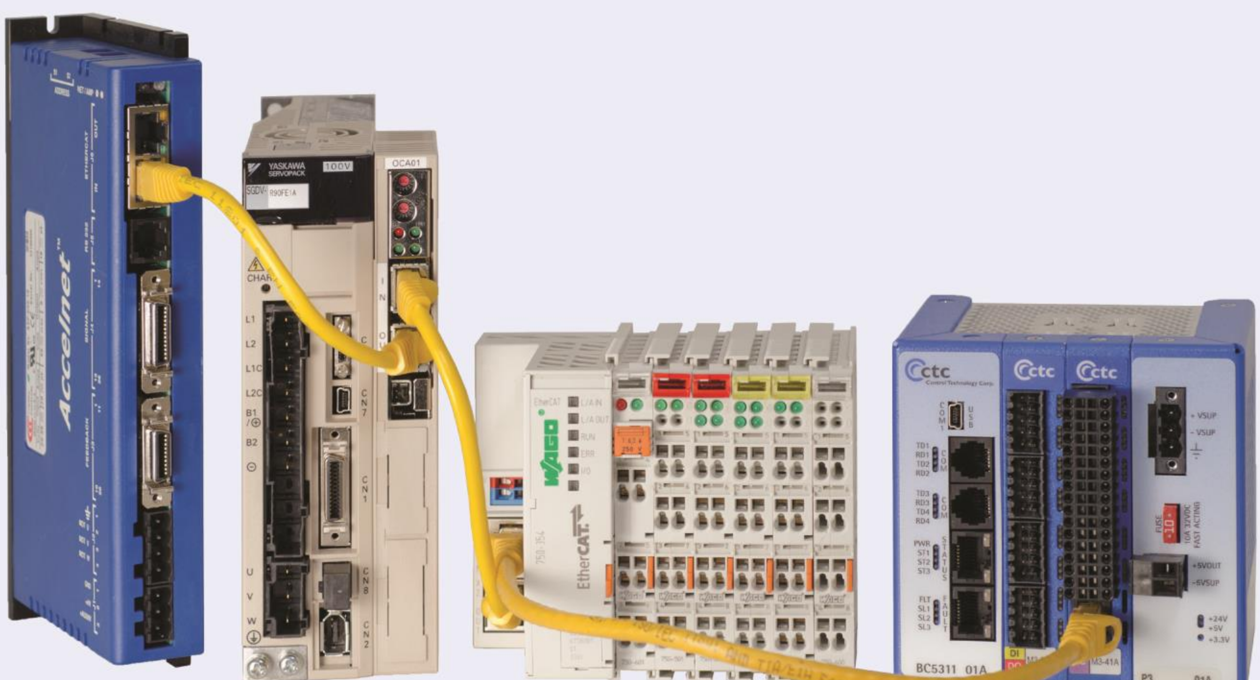
Control Technology Corp.

EtherCAT Applications Guide



Control Technology Corporation
Hopkinton, MA • ctc-control.com

CONTROL TECHNOLOGY CORPORATION

Incentive EtherCAT Applications Guide

EtherCAT Applications Guide

December 15, 2022

Document 951-534101-074

Blank

WARNING: Use of CTC Controllers and software is to be done only by experienced and qualified personnel who are responsible for the application and use of control equipment like the CTC controllers. These individuals must satisfy themselves that all necessary steps have been taken to assure that each application and use meets all performance and safety requirements, including any applicable laws, regulations, codes and/or standards. The information in this document is given as a general guide and all examples are for illustrative purposes only and are not intended for use in the actual application of CTC product. CTC products are not designed, sold, or marketed for use in any particular application or installation; this responsibility resides solely with the user. CTC does not assume any responsibility or liability, intellectual or otherwise for the use of CTC products.

The information in this document is current as of the following Hardware and Firmware revision levels. Some features may not be supported in earlier revisions. See www.ctc-control.com for the availability of firmware updates or contact CTC Technical Support.

Model Number	Hardware Revision	Firmware Revision
5300/Incentive M3-41/Incentive (EtherCAT Master)	All Revisions	>= V050090R70.33 >= V1.113

Blank

TABLE OF CONTENTS

Blank

CHAPTER 1

[1] Overview

This manual is intended to be used in conjunction with the *Model 5300 QuickMotion Reference Guide* and an understanding of that manual is assumed. The model 5300 M3-40 motion module supports local drives and I/O; this manual discusses CTC motion control as it pertains to the 5300 M3-41 EtherCAT hardware module as well as the PC based Incentive software package. Both products offer an advanced EtherCAT Master providing distributed motion control, using the CAN application protocol over EtherCAT, supporting such devices as servo drives, RFID readers, and I/O control. Although the M3-41 is a PLC hardware based solution, it shares the same code base as the advanced PC based Incentive software solution.

Unlike EtherCAT Masters from other vendors, Incentive attempts to isolate the user from the complexity of the EtherCAT environment by automatically scanning the network and configuring supported devices. The programming interface uses the same high-level language that the previous 5300 M3-40 series of modules uses: QuickBuilder MSBs (Motion Sequence Blocks). You no longer have to deal with a complex configurator, poking drive objects, or figuring out how an interface works. Each of the supported motion and/or I/O devices has been verified, and all setup and initialization is done for you. This greatly simplifies an EtherCAT installation, enabling you to concentrate on motion control and your system, not a complicated configurator. Multiple EtherCAT Master Network modules (M3-41 and PC based) can be intermixed with other networks such as Modbus®, as well as other modules offered within the embedded 5300 controller family.

The M3-41 EtherCAT Master is available both as a hardware device, module within the 5300 Controller, or as a soft device, executing in real-time on a Windows® based platform (IncentivePLC/IncentiveECAT). In both environments the same application programming environment is used, QuickBuilder. Additionally, a .Net API (IncentiveAPI), exposes the entire MSB language is available on a Windows® platform. The biggest difference between the two is the PC environment is limited to EtherCAT only IO while the embedded 5300 controller has numerous local IO and network possibilities, in addition to EtherCAT. Execution on a Windows based PC presents the developer with an open and diverse architecture with which to implement their automation solution. Below is a quick comparison of the two environments:

Windows 11 and/or secure boot are not currently supported by the Incentive PC runtime.

Feature	*Embedded 5300 PLC*	*Incentive PC, Dual Core Processor*	*Incentive PC, Quad Core Processor*
Executes QuickBuilder	Yes	Yes	Yes
Programmable using 'C' steps	Yes	Yes	Yes
Modbus, CTC Binary, UDP, TCP	Yes	Yes	Yes
Serial Port Support	Yes	Yes	Yes
QuickBuilder Performance	1 X	1X to 4 X	2.3 to 8 X+
Local IO Expansion modules	Yes 2/4/8	No	No
Axes/EtherCAT Network	16	8 (Atom) 16 (i7)	64 (I210) 32 (other)
Max EtherCAT Networks	4	1	2
EtherCAT Segmentation	No	Yes	Yes
Fast Packet Retry on Loss	Yes	No	No
RFID Channels/Network	16	8 (Atom) 16 (i7)	32
Scan Rates	500us-4ms	500us-4ms	500us-4ms
Redundant Master	Secondary only	Secondary/Ring	Secondary/Ring
BACnet Support	Yes	TBD	TBD
Digital IN/Digital OUT	1024	1024	1024
Analog IN/Analog OUT	256	256	256

Many factors affect the maximum number of axis/network with the Incentive PC. The usage of an Intel I210 network adapter also significantly boosts performance. This adapter is able to periodically transmit packets via hardware whereas others (Realtek) place the overhead on the software, thus limiting performance.

The following motion control devices are currently supported, with more available in the future:

	Cyclic Sync Position	*Interpolated*	*Homing*	*Profile Position*	*Profile Velocity*	*Registration*
ABB MicroFlex e150	✓	Not supported by manuf	Not supported by manuf	Not supported by manuf	Not supported by manuf	✓ Probe 1 & 2
ADVANCED Motion Controls (Digiflex DPE/STF)	✓	Not supported by manuf.	✓			
Applied Motio Products (SS-EC &	✓		✓			

	Cyclic Sync Position	Interpolated	Homing	Profile Position	Profile Velocity	Registration
STF)[8]						
Delta Electronics ASDA-A2	✓		✓	✓		
Copley Accelnet (AEP-055-18) and Stepnet	✓		✓			
Copley Xenus Plus 1 & 2 Axis (XEL-230-40 XE2-230-20, 800-1849)	✓		✓			
Emerson[2] Digitax ST & Unidrive SP (single axis)	✓	Not supported	✓	Not supported by manuf.	Not supported by manuf.	
Festo EMCA-EC-67 Drive[9]	✓				.	
IAI ACON Controller[3]	Not supported by manuf.	Not supported by manuf.	✓	✓	Not supported by manuf.	
Kollmorgen AKD[4]	✓	Not supported	✓	Not supported	✓	✓ Probe 1
LINMOT C1250 & E1450[5]	✓	Not supported by manuf.	Not supported by manuf	Not supported by manuf	Not supported by manuf.	
Parker Hannifin P Series	✓		✓		.	
Maxon EPOS4	✓		✓	✓		

	Cyclic Sync Position	Interpolated	Homing	Profile Position	Profile Velocity	Registration
PULSEROLLER MOTIONLINX-AI	Not supported by manuf	Not supported by manuf.	Not supported by manuf	Not supported by manuf	$VELOCITY_ MODE Only	
Mitsubishi J4/J5[6]	✓	Not supported by manuf.	✓	Not supported by manuf	Not supported by manuf	✓ Probe 1 & 2 (J4 Only)
Mitsubishi Fr-E800 series VFD drives					✓	
Sanyo Denki SANMotion RS2E	✓	Not supported by manuf.	✓	✓	✓	✓ Probe 1 & 2
Yaskawa Sigma 5 & 7(Rotary and Linear)	✓	✓	✓	✓	✓	✓ Probe 1 & 2
Virtual Axis[7]	✓					

[2]Emerson must have DC Sync enabled prior to 'drive enable' for all operations.
[3]IAI ACON Controllers do not support Profile Position mode but use a proprietary setting similar to it with simulation provided by the M3-41 module. Only Full Direct Value Mode is supported (3).
[4]Support for Kollmorgen Interpolated Position mode was removed to allow PDO space for registration. Cyclic Sync Position mode is the preferred mode.
[5]Due to timing issues with LinMOT drives a cycle time of 2 mS or slower must be used or a PVT overflow error may occur within the drive.
[6]Probe 1 & 2 Registration has been implemented but not fully tested for J4 drive only. Mitsubishi cannot run any slower than 2 ms scan rate (manufacturer limitation).
[7]A Virtual Axis does not impact the number of online drives licensed although does count towards the limitation of 16 axis/M3-41 module. Wago Incremental Encoder modules automatically add one Virtual Axis per encoder.
[8]Only 1 ms scan rate has been implemented. This is a stepper motor, not a servo. Torque control during CSP not supported by manufacturer.
[9]Festo only supported in CSP mode with free run, no DC Sync. Firmware issues were discovered which limit operations of other modes. DC Sync is disabled by our software and homing can be attempted but existing firmware is not functional thus it is untested should new firmware become available.

Firmware Versions:

- Kollmorgen V1.8.0.3 (must be this or greater to support Cyclic Sync Position mode)
- Emerson SM-EtherCAT V1.07.01, Unidrive SP02X1 V1.15, Digitax DST1201 V1.06
- Yaskawa V3.05
- ABB MicroFlex e150 build 5711.4.0
- Applied Motion Products SS-EC V2.1, STF V1
- LinMOT – See appendix.
- PULSEROLLER MOTIONLINX-AI V1.4, Hardware V1.0
- Copley drives must have V3.06 or greater firmware
- Festo V1.4.0.9
- Parker Hannifin PD-xxC drives V1.02 firmware
- Delta Electronics ASDA-A2 V1.651.408
- Maxon EPOS4 Firmware 0x161.

Supported Gateway Devices:

HMS Anybus X Gateway EtherCAT Slave (Incentive side)

- Profibus
- Modbus-TCP
- Other Anybus X Gateway devices may work depending upon the Product ID.

Supported I/O Devices:

Beckhoff (8 bit digital IO boundaries only)

- EK1100 – EtherCAT Coupler
- Digital Inputs – EL1018
- Digital Outputs – EL2008
- Analog Inputs – EL3102, EL3112, EL 3122, EL3142, EL3152, EL3162
- Analog Outputs – EL4032, EL4102, EL4112, EL4122, EL4132

Omron

- GX-JC06 – EtherCAT Junction Slave, 6 port.
- Pending: NX-ECC201, NX-ECC202, and NX-ECC203 EtherCAT Couplers with following NX- modules:
    - ID4342 DIN-8 NPN 24VDC
    - ID4442 DIN-8 PNP 24VDC
    - ID5142-1 DIN-16 N/PNP 24VDC
    - ID5142-5 DIN-16 N/PNP 24VDC
    - ID5342 DIN-16 NPN 24VDC
    - ID5442 DIN-16 PNP 24VDC
    - ID6142-5 DIN-32 N/PNP 24VDC
    - ID6142-6 DIN-32 N/PNP 24VDC
    - OD4121 DOUT-8 TNPN 12-24VDC
    - OD4256 DOUT-8 TPNP 12-24VDC
    - OD5121-1 DOUT-16 TNPN 12-24DC
    - OD5121-5 DOUT-16 TNPN 12-24DC
    - OD5121 DOUT-16 TNPN 12-24VDC

  - OD5256-1 DOUT-16 TPNP 24VDC
  - OD5256-5 DOUT-16 TPNP 24VDC
  - OD5256 DOUT-16 TPNP 24VDC
  - OD6121 DOUT-32 TNPN 12-24DC
  - OD6121-6 DOUT-32 TNPN 12-24DC
  - OD6256-5 DOUT-32 TPNP 24VDC
  - TC2406 2 channel Temperature Control
  - TC3406 4 channel Temperature Control
  - PC0010 IOG 16 term

  Free-run mode only.

Koganei YS7K1/YS7K2 F10/15 Series Solenoid Valves

- YS7K1 16 output
- YS7K3 32 output

Numatics Incorporated (Emerson) 501/G3 Series ***(1.1 Build 42194, Boot 1.1 Build 41544)***

- Diagnostic inputs must be turned off.
- 240-310 EtherCAT Module
- 219-828 Valve Driver Output Module
- 425186-001 Atlas Valve Driver Output Module (P599AE42518800x)
- 240-203 16DI PNP Terminal Strip
- 240-204 16DI NPN Terminal Strip
- 240-205 16DI PNP M12 x 8
- 240-206 8DI PNP M12 x 8
- 240-207 16DO PNP M12 x 8
- 240-208 8DO PNP M12 x 8
- 240-209 16DI NPN M12 x 8
- 240-210 8DI NPN M12 x 8
- 240-211 8DI/8DO PNP M12 x 8
- 240-300 8DO High Current PNP M12 x 4
- 240-316 8DI PNP Terminal Strip
- 240-323 16DI PNP 19 Pin
- 240-330 16DO PNP Terminal Strip

Numatics Incorporated (Emerson) 580 ***(Firmware 1.1 Build 42656)***

- 32 outputs, no inputs on this device.
- Firmware version is critical since older firmware will not work properly.
- Connects to Numatics 501, 502, and 503 series valves.
- Diagnostic inputs must be turned off.
- Lower port is EtherCAT IN, not upper. Documentation is incorrect.

SMC Corporation EX600 (8 bit digital IO boundaries only, ***not first slave due to clock issues****)*

- Digital Inputs – EX600-DX*B, C, C1, D, E, F modules
- Digital Outputs – EX600-DY*B, E, F modules

- Digital Input/Outputs – EX600-DM*E, F modules
- Analog Inputs – EX600-AXA module
- Analog Outputs – EX600-AYA module
- Analog Input/Outputs – EX600-AMB module
- Valves – EX600-SEC*, 8, 16, and 24 valves

Wago 750-354 EtherCAT Coupler

- Digital Inputs – 750-4XX modules (non-8 bit boundaries supported)
- Digital Outputs – 750-5XX modules (non-8 bit boundaries supported)
- Analog Inputs – 750-452, 453, 454, 455, 456, 457, 459, 460, 461, 462, 464, 465, 466, 467, 468, 469, 470, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 483, 485, 486, 487, 491, 492, 496, 497, 630
- Analog Outputs – 750-550, 551, 552, 553, 554, 555, 556, 557, 559, 560, 562, 563, 565
- Incremental Encoder – 750-631 and 637 (each channel appears as an incremental encoder on axis).
- SSI Transmitter – 750-630/000-008 25 bits/125khz gray code (others may work but are not tested, each appears as an axis, set axis property encoder_mode = 1 or the axis encoder properties to absolute to power up with present position, execute ‘drive enable’).

Turck BL20-E-GW-EC EtherCAT Coupler

- Digital Inputs (non-8 bit boundaries supported)
    - BL20-2DI-24VDC-P (2 digital inputs, 24VDC PNP switching)
    - BL20-2DI-24VDC-N (2 digital inputs, 24VDC NPN switching)
    - BL20-2DI-120/230VAC (2 digital inputs, 120/230VAC)
    - BL20-4DI-24VDC-P (4 digital inputs, 24VDC PNP switching)
    - BL20-4DI-24VDC-N (4 digital inputs, 24VDC NPN switching)
    - BL-20-E-8DI-24VDC-P (8 digital inputs, 24VDC PNP switching)
    - BL-20-E-16DI-24VDC-P (16 digital inputs, 24VDC PNP switching)
    - BL-20-16DI-24VDC-P P (16 digital inputs, 24VDC PNP switching, optocouplers)
    - BL-20-32DI-24VDC-P (32 digital inputs, 24VDC PNP switching, optocouplers)
- Digital Outputs (non-8 bit boundaries supported)
    - BL20-2DO-24VDC-0.5A-P (2 digital outputs, 24VDC PNP switching, 0.5 Amp)
    - BL20-2DO-24VDC-2A-P (2 digital outputs, 24VDC PNP switching, 2 Amp)
    - BL20-2DO-120/230VAC-0.5A (2 digital outputs, 120/230VAC)
    - BL20-2DO-R-NO (2 relay outputs, normally open)
    - BL20-2DO-R-NC (2 relay outputs, normally closed)
    - BL20-2DO-R-CO (2 relay outputs)
    - BL20-4DO-24VDC-0.5A-N (4 digital outputs, 24VDC NPN switching)
    - BL20-4DO-24VDC-0.5A-P (4 digital outputs, 24VDC PNP switching)
    - BL-20-E-8DO-24VDC-0.5A-P (8 digital outputs, 24VDC PNP switching)
    - BL-20-E-16DO-24VDC-0.5A-P (16 digital outputs, 24VDC PNP switching)

- BL-20-16DO-24VDC-0.5A-P (16 digital outputs, 24VDC PNP switching, optocouplers)
- BL-20-32DO-24VDC-0.5A-P (32 digital outputs, 24VDC PNP switching, optocouplers)

- Analog Inputs
  - BL20-E-8AI-U/I-4PT/NI (8 analog inputs, U/I configurable, -10/0..+10VDC & 0/4..20MA)
  - BL20-2AI-U (2 analog inputs, U/I configurable, -10/0..+10VDC & 0/4..20MA)
  - BL20-2AI-I (2 analog inputs, 0/4..20MA)
  - BL20-1AI-I (1 analog input, 0/4..20MA)
  - BL20-1AI-U (1 analog input, -10/0..+10VDC)
  - BL20-2AI-PT/NI-2/3 Temp (2 analog inputs for temperature measurement)
  - BL20-4AI-U/I (4 analog inputs, U/I configurable)
  - BL20-2AI-THERMO-PI (2 analog inputs for thermocouples)
- Analog Outputs
  - BL20-E-4AO-U/I (4 analog outputs, configurable, -10/0..+10VDC & 0/4..20MA)
  - BL20-1AO-I (1 analog output, 0/4..20MA)
  - BL20-2AO-I (2 analog outputs, 0/4..20MA)
  - BL20-2AO-I (2 analog outputs, -10/0..+10VDC)
- RFID Reader
  - BL20-2RFID-S (2 channel RFID reader)

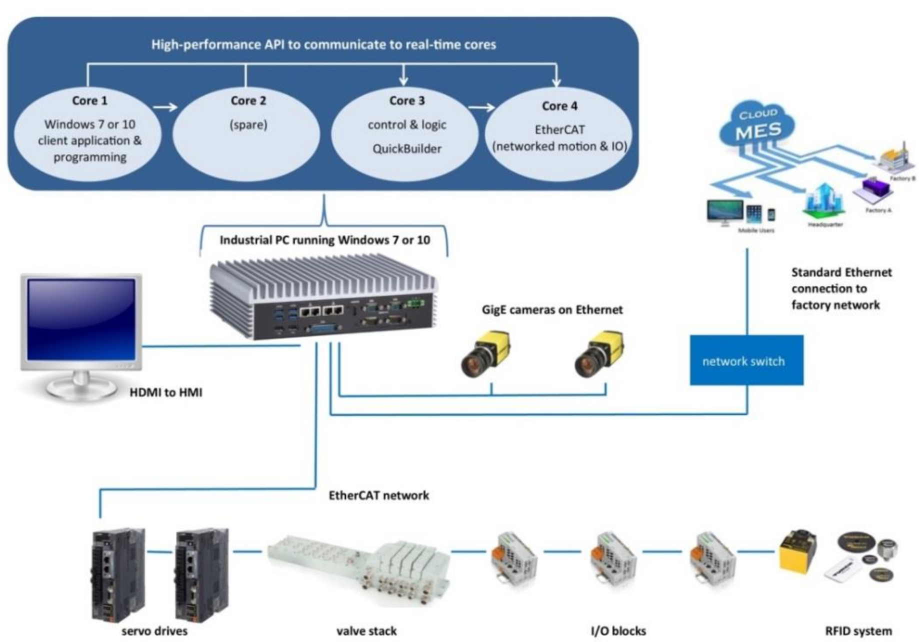
Only digital input and digital output modules evenly divisible by 8 are supported on all slaves except Wago and Turck; this keeps the EtherCAT packet aligned for faster operation. Usage of non 8 bit divisible digital IO on Wago and Turck will affect performance due to the bit shifting required during each control loop. This typically reduces the total axes supported by 1, depending on the number of IO points. If hundreds of mis-aligned IO are used the performance impact will be greater and will require specific application testing. <u>Additionally SMC Corp devices must not be the first slave device, they do not support ARMW EtherCAT packets required for clock synchronization.</u>

CHAPTER 2

[2] Incentive PC Runtime

Those using the 5300 PLC Controller may skip to the next chapter, this Chapter details the embedded PC platform. The 5300 Controller QuickBuilder run-time has been ported to the Windows® PC where all your EtherCAT programs can be used on both environments with just the selection of a translation combo box within QuickBuilder. A virtual, soft 5300 PLC (IncentivePLC) can now execute on multiple platforms allowing for versatility in your automation decisions. In addition to QuickBuilder a complete Incentive .Net Managed API (IncentiveAPI) is available allowing for programming of all functions directly from languages such as C#, VB.Net, and C++ using Visual Studio®. IncentiveAPI works both on the local computer and transparently over a network. Support is even included for direct interaction with Amazon AWS IoT cloud services. Complex, highly integrated solutions can now be integrated into a single platform, programmed in your language of choice.

Incentive… for PC-based Programmable Automation



Incentive Runtime

CTC Incentive offers system flexibility not only in its open vendor support of numerous EtherCAT drives and IO selections but its tightly integrated programming offerings, fully supporting Microsoft Visual Studio®, and transforming Windows® into a real-time automation environment. Standard Windows® based programs such as HMI's, vision systems, and custom applications run transparently, in parallel to the Incentive real-time EtherCAT Master.

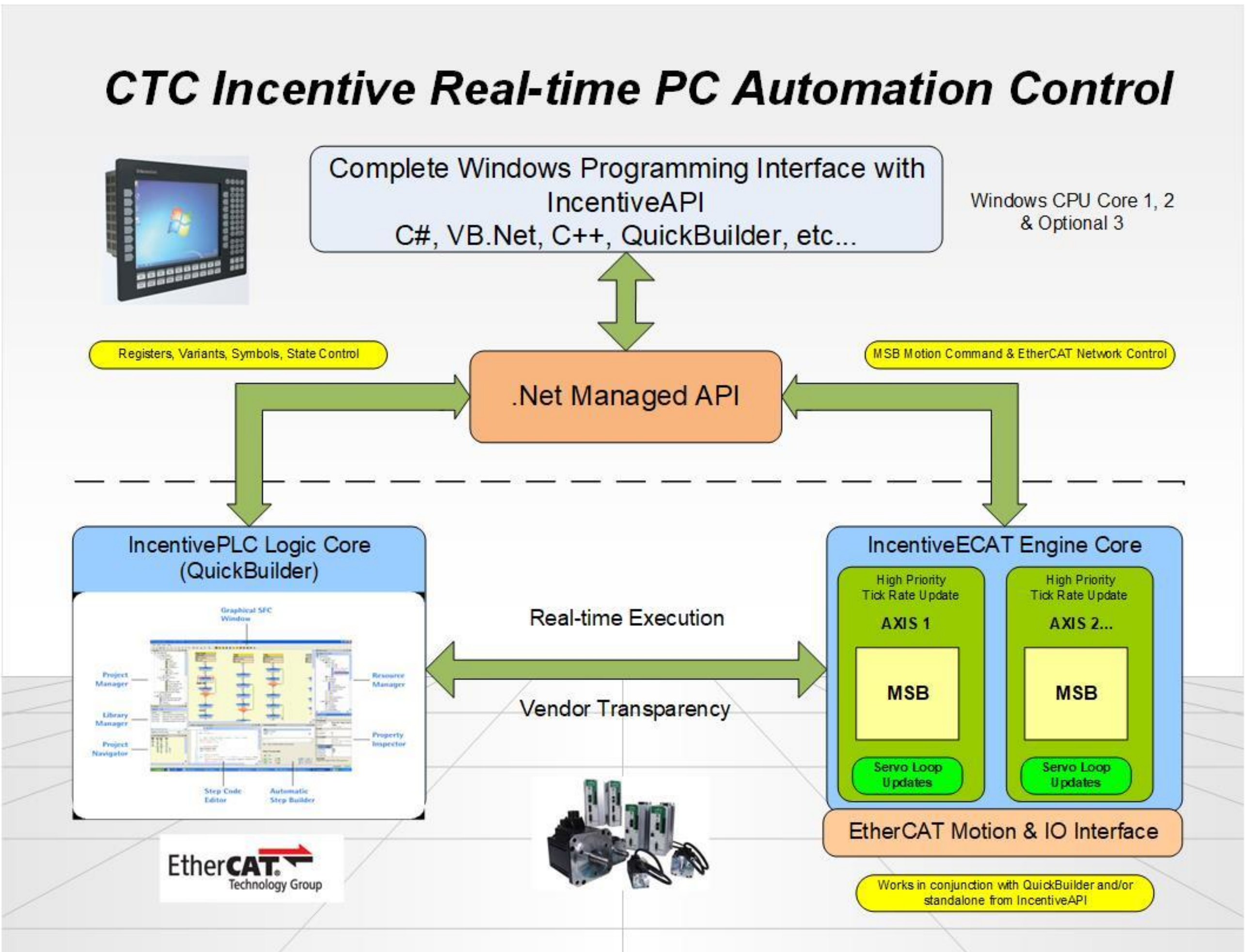
The Incentive Runtime can run on systems with:

- Windows® 7 64 bit or greater, 32 bit is not supported. Windows 10 Pro preferred, **11 not supported nor is secure boot**.
- Dual or Quad core processors such as J1900, E3845, i5, i7. Quad core is recommended although dual core processors may be used in smaller, cost sensitive environments with reduced performance, 8 cores may also be used where available. 4 G memory minimum, 8G-16G recommended for efficient Windows® operation.
- Some processors may have jitter issues and need to be evaluated by CTC. Currently Intel processor families such as Bay Trail, Haswell, and Broadwell work fine. Skylake and Kaby Lake work fine with the latest Incentive release but do require an i210 network adapter for EtherCAT and are limited to 1 mS cycle time or slower since these processors have memory cache problems that can cause 300uS timing problems every once in a while (hours apart). The timing is not a problem with the i210 since it uses its own timer to periodically transmit EtherCAT packets. All Skylake and Kaby Lake systems should be tested prior to deployment as there are some where jitter may be unacceptable. CTC has validated a few systems.
- Solid state drives must be used to ensure data integrity of registers during shutdown as well as support the fast access time required by the real time environment. The Samsung EVO and PRO SSD drives are recommended, 128G or greater.
- BIOS with limited System Management Interrupt (SMI) usage. SMI's can cause excessive jitter when not using an I210 Ethernet Adapter and reduce performance. They can prevent the real-time operating system from gaining control for several hundred microseconds. Refer to the section within this chapter entitled "System Management Interrupt Detection" for additional information.
- Intel I210 Ethernet Adapter preferred (Realtek 8111E PCIe Gbe, i211 and i219 family adapters with reduced performance). Realtek 8168E did not pass EtherCAT performance testing. Other adapters are possible but subject to testing.
- Onboard Graphics controllers such as Intel HD typically offer better timing stability than Nvidia or AMD but when an i210 network adapter is used timing is not a problem.
- The BIOS of the PC must have controls to disable such things as SpeedStep, Hyper-Threading and C-States. Failure to do so can cause excessive jitter and possible EtherCAT errors. In many cases it is less expensive, with similar performance, to use an i5 rather than an i7 since i5 is essentially an i7 with no Hyper-Threading and a smaller cache. Secure boot must also be disabled.
- A small UPS with Windows UPS shutdown utilities is suggested for register preservation upon power loss. Both CyberPower and APC have been tested to work well.

To optimize performance the Windows® processor can run different operating systems on each of its cores. With CTC Incentive, in a quad core system, one core runs the QuickBuilder PLC Logic (IncentivePLC); a second core runs a virtual EtherCAT Master (IncentiveECAT), all communicating through high speed shared memory. Both of these cores run in real-time using an operating system provided by TenAsys, called

INtime. This leaves the remaining two cores to run any normal Windows® application. A second EtherCAT Master can also be run for larger systems, where two networks are desired, using 3 cores in real-time and leaving one core for Windows®. For even greater flexibility the EtherCAT Master can be run standalone programmed solely by a Windows Application (C#, VB.Net, C++., etc…), using a single core, leaving 3 cores for greater Windows performance.

In a dual core environment, both the QuickBuilder application and EtherCAT Master are loaded as separate processes and execute on the same core. This should only be used in smaller systems where performance is not as important given the QuickBuilder application will affect the EtherCAT performance, and vice versa. Quad core allows each to run at maximum speed, independently, neither affecting the other. Where Windows® performance is more important the dual core environment can also be used on a quad core processor. In this case Windows® would run 3 cores for itself and one for IncentivePLC/IncentiveECAT.



A significant improvement with the Incentive PC Runtime is that of performance. For raw QuickBuilder execution the PC can be tailored to improve performance, especially in EtherCAT IO update rates. A simple itineration test provided some comparison numbers showing a quad core J1900 Celeron 2GHZ 8GB ram, at 2.3 X that of a 5300 and an i7-3770 3.40GHZ 16GB ram, 5.3 X. Much of the IO speed improvement is from the tightly coupled independent cores, each running their own environments, communicating through shared memory. One dedicated to QuickBuilder execution and the other to EtherCAT IO and motion control.

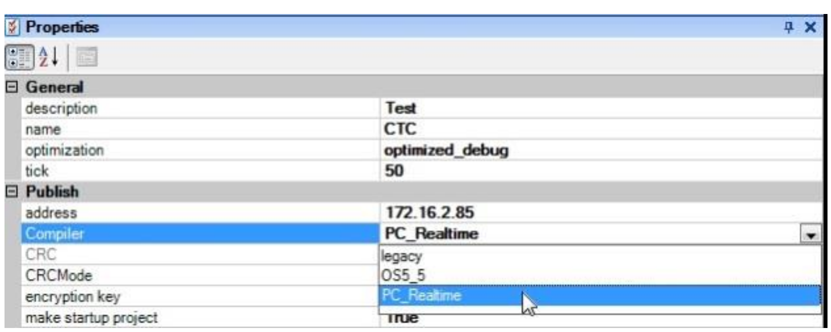
With the increased performance of the Incentive Runtime it also now supports up to 64 axes per network when the I210 network controller is used at a 1 ms scan rate as compared to that of 16 per M3-41 hardware version. For large networks segmentation is also supported, where data can be chained in multiple packets to allow for a greater number of IO and axes. With this also comes the importance of good cabling practices as, unlike the M3-41, the PC cannot do timely packet re-transmission and still maintain DC SYNC. A final improvement is the support of twice the number of RFID channels, 32, versus 16 of the M3-41. In summary:

Feature	***5300 M3-41 Hardware***	***IncentiveECAT Dual Core***	***IncentiveECAT Quad Core***
Max Coordinated Axes/Network	16	8 (Atom) 16 (i7)	64 (I210) 32 (other)
Max Networks	4	1	2
QuickBuilder Performance	1 X	1X to 4 X	2.3 to 8 X+
EtherCAT Segmentation	No	Yes	Yes
Fast Packet Retry on Loss	Yes	No	No
Turck RFID Channels	16	8 (Atom) 16 (i7)	32
Scan Rates	500us-4ms	1ms-4ms	500us-4ms

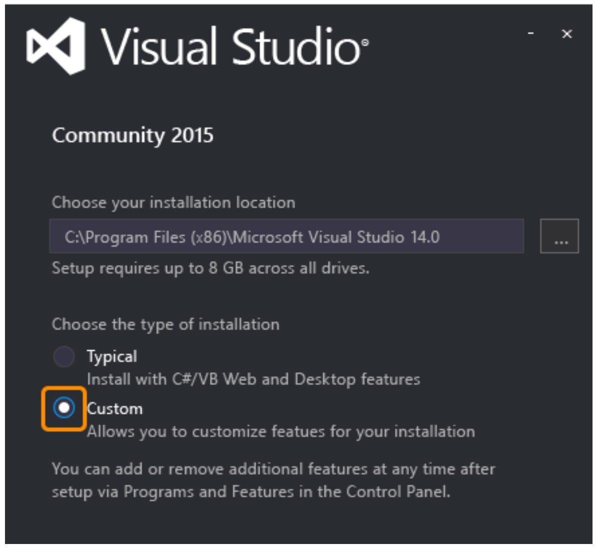
Currently 500uS, 1ms and 2ms scan rates work on all devices. 1ms is the recommended default. 4ms does not work on Mitsubishi. Remember you must set the DC SYNC SYNC0 timing to match the scan rate or an error from the drive may occur. When using a non-I210 Ethernet Adapter it may best to offset the SYNC0 by up to 500uS (dcsync -1, 1000000, 0, 500000, 100000000).

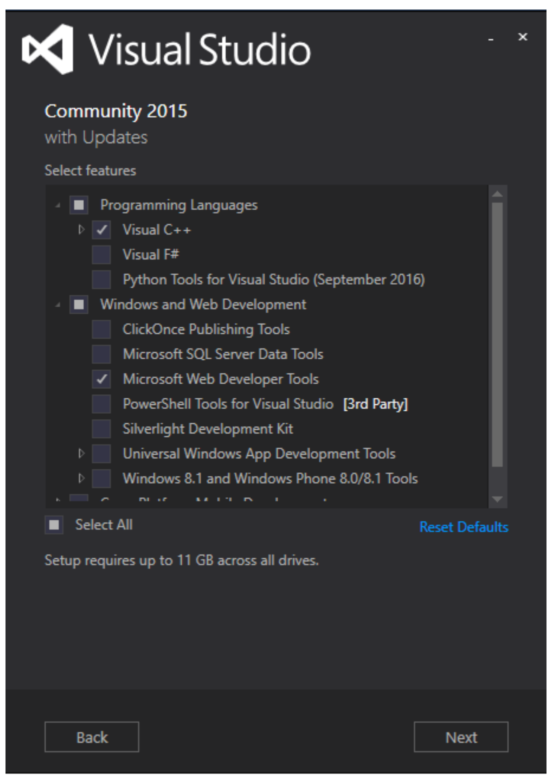
QuickBuilder Programming for Windows®

QuickBuilder generates 'C' code which can execute on multiple environments. Programming for the 5300 Controller is identical to that of the PC, you simply have to select 'PC_Runtime' from the Controller Compiler property and click Translate.



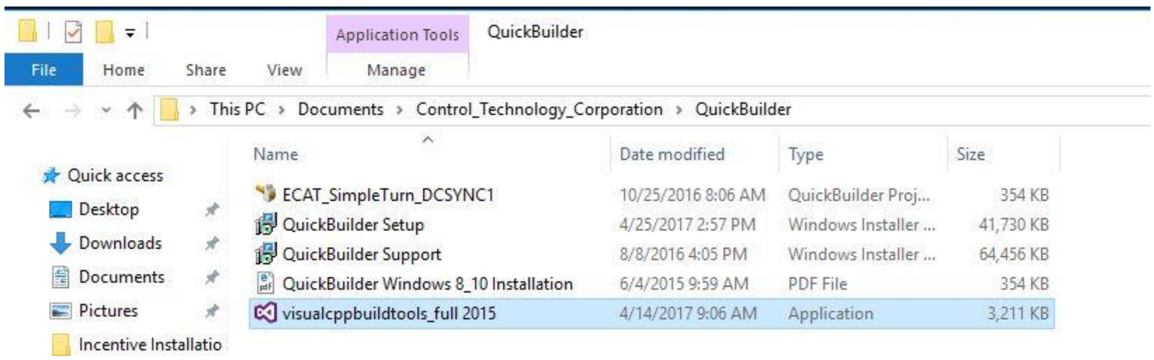
This assumes that you have previously installed a copy of Microsoft's Visual Studio® Express for Windows Desktop, or Community Edition, and the latest QuickBuilder Support libraries. Any Visual Studio® version greater than Visual Studio® 2012 can be used, as long as it has Desktop support and C++, this includes the free 2015 Community version (https://www.visualstudio.com/products/visual-studio-community-vs). Just make sure to do the custom install if installing 2015 or greater since Microsoft stopped installing C++ automatically beginning with that revision. The C++ compiler is required for QuickBuilder.





When 'Select features' appears expand 'Programming Languages' and make sure to select C++ as well as your desired languages if using the Incentive managed API.

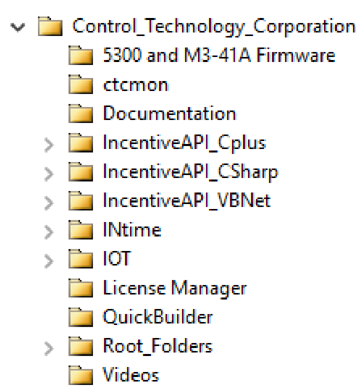
If there is no need for Visual Studio and only the C++ compiler for QuickBuilder is needed the Incentive Installation kit provides a program which will allow you to install only the needed tools, “visualcppbuildtools_full 2015.exe”. This is located in the “Control_Technology_Corporation\QuickBuilder” directory:



If you are installing Visual Studio 2017, or greater, make sure to first install “visualcppbuildtools_full_2015.exe”, located in the “Control_Technology_Corporation\QuickBuilder” installation directory. Visual Studio 2017 significantly changed the C++ compiler tools and once installed you are not allowed to install an older version, but if 2015 is installed first they will work together fine.

Incentive Installation Overview

A single installation file, Incentive_Setup.exe, is available which provides the Incentive runtime, API, documentation, QuickBuilder, CTCMon, test projects, and installation video. Incentive_Setup.exe installs and registers the CTC_Incentive.dll, managed .Net dll, as well as provides additional folders within which are other programs to be installed. Upon installation the directory tree will look something like:



5300 and M3-41A Firmware – Embedded 5300 PLC and M3-41A EtherCAT Option module latest firmware at the time of this release.

ctcmon – Contains an installation program called *mon##setup.exe* (where ## is the version). This program will install a communications dll called ctccom32.dll as well as CTCmon. The dll may be used by any

Windows program to interact with a controller on a network or serial port. The utility is used to monitor registers, program status, and IO via TCP, UDP, and serial ports.

Documentation – This manual in PDF format as well as the Incentive API chm help file.
IncentiveAPI_Cplus – A Visual Studio 2013 sample project using C++ with the Incentive DLL.

IncentiveAPI_CSharp - A Visual Studio 2013 sample project using C# with the Incentive DLL.

IncentiveAPI_VBNet - A Visual Studio 2013 sample project using VB.Net with the Incentive DLL.

INtime – Contains the installation programs for both a host only (remote PC with no resident runtime), *host61-17004_installer.exe* (where the numerics are the revision level), as well as the realtime runtime environment required by the EtherCAT Master, *runtime61-17004_installer.exe*. By default the runtime will install as a demo for 60 days. No license is required for the host environment. Patch files are also included and are updated as problems are identified and resolved. Reference the readme in the INtime_Patches sub-directory for current patch requirements. An upgrade is also available, *host63-19040-2.exe and runtime63-19040-2.exe, or newer.* The older version is best to use initially since it allows a 60 day demo period. You may immediately upgrade to 6.3 and the demo license will remain active. Make sure to install all patches.

IOT – Contains the Internet of Things DLL's and utilities for mqtt and amqp protocols. Reference the IOT sections of this manual for further discussion.

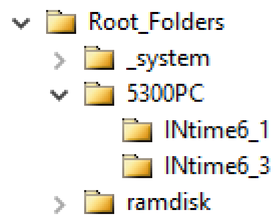
License Manager – Contains the installation programs for both a host only (remote PC with no resident runtime),

- *Incentive License Setup.msi* installation program which contains a licensing utility used for newer installations of both the Incentive real-time environment as well as QuickBuilder. A license monitor service is also included, which provides an interface between Windows and the real-time EtherCAT Master runtimes for license verification and available options. Reference the section dedicated to Licensing for further details.

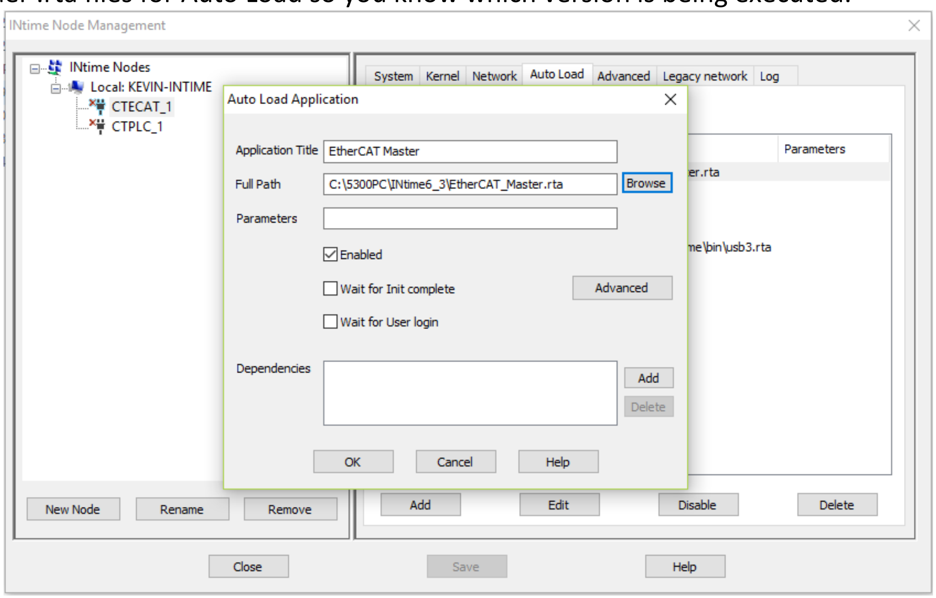
QuickBuilder –

- *QuickBuilder_Support.msi* installation program which contains the gcc compiler tools required for translating application programs for the 5300 embedded PLC. Note that Visual Studio 2015 Community or later is required to translate programs for execution on the PC realtime environment (available directly from Microsoft).
- *QuickBuilder_Setup.msi* installs the Quickbuilder development system which by default will install as a demo for 30 days.
- *ECAT_SimpleTurn_DCSYNC1.qbp* is a simple EtherCAT application program to control a single servo motor.
- *QuickBuilder_Windows_8_10_Installation.pdf* contains special instruction required for installation of QuickBuilder on that platform.
- *Visualcppbuildtools_full_2015.exe* is a utility provided by Microsoft to download only the Visual Studio 2015 C++ compiler and libraries, not full Visual Studio. This is all that is needed by QuickBuilder for execution of applications on a real-time PC. If Visual Studio 2017 or newer is going to be installed this program needs to be installed first.

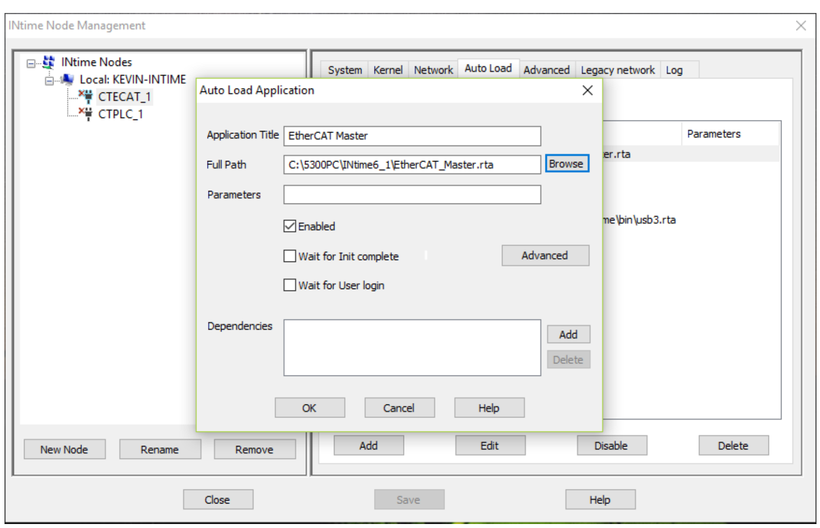
Root_Folders – Contain three subfolders, *_system*, *5300PC*, and *ramdisk*, all of which should be copied to the root directory of the C: drive. The *_system* and *ramdisk* folder contents replicate that of the 5300 embedded PLC while the *5300PC* subdirectory contains the executable files for both the PLC Logic and EtherCAT Master realtime Windows processor cores.



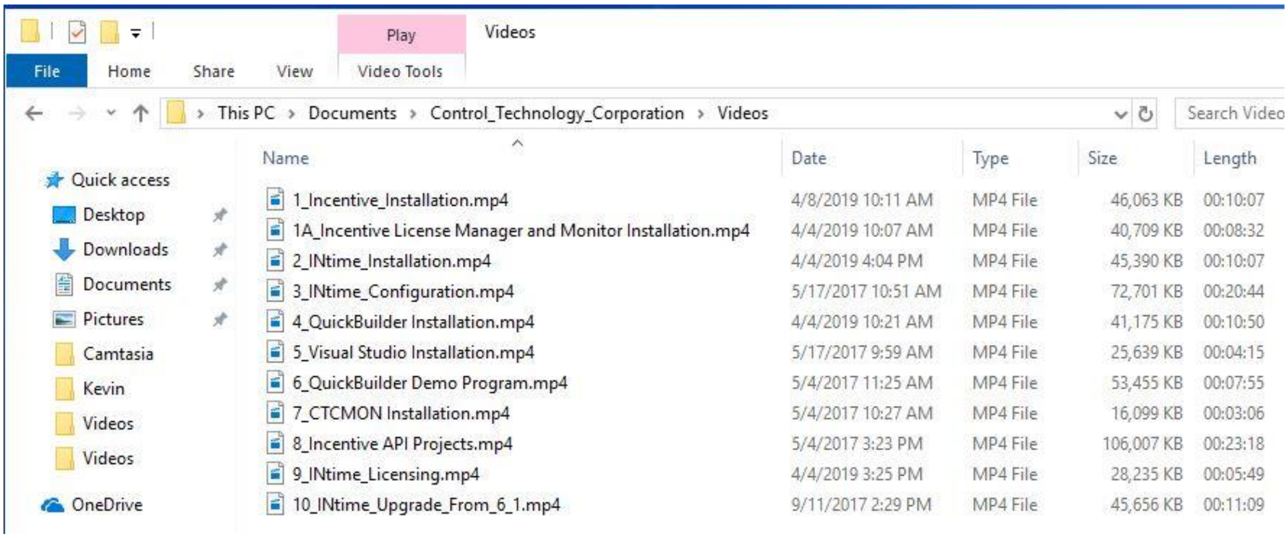
By default the TenAsys INtime 6.1 runtime is in the 5300PC folder and will be executed. This is mainly for demo installations that use V6.1.17004 and legacy installations. When using INtime 6.3 you should copy the .rta files from INtime6_3 to 5300PC or better, setup the INtime configuration to directly point to the INtime6_3 folder .rta files for Auto Load so you know which version is being executed.



If INtime 6.1:



Videos – This folder contains installation videos to help you with the install, configuration, and programming of Incentive. They are numbered from 1 to 9 in the recommended viewing order:



The recommended sequence for installation is as follows:

First watch the “1_Incentive_Installation.mp4” video and run Incentive_Setup as directed, then “1A_Incentive License Manager and Monitor Installation.mp4” to ensure you have obtained the proper licensing required. QuickBuilder will need a demo license whereas the Incentive real-time product will run for 3 hours as a full product before shutting down if in demo mode. Once complete then proceed as below:

Application Development Only or Remote PC with no EtherCAT Master:

1. Perform all Windows updates prior to installation.
2. Install ‘Incentive License Setup.msi’ in the Licensing sub-folder. This is required for QuickBuilder demos but not Incentive real-time demos. All fully licensed products will require it in order to authenticate licenses. Reference the section on Incentive Licensing. Note that upgrades of older installations do not need the new licensing tools and your existing licenses will be used.
3. Install CTCmon – mon##setup.exe, optional for debug, in the ctcmon sub-folder.
4. Install QuickBuilder_Support.msi, required for QuickBuilder programming, from the QuickBuilder sub-folder.
5. Install QuickBuilder_Setup.msi, required for QuickBuilder programming, from the QuickBuilder sub-folder.
6. Download and install Visual Studio 2015 or greater Community edition from Microsoft. (https://www.visualstudio.com/downloads/) or install just the C++ tools using the file provided in the QuickBuilder Incentive installation directory (visualcppbuildtools_full 2015.exe). This is needed in order for QuickBuilder to compile your application program for the Windows PC target system.
7. Install host61-17004_installer.exe or host63-19040-2.exe (or newer) to match that of the real-time INtime version being used. (where the numerics are the revision level).

8. Open up one of the Visual Studio projects and remove the CTC_Incentive reference. Right click ‘Add Reference’ and browse to the installation of the new CTC_Incentive.dll, adding it as a reference. (typically: C:\Program Files\Control Technology Corporation\Incentive\CTC_Incentive.dll). Rebuild the test program to confirm successful installation.

Realtime Incentive EtherCAT Master PC:

1. Perform all Windows updates prior to installation.
2. Install ‘Incentive License Setup.msi’ in the Licensing sub-folder. This is required for QuickBuilder demos but not Incentive real-time demos. All fully licensed products will require it in order to authenticate licenses. Reference the section on Incentive Licensing. Install the Incentive License Monitor service after you activate your licenses online. Note that upgrades of older installations do not need the new licensing tools and your existing licenses will be used.
3. Watch the “2_INtime_6_1_Installation.mp4” video and install runtime61-17004_installer.exe or “2_INtime_6_3_Installation.mp4” video and install runtime63-19040-2_installer.exe (or newer) as directed.
4. Watch the “3_INtime_Configuration.mp4” video and setup/configure the system as needed. Make sure to copy each folder found in ‘Root_Folders’ to the root of the C: drive.
5. Watch the other videos based upon what you wish to install and follow the directions.
6. Once operational fully license the product otherwise it will run in demo mode.

It is recommended to increase the INtime shutdown time to 5 seconds to ensure disk flushing of non-volatile variants. The default is 500 ms but as programs get larger, and more non-volatile variables are used, the greater time is needed to ensure Windows closes the real-time disk files. Reference the “Disk Cache not fully flushed during INtime menu ‘Stop All’ on large programs” section within “Some Common Issues and Resolutions”. Also, Windows updates should also be disabled and only done manually since Windows does an automatic reboot of the system after an update is applied. For a device running 24/7 this would not be acceptable and would cause Incentive to stop running.

BIOS Settings and Configuration

It is important to set the BIOS of your PC correctly. Failure to do so will cause jitter and possible device faults on the EtherCAT network due to timing issues. Some BIOS settings can cause unstable real-time operation, like turbo mode. It is suggested you run the INtime Graphical Jitter program prior to Incentive for at least 30 minutes on a new, unproven system, to ensure there are no jitter deviation flagged in red. Below is a list of some BIOS settings you may see on your computer and how they should be set, the example is from a Dell Optiplex PC. Each system tends to be different so you may not have access to all that appear below.

Virtualization Support

Virtualization – Disable (or do not check Enabled)

Power Management

Intel Ready Mode – Disabled (or do not check Enabled.

Block Sleep – Enable Block Sleep (S3 State)
Deep Sleep Control – Disabled

Performance

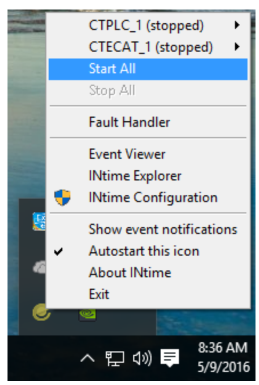
HyperThread control – Disabled
Intel TurboBoost – Disable (or do not check Enabled)
Limit CPU ID Value – Disabled (or do not check Enabled)
C-State Control – Disabled (or do not check Enabled)
Intel SpeedStep – Disabled (or do not check Enabled)

Intel Software Guard Extensions

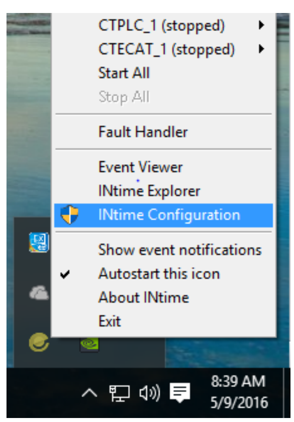
Intel SGX Enable – Software Controlled

Startup and Network Configuration

Upon installation the Windows® PC appears and operates identical to a normal PC. The EtherCAT environment can be setup to run automatically when the PC boots or started/stopped manually by using a tray icon. The environment should only be started when the EtherCAT devices are ready for operation otherwise it will timeout and require a restart, just like the hardware based 5300 controller.

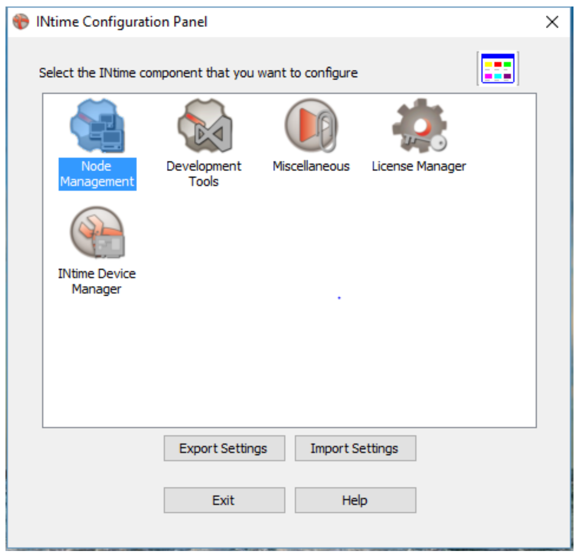


To modify the properties of the Incentive real-time environment you may invoke the INtime Configuration Manager. The video, ""3_INtime_Configuration.mp4", contains the latest configuration information and should be referenced as well as the information below.

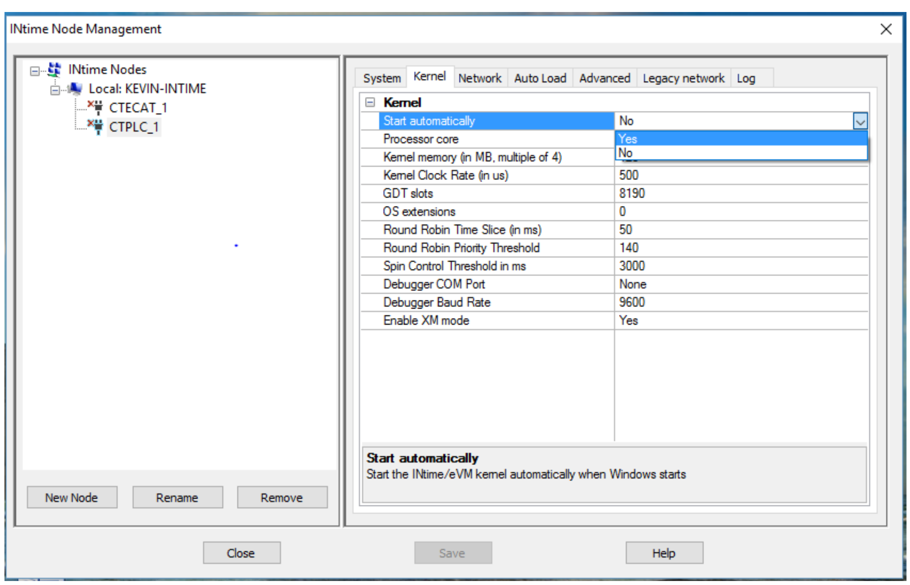


If for some reason the icon for INtime does not appear in the system tray, or INtime was installed by another user, simply run 'intimestatus' from a cmd window or located at "C:\Program Files (x86)\INtime\bin". Nothing will appear to happen but the icon will be in the system tray now. Click the "Autostart this icon" for it to appear upon each login of this user. Note that it can take several seconds after logging in for the icon to appear, depending upon the speed of your processor and other programs run by your startup menu.

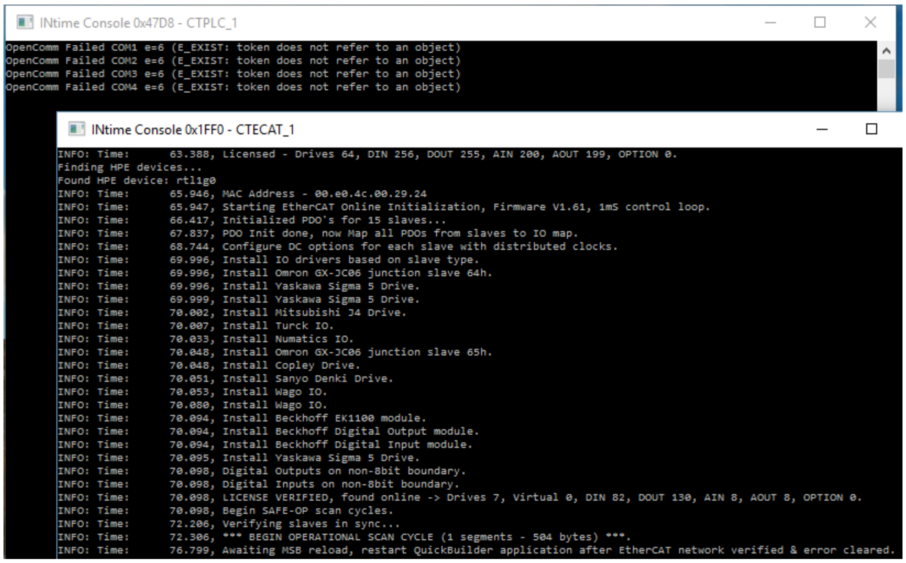
Select 'Node Management':



If you wish to have the Incentive EtherCAT environment start automatically with Windows® you can select 'Yes' to the 'Start Automatically' prompt within the INtime Node Management screen. This must be done for both CTPLC_1 and CTECAT_1 in a quad core system, CTECAT_1 only with dual core.

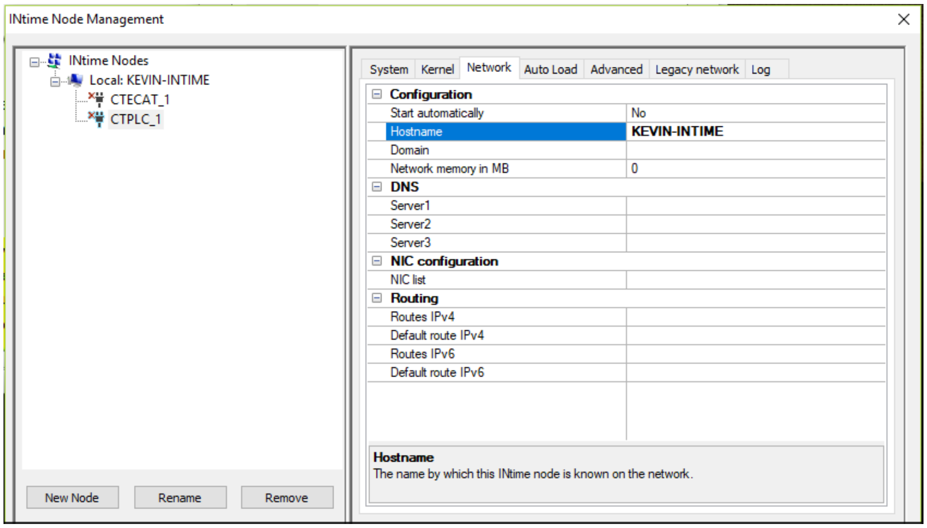


When started, Incentive will open up to two console screens, one for the EtherCAT core and one for the QuickBuilder PLC Logic core (non-automatic mode); expect several seconds of delay, especially if the optional LCD display is used. Diagnostic messages will be presented to these screens identical to what is typically viewed via the EtherCAT Explorer log, remotely. Note the "OpenComm Failed" errors, that is normal for each COM port not found and is meant to notify you that if you intend to use a serial port, none was found.

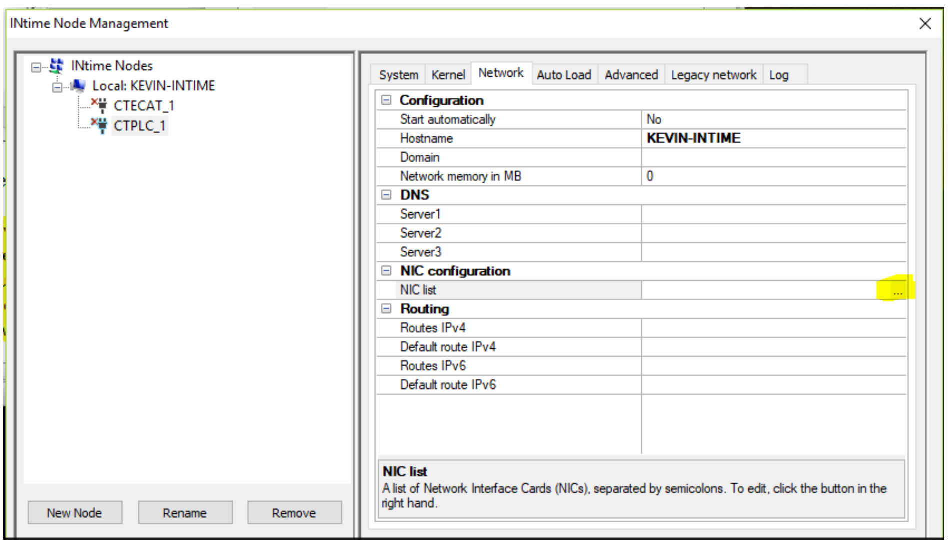


The QuickBuilder PLC Logic core shares the main PC Ethernet adapter (bridged) and requires its own IP address. This is the IP address which all the communication protocols will use and it is different than the main Windows® PC even though using the same network adapter. You may assign this dynamically with DHCP or use a fixed static IP address, depending upon your network.

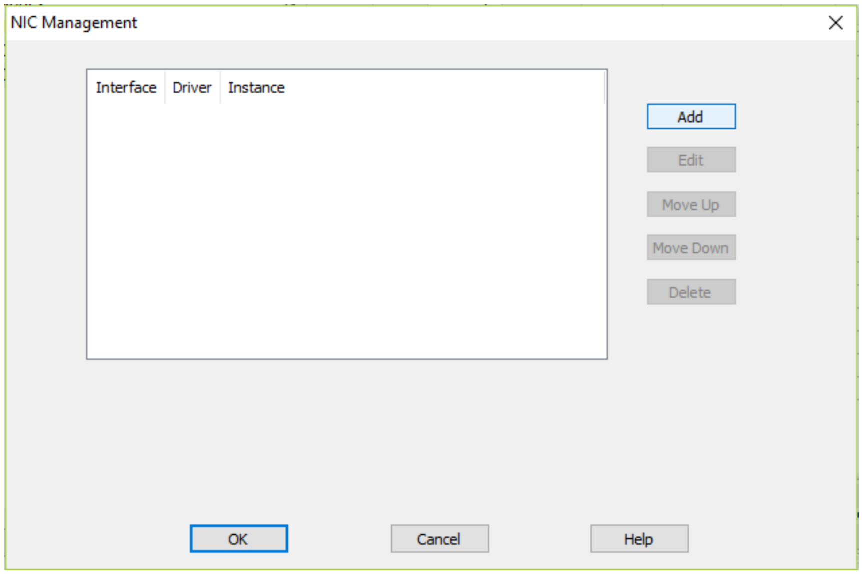
To begin we must first create a Virtual Ethernet Adapter. On the CTPLC_1 core select the 'Network' tab (CTECAT_1 if using a dual core and not running CTPLC_1). Set 'Start Automatically' to No since Incentive will automatically start the network itself when started. The 'Hostname' is that which is registered with the DHCP server when requesting an IP address and/or used by a remote Incentive API to address the computer on a network. The host path for the API would use 'KEVIN-INTIME' to open a connection to the node defined below.

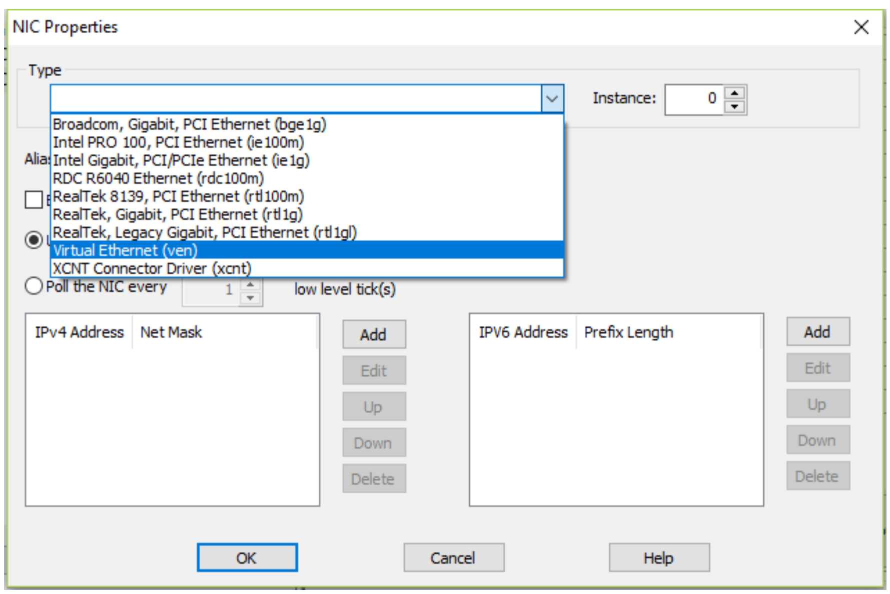


Next the adapter itself will be created. Select the '...' box to the right of NIC list:

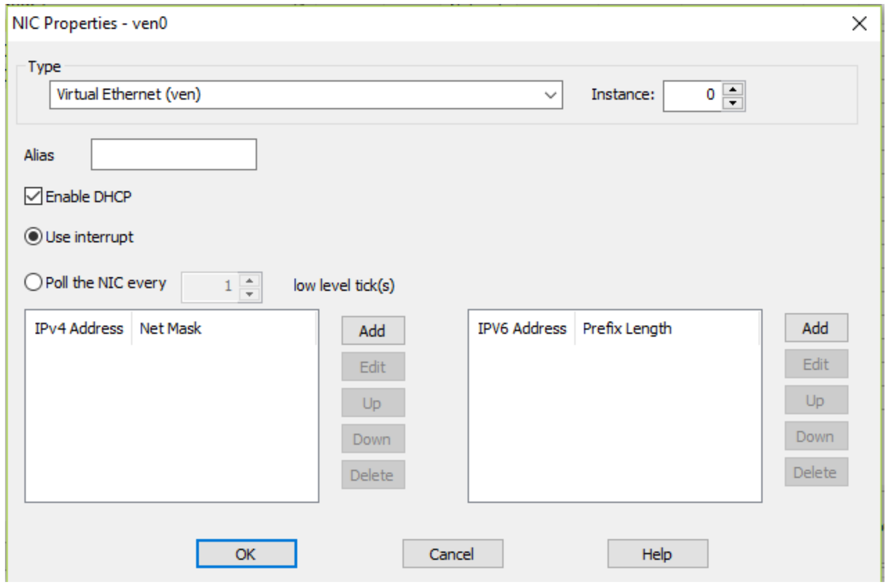


This will have a dialog appear where you will click 'Add':

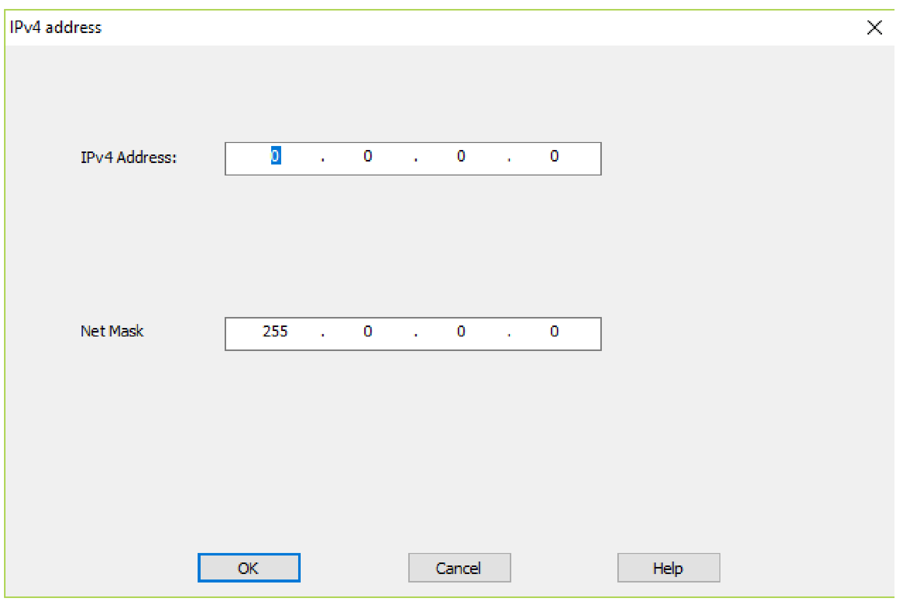




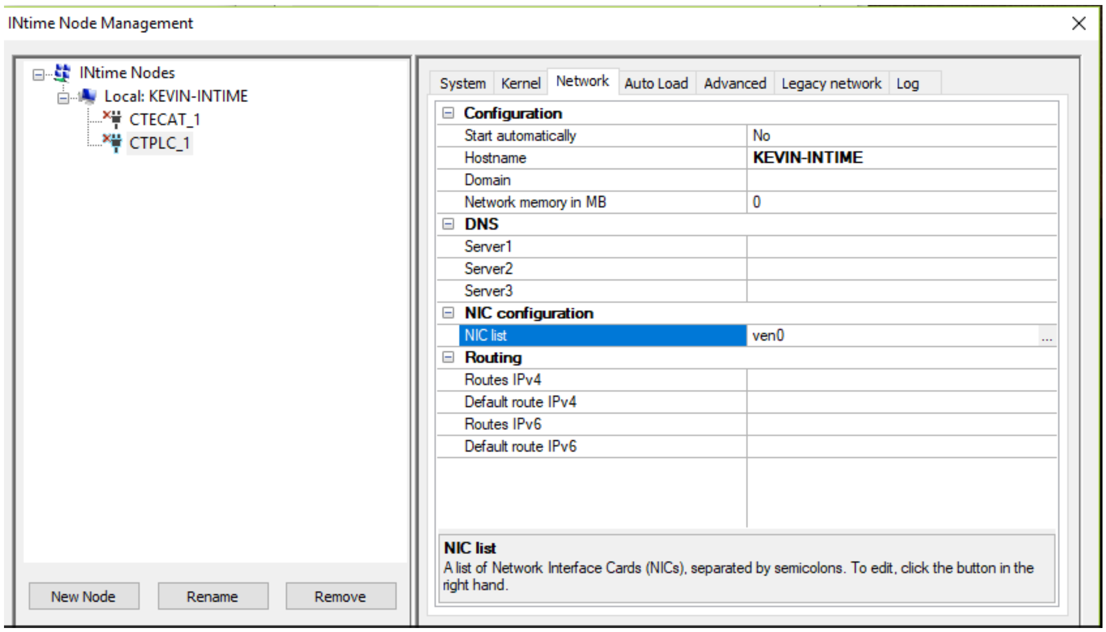
If running DHCP check the Enable DHCP box followed by OK:



If running a static IP address do not select 'Enable DHCP', select the 'Add' and enter the IP Address and network mask for a static IP address (Windows PC and Incentive must be on the same subnet mask):

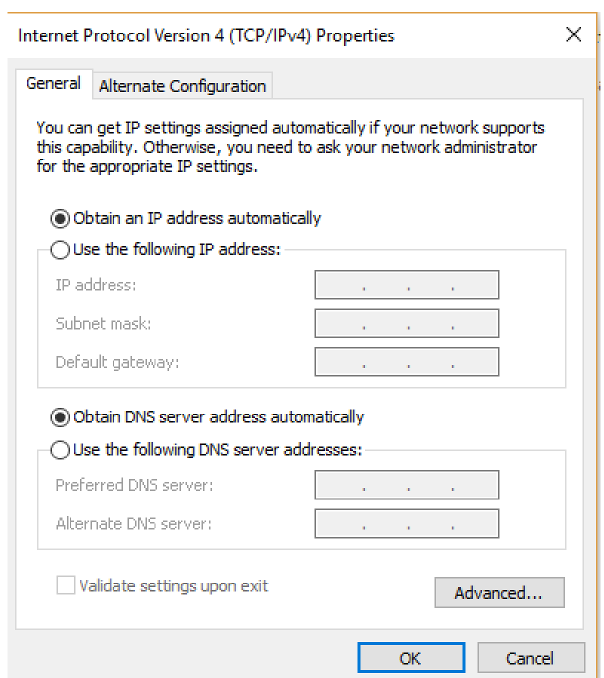


Select OK to accept all entries until back at the 'Network' tab and a summary will appear:

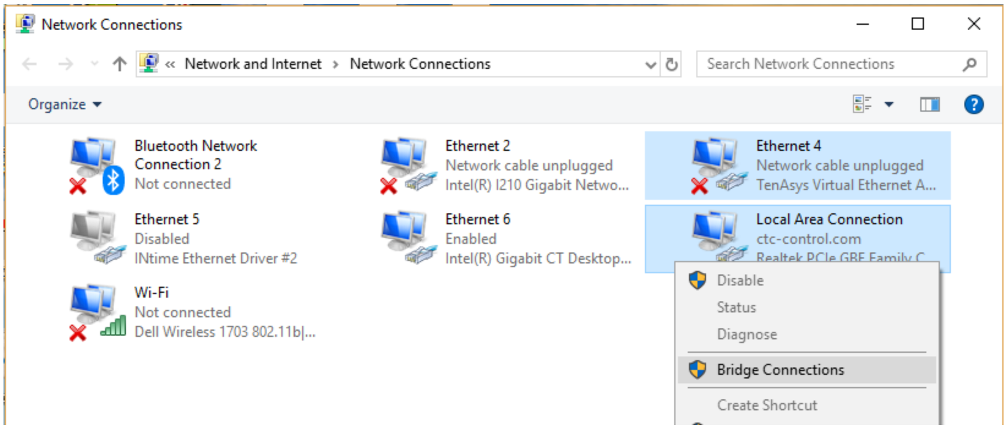


Note that it can take up to 30 seconds for DHCP addresses to be resolved when starting Incentive. Now that a Virtual Ethernet Adapter has been created it must be shared (bridged) to the main Windows network adapter that will be used for any outside communications.

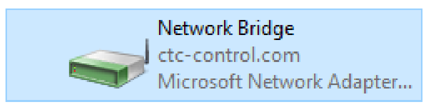
The Windows network adapter should be set for DHCP, if static will be used the IP Address it will be set in the Bridge after creation:



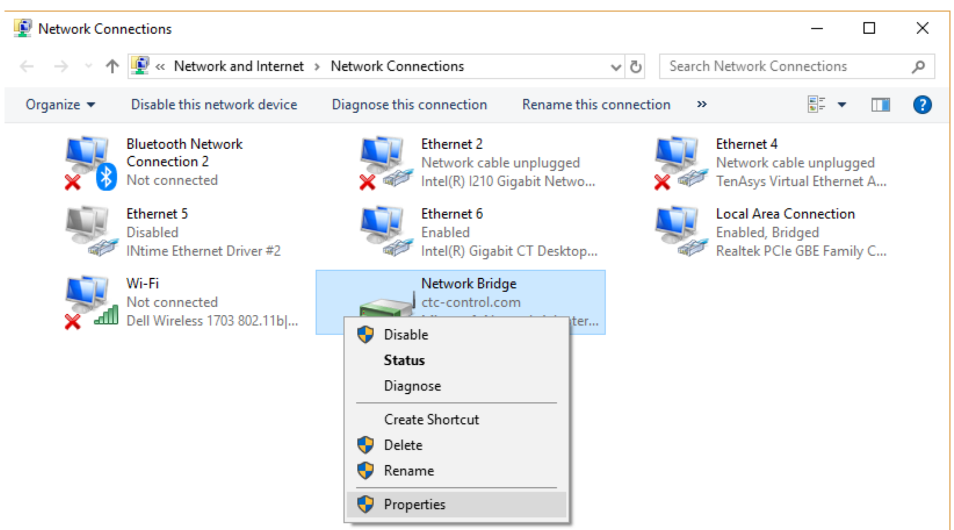
In order to share the main PC network adapter a bridge must be created with the virtual TenAsys INtime adapter. This is done by going to your Network Connections within Windows®, hold the control key and click on the Network adapter desired as well as the TenAsys Virtual Adapter, right click and click “Bridge Connections”.



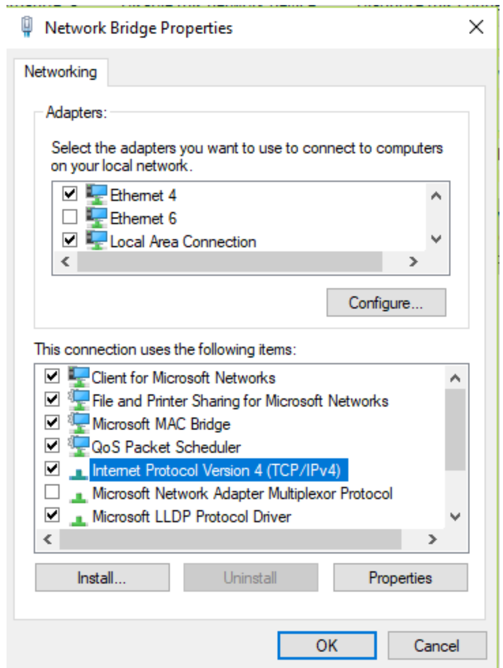
Windows will create a new network adapter called ‘Network Bridge’:



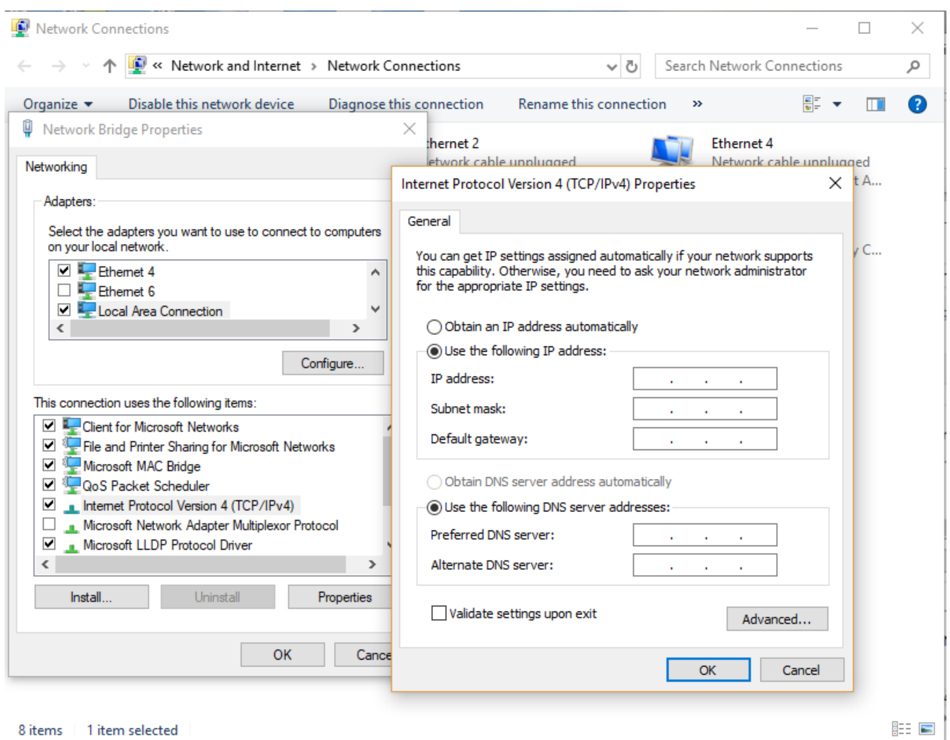
If you are running DHCP you are done and the PC typically has to be rebooted. If static is needed you must right click the Network Bridge and select ‘Properties’:



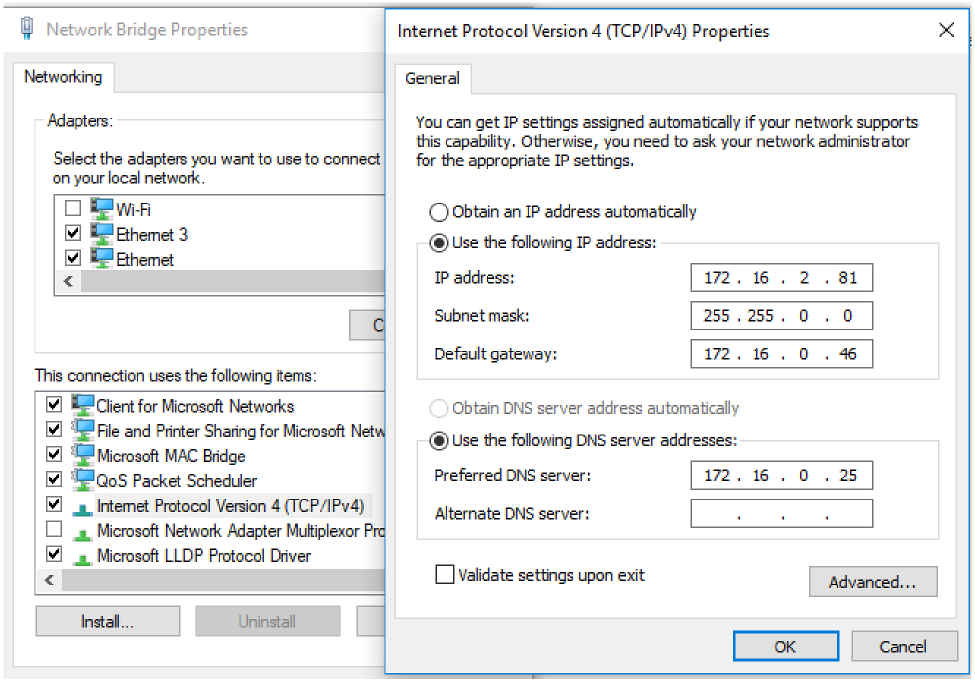
Select 'Internet Protocol Version 4' under Network Bridge Properties:



Double click the 'Internet Protocol Version 4 (TCP/IPv4)' to set the desired parameters for the Windows side (for example if Incentive real-time was 192.168.1.2, Windows should be different such as 192.168.1.3, both with the same subnet mask of 255.255.255.0, gateway and dns are optional on non-routable networks):



An example where the Incentive INtime real-time IP address is 172.16.2.80 and the subnet mask is 255.255.0.0 is below with a Windows PC side of 172.16.2.81 and a gateway and dns are also set:



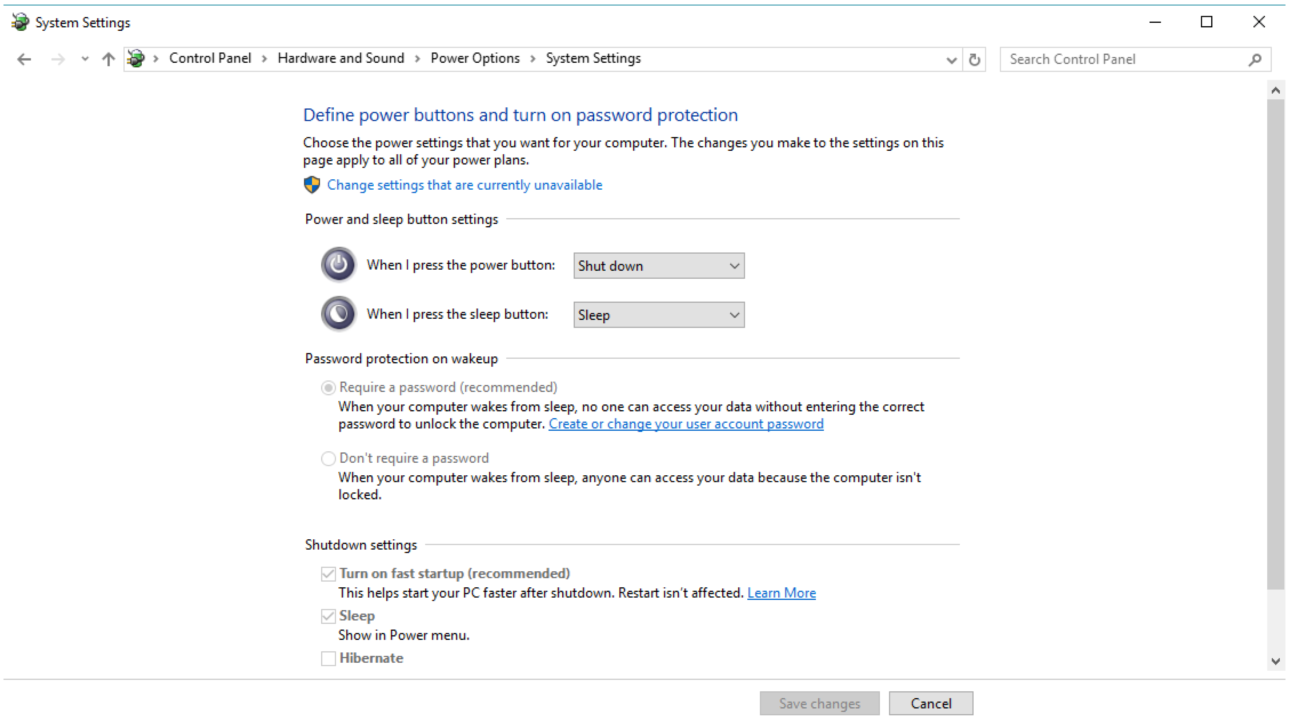
The private address segments (10.0.0.0 - 10.255.255.255, 172.16.0.0 - 172.31.255.255, 192.168.0.0 - 192.168.255.255) are commonly referred to as "non-routable" addresses.

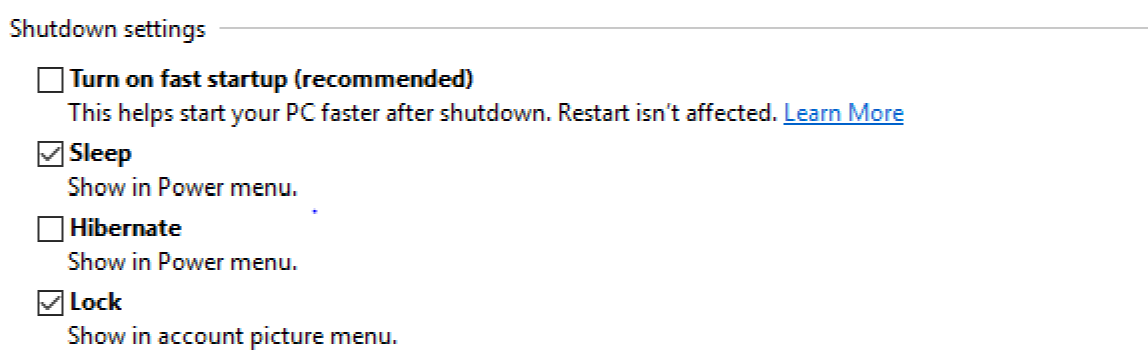
To remove the Bridge, simply right click on the adapter and click “Remove from Bridge”. The QuickBuilder PLC Logic core will not be able to communicate via UDP or TCP without the Bridge in place. Reboot after creation.

When running DHCP, to find out the current IP Address assigned to the Incentive PLC Logic environment reference the CTPLC_1 IO Console. A message will appear at startup displaying that assigned either by DHCP or static IP. CTECAT_1 IO Console will display the MAC Address of the assigned Ethernet port, typically referenced for licensing.

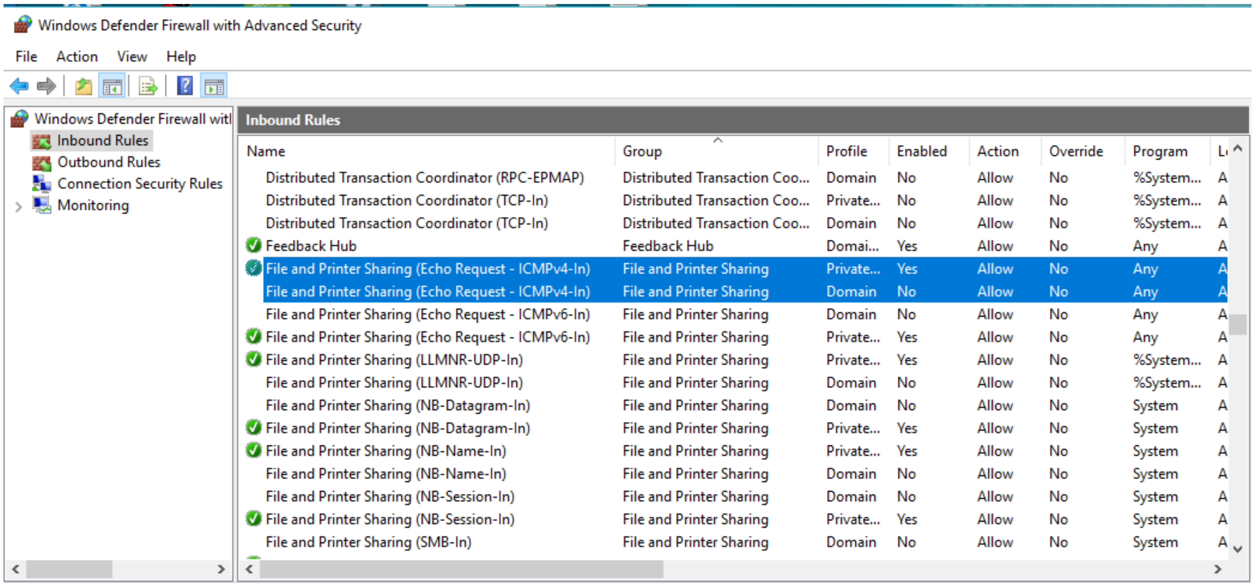
Make sure when using a network that your wifi adapter is not on the same network otherwise packets may get confused and be sent out the wrong network adapter. For example if the hard wired LAN is 192.168.1.5 and the wifi is 192.168.1.25 (255.255.255.0). It is best to change one of them to something outside the subnet range of the other. This is especially needed if the wifi adapter is used for remote support.

Once the Bridge is created we have discovered a problem within Windows where doing a shutdown on some systems causes the Bridge to not function anymore. This has something to do with a shutdown being a memory image dump on Windows 10 and not really a full shutdown, more a hibernate. The resolution to this problem is to go into ‘Power Options’ within the Control Panel, on the left side of the panel there is an option “Choose what the power buttons do”, select that and the screen below will appear. Note that the option “Turn on fast startup (recommended)” is grayed out. Select “Change settings that are unavailable” and you will be able to modify it. Deselect it and you may then press the “Save Changes” button. This will allow a full shutdown and resolve any bridge issues caused by rebooting. Note that “Restart” does not have the issue, only “Shutdown”, due to hibernation mode being entered for fast startup.





Note that if you wish to test your network using Ping, Windows 10 disables inbound ping responses by default and must be enabled under the Windows Firewall advanced settings, Inbound Rules:



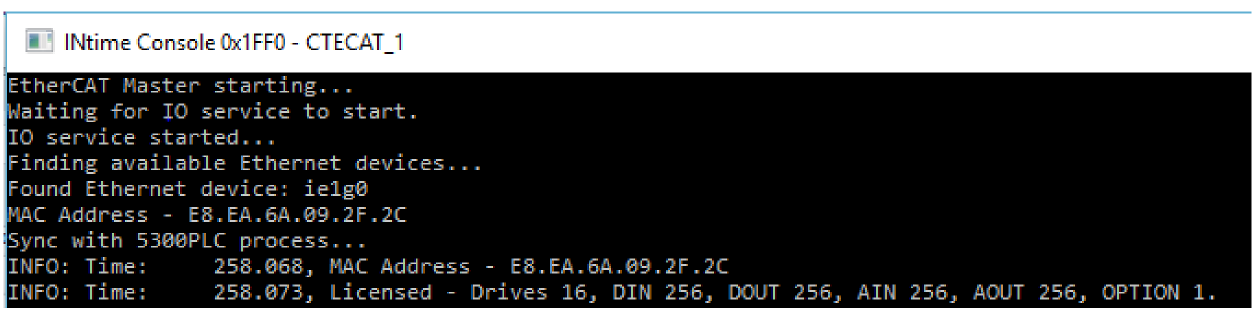
MAC Address for Large Systems

The MAC Address used by the INtime Bridged virtual adapter is dynamically created. In large systems there is the potential for duplicates to occur. If this should happen or to prevent it from happening a unique MAC Address should be used. It is recommended to use the MAC Address that appears on The CTECAT_1 screen for your EtherCAT network. This network will be private and ensure a unique MAC Address. Thus edit the file loader.cfg found in: C:\ProgramData\TenAsys\INtime\CTPLC_1\etc

At the end of the file is something like:

```
dev.ven0.macaddr=1A:EF:C8:5B:BC:00
```

Change the macaddr to that found on the EtherCAT IO Console screen:



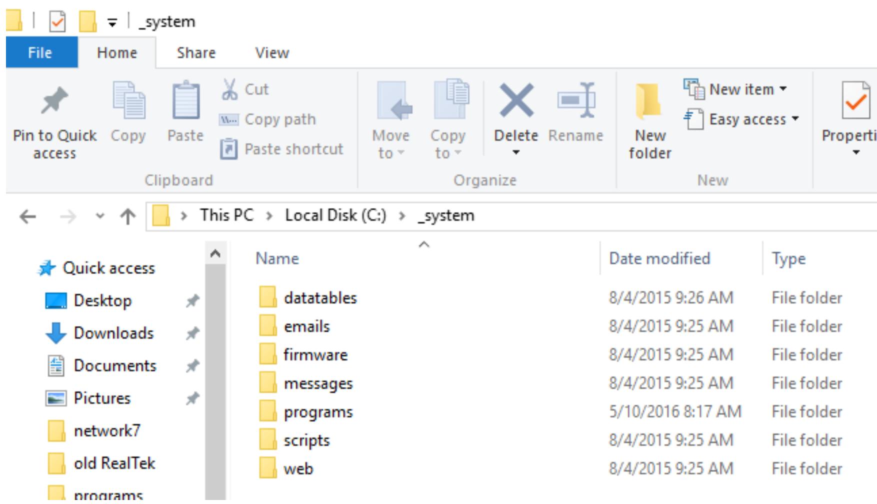
In order to use that show above the line would be changed to:

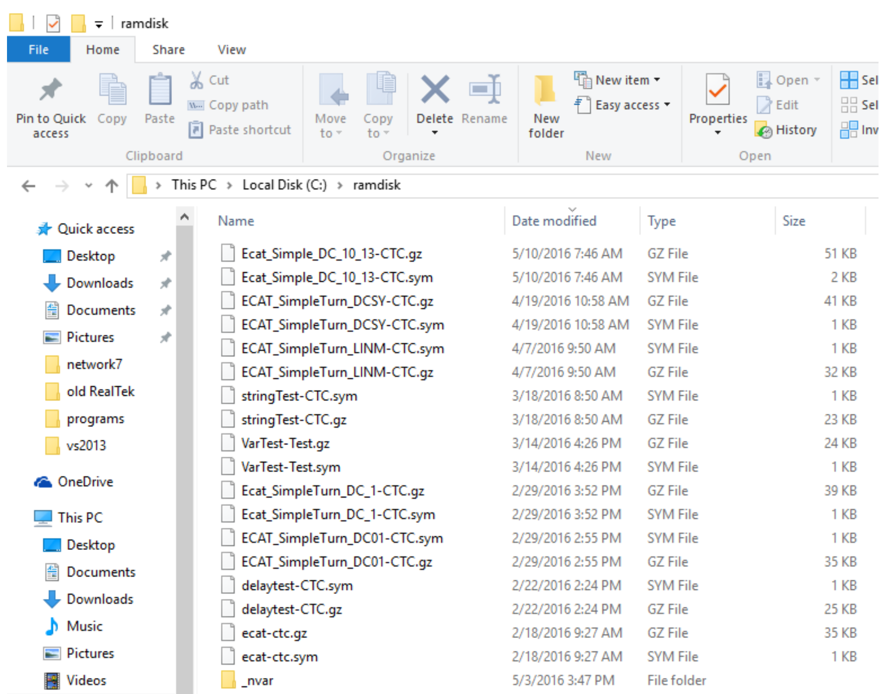
```
dev.ven0.macaddr=E8:EA:6A:09:2F:2C
```

Reboot the controller for it to take effect.

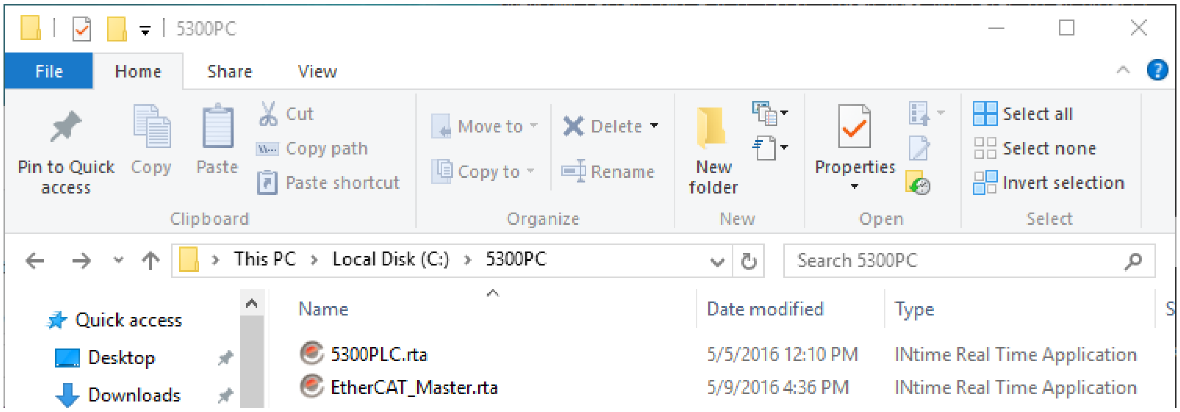
File System

The PC file system mimics that of the standard embedded 5300 controller, using two subdirectories off the C: drive, '_system' and 'ramdisk'. The files within these directories are the same as described in the 5300 manual. The main difference is the local SATA solid state drive is used instead of the flash file system of the 5300 for the '_system' directory, as well as 'ramdisk'. As with the embedded 5300, the \ramdisk\nvar subdirectory contains any variant storage as well as some scratch files called 'nv#501.reg' and 'nv#32001.reg' (where # is 0 or 1). These files contain the non-volatile registers 501 to 1000 and 32001 to 36000. If any of these 5300 registers are modified, these files are updated every 5 seconds and saved during a normal shutdown for reloading at the next power-up.

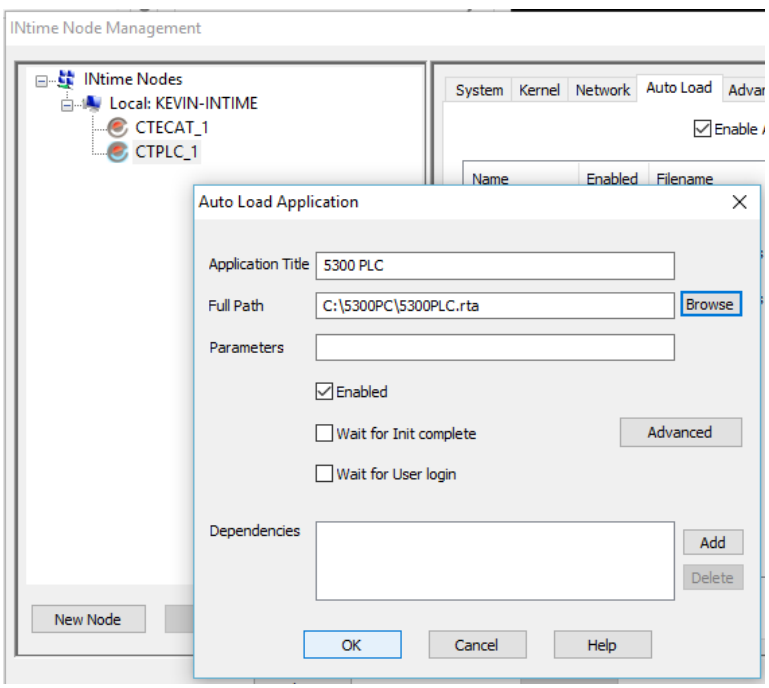


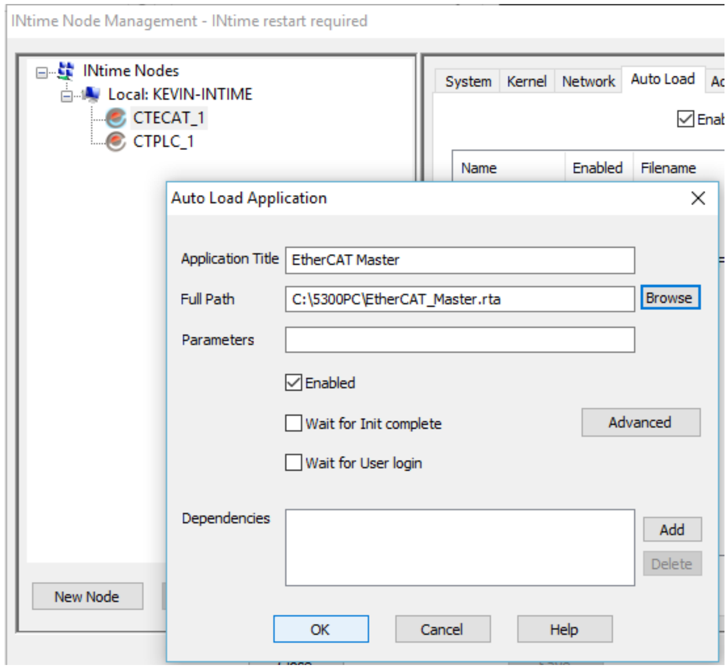


The real-time applications which run on each CPU core are maintained in the \5300PC directory, 5300PLC.rta and EtherCAT_Master.rta and are referenced from the INtime Node Management Auto Load tab.



Auto Load configuration, starts EtherCAT from tray icon, the default for quad core:





In a dual core system (single core used by Incentive) both EtherCAT_Master.rta and 5300PLC.rta would be listed under CTECAT_1, CTPLC_1 would not exist (freeing up a core for Windows®). This is also true for standalone mode, when driven solely by the Windows API.

The _system\Programs directory contains various configuration files. These files are set by using the EtherCAT Explorer ‘User Options’, ‘Save Config, and ‘License’ forms. Since more than one EtherCAT network can run at a time there are some files that contain the MAC Address of the EtherCAT Ethernet adapter as part of its file name to make the name unique. Files in this directory are:

Used by the 5300PLC.rta process –

_lcddisplay.txt : Some embedded PC devices have an LCD display for status messages to be displayed. If present this file contains the IP address in text format, for example: 172.16.2.190
_runprogram.txt : Contains the path\filename of any QuickBuilder project to automatically startup at boot. It is set by QuickBuilder during a download of a project file if the option is enabled in the project properties. For example: \RAMDISK\ECAT_SimpleTurn_DCSY-CTC.gz
_Init.bin: Saves configuration information like custom serial port setting, baud rates, etc. It is automatically created with the system defaults if it does not exist.

Used by the EtherCAT_Master.rta process –

options[MAC ADDRESS].txt : This file contains the settings from the ‘User Options’ EtherCAT Explorer configuration form. If no file exists a file will be created with default settings of 1 mS cycle time, 300uS pdo timeout, 5 initialization retries, and no virtual axis. Example filename: _options_E8EA6A092F2C.txt where E8EA6A092F2C is the 6 byte mac address of the EtherCAT Ethernet network adapter being used.

ioOptions[MAC ADDRESS].txt : This file contains the license settings from the EtherCAT Explorer license configuration form. It is an encrypted binary file. When the software is first installed a file called _license_[MAC ADDRESS].txt will be emailed to you. That file should be placed in the same directory as this file. When the _ioOptions file is not present the system will look for a valid _license file. If found that file will be imported and the _ioOptions file generated. The _ioOptions file can also be generated from the QuickBuilder EtherCAT Explorer configuration form, as it is with the 5300 M3-41A. In a PC system it is typically easier to have the license file emailed to you and then placed in the _system\Programs directory for automatic import.

license[MAC ADDRESS].txt : This file is generated by CTC Technical Support to enable the number of drives and IO your EtherCAT runtime is licensed for. If this file and the _ioOptions file are both missing the system will run in demo mode. In this mode EtherCAT will control up to 16 drives and 256 digital and analog IO for a period of about 3 hours. After that the network will reset and power will have to be cycled to restart another 3 hour period. The generation of this file requires the MAC Address of the EtherCAT network adapter. You can get the MAC Address by starting the EtherCAT_Master.rta process and it will appear on the console screen. Either email Technical Support or have it available when calling so that a proper license file can be generated. Systems pre-configured by CTC will already have the license installed. Example filename: _license_E8EA6A092F2C.txt.

log[MAC ADDRESS].txt : This file is written to with the same information as the IO diagnostic console and provides a hard copy of each session. The file is automatically appended to. Example filename: _log_E8EA6A092F2C.txt where E8EA6A092F2C is the 6 byte mac address of the EtherCAT Ethernet network adapter being used.

It is recommended that an inexpensive UPS be connected to a serial port for automatic shutdown should extended or abrupt power failures occur. The will ensure file integrity as well as preservation of non-volatile registers.

Demo Mode and TenAsys INtime Licensing

Both the TenAsys INtime runtime, Incentive API and EtherCAT Master can run in demo mode. Demo mode for INtime Version 6.1.17004 is currently valid for 60 days after which the QuickBuilder EtherCAT environment must be purchased. It cannot be extended due to limitations imposed by TenAsys. The QuickBuilder EtherCAT network will run for up to about 3 hours continuously, before resetting. In this mode EtherCAT will control up to 16 drives and 256 digital and analog IO. In order to purchase a full EtherCAT license please contact Control Technology with your motion control and IO needs as well as the MAC Address of the Ethernet adapter that will be used. From Windows the MAC Address can be found by opening a ‘cmd’ window and typing ‘ipconfig /all’. The Physical Address of your adapter is the MAC Address.

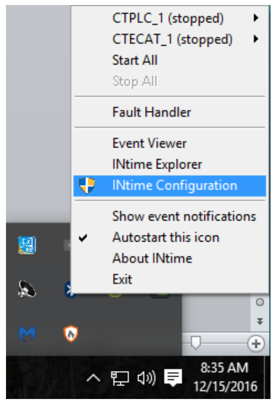
With TenAsys INtime 6.3 demo licensing is no longer available. It is recommended that if a demo unit is needed or testing prior to licensing then install 6.1.17004, conduct the needed testing, then update to 6.3 after full licensing. A demo license can be provided for 6.3 but only after the system fingerprint is emailed to CTC, a license will then be generated, similar to the actual license, valid for 30 days.

If 6.1.17004 is installed as a demo and then upgraded to 6.3, during the demo period; the demo license will currently work and be carried over to the 6.3 revision. The video 10_INtime_Upgrade_From_6_1.mp4 is available to show how to upgrade your embedded PC.

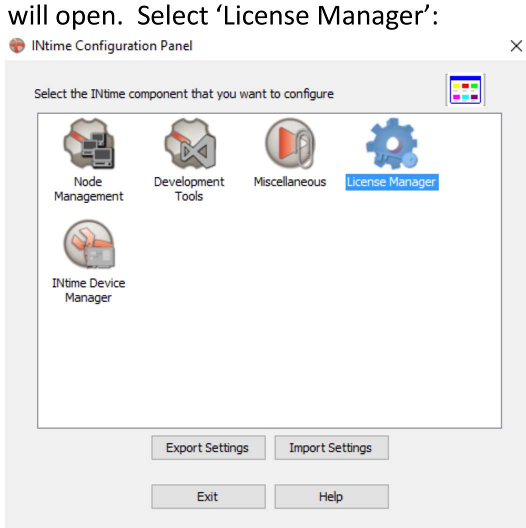
TenAsys INtime Licensing

If your INtime license has not already been installed you must send a system fingerprint to CTC (along with purchase order) so that a license string can be generated for your system. This license is keyed to your hard drive so should that be replaced, you will need a new license. You may reference the below information or watch the video, “9_Licensing.mp4”.

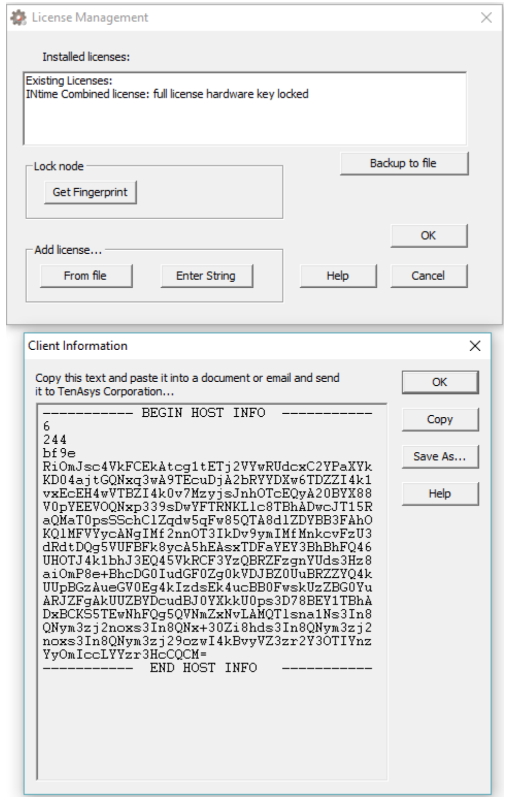
In order to retrieve the fingerprint go to the system tray and select ‘INtime Conguration’, right clicking on the TenAsys INtime icon:



The INtime Configuration Panel will open. Select ‘License Manager’:



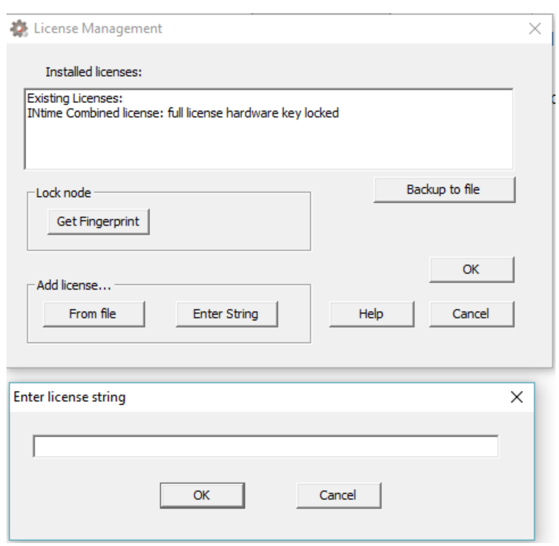
Select ‘Get Fingerprint’ followed by ‘Copy’ or ‘Save As…’ on the Client Information form that opens:



Copy (or attach) the information into an email and send it to CTC for validation. Once validate a license string will be generated and returned. An example of this would look something like:

*AA2KZ4OIAJ75NDSYKM6GN3ETSEVKHRA44DRS3KBJZIB8IEEWSVYY2# "16" version "", no expiration date, exclusive

Select 'Enter String' from the License Management screen, copy and paste the license you sent them followed by clicking 'OK':



Incentive Legacy Licensing

Your Incentive license is keyed to the MAC Address of the Ethernet port used for EtherCAT. While in demo mode it will run for 3 hours and then need restarting. To fully unlock a license file (_license_[MAC ID].txt must be requested from CTC which defines the number and type of IO licensed as well as to whether the API is fully enabled. This file would be placed in the C:_system\programs directory. All _ioOptions* files must be removed if only a EtherCAT Master. If multiple then only remove the existing _ioOptions* file that matches the MAC Address of the network adapter used. Reference the the video, “9_Licensing.mp4”.

> ***_license_[MAC ADDRESS].txt*** : This file is generated by CTC Technical Support to enable the number of drives and IO your EtherCAT runtime is licensed for. If this file and the _ioOptions file are both missing the system will run in demo mode. In this mode EtherCAT will control up to 16 drives and 256 digital and analog IO for a period of about 3 hours. After that the network will reset and power will have to be cycled to restart another 3 hour period. The generation of this file requires the MAC Address of the EtherCAT network adapter. You can get the MAC Address by starting the EtherCAT_Master.rta process and it will appear on the console screen. Either email Technical Support or have it available when calling so that a proper license file can be generated. Systems pre-configured by CTC will already have the license installed. Example filename: _license_E8EA6A092F2C.txt.

Windows® Updates

Windows® updates are pushed by Microsoft periodically and are required for proper operation as well as security. With the release of Windows® 10 Micosoft forces updates rather than allowing you to be prompted. Doing an update while you are controlling an automation environment is generally not a good thing since in many cases Microsoft reboots the PC. Thus it is strongly suggested that updates be turned off or at least request a prompt. On Windows® 10 you can set a connection, network or WiFi, as metered to prevent updates. Wifi can be set within the Control Panel whereas an Ethernet network must be set via a registry entry. There are numerous articles available about how to do this with that below being a particularly good reference:

Windows 10 Pro (schedule updates):

http://www.windowscentral.com/how-schedule-windows-updates-windows-10
You can also go to “Settings->Advanced Options->Defer feature updates” should be selected.

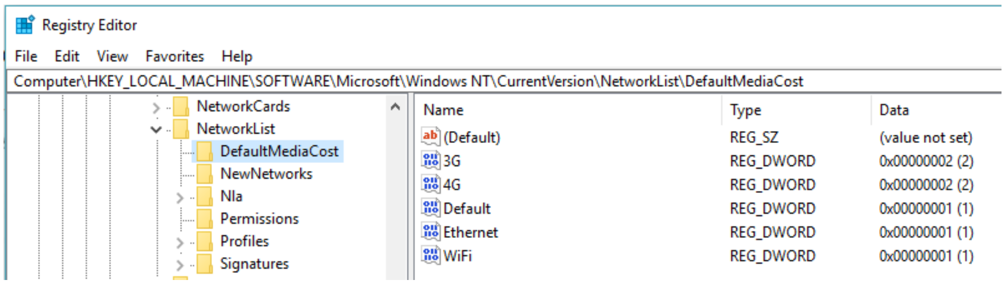
Windows 10 Home & Pro (metered connection):

http://www.windowscentral.com/how-set-ethernet-connection-metered-windows-10

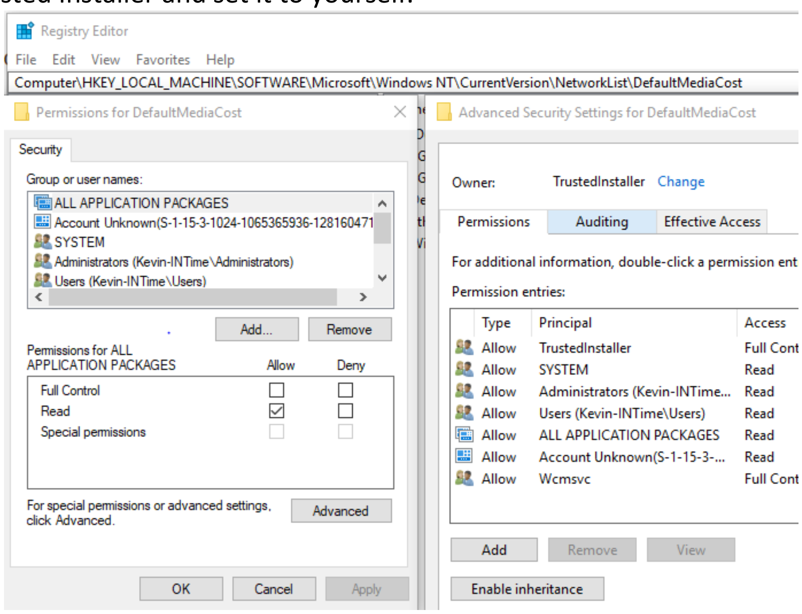
Failure to defer your updates until you are ready for them will cause your PC to reboot automatically at unpredictable times due to Microsoft pushing out Windows Updates. This will cause your real time control to stop operation as well. Also be warned that Microsoft has begun overwriting any registry changes you make, such as metered connection, during major updates, therefore always verify prior changes made are still present.

Below is a summary of what registry entry to change to prevent updates, basically setting all possible connections to metered:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\NetworkList\DefaultMediaCost



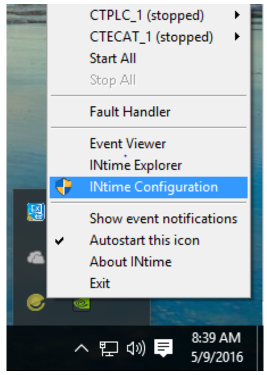
Right click DefaultMediaCost and select permissions followed by Advanced in the popup dialog. Click 'Change' next to Trusted Installer and set it to yourself.

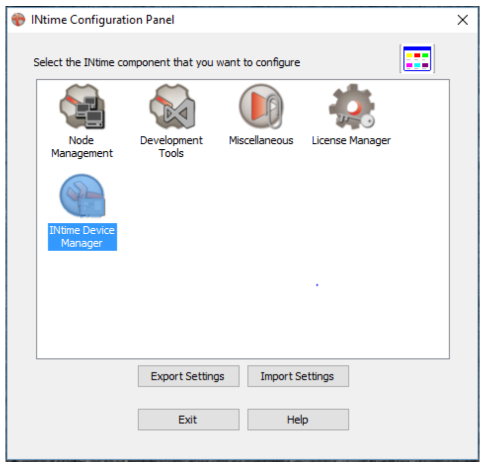


Set Default, WiFi, and Ethernet to a 2 for metered after you set the TrustedInstaller to yourself. Reference the link previously provided for more detail.

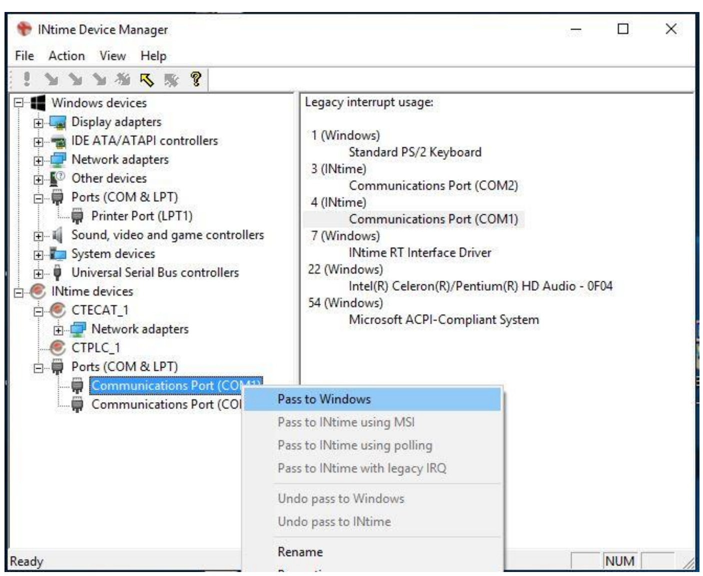
Serial Ports

At present time the QuickBuilder PLC Logic core will attempt to open COM1 through COM4 for its own use. Whichever ports are found will be used by the standard CTC Binary Protocol and available for QuickBuilder programming, those not found will cause a diagnostic warning on the Incentive PLC console screen. Serial ports may be mapped back and forth between the INtime real-time environment or for Windows® programming use. Mapping is required in order for ports to be disabled from Windows® and enabled for INtime. To map serial ports between the INtime real-time environment and Windows® the Intime Device Manager must be used:

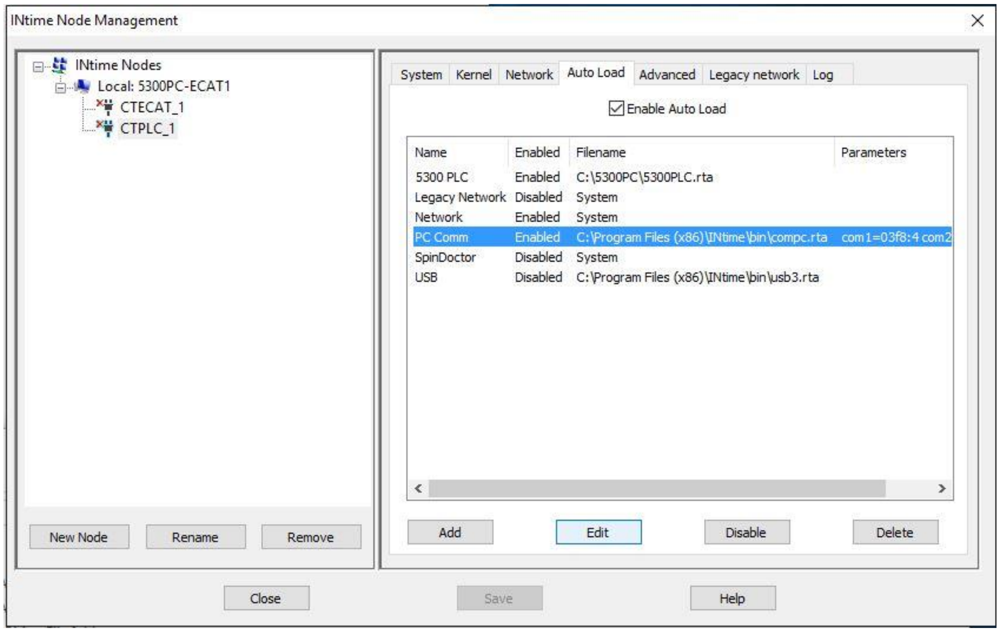




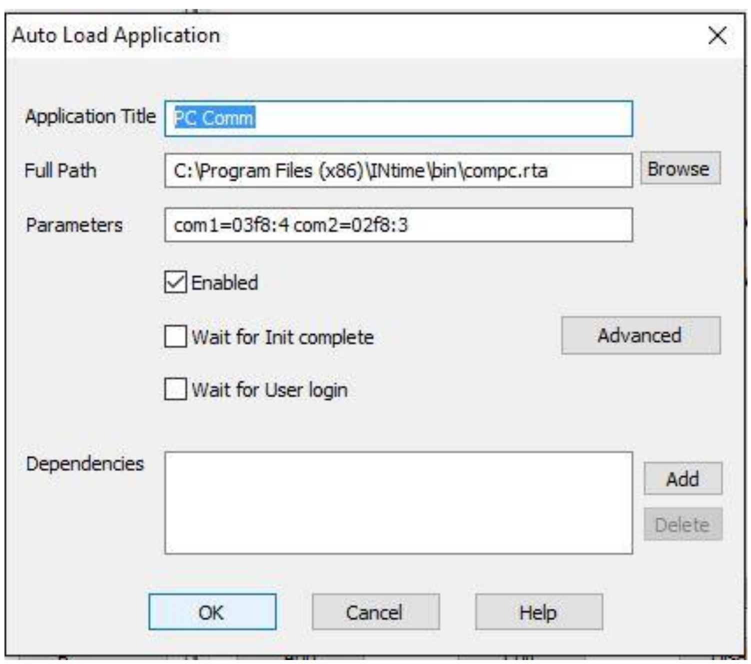
Note to return a serial port to Windows control, right click on the COM port under INtime devices and select ‘Pass to Windows’. To map it back select the COM port under Windows devices and select ‘Pass to INtime with legacy IRQ’. Rebooting is required for it to map properly.



In addition, when mapping serial ports a special driver (compc.rta) must be passed the IO address of the port and interrupt used. This is configured within INtime Node Management, CTPLC_1 for quad core, CTECAT_1 if dual:



Modify the 'Parameters' section as required. You may refer to the Windows® Device Manager->Ports (COM & LPT) section properties to determine what is currently available and IO address:interrupts used.



For example the parameters for COM1 with a hex address of 0x03f8, using IRQ 4, and COM2 with a hex address of 0x02f8, IRQ 3 are shown above.

Serial ports are referenced by Incentive starting with COM1 and added sequentially. This means if COM2 is mapped by INtime but not COM1 then COM2 will become COM1 within the Incentive environment. If COM2 and COM3 were mapped then Incentive COM1 is Windows COM2 and Incentive COM2 is Windows COM3. Incentive has no idea of how Windows referenced the COM ports, just each assigned will become sequential, from 1 to 4.

Another thing to keep in mind is that you cannot change the serial port parameters while the port is open (Windows must have the port closed), thus the registers for changing baud rate, parity, data and stop bits, will have no effect. The proper way to set serial port parameters is via the _Init.bin file, where settings are stored and read during initialization. The 'set COMM#' command, available via telnet, can be used to retrieve and save serial port setups.

Set COMM[Port] [Baud Rate], [Data Bits], [Parity], [Stop Bits], [Protocol], [Flow Control], [Address]

Port

1, 2, 3, or 4 are valid entries as long as the port is available. Example: 'set COMM1', 'set COMM2', …

Baud Rate

Baud Rate may be one of the following (19200 is the default):

- 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200

Data Bits

"7" or "8" data bits with 8 being the default.

Parity

"None", "Odd", or "Even" parity, with "None" being the default.

Stop Bits

"1" or "2" stop bits with "1" being the default.

Protocol

Protocol may be one of below, case sensitive:

- CTC Binary (Default, compatible with CTCMON and ctccom32.dll)
- Modbus Master/RTU – controller polls the device, binary mode.
- Modbus Master/ASCII – controller polls the device, ASCII mode.
- Modbus Slave/RTU – controller polled by external device, binary mode
- Modbus Slave/ASCII – controller polled by external device, ASCII mode
- Diag. Terminal – Diagnostic Terminal
- Philips ISP – Programmable chip mode
- Scale – Custom scale protocol

Flow Control (not supported on 5300, only Incentive PC environment)

"None", "Xon", or "Hardware" are available options.

None – Typically 2 wire, TX/RX half duplex protocol like CTC Binary.

Xon – Xoff (0x13) is sent when receive buffer is full, Xon (0x11) when OK to send again.

Hardware – DTR is enabled when the port is open. RTS is active when OK to receive characters, CTS controls transmitter, when not active transmission will not occur.

Address

This is the address to be used when Modbus protocols are selected. When in Master mode only a single device may be polled. To poll multiple devices the Address register must be changed by the Quickstep program, dynamically. An address from 1 to 255 is valid with 1 as the default.

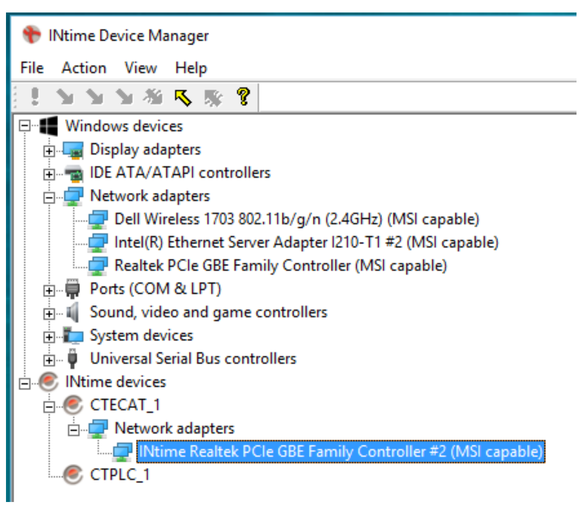
Example:

Set COMM1 19200, 8, None, 1, CTC Binary, None, 1

You can also use 'get COMM[#]' to retrieve the current settings.

EtherCAT Ethernet Adapter

Within the INtime configuration screens you will notice CTECAT_1 for a single network configuration, CTECAT_2 would be added for dual network. These are the real-time processor cores responsible for the EtherCAT Master network. Reference the 'INtime Device Manager' for those available and/or presently being used.

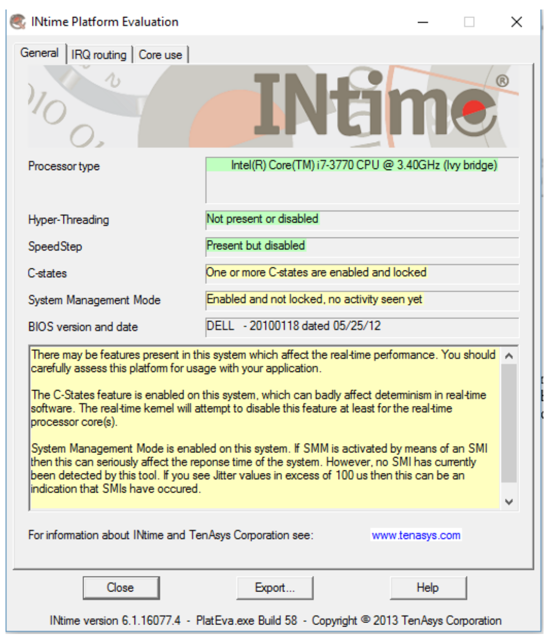


The above configuration shows a Realtek PCIe GBE Controller being used under 'INtime devices.' Referencing 'Windows devices', there are three additional adapters currently used by Windows® which are available. The ideal adapter to use would be the I210-T1 adapter as that offers special high speed queuing and an internal precision timer that is optimum for EtherCAT packet timing. For example the Realtek adapter will typically be about 30 uS average jitter with a max of 200 uS (400 uS dual core) whereas the I210 is typically has sub-microsecond average jitter and significantly offloads the processor allowing for greater number of EtherCAT devices to be supported on a single network.

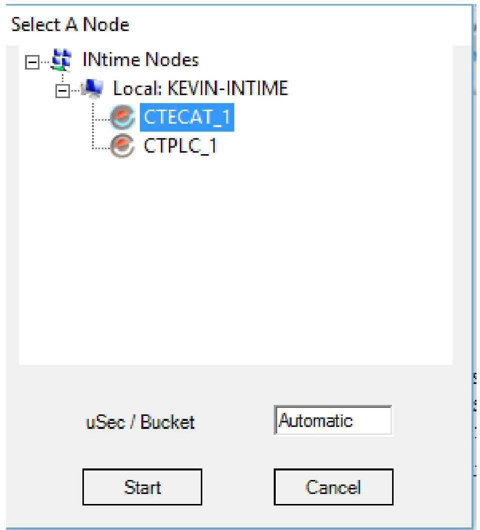
The CTECAT core will automatically detect which adapter is selected and attempt to use it. When mapping an adapter to the INtime environment be sure to select the 'Pass to INtime using MSI' property. Only a single Ethernet adapter is supported by each CTECAT core, where _1 is incremented to __2 when additional networks are used for EtherCAT. As with serial ports, reboot after any assignment changes.

Platform Evaluation

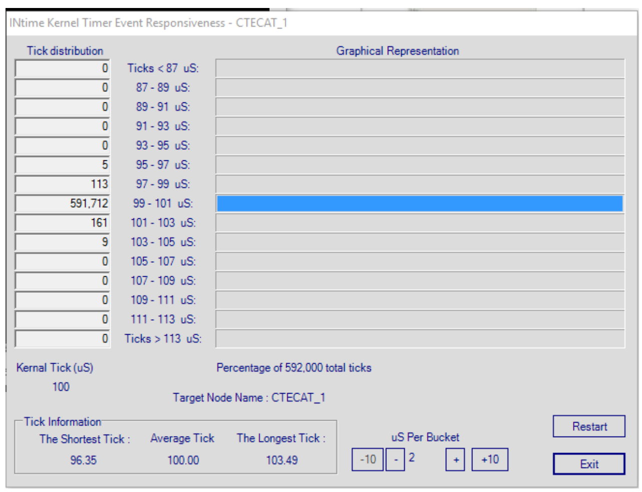
Once the real-time environment is installed, even in demo mode, an actual evaluation of the suitability of that PC can be made by running two programs. The first is the INtime Platform Evaluation utility. This utility will check various BIOS settings and give warnings of things that may need changing. Additionally IRQ routine and how the CPU cores are presently assigned are available via that tabs:

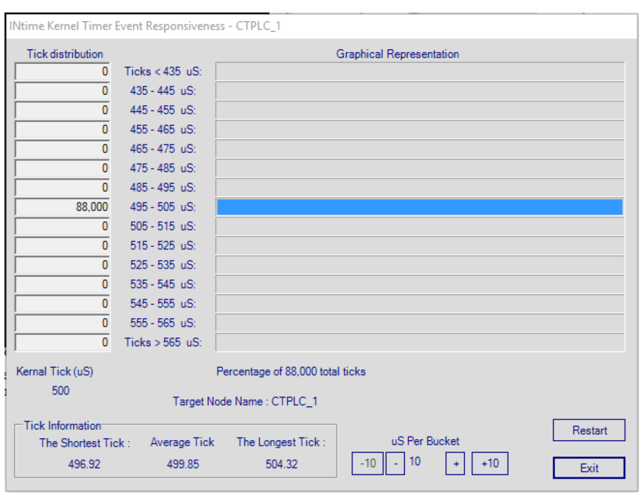


The actual final test is the "Graphical Jitter Display", this test actually checks the ability of the real-time environment to run in real-time, ideally with no jitter caused by SMI's. Upon startup you are presented with the available cores to test (assuming they have been started). Select and test each independently:



The screen should look something like this, with no red indicators showing excessive jitter after you perform the necessary operations on your PC:





CTC Incentive® .Net API

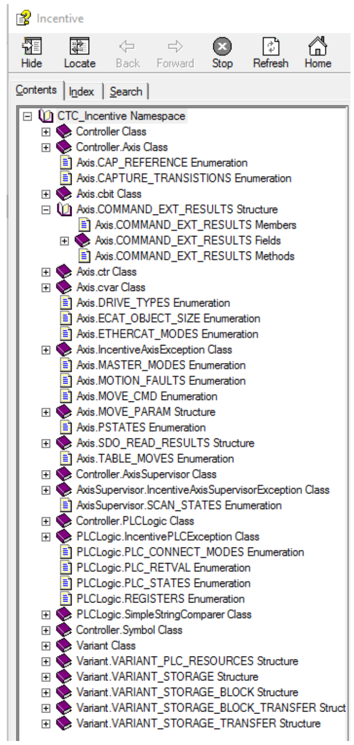
CTC has created a very powerful tool for .Net programmers, a tightly integrated dynamic link library (DLL) interface which allows users to program the PC based controller using C#, managed C++, and VB.Net. This DLL includes the complete MSB motion control language, axis properties, registers, variant storage, as well as the ability to access variable registers and symbolic names, both locally and over a network. It runs standalone or in conjunction with the QuickBuilder application programs where programs can execute in mixed mode with .Net programs controlling one aspect of the automation and QuickBuilder, another.

Typically QuickBuilder would be used where absolute real-time is required whereas .Net can be leveraged to coordinate larger systems, perform database queries, provide vision control interfaces, complex calculations, HMI updates, and/or real-time motion control of some of the axis, limited only by the programmer's imagination. The only difference between QuickBuilder and the .Net interface is that the .Net program will execute under the Windows operating system (non-realtime) whereas QuickBuilder runs in real-time on a dedicated processor core.

Some of the features of the API are as follows:

- Portable .Net DLL, written in C#, abstracting the user from unmanaged code.
- Sample Applications included for C#, VB.Net, and managed C++.
- Connection based high speed parallel threaded interface to both the PLC logic and EtherCAT Master processes.
- Same exact interface whether communicating locally or over a network, simply include the defined remote hostname for CTPLC_1 process when using the API 'openConnection' function in a networked environment. Multiple connections supported, both local and networked.
- Full support and mapping of the QuickBuilder Motion language to .Net. .Net Methods mimic those of the QuickBuilder MSB language helping to shorten learning curves when moving between programming environments.
- Full register interface, including Variant support. Exception based error processing.
- Requests are passed to the real-time processes where they execute using existing MSB instruction objects, thus executes exactly as an MSB would.
- Full access to a QuickBuilder programs' symbol tables for variable reference, both at the QuickBuilder and MSB level. All axis properties are directly supported as well as access to user variables. This allows QuickBuilder programs to be re-compiled, variable registers re-assigned, and still work with .Net since the symbol name is the reference, not the low level register number.
- System state control to restart programs, restart EtherCAT, monitor current execution state, etc. Including managing the starting and stopping of the real time operating system INtime, as well as monitor the status of programs running on its dedicated processor cores.
- Optional standalone EtherCAT Master operation for Windows only programs. Requires only a single core for real-time versus the preferred two when using QuickBuilder applications.

Some of the class features of the CTC_Incentive Namespace (from CTC_Incentive.chm):

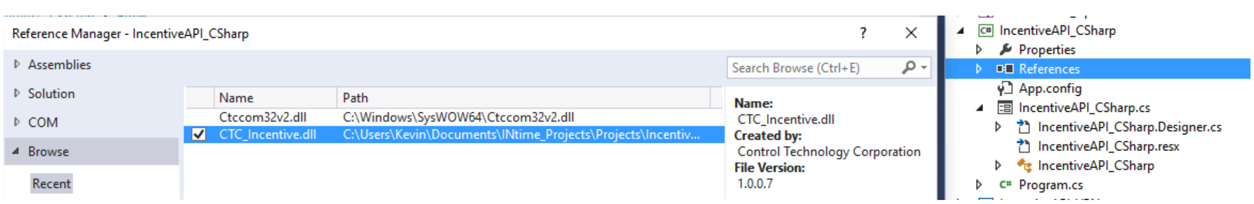


Some examples of supported motion commands:

_fpos	Read feedback position for a specific axis.
_tpos	Read target position for specific axis.
closeConnection	This method closes a connection to the EtherCAT Master runtime.
clrout	Clear drive outputs.
dc_sync	This method requests DC Sync mode with the drive attached to this axis. It does not return until completely executed and the drive
delayMSB	Delays the specified milliseconds using an MSB delay loop.
drive_disable	Disable the drive, turning voltage off to it. Drive will then free wheel and will continue to have its position tracked.
drive_enable	Enables the axis, if not already. This tells the drive to activate voltage and prepare for operation via EtherCAT.
Equals(System.Object)	Determines whether the specified Object is equal to the current Object. (Inherited from Object.)
Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
gear_at	Changes the gear ratio of a slaved axis to the specified values.
gear_at_in	Changes the gear ratio of a slaved axis to the specified values over some number of master counts.
gear_at_in_after	Changes the gear ratio of a slaved axis to the specified values over some number of master counts.
gear_for_in	Temporarily changes the gear ratio of a slaved axis such that a slave_counts correction (offset) occurs over a master-feedback dis
gear_for_in_after	Temporarily changes the gear ratio of a slaved axis such that a slave_counts correction (offset) occurs over a master-feedback dis number of counts.
get_drive_properties	Get drive properties, reads COMMAND_EXT_RESULTS drive properties all at once versus individually.
GetHashCode	Serves as a hash function for a particular type. (Inherited from Object.)
getResources	Gets the resources.
GetType	Gets the Type of the current instance. (Inherited from Object.)
getVar	Gets the variable.
MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
move_at_for(Double, Double, Double, Double)	Move relative, speed limited, trapazoidal.
move_at_for(Double, Double, Double, Double, Controller.Axis.COMMAND_EXT_RESULTS)	Move relative, speed limited, trapazoidal. Extended drive information results.
move_at_for(Double, Double)	Move relative, speed limited, trapazoidal using default acc and dec.
move_at_for(Double, Double, Controller.Axis.COMMAND_EXT_RESULTS)	Move relative, speed limited, trapazoidal using default acc and dec. Extended drive information
move_at_to(Double, Double, Double, Double)	Move absolute, speed limited, trapazoidal.
move_at_to(Double, Double, Double, Double, Controller.Axis.COMMAND_EXT_RESULTS)	Move absolute, speed limited, trapazoidal. Extended drive information results.
move_at_to(Double, Double)	Move absolute, speed limited, trapazoidal, use default acc and dec.
move_at_to(Double, Double, Controller.Axis.COMMAND_EXT_RESULTS)	Move absolute, speed limited, trapazoidal, use default acc and dec. Extended drive information results.
move_for(Double, Double, Double)	Move relative, triangular.
move_for(Double)	Move relative, triangular. Uses default acc and dec.
move_in_for(Double, Double, Double)	Move relative, trapazoidal, in time requested with acc/dec ramp multiplier.
move_in_for(Double, Double)	Move relative, trapazoidal, in time requested.
move_in_to(Double, Double, Double)	Move absolute, trapazoidal, in time requested with acc/dec ramp multiplier.
move_in_to(Double, Double)	Move absolute, trapazoidal, in time requested.
move_to(Double, Double, Double)	Move absolute, triangular.
move_to(Double)	Move absolute, triangular. Uses default acc and dec.
move_trap_for	Move relative, trapazoidal 1/3 acc, 1/3 constant, 1/3 dec.
move_trap_to	Move absolute, trapazoidal 1/3 acc, 1/3 constant, 1/3 dec.
new_endposition	Modifies the end point of an active move command to an absolute position. If not moving then ignored.
new_endposition_relative	Modifies the end point of an active move command relative to current position. If not moving then ignored.
offset_position	Modify the target and feedback positions simultaneously by adding the offset to both.
offset_position_counts	Modify the target and feedback positions simultaneously by adding the offset to both.
offset_slave_by	Offsets the position (and therefore phase) of the axis such that a 'slavecounts' correction (offset) occurs over a period
openConnection	This method opens a connection to the EtherCAT Master runtime for motion control.
pulse	Pulse the specifed output for the desired number of milliseconds.
sdo_read	Read an object property of the drive over EtherCAT. This call is blocking and can take about 7 milliseconds to execute.
sdo_write	Write an object property to the drive over EtherCAT. This call is blocking and can take about 7 milliseconds to execute.
segmove_accdec_to_for	Adds an acc/dec segment from the current velocity to the new 'velocity' over some 'displacement'.
segmove_accdec_to_using	Adds an acc/dec segment from the current velocity to the new 'velocity' at the specified 'rate'.
segmove_clear	Clears the specified segment table.
segmove_slew_until_position	This method adds a constant velocity segment until reaching some specified absolute 'position' from the start of the p movement before this command is accedpted otherwise an error will occur.
segmove_start_relative	Starts a relative segmented move, 'zero feedback position' occurs automatically upon executing this command.
segmove_stop_at	Stops motion at the specified 'position', with a given 'rate'. This will cause motion to stop at an absolute position at the table must represent movement before this command is accepted.
set_capture(Controller.Axis.CAPTURE_TRANSISTIONS, Int32, Int32,	Initializes the parameters to be used for all captures on this axis, specifying the input to use and the optional gated in

	set_capture(Controller.Axis.CAPTURE_TRANSISTIONS, Int32)	Initializes the parameters to be used for all captures on this axis, specifying the input to use.
	set_capwin_range	Initializes a window to be monitored for valid captures to occur, anything outside this window is considered invalid and ignored.
	set_common_bit	Set_common_bits the specified number.
	set_common_var	Sets the common var, 1 to 255 to the specified value.
	set_feedback_position	Set feedback position.
	set_master	Sets the source of the master axis encoder.
	set_mode	Set desired operating mode of the axis, positioning or tracking.
	set_simulated_feedback	Set simulated feedback position mode (true/false).
	set_target_position	Set target position.
	setout	Set drive outputs.
	setVar	Sets the variable.
	slew_at_in	Alters the current slew velocity, changing smoothly over the specified time. For immediate change set time to 0.0.
	slew_begin	Changes the operating mode of the axis to slewing.
	slew_end	Changes the operating mode of the axis to positioning from slewing.
	slew_for	Alters the current slew velocity, over time, to a slew velocity of 0.0 such that some displacement is consumed.
	stop()	Stops the axis quickly.
	stop(Single)	Stops the axis at the specified slew rate.
	table_clear	Clear the specified table of all entries.
	table_continue	Continues a cam table that was stopped by the 'stop table' command. Note that this command should only be used if the master positi next table row position otherwise any row that is currently being executed during a 'stop table' will be re-executed.
	table_precompute	Readies a table for use by a spline/CAM motion. After points have been added to a table, there are a series of computations that need utilized for spline and CAM motion operations.
	table_start	Starts spline motion using the specified table.
	table_start_cam	Starts CAM motion using the specified table.
	table_stop	Stops spline or CAM motion. If in CAM motion then the current table state is saved in case a 'table continue' command is executed.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	wait_capture	Waits for the capture and arms the capture input. If the capture occurs control will be returned else returns after the specified 'limit'.
	wait_common_bit	Wait for a common bit to be true or false.
	wait_common_var(Int32, Int32)	Wait for a common var to be a certain value. This method will wait until the condition is true to get a message back.
	wait_common_var(Int32, Boolean, Int32, Int32)	Wait for a common var to be in or out of range. This method will wait until the condition is true to get a message back.
	wait_for_in_pos	Wait for motion in position (inpos == 1) or timeout.
	wait_master	Waits until the specified number of master encoder counts has been generated.

In order to access the features of the CTC Incentive API simply add the CTC_Incentive.dll resource to your Visual Studio project (Visual Studio 2013 or greater). This is done by right clicking on your project in the Solution Explorer, followed by selection of Add->Reference. Once the Reference Manager appears select Browse and browse to where you installed the dll, typically "C:\Program Files\Control Technology Corporation\CTC_IncentiveAPI\CTC_Incentive.dll", make the selection and it will be added as a Reference to your .Net project. Example projects are installed with the dll installation for your review. Reference the "CTC Incentive® .Net API User Manual" for additional details (chm help file format).



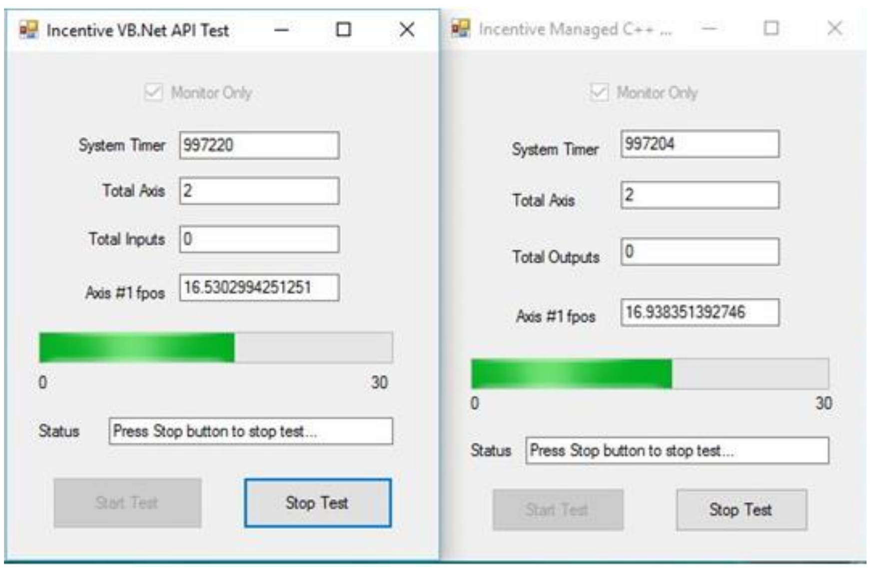
.Net API Sample Program Overview

The sample programs are available in three different languages; C#, VB.Net, and managed C++, reference the "Incentive_API_Projects.mp4" video. The Monitor checkbox, when checked, allows the program to attach to the Incentive PLC environment in a passive, monitor only mode. When not checked, any QuickBuilder program will be shutdown, EtherCAT network reset and the sample program will take control of the axis motion. Only one instance should be run with the checkbox unchecked.

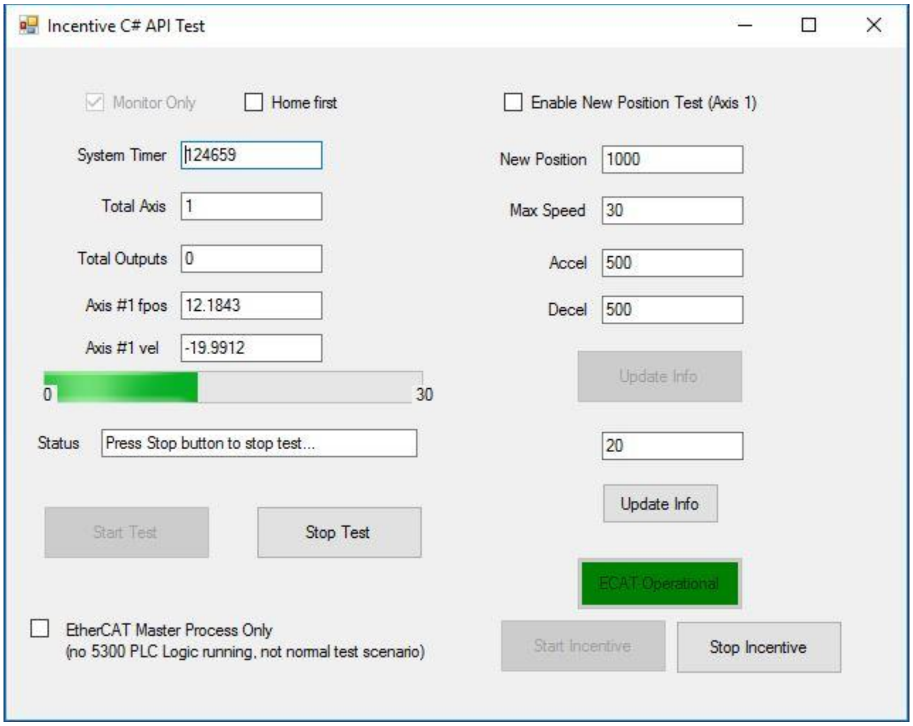
Upon selection of the 'Start Test' button the program will first attempt to connect to the Incentive PLCLogic process, CTPLC_1. Once connected it will check to see if 'Monitor Only' is selected. If not selected any QuickBuilder program will be stopped and the EtherCAT network reset. Next the available resources will be checked and a connection as well as a thread spawned for each axis found. A PLCLogic monitor thread will also be spawned to periodically display register 13002, system tick, as well as the network status register.

If not monitoring, each thread will initialize the servo drive for operation and begin a back and forth move, displaying the axis #1 feedback (fpos) position in a textbox. Either the 'Stop Test' button or the 'X' at the upper right of the GUI form will cause the threads to stop and connections to be released.

The sample programs themselves all perform similar operations, with a few extra in the C# example, but in different programming languages. Only one test program should control the axis at a time, the others can execute in Monitor Only mode. If you wish more than one to run the axis then changes must be made to ensure the EtherCAT network is not reset while the other application is running, else an error will result. Below shows three application test programs, one for each language monitoring the PLC system timer and axis 1.



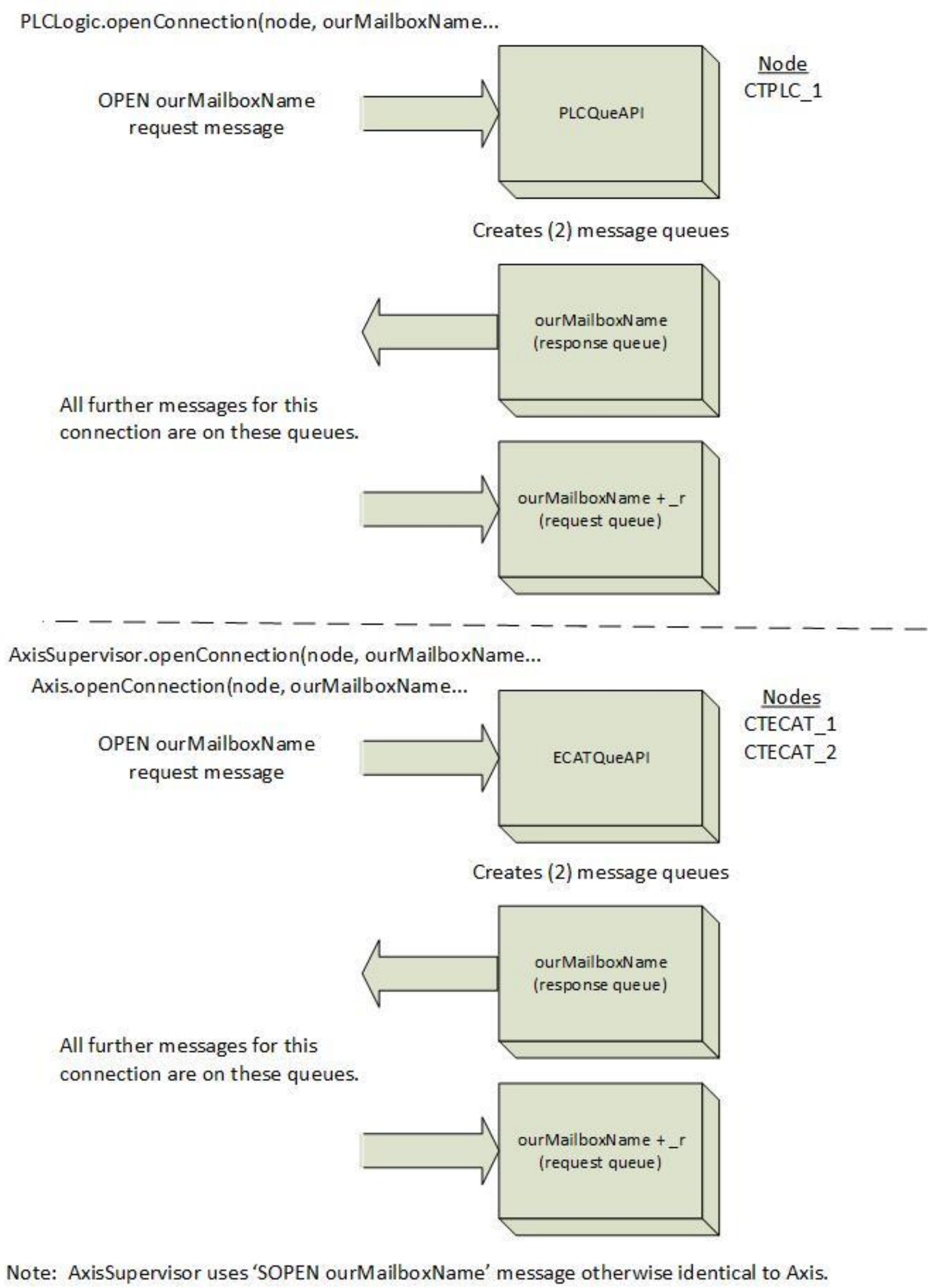
C# has been enhanced beyond that of VB.Net and C++ to demonstrate such features as running in Standalone mode (EtherCAT Master only), homing, new position commands, and starting/stopping the Incentive environment:



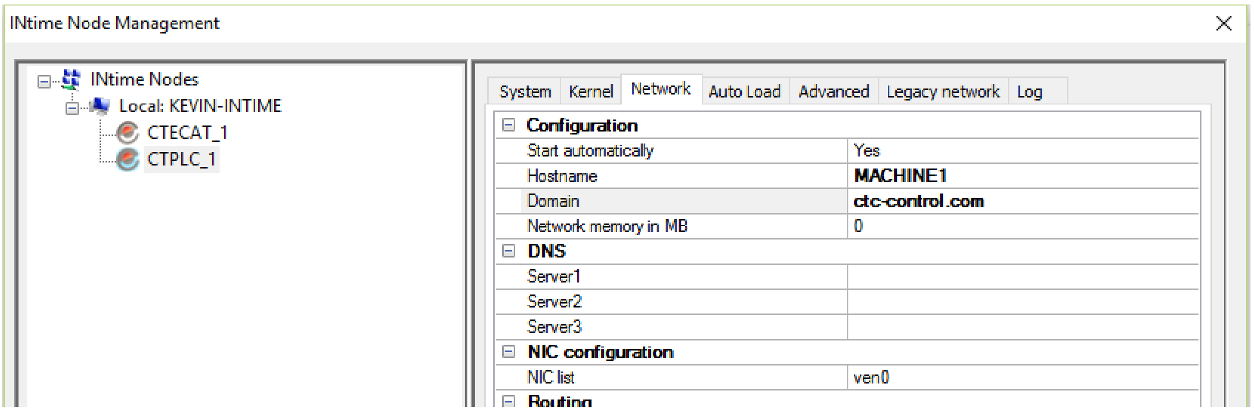
C# has the same basic tests of the other programming languages with the additions of some of the enhancements discussed. Start/Stop of Incentive realtime environment is controlled by the buttons on the lower right of the form, with the status now displaying in green, “ECAT Operational”. In an EtherCAT only environment (usually controlled directly by a high level Windows application), the checkbox on the lower right of the form should be checked. This allows the logic to then connect to an AxisSupervisor class to start and stop EtherCAT.

.Net API Opening a Connection Locally and Remote

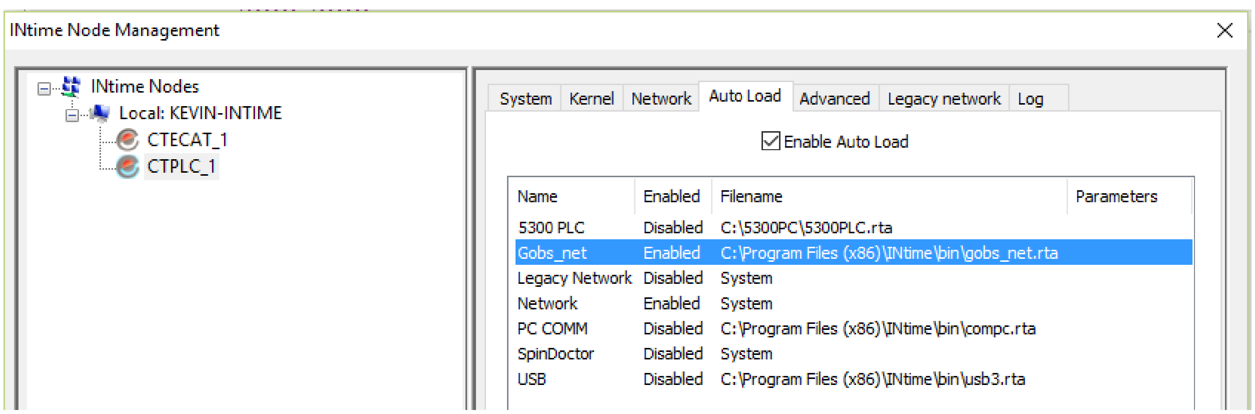
In order to establish a connection with a runtime process the ‘openConnection’ function is invoked. This function creates two private high speed mailboxes within the Incentive runtime with which to communicate, one for sending messages and one for receiving. The parameters passed to the ‘openConnection’ differ slightly for the PLC Logic runtime, CTPLC_1, and that of the EtherCAT Master, CTECAT_1. The first parameter is the node name and the second mailbox name, common for all ‘openConnection’ functions. The mailbox name must be unique on every node for every connection since the name is published globally for each node. Different nodes can have the same mailbox name since the mapping includes the node name when addressing a mailbox.



When running the API locally only the node name of CTPLC_1 or CTECAT_1 is used. In order to communicate over a network the node name is expanded to include the unique Hostname assigned to CTPLC_1 within the INtime Node Management configuration form. Thus for a local connection to the PLC logic process (QuickBuilder program), the parameter would be CTPLC_1. For a remote host it would be Hostname plus any domain name entered in the configuration form or returned by DHCP followed by /CTPLC_1. For example if the Hostname was MACHINE1 with no domain information the parameter passed to 'openConnection' would be 'MACHINE1/CTPLC_1'. With a domain of ctc-control.com it would become 'MACHINE1.ctc-control.com'/CTPLC_1. Note that a Hostname must be passed to 'openConnection' not a raw IP address. Also if you find that using the Hostname does not work check your network as some systems automatically assign a domain during the DHCP process. You can usually find this by opening a command window and typing ipconfig. The domain is listed under the first item: “Connection-specific DNS Suffix”. Below is an example of setting the name for an Incentive node.



In order for automatic remote access to work a driver must also be loaded called ‘gobs_net.rta’. This driver is responsible for all connections to both CTPLC_1 and CTECAT_1 and will only be enabled and loaded on the node CTPLC_1. The driver will automatically create a shared memory path to CTECAT_1. Note that if there is a Gobs_net entry that says System under Filename you may use that only if the Network tab Start Automatically is set to Yes. By default we set this to No during installation and thus requires a manual entry be placed in the Auto Load table for Gobs_net, as shown below. If there is one already there that is a System type then just name your Gobs_net_new or something similar.



Gobs_net uses UDP traffic on port 48271 for communication and broadcasts of Hostname information and status periodically, therefore not relying on a DNS server or dhcp for name registration. Due to the high volume of UDP traffic it is advisable to isolate groups of computers using Gobs_net from normal LAN traffic. For example you would not want 100 computers on the same network running the peer to peer communications without using a switch to isolate some of the traffic.

In order for a remote Visual Studio project to be able to communicate remotely to an Incentive environment the standard Incentive installation kit must be installed. This provides the API dll, CTC_Incentive.dll, as well as the INtime host runtime (host61 or host63 installer) that will need to be installed and patched (no license required). Located in your Control_Technology_Corporation\INtime subdirectory, 6.1 and 6.3 host & runtimes cannot be mixed on the same network since changes made to 6.3 are not backward compatible.

There is a property for PLC connections of Controller_Timeout_ms and one for the EtherCAT Master connection of Command_timout_ms. Each of these should be set to 5000 or greater when using the network based API. Failure to do so can cause E_TIMEOUT errors.

Simple API Programming Concepts

PLClogic Class

All API transactions typically begin with an openConnection call (reference the Incentive.chm help file). This establishes the communication queues between the non-realtime Windows environment and the realtime Incentive. As detailed in the previous section you must specify a node name and desired unique mailbox name.

C# VB C++

```
public bool openConnection(
        string node,
        string our_mailbox_name,
        bool get_resources,
        bool get_symbols
)
```

Parameters

node
Type: String
PLC runtime name, typically CTPLC_1.

our_mailbox_name
Type: String
Base name of our queue, limited to 8 characters.

get_resources
Type: Boolean
True if want controller resources loaded immediately after connection.

get_symbols
Type: Boolean
True to request controller symbols be loaded immediately after connection.

Return Value

True if successful, false if failed.

The PLCLogic class is used for communications with the QuickBuilder Logic environment. There you will primarily being accessing registers and variants. Both are where QuickBuilder programs store their program information. Registers are of type Integer and can also be used to access IO data and certain aspects of operation. Reference the Model 5300 Quick Reference Register Guide: http://support.ctc-control.com/customer/techinfo/docs/5300_951/951-530006.pdf. Variants store any type of information and can also consist of one and two dimensional tables, reference the QuickBuilder Reference Guide: http://support.ctc-control.com/customer/techinfo/docs/5300_951/951-530020.pdf.

For the PLCLogic class you can also load the QuickBuilder symbol table for symbolic name access, ‘get_symbols’. Typically direct numeric register access is used to expedite reads since symbols tend to slow performance. The ‘get_resources’ optionConnection option is useful to find out how much IO is present in the system before interacting with it. In a local environment the node name is always 'CTPLC_1', with a unique 'our_mailbox_name':

```
using CTC_Incentive;
…

   Controller.PLCLogic testplc = new Controller.PLCLogic();
   int value = 0;
  try
  {
     if (testplc.openConnection("CTPLC_1", "PQ001", true, false))
    {
          // Connection established… Now do something, lets read system tick register 13002.
          testplc.getRegister((int)Controller.PLCLogic.REGISTERS.MILLISECOND_COUNTER, ref value);
          // 'value' now has the tick value read.

          // To poke the value of 1 into register 5 you could do the following.
          testplc.putRegister(5, (int)1);
    }
  }
  catch (Controller.PLCLogic.IncentivePLCException e2)
  {
          // Error processing of Incentive specific error.
          MessageBox.Show("Error occurred:  " + e2.ErrMessage);
  }
  catch (Exception)
  {
         // Handle other errors...
  }

  // When all done and ready to exit your program you should close the connection.
 testplc.closeConnection();
```

Each thread needs its own private connection; multiple connections can be made for parallel operations.

Beyond simple integer register storage there is also something called Variants. Variants can automatically assume the types you wish them do. They can be integer, double, float, and/or string. As defined in the 5300 PLC manual, certain register blocks have certain storage capabilities.

1 – 500: volatile integer registers
501-1000: non-volatile integer registers
32001-36000: non-volatile integer registers
36101 – 36700: 600 volatile Variant registers
36701 – 36800: 100 non-volatile Variant registers

Non-volatile registers are stored on the disk, and thus the importance of a UPS, so the file storage can be updated and closed when written to. Also the NTFS file system should be used, not FAT32.

Variants not only can be of any type they also can be one (vector) and two dimensional (table) arrays. The getRegister and putRegister methods have optional parameters to handle this, where the basic call is the same as previously described. The enhanced method lists the row, column, and precision:

PLCLogic.getRegister Method (Int32, Int32, Int32, Byte, Double)

This method reads a 'double' from a Variant cell.

Namespace: CTC_Incentive
Assembly: CTC_Incentive (in CTC_Incentive.dll)

Syntax

C# | VB | C++

```
public bool getRegister(
        int regnum,
        int row,
        int col,
        byte precision,
        ref double value
)
```

Parameters

regnum
Type: Int32
Register number to access.

row
Type: Int32
Row, X value if array.

col
Type: Int32
Column, Y value if array.

precision
Type: Byte
Number of decimal precision.

value
Type: Double
Reference to where to store what read.

Return Value

Thus to read a double from a table, 36105[3][5], register 36105, row 3 and column 5:

```
double dValue = 0;
testplc.getRegister(36105, 3, 5, 6, ref dValue);
```

The precision by default is set to 6. Precision is only use on a read operation where a string is involved and defines the number of decimal places for the conversion. Therefore doing the same operation but in this case reading the data as a string where the double value was 26.157604532678:

```
string sValue = 0;
testplc.getRegister(36105, 3, 5, 6, ref sValue);
```

After execution the string 'sValue' would contain "26.157604".
Writing to a Variant or integer is similar to reading except instead of reference the value, it is passed to the method:

```
string sValue = "26.157604";
testplc.putRegister(36105, 3, 5, 6, sValue);

float fValue = 2.6753;
testplc.putRegister(36105, 3, 5, 6, fValue);
```

Axis Class

Since Incentive consists of multiple realtime processes and threads we may also have a requirement to access the EtherCAT environment directly. Possibly to simply read servo position, interact with MSB (QuickBuilder Motion Sequence Blocks), or totally control motion from within our Windows program. Like the PLCLogic class there is the Axis class. The Axis class contains the entire MSB program language, mapped to Windows, as well as access to all motion and EtherCAT IO variables. An open connection is required as well to establish direct communications with the EtherCAT process node, CTECAT_1 (CTECAT_2 would be a second parallel network if used).

Axis.openConnection Method

This method opens a connection to the EtherCAT Master runtime for motion control.

Namespace: CTC_Incentive
Assembly: CTC_Incentive (in CTC_Incentive.dll)

Syntax

C# | VB | C++

```
public bool openConnection(
        string node,
        string our_mailbox_name
)
```

Parameters

node
Type: String
EtherCAT runtime name, typically CTECAT_1.

our_mailbox_name
Type: String
Base name of our queue, limited to 8 characters.

Return Value

true if successful, false if failed.

In a local environment the node name is always 'CTECAT_1', with a unique 'our_mailbox_name':

```
using CTC_Incentive;
...

   Controller.Axis testAxis = new Controller.Axis();
   int value = 0;
  try
  {
     if (testAxis.openConnection("CTECAT_1", "MQ001"))
    {
           // Connection established...  Now do something, read the current feedback position.
           double ourPos = testAxis.fpos;

           // Let spin the motor 100 revolutions greater than present position.
           // Velocity is 30 revs/sec, acceleration 50 revs/sec2, deceleration 250 revs/sec2.
           testAxis.move_at_to(30, ourPos + 100, 50, 250);

           // Lets wait until we are in position.
           testAxis.wait_for_in_pos(-1);
    }
  }
  catch (Controller.Axis.IncentiveAxisException e2)
  {
          // Error processing of Incentive specific error.
          MessageBox.Show("Motion error occurred:  " + e2.ErrMessage);
  }
  catch (Exception)
  {
         // Handle other errors...
  }

  // When all done and ready to exit your program you should close the connection.
 testAxis.closeConnection();
```

Each thread needs its own private connection; multiple connections can be made for parallel operations. Note that each command runs as its own simulated MSB.

The Axis move_at_to command was introduced above and is described in a bit more detail below:

Axis.move_at_to Method (Double, Double, Double, Double)

Move absolute, speed limited, trapazoidal.

Namespace: CTC_Incentive
Assembly: CTC_Incentive (in CTC_Incentive.dll)

Syntax

C# | VB | C++

```
public bool move_at_to(
	double max_velocity,
	double position,
	double acc,
	double dec
)
```

Parameters

max_velocity
Type: Double
Maximum velocity to attain, if not possible becomes a triangular move, user units/sec.

position
Type: Double
Absolute end position, user units.

acc
Type: Double
Acceleration rate, user units/sec/sec.

dec
Type: Double
Deceleration rate, user units/sec/sec.

Return Value

true if successful, false if failed.

RuntimeManagement Class

The Incentive API uses a realtime operating system created by TenAsys called INtime. The API contains a system management class called RuntimeManagement. This class can be used to start and stop the INtime nodes as well as determine what state they are in with regards to EtherCAT being executed.

The updateCurrentStateInformation method updates the class properties EtherCAT_started, Plclogic_started, and SameCore with the proper current values:

RuntimeManagement Constructor

Initializes a new instance of the RuntimeManagement class.

Namespace: CTC_Incentive
Assembly: CTC_Incentive (in CTC_Incentive.dll)

Syntax

C# | VB | C++

```
public RuntimeManagement(
	int inst
)
```

Parameters

inst
Type: Int32
Instance of node, 1 or 2, CTPLC/CTECAT_1 or CTECAT_2.

RuntimeManagement Properties

The RuntimeManagement type exposes the following members.

Properties

	Name	Description
	EtherCAT_started	Gets a value indicating whether EtherCAT process was started.
	Plclogic_started	Gets a value indicating whether QB PLC Logic process was started.
	SameCore	Gets a value indicating whether a single core is being used.

RuntimeManagement.updateCurrentStateInformation Method

Updates the online information status information in case Incentive is already running. Best to call before attempting to start in case was set to automatically start when Windows booted.

Namespace: CTC_Incentive
Assembly: CTC_Incentive (in CTC_Incentive.dll)

Syntax

C# | VB | C++

```
public void updateCurrentStateInformation()
```

Return Value

`true` if successful, `false` otherwise.

The start/stop methods are equally as simple:

RuntimeManagement.startIncentive Method

Starts the Incentive nodes for the desired instance (typically 1). Auto start for the processes must be set with the INtime Configurator.

Namespace: CTC_Incentive
Assembly: CTC_Incentive (in CTC_Incentive.dll)

Syntax

C# | VB | C++

```
public bool startIncentive(
        int maxTimeout_ms,
        bool EtherCAT_only,
        bool sameCore
)
```

Parameters

maxTimeout_ms
Type: Int32
Maximum time to wait for starting each node before giving up, minimum 1000 ms, 5000 ms recommended.

EtherCAT_only
Type: Boolean
if set to `true` only start EtherCAT process.

sameCore
Type: Boolean
if set to `true` start both processes on CTECAT.

Return Value

`true` if started, `false` otherwise.

RuntimeManagement.stopIncentive Method

Stops the Incentive processes for the desired instance (typically 1).

Namespace: CTC_Incentive
Assembly: CTC_Incentive (in CTC_Incentive.dll)

Syntax

C# | VB | C++

```
public bool stopIncentive(
        int maxTimeout_ms
)
```

Parameters

maxTimeout_ms
Type: Int32
Maximum time to wait for stopping on each node before giving up, minimum 500 ms, 3000 ms recommended.

Return Value

`true` if stopped, `false` otherwise or timed out.

```
using CTC_Incentive;
```

…

```
Controller.RuntimeManagement RuntimeSystem = new Controller.RuntimeManagement(1);
// Update the latest node execution information.
RuntimeSystem.updateCurrentStateInformation();
If (Plclogic_started == false)
{
    // Start the INtime kernel on their nodes with a 5 second timeout.
    RuntimeSystem.startIncentive(5000);
}
```

Selective Offline and Online of EtherCAT Devices (Beta Only)

The Incentive API contains methods to support the setting of slave devices to an online or offline state within a fully active network. This feature is also called 'Offlining'. It can be useful when sections of your machine needs to be taken offline for maintenance or when a device needs to be replaced without shutting the EtherCAT network down.

Without any special provisions EtherCAT devices are daisy chained, into one device, out and into the next. This means once you break the daisy chain all following devices will not respond. If a device does not respond it is an error condition to Incentive and a network fault will occur. The way around this is to notify Incentive the device is being taken offline and allow it to prepare for it.

Certain guidelines must be followed to have this work properly and prevent devices from faulting or gaps in EtherCAT cabling causing packet loss:

1. ***All Online at Startup*** - It is important that when the EtherCAT network is first brought online, all devices are present. This allows Incentive to establish the maximum required packet size for communications and the data placement for each within a packet.
2. ***Offline all following devices*** – If removing a device you must offline that device and all following in the daisy chain before performing service. This is only required if the device is being turned off.
3. ***Omron EtherCAT Junction GS-JC06*** – It is required to use an Omron EtherCAT junction as the first slave device. This device provides up to 5 independent EtherCAT segments as well as a 64 bit clock for distribution. If you disconnect a slave device on one of the segments the network can continue to operate the other segments. The same rules apply with daisy chaining, all devices after the device you are powering off must be offline as well or an error will result, with the EtherCAT junction you have up to 5 independent segments instead of just one.
4. ***Only one segment*** – Only one segment of daisy chained devices can be offline at a time. These devices must then all be brought back online before the next segment is offline. This allows Incentive to ensure all devices are responding correctly prior to shutting down another segment of devices. It also allows Incentive to recover from bad onlining requests when the device is not powered up.
5. ***A/B Ethernet switch*** – An A/B Ethernet switch must be placed on the output of the EtherCAT Junction. Once the devices are offline the switch is deactivated, disconnecting the segment from the network. Offline devices can then be power cycled as needed. The switch is needed to keep power cycling from corrupting the EtherCAT network. A requirement of EtherCAT is the device is not supposed to activate its link until it is ready to forward packets, regrettably many devices do

not follow this requirement and network packets can be lost if the cable is not physically disconnected during power cycling. Once the segment is serviced and the devices are powered back on, the A/B switch can once again be enabled and devices placed back online. The A/B switch is only needed if devices are going to be fully power cycled.

When a device is offline it is placed in its EtherCAT INIT state, outputs (off) do not change or work and data is not available from the device. It does replicate EtherCAT packets to the next device. It is only when you power off the device or disconnect the EtherCAT cable that communications is affected. Normally each slave device increments a counter in the packet, by warning Incentive you are disconnecting devices it will know how many counts are allowed to be missing in the packet before it is a real error.

AxisSupervisor Class – online/offline Methods

The AxisSupervisor Class is used to interface with EtherCAT at a system level. API functionality such as restarting an EtherCAT network or placing a slave device online or offline is available. These methods are as follows:

AxisSupervisor.offlineSlave Method

Offline the requested EtherCAT Slave device. This will mask its inputs so they are 0.

Namespace: CTC_Incentive
Assembly: CTC_Incentive (in CTC_Incentive.dll)

Syntax

C# | VB | C++

```
public bool offlineSlave(
        int slave,
        bool opt_slew
)
```

Parameters

slave
Type: Int32
The EtherCAT slave number that is assigned to the device.

opt_slew
Type: Boolean
Slew option if slave is an axis device, false means immediate stop else will decelerate based on value of 'stoprate' axis property.

Return Value

`true` if offline, `false` otherwise. Alternatively IncentiveAxisSupervisorException if fails where ErrCode is Axis.MOTION_FAULTS..

AxisSupervisor.offlineSlaves Method

Offline the requested sequence of slaves in the order given.

Namespace: CTC_Incentive
Assembly: CTC_Incentive (in CTC_Incentive.dll)

Syntax

C# | VB | C++

```
public bool offlineSlaves(
        int[] slaves,
        int num_slaves,
        bool opt_slew,
        ref int lastSlave
)
```

Parameters

slaves
Type: Int32[]
The array containing the sequence of slaves to offline.

num_slaves
Type: Int32
The number slaves in the array.

opt_slew
Type: Boolean
Slew option if slave is an axis device, false means immediate stop else will decelerate based on value of 'stoprate' axis property.

lastSlave
Type: Int32
The last slave device attempting to offline is stored here for reference should an error occur.

Return Value

`true` if offline, `false` otherwise. Alternatively IncentiveAxisSupervisorException if fails where ErrCode is Axis.MOTION_FAULTS.

AxisSupervisor.offlineAxis Method

Offline the requested axis

Namespace: CTC_Incentive
Assembly: CTC_Incentive (in CTC_Incentive.dll)

Syntax

C# | VB | C++

```
public bool offlineAxis(
        int axis,
        bool opt_slew
)
```

Parameters

axis
Type: Int32
The axis that should be offline, 1 to N.

opt_slew
Type: Boolean
Slew option, false means immediate stop else will decelerate based on value of 'stoprate' axis property.

Return Value

`true` if offline, `false` otherwise. Alternatively IncentiveAxisSupervisorException if fails where ErrCode is Axis.MOTION_FAULTS.

AxisSupervisor.onlineSlave Method

Online the requested EtherCAT device. The inputs will begin returning valid data.

Namespace: CTC_Incentive
Assembly: CTC_Incentive (in CTC_Incentive.dll)

Syntax

C# | VB | C++

```
public bool onlineSlave(
        int slave,
        bool restartMSB
)
```

Parameters

slave
Type: Int32
The EtherCAT slave number that is assigned to the device.

restartMSB
Type: Boolean
If true, and slave is an axis with an MSB that existed when placed offline, the first MSB will be restarted.

Return Value

`true` if online, `false` otherwise. Alternatively IncentiveAxisSupervisorException if fails where ErrCode is Axis.MOTION_FAULTS.

AxisSupervisor.onlineSlaves Method

Online the requested sequence of slaves in the order given.

Namespace: CTC_Incentive
Assembly: CTC_Incentive (in CTC_Incentive.dll)

Syntax

C# | VB | C++

```
public bool onlineSlaves(
	int[] slaves,
	int num_slaves,
	bool restartMSB
)
```

Parameters

slaves
Type: Int32[]
int array of slave node number to online.

num_slaves
Type: Int32
The number of slaves nodes in the array.

restartMSB
Type: Boolean
If true, and slave is an axis with an MSB that existed when placed offline, the first MSB will be restarted.

Return Value

`true` if online, `false` otherwise. Alternatively IncentiveAxisSupervisorException if fails where ErrCode upper 16 bits is slave number and lower 16 bits is Axis.MOTION_FAULTS. If slave number is 0 then not valid, was a timeout or similar error.

AxisSupervisor.onlineAxis Method

Online an axis that was offline temporarily.

Namespace: CTC_Incentive
Assembly: CTC_Incentive (in CTC_Incentive.dll)

Syntax

C# VB C++

```
public bool onlineAxis(
        int axis,
        bool restartMSB
)
```

Parameters

axis
Type: Int32
The axis that should be online and initialized, 1 to N.

restartMSB
Type: Boolean
If true, and an MSB existed when placed offline, the first MSB will be restarted.

Return Value

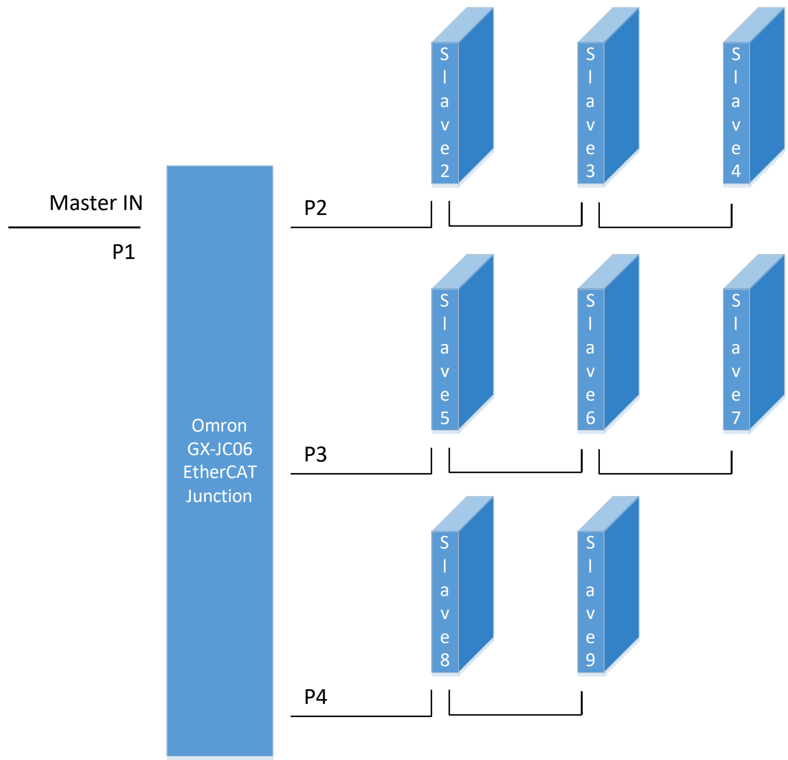
`true` if online, `false` otherwise. Alternatively IncentiveAxisSupervisorException if fails where ErrCode is Axis.MOTION_FAULTS..

If you would like to know the slave # of a particular axis you can reference that axis property dwSlaveID. Additionally, if you would like an axis to always be assigned to a specific drive reference the 'Station Alias' capability in chapter 3 of this manual. Be careful when you are replacing a drive and are using the alias feature, don't install a new drive with the wrong alias or unknown motion could occur with more than one drive assigned the same axis number.

Onlining is device dependent and only should be used in a production environment after extensive testing. Cabling and topology can greatly affect your installation.

Offline/Online Example

Below is an example of a network with the first slave device being the Omron EtherCAT Junction, GX-JC06. The slave devices are daisy chained on each port, 5 are available, each acting as a separate network segment. An A/B switch would exist on each segment, P2, P3, and P4.



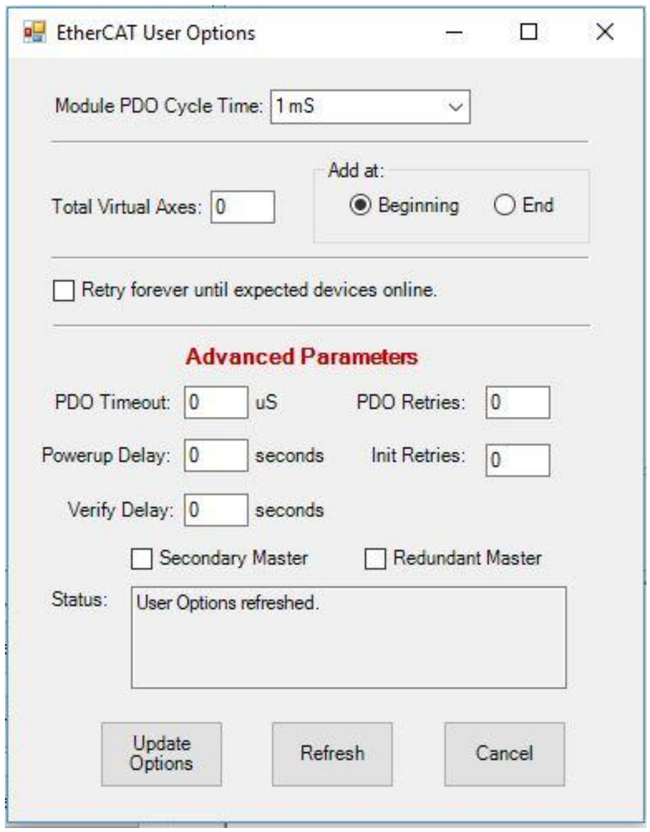
With the example configuration we could execute the offline on each slave in a group (P2, 3, or 4), followed by disconnecting the network with the A/B switch, and then powering down that group for service. To bring it back online simply power the devices back up, once stable reconnect the A/B switch, and then execute the online method.

```
int[] slaves = {5, 6, 7};
int lastSlave = 0;
offlineSlaves(slaves, 3, true, ref lastSlave);  // Offline the 3 slave devices, slew first if axis.

… EtherCAT cable disconnected via A/B switch, power down offline devices…
… service is performed…
…power up offline devices, connect EtherCAT cable once power is stable…

onlineSlaves(slaves, 3, true);  // Online the 3 slave devices and restart any first MSB.
```

Due to the sequence in which a string of devices can be powered on and fluctuations in power the network must be disconnected (Ethernet A/B switch) prior to turning off downstream devices and only re-connected after the devices have stable power. Failure to cycle power in this manner can cause packet loss and faults on the main EtherCAT network.

Master Redundancy

In some applications redundancy is important. Incentive currently supports both a Secondary Master (warm backup) and a Redundant Master (or Ring) simply by selecting the proper user option within the QuickBuilder EtherCAT Explorer.



Secondary Master (Warm Backup)

A warm backup is when two EtherCAT Masters are placed on the same network, using different embedded PC's or a mix of an embedded PC and a 5300 PLC with an M3-41A EtherCAT Master module. One master is designated a secondary master via the QuickBuilder EtherCAT Explorer Options dialog, by default the other is the primary. Upon enabling each will listen for the other and if not seen on the network it will take control and become the EtherCAT Master. Should one fail the other will automatically reset and restart the network in a few seconds. It is up to the user to sync their application software to know how to handle recovery. Once the failing master has been replaced it can be powered on and will once again monitor for any failures, seizing control should it occur.

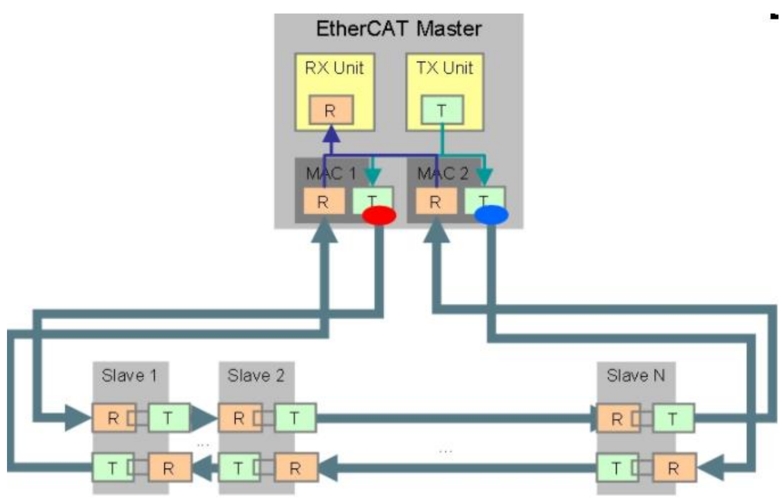
This is a warm restart which means the system will undergo a reset and restart. It is not possible to detect failure in a running system and keep the system running since it is impossible to know the trajectory of running servo motors.

Redundant Master (Ring)

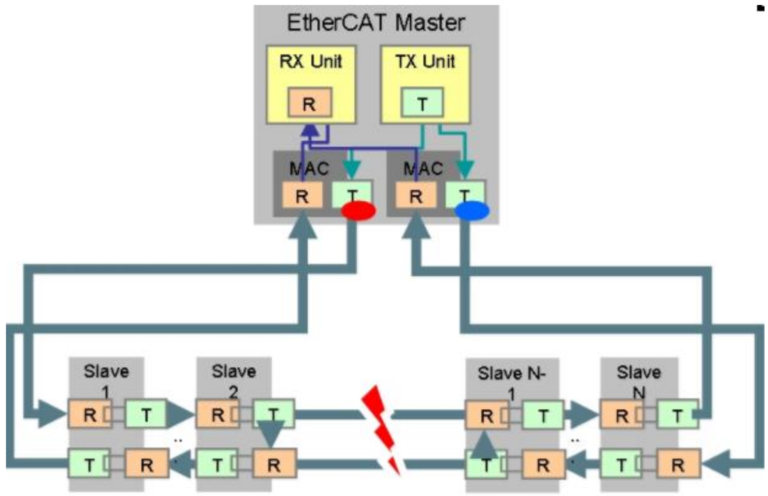
A redundant master (ring or loop) allows for operation to continue, unstopped, should a cable be disconnected or cut. Each slave controller has a fast port closing feature on its IN and OUT ports which allow data to be looped back should the cable disconnect. When using a loop wiring scheme only one embedded PC system is needed, with 3 Ethernet ports. One Ethernet port is used for the main house network and the other two ports as EtherCAT Master Ports. The primary master is connected normally, to the IN port of the slave. Each slave is daisy chained as before except on the last slave device the OUT port is connected to the backup Ethernet port. That port will act as the redundant master.

In a normal sequence the network packet goes in and out of each slave device, data placed into the packet by each slave. The backup Master Ethernet port will then receive the data and act like a slave device itself, transmitting the received packet back to the primary master. When the cable breaks the primary master will see the MAC Address of the slave device instead of the backup Ethernet port and know it is disconnected. The primary master then immediately sends its original packet out the backup Ethernet port to communicate with the remaining slave devices on the other side of the loop.

In the loop scenario both Ethernet ports are always active and there typically is a 90uS delay in packet propagation. The EtherCAT network will remain operational, even controlling servos, with a single disconnect anywhere after the first slave device. The network can even startup and initialize properly with a broken cable. A warning message will appear on the IO console as well as in the log buffer when a cable problem occurs. In order to repair the cable the network should be shut down in an orderly function. Typically it is not possible to just replace the cable and have the network stay operational during plug-in. It has been observed that a number of slave devices do not re-open their ports properly unless re-initialized, thus preventing hot cable repair. Sometimes it works, most of the time the network will fault. Incentive software is designed to handle the hot cable replacement.



Normal Operation, second Ethernet port loops data back to first



Cable has broken some slaves on left port, others on redundant Ethernet port.

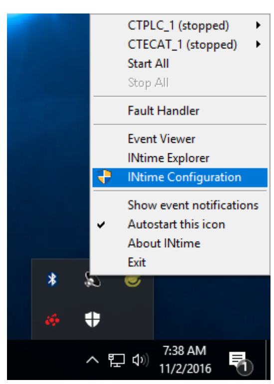
Restoration of the broken cable is not supported during operation due to slave limitations. When a single cable failure does occur operation will continue normally for all slaves.

Removal of CTPLC_1 or CTECAT_1 for Standalone Operation

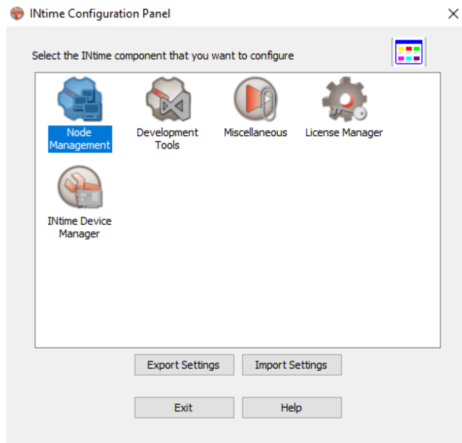
Both CTPLC_1, PLC logic, and CTECAT_1, EtherCAT Master can be used as standalone products, independent of each other. By default the installation procedures include the setup for running both products working in unison.

When running the EtherCAT Master in standalone operation the Master interfaces with a Windows program directly via the API and QuickBuilder is not present. In this mode we can run on a single core with just the EtherCAT_Master.rta program running, being driven by the Incentive API. It is assumed that two cores have been reserved for Incentive and you wish to remove CTPLC_1, freeing it up for Windows operation.

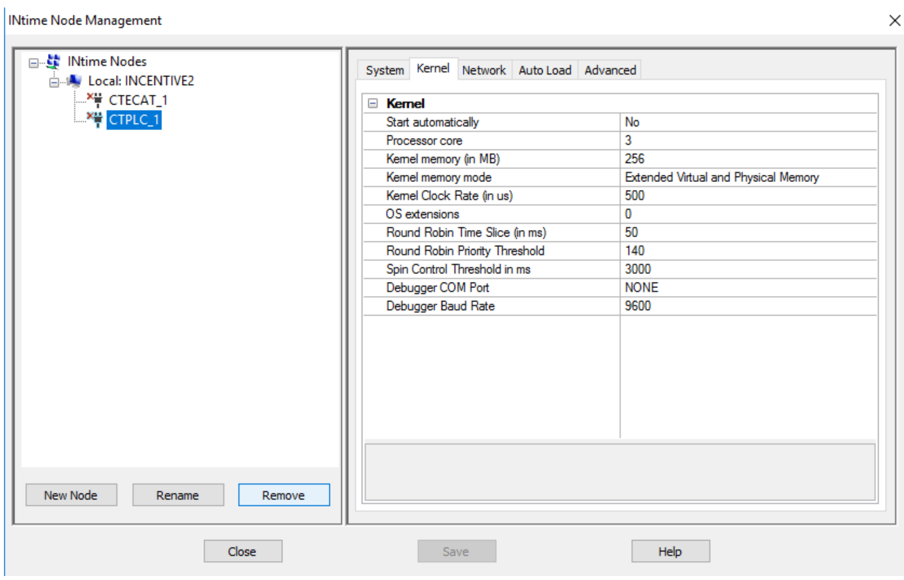
Begin by invoking the INtime Configurator, right clicking on the olive green INtime icon and selecting INtime Configuration:

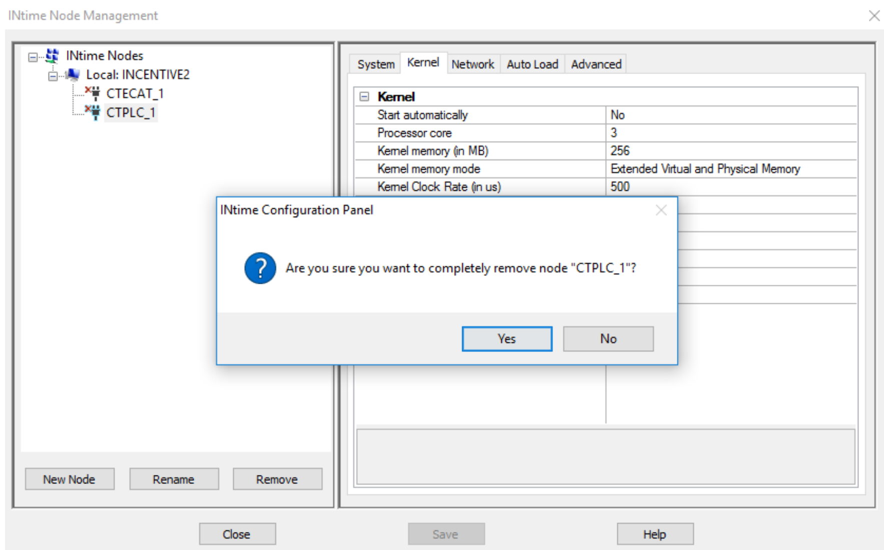


Select Node Management:

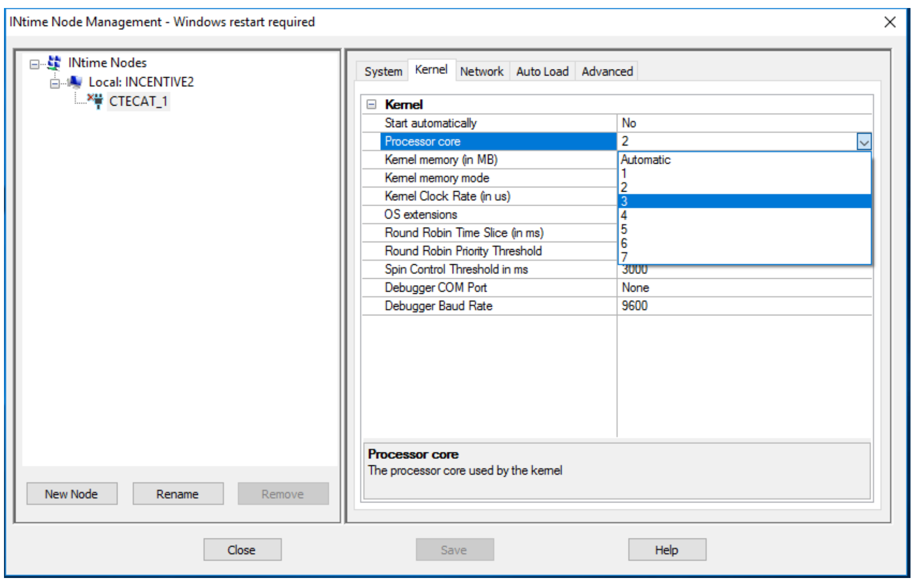


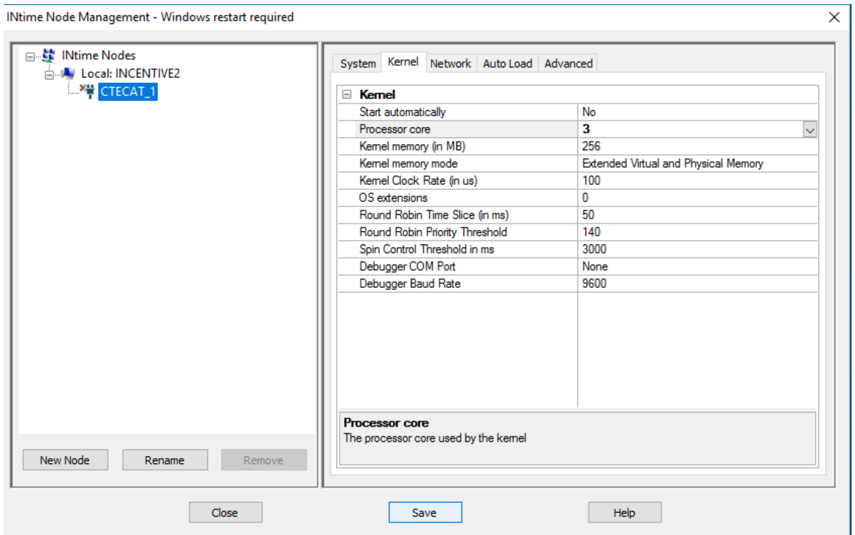
Once the Node Management screen appears select CTPLC_1, followed by Remove. Select 'Yes' when prompted "Are you sure?":



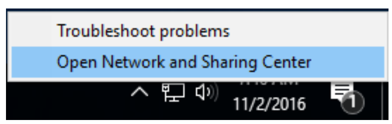


Select the CTECAT_1 node and change the Processor core to 3, if it is not already that, and click 'Save':

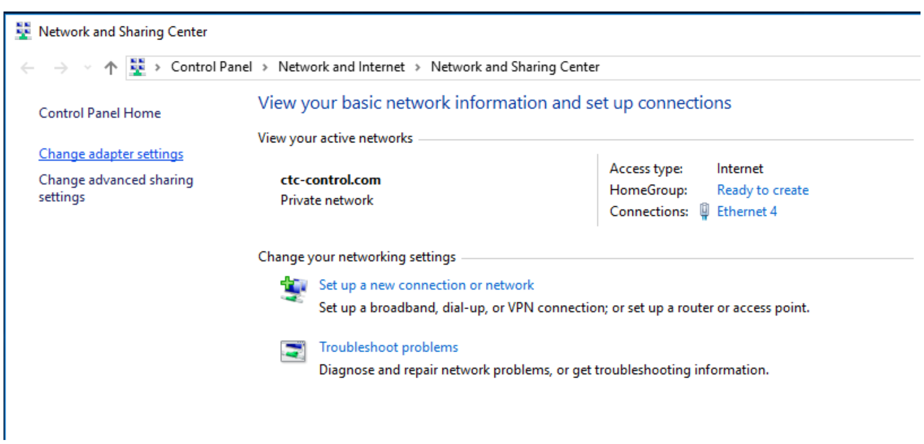




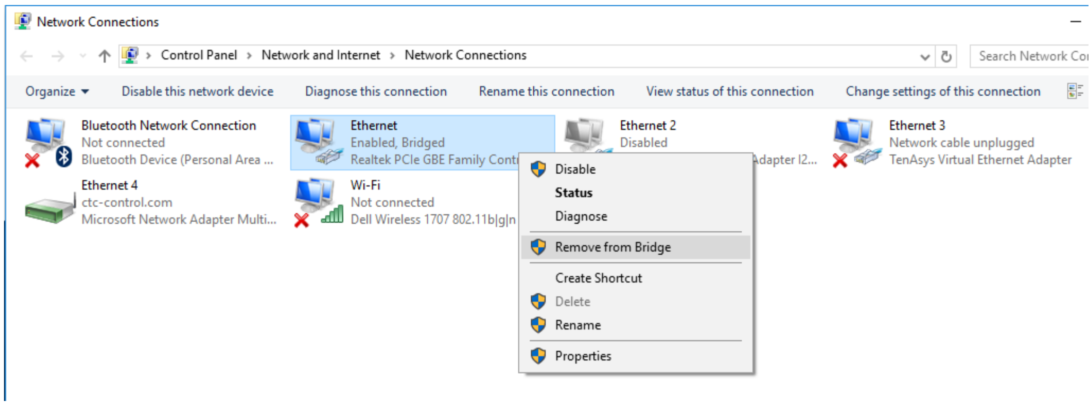
Close the Node Management screen and exit out of the INtime Configuration Panel. You will be prompted to reboot for the changes to take effect. Prior to rebooting we want to remove any bridged network adapters. Invoke 'Open Network and Sharing Center' by right clicking on the network tray icon or using Control Panel:



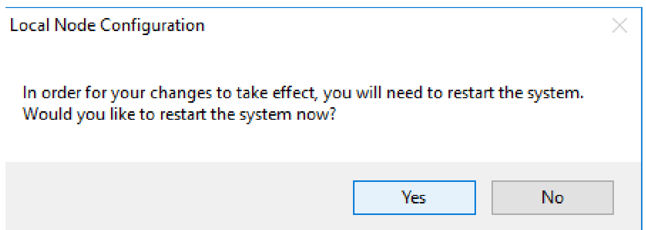
Select 'Change adapter settings':



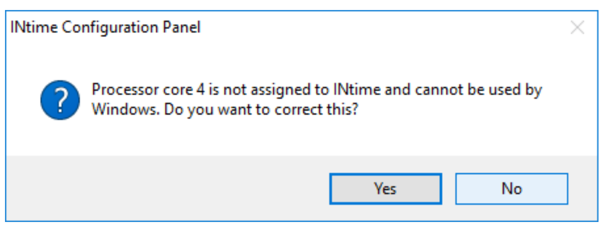
Right click on the bridged network adapters and select 'Remove from bridge'. This is usually your main network adapter and the TenAsys Virtual Ethernet Adapter:



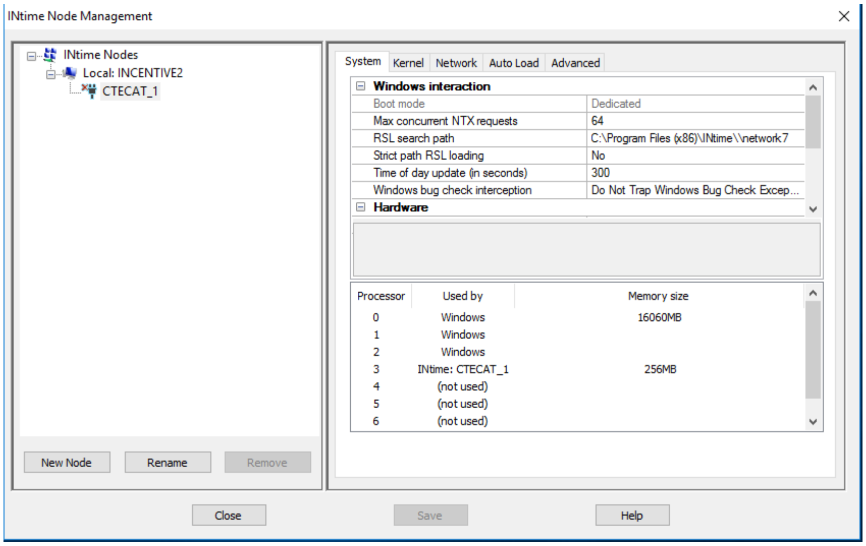
You may now reboot your PC by selecting ‘Yes’ on the Local Node Configuration prompt you received when you exited the Configurator:



After rebooting open the INtime Configuration Panel again and you may be prompted that ‘Processor core 4 is not assigned to INtime…’, select ‘No’:



Return to the INtime Node Management screen again and select ‘System’ on the tabs. Note that Windows now has 3 cores and core 3 is assigned to CTECAT_1:



To execute in standalone mode make sure a file called EtherCAT_Master_Only.txt exists in the C:\5300PC directory. Contents of the file do not matter, it is just checked for its existence. You may now run your application under full control of the CTC_Incentive API, without QuickBuilder or its CTPLC_1 process.

In some cases it is desirable to only run the Incentive 5300 PLC process and not EtherCAT Master. For instance when only using serial ports and/or protocols such as Modbus. This can be done by placing a file called 5300PLC_Only.txt in the C:\5300PC directory and only starting the CTPLC_1 node. CTECAT_1 node can be deleted if desired and a single core used as explained previously . The presence of the file will remove any error messages when CTECAT_1 is not detected at startup. When using only CTPLC_1 the virtual network bridge is not removed as it was with CTECAT_1.

QuickBuilder Programming and Atomicity

This is just a quick recommendation of something to think about as you structure your QuickBuilder program and your system architecture. Access to variables, whether normal registers or variants, is atomic and protected during read and write access. Also when writing a QuickBuilder program things within a conditional or loop { } are protected from having other QuickBuilder threads from accessing the same variables. Where subtle problems can occur is in the simple instance of writing something like this with inline code:

```
i = i+1;
```

Now that may seem straight forward and is protected between QuickBuilder threads but if you have something on the network or via the API poking ‘i’ with a value, say a 0, the result may not always be what you think it might. Consider the situation where QuickBuilder reads the value of ‘i’ (say containing 2), it then adds a 1 to it but before it can be stored back to ‘i’ the network pokes a 0 in ‘i’. Well for a brief period of time ‘i’ will be 0 but then the result of 2 + 1, or 3 will overwrite the 0. Atomicity or protection will not last for the full arithmetic formula unless it is between QuickBuilder tasks. A way around this is to have the network or API write public variables that are not involved in QuickBuilder calculations and have QuickBuilder access those variables at the beginning of their calculations, copying them to internal variables.

Some Common Issues and Resolutions

Shutdown and Restart Leave Power Switch Lit

Some computers have a problem with the INtime environment with Shutdown and Restart where Restart actually attempts a Shutdown and Shutdown does close Windows but the power switch stays lit. Only way out of this is to hold the power switch for 5 seconds to completely shutdown. This was recently observed on a Dell Optiplex system. An easy resolution to this is a registry change:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\rtif\Parameters

Change Flags from 0x05 to 0x25.

Bridged Network Fails after Windows Shutdown

This problem is a bug in some Windows 10 network drivers. We have noticed doing a Restart of Windows works fine but some computers after Shutdown you cannot communicate to our Incentive environment via UDP or TCP where we share a network controller with Windows (TenAsys virtual adapter). When Windows does a Shutdown it actually saving a memory image and not doing a full reboot. In doing so it is not properly saving the bridged network information. The resolution is simple, have Windows do a full shutdown (not a fast boot which is default) so a fresh memory image is loaded. The solution to this issue is detailed in the "*Startup and Network Configuration*" section.

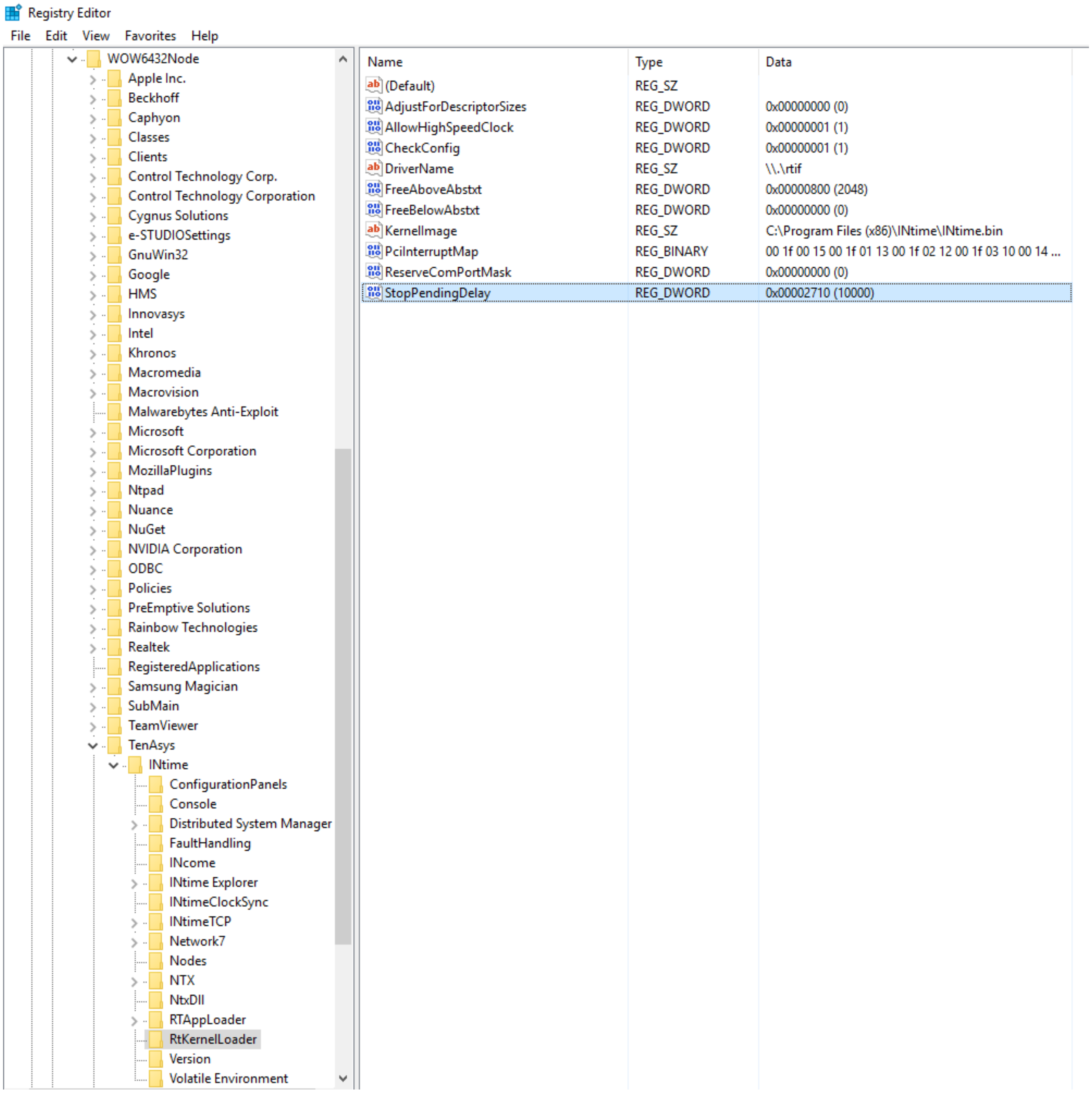
Disk Cache not fully flushed during INtime menu 'Stop All' on large programs

By default when you select 'Stop All' from the INtime menu to terminate Incentive we are given a 500 millisecond notification before stopping. During this time we shut the program down, clean things up, and flush any open files to disk. It is necessary to set this to a larger value to properly allow for system cleanup. Especially on larger programs this can cause a problem and corruption of 'nvar' tables since they are stored on the hard drive. Future releases of INtime will allow for an adjustment to this shutdown time but for now the registry must be modified after installation. CTC recommends a large number like 5-10 seconds to ensure cleanup of the environment. The following information will provide a resolution to this problem.

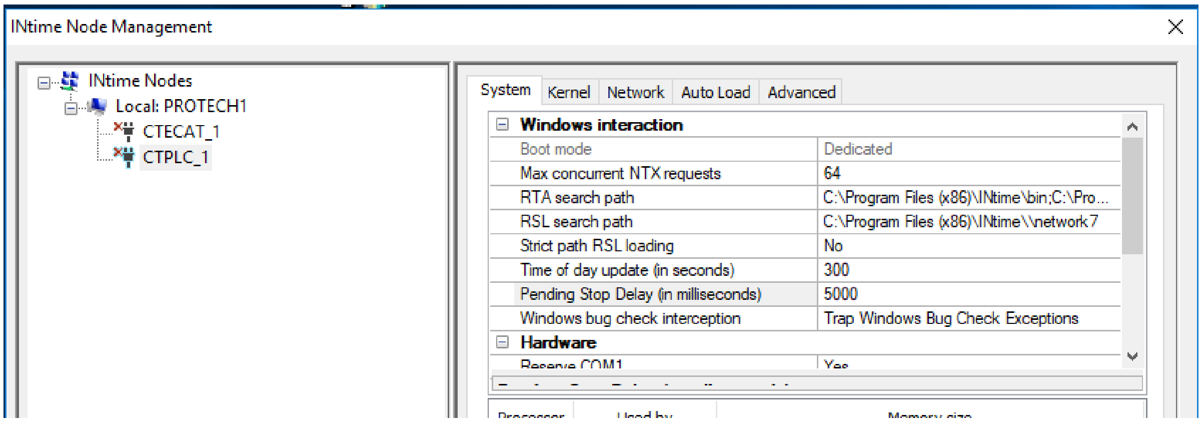
From TenAsys registry change, INtime version 6.1.17004:

Add a DWORD parameter named StopPendingDelay to the registry under HKLM\SOFTWARE\Wow6432Node\TenAsys\INtime\RtKernelLoader. StopPendingDelay specifies the number of milliseconds to wait between the KERNEL_SHUTDOWN_PENDING and the KERNEL_STOPPING notifications. The default is 500 milliSec. The delay will come into play every time one or all INtime nodes is stopped (via the tray icon, command line or programmatically

Suggest making it 5000 or 10000 (set to decimal).


Registry Editor
File Edit View Favorites Help
WOW6432Node
Apple Inc.
Beckhoff
Caphyon
Classes
Clients
Control Technology Corp.
Control Technology Corporation
Cygnus Solutions
e-STUDIOSettings
GnuWin32
Google
HMS
Innovasys
Intel
Khronos
Macromedia
Macrovision
Malwarebytes Anti-Exploit
Microsoft
Microsoft Corporation
MozillaPlugins
Ntpad
Nuance
NuGet
NVIDIA Corporation
ODBC
Policies
PreEmptive Solutions
Rainbow Technologies
Realtek
RegisteredApplications
Samsung Magician
SubMain
TeamViewer
TenAsys
INtime
ConfigurationPanels
Console
Distributed System Manager
FaultHandling
INcome
INtime Explorer
INtimeClockSync
INtimeTCP
Network7
Nodes
NTX
NtxDll
RTAppLoader
RtKernelLoader
Version
Volatile Environment
Name Type Data
(Default) REG_SZ
AdjustForDescriptorSizes REG_DWORD 0x00000000 (0)
AllowHighSpeedClock REG_DWORD 0x00000001 (1)
CheckConfig REG_DWORD 0x00000001 (1)
DriverName REG_SZ \\.\rtif
FreeAboveAbstxt REG_DWORD 0x00000800 (2048)
FreeBelowAbstxt REG_DWORD 0x00000000 (0)
KernelImage REG_SZ C:\Program Files (x86)\INtime\INtime.bin
PciInterruptMap REG_BINARY 00 1f 00 15 00 1f 01 13 00 1f 02 12 00 1f 03 10 00 14 ...
ReserveComPortMask REG_DWORD 0x00000000 (0)
StopPendingDelay REG_DWORD 0x00002710 (10000)

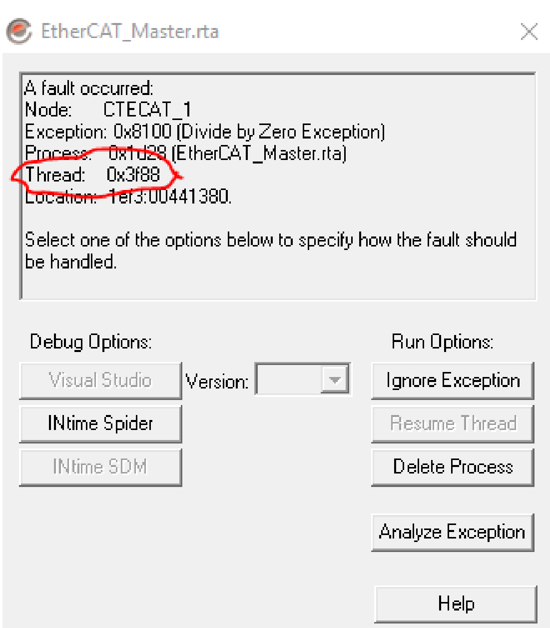
From TenAsys INtime version 6.3.18220 (or newer) the INtime Configurator has a setting now, set “Pending Stop Delay” for 5000 (5 seconds) for each node (CTPLC_1, CTECAT_1 and optional CTECAT_2):



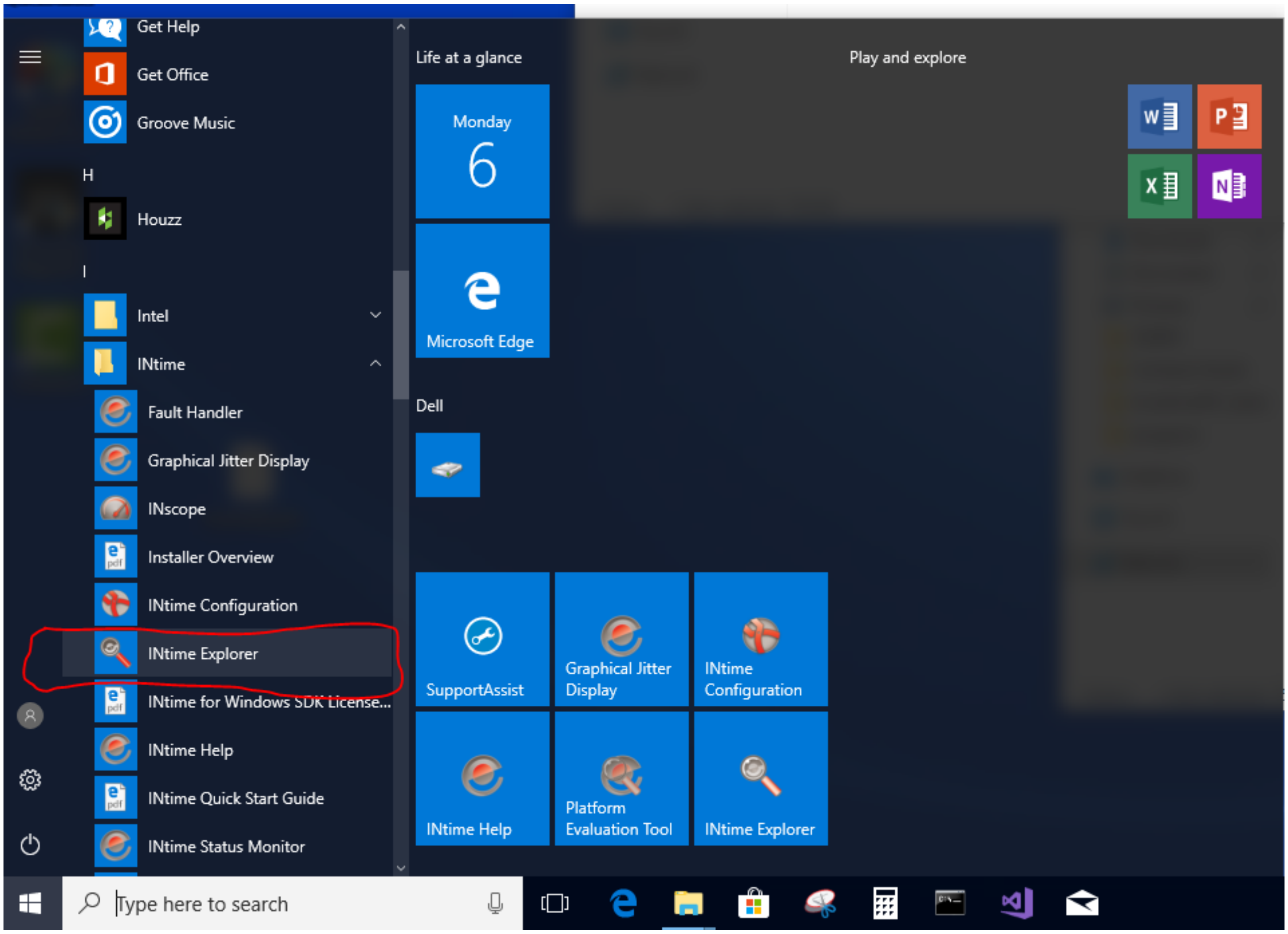
GPF’s (Exception Faults)

General Protection Faults (GPF) are a way for a program to notify the user that something wrong has occurred. These can be turned off but in the case of Incentive we have left them enabled to ensure proper operation. Ignoring them could cause a thread to fail but no notification given. GPF’s should not but at times, during an upgrade or through some unique implementation they can occur and when do it can be difficult to determine why. With the help of a tool called INtime Explorer, Incentive can locate the thread in which the GPF occurred as well as the process. This is useful information when reporting a problem to help lead to a quick resolution.

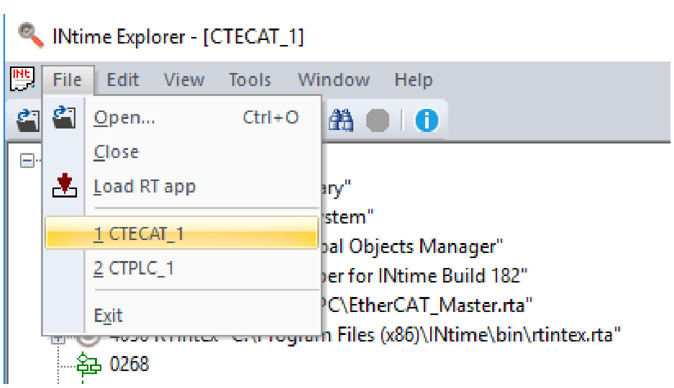
In the example below Incentive was modified to force a “Divide by Zero” exception for demonstration purposes. Upon execution the following occurred:



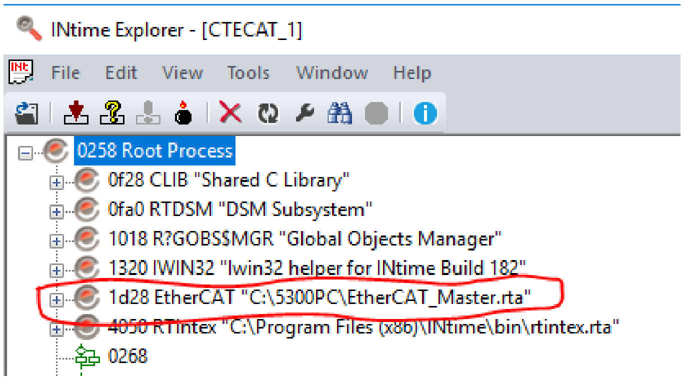
The title in the GPF says it is from the program EtherCAT_Master.rta, The "Exception" is Divide by Zero Exception with a Thread ID of 0x3f88, circled in red. In order to find out what the Thread name is the INtime Explorer program must be run:



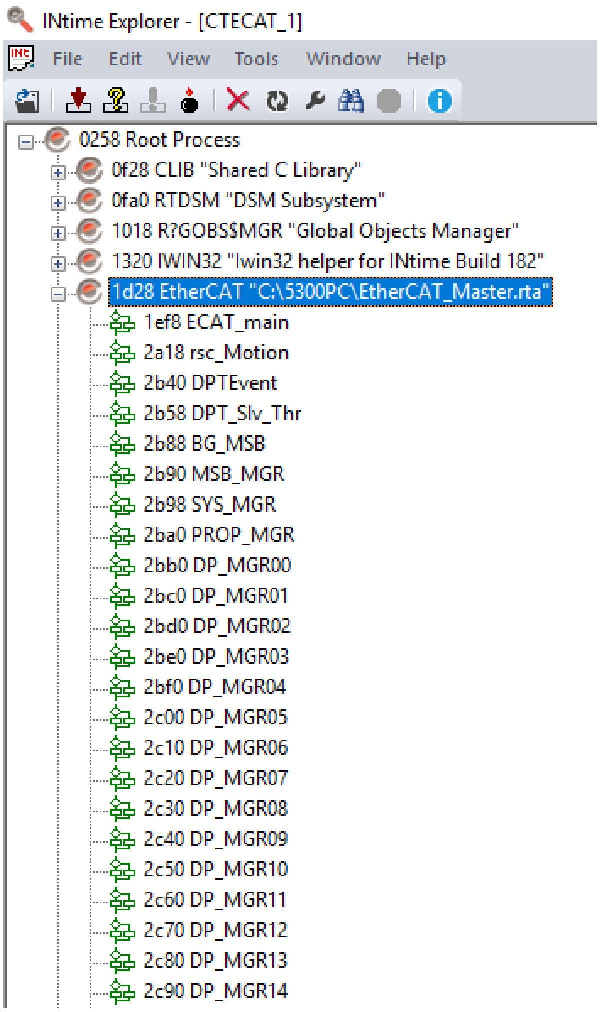
Once invoked one of two nodes must be selected depending upon which failed. EtherCAT_Master.rta is the CTECAT_1 node. 5300PLC.rta is the CTPLC_1 node. Make the selection as requested by INtime Explorer either when first invoked or via the File menu:



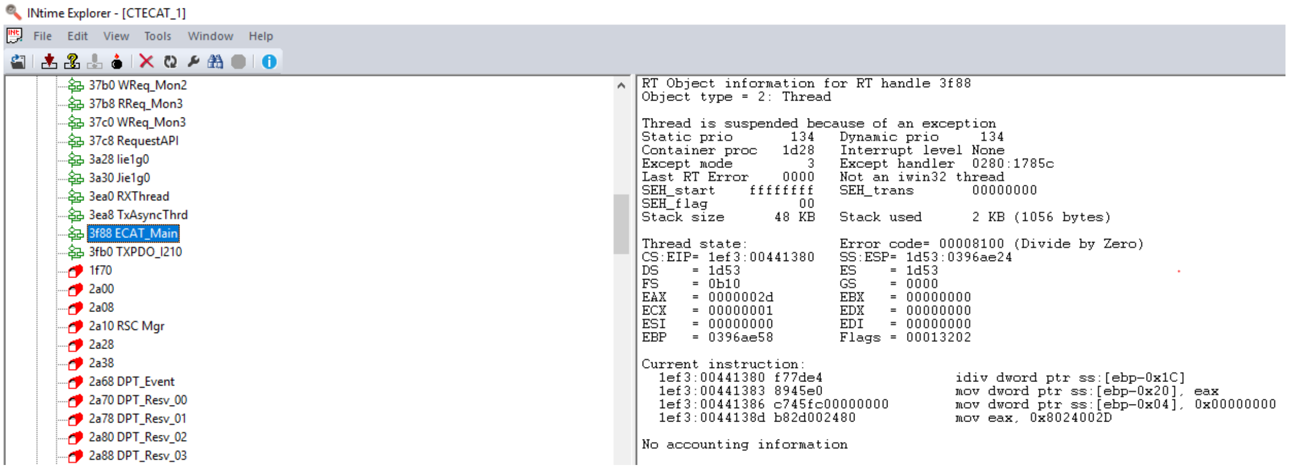
Next expand the EtherCAT process as circled in red below:



When expanded you will see a number of thread names with their thread id's to the left of them. Each thread serves a purpose with the execution of Incentive:



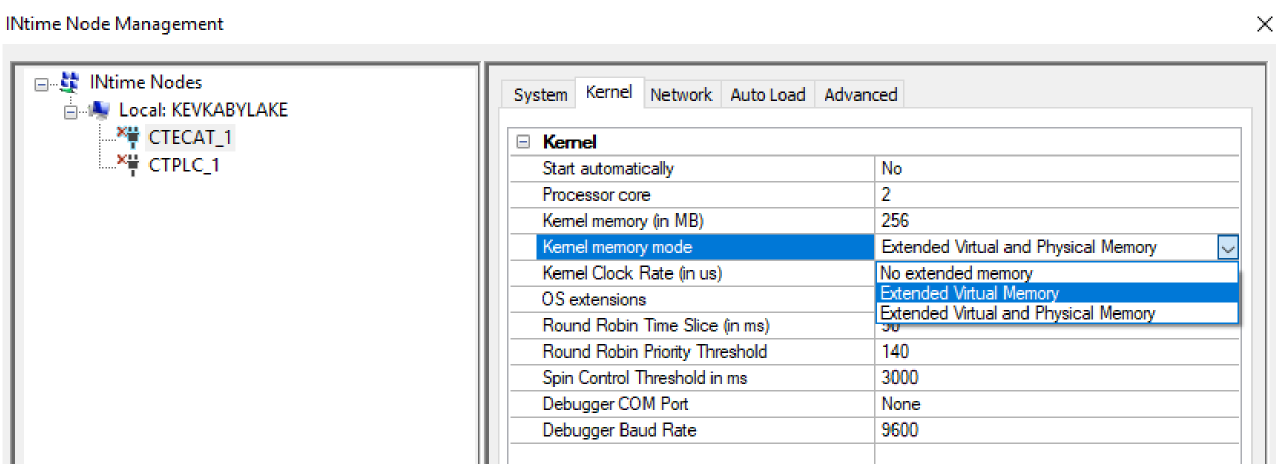
Scrolling down to the thread id that matches the exception the name of the thread can be identified:



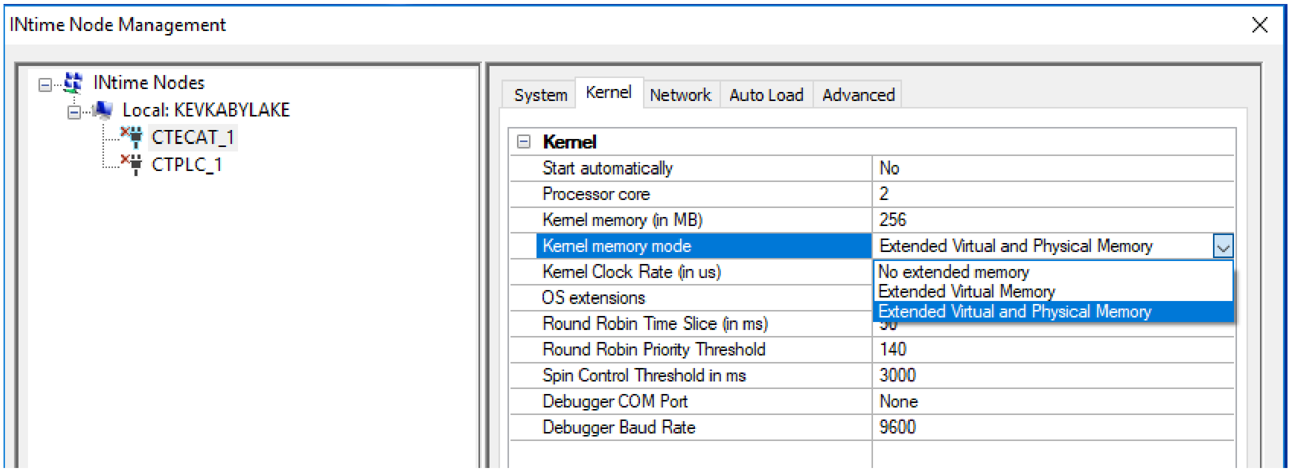
That highlighted above is the thread call “ECAT_Main”, it is responsible for initialization and startup of EtherCAT. The right frame shows details of the fault, with the Thread state detailing the cause. When contacting technical support, knowing the name of the thread and the reason of the fault is very useful in determining if it is a bug in the program, a user application, or operating system.

Running INtime 6.1 applications on 6.3 (PC_Realtime versus PC_Realtime6_3)

A new feature in QuickBuilder allows programs to be built as PC_Realtime or PC_Realtime6_3 (Compiler property of controller). PC_Realtime applications will run on TenAsys INtime 6.1 and > 6.3, the difference is the libraries the application is linked to when translated by QuickBuilder. For future migration it is best to move towards INtime 6.3/6.4 given that is the latest supported revision by the manufacturer. When running 6.1 (PC_Realtime) translated programs within a 6.3/6.4 environment the memory type must be set to “Extended Virtual” on the System tab within the INtime Configurator for both Incentive programs, CTPLC_1 and CTECAT_1.



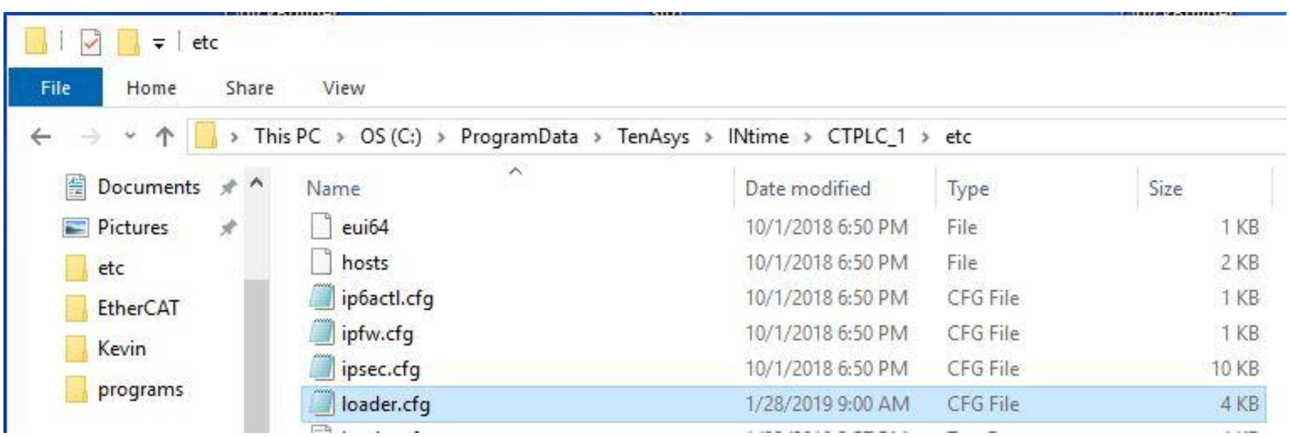
When built as PC_Realtime6_3 the memory type is set to “Extended Virtual and Physical”, the default.



Note that QuickBuilder programs translated with PC_Runtime6_3 can not run on a 6.1 environment. Also any network messaging using gobsnet is not compatible.

Using Socket Sessions with TCP Serial, Modbus, etc…

When using a lot of Network connections it may be necessary to increase the number of available worker threads available to the TenAsys INtime kernel. The default is 32 but this can be increased as needed. Typically for an extra 10 TCP serial ports and and 10 Modbus a value of 64 is needed. If there are too few threads you may see an error message of something like "RtrslScope: AllocateRtMemory failed” or a socket error 9 or 114 on the Incentive PLC console. Thus far this has not been seen in an actual application and could only be forced by creating 40 sockets and shutting down 20 of them every 5 seconds and then re-connecting over a multi-day period, about 15,000 cycles. This is not the norm but is included here should an application ever encounter a similar problem. To change the worker threads a file called loader.cfg must be edited and the line kern.workers_max=32 changed as shown below:



loader.cfg - Notepad

File Edit Format View Help

```
#
# INtime networking stack load time parameters
#
#
# Interfaces to load
# Can also be specified as stack command line parameters
# istack.rta -i ven0 ie100m0
#net.interfaces=ven0,ie100m0

# Amount of memory allocated for stack buffers
vm.kmem_size=0x0

#
# Priorities are relative (less than or equal) to this
kern.maxpri=130
# Size of worker thread stack
kern.workers_stack=4096
# Number of pesistant worker threads in pool (# of simultaneous operations)
# Additional threads are created/deleted as needed but a stack performance
# penalty is associated with their on the fly creation.
kern.workers_max=64
# Number of seconds to timeout a worker > kern.workers_max
kern.workers_timeout=10
# Cache this number of UMA allocations
```

The security properties of the file may need to be changed so that you have write permission. Also, if you ever want to know how many sockets are being used by the system you can log into Incentive using telnet and execute the ‘get numsockets’ command.

Downgrading TenAsys INtime Run-time Versions (6.4 or 6.3 to 6.1)

It is very important that when downgrading an INtime Run-time, for example for 6.3 to 6.1, delete the installation directory after uninstalling. It has been found that any patches will be left in the directory and could cause installation problems.

EtherCAT Device Manufacture and Product Codes Supported

A typical EtherCAT master system supports an XML based ESI file (EtherCAT Slave Information). Incentive takes another approach which removes the complexity of an EtherCAT system from the user. Rather than randomly import and support complicated ESI files Incentive imbeds the supported and fully tested devices within its software. This allows for complete auto-discovery and configuration of devices that the user knows will work. In some cases a manufacturer may release a similar product but with a different EtherCAT product code. Rather than not allow operation Incentive accepts an alternate product code file (alt_productcodes.txt) which can map these new devices to existing devices. This file is stored in the C:_system\programs directory and loaded during initialization, enhancing that which is built-in.

The file format is free text where all lines not starting with DEVICE: are ignored, therefore comments may be included. An example is seen below:

DEVICE: [MANUF_ID] = 1337, [PRODUCT_CODE] = 0x02200008, [SIMILIAR_PRODUCT_CODE] = 0x02200001, [NAME] = Sigma V Replacement

The above entry allows a Yaskawa drive with the product code of 0x02200008 to use the same configuration and settings as that already built into the Incentive product. The [MANUF_ID], [PRODUCT_CODE], and [SIMILAR_PRODUCT_CODE] are required fields, all others would use that already defined unless overridden by the alternate product code file. Fields available are below, either decimal or hex (0x) values may be used and all must appear on the same line for the device defining:

[MANUF_ID] – Vendor ID from the EtherCAT ESI file (required).

[SIMILAR_MANUF_ID] – Vendor ID driver to use within Incentive, from the table following this section (optional). This is only needed if mapping the [MANUF_ID] to a different manufacturer.

[PRODUCT_CODE] – Vendor Product Code from the EtherCAT ESI file (required).
[SIMILAR_PRODUCT_CODE] – Vendor Product Code driver to use that is already defined within Incentive from the table following this section (required).

[MANUF] – Manufacturer name to be displayed for this device (optional). This is only needed if mapping to a different manufacturer.

[NAME] – Name to be displayed for this device when online (optional).

[GROUP] – Group this device belongs to; Drive, Gateways, SystemBk… (optional).

[AINS] – Number of analog inputs.

[AOUTS] – Number of analog outputs.

[DINS] – Number of digital inputs if this device is a servo drive. Estimate only, actual is limited to 32.

[DOUTS] – Number of digital outputs if this device is a servo drive. Estimate only, actual is limited to 32.

[SERVOS] – Number of axis available, 1 or 2 depending upon the driver mapping to.

[STEPPERS] – Not currently used.

[COUNTERS] – Not currently used.

Incentive built-in Manufacture and product codes:

Manufacturer	MANUF_ID	PRODUCT_CODE	NAME	GROUP	AIN	AOUT	DIN	DOUT	SERVOS

Manufacturer	MANUF_ID	PRODUCT_CODE	NAME	GROUP	AIN	AOUT	DIN	DOUT	SERVOS
ABB	183	0x000002bc	MicroFlex e150	Drive			6	3	1
ABB	183	0x000002c0	MicroFlex e180	Drive			6	3	1
ABB	183	0x000002c1	MicroFlex e190	Drive			6	3	1
Adv Motion	189	0x012D0000	DP/DZ	Drive			6	3	1
Adv Motion	189	0x012E0000	DP/DZ	Drive			6	3	1
Adv Motion	189	0x012F0000	DP/DZ	Drive			6	3	1
Adv Motion	189	0x01300000	DP/DZ	Drive			6	3	1
Applied Motion	1028	0x00000001	SS-EC	Drive			11	4	1
Applied Motion	1028	0x00000002	STF	Drive			32	32	1
Beckhoff	2	0x044c2c52	EK1100	SystemBk					0
Beckhoff	2	0x044d2c52	EK1101	SystemBk					0
Beckhoff	2	0x04562c52	EK1110	SystemBk					0
Beckhoff	2	0x03fa3052	EL1018	DigIn			8		0
Beckhoff	2	0x07d83052	EL2008	DigOut				8	0
Beckhoff	2	0x0c1e3052	EL3102	AnaIn	2				0
Beckhoff	2	0x0c283052	EL3112	AnaIn	2				0
Beckhoff	2	0x0c323052	EL3122	AnaIn	2				0
Beckhoff	2	0x0c463052	EL3142	AnaIn	2				0
Beckhoff	2	0x0c503052	EL3152	AnaIn	2				0
Beckhoff	2	0x0c5a3052	EL3162	AnaIn	2				0
Beckhoff	2	0x0fc03052	EL4032	AnaOut		2			0
Beckhoff	2	0x10003052	EL4102	AnaOut		2			0
Beckhoff	2	0x10103052	EL4112	AnaOut		2			0
Beckhoff	2	0x101a3052	EL4122	AnaOut		2			0
Beckhoff	2	0x10243052	EL4132	AnaOut		2			0
Copley	171	0x00000380	Accelnet-AEP	SystemBk			8	3	1
Copley	171	0x00001000	Xenus Plus-XEL	Drive			15	6	1
Copley	171	0x000010b0	Xenus Plus-XE2	Drive			22	7	2
Copley	171	0x00001110	Accelnet-BEL	Drive			11	4	1
Copley	171	0x000011b0	Xenus Plus-XEL	Drive			15	6	1
Copley	171	0x00001030	Stepnet-AEM	Drive			11	6	1
Copley	171	0x00001130	Stepnet-TEL	Drive			14	6	1
Copley	171	0x000010c0	Accelnet-BE2	Drive			18	7	2
Copley	171	0x00001050	Accelnet-AE2	Drive			20	7	2
Copley	171	0x00001090	Accelnet-SE2	Drive			26	7	2
Copley	171	0x000010f0	Accelnet-TE2	Drive			26	7	2
CTC	0xfffffffe	0xfffffffe	Virtual	Drive			0	0	1
CTC	0x00000989	0x00000002	5300 PLC	SystemBk	256	256	1024	1024	0
Delta Elec.	0x000001dd	0x10305070	ASDS-A2-E	Drive			6	6	2
Emerson	249	0x00040003	Unidrive SP	Drive			6	3	1
Emerson	249	0x00040203	Digitax ST	Drive			6	3	1
Festo	29	0x001706AC	EMCA-EC	Drive			6	3	1
HMS	27	0x0000001b	ANYBUS X	Gateways					0

Manufacturer	MANUF_ID	PRODUCT_CODE	NAME	GROUP	AIN	AOUT	DIN	DOUT	SERVOS
			MODBUS-TCP						
HMS	27	0x0000003C	ANYBUS X PROFINET	Gateways					0
HMS	27	0x0000003D	ANYBUS X PROFINET	Gateways					0
HMS	27	0x00000032	ANYBUS X-GTWY	Gateways					0
Koganei		0x00000001	YS7K1 Sol Valve 16	SystemBk				16	0
Koganei		0x00000002	YS7K1 Sol Valve 32	SystemBk				32	0
Kollmorgen	106	0x00414b44	AKD	Drive			6	3	1
IAI	1416	0x00A00003	RC-ECT-FMOD3	Drive			0	0	1
LinMot	0x4c4e5449	0x00960971	C1150-DS-XC-0S	Drive			0	0	1
LinMot	0x4c4e5449	0x0096096F	C1250-DS-XC-0S	Drive					1
LinMot	0x4c4e5449	0x00960970	C1250-DS-XC-1S	Drive					1
LinMot	0x4c4e5449	0x00960A4C	C1450-DS-VS-0S	Drive					1
LinMot	0x4c4e5449	0x0096096B	E1450-DS-QN-0S	Drive					1
LinMot	0x4c4e5449	0x0096096C	E1450-DS-QN-1S	Drive					1
LinMot	0x4c4e5449	0x00960972	C1150-DS-XC-1S	Drive					1
LinMot	0x4c4e5449	0x0096096A	E1250-DS-UC	Drive					1
Maxon	0x000000fb	0x00006050	EPOS-4	Drive					1
Mitsubishi	2590	0x00000201	MR-J4	Drive			0	0	1
Mitsubishi	2590	0x00000301	MR-J5	Drive			0	0	1
Mitsubishi	2590	0x00000302	MR-J5	Drive			0	0	1
Mitsubishi	2590	0x00000303	MR-J5	Drive			0	0	1
Mitsubishi	2590	0x00000304	MR-J5	Drive			0	0	1
Mitsubishi	2590	0x00000307	MR-J5	Drive			0	0	1
Mitsubishi	2590	0x00000308	MR-J5	Drive			0	0	1
Mitsubishi	2590	0x00000309	MR-J5	Drive			0	0	1
Numatics	0x000001e1	0x00004733	G3	SystemBk			0	0	0
Numatics	0x000001e1	0x00000580	580	SystemBk				32	0
Omron	131	0x00000064	GX-JC06	SystemBk			0	0	0
Omron	131	0x00000065	GX-JC06	SystemBk			0	0	0
Omron	131	0x00000081	GX-JC06-H	SystemBk			0	0	0
Omron	131	0x00000082	GX-JC06-H	SystemBk			0	0	0
Omron	131	0x000000AA	NX-ECC203	SystemBk			0	0	0
Omron	131	0x000000A6	NX-ECC202	SystemBk			0	0	0
Omron	131	0x00000083	NX-ECC201	SystemBk			0	0	0
Panasonic	1647	0x511050A1	MINAS	Drive			6	3	1
Parker Hannin	0x09000089	0x00010001	PD-xxC	Drive			0	0	1
Parker Hannin	0x09000089	0x00010000	iPM	Drive			0	0	1
Parker Hannin	0x09000089	0x00010002	PD-xxP	Drive			0	0	1
PULSEROLLER	73	0x00000491	MOTIONLINX	Drive					2
Sanyo Denki	441	0x00000002	RS2E	Drive			6	3	1
SMC Corp	0x00000114	0x01000005	EX600	SystemBk			0	0	0

Manufacturer	MANUF_ID	PRODUCT_CODE	NAME	GROUP	AIN	AOUT	DIN	DOUT	SERVOS
Turck	0x0000009c	0x00682D74	Series SI Unit BL20 EtherCAT Gateways	SystemBk			0	0	0
Wago	33	0x07500354	750-354	SystemBk			6	3	0
Yaskawa	1337	0x02200001	SIGMA 5 Rotary	Drive					1
Yaskawa	1337	0x02200002	SIGMA 5 Linear	Drive			6	3	1
Yaskawa	1337	0x02200301	SIGMA 7 Rotary	Drive			6	3	1
Yaskawa	1337	0x02200302	SIGMA 7 Linear	Drive			6	3	1
Yaskawa	1337	0x02200401	SIGMA 7 Rotary	Drive			6	3	1
Yaskawa	1337	0x41313030	A1000 VFD	Drive			6	3	1
Yaskawa	1337	0x53455333	V1000 VFD	Drive			6	3	1
Yaskawa	1337	0x53455333	V1000 VFD	Drive			6	3	1

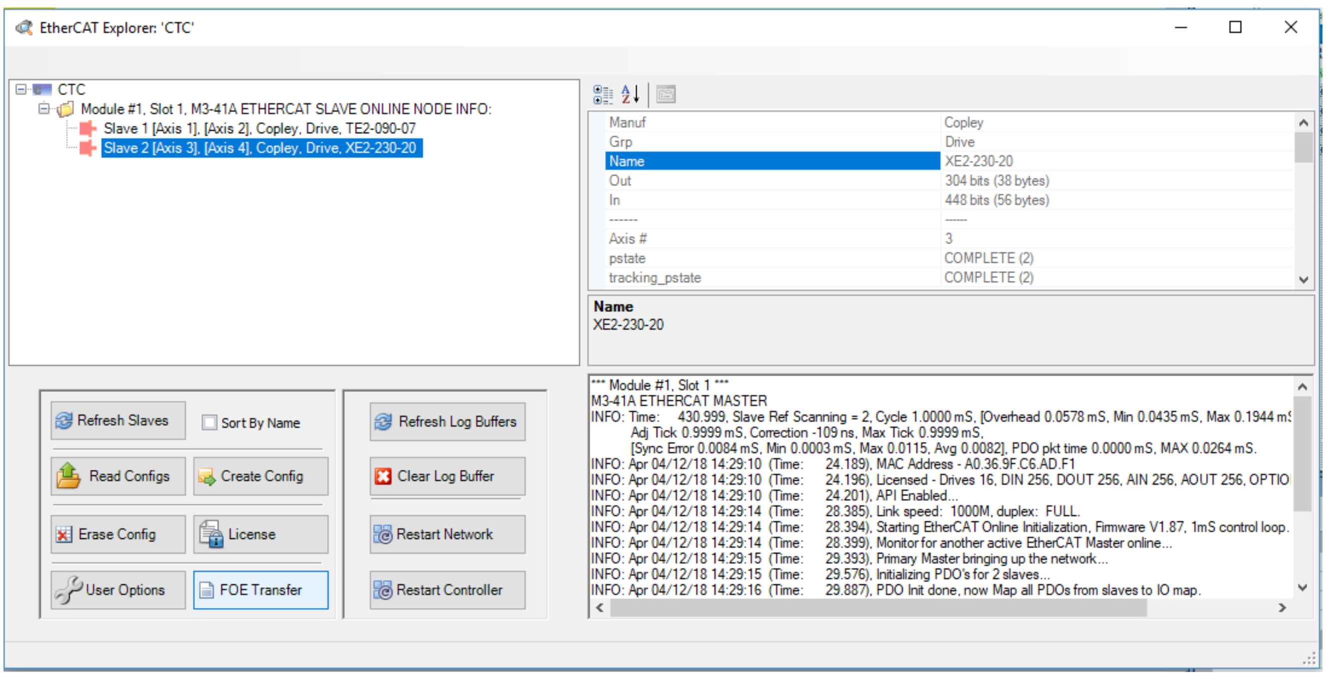
Product Code mapping is only supported on PC platforms, not the 5300 M3-41A.

FOE (File Over EtherCAT)

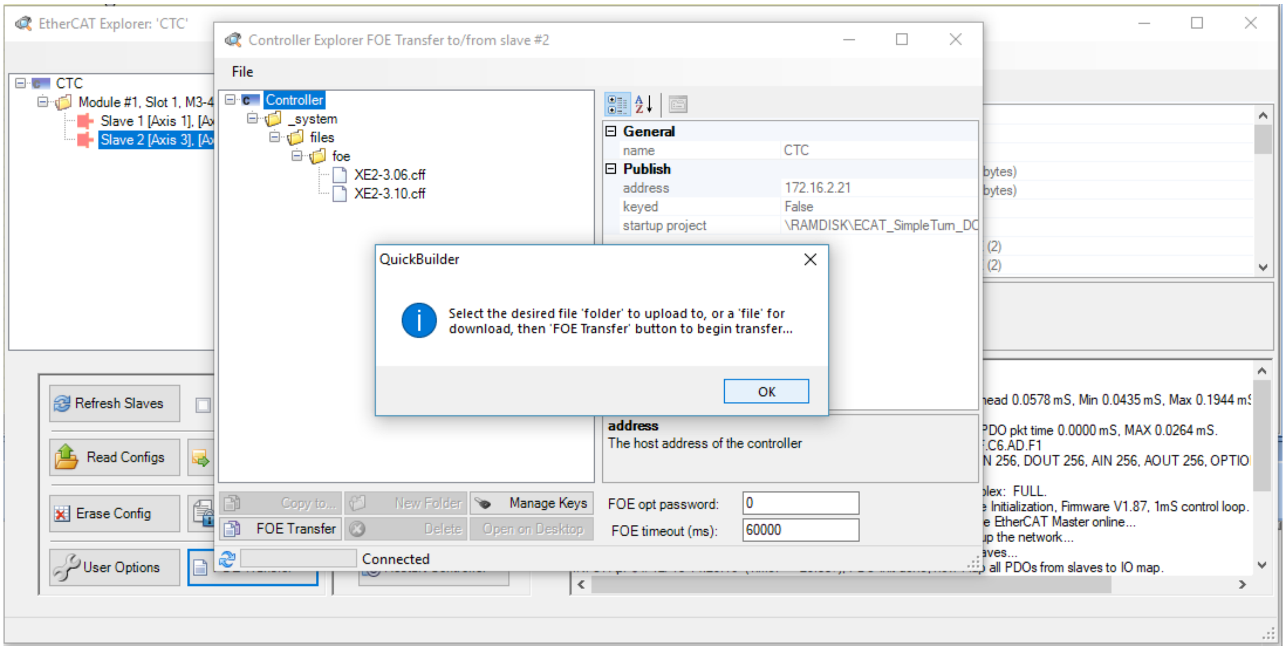
QuickBuilder File Transfer

The QuickBuilder EtherCAT Explorer supports an EtherCAT protocol known as FOE or File over EtherCAT. This protocol allows a means to transfer configuration and firmware files to slave devices on the network using EtherCAT. Previously slave devices had to be manually taken offline and serial ports or other network connections used to update manufacturer specific information. Not all manufacturers support this protocol and for the moment it is only available and tested on Copley drives. Mitsubishi also supposedly has support but it has yet to be tested and requires special firmware in their drives.

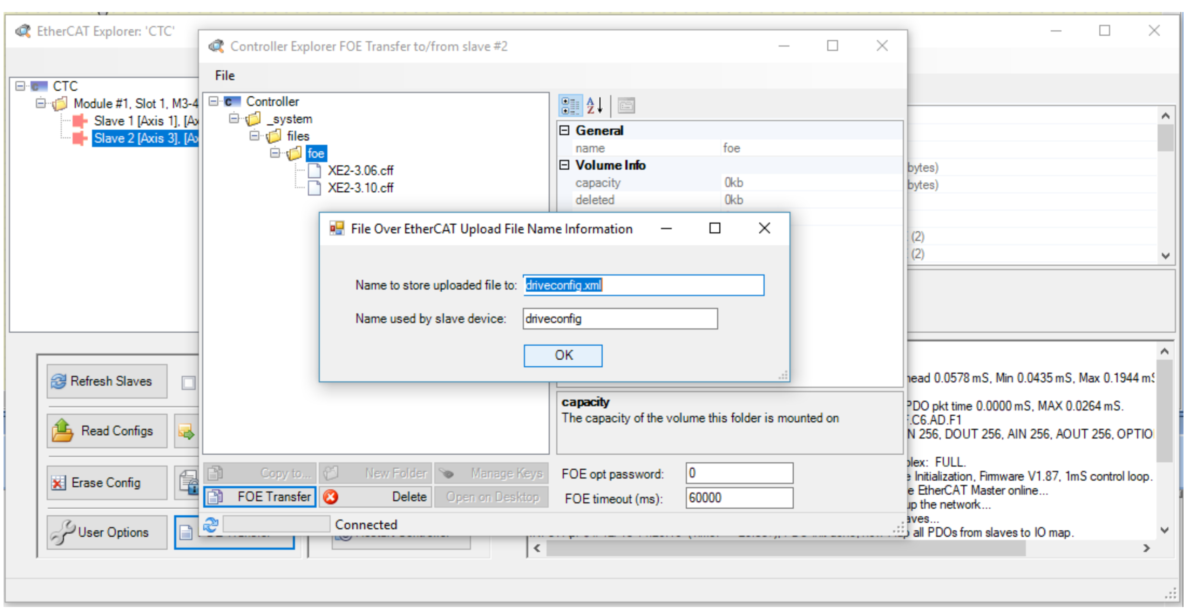
Only Incentive PC supports the protocol. It could be made available on the 5300 PLC but given its limited storage abilities it is yet to be fully implemented. If the need arises it could be easily added. To invoke the FOE transfer window you must first select the slave device of interested followed by pressing the 'FOE Transfer' button:



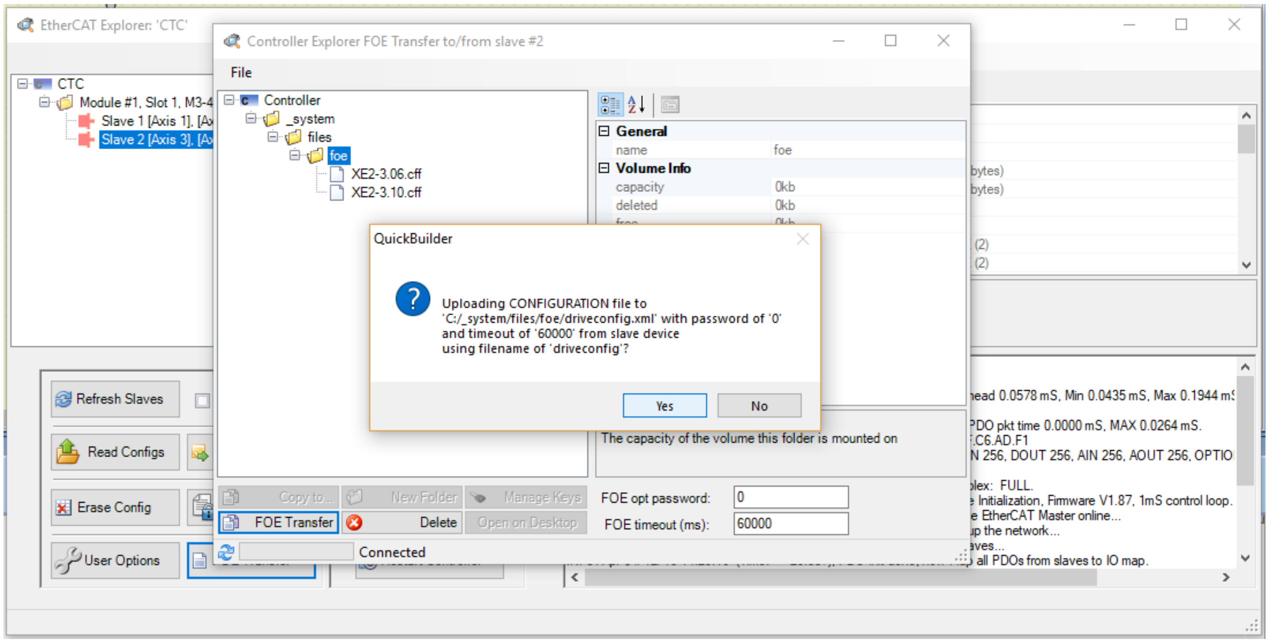
The Controller Explorer window will then appear showing the default FOE storage directory C:_system\files\foe. This, and additional subfolders is where files for transfer should be placed on the Incentive PC. The Controller Explorer uses the FTP protocol to obtain a directory of the remote PC for your presentation:



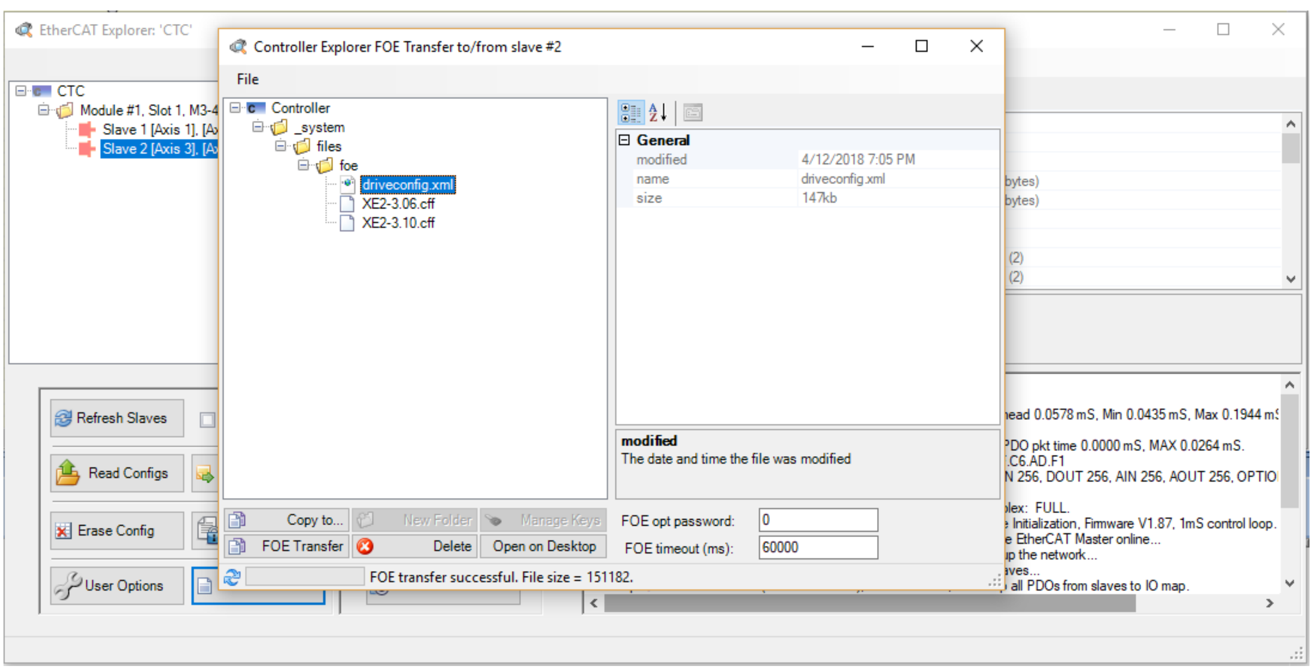
Upon opening a dialog box will tell you that if you select just a folder followed by pressing the 'FOE Transfer' button then you will be uploading a file to that directory and will have additional forms which will let you set any custom file names. The default is driveconfig.xml since at the moment all uploads are drive configuration files. This may change as the protocol is more widely implemented. If you select a file followed by the 'FOE Transfer' button then that is for download to the slave device. Note that you can set a generic 'FOE opt password' (numeric only) and a timeout. Typically 60000 milliseconds or 1 minute is needed as that is the length of time it takes for drives to re-flash their firmware. Below is an example of uploading a configuration from a drive. It may be used to download to other drives that may use the same configuration (drive settings per the manufacturer). First the folder to store the uploaded file into is selected, followed by pressing the 'FOE Transfer' button:



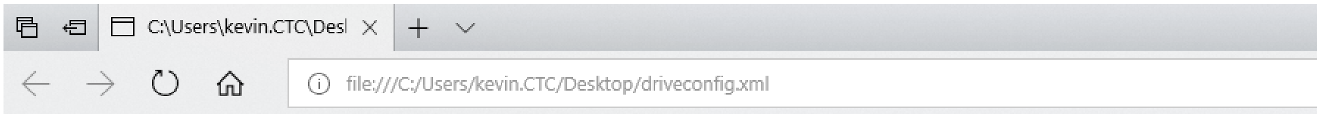
A form will appear which allows you to change the default system file name from 'driveconfig.xml' to something else. The name used by the slave device defaults to 'driveconfig' which is the standard for all drives. Upon pressing OK another dialog will appear giving you a summary of what is about to occur. Hit the 'No' button to abort or 'Yes' to continue:



Upon selecting 'Yes' QuickBuilder will interact with the remote Incentive PC requesting the file be uploaded, the results will be displayed at the bottom of the form and the directory will be updated to show the new directory:



The highlighted file 'driveconfig.xml' was uploaded and at the bottom of the form it says it was successful and the file size was 151182 bytes. You can further select 'Copy to' to copy the file to you local QuickBuilder PC and/or the 'Open on Desktop' button to open the file locally for inspection. Below is 'Open on Desktop' showing the file with the Microsoft Edge browser:




```xml
<?xml version="1.0" encoding="UTF-8"?>
- <DeviceConfig>
      <!-- This section contains general info about the config file. -->
  - <FileInfo>
      <FileVersion>2.0</FileVersion>
      <FirmwareVersion>3.06</FirmwareVersion>
    </FileInfo>
      <!-- This section contains some options used when uploading the configuration to the drive -->
      <!-- Modify the settings here to control drive behavior on upload. Settings here effect following sections. -->
  - <DownloadOptions>
        <!-- Set this option to TRUE to ignore unknown parameters. FALSE to fail on unknown parameters. -->
      <IgnoreUnknownParams>TRUE</IgnoreUnknownParams>
        <!-- If true, all files in the flash file system will be deleted when the FileSystem tag is first encountered. -->
        <!-- If false, new files will be added, but old files will only be removed if a new file is added to the same slot. -->
      <ClearFileSystem>TRUE</ClearFileSystem>
        <!-- Set this option to TRUE if the file should only be loaded into a drive with the same product code -->
        <!-- as the drive that created the file. Set to FALSE to allow the file be loaded into any drive. -->
      <MustMatchProductCode>TRUE</MustMatchProductCode>
        <!-- Set this option to TRUE if the file should only be loaded into a drive with the same firmware version -->
        <!-- Note that the firmware version located in the Identity section of this file is checked, not the FileInfo section. -->
      <MustMatchFirmwareVersion>FALSE</MustMatchFirmwareVersion>
    </DownloadOptions>
      <!-- This section gives information about the device that created the file. -->
  - <IdentityInfo>
        <!-- EtherCAT/CANopen vendor ID. -->
      <VendorID>0x00ab</VendorID>
        <!-- Product code, identifies the drive model. -->
      <ProductCode>0x10b0</ProductCode>
        <!-- CRC of drive parameters contained in the file. -->
      <Checksum>0xf363a930</Checksum>
      <ChecksumMethod>0x01</ChecksumMethod>
        <!-- Firmware version of drive that generated the file. -->
      <FirmwareVersion>3.06</FirmwareVersion>
    </IdentityInfo>
      <!-- This section contains all drive parameters. -->
  - <ParameterList>
    - <ParameterGroup N="0">
          <!-- This group of parameters contains global settings shared by all axes on the drive -->
      - <Parameter>
          <VarID>0x0070</VarID>
          <Name>out0</Name>
          <Description>Output 0 configuration</Description>
          <Index>0x2193</Index>
          <Subindex>0x01</Subindex>
          <Type>OUTCFG</Type>
          <Size>10</Size>
          <Value>00017F00440000000000</Value>
        </Parameter>
```


To download a file it is very similar to uploading except the file is selected instead of the folder. If a .cff file is selected (firmware) a notice will appear that the EtherCAT network will be stopped prior to downloading the file for safety reasons. Firmware can take up to a minute to flash to be patient. The status window will appear with the results after flashing. If the .xml file is downloaded then it is assumed to be a configuration file and the network will not be stopped. Be careful as the configuration file may change parameters within the drive that effect operation.

Incentive API File Transfer

The Incentive API Controller.Axis and Controller.AxisSupervisor classes both support the EtherCAT protocol known as FOE or File over EtherCAT. As with QuickBuilder, this protocol allows a means to transfer configuration and firmware files to slave devices on the network.

Transfers may be done by axis number (1 to N) as well as slave number using FOE_DownloadByAxis, FOE_DownloadBySlave, FOE_UploadByAxis, and FOE_UploadBySlave. The syntax for download is shown below as reference:

AxisSupervisor.FOE_DownloadBySlave Method

Download a file to a slave device using the EtherCAT FOE protocol

Namespace: CTC_Incentive
Assembly: CTC_Incentive (in CTC_Incentive.dll)

Syntax

C# VB C++

```
public bool FOE_DownloadBySlave(
        int slavenum,
        string slave_filename,
        string system_filename,
        ECAT_STATES opt_specialMode,
        int ms_timeout
)
```

Parameters

slavenum
Type: Int32
The slave device number as designated on the network to be downloaded to, 1 to N, depending upon position.

slave_filename
Type: String
File name as used on the axis (slave device).

system_filename
Type: String
File name with path as referenced from Windows.

opt_specialMode
Type: ECAT_STATES
Default to 0 for any mode, some devices must be placed in a special EtherCAT state mode depending on the file type, prior to transfer.

ms_timeout
Type: Int32
The maximum number of milliseconds to allow the device to be continously busy for. Sometimes re-flashing can cause long delays, up to 30 seconds.

Return Value

`true` if downloaded, `false` otherwise. Alternatively IncentiveAxisSupervisorException if fails where ErrCode is Axis.MOTION_FAULTS.

AxisSupervisor.FOE_DownloadByAxis Method

Download a file to an axis using the EtherCAT FOE protocol

Namespace: CTC_Incentive
Assembly: CTC_Incentive (in CTC_Incentive.dll)

Syntax

C# | VB | C++

```
public bool FOE_DownloadByAxis(
        int axis,
        string slave_filename,
        string system_filename,
        ECAT_STATES opt_specialMode,
        int ms_timeout
)
```

Parameters

axis
Type: Int32
The axis to be downloaded to, 1 to N.

slave_filename
Type: String
File name as used on the axis (slave device).

system_filename
Type: String
File name with path as referenced from Windows.

opt_specialMode
Type: ECAT_STATES
Default to 0 for any mode, some devices must be placed in a special EtherCAT state mode depending on the file type, prior to transfer.

ms_timeout
Type: Int32
The maximum number of milliseconds to allow the device to be continously busy for. Sometimes re-flashing can cause long delays, up to 30 seconds.

Return Value

`true` if downloaded, `false` otherwise. Alternatively IncentiveAxisSupervisorException if fails where ErrCode is Axis.MOTION_FAULTS.

Note that if firmware is being re-flashed the EtherCAT network must be stopped or at a minimum the slave node that will be updated must be placed offline. A request to stop the EtherCAT network can be made using the AxisSupervisor stopEtherCAT() function.

AxisSupervisor.stopEtherCAT Method

Stop the EtherCAT network. MSB connections will be cleared so best to close axis connections prior to this call. Usually used prior to FOE firmware download.

Namespace: CTC_Incentive
Assembly: CTC_Incentive (in CTC_Incentive.dll)

Syntax

C# | VB | C++

```
public bool stopEtherCAT(
        int timeout_ms
)
```

Parameters

timeout_ms
Type: Int32
Optional timeout in milliseconds, -1 is forever waiting for OFFLINE state, 500 is min value (0), 5000 recommended.

Return Value

`true` if successful, `false` otherwise.

To upload configuration files the FOE_UploadBySlave or FOE_UploadByAxis functions can be used:

AxisSupervisor.FOE_UploadBySlave Method

Upload a file from a slave device using the EtherCAT FOE protocol

Namespace: CTC_Incentive
Assembly: CTC_Incentive (in CTC_Incentive.dll)

Syntax

C# | VB | C++

```
public bool FOE_UploadBySlave(
        int slavenum,
        string slave_filename,
        string system_filename,
        ECAT_STATES opt_specialMode,
        int ms_timeout
)
```

Parameters

slavenum
Type: Int32
The slave device number as designated on the network to upload the file from, 1 to N, depending upon position.

slave_filename
Type: String
File name as used on the axis (slave device).

system_filename
Type: String
File name with path as referenced from Windows.

opt_specialMode
Type: ECAT_STATES
Default to 0 for any mode, some devices must be placed in a special EtherCAT state mode depending on the file type, prior to transfer.

ms_timeout
Type: Int32
The maximum number of milliseconds to allow the device to be continously busy for. Set it to 0 for now, not used for uploads.

Return Value

`true` if uploaded, `false` otherwise. Alternatively IncentiveAxisSupervisorException if fails where ErrCode is Axis.MOTION_FAULTS.

AxisSupervisor.FOE_UploadByAxis Method

Upload a file from an axis using the EtherCAT FOE protocol

Namespace: CTC_Incentive
Assembly: CTC_Incentive (in CTC_Incentive.dll)

Syntax

C# | VB | C++

```
public bool FOE_UploadByAxis(
	int axis,
	string slave_filename,
	string system_filename,
	ECAT_STATES opt_specialMode,
	int ms_timeout
)
```

Parameters

axis
Type: Int32
The axis to upload the file from, 1 to N.

slave_filename
Type: String
File name as used on the axis (slave device).

system_filename
Type: String
File name with path as referenced from Windows.

opt_specialMode
Type: ECAT_STATES
Default to 0 for any mode, some devices must be placed in a special EtherCAT state mode depending on the file type, prior to transfer.

ms_timeout
Type: Int32
The maximum number of milliseconds to allow the device to be continously busy for. Set it to 0 for now, not used for uploads.

Return Value

`true` if uploaded, `false` otherwise. Alternatively IncentiveAxisSupervisorException if fails where ErrCode is Axis.MOTION_FAULTS.

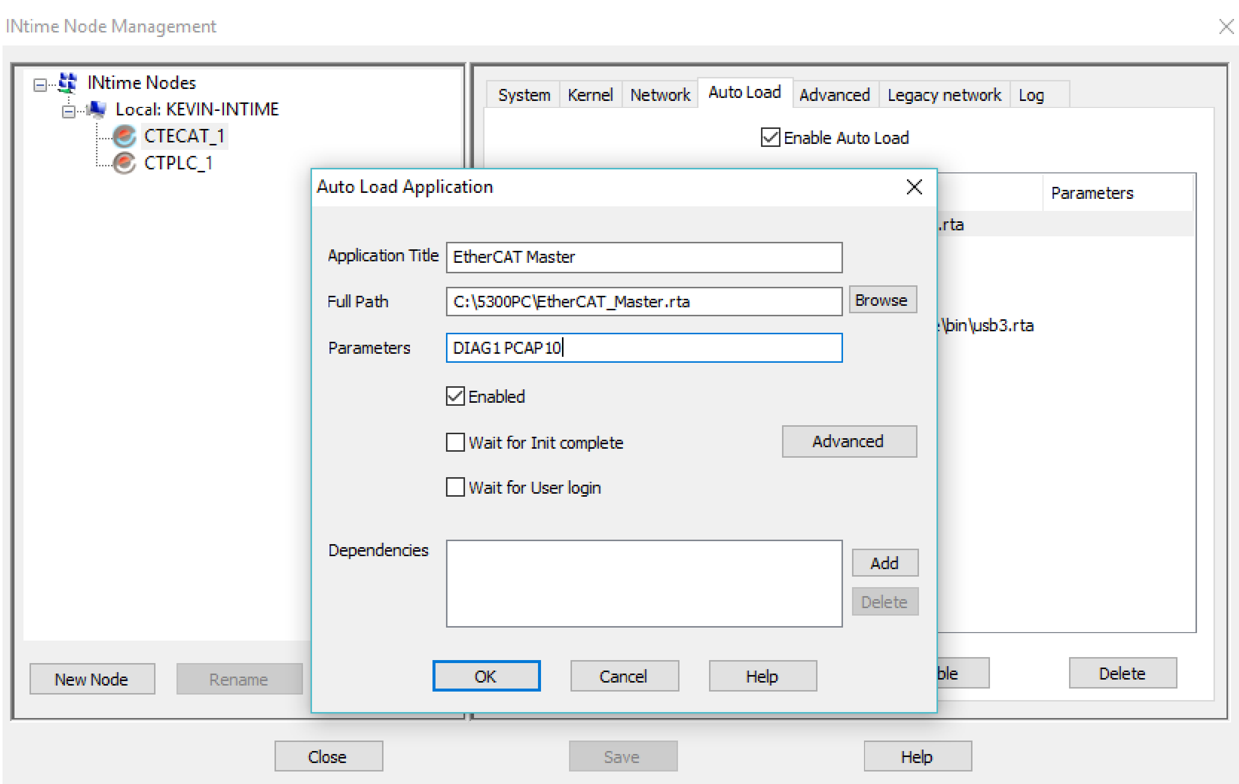
Note that the maximum length system_filename string is 358 characters and the maximum length slave_filename is limited by the device as well as a maximum of 128 by the Incentive API.

Enabling Diagnostics and EtherCAT Packet Collection (Wireshark)

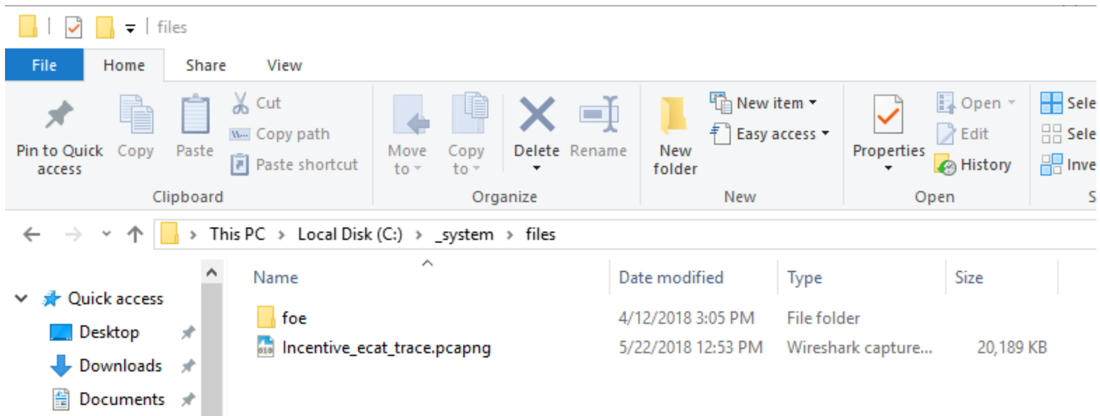
More enhanced diagnostic messages are available when running the CTPLC and CTECAT applications. This is set using the INtime Node Configuration. Select the Auto Load Application tab and edit the Parameter line for the application of interest. Both CTPLC and CTECAT accept DIAG1 for more enhanced console messages. CTECAT also accepts additional commands to save EtherCAT packets to a Wireshark compatible file: “c:_system\files\Incentive_ecat_trace##.pcapng”, where ## starts at 01 and increments to 16 each time EtherCAT is started (reseting to 01 after 16). Three unique parameters are available for packet collection:

1. PCAPSTARTUPONLY which will collect initialization packets but stop collection when the network goes operational.
2. PCAP#### where #### is the number of seconds to collect packets while operational (PCAP0 will collect forever).
3. PCAP_BACKTRACE which will record the last 512 packets each time EtherCAT stops. This is probably the most conservative with storage and results in about 62K bytes per file.

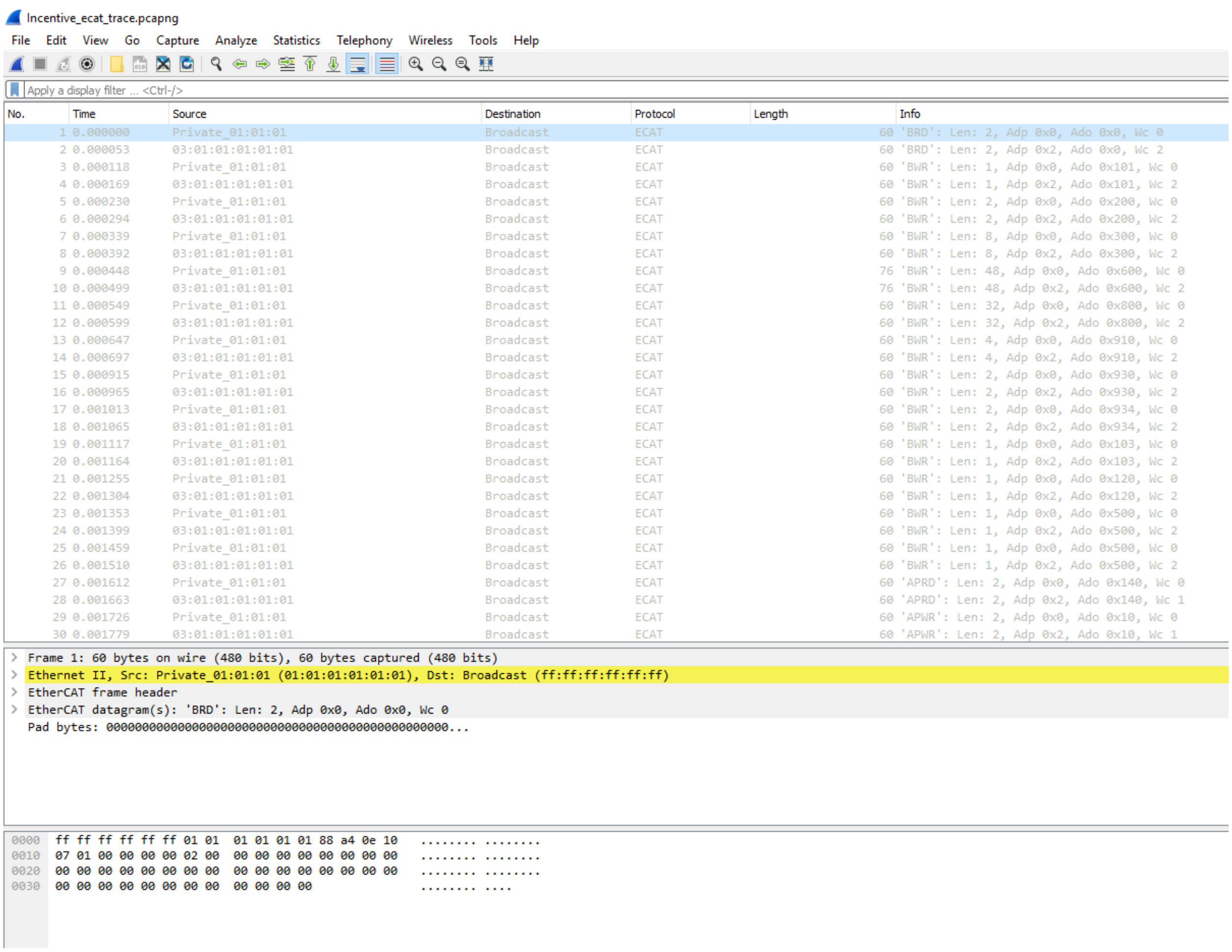
Be careful since your disk can fill up quickly since at a 1mS scan rate this is typically 5.55 megabytes/10 seconds of collection time with a base of about 3.5 megabytes for startup initialilzation. Below is an example of running DIAG1 messages as well as collecting packets at both startup and for the first 20 seconds of operation. Note that each time the network is restarted the collected packets will be overwritten. The file size will typically show 0 until it has finished collecting data.



Packets file storage location:



Loaded into Wireshark (https://www.wireshark.org/):



Typically the Wireshark file would be sent to CTC technical support for analysis.

Additional parameters, which may be used in addition to DIAG1, or independently, are available with the CTPLC_1 node are:

DIAG2 – Dump memory usage information during each log message.
DIAG3 – Log TCPSerial thread connection information as connections are initiated and disconnected.

CHAPTER
3

[3] Incentive License Manager

The Incentive License Manager is used to generate licenses for all Incentive products including QuickBuilder, real time Incentive PLC Logic, EtherCAT Master, and the Incentive API. The license process if fairly simple and consists of receiving a License ID and password from CTC for the products purchased. The ID and password are then entered into the License Manager where it is validated with an online server and license file created. The license will then be locked to the computer you are using. By using an online server licenses can later be refreshed for any purchased upgrades or re-activated should your computer become damaged, demo times extended, etc.

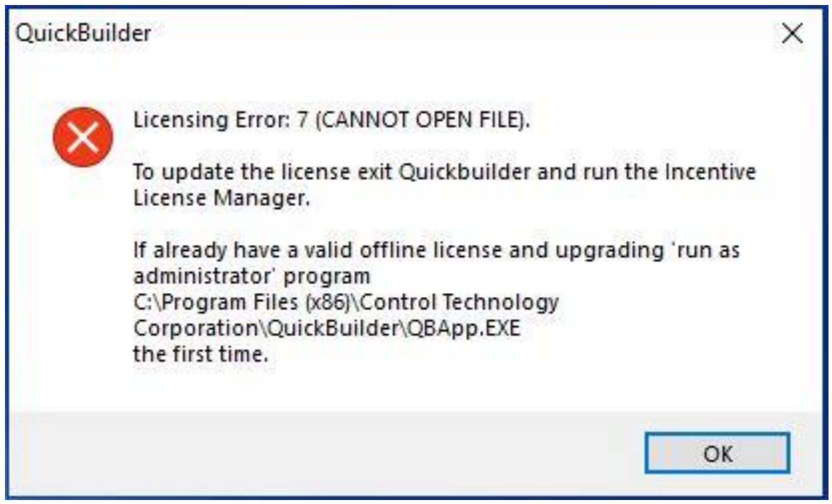
Licenses for the real-time EtherCAT environment require an additional product, the Incentive License Monitor Service. It is a background service that is used to authenticate your license with the real-time EtherCAT and PLC environment each time it is invoked. QuickBuilder does not need the service to be running since it is a Windows based program and can validate directly.

Installation

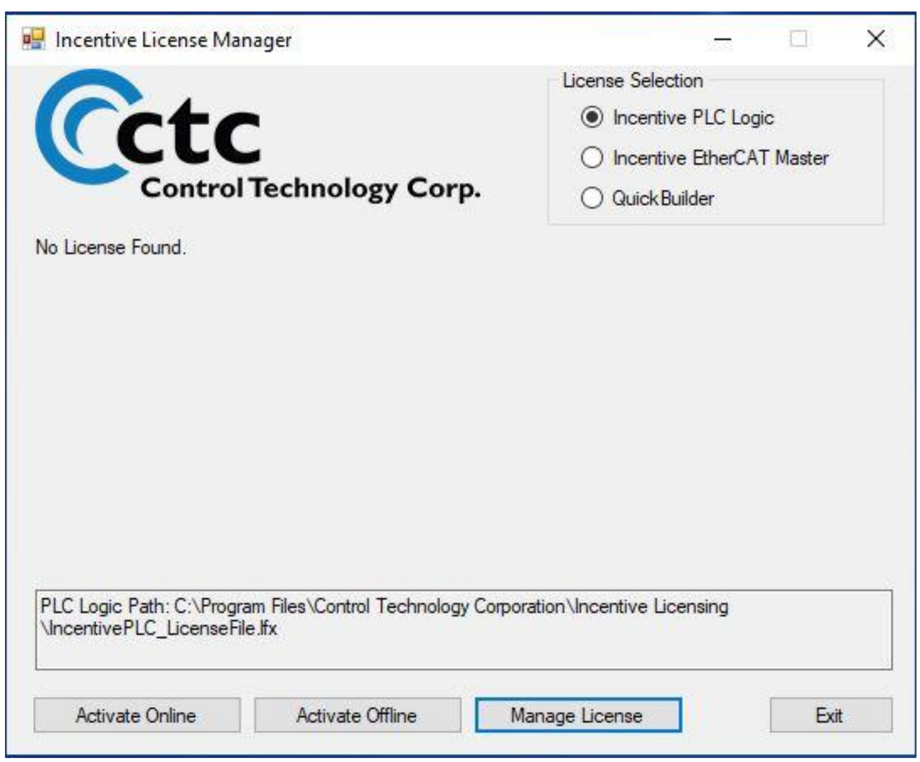
The Incentive License Manager can be installed by executing the "Incentive License Setup.msi" installation file. Follow the prompts for installation and it is suggested you accept the default installation location as all licenses will be store there once generated. You will also need administrative rights and Windows should prompt you accordingly. Upon installation, should you need to run the real-time Incentive environment for EtherCAT control you would next select "Incentive License Monitor – Install" from your Windows CTC Programs menu. This will install a background license service. Installation is automatic and it will open up a Windows dialog to run as administrator so the program can be installed. The service will automatically start upon boot and monitor for available licenses. If a license is updated the service will have to be restarted for it to take effect.

QuickBuilder Licensing

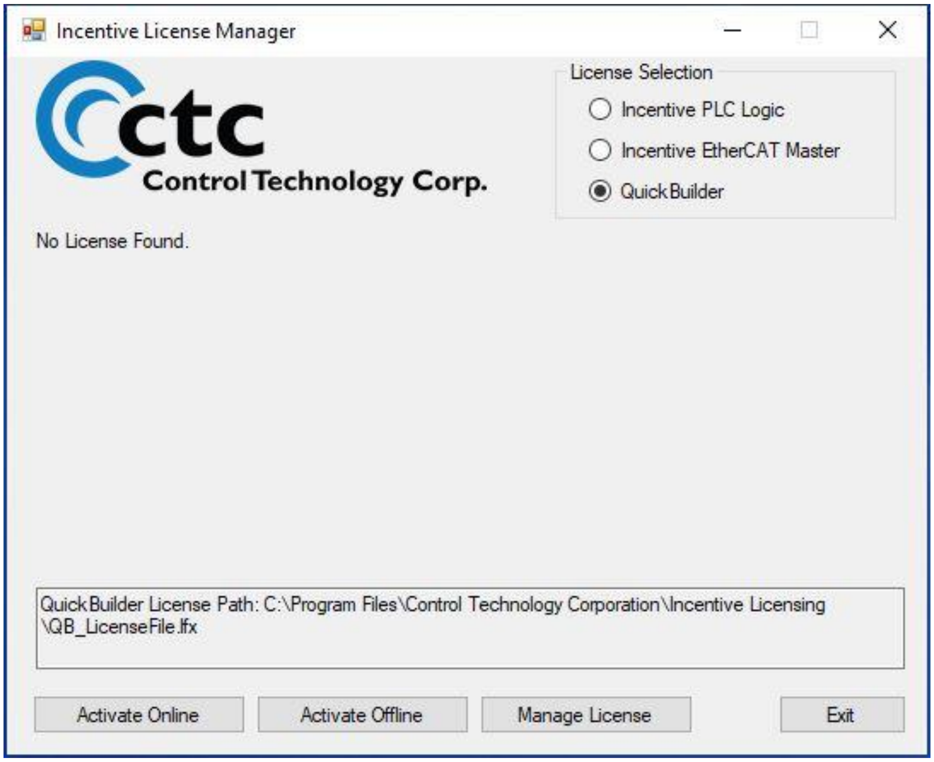
QuickBuilder requires a license both as a fully licensed product and in demo mode. Older existing installations, which are only upgrading, use a different licensing technique and will automatically detect that license thereby not requiring the use of the License Manager. In a new installation should you attempt to execute QuickBuilder after installation a dialog box will appear warning of the license requirement:

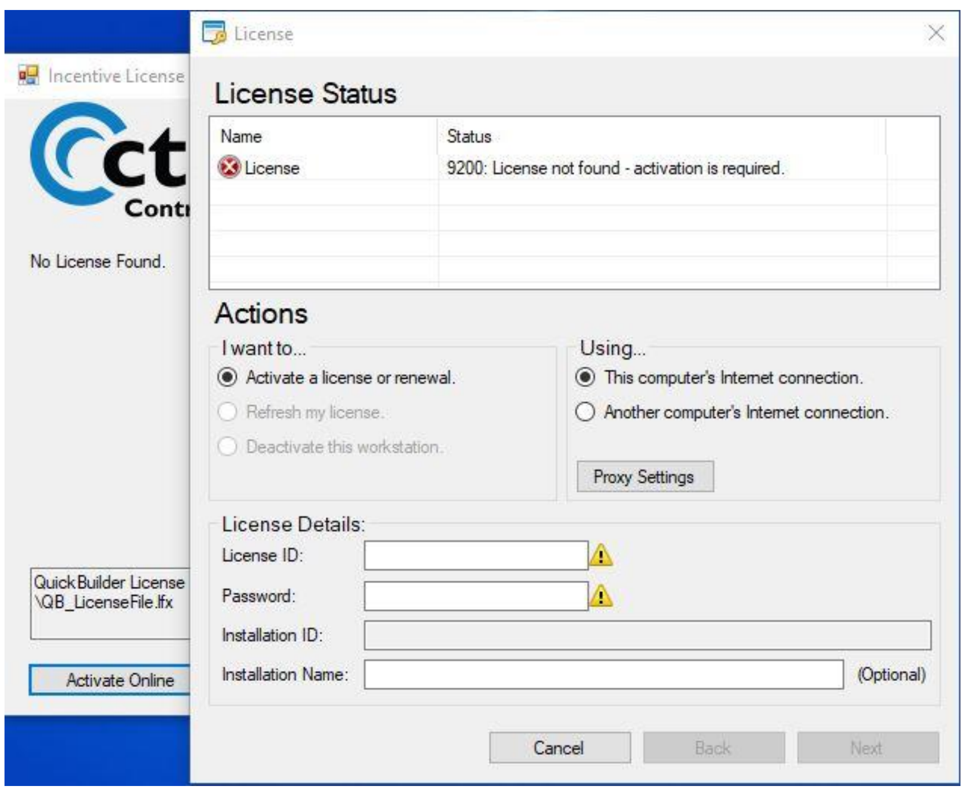


In order to activate invoke the Incentive License Manager from the Windows CTC Program Menu and click Yes when prompted by the User Access Control prompt to allow changes to your PC.

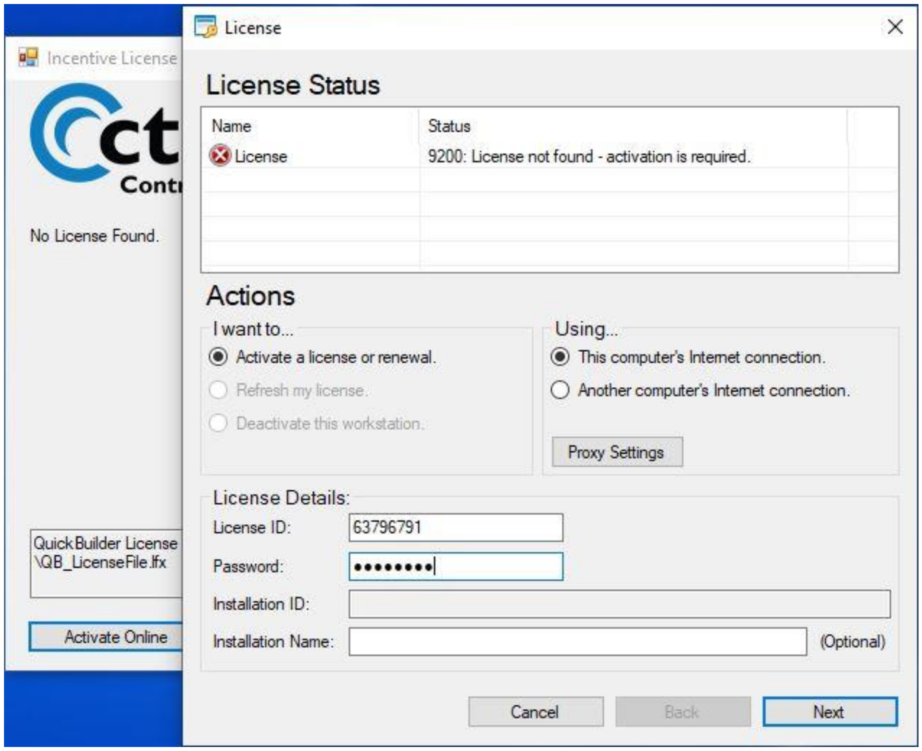


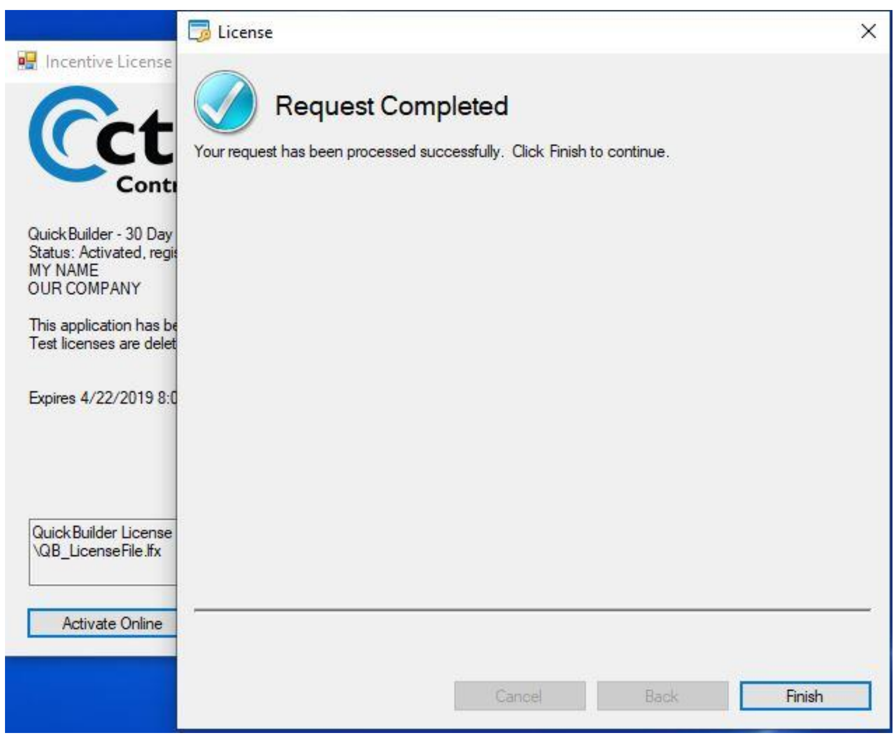
Select 'QuickBuilder' under the 'License Selection' followed by Activate Online.



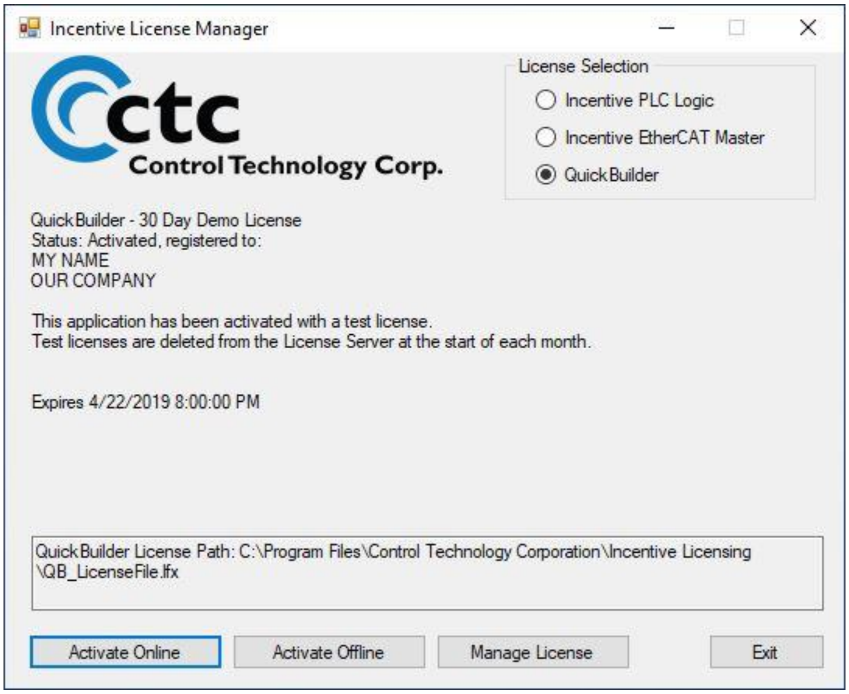


CTC must supply you with a License ID and a Password. Once this is received you may click 'Next' for the validation to be verified with the online server.





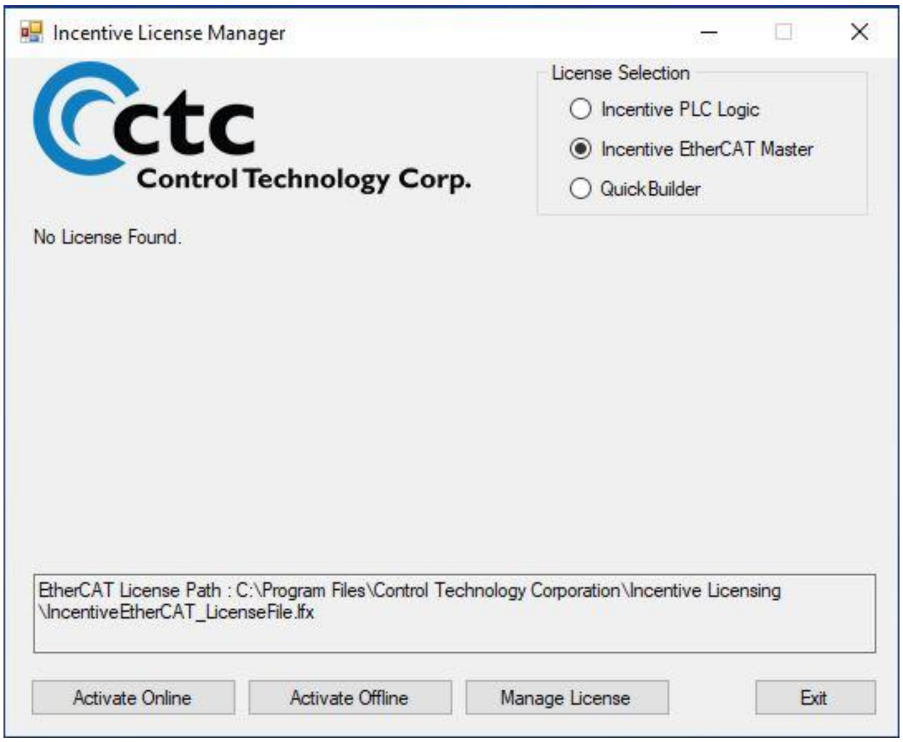
The online server is provided by a third party, SoftwareKey.com, whom handles the license validation. CTC subscribes to their service to maintain its licenses. There are two types of licenses, time limited, and full. A time limited is considered a demo and is typically for a little less than 30 days. The license works on Universal time, that without a time zone so your license is shorter or longer by the distance you are from GMT. The Expiration shown by the License Manager and Quickbuilder is that of your local time. A demo license can be refreshed to extend it with the permission of CTC or changed to a full license at any time, also by refreshing and being authorized by CTC.



Above is shown a demo time limited license. When you are ready to activate the full license simply place the order with CTC and the license will be set to a full license. Alternatively, you can request an extension to your demo. To update the license click Manage License and 'Refresh my license'. Once refreshed the license will be updated with the requested changes. The location of the license file is shown on the form, 'QuickBuilder License Path'.

Incentive PLC Logic and EtherCAT Master Licensing

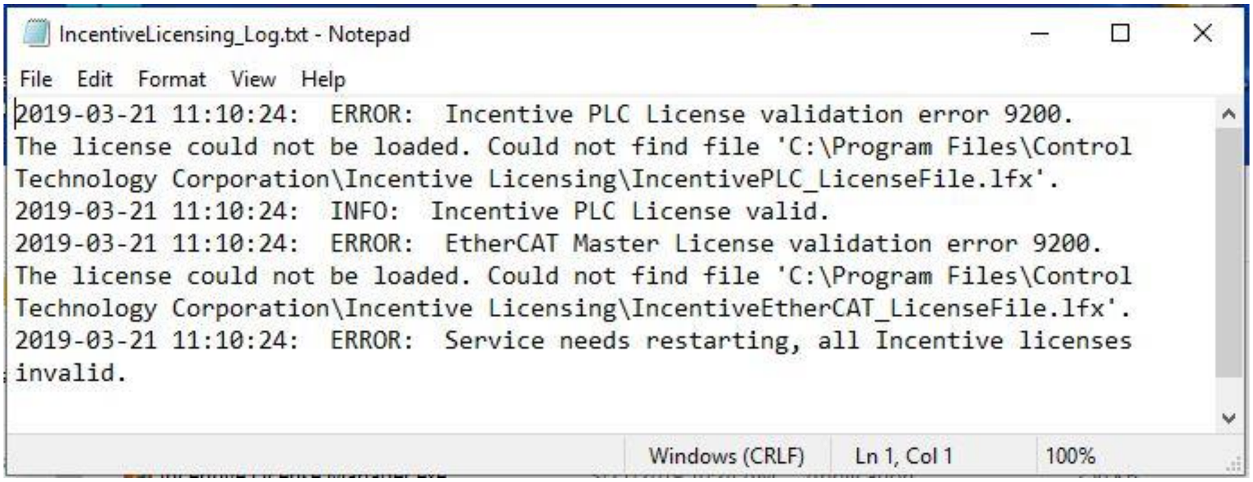
Both the Incentive PLC Logic and EtherCAT Master products require a license for full operation. If a license is not available the IO Console for each will warn you that it is running in demo mode, after about 3 hours it will shutdown and require restarting to run another 3 hours. Licenses are entered exactly as the QuickBuilder license was except that the License Selection radio button will be for the desired license. For example below shows Incentive EtherCAT Master:



Once licensed a licensing service must be installed since the Incentive PLC Logic and EtherCAT Master do not run under the control of Windows, they execute under the real-time TenAsys INtime operating system. The service is need to act as a proxy, validating the Windows based licenses and authorizing the real-time execution to commence. To install the service, select 'Incentive License Monitor – Install' from the Program Menu. Allow it to run to completion and the service will be installed and automatically started, restarting whenever Windows runs. Should you install or update a license then the License Monitor service should be restarted either manually or reboot your computer.

NMake sure the TenAsys INtime environment is fully installed and configured for CTPLC_1 and CTECAT_1 prior to installing the Incentive PLC Logic or EtherCAT Licenses. Changing the number of cores assigned to Windows causes the license to view the computer as being different and will invalidate the license if installed prior to INtime. Should a problem occur CTC Technical Support can increase your activations so you can reactivate with the same ID and password.

During execution the service maintains an activity log in a file called 'IncentiveLicensing_Log.txt'. This log file is located in the installation directory where the license was stored, appearing at the bottom of the Incentive License Manager form. For example, if no licenses are present the following logs will appear:



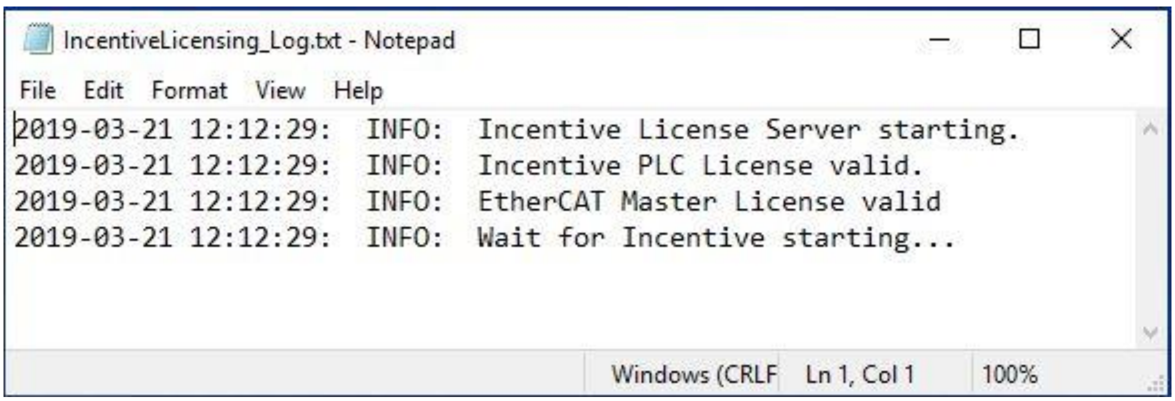
IncentiveLicensing_Log.txt - Notepad

File Edit Format View Help

```
2019-03-21 11:10:24:  ERROR:  Incentive PLC License validation error 9200.
The license could not be loaded. Could not find file 'C:\Program Files\Control
Technology Corporation\Incentive Licensing\IncentivePLC_LicenseFile.lfx'.
2019-03-21 11:10:24:  INFO:  Incentive PLC License valid.
2019-03-21 11:10:24:  ERROR:  EtherCAT Master License validation error 9200.
The license could not be loaded. Could not find file 'C:\Program Files\Control
Technology Corporation\Incentive Licensing\IncentiveEtherCAT_LicenseFile.lfx'.
2019-03-21 11:10:24:  ERROR:  Service needs restarting, all Incentive licenses
invalid.
```

Windows (CRLF) Ln 1, Col 1 100%

If both licenses were valid the following would appear and the License Monitor would wait for the real-time Incentive environment to start, once started it would authorize full execution.



IncentiveLicensing_Log.txt - Notepad

File Edit Format View Help

```
2019-03-21 12:12:29:  INFO:  Incentive License Server starting.
2019-03-21 12:12:29:  INFO:  Incentive PLC License valid.
2019-03-21 12:12:29:  INFO:  EtherCAT Master License valid
2019-03-21 12:12:29:  INFO:  Wait for Incentive starting...
```

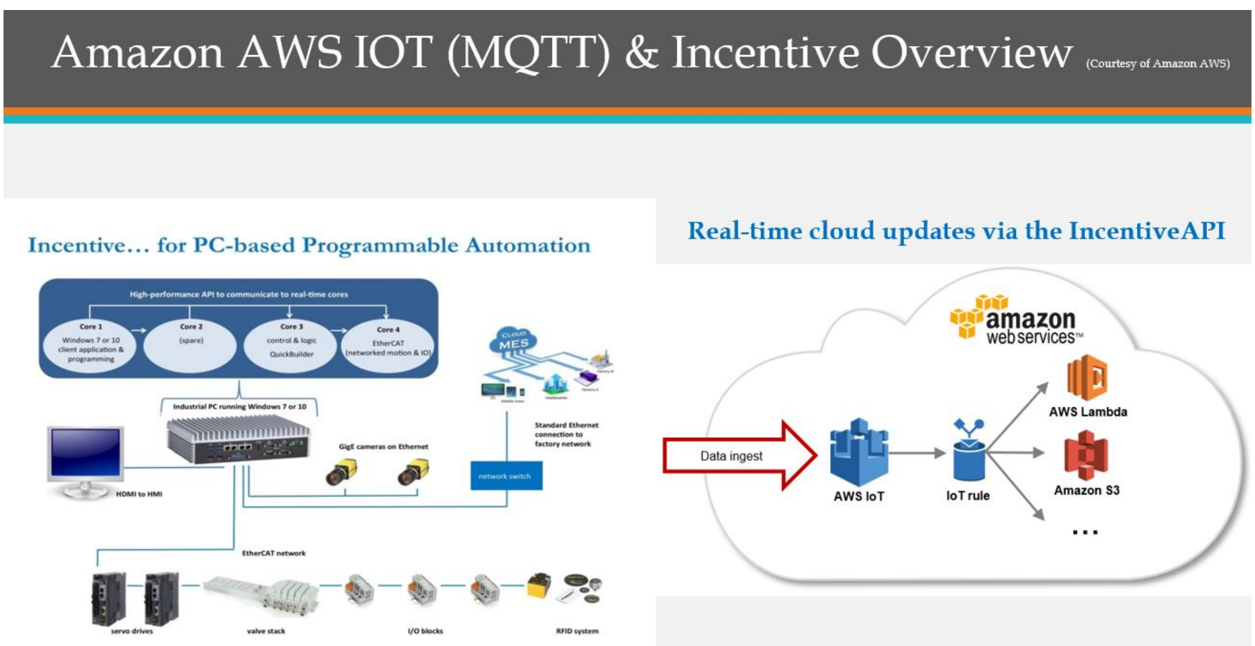
Windows (CRLF Ln 1, Col 1 100%

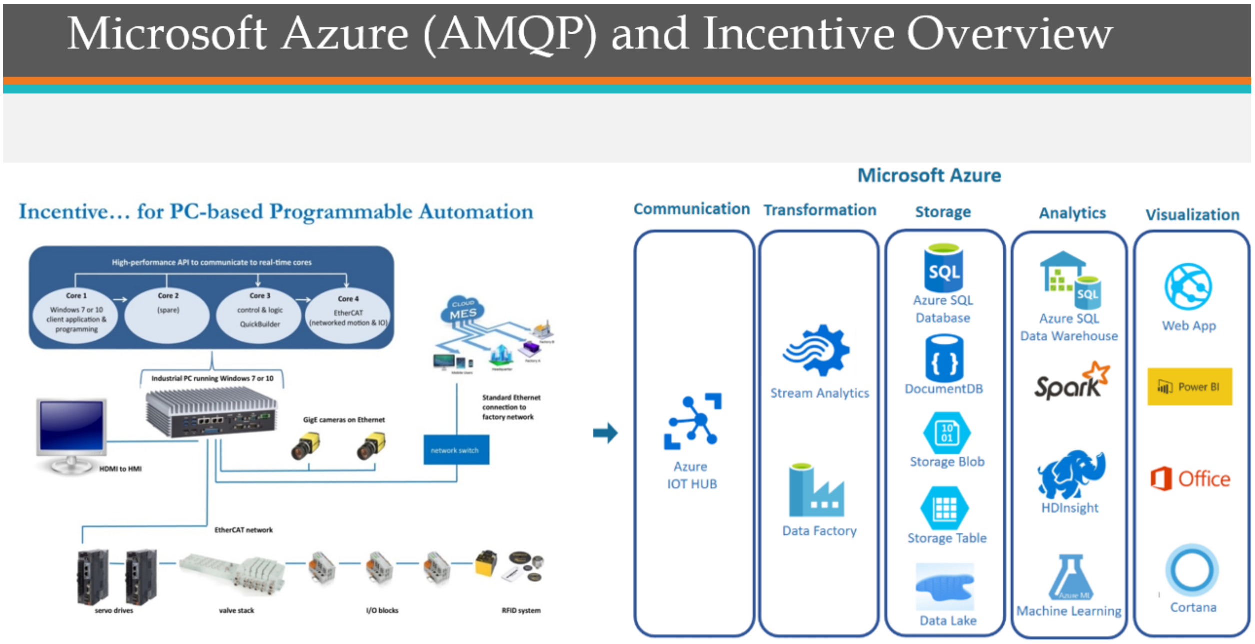
Blank

CHAPTER 4

[4] IIS & IIOT Solutions

Incentive includes both IIS (Internet Information Service) and IIOT (Industiral Internet of Things) solutions. A soap webservice interface to the IncentiveAPI is available to allow for remote access of device information, alternatively or in addition to an IIOT solution. IIOT is the industrial version of the consumer oriented Internet of Things (IOT) where devices can seamlessly interact with the cloud. The Incentive module 'IncentiveAPI' includes a full cloud interface to Amazon AWS and Microsoft Azure Hub IOT. The interface to IIOT is simplified by the same managed DLL that implements a real-time interface to IncentivePLC and IncentiveECAT. Simple connectIOT, publishIOT, and subscribeIOT methods are available to send and receive messages. Both MQTT (X.509 certificates) and AMQP (SAS Tokens) protocols are supported using secure TLS 1.2 communications. Securely send and receive messages globally. The use of the API allows redirection of messages to databases, analytics processing, diagnostic monitoring, recipe distribution, and numerous other solutions that may be required by an enterprise.

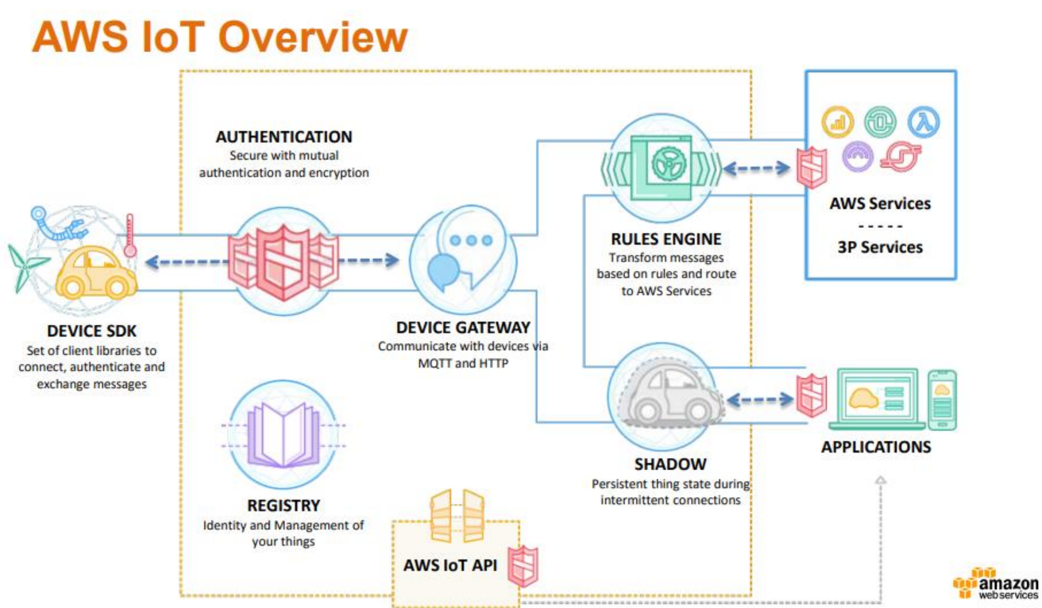




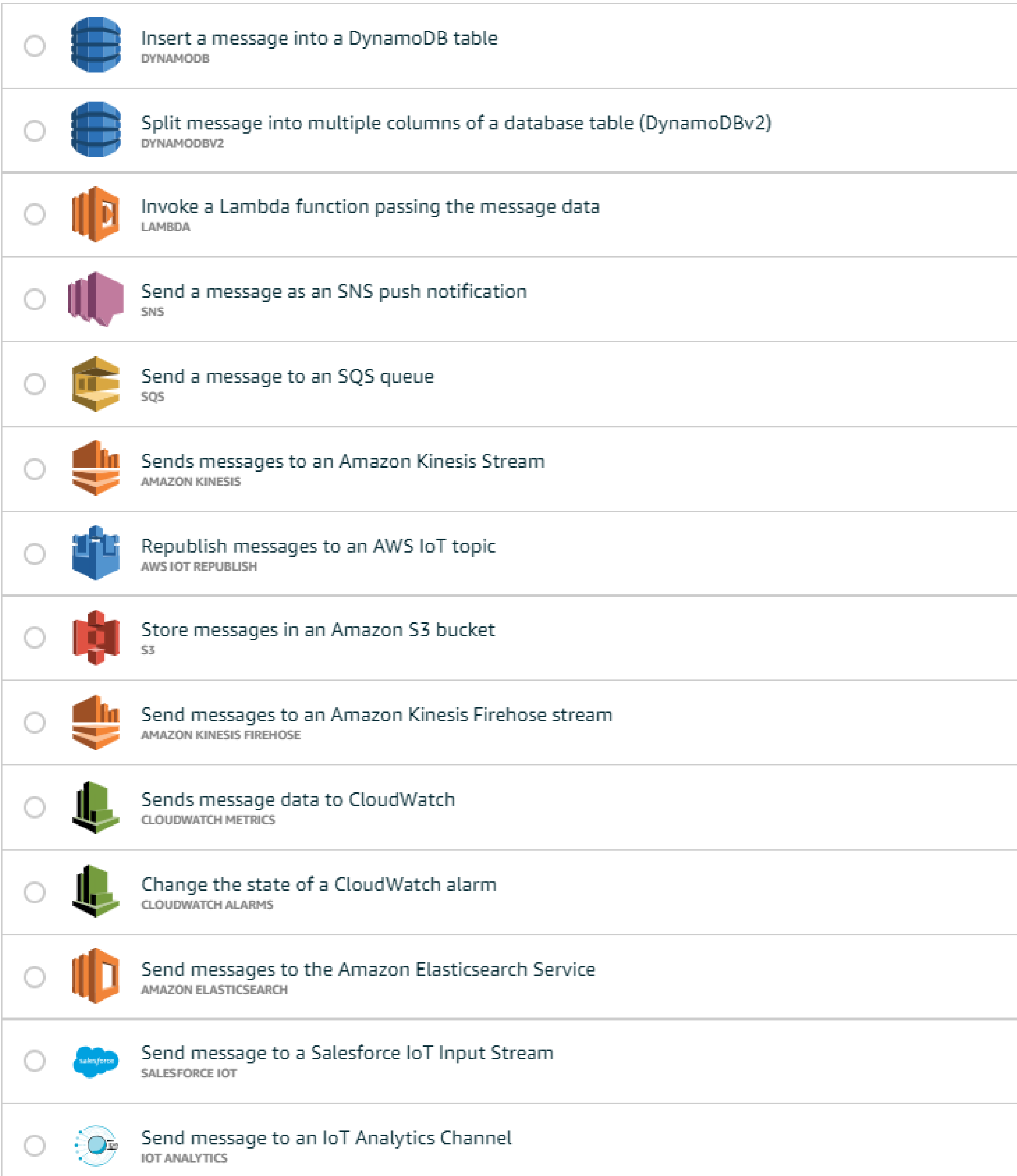
Support is only available within the IncentiveAPI, thus not available for the 5300 PLC controller.

Amazon AWS IoT (Internet of Things) API Support

The Incentive API contains a cloud interface to Amazon's AWS IoT Core using the MQTT protocol and X.509 certificate security. The complexity of this connection is hidden from the user and allows device data to be easily sent to the cloud for storage, monitoring by IoT subscribers, peer to peer communications to other Incentive devices, as well as database storage. A good high level overview and video of Amazon IoT is available at https://aws.amazon.com/iot/ and https://aws.amazon.com/iot-core/, as well as https://aws.amazon.com/iot/solutions/industrial-iot/. From an Amazon presentation the below gives an overview of their structure:



The services offered by Amazon once the data reaches the cloud are vast and can range anywhere from complex analytics to simple data collection. Some of the actions available are:



For additional information on the options available please refer to the amazon AWS IoT documentation https://aws.amazon.com/documentation/.

The creation of an Amazon AWS IoT Thing is left to the user and beyond the scope of this document. It will be necessary to have one or more Topics available to send data to. Some documents that are helpful are available at:

https://docs.aws.amazon.com/iot/latest/developerguide/register-device.html

In addition to creating a 'Thing' you will need to create a Certificate, Rules and a Policy. Screen shots of the IOT Thing created for the C# demo included with the Incentive distribution are included within the appendix.

Mqtt_IOT Class

The API contains support for the Internet of Things (IoT) MQTT communication protocol to enable a simple interface to the cloud for storage. Amazon AWS IOT and Microsoft Auzre IOT Hub are presently supported, with the Mqtt_IOT Class typically used for Amazon AWS. The Mqtt_IOT class puts a wrapper around the open source MQTT client software from the Eclipse Paho project: https://www.eclipse.org/paho/clients/c/. A very limited ability is exposed to simplify the interface and the DLL (M2Mqtt.Net.dll) is included with our API distribution. If for some reason the IncentiveAPI is missing some feature you require you can always add the DLL as a resource in your project and work with it directly. The creation of an Amazon account and AWS IoT device is beyond the scope of this document and available online from Amazon: https://docs.aws.amazon.com/iot/latest/developerguide/iot-console-signin.html.

To simplify the interface a generic connectIOT, publishIOT, subscribeIOT, unsubscribeIOT, and disconnectIOT methods have been added to the class. The C# example program distributed with Incentive contains additional code showing the publishing of data to Amazon AWS IoT as well as how to receive data from IoT topics, upload and download files (subclass S3 from within class TestDynamoDBandFileUploadDownload), even update a database using Amazon's DynamoDB. File uploading/downloading and the DynamoDB interface has been kept out of the DLL and only included within the C# example program so that users can customize their programs and not have to maintain a large overhead of modules and libraries that may not be needed for building their application. Note that methods may throw an MqttConnectionException with an MqttClientErrorCode or MqttClientException.

Methods

	Name	Description
	connectIOT	Connect to IOT client using MQTT protocol over TLS 1.2, port 8883 is suggested.
	disconnectIOT	Disconnect the client IOT session.
	Equals(System.Object)	Determines whether the specified Object is equal to the current Object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as a hash function for a particular type. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	publishIOT(String, String, Boolean)	Publish a message asynchronously and optionally wait for acknowledgement it was received (QoS Level 1, wait for ACK).
	publishIOT(String, Byte[], Boolean)	Publish a message asynchronously and optionally wait for acknowledgement it was received (QoS Level 1, wait for ACK).
	subscribeIOT	Subscribe to receive messages asynchronously.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	unsubscribeIOT	Unsubscribe from any topics wish to stop receiving messages from.

Fields

	Name	Description
	publishEventWaitHandle	publishEventWaitHandle AutoResetEvent
	QOS_LEVEL_AT_LEAST_ONCE	QOS Level At Least Once
	QOS_LEVEL_AT_MOST_ONCE	QOS Level At Most Once
	QOS_LEVEL_EXACTLY_ONCE	QOS Level Exactly Once

Events

	Name	Description
	Callback	MessageReceivedCallBack event. This event must be set by the application so that any topics subscribed to will have the callback routine invoked upon reception of a message.

Mqtt_IOT.connectIOT Method

Connect to IOT client using MQTT protocol over TLS 1.2 on port 8883.

Namespace: CTC_Incentive
Assembly: CTC_Incentive (in CTC_Incentive.dll)

Syntax

C# VB C++

```
public bool connectIOT(
        string CustomEndpoint,
        byte QOS_Level,
        string clientCertFilePath,
        string clientCertPassword,
        string caCertFilePath,
        ushort keepalive
)
```

Parameters

CustomEndpoint
Type: String
Custom endpoint that allows you to connect to AWS IoT. You may obtain it from https://aws.amazon.com/console/ after you login. I will look something like "q1b1s8partcxdl.iot.us-east-1.amazonaws.com".

QOS_Level
Type: Byte
QOS Level 0, 1, or 2, level 1 recommended (QOS_LEVEL_AT_LEAST_ONCE, wait for ACK).

clientCertFilePath
Type: String
Client X509 certificate file path and name that was created using openssl. Example - C:\OpenSSL-Win64\bin\openssl pkcs12 -export -out f366b02e9c-certificate.pfx -inkey f366b02e9c-private.pem.key -in f366b02e9c-certificate.pem.crt -certfile root-CA.crt

clientCertPassword
Type: String
Password given to openssl when creating the client X509 certificate.

caCertFilePath
Type: String
Root X509 certificate file path and name that is used by IOT site. For Amazon AWS this is available from https://www.symantec.com/content/en/us/enterprise/verisign/roots/VeriSign-Class%203-Public-Primary-Certification-Authority-G5.pem

keepalive
Type: UInt16
Number of seconds of inactivity to wait before send an empty packet letting hub know we still want to maintain connection. 60 is recommended.

Return Value

`true` if connected. `false` otherwise. MqttConnectionException on connect error.

Mqtt_IOT.disconnectIOT Method

Disconnect the client IOT session.

Namespace: CTC_Incentive
Assembly: CTC_Incentive (in CTC_Incentive.dll)

Syntax

C# VB C++

```
public void disconnectIOT()
```

Mqtt_IOT.publishIOT Method

Publish a message asynchronously and optionally wait for acknowledgement it was received (QoS Level 1, wait for ACK).

Namespace: CTC_Incentive
Assembly: CTC_Incentive (in CTC_Incentive.dll)

Syntax

C# VB C++

```
public ushort publishIOT(
        string IOT_Topic,
        string iot_data,
        bool waitPublished
)
```

Parameters

IOT_Topic
Type: String
Message topic

iot_data
Type: String
Message data (payload)

waitPublished
Type: Boolean
If true we will block until we get an acknowledgement sent else just queue it

Return Value

Message Id related to PUBLISH message, MqttClientException if error.

Mqtt_IOT.subscribeIOT Method

Subscribe to receive messages asynchronously.

Namespace: CTC_Incentive
Assembly: CTC_Incentive (in CTC_Incentive.dll)

Syntax

C# | VB | C++

```
public ushort subscribeIOT(
        string[] IOT_Topics,
        byte[] QOS_levels
)
```

Parameters

IOT_Topics
Type: String[]
string array of topics to subscribe to (receive from).

QOS_levels
Type: Byte[]
MqttMsgConnect.QOS_LEVEL... QOS level subscribing at. Typically MqttMsgConnect.QOS_LEVEL_AT_LEAST_ONCE.

Return Value

Message Id related to PUBLISH message, MqttClientException if error.

Mqtt_IOT.unsubscribeIOT Method

Unsubscribe from any topics wish to stop receiving messages from.

Namespace: CTC_Incentive
Assembly: CTC_Incentive (in CTC_Incentive.dll)

Syntax

C# | VB | C++

```
public ushort unsubscribeIOT(
        string[] IOT_Topics
)
```

Parameters

IOT_Topics
Type: String[]
string array of topics unsubscribing from

Return Value

Message Id related to subscribed message

Sample Mqtt_IOT Class usage

```
/// <summary>
/// IOT_Test class for testing API access to Amazon AWS..
/// </summary>
class IOT_Test
{
   // This is the Amazon Topic or Azure Device to which messages are published (Send) or subscribed to (Receive).
   public string IOT_Topic_Device = "IncentivePLC_Data";
   // Queue to store received message to for testing
   public Queue subscribed_Data = new Queue();

   // Generic callback for subscriber messages received.  Each message is placed in the queue for later use.  'handle' is used by Azure but not
   //Amazon AWS interface.
   // With Azure there is a receive thread for each subscription, with Amazon all subscriptions use the same thread.  'topic' is the device
   // subscribed to whereas 'data' is what is received.
   private void subscribedMessageCallback(int handle, string topic, byte[] data)
   {
      string ReceivedMessage = topic + ":  " + Encoding.UTF8.GetString(data);  // This assumes the message is a string, modify as desired.
      subscribed_Data.Enqueue(ReceivedMessage);
   }

   // ***************** Amazon AWS IOT Hub Access (Modify for your account setup) ************************
   // You can obtain AWSCustomEndpoint (Custom endpoint) from the Settings menu on the navigation bar of the AWS IoT Console
   // https://aws.amazon.com/console/
   public string AWSCustomEndpoint = "a1g1s7parycwdl.iot.us-east-1.amazonaws.com";
   // To create the client certificate use openssl and run command line as follows from the directory that has the certificates downloaded from
   // AWS:
   // C:\OpenSSL-Win64\bin\openssl pkcs12 -export -out f366b02e9c-certificate.pfx -inkey f366b02e9c-private.pem.key -in f366b02e9c-
   /   certificate.pem.crt -certfile root-CA.crt
   // openssl is available from:  https://slproweb.com/download/Win64OpenSSL_Light-1_1_0h.exe
   public string clientCertFilePath = "C:\\Users\\Kevin\\Documents\\Control_Technology_Corporation\\iot\\mqtt\\certificates\\f366b02e9c-
   certificate.pfx";
   // Password entered when prompted when openssl was run creating the pfx file (your private password to generate your certificate):
   public string clientCertPassword = "Incentive";
   // AWS root certificate is opened from Symantec and copied to a file called root-CA.crt:
   // https://www.symantec.com/content/en/us/enterprise/verisign/roots/VeriSign-Class%203-Public-Primary-Certification-Authority-G5.pem
   public string caCertFilePath = "C:\\Users\\Kevin\\Documents\\Control_Technology_Corporation\\iot\\mqtt\\certificates\\root-CA.crt";
   // IOT Topic details are available:  https://docs.aws.amazon.com/iot/latest/developerguide/topics.html

   public void testAmazon_IOT()
   {
      // Request and IOT communications interface from the Incentive API for MQTT protocol which is used by Amazon AWS.
      Mqtt_IOT myMqtt_iot = new Mqtt_IOT();

      // Connect to the Amazon AWS IOT cloud, TLS protocol is used on the standard port 8883.
      if (!myMqtt_iot.connectIOT(AWSCustomEndpoint, 8883, myMqtt_iot.QOS_LEVEL_AT_LEAST_ONCE, clientCertFilePath, clientCertPassword,
         caCertFilePath, 60))
      {
         return;  // Connection failed
      }

      // Add the callback function so can queue messages received
      myMqtt_iot.Callback += new Mqtt_IOT.MessageReceivedCallBack(subscribedMessageCallback);

      // As a test we will subscribe to the same topic we are publishing to so should get the message back after we send it to the cloud.
      string[] subscribeList = new string[1];  // List of desired topics, in this case just the one.
      subscribeList[0] = IOT_Topic_Device;
      byte[] QOS_List = new byte[1];  // QOS level desired
      QOS_List[0] = myMqtt_iot.QOS_LEVEL_AT_LEAST_ONCE;
      // Subscribe to the desired IOT Topics, in this case just one.  When done the same messages sent should be available in
      // mMqtt_iot.subscribed_Data queue.
      myMqtt_iot.subscribeIOT(subscribeList, QOS_List);

      // Generate some random data to send to the cloud, in this case send 30 IOT messasges which will be redirected to the Amazon DynamoDB
```

```
        // database for later reference
        int key = 1;
        string timeStamp;
        string iot_data;
        float temperature = 400.0F;
        for (int i = 0; i != 30; i++)
        {
          // Always useful to have a timestamp on an IOT entry
          timeStamp = DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss.fff");
          // This can be written as desired but for this example it is a json record for writing to a DynamoDBv2 database
          // In this particular example the DynamoDB primary key field name is 'key' and all the other fields are created dynamically by the
          // database.  Only the primary key and an optional sort key are always required and defined ahead of time
          // Make up some data where we keep the key unique by incrementing each time and just bump the temp data so can see some changes in
          // the database later when we look at it.  As a side note if the key does not change the record will be overwritten for that key.
          iot_data = "{ \"timestamp\" : \"" + timeStamp + "\", \"key\" : \"" + key + "\", \"deviceid\" : \"Monitor_01\", \"temp\" : " + temperature +
            ", \"humidity\" : 47.463, \"Co2_level\" : 644, \"lighting_level\" : 85, \"pressure\" : 28, \"light_state\" : 1, \"CO Alarm\" : 0 }";
          // The IOT_Topic can change for different data using the same connection.  This is useful for having AWS do different things
          // with different topics are used.
          ushort result = myMqtt_iot.publishIOT(IOT_Topic_Device, iot_data, true);
          MessageBox.Show("Published " + key + ". Result = " + result);
          key++;  // bump the key
          temperature = temperature + 25.2F;  // Arbitrarily bump the temperature data so see changes in the database
        }
        // Unsubscribe from topic service using, if set a breakpoint here we could look at 'subscribed_Data' Queue and would have all published
        // messages
        // that we received back in it.
        myMqtt_iot.unsubscribeIOT(subscribeList);

        // All records published so disconnect from Amazon AWS now that we are done
        myMqtt_iot.disconnectIOT();
      }
    }
```

Mqtt_IOT Certificate Generation

Open_sll from https://slproweb.com/download/Win64OpenSSL_Light-1_1_0h.exe is used to create the required security certificate (assuming 64 bit Windows it is included with the installation in the '..\Control_Technology_Corporation\iot\mqtt' directory). Once installed and the appropriate Amazon AWS certificates are downloaded the following command line will generate the needed certificate (pfxcreation.bat in the '..\Control_Technology_Corporation\iot\mqtt\certificates' directory):

C:\OpenSSL-Win64\bin\openssl pkcs12 -export -out f366b02e9c-certificate.pfx -inkey f366b02e9c-private.pem.key -in f366b02e9c-certificate.pem.crt -certfile root-CA.crt

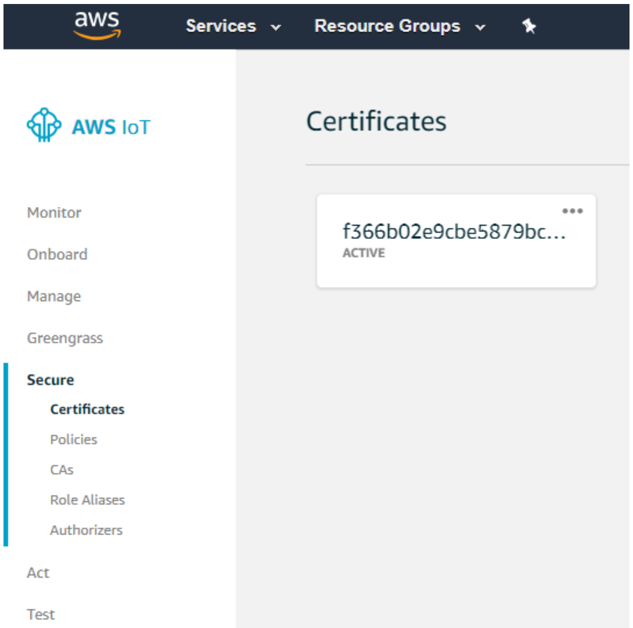
Also remember to save the password used to generate the security certificate as it will be needed by your application when accessing the API for the MQTT connection (connectIOT).

The referenced certificates used for input may be found as follows in the iot\mqtt\certificates folder:

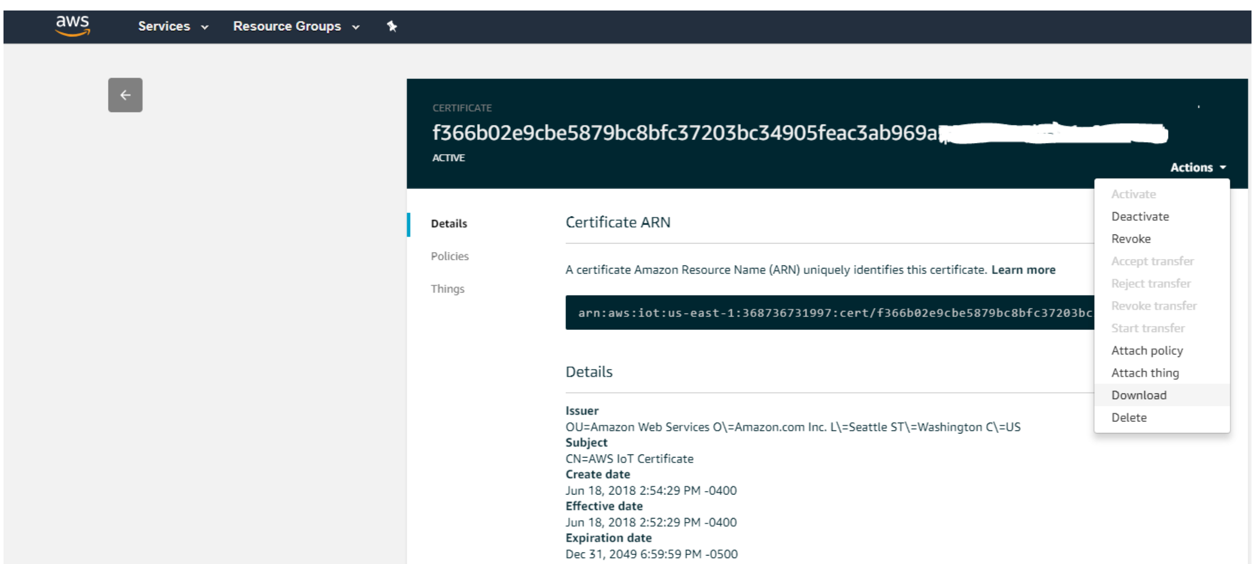
f366b02e9c-private.pem.key – The AWS IoT Thing's private key which is available at the time you create the certificate. Reference Amazon docs https://docs.aws.amazon.com/iot/latest/developerguide/create-device-certificate.html.

f366b02e9c-certificate.pem.crt – Certificate attached to your AWS IoT Thing. See the private key creation and download it then or it may be downloaded later via the below screen.

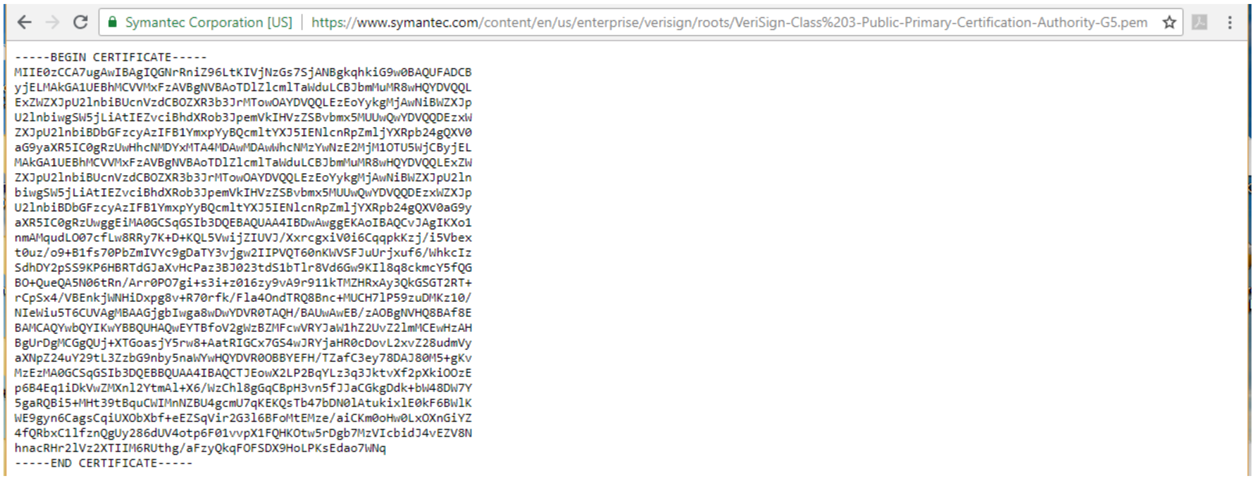
Should you use your own certificate and the format need changing openssl can take care of that for you. Reference the information at the following link:
https://www.digitalocean.com/community/tutorials/openssl-essentials-working-with-ssl-certificates-private-keys-and-csrs



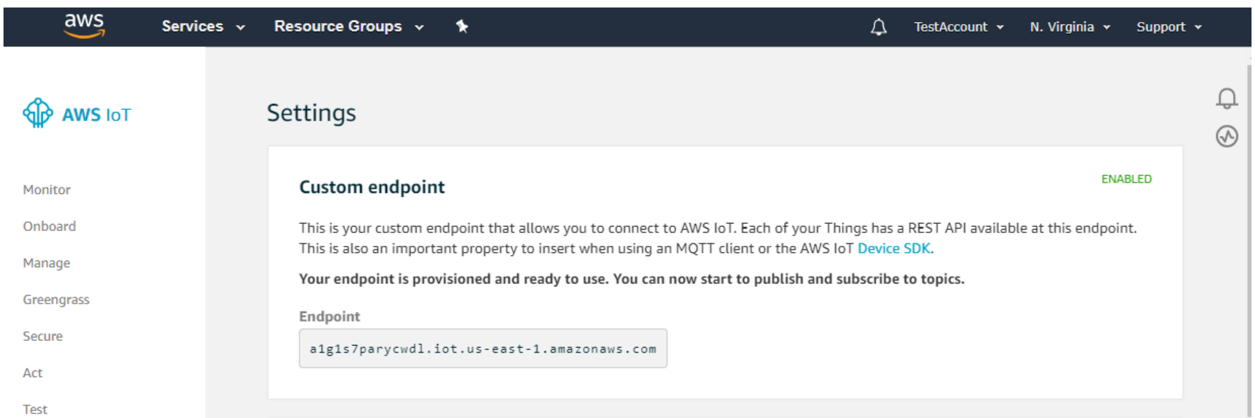
Click Download to download the certificate file attached to your AWS IoT thing:



root-CA.crt – Name of the Verisign root CA server certificate used to authenticate the AWS IoT server. By default, 'rootCA.crt'. You can download it from https://www.symantec.com/content/en/us/enterprise/verisign/roots/VeriSign-Class%203-Public-Primary-Certification-Authority-G5.pem and save it to a text file called root-CA.crt.

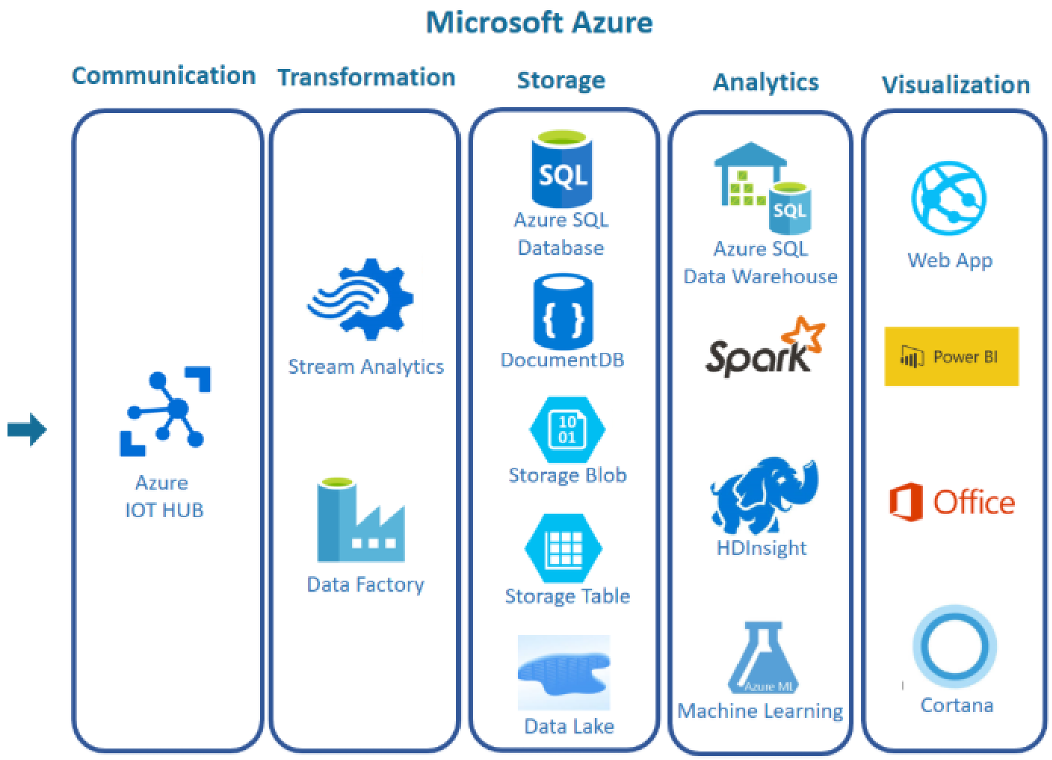


AWSCustomEndpoint – Your AWS IoT endpoint.



Azure IoT Hub (Internet of Things) API Support

The Incentive API contains a cloud interface to Azure's IoT Hub using the Amqp protocol and SAS tokens for security. Microsoft best describes the "IoT Hub is a managed service, hosted in the cloud, that acts as a central message hub for bi-directional communication between your IoT application and the devices it manages. You can use Azure IoT Hub to build IoT solutions with reliable and secure communications between millions of IoT devices and a cloud-hosted solution backend." The complexity of this communications is hidden from the user and allows device data to be easily sent to the cloud for storage, monitoring by IoT subscribers, peer to peer communications to other Incentive devices, as well as database storage.



The creation of an Azure IoT Hub and devices is left to the user and beyond the scope of this document. It will be necessary to have one or more devices available to send data to. Some documents that are helpful are available at:

https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-create-through-portal

In addition to creating a 'Thing' you will need to create an SAS Token for access to your Hub. This is detailed later in this document. Screen shots of the IoT Device created for the C# demo included with the Incentive distribution are included within the appendix.

Amqp_IOT Class

The API contains support for the Internet of Things (IoT) AMQP communication protocol to enable a simple interface to the cloud for storage. Amazon AWS IoT and Microsoft Auzre IoT Hub are presently supported, with the Amqp_IOT Class typically used for Azure. The Amqp_IOT class puts a wrapper around the open source Amqp client software from the Amqp .Net Lite: http://azure.github.io/amqpnetlite/. A very limited ability is exposed to simplify the interface and the DLL (Amqp.Net.dll) is included with our API distribution. If for some reason the IncentiveAPI is missing some feature you require you can always add the DLL as a resource in your project and work with it directly. The creation of an Azure account and IoT device is beyond the scope of this document and available online from Microsoft: https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-create-through-portal .

To simplify the interface a generic connectIOT, publishIOT, subscribeIOT, unsubscribeIOT, and disconnect IOTmethods have been added to the class. Note that methods may throw an AmqpConnectionException or AmqpClientException.

Methods

	Name	Description
	connectIOT	Connect to AMQP (Azure) IOT Hubclient using TLS1.2. Note that connection will be disconnected by the Azure IOT server if > than 10 minutes of inactivity (default may change and is controlled by Microsoft). Port 5671 is typically used.
	disconnectIOT	Disconnect the client IOT session. Will automatically unsubscribe and unpublish all links as well, first, prior to ending the session.
	Equals(System.Object)	Determines whether the specified Object is equal to the current Object. (Inherited from Object.)
	Finalize	Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)
	GetHashCode	Serves as a hash function for a particular type. (Inherited from Object.)
	GetType	Gets the Type of the current instance. (Inherited from Object.)
	MemberwiseClone	Creates a shallow copy of the current Object. (Inherited from Object.)
	publishIOT(Int32, String, String)	Publish data to IOT
	publishIOT(Int32, String, Byte[])	Publish data to IOT
	subscribeIOT	Subscribe to receive data from IOT device endpoint.
	ToString	Returns a string that represents the current object. (Inherited from Object.)
	unpublishIOT	Unpublish from sending data on a specified IOT link.
	unsubscribeIOT	Unsubscribe from receiving data on an open endpoint link.

Events

	Name	Description
	Callback	MessageReceivedCallBack event. This event must be set by the application so that any topics subscribed to will have the callback routine invoked upon reception of a message.

Amqp_IOT.connectIOT Method

Connect to AMQP (Azure) Hub IOT client using TLS1.2. Note that connection will be disconnected by the Azure IOT server if > than 10 minutes of inactivity (default may change and is controlled by Microsoft). Port 5671 is typically used.

Namespace: CTC_Incentive
Assembly: CTC_Incentive (in CTC_Incentive.dll)

Syntax

C# VB C++

```
public bool connectIOT(
        string host,
        int port,
        string username_hublevel,
        string password_hublevel,
        string device_id
)
```

Parameters

host
Type: String
Host name to connect to, used to find IP address. For example: [Hub Name].azure-devices.net, where [Hub name] is the name of your IOT hub

port
Type: Int32
What port to attempt connection on, 5671 is normally used.

username_hublevel
Type: String
Hub user name for example iothubowner@sas.root.[Hub Name]. Where [Hub Name] is name of your IOT hub.

password_hublevel
Type: String
SAS Hub Token from Device Explorer - Configuration - SharedAccessSignature - Generate SAS.

device_id
Type: String
Name of your device on the Azure IOT Hub.

Return Value

`true` if connected, `false` otherwise. AmqpConnectionException on connect error.

Amqp_IOT.disconnectIOT Method

Disconnect the client IOT session. Will automatically unsubscribe and unpublish all links as well, first, prior to ending the session.

Namespace: CTC_Incentive
Assembly: CTC_Incentive (in CTC_Incentive.dll)

Syntax

C# VB C++

```
public void disconnectIOT()
```

Amqp_IOT.publishIOT Method (Int32, String, Byte[])

Publish data to IOT

Namespace: CTC_Incentive
Assembly: CTC_Incentive (in CTC_Incentive.dll)

Syntax

C# | VB | C++

```
public int publishIOT(
        int handle,
        string sendEndpoint,
        byte[] iot_data
)
```

Parameters

handle
Type: Int32
Handle to use for sending, -1 if a new one, else actual used before.

sendEndpoint
Type: String
Endpoint name to use for the device. If simple message it is normally: "/devices/[DEVICE_ID]/messages/events", where [DEVICE_ID] is the name of the device.

iot_data
Type: Byte[]
byte array data to send to the IOT Device, sendEndpoint.

Return Value

Returns int 'handle' use for send link, this handle must be used for future calls to same endpoint prior to a unpublishIOT () call which closes the send link. -1 if failed. AmqpClientException is also possible.

Amqp_IOT.unpublishIOT Method

Unpublish from sending data on a specified IOT link.

Namespace: CTC_Incentive
Assembly: CTC_Incentive (in CTC_Incentive.dll)

Syntax

C# | VB | C++

```
public void unpublishIOT(
        int handle
)
```

Amqp_IOT.subscribeIOT Method

Subscribe to receive data from IOT device endpoint.

Namespace: CTC_Incentive
Assembly: CTC_Incentive (in CTC_Incentive.dll)

Syntax

C# | VB | C++

```
public int subscribeIOT(
	string receiveEndpoint
)
```

Parameters

receiveEndpoint
Type: String
Endpoint string to monitor for data. For a message this is generally: "/devices/[DEVICE_ID]/messages/deviceBound", where [DEVICE_ID] is the name of the device.

Return Value

int 'handle' to be used by any thread monitoring data received on the link established for that endpoint. -1 if failed. AmqpClientException is also possible.

Amqp_IOT.unsubscribeIOT Method

Unsubscribe from receiving data on an open endpoint link.

Namespace: CTC_Incentive
Assembly: CTC_Incentive (in CTC_Incentive.dll)

Syntax

C# | VB | C++

```
public void unsubscribeIOT(
	int handle
)
```

Sample Amqp_IOT Class usage

```
/// <summary>
/// IOT_Test class for testing API access to Azure IOT.
/// </summary>
class IOT_Test
{
   // This is the Amazon Topic or Azure Device to which messages are published (Send) or subscribed to (Receive).
   public string IOT_Topic_Device = "IncentivePLC_Data";
   // Queue to store received message to for testing
   public Queue subscribed_Data = new Queue();
```

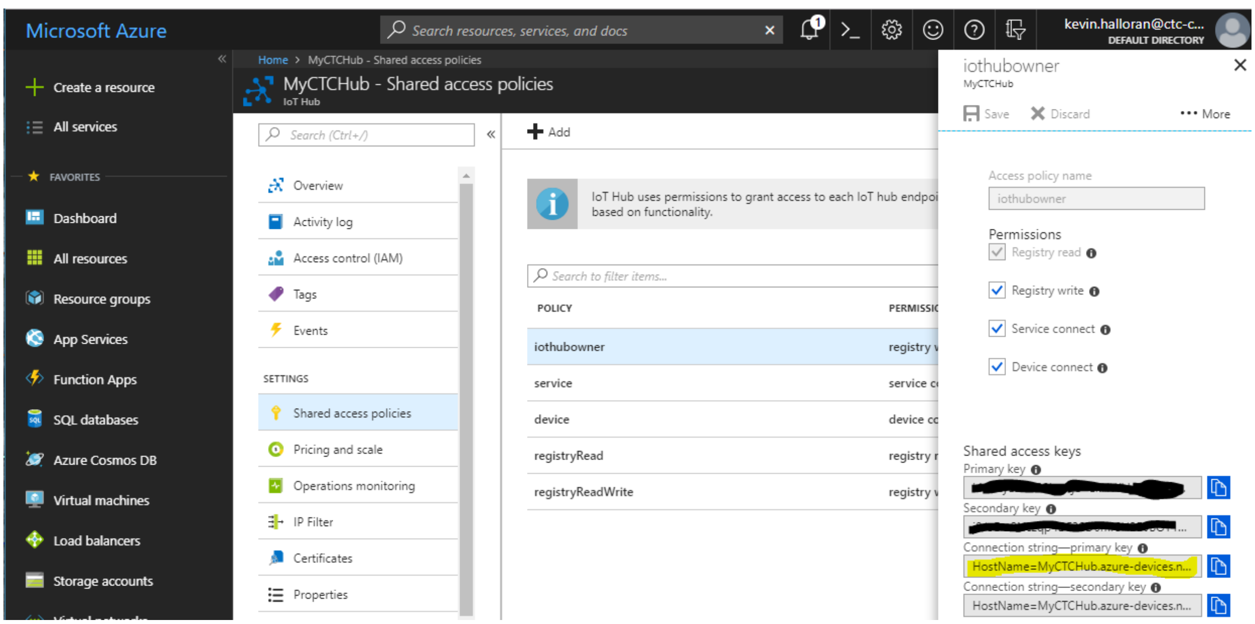
```
// Generic callback for subscriber messages received.  Each message is placed in the queue for later use.  'handle' is used by Azure but not
//Amazon AWS interface.
// With Azure there is a receive thread for each subscription, with Amazon all subscriptions use the same thread.  'topic' is the device
// subscribed to whereas 'data' is what is received.
private void subscribedMessageCallback(int handle, string topic, byte[] data)
{
   string ReceivedMessage = topic + ":  " + Encoding.UTF8.GetString(data);  // This assumes the message is a string, modify as desired.
   subscribed_Data.Enqueue(ReceivedMessage);
}

// ***************** Azure IOT Hub Access (Modify for your account setup) *************************
public string HOST = "MyCTCHub.azure-devices.net";    // Azure Hub Hostname
public int PORT = 5671;                               // TLS 1.2 port to use

// Hub username iothubowner@sas.root.[Hub Name]
public string username_hublevel = "iothubowner@sas.root.MyCTCHub";
// SAS Token from Microsoft Device Explorer - Configuration - SharedAccessSignature - Generate SAS
public string password_hublevel = "SharedAccessSignature sr=MyCTCHub.azure-devices.net&sig=X89kjkl;hkj45xxfh%68403mf
r%3D&se=1534575847&skn=iothubowner";
public void testAzure_IOT()
{
   // Request and IOT communications interface from the Incentive API for Amqp protocol which is used by Microsoft Azure.
   Amqp_IOT myAmqp_iot = new Amqp_IOT();

   // Establish a connection to the Azure device
   if (!myAmqp_iot.connectIOT(HOST, PORT, username_hublevel, password_hublevel, IOT_Topic_Device))
   {
      return;  // Connection failed
   }

   // Add the callback function so can queue messages received otherwise they are ignored.
   myAmqp_iot.Callback += new Amqp_IOT.MessageReceivedCallBack(subscribedMessageCallback);

   // Subscribe to an Endpoint, in this case all messages sent to messages/event by other connections will be received
   string rx_endpoint = "/devices/" + IOT_Topic_Device + "/messages/deviceBound";
   int rxHandle = myAmqp_iot.subscribeIOT(rx_endpoint);
   if (rxHandle < 0)
   {
      // Session is closed, if 10 minutes idle Azure will disconnect fo may want to reconnect.
      return;
   }
   int txHandle = -1;  // request a handle the first time publish a message

   // Send a bunch of messages, suggest using Device Explorer to send a couple as well so can see receive thread working.
   int key = 1;
   string timeStamp;
   string iot_data;
   float temperature = 200.0F;
   for (int i = 0; i != 30; i++)
   {
      // Always useful to have a timestamp on an IOT entry
      timeStamp = DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss.fff");
      // This can be written as desired but for this example it is a json record for writing to a DynamoDBv2 database
      // In this particular example the DynamoDB primary key field name is 'key' and all the other fields are created dynamically by the
      // database.  Only the primary key and an optional sort key are always required and defined ahead of time
      // Make up some data where we keep the key unique by incrementing each time and just bump the temp data so can see some changes in
      //  the database later when we look at it.  As a side note if the key does not change the record will be overwritten for that key.
      iot_data = "{ \"timestamp\" : \"" + timeStamp + "\", \"key\" : \"" + key + "\", \"deviceid\" : \"Monitor_01\", \"temp\" : " + temperature + ",
\"humidity\" : 47.43, \"Co2_level\" : 644, \"lighting_level\" : 85, \"pressure\" : 28, \"light_state\" : 1, \"C0 Alarm\" : 0 }";
      // Each unique endpoint will need its own handle, thus if first time for this endpoint set the handle to -1, otherwise you can use the handle
      //  returned by the prior call to publish.
      string tx_endpoint = "/devices/" + IOT_Topic_Device + "/messages/events";
      txHandle = myAmqp_iot.publishIOT(txHandle, tx_endpoint, iot_data);
      if (txHandle < 0)
      {
```

```
                // session closed?
                return;
            }
            MessageBox.Show("Published " + key + ".");
            key++;  // bump the key
            temperature = temperature + 25.2F;  // Arbitrarily bump the temperature data so see changes in the database
        }
        // Cleanup and disconnect.
        myAmqp_iot.disconnectIOT();
    }
}
```

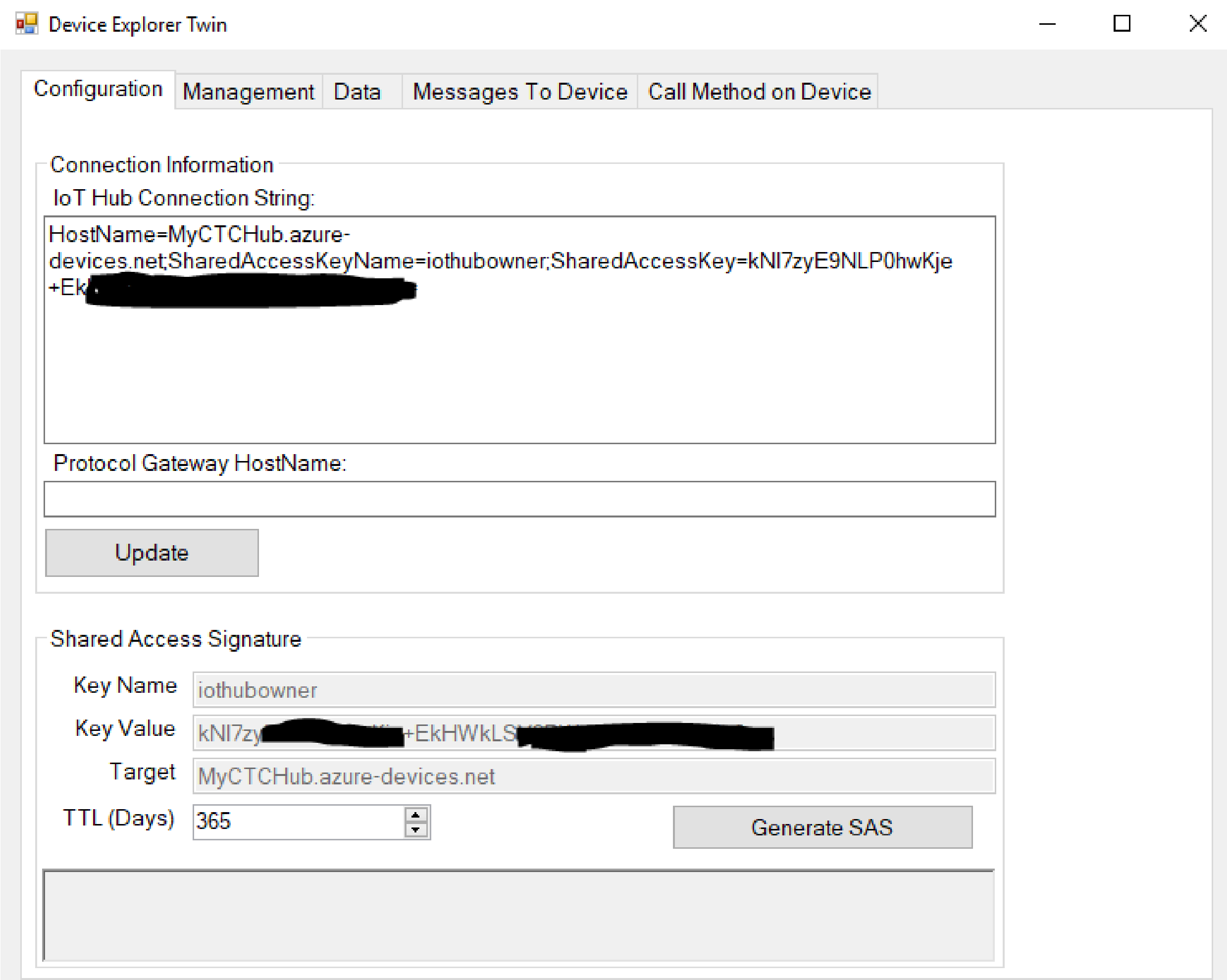
Amqp_IOT SAS Token Generation – Device Explorer

A program utility provided by Microsoft called 'SetupDeviceExplorer.msi' is included in you '..\Control_Technology_Corporation\iot\amqp' directory. This program must be installed to generate a SAS Token used in the previous section for the variable 'password_hublevel'.

Once installed you must first enter the Iot Hub Connection String for your Hub. It is located under 'Shared access policies' for the 'iothubowner'. Reference the 'IoT Hub Connection String' highlighted in yellow below:



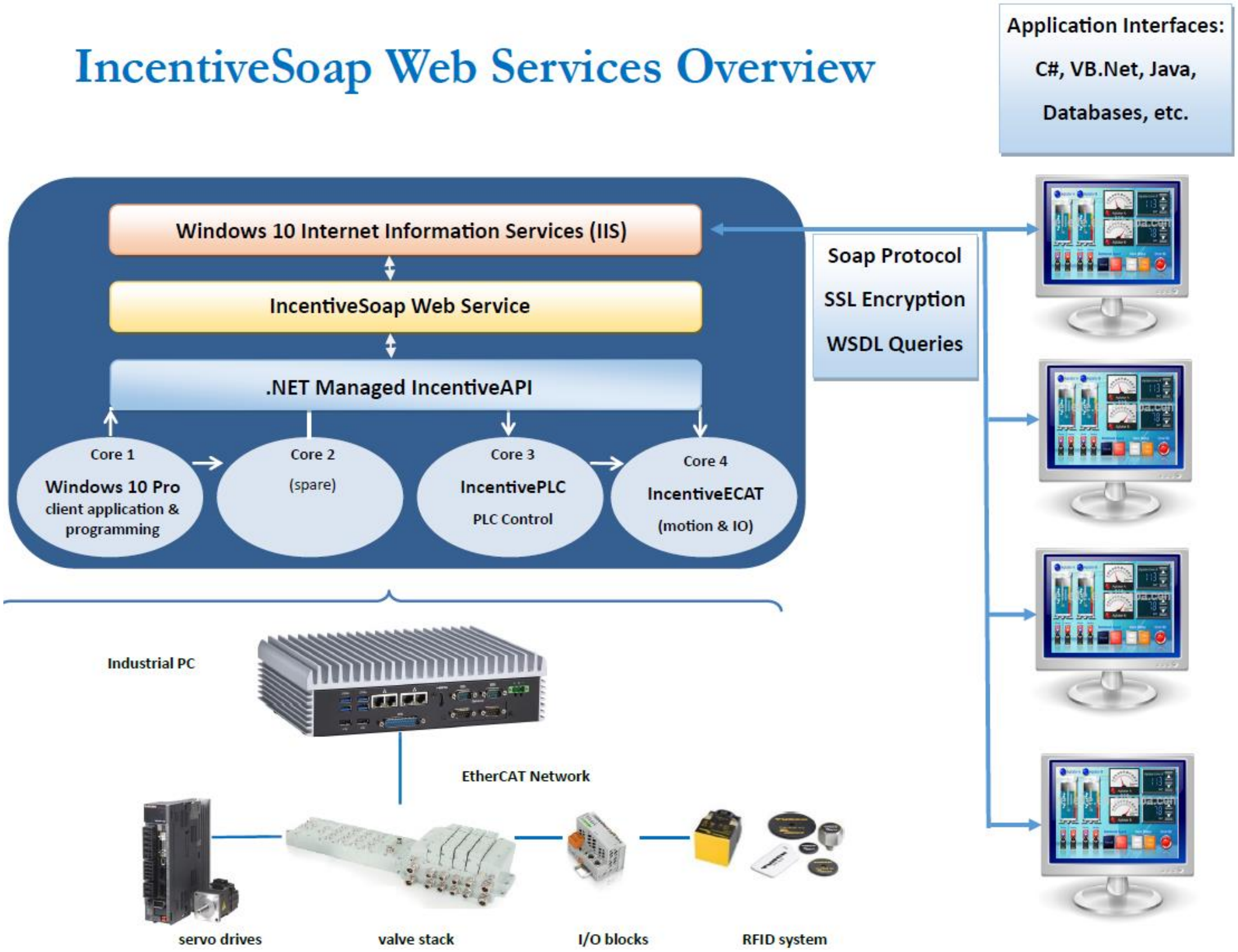
This string is then added to the Connection Information of the Device Explorer and Update is selected. Upon doing so a connection should be established and the iothubowner appear similar to below.



Click 'Generate SAS' in order to create a token for your application program, 'password_hublevel' in the preceeding example program. Additional information on the 'Device Explorer' is available from Microsoft at: https://github.com/Azure/azure-iot-sdk-csharp/tree/master/tools/DeviceExplorer.

IncentiveSoap - IIS Soap Web Service

IncentiveSoap is a web service that can provide secure access to data within the Incentive environment. Soap can be used to remotely read/write Incentive variables and motion control information for presentation on a web page or storage to a database. The available Soap operations can be queried by reading the services Web Service Description Language (wsdl) file, IncentiveSoapWebService.svc?wsdl. The "..\Control_Technology_Corporation_\WebServices\IncentiveSoapWebDeploy" directory contains the soap web service interface for the IncentiveAPI. This service can run locally as an Internet Information Service (IIS) Windows Web site using asp.net. The figure below gives an overview of how the Web Service may be used:



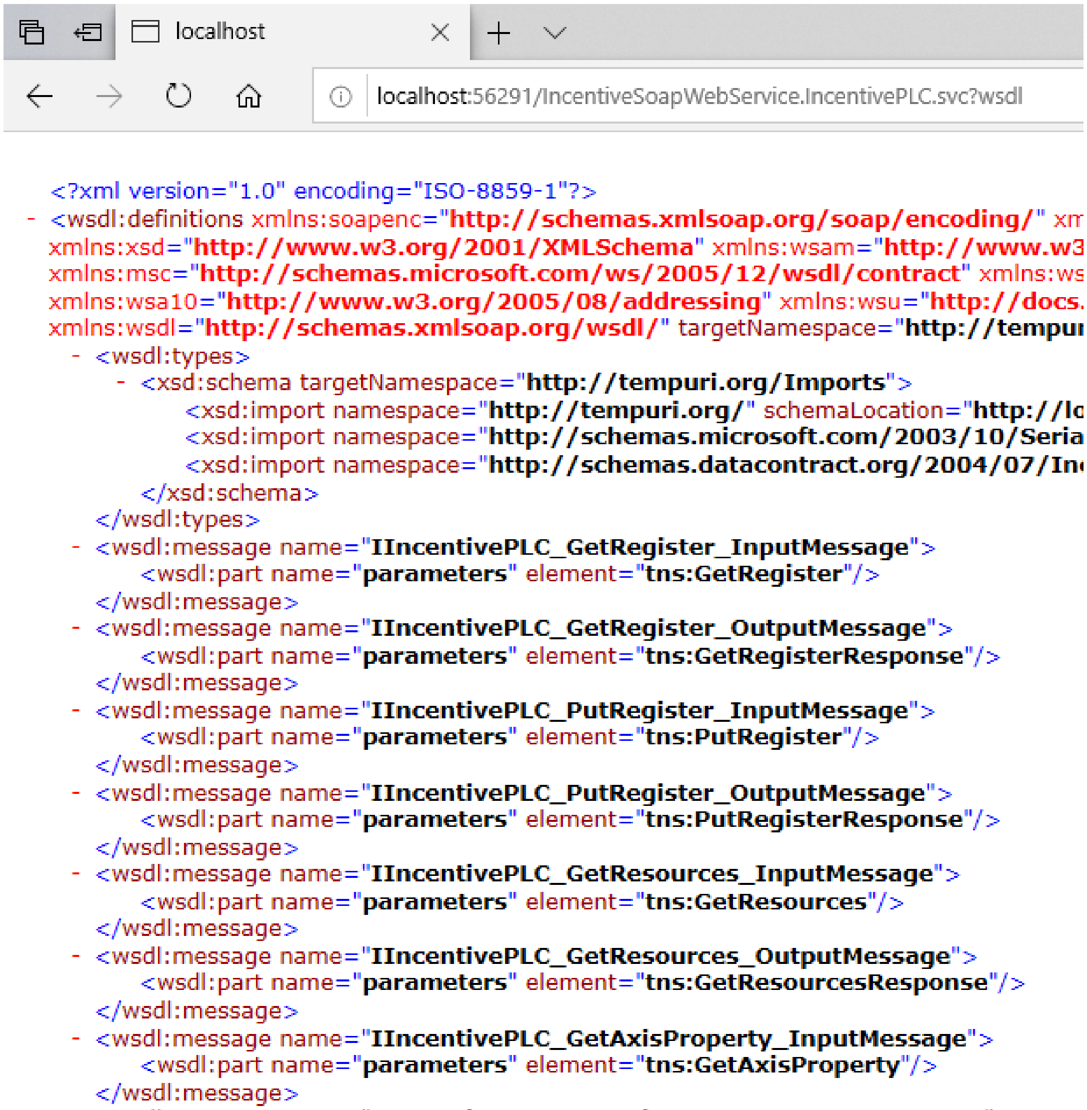
A utility for testing the soap interface is the SmartBear ReadyAPI SoapUI Pro. It is a great product and available with a 14 day free trial within which you can usually complete most of your testing.

https://smartbear.com/product/ready-api/soapui/overview/

The setup of the SoapUI testing is beyond the scope of this document but is fairly automatic as it will query the WSDL file and automatically generate the proper test cases for you where you can perform manual queries. The port the service will listen on is dependent on that to which you assigned to it in IIS. To read the WSDL definition file the link, assuming you are local and using port 56291, would be:

http://localhost:56291/IncentiveSoapWebService.IncentivePLC.svc?wsdl

As a test you can access this from your own web browser and the xml format will be returned to you:




```
<?xml version="1.0" encoding="ISO-8859-1"?>
- <wsdl:definitions xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xm
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsam="http://www.w3
  xmlns:msc="http://schemas.microsoft.com/ws/2005/12/wsdl/contract" xmlns:ws
  xmlns:wsa10="http://www.w3.org/2005/08/addressing" xmlns:wsu="http://docs.
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://tempu
  - <wsdl:types>
    - <xsd:schema targetNamespace="http://tempuri.org/Imports">
        <xsd:import namespace="http://tempuri.org/" schemaLocation="http://lc
        <xsd:import namespace="http://schemas.microsoft.com/2003/10/Seria
        <xsd:import namespace="http://schemas.datacontract.org/2004/07/In
      </xsd:schema>
    </wsdl:types>
  - <wsdl:message name="IIncentivePLC_GetRegister_InputMessage">
      <wsdl:part name="parameters" element="tns:GetRegister"/>
    </wsdl:message>
  - <wsdl:message name="IIncentivePLC_GetRegister_OutputMessage">
      <wsdl:part name="parameters" element="tns:GetRegisterResponse"/>
    </wsdl:message>
  - <wsdl:message name="IIncentivePLC_PutRegister_InputMessage">
      <wsdl:part name="parameters" element="tns:PutRegister"/>
    </wsdl:message>
  - <wsdl:message name="IIncentivePLC_PutRegister_OutputMessage">
      <wsdl:part name="parameters" element="tns:PutRegisterResponse"/>
    </wsdl:message>
  - <wsdl:message name="IIncentivePLC_GetResources_InputMessage">
      <wsdl:part name="parameters" element="tns:GetResources"/>
    </wsdl:message>
  - <wsdl:message name="IIncentivePLC_GetResources_OutputMessage">
      <wsdl:part name="parameters" element="tns:GetResourcesResponse"/>
    </wsdl:message>
  - <wsdl:message name="IIncentivePLC_GetAxisProperty_InputMessage">
      <wsdl:part name="parameters" element="tns:GetAxisProperty"/>
    </wsdl:message>
```


The WSDL description link will be required by the SoapUI project, upon which it will interrogate the service and generate the available operations. The link for the operation request would be something like:

http://localhost:56291/IncentiveSoapWebService.IncentivePLC.svc/soap

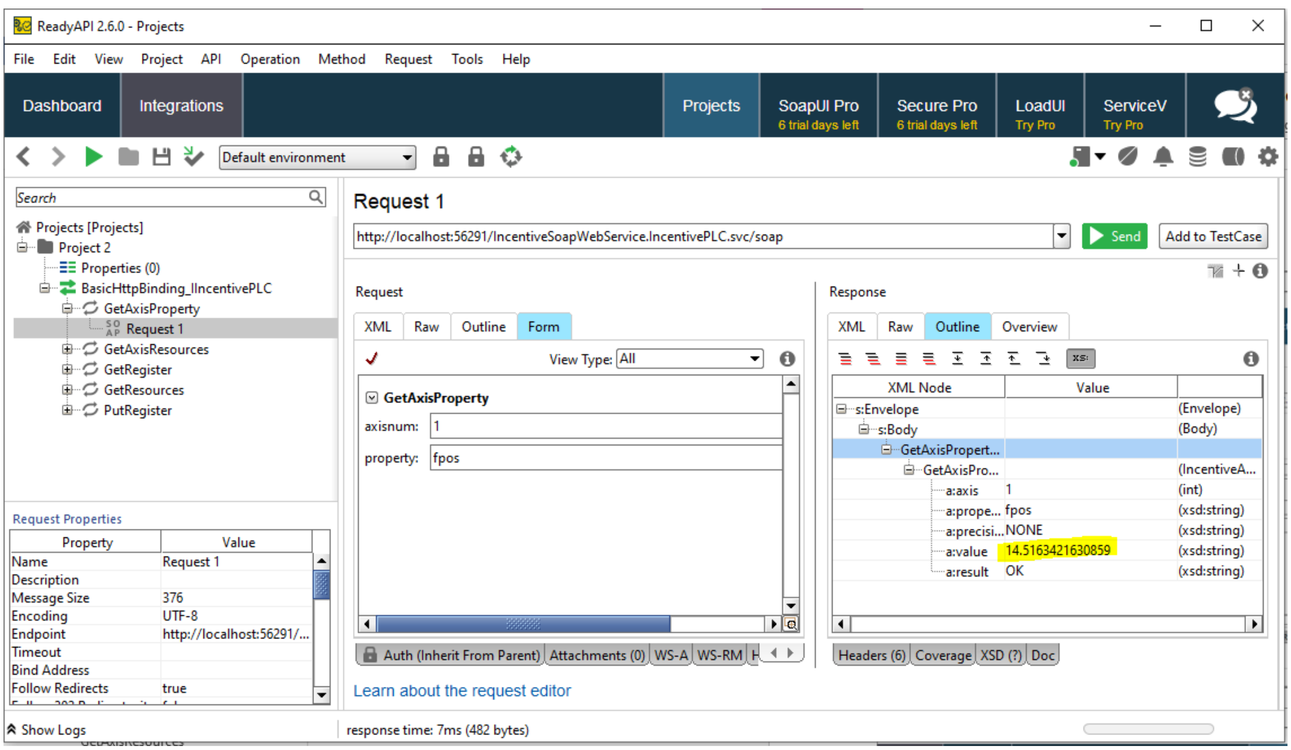
The operations mimic that of the IncentiveAPI. The available operations are:

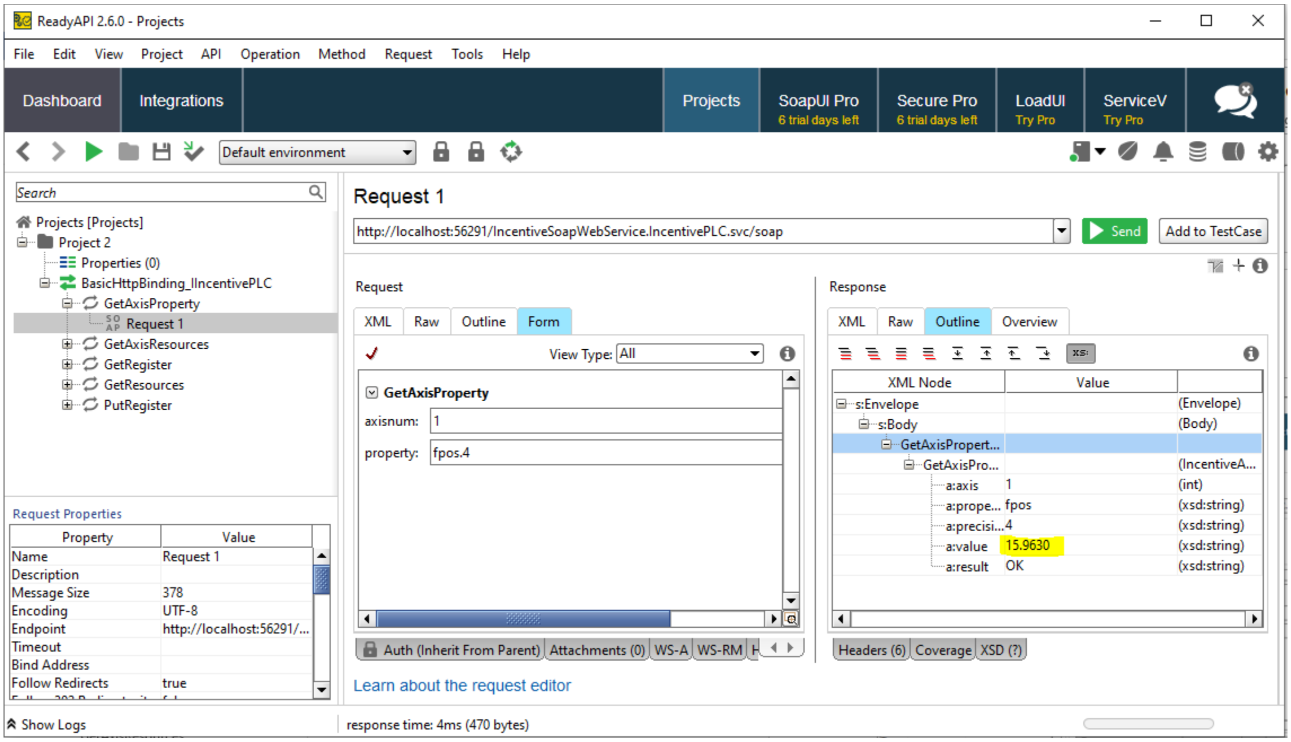
GetAxisProperty

GetAxisProperty can be used to read the properties of an axis. The property is specified as a string, such as fpos for feedback postion. Since these values are typically doubles you can limit the number of characters after the decimal point by using a property.precision value. For example fpos would return the entire double value while fpos.4 would return only 4 locations after the decimal point.

Available properties are as follows:

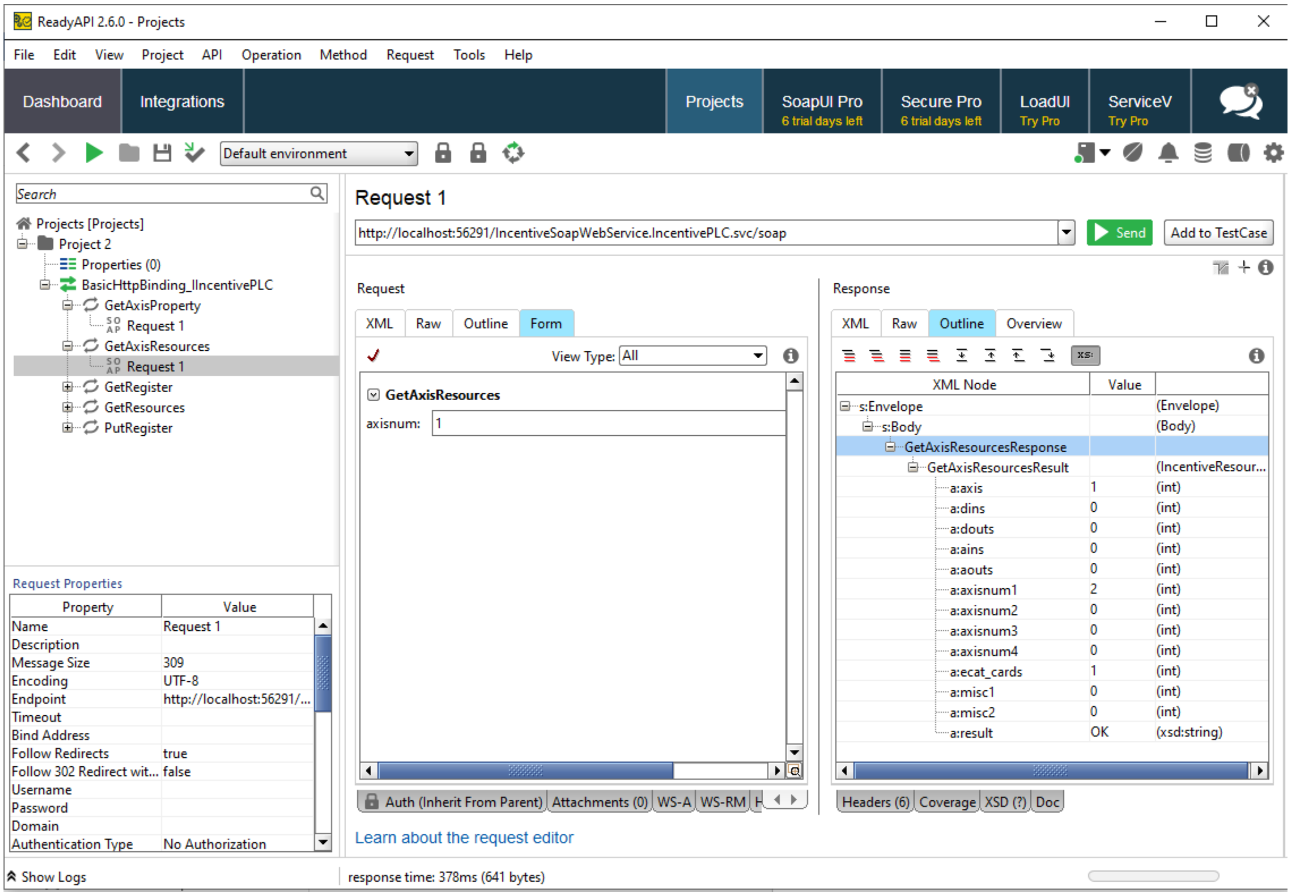
Property Names		
fpos	enabled	tlim
tpos	global_inputs	tmax
mpos	global_outputs	uud
mppr	gratio	uun
ppr	inpo	units_ratio
tposc	pstate	vmax
mposc	perr	vmin
driveType	vel	faulted
dwSlaveID		




ReadyAPI 2.6.0 - Projects
File Edit View Project API Operation Method Request Tools Help
Dashboard
Integrations
Projects
SoapUI Pro
6 trial days left
Secure Pro
6 trial days left
LoadUI
Try Pro
ServiceV
Try Pro
Default environment
Projects [Projects]
Project 2
Properties (0)
BasicHttpBinding_IIncentivePLC
GetAxisProperty
Request 1
GetAxisResources
GetRegister
GetResources
PutRegister
Request Properties
Property Value
Name Request 1
Description
Message Size 378
Encoding UTF-8
Endpoint http://localhost:56291/...
Timeout
Bind Address
Follow Redirects true
Request 1
http://localhost:56291/IncentiveSoapWebService.IncentivePLC.svc/soap
Send
Add to TestCase
Request
XML Raw Outline Form
View Type: All
GetAxisProperty
axisnum: 1
property: fpos.4
Auth (Inherit From Parent) Attachments (0) WS-A WS-RM
Response
XML Raw Outline Overview
XML Node Value
s:Envelope (Envelope)
s:Body (Body)
GetAxisPropert...
GetAxisPro... (IncentiveA...
a:axis 1 (int)
a:prope... fpos (xsd:string)
a:precisi... 4 (xsd:string)
a:value 15.9630 (xsd:string)
a:result OK (xsd:string)
Headers (6) Coverage XSD (?)
Learn about the request editor
Show Logs
response time: 4ms (470 bytes)

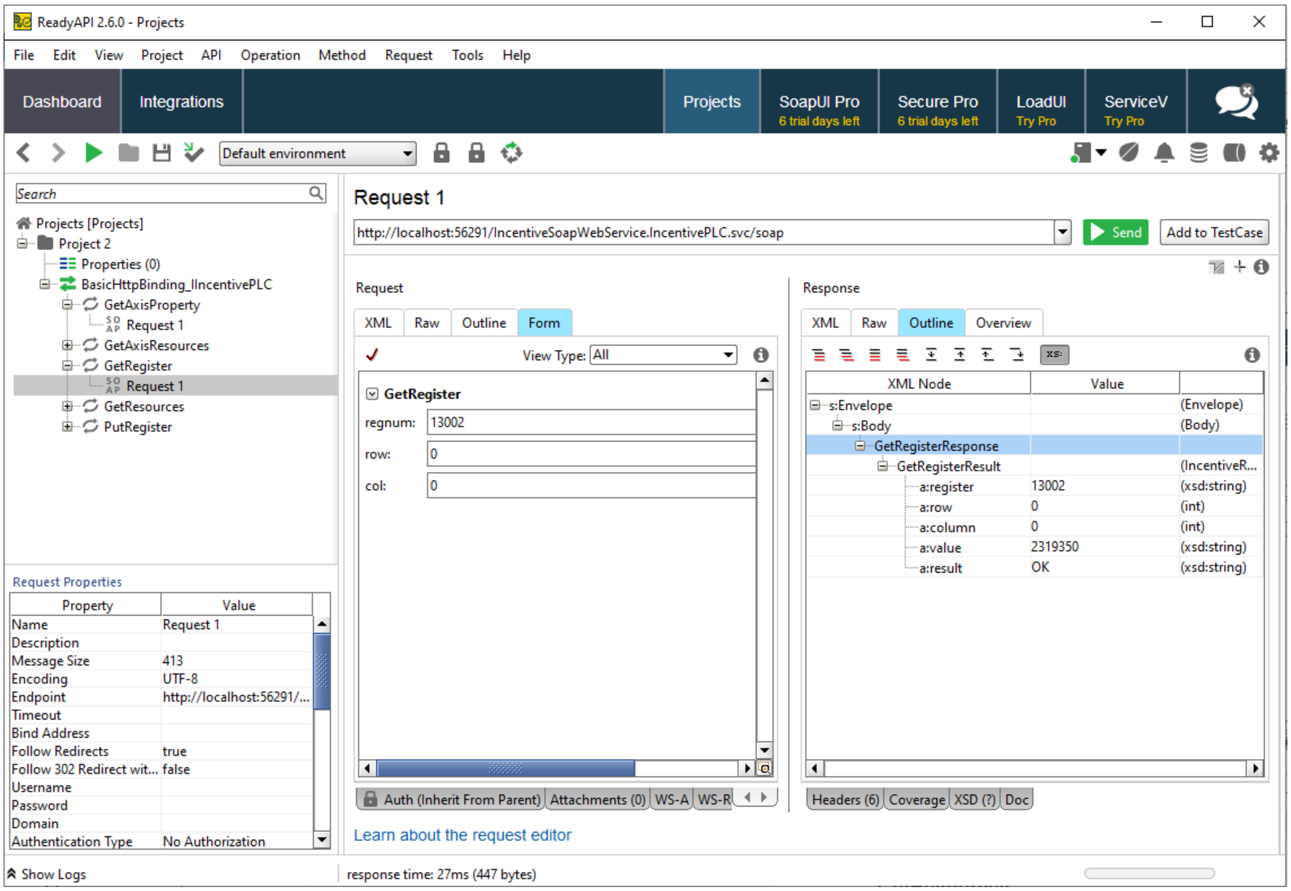
GetAxisResources

GetAxisResources is similar to GetResources except it is for a specific EtherCAT master and not system wide.



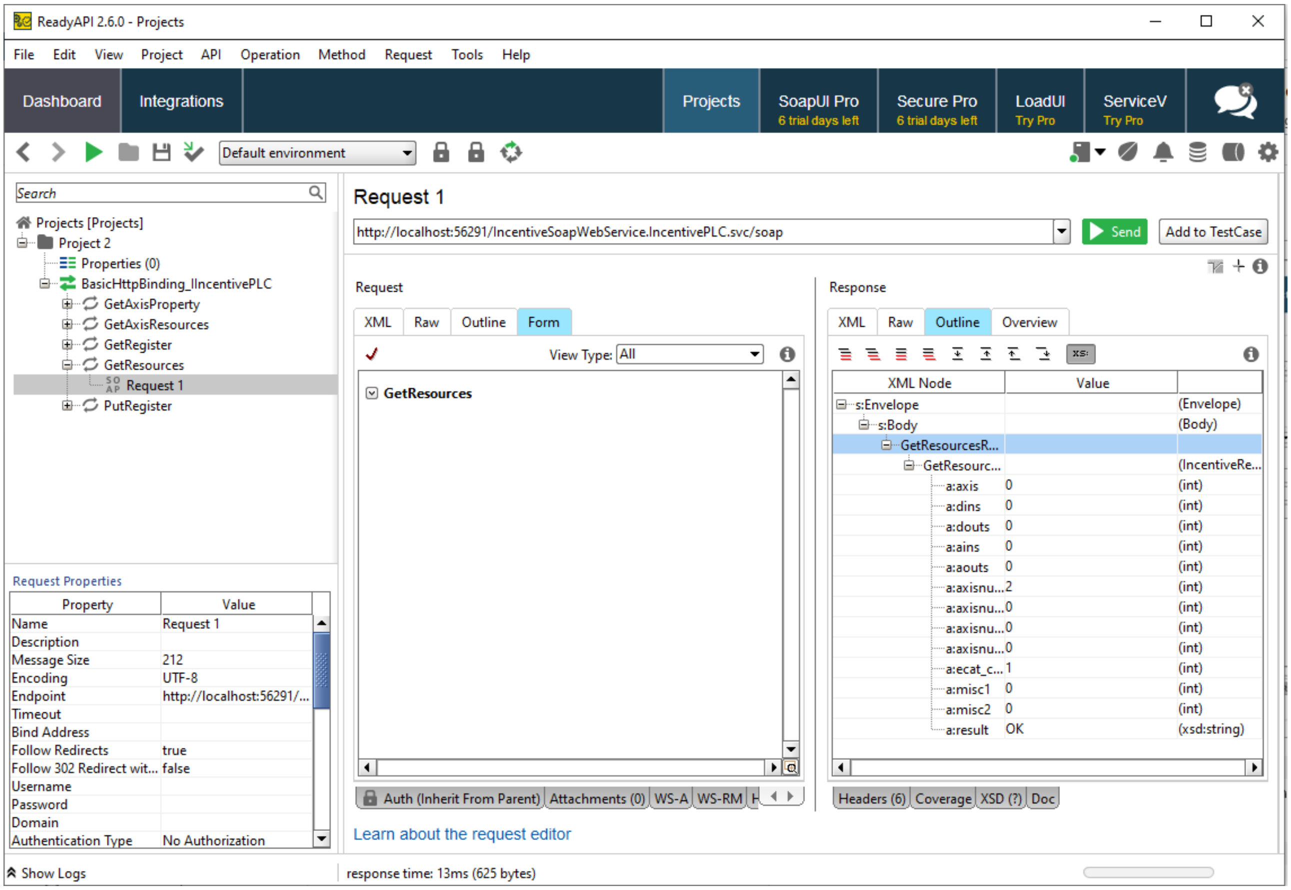
GetRegister

GetRegister is the same as the Incentive API where a register number is specified as well as a row and column should it be a variant. The String value is returned. Note that a symbol defined in the program can be used as the register reference as well.



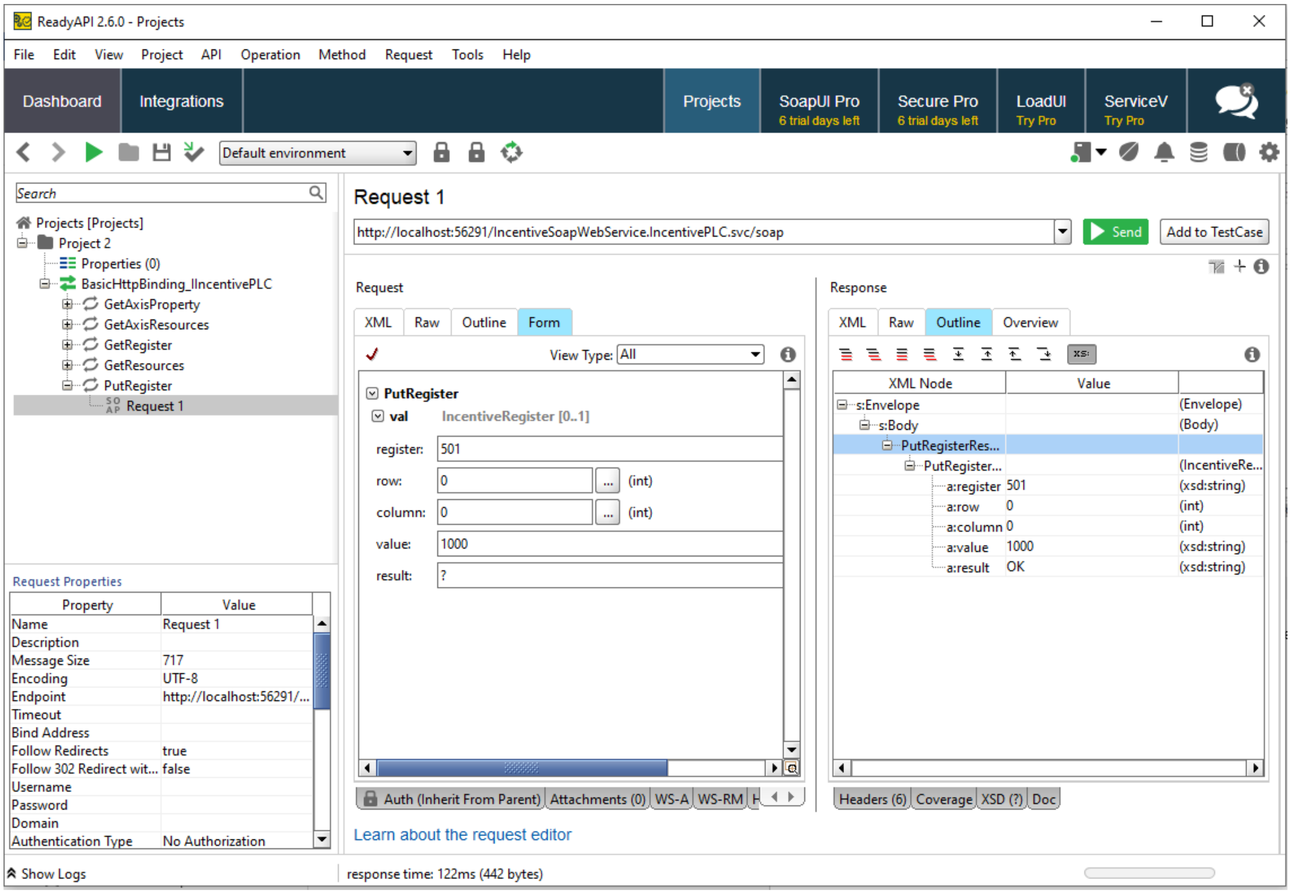
GetResources

The GetResources operation returns all the resources system wide available to Incentive PLC.



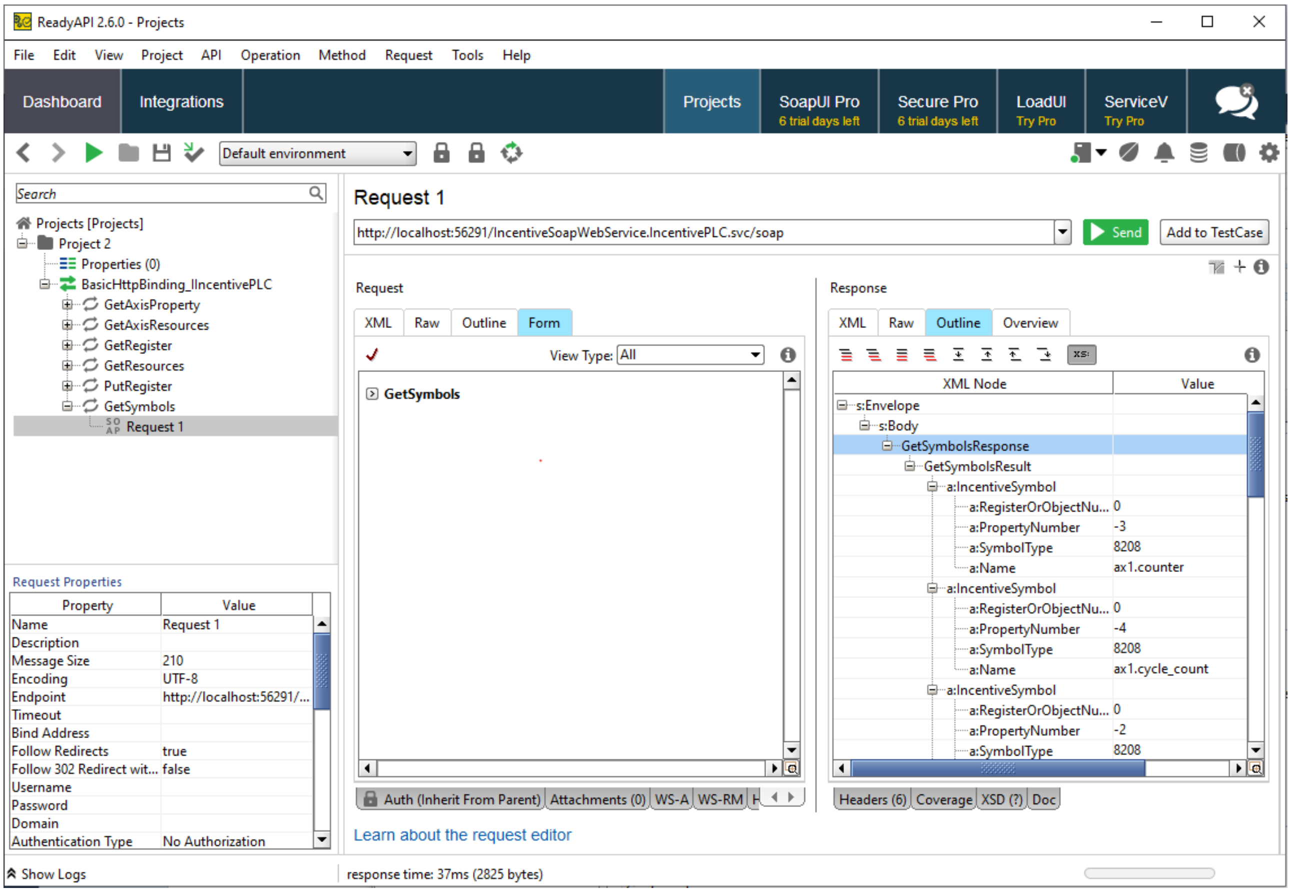
PutRegister

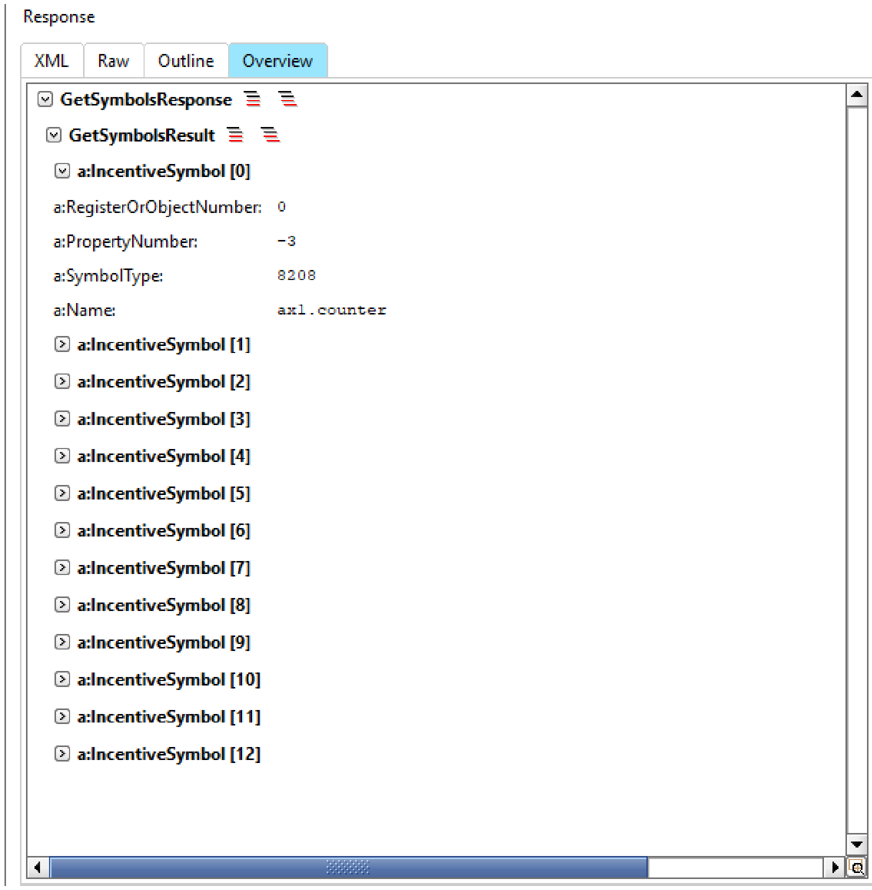
The PutRegister operation is used to write a value to a register, the example below is writing 1000 to register 501. Note that a symbol defined in the program can be used as the register reference as well.



GetSymbols

The GetSymbols operation returns a list of all the symbols as an array. The properties are the same as that defined by the IncentiveAPI. If there is only one symbol returned and the SymbolType is set to -1 then an error occurred and the Name field details the error that occurred.

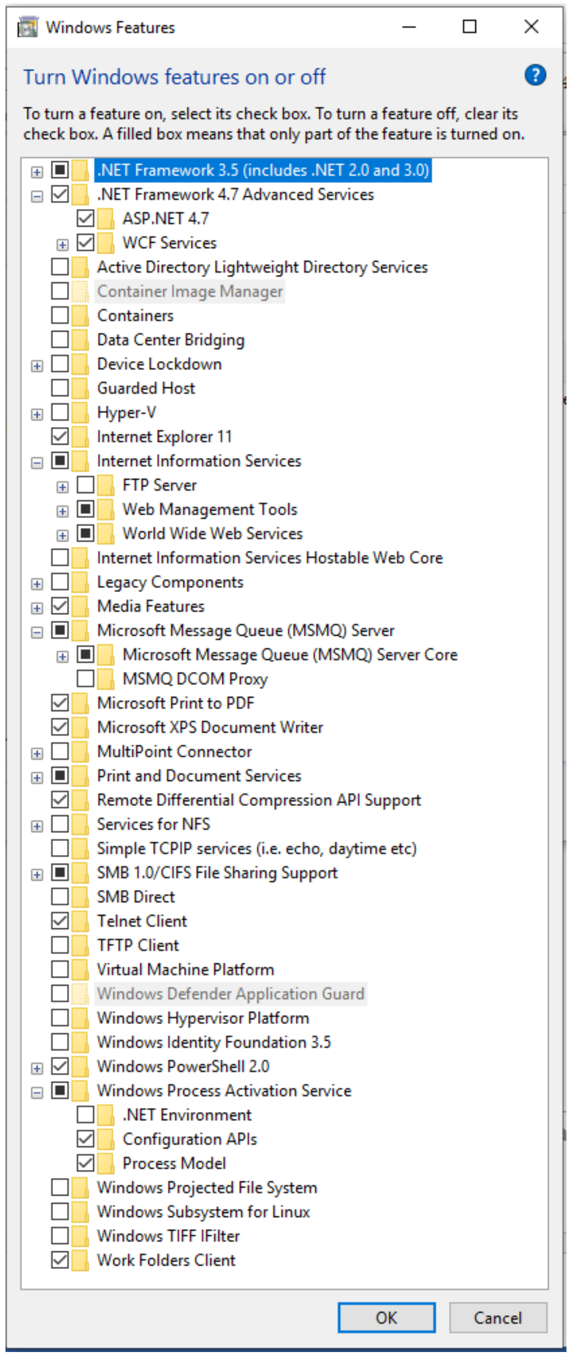



Response
XML
Raw
Outline
Overview
GetSymbolsResponse
GetSymbolsResult
a:IncentiveSymbol [0]
a:RegisterOrObjectNumber: 0
a:PropertyNumber: -3
a:SymbolType: 8208
a:Name: ax1.counter
a:IncentiveSymbol [1]
a:IncentiveSymbol [2]
a:IncentiveSymbol [3]
a:IncentiveSymbol [4]
a:IncentiveSymbol [5]
a:IncentiveSymbol [6]
a:IncentiveSymbol [7]
a:IncentiveSymbol [8]
a:IncentiveSymbol [9]
a:IncentiveSymbol [10]
a:IncentiveSymbol [11]
a:IncentiveSymbol [12]

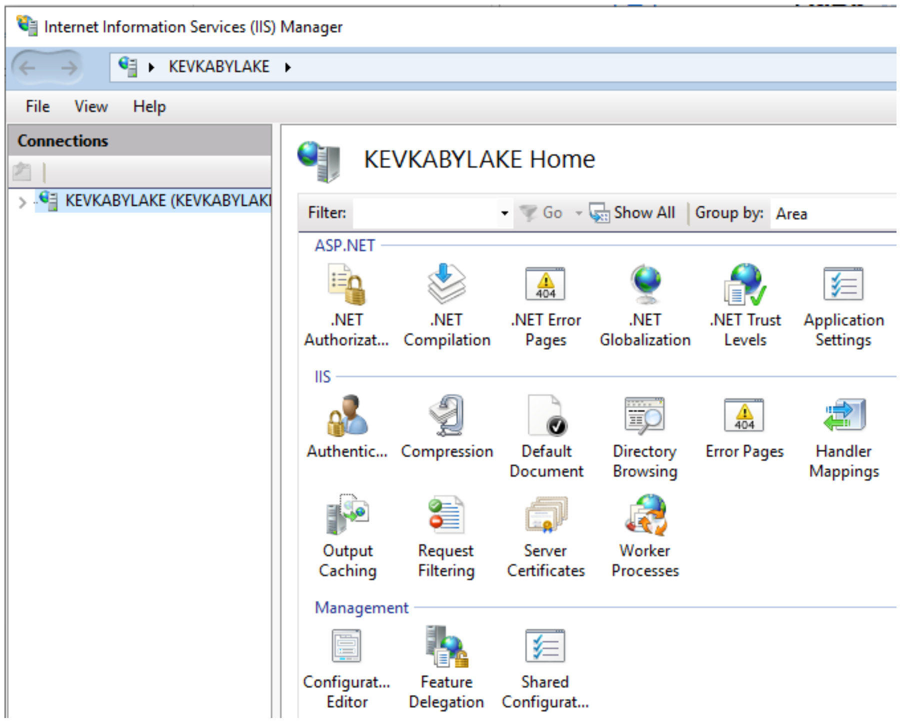
IIS – Web Service Configuration

IIS can be added to Windows 10 Pro by accessing the Programs and Features settings and then selecting "Turn Windows Features on or off".

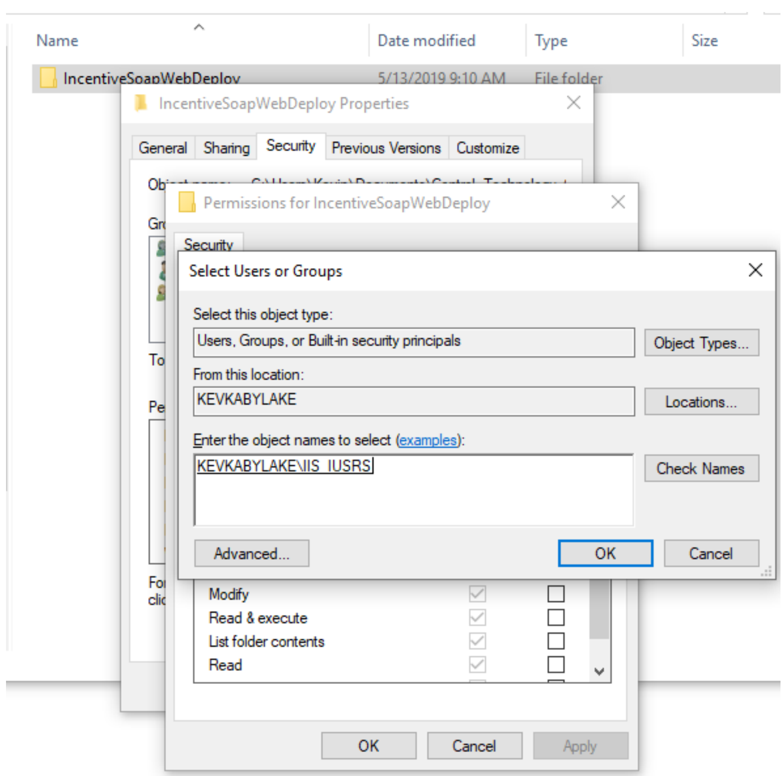
It should be set something like the following:



Add the features as needed and when done open the IIS Manager under Windows Administrative Tools.



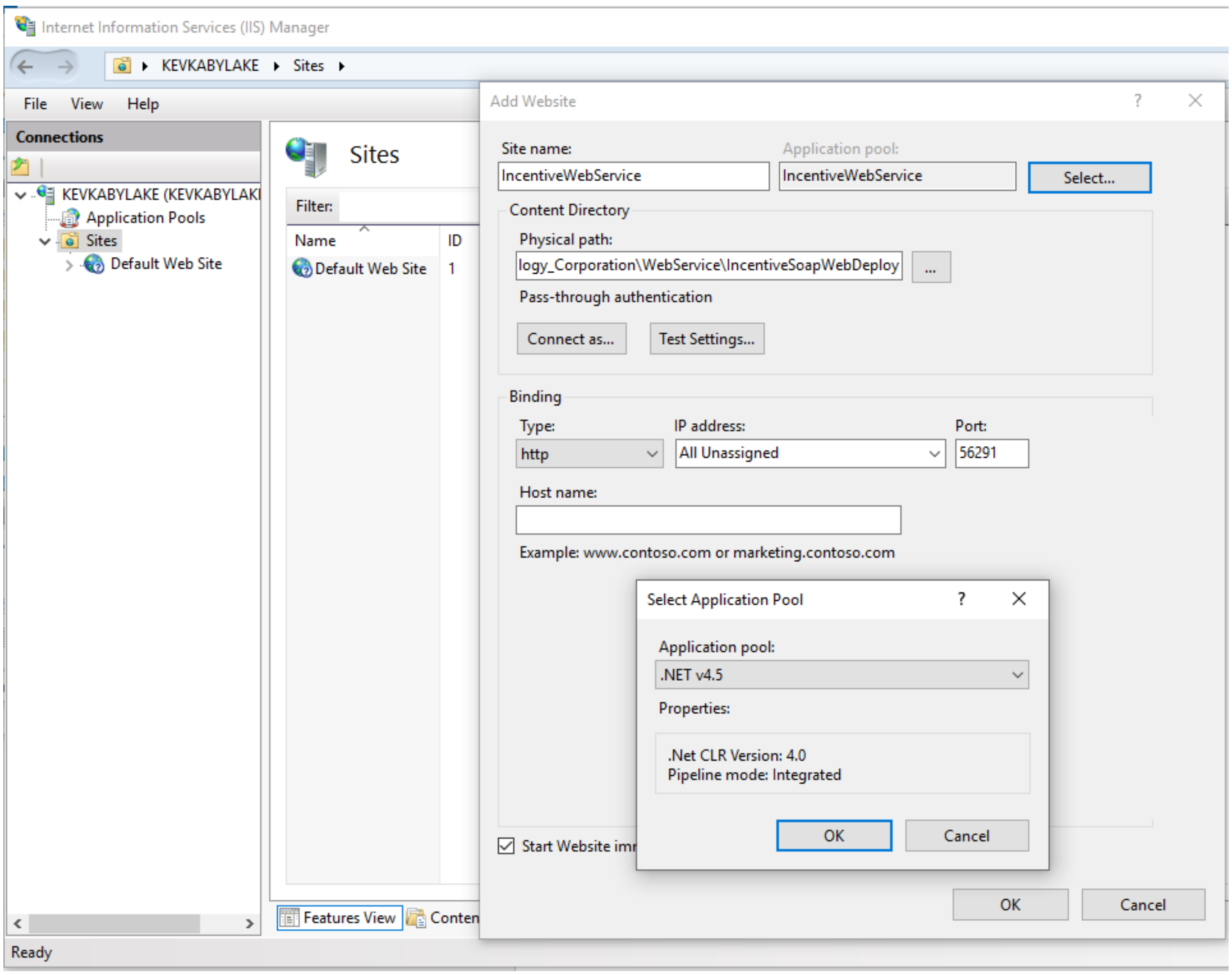
Make sure to add IIS_USRS permission to the folder IncentiveSoapWebDeploy, it is installed under “..\ Control_Technology_Corporation\WebService\IncentiveSoapWebDeploy”:



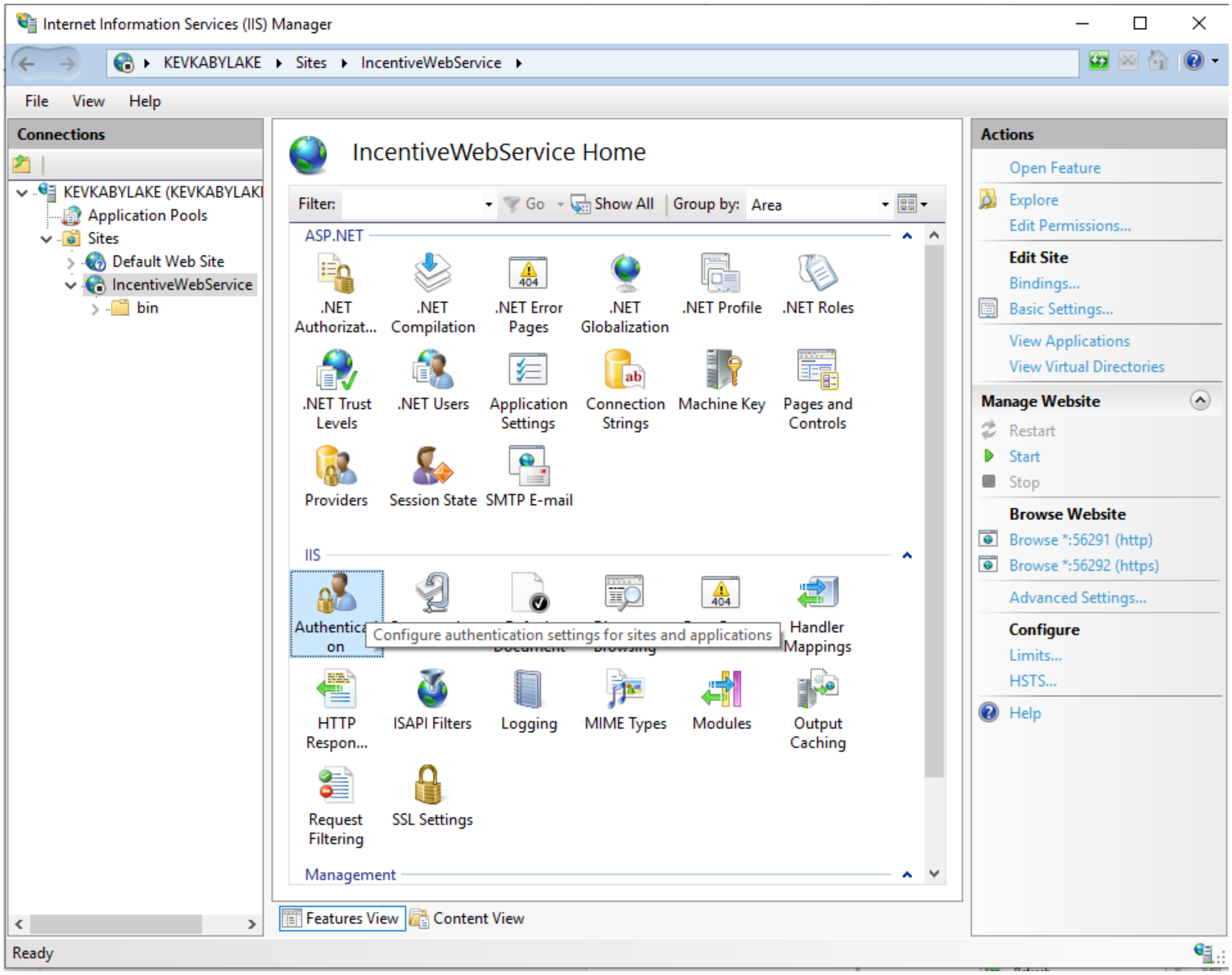
Within IIS Manager expand the hostname listed under Connections and right click “Add Website”. Fill out the desired site name, physical path for the default installation, something like:

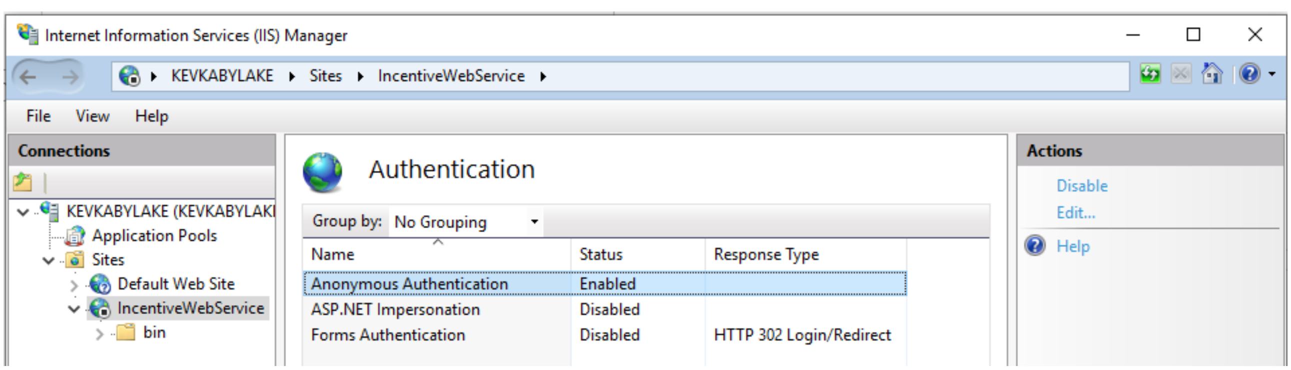
C:\Users\Kevin\Documents\Control_Technology_Corporation\WebService\IncentiveSoapWebDeploy

Also add the desired port for the service to use, in this example 56291. Click Select and select .NET V4.5 as the application pool. Click OK on each of the forms to create the site.

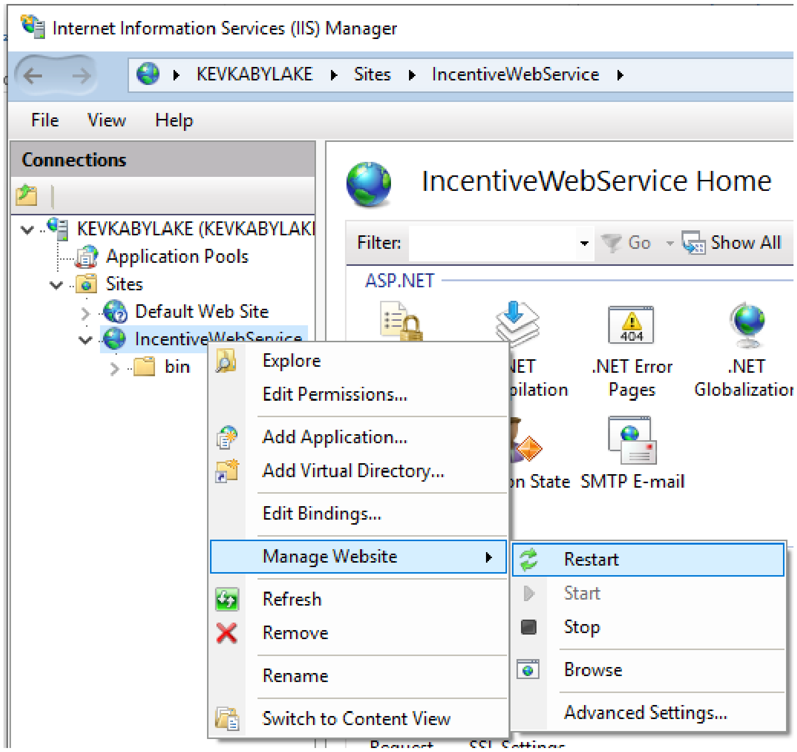


Ensure that Anonymous Authentication is enabled for the web site created.

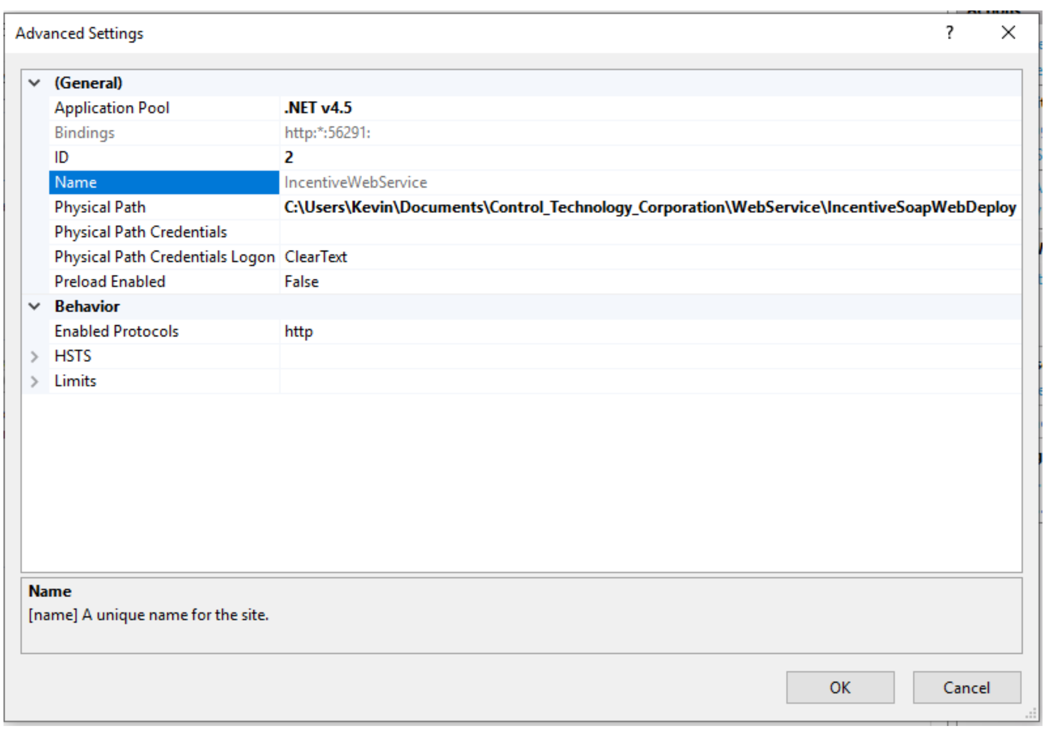




Next restart the web site, right clicking on it and selecting the “Manage Website” selection:

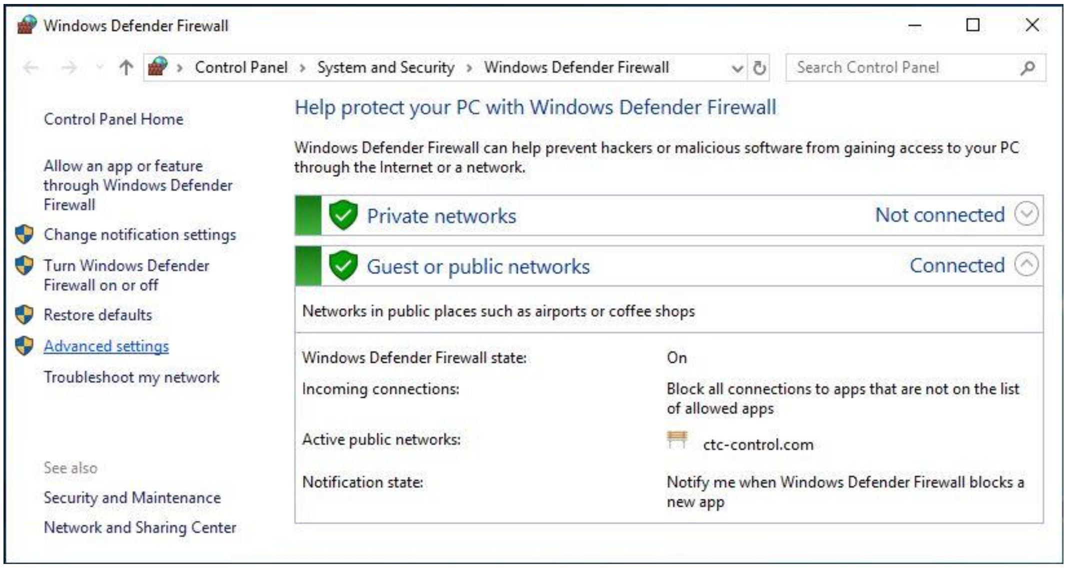


If you select 'Advanced Settings' your web site should look something like below and be operational:

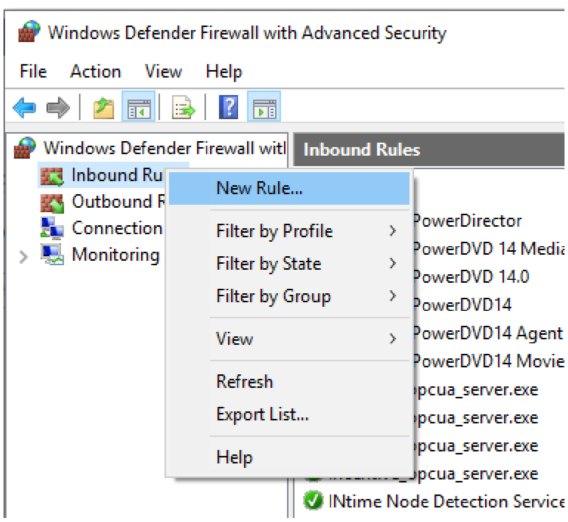


Remember that an Inbound rule will need to be added to the Windows Defender Firewall for Http TCP port 56291 (56292 for https example in later section) or that which was used.

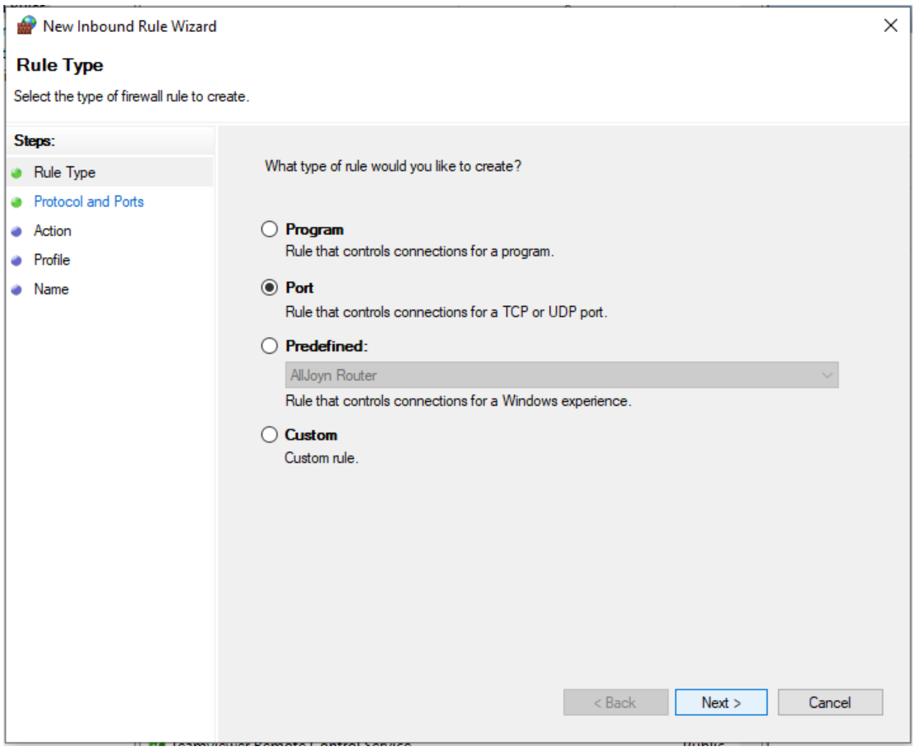
For Inbound firewall settings for access to the Http web site open Windows Defender Firewall->Advanced.



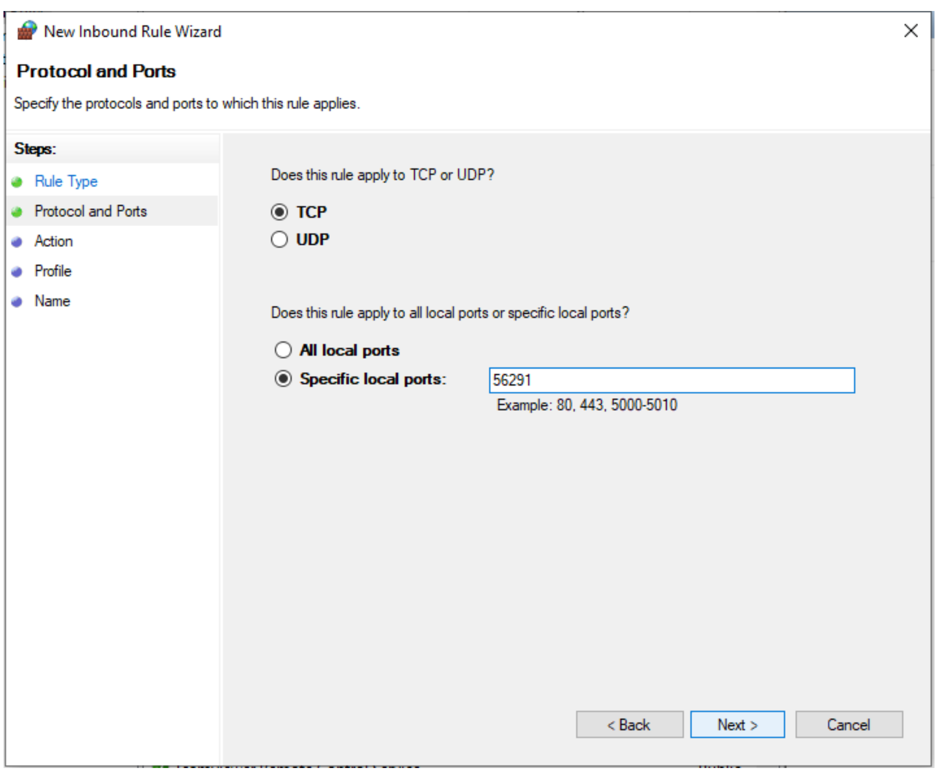
Depending upon the port used you will need to create an Inbound Rule, right click on 'Inbound Rules' and select 'New Rule':



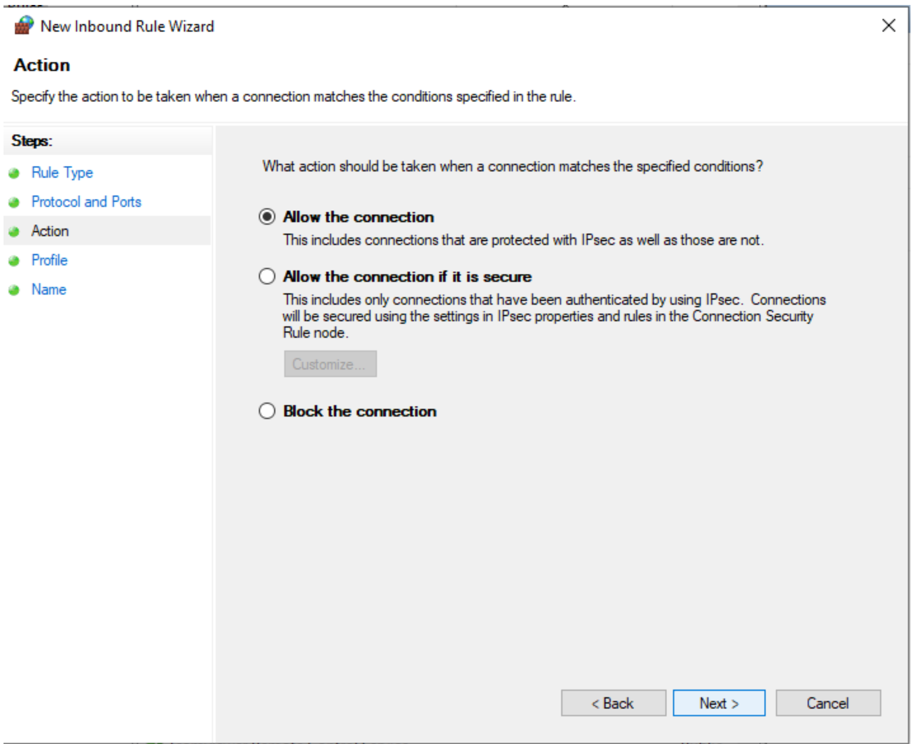
The New Inbound Rule Wizard will appear, select ‘Port’ followed by ‘Next’.



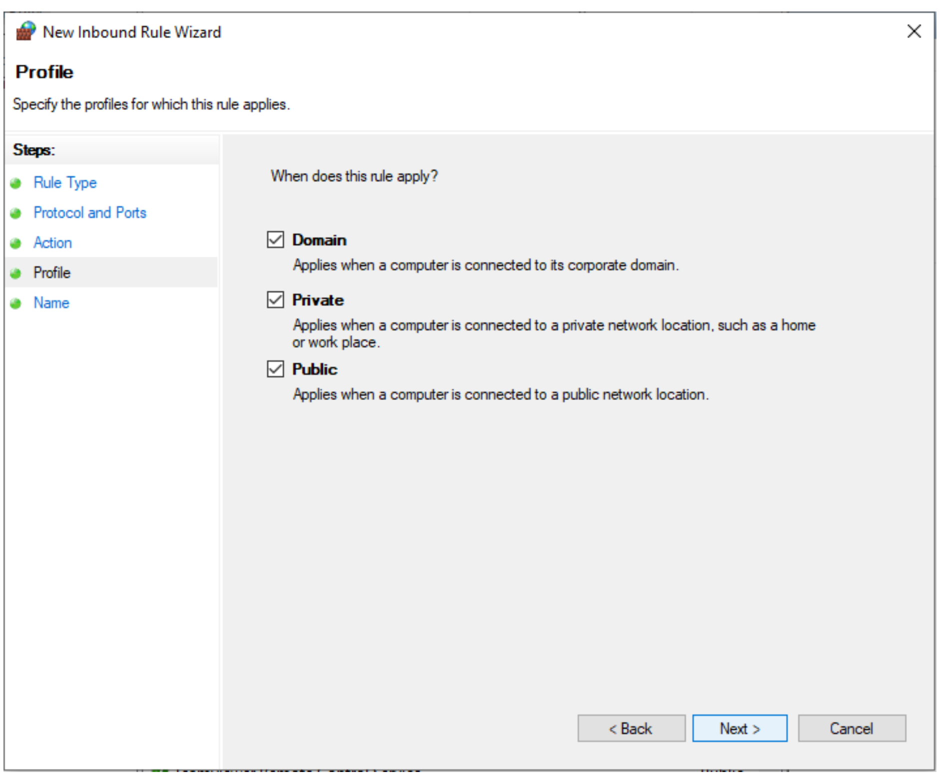
Enter the specific port being using, in this case 56291 using TCP, click Next.



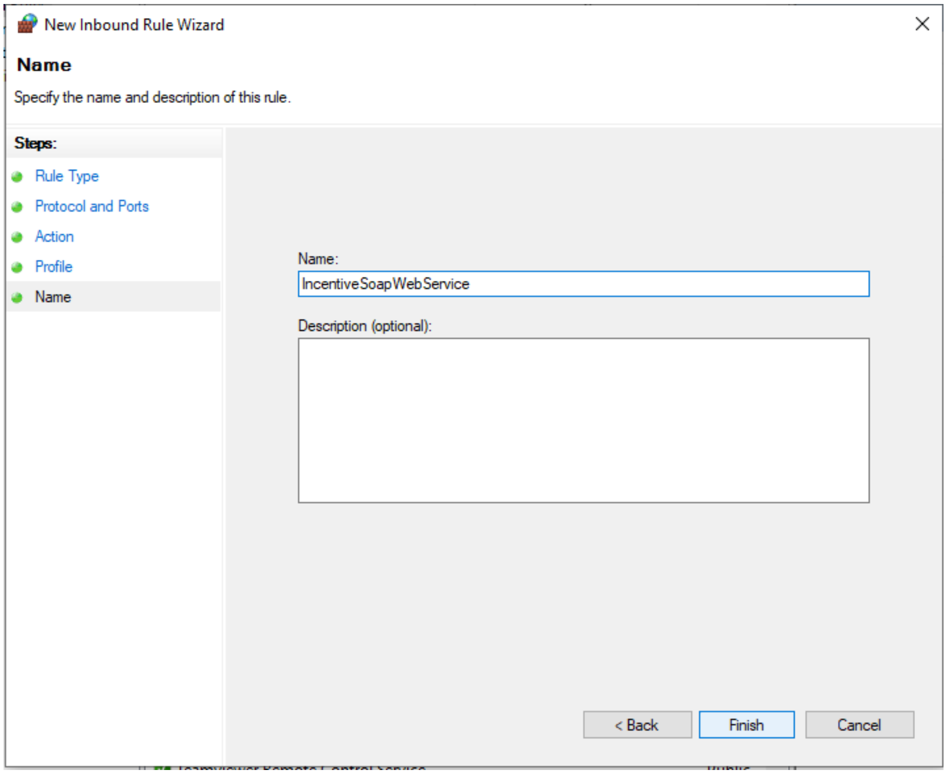
Make sure the action is set to 'Allow the connection', click Next.



Set who this rule will apply to and click 'Next':



Set the Name the rule is to be known by and click Finish.



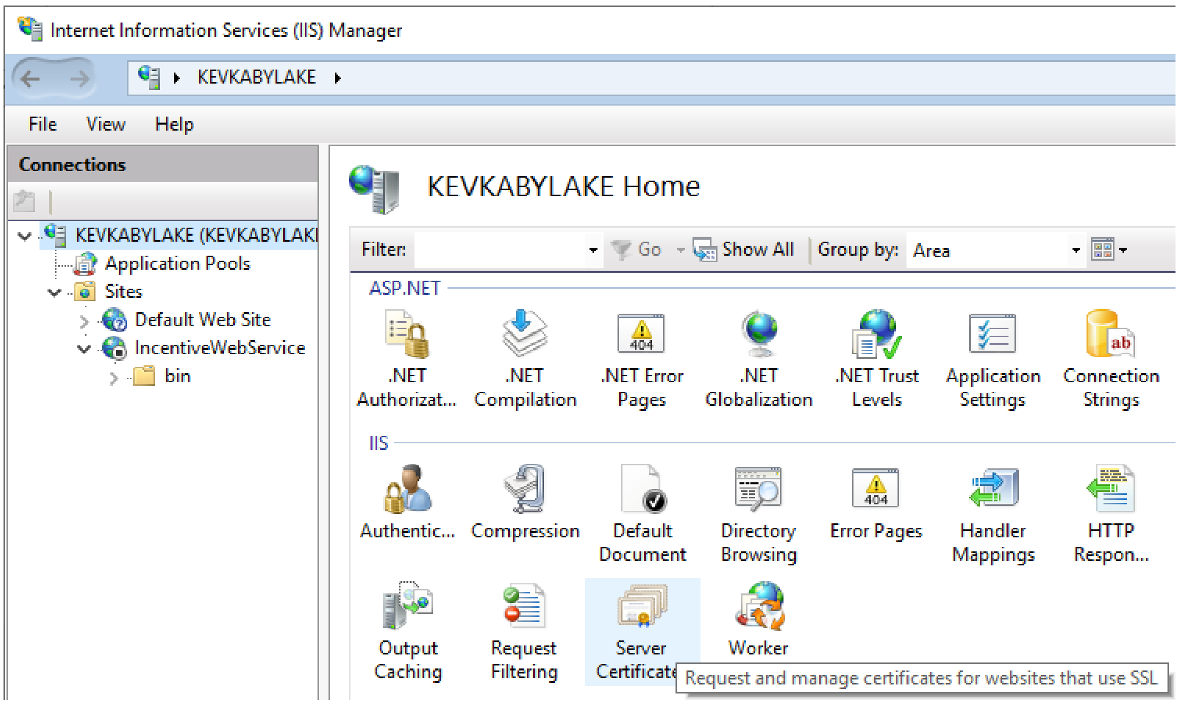
Below shows two Inbound Rules added for 56291 (Http) and 56292 (Https/SSL).

IIS with SSL Option

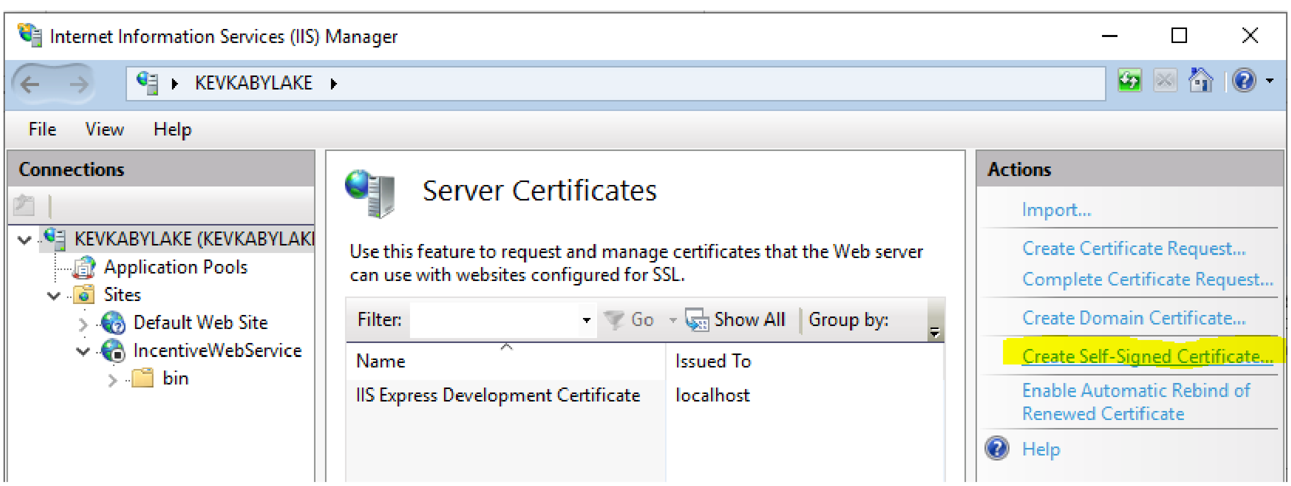
If you wish to add SSL security to the web service you typically would get a certificate from a third party provider. For test purposes a self-signed certificate can be generated. You will also need to assign a port to https references. Using the previous web site as an example lets add port 56292 as an https port as well as create a self-signed certificate.

Creating Self-Signed SSL Certificate

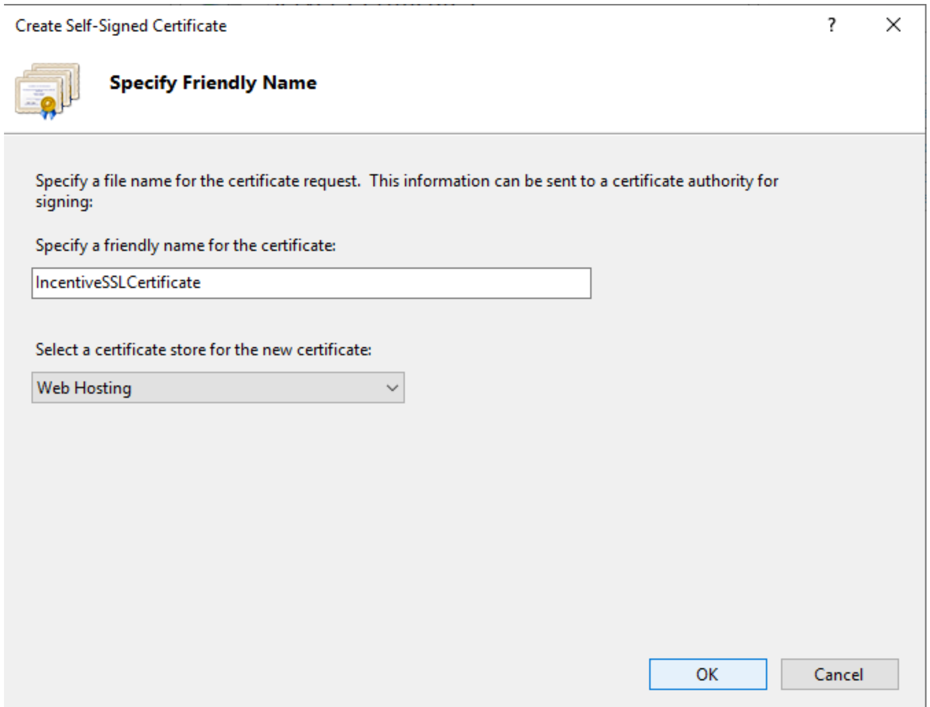
Begin by selecting the Server Certificates icon for all websites.



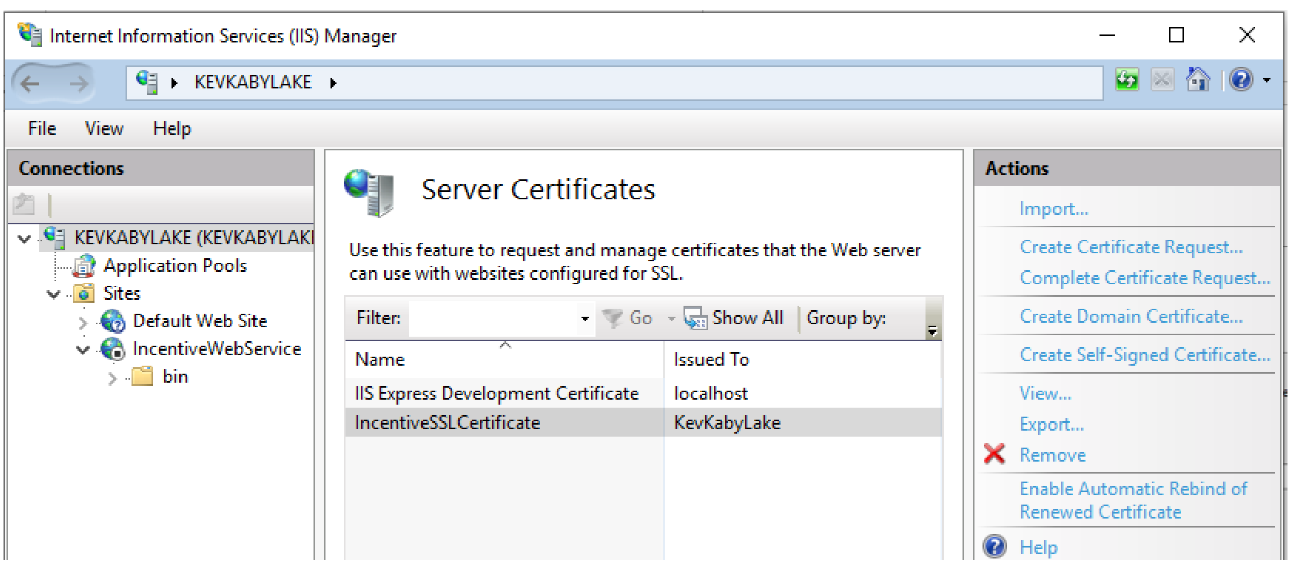
Select 'Create Self-Signed Certificate on the right side of the form, highlighted in yellow.



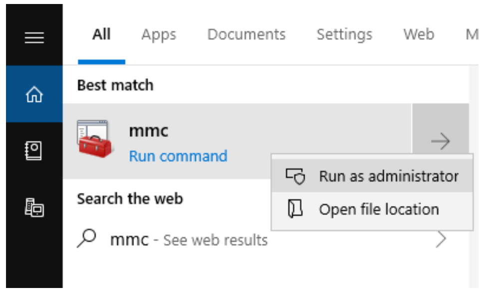
A prompt will appear, enter the name desired for the certificate and select Web Hosting.



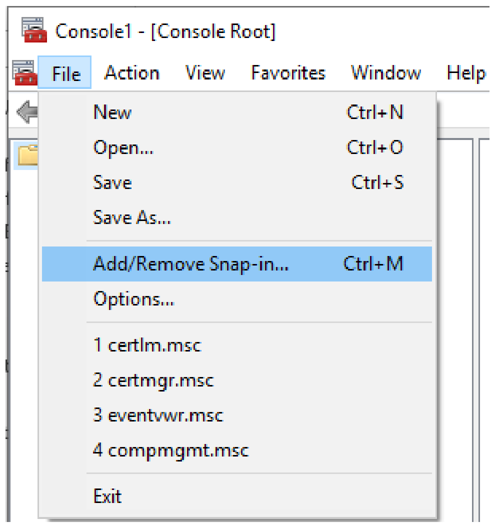
Click OK and the certificate is created and listed as a server certificate.



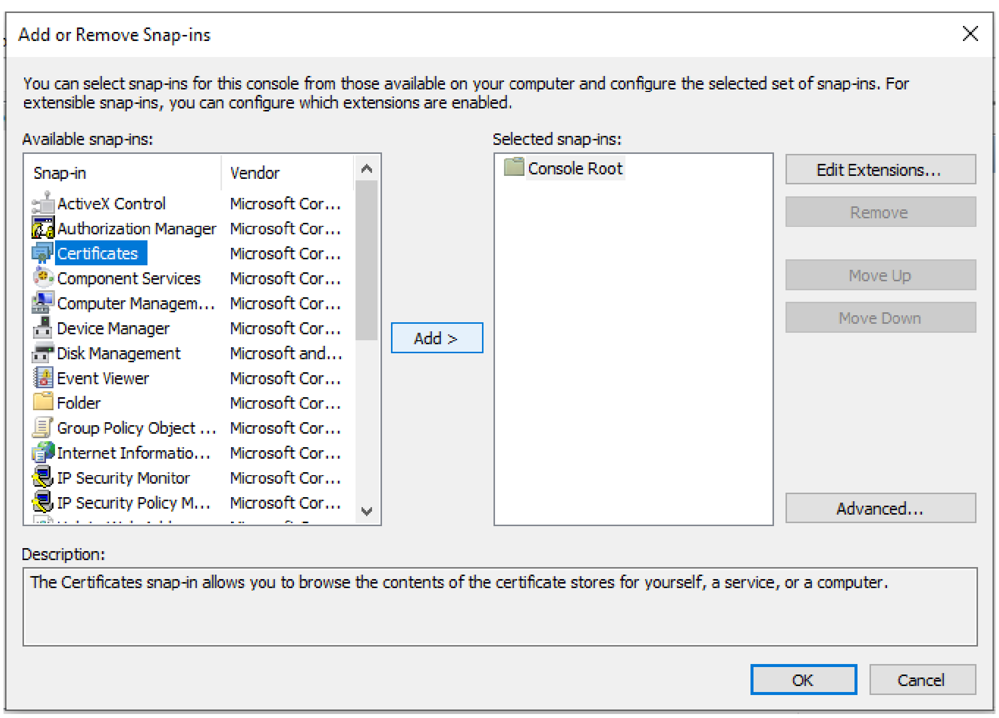
The Certificate Management console for your computer will need to be invoked. In the bottom left of your computer where it says 'Type here to search' type in mmc, then right click the match and click 'Run as administrator'.



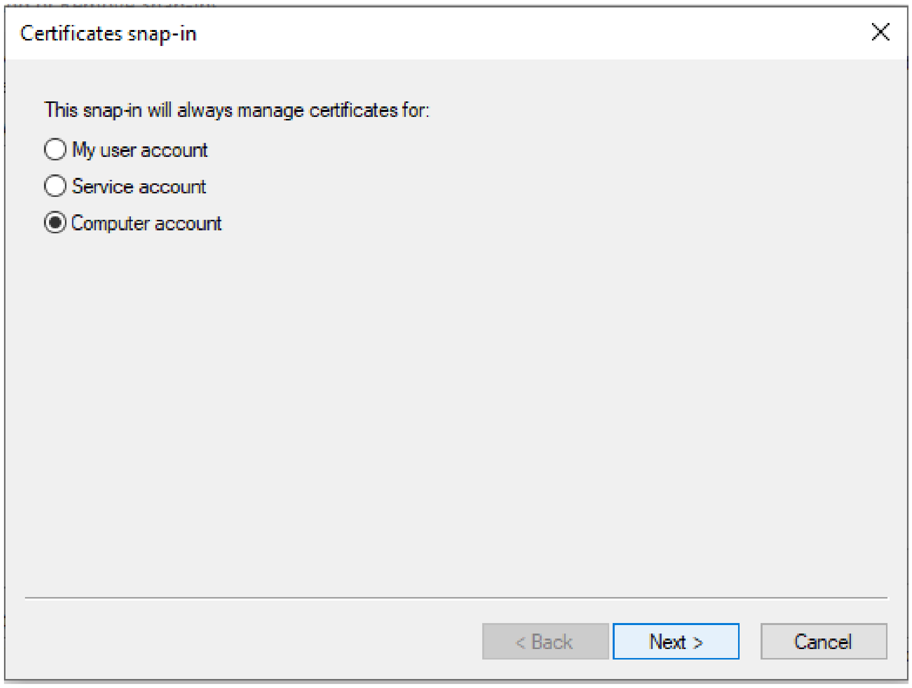
You will need to add the Snap-in for Certificates – Computer Account.

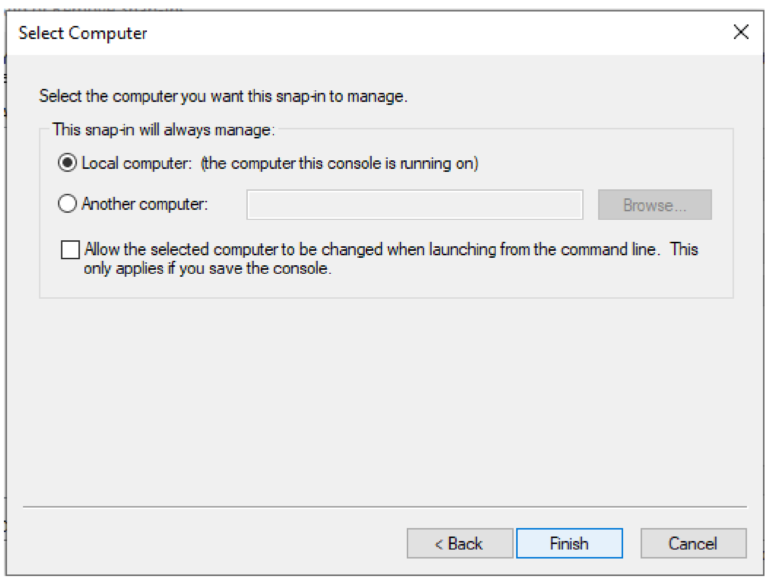


Highlight Certificates and click Add.

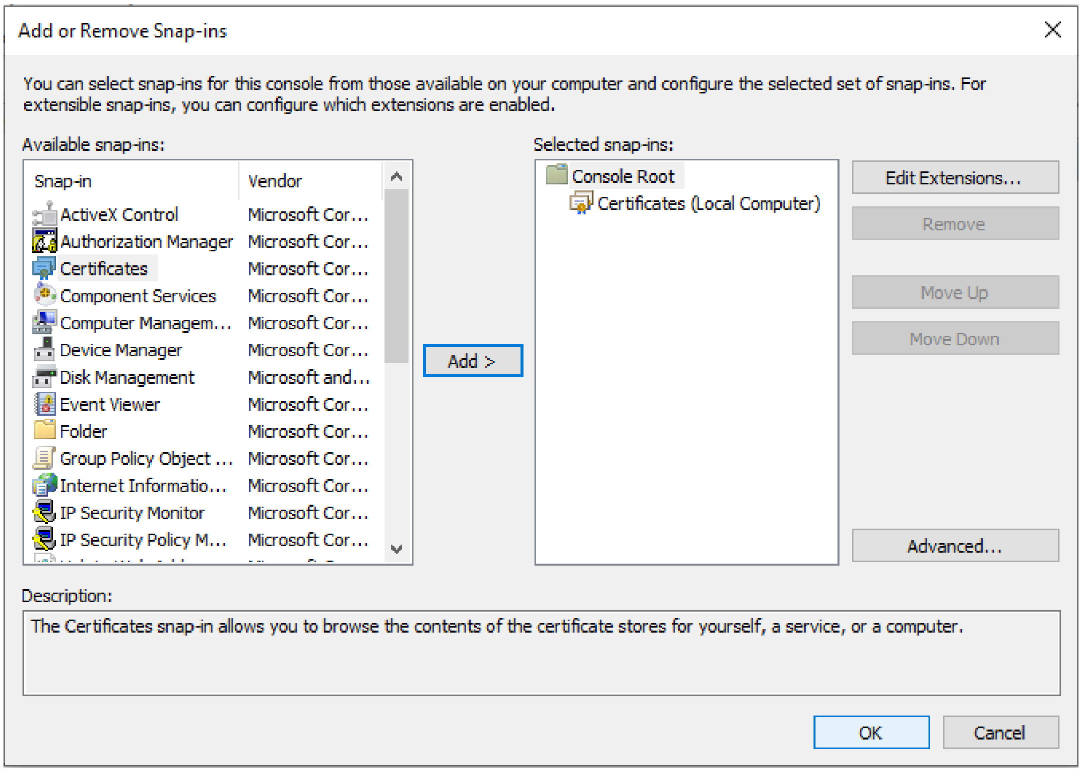


Select 'Computer account' to manage those certificates.

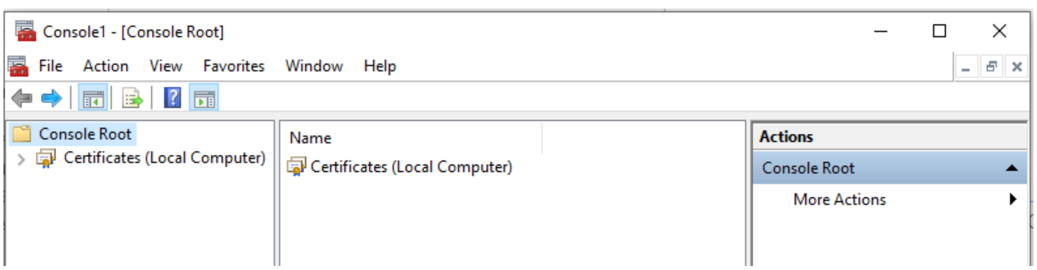




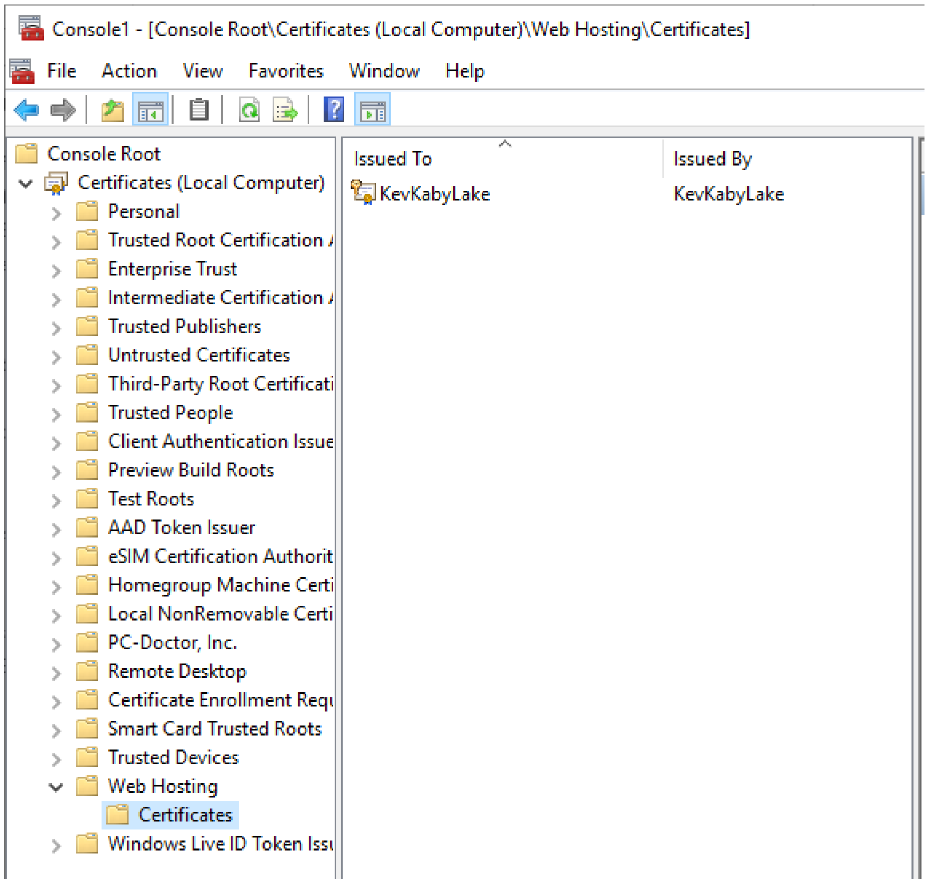
Click Finish.



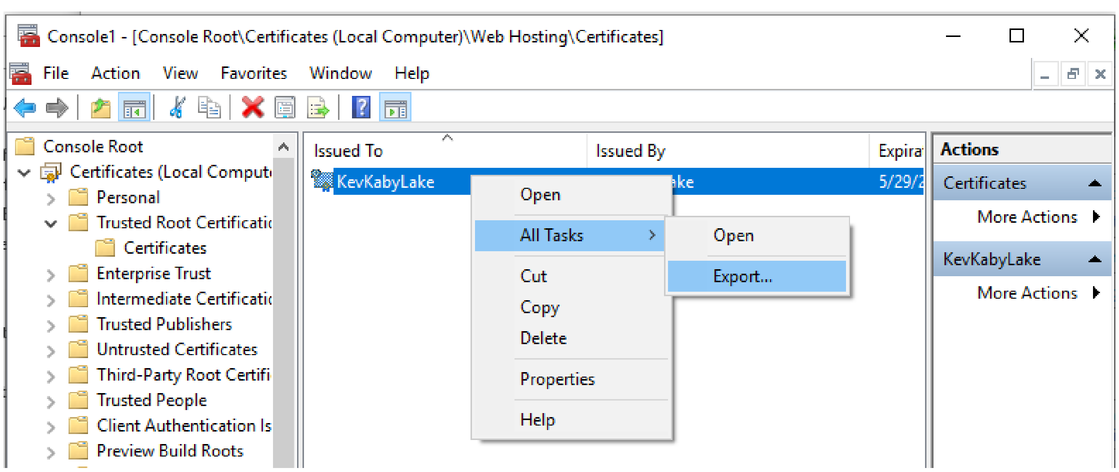
Click OK and the console will appear the the Certificate add-in.



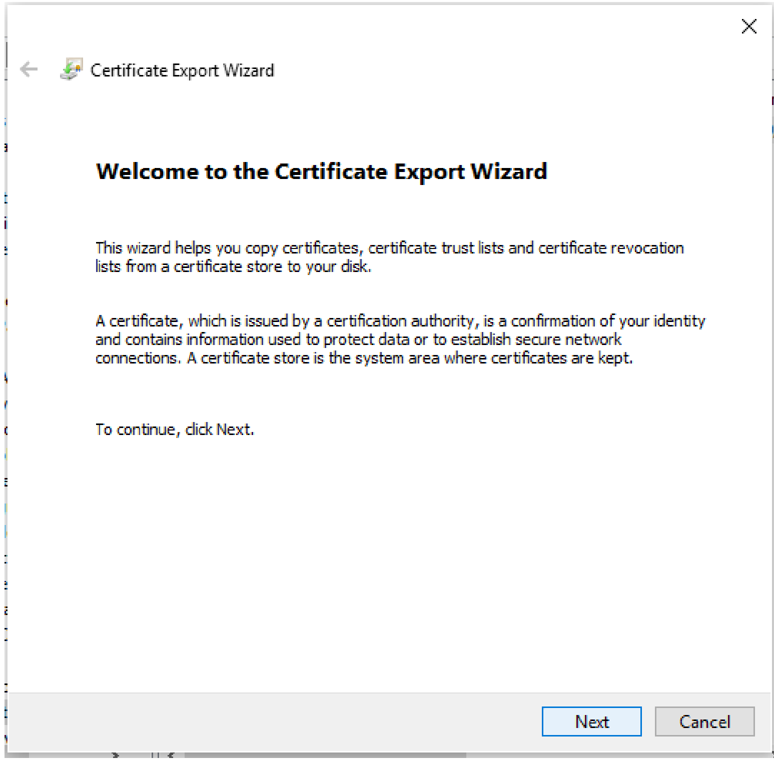
Expand Certificates and click on Web Hosting -> Certificates. The Certificate you created will appear here, Issued to your computer name.



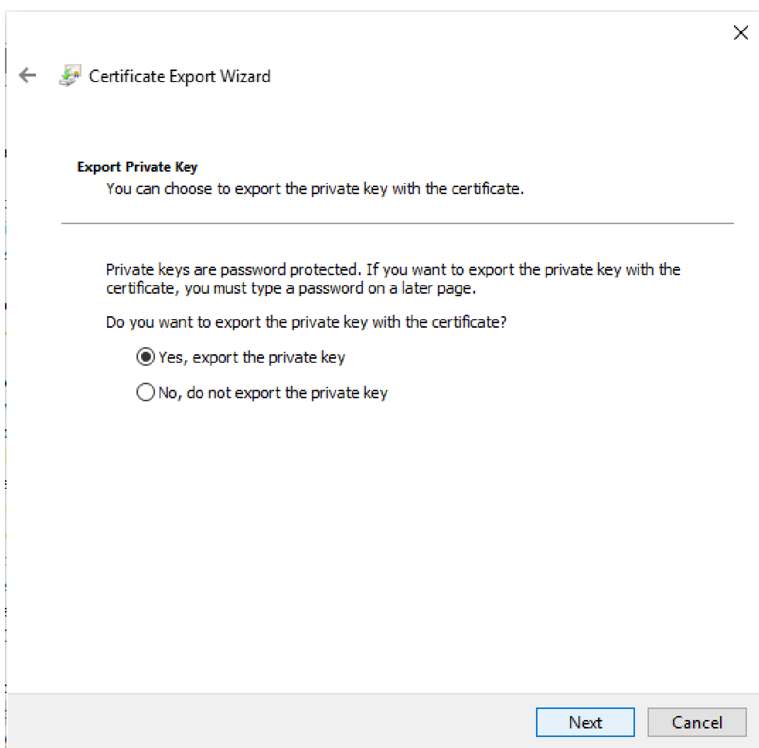
The certificate will need to be exported so the client computer can import it into its Certificate database to setup a trusted authority. Right click on the certificate and select Export.



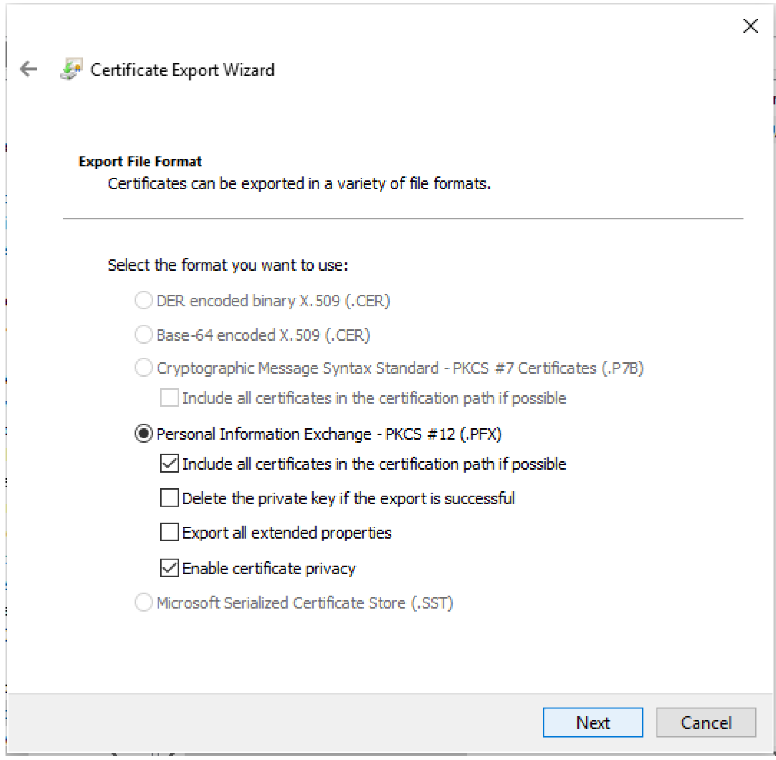
A Certificate Export Wizard will appear.



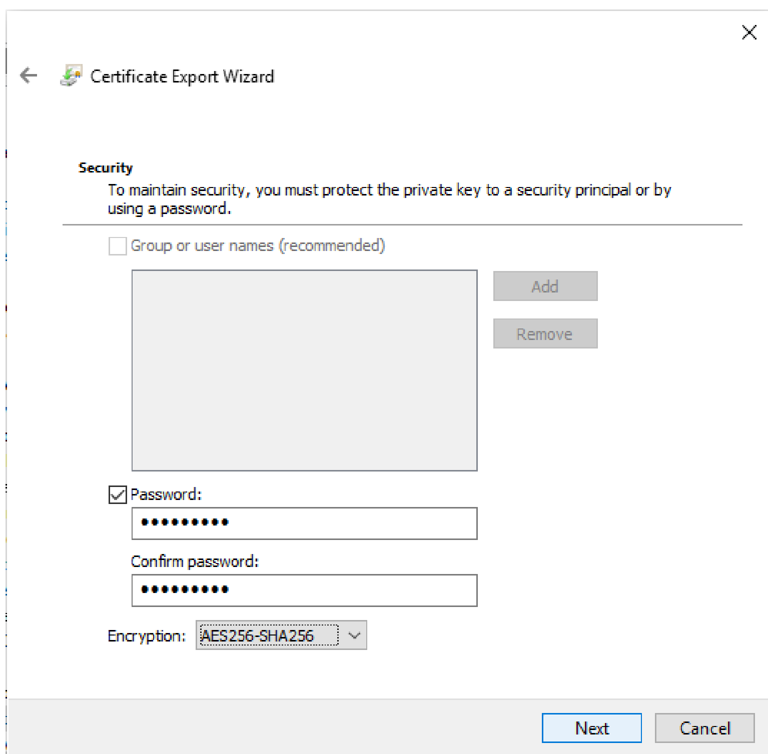
Click Next and select ‘Yes, export the private key’.



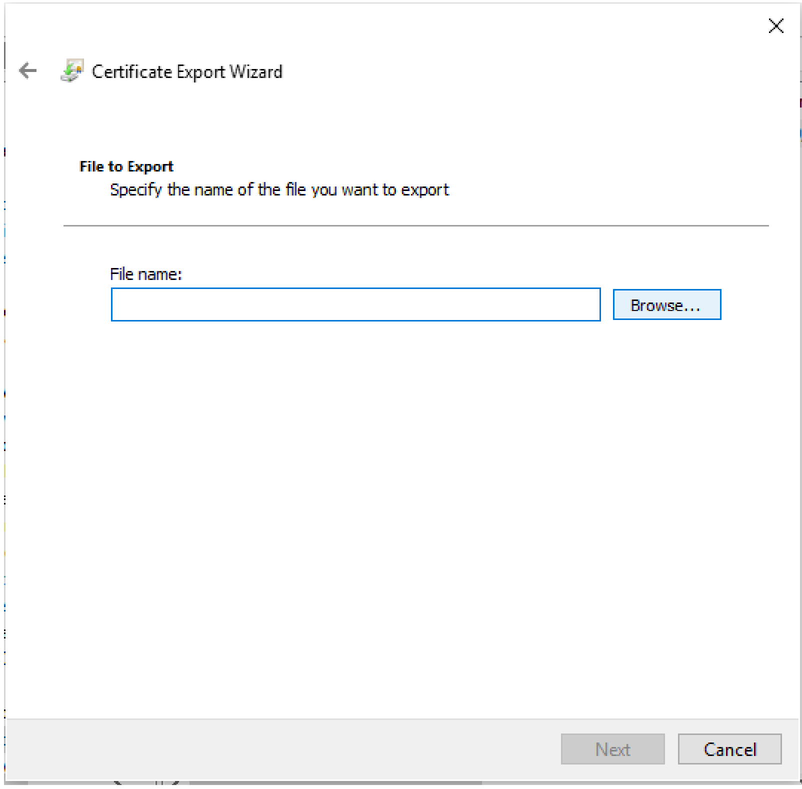
You will use the Personal Information Exchange, click Next.

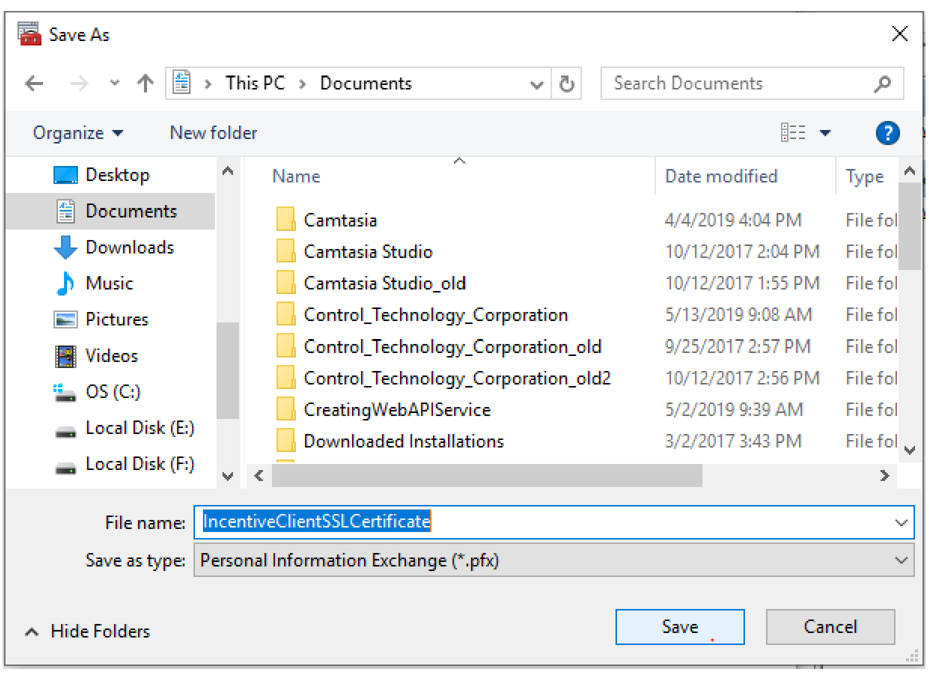


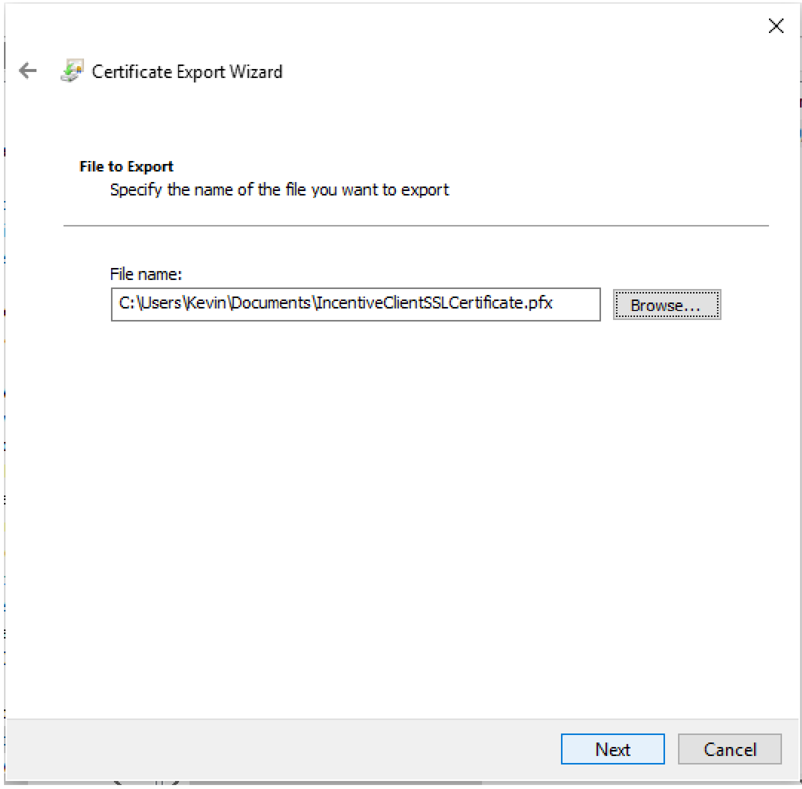
Enter a desired Certificate password and click Next.



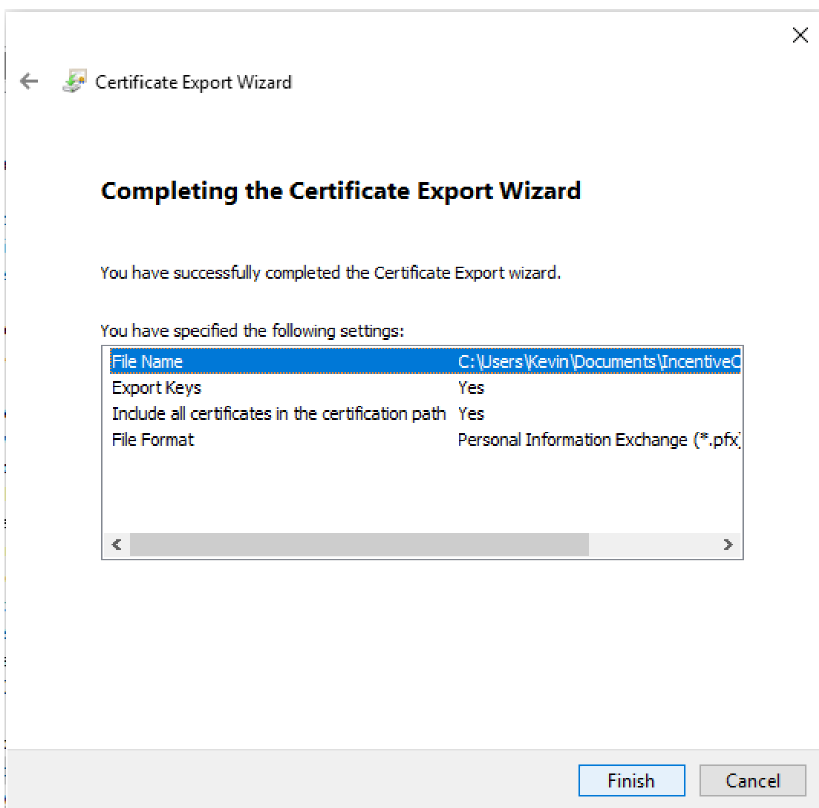
Select where to store the exported certificate and what to name the file.

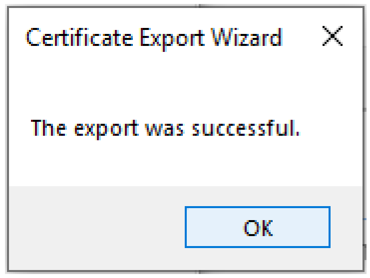

Certificate Export Wizard
File to Export
Specify the name of the file you want to export
File name:
Browse...
Next
Cancel


Save As
This PC > Documents
Search Documents
Organize
New folder
Desktop
Documents
Downloads
Music
Pictures
Videos
OS (C:)
Local Disk (E:)
Local Disk (F:)
Name
Date modified
Type
Camtasia
4/4/2019 4:04 PM
Camtasia Studio
10/12/2017 2:04 PM
Camtasia Studio_old
10/12/2017 1:55 PM
Control_Technology_Corporation
5/13/2019 9:08 AM
Control_Technology_Corporation_old
9/25/2017 2:57 PM
Control_Technology_Corporation_old2
10/12/2017 2:56 PM
CreatingWebAPIService
5/2/2019 9:39 AM
Downloaded Installations
3/2/2017 3:43 PM
File name:
IncentiveClientSSLCertificate
Save as type:
Personal Information Exchange (*.pfx)
Hide Folders
Save
Cancel

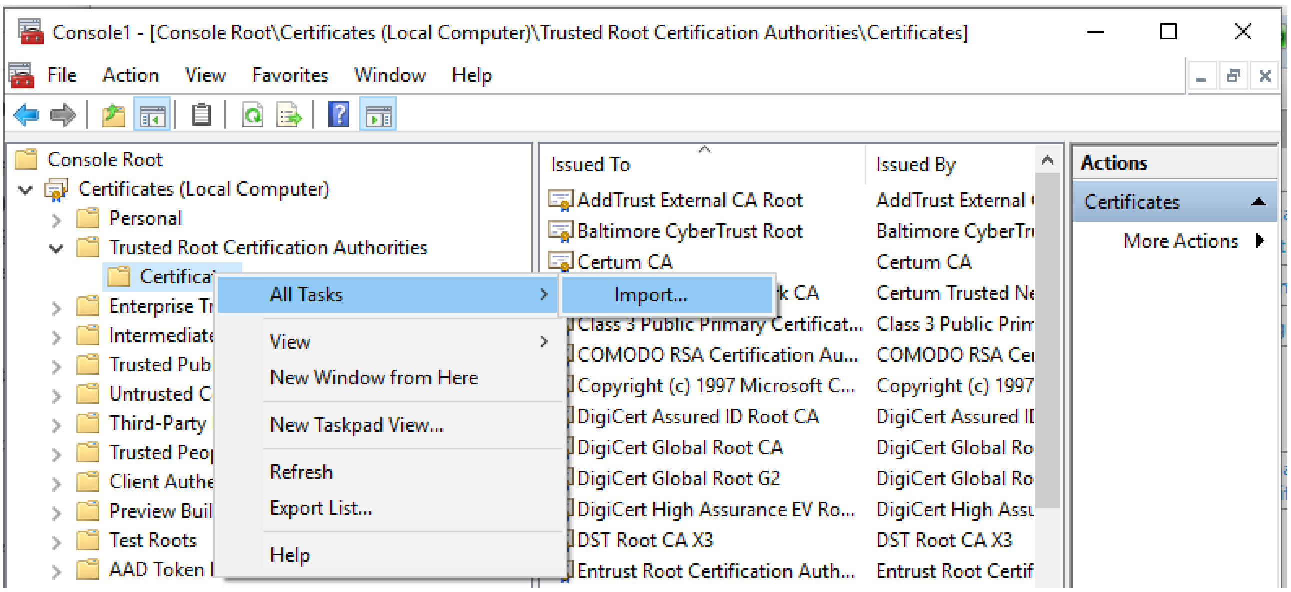


Click Next followed by Finish to complete the export.

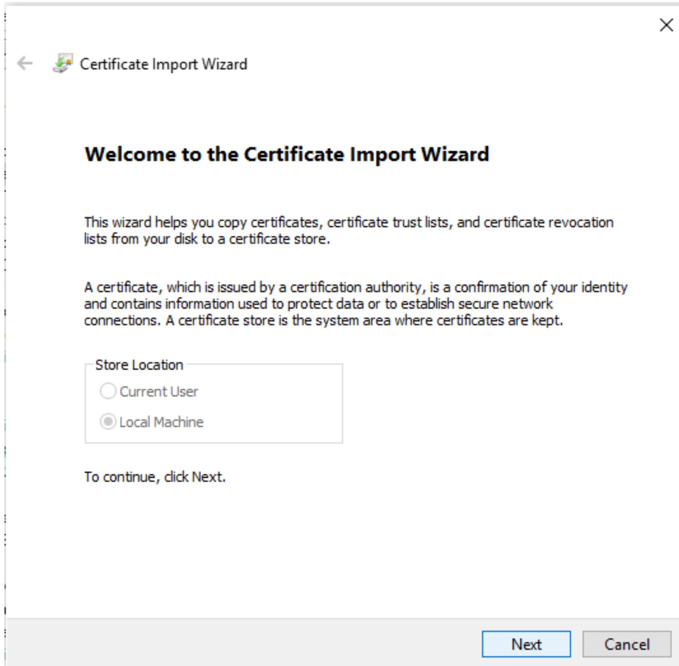




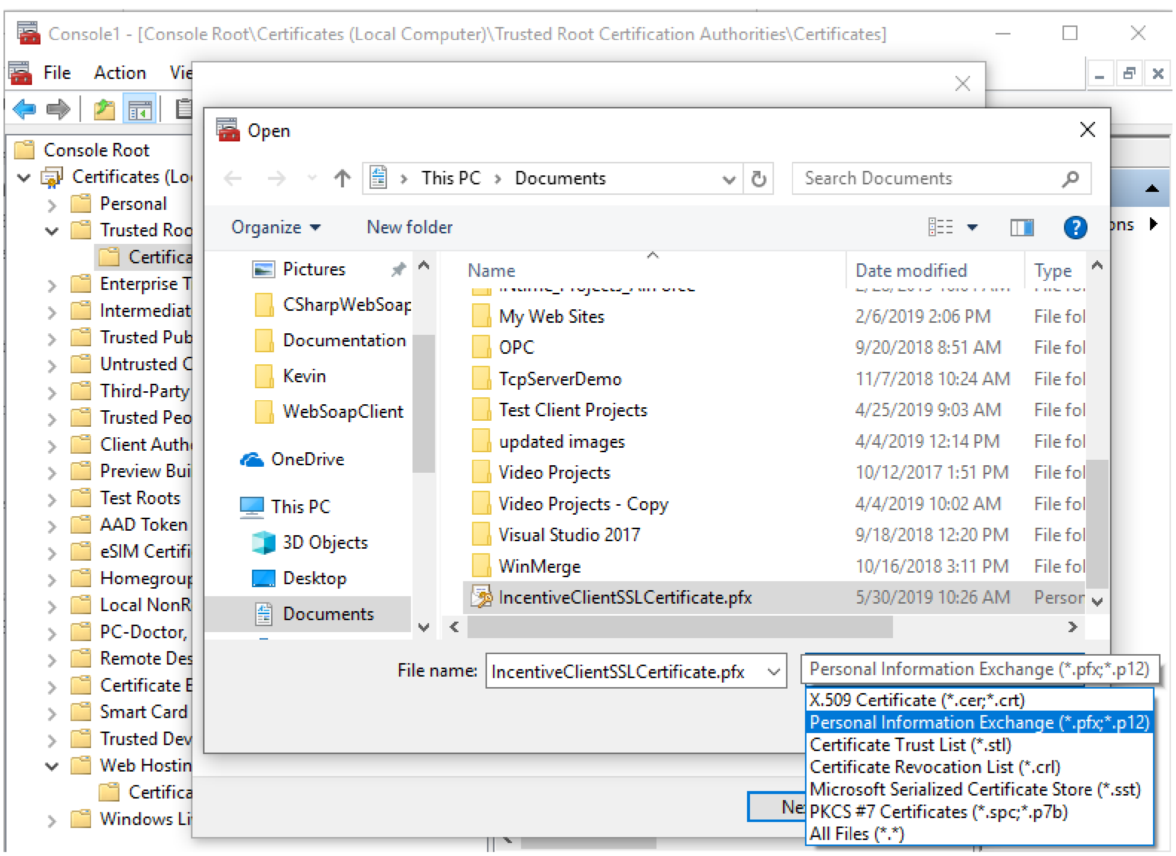
The Certificate IncentiveClientSSLCertificate.pfx is now available to be imported into the client computer's Trusted Root Certificate Authorities. On the client computer open the mmc console as before and add Certificates – Local Computer. Select the Certificate folder under Trusted Root Certification Authorities, right click and select All Tasks->Import.



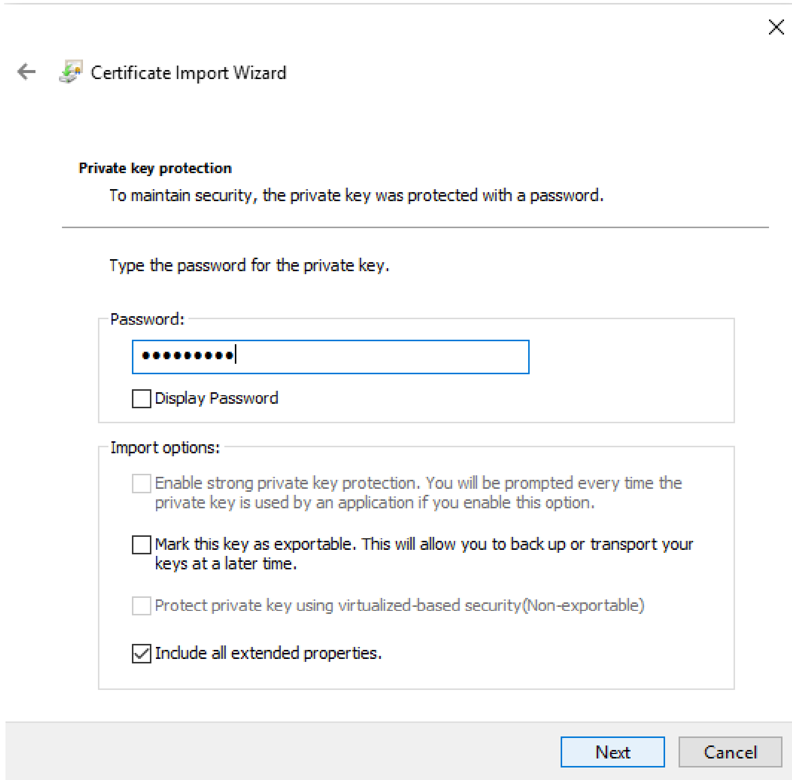
Select 'Next' when the Wizard appears.



Select the pfx file exported by the server, the *.pfx file type must be selected as well.

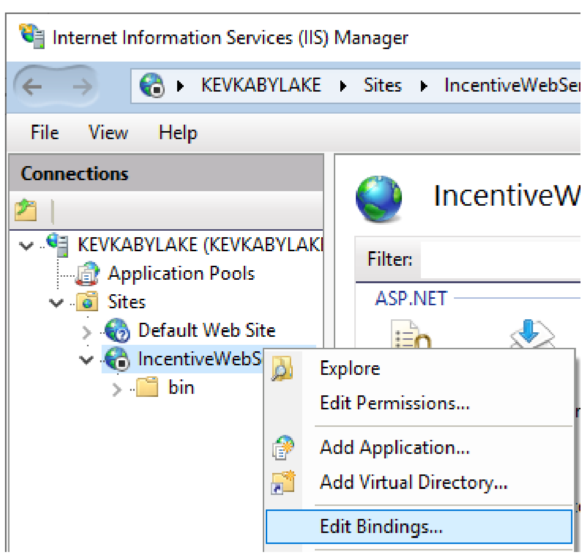


Enter the password used during the server export, click ‘Next’ and the certificate is now imported.

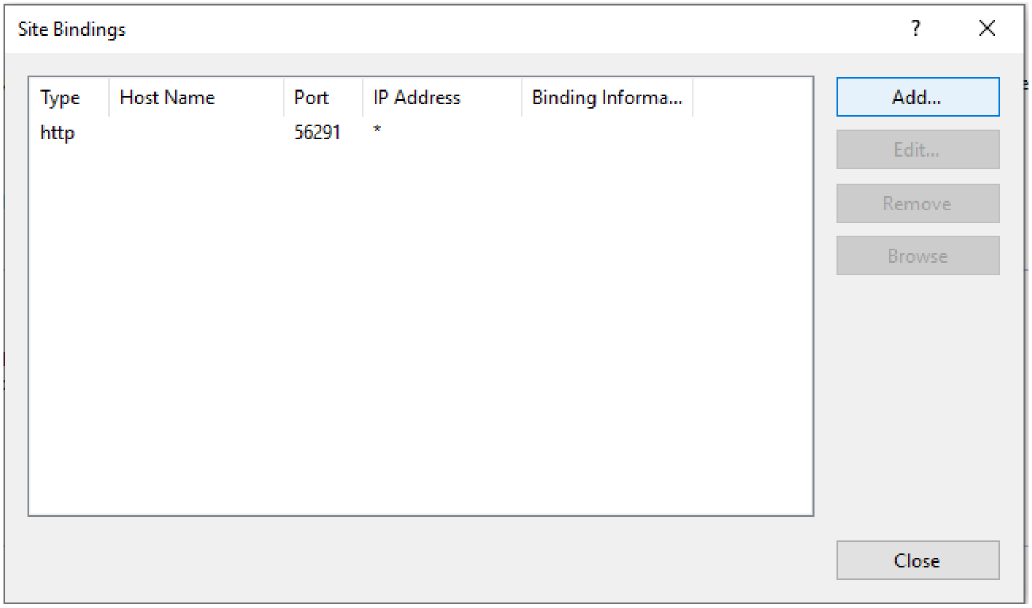


Adding an HTTPS Port to the IIS Web Site

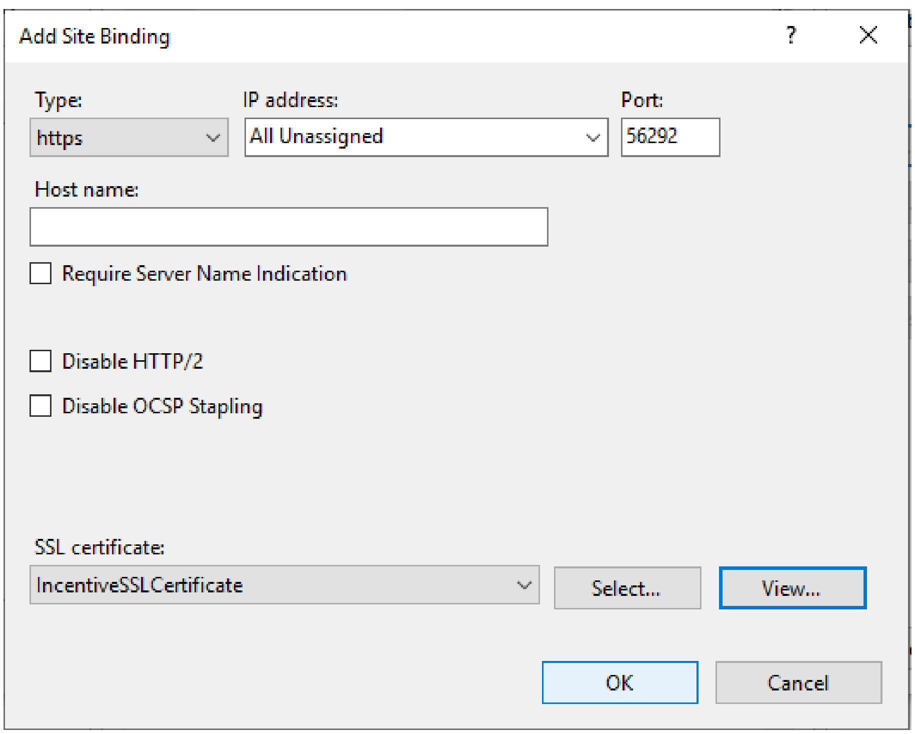
Right click the web site and select ‘Edit Bindings’.

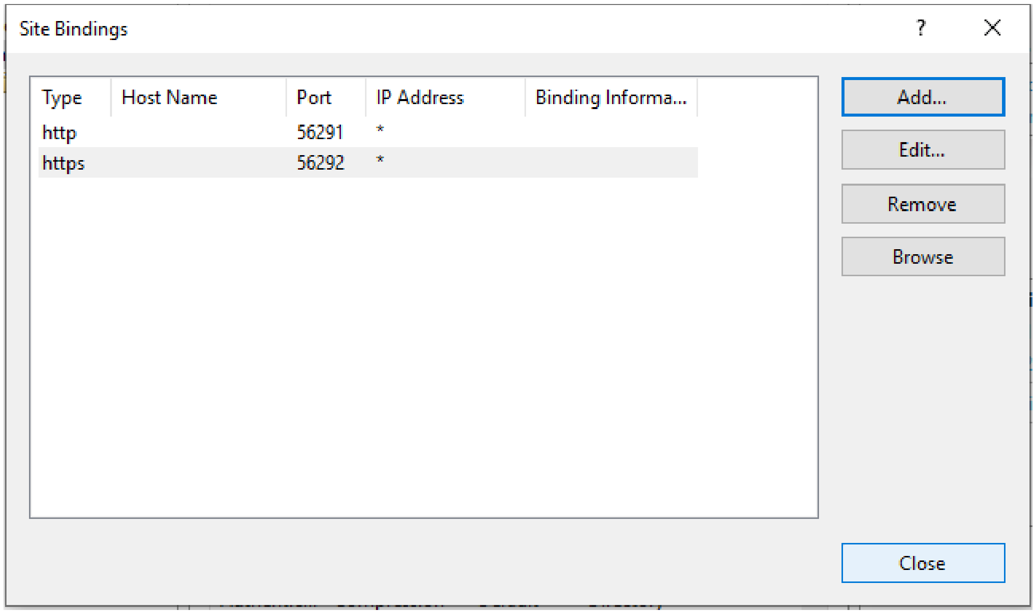


Select ‘Add’ to add a binding. The http binding shoud be removed since this will become a secure SSL site.



Change the type to 'https' and enter the desired port to use. Select the SSL Certificate that was created and click 'OK'.





The http binding should be removed.

The Web.config, in the '..\Control_Technology_Corporation\WebService\IncentiveSoapWebDeploy' directory, needs to be changed to the contents of WebSSL.config and the web site restarted to enable https.

WebSSL.config:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
 <system.serviceModel>
  <bindings>
   <basicHttpBinding>
    <binding name="secureHttpBinding">
     <security mode="Transport">
      <transport clientCredentialType="None"></transport>
     </security>
    </binding>
   </basicHttpBinding>
  </bindings>
  <services>
   <service name="IncentiveSoapWebService.IncentivePLC" behaviorConfiguration="MyServiceTypeBehaviors">
    <endpoint address="soap" contract="IncentiveSoapWebService.IIncentivePLC" binding="basicHttpBinding"
bindingConfiguration="secureHttpBinding" />
    <endpoint contract="IMetadataExchange" binding="mexHttpsBinding" address="mex" />
   </service>
  </services>
  <behaviors>
   <serviceBehaviors>
    <behavior name="MyServiceTypeBehaviors">
     <serviceMetadata httpsGetEnabled="true" />
     <serviceDebug includeExceptionDetailInFaults="true" />
    </behavior>
   </serviceBehaviors>
  </behaviors>
 </system.serviceModel>
 <startup>
  <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" />
```

```
  </startup>
   <system.web>
      <identity impersonate="false" />
   </system.web>
</configuration>
```

Simple Java Web Client Application

Numerous languages can be used to access the Incentive Web Service. This particular example is for Java version 8 update 211. Install both the jdk and the jre. One of the nice features of Java is there is a utility called wsimport (contained in the jdk in bin directory) that can read the WSDL definition file from a web service and automatically generate the needed class structures. Some useful information can be found here:

http://java.boot.by/ocewsd6-guide/ch06.html

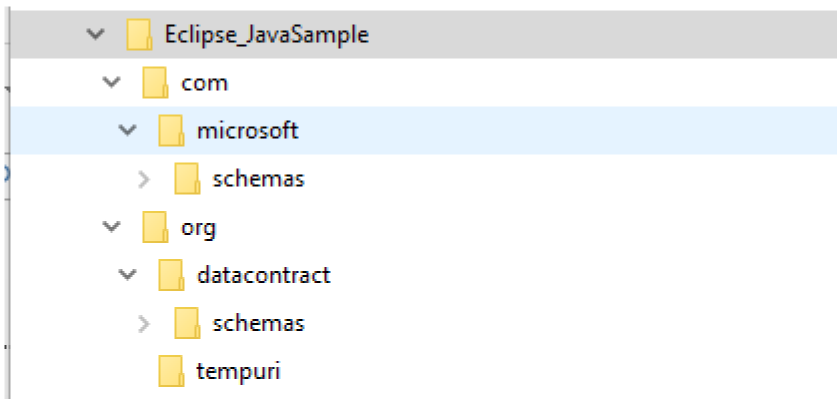
To invoke the utility a command line similar to the following can be used after a path is added for wsimport.exe. When executed from the desired project directory (Eclipse_JavaSample) the following can be expected:

```
wsimport -keep -verbose http://localhost:56291/IncentiveSoapWebService.IncentivePLC.svc?wsdl
parsing WSDL...

Generating code...

com\microsoft\schemas\_2003\_10\serialization\ObjectFactory.java
org\tempuri\GetAxisProperty.java
org\tempuri\GetAxisPropertyResponse.java
org\tempuri\GetAxisResources.java
org\tempuri\GetAxisResourcesResponse.java
org\tempuri\GetData.java
org\tempuri\GetDataResponse.java
org\tempuri\GetResources.java
org\tempuri\GetResourcesResponse.java
org\tempuri\GetShortData.java
org\tempuri\GetShortDataResponse.java
org\tempuri\IIncentivePLC.java
org\tempuri\IncentivePLC.java
org\tempuri\ObjectFactory.java
org\tempuri\PostData.java
org\tempuri\PostDataResponse.java
org\tempuri\package-info.java
org\datacontract\schemas\_2004\_07\incentiveSoapWebService\IncentiveAxis.java
org\datacontract\schemas\_2004\_07\incentiveSoapWebService \IncentiveRegister.java
org\datacontract\schemas\_2004\_07\incentiveSoapWebService \IncentiveResources.java
org\datacontract\schemas\_2004\_07\incentiveSoapWebService \ObjectFactory.java
org\datacontract\schemas\_2004\_07\incentiveSoapWebService \package-info.java

Compiling code...

Javac -d C:\Users\Kevin\Documents\INtime_Projects\webmon\. -classpath C:\Program Files\Java\jdk1.8.0_211/lib/tools.jar;C:\Program
Files\Java\jdk1.8.0_211/classes -Xbootclasspath/p:C:\Program Files\Java\jdk1.8.0_211\jre\lib\rt.jar;C:\Program
Files\Java\jdk1.8.0_211\jre\lib\rt.jar
C:\Users\Kevin\Documents\INtime_Projects\webmon\.\com\microsoft\schemas\_2003\_10\serialization\ObjectFactory.java
C:\Users\Kevin\Documents\INtime_Projects\webmon\.\org\tempuri\GetAxisProperty.java
C:\Users\Kevin\Documents\INtime_Projects\webmon\.\org\tempuri\GetAxisPropertyResponse.java
C:\Users\Kevin\Documents\INtime_Projects\webmon\.\org\tempuri\GetAxisResources.java
C:\Users\Kevin\Documents\INtime_Projects\webmon\.\org\tempuri\GetAxisResourcesResponse.java
C:\Users\Kevin\Documents\INtime_Projects\webmon\.\org\tempuri\GetData.java
```

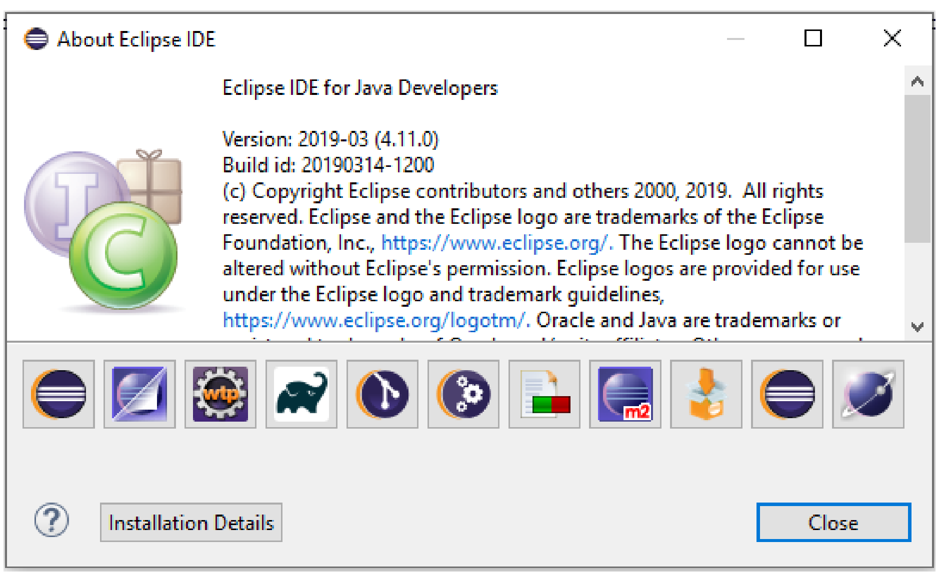
```
C:\Users\Kevin\Documents\INtime_Projects\webmon\.\org\tempuri\GetDataResponse.java
C:\Users\Kevin\Documents\INtime_Projects\webmon\.\org\tempuri\GetResources.java
C:\Users\Kevin\Documents\INtime_Projects\webmon\.\org\tempuri\GetResourcesResponse.java
C:\Users\Kevin\Documents\INtime_Projects\webmon\.\org\tempuri\GetShortData.java
C:\Users\Kevin\Documents\INtime_Projects\webmon\.\org\tempuri\GetShortDataResponse.java
C:\Users\Kevin\Documents\INtime_Projects\webmon\.\org\tempuri\IIncentivePLC.java
C:\Users\Kevin\Documents\INtime_Projects\webmon\.\org\tempuri\IncentivePLC.java
C:\Users\Kevin\Documents\INtime_Projects\webmon\.\org\tempuri\ObjectFactory.java
C:\Users\Kevin\Documents\INtime_Projects\webmon\.\org\tempuri\PostData.java
C:\Users\Kevin\Documents\INtime_Projects\webmon\.\org\tempuri\PostDataResponse.java
C:\Users\Kevin\Documents\INtime_Projects\webmon\.\org\tempuri\package-info.java
C:\Users\Kevin\Documents\INtime_Projects\webmon\.\org\datacontract\schemas\_2004\_07\incentivejsonrestservice\IncentiveAxis.java
C:\Users\Kevin\Documents\INtime_Projects\webmon\.\org\datacontract\schemas\_2004\_07\incentivejsonrestservice\IncentiveRegister.java
C:\Users\Kevin\Documents\INtime_Projects\webmon\.\org\datacontract\schemas\_2004\_07\incentivejsonrestservice\IncentiveResources.java
C:\Users\Kevin\Documents\INtime_Projects\webmon\.\org\datacontract\schemas\_2004\_07\incentivejsonrestservice\ObjectFactory.java
C:\Users\Kevin\Documents\INtime_Projects\webmon\.\org\datacontract\schemas\_2004\_07\incentivejsonrestservice\package-info.java
```

Remember to substitute ‘localhost’ for the actual host name if on a remote system for the url.

After execution the directory will contain the generated files:



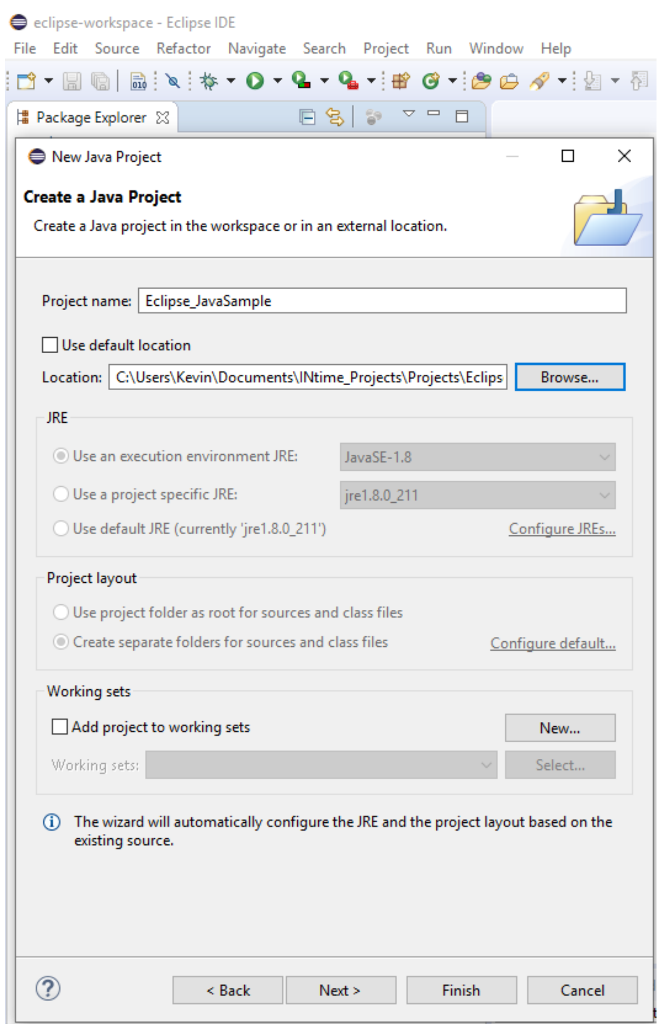
For this development Eclipse was used:

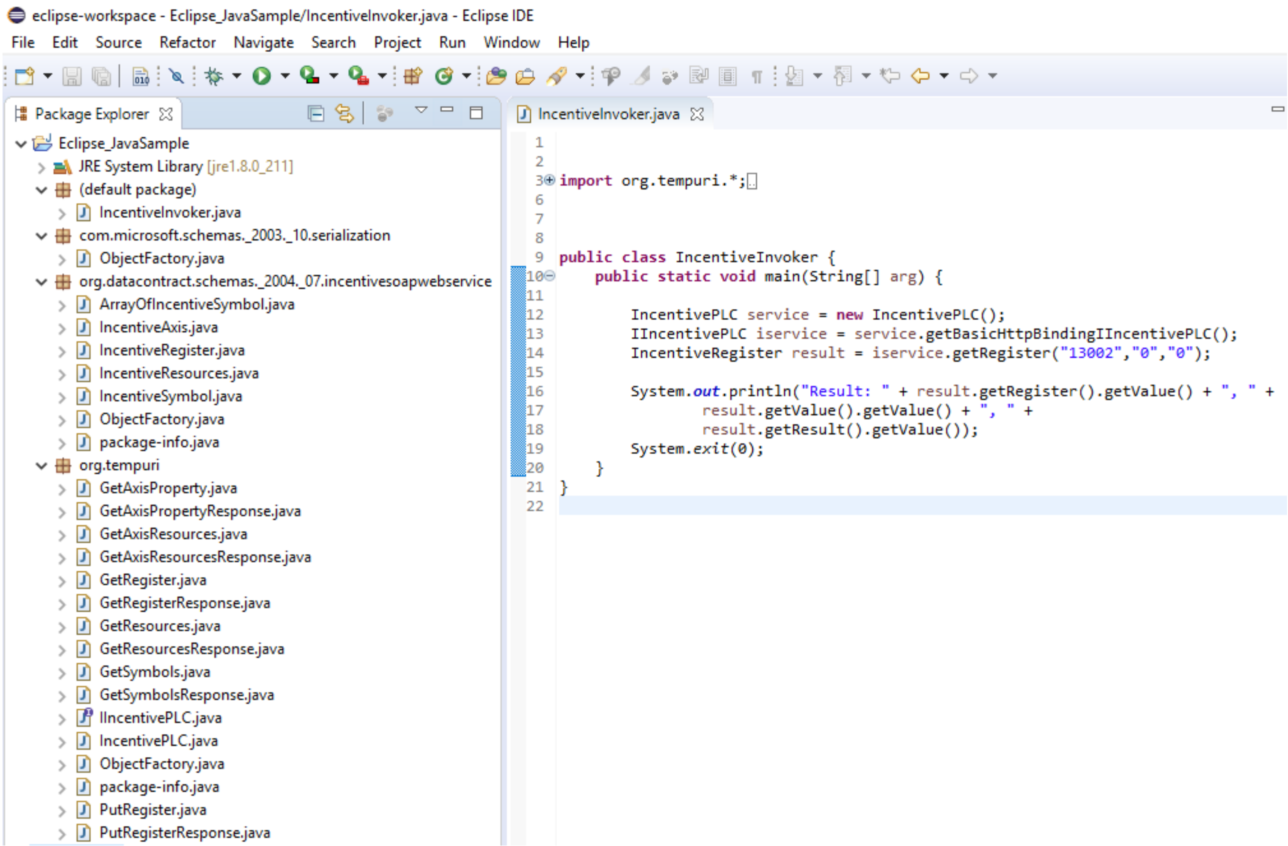


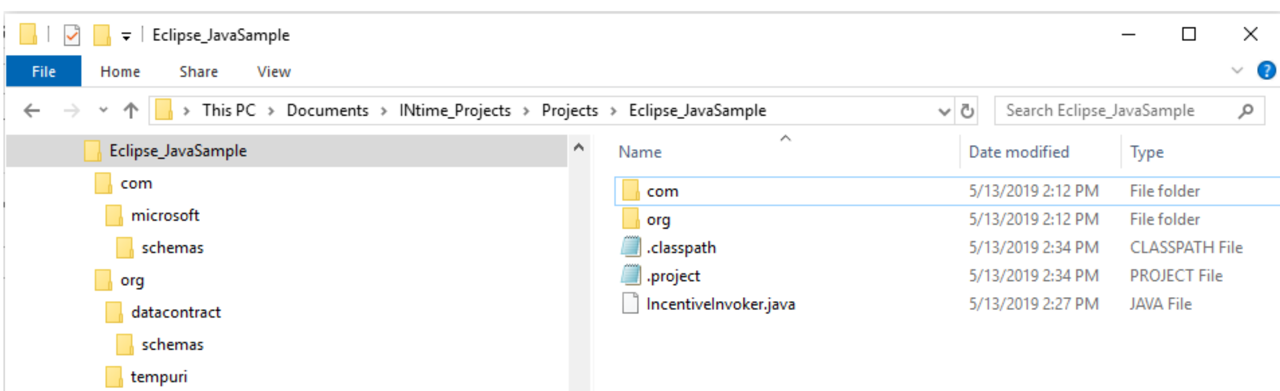
In the root of your project folder create a java file called IncentiveAPIInvoker.java with its contents something like:

```java
import org.tempuri.*;

import org.datacontract.schemas._2004._07.incentivesoapwebservice.IncentiveRegister;


public class IncentiveInvoker {
    public static void main(String[] arg) {

        IncentivePLC service = new IncentivePLC();
        IIncentivePLC iservice = service.getBasicHttpBindingIIncentivePLC();
        IncentiveRegister result = iservice.getRegister("13002","0","0");

        System.out.println("Result: " + result.getRegister().getValue() + ", " +
                result.getValue().getValue() + ", " +
                result.getResult().getValue());
        System.exit(0);
    }
}
```

Now create a new project using Eclipse, in this case Eclipse_JavaSample:





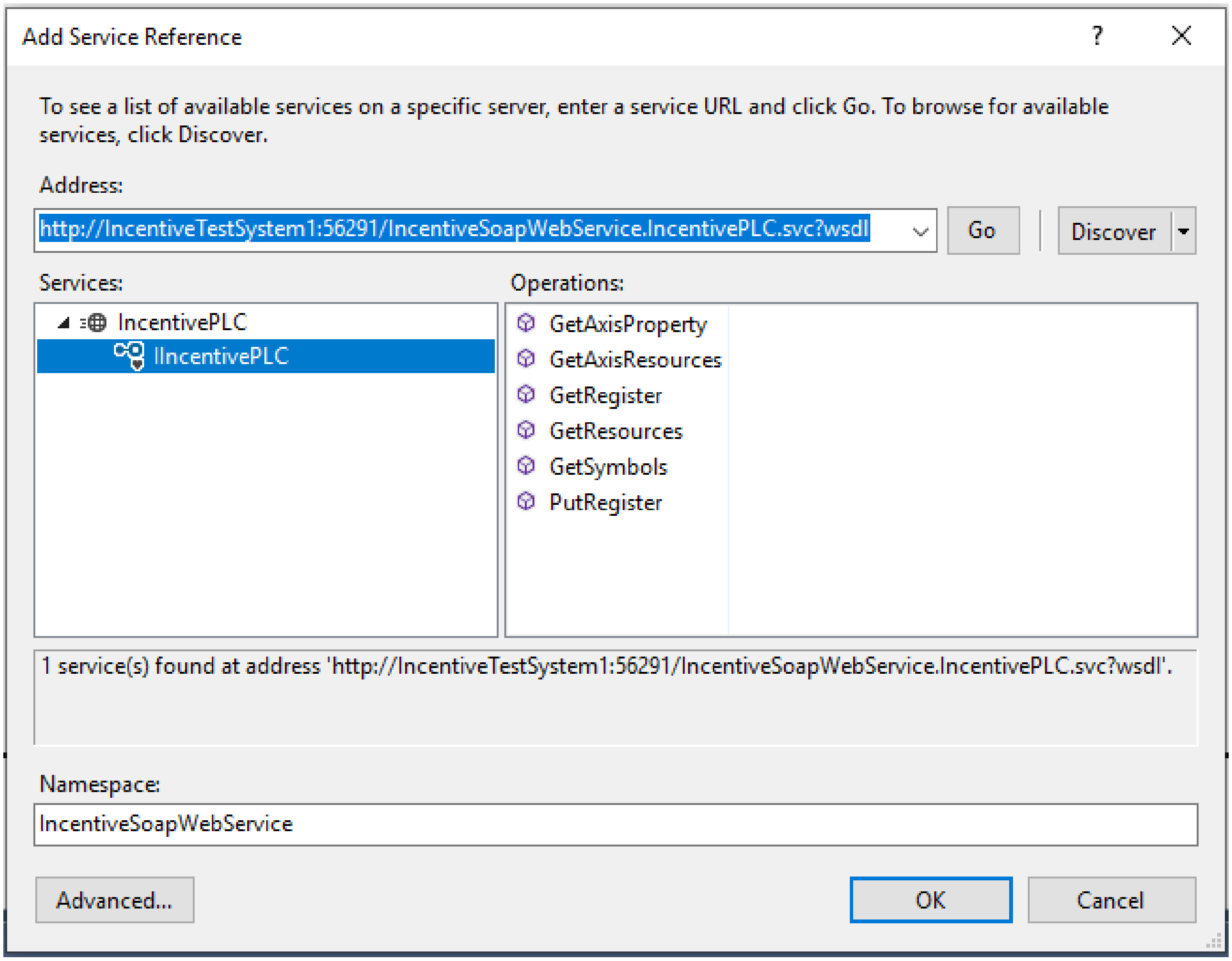


The project is ready for debugging. The default url used will be that invoked by wsimport. Reference IncentivePLC.java line 33.

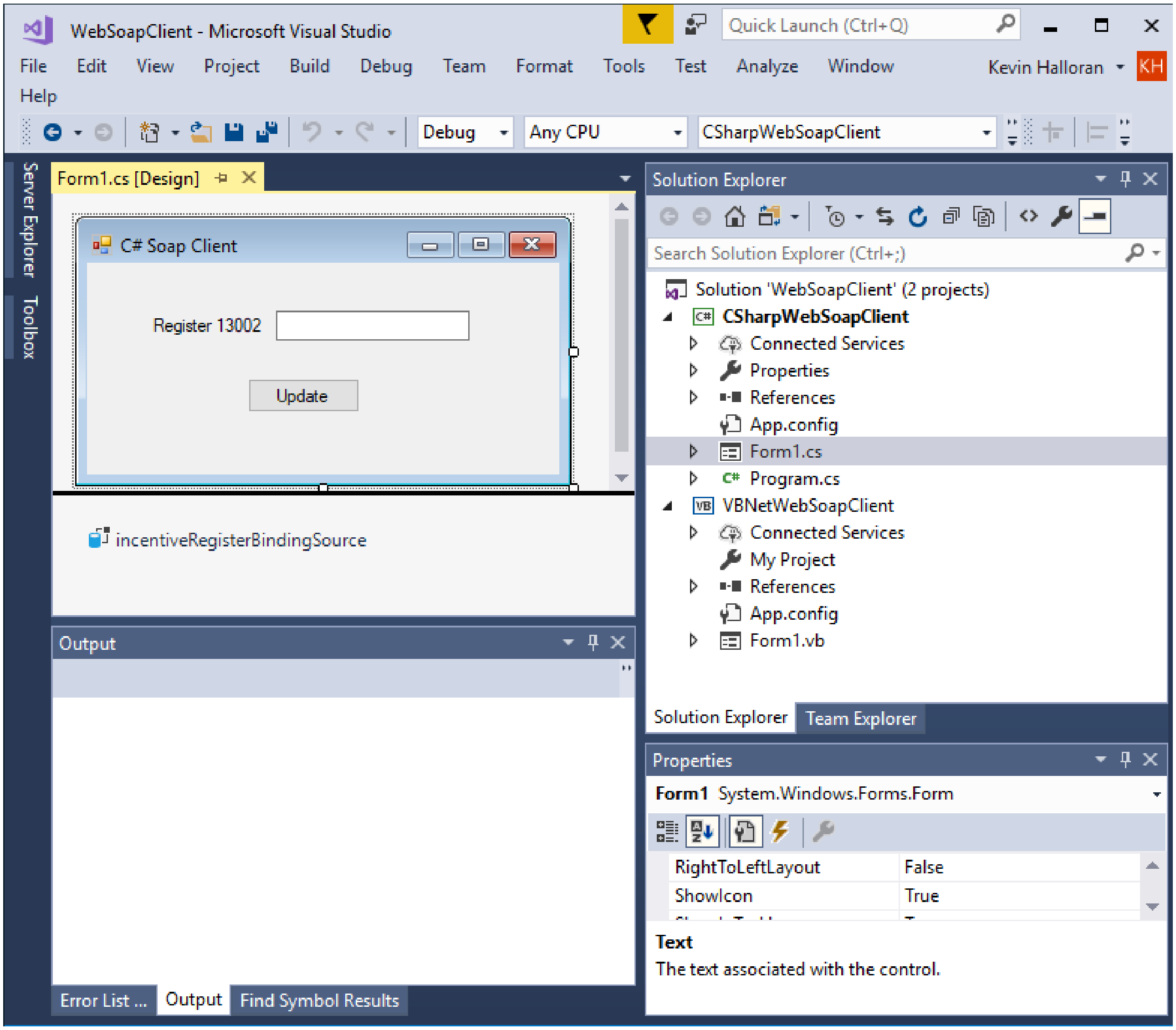
Simple C# and VB.Net Web Client Applications

Both a C# and VB.Net Web Client example projects are included with the Incentive installation within the WebServices subdirectory. They may be opened by using Visual Studio 2017, or greater, and double clicking on the WebSoapClient.sln Visual Studio Solution file within the ..\WebServices\WebSoapClient subdirectory. Both examples do the exact same thing, read the timer tick register 13002 using Soap. The C# example adds the option to build for SSL secure transport by uncommenting the define at the top of the

Form1.cs file. Creating the project is very simple. It mearly consists of creating a new project and then right clicking on Reference in Solution Explorer, followed by 'Add Service Reference'. The Web Service must be running on the Incentive system under IIS since its WSDL file will be accessed to create the appropriate service model classes. A form similar to below will appear where you will need to enter the desired URL for the service, clicking Go will allow you to browse the available services offered. Click Okay to create the class files.



The supplied project already has the service reference added and should appear similar to below when loaded into Visual Studio:



The code is very simple and is below. Note that optionally you can specify a different endpoint to override that defined in the App.config file. Different instances of the service with different endpoints can be used for multiple remote IncentivePC installations.

```
// Uncomment below if will be using SSL
// #define SSL  // If using SSL make sure to export SSL Certificate from IIS machine with private key
             // (pfx file with password) and import to Trusted Root Certification Authorities
             // (Local Computer) of Client.

using System;
using System.Threading.Tasks;
using System.Windows.Forms;
#if SSL
using System.Security.Cryptography.X509Certificates;
#endif
using CSharpWebSoapClient.IncentiveSoapWebService;
```

```
using System.ServiceModel;

namespace CSharpWebSoapClient
{
    public partial class Form1 : Form
    {
        CSharpWebSoapClient.IncentiveSoapWebService.IncentivePLCClient incentivePLC;
        public Form1()
        {
            InitializeComponent();
            incentivePLC = new CSharpWebSoapClient.IncentiveSoapWebService.IncentivePLCClient();
#if SSL
            // This is SSL soap transport
            this.Text = "C# Soap SSL Client";
            // Create the proper binding
            BasicHttpBinding binding = new BasicHttpBinding();
            binding.ReaderQuotas.MaxArrayLength = int.MaxValue;
            binding.MaxBufferSize = int.MaxValue;
            binding.MaxReceivedMessageSize = int.MaxValue;
            binding.MaxBufferPoolSize = int.MaxValue;
            binding.ReaderQuotas.MaxDepth = int.MaxValue;
            binding.ReaderQuotas.MaxStringContentLength = int.MaxValue;
            binding.ReaderQuotas.MaxBytesPerRead = int.MaxValue;
            binding.ReaderQuotas.MaxNameTableCharCount = int.MaxValue;

            binding.Security.Mode = BasicHttpSecurityMode.Transport;
            binding.TransferMode = TransferMode.Buffered;
            incentivePLC.Endpoint.Binding = binding;
            // Change the url below to that desired to use another system to access where
incentivetestsystem1
            // is the DNS host name and 56291 is the IIS port for the web service on the remote device, 56292
for SSL.
            incentivePLC.Endpoint.Address = new
System.ServiceModel.EndpointAddress("https://incentivetestsystem1:56292/IncentiveSoapWebService.IncentivePLC.
svc/soap");
            // We are using SSL so access https and set the certificate using that is in Trusted Root
Certification Authorities (Local Computer)
           incentivePLC.ClientCredentials.ClientCertificate.SetCertificate(StoreLocation.LocalMachine,
StoreName.Root, X509FindType.FindByIssuerName, "IncentiveTestSystem1");
#else
            // We are not using SSL so norma/ http request
            this.Text = "C# Soap Client";
            // Create the proper binding
            BasicHttpBinding binding = new BasicHttpBinding();
            binding.ReaderQuotas.MaxArrayLength = int.MaxValue;
            binding.MaxBufferSize = int.MaxValue;
            binding.MaxReceivedMessageSize = int.MaxValue;
            binding.MaxBufferPoolSize = int.MaxValue;
            binding.ReaderQuotas.MaxDepth = int.MaxValue;
            binding.ReaderQuotas.MaxStringContentLength = int.MaxValue;
            binding.ReaderQuotas.MaxBytesPerRead = int.MaxValue;
            binding.ReaderQuotas.MaxNameTableCharCount = int.MaxValue;

            binding.Security.Mode = BasicHttpSecurityMode.None;
            binding.TransferMode = TransferMode.Buffered;
            incentivePLC.Endpoint.Binding = binding;
            // Change the url below to that desired to use another system to access where
incentivetestsystem1
            // is the DNS host name and 56291 is the IIS port for the web service on the remote device, 56292
for SSL.
            incentivePLC.Endpoint.Address = new
System.ServiceModel.EndpointAddress("http://incentivetestsystem1:56291/IncentiveSoapWebService.IncentivePLC.s
vc/soap");
            incentivePLC.Open();

#endif
            incentivePLC.Endpoint.Binding.SendTimeout = TimeSpan.FromSeconds(5);  // set connection timeout
in seconds
       }
```

```
        private void button1_Click(object sender, EventArgs e)
        {
            try
            {
                // Read the desired register
                IncentiveRegister reg = incentivePLC.GetRegister("13002", "0", "0");
                textbox_RegValue.Text = reg.value;
                // Below is example of reading an axis property, in this example 'fpos'
                // or feedback position and limiting it to 4 decimal places
//                IncentiveAxis axis = incentivePLC.GetAxisProperty("1", "fpos.4");
//                textbox_RegValue.Text = axis.value;

            }
            catch (System.ServiceModel.EndpointNotFoundException)
            {
                MessageBox.Show("Endpoint is offline: " + incentivePLC.Endpoint.Address.ToString());
            }
            catch (System.TimeoutException)
            {
                MessageBox.Show("Endpoint is offline: " + incentivePLC.Endpoint.Address.ToString());
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.ToString());
            }
        }
    }
}
```

CHAPTER

5

[5] Drives & I/O Mapping

QuickBuilder assigns drive and I/O as it sees them, sequentially on the local backplane. EtherCAT is simply an extension of the backplane — as each device is seen, it is added to the local I/O and drive table. Thus the first device on the network has its I/O added to the end of the local 5300 rack I/O table, and sequentially for IncentivePC, as it is placed online. The EtherCAT Explorer, a QuickBuilder utility, can be used to observe how IO is assigned per device. EtherCAT devices are typically addressed by their sequence of daisy chained wiring. By default each EtherCAT device has an automatically assigned 'Configured Address'. The lower part of this address is masked with 0x00ff and the result becomes the default address. Axis numbers are assigned as they appear on the network. The problem with this approach is that the addresses can change as devices are moved in the daisy chain of wiring. This may be fine with I/O, but does not work for mapping MSBs to the drives they are controlling. Insertion of a new drive will shift all the following axis numbers up by one. To resolve this, if the 'Station Alias' is nonzero it will override the automatic numbering and be used as the Axis assignment. If 'Station Alias' is used on one drive then all drives should be use it to ensure there is no overlapping and re-use of the same axis. Should this not be desired then only the drives at the end of the EtherCAT network should have an alias assigned since those prior will have the axis numbers assigned sequencially, as identified.

I/O may be intermixed with drives, in any order.

The I/O will be ignored in the numbering of the axes. Using the Station Alias enables you to insert other devices and change wiring without worrying about your MSBs controlling the wrong drive. Most Station Aliases are set using the drive's dip switches. Due to a firmware anomaly, Kollmorgen devices typically must have their addresses set with the EtherCAT configurator and programmed to their individual EEPROMs.

Optionally, virtual drives can be defined, starting as the first or added after the online drives. These drives run in Cyclic Sync Position mode only and execute the same as online drives. In many cases a virtual drive will be configured as a master and an online drive will track to its position. The eCAT_driveType variable will contain a 9 for a Virtual axis.

The EtherCAT Master can only assign I/O and axis numbers based upon what devices respond on the network. Configuration files must be used in a production environment to ensure all the required devices are online prior to executing their controller MSBs. Differing devices power up at different times and may not initially respond to the

EtherCAT Master online broadcast. Having a configuration file to compare against informs the master that it must wait for devices to come online prior to proceeding with the boot operation. Reference the EtherCAT Explorer chapter for information on how to automatically create a configuration file.

Drives & 5300 M3-41A Local I/O

Some drives support inputs and outputs at the remote drive level. The MSB property 'dins' represents the raw inputs provided by the drive, up to 32 inputs (object 0x60FD.0). The first 10 inputs may be accessed using 'din1' to 'din10' bit properties; as with the M3-40 modules.

Outputs operate as they do on the M3-40 module, limited to 8 outputs at the remote drive level (object 0x60FE.1). Use the 'setout' and 'clrout' MSB instructions for access, where the first output is 1.

Local I/O is also present on the 5300 M3-41 EtherCAT module. This module has 6 inputs and 2 outputs which are global to all MSBs. The outputs are referenced as 9 and 10 when using the 'setout'/'clrout' instructions. The MSB property 'global_inputs' is used to read the 6 inputs, with the first bit being the first input. The MSB property 'global_outputs' can be used in addition to 'setout'/'clrout' for read/write operations of the local outputs.

Local global Inputs, P1 connector pins:

P1 -11 DIN1
P1-13 DIN2
P1-15 DIN3
P1 -12 DIN4
P1-14 DIN5
P1-16 DIN6

Local global Outputs, P1 connector pins:

P1 -9 DOUT1
P1-10 DOUT2

1. 'global_inputs' and 'global_outputs' axis properties may be accessed by QuickBuilder using the Axis name/property method: axisname.property. These two properties will contain the same value on all EtherCAT axes.

2. Chapter 6 discusses additional IO capabilities available from the MSB language using various IO arrays. These arrays give access not only to drive and module based IO but remote EtherCAT IO blocks such as those from Wago, Turck and Beckhoff. Some of the features include PLS, PWM, pulse, and atomic multi-bit access of 32 drive inputs/outputs, local and remote IO.

3. IncentiveECAT only supports drive I/O.

CHAPTER

6

[6] Drives: Modes of Operation & General Programming

CTC's programming environment allows you to place a drive in a number of different modes, each offering unique features. The MSB 'cmode' variable controls the active mode that a drive runs in when motion commands are executed. The variable can be set programmatically and/or its initial value set via the QuickBuilder axis property sheet. The following values are currently supported for programming entry:

```
$CYCLIC_SYNC_POSITION_MODE   0
$PROFILE_VELOCITY_MODE       1
$INTERPOLATED_POSITION_MODE   3
$PROFILE_POSITION_MODE       4
$PROFILE_TORQUE_MODE          5 (not supported, reference Chapter 4, Torque Control)
$HOMING_MODE                 8
$VELOCITY_MODE               9  (velocity only drives like MotionLinx-AI)
```

When setting 'cmode', either the constant predefined name can be used (starting with $) or the immediate numeric value. When using the constant make sure it is spelled correctly or it will be defined as a user variable with a default value of 0.

By default all drives are placed in Cyclic Sync Position mode at power up unless overridden by the QuickBuilder axis property sheet. This allows direct control of all motion. Initially the current position becomes the base position and is cyclically written to each drive as a holding position until commanded differently. The amplifier is only turned on once the 'drive enable' command is executed. This results in a special power-up sequence. The 'drive enable' command waits until the drive is powered up before it allows additional commands to be executed.

A typical simple move is displayed below:

```
cmode = $CYCLIC_SYNC_POSITION_MODE;  // CSP mode (default)
[top]
move at 1 for 2;
wait for in position;
delay 1000 ms;
move at 1 for -2;
wait for in position;
delay 1000 ms;
goto top;
```

In the above example, the drive will move at 1 rev/sec for 2 revolutions and then move back 2 revolutions. The move is a relative move, since it does not designate an actual position. The delay is arbitrary and for viewing purposes only.

To run in Profile Position mode, simply change cmode to a 4 or $PROFILE_POSITION_MODE and execute the same commands.

Cyclic Synchronous Position Mode (CSP)

Cyclic sync position mode provides linear interpolation, where the drive will always insert a delay of one position command. It gives the EtherCAT Master the greatest flexibility since it is directing the drive exactly where to go on each scan cycle (delayed one scan cycle by the drive). The trajectory is calculated on the fly. It also takes the most overhead, as each drive requires around 50µS for this calculation. CSP mode is the preferred mode for CTC EtherCAT products.

Example:

```
// This is a background MSB.  Make sure inposw is set for the drive,
// typically .001 for 1048576 ppr.  Also set the ppr and mppr.  For
// Yaskawa this is typically 1048576.  Enable the drive, turning power
// on to the amplifier.  The current position will be constantly
// updated so the drive does not move.

drive enable;
zero feedback position;

// CSP mode is the default but set anyway for documentation purposes
// and await any drive settling.
cmode = $CYCLIC_SYNC_POSITION_MODE;
wait for in position;

[top]
// Begin the move, 1 rev/second for 2 revolutions
move at 1 for 2;
wait for in position;
// Delay 1 second once in position
delay 1000 ms;
// Do a relative move back 2 revolutions at 1 rev/second
move at 1 for -2;
wait for in position;
// Delay 1 second once in position
delay 1000 ms;
// Do it again, forever...
goto top;
```

When moving out of a Profile mode and back into CSP mode, make sure the following is executed first:

```
cmode = $CYCLIC_SYNC_POSITION_MODE;
wait for in position;
```

Reference the DC Sync section of this manual for additional information when using CSP mode.

Profile Velocity Mode

In Profile Velocity mode, the speed is output in accordance with the Profile acceleration and Profile deceleration. The drive attempts to maintain the velocity commanded. CSP mode can also be used for constant velocity, but when using Profile Mode, control loop time is optimized since the EtherCAT Master does not have to constantly calculate the trajectory for that drive. The following is a simple example of Profile Velocity Mode:

```
vmax = 100;         // Max velocity.
stoprate = 1;       // This is the rate (rev/sec) used on Yaskawa and
                    // Copley drives for decel when STOP.
invel_t = 1;        // Time required, in milliseconds to be at target
                    // velocity before considered AT TARGET.
invel_w = .01;      // Must be at target velocity +/- (.01 X target vel)
                    // with drive AT TARGET to satisfy move.
                    // If target is 0 then +/- .01 rev/sec

[top]

cmode = $PROFILE_VELOCITY_MODE;      // Profile Velocity mode

// When in velocity mode distance is just sign of direction.
move at 2 for 1;
wait for in position;   // This is when attain requested velocity
delay 5000 ms;

// Now speed up to 20 rev/sec in the same direction.
move at 20 for 1;
wait for in position;
delay 1000 ms;

// If the motor is not tuned may never get to here but
// this is the proper way to stop in velocity mode.
move at 0 for 1;
wait for in position;

// Ensure stopped before changing to another mode, like CSP.
stop;
wait for in position;
goto top;
```

The EtherCAT Master uses the properties ‘invel_t’ (at velocity time) and ‘invel_w’ (at velocity window) to monitor when the drive is actually AT TARGET in Profile Velocity mode.

Interpolated Position Mode

Interpolated position mode is used to control multiple coordinated axes or a single axis with the need for time-interpolation of setpoint data. In interpolated position mode, the trajectory is calculated by the EtherCAT Master and passed to the amplifier's interpolated position buffer as a set of points. The amplifier reads the points from the buffer and performs linear or cubic interpolation between them. This mode is provided for compatibility with some drives with the preferred mode being CSP. The following is a simple example of Interpolated Position Mode:

```
// This is a background MSB.  Make sure inposw is set for the drive,
// typically .001 for 1048576 ppr. Also set the ppr and mppr.  For
// Yaskawa this is typically 1048576.

// Enable the drive, turning power on to the amplifier.  The current
// position will be constantly updated so the drive does not move.

drive enable;
zero feedback position;

// Make sure we are stopped and are in CSP mode
cmode = $CYCLIC_SYNC_POSITION_MODE;
wait for in position;

// Drop into Interpolated Position Mode
cmode = $INTERPOLATED_POSITION_MODE;
[top]
// Begin the move, 1 rev/second for 2 revolutions
move at 1 for 2;
wait for in position;
// Delay 1 second once in position
delay 1000 ms;
// Do a relative move back 2 revolutions at 1 rev/second
move at 1 for -2;
wait for in position;
// Delay 1 second once in position
delay 1000 ms;
// Do it again, forever...
goto top;
```

Profile Position Mode

In Profile Position mode the drive is given a velocity, and acceleration, and final position to move to and it calculates the trajectory. There is lower overhead on the EtherCAT Master when using Profile Position Mode than for Cyclic Sync Position, since no trajectory needs to be calculated but it does impact the cyclic nature of other drives running in CSP mode given the initial SDO transmissions for profile setup. The following is a simple example of Profile Position Mode:

```
// Place the drive in Profile Position mode (Note Profile Position is
// not supported by Kollmorgen).
cmode = $PROFILE_POSITION_MODE;
[top]
```

```
// Set Profile Velocity (0x6081) to 5 rev/sec, acc (0x6083), and dec
// (0x6084).  Request move of 2 revolutions
move at 5 for 2;
// Wait until drive says we are in position
wait for in position;
delay 1000 ms;     // Delay for visual
// Set Profile Velocity (0x6081) to 5 rev/sec, acc (0x6083), and dec
// (0x6084).  Request move of -2 revolutions, relatively back to 0.
move at 5 for -2;
// Wait until drive says we are in position
wait for in position;
delay 1000 ms;     // Delay for visual

goto top;          // Continue forever
```

The properties of inpos_w variable (in position window) and inpos_t variable (in position time) can be set prior to initiating the command. ‘inpos_t’ maps to object 0x6068 and ‘inpos_w’ maps directly to the drive property ‘inposw’. ‘inpos_t’ is used in Profile Position mode to set the length of time a drive must be in the commanded position before it is determined to be ‘in position’ (inpos variable).

Homing Mode

In Homing mode the drive finds the home position based on a supplied motion profile and the designated homing method (0x6098 object). Most homing methods are generic but not all are supported by every drive. Consult the drive manufacturer’s manual to determine your drive’s homing methods.

Before requesting a Homing Move, set the following parameters:

- ***inpos_t*** – The number of milliseconds the drive should settle after finding home before the move is considered complete.
- ***homing_speed1*** – Some homing modes require multiple speeds, with this one being the speed to the switch.
- ***homing_speed2*** – Some homing modes require multiple speeds, with this one being the speed to zero or the index pulse.
- ***homing_method*** – The method that the drive manufacturer designates for the desired move. For example, 33 homes to the index pulse, while 34 does the same thing but in the opposite direction. There are numerous modes available, all controlled by the drive itself. The MSB is simply setting the move up for the drive to take control.

Once the above parameters are defined, ‘cmode’ is set to $HOMING_MODE (8) to command Homing mode and a move absolute command to 0 is initiated. The acceleration value in the move command will be used for the homing acceleration and deceleration rate. The move will not start until the ‘move’ command is executed. For example: ‘**move to** 0 **using** 10000,10000;’ initiates the drive homing move with an acceleration/deceleration profile of 10000 rev/sec^2. This is the value written to object 0x609A, Homing Acceleration. Note that some drives, such as Mitsubishi call this an acceleration/deceleration constant where upon the raw value is written. For example for a time constant of 0 on Mitsubishi the command would be: ‘**move to** 0 **using** 0, 0;’. Check the manufacturers manual for the meaning of object 0x609a. Note that Mitsubishi currently does not support the object but may in the future.

The following is a simple example of Homing Mode:

```
drive enable;
zero feedback position;

// Let the drive home to the index pulse
[home]
inpos_t = 250;      // 250 millisecond settling (can make it anything want
                    // but this is the time the drive will wait at position
                    // before notifying QuickBuilder MSB that it is home.
homing_method = 34;     // Yaskawa method setting for home to index pulse
homing_speed1 = 1; // homing_speed1 is not used but set for default anyways
homing_speed2 = 1; // in mode 34 only the homing_speed2 is used
cmode = $HOMING_MODE;         // Homing mode for drive
move to 0 using 10000,10000;  // Tell the drive to initiate the move with
                              // accel of 10000.
wait for in position;     // The drive will stop once it sees the index
                          // pulse and tpos = fpos at that position
                          // therefore we will have an offset past home in
                          // tpos/fpos, not really absolute 0.

// Using CSP mode we can move back to absolute home position or 'zero
// feedback' to 0 out tpos/fpos.

// Drop back to CSP mode so we command the drive
cmode = $CYCLIC_SYNC_POSITION_MODE;
move at .1 to 0 using 10000,10000;  // Do any kind of absolute move back
                                    // to 0 to remove our offset.
wait for in position;               // Wait until move is complete

// We are now at home, 0.
```

The properties of inpos_w variable (in position window) and inpos_t variable (in position time) can be set prior to initiating the command. 'inpos_t' maps to object 0x6068 and 'inpos_w' maps directly to the drive property 'inposw'. 'inpos_t' is used in Homing mode to set the settling time a drive must be in the commanded position before it is determined to be 'in position' (inpos variable).

Should you need to abort a Homing, once it has begun, a '**`stop`**' command may be issued followed by a '**`wait for in position`**'. After the '**`wait for in position`**' you may change the mode back to CSP if desired or begin another Home procedure.

Velocity Mode

In Velocity mode, the speed is output in accordance with the acceleration and deceleration. The drive attempts to maintain the velocity commanded. The drive itself generates the proper profiles based upon the move instructions used. The velocity is positive and negative to set direction and the position parameter is ignored and can be 0. Velocity can be changed on the fly by executing a new move instruction once inpos is true (at velocity) or pstate is and if the acc/dec is not specified by the move instruction the default acc/dec variable value will be used.

```
acc = 50;
```

```
dec = 10;
cmode = $VELOCITY_MODE;  // Set type of motion control to do
ever = 0;  // Assign 0 to a variable to make code more readable
[top]
move at 200 for ever;   // Move in the positive direction
// Wait until drive is at velocity
wait for in position;
delay 5000 ms;    // Let it run a bit
move at -200 for ever;  // Run the opposite direction
// Wait until drive says we are in position
wait for in position;
delay 5000 ms;    // Let it run a bit
stop;                   // Issue the stop command
wait for in position;   // Wait for drive to stop moving
delay 1000 ms;    // Pause at rest for a bit

goto top;         // Continue forever
```

Presently only the MOTIONLINX conveyor controller is supported in Velocity Mode.

Gearing/Tracking & Local Quadrature Encoders

Gearing and tracking modes work the same as with the 5300 M3-40 servo module, with a few enhancements to allow for the larger number of axes.

Any axis can track another axis by simply dropping into tracking mode. To reference which axis to track, use the variable 'master_feedback', which by default is 1. Set this variable to the axis you wish to track, set whether to reference fpos or tpos of the master axis, and then drop into tracking mode. The following is a simple example of Tracking Mode:

```
cmode = $CYCLIC_SYNC_POSITION_MODE;
// Establish which axis we are for, axisnum is our axis number
// and is a new available property.
AxisNum = axisnum;

// ******** TRACKING MODE **********
[SetTracking]
zero master counters;
master_feedback = 1;  // Set we will track axis 1
// Set the feedback mode first so when enter tracking it is referencing
// correct master.  'set master feedback' references the other axis
// fposc or feedback position.  'set master target1' references the other
// axis tpos position.
set master feedback;
set mode tracking;          // Enter tracking mode
gear at 1:4;

[stall]
goto stall;
```

As described in the example there are two major ways to track the master axis:

set master feedback – tracks fpos (encoder feedback) of the master.

set master target1 – tracks tpos (target position) of the master.

With an analog servo fpos (encoder feedback position) is typically used. When using EtherCAT tpos (target position) may be more accurate, depending on the drive's lag. Lag is the time it takes for the drive to take a commanded position and make it current. Different drives have different averaging or smoothing algorithms as they calculate their next profile. For example 'fpos' will also lag more than 'tpos' since it will take the drive time to catch up to the commanded position. The benefit of tracking 'fpos' is when the master is in a high torque situation, with a larger position error, the slave will track the actual position rather than the desired position (tpos). Which to use will be application and drive dependent, with the preference towards 'fpos'.

Another option, only available on the M3-41 hardware module, is the provision for directly connecting up to 3 local quadrature encoders. These encoders can be used as master references by the EtherCAT axis. To reference these encoders, the master_feedback variable is set to 1001, 1002, or 1003, for each of the respective encoder inputs. Once master_feedback is referencing a local encoder, its present value will appear in 'mpos' and 'mposc' MSB variables. In addition, all local encoder counts can be accessed using the 'ctr' array, index 5 to 7 from an MSB, or ctr5, ctr6, and ctr7 from QuickBuilder.

Master_feedback = 1001, ctr[5]/ctr5, P1 connector pins:

P1-19 A0+
P1-20 A0-
P1-21 B0-
P1-22 B0+

Master_feedback = 1002, ctr[6]/ctr6, P1 connector pins:

P1-23 A1+
P1-24 A1-
P1-25 B1-
P1-26 B1+

Master_feedback = 1003, ctr[7]/ctr7, P1 connector pins:

P1-27 A2+
P1-28 A2-
P1-29 B2-
P1-30 B2+

Power and ground for the encoder can be sourced from the +5V/GND connector on the model 5300 power supply. Additionally, when using 'set master target1', it is best to set the lower numbered axis as the master otherwise the position loop update will be delayed by one control loop cycle on the slave device. In most instances this delay is irrelevant.

The Incentive PC could use EtherCAT based incremental encoders such as the Wago 750-631/637.

Tracking operation can be modified with two variables, perrlimit and vcmd. During tracking operations the amount the slave axis master position has moved, each servo cycle, will be multiplied by 'vcmd'. By default this is 1.0 but may be modified to any positive value. A second variable 'perrlimit' can be used to dump the error when it reaches the value set in 'perrlimit'. This is useful in winding operations. Setting 'perrlimit' to 0 disables this feature and is the default.

Move on a Gear

One of the latest features introduced is the ability to be geared to a master axis and then do a 'move' command (CSP mode only) which will transpose that move commands profile on top of that tracking the master axis. This is useful in a number of applications, particularly where the slave may be tracking with a master but needs to catch up to that axis and maintain gearing.

Specific property variables:

tracking_pstate – Read only, contains the current execution state of the drive during a 'move on a gear' operation. '*pstate'* must be in TRACKING mode for this property to be valid.

tracking_tpos – Read only, same as '*tpos'* property but used only for 'move on a gear' operation. It will depict the relative position reference with an initial position of 0. This is then added to the tracked position as calculated from the master position.

tracking_tposc = Read only, number of counts required for the 'move on a gear' move for this increment of its profile.

tracking_sign – Read only, 0 if not used, 1 for positive rotation, -1 for negative rotation when adding '*tracking_tposc*' to '*tposc'* to derive resulting motion.

tracking_status – Read only, contains the current state of the move as referenced by the 'QS2_Status' property.

```
NOT_INITIALIZED           = 0
STOPPED_READY             = 1
ACCELERATING              = 3
AT_MAX_SPEED              = 4
DECEL_TO_NEW_MAX_SPEED    = 5
DECEL_TO_STOP             = 6
```

The below example is two axes, where the second axis is the master:

MSB axis 2, Master Axis (can be virtual axis):

```
/**************** ENABLE DRIVE *******************/
delay 1000 ms;
 [Drive_Enable]
drive enable;              // Enable the drive
set common bit 1 true;     // Tell slave drive is enabled and it can gear
wait common bit 0 true;    // Wait for slave to be geared
```

```
[loop]

move at 10 for 1000000;        // make a long move
wait for in position;
delay 1000;

goto loop;  // repeat
```

MSB axis 1, Slave Axis (start this axis first):

```
/**************** ENABLE DRIVE ******************/
// Clear common bits used to handshake with the master
set common bit 0 false;
set common bit 1 false;

[Drive_Enable]
drive enable;
wait common bit 1 true;        // Wait for master to tell us its drive is enabled
zero master counters;
master_feedback = 2;           // Reference axis 2 for fposc so we can track
set master feedback;
set mode tracking;             // Enable tracking mode

// Set our gear ratio
gear at 1:1;

set common bit 0 true;         // tell master we are all geared and ready to go

[reverse]
delay 3000 ms;                 // Track 1:1 with master at speed of master
move at 5 for 50;              // Simulate catching up to Master.
                               // If Master at 10 rev/s then doing 15 rev/sec
                               // during move, 'wait for in position' not usable.
[stall]
if tracking_pstate != 2 goto stall; // wait for move to COMPLETE
goto reverse;
```

Only 'move' commands can be executed while in TRACKING mode. Slew and camming are not supported by the slave axis.

Axis property 'pstate' and 'tracking_pstate', where 'pstate' is equal to TRACKING mode for 'move on a gear' to occur, both can have the following values:

```
IDLE = 0,           // Ready to run new move command
RUNNING = 1,        // Processing sub-steps
COMPLETE = 2,       // Done running, awaiting IDLE to begin new move
STOP = 3,           // Stop
SLEWSTOP = 4,       // Slewed stop
SLEWING = 5,        // Slewing
PRESPLINE = 6,      // Pre 'SPLINE' move
PRECAM = 7,         // Pre 'CAM' move
PRECAM_REVERSE=8,   // Run cam table in reverse
```

```
CONT_CAM=9,          // Continue cam from where stopped (assumes did jog)
INSPLINE=10,         // In 'SPLINE' move
INCAM=11,            // In 'CAM' move
TABLESTOP=12,        // Stop table
TRACKING=13,         // Geared mode, state of ‘pstate’ when
                     //     ‘tracking_pstate’ valid
PRETRACKING=14,      // Initialization for TRACKING (geared) mode
EXIT_TRACKING=15,
ECAT_COMPLETE_PENDING=16,
ECAT_PROFILE_POS_INIT=17,
ECAT_PROFILE_POS_STARTING1=18,
ECAT_PROFILE_POS_STARTING1A=19,
ECAT_PROFILE_POS_STARTING2=20,
ECAT_PROFILE_POS_RUNNING=21,
ECAT_PROFILE_POS_WAIT_INPOS=22,
ECAT_PROFILE_VEL_INIT=23,
ECAT_PROFILE_VEL_WAIT_DELAY1=24,
ECAT_PROFILE_VEL_WAIT_DELAY2=25,
ECAT_PROFILE_VEL_WAIT_DELAY3=26,
ECAT_PROFILE_VEL_WAIT=27,
ECAT_PROFILE_TORQUE_INIT=28,
ECAT_PROFILE_INIT_CSP=29,
ECAT_PROFILE_INIT_INTERPOLATED=30,
ECAT_PROFILE_WAIT_CSP1=31,
ECAT_PROFILE_WAIT_CSP2=32,
ECAT_MODE_WAIT_CSP=33,
ECAT_MODE_WAIT_INTERPOLATED=34,
ECAT_MODE_WAIT_PROFILE=35, //Idling, awaiting Profile request to be processed.
ECAT_PROFILE_INIT_QSTOP=36,
ECAT_PROFILE_WAIT_QSTOP1=37,
ECAT_PROFILE_WAIT_QSTOP2=38,
ECAT_PROFILE_WAIT_QSTOP=39,
ECAT_PROFILE_AT_VEL=40,
ECAT_PROFILE_AT_TORQUE=41,
ECAT_HOMING_INIT=42,
ECAT_HOMING_STARTING1=43,
ECAT_HOMING_STARTING1A=44,
ECAT_HOMING_STARTING2=45,
ECAT_HOMING_RUNNING=46,
ECAT_HOMING_WAIT_INPOS_KOLLMORGEN=47,
ECAT_HOMING_WAIT_INPOS=48,
ECAT_HOMING_WAIT_INPOS_IAI=49,
ECAT_PROFILE_POS_WAIT_ABORTING=50,
ECAT_PROFILE_VEL_STOP=51,
ECAT_PROFILE_VEL_STOPPING=52,

ECAT_OFFLINE = -1,
ECAT_USER_OFFLINE=-2,
```

Camming Moves and Optional Timeouts

Camming Moves are supported in Cyclic Sync Position and Interpolated Position modes. You are allowed up to 2000 rows (master/slave entries) per table, with up to 6 tables available (numbered 0 to 5). Consult the *QuickMotion Reference Guide* for more detailed information. An MSB example is as below:

```
drive enable;
zero feedback position;
cmode = $CYCLIC_SYNC_POSITION_MODE;

// This MSB effectively sets up a 1:1 gear ratio with the Master
// The first item in the table is master revolutions/second.
// The second entry is that of the slave, this MSB.
table 1 clear;           // clear out the old data

table 1 addseries        // load new data to table 1
0.000 ,     0.0000:      //set up a 1:1 ratio
1.000 ,     1.0000:
2.000 ,     3.000 :
3.000 ,     5.0000:
5.000 ,     0.0000;      // CAMS wrap back to zero, like a mechanical cam

table 1 precompute;      // compute the cam (about 1/4 sec per 1000 pts)
// The master can reference another drive with the 'set master feedback1'
// command and setting the master_feedback variable to the drive desired.
// It can also reference its own created master, as shown below.
// This master is virtual and will increment by 1000 counts every
// Control Loop tick (1 ms).  This way only one drive is needed for
// testing camming.
set master virtual;            // We will use our own virtual master
// Set up the number of counts to increment per control loop.
move master at 1000 forever;
// This value will be divided by ppr for actual master revs.
zero feedback position;
[top]
// With the table repeat count set to 0 it will continually recycle
// through the table, forever, until a 'stop table' command is executed.
// This one will do 6 test cycles, repeating the cam table.
table 1 start linear cam 1.0, 1.0, 6;
wait for in position;    // Wait for CAM table to be done and exit

[loop]
activeCAM_row = 0;       // This contains where the table left off when it
                         // exited and must be zero'd as it
                         // is where it will start next time through.
delay 3000;              // Stall for now...
goto loop;
```

Below, a small section of the code is re-written to show how timeouts can be used in case a move does not occur becauseof an error condition. Timeouts can be used anywhere, not just in camming.

```
[top]
mytime = 10000;          // About 10 seconds
set timeout mytime;
```

```
on timeout goto timedout;  // We will expect the CAM move to finish in 10
                           // seconds or abort.
// With the table repeat count set to 0 it will continually recycle
// through the table, forever, until a 'stop table' command is executed.
// This one will do 2 test cycles, repeating the cam table.
table 1 start linear cam 1.0, 1.0, 2;
wait for in position;   // Wait for CAM table to be done or timeout
set timeout 0;          // Cancel the timer since we finished the move

[loop]
activeCAM_row = 0;      // This contains where the table left off when it
                        // exited and must be zero'd as it
                        // is where it will start next time through.
delay 3000;             // Stall for now...
goto loop;

[timedout]
// Timed out on camming table move
stop table;         // Stop the camming
[stall]
i = i+1;            // Increment i so know timeout worked from QuickView
goto stall;
```

Segmented Moves

Segmented Moves are supported in Cyclic Sync Position and Interpolated Position modes. You are allowed up to 20 segments. As shown in the example below, first clear the segment table, then add each segment, followed by a 'start' command. An MSB example:

```
// Enable the drive and clear our position
drive enable;
zero feedback position;
cmode = $CYCLIC_SYNC_POSITION_MODE;

// Initialize the move variables
vel1=5;
vel2=11;
rate1=50;
rate2=5;
dist1=10;
dist2=60;

stop_dist=1;

// Start the move loop
[top]
segmove 1 clear;        // Clear any prior segments stored

// Add each of the desired segments for the move desired
segmove 1 accdec to vel1 using rate1;
segmove 1 slew until dist1;
segmove 1 accdec to vel2 using rate2;
segmove 1 slew until dist2;
segmove 1 accdec to 0 for stop_dist;
```

```
// Start the move now
segmove 1 start relative;
// Wait for it to run the segments
wait for in position;
// Delay a bit and then do it again
delay 1000 ms;
goto top;
```

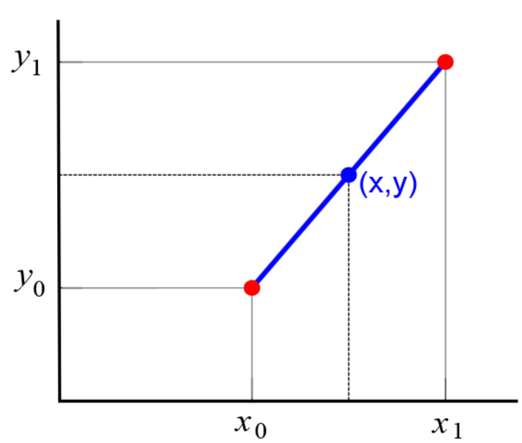
Slewed Move

A slewed move can be used for jogging as well as other desired motions. It is basically a move at some velocity that is reached in the time specified. An example of a slewed move:

```
drive enable;
zero feedback position;

cmode = $CYCLIC_SYNC_POSITION_MODE;
Speed = .5;               // Set our desired slew speed

[Slew]
slew begin;               // Enter slewing mode
slew at Speed in 1; // Slew to speed in 1 second
delay 5000 ms;            // Maintain speed for 5 seconds
slew at 0 in 0.1; // Slew to a stop in .1 seconds
slew end;                 // Drop out of slewing mode
delay 1000 ms;            // Pause for 1 second
Speed = Speed * -1;       // Change directions and slew the other way

goto Slew;
```

Linear Interpolation (2D)

Two dimensional (2D) linear interpolation allows any two drives to arrive at the same point, at the same time. This is referred to as the target X, Y position, when programming with MSB's, with the move from the present position to the target being interpolated.



The controlling axis is the X axis MSB while the axis that will be controlled in unison is the Y axis. A number of properties are available for an interpolated move:

axisY – The axis number, from 1 to N, which will be the Y axis, commanded from the X axis. The Y axis must be set for interpolation to occur.
vectorY – The desired Y position on an X/Y grid in user units, based upon revolutions.
Note that this value is overwritten after a circular interpolated move for diagnostic purposes.

Read only variables for viewing in the debug watch window:

angle – Read only, calculated angle of the last vector move.
magnitude – Read only, calculated size of the last vector move.
velX – Read only, calculated velocity along the X axis of the last vector move.
velY – Read only, calculated velocity along the Y axis of the last vector move.
accX – Read only, calculated acceleration along the X axis of the last vector move.
accY – Read only, calculated acceleration along the Y axis of the last vector move.
decX – Read only, calculated deceleration along the X axis of the last vector move.
decY – Read only, calculated deceleration along the Y axis of the last vector move.

Any of the normal 'move' commands can be used. The velocity and acceleration parameters are that of the vector while the position information is for the X axis position. The Y axis MSB should not attempt to control the axis while the X axis has it in motion or an error will result. Common bits can be used to synchronize the tasks if needed. Additionally, both drives must be in CSP mode.

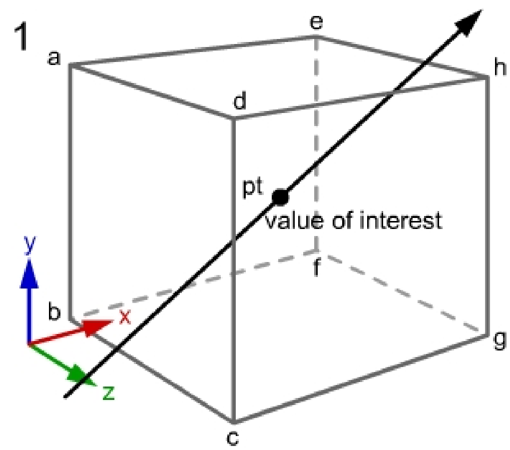
Sample Program:

```
[beginTest]
axisY = 2;          // set the Y axis
delay 1000 ms;      // make sure the axis is running first
speed = 40;         // Set the vector velocity in rev/sec

[interpolate]
// For simplicity user units are 1:1 with revolutions (uun/uud)
cmode = $CYCLIC_SYNC_POSITION_MODE; // CSP mode
vectorY = 10;       // Y position of 10 revolutions
// Below will make a 45 degree angle, 100 is the vector accel/decel
move at speed for 10 using 100,100;  // vector move where 'for' is the
                                     // X position.
wait for in position;    // Wait for move to finish on both X & Y axis

// move along the X Axis only
vectorY = 0;
move at speed for -10 using 100, 100;
wait for in position;

// Move along the Y Axis only, should be back to start after the move.
vectorY = -10;
move at speed for 0 using 100, 100;
wait for in position;

// should be back home now, do again
goto interpolate;
```

Splines may be used to with multiple X/Y vectors to provide smooth motion through multiple 2D/3D positions residing in a table. Linear, cubic, and quadratic splines are supported.

Linear Interpolation (3D)

Three dimensional (3D) linear interpolation allows any three drives to arrive at the same point, at the same time. This is referred to as the target X, Y, and Z position, when programming with MSB's, with the move from the present position to the target being interpolated.



As with 2D linear interpolation, the controlling axis is the X axis MSB while the axes that will be controlled in unison are the Y and Z axis. A number of properties are available for an interpolated move:

axisY – The axis number, from 1 to N, which will be the Y axis, commanded from the X axis. The Y axis must be set for either 2D or 3D interpolation to occur.
vectorY – The desired Y position on an X/Y/Z grid in user units, based upon revolutions. Note that this value is overwritten after a circular interpolated move for diagnostic purposes.
axisZ – The axis number, from 1 to N, which will be the Z axis, commanded from the X axis.
vectorZ – The desired Z position on an X/Y/Z grid in user units, based upon revolutions. Note that this value is overwritten after a circular interpolated move for diagnostic purposes.

Read only variables for viewing in the debug watch window:

angle – Read only, calculated angle of the last vector move.
magnitude – Read only, calculated size of the last vector move.
velX – Read only, calculated velocity along the X axis of the last vector move.
velY – Read only, calculated velocity along the Y axis of the last vector move.
velZ – Read only, calculated velocity along the Y axis of the last vector move.
accX – Read only, calculated acceleration along the X axis of the last vector move.
accY – Read only, calculated acceleration along the Y axis of the last vector move.
accZ – Read only, calculated acceleration along the Z axis of the last vector move.
decX – Read only, calculated deceleration along the X axis of the last vector move.

decY – Read only, calculated deceleration along the Y axis of the last vector move.
decZ – Read only, calculated deceleration along the Z axis of the last vector move.

Any of the normal 'move' commands can be used. The velocity and acceleration parameters are that of the vector while the position information is for the X axis position. The Y/Z axis MSB's should not attempt to control their axis while the X axis has it in motion or an error will result. Common bits can be used to synchronize the tasks if needed. Additionally, all 3 drives must be in CSP mode.

Sample Program:

```
[beginTest]
axisY = 2;          // set the Y axis, required for interpolation
axisZ = 3;          // set the Z axis
delay 1000 ms;      // make sure the axis is running first
speed = 40;         // Set the vector velocity in rev/sec

[interpolate]
// For simplicity user units are 1:1 with revolutions (uun/uud)
cmode = $CYCLIC_SYNC_POSITION_MODE; // CSP mode
vectorY = 10;       // Y position of 10 revolutions
vectorZ = 20;       // Z position of 20 revolutions
// Below will make a 45 degree X/Y angle, 100 is the vector accel/decel
move at speed for 10 using 100,100;  // vector move where 'for' is the
                                     // X position.
wait for in position;   // Wait for move to finish on X, Y & Z axis

// move along the X Axis only
vectorY = 0;
vectorZ = 0;
move at speed for -10 using 100, 100;
wait for in position;

// Move along the Y Axis only, should be back to start after the move.
vectorY = -10;
vectorZ = 0;
move at speed for 0 using 100, 100;
wait for in position;

// Move along the Z Axis only, should be back to start after the move.
vectorY = 0;
vectorZ = -20;
move at speed for 0 using 100, 100;
wait for in position;

// should be back home now, do again
goto interpolate;
```

When running 3D linear interpolation all the vector parameters are based upon the 2D X/Y axis calculations. The Z axis will attempt the same type of move based upon the required distance and calculated time to target, ignoring the vector motion parameters. Since both X and Y axis arrives at the target at the same time, moving the Z axis in this way ensures it arrives simultaneously.

Circular Interpolation (2D)

Two dimensional (2D) circular interpolation allows any two drives to work in coordinated motion to draw an arc or full circle. This functions similarly to that of 2D linear interpolation but with the addition of a few parameters to define the arc.

The controlling axis is the X axis MSB while the axis that will be controlled in unison is the Y axis. A number of properties are available for an interpolated move:

axisY – The axis number, from 1 to N, which will be the Y axis, commanded from the X axis. The Y axis must be set for interpolation to occur.
radius – The radius in user units of the arc to be drawn. A negative radius flips the arc.
angleSweep – The desired amount of angular motion that is to occur relative to the radius center point. A positive angleSweep rotates clockwise, negative, counter clockwise.
angleStart – The angle at which motion should start where 0 is vertical on the Y axis, minus angle moves left, and positive angle moves right. The 'angleSweep' variable is added to this angle.
vectorZ – The calculated center of the arc for the X axis will be stored here for diagnostic reference, in machine units.
vectorY – The calculated center of the arc for the Y axis will be stored here for diagnostic reference, in machine units. Make sure you update vectorY after a circular move if the next move is linear interpolation.

Read only variables for viewing in the debug watch window:

angle – Read only, initialized to 0 and records the calculated angle as it sweeps.
velVector – Read only, velocity in radians/second that is being used for the calculated profile.
accVector – Read only, acceleration in radians/second^2 that is being used for the calculated profile.
decVector – Read only, deceleration in radians/second^2 that is being used for the calculated profile.

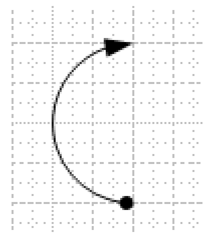
Any of the normal 'move' commands can be used. The velocity and acceleration parameters are in user units and represent the feed rate at a point on the arc. These parameters are then converted into radian units. The Y axis MSB should not attempt to control the axis while the X axis has it in motion or an error will result. Common bits can be used to synchronize the tasks if needed. Additionally, both drives must be in CSP mode.

Below are two sets of arcs drawn, one specifying the feed rate and the second time based, with all the needed velocity, acceleration, and deceleration dynamically calculated:

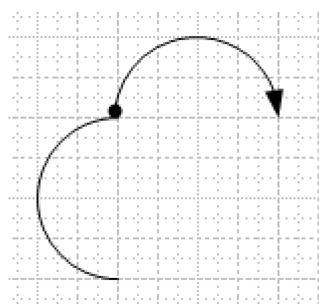
Example absolute moves:

```
[beginTest]

axisY = 2;  // set that we will coordinate with axis 2

[top]

// CW rotation
radius = .5;            // 1/2 inch radius, 5 rev/inch., uud = 5.
angleSweep = 180; // 180 degree sweep
```

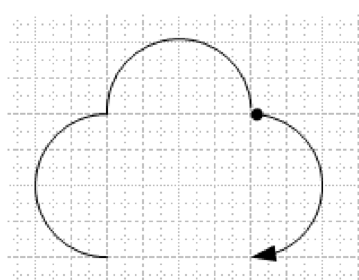
```
angleStart = -90; // negative 90 degree offset on Y axis
// move at 1.5 inches/sec feed rate with accel/decel of 4 inches/sec2
// velocity / radius = rad/sec
move at 1.5 to 0 using 4,4;
wait for in position;
```



```
// CW rotation
radius = .5;            // 1/2 inch radius.
angleSweep = 180;
angleStart = 0;         // no offset on Y axis
// move at 1.5 inches/sec feed rate with accel/decel of 4 inches/sec2
// velocity / radius = rad/sec
move at 1.5 to 0 using 4,4;
wait for in position;
```

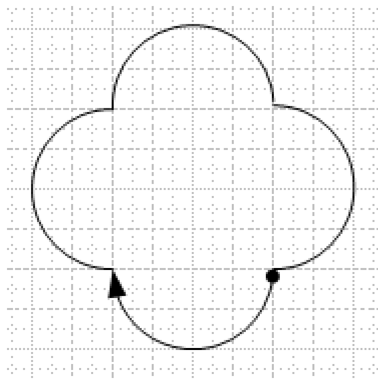


```
// CW rotation
radius = .5;            // 1/2 inch radius.
angleSweep = 180;
angleStart = 90;        // 90 degree offset on radius
// move at 1.5 inches/sec feed rate with accel/decel of 4 inches/sec2
// velocity / radius = rad/sec
move at 1.5 to 0 using 4,4;
wait for in position;
```



```
// CW rotation & flip
radius = -.5;           // 1/2 inch radius but flip the curve, CW.
angleSweep = 180;
angleStart = 0;         // no offset to the radius
```

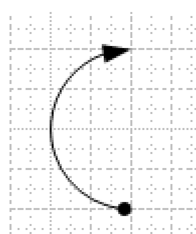
```
// move at 1.5 inches/sec feed rate with accel/decel of 4 inches/sec2
// velocity / radius = rad/sec
move at 1.5 to 0 using 4,4;
wait for in position;
```



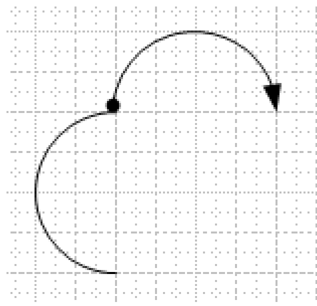
```
[stall]
delay 4000 ms;
// Do it again, forever...
goto top;
```

Example time based moves:

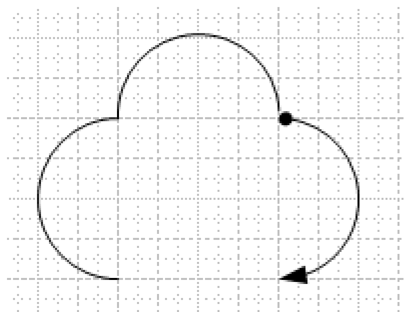
```
[beginTest]

axisY = 2;  // set that we will coordinate with axis 2

[top]

// CW rotation
radius = .5;            // 1/2 inch radius, 5 rev/inch., uud = 5.
angleSweep = 180; // 180 degree sweep
angleStart = -90; // negative 90 degree offset on Y axis
// draw arc in 1 second and calculate required velocity and acceleration
move in 1 to 0;
wait for in position;
```



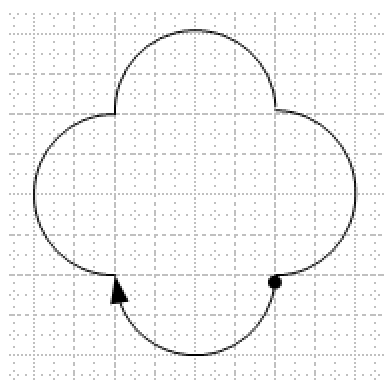
```
// CW rotation
radius = .5;            // 1/2 inch radius.
angleSweep = 180;
angleStart = 0;         // no offset on Y axis
// draw arc in 1 second and calculate required velocity and acceleration
move in 1 to 0;
wait for in position;
```



```
// CW rotation
radius = .5;            // 1/2 inch radius.
angleSweep = 180;
angleStart = 90;        // 90 degree offset on radius
// draw arc in 1 second and calculate required velocity and acceleration
move in 1 to 0;
wait for in position;
```



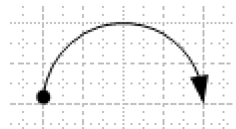
```
// CW rotation & flip
radius = -.5;           // 1/2 inch radius but flip the curve, CW.
angleSweep = 180;
angleStart = 0;         // no offset to the radius
// draw arc in 1 second and calculate required velocity and acceleration
move in 1 to 0;
wait for in position;
```



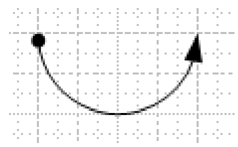
```
[stall]
delay 4000 ms;
// Do it again, forever...
goto top;
```

Arc summary with positive/negative radius and sweep angles:

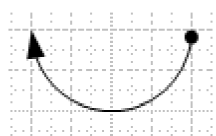
Radius = -.5, angleSweep = 180, angleStart = 0



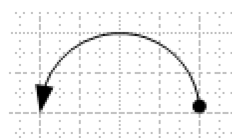
Radius = -.5, angleSweep = -180, angleStart = 0



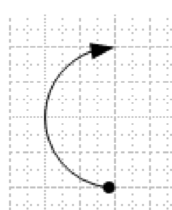
Radius = .5, angleSweep = 180, angleStart = 0



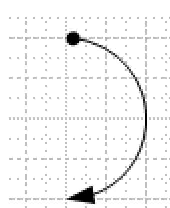
Radius = .5, angleSweep = -180, angleStart = 0



Radius = .5, angleSweep = 180, angleStart = 90



Radius = .5, angleSweep = 180, angleStart = -90



If you wish to exit circular interpolation mode and go back to normal single axis moves first execute a 'stop' command with the axisY still set. This will cause a stop on both the X and Y axis with the single command. After a 'wait for in position' command you may begin single axis moves (>= V1.83 required).

Camming & Splines (2D/3D) – Linear, Cubic & Quadratic

Camming tables in QuickMotion are two-dimensional arrays of floating-point data. There are 8 tables available for use with the M3-41, numbered 0 through 7, each having up to 2000 rows and always 2 columns. These columns are named “x” and “y”. Although their primary use is to hold data for *spline*- and *CAM*-based motion, they can be used to hold arbitrary data such as positions for recipe-based motion. Although limited to 8 tables per axis, these tables can also be swapped out dynamically and refreshed with new data when loaded from the controller file system.

Spline tables use the “x” column as time and the “y” column as a *relative* position. *CAM* tables use the “x” column as a *relative master* position and the “y” column as a *relative slave* position.

Since *spline* and *CAM* tables use *relative* position data, the first point pair in these tables must be 0.0, 0.0 (time/master-position of 0, position/slave-position of 0). The exception to this is with CAM tables where the y component can be non-zero, thereby establishing an offset. In addition, for any tables used for *spline* and *CAM* operations, all “x” values must be increasing, that is: a given row’s “x” must be greater than the previous row’s “x”. Also, the minimum number of rows (pairs) in these tables is 3.

It is recommended that CAM tables and instructions be used whenever possible. Significant enhancements have been made to camming which have currently not been carried forward to splines. Some of this consists of the ability to start on non-zero y column values, ability to start anywhere within a table, and forward and reverse table traversing.

Points in a *spline* or *CAM* table are also referred to as *knots*, as they represent critical loci that must be passed through when interpolation occurs.

For example, in the following *spline* table:

0.0	0.0
1.5	2.0
2.0	2.5
3.0	3.0
4.0	2.0
5.0	0.0

There are 6 knots. Since this is a *spline* table, the last 5 knots are interpreted as follows:

- At time = 1.5 seconds, the position of the axis should be 2.0 user-units beyond where the axis started this spline move.
- At time = 2.0 seconds, the position of the axis should be 2.5 user-units beyond where the axis started this spline move.
- At time = 3.0 seconds, the position of the axis should be 3.0 user-units beyond where the axis started this spline move.
- At time = 4.0 seconds, the position of the axis should be 2.0 user-units beyond where the axis started this spline move.

- At time = 5.0 seconds, the position of the axis should be back where the axis started this spline move.

The position of the axis between these "knots" is determined by the interpolation method specified by the MSB code when the table is *started*.

The three available interpolation methods in QM for *spline* (and *CAM* tables) are:

Linear - A straight-line joins each knot.

Quadratic - A piecewise 2nd degree polynomial is fitted between this knot and the next; the first derivative of the first point is forced to 0.

Cubic - A piecewise 3rd degree polynomial is fitted between this knot and the next two knots; the first and second derivatives of the first point is forced to 0.

Splines and Camming work identical to the M3-40 module. The M3-41 adds the ability to synchronize multiple axes, up to 3, using splines and camming tables. As with 2D & 3D Linear Interpolation, the axisY and axisZ properties may be assigned to the desired axis with which to synchronize the controlling X axis.

axisY – The axis number, from 1 to N, which will be the Y axis, commanded from the X axis. The Y axis must be set for either 2D or 3D interpolation to occur.
axisZ – The axis number, from 1 to N, which will be the Z axis, commanded from the X axis in 3D operations.

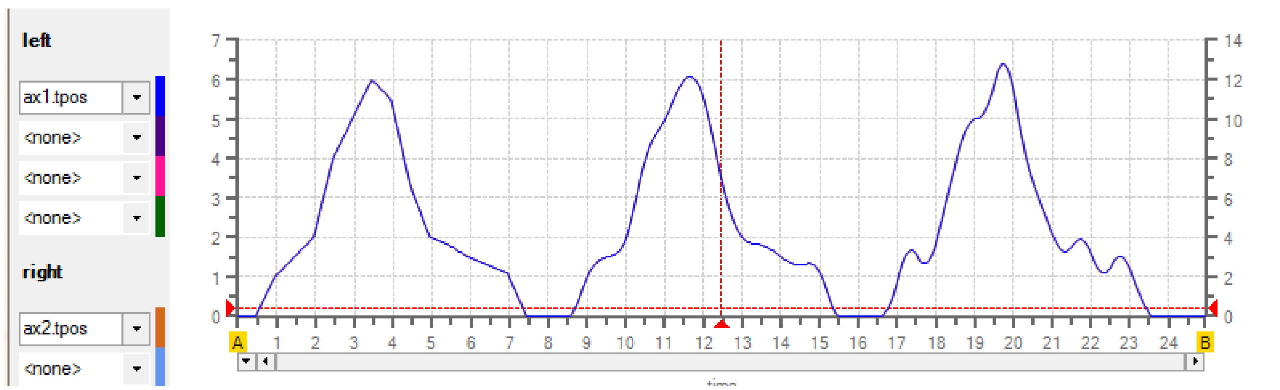
In order to operate in 2D mode the first table has its data assigned, for 3D, the first table # +1 is used for the Y axis, first table #+2 for the Z axis. The Z axis cannot be used without the Y axis. When the table pre-compute is executed on the first table, it will automatically computer the other tables based upon the axisY and axisZ contents. The same is true for the 'table start', start the first table and the other axis will start simultaneously. Note that there must be the exact same number of entries in all tables, using the same time references for splines, or master position for camming. Also all axis must be in Cyclic Synchronous Position (CSP) mode. The following is an example of a 2D spline operation:

```
[beginTest]
// Demonstrates the three types of splines:  linear, cubic,
// and quadratic

// Set the Axis to use for Y, this can be done anytime
// prior to the precompute.
axisY = 2;

// Clear the X and Y axis
table 0 clear;
table 1 clear;

// Add the spline data points for the X axis.
// First is time, send if position.
```

```
table 0 addseries
0.000 ,     0.0000      :                  // simple spline table
0.500 ,     1.0000      :
1.000 ,     1.5000      :
1.500 ,     2.0000      :
2.000 ,     4.0000      :
2.500 ,     5.0000      :
3.000 ,     6.0000      :
3.500 ,     5.5000      :
4.000 ,     3.3000      :
4.500 ,     2.0000      :
5.000 ,     1.8000      :
5.500 ,     1.5000      :
6.000 ,     1.3000      :
6.500 ,     1.1000      :
7.000 ,     0.000 ;

// The Y table will be X table number + 1
// Now enter the spline data points for the Y axis.
// If Z was used it would be table #3, there
// may be up to 6 tables, 0 to 5.
table 1 addseries
0.000 ,     0.0000      :                  // simple spline table
0.500 ,     2.0000      :
1.000 ,     3.0000      :
1.500 ,     4.0000      :
2.000 ,     8.0000      :
2.500 ,     10.0000     :
3.000 ,     12.0000     :
3.500 ,     11.000      :
4.000 ,     6.6000      :
4.500 ,     4.0000      :
5.000 ,     3.6000      :
5.500 ,     3.0000      :
6.000 ,     2.6000      :
6.500 ,     2.2000      :
7.000 ,     0.0000      ;

// Now calculate the move, note that since
// axisY is set the table for X and X#+1
// will be done by this single instruction.
// If axisZ was set then table X#+2 would
// also be done.
table 0 precompute;            // precomputes 1 & 2

[top]
// zero our position since table is 0 based
zero feedback position;

// Do the X & Y linear move
table 0 start linear  1.0, 1.0, 1;
wait for in position;
delay 1000;
// Do the X & Y cubic move
table 0 start cubic   1.0, 1.0, 1;
wait for in position;
delay 1000;
```

```
// Do the X & Y quadratic move
table 0 start quadratic   1.0, 1.0, 1;
wait for in position;

// pause and do again...
delay 3000;
goto top;
```

The resulting target positions calculated for each move type, on both axes, are shown below. The first is a linear move, second cubic, and third quadratic, note that both axis are exactly synced and overlaying one another:



Camming Tables Special Variant Registers

Camming tables are typically loaded from a QuickBuilder program using the 'table .. addseries' instruction or via ftp and loaded with QuickBuilder. For EtherCAT applications they can also be loaded by writing to a special Variant register group that maps directly to the M3-41A or EtherCAT process (Incentive-PC). This register group allows you to create and modify the 8 available tables on each axis as well as initiate precompute, set the size of the table (number of knots), and computation type (linear, quadratic and cubic).

Each Variant table register 36830 to 36837 operates identically and allows for up to 8 tables to be accessed simultaneously on any available EtherCAT axis, one per register. Each register consists of 3 columns and up to 2001 rows. The table itself will have knot 1 starting in row 1, with column 1 representing time in milliseconds and column 2 positions, as doubles. Row 0 is not used in columns 1 and 2. Column 0 is reserved as an interface to the table to cause some function to occur or read information depending upon the row selected. Details below:

Column	0 (Function)	1 (Time)	2 (Position)
Row 0	Table #0-7 (R/W)	Not used	Not used
Row 1	Axis 1 to N (R/W)	Time mS	Position User Units
Row 2	Number of Knots (R/W)	...	...
Row 3	Calculation Type (R/W) Bit OR Linear 0x01 Quadratiic 0x02	...	...

	Cubic 0x04		
Row 4	Status (R/W)	…	…
Row 5	PreCompute (Write 0x10)	…	…
Row 6	String Knots Write	…	…
Row 7	Clear Table (Write 0x11)	…	…

Column 0 controls the properties of the overall cam table and is independent of columns 1 and 2 which represent a view of the selected table. Row 0, columns 1 and 2 are not used so as to make the terminology of Row 1 equivelent to Knot 1 when accesing the table..

[Row, Column]

[0, 0] – Table number, R/W. Table number starts at 0 and goes to 7, integer based.

[1, 0] – Axis number, R/W. Axis number starts at 1 and goes to N where N is the maximum number of EtherCAT Axis in the system, integer based.

[2, 0] – Number of Knots, R/W. This is the total number of knots in a table starting with Knot 1, integer based. This must be set when a new table is created or existing table is extended. It is not automatically incremented when knot writes occur and should be set to the proper length prior to a precompute. Integer based. If the table is created by an MSB or loaded via ftp then the proper number will appear.

[3, 0] – Calculation type, R/W. By default this is set to 7, all types. Otherwise it is a bit OR of Linear 0x01, Quadratiic 0x02, and Cubic 0x04.

[4, 0] – Status, R/W. Any value may be written here. When a table is cleared it is set to 0 and after each successful write operation to a Knot or functions in row 2, 3, 5 to 7, it is incremented by 1. If it returns a -1 after a write operation, then the write operation has failed. The status will increment to 10,000 and then wrap back to 0..

[5, 0] – PreCompute, Write Only. Write a 0x10 integer to this location to cause a precompute of the table to occur.

[6, 0] – String Knots Write, Write only. Knot information can be written in a string format instead of poking individual time/position values with doubles. The string can be written to one or more knots which do not have to be consecutive. Format is KNOT#:Time mS Position, KNOT#:Time mS Position

Example: 1:0.0 0.0, 2:.01 .00345, 3:.02: .0053

[7, 0] – Table clear, Write Only of 0x11. Clears the table back to empty and initializes it to 0, enabling all compute types.

Torque Control

Some applications, such as a press, may want to dynamically control the maximum torque applied. Yaskawa, Sanyo Denki, and Emerson drives support a variable called 'tmax' which when set represents the

% of maximum torque. At power up tmax is initialized to whatever the drive default is, for example Emerson is 175% or ‘tmax’ = 175. The variable ‘rmstrq’ represents the current torque requirements in %, maximum being that set by ‘tmax’. Note that the property sheet of Quickbuilder is ignored and the actual default powerup/reset value from the drive is read and written to ‘tmax’.

If you exceed the ‘tmax’ value and stall the drive, motion will stop and ‘inpos’ will never be set due to the failure of ‘fpos’ (feedback position) to equal ‘tpos’ (target position). MSB programs must not use the ‘wait for in position’ command or they will hang forever. Either the ‘pstate’ (COMPLETE, 2) or ‘QS2_status’ (STOPPED_READY, 1) may be monitored for the moves completion.

EtherCAT drive specific object access is as follows:

Delta

tmax – 0x6072.0 / 10.0 0 (Max torque %)

Initialized to value in 0x6072.0 or 300.0 if not available.

rmstrq – 0x6077.0 / 10.0 0 (Actual torque, absolute value)

Emerson

tmax – 0x2004.7 / 10.0 (Symmetrical Current Limit PR4.07)

Initialized to value in 0x2004.7 or 175.0 if not available.

rmstrq – 0x2004.4 / 10.0 (Current Demand PR4.04, absolute value)

Sanyo Denki

tmax – 60E0.0 / 10.0 (positive torque %) & 60E1.0 / 10.0 (negative torque %), 0 to 500%.

Initialized to 500.0 during initialization.

rmstrq – 0x6077.0 / 10.0 (Actual torque, absolute value)

Yaskawa

tmax – 0x6072.0 / 10.0 0 (Max torque)

Initialized to value in 0x6072.0 or 300.0 if not available.

rmstrq – 0x6077.0 / 10.0 0 (Actual torque, absolute value)

LinMOT

tmax – 0x13A6.0 / 10.0 (Max Current A) & 0x13BA.0 / 10.0 (Max Current B)

Limit is expressed in Amps, not %. Initialized to value in 0x13A6.0 or 4.0 if not available.

rmstrq – 0x1b93.0 / 1000.0 (Current Demand /1000.0 = Amps)

Mitsubishi

tmax – 60E0.0 / 10.0 (positive torque %) & 60E1.0 / 10.0 (negative torque %), 0 to 100%

Initialized to 100.0 during initialization, not read from the drive.

rmstrq – 0x6077.0 (Actual torque)

IAI

tmax – Initialized to value from properties of drive in QuickBuilder.

Pressing Current Limit set to (tmax/100.0) * 255.0

Load Current Threshold set to (tlim/100.0) * 255.0

tlim initialized to 0.

Below drives do not support the ‘tmax’ function but do provide ‘rmstrq’.

Kollmorgon

tmax – not supported

rmstrq – 0x6077.0 (Actual torque)

Advance Motion Controls

tmax – not supported
rmstrq – 0x6077.0 (Actual torque)

Copley

tmax – not supported
rmstrq – 0x6077.0 (Actual torque)

Parker Hannifin

tmax – Only supported during homing, firmware issue with the drive.
rmstrq – 0x6077.0 (Actual torque)

’tmax’ must be set after ‘drive enable’ in order to take effect. Prior to ‘drive enable’ the current value in the drive is read and ‘tmax’ is initialized to that value or the constants detailed above for each drive. This allows the offline PDO scan to occur without changing the existing value in the drive.

Restarting EtherCAT Programmatically

A request to restart the EtherCAT network can be made programmatically by setting the number of the network module you wish to access (1 = first) in register 12333, followed by writing a 21930 (0x55AA) to register 13464. The network will be taken off line, rescanned, and MSBs restarted. Monitor the online status (13464) prior to access. Note that the I/O count will not be updated in the controller; a restart to the network should not be used to add or remove devices without cycling power on the controller.

12333 – (R/W): Network module bank select register; each EtherCAT and other supported network modules are selectable for further access. ‘1’ is the first module.

13464 – (R/W): Network online status; 1 = online, 0 = offline. Writing a 21930 (0x55AA) to this register causes the network to reset and, in the case of EtherCAT, rescan all I/O and drives. MSBs will be restarted.

The network can also be restarted via telnet and the QuickBuilder EtherCAT Explorer. When restarting the network, any axis properties that were changed programmatically from those shown on the property sheet should be manually initialized. The restart does not re-initialize the property values; those are only set during a hardware reset or power cycle.

Special Register Access

12333 – (R/W): Network module bank select register; each EtherCAT and other supported network modules are selectable for further access. ‘1’ is the first module.

13464 – (R/W): Network online status; 1 = online, 0 = offline. Writing a 21930 (0x55AA) to this register will cause the network to reset and in the case of EtherCAT, rescan all I/O and drives. MSBs will be restarted.

13025 – (RO): Number of digital inputs in the system

13026 – (RO): Number of digital outputs in the system

13027 – (RO): Number of analog inputs in the system

13028 – (RO): Number of analog outputs in the system

13029 – (RO): Number of motors (axes) in the system

Registers 13025 to 13029 are useful for the application to verify the number of I/O it expects to find in the system. Some EtherCAT Masters use a specially configured XML file to simplify I/O count verification. It is up to the application to confirm the correct number of I/O and axes during startup and notify the user of any discrepancies.

Global MSB Registers and Flags

There are 48 double precision user variables per MSB. These variables are local to an MSB and not available to the MSBs executing on a different axis. Global registers, which are local to each EtherCAT Master module, can be used as a shared variable resource amongst the axes. There are 32 registers named 'global_reg1' to 'global_reg32'. Storage operations consisting of arithmetic operations are atomic to each MSB and these registers can be accessed from QuickBuilder just like axis properties. Like user variables, global registers are double precision variables.

Global flag registers are also available, operating the same as global registers. There are 5 flags registers named 'global_flags1' to 'global_flags5'. Unlike variables the flag registers are 32 bit integers. Atomic math operations like '|', '&', and arithmetic bit shifting (<<, and >>) may be used. For example:

```
global_flags1 = global_flags1 << 1;   // shift bits left by 1.
```

QuickBuilder access of global registers and flags are atomic for read operations but not write. If atomic operation is needed it is recommended common bits be used as semaphore flags and is application dependent.

Accessing Properties of another Axis from MSB

Properties such as fpos, tpos, etc., are local to an axis and not shared with other axis MSBs. This limitation can be overridden by using the 'axisptr' property of an MSB. This property controls what axis the MSB will retrieve its property value on a read and write operation. It is typically set to the value of 'axisnum', which is the axis number of that axis executing the MSB. Setting this axis number to any other value will override what axis the property is retrieved from. An example follows:

```
axisptr = 3;        // Monitor axis 3 until the drive is enabled and running
[stall]
if enabled > 0 goto online;
goto stall;         // Wait until available
[online]
axisptr = axisnum;        // Future property access will be for out axis.
```

Any property value can be accessed but make sure you do not attempt to mix property access of more than one axis in a statement since all properties will reference that set by the axisptr. If you wish to read, or write, a value a user variable should be used. For example the following reads fpos on axis 3:

```
axisptr = 3;       // Access axis 3 properties
myvar = fpos;      // read fpos from axis 3
axisptr = axisnum;  // set property access back to our axis
```

Mapping QuickBuilder Axis references to alternate Axis

QuickBuilder has a system variable called $AXISMAP which is an array that contains the axis number of each axis, such that $AXISMAP[1] is initialized with a 1, $AXISMAP[2] with a 2, etc., up to the number of axis. In some application multiple axis may be executing the same MSB and rather than duplicating Quickbuilder code for each axis to access user variables and properties you can simply change the map. For example if axis 1 had a name assigned of ax1_FrontAxis and axis 2 ax2_RearAxis by default ax1_FrontAxis.fposc gets the fposc from the first axis. If you wanted to use the same QB code to access fposc on axis 2 you would se the map, $AXISMAP[1] = 2. Now ax1_FrontAxis references for properties and user variables from within your QB code will be mapped to axis 2, not 1. Note that ax2_RearAxis is also still mapped to 2, unless changed.

Drive Type & Axis Number

The type of drive and axis number that an MSB is executing can be referenced programmatically via the 'eCAT_driveType' and 'axisnum' property variables. 'axisnum' contains the axis number, where 1 is the first. 'eCAT_driveType' is defined as follows:

```
$DRIVE_COPLEY            2
$DRIVE_YASKAWA           3
$DRIVE_ELMO (not supported)    4
$DRIVE_KOLLMORGEN        5
$DRIVE_SANYO_DENKI       6
$DRIVE_EMERSON           7
$DRIVE_AMC               8
$DRIVE_VIRTUAL           9
$DRIVE_IAI_ACON_MODE3    11
$DRIVE_ABB_MICROFLEX     12
$DRIVE_ABB_MITSUBISHI    13
$DRIVE_ABB_PANASONIC     14
$DRIVE_ABB_LINMOT        15
$DRIVE_YASKAWA_V1000     16
$DRIVE_MOTIONLINX        17
$DRIVE_WAGO_ENCODER_631  18
$DRIVE_WAGO_ENCODER_637  19
$DRIVE_APPLIED_MOTION    20
$DRIVE_FESTO             21
$DRIVE_WAGO_ENCODER_630  22
$DRIVE_PARKER_HANNIFIN   23
$DRIVE_MAXON             24 (reserved)
$DRIVE_DELTA       25
```

As with 'cmode', either the constant name beginning with '$' or the actual numeric value may be used in an expression.

Example:

```
if eCAT_driveType == $DRIVE_AMC goto AMC;
if eCAT_driveType == $DRIVE_VIRTUAL goto Virtual;
```

Drive Object Access (SDO)

An MSB program can directly access any remote drive object, assuming the drive is in the correct state for the write operation. This is implemented by allowing SDO (Service Data Object) reads and writes. The command syntax is as follows:

```
sdo write <value>, <slave>, <object #>, <object index>, <object size, 1 to 4>;
sdo read <result storage>, <slave>, <object #>, <object index>, <object size, 1 to 4>;
```

If an error occurs, the MSB will enter a fault state. The Object size is the size of the data, in bytes, as specified by the manufacturer. The <slave> is the destination of the read or write: -1 for the current slave the MSB is operating on, or the slave index, starting at 1, as it appears in the discovery tree.

Example:

```
// 0x609a.00 is the homing acceleration object.  This is not used
// on Yaskawa so we can use it for general storage
sdo write counter, -1, 0x609a, 1, 4;
// Read the object value back, should be the same...
sdo read counter, -1, 0x609a, 1, 4;
```

Many objects cannot be accessed while the drive is in the operational state, resulting in a state error message and thus a fault. Also Virtual Drives support intermittent sdo read/writes to a specific slave, not to -1 since it is not actually online.

Although SDO access to slaves is supported it is suggested it be used sparingly as it limits the bandwidth available to the cyclic PDO transmissions. It may also result in warning messages of PDO re-transmission due to timeouts caused by SDO messaging. This is especially true when using a 500 µS EtherCAT control loop time.

CHAPTER
7

[7] Registration, Absolute Positioning, & Distributed Clock

Registration

Registration allows the present servo position to be captured when an input is triggered. The standard MSB commands can be used for registration as on the M3-40 card. Registration is only supported on certain drives: Kollmorgen, Sanyo Denki, and Yaskawa (not Wago encoder modules). These drives support the Touch Probe Function object, 0x60B8, as well as the Touch Probe Status, 0x60B9, and Touch Probe Value 1, 0x60BA. The Sanyo Denki and Yaskawa both support Touch Probe Value 2, 0x60BC for an additional registration input.

The QuickMotion language isolates the programmer from the interaction with these objects making registration simple to implement. The ‘set capture’ command selects the drive input probe 1 or 2. ‘set capwin’ sets the window within which the capture is allowed to occur, the reference for comparison, as well as optionally arming the touch probe input. The MSB instructions are summarized below:

syntax

set capture *transition* **of input** *input*

parameters

transition	*rise*, ~~*fall* or *edge* (any)~~ (drive dependent but only rise supported)
input	drive touch probe 1 or 2, reference specific drive for proper input wiring

This statement initializes the parameters to be used for all captures on this axis, specifying the input (*capInput*) to use. The following variables are computed and available after a successful capture:

capposc	capture position in encoder counts
cappos	capture position in user units
capTriggered	flag set to 1 when capture occurs

Note: *capposc* and *cappos* are only valid when *capTriggered* is a 1. Once armed *capposc*/*cappos* will reflect the value latched when the capture input goes active but is not necessarily within the defined capture window. *capTriggered* verifies the capture window against the latched *capposc*/*cappos*, prior to setting. If more than one running MSB on an M3-41 module arms the *same* input for capture, unexpected capture results may occur.

Only one input may be armed for capture at a time *per axis*. If another input is presently armed when this command is issued, the other input is effectively *disarmed*

syntax

```
set capwin range start, end using reference { arm }
```

parameters

`start` — Start window position to compare against *reference*. *Reference* >= *start*.

`end` — End window position to compare against *reference*. If equals *start* then no window exists and capture will occur based on input. *Reference* <= *end*.

`reference` — the encoder count scaled reference variable to compare to:

fposc	feedback position
mposc1 - **mposc5**	master position counters #1 through #5
mposc	master position counter
smodc	slave position (modulo)
smark	slave marked position
tmc1 tmc2	temporary master counters #1 & #2
tsc1 tsc2	temporary slave counters #1 & #2
sdc	slave decrement counter
fposc1	feedback position of axis 1 (fposcA)
fposc2	feedback position of axis 2 (fposcB)
tmodc	Temporary master counter mod mmc
sfposc	Secondary feedback position of axis
tposc	Target position of axis
ctr0	din1 mapped input counter
ctr1	din2 mapped input counter
ctr2	din3 mapped input counter
ctr3	din4 mapped input counter
ctr4	din5 mapped input counter
ctr5	Local quadrature encoder 1
ctr6	Local quadrature encoder 2
ctr7	Local quadrature encoder 3

`arm` — If included will arm the capture, if not arm will need to be done by a Wait or On

	command.

This statement initializes a window to be monitored for valid captures to occur, anything outside this window is considered invalid and ignored. If the capture occurs outside this window it will automatically be re-armed. If 'arm' is specified this statement will automatically arm the capture prior to completing this instruction. The *capwinStart* variable is the start of range and the *capwinEnd* variable is the end of range, inclusive. The '*capMod'* variable is used to perform a modulus on the reference value prior to comparison to the *start/end*, the remainder after the modulus is the value compared. This helps in situations that may experience rollover.

Below is an example program to latch the position whenever Touch Probe 1 occurs during a set of moves:

```
[beginTest]
// Setup registration
// First set which probe to use.
// Yaskawa & Sanyo Denki support Touch Probe 1 & 2.
// Kollmorgen supports only Touch Probe 1.
// Yaskawa fires on input being active, since active low this is falling edge
set capture rise of input 1;  // Select probe 1 (enter 2 for probe 2)
// Clear the window so capture will happen moment probe occurs
set capwin range 0, 0 using fposc arm;     // Arm at same time
// capTriggered will be set to a 1 when the capture occurs.

[run]
// Begin the move, 1 rev/second for 2 revolutions
if capTriggered != 1 goto notTriggered;
// Clear the window so capture will happen moment probe occurs
set capwin range 0, 0 using fposc arm;     // Clear the range, arm at same time
                                           // 'wait capture' will arm as well.
[notTriggered]
move at 1 for 2;
wait for in position;
// Delay 1 second once in position
delay 3000 ms;
// Do a relative move back 2 revolutions at 1 rev/second
move at 1 for -2;
wait for in position;
// Delay 1 second once in position
delay 3000 ms;
// Do it again, forever...
goto run;
```

'touchProbeStatus' MSB variable maps directly to object 0x60b9.

Absolute versus Incremental Positioning Modes

By default the Incentive uses Incremental Positioning mode. At power-up it records the current absolute position and zeros it; thereby fpos and tpos are at 0. If the encoder is battery backed, you can use Absolute Positioning mode to maintain position. This mode sets fpos and tpos to the current actual position (0x6064). You can later clear it to 0 by using the 'zero feedback position' command or do moves based on an offset from tpos. Regardless, Absolute Positioning mode allows you to maintain position after a power cycle.

Enable Absolute Positioning mode by setting the encoder_mode axis property variable to a 1, prior to executing the 'drive enable' command. A value of 0 is for an incremental encoder. This variable can also be automatically set by using the axis property pull-down menu option: absolute.

For example, with Yaskawa and a circular table, this is done by using SigmaWin+. Perform this operation with the following sequence:

1. Make sure the drive is set for Absolute Encoder (Pn002.2). If not, set it, exit the program, and re-enter (needed to enable menu functions).

2. Set the Servopack for Multi-turn (Pn205), setting the number of revolutions of the motor to one revolution of the circular table. Write to the drive and cycle power (note that a setting of zero is 1:1).

3. Setup->Multiturn Limit Setup is invoked. The number must match the number previously set as this sets it in the motor; cycle power.

4. Setup->Reset Absolute Encoder, cycle power. This establishes a position within the Multi-turn window. Example: If the Multi-turn value (Pn205) is set to '0', resetting the absolute encoder places the motor's position somewhere between '0' and '1048576' counts; cycle power.

6. Setup->Search Origin is done to locate the actual rotation reference position of the motor. The function allows you to jog the axis into the encoder's marker position, also called the "Point of Origin."

Using the Multi-turn functionality of the Yaskawa drive ensures that whenever the power is cycled, the actual position (0x6064) will be based on one rotation of the circular table. While running, the actual position (0x6064) will increment/decrement normally as a 32-bit number.

If you wish to zero the home position, the Absolute Encoder Home Offset object (0x607C) can be set. The value in this object is added to the Absolute Position and is the value placed in the actual position (0x6064) object, thus zeroing position. In most drives this is the 0x607C object. The value written is the complement of the position when at home, upon power-up. This causes fpos and the actual position (0x6064) object to appear to be 0 at home. For Yaskawa drives, use the SigmaWin+ utility to initialize 0x607C, since it is a non-volatile object that must be set prior to the drive being operational.

Distributed Clock & DC Sync

By default the distributed clock is always read from the first DC slave and distributed to all the slaves in the system using the EtherCAT ARMW command. The first slave in the EtherCAT cabling that supports it will be assigned the role of distributed clock master and the EtherCAT Master will read the time from this slave on every control loop cycle and write it to all following slaves on the network.

Both 32 & 64 bit reference clocks are supported with the release of M3-41 V1.46 and above. Earlier revisions only supported 64 bit.

Currently 64-bit clocks are available in the following supported devices:

- Beckhoff EK1100 couplers
- Wago 750-354 couplers
- Omron GX-JC06 EtherCAT Junction Slaves
- Sanyo Denki drives.
- ABB e150 Drives
- SMC Corp not supported, must not be first slave device.

Currently 32-bit clocks are available in the following supported devices:

- Yaskawa Sigma 5 & 7
- Copley Accelnet
- SMC Corp not supported, must not be first slave device.

With the Beckhoff EK1100, no other modules are needed, just the coupler as the first node. Wago requires an IO module as well as an end module.

Currently the only drive that requires DC Sync all the time are the Emerson/Control Techniques, ABB, Mitsubishi, and Sanyo Denki drives; all others can run in free-run mode and will interpolate the commanded position. There are two possible syncs: Sync0 and Sync1. Sync0 alone is used more commonly than both Sync0 and Sync1. The example below shows how to enable both. To disable Sync1 in the example, set its shift time from Sync0 to 0.

Only Yaskawa Sigma 5 & 7 supports Sync0 & Sync1. Other drives are limited by the manufacturer. AMC drives do not support DC Sync. Fully tested and supported drives with DC Sync are Yaskawa, Copley, Sanyo Denki, Mitsibushi, and Emerson. Mitsibushi and Sanyo Denki must have DC Sync enabled prior to being operational thus they are always set to 1mS clock and 250uS offset on Sync0.

The master will attempt to sync to the clock of the slave that is assigned the role of distributed clock master, but expect drift to occur due to variations in the operating system and interrupt overhead. The more drives in the system, the more drift. The cyclic data will be consistent (control loop time), but the point at which the slave receives the data will drift anywhere from a few nanoseconds up to an estimated 75µS, constantly re-syncing, with an average jitter of about 2 µS. On most drives this is not a problem as the drive will interpolate. Many drives, such as the Yaskawa, sample at very high rates (62.5 µS). It is best to set the dc sync to the same value as the Master PDO control loop update time.

```
// Activate DC Sync0 each cycle time with no Sync1, always do before
// 'drive enable'
// dcsync <slave node or -1 for current>,
//     <Sync0 Cycle Time in nanoseconds, ns>,
//     <Sync1 shift from Sync0, ns>,
//     <Sync0 shift from Cycle Time, ns>,
```

```
//    <Sync start delay in ns>
// Set all parameters to 0 except the slave node to deactivate.
// Below is a 1mS Sync0 cycle time with no Sync1, control loop is 1 mS.
// Sync0 starts at cycle time and is not shifted and there is a
// 100mS delay before it all starts the first sequence.
      delay 2000 ms;    // Needed for restarts so have idle time on clock off.
      dcsync -1, 1000000, 0, 0, 100000000;
      delay 105 ms;     // starts 100 milliseconds into the future

// Enable the drive, turning power on to the amplifier.  The current position
// will be constantly updated so the drive does not move.

      drive enable;
```

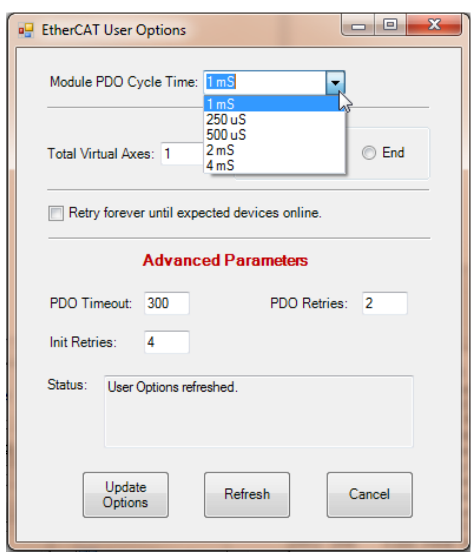
It is best to use DC Sync when using Cyclic Synchronous Position mode, especially with machining operations. Failure to enable can cause a small amount of infrequent servo noise as the drive interpolates the commanded position.

Some devices will not operate correctly unless the DC Sync is enabled (e.g., Emerson/Control Techniques). It is also best to set the DC Sync prior to enabling the drive since some drives, such as Emerson, require this.

When using DC Sync, or any multiple drive systems, it is best to verify all drives are enabled prior to executing a MOVE command. The DC Sync command can take time to execute since it must place the drive in a non-operational state, initiate numerous commands and then make it operational again. Only one DC Sync command will execute at a time, with other axis locked out until completion. In a large system this can cause several seconds of delay where the first drive is ready to execute but the last drive is still enabling DC Sync and the drive. Reference Chapter 4, "*Accessing Properties of Another Access*".

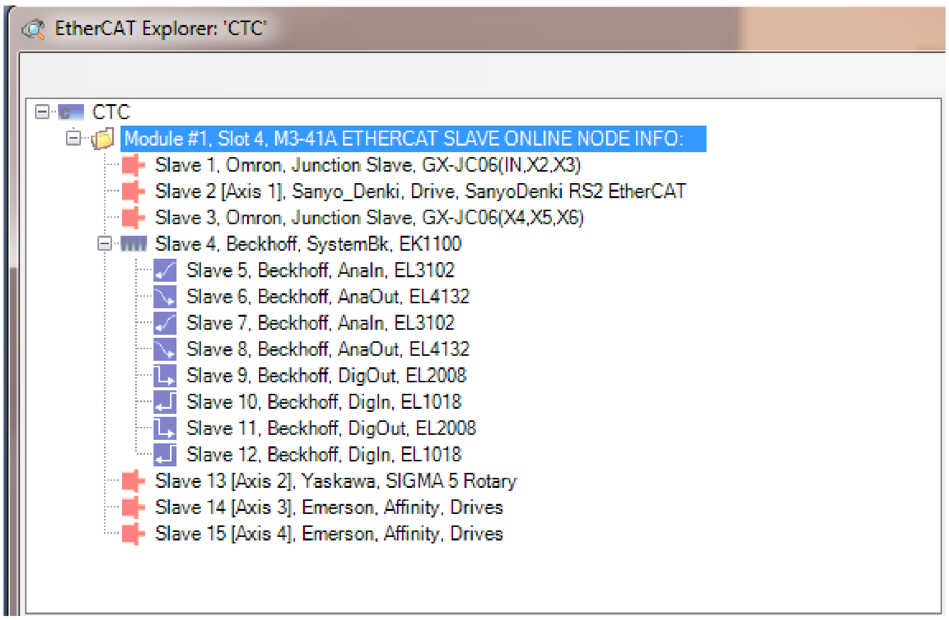
EtherCAT Master Control Loop Cycle Time

The M3-41 and Incentive PC can support an EtherCAT scan time of 500 µS, 1 mS, 2 mS, or 4 mS. The User Options form (Chapter 8), available within the EtherCAT Explorer, is used to set the desired speed. The network has to be restarted or controller rebooted to change the control loop time.



For most applications it is best to use the default 1 mS scan time. Drives tend to lag regardless of how fast the position information is updated (CSP mode), this is dependent on the smoothing algorithm resident in all drives. **When running 500 µS it is recommended that no more than 4 drives be used**, some applications may support up to 6 but it is dependent on optional IO and the application program being executed. Applications which do not use the 'host read/write' commands can support more drives.

One way to determine if your application is approaching the limit is to view the Log Buffers. The following logs are from the Test Suite application in the Appendix, running two Emerson, one Sanyo Denki, and one Yaskawa drive with a DC Sync of 500 µS:



*** Module #1, Slot 4 ***
M3-41A ETHERCAT MASTER
INFO: Time: 2308.792, Scanning = 2, Cycle 0.5000 mS, [Overhead 0.3326 mS, Min 0.1584 mS, **Max 0.4522 mS**],
Adjusted Tick 0.4992 mS, Correction -820 ns, Max Tick 0.5000 mS, [Idle Time 0.0860 mS, Min 0.0031 mS, Max 0.3465].
[Sync Time 0.0070 mS, Min 0.0000 mS, **Max 0.2500, Avg 0.0095**].

Reference the 'Overhead' information with the Max at .4522 mS. This means that you had 500 – 452 or 48 µS of spare control loop time, worst case. Once you exceed 500 a warning will occur, 550 an error. It is recommend that this number stay below .425 mS. This particular application timing was from a stress test with 4 drives, each of which were reading and writing local controller registers (host read/write) while at the same time executing motion commands. Also note the Error Time of a Max .250 mS, with an average jitter of 9.5 µS, this is excessive but still functional.

The following is the same test with no host read/write instructions:

*** Module #1, Slot 4 ***
M3-41A ETHERCAT MASTER
INFO: Time: 2815.575, Scanning = 2, Cycle 0.5000 mS, [Overhead 0.3168 mS, Min 0.2004 mS, **Max 0.4157 mS**],
Adjusted Tick 0.5000 mS, Correction -3 ns, Max Tick 0.5001 mS, [Idle Time 0.1534 mS, Min 0.0042 mS, Max 0.2542].
[Sync Time 0.0005 mS, Min 0.0000 mS, **Max 0.0651, Avg 0.0026**].

Note how the ‘Overhead’ Max has dropped to .415 mS and the Error Time Max is .065 mS with an average jitter just 2.6 µS. This application will run fine at 500 µS scan time.

DC Sync should be set to match the cycle loop time, for example, 500000 nS, not 1000000 nS if 500 µS is enabled. This is especially critical on Emerson drives. It is also recommended that common bits be used instead of the host read/write instruction, wherever possible.

CHAPTER 8

[8] EtherCAT IO, PLS, & PWM

Inputs/Outputs

EtherCAT IO can be accessed by QuickBuilder application programs as though it is local IO, appearing transparently as digital and analog inputs/outputs. QuickBuilder MSB's have additional capabilities to access IO that is resident on a drive, the local M3-41 module, and/or remotely on a slave device such as a Wago, Turck or Beckhoff IO block. IO can not only be accessed by an MSB controlling a physical drive but a virtual drive, in fact, an EtherCAT network with no drives can be designed with nothing but virtual drives controlling the IO allowing for a very flexible system.

A number of MSB IO arrays are available with the indexes into these arrays determining the source or destination of the operation.

<index>: (may be an immediate numeric or indirect via a variable reference)

Bit oriented indexes: an array index of 1 to 32 references each input or output bit available on the drive whose axis is assigned to the MSB. An array index of 501 to 1000 is reserved for local module IO (global_inputs/global_outputs), where 501 is the first. An array index of 1001 to 2025 is reserved for remote IO device blocks, such as Wago, Turck, Beckhoff, and SMC, where 1001 is the first.

Byte oriented indexes: an array index of 1 to 4 references each input or output byte available on the drive whose axis is assigned to the MSB. An array index of 501 to 1000 is reserved for local module IO bytes (global_inputs/global_outputs). An array index of 1001 to 2025 is reserved for remote IO device blocks, such as Wago, Turck, Beckhoff, and SMC.

Word (32 bit) oriented indexes: An array index of 1001 to 2025 is reserved for remote analog IO device blocks, such as Wago, Turck, Beckhoff, and SMC.

Arrays:

inputs[ ] – Digital input bits, where 1 is the first input.
inputs8[ ] – Digital input bytes, atomic to the byte level, where 1 is the first byte.
outputs[ ] – Digital output bits, where 1 is the first output.
outputs8[ ] - Digital output bytes, atomic to the byte level, where 1 is the first byte.
ains[ ] – Analog inputs, 32 bits. If the device being accessed is 16 bits, 0x0000 is added to the high bytes.
aouts[ ] – Analog outputs, 32 bits. If the device being accessed is 16 bits the data will be truncated.

Example using Output bytes:

```
// Attached device is a Wago IO block with 40 digital outputs,
// 40 digital inputs, 8 analog outputs, and 8 analog inputs.
//
// Shift 8 bits on the output
//
// Write a byte output, index can be immediate numeric or variable
// reference.
[test1]
 [_begin]
outputs8[1001] = 0x01;  // 1001 is the first remote 8 bit output block,
                        // write a 1 to first bit
[loop]
delay 250 ms;
// Shift the bit up by one and update the output
outputs8[1001] = outputs8[1001] << 1;
if (outputs8[1001] != 0) goto loop;
goto _begin;
```

Example using Output bits:

```
// Attached device is a Wago IO block with 40 digital outputs,
// 40 digital inputs, 8 analog outputs, and 8 analog inputs.
//
// Shift 8 bits on the output
//
// Write each output a bit at a time, shifting active output up by
// 1 each time.
[test2]
index = 1001;
[_begin1]
outputs[index] = 0x01;  // 1001 is the first remote output block,
                        // write a 1 to first bit
[loop1]
delay 250 ms;
// Turn the current output off
outputs[index] = 0;
// Point to next output bit
index = index+1;
// turn the next output on
if (index != 1009) goto _begin1;
```

```
index = 1001;      // start at beginning again
goto _begin1;
```

Example using analog input/output:

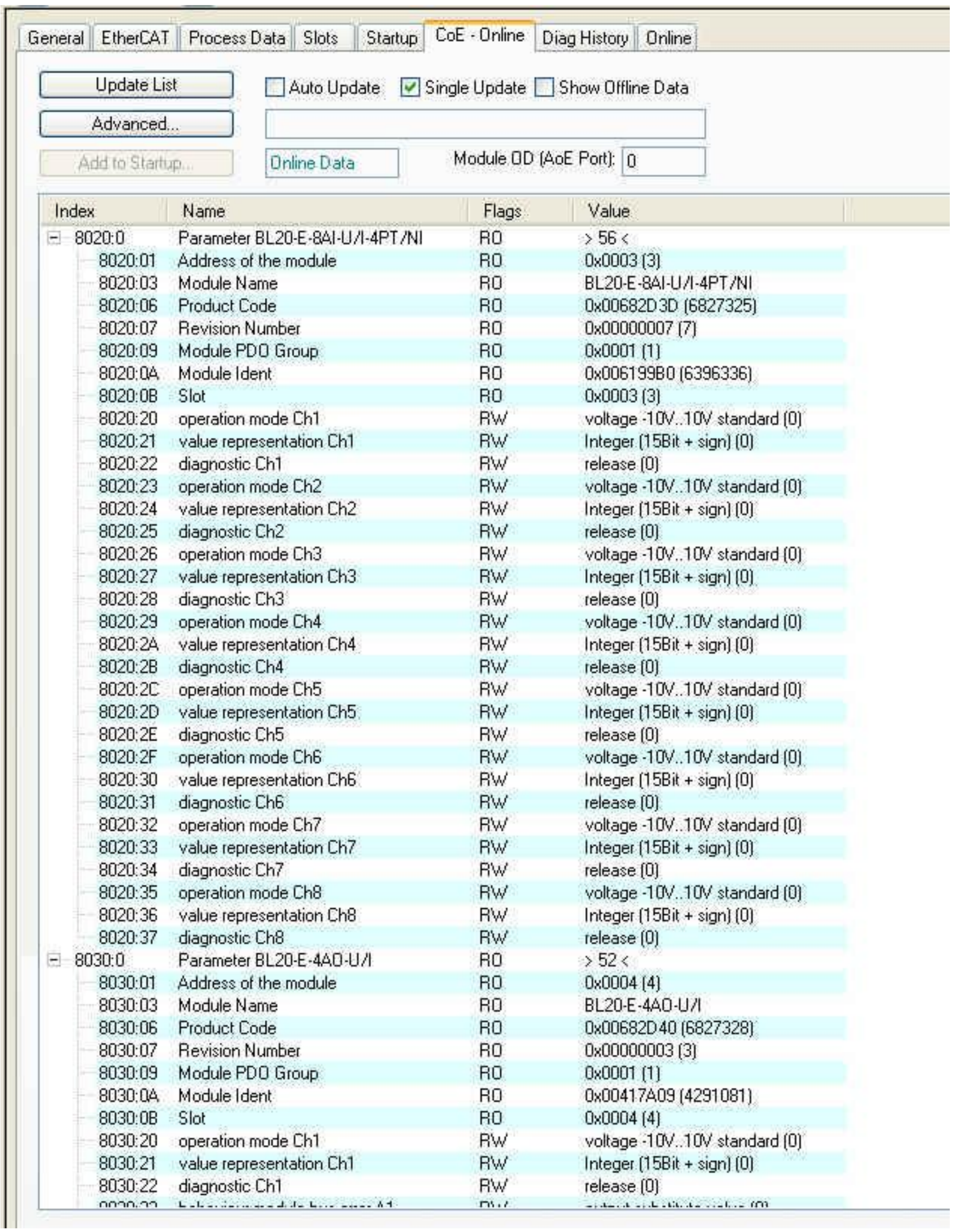
```
// Test remote Analog input/outputs #5
[test3]
index = 1005;
val = 0;
[loop3]
// update the analog output value
aouts[index] = val;
// Allow analog out to stabalize
delay 250 ms;
// Analog output is looped back to Analog input via external wire
myval = ains[index];
val = val + 100;        // increase analog output value
if (val <= 10000) goto loop3;
goto test3;  // reset value
```

Example using digital input:

```
// Mapped global input example
// Map drive input 1 to global input 1 so can use falloff or
// riseof commands to monitor its state
Set mapped input 1 to input 501;
[top]
On fallof 1 goto falledge;
goto top;
[falledge]
// do whatever desired when falling edge detected
// and await level state again…
// … do something until back to high again …
if inputs[1] goto top;
goto falledge;  // waiting for signal to go high again
```

EtherCAT IO Configuration

Some EtherCAT IO, such as Wago and Turck have modules that are configurable. If a special mode, other than the default is needed it is suggested that either that mode be saved to the devices EEPROM using the Beckhoff EtherCAT Configurator or a specific SDO write command be issued to the device's configuration object by a virtual axis, at initialization. Refer to the specific manufacture's manual for object address information. For Wago and Turck this is typically the 0x8000 * (slot number -1) * 0x0010 object, with each index offering a specific property. Below shows a Turck IO device with an analog input module in slot 3 and analog output in slot 4.



Mappable Input IO & Counters

The first five inputs of each axis are by default mapped to the available drive inputs, with up to 32, depending on those available for the specific drive. These inputs have a number of MSB commands available for special handling (reference the QuickMotion Reference Guide):

```
on asynchevent asynchhandler
wait for transition of input { or condition }
pls output using reference definitions
set capture transition of input input
variable = ctr[n]
ctr[n] = expression
ctr[n] = offset
variable = dins
variable = din1
variable = din2
variable = din3
variable = din4
variable = din5
```

An additional feature is the ability to remap the first five inputs to any other available input. This can then be used to drive the above commands, including counters, ctr0 to ctr4. The 32 bit property 'dins' will have its first 5 bits originating from the mapped location, with the remainder from the drive. If the first 5 bits are needed from the drive they can be read using the input[ ] array.

The command to remap an input is as follows:

syntax

```
set mapped input index to input input { count edge }
```

parameters

index	The index of the mapped input assigned the input to control. 1-5 index
input	the input to assign to the index 0 to disable mapping and restore to default drive input, counter disabled 1-32 on drive 501 to 1000 local to M3-41 module 1001 to 2025 remote EtherCAT IO block
edge	*rising* or *falling*, optional, with default being *rising*
	Note: index 1 is assigned to ctr0, 2 to ctr1, etc., also counter cleared upon execution.

Example:

```
// Map dins bits 0 to 4 to remote EtherCAT IO inputs 25 to 29.
// The counters ctr0 to ctr4 will be observed using QuickBuilder
// watch window since remote output #25 is connected to input
// #25, it will count on the rising edge once per second.
set mapped input 1 to input 1025 count rising;
```



```
set mapped input 2 to input 1026 count falling;
set mapped input 3 to input 1027 count rising;
set mapped input 4 to input 1028 count rising;
set mapped input 5 to input 1029 count rising;
[loop]
outputs[1025] = 1;
delay 500 ms;
outputs[1025] = 0;
delay 500 ms;
goto loop;
```

Pulse & PWM Generation

Timed pulses and PWM can be generated using the EtherCAT control loop time as a time tick. The MSB language has a number of instructions in this regard:

syntax

pulse_ext *output* **for** *n*

parameters

output	the output to pulse 1-32 on drive 501 to 1000 local to M3-41 module 1001 to 2025 remote EtherCAT IO block
n	the time to pulse the output, expressed as control loop ticks (up to 5 pulsed outputs may be active at one time)

This statement causes the specified *output* to pulse for the specified duration of EtherCAT control loop *ticks*. The *output* follows the same access numbering convention as the outputs[ ] array. If the output is already *on* when this statement executes, the output state is unchanged, however it will be turned *off* after the specified time.

If another statement changes the state of the output to off before the allotted duration, the generation of the pulse is aborted. The generated pulse will be synced & set active on the next control loop tick.

Example:

```
// Test pulse and output on
[test4]
pulse_ext 1002 for 500; // turn output 2 on for 500 ms
delay 1000 ms;          // this should make it appear as on 1/2, off 1/2
second.
goto test4;
```

syntax

```
generate output output rate freq
```

parameters

`output`	the output to pulse 1-32 on drive 501 to 1000 local to M3-41 module 1001 to 2025 remote EtherCAT IO block
`freq`	the frequency (in control loop ticks) to generate 50% duty cycle pulses; rounded to an integer. Specifies the on time.

This statement begins or ends generation of pulses using a specific output. When a frequency of 0 is specified, no pulse generation occurs. This effectively turns the output back into a general-purpose output. A 50% duty cycle is generated, thus a rate of 1 would turn the output on during one control loop cycle and off on the next. A 1 mS control loop would yield a 500 HZ output square wave.

Example:

```
// Test square wave on output #1
[test5]
// Turn output 1 on for 75 ticks, off for 75 ticks
generate output 1001 rate 75;
// Turn output 2 on for 125 ticks, off for 125 ticks
generate output 1002 rate 125;
// Turn output 3 on for 250 ticks, off for 250 ticks
generate output 1003 rate 250;
// Turn output 4 on for 500 ticks, off for 500 ticks
generate output 1004 rate 500;
// Turn output 5 on for 1000 ticks, off for 1000 ticks
generate output 1005 rate 1000;
[loop5]
goto loop5;       // stall, pulses will continue forever...
```

syntax

```
pwm output output on tickson off ticksoff cycles n
```

parameters

`output`	the output to pulse 1-32 on drive 501 to 1000 local to M3-41 module 1001 to 2025 remote EtherCAT IO block
`tickson`	the number of control loop ticks to activate the output.
`ticksoff`	the number of control loop ticks to de-activate the output.
`n`	The number of complete PWM cycles to do prior to stopping where 0 terminates an active PWM cycle and restores the state to an inactive output and -1 runs forever.

This statement will generate a certain number of variable width output *cycles* with the desired *on* and *off* time in control loop *ticks*. If the *cycles* or *on* time are 0 the PWM will terminate immediately and the output set inactive. If the *cycles* is '-1' the PWM will run forever or until stopped by another instruction with cycles/on time set to 0.

Example with immediate references:

```
// Test PWM output on output 2
[test6]
pwm output 1002 on 125 off 1000 cycles -1;
[loop6]
goto loop6;              // stall, pwm will continue forever...
```

Example with variable references:

```
// Test PWM output on output 2
[test6]
index = 1002;
ontime = 125;
offtime = 1000;
pwm_cycles = -1;
pwm output index on ontime off offtime cycles pwm_cycles;
[loop6]
goto loop6;              // stall, pwm will continue forever...
```

Example with embedded math:

```
// Test PWM output on output 2
[test6]
ontime = 1000;
pwm output 1002 on (ontime/8) off 1000 cycles -1;
[loop6]
goto loop6;              // stall, pwm will continue forever...
```

PLS Outputs

PLS, or Programmable Limit Switch, is an output which can be configured to become active at a high rate of speed based upon some monitored event or state. Typically PLS outputs must become active faster than instructions in an MSB can execute testing the required conditions. For example with a 500 uS control loop the PLS output conditions and state would be tested every 500 uS, much faster than the MSB could execute. The EtherCAT Master allows any output that is resident on a drive, local to the M3-41 module, or a general EtherCAT output to be used as a PLS output. The *output* follows the same access numbering convention as the outputs[ ] array. The available commands are as follows:

syntax

```
set pls index to output output
```

parameters

`index`	The index of the PLS assigned the output to control. 1-5 index

`output`	the output to activate under PLS control 1-32 on drive 501 to 1000 local to M3-41 module 1001 to 2025 remote EtherCAT IO block

There exist five possible PLS functions that may be active per axis. Each of these PLS functions must be assigned the output they are to trigger when the specified event occurs. This instruction assigns the output number to each of the PLS index functions. The index requires this assignment in order to know what output to operate on for all the other PLS instructions. Note that the use of this command will disable and active PLS on the specified index and clear the 'mod' to 0 (not used).

syntax

`set pls` *`index`* `mod` *`mod`*

parameters

`index`	The index of the PLS assigned the output to control. 1-5 index
`mod`	The modulus to apply to the PLS reference, whereby the remainder is the resulting value, prior to comparison of a window value. This allows for taking into account rollover. For example if there are 65536 counts per revolution and the reference window is defined for the first revolution, then each revolution would be able to reference the same set of window values. Default is 0 meaning the reference value is not changed.

There exist five possible PLS functions that may be active per axis. Each of these PLS functions may be assigned a modulus value to apply to the reference value prior to window comparison. For example if 'fposc' is the reference and its value is 1025674 and the modules was 65536 (possibly pulses per revolution), then the actual reference used would be 1025674 % 65536 or 42634, the remainder.

syntax

`pls` *`index state`*

parameters

`index`	The index of the PLS assigned the output to control. 1-5 index
`state`	**`on`** or **`off`**

This statement enables ("on") or disables ("off") a PLS for an output active for the given index.

- **on** - Enables the pls functionality initialized for a particular output with the PLS Define statement.

- **off** – Disables the pls functionality initialized for a particular output with the PLS Define statement.

If the output is on when a PLS is disabled, it will remain on – unless the user re-enables the PLS (to re-compute the PLS output).

```
pls index using reference definitions
```

parameters

```
index          The index of the PLS assigned the output to control, 1 to 5.

reference      the encoder count scaled reference variable to compare to:
               fposc          Feedback position of axis msb
               mposc1 - mposc5 Master position counters #1
                                through #5
               mposc          Master position counter
               smodc          Slave position (modulo)
               smark          Slave marked position
               tmc1 tmc2      Temporary master counters #1 & #2
               tsc1 tsc2      Temporary slave counters #1 & #2
               sdc            Slave decrement counter
               fposc1         Feedback position of axis 1
                                (fposcA)
               fposc2         Feedback position of axis 2
                                (fposcB)
               tmodc          Temporary master counter mod mmc
               sfposc         Secondary feedback position of axis
               tposc          Target position of axis
               ctr0           din1 mapped input counter
               ctr1           din2 mapped input counter
               ctr2           din3 mapped input counter
               ctr3           din4 mapped input counter
               ctr4           din5 mapped input counter
               ctr5           Local quadrature encoder 1
               ctr6           Local quadrature encoder 2
               ctr7           Local quadrature encoder 3

definitions    a comma-separated list of up to 16 PLS definitions:

               on x to y       Turn output on when the reference
                               is within the bounds specified
                               by x through y (may be
                               expressions)
```

This statement defines, or redefines, a PLS associated with a given output and its operation. When a PLS is defined/re-defined it will be disabled and will not compute the state for the output. To enable a PLS after it is defined/re-defined, a *pls <index> on* statement must be issued.

Example:

```
[beginTest]

// Assign the first remote output on a Wago IO Block to PLS index 1.
set pls 1 to output 1001;
```

```
// Output will be on when fposc is within 100000-250000 or 1750000-3000000
pls 1 using fposc on 100000 to 250000, on 1750000 to 3000000;
// enable the PLS for index #1
pls 1 on;

// Assign the first remote output on a Wago IO Block to PLS index 1.
set pls 2 to output 1005;
// Output will be on when fposc is within10-200000 or 250000-1750000
pls 2 using fposc on 0 to 200000, on 250000 to 1750000;
// enable the PLS for index #1
pls 2 on;

[run]
// Begin the move, .05 rev/second for 2 revolutions
move at 0.05 for 2;
wait for in position;
// Delay 1 second once in position
delay 1000 ms;
// Do a relative move back 2 revolutions at .05 rev/second
move at 0.05 for -2;
wait for in position;
// Delay 1 second once in position
delay 1000 ms;
// Do it again, forever...
goto run;
```

Blank

CHAPTER

9

[9] Error Handling

Each drive manufacturer lists its own error codes. The combination of EtherCAT and the many different servo drives available results in thousands of different error possibilities. The network is continually monitored for error conditions and emergency messages from the slave devices. In the event of an error, the MSB typically performs a Quick Stop operation and the EtherCAT master stops scanning and awaits program RESTART. Power to the amplifier is turned off when scanning stops, causing the servo to freewheel. Any braking needed during this state should be considered.

Both informational and error conditions are constantly logged to a universal message buffer. The message buffer contents can be viewed in the QuickBuilder EtherCAT Explorer or by using telnet commands.

Retry Logic

EtherCAT networks are fairly robust but intermittently a packet can be lost due to noise affecting the cabling. The EtherCAT Master has built in retry logic to attempt a recovery before considering a lost packet a non-recoverable error. With poll times set to the standard 1 mS a retry will occur up to two times if a packet is not returned within 300 uS of transmission. At a poll time of 500 uS this is shortened to 200 uS. If a retry occurs an information message will be logged to alert the user. With the EtherCAT Master dynamically adjusting its scan timer, the Master will quickly re-sync to the reference slave clock if a retry is successful. Reference Chapter 8, the EtherCAT Explorer 'User Options', if the default retry logic needs to be modified.

Drive Diagnostic Variables and Registers

Special QuickBuilder variables and controller registers are available to monitor EtherCAT operation and provide post analysis after faulting. The best method is by the use of the EtherCAT Explorer but in some cases a remote HMI or user may want to access the information that the Explorer does, except at the error code/register level.

QuickBuilder MSB variables by axis:

dwSlaveID – The EtherCAT slave ID as it appears in the EtherCAT Explorer.

faulted – 0, no fault, 1 fault on this axis.
faultOpcode – Type of fault, reference following section for definition.
wStatus – Current PDO status read from the drive, object 0x6041.
wControlWord – Current PDO control command to the drive, object 0x6040, reference manufacturer manual.
errorType – Last error type logged.
errorRegister – Object 0x1001 error register in drive if supported, see appendix for drive manufacturer.
errorCode – Last error from drive, typically object 0x603F if supported, see appendix for drive manufacturer.
last_ALStatusCode – Additional error information from drive, object 0x134:0x135.

By Controller Register:

Register 13700 – Axis Display Index Register. Set to 0, default, 14XX0 block same as defined in the 'Model 5300 Quick Reference Register Guide', set to 1 and maps as follows where XX is the axis number starting with 00 for axis 1 allowing for up to 100 drives:

Register	QuickBuilder MSB Variable
14XX0	dwSlaveID
14XX1	faulted
14XX2	faultOpcode
14XX3	wStatus
14XX4	wControlWord
14XX5	errorType
14XX6	errorRegister
14XX7	errorCode
14XX8	last_ALStatusCode
14XX9	Not Available

MSB 'wStatus' Variable Bit Definitions

'wStatus' references the drive's last Status Word, object 0x6041, read. References the specific drive Manufacture for AL additional details:

Bit	QuickBuilder MSB Variable
0	Ready to switch on
1	Switched on
2	Operation enabled

3	Fault
4	Voltage enabled
5	Quick stop
6	Switch on disabled
7	Warning
8	-
9	Remote
10	Target reached
11	Internal limit active
12	Operation mode specific
13	Operation mode specific
14	Torque limit active
15	-

MSB 'errorType' Variable Value Definitions

The 'errorType' variable defines the type of error that has occurred as identified by the M3-41 module. This typically allows CTC to locate where it was during software execution when the error occurred.

errorType	Description
0	*INFORMATIONAL_ONLY*
1	*INFORMATIONAL_ONLY_MALLOCED*
2	*ERROR_DEFAULT*
3	*ERROR_ECAT_PROFILE_POS_INIT*
4	*ERROR_ECAT_PROFILE_POS_STARTING1*
5	*ERROR_ECAT_PROFILE_POS_STARTING2*
6	*ERROR_ECAT_PROFILE_POS_RUNNING*
7	*ERROR_ECAT_HOMING_INIT*
8	*ERROR_ECAT_HOMING_STARTING1*
9	*ERROR_ECAT_HOMING_STARTING1A*
10	*ERROR_ECAT_HOMING_STARTING2*
11	*ERROR_ECAT_HOMING_RUNNING*
12	*ERROR_ECAT_PROFILE_VEL_WAIT*

errorType	Description
13	*ERROR_ECAT_PROFILE_WAIT_QSTOP*
14	*ERROR_RUNNING*
15	*ERROR_TRACKING*
16	*ERROR_LOST_CONNECTION*
17	*ERROR_Network_Interface*
18	*ERROR_No_Slaves_Found*
19	*ERROR_Not_All_Slaves_PreOp*
20	*ERROR_Not_All_Slaves_Operational*
21	*ERROR_PDO_Init_Failed*
22	*ERROR_ec_config_map_Failed*
23	*ERROR_Slave_Unknown*
24	*ERROR_Init_Send_Processdata*
25	*ERROR_Init_Receive_Processdata*
26	*ERROR_No_Slaves_For_DC*
27	*WARNING_No_Station_Alias*
28	*ERROR_DC_SYNC0_Failed*
29	*ERROR_ECATLoop_Execution_Exceed_Scantime*
30	*ERROR_MEMORY_BUFFER_EXCEEDED*
31	*ERROR_ECATLoop_Event_Timeout*
32	*ERROR_ECATLoop_RX_Timeout*
33	*ERROR_ECATLoop_TX_Timeout*
34	*ERROR_USER_SDO_READ*
35	*ERROR_USER_SDO_WRITE*
36	*ERROR_Profile_Thread*
37	*ERROR_Motion_Fault*
38	*ERROR_Dump*
39	*ERROR_USER_DCSYNC*
40	*ERROR_Not_All_Slaves_SafeOp*
41	*ERROR_Not_All_Slaves_Found*
42	*ERROR_Slaves_Not_Match_Expected*
43	*ERROR_Duplicate_Axis*
44	*WARNING_ECATLoop_Execution_Exceed_Scantime*
45	*ERROR_Slave_ALState_NotOperational*
46	*ERROR_ec_config_map_Too_Many_Segments*

errorType	Description
47	*ERROR_Slave_Not_Synced*
48	*ERROR_Drive_PowerUP_Fault*
49	WARNING_Too_Many_Servos

MSB 'last_ALStatusCode' Variable Value Definitions

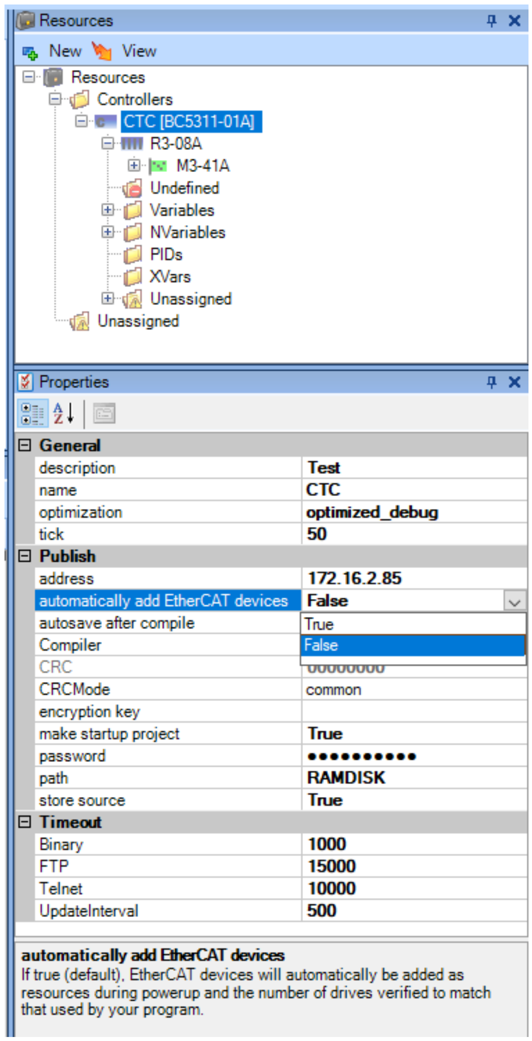
References the specific drive Manufacture for AL Status Codes that are not listed below or where additional definition is required. Codes are shown in hexadecimal representation:

AL Status Code	Description
0x0000	No error
0x0001	Unspecified error
0x0002	No Memory
0x0011	Invalid requested EMS change
0x0012	Unknown requested state
0x0013	Bootstrap not supported
0x0014	No valid firmware
0x0015	Invalid mailbox configuration
0x0016	Invalid mailbox configuration
0x0017	Invalid sync manager configuration
0x0018	No valid inputs available
0x0019	No valid outputs
0x001A	Synchronization error
0x001B	Sync manager watchdog
0x001C	Invalid Sync Manager Types
0x001D	Invalid Output Configuration
0x001E	Invalid Input Configuration
0x001F	Invalid Watchdog configuration
0x0020	Slave needs cold start
0x0021	Slave needs INIT
0x0022	Slave needs PREOP
0x0023	Slave needs SAFEOP
0x0024	Invalid Input Mapping
0x0025	Invalid Output Mapping

AL Status Code	Description
0x0026	Unmatched setting
0x0027	Free-run mode unsupported
0x0028	SYNC mode unsupported
0x0029	Free-run mode, 3 Buffer mode not set
0x002A	Background watchdog
0x002B	No valid inputs and outputs
0x002C	Fatal sync error
0x002D	No sync error
0x0030	Invalid DC Sync configuration
0x0031	Invalid DC Latch configuration
0x0032	PLL error
0x0033	Invalid DC IO error
0x0034	Invalid DC timeout error
0x0035	DC invalid Sync cycle time
0x0036	DC SYNC0 cycle time
0x0037	DC SYNC1 cycle time
0x0042	MBX_EOE
0x0043	MBX_COE
0x0044	MBX_FOE
0x0045	MBX_SOE
0x004F	MBX_VOE
0x0050	EEProm No Access
0x0051	EEProm Error?
0x????	Unknown error, reference manuf.

Automatically Add EtherCAT Devices Option (ERROR_NO_SERVO)

By default Incentive QuickBuilder application programs verify that the number of drives used by the program is the same or less than that which is currently fully operational on the EtherCAT network. If they are not the same or less an ERROR_NO_SERVO fault will occur and the program will not run. In addition all IO devices found on the network are appended to the local IO or other network IO already found. There are installations where it is desirable for QuickBuilder to be fully operational before EtherCAT or for example have the 5300 PLC local IO control power to drives and then bring up the EtherCAT network (and/or bring it down as well). This can be done by setting the “Automatically add EtherCAT devices” option false under the controller properties within QuickBuilder. This will put the responsibility of resetting the EtherCAT network and adding whatever drives and IO are found when the QuickBuilder application program wants to, not at startup.



When set to False it it necessary for the QuickBuilder program to reset the network (write 0x55AA to register 13464) and then once operational, tell QuickBuilder to install whatever drives and IO was found as well as load your MSB programs (write 0x55AB to register 13464). QuickBuilder will not check the number of drives available so the application program can do this by reading register 13465. If there are more MSB programs for multiple axes than there are drives available an ERROR_NO_SERVO fault will occur as it did when the QuickBuilder option was set to True. Below is example code for implementation of this feature.

```
foobar = 1;
…
init_network // Step
// Assume we are starting the network since we control the power to the drives and can come up with
// drives off. If more than one network we must get first network up first and then select second
// and get it up else the count will be messed up for servo's.  In theory this will work for IO on
// the EtherCAT network as well.
$REGISTERS[12333] = 1;  // Select first card
while (foobar != 0) repeat { };  // do what is needed and when this is clear then attempt restart
foobar = 1;
// Attempt to bring network up
```

```
while (1) repeat
{
      $REGISTERS[13464] = 0x55AA;     // Reset the network in case this is a new QB download.
                                      // Suggest setting User options on EtherCAT init retries to 2.
      loopcounter = 0;
      while ($REGISTERS[13464] == 0) repeat
      {
            delay 100 ms;
            loopcounter = loopcounter + 1;
            if (loopcounter > 300) then
            {
                 goto init_network;   // timeout and reset network again
            }
      }   // Wait for network to come up if not already
          delay 2000 ms;  // let network settle in case there is a drive issue
      if ($REGISTERS[13464] == 0) then goto init_network; // Network down...
      // Network is up so see if there are any drives available on the EtherCAT card
      if ($REGISTERS[13465] == 1) then goto init_motors; // Have one drive so all set...
}
```

init_motors // Step

```
// Initialize and load msb since network is up.  We do this manually since controlling power to
// motors and need local IO to turn power on and off.
$REGISTERS[13464] = 0x55AB;
delay 1000 ms;
goto motion1;  // Things should be good now to start
```

motion1 // Step

```
start ax1 Motion_loop BG;
```

This option is for servo only systems, no IO, and must be enabled or disabled by using the Incentive telnet command `disable EtherCAT automatic device installation` or `enable EtherCAT automatic device installation` . Followed by writing a 1 to register 20096 to save the configuration. The system would then be restarted to enable/disable the option for any futher power cycles. For volume distributions of Incentive PC the initialization file _Init.bin found in the C:_system\programs folder can be copied to other PC's to clone the settings. If not sure what the enable/disable flag is set to you can execute the `get EtherCAT automatic device installation` command.

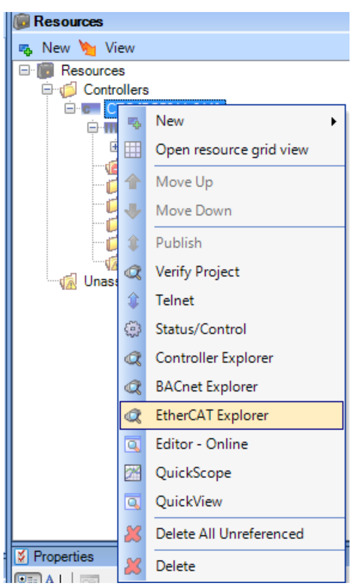
CHAPTER 10

[10] QuickBuilder EtherCAT Explorer

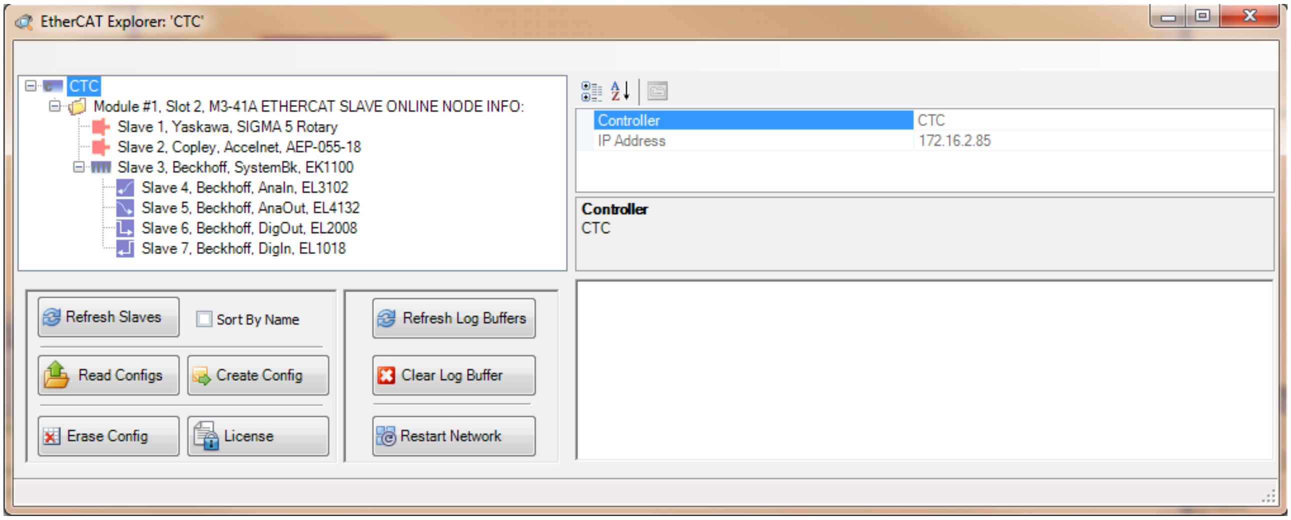
QuickBuilder provides a simple-to-use EtherCAT Explorer. The Explorer communicates directly with any model 5300 that has one or more M3-41 modules and graphically presents the network information. It also provides a high-level diagnostics capability. When using Incentive PC each EtherCAT Master process which is executing appears as a M3-41 module.

QuickBuilder EtherCAT Explorer Status Window

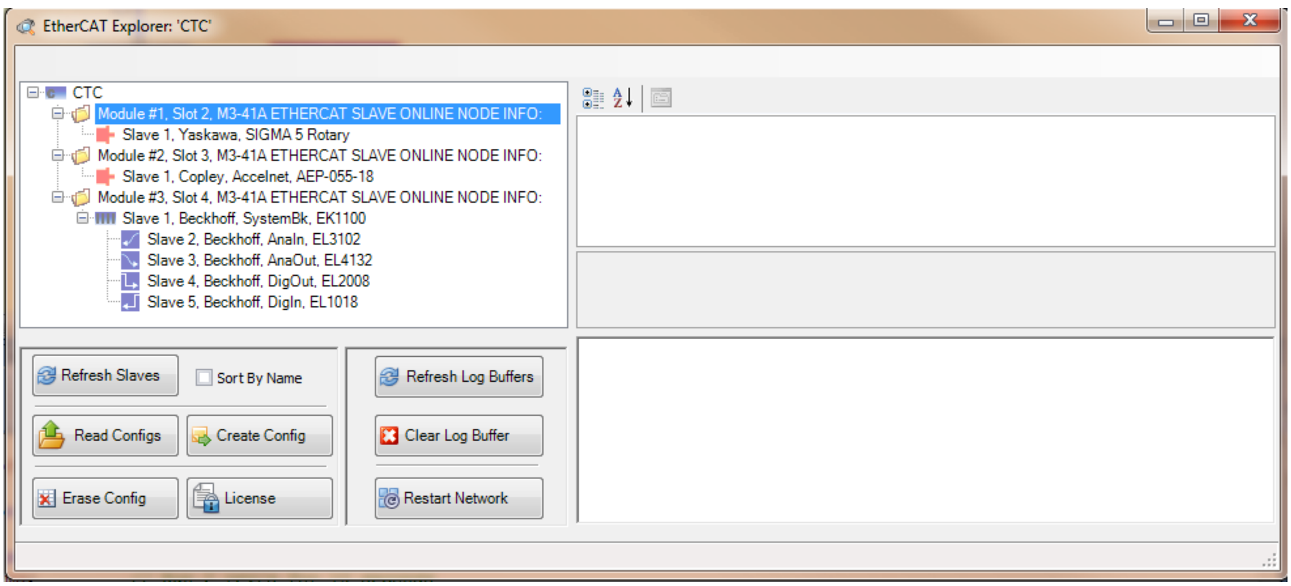
The EtherCAT Explorer window is a feature of the QuickBuilder environment. To open the EtherCAT Explorer, right click on the controller available in Resources. A menu of options will appear; select 'EtherCAT Explorer' to connect to the defined controller.



Once invoked, a window similar to the one below will appear, enabling you to monitor the model 5300's EtherCAT module and its Master configuration.

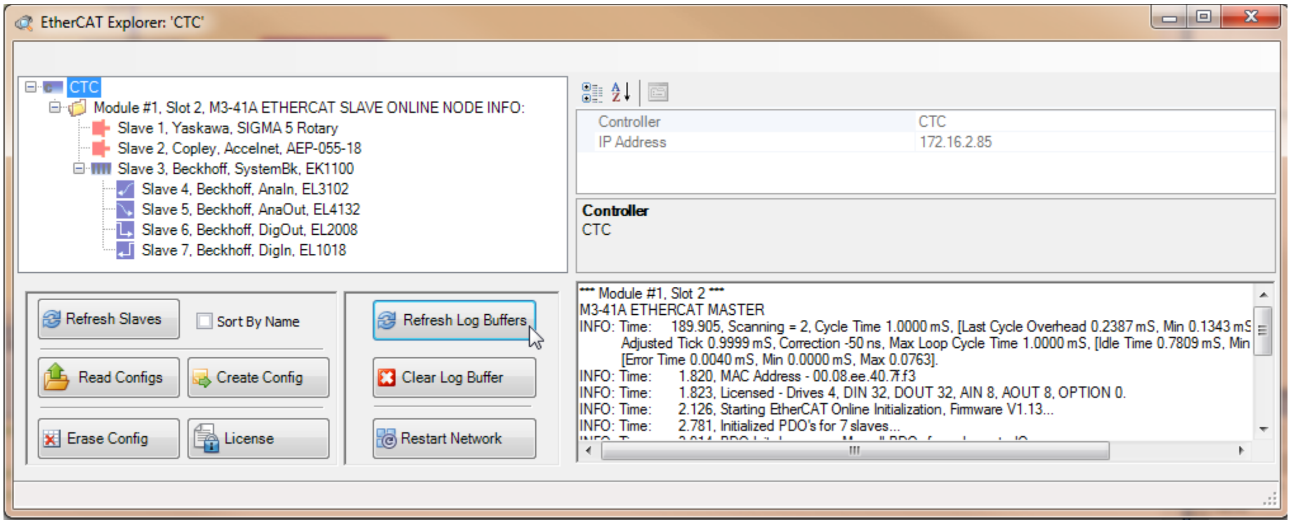


If multiple M3-41 modules (or IncentiveECAT) are present, a folder will appear for each module, with the slaves it controls listed below. Here is an example of three M3-41 EtherCAT Master networks in one model 5300 controller with the first M3-41 module selected (highlighted).



The top left tree is known as the Slave Discovery Window. Both online slaves and the expected slave configurations appear here. Select a slave entry, and the available property information appears within the window to its right.

The window on the bottom right is known as the Message Window. As the EtherCAT Master executes, diagnostic log information is stored in the M3-41 module. By selecting ‘Refresh Log Buffer’ the most current contents of the log buffer will be displayed. Note that at power up the Licensing information appears in the list, “Licensed: Drives 4…”. This is the total number of I/O and drives that your EtherCAT Master is authorized to control.

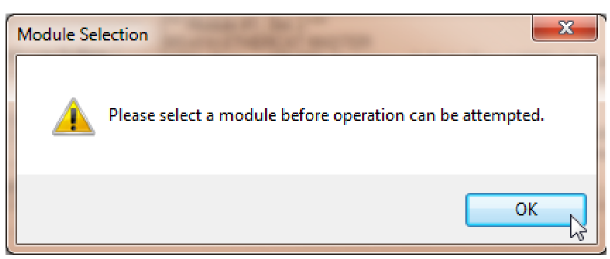


The panel on the lower left contains a number of buttons. Some are for global access; others are for individual M3-41 modules. Those operating an individual EtherCAT Master Module (M3-41) require the Slave Discovery Tree entry with the folder icon (Module #, Slot) to be selected to identify which module is to be accessed. The following operations are available:

Refresh Slaves – Updates the Slave Discovery Window with all online slaves observed at the last restart for all installed M3-41 modules. Each slave's properties are also refreshed to the most current. These properties vary by drive. Drive information will contain present PDO contents, position and state information, etc.

Read Configs – Updates the Slave Discovery Window with any saved configuration file whose content resides in the module's non-volatile storage for all installed M3-41 modules. The information displayed is what is required to be online for the network to become active.

Create Config – This button erases the non-volatile memory stored in the selected module and writes XML information matching the current online slave's to the EtherCAT Master module. This is an alternate approach to using an EtherCAT Configurator, such as Beckhoff's, allowing the configuration to be dynamically created from within QuickBuilder. The actual creation and storage is performed by the EtherCAT Master module, thereby requiring no file transfer. Note that this operation takes about 20 seconds to complete, because of the length of time required to erase non-volatile memory. The appropriate Module # must be selected from within the tree list prior to pressing the **Create Config** button, or an error message will be displayed:

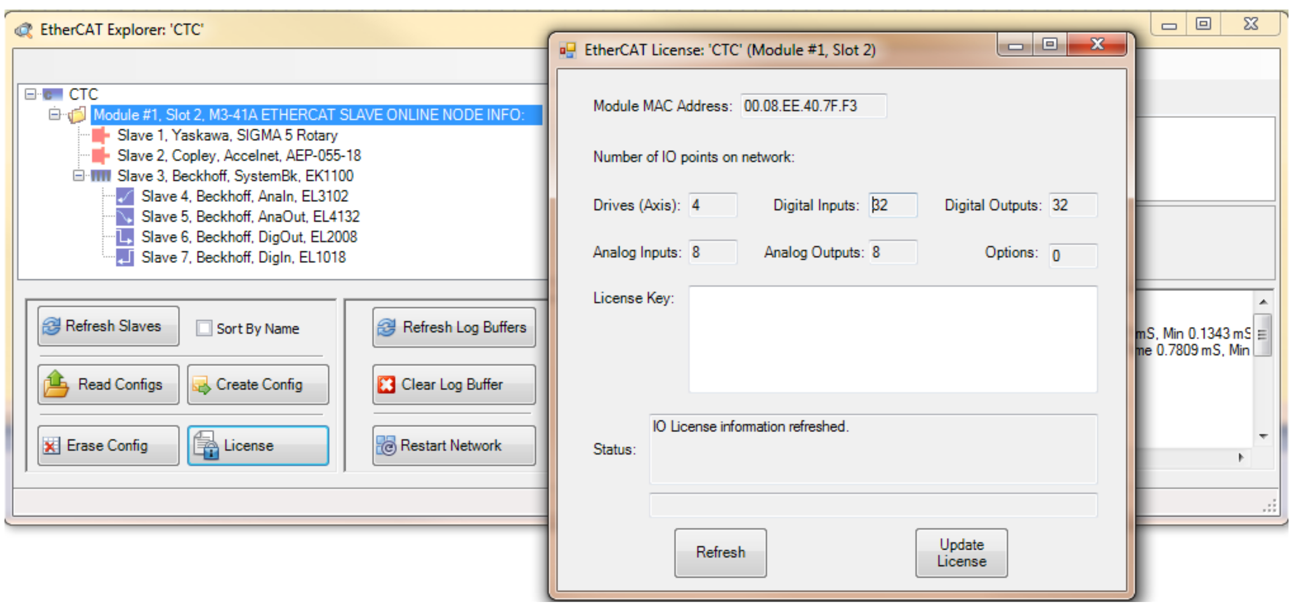


Note that the PC runtime stores this information in a file called _slaveConfig_[MAC Address].txt located in the _system\Programs directory.

Configuration files must be used in a production environment to ensure all the required devices are online prior to executing their controller MSBs. Differing devices power up at different times and may not initially respond to the EtherCAT Master online broadcast. Having a configuration file to compare against informs the master that it must wait for devices to come online prior to proceeding with the boot operation.

Erase Config – This button erases the current configuration file stored in the selected EtherCAT Master Module's non-volatile memory. By default, when no file is stored, and the network is restarted, no verification of online slaves occurs, and the controller begins operation with whatever devices and I/O are found on the network. This is known as Slave Discovery Mode and is useful when initially setting up a network. It can take up to 20 seconds for this command to complete. The appropriate Module # must be selected from within the tree list prior to pressing the **Create Config** button or an error message will be displayed.

License – This button displays the EtherCAT Master License form for the selected module. The MAC Address of the module appears along with the type and number of devices authorized for control by the master. New license keys can be purchased from CTC technical support, and entered within this form to change the current authorization.



Copy the license key you receive by email and paste it into the 'License Key" text box. Click the **Update License** button to update the number of I/O authorized. Click the **Refresh** button to verify the changes have been made. You must reboot the controller for the changes to take effect at the network level. Note that the PC runtime stores this information in a file called _ioOptions_[MAC Address].txt located in the _system\Programs directory.

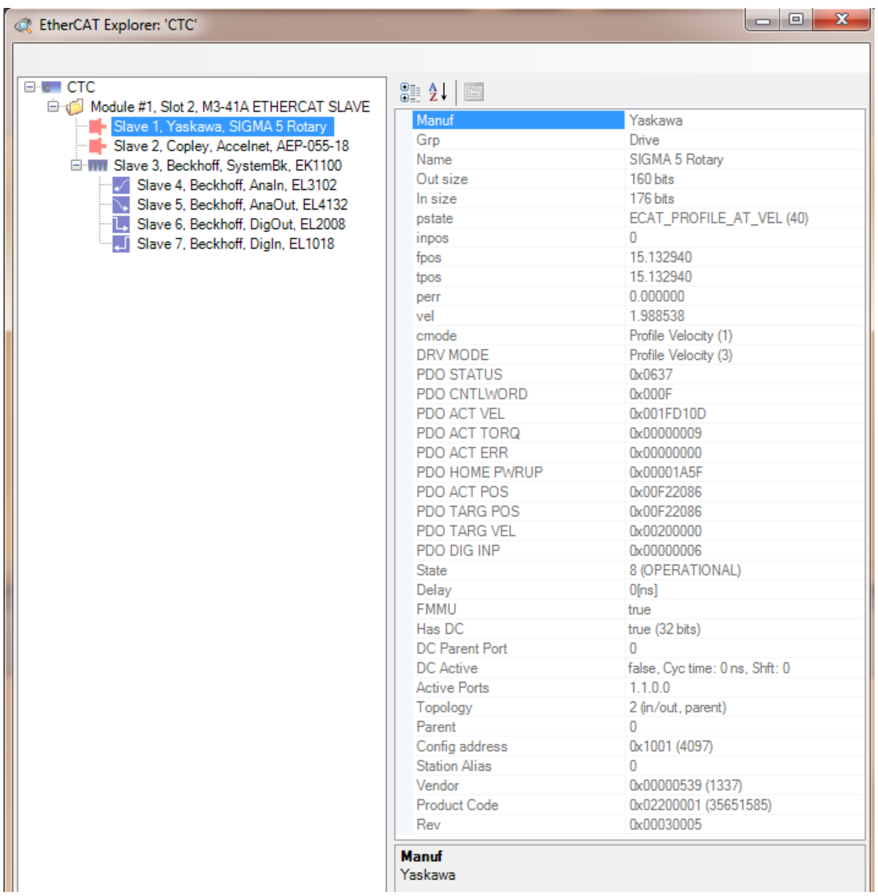
Refresh Log Buffers – This button displays log messages residing in all EtherCAT Master Modules in the lower right window. It is useful for diagnostic purposes.
Clear Log Buffer – This button clears the log messages for the selected EtherCAT Master module. Only new messages that occur after the clear operation will appear after the **Refresh Log Buffers** button is selected.

Restart Network – This button causes the selected EtherCAT Master module to re-scan the network and display whatever slaves are found. Prior scan results are overwritten. Available I/O in the controller will not be updated, and a reboot is required if the configuration changes. The network will not be available until after the restart is completed. Restarting the network is useful when connecting new slaves to the network or after power cycling a slave to verify that it is seen on the network. Note that MSBs will also restart.

Reboot Controller – This button causes the controller to be rebooted remotely. This is a hard reset and can take up to 30 seconds before the controller will be back online.

User Options – This button allows customization of the EtherCAT Master parameters, such as PDO cycle time, number of virtual drives, timeouts, and retries. Note that the PC runtime stores this information in a file called _options_[MAC Address].txt located in the _system\Programs directory.

EtherCAT Explorer Properties



Manuf - Manufacturer description

Grp – Group description

Name – Device Name

Out size – Number of bits the device consumes in the output Ethernet packet

In size – Number of bits the device consumes in the input Ethernet packet

Program Variables (MSB variables, if a drive):

pstate – present MSB program operational state
inpos – 1 if motor is in position, 0 if it is not
fpos – Current motor position in revolutions
tpos – Present target position in revolutions
perr – Present error (tpos – fpos)
vel – Present motor velocity in revolutions/second
cmode – MSB program commanded mode for the motor

DRV MODE – CANOpen DS402 mode the drive is in for motion control

PDO STATUS – Object 0x6041 representing the device state

PDO CNTLWORD – Object 0x6040 representing the currently written Control Word

PDO ACT VEL – Object 0x606C representing the current velocity

PDO ACT TORQ – Object 0x6077 representing the current torque. On some drives this is the Actual Current when torque is not available.

PDO ACT ERR – Object 0x60F4 representing the current servo position error

PDO HOME PWRUP – Power up position first seen by the EtherCAT Master

PDO ACT POS – Object 0x6064 representing the actual current position in increments

PDO TARG POS – Object 0x607A representing the target position in increments. This is relevant in Cyclic Sync Position and Profile Position modes. Interpolated motion mode uses 0x60C1 subindex 1.

PDO TARG VEL - Object 0x60FF representing the target velocity in increments/sec. This is only relevant in Profile Velocity mode.

PDO DIG INP – Drive Inputs as reported by the cyclic PDO scan

State – Last seen EtherCAT state of this device

Delay – Propagation delay, in ns, of this device as cabled on the network.

FMMU – True if Fieldbus Memory Management unit is bit oriented, false if it is not.

Has DC – Set to 1 if the drive can support being the source of the distributed clock. The EtherCAT Master selects the first device that 'Has DC' as the source of the clock and then periodically reads the time from that device and writes it to the rest of the slaves. The Master attempts to sync its internal clock to that device as well.

DC Parent Port – The slave named as the Distributed Clock master

DC Active – Information relative to DC Sync, if enabled

Active Ports – Each device can typically have up to 4 ports. This represents the ports being used on this device.

Topology – Each EtherCAT slave has up to 4 internal ports, each represented by a bit.

Parent – Set to the parent node. 0 means it is the parent.

Config address – EtherCAT assigned device address representing its place with regards to being cabled on the network. First device would be 0x1001.

Station Alias – Programmable station alias used to define axis numbers for MSB assignment

Vendor – Device vendor code

Product Code – Device product code

Rev – Firmware revision of the device

Log Buffer Timings

When the Log Buffer is viewed the first line after the module identification information contains internal timing information. This information can be critical in troubleshooting problems or possibly preventing them. The timing information contains the state of the EtherCAT network scanning, control loop overhead, idle, and slave sync timings. Below was observed in a six axis system do simple back and forth motion on all drives:

```
*** Module #1, Slot 4 ***
M3-41A ETHERCAT MASTER
INFO: Time:     275.943, Scanning = 2, Cycle 1.0000 mS, [Overhead 0.3180 mS, Min 0.1580 mS, Max 0.3995 mS, Avg 0.2951 mS],
        Adjusted Tick 1.0000 mS, Correction -6 ns, Max Tick 1.0000 mS, [Idle 0.6640 mS, Min 0.5234 mS, Max 0.8462].
        [Sync Error 0.0002 mS, Min 0.0000 mS, Max 0.0730, Avg 0.0022].
```

The following is how to interpret these timings:

[Scanning = 2] – The EtherCAT network scanner has three possible states.

- Initializing, 0.
- Scanning but for initial sync, 1.
- Online and executing, 2.

[Cycle 1.0000 mS] – Network control loop scan time, typically 1 mS or 500 µS.

[Overhead] – The time needed to process the PDO packet from the slaves, calculate the new trajectories, update IO, and prepare the PDO packet for transmission. The Cycle time minus the Overhead time is how much time the rest of the system has to execute. The Overhead may never be greater than the Cycle time. If it occurs once in a while and is less than 50 µS a warning will result and recovery attempted, otherwise a fault error. The first time listed is the time for the last completed cycle.

[Adjusted Tick] – The time that the FPGA timer was last set to, on the last cycle. This will shift slightly to sync with the reference slave.

[Correction] – The amount of correction added to the last time cycle in order to more closely sync to the reference slave.

[Max Tick] – The maximum time that the FPGA timer was set to for its periodic interrupt.

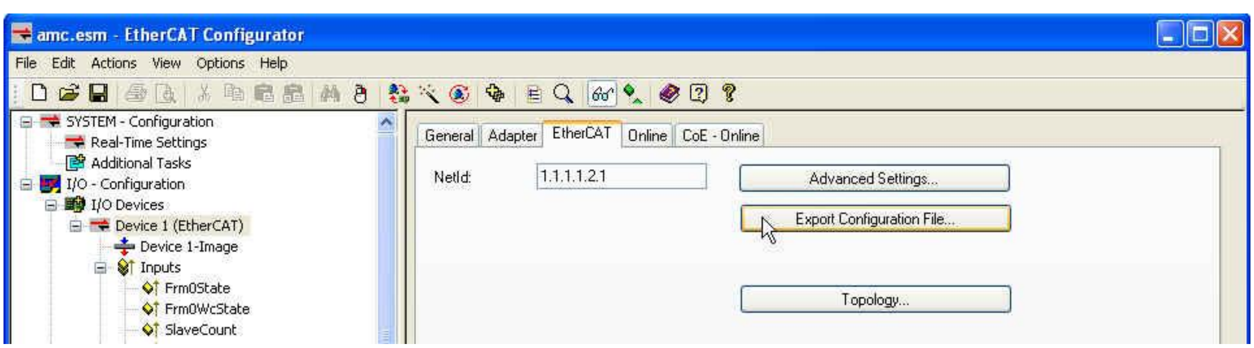
[Idle] – The amount of idle time available for the rest of the system to run, with the current time for the last completed cycle listed first.

[Sync Error] – This is the amount of error or jitter that the master has experienced while attempting to sync to the slave reference. The first listed time is from the last completed control loop cycle. Note that the average jitter is only 2.2 µS with a maximum of 73 µS. The 73 µS only lasts for a single cycle as corrections are applied.

EtherCAT Master ENI Configuration Files

Standard EtherCAT configuration files can be generated either automatically, via the **Create Config** button of the QuickBuilder EtherCAT Explorer, or through third-party tools such as Beckhoff's TwinCAT or EtherCAT Configurator (5300 M3-41 only). The file format stored in the M3-41 is standard XML.

To generate a configuration file using Beckhoff's EtherCAT Configurator, first create a network of the desired configuration and then select (as shown below) to export to an XML file (I/O-Configuration->I/O Devices->Device 1 (EtherCAT) followed by the EtherCAT tab and the **Export Configuration File** option button):



The saved file must be renamed to M341ACFGV0100.xml and placed in the model 5300 controller's _system->Firmware directory. You may set the V#### part of the file name to anything you desire, but the first part must be M341ACFG. Use the standard firmware update commands to load the file into a module. 'update M341ACFGV0100.xml' for all modules in the rack, or 'force update slot # M341ACFGV0100.xml' for a specific slot. Note that the 'force update' command is also needed if the M3-41 module I/O/Drives are not online. The 'update' command only works when the card is fully operational and the I/O/Drives have been added by the controller.

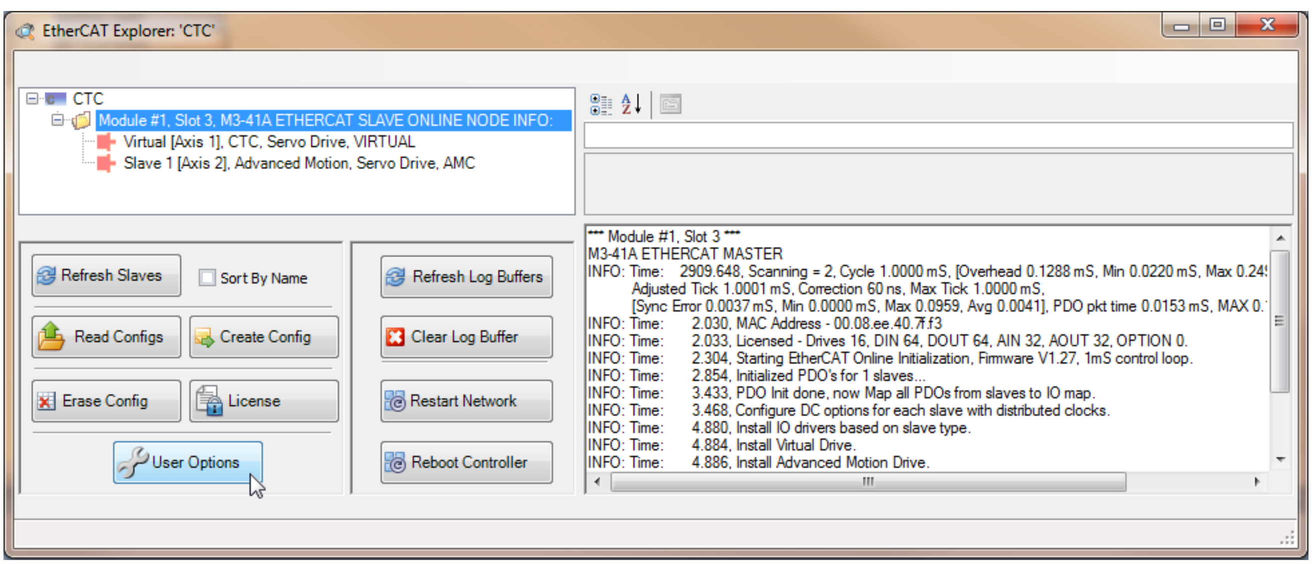
The configuration file size limit is 1,572,860 bytes. For larger configuration files, you must use QuickBuilder to create the configuration file, which is the preferred method and supported by both IncentiveECAT and the M3-41 module. When using QuickBuilder simply have the desired network online, press the 'Create Config' EtherCAT Explorer button and the configuration file will be saved. Any further attempts to start the network will be validated against this file.

User Options

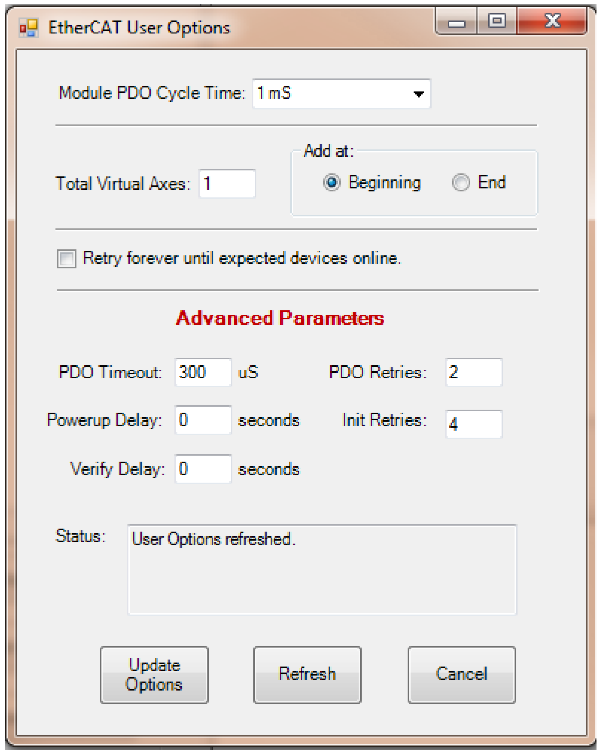
The EtherCAT Explorer allows the user the ability to customize the EtherCAT Master's operation. The customization currently supported consists of:

- Master PDO cycle loop times of 500 µS, 1 mS, 2 mS, or 4 mS.
- Automatic virtual axis creation.
- Capability to add the virtual axis to the end or beginning of those drives online.
- Retry forever option.
- PDO Timeout & retries.
- Initialization retries.

Invoking the 'User Option' form is done by clicking that button within the EtherCAT Explorer:



Once invoked the currently programmed options will appear:



Module PDO Cycle Time – This option sets the EtherCAT master control loop cycle time. The default of 1 mS is typically fine but in some situations the user may wish to speed up or slow down the loop. For example if the system is heavily loaded a 2 or 4 mS control loop will work well with most drives. Selections of 500 µS, 1 mS, 2 mS, and 4 mS are available.

Total Virtual Axes – This option sets the number of virtual axis to 'Add at' the 'Beginning' or 'End' of the online drive list. A virtual axis runs an MSB just like an online axis except that its feedback position (fpos) is updated automatically to its incremental tpos on each control loop, thereby simulating motion. The Virtual Axis is reported to the QuickBuilder as a normal axis.

Retry forever – This option, when selected, will cause the controller and EtherCAT network not to boot until the stored online configuration is observed. If the option is not selected then the 'Init Retries' parameter within the Advanced Parameters will be referenced and that many retries attempted prior to reporting the fault state to the controller and aborting operation.

PDO Timeout – This option should not be set unless instructed by CTC technical support. It will automatically be optimized to the proper setting based upon the PDO Cycle time selected. The option is the amount of time that the EtherCAT Master will wait for the response to the cyclical PDO packet transmission. PDO Timeout X PDO Retries should be less than the cycle time to ensure no DC Sync errors.

PDO Retries – This option should not be set unless instructed by CTC technical support. It will automatically be optimized to the proper setting based upon the PDO Cycle time selected. The option is the number of PDO Timeouts that are allowed before aborting operation and faulting. PDO Timeout X PDO Retries should be less than the cycle time to ensure no DC Sync errors.

Powerup Delay – This option defines how long, in seconds, the M3-41 module should wait, after power up, prior to beginning its identification of network slaves and initializing the EtherCAT network. It is useful when attempting to prevent timeouts on equipment that may take a long time to power up and come online.

Init Retries – This option is the number of times the EtherCAT Master will attempt to activate the network and initialize devices. If a Network Configuration is saved and those devices observed online do not match the network will be re-initialized and scanned again, with this count decremented by 1. Once a count of 0 is reached the module will abort, fault, and report an error.

Verify Delay – This option sets the amount of time, in seconds, the M3-41 module should delay after it identifies all required slave devices online (INIT state) and initializes their PDO's (PRE_OP state). After the delay occurs one more INIT cycle will be done and delay prior to updating the PDO mappings within the slave and marking the devices as online. This is required in some networks where slaves can report that they are online and ready but in fact other equipment is powering up or the slave still needs a small amount of time to continue initialization.

Available buttons:

Update Options – Clicking this button will cause the module to be set to the settings currently displayed. The status window will display the results of the operation.

Refresh – Clicking this button will cause the form to be updated with the options currently programmed within the module.

Cancel – Clicking this button will close the form, without changes and return to the EtherCAT Explorer form.

Blank

CHAPTER

11

[11] Telnet Commands

The model 5300 has an administrative mode that can be accessed via standard telnet. The use of telnet is beyond the scope of this manual and discussion here is limited to the specific commands available that affect the M3-41 EtherCAT module. Many of the commands are the same as those used by the QuickBuilder EtherCAT Explorer.

Status Commands

***get ethercat info all* –** Displays all available information for online slave devices, including mailbox information

***get ethercat info summary* –** Displays all available information for online slave devices, less the mailbox information. This is what is used by the QuickBuilder EtherCAT Explorer.

***get ethercat slave coe <slave #>* -** Displays the current PDO mapping and mailbox information for a specific slave

Network Commands

***restart ethercat networks* –** Restarts all EtherCAT networks. I/O and drives should remain the same. If changes are made to any network devices, cycle power rather than using this command. Otherwise the new modules will not be usable by the controller or a configuration error could occur.

***restart ethercat network slot <slot #>* –** This command will restart only the EtherCAT network for the M3-41 module in the specified slot, I/O and drives should remain the same. If changes are made to any network devices cycle power rather than using this command. Otherwise the new modules will not be usable by the controller or a configuration error could occur.

***get ethercat mac address* slot <slot # or -1 for first> -** Retrieves the MAC Address of the EtherCAT Master module

Message Log Commands

***clear ethercat messages* –** Erases the message log buffer on all M3-41 EtherCAT modules

***clear ethercat messages slot <slot # >* –** Erases the message log buffer for only the M3-41 EtherCAT module in the specified slot, with 1 being the first

***get ethercat messages* –** Displays all the message log buffers of all M3-41 EtherCAT modules installed.

Configuration File Commands

***erase ethercat config files* –** Erases the configuration files stored in the M3-41 modules. If the controller is rebooted, the network will be used without verification.

***erase ethercat config file slot <slot #>* –** Erases the configuration file only in the M3-41 module at the specified slot, with 1 being the first. If the controller is rebooted, the network will be used without verification.

***generate ethercat config files* –** Generates an XML file of the existing online slaves and stores it to serial flash memory of all M3-41 modules. Upon re-start, this file will be read and used to verify the slave devices found on the network. Operation will not begin until a full match is found. The use of this command greatly simplifies using a third-party configurator: connect all the slaves to the EtherCAT Master; confirm they are present with the 'get ethercat info summary' command; and use this command to save the configuration.

***generate ethercat config file slot <slot #>* –** Generates an XML file of the existing online slaves and stores it to serial flash memory for only the M3-41 module in the specified slot. Upon re-start, this file will be read and used to verify the slave devices found on the network. Operation will not begin until a full match is found. The use of this command greatly simplifies using a third-party configurator: connect all the slaves to the EtherCAT Master; confirm they are present with the 'get ethercat info summary' command; and use this command to save the configuration.

get ethercat expected info – Displays any saved configuration on the M3-41 module. It will be displayed in the 'get ethercat info summary' format containing information such as the VendorID, Product Code, etc.

Firmware Update Commands

***update <Filename>* -** The 'update' command is the standard command supported by the model 5300 to update online firmware modules. The file should reside in the /_system/Firmware directory and that should be made current using the change directory command, 'cd'. The M3-41 module supports two different files: an .xml used as a master configuration file, as exported from a program such as TwinCat, and a .bin file to update the module firmware. The file naming convention is fixed and must use the following format:

> M341ACFGV0100.xml – used as the configuration file. The V0100 represents the version number and can be anything desired.
>
> M341ASOMV0100.bin – used as the module firmware file. The V0100 represents the version number and can be anything desired.

***fupdate slot <#> <Filename>* -** The 'fupdate' command is the standard command supported by the model 5300 to update firmware modules that are offline or online. It is considered a forced update to a specific slot, regardless of module type. The <#> represents the slot number, starting at 1, with the <Filename> convention the same as for the 'update' command.

'update' will only work if the module is fully operational and devices are online. If not online and the controller is faulted, use the 'fupdate' command.

The equivalent of 'update' will also occur if a file is dragged and dropped onto the root 5300 directory via Internet Explorer.

License Commands

***get ethercat IO enabled slot <slot # or -1 for first>* -** Used to display the current licensed I/O totals authorized for use on the EtherCAT Master module.

***set ethercat IO enabled slot <slot # or -1 for first> <Encrypted Key>* -** Used to set a new authorization license total within the EtherCAT Master module.

Blank

Appendix

A

[A] ABB

ABB manufactures a number of drives. The drive currently supported is the single axis Microflex e150 servo drive. This section provides information that may be specific to this manufacturer.

eCAT_driveType – 12

DC Sync

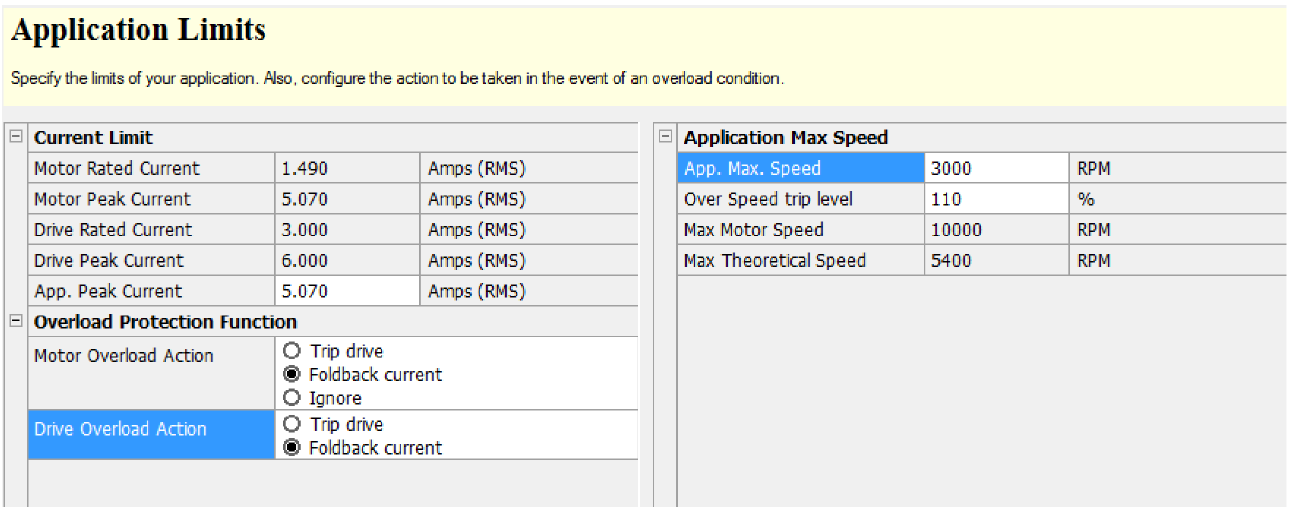
ABB drives must have DC Sync0 enabled prior to enabling the drive as well as a 32 or 64-bit slave reference as the first node on the EtherCAT network. Reference the 'dcsync' command example within the main body of this manual.

MSB code sample prior to drive enable:

```
// This is only needed for ABB & Emerson/Control Techniques
delay 2000 ms;  // needed in case restart so syncs when cycle DC Sync on/off.
// Cycle time is 1 ms, start it 100 ms in the future.
// Note that we need to make sure that the first slave device in the EtherCAT
// Network supports 32/64-bit distributed clocks for this to work properly.
// Thus far that is Beckhoff, Wago, ABB, and Sanyo Denki.
dcsync -1, 1000000, 0, 0, 100000000;
delay 200 ms;     // starts 100 milliseconds into the future

/**************** ENABLE DRIVE *******************/
[Drive_Enable]
// Issue any hardware enable output commands
//
// Power up the drive amplifier
drive enable;
```

Drive Configuration

When configuring the drive and application using the ABB Mint Workbench a few parameters should be modified from the factory defaults. The first is the Application Maximum Speed and the next is the Motor & Drive Overload Action being set to 'Foldback current'. Failure to modify the overload action can result in a 'Following Error' as the drive attempts to increase torque to attain the commanded position from the EtherCAT Master.



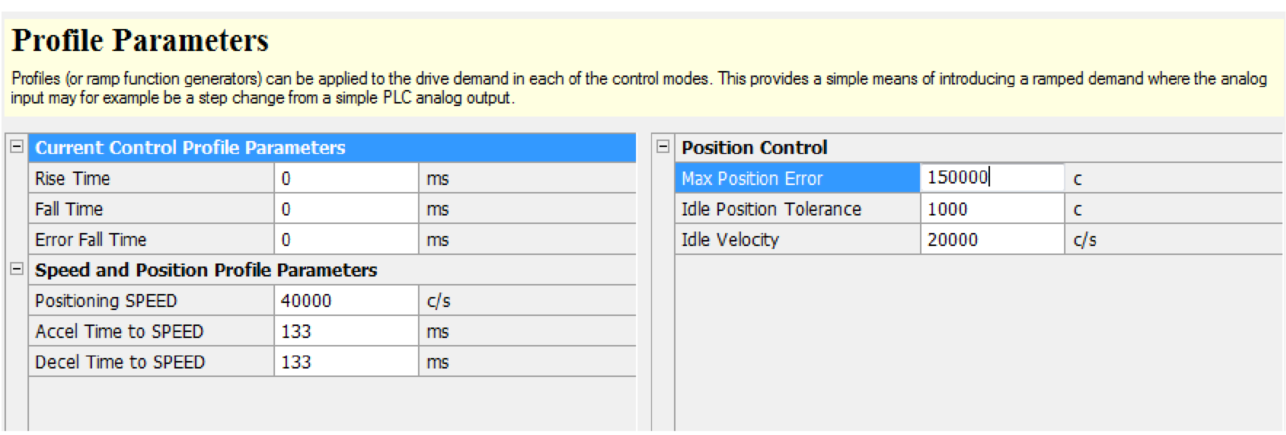
Application Limits

Specify the limits of your application. Also, configure the action to be taken in the event of an overload condition.

Current Limit		
Motor Rated Current	1.490	Amps (RMS)
Motor Peak Current	5.070	Amps (RMS)
Drive Rated Current	3.000	Amps (RMS)
Drive Peak Current	6.000	Amps (RMS)
App. Peak Current	5.070	Amps (RMS)
Overload Protection Function		
Motor Overload Action	○ Trip drive ◉ Foldback current ○ Ignore	
Drive Overload Action	○ Trip drive ◉ Foldback current	

Application Max Speed		
App. Max. Speed	3000	RPM
Over Speed trip level	110	%
Max Motor Speed	10000	RPM
Max Theoretical Speed	5400	RPM

Another parameter that must be changed is the acceptable position error. This is dependent upon the 'ppr' of the drive. The default is typically around 1K counts but with a 512K/rev ppr this is a very small value and can happen quite easily. Below shows a setting of 150K pulses. This is how much the commanded EtherCAT Master position can lag before the drive issues a 'Following Error'. A large lag is acceptable since the drive will catch up the commanded position prior to stopping.



Profile Parameters

Profiles (or ramp function generators) can be applied to the drive demand in each of the control modes. This provides a simple means of introducing a ramped demand where the analog input may for example be a step change from a simple PLC analog output.

Current Control Profile Parameters		
Rise Time	0	ms
Fall Time	0	ms
Error Fall Time	0	ms
Speed and Position Profile Parameters		
Positioning SPEED	40000	c/s
Accel Time to SPEED	133	ms
Decel Time to SPEED	133	ms

Position Control		
Max Position Error	150000	c
Idle Position Tolerance	1000	c
Idle Velocity	20000	c/s

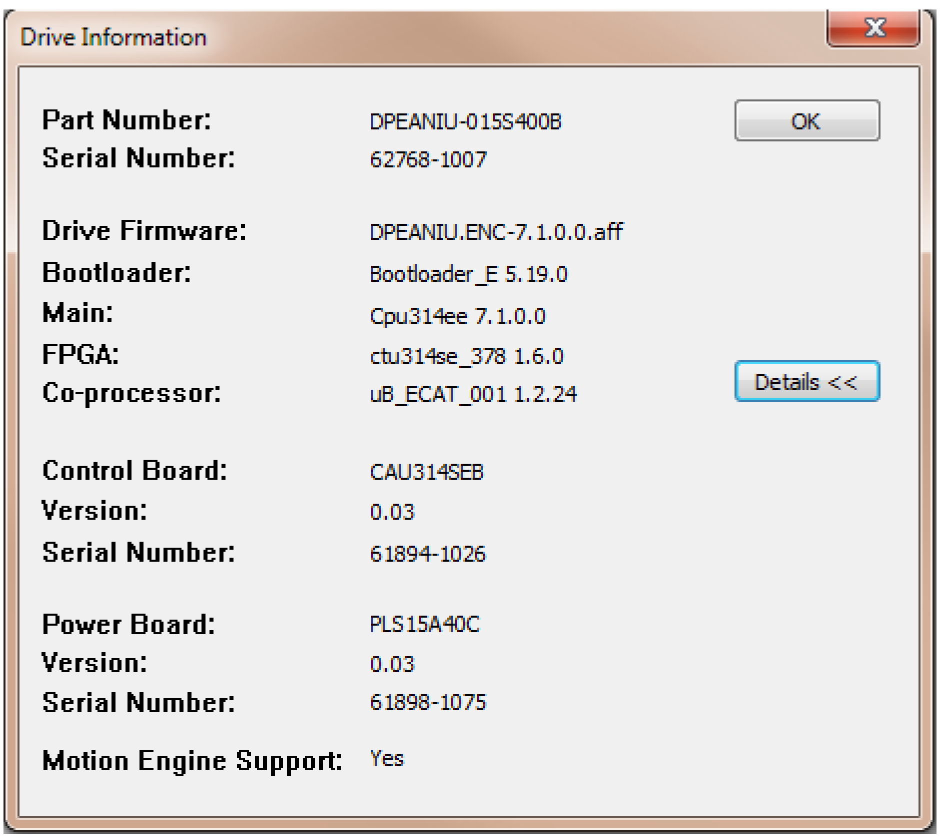
Appendix B

[B] Advanced Motion Controls

Advanced Motion Controls (AMC) manufactures a number of drives. The drive currently supported is the single axis DigiFlex servo drive (DPE series). This section provides information that may be specific to this manufacturer.

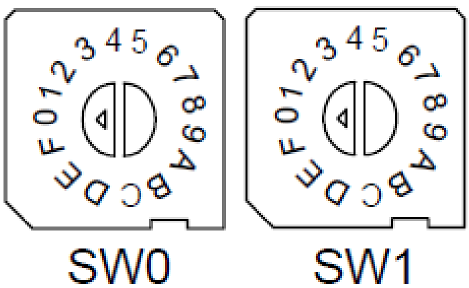
eCAT_driveType – 8

Drive Information & Firmware



Station Alias

In an EtherCAT network, slaves are automatically assigned addresses based on their position in the bus. When a device, such as a drive, must have a fixed assigned identification that is independent of cabling, a Station Alias is needed. Advanced Motion provides two 16-position rotary switches with hexadecimal encoding for this purpose. This allows for a setting of 0 to 255 (FFh), where 0 defaults to the automatic address assignment. As an example, if SW0 is set to a 1 and SW1 to an A this would be 1Ah or 1 X 16 + 10 = 26. Since the M3-41 only supports up to 16 drives SW0 would always be set to 0 and only SW1 used. Setting both switches to 0 defaults to automatic addressing.



SW1	SW0	Node ID
0	0	000
0	1	001
0	2	002
...	...	...
F	D	253
F	E	254
F	F	255

DC Sync

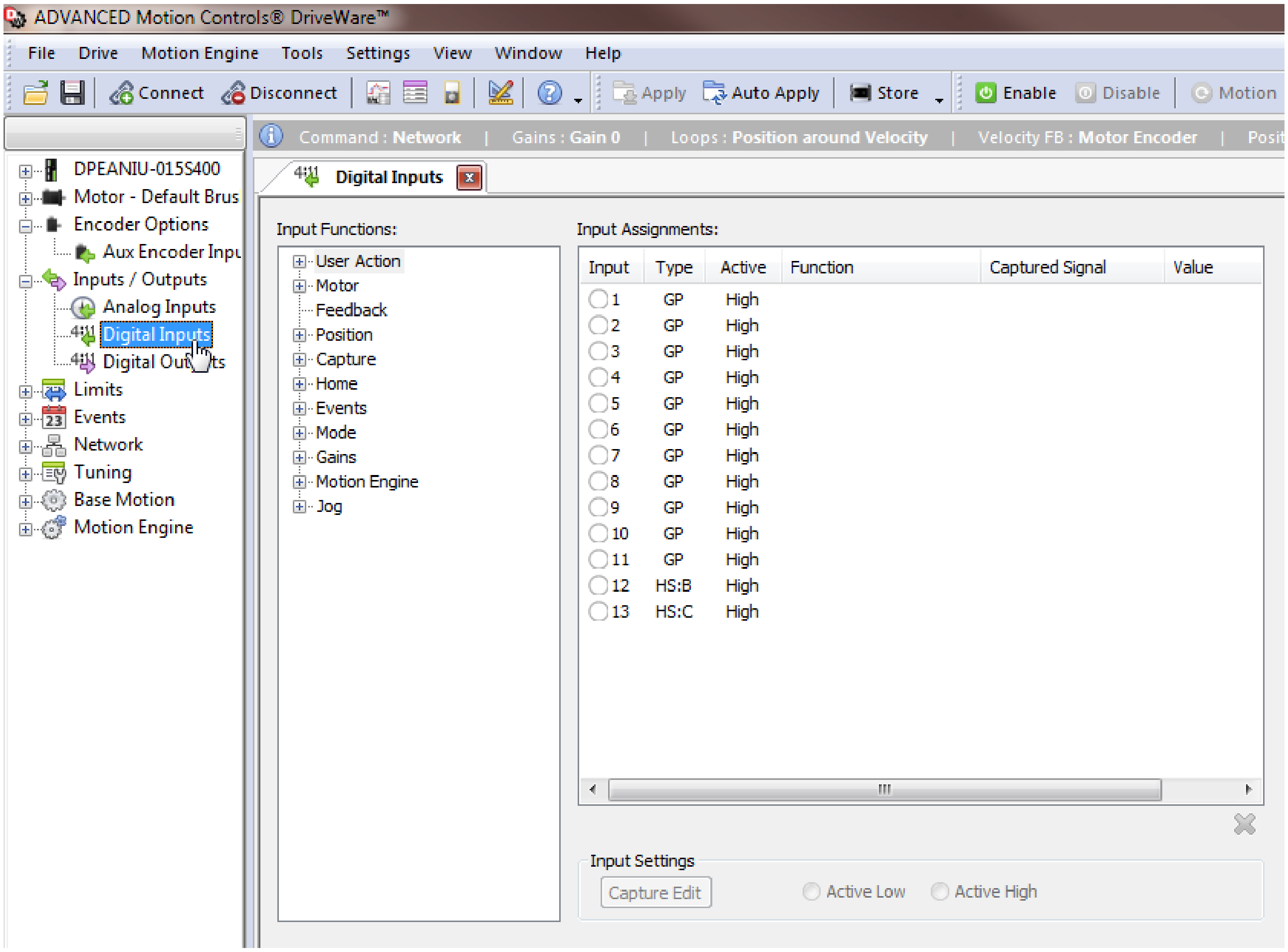
Advanced Motion Controls drives previously did not support DC Sync but recent updates have enabled this function on its newer drives.

Inputs/Outputs

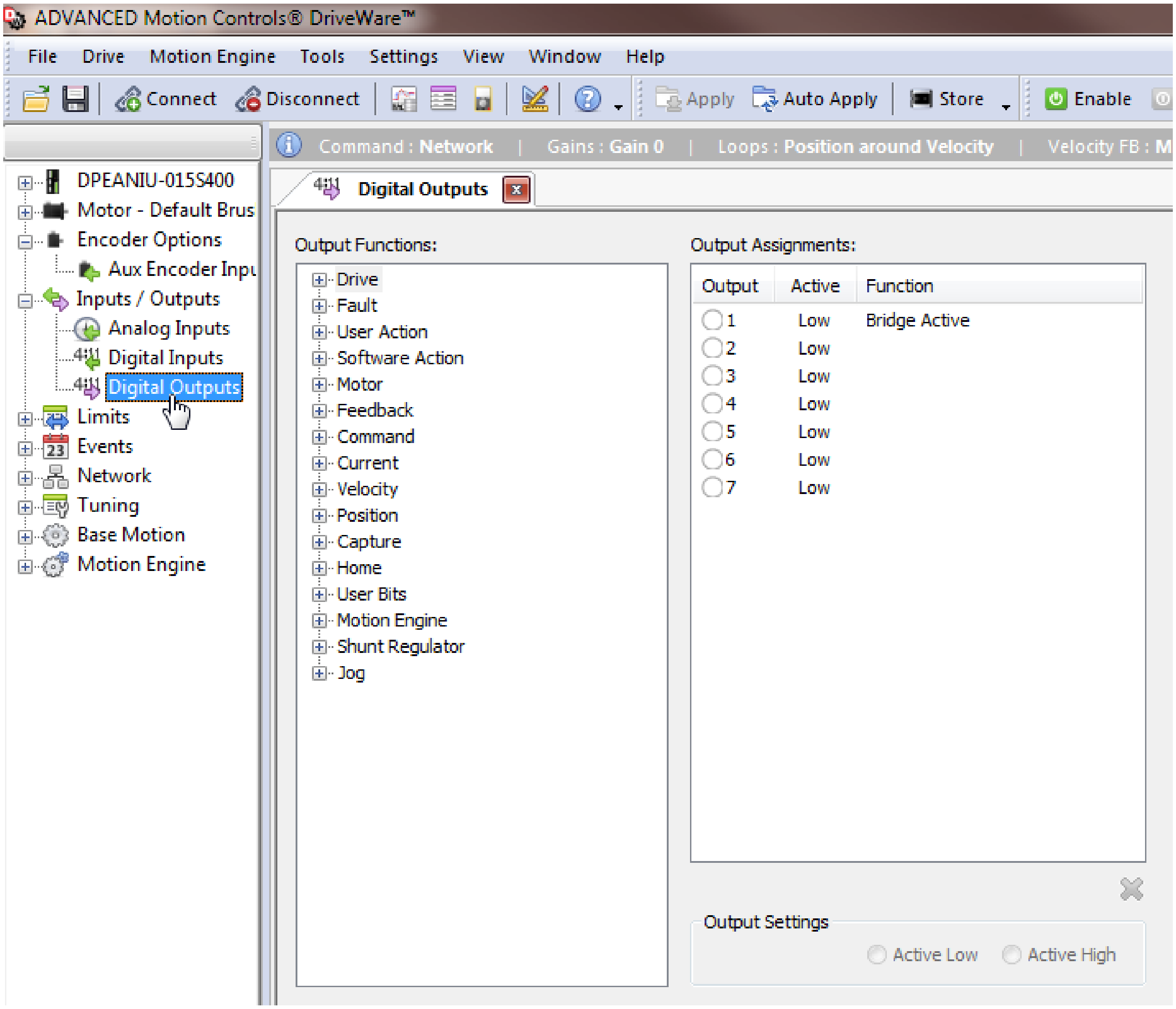
The inputs and outputs of the AMC drive are highly configurable and it is left to the user as to how they should be set up and used. The MSB 'din' variable maps to object 0x2023.1 (Digital Input Value) and 'dout' maps to object 0x2001.3 (User Bit Control). These objects are part of the PDO update cycle and refreshed every millisecond.

DriveWare can be used to configure the functionality of the inputs and outputs. (Make sure you apply the changes to the drive and save the modified configuration to non-volatile memory.)

Inputs:

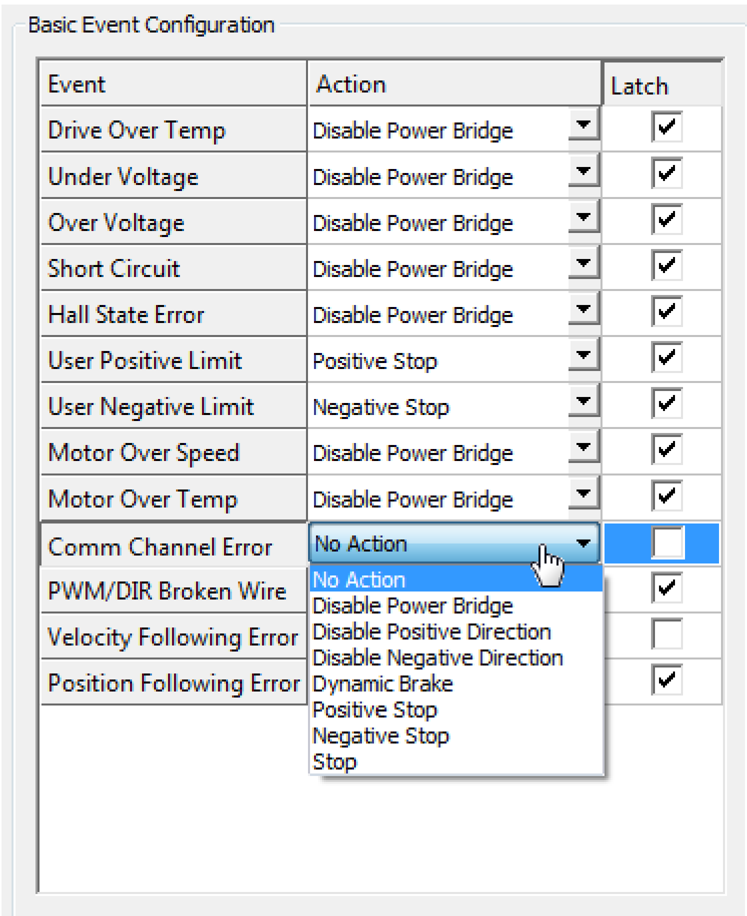

ADVANCED Motion Controls® DriveWare™
File Drive Motion Engine Tools Settings View Window Help
Connect Disconnect Apply Auto Apply Store Enable Disable Motion
Command : Network | Gains : Gain 0 | Loops : Position around Velocity | Velocity FB : Motor Encoder
DPEANIU-015S400
Motor - Default Brus
Encoder Options
Aux Encoder Inpu
Inputs / Outputs
Analog Inputs
Digital Inputs
Digital Outputs
Limits
Events
Network
Tuning
Base Motion
Motion Engine
Digital Inputs
Input Functions:
User Action
Motor
Feedback
Position
Capture
Home
Events
Mode
Gains
Motion Engine
Jog
Input Assignments:
Input Type Active Function Captured Signal Value
1 GP High
2 GP High
3 GP High
4 GP High
5 GP High
6 GP High
7 GP High
8 GP High
9 GP High
10 GP High
11 GP High
12 HS:B High
13 HS:C High
Input Settings
Capture Edit
Active Low
Active High

Outputs:


ADVANCED Motion Controls® DriveWare™
File Drive Motion Engine Tools Settings View Window Help
Connect Disconnect Apply Auto Apply Store Enable
Command : Network | Gains : Gain 0 | Loops : Position around Velocity | Velocity FB : M
Digital Outputs
DPEANIU-015S400
Motor - Default Brus
Encoder Options
Aux Encoder Inpu
Inputs / Outputs
Analog Inputs
Digital Inputs
Digital Outputs
Limits
Events
Network
Tuning
Base Motion
Motion Engine
Output Functions:
Drive
Fault
User Action
Software Action
Motor
Feedback
Command
Current
Velocity
Position
Capture
Home
User Bits
Motion Engine
Shunt Regulator
Jog
Output Assignments:
Output Active Function
1 Low Bridge Active
2 Low
3 Low
4 Low
5 Low
6 Low
7 Low
Output Settings
Active Low Active High

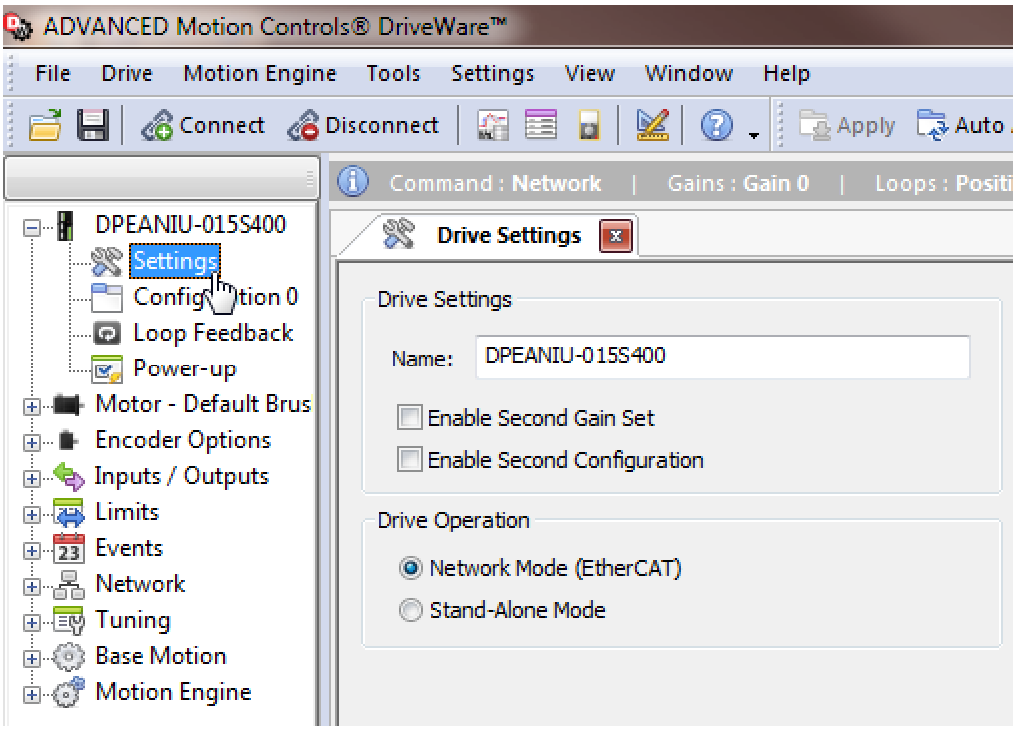
Communications Error

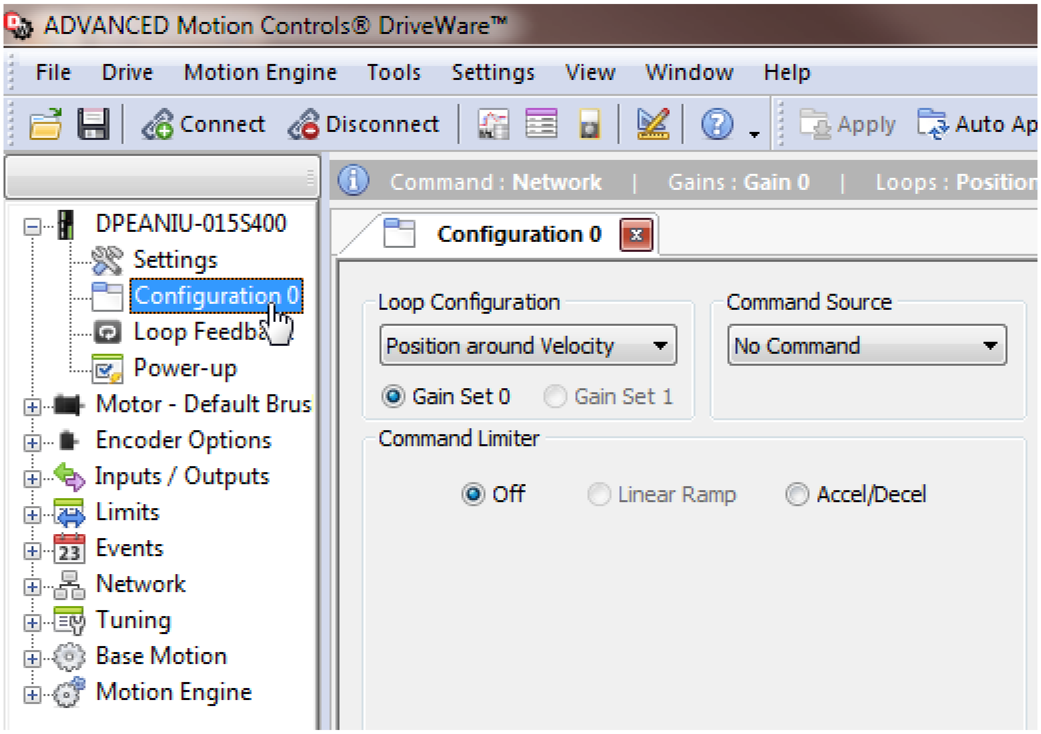
Typically when a communication error occurs it is ignored by the AMC drive. When running EtherCAT this is not acceptable — it means the M3-41 has lost control of the drive. DriveWare allows a very detailed configuration of what should happen when an Error Event occurs. If the "Comm Channel Error" check box is not selected, the M3-41 module automatically enables it at the drive level with the "Disable Power Bridge" setting. If the user selects some other setting within DriveWare (other than No Action) this setting will override that of the M3-41 module and become the default setting.

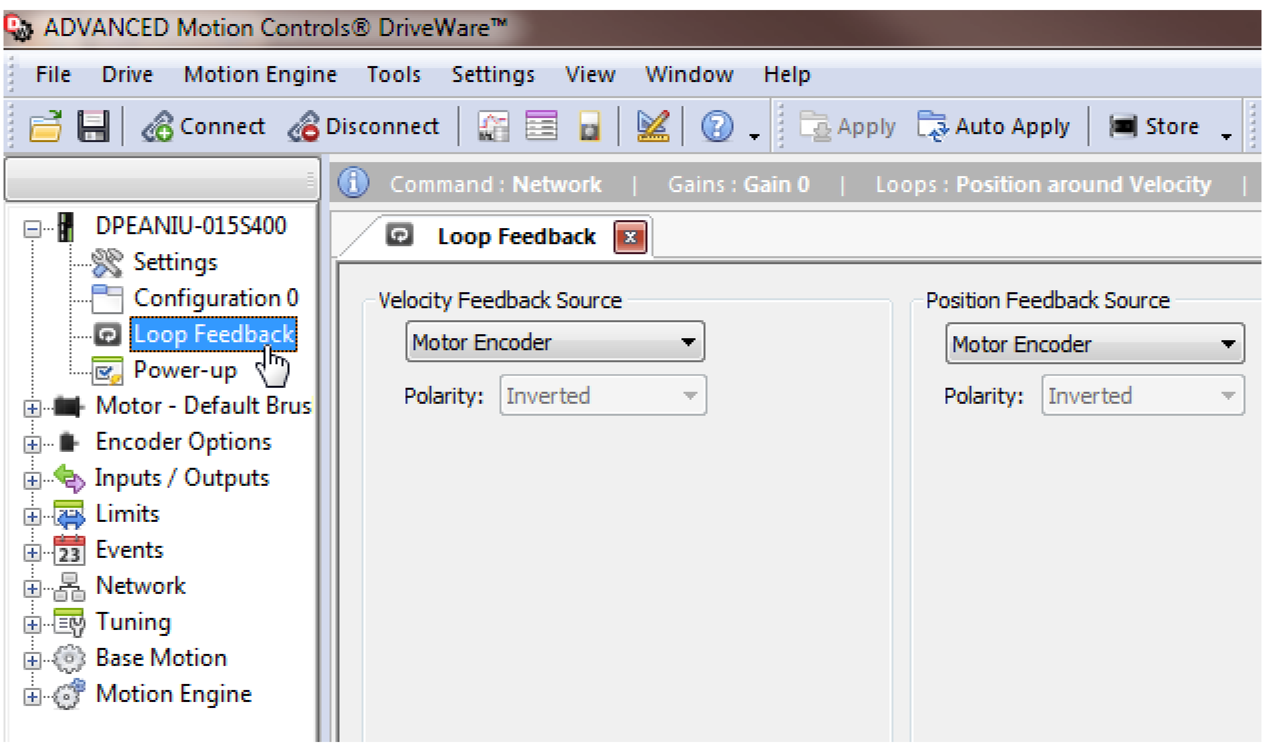


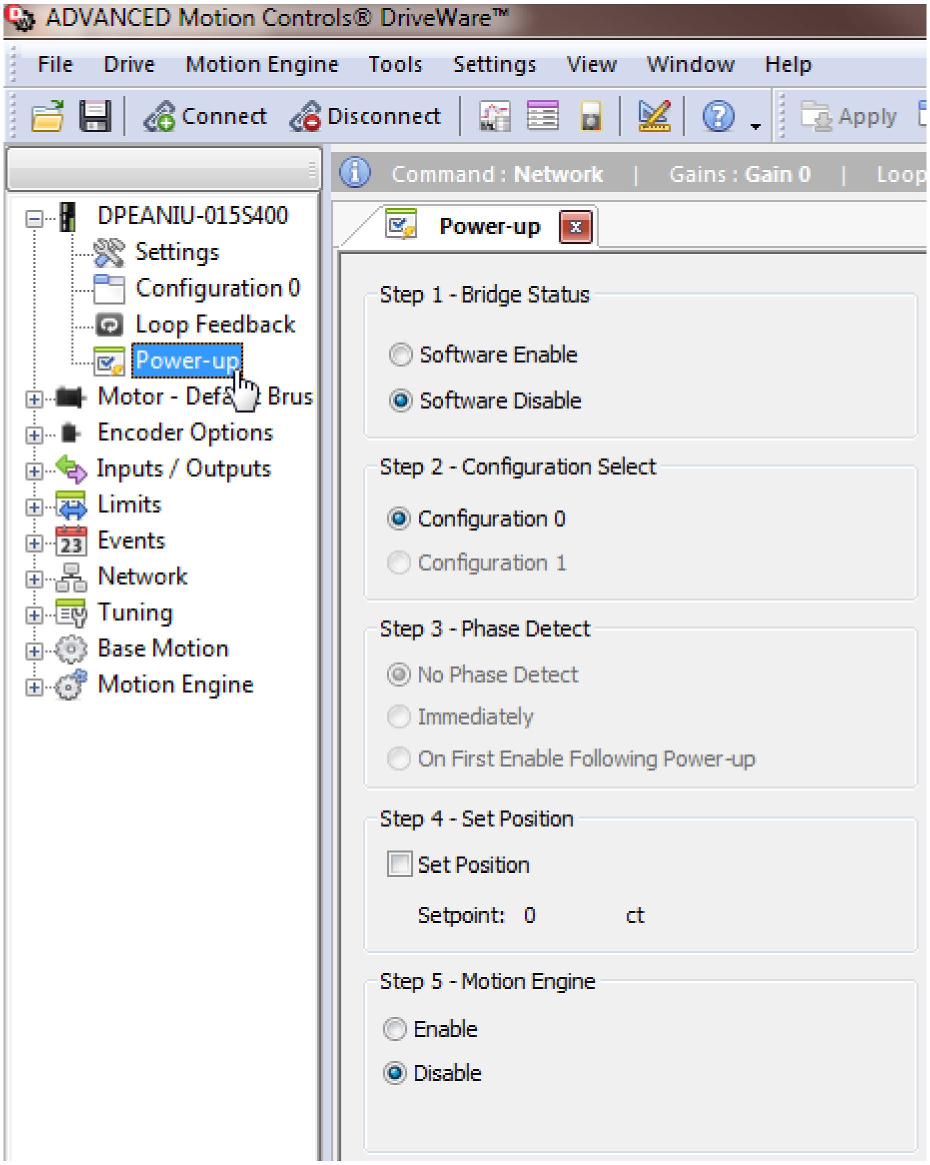
DriveWare

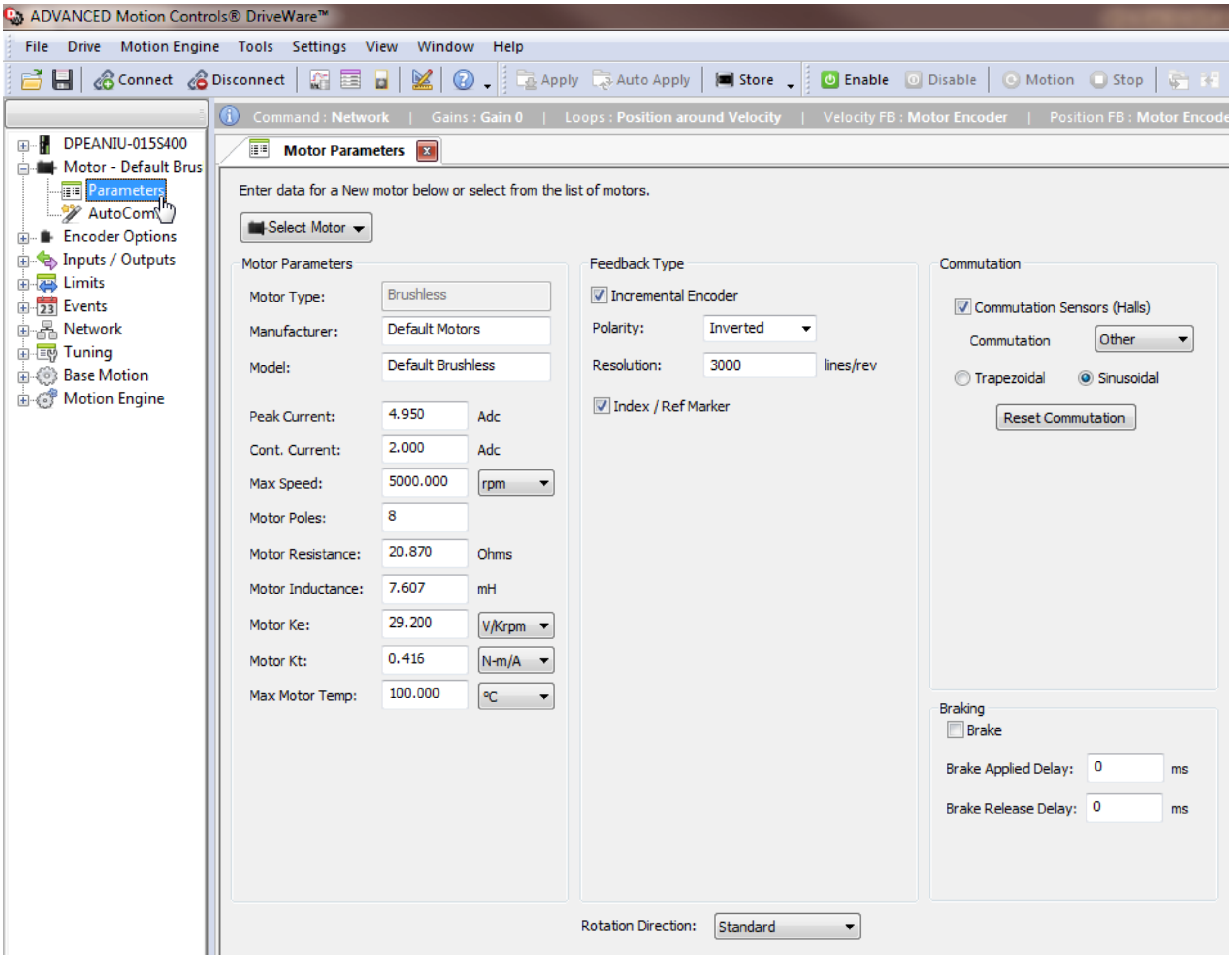
CTC used DriveWare 7.1 on a Windows 7 64-bit computer to test and configure the drives. Typically DriveWare will be needed for tuning and limited setup. Below are some setup screens and their typical parameters. Note that the PDO setup will be overridden by the M3-41 module and may be left at the default.

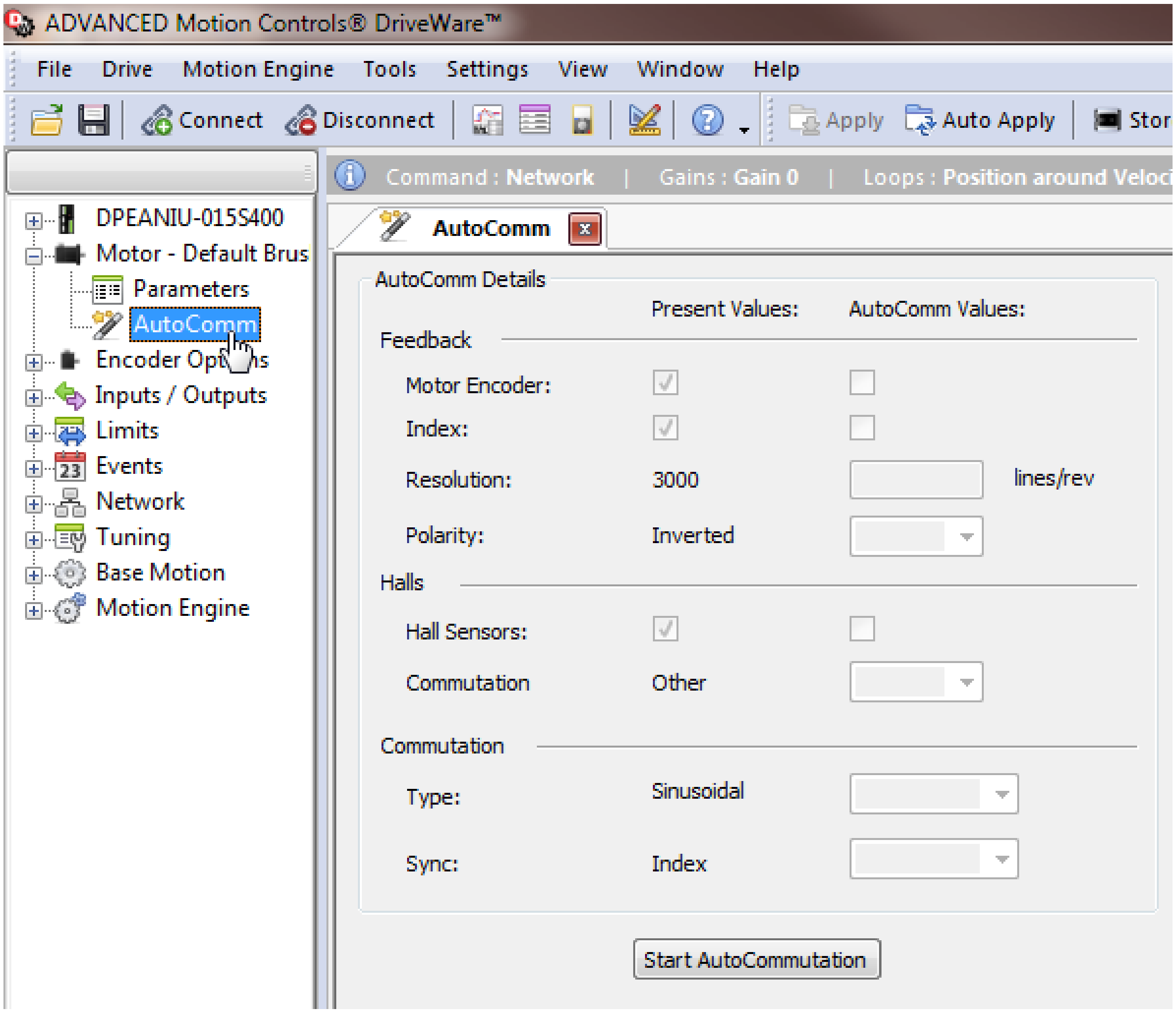


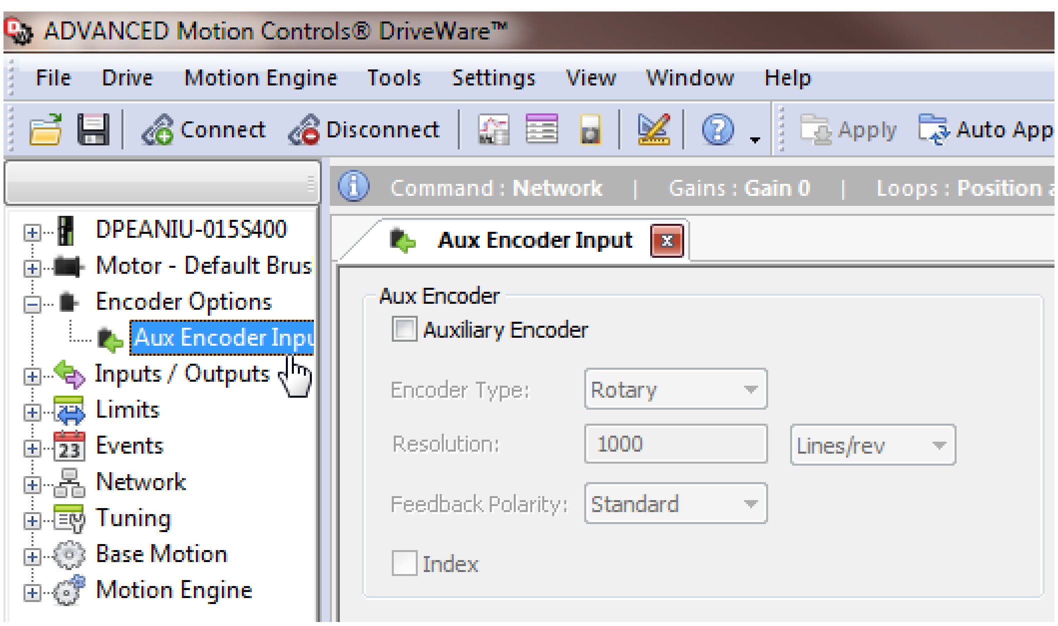

ADVANCED Motion Controls® DriveWare™
File Drive Motion Engine Tools Settings View Window Help
Connect Disconnect Apply Auto Ap
Command : Network | Gains : Gain 0 | Loops : Position
DPEANIU-015S400
Settings
Configuration 0
Loop Feedb
Power-up
Motor - Default Brus
Encoder Options
Inputs / Outputs
Limits
Events
Network
Tuning
Base Motion
Motion Engine
Configuration 0
Loop Configuration
Position around Velocity
Gain Set 0
Gain Set 1
Command Source
No Command
Command Limiter
Off
Linear Ramp
Accel/Decel

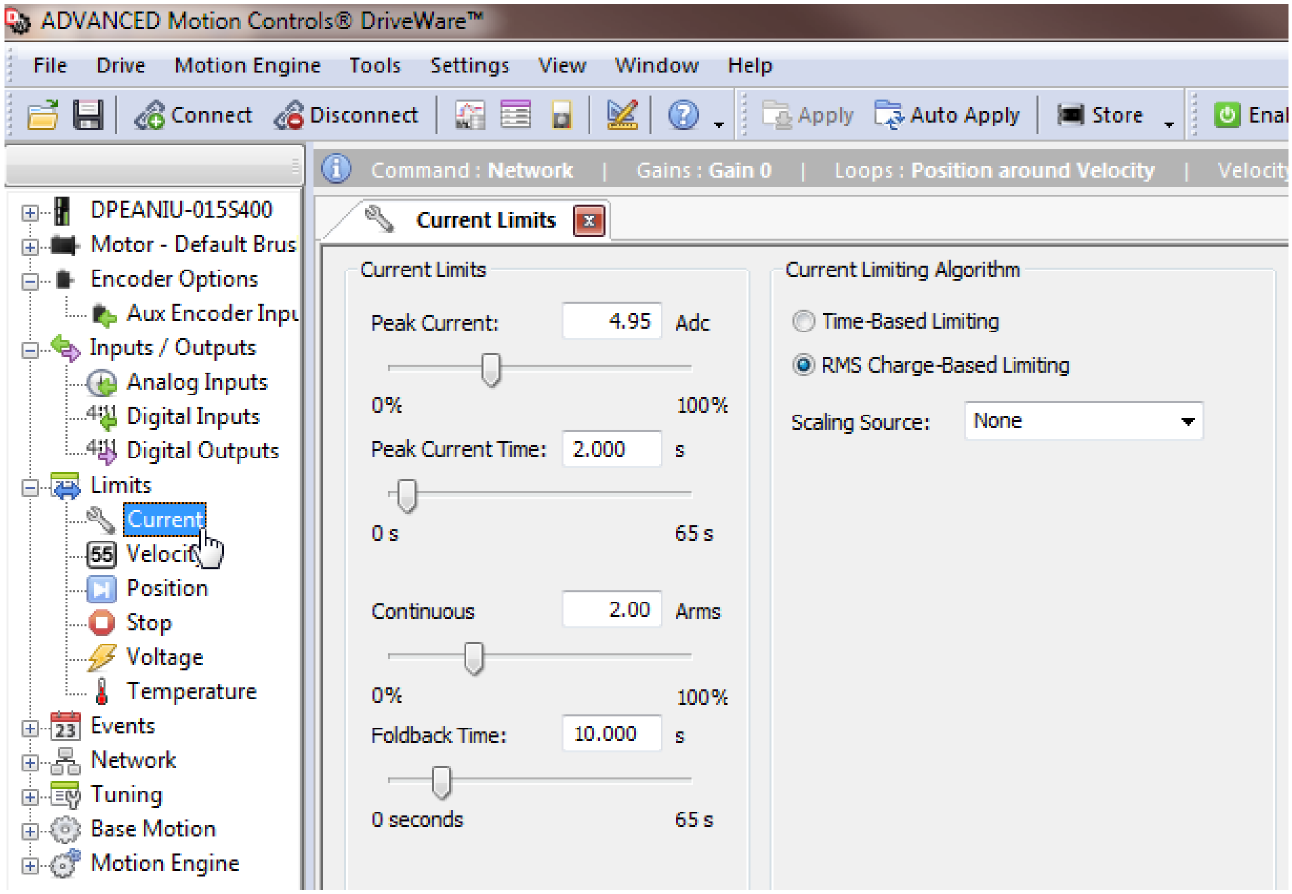

ADVANCED Motion Controls® DriveWare™
File Drive Motion Engine Tools Settings View Window Help
Connect Disconnect Apply Auto Apply Store
Command : Network | Gains : Gain 0 | Loops : Position around Velocity |
DPEANIU-015S400
Settings
Configuration 0
Loop Feedback
Power-up
Motor - Default Brus
Encoder Options
Inputs / Outputs
Limits
Events
Network
Tuning
Base Motion
Motion Engine
Loop Feedback
Velocity Feedback Source
Motor Encoder
Polarity: Inverted
Position Feedback Source
Motor Encoder
Polarity: Inverted

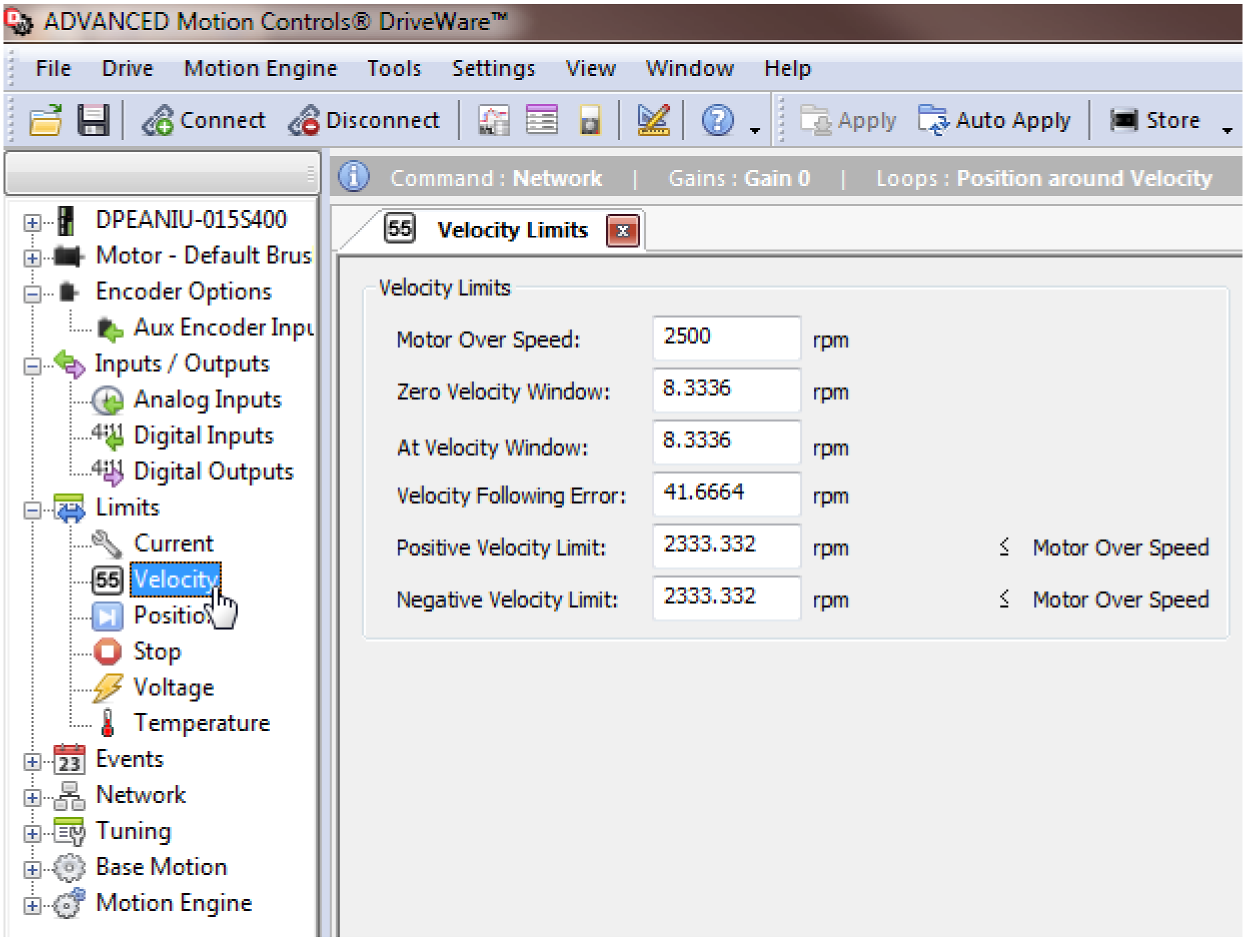

ADVANCED Motion Controls® DriveWare™
File Drive Motion Engine Tools Settings View Window Help
Connect Disconnect Apply
Command : Network | Gains : Gain 0 | Loop
DPEANIU-015S400
Settings
Configuration 0
Loop Feedback
Power-up
Motor - Def Brus
Encoder Options
Inputs / Outputs
Limits
Events
Network
Tuning
Base Motion
Motion Engine
Power-up
Step 1 - Bridge Status
Software Enable
Software Disable
Step 2 - Configuration Select
Configuration 0
Configuration 1
Step 3 - Phase Detect
No Phase Detect
Immediately
On First Enable Following Power-up
Step 4 - Set Position
Set Position
Setpoint: 0 ct
Step 5 - Motion Engine
Enable
Disable

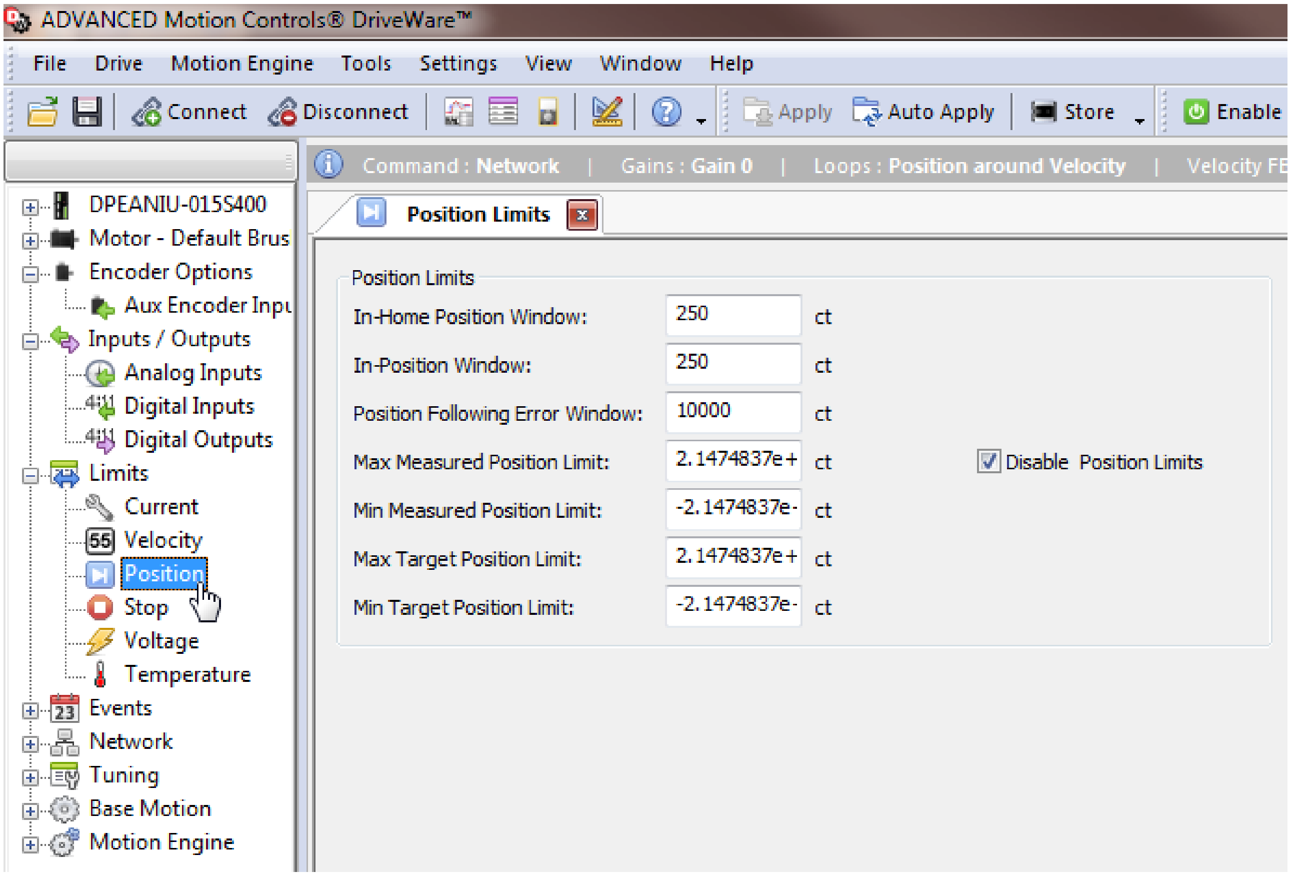

ADVANCED Motion Controls® DriveWare™
File Drive Motion Engine Tools Settings View Window Help
Connect Disconnect Apply Auto Apply Store Enable Disable Motion Stop
Command : Network | Gains : Gain 0 | Loops : Position around Velocity | Velocity FB : Motor Encoder | Position FB : Motor Encode
DPEANIU-015S400
Motor - Default Brus
Parameters
AutoCom
Encoder Options
Inputs / Outputs
Limits
Events
Network
Tuning
Base Motion
Motion Engine
Motor Parameters
Enter data for a New motor below or select from the list of motors.
Select Motor
Motor Parameters
Motor Type: Brushless
Manufacturer: Default Motors
Model: Default Brushless
Peak Current: 4.950 Adc
Cont. Current: 2.000 Adc
Max Speed: 5000.000 rpm
Motor Poles: 8
Motor Resistance: 20.870 Ohms
Motor Inductance: 7.607 mH
Motor Ke: 29.200 V/Krpm
Motor Kt: 0.416 N-m/A
Max Motor Temp: 100.000 °C
Feedback Type
Incremental Encoder
Polarity: Inverted
Resolution: 3000 lines/rev
Index / Ref Marker
Commutation
Commutation Sensors (Halls)
Commutation Other
Trapezoidal Sinusoidal
Reset Commutation
Braking
Brake
Brake Applied Delay: 0 ms
Brake Release Delay: 0 ms
Rotation Direction: Standard

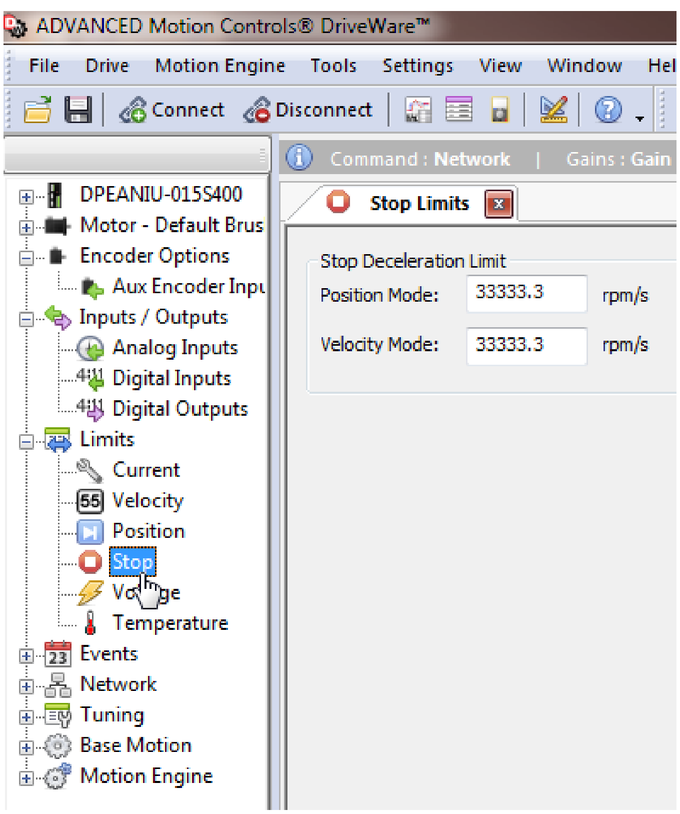

ADVANCED Motion Controls® DriveWare™
File Drive Motion Engine Tools Settings View Window Help
Connect Disconnect Apply Auto Apply Stor
Command : Network | Gains : Gain 0 | Loops : Position around Veloci
DPEANIU-015S400
Motor - Default Brus
Parameters
AutoComm
Encoder Options
Inputs / Outputs
Limits
Events
Network
Tuning
Base Motion
Motion Engine
AutoComm
AutoComm Details
Present Values:
AutoComm Values:
Feedback
Motor Encoder:
Index:
Resolution:
3000
lines/rev
Polarity:
Inverted
Halls
Hall Sensors:
Commutation
Other
Commutation
Type:
Sinusoidal
Sync:
Index
Start AutoCommutation

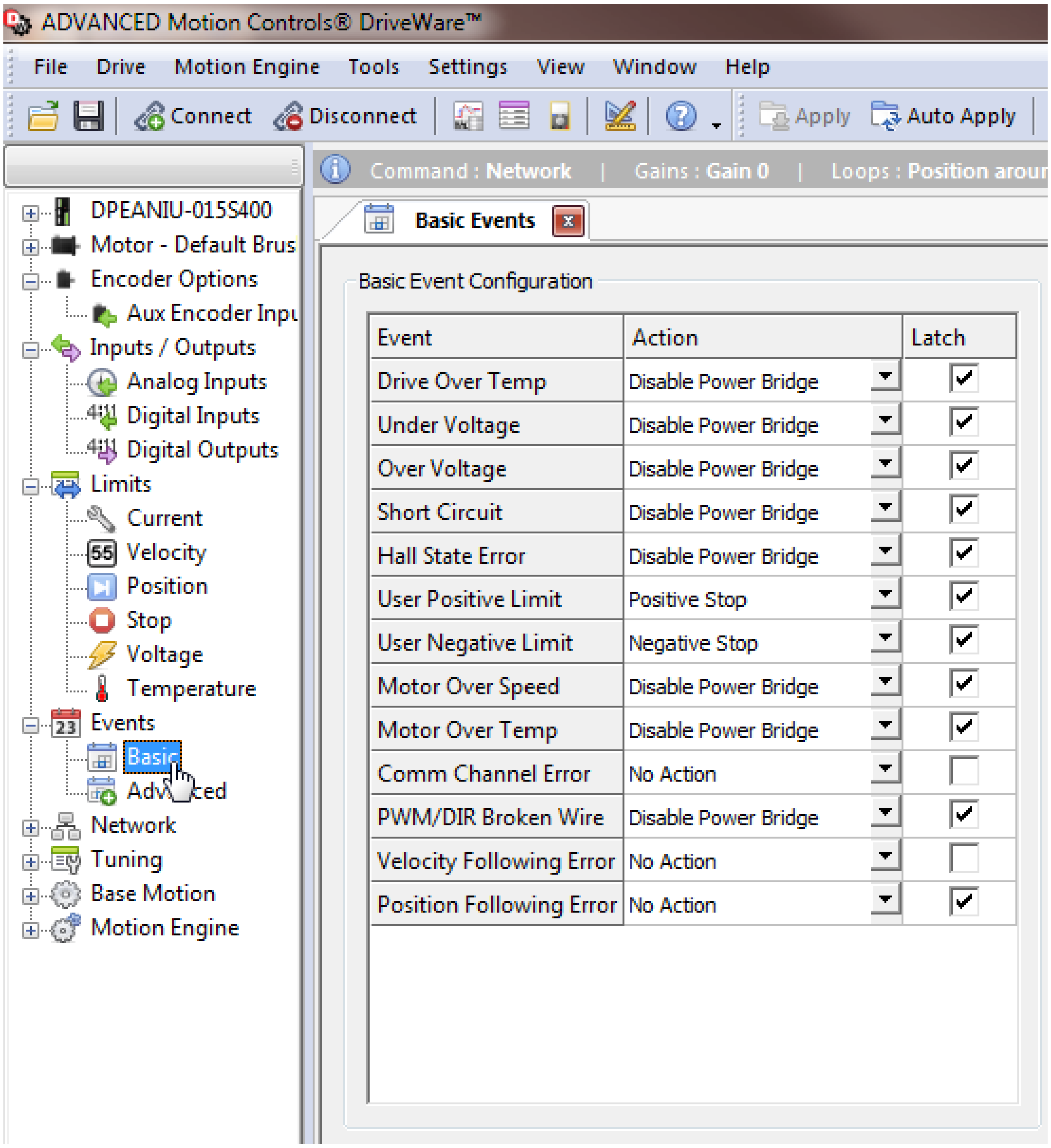

ADVANCED Motion Controls® DriveWare™
File Drive Motion Engine Tools Settings View Window Help
Connect Disconnect Apply Auto App
Command : Network | Gains : Gain 0 | Loops : Position a
DPEANIU-015S400
Motor - Default Brus
Encoder Options
Aux Encoder Inpu
Inputs / Outputs
Limits
Events
Network
Tuning
Base Motion
Motion Engine
Aux Encoder Input
Aux Encoder
Auxiliary Encoder
Encoder Type:
Rotary
Resolution:
1000
Lines/rev
Feedback Polarity:
Standard
Index


ADVANCED Motion Controls® DriveWare™
File Drive Motion Engine Tools Settings View Window Help
Connect
Disconnect
Apply
Auto Apply
Store
Command : Network | Gains : Gain 0 | Loops : Position around Velocity |
DPEANIU-015S400
Motor - Default Brus
Encoder Options
Aux Encoder Inpu
Inputs / Outputs
Analog Inputs
Digital Inputs
Digital Outputs
Limits
Current
Position
Stop
Voltage
Temperature
Events
Network
Tuning
Base Motion
Motion Engine
Current Limits
Current Limits
Peak Current: 4.95 Adc
0% 100%
Peak Current Time: 2.000 s
0 s 65 s
Continuous 2.00 Arms
0% 100%
Foldback Time: 10.000 s
0 seconds 65 s
Current Limiting Algorithm
Time-Based Limiting
RMS Charge-Based Limiting
Scaling Source: None

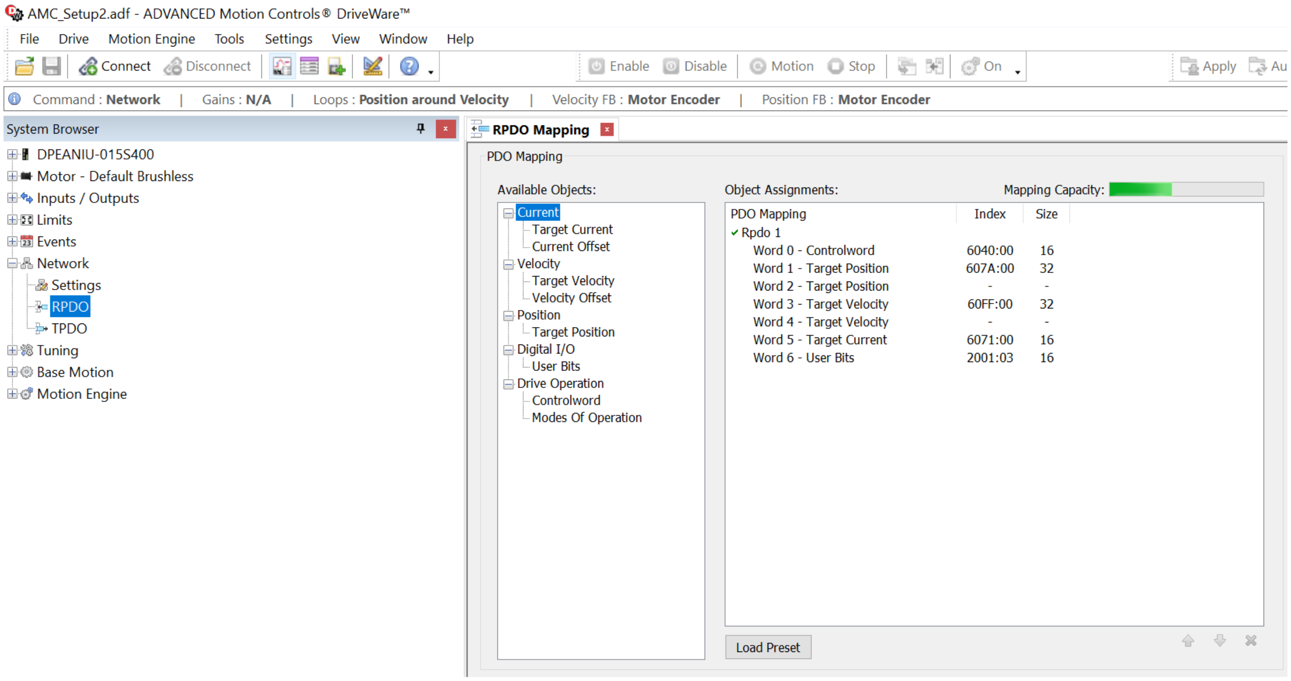

ADVANCED Motion Controls® DriveWare™
File Drive Motion Engine Tools Settings View Window Help
Connect Disconnect Apply Auto Apply Store
Command : Network | Gains : Gain 0 | Loops : Position around Velocity
DPEANIU-015S400
Motor - Default Brus
Encoder Options
Aux Encoder Inpu
Inputs / Outputs
Analog Inputs
Digital Inputs
Digital Outputs
Limits
Current
Velocity
Positio
Stop
Voltage
Temperature
Events
Network
Tuning
Base Motion
Motion Engine
Velocity Limits
Velocity Limits
Motor Over Speed: 2500 rpm
Zero Velocity Window: 8.3336 rpm
At Velocity Window: 8.3336 rpm
Velocity Following Error: 41.6664 rpm
Positive Velocity Limit: 2333.332 rpm ≤ Motor Over Speed
Negative Velocity Limit: 2333.332 rpm ≤ Motor Over Speed


ADVANCED Motion Controls® DriveWare™
File Drive Motion Engine Tools Settings View Window Help
Connect Disconnect Apply Auto Apply Store Enable
Command : Network | Gains : Gain 0 | Loops : Position around Velocity | Velocity FE
Position Limits
Position Limits
In-Home Position Window: 250 ct
In-Position Window: 250 ct
Position Following Error Window: 10000 ct
Max Measured Position Limit: 2.1474837e+ ct
Min Measured Position Limit: -2.1474837e- ct
Max Target Position Limit: 2.1474837e+ ct
Min Target Position Limit: -2.1474837e- ct
Disable Position Limits
DPEANIU-015S400
Motor - Default Brus
Encoder Options
Aux Encoder Inpu
Inputs / Outputs
Analog Inputs
Digital Inputs
Digital Outputs
Limits
Current
Velocity
Position
Stop
Voltage
Temperature
Events
Network
Tuning
Base Motion
Motion Engine

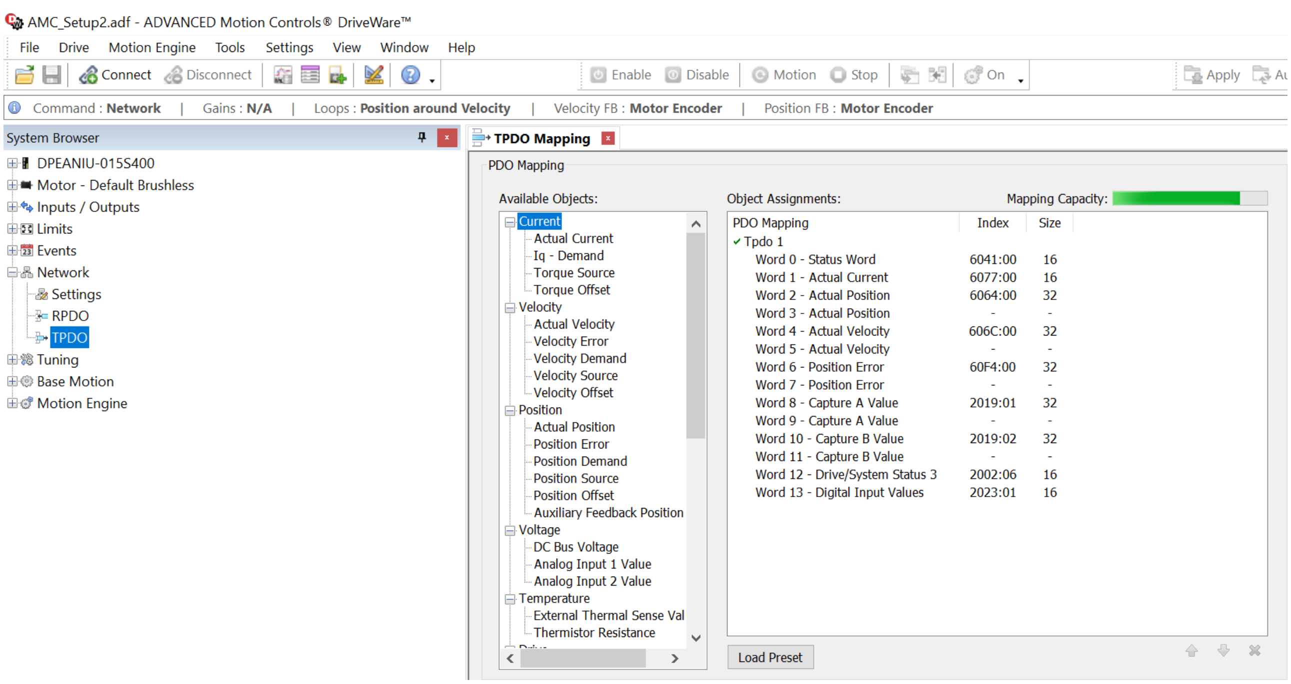

ADVANCED Motion Controls® DriveWare™
File Drive Motion Engine Tools Settings View Window Hel
Connect Disconnect
Command : Network | Gains : Gain
Stop Limits
Stop Deceleration Limit
Position Mode: 33333.3 rpm/s
Velocity Mode: 33333.3 rpm/s
DPEANIU-015S400
Motor - Default Brus
Encoder Options
Aux Encoder Inpu
Inputs / Outputs
Analog Inputs
Digital Inputs
Digital Outputs
Limits
Current
Velocity
Position
Stop
Temperature
Events
Network
Tuning
Base Motion
Motion Engine

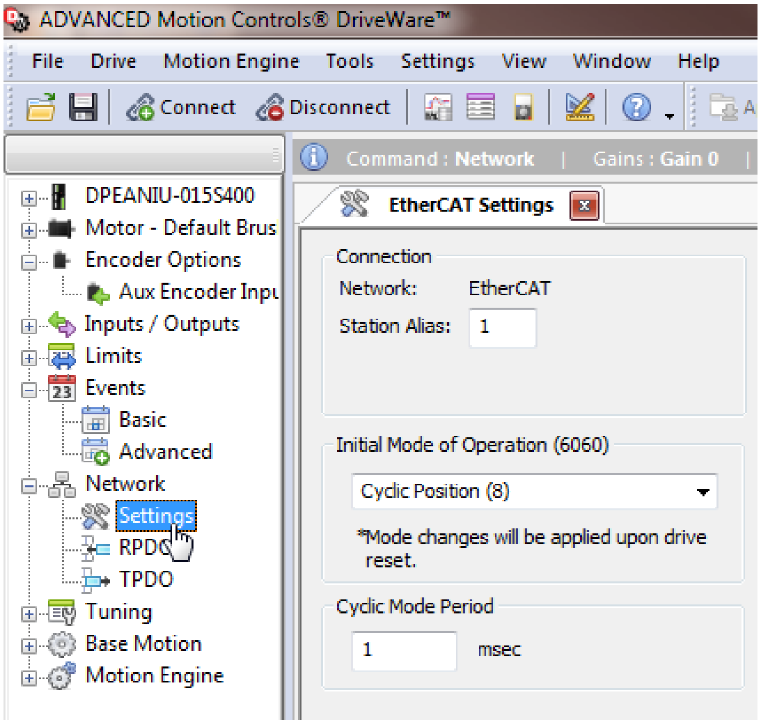


Comm Channel Error Event will automatically be enabled by the M3-41 module, with “Disable Power Bridge” as the default, unless otherwise selected on the above setup screen.



RPDO needs to be set as above and TXPDO as below. The mapping can be done by using a EEProm Programming utility provided by AMC that allows the xml file to be loaded prior to use using a regular Ethernet connection. Contact CTC Technical support to obtain the proper EtherCAT ESi xml file formatted for the required mapping. Do not use an xml file supplied from AMC as the one we have seen recently did not work properly with their programming utility.




ADVANCED Motion Controls® DriveWare™
File Drive Motion Engine Tools Settings View Window Help
Connect Disconnect
Command : Network | Gains : Gain 0 |
DPEANIU-015S400
Motor - Default Brus
Encoder Options
Aux Encoder Inpu
Inputs / Outputs
Limits
Events
Basic
Advanced
Network
Settings
RPDO
TPDO
Tuning
Base Motion
Motion Engine
EtherCAT Settings
Connection
Network: EtherCAT
Station Alias: 1
Initial Mode of Operation (6060)
Cyclic Position (8)
*Mode changes will be applied upon drive reset.
Cyclic Mode Period
1 msec

EtherCAT Explorer View

Manuf	Advanced Motion
Grp	Servo Drive
Name	AMC
Out	112 bits (14 bytes)
In	224 bits (28 bytes)
Axis #	4
pstate	RUNNING (1)
tracking_pstate	COMPLETE (2)
inpos	0
fpos	9.034667
tpos	9.063000
perr	0.027334
vel	1.044909
DRV MODE	Cyclic Sync Position (8)
PDO STATUS	0x0237
PDO CNTLWORD	0x000F
PDO ACT VEL	0x000140FF
PDO ACT TORQ	0x0000000F
PDO ACT ERR	0x0000013A
PDO HOME PWRUP	0x00000001
PDO ACT POS	0x0001A781
PDO TARG POS	0x0001A8D5
PDO TARG VEL	0x00000000
PDO DIG INP	0x00000000
State	8 (OPERATIONAL)
Delay	3030 ns
Has DC	true (32 bits)
DC Parent	1
DC Active	false, Cyc time: 0 ns, Shft: 0
Parent	3
Config addr	0x1004 (4100)
Station Alias	0
Vndr	0x000000bd (189)
Product Code	0x012d0000 (19726336)
Rev	0x01000101

EtherCAT Esi Xml File

Below is modified from the Advanced Motion Esi file AMC_DPE_01.00.04.00.xml, as previously stated newer xml files did not work properly with their “EEPROM Programmer.exe” slave utility program V.1.1.0.0. That highlighted are the changes required should a newer utility become available.

```
<?xml version="1.0" encoding="utf-8"?>
<EtherCATInfo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="EtherCATInfo.xsd" Version="1.6">
  <Vendor>
    <Id>189</Id>
    <Name>ADVANCED Motion Controls</Name>
```

```xml
<ImageData16x14>424DD60200000000000036000000280000001000000000E0000000100180000000000A00200000000000000000000000000000
000000FFFFFFFFFFFF0000FF0000FF0000FF0000FF0000FF0000FFFFFFFF0000FF0000FF0000FF0000FF0000FF0000FF0000FFFFFFFFFFFFFFFFFFFFF0000
FF0000FF0000FF0000FF0000FF0000FFFFFFFF0000FF0000FF0000FF0000FF0000FF0000FFFFFFFFFFFFFFFFFFFFFFFFFFF0000FF0000FF0000FF0000FF00
00FF0000FFFFFFFF0000FF0000FF0000FF0000FF0000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000FF0000FF0000FF0000FF0000FF0000FFFFFFFF0000FF0
000FF0000FF0000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000FF0000FF0000FF0000FF0000FF0000FFFFFFFF0000FF0000FF0000FFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000FF0000FF0000FF0000FF0000FF0000FFFFFFFF0000FF0000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFF0000FF0000FF0000FF0000FF0000FF0000FFFFFFFF0000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000FF0000FF0000FF00
00FF0000FF0000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000FF0000FF0000FF0000FF0000FF0000FFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000FF0000FF0000FF0000FF0000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000FF0000FF0000FF0000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFF0000FF0000FF0000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000FF0000FFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000FF</ImageData16x14>
 </Vendor>
 <Descriptions>
  <Groups>
   <Group SortOrder="0">
    <Type>Servo Drive</Type>
    <Name LcId="1031">Servo Drives</Name>
    <Name LcId="1033">Servo Drives</Name>

<ImageData16x14>424DD60200000000000036000000280000001000000000E0000000100180000000000A00200000000000000000000000000000
000000FFFFFFFFFFFFFFFFFFFFFFFFF000000000000000000000000000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
000000C0C0C080808000000000800000FF00000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF000000C0C0C0C0C0C0000000008
00000FF00000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF000000C0C0C080808000000000800000FF00000000FFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000000000000000000000000800000FF00000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFF000000C0C0C080808000000000800000FF00000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF000000C0C0C0C0C0C0
000000000800000FF00000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF000000C0C0C0808080000000000000000000000000FFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF000000000000000000000000000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFF000000000000000000000000000000000000FF000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF000000C0C
0C000FFFF000000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF000000C0C0C08080800000000000000000000
000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF000000C0C0C000FF000000000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000000000000000000000000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF</ImageData16x14>
   </Group>
  </Groups>
  <Devices>
   <Device Physics="YY">
    <Type ProductCode="#x012D0000" RevisionNo="#x01000400" CheckProductCode="NONE" CheckRevisionNo="NONE"
CheckSerialNo="NONE">AMC Servo Drive</Type>
    <Name LcId="1033">AMC Servo Drive</Name>
    <Name LcId="1031">AMC Servo Drive</Name>
    <URL>http://www.a-m-c.com</URL>
    <Info>
     <IdentificationReg134>1</IdentificationReg134>
    </Info>
    <GroupType>Servo Drive</GroupType>
    <Profile>
     <ChannelInfo>
      <ProfileNo>402</ProfileNo>
     </ChannelInfo>
     <DictionaryFile>AMC DP_DZ\AMC_DP_DZ1 Dict.xml</DictionaryFile>
    </Profile>
    <Fmmu>Outputs</Fmmu>
    <Fmmu>Inputs</Fmmu>
    <Fmmu>MBoxState</Fmmu>
    <Sm MinSize="34" MaxSize="612" DefaultSize="612" StartAddress="#x1000" ControlByte="#x26" Enable="1">MBoxOut</Sm>
    <Sm MinSize="34" MaxSize="612" DefaultSize="612" StartAddress="#x1600" ControlByte="#x22" Enable="1">MBoxIn</Sm>
    <Sm StartAddress="#x1C00" ControlByte="#x64" Enable="1">Outputs</Sm>
    <Sm StartAddress="#x1E00" ControlByte="#x20" Enable="1">Inputs</Sm>
    <RxPdo Mandatory="0" Fixed="1" Virtual="0" Sm="2">
     <Index>#x1600</Index>
     <Name>Outputs</Name>
     <Entry>
      <Index>#x6040</Index>
      <SubIndex>0</SubIndex>
      <BitLen>16</BitLen>
```

```
        <Name>Control Word</Name>
        <DataType>UINT</DataType>
      </Entry>
      <Entry>
        <Index>#x607A</Index>
        <SubIndex>0</SubIndex>
        <BitLen>32</BitLen>
        <Name>Target Position</Name>
        <DataType>DINT</DataType>
      </Entry>
      <Entry>
        <Index>#x60FF</Index>
        <SubIndex>0</SubIndex>
        <BitLen>32</BitLen>
        <Name>Target Velocity</Name>
        <DataType>DINT</DataType>
      </Entry>
      <Entry>
        <Index>#x6071</Index>
        <SubIndex>0</SubIndex>
        <BitLen>16</BitLen>
        <Name>Target Current</Name>
        <DataType>INT</DataType>
      </Entry>
      <Entry>
        <Index>#x2001</Index>
        <SubIndex>3</SubIndex>
        <BitLen>16</BitLen>
        <Name>User Bits</Name>
        <DataType>UINT</DataType>
      </Entry>
    </RxPdo>
    <TxPdo Mandatory="0" Fixed="1" Virtual="0" Sm="3">
      <Index>#x1A00</Index>
      <Name>Inputs</Name>
      <Entry>
        <Index>#x6041</Index>
        <SubIndex>0</SubIndex>
        <BitLen>16</BitLen>
        <Name>Status Word</Name>
        <DataType>UINT</DataType>
      </Entry>
      <Entry>
        <Index>#x6077</Index>
        <SubIndex>0</SubIndex>
        <BitLen>16</BitLen>
        <Name>Actual Torque</Name>
        <DataType>INT</DataType>
      </Entry>
      <Entry>
        <Index>#x6064</Index>
        <SubIndex>0</SubIndex>
        <BitLen>32</BitLen>
        <Name>Actual Position</Name>
        <DataType>DINT</DataType>
      </Entry>
      <Entry>
        <Index>#x606C</Index>
        <SubIndex>0</SubIndex>
        <BitLen>32</BitLen>
        <Name>Actual Velocity</Name>
        <DataType>DINT</DataType>
      </Entry>
      <Entry>
        <Index>#x60F4</Index>
        <SubIndex>0</SubIndex>
```

```xml
        <BitLen>32</BitLen>
        <Name>Position Following Error</Name>
        <DataType>DINT</DataType>
      </Entry>
      <Entry>
        <Index>#x2019</Index>
        <SubIndex>1</SubIndex>
        <BitLen>32</BitLen>
        <Name>Capture A</Name>
        <DataType>DINT</DataType>
      </Entry>
      <Entry>
        <Index>#x2019</Index>
        <SubIndex>2</SubIndex>
        <BitLen>32</BitLen>
        <Name>Capture B</Name>
        <DataType>DINT</DataType>
      </Entry>
      <Entry>
        <Index>#x2002</Index>
        <SubIndex>6</SubIndex>
        <BitLen>16</BitLen>
        <Name>Capture Status</Name>
        <DataType>UINT</DataType>
      </Entry>
      <Entry>
        <Index>#x2023</Index>
        <SubIndex>1</SubIndex>
        <BitLen>16</BitLen>
        <Name>Digital Inputs</Name>
        <DataType>UINT</DataType>
      </Entry>
    </TxPdo>
    <Mailbox DataLinkLayer="1">
      <EoE />
      <CoE SdoInfo="1" PdoAssign="0" PdoConfig="0" PdoUpload="1" SegmentedSdo="1" CompleteAccess="0" />
    </Mailbox>
    <Dc>
      <OpMode>
        <Name>DC Off</Name>
        <Desc>FreeRun</Desc>
        <AssignActivate>#x0</AssignActivate>
      </OpMode>
      <OpMode>
        <Name>DC On</Name>
        <Desc>DC-Synchronous</Desc>
        <AssignActivate>#x300</AssignActivate>
        <CycleTimeSync0 Factor="1">0</CycleTimeSync0>
        <ShiftTimeSync0>0</ShiftTimeSync0>
      </OpMode>
    </Dc>
    <Eeprom AssignToPdi="0">
      <ByteSize>2048</ByteSize>
      <ConfigData>8D0E00448813000000000000800000</ConfigData>
    </Eeprom>
   </Device>
  </Devices>
 </Descriptions>
</EtherCATInfo>
```

Appendix C1

[C1] Applied Motion Products

Applied Motion Products manufactures a number of drives. That currently supported by Incentive is the SS-EC and STF. This drive is actually a stepper, not a servo, but it is made to appear like a servo. For test purposes the SS-05EC-D and STF10-EC drive were tested. This section provides information that may be specific to this manufacturer.

eCAT_driveType – 20 (SS-EC) and 26 (STF)

Overview

Only CSP and Homing mode are supported with the SS-EC and STF drives. Additionally the DC SYNC motion control instruction is ignored since it is automatically enabled prior to network operation. Thus it is automatically enabled by the EtherCAT Master with the SYNC0 always having an offset from the network cycle time, typically 1 mS cycle time, with a shift of 'cycle time/4' or 250uS for 1 mS, whichever is smaller. SYNC1 is not available. Only touch probe value rising edge is available although at present time it has not been tested. All status and function parameters are mapped to the drive as required.

Drive PPR selection:

For speed and acceleration and deceleration parameters, it is based on the encoder resolution of the motor.

When your Drive connect with **HT17/23/24/34-SS** Motors, the encoder resolution is 20000 counts/rev

Parameter Type	Speed	Value
Distance	1 rev	20000
Speed	1 rps	20000
Accel/Decel	100 eps/s	2000000

When your Drive connect with **HT11-SS** Motors, the encoder resolution is 4096 counts/rev

Parameter Type	Speed	Value
Distance	1 rev	4096
Speed	1 rps	4096
Accel/Decel	100 eps/s	409600

STF PPR was 20000 counts/rev as tested.

COE Objects of Interest

With regards to torque control an SDO write is needed to object 0x6073 (max current). This object controls the maximum current. It is a 16 bit value and is in increments of .01 Amps.

0x6073 Max Current

This object configures the max continues current permissible value of the drive. The unit of this object is 0.01Amps.

Object Type	Data Type	Access Type	PDO Mapping	Default Value
Var	UNSIGNED16	rw	no	-

Digital Inputs are available via the 'dins' variable. X1 input is assigned to Bit 0 with X2 to X8 being Bit 1 to Bit7. CCW Limit is Bit 16, CW Limit is Bit 17, and Home Sensor is Bit 18.

Digital Outputs, 'douts' variable, have Y1 assigned to Bit 0, Y2 to Bit1, Y3 to Bit2, and Y4 to Bit3.

If needed a Home Offset can be set via an SDO write:

0x607C Home Offset

This object is the difference between the zero position for the application and the home sensor position (found during homing). During homing the home sensor position is found and once the homing is completed the zero position is offset from the home position by adding the Home Offset to the home position. All subsequent absolute moves shall be taken relative to this new zero position. The unit of this object is step.

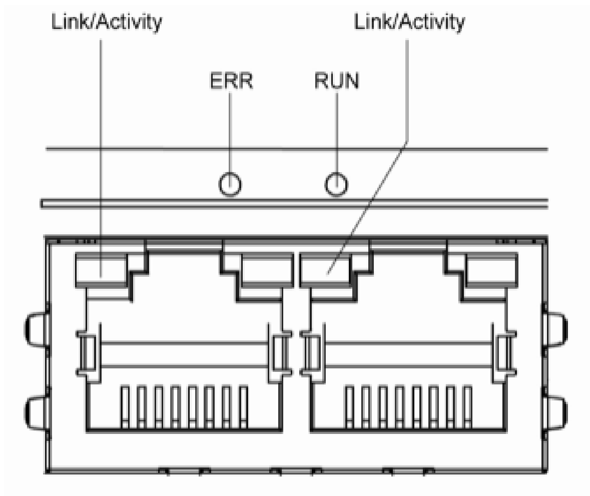
Object Type	Data Type	Access Type	PDO Mapping	Default Value
Var	INTEGER32	rw	yes	0

The minimum acceleration is .167 rev/sec^2 (for homing 1 rev/sec^2), maximum is 5461.167 rev/sec^2. If a greater acceleration is given the drive does not limit is and will move at extremely slow speeds, about 1 pulse per second. This was observed with firmware level 2.1. For other limits reference the "Host Command Reference" document available on Applied Motion's web site:

https://www.applied-motion.com/sites/default/files/hardware-manuals/Host-Command-Reference_920-0002P.PDF

EtherCAT Connection

There are two EtherCAT connectors labeled LA IN and LA OUT. The EtherCAT input cable should be connected to LA IN. Additionally there are a number of status LED's, those for Link/Activity, ERR, and RUN as detailed below:



LED indicator descriptions:

LED	Color	Status	Description
Link / Activity	Green	OFF	no Ethernet connection
		ON	Ethernet is connected
		Flickering	activity on line
RUN	Green	OFF	initialization state
		Blinking	pre-operational state
		Single Flash	safe-operational state
		ON	operational state
ERR	Red	OFF	no error
		Blinking	general error
		Single Flash	sync error
		Double Flash	watch dog error

Notes:

Flickering: Rapid flashing with a period of approx. 50ms (10 Hz)

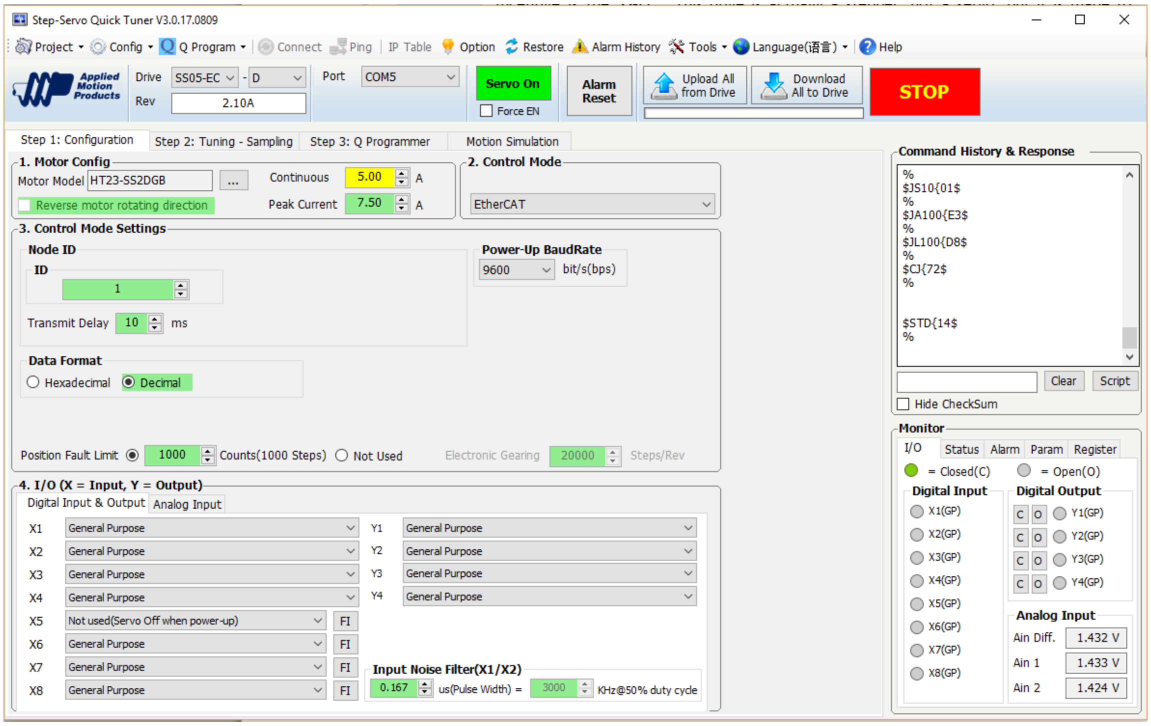
Blinking: Flashing with equal on and off periods of 200ms (2.5Hz)

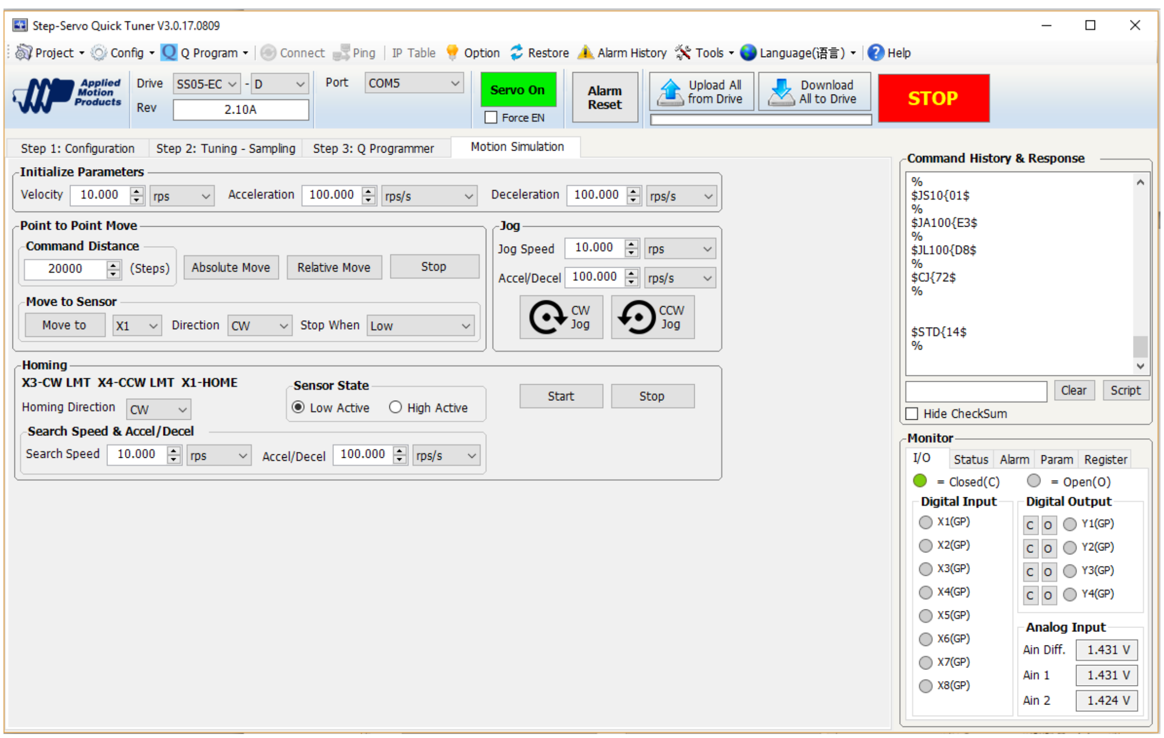
Single Flash: Repeating ON for 200ms and OFF for 1s

Double Flash: Two flashes with a period of 200ms followed by 1s OFF period

Step-Servo Quick Tuner Screens

Before operation it is best to use Applied Motion's Quick Tuner to tune, jog the motor and monitor sensor inputs. Once operational it may then be connected for EtherCAT operation.





Appendix

C2

[C2] Delta Electronics ADSA-A2

Delta Electronics manufactures a number of drives. That currently supported by the M3-41A is the ADSA-A2. For test purposes two drives in particular were tested: ADSA-A2-0421-E. This section provides information that may be specific to this manufacturer.

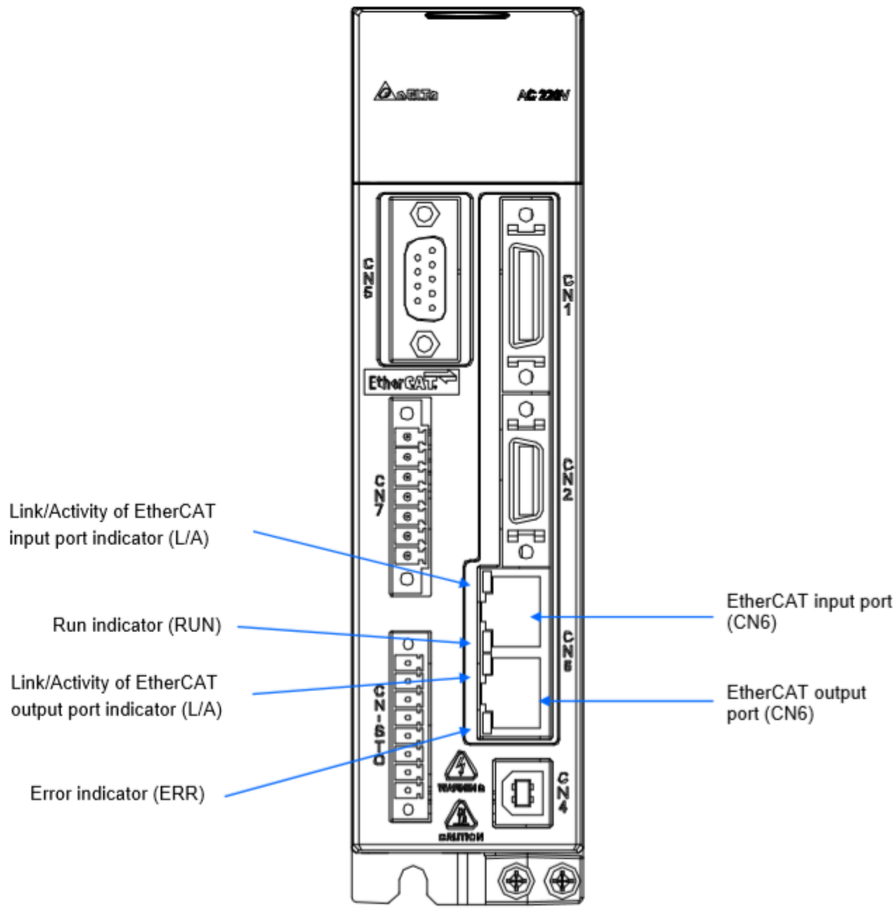
eCAT_driveType – 25

Overview

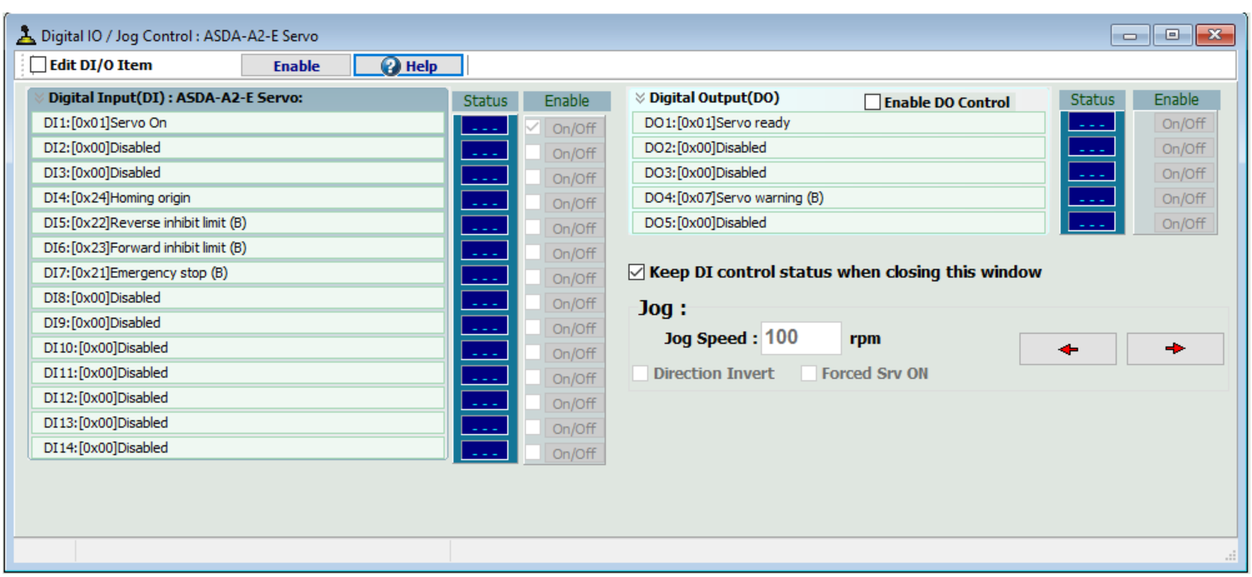
Only CSP, Homing and Profile Position modes are supported with the Delta drives. Additionally the DC SYNC motion control instruction is ignored since it is automatically enabled prior to network operation since the Delta drive does not support free run mode (manual says it does, but it does not). Thus it is automatically enabled by the EtherCAT Master with the SYNC0 always having an offset from the network cycle time, typically 1 mS cycle time, with a shift of 'cycle time/4' or 250uS for 1 mS, whichever is smaller. SYNC1 is not supported by the drive.

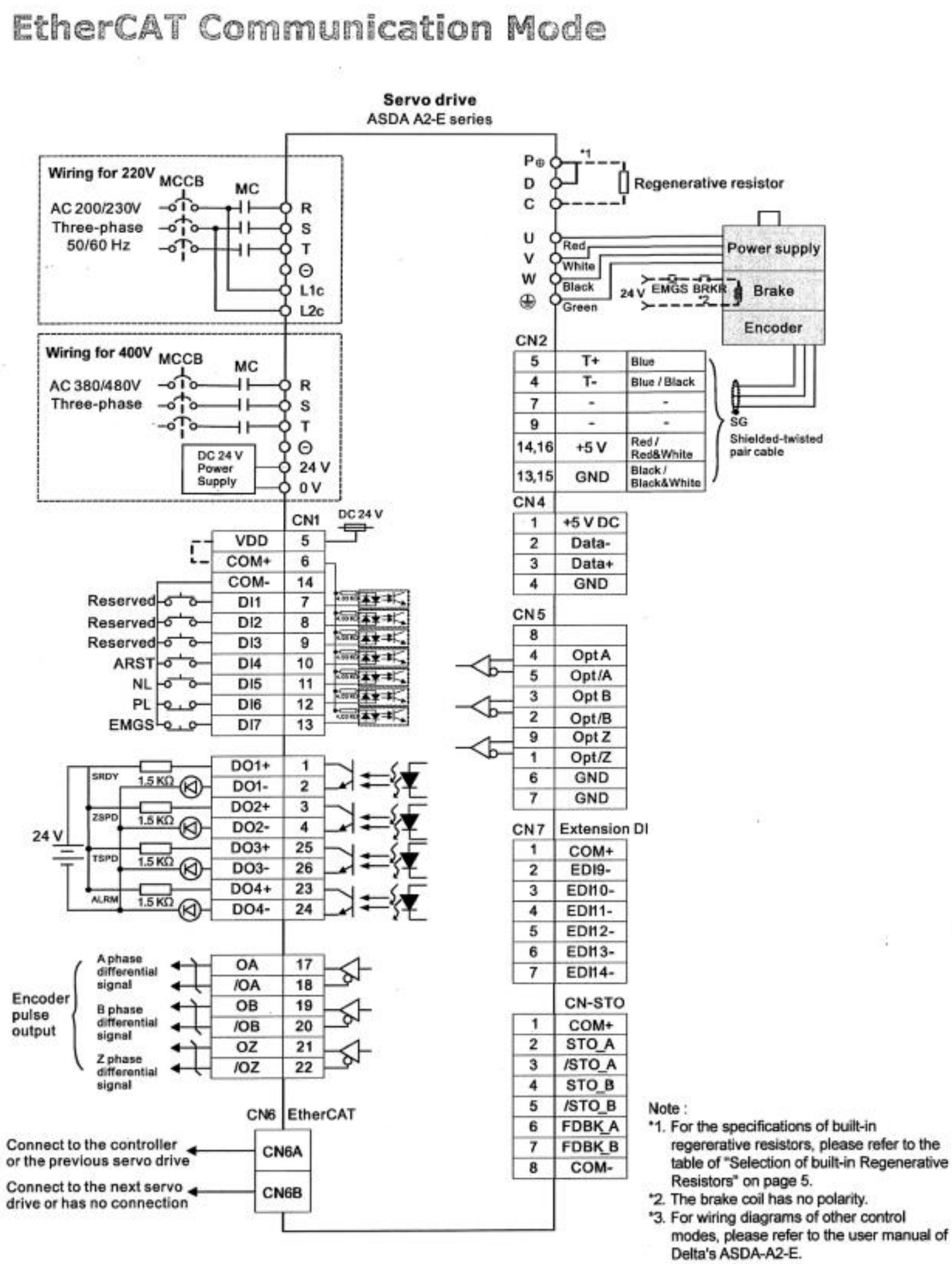
Connectors

For test purposes the drive was wired as recommend in the Delta manuals, note that the Negative Limit, NL is actually a Normally Closed (NC, their type B) connection and not Normally Open, it is a manual misprint. The STO bypass PCB included with the drive was installed. CN1 was configured both with wiring and by settings within their ADSA-Soft setup software for digital inputs and outputs. Homing origin was set for DI4 since it was the factory default versus that shown in the diagram. Inputs can be set as you require for your application. Before attempting to run the CTC Incentive EtherCAT system it is recommended that you verify that you can jog the motor first, using the Delta PC software and the drive is tuned.



Digital inputs and outputs may be changed as desired but for test purposes were as follows and that recommended in the manual:

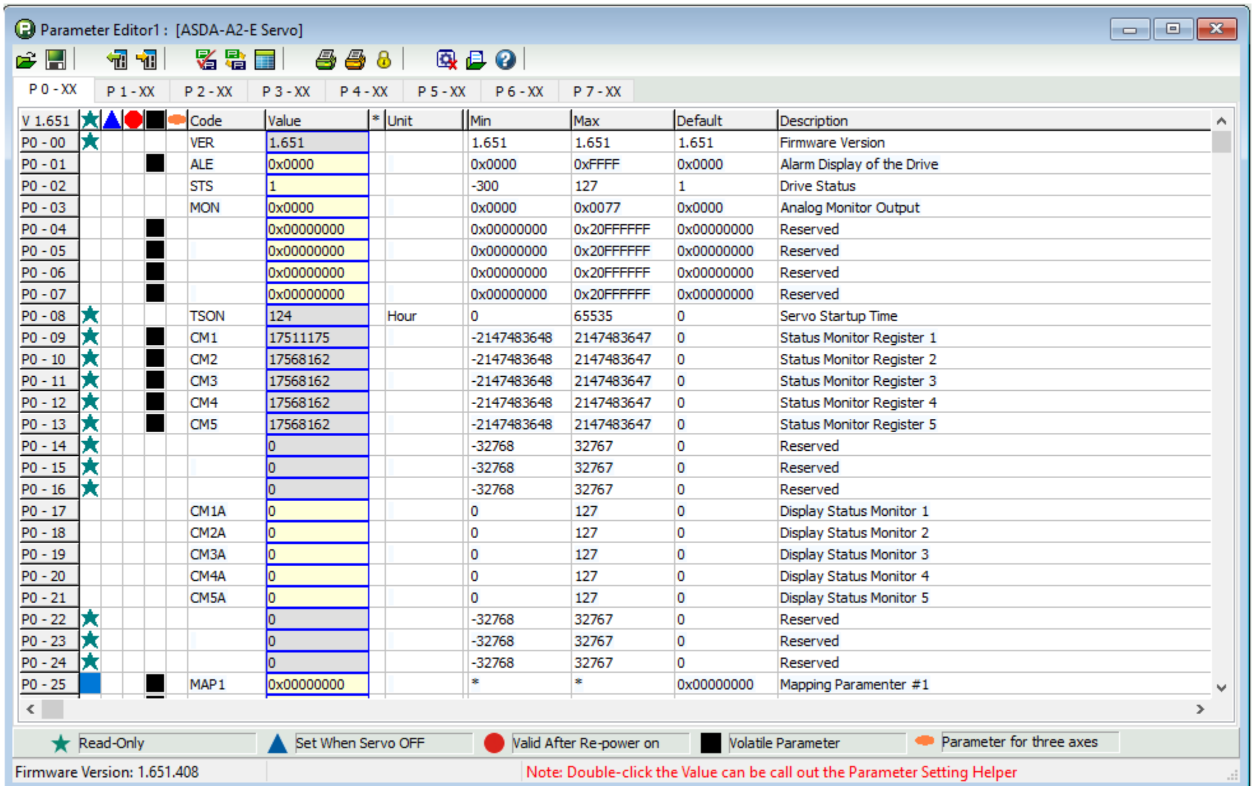




ADSA-Soft(V5)

The Delta Windows PC Software allows for configuration, tuning, and monitoring of the drive via a USB connection. It is recommended that prior to starting your configuration that you initialize the drive to its factory defaults by writing a 10 to P2-08, save the parameter, then cycle power on the drive. This will allow you to start from a known state. There are numerous settings possible with this drive and the screens in this section list that used for testing. Make sure that you do not change the units in the drive as it will effect that used by our EtherCAT Master. We expect the default which is rpm based for most velocity/acceleration based settings. QuickBuilder will use rps, revolutions per second and convert accordingly. Feedback position is based on ppr which was 1280000 pulses per revolution for the drive tested. Make sure P1-01 is set to 0x000C which is CANopen (Full). The software does not have an EtherCAT setting but will run CANopen over EtherCAT. Also cycle power whenever parameters are written to the drive, then read the parameters back to ensure they are set properly. Below are some of the parameters used during testing:

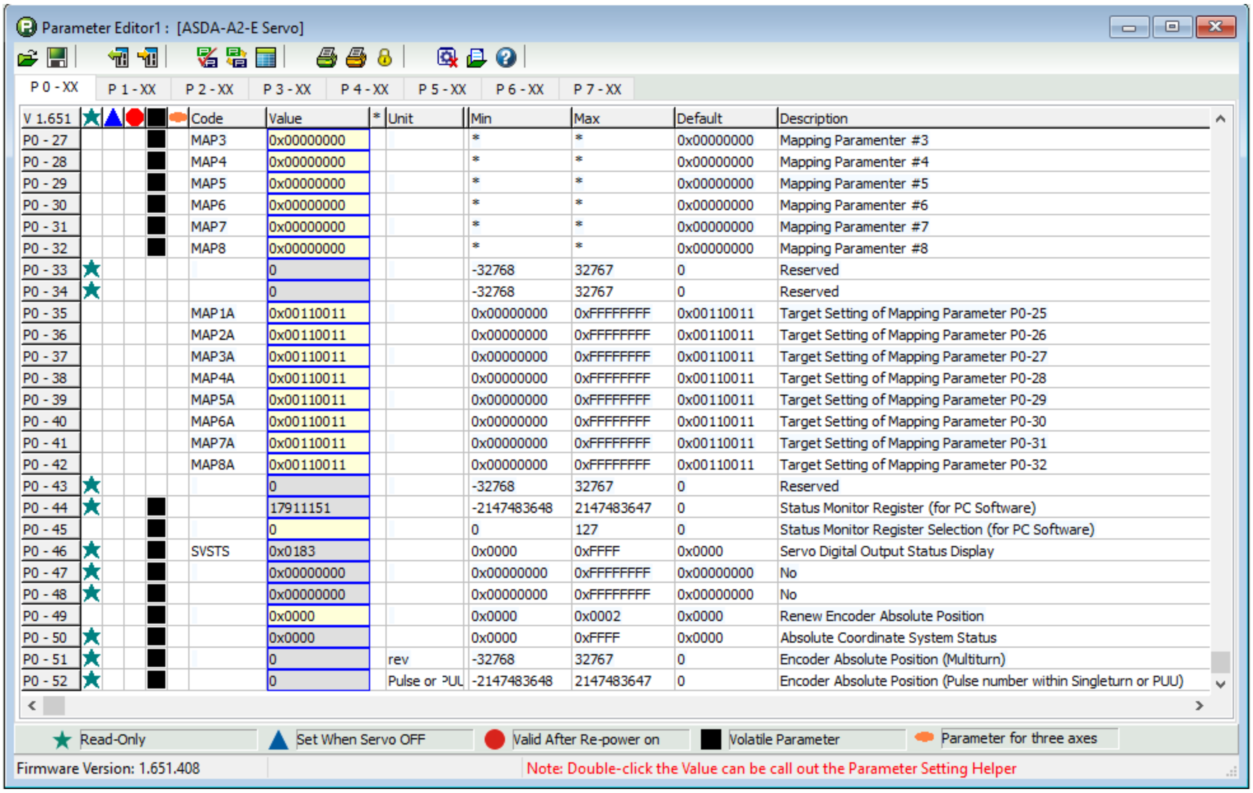
P0-XX



V 1.651	★	▲	●	■	Code	Value	*	Unit	Min	Max	Default	Description
P0 - 00	★				VER	1.651			1.651	1.651	1.651	Firmware Version
P0 - 01				■	ALE	0x0000			0x0000	0xFFFF	0x0000	Alarm Display of the Drive
P0 - 02					STS	1			-300	127	1	Drive Status
P0 - 03					MON	0x0000			0x0000	0x0077	0x0000	Analog Monitor Output
P0 - 04				■		0x00000000			0x00000000	0x20FFFFFF	0x00000000	Reserved
P0 - 05				■		0x00000000			0x00000000	0x20FFFFFF	0x00000000	Reserved
P0 - 06				■		0x00000000			0x00000000	0x20FFFFFF	0x00000000	Reserved
P0 - 07				■		0x00000000			0x00000000	0x20FFFFFF	0x00000000	Reserved
P0 - 08	★				TSON	124		Hour	0	65535	0	Servo Startup Time
P0 - 09	★			■	CM1	17511175			-2147483648	2147483647	0	Status Monitor Register 1
P0 - 10	★			■	CM2	17568162			-2147483648	2147483647	0	Status Monitor Register 2
P0 - 11	★			■	CM3	17568162			-2147483648	2147483647	0	Status Monitor Register 3
P0 - 12	★			■	CM4	17568162			-2147483648	2147483647	0	Status Monitor Register 4
P0 - 13	★			■	CM5	17568162			-2147483648	2147483647	0	Status Monitor Register 5
P0 - 14	★					0			-32768	32767	0	Reserved
P0 - 15	★					0			-32768	32767	0	Reserved
P0 - 16	★					0			-32768	32767	0	Reserved
P0 - 17					CM1A	0			0	127	0	Display Status Monitor 1
P0 - 18					CM2A	0			0	127	0	Display Status Monitor 2
P0 - 19					CM3A	0			0	127	0	Display Status Monitor 3
P0 - 20					CM4A	0			0	127	0	Display Status Monitor 4
P0 - 21					CM5A	0			0	127	0	Display Status Monitor 5
P0 - 22	★					0			-32768	32767	0	Reserved
P0 - 23	★					0			-32768	32767	0	Reserved
P0 - 24	★					0			-32768	32767	0	Reserved
P0 - 25				■	MAP1	0x00000000			*	*	0x00000000	Mapping Paramenter #1

★ Read-Only ▲ Set When Servo OFF ● Valid After Re-power on ■ Volatile Parameter ⬬ Parameter for three axes

Firmware Version: 1.651.408 Note: Double-click the Value can be call out the Parameter Setting Helper



Parameter Editor1 : [ASDA-A2-E Servo]

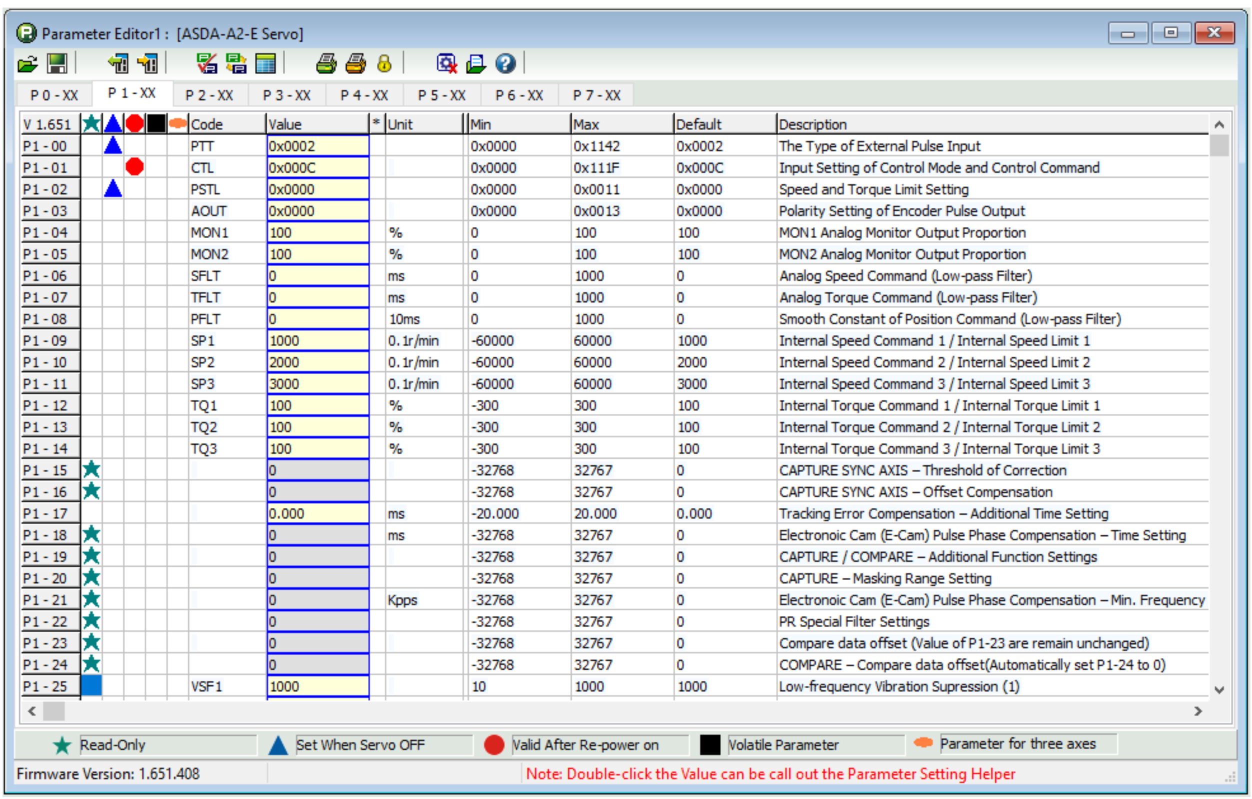
P 0 - XX | P 1 - XX | P 2 - XX | P 3 - XX | P 4 - XX | P 5 - XX | P 6 - XX | P 7 - XX

V 1.651	★	▲	●	■	Code	Value	Unit	Min	Max	Default	Description
P0 - 27				■	MAP3	0x00000000		*	*	0x00000000	Mapping Paramenter #3
P0 - 28				■	MAP4	0x00000000		*	*	0x00000000	Mapping Paramenter #4
P0 - 29				■	MAP5	0x00000000		*	*	0x00000000	Mapping Paramenter #5
P0 - 30				■	MAP6	0x00000000		*	*	0x00000000	Mapping Paramenter #6
P0 - 31				■	MAP7	0x00000000		*	*	0x00000000	Mapping Paramenter #7
P0 - 32				■	MAP8	0x00000000		*	*	0x00000000	Mapping Paramenter #8
P0 - 33	★					0		-32768	32767	0	Reserved
P0 - 34	★					0		-32768	32767	0	Reserved
P0 - 35					MAP1A	0x00110011		0x00000000	0xFFFFFFFF	0x00110011	Target Setting of Mapping Parameter P0-25
P0 - 36					MAP2A	0x00110011		0x00000000	0xFFFFFFFF	0x00110011	Target Setting of Mapping Parameter P0-26
P0 - 37					MAP3A	0x00110011		0x00000000	0xFFFFFFFF	0x00110011	Target Setting of Mapping Parameter P0-27
P0 - 38					MAP4A	0x00110011		0x00000000	0xFFFFFFFF	0x00110011	Target Setting of Mapping Parameter P0-28
P0 - 39					MAP5A	0x00110011		0x00000000	0xFFFFFFFF	0x00110011	Target Setting of Mapping Parameter P0-29
P0 - 40					MAP6A	0x00110011		0x00000000	0xFFFFFFFF	0x00110011	Target Setting of Mapping Parameter P0-30
P0 - 41					MAP7A	0x00110011		0x00000000	0xFFFFFFFF	0x00110011	Target Setting of Mapping Parameter P0-31
P0 - 42					MAP8A	0x00110011		0x00000000	0xFFFFFFFF	0x00110011	Target Setting of Mapping Parameter P0-32
P0 - 43	★					0		-32768	32767	0	Reserved
P0 - 44	★			■		17911151		-2147483648	2147483647	0	Status Monitor Register (for PC Software)
P0 - 45				■		0		0	127	0	Status Monitor Register Selection (for PC Software)
P0 - 46	★			■	SVSTS	0x0183		0x0000	0xFFFF	0x0000	Servo Digital Output Status Display
P0 - 47	★			■		0x00000000		0x00000000	0xFFFFFFFF	0x00000000	No
P0 - 48	★			■		0x00000000		0x00000000	0xFFFFFFFF	0x00000000	No
P0 - 49				■		0x0000		0x0000	0x0002	0x0000	Renew Encoder Absolute Position
P0 - 50	★			■		0x0000		0x0000	0xFFFF	0x0000	Absolute Coordinate System Status
P0 - 51	★			■		0	rev	-32768	32767	0	Encoder Absolute Position (Multiturn)
P0 - 52	★			■		0	Pulse or PUU	-2147483648	2147483647	0	Encoder Absolute Position (Pulse number within Singleturn or PUU)

★ Read-Only ▲ Set When Servo OFF ● Valid After Re-power on ■ Volatile Parameter Parameter for three axes

Firmware Version: 1.651.408 Note: Double-click the Value can be call out the Parameter Setting Helper

P1-XX



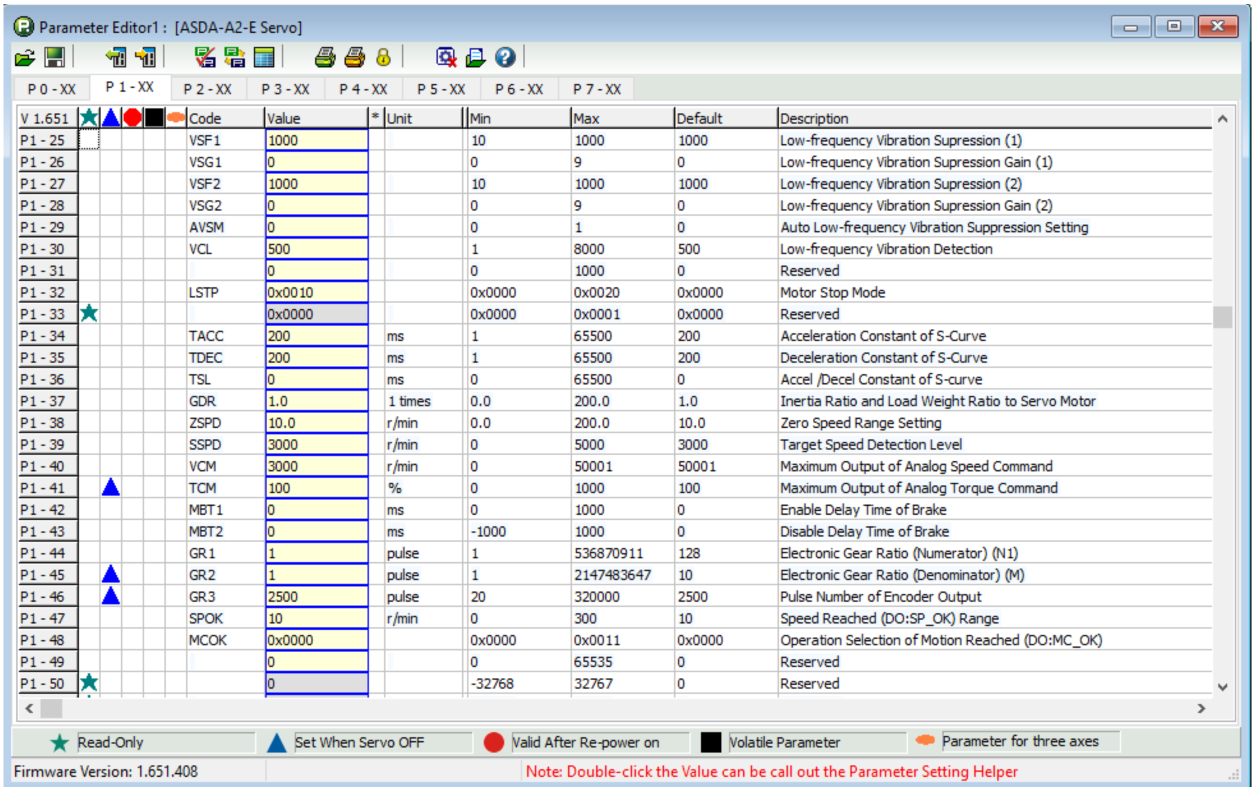
Parameter Editor1 : [ASDA-A2-E Servo]

P 0 - XX | P 1 - XX | P 2 - XX | P 3 - XX | P 4 - XX | P 5 - XX | P 6 - XX | P 7 - XX

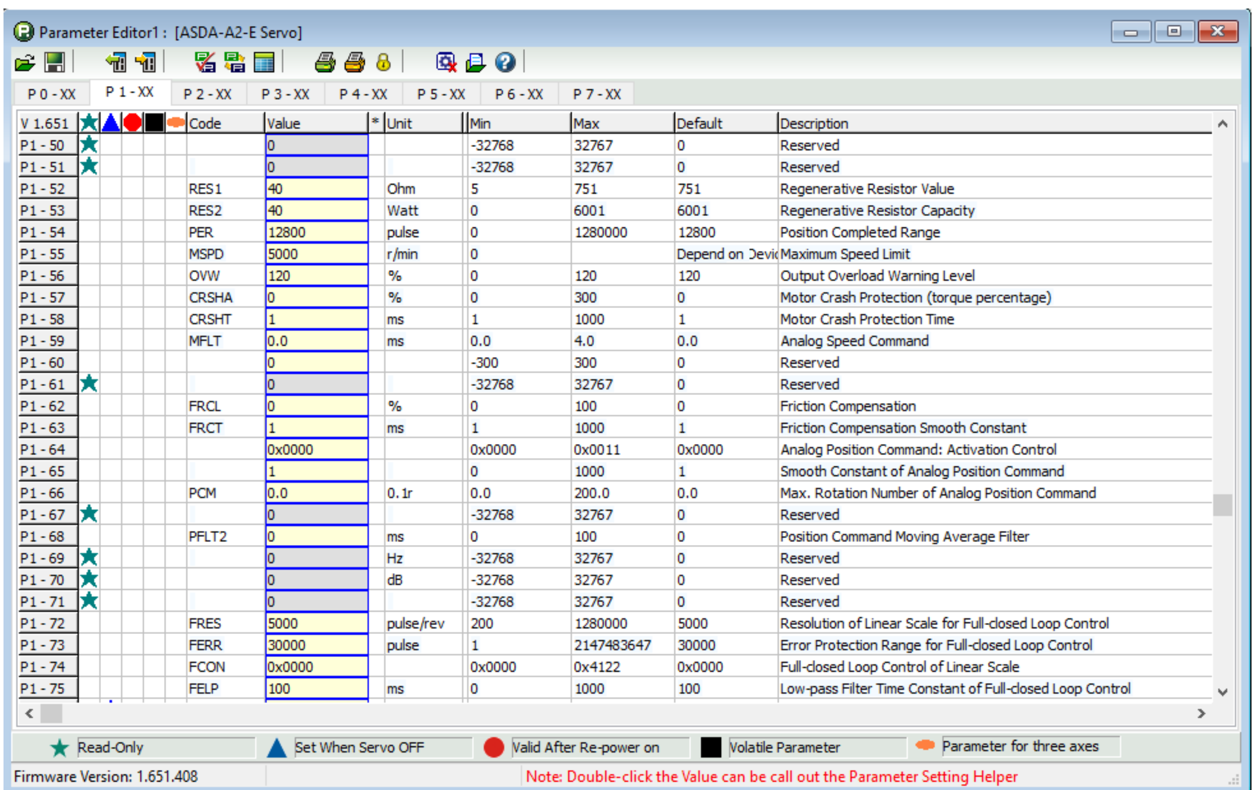
V 1.651	★	▲	●	■	Code	Value	Unit	Min	Max	Default	Description
P1 - 00		▲			PTT	0x0002		0x0000	0x1142	0x0002	The Type of External Pulse Input
P1 - 01			●		CTL	0x000C		0x0000	0x111F	0x000C	Input Setting of Control Mode and Control Command
P1 - 02		▲			PSTL	0x0000		0x0000	0x0011	0x0000	Speed and Torque Limit Setting
P1 - 03					AOUT	0x0000		0x0000	0x0013	0x0000	Polarity Setting of Encoder Pulse Output
P1 - 04					MON1	100	%	0	100	100	MON1 Analog Monitor Output Proportion
P1 - 05					MON2	100	%	0	100	100	MON2 Analog Monitor Output Proportion
P1 - 06					SFLT	0	ms	0	1000	0	Analog Speed Command (Low-pass Filter)
P1 - 07					TFLT	0	ms	0	1000	0	Analog Torque Command (Low-pass Filter)
P1 - 08					PFLT	0	10ms	0	1000	0	Smooth Constant of Position Command (Low-pass Filter)
P1 - 09					SP1	1000	0.1r/min	-60000	60000	1000	Internal Speed Command 1 / Internal Speed Limit 1
P1 - 10					SP2	2000	0.1r/min	-60000	60000	2000	Internal Speed Command 2 / Internal Speed Limit 2
P1 - 11					SP3	3000	0.1r/min	-60000	60000	3000	Internal Speed Command 3 / Internal Speed Limit 3
P1 - 12					TQ1	100	%	-300	300	100	Internal Torque Command 1 / Internal Torque Limit 1
P1 - 13					TQ2	100	%	-300	300	100	Internal Torque Command 2 / Internal Torque Limit 2
P1 - 14					TQ3	100	%	-300	300	100	Internal Torque Command 3 / Internal Torque Limit 3
P1 - 15	★					0		-32768	32767	0	CAPTURE SYNC AXIS – Threshold of Correction
P1 - 16	★					0		-32768	32767	0	CAPTURE SYNC AXIS – Offset Compensation
P1 - 17						0.000	ms	-20.000	20.000	0.000	Tracking Error Compensation – Additional Time Setting
P1 - 18	★					0	ms	-32768	32767	0	Electronoic Cam (E-Cam) Pulse Phase Compensation – Time Setting
P1 - 19	★					0		-32768	32767	0	CAPTURE / COMPARE – Additional Function Settings
P1 - 20	★					0		-32768	32767	0	CAPTURE – Masking Range Setting
P1 - 21	★					0	Kpps	-32768	32767	0	Electronoic Cam (E-Cam) Pulse Phase Compensation – Min. Frequency
P1 - 22	★					0		-32768	32767	0	PR Special Filter Settings
P1 - 23	★					0		-32768	32767	0	Compare data offset (Value of P1-23 are remain unchanged)
P1 - 24	★					0		-32768	32767	0	COMPARE – Compare data offset(Automatically set P1-24 to 0)
P1 - 25					VSF1	1000		10	1000	1000	Low-frequency Vibration Supression (1)

★ Read-Only ▲ Set When Servo OFF ● Valid After Re-power on ■ Volatile Parameter Parameter for three axes

Firmware Version: 1.651.408 Note: Double-click the Value can be call out the Parameter Setting Helper



V 1.651	★	▲	Code	Value	Unit	Min	Max	Default	Description
P1 - 25			VSF1	1000		10	1000	1000	Low-frequency Vibration Supression (1)
P1 - 26			VSG1	0		0	9	0	Low-frequency Vibration Supression Gain (1)
P1 - 27			VSF2	1000		10	1000	1000	Low-frequency Vibration Supression (2)
P1 - 28			VSG2	0		0	9	0	Low-frequency Vibration Supression Gain (2)
P1 - 29			AVSM	0		0	1	0	Auto Low-frequency Vibration Suppression Setting
P1 - 30			VCL	500		1	8000	500	Low-frequency Vibration Detection
P1 - 31				0		0	1000	0	Reserved
P1 - 32			LSTP	0x0010		0x0000	0x0020	0x0000	Motor Stop Mode
P1 - 33	★			0x0000		0x0000	0x0001	0x0000	Reserved
P1 - 34			TACC	200	ms	1	65500	200	Acceleration Constant of S-Curve
P1 - 35			TDEC	200	ms	1	65500	200	Deceleration Constant of S-Curve
P1 - 36			TSL	0	ms	0	65500	0	Accel /Decel Constant of S-curve
P1 - 37			GDR	1.0	1 times	0.0	200.0	1.0	Inertia Ratio and Load Weight Ratio to Servo Motor
P1 - 38			ZSPD	10.0	r/min	0.0	200.0	10.0	Zero Speed Range Setting
P1 - 39			SSPD	3000	r/min	0	5000	3000	Target Speed Detection Level
P1 - 40			VCM	3000	r/min	0	50001	50001	Maximum Output of Analog Speed Command
P1 - 41		▲	TCM	100	%	0	1000	100	Maximum Output of Analog Torque Command
P1 - 42			MBT1	0	ms	0	1000	0	Enable Delay Time of Brake
P1 - 43			MBT2	0	ms	-1000	1000	0	Disable Delay Time of Brake
P1 - 44			GR1	1	pulse	1	536870911	128	Electronic Gear Ratio (Numerator) (N1)
P1 - 45		▲	GR2	1	pulse	1	2147483647	10	Electronic Gear Ratio (Denominator) (M)
P1 - 46		▲	GR3	2500	pulse	20	320000	2500	Pulse Number of Encoder Output
P1 - 47			SPOK	10	r/min	0	300	10	Speed Reached (DO:SP_OK) Range
P1 - 48			MCOK	0x0000		0x0000	0x0011	0x0000	Operation Selection of Motion Reached (DO:MC_OK)
P1 - 49				0		0	65535	0	Reserved
P1 - 50	★			0		-32768	32767	0	Reserved



V 1.651	★	▲	Code	Value	Unit	Min	Max	Default	Description
P1 - 50	★			0		-32768	32767	0	Reserved
P1 - 51	★			0		-32768	32767	0	Reserved
P1 - 52			RES1	40	Ohm	5	751	751	Regenerative Resistor Value
P1 - 53			RES2	40	Watt	0	6001	6001	Regenerative Resistor Capacity
P1 - 54			PER	12800	pulse	0	1280000	12800	Position Completed Range
P1 - 55			MSPD	5000	r/min	0		Depend on Devi	Maximum Speed Limit
P1 - 56			OVW	120	%	0	120	120	Output Overload Warning Level
P1 - 57			CRSHA	0	%	0	300	0	Motor Crash Protection (torque percentage)
P1 - 58			CRSHT	1	ms	1	1000	1	Motor Crash Protection Time
P1 - 59			MFLT	0.0	ms	0.0	4.0	0.0	Analog Speed Command
P1 - 60				0		-300	300	0	Reserved
P1 - 61	★			0		-32768	32767	0	Reserved
P1 - 62			FRCL	0	%	0	100	0	Friction Compensation
P1 - 63			FRCT	1	ms	1	1000	1	Friction Compensation Smooth Constant
P1 - 64				0x0000		0x0000	0x0011	0x0000	Analog Position Command: Activation Control
P1 - 65				1		0	1000	1	Smooth Constant of Analog Position Command
P1 - 66			PCM	0.0	0. 1r	0.0	200.0	0.0	Max. Rotation Number of Analog Position Command
P1 - 67	★			0		-32768	32767	0	Reserved
P1 - 68			PFLT2	0	ms	0	100	0	Position Command Moving Average Filter
P1 - 69	★			0	Hz	-32768	32767	0	Reserved
P1 - 70	★			0	dB	-32768	32767	0	Reserved
P1 - 71	★			0		-32768	32767	0	Reserved
P1 - 72			FRES	5000	pulse/rev	200	1280000	5000	Resolution of Linear Scale for Full-closed Loop Control
P1 - 73			FERR	30000	pulse	1	2147483647	30000	Error Protection Range for Full-closed Loop Control
P1 - 74			FCON	0x0000		0x0000	0x4122	0x0000	Full-closed Loop Control of Linear Scale
P1 - 75			FELP	100	ms	0	1000	100	Low-pass Filter Time Constant of Full-closed Loop Control

Parameter Editor1 : [ASDA-A2-E Servo]

P 0 - XX | P 1 - XX | P 2 - XX | P 3 - XX | P 4 - XX | P 5 - XX | P 6 - XX | P 7 - XX

V 1.651	★	▲	●	■	Code	Value	*	Unit	Min	Max	Default	Description
P1 - 73					FERR	30000		pulse	1	2147483647	30000	Error Protection Range for Full-closed Loop Control
P1 - 74					FCON	0x0000			0x0000	0x4122	0x0000	Full-closed Loop Control of Linear Scale
P1 - 75					FELP	100		ms	0	1000	100	Low-pass Filter Time Constant of Full-closed Loop Control
P1 - 76		▲			AMSPD	5500		r/min	0	6000	5500	Max. Rotation of Encoder Output Setting (OA, OB)
P1 - 77		▲				0x0000			0x0000	0x0001	0x0000	Reserved
P1 - 78	★					0			-32768	32767	0	Reserved
P1 - 79	★					0			-32768	32767	0	Reserved
P1 - 80	★					0			-32768	32767	0	Reserved
P1 - 81	★					0		rpm/10V	-32768	32767	0	Max. Speed of 2nd Analog Speed Command
P1 - 82	★					0		ms	-32768	32767	0	Filter Switching Time Between P1-40 and P1-81
P1 - 83	★					0		mV	-32768	32767	0	Abnormal Analog Input Voltage Level
P1 - 84	★					0			-32768	32767	0	
P1 - 85	★					0			-32768	32767	0	
P1 - 86	★					0			-32768	32767	0	
P1 - 87	★					0			-32768	32767	0	
P1 - 88	★					0			-32768	32767	0	
P1 - 89	★					0			-32768	32767	0	
P1 - 90	★					0			-32768	32767	0	
P1 - 91	★					0			-32768	32767	0	
P1 - 92	★					0			-32768	32767	0	
P1 - 93	★					0			-32768	32767	0	
P1 - 94	★					0			-32768	32767	0	
P1 - 95	★					0			-32768	32767	0	
P1 - 96	★					0			-32768	32767	0	
P1 - 97	★					0			-32768	32767	0	
P1 - 98						0			0	800	0	

★ Read-Only ▲ Set When Servo OFF ● Valid After Re-power on ■ Volatile Parameter Parameter for three axes

Firmware Version: 1.651.408 Note: Double-click the Value can be call out the Parameter Setting Helper

P2-XX

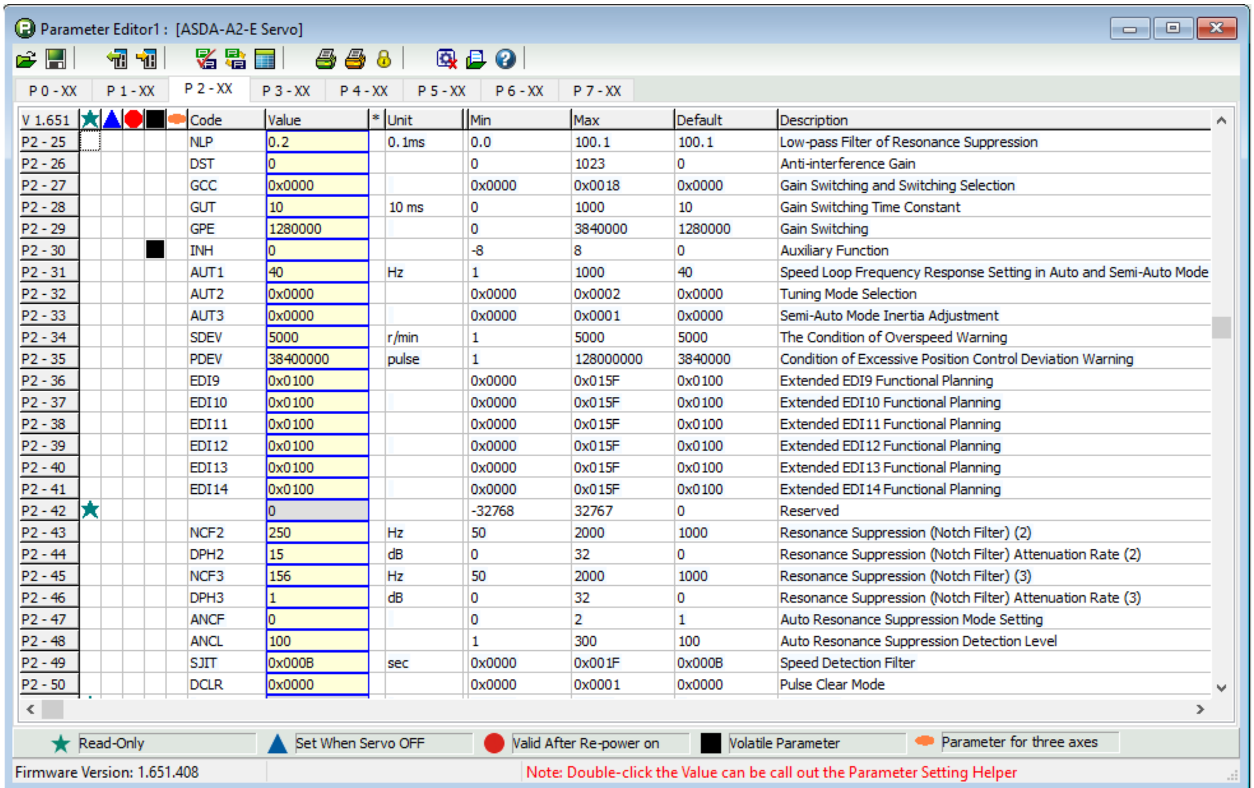
Parameter Editor1 : [ASDA-A2-E Servo]

P 0 - XX | P 1 - XX | P 2 - XX | P 3 - XX | P 4 - XX | P 5 - XX | P 6 - XX | P 7 - XX

V 1.651	★	▲	●	■	Code	Value	*	Unit	Min	Max	Default	Description
P2 - 00					KPP	35		rad/s	0	2047	35	Position Loop Gain
P2 - 01					PPR	100		%	10	500	100	Switching Rate of Position Loop Gain
P2 - 02					PFG	50		%	0	100	50	Position Feed Forward Gain
P2 - 03					PFF	5		ms	2	100	5	Smooth Constant of Position Feed Forward Gain
P2 - 04					KVP	500		rad/s	0	8191	500	Speed Loop Gain
P2 - 05					SPR	100		%	10	500	100	Switching Rate of Speed Loop Gain
P2 - 06					KVI	100		rad/s	0	1023	100	Speed Integral Compensation
P2 - 07					KVF	0		%	0	100	0	Speed Feed Forward Gain
P2 - 08				■	PCTL	36			0	501	0	Special Parameter Write-in
P2 - 09					DRT	2		2ms	0	20	2	DI Debouncing Time
P2 - 10					DI1	0x0101			0x0000	0x015F	0x0100	DI1 Functional Planning
P2 - 11					DI2	0x0100			0x0000	0x015F	0x0100	DI2 Functional Planning
P2 - 12					DI3	0x0100			0x0000	0x015F	0x0100	DI3 Functional Planning
P2 - 13					DI4	0x0124			0x0000	0x015F	0x0124	DI4 Functional Planning
P2 - 14					DI5	0x0022			0x0000	0x015F	0x0022	DI5 Functional Planning
P2 - 15					DI6	0x0023			0x0000	0x015F	0x0023	DI6 Functional Planning
P2 - 16					DI7	0x0021			0x0000	0x015F	0x0021	DI7 Functional Planning
P2 - 17					DI8	0x0100			0x0000	0x015F	0x0100	DI8 Functional Planning
P2 - 18					DO1	0x0101			0x0000	0x013F	0x0101	DO1 Functional Planning
P2 - 19					DO2	0x0100			0x0000	0x013F	0x0100	DO2 Functional Planning)
P2 - 20					DO3	0x0100			0x0000	0x013F	0x0100	DO3 Functional Planning
P2 - 21					DO4	0x0007			0x0000	0x013F	0x0007	DO4 Functional Planning
P2 - 22					DO5	0x0100			0x0000	0x013F	0x0100	DO5 Functional Planning
P2 - 23					NCF1	1000		Hz	50	1000	1000	Resonance Suppression (Notch Filter) (1)
P2 - 24					DPH1	0		dB	0	32	0	Resonance Suppression (Notch Filter) Attenuation Rate (1)
P2 - 25					NLP	0.2		0.1ms	0.0	100.1	100.1	Low-pass Filter of Resonance Suppression

★ Read-Only ▲ Set When Servo OFF ● Valid After Re-power on ■ Volatile Parameter Parameter for three axes

Firmware Version: 1.651.408 Note: Double-click the Value can be call out the Parameter Setting Helper



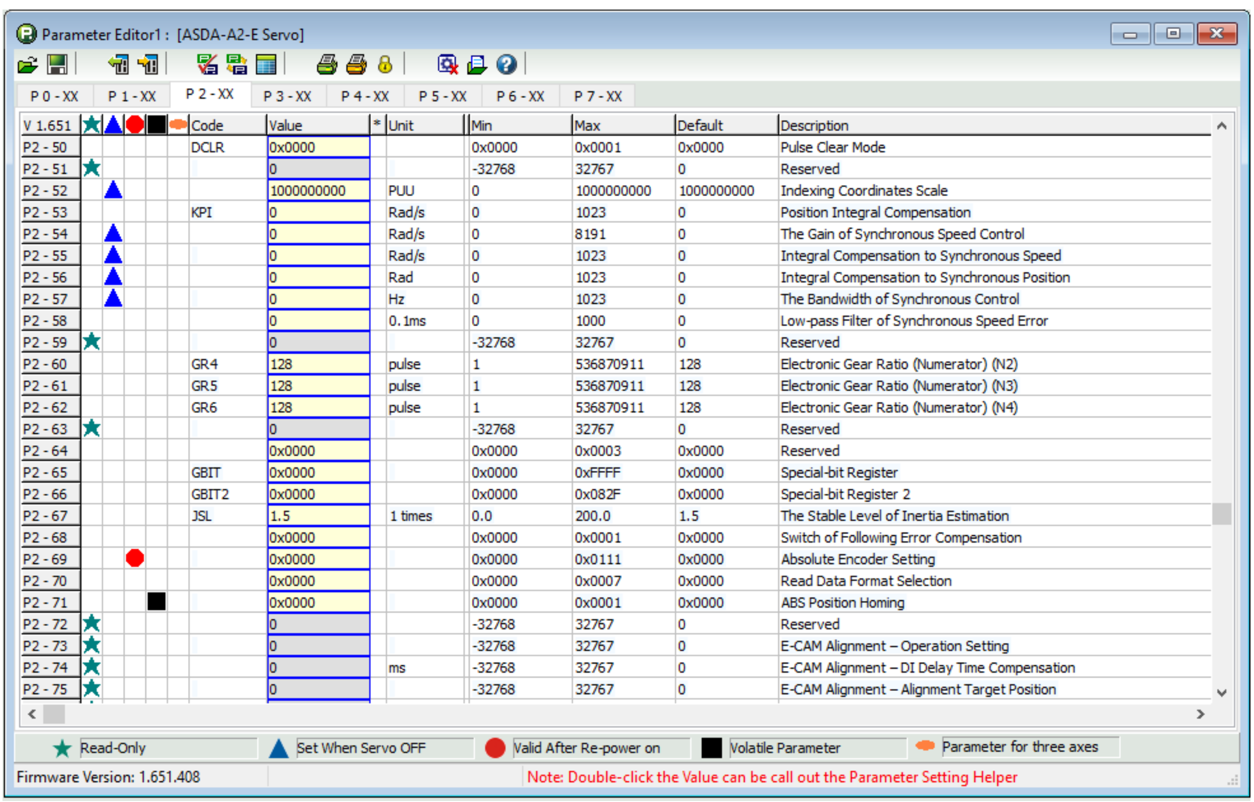
Parameter Editor1 : [ASDA-A2-E Servo]

P 0 - XX | P 1 - XX | P 2 - XX | P 3 - XX | P 4 - XX | P 5 - XX | P 6 - XX | P 7 - XX

V 1.651	★	▲	●	■	Code	Value	Unit	Min	Max	Default	Description
P2 - 25					NLP	0.2	0.1ms	0.0	100.1	100.1	Low-pass Filter of Resonance Suppression
P2 - 26					DST	0		0	1023	0	Anti-interference Gain
P2 - 27					GCC	0x0000		0x0000	0x0018	0x0000	Gain Switching and Switching Selection
P2 - 28					GUT	10	10 ms	0	1000	10	Gain Switching Time Constant
P2 - 29					GPE	1280000		0	3840000	1280000	Gain Switching
P2 - 30				■	INH	0		-8	8	0	Auxiliary Function
P2 - 31					AUT1	40	Hz	1	1000	40	Speed Loop Frequency Response Setting in Auto and Semi-Auto Mode
P2 - 32					AUT2	0x0000		0x0000	0x0002	0x0000	Tuning Mode Selection
P2 - 33					AUT3	0x0000		0x0000	0x0001	0x0000	Semi-Auto Mode Inertia Adjustment
P2 - 34					SDEV	5000	r/min	1	5000	5000	The Condition of Overspeed Warning
P2 - 35					PDEV	38400000	pulse	1	128000000	3840000	Condition of Excessive Position Control Deviation Warning
P2 - 36					EDI9	0x0100		0x0000	0x015F	0x0100	Extended EDI9 Functional Planning
P2 - 37					EDI10	0x0100		0x0000	0x015F	0x0100	Extended EDI10 Functional Planning
P2 - 38					EDI11	0x0100		0x0000	0x015F	0x0100	Extended EDI11 Functional Planning
P2 - 39					EDI12	0x0100		0x0000	0x015F	0x0100	Extended EDI12 Functional Planning
P2 - 40					EDI13	0x0100		0x0000	0x015F	0x0100	Extended EDI13 Functional Planning
P2 - 41					EDI14	0x0100		0x0000	0x015F	0x0100	Extended EDI14 Functional Planning
P2 - 42	★					0		-32768	32767	0	Reserved
P2 - 43					NCF2	250	Hz	50	2000	1000	Resonance Suppression (Notch Filter) (2)
P2 - 44					DPH2	15	dB	0	32	0	Resonance Suppression (Notch Filter) Attenuation Rate (2)
P2 - 45					NCF3	156	Hz	50	2000	1000	Resonance Suppression (Notch Filter) (3)
P2 - 46					DPH3	1	dB	0	32	0	Resonance Suppression (Notch Filter) Attenuation Rate (3)
P2 - 47					ANCF	0		0	2	1	Auto Resonance Suppression Mode Setting
P2 - 48					ANCL	100		1	300	100	Auto Resonance Suppression Detection Level
P2 - 49					SJIT	0x000B	sec	0x0000	0x001F	0x000B	Speed Detection Filter
P2 - 50					DCLR	0x0000		0x0000	0x0001	0x0000	Pulse Clear Mode

★ Read-Only ▲ Set When Servo OFF ● Valid After Re-power on ■ Volatile Parameter Parameter for three axes

Firmware Version: 1.651.408 Note: Double-click the Value can be call out the Parameter Setting Helper



Parameter Editor1 : [ASDA-A2-E Servo]

P 0 - XX | P 1 - XX | P 2 - XX | P 3 - XX | P 4 - XX | P 5 - XX | P 6 - XX | P 7 - XX

V 1.651	★	▲	●	■	Code	Value	Unit	Min	Max	Default	Description
P2 - 50					DCLR	0x0000		0x0000	0x0001	0x0000	Pulse Clear Mode
P2 - 51	★					0		-32768	32767	0	Reserved
P2 - 52		▲				1000000000	PUU	0	1000000000	1000000000	Indexing Coordinates Scale
P2 - 53					KPI	0	Rad/s	0	1023	0	Position Integral Compensation
P2 - 54		▲				0	Rad/s	0	8191	0	The Gain of Synchronous Speed Control
P2 - 55		▲				0	Rad/s	0	1023	0	Integral Compensation to Synchronous Speed
P2 - 56		▲				0	Rad	0	1023	0	Integral Compensation to Synchronous Position
P2 - 57		▲				0	Hz	0	1023	0	The Bandwidth of Synchronous Control
P2 - 58						0	0.1ms	0	1000	0	Low-pass Filter of Synchronous Speed Error
P2 - 59	★					0		-32768	32767	0	Reserved
P2 - 60					GR4	128	pulse	1	536870911	128	Electronic Gear Ratio (Numerator) (N2)
P2 - 61					GR5	128	pulse	1	536870911	128	Electronic Gear Ratio (Numerator) (N3)
P2 - 62					GR6	128	pulse	1	536870911	128	Electronic Gear Ratio (Numerator) (N4)
P2 - 63	★					0		-32768	32767	0	Reserved
P2 - 64						0x0000		0x0000	0x0003	0x0000	Reserved
P2 - 65					GBIT	0x0000		0x0000	0xFFFF	0x0000	Special-bit Register
P2 - 66					GBIT2	0x0000		0x0000	0x082F	0x0000	Special-bit Register 2
P2 - 67					JSL	1.5	1 times	0.0	200.0	1.5	The Stable Level of Inertia Estimation
P2 - 68						0x0000		0x0000	0x0001	0x0000	Switch of Following Error Compensation
P2 - 69			●			0x0000		0x0000	0x0111	0x0000	Absolute Encoder Setting
P2 - 70						0x0000		0x0000	0x0007	0x0000	Read Data Format Selection
P2 - 71				■		0x0000		0x0000	0x0001	0x0000	ABS Position Homing
P2 - 72	★					0		-32768	32767	0	Reserved
P2 - 73	★					0		-32768	32767	0	E-CAM Alignment – Operation Setting
P2 - 74	★					0	ms	-32768	32767	0	E-CAM Alignment – DI Delay Time Compensation
P2 - 75	★					0		-32768	32767	0	E-CAM Alignment – Alignment Target Position

★ Read-Only ▲ Set When Servo OFF ● Valid After Re-power on ■ Volatile Parameter Parameter for three axes

Firmware Version: 1.651.408 Note: Double-click the Value can be call out the Parameter Setting Helper

P3-XX

Parameter Editor1 : [ASDA-A2-E Servo]

P 0 - XX | P 1 - XX | P 2 - XX | P 3 - XX | P 4 - XX | P 5 - XX | P 6 - XX | P 7 - XX

V 1.651	★	▲	●	■	Code	Value	Unit	Min	Max	Default	Description
P3 - 00			●		ADR	0x007F		0x0001	0x007F	0x007F	Communication Address Setting
P3 - 01					BRT	0x0203	bps	0x0000	0x0405	0x0203	Transmission Speed
P3 - 02					PTL	0x0006		0x0000	0x0008	0x0006	Communication Protocol
P3 - 03					FLT	0x0000		0x0000	0x0001	0x0000	Communication Error Disposal
P3 - 04					CWD	0	sec	0	20	0	Communication Time Out
P3 - 05					CMM	0x0000		0x0000	0x0001	0x0000	Communication Mechanism
P3 - 06				■	SDI	0x0000		0x0000	0x3FFF	0x0000	Control Switch of Digital Input (DI)
P3 - 07					CDT	0	0.5ms	0	1000	0	Communication Response Delay Time
P3 - 08				■	MNS	0x0000		0x0000	0x00F3	0x0000	Monitor Mode
P3 - 09		▲			SYC	0x0100		0x0100	0x0F00	0x0100	CANopen Synchronize Setting
P3 - 10						0x0000		0x0000	0xFFFF	0x0000	Reserved
P3 - 11				■		0x0000		0x0000	0x0001	0x0000	Reserved
P3 - 12			●			0x0000		0x0000	0x0111	0x0000	
P3 - 13						0x0000		0x0000	0xFFFF	0x0000	
P3 - 14	★					0		-32768	32767	0	
P3 - 15	★					0		-32768	32767	0	
P3 - 16	★					0		-32768	32767	0	
P3 - 17	★					0		-32768	32767	0	
P3 - 18			●			0x00002000		0x00000000	0x00112111	0x00002000	
P3 - 19						0x00000021		0x00000000	0x0001FFFF	0x00000021	
P3 - 20	★					0		-32768	32767	0	
P3 - 21	★					0		-32768	32767	0	
P3 - 22			●			0xFF04		0x0002	0xFF14	0xFF04	

★ Read-Only ▲ Set When Servo OFF ● Valid After Re-power on ■ Volatile Parameter Parameter for three axes

Firmware Version: 1.651.408 Note: Double-click the Value can be call out the Parameter Setting Helper

P4-XX

Parameter Editor1 : [ASDA-A2-E Servo]

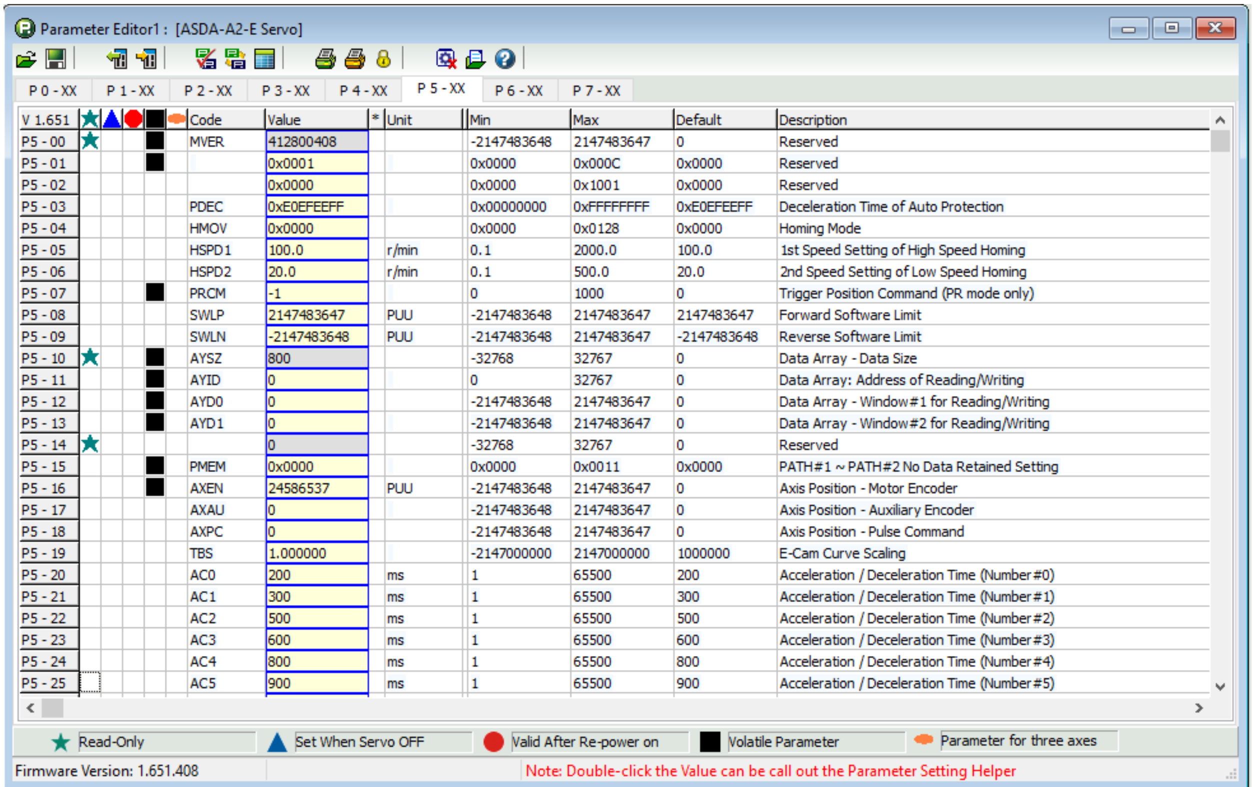
P 0 - XX | P 1 - XX | P 2 - XX | P 3 - XX | P 4 - XX | P 5 - XX | P 6 - XX | P 7 - XX

V 1.651	★	▲	●	■	Code	Value	Unit	Min	Max	Default	Description
P4 - 00				■	ASH1	0x81300180		0x00000000	0xFFFFFFFF	0x00000000	Fault Record (N)
P4 - 01	★			■	ASH2	0x620003E3		0x00000000	0xFFFFFFFF	0x00000000	Fault Record (N-1)
P4 - 02	★			■	ASH3	0x81300180		0x00000000	0xFFFFFFFF	0x00000000	Fault Record (N-2)
P4 - 03	★			■	ASH4	0x620003E3		0x00000000	0xFFFFFFFF	0x00000000	Fault Record (N-3)
P4 - 04	★			■	ASH5	0x81300180		0x00000000	0xFFFFFFFF	0x00000000	Fault Record (N-4)
P4 - 05					JOG	100	r/min	0	5000	20	JOG Operation
P4 - 06				■	FOT	0x0000		0x0000	0x00FF	0x0000	Digital Output Register (Readable and Writable)
P4 - 07				■	ITST	0x0000		0x0000	0x3FFF	0x0000	Multi-funciton of Digital Input
P4 - 08	★			■	PKEY	0x0000		0x0000	0x00FF	0x0000	Input Status of the Drive Keypad (Read-only)
P4 - 09	★			■	MOT	0x0009		0x0000	0x001F	0x0000	Digital Output Status
P4 - 10		▲		■	CEN	0		0	6	0	Adjustment Selection
P4 - 11					SOF1	16384		0	32767	16352	Analog Speed Input Offset Adjustment 1
P4 - 12					SOF2	16384		0	32767	16352	Analog Speed Input Offset Adjustment 2
P4 - 13					TOF1	16384		0	32767	16352	Analog Torque Input Offset Adjustment 1
P4 - 14					TOF2	16384		0	32767	16352	Analog Torque Input Offset Adjustment 2
P4 - 15					COF1	16680		0	32767	16352	Current Detector (V1 phase) offset Adjustment
P4 - 16					COF2	16463		0	32767	16352	Current Detector (V2 phase) Offset Adjustment
P4 - 17					COF3	16659		0	32767	16352	Current Detector (W1 phase) Offset Adjustment
P4 - 18					COF4	16456		0	32767	16352	Current Detector (W2 phase) Offset Adjustment
P4 - 19					TIGB	2		1	4	2	IGBT NTC Adjustment Deteciton Level (cannot reset)
P4 - 20					DOF1	0	mV	-800	800	0	Offset Adjustment Value of Analog Monitor (Ch1)
P4 - 21					DOF2	0	mV	-800	800	0	Offset Adjustment Value of Analog Monitor (Ch2)
P4 - 22					SAO	0	mV	-5000	5000	0	Analog Speed Input OFFSET
P4 - 23					TAO	0	mV	-5000	5000	0	Analog Torque Input OFFSET
P4 - 24					LVL	160	V	140	381	381	Level of Undervoltage Error

★ Read-Only ▲ Set When Servo OFF ● Valid After Re-power on ■ Volatile Parameter Parameter for three axes

Firmware Version: 1.651.408 Note: Double-click the Value can be call out the Parameter Setting Helper

P5-XX



V 1.651	★	▲	●	■	Code	Value	*	Unit	Min	Max	Default	Description
P5 - 00	★			■	MVER	412800408			-2147483648	2147483647	0	Reserved
P5 - 01				■		0x0001			0x0000	0x000C	0x0000	Reserved
P5 - 02						0x0000			0x0000	0x1001	0x0000	Reserved
P5 - 03					PDEC	0xE0EFEEFF			0x00000000	0xFFFFFFFF	0xE0EFEEFF	Deceleration Time of Auto Protection
P5 - 04					HMOV	0x0000			0x0000	0x0128	0x0000	Homing Mode
P5 - 05					HSPD1	100.0		r/min	0.1	2000.0	100.0	1st Speed Setting of High Speed Homing
P5 - 06					HSPD2	20.0		r/min	0.1	500.0	20.0	2nd Speed Setting of Low Speed Homing
P5 - 07				■	PRCM	-1			0	1000	0	Trigger Position Command (PR mode only)
P5 - 08					SWLP	2147483647		PUU	-2147483648	2147483647	2147483647	Forward Software Limit
P5 - 09					SWLN	-2147483648		PUU	-2147483648	2147483647	-2147483648	Reverse Software Limit
P5 - 10	★			■	AYSZ	800			-32768	32767	0	Data Array - Data Size
P5 - 11				■	AYID	0			0	32767	0	Data Array: Address of Reading/Writing
P5 - 12				■	AYD0	0			-2147483648	2147483647	0	Data Array - Window#1 for Reading/Writing
P5 - 13				■	AYD1	0			-2147483648	2147483647	0	Data Array - Window#2 for Reading/Writing
P5 - 14	★					0			-32768	32767	0	Reserved
P5 - 15				■	PMEM	0x0000			0x0000	0x0011	0x0000	PATH#1 ~ PATH#2 No Data Retained Setting
P5 - 16				■	AXEN	24586537		PUU	-2147483648	2147483647	0	Axis Position - Motor Encoder
P5 - 17					AXAU	0			-2147483648	2147483647	0	Axis Position - Auxiliary Encoder
P5 - 18					AXPC	0			-2147483648	2147483647	0	Axis Position - Pulse Command
P5 - 19					TBS	1.000000			-2147000000	2147000000	1000000	E-Cam Curve Scaling
P5 - 20					AC0	200		ms	1	65500	200	Acceleration / Deceleration Time (Number#0)
P5 - 21					AC1	300		ms	1	65500	300	Acceleration / Deceleration Time (Number#1)
P5 - 22					AC2	500		ms	1	65500	500	Acceleration / Deceleration Time (Number#2)
P5 - 23					AC3	600		ms	1	65500	600	Acceleration / Deceleration Time (Number#3)
P5 - 24					AC4	800		ms	1	65500	800	Acceleration / Deceleration Time (Number#4)
P5 - 25					AC5	900		ms	1	65500	900	Acceleration / Deceleration Time (Number#5)

★ Read-Only ▲ Set When Servo OFF ● Valid After Re-power on ■ Volatile Parameter Parameter for three axes

Firmware Version: 1.651.408 Note: Double-click the Value can be call out the Parameter Setting Helper

Appendix

D

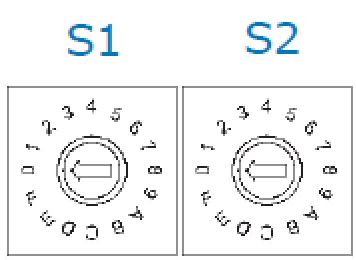
[D] Copley (Accelnet, Stepnet, Xenus Plus)

Copley Controls manufactures a number of drives. The drives currently supported are the single axis Accelnet servo drive, Xenus Plus line (800-1849, XE2) and Stepnet Plus TE2. SEM and TEL are supposed to work identical but have not been tested; although their product code is enabled for operation. Version 3.06 or greater firmware must be used (beta at time of testing). This section provides information that may be specific to this manufacturer.

eCAT_driveType – 2

Station Alias

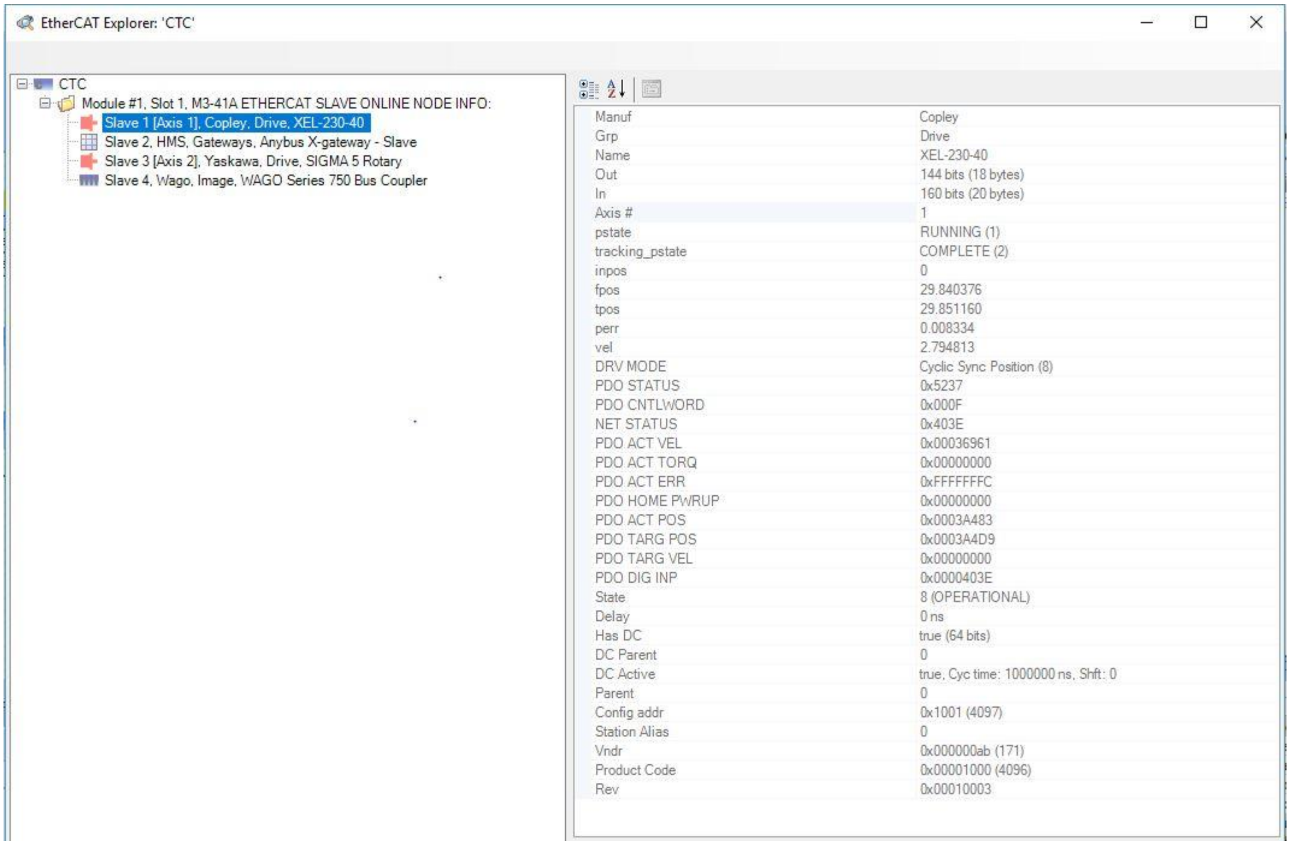
In an EtherCAT network, slaves are automatically assigned addresses based on their position in the bus. When a device, such as a drive, must have a fixed assigned identification that is independent of cabling, a Station Alias is needed. Accelnet provides two 16-position rotary switches with hexadecimal encoding for this purpose. This allows for a setting of 0 to 255 (FFh), where 0 defaults to the automatic address assignment. As an example if S1 is set to a 1 and S2 to an A this would be 1Ah or 1 X 16 + 10 = 26. Since the M3-41 only supports up to 16 drives S1 would always be set to 0 and only S2 used.



EtherCAT
Address Switch
Decimal values

	S1	S2
HEX	DEC	
0	0	0
1	16	1
2	32	2
3	48	3
4	64	4
5	80	5
6	96	6
7	112	7
8	128	8
9	144	9
A	160	10
B	176	11
C	192	12
D	208	13
E	224	14
F	240	15

EtherCAT Explorer View (Xenus XEL Single Axis)

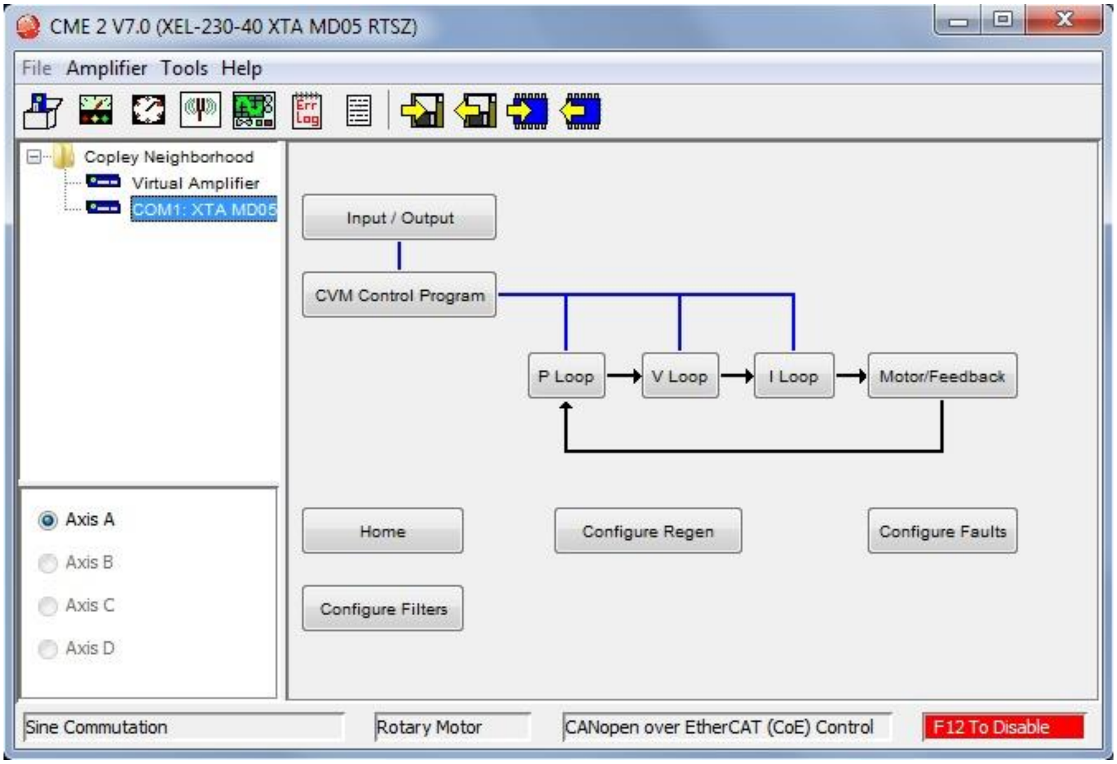


EtherCAT Explorer View (Xenus XE2 Dual Axis)

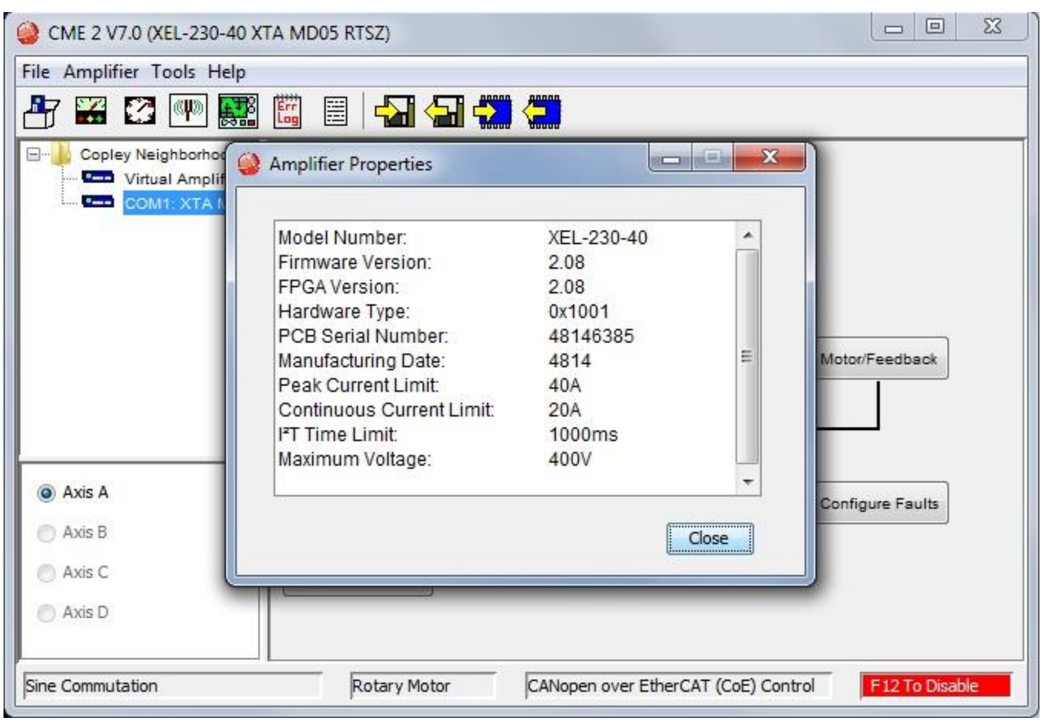
Manuf	Copley
Grp	Drive
Name	XE2-230-20
Out	272 bits (34 bytes)
In	288 bits (36 bytes)
------	------
Axis #	6
pstate	RUNNING (1)
tracking_pstate	COMPLETE (2)
inpos	0
fpos	9.892500
tpos	9.897000
perr	0.003500
vel	0.969600
DRV MODE	Cyclic Sync Position (8)
PDO STATUS	0x5237
PDO CNTLWORD	0x000F
------	------
Axis #	7
pstate	RUNNING (1)
tracking_pstate	COMPLETE (2)
inpos	0
fpos	9.891875
tpos	9.897000
perr	0.004125
vel	1.000975
DRV MODE	Cyclic Sync Position (8)
PDO STATUS	0x5237
PDO CNTLWORD	0x000F
------	------
State	8 (OPERATIONAL)
Delay	4210 ns
Has DC	true (64 bits)
DC Parent	1
DC Active	true, Cyc time: 1000000 ns, Shft: 0
Parent	5
Config addr	0x1006 (4102)
Station Alias	0
Vndr	0x000000ab (171)
Product Code	0x000010b0 (4272)
Rev	0x00010003

Copley Motion Explorer (CME 2)

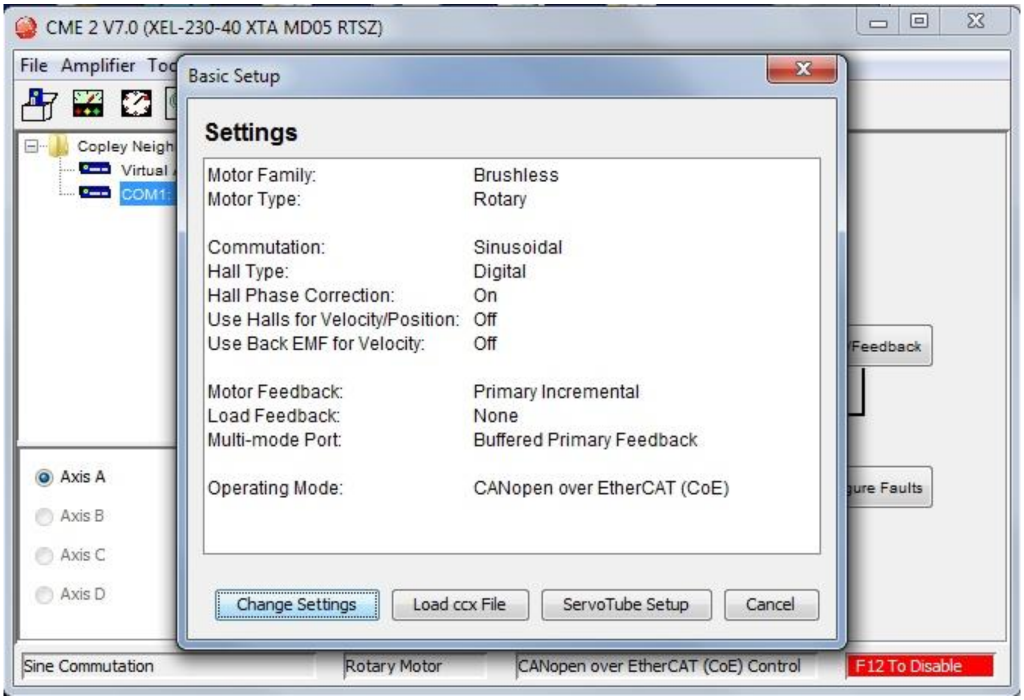
Overall Explorer screen:



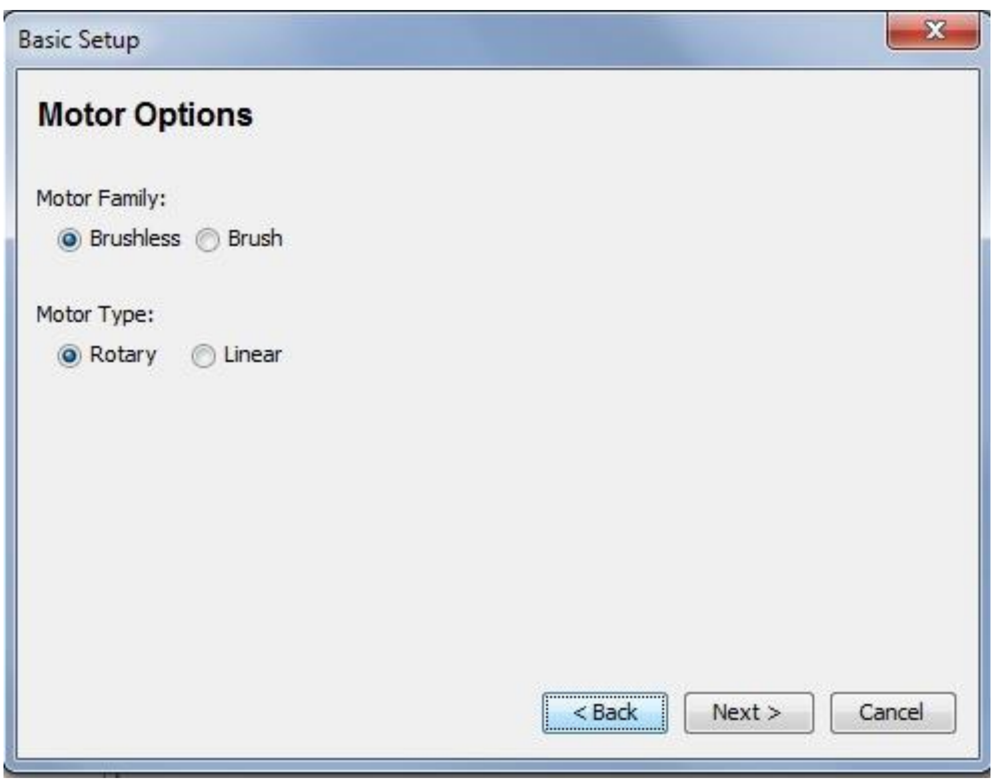
Amplifier Properties:

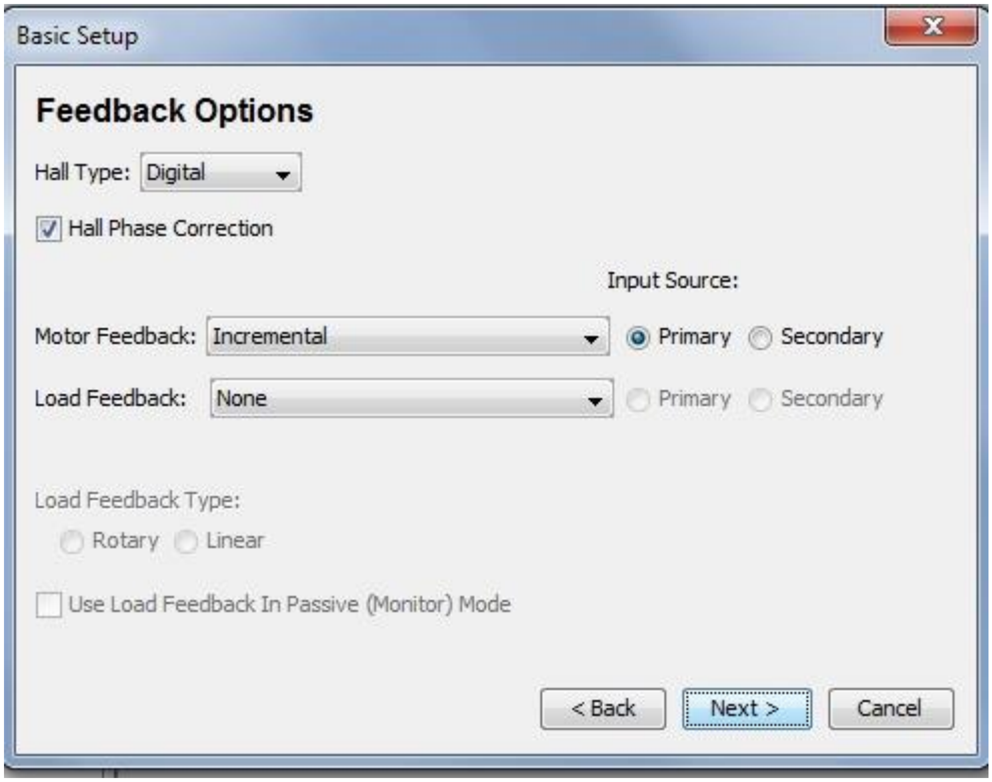


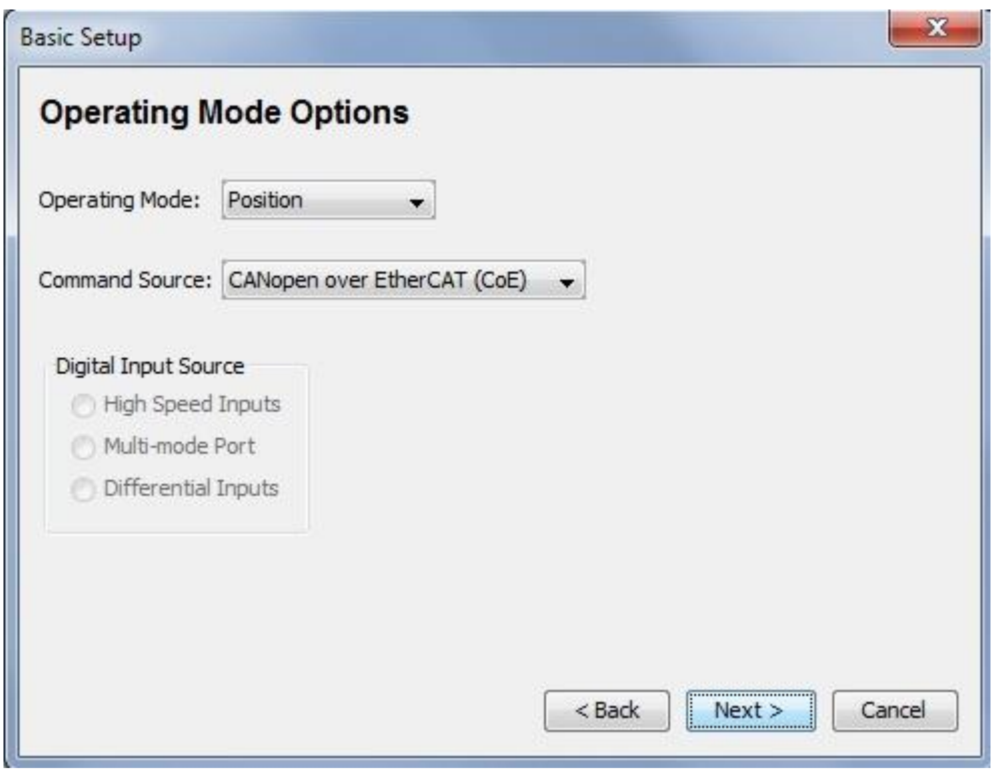
Basic Setup:

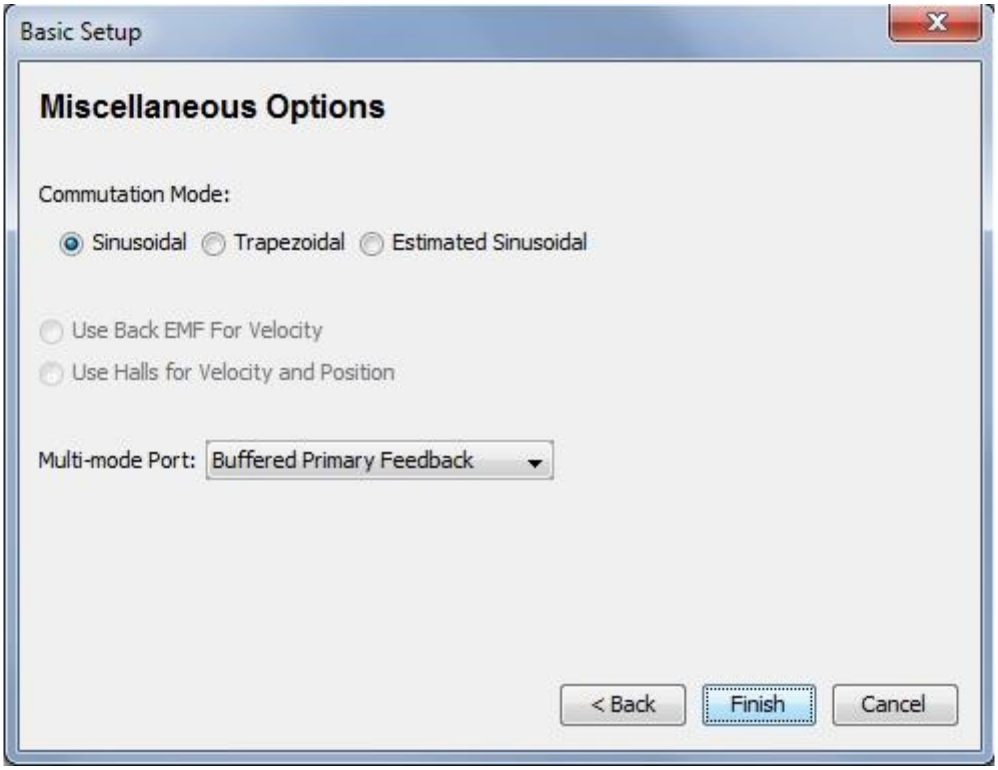


Change Settings:




Basic Setup
Feedback Options
Hall Type: Digital
Hall Phase Correction
Input Source:
Motor Feedback: Incremental
Primary
Secondary
Load Feedback: None
Primary
Secondary
Load Feedback Type:
Rotary
Linear
Use Load Feedback In Passive (Monitor) Mode
< Back
Next >
Cancel


Basic Setup
Operating Mode Options
Operating Mode: Position
Command Source: CANopen over EtherCAT (CoE)
Digital Input Source
High Speed Inputs
Multi-mode Port
Differential Inputs
< Back
Next >
Cancel


Basic Setup
Miscellaneous Options
Commutation Mode:
Sinusoidal
Trapezoidal
Estimated Sinusoidal
Use Back EMF For Velocity
Use Halls for Velocity and Position
Multi-mode Port:
Buffered Primary Feedback
< Back
Finish
Cancel

Appendix E

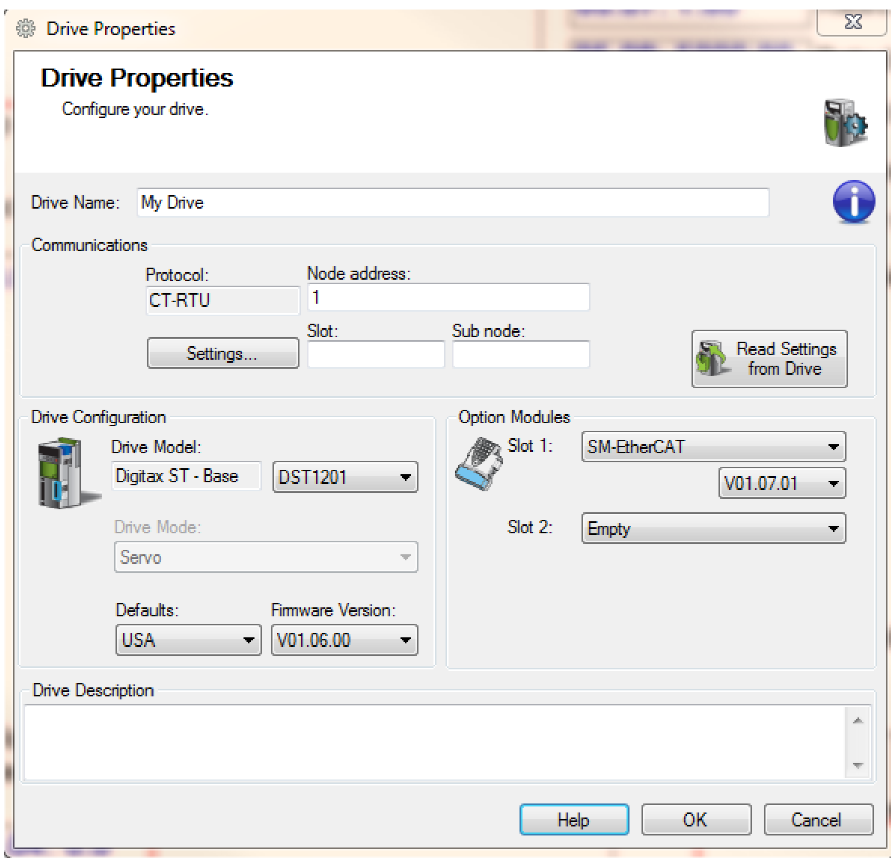
[E] Emerson (Control Techniques)

Emerson (Control Techniques) manufactures a number of drives. The drive currently supported is the single axis Unidrive SP and Digitax ST servo drive. This section provides information that may be specific to this manufacturer.

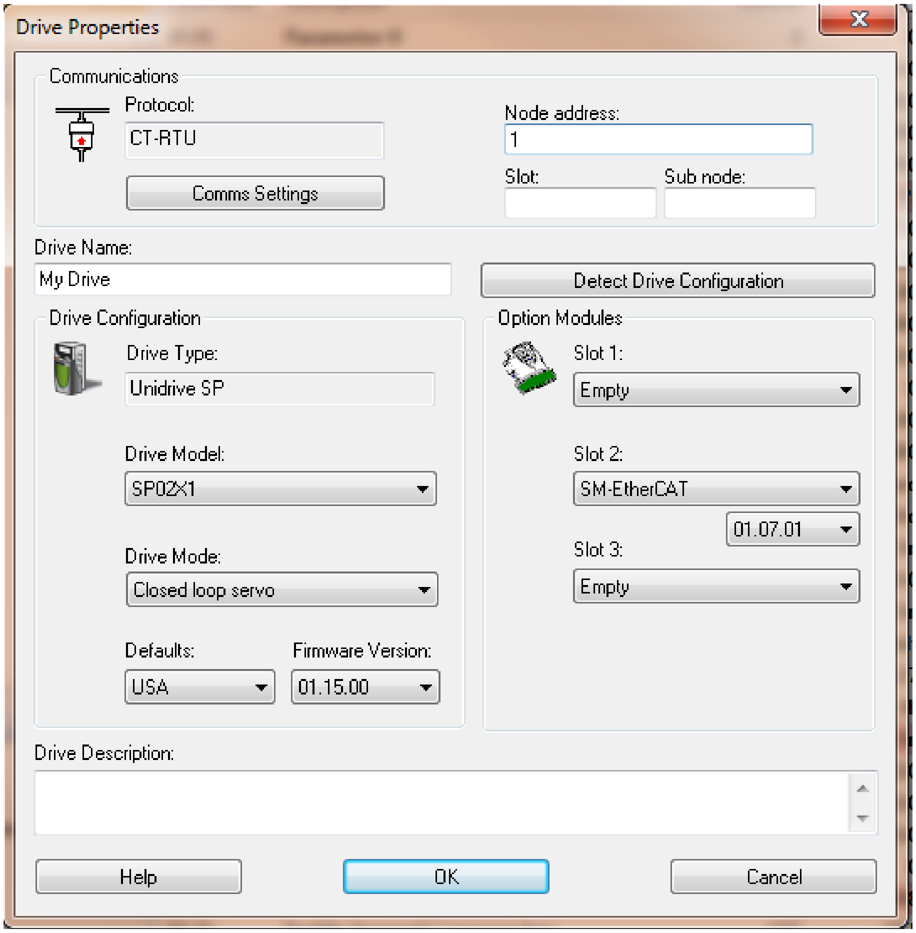
eCAT_driveType – 7

Drive Information & Firmware

Digitax:



Unidrive SP:



Inputs/Outputs

CTSoft Menu 8 is used to set up Digital I/O on the Unidrive SP. This is object 0x2008 when mapped to EtherCAT with the subindex being the menu item number. Thus item 20 is a read-only Digital I/O state object; item 20 in hex is subindex 0x14.

Bit	Digital I/O
0	T24 input / output 1
1	T25 input / output 2
2	T26 input / output 3
3	T27 input 4
4	T28 input 5
5	T29 input 6
6	Relay
7	T22 24V output
8	Secure disable

To activate outputs, a specific ‘sdo write’ operation is required, since the Emerson drive only has single-bit access to outputs, and the first 3 inputs are reconfigurable. In referencing the parameters as defined in the “Advanced User Guide Unidrive SP” (Part Number 0471-0002-10), subindex 31 to 33 enables T24 to T26 as outputs. Subindex 21 to 23 maps where the outputs would come from depending on whether an input or output.

DC Sync

Emerson drives must have DC Sync0 enabled prior to enabling the drive as well as a 32 or 64-bit slave reference as the first node on the EtherCAT network. Reference the ‘dcsync’ command example within the main body of this manual.

MSB code sample prior to drive enable:

```
// This is only needed for Emerson/Control Techniques
delay 2000 ms;  // needed in case restart so syncs when cycle DC Sync on/off.
// Cycle time is 1 ms, start it 100 ms in the future.
// Note that we need to make sure that the first slave device in the EtherCAT
// Network supports 32/64-bit distributed clocks for this to work properly.
// Thus far that is Beckhoff, Wago, and Sanyo Denki.
dcsync -1, 1000000, 0, 0, 100000000;
delay 200 ms;     // starts 100 milliseconds into the future

/**************** ENABLE DRIVE *******************/
[Drive_Enable]
// Issue any hardware enable output commands
//
// Power up the drive amplifier
drive enable;
```

Drive Enable Command

Emerson drives appear to have an anomaly which in some cases can cause the motor to creep or move when it is not commanded. This may have something to do with the Pr3.22 (Hard speed reference) setting but has not been verified. During EtherCAT initialization the drive is placed in Cyclic Sync Position mode and it target position is set equal to its present position, which should result in no motion. If an external hardware enable is used and this enable is turned on prior to the MSB executing unpredictable results can occur since the M3-41 is not in full control. From an EtherCAT perspective the drive is disabled via the control word but from a drive perspective it may move based upon internal parameter settings, overriding the EtherCAT command.

The ideal way to overcome this problem is to have the drive’s MSB issue the command to enable the hardware enable after DC Sync has been executed but just prior to the ‘drive enable’ command. This should be tested in a safe environment with the proper interlocks. A second approach, which has been proven to work, would be to start the axis MSB from QuickBuilder, have the MSB enable DC Sync and just prior to executing the ‘drive enable’ set a flag for the QuickBuilder application to observe. While the MSB hangs in the ‘drive enable’ command waiting for the hardware enable the QuickBuilder application would then activate the drives hardware enable using outputs under its control followed by monitoring the drive's 'enabled' flag to know when the drive amplifier has been fully powered up and is holding position.

Station Alias

The station alias may be set through the main menu screen, selection Pr 15.03. A setting of 0 is for automatic. Note that once an alias is set, if you cycle power, the Emerson drive will display a 0 even though that is not what is set. This was an anomaly in Emerson's firmware at the time testing was done. If an alias is displayed in the QuickBuilder EtherCAT Explorer and a 0 appears in Pr 15.03, you must manually select a different value and then go back to 0 and enter it, cycling power.

Menu to Object Mapping

The format used when mapping drive parameters to PDOs is as follows:

- Index: 0x2000 + menu number
- Sub-index: 0x00 + parameter number
- Size: Dependant on the size (in bytes) of the object to be mapped (range: 1-4)

For example Pr **20.21** would be index 0x2014, sub-index 0x15 and the size would be 4 (it is a 32-bit signed value when referenced in the manual).

RPM Limiting

Should an issue occur where the default maximum RPM of 3000 cannot be overridden by saving the parameters using the Emerson Configuration software, or it needs to be modified on the fly, you may do an SDO write to object 0x6080, unsigned 32 bit integer, with the desired maximum RPM.

Appendix

F

[F] Festo

Festo manufactures a number of drives. The drive currently supported is the single axis EMCA-EC series. This section provides information that may be specific to this manufacturer.

eCAT_driveType – 21

Overview

Only CSP mode is supported with firmware V1.4.0.29 tested. Homing is in place but the current revision of Festo Firmware did not work properly so although the drivers are in place for it, it is not supported until a later revision is tested. The drive does home but has a firmware bug on the transition from homing to CSP mode where it moves on its own for a short period before we obtain control. Presently DC Sync is not supported as well, just free run mode which seems to work fine. The manufacturer specifies a maximum scan rate of 2 mS although 1 mS was tested with no known problems. The interpolation will be 15 mS, regardless, as set by the manufacturer. It is recommended to follow the manufacturer's guidelines and set the scan rate to 2 mS or slower.

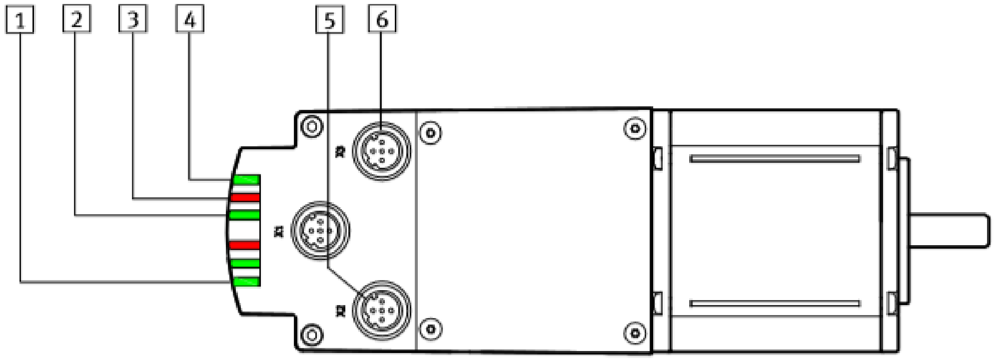
Festo also specifies their units of measure in something called SINC versus encoder counts. This is set to .0006 rpm = 1 SINC/s and not changeable. Thus if we convert this to pulses/second, .0006 rpm/60 sec in minute = .00001 rev = 1 SINC or 100,000 SINC/rev. Thus ppr should be set to 100,000 and is left to the user to adjust as desired.

Overall they are early in their EtherCAT implementation thus later revision firmware should eventually resolve some of the issues.

EtherCAT Cabling

Festo specifies X2 as the EtherCAT input and X3 as the output but it has been observed that this causes the drive to be placed at the end of the slave devices regardless of wiring. This is due to the fact that an internal EtherCAT object is set to -1 (Auto Increment) instead of the standard of 0. Using X3 as the input corrects this and proper slave numbering will result. Thus use X3 (Port 2) as the input and X2 (Port 1) as the output.

The following integrated EtherCAT interfaces of the EMCA-...-EC are available for EtherCAT operation:

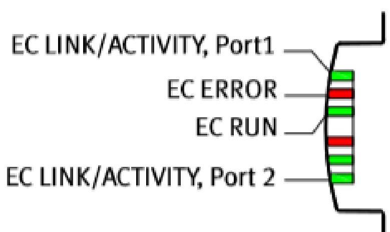


1	LED indicator: EC LINK/ACTIVITY (communication activity/line monitoring) from Port 2, connection [X2]	4	LED indicator: EC LINK/ACTIVITY (communication activity/line monitoring) from Port 1, connection [X3]
2	LED indicator: EC RUN	5	Connection [X2]: EtherCAT, Port 2
3	LED indicator: EC ERROR	6	Connection [X3]: EtherCAT, Port 1

Fig. 4.1 EtherCAT interface on the EMCA

4.2.1 EtherCAT display elements

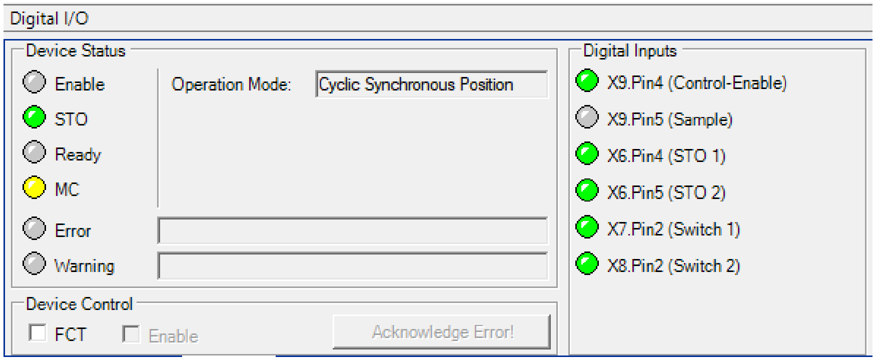
The status of EtherCAT is displayed over the following four LEDs.

LED	Description
	The following EtherCAT statuses are displayed: – EtherCAT communication – Warnings/malfunctions For additional information ➔ Page 252

Tab. 4.2 LED indicator

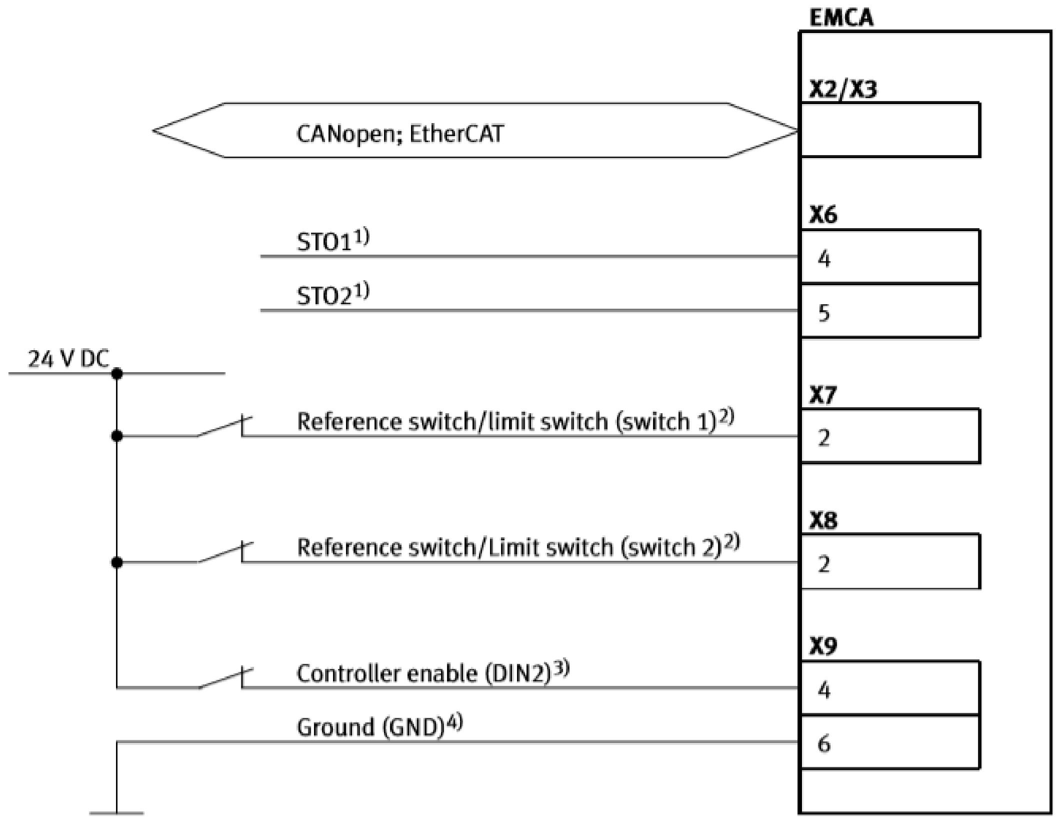
Digital IO

Digital outputs are not currently available under direction of the manufacturer. Digital Inputs can be monitored using the Festo Configuration Tool. The inputs should appear similar to below for operation:



Wiring should be connected as follows with inputs connected to 24V for enable:

The connection diagram shows the required digital inputs "Controller enable", "Safety function" and "Reference switch or limit switch" for bus operation.



1) Further information on the wiring of input channels STO1/STO2 → Description EMCA-EC-S1-...
2) Only required with use reference switch or limit switch → Description EMCA-EC-SY-...
3) Parameterisation of the controller enable signals → FCT
4) Reference potential for the controller

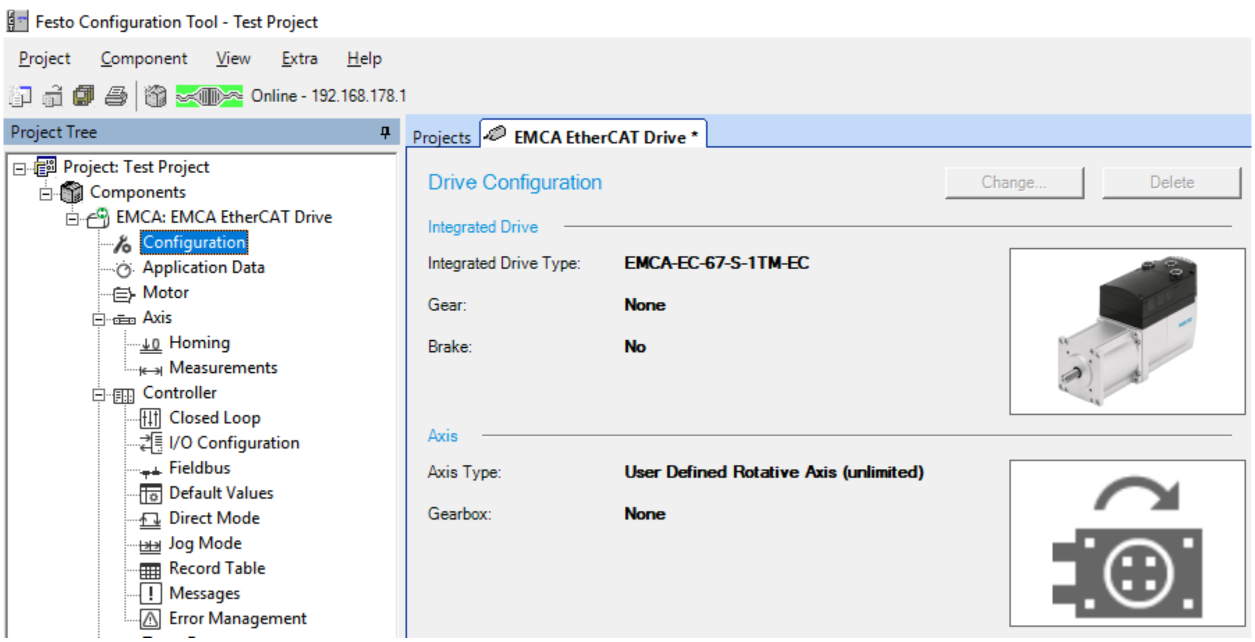
Fig. 1.3 Digital inputs/outputs for operation

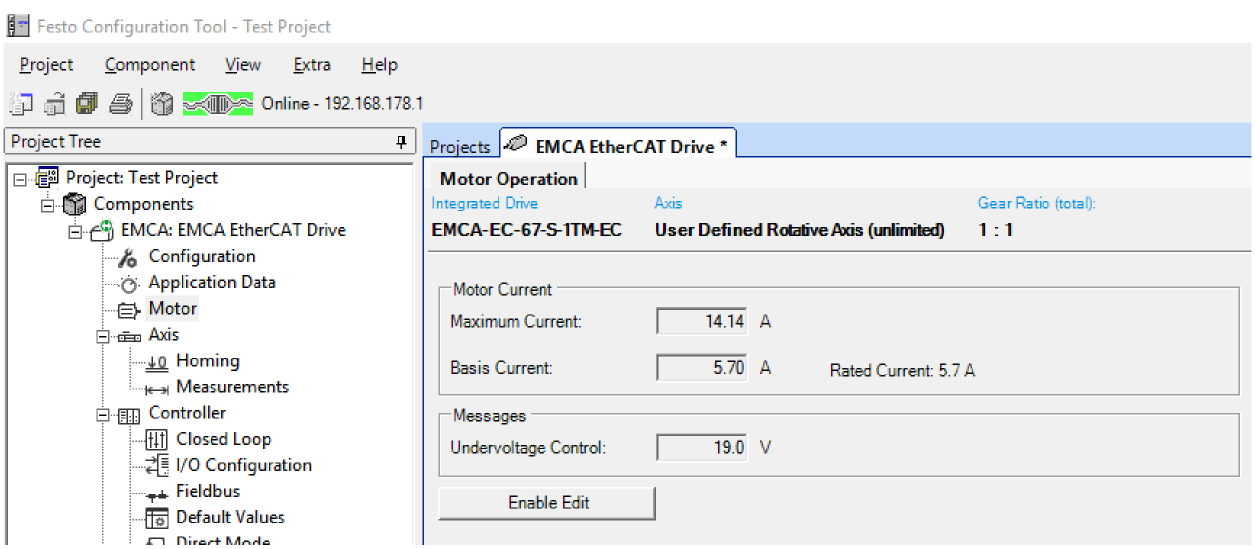
The din bits for an axis map as follows when setup using the Festo Configuration Tool (FCT):

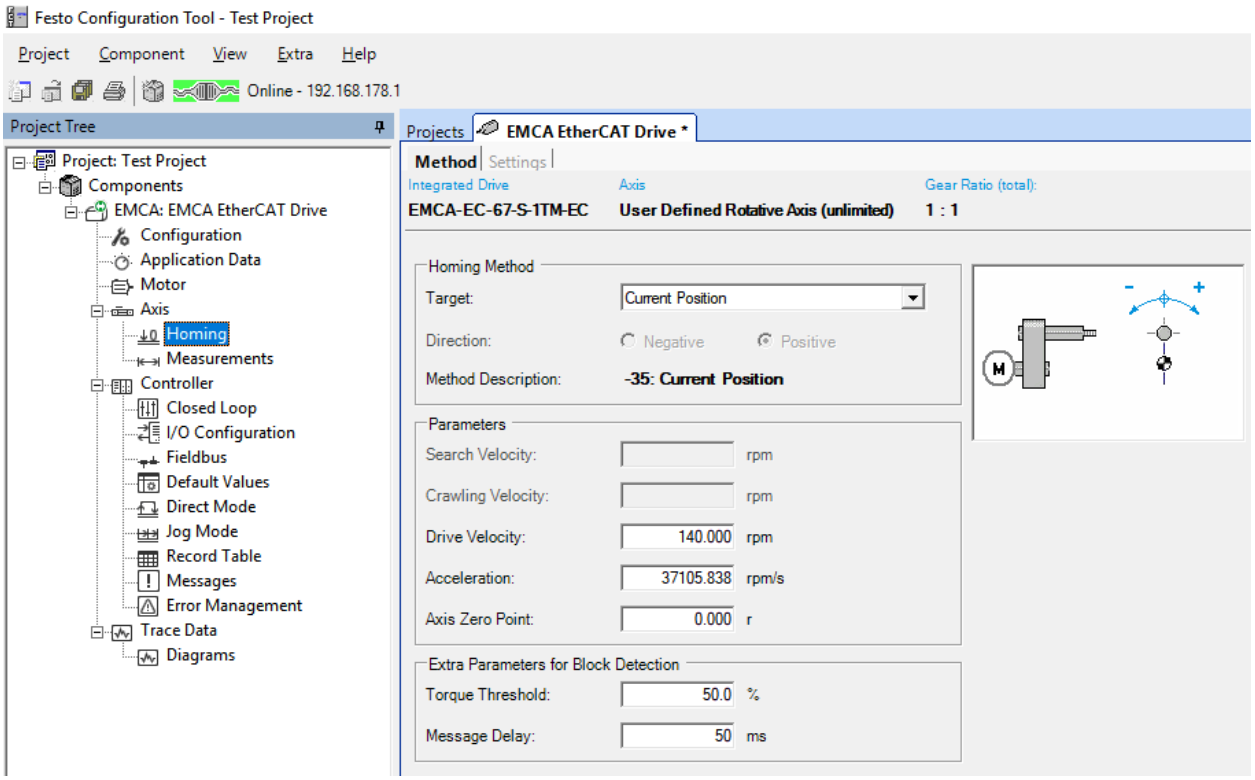
Bit 0: negative limit switch (FCT: Axis<<Axis Options<<Switch Definition)
Bit 1: positive limit switch (setting in FCT: Axis<<Axis Options<<Switch Definition)
Bit 2: home switch (setting in FCT: Axis<<Axis Options<<Switch Definition)

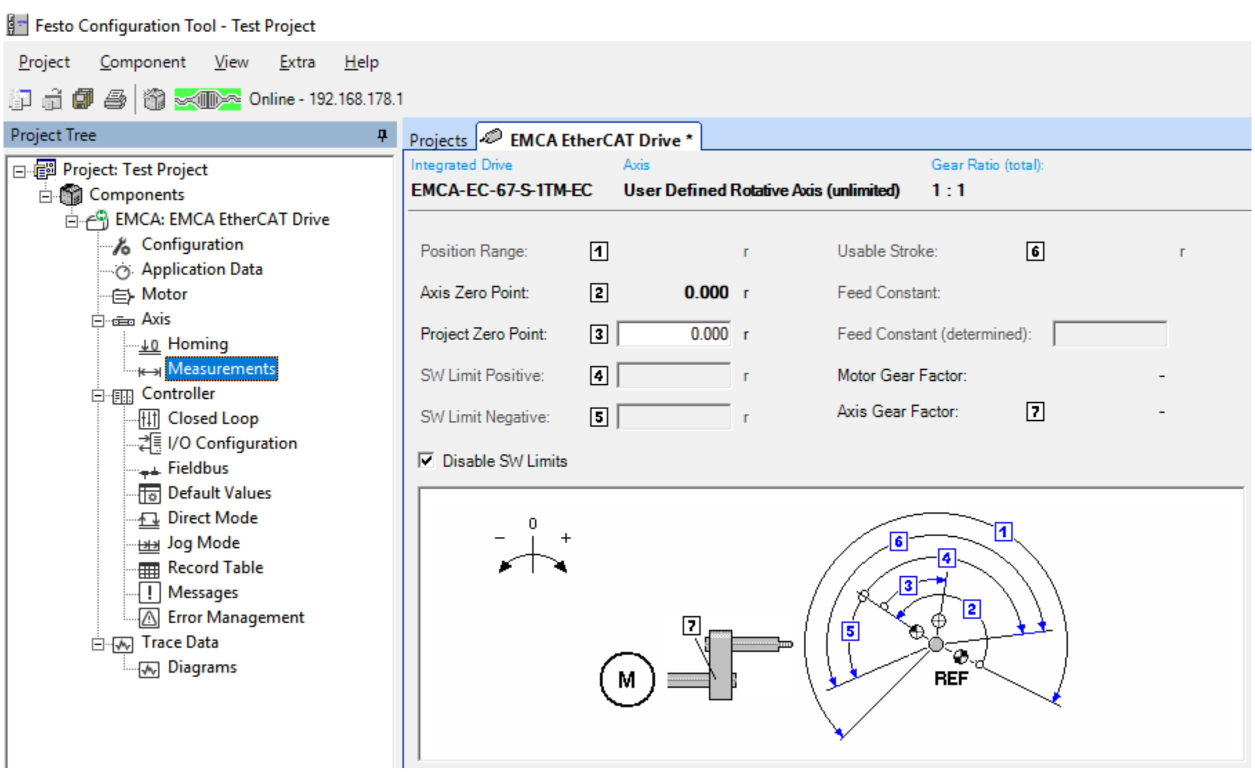
Configuration Tool Screens

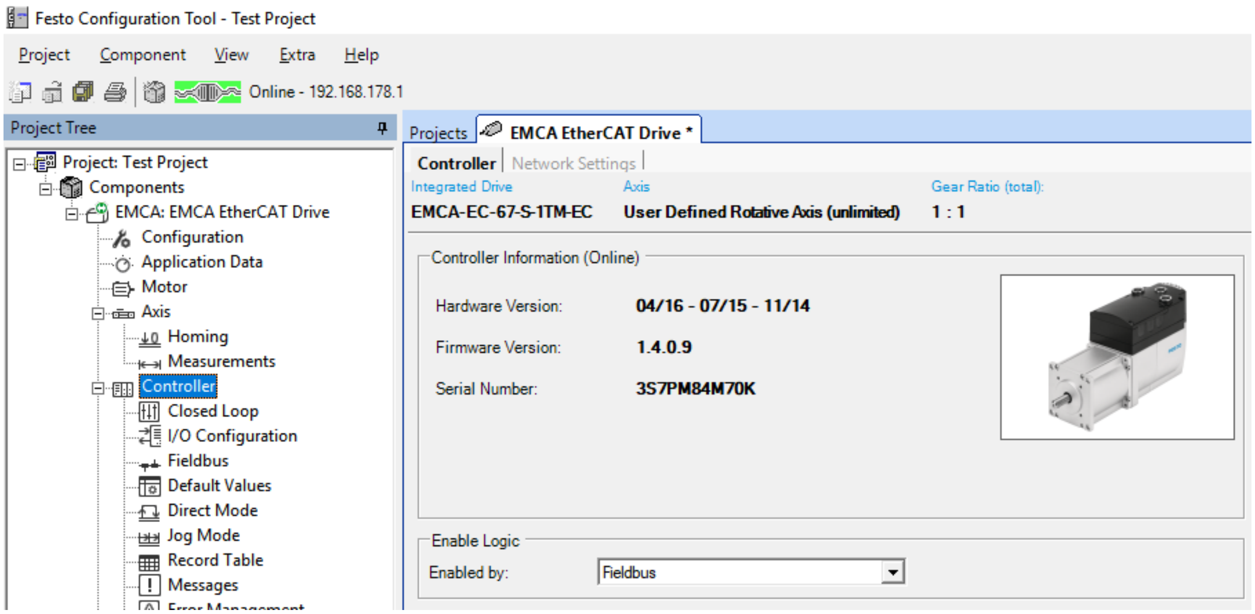
Some of the setup parameters are as follows:

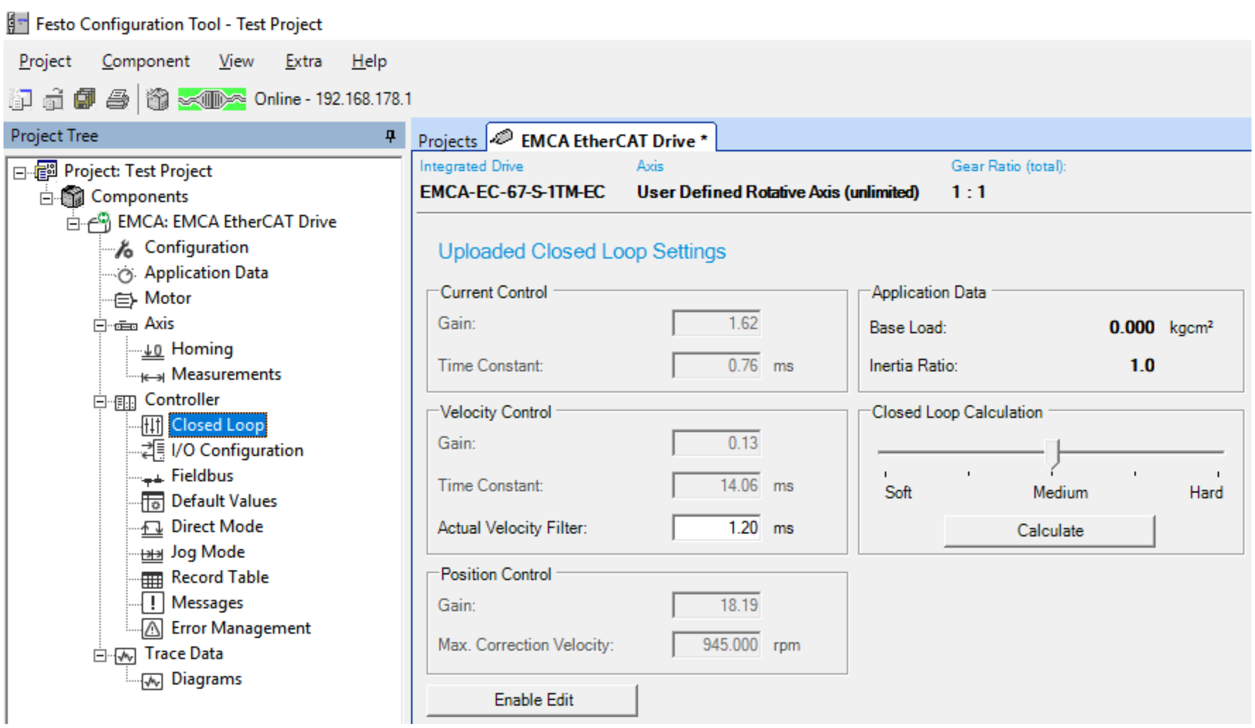


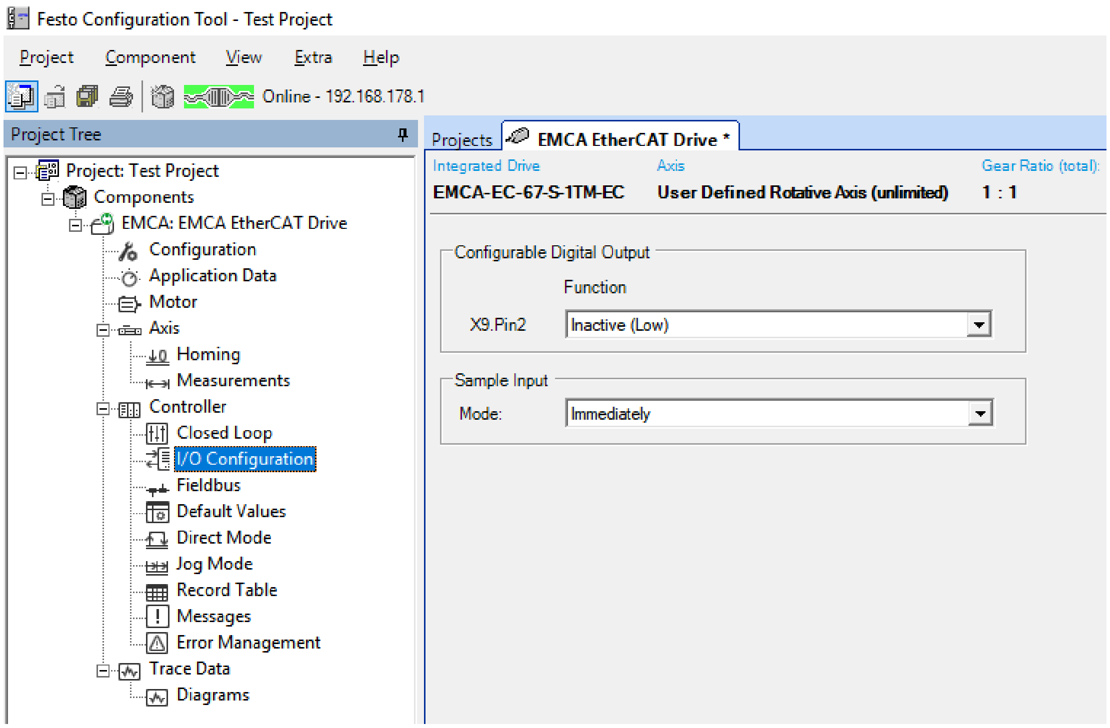


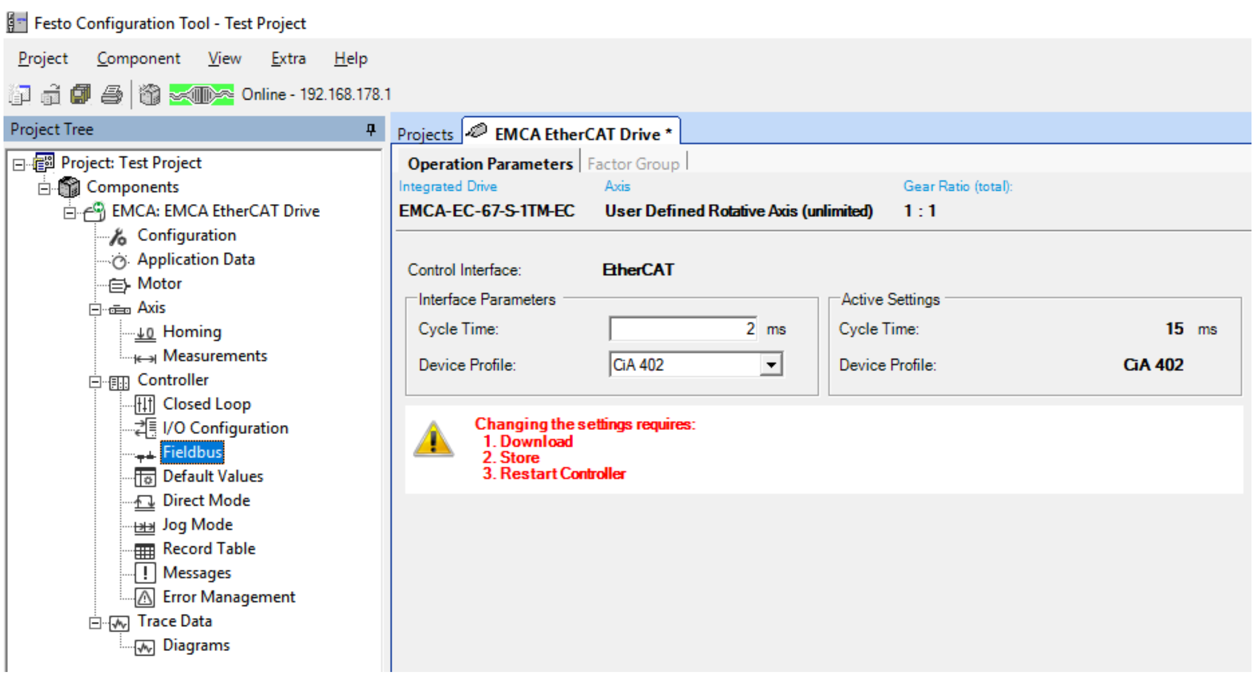

Festo Configuration Tool - Test Project
Project Component View Extra Help
Online - 192.168.178.1
Project Tree
Project: Test Project
Components
EMCA: EMCA EtherCAT Drive
Configuration
Application Data
Motor
Axis
Homing
Measurements
Controller
Closed Loop
I/O Configuration
Fieldbus
Default Values
Direct Mode
Jog Mode
Record Table
Messages
Error Management
Trace Data
Diagrams
Projects
EMCA EtherCAT Drive *
Method
Settings
Integrated Drive
EMCA-EC-67-S-1TM-EC
Axis
User Defined Rotative Axis (unlimited)
Gear Ratio (total):
1 : 1
Homing Method
Target:
Current Position
Direction:
Negative
Positive
Method Description:
-35: Current Position
Parameters
Search Velocity: rpm
Crawling Velocity: rpm
Drive Velocity: 140.000 rpm
Acceleration: 37105.838 rpm/s
Axis Zero Point: 0.000 r
Extra Parameters for Block Detection
Torque Threshold: 50.0 %
Message Delay: 50 ms
M

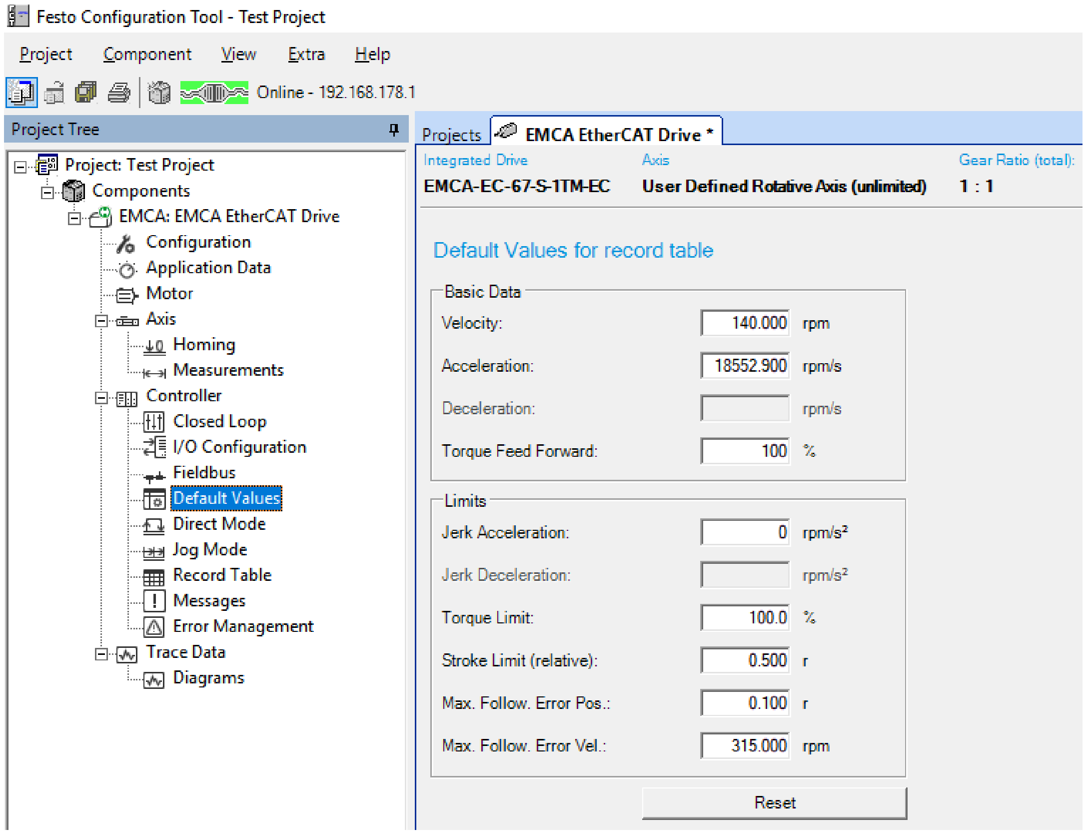

Festo Configuration Tool - Test Project
Project Component View Extra Help
Online - 192.168.178.1
Project Tree
Project: Test Project
Components
EMCA: EMCA EtherCAT Drive
Configuration
Application Data
Motor
Axis
Homing
Measurements
Controller
Closed Loop
I/O Configuration
Fieldbus
Default Values
Direct Mode
Jog Mode
Record Table
Messages
Error Management
Trace Data
Diagrams
Projects
EMCA EtherCAT Drive *
Integrated Drive
EMCA-EC-67-S-1TM-EC
Axis
User Defined Rotative Axis (unlimited)
Gear Ratio (total):
1 : 1
Position Range: 1 r
Axis Zero Point: 2 0.000 r
Project Zero Point: 3 0.000 r
SW Limit Positive: 4 r
SW Limit Negative: 5 r
Usable Stroke: 6 r
Feed Constant:
Feed Constant (determined):
Motor Gear Factor: -
Axis Gear Factor: 7 -
Disable SW Limits
0
M
REF

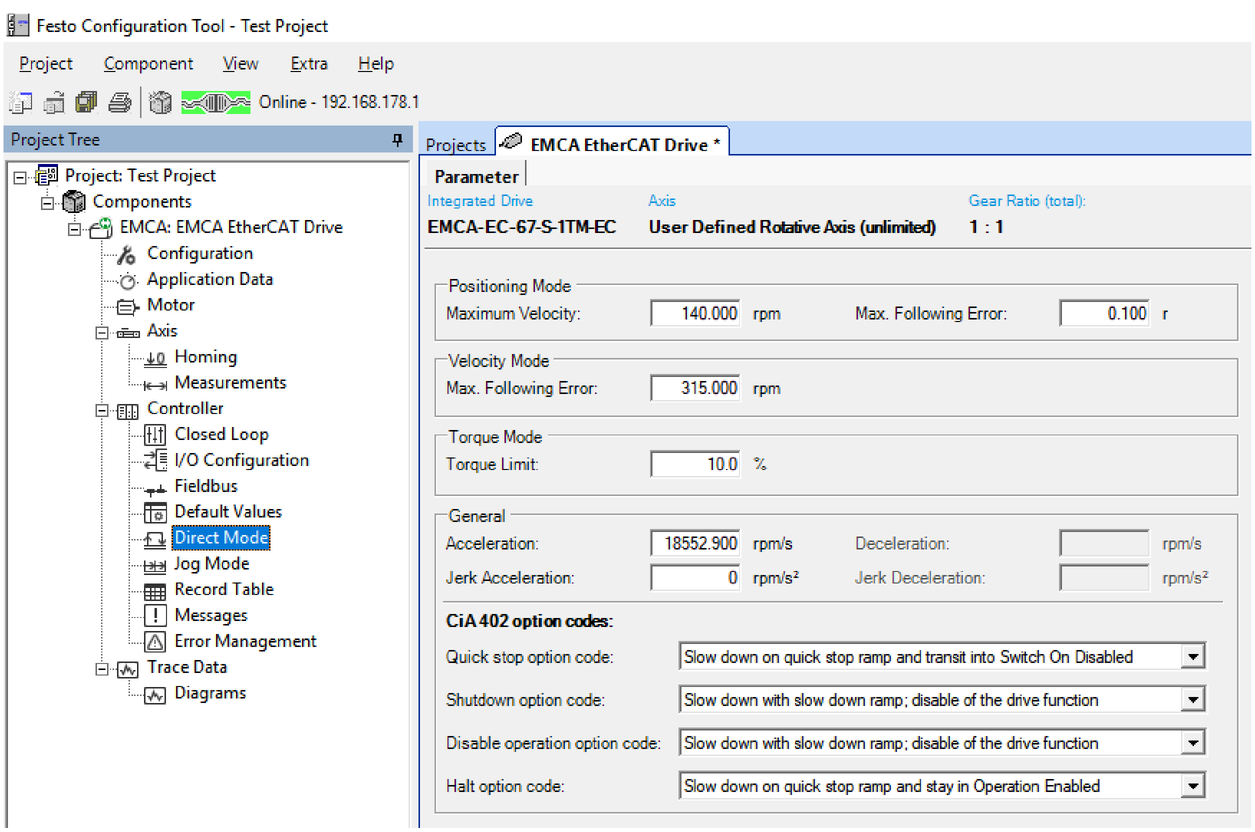

Festo Configuration Tool - Test Project
Project Component View Extra Help
Online - 192.168.178.1
Project Tree
Project: Test Project
Components
EMCA: EMCA EtherCAT Drive
Configuration
Application Data
Motor
Axis
Homing
Measurements
Controller
Closed Loop
I/O Configuration
Fieldbus
Default Values
Direct Mode
Jog Mode
Record Table
Messages
Projects
EMCA EtherCAT Drive *
Controller
Network Settings
Integrated Drive
Axis
Gear Ratio (total):
EMCA-EC-67-S-1TM-EC
User Defined Rotative Axis (unlimited)
1 : 1
Controller Information (Online)
Hardware Version:
04/16 - 07/15 - 11/14
Firmware Version:
1.4.0.9
Serial Number:
3S7PM84M70K
Enable Logic
Enabled by:
Fieldbus

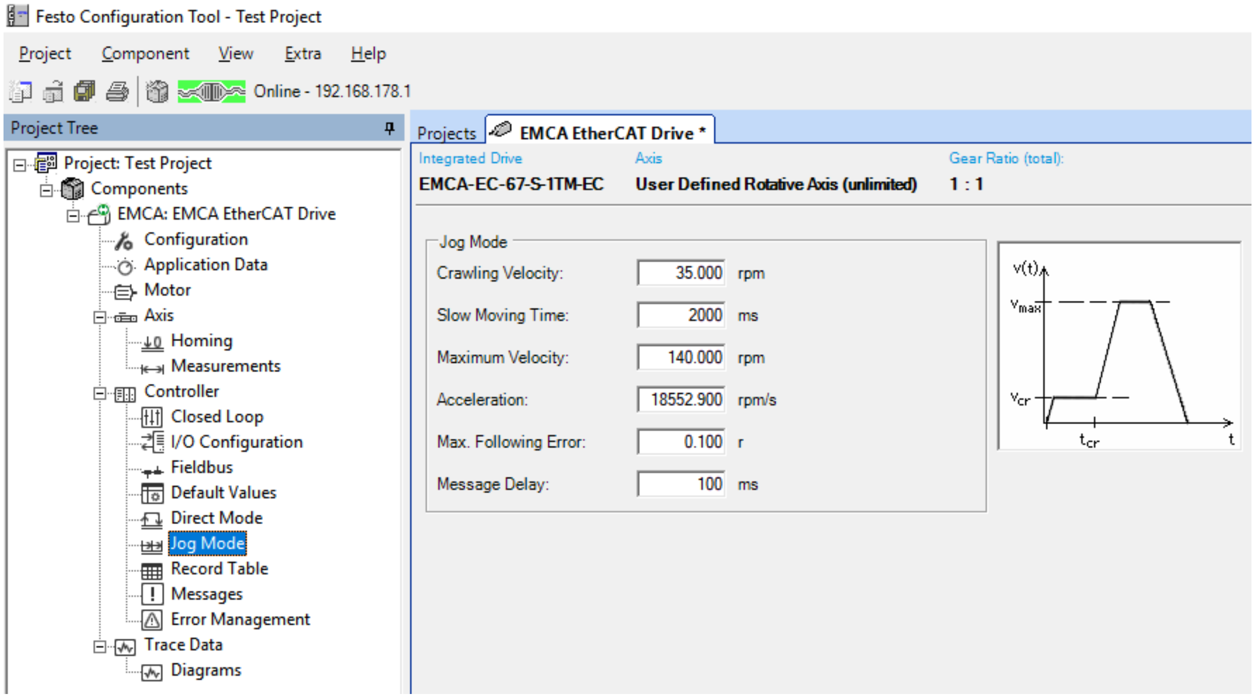

Festo Configuration Tool - Test Project
Project Component View Extra Help
Online - 192.168.178.1
Project Tree
Project: Test Project
Components
EMCA: EMCA EtherCAT Drive
Configuration
Application Data
Motor
Axis
Homing
Measurements
Controller
Closed Loop
I/O Configuration
Fieldbus
Default Values
Direct Mode
Jog Mode
Record Table
Messages
Error Management
Trace Data
Diagrams
Projects
EMCA EtherCAT Drive *
Integrated Drive
Axis
Gear Ratio (total):
EMCA-EC-67-S-1TM-EC
User Defined Rotative Axis (unlimited)
1 : 1
Uploaded Closed Loop Settings
Current Control
Gain: 1.62
Time Constant: 0.76 ms
Application Data
Base Load: 0.000 kgcm²
Inertia Ratio: 1.0
Velocity Control
Gain: 0.13
Time Constant: 14.06 ms
Actual Velocity Filter: 1.20 ms
Closed Loop Calculation
Soft
Medium
Hard
Calculate
Position Control
Gain: 18.19
Max. Correction Velocity: 945.000 rpm
Enable Edit

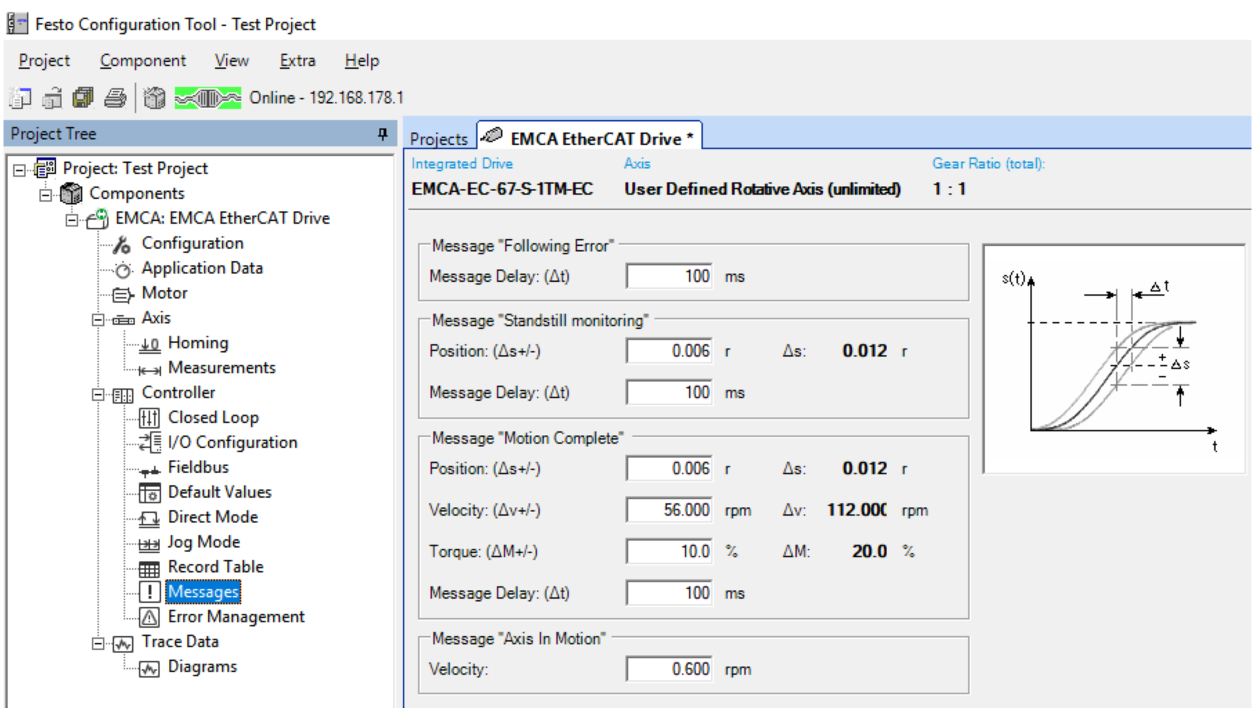

Festo Configuration Tool - Test Project
Project
Component
View
Extra
Help
Online - 192.168.178.1
Project Tree
Project: Test Project
Components
EMCA: EMCA EtherCAT Drive
Configuration
Application Data
Motor
Axis
Homing
Measurements
Controller
Closed Loop
I/O Configuration
Fieldbus
Default Values
Direct Mode
Jog Mode
Record Table
Messages
Error Management
Trace Data
Diagrams
Projects
EMCA EtherCAT Drive *
Integrated Drive
Axis
Gear Ratio (total):
EMCA-EC-67-S-1TM-EC
User Defined Rotative Axis (unlimited)
1 : 1
Configurable Digital Output
Function
X9.Pin2
Inactive (Low)
Sample Input
Mode:
Immediately

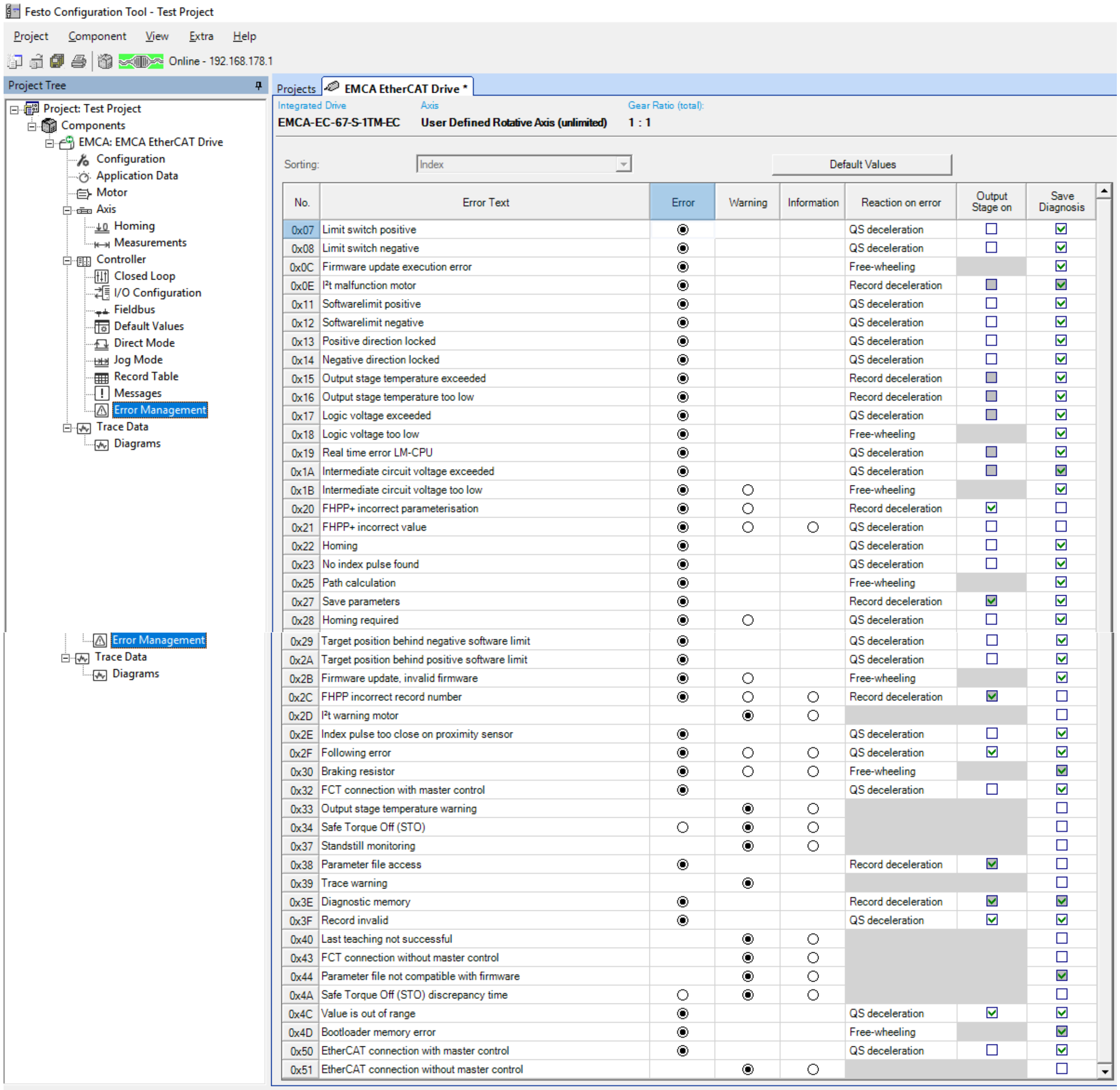

Festo Configuration Tool - Test Project
Project
Component
View
Extra
Help
Online - 192.168.178.1
Project Tree
Project: Test Project
Components
EMCA: EMCA EtherCAT Drive
Configuration
Application Data
Motor
Axis
Homing
Measurements
Controller
Closed Loop
I/O Configuration
Fieldbus
Default Values
Direct Mode
Jog Mode
Record Table
Messages
Error Management
Trace Data
Diagrams
Projects
EMCA EtherCAT Drive *
Operation Parameters
Factor Group
Integrated Drive
Axis
Gear Ratio (total):
EMCA-EC-67-S-1TM-EC
User Defined Rotative Axis (unlimited)
1 : 1
Control Interface:
EtherCAT
Interface Parameters
Cycle Time:
2
ms
Device Profile:
CiA 402
Active Settings
Cycle Time:
15
ms
Device Profile:
CiA 402
Changing the settings requires:
1. Download
2. Store
3. Restart Controller


Festo Configuration Tool - Test Project
Project Component View Extra Help
Online - 192.168.178.1
Project Tree
Project: Test Project
Components
EMCA: EMCA EtherCAT Drive
Configuration
Application Data
Motor
Axis
Homing
Measurements
Controller
Closed Loop
I/O Configuration
Fieldbus
Default Values
Direct Mode
Jog Mode
Record Table
Messages
Error Management
Trace Data
Diagrams
Projects
EMCA EtherCAT Drive *
Integrated Drive
EMCA-EC-67-S-1TM-EC
Axis
User Defined Rotative Axis (unlimited)
Gear Ratio (total):
1 : 1
Default Values for record table
Basic Data
Velocity: 140.000 rpm
Acceleration: 18552.900 rpm/s
Deceleration: rpm/s
Torque Feed Forward: 100 %
Limits
Jerk Acceleration: 0 rpm/s²
Jerk Deceleration: rpm/s²
Torque Limit: 100.0 %
Stroke Limit (relative): 0.500 r
Max. Follow. Error Pos.: 0.100 r
Max. Follow. Error Vel.: 315.000 rpm
Reset


Festo Configuration Tool - Test Project
Project Component View Extra Help
Online - 192.168.178.1
Project Tree
Project: Test Project
Components
EMCA: EMCA EtherCAT Drive
Configuration
Application Data
Motor
Axis
Homing
Measurements
Controller
Closed Loop
I/O Configuration
Fieldbus
Default Values
Direct Mode
Jog Mode
Record Table
Messages
Error Management
Trace Data
Diagrams
Projects
EMCA EtherCAT Drive *
Parameter
Integrated Drive
EMCA-EC-67-S-1TM-EC
Axis
User Defined Rotative Axis (unlimited)
Gear Ratio (total):
1 : 1
Positioning Mode
Maximum Velocity: 140.000 rpm
Max. Following Error: 0.100 r
Velocity Mode
Max. Following Error: 315.000 rpm
Torque Mode
Torque Limit: 10.0 %
General
Acceleration: 18552.900 rpm/s
Deceleration: rpm/s
Jerk Acceleration: 0 rpm/s²
Jerk Deceleration: rpm/s²
CiA 402 option codes:
Quick stop option code: Slow down on quick stop ramp and transit into Switch On Disabled
Shutdown option code: Slow down with slow down ramp; disable of the drive function
Disable operation option code: Slow down with slow down ramp; disable of the drive function
Halt option code: Slow down on quick stop ramp and stay in Operation Enabled


Festo Configuration Tool - Test Project
Project Component View Extra Help
Online - 192.168.178.1
Project Tree
Project: Test Project
Components
EMCA: EMCA EtherCAT Drive
Configuration
Application Data
Motor
Axis
Homing
Measurements
Controller
Closed Loop
I/O Configuration
Fieldbus
Default Values
Direct Mode
Jog Mode
Record Table
Messages
Error Management
Trace Data
Diagrams
Projects
EMCA EtherCAT Drive *
Integrated Drive
EMCA-EC-67-S-1TM-EC
Axis
User Defined Rotative Axis (unlimited)
Gear Ratio (total):
1 : 1
Jog Mode
Crawling Velocity: 35.000 rpm
Slow Moving Time: 2000 ms
Maximum Velocity: 140.000 rpm
Acceleration: 18552.900 rpm/s
Max. Following Error: 0.100 r
Message Delay: 100 ms
v(t)
vmax
vcr
tcr
t


Festo Configuration Tool - Test Project
Project Component View Extra Help
Online - 192.168.178.1
Project Tree
Project: Test Project
Components
EMCA: EMCA EtherCAT Drive
Configuration
Application Data
Motor
Axis
Homing
Measurements
Controller
Closed Loop
I/O Configuration
Fieldbus
Default Values
Direct Mode
Jog Mode
Record Table
Messages
Error Management
Trace Data
Diagrams
Projects
EMCA EtherCAT Drive *
Integrated Drive
EMCA-EC-67-S-1TM-EC
Axis
User Defined Rotative Axis (unlimited)
Gear Ratio (total):
1 : 1
Message "Following Error"
Message Delay: (Δt) 100 ms
Message "Standstill monitoring"
Position: (Δs+/-) 0.006 r Δs: 0.012 r
Message Delay: (Δt) 100 ms
Message "Motion Complete"
Position: (Δs+/-) 0.006 r Δs: 0.012 r
Velocity: (Δv+/-) 56.000 rpm Δv: 112.000 rpm
Torque: (ΔM+/-) 10.0 % ΔM: 20.0 %
Message Delay: (Δt) 100 ms
Message "Axis In Motion"
Velocity: 0.600 rpm
s(t)
Δt
Δs
t



No.	Error Text	Error	Warning	Information	Reaction on error	Output Stage on	Save Diagnosis
0x07	Limit switch positive	◉			QS deceleration	☐	☑
0x08	Limit switch negative	◉			QS deceleration	☐	☑
0x0C	Firmware update execution error	◉			Free-wheeling		☑
0x0E	I²t malfunction motor	◉			Record deceleration	☐	☑
0x11	Softwarelimit positive	◉			QS deceleration	☐	☑
0x12	Softwarelimit negative	◉			QS deceleration	☐	☑
0x13	Positive direction locked	◉			QS deceleration	☐	☑
0x14	Negative direction locked	◉			QS deceleration	☐	☑
0x15	Output stage temperature exceeded	◉			Record deceleration	☐	☑
0x16	Output stage temperature too low	◉			Record deceleration	☐	☑
0x17	Logic voltage exceeded	◉			QS deceleration	☐	☑
0x18	Logic voltage too low	◉			Free-wheeling		☑
0x19	Real time error LM-CPU	◉			QS deceleration	☐	☑
0x1A	Intermediate circuit voltage exceeded	◉			QS deceleration	☐	☑
0x1B	Intermediate circuit voltage too low	◉	○		Free-wheeling		☑
0x20	FHPP+ incorrect parameterisation	◉	○		Record deceleration	☑	☐
0x21	FHPP+ incorrect value	◉	○	○	QS deceleration	☐	☐
0x22	Homing	◉			QS deceleration	☐	☑
0x23	No index pulse found	◉			QS deceleration	☐	☑
0x25	Path calculation	◉			Free-wheeling		☑
0x27	Save parameters	◉			Record deceleration	☑	☑
0x28	Homing required	◉	○		QS deceleration	☐	☑
0x29	Target position behind negative software limit	◉			QS deceleration	☐	☑
0x2A	Target position behind positive software limit	◉			QS deceleration	☐	☑
0x2B	Firmware update, invalid firmware	◉	○		Free-wheeling		☑
0x2C	FHPP incorrect record number	◉	○	○	Record deceleration	☑	☐
0x2D	I²t warning motor		◉	○			☐
0x2E	Index pulse too close on proximity sensor	◉			QS deceleration	☐	☑
0x2F	Following error	◉	○	○	QS deceleration	☑	☑
0x30	Braking resistor	◉	○	○	Free-wheeling		☑
0x32	FCT connection with master control	◉			QS deceleration	☐	☑
0x33	Output stage temperature warning		◉	○			☐
0x34	Safe Torque Off (STO)	○	◉	○			☐
0x37	Standstill monitoring		◉	○			☐
0x38	Parameter file access	◉			Record deceleration	☑	☐
0x39	Trace warning		◉				☐
0x3E	Diagnostic memory	◉			Record deceleration	☑	☑
0x3F	Record invalid	◉			QS deceleration	☑	☑
0x40	Last teaching not successful		◉	○			☐
0x43	FCT connection without master control		◉	○			☐
0x44	Parameter file not compatible with firmware		◉	○			☑
0x4A	Safe Torque Off (STO) discrepancy time	○	◉	○			☐
0x4C	Value is out of range	◉			QS deceleration	☑	☑
0x4D	Bootloader memory error	◉			Free-wheeling		☑
0x50	EtherCAT connection with master control	◉			QS deceleration	☐	☑
0x51	EtherCAT connection without master control		◉	○			☐

Appendix

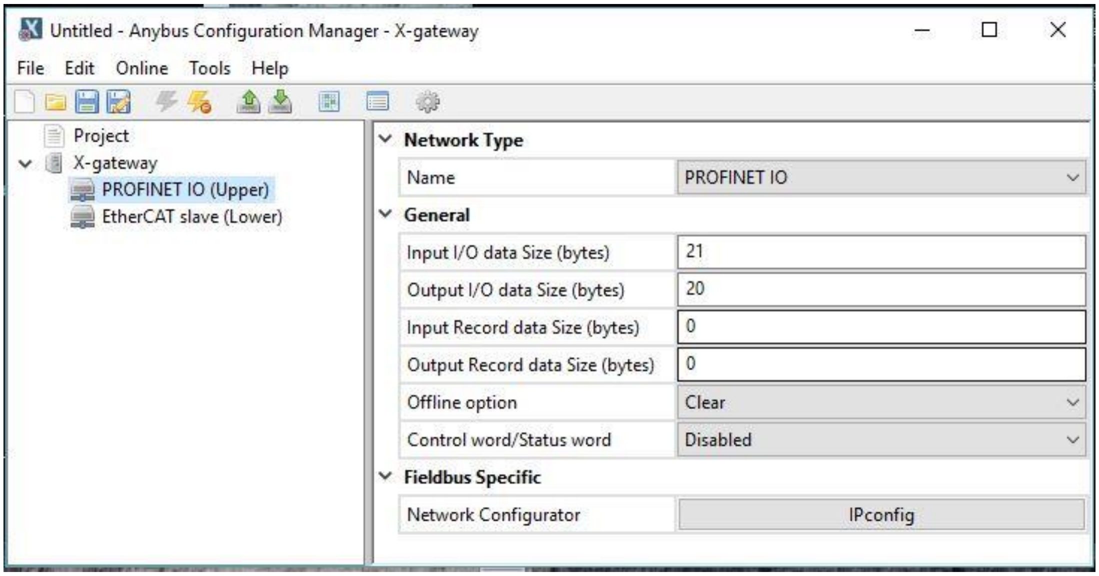
G

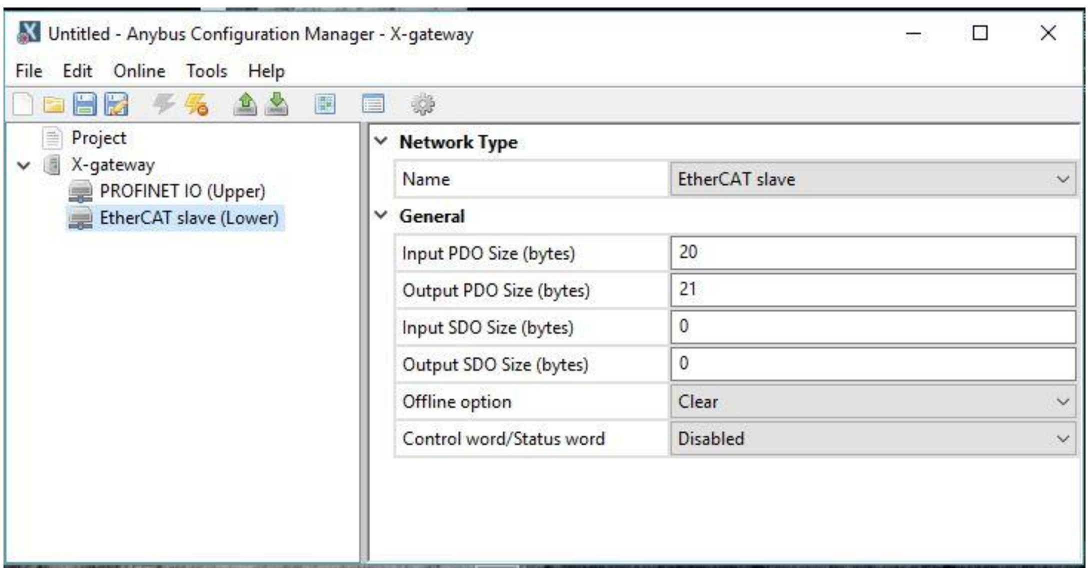
[G] HMS Anybus X Gateway

HMS is a manufacturer of a number of protocol converter products. One in particular is the Anybus X Gateway which connects two dissimilar protocols, one of which is EtherCAT Slave, the other a User's choice. PROFINET IO Slave is documented below but others should be similar.

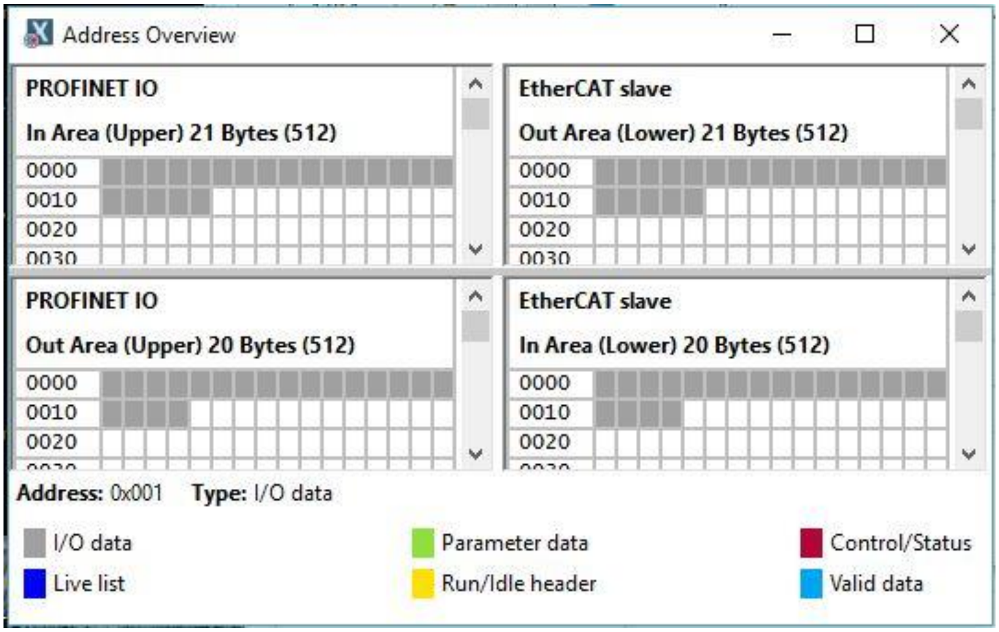
Protocol Mapping

The Anybus Configuration Manager – X gateway tool must be downloaded and installed on your PC. This tool connects to the X4 port of the Gateway via a USB cable. For the PROFINET/EtherCAT Slave unit the 'Lower' port refers to the two RJ45 jacks on the bottom of the unit and is to be used for the EtherCAT Slave protocol. In this example the upper port is mapped to the PROFINET IO, similar for Modbus/TCP. You basically set the input and output size for each mapped area and it will appear as an 8 bit IO device to Incentive. Using registers and input/output instructions this can also be read on 16 and 32 bit boundaries.





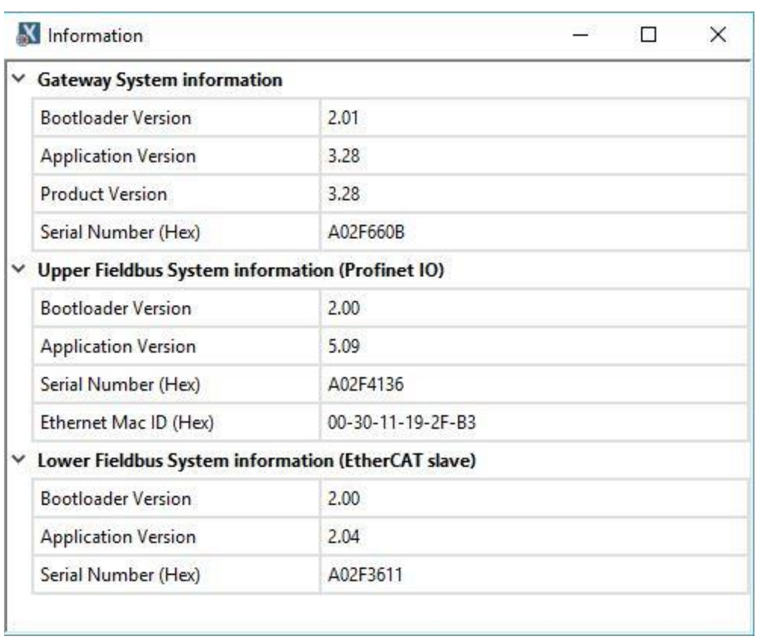
It is important to note that the Input I/O data of the User protocol, PROFINET in this case, maps to the Output PDO Size of the EtherCAT slave. EtherCAT Slave devices specify the output relative to them and since they are sending it to the Incentive host, Incentive inputs are data that is 'Output' by the EtherCAT slave. Incentive outputs become EtherCAT Slave Input PDO Size and PROFINET Output I/O data Size.



After mapping is complete be sure to save the configuration back into the Gateway unit, reboot it, and restart Incentive. Restarting Incentive is necessarily since it will dynamically adjust the size of its EtherCAT IO based on the mapping defined. This particular example mapped as 160 (20 X 8) data outputs and 168 (21 X 8) data inputs which are what will appear in the IO map of Incentive. Note that there is a maximum of 1024 inputs and 1024 outputs within Incentive.

Gateway Software Revisions

Below are the revisions as tested, Part number AB7684-F:

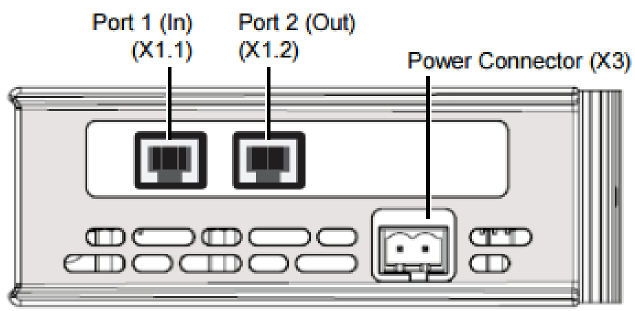

Information

Gateway System information	
Bootloader Version	2.01
Application Version	3.28
Product Version	3.28
Serial Number (Hex)	A02F660B
Upper Fieldbus System information (Profinet IO)	
Bootloader Version	2.00
Application Version	5.09
Serial Number (Hex)	A02F4136
Ethernet Mac ID (Hex)	00-30-11-19-2F-B3
Lower Fieldbus System information (EtherCAT slave)	
Bootloader Version	2.00
Application Version	2.04
Serial Number (Hex)	A02F3611

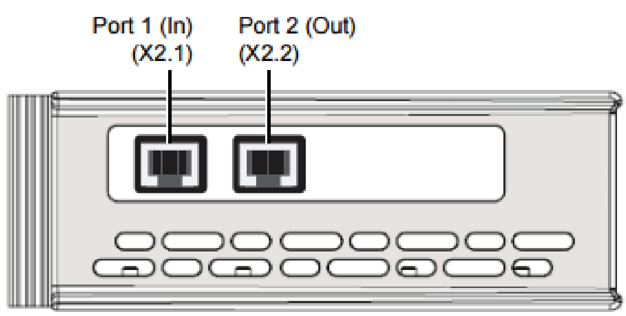
EtherCAT Slave Connections

Depending upon the unit the EtherCAT ports will be on the top or bottom. Please note that X1.1/X2.1 is the EtherCAT Slave input and X1.2/X2.2 is the output. Failure to connect correctly will cause the network to fail.

Top-mounted Interface



Bottom-mounted Interface



Appendix H

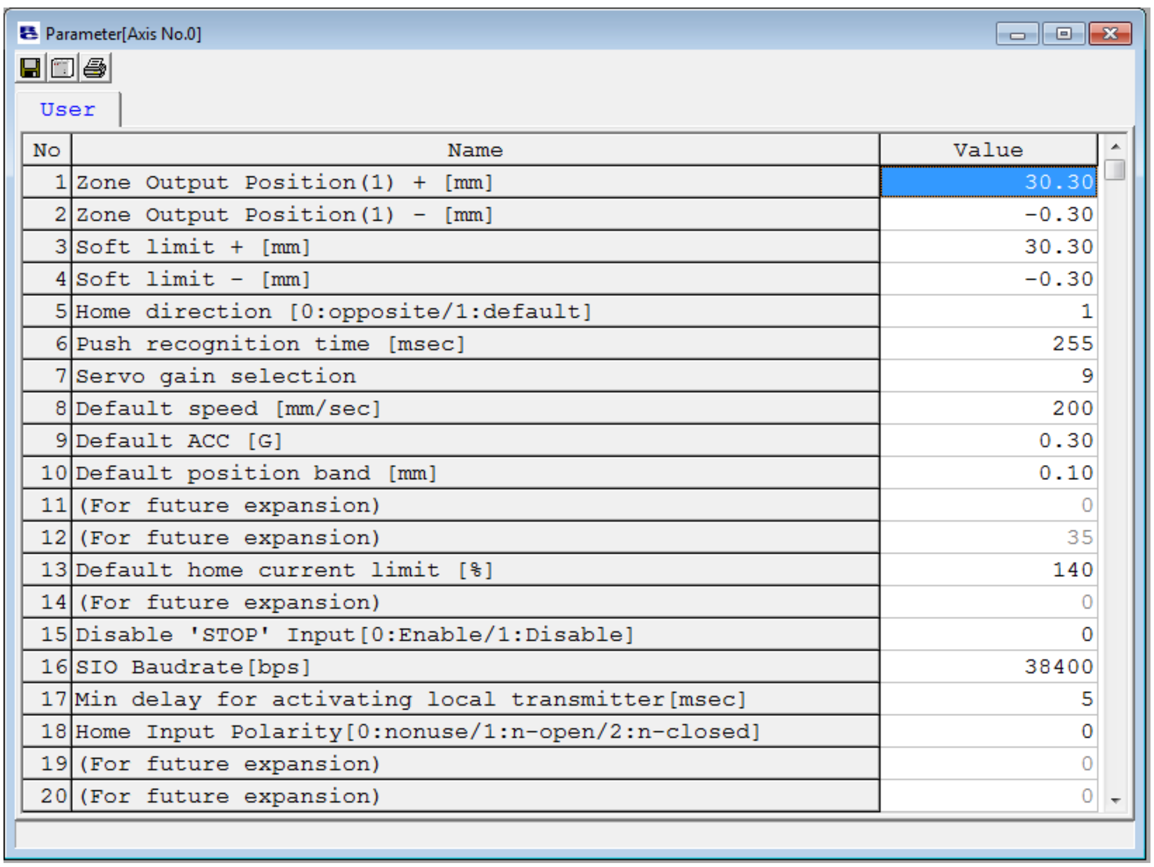
[H] IAI

IAI is a manufacturer of intelligent linear actuators. The M3-41 supports the ACON controller in Full Direct Value Mode. The IAI Windows based ROBO Cylinder Software was used for initial setup of the controller. This section provides information that may be specific to this manufacturer.

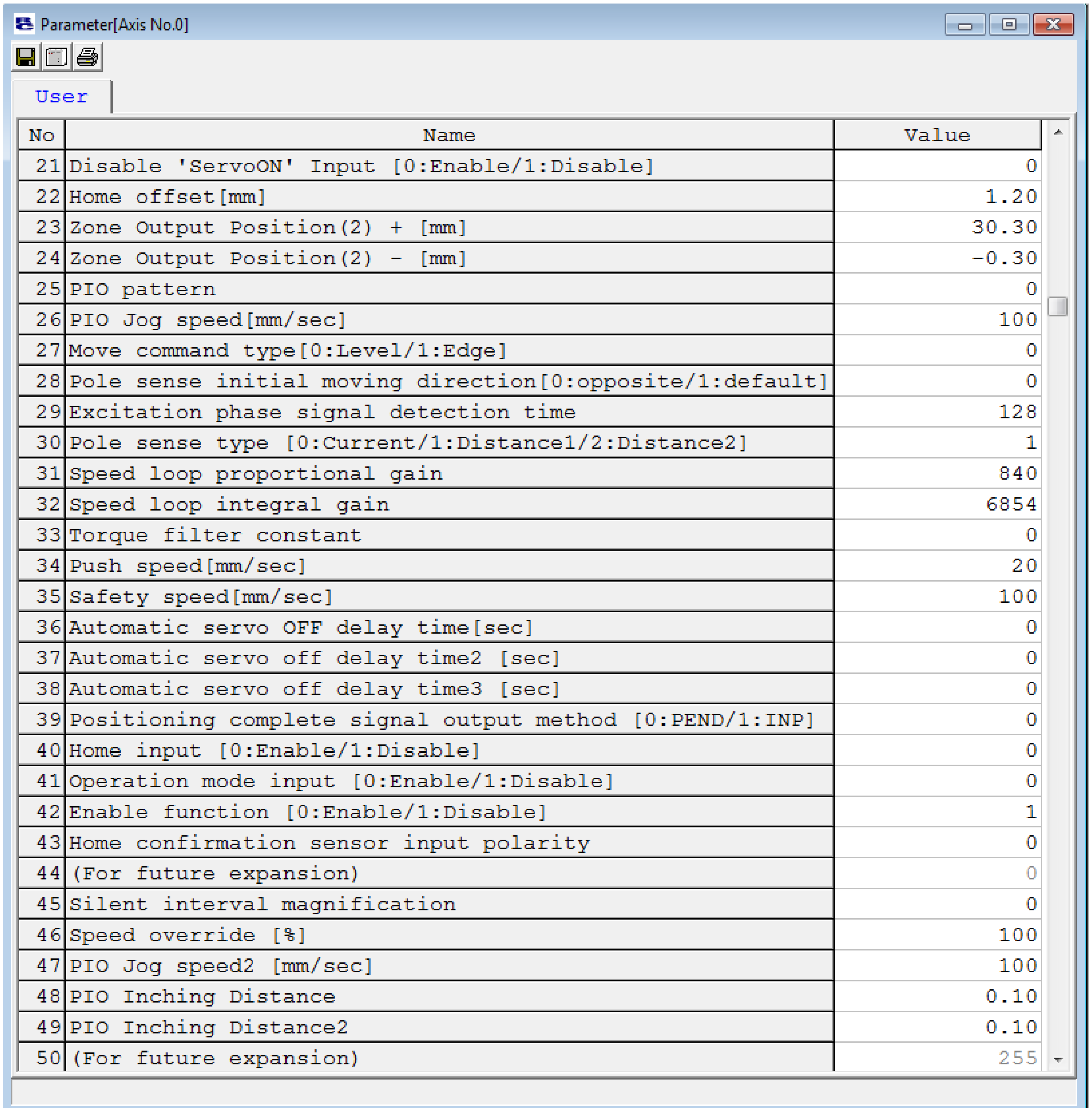
eCAT_driveType – 11

Parameter Information from Test Setup

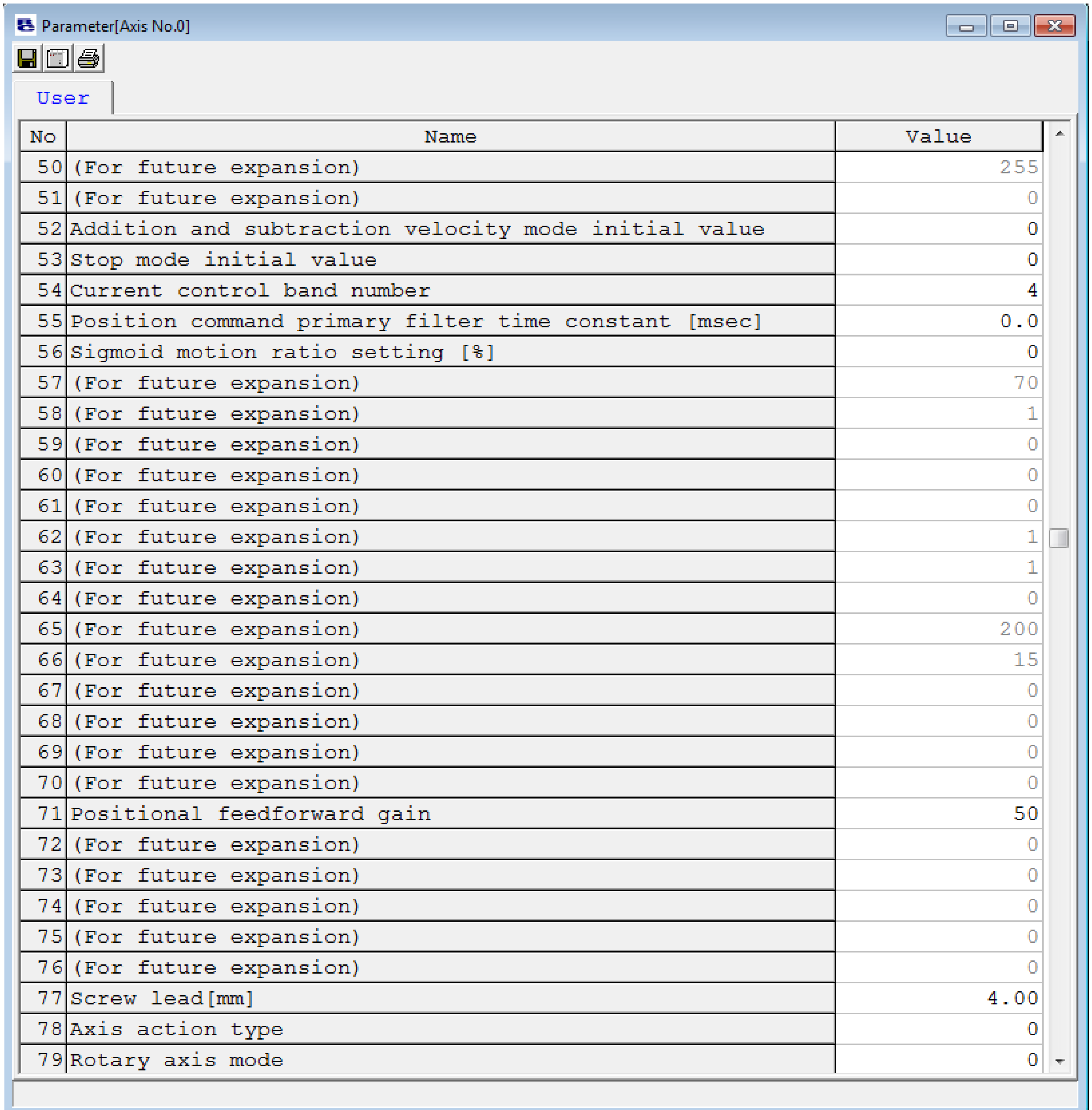
Test setup: Controller – ACON-C-30I-EC-0-0, Actuator - RCA2-TF4N-I-20-4-30-A3-P



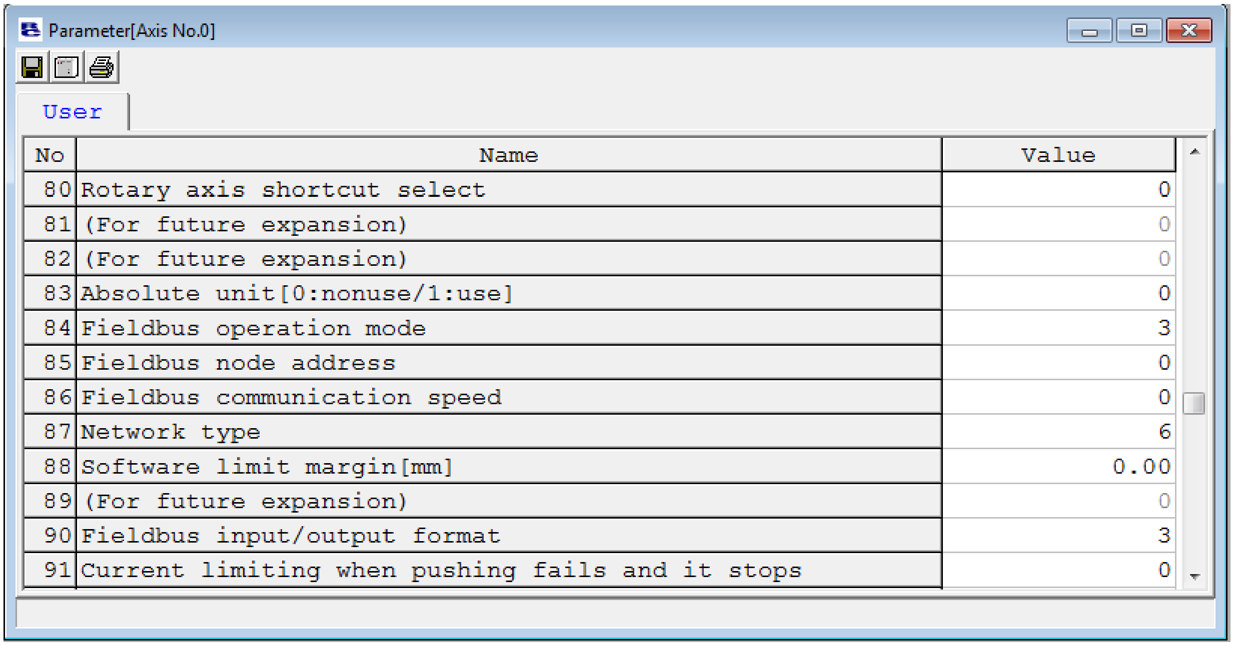
No	Name	Value
1	Zone Output Position(1) + [mm]	30.30
2	Zone Output Position(1) - [mm]	-0.30
3	Soft limit + [mm]	30.30
4	Soft limit - [mm]	-0.30
5	Home direction [0:opposite/1:default]	1
6	Push recognition time [msec]	255
7	Servo gain selection	9
8	Default speed [mm/sec]	200
9	Default ACC [G]	0.30
10	Default position band [mm]	0.10
11	(For future expansion)	0
12	(For future expansion)	35
13	Default home current limit [%]	140
14	(For future expansion)	0
15	Disable 'STOP' Input[0:Enable/1:Disable]	0
16	SIO Baudrate[bps]	38400
17	Min delay for activating local transmitter[msec]	5
18	Home Input Polarity[0:nonuse/1:n-open/2:n-closed]	0
19	(For future expansion)	0
20	(For future expansion)	0



No	Name	Value
21	Disable 'ServoON' Input [0:Enable/1:Disable]	0
22	Home offset[mm]	1.20
23	Zone Output Position(2) + [mm]	30.30
24	Zone Output Position(2) - [mm]	-0.30
25	PIO pattern	0
26	PIO Jog speed[mm/sec]	100
27	Move command type[0:Level/1:Edge]	0
28	Pole sense initial moving direction[0:opposite/1:default]	0
29	Excitation phase signal detection time	128
30	Pole sense type [0:Current/1:Distance1/2:Distance2]	1
31	Speed loop proportional gain	840
32	Speed loop integral gain	6854
33	Torque filter constant	0
34	Push speed[mm/sec]	20
35	Safety speed[mm/sec]	100
36	Automatic servo OFF delay time[sec]	0
37	Automatic servo off delay time2 [sec]	0
38	Automatic servo off delay time3 [sec]	0
39	Positioning complete signal output method [0:PEND/1:INP]	0
40	Home input [0:Enable/1:Disable]	0
41	Operation mode input [0:Enable/1:Disable]	0
42	Enable function [0:Enable/1:Disable]	1
43	Home confirmation sensor input polarity	0
44	(For future expansion)	0
45	Silent interval magnification	0
46	Speed override [%]	100
47	PIO Jog speed2 [mm/sec]	100
48	PIO Inching Distance	0.10
49	PIO Inching Distance2	0.10
50	(For future expansion)	255



No	Name	Value
50	(For future expansion)	255
51	(For future expansion)	0
52	Addition and subtraction velocity mode initial value	0
53	Stop mode initial value	0
54	Current control band number	4
55	Position command primary filter time constant [msec]	0.0
56	Sigmoid motion ratio setting [%]	0
57	(For future expansion)	70
58	(For future expansion)	1
59	(For future expansion)	0
60	(For future expansion)	0
61	(For future expansion)	0
62	(For future expansion)	1
63	(For future expansion)	1
64	(For future expansion)	0
65	(For future expansion)	200
66	(For future expansion)	15
67	(For future expansion)	0
68	(For future expansion)	0
69	(For future expansion)	0
70	(For future expansion)	0
71	Positional feedforward gain	50
72	(For future expansion)	0
73	(For future expansion)	0
74	(For future expansion)	0
75	(For future expansion)	0
76	(For future expansion)	0
77	Screw lead[mm]	4.00
78	Axis action type	0
79	Rotary axis mode	0



No	Name	Value
80	Rotary axis shortcut select	0
81	(For future expansion)	0
82	(For future expansion)	0
83	Absolute unit[0:nonuse/1:use]	0
84	Fieldbus operation mode	3
85	Fieldbus node address	0
86	Fieldbus communication speed	0
87	Network type	6
88	Software limit margin[mm]	0.00
89	(For future expansion)	0
90	Fieldbus input/output format	3
91	Current limiting when pushing fails and it stops	0

Note that Parameter #84, **Fieldbus operation mode is set to 3**. This setting is required for Full Direct Value Mode. The factory default setting is 0 therefore a change is required. Make sure the switch on the front panel is set to Manual, modify the parameter, then click the “Load to CTL” icon. Once loaded, restore the switch to the Auto position for EtherCAT communications.

Parameter #85 is the Fieldbus node address, in EtherCAT terminology this is the Station Alias. The factory default is 17. **For automatic axis assignment set this address to 0**, otherwise an axis number from 1 to 16 that is unique for the network being controlled by the M3-41 module.

Special Variable Mapping

Due to the unique nature of the IAI interface a number of MSB variables have been mapped to be cyclically updated on each EtherCAT scan cycle. These variables are:

Outputs:

cmode – Mode of motion control, must be set to $HOMING_MODE or $PROFILE_POSITION_MODE.

homing_speed1 – Homing speed in mm/sec. Used in conjunction with the $HOMING_MODE ‘cmode’ command.

zone_limit_pos – Zone Boundry + in mm. Positive zone boundary limit. After completion of home return, an effective zone signal can be output separately from the zone boundaries specified by parameters. The status signal PZONE turns ON when the current position is inside the Zone Boundary +/- boundaries.

zone_limit_neg – Zone Boundry - in mm. Negative zone boundary limit. After completion of home return, an effective zone signal can be output separately from the zone boundaries specified by parameters. The status signal PZONE turns ON when the current position is inside the Zone Boundary +/- boundaries.

IAI_conrol – Maps to IAI Control Signal 1 & 2. Reference the Control Output Mapping section that follows.

tmax – Pressing Current Limit (%), specifies the current limit value to be used during the push-motion operation. The allowable specification range is 0 to 100%. The actual settable range varies depending on each actuator. If a move command is issued by specifying a value exceeding the maximum push-motion current and alarm and fault will occur.

tlim – Load Current Threshold (%), sets the current threshold when whether or not the load current exceeds the threshold is judged. The allowable range is 0 to 100% with 0 the threshold judgment if it is not required.

Inputs:

dins – The 'dins' variable maps to the IAI Control Status inputs. Reference the Control Status Mapping section that follows.

rmstrq – IAI Current in mA.

IAI_alarmcode - Maps directly to the IAI Alarm Code for general monitoring. This is the code that is referenced when a fault occurs and an EtherCAT Explorer log is generated.

PDO Mappings for EtherCAT Explorer:

PDO CNTLWORD – IAI Control Signal 2 outputs.

PDO STATUS – IAI Control Status inputs.

Control Output Mapping

The ACON controller has two Control Signal Output blocks, 1 & 2. These can be accessed directly referencing the 'IAI_control' MSB variable. Any changes to this variable will be output on the next EtherCAT scan cycle. 'IAI_control' is a 32 bit value with its lower 16 bit word being Control Signal 1 and upper 16 bits Control Signal 2:

<table>
<tr><td rowspan="13">PLC Output</td><td rowspan="13">Control Signal 1</td><td>b15</td><td colspan="2" rowspan="5">-</td><td colspan="2" rowspan="5">Unavailable</td></tr>
<tr><td>b14</td></tr>
<tr><td>b13</td></tr>
<tr><td>b12</td></tr>
<tr><td>b11</td></tr>
<tr><td rowspan="2">b10</td><td>ACON</td><td>-</td><td colspan="2">Unavailable</td></tr>
<tr><td>PCON</td><td>SMOD</td><td colspan="2">Stopping control mode: When this signal is ON, servo control is performed during stopping.</td></tr>
<tr><td>b9</td><td colspan="2">ASO1</td><td>Stop Mode 1</td><td rowspan="2">Select stop mode while standing by
ASO1 / ASO0 / Functions
OFF / OFF / Disable (Servo is ON at all times)
OFF / ON / Sever turns OFF in time set in Parameter No. 36
ON / OFF / Sever turns OFF in time set in Parameter No. 37
ON / ON / Sever turns OFF in time set in Parameter No. 38</td></tr>
<tr><td>b8</td><td colspan="2">ASO0</td><td>Stop Mode 0</td></tr>
<tr><td>b7</td><td rowspan="2">ACON</td><td>MOD1</td><td colspan="2" rowspan="2">Acceleration / deceleration mode:
When both signals are OFF, the trapezoid pattern mode is selected.
When one signal is OFF and the other signal is ON, the S-motion mode is selected. When one signal is ON and the other signal is OFF, the primary delay filter mode is selected.</td></tr>
<tr><td>b6</td><td>MOD0</td></tr>
</table>

Address		Bit	Symbol		Function
	Control Signal 1	b7	PCON	-	Unavailable
		b6			
		b5	-		Unavailable
		b4			
		b3	INC		Incremental Command: Absolute position commands are issued when this signal is OFF, and incremental position commands are issued when the signal is ON.
		b2	DIR		Push direction specification: "OFF" for the direction reducing the positioning band from the target position "ON" for the direction adding the positioning band to the target position
		b1	PUSH		Push-motion specification : Positioning operation is performed when this signal is OFF, and push-motion operation is performed when the signal is ON.
		b0	-		Unavailable

All but 'b3' is masked low level and available to the programmer. 'b3' is forced to 0 since the MSB command language supports incremental commands and automatically converts to absolute. When not using PUSH mode Control Signal 1 defaults to a 0, as does the variable 'IAI_control'. Control Signal 1 represents bits 0 to 15 in 'IAI_control'.

PLC Output	Control Signal 2	b15	BKRL	Forced brake release: When it is turned ON, the brake is released
		b14	RMOD	Operating mode selector: The AUTO mode is selected when this signal is OFF, and the MANU mode is selected when the signal is ON.
		b13	-	Unavailable
		b12		
		b11		
		b10		
		b9		
		b8	JOG+	+Jog: "ON" for Movement in the Opposite Direction of Home
		b7	JOG-	-Jog: "ON" for Movement to the Home Direction
		b6	JVEL	Jog-speed/inch-distance switching: The values set in parameter No. 26, "Jog speed" and parameter No. 48, "Inch distance" are used when this signal is OFF, and the values set in parameter No. 47, "Jog speed 2" and parameter No. 49, "Inch distance 2" are used when the signal is ON.
		b5	JISL	Jog/inch switching: Jog operation is performed when this signal is OFF, and inch operation is performed when the signal is ON.
		b4	SON	Servo ON Command: The servo turns ON when this signal turns ON.
		b3	RES	Reset: A reset is performed when this signal turns ON.
		b2	STP	Pause: A pause command is issued when this signal turns ON.
		b1	HOME	Home return: A home-return command is issued when this signal turns ON.
		b0	DSTR	Positioning Start: A move command is issued when this signal turns ON.

All but 'b4' is masked low level and available to the programmer. 'b4' is controlled by the 'drive enable' command, set to a 1 upon execution and defaulting to 0 at power up. The M3-41 will control RES, HOME, and DSTR as needed under program control. RES is used to clear an alarm condition. Control Signal 2 represents bits 16 to 31 in 'IAI_control'.

Control Status Mapping

The ACON controller has one 16 bit Control Status Input block. These can be accessed directly referencing the 'dins' and 'din#' MSB variables. The variable is updated every scan cycle. From the IAI manual the status flag are defined as below and represent what appears in the PDO STATUS row of the EtherCAT Explorer.

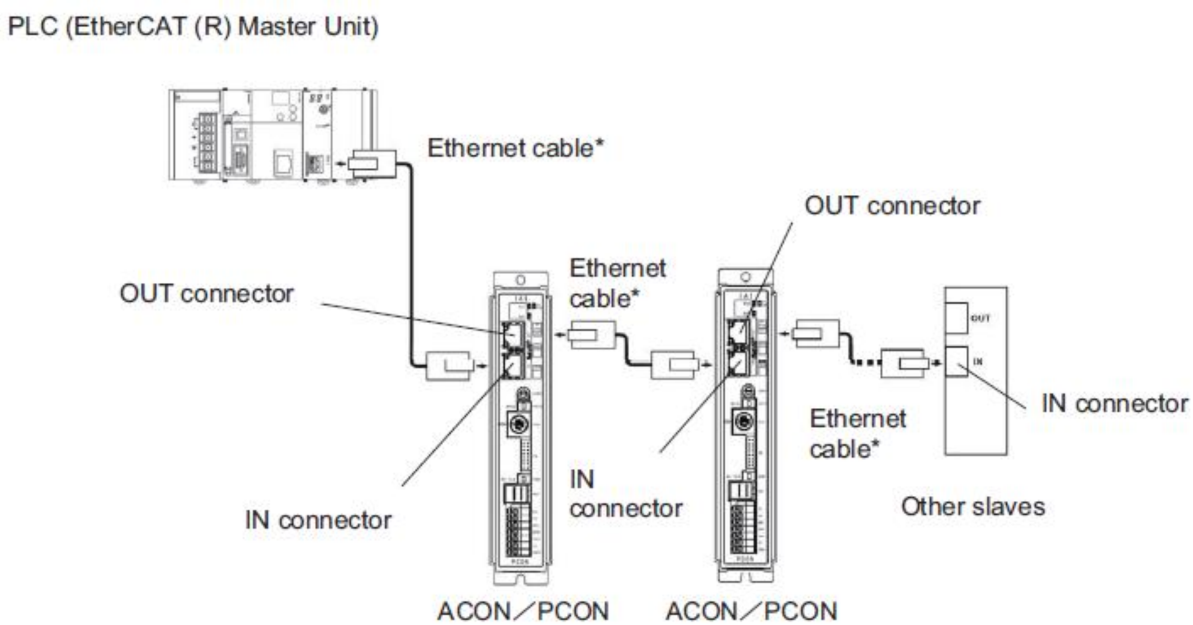
PLC Input	Status Signal	b15		EMGS	Emergency stop: An emergency stop is actuated when this signal turns ON.
		b14		PWR	Controller ready : This signal turns ON when the controller becomes ready.
		b13		ZONE2	Zone 2:"ON" for the current position within the zone set range
		b12		ZONE1	Zone 1:"ON" for the current position within the zone set range
		b11		PZONE	Position zone: This signal turns ON when the current position is inside the specified position zone.
		b10	ACON	-	Unavailable (ON/OFF is undefined)
			PCON	LOAD	Load output judgment: When this signal is ON, the specified load has been reached. When the signal is OFF, the load has not been reached yet. (Refer to Operation Manual for the controller main body for more information)
		b9	ACON	-	Unavailable (ON/OFF is undefined)
			PCON	TROS	Torque level: When this signal is ON, the specified load has been reached. When the signal is OFF, the load has not been reached yet. (Refer to Operation Manual for the controller main body for more information)
		b8		RMDS	Operation Mode Status: This signal is OFF when the current mode is AUTO, or ON when the current mode is MANU.
		b7		GHMS	Under Home return Operation: This signal remains ON while home return is in progress.
		b6		PUSHS	Push motion in progress: This signal remains ON while push-motion operation is in progress.
		b5		PSEL	Pressing and a Miss: This signal turns ON when the actuator missed the work part in push-motion operation.
		b4		SV	Operation preparation end: This signal turns ON when the servo turns ON.
		b3		ALM	Alarm: This signal turns ON when an alarm occurs.
		b2		MOVE	Moving Signal: This signal remains ON while the actuator is moving.
		b1		HEND	Home return completion: This signal turns ON when home return is completed.
		b0		PEND	Positioning completion signal: This signal turns ON when positioning is completed.

When referencing '*dins'* and '*din#*' variables the bits have been reordered to allow the use of the din1 to din10 bit referencing within the MSB language. Thus when referencing these variables the status flags are ordered as follows:

* b15 - LOAD (not used)
* b14 - TROS (not used)
* b13 - RMDS (Auto/manual)
* b12 - GHMS
* b11 - PUSHS
* b10 - PWR
* b09 - SV
* b08 - ALM
* b07 - MOVE
* b06 - HEND
* b05 - PEND
* b04 - EMGS
* b03 - PSEL
* b02 - ZONE2
* b01 - ZONE1
* b00 - PZONE

Operation

The ACON controller is initially configured using the ROBO Cylinder Software from IAI. This software allows for the setting of Parameters in non-volatile storage within the controller via a serial cable. As previously noted Fieldbus Operation Mode, parameter 84, should be set to a 3 for Profile Position mode type motions and Fieldbus Node Address, parameter 85, to 0 for automatic sequential axis assignment or an axis number from 1 to 16 (station alias). Once the parameters are saved the Manual/Auto switch on the front panel of the controller must be set to Auto for M3-41 EtherCAT control. The address switch should also be set to 0 and the BKLS switch to NORMAL.

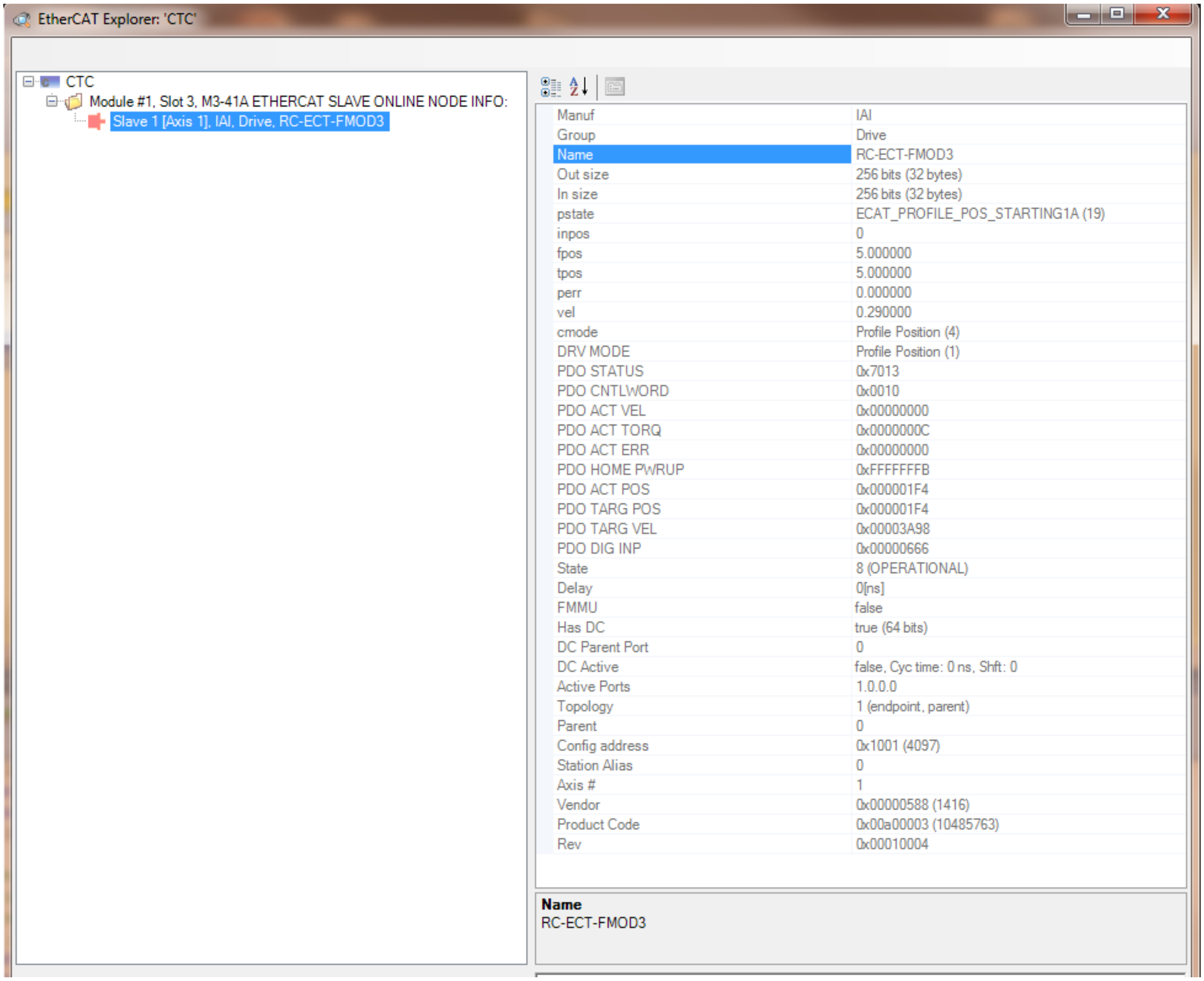


* Ethernet cable: Straight cable of category 5e or above, 100 m max
(Aluminum tape and braided double-shielded cable are recommended.)

The ACON Controller is supported only in Full Direct Value Mode (parameter 84 = 3). This allows for the most flexibility from an M3-41 perspective, controlling acceleration, velocity, torque, and position dynamically. The factory default mode uses a table stored in the ACON Controllers non-volatile memory. This mode, known as Remote I/O mode is not supported and is the least flexible.

Other parameters that may be of interest are 1/2 and 23/24 for Zone Position 1 and 2 respectively. This controls the ZONE1 and ZONE2 bits in *dins* when the position is within the assigned window. Variables *zone_limit_pos/zone_limit_neg* are used for the dynamic PZONE bit window control.

A properly configured IAI unit will appear online as is shown below within the EtherCAT Explorer, note the RC-ECT-FMOD3 drive type. The FMOD3 represent Full Direct Mode, anything else appearing here means that Parameter 84 is set wrong.



All programming units are in mm by default and the *'ppr'* should be set to 100 since the IAI controller is using .01 mm per unit for default positioning, thus 100 * .01 = 1 mm/unit. Normal MSB move commands may be used, specifying position (mm), velocity (mm/s), and acceleration (G's).

```
inposw = .01;      // Positioning Band, .01mm
// Zone boundary can optionally be used
zone_limit_pos = 24.0;
zone_limit_neg = 16.0;  // PZONE bit in dins will go active from 16 to 24 mm
tmax = 70;         // 70% Pressing Current Limit
tlim = 0;          // 0 % Load Current (threshold judgement)
drive enable;

// Let the drive home
[home]
homing_speed1 = 20;     // homing_speed1 is used, 20 mm/sec

cmode = $HOMING_MODE;   // Homing mode for drive
move to 0 using .3,.3;  // Tell the drive to initiate the Home.
wait for in position;
```

```
delay 100 ms;

// IAI only support a full profile position type mode but we can setup the
profile.
cmode = $PROFILE_POSITION_MODE;

[top]
// Move at a velocity of up to 150 mm/sec for 5 mm
// at an acceleration of .3 G and deceleration
// of .3G incrementally
move at 150 for 5 using 0.3,0.3;
wait for in position;
delay 100 ms;
move at 150 for 5 using 0.3,0.3;
wait for in position;
delay 100 ms;
move at 150 for 5 using 0.3,0.3;
wait for in position;
delay 100 ms;
move at 150 for 5 using 0.3,0.3;
wait for in position;
delay 100 ms;
move at 150 for 5 using 0.3,0.3;
wait for in position;
delay 100 ms;
move at 150 for 5 using 0.3,0.3;
wait for in position;
delay 100 ms;

// Move all the way back at a velocity
// of up to 200 mm/sec for -30 mm
// at an acceleration of .3 G and deceleration
// of .3G incrementally.
move at 200 for -30 using 0.3,0.3;
wait for in position;
delay 100 ms;
goto top;          // repeat
```

By default POSITION mode is used. IAI supports a PUSH mode as well and that can be enabled by setting bits b2 and b1 of Control Signal 1 to their proper value, with b1 ON enabling PUSH mode. When using PUSH mode a ‘Pressing and a Miss’ flag, PSEL b4, must be monitored since this is not considered an alarm condition and is recoverable by the program. When a miss occurs *tpos* will be set equal to *fpos* to allow exiting from the ‘wait for in position’ instruction. The variables *ztpos* and *zfpos* will contain the values prior to the change should they need to be referenced. Below is a code sample using the PUSH capability:

```
inposw = .01;      // Positioning Band, .01mm
// Zone boundary can optionally be used
zone_limit_pos = 24.0;
zone_limit_neg = 16.0;  // PZONE bit in dins will go active from 16 to 24 mm
tmax = 70;         // 70% Pressing Current Limit
tlim = 0;          // 0 % Load Current (threshold judgement)
drive enable;

// Let the drive home
[home]
```

```
homing_speed1 = 20;     // homing_speed1 is used, 20 mm/sec

cmode = $HOMING_MODE;                 // Homing mode for drive
move to 0 using .3,.3;  // Tell the drive to initiate the Home.
wait for in position;

delay 100 ms;

// IAI only support a full profile position type mode but we can setup the
profile.
cmode = $PROFILE_POSITION_MODE;

// enable push motion with positive direction
IAI_control = 0x00000006;
inposw = 1; // 1 mm push

[top]
// Move at a velocity of up to 150 mm/sec for 5 mm
// at an acceleration of .3 G and deceleration
// of .3G incrementally.  Check for Push/Miss after move.
move at 150 for 5 using 0.3,0.3;
wait for in position;
if (dins & 0x00000008) goto PSEL_Error;
delay 100 ms;
move at 150 for 5 using 0.3,0.3;
wait for in position;
if (dins & 0x00000008) goto PSEL_Error;
delay 100 ms;
move at 150 for 5 using 0.3,0.3;
wait for in position;
if (dins & 0x00000008) goto PSEL_Error;
delay 100 ms;
move at 150 for 5 using 0.3,0.3;
wait for in position;
if (dins & 0x00000008) goto PSEL_Error;
delay 100 ms;
move at 150 for 5 using 0.3,0.3;
wait for in position;
if (dins & 0x00000008) goto PSEL_Error;
delay 100 ms;
move at 150 for 5 using 0.3,0.3;
wait for in position;
if (dins & 0x00000008) goto PSEL_Error;
delay 100 ms;

// Move all the way back at a velocity
// of up to 200 mm/sec for -30 mm
// at an acceleration of .3 G and deceleration
// of .3G incrementally.
move at 200 for -30 using 0.3,0.3;
wait for in position;
// Check for Pressing and Miss error
if (dins & 0x00000008) goto PSEL_Error;
delay 100 ms;
goto top;

[PSEL_Error]
```

```
i = i+1;                    // Press and miss error occurred
delay 1000 ms;
zero following error;
delay 1000 ms;
goto home;
```

Errors and Alarms

The ACON controller will generate cause an MSB fault and EtherCAT log entry under certain error conditions, some are recoverable, and some require power cycling of the IAI Controller. Reference the IAI Controller manual for details. In general it has been found that the Overload error, alarm code 0x00E0 requires power cycling.

The most common error is the Position Command Error, 0x00a3. This error typically occurs when a profile parameter has been exceeded, such as velocity, acceleration, position out of range, or pressing/load current %. Sometimes referencing the Parameter table stored within the controller, via the ROBO Cylinder Software, is helpful in discovering what profile parameter exceeded a limit. For example for the test cylinder used during EtherCAT integration 0.30 G's was the maximum acceleration/deceleration allowable by the model RCA2-TF4N-I-20-4-30-A3-P actuator. Setting this to 0.31 G's would cause an ALM condition and 0x00a3 error to be logged.

Blank

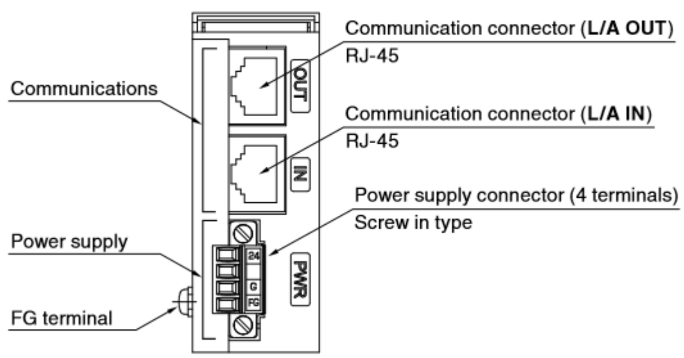
Appendix

I

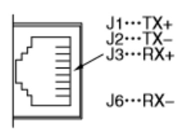
[I] Koganei

Koganei manufactures a number of solenoid valves, specifically supported here are the YS7K1 (16 output) and YS7K3 (32 output). The outputs appears as digital outputs within QuickBuilder. This section provides information that may be specific this device.

Connectors



Communications
(RJ-45 communication connector)
J1···TX+
J2···TX–
J3···RX+
J6···RX–



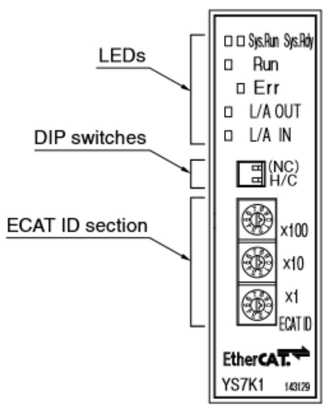
Power supply (applicable wire sizes: AWG 28 to 16, tightening torque: 0.22 to 0.25 N·m [1.95 to 2.21 in·lbf])

24···24 VDC + terminal
G···24 VDC – terminal
FG···Frame ground terminal

FG terminal (tightening torque: 0.49 N·m [4.34 in·lbf])
Ground terminal connection (connected internally to the FG on the power supply terminal)

Display Panel

Display panel

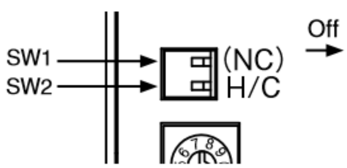


* The illustration shows the panel for the YS7K1, but the YS7K3 panel is the same.

LED displays and descriptions

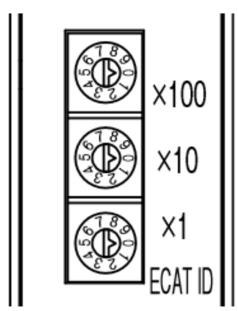
Name	State	Color	Description
Sys.Run/Sys.Rdy	Lit/Not lit	Green/yellow	• Transmission block operation normal
	Flashing/flashing	Green/yellow	• Transmission block initialization
	Not lit/lit or flashing	Green/yellow	• Transmission block error
	Not lit/Not lit	Green/yellow	• Transmission block power off
Run	Off	Green	• INIT
	Flashing (blinking)	Green	• PRE-OPERATIONAL
	Flashing (single flash)	Green	• SAFE-OPERATIONAL
	Lighted	Green	• OPERATIONAL
Err	Off	Red	• No error
	Flashing (blinking)	Red	• Invalid setting
	Flashing (single flash)	Red	• Unrequested change in status
	Flashing (double flash)	Red	• Communication disconnect
L/A OUT L/A IN	Lighted	Green	• Normal communication
	Flashing	Green	• EtherCAT frame sending/receiving
	Off	Green	• Not connected

Description of DIP switches



SW1···(NC): Not used
SW2···On (H: Hold output), Off (C: Clear output)

Description of ECAT ID section



Range of ECAT ID settings (0 to 999)
0···Writes slave node address
1 to 999···Possible range of settings

* Open the cover and use a flat screwdriver to set hold or clear (H/C) and to set the address on the rotary switches.
* Always turn off the power supply before doing settings.

Output Mapping

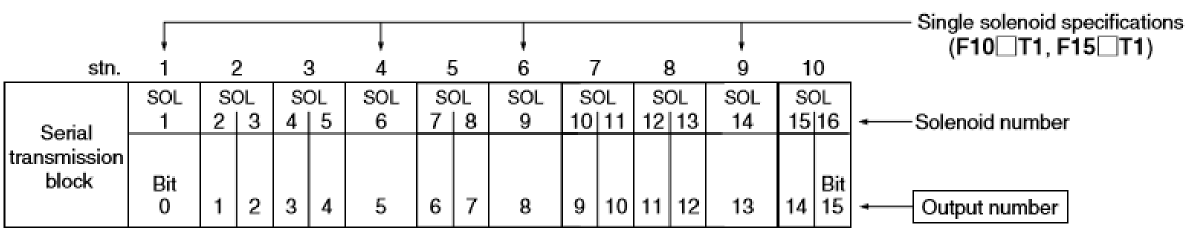
The solenoid manifold block has either 16 or 32 outputs. The relationship between the output numbers in the program and the actually mounted solenoid valves is as show below and different depending upon what is specified for "Wiring specifications" in the manifold ordering codes.

Wiring specifications:
Blank (packed wiring): wired according to specifications of mounted valves.
-W (double wiring): All wiring is for double solenoids, regardless of the specifications of the mounted valves.

1. If wiring specifications are "blank" (packed wiring)

The valves specified as the single solenoid specifications (**F□T1**) when ordering are wired to solenoid A only and are not wired to solenoid B because wiring is done according to the mounted valve specifications.

This means that it cannot function as a double solenoid valve after it is delivered because no current flows to solenoid B, even if it is switched from single solenoid valve to double solenoid valve.



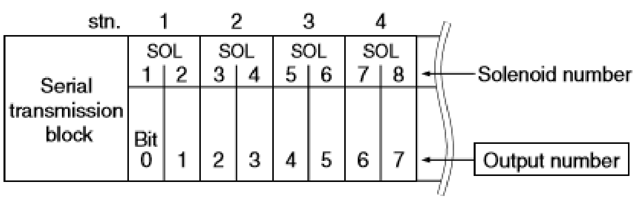
For 16 outputs
WORD (16bit) **15· · · · · · · · · ·0**
MSB LSB
For 32 outputs
DWORD (32bit) **31· · · · · · · · · ·0**

Output number relationship chart (example)

Output number	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Solenoid number	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

2. When wiring specifications are "-W" (double wiring)

All wiring is for double solenoids.



Blank

Appendix

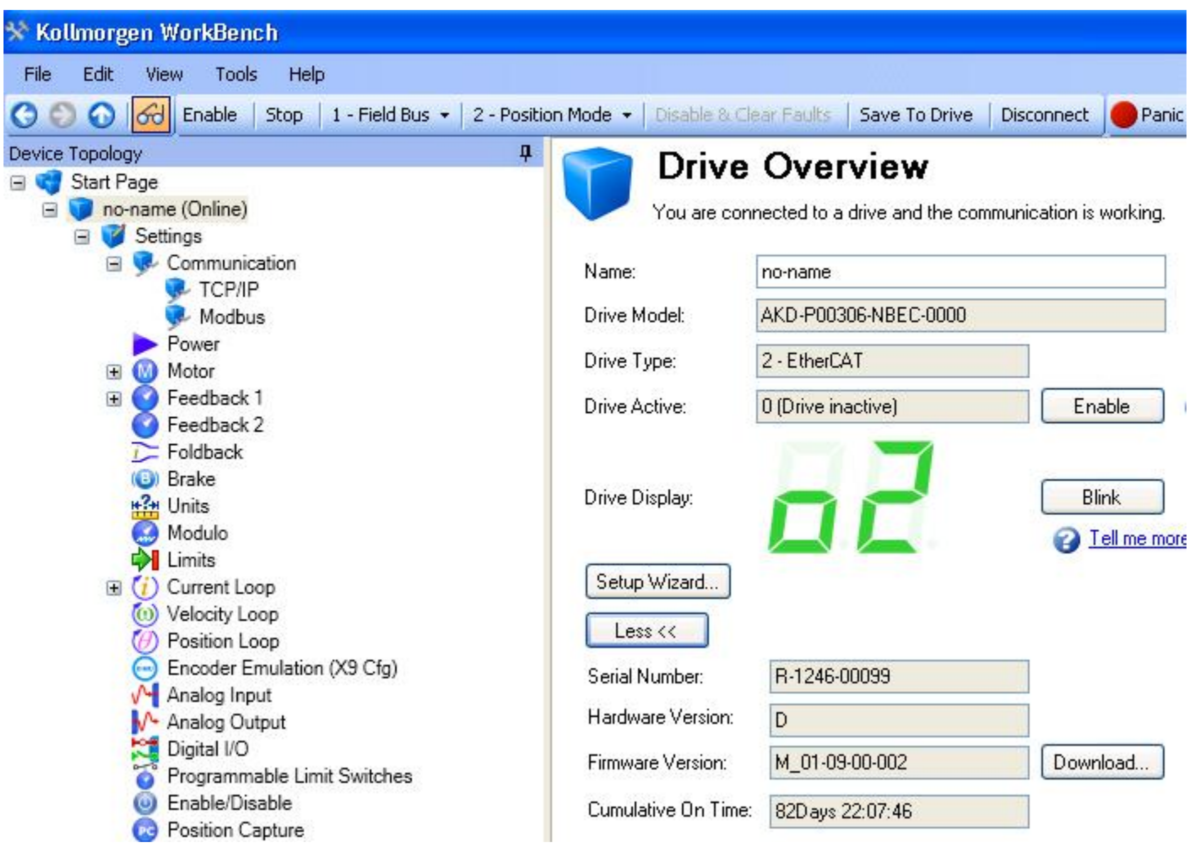
J

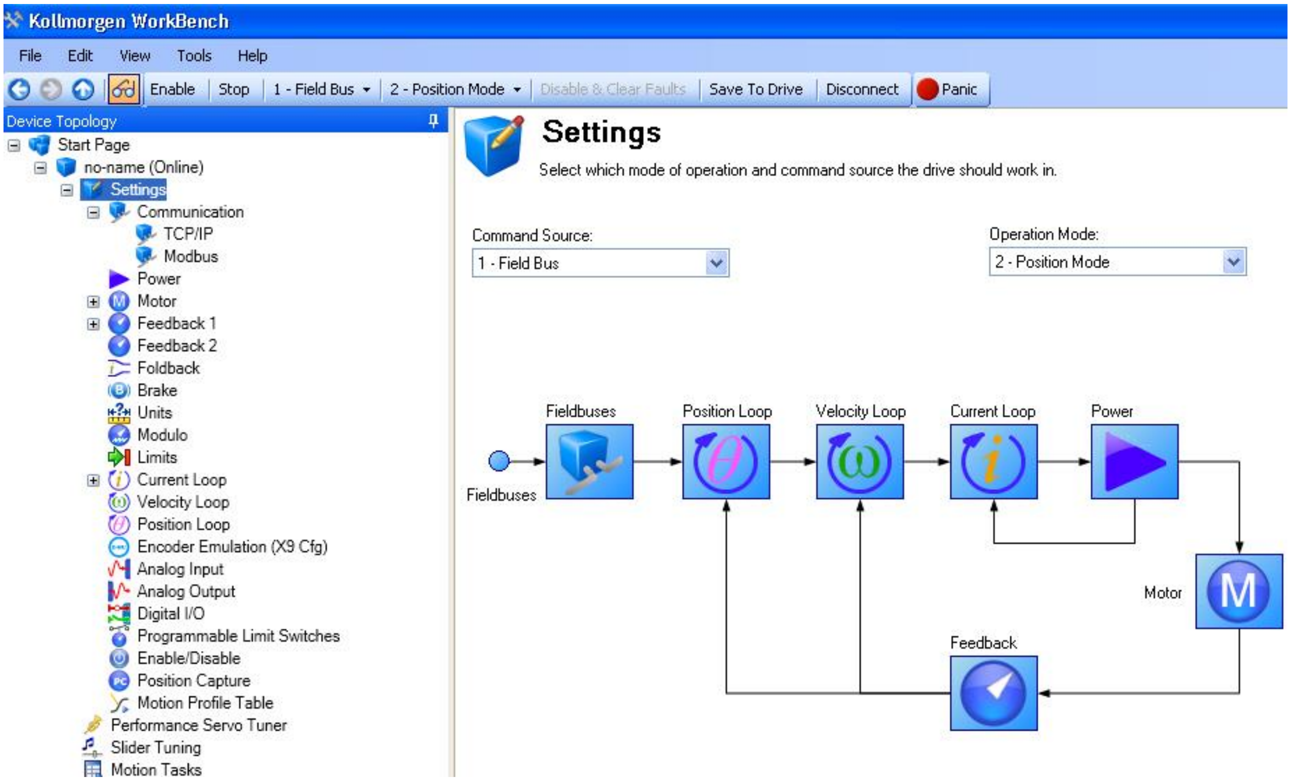
[J] Kollmorgen

Kollmorgen manufactures a number of drives. The drive currently supported is the single axis AKD servo drive. This section provides information that may be specific to this manufacturer.

eCAT_driveType – 5

Drive Information & Firmware





Positioning Mode

Kollmorgen AKD drives do not support Profile Position mode; Cyclic Sync Position must be used. With this drive, the EtherCAT Master automatically switches to Cyclic Sync Position mode even if Profile Position mode is selected. Interpolated Position mode will lag one scan cycle behind in the move operation from that commanded.

Powerup Delay

Unlike most drives, Kollmorgen can take up to 20 seconds to power up and be recognized on the network. Make sure all drives are powered on and ready before powering up the controller. Otherwise the drive(s) will not be seen. If a configuration file is loaded into the EtherCAT Master, it will retry for the 20-second period prior to aborting, allowing both the controller and the drive to be power cycled at the same time. Combining this with Option Switch #1 (retry override, which continuously scans the network for proper configuration when enabled) will ensure successful power up.

Appendix

K

[K] LinMot

LinMot manufactures a number of linear motor drives. Those currently supported are the C1250-DS-XC-0S and the E1450-DS-QN-0S. This section provides information that may be specific to this manufacturer.

eCAT_driveType – 15

Overview

Only CSP mode is supported with the LinMot series of drives using their CiA402 interface. Homing is done with a series of move commands, executing a 'zero feedback position' to zero out the position. Position capture is not supported, only simple moves. Default units are as follows:

Velocity – mm/s
Acceleration – mm/s^2
Position – millimeters (mm)
ppr – 10000 (.0001 mm per position count)
inposw – typically .01 mm, E1450 may require a greater value depending upon load and tuning.

For proper operation you must verify the firmware revision levels installed in the drive using their LinMot-Talk PC software. During testing a number of anomalies were discovered that required firmware updates that at the time were not the default shipped with their LinMot-Talk software. The tested revisions are as follows:

C1250:
Firmware Release – 6.4 Build 20151105
OS – OSSW_C1200_V6S4_b01
MC – MCSW1200_S21_V6S4_b01
INTF – IntfSWEC_SG6_V6S5_a01 (contains fixes supplied by LinMot, .HX3 file)

E1450:
Firmware Release – 6.4 Build 20151105
OS – OSSW_E1400V2_V6S4_b01
MC – MCSW1400_S32_V6S4_b01
INTF – IntfSWEC_SG6_V6S5_a01 (contains fixes supplied by LinMot, .HX3 file)

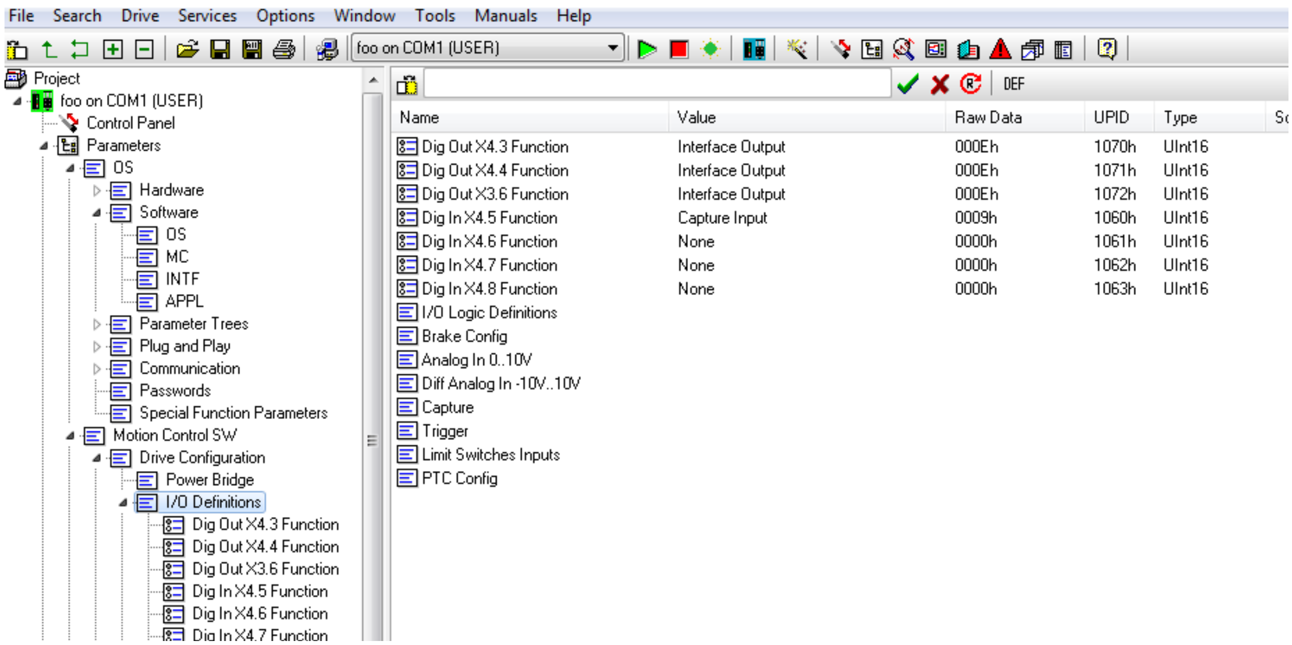
EtherCAT Connectors:

C1250: X17 IN, X18 Out

E1450: X17 IN, X18 Out

I/O

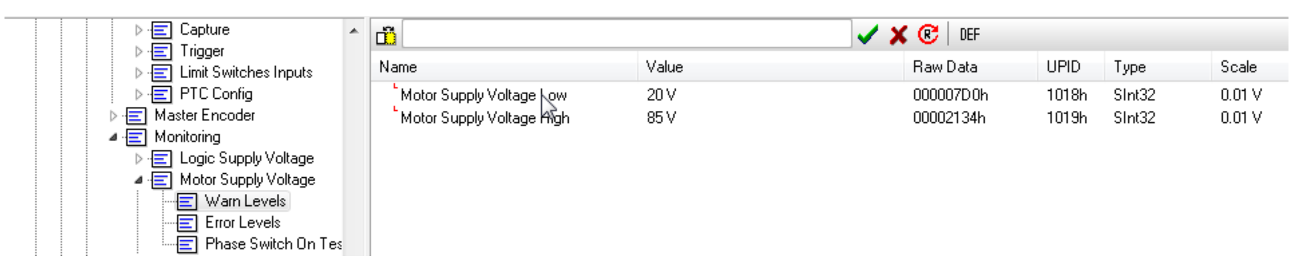
For I/O to be functional its interface must first be enabled using LinMot-Talk:

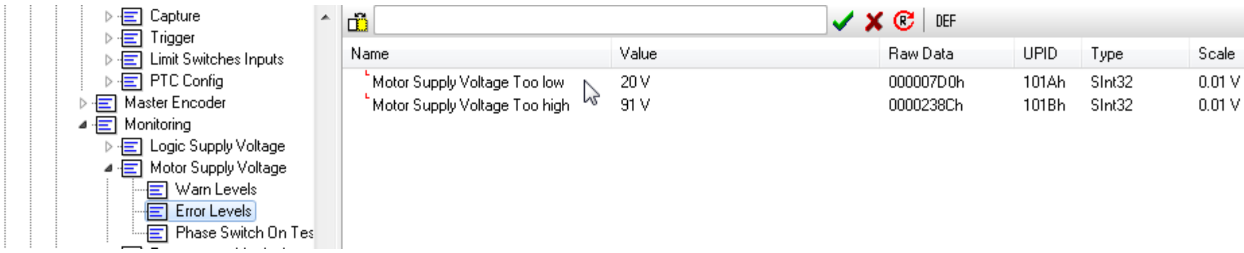


IO maps direction to the X4 interface where X4.3 is bit 3 of the output and X4.5 is bit 5 of the input, as referenced by LinMot-Talk.

Motor Voltage Levels

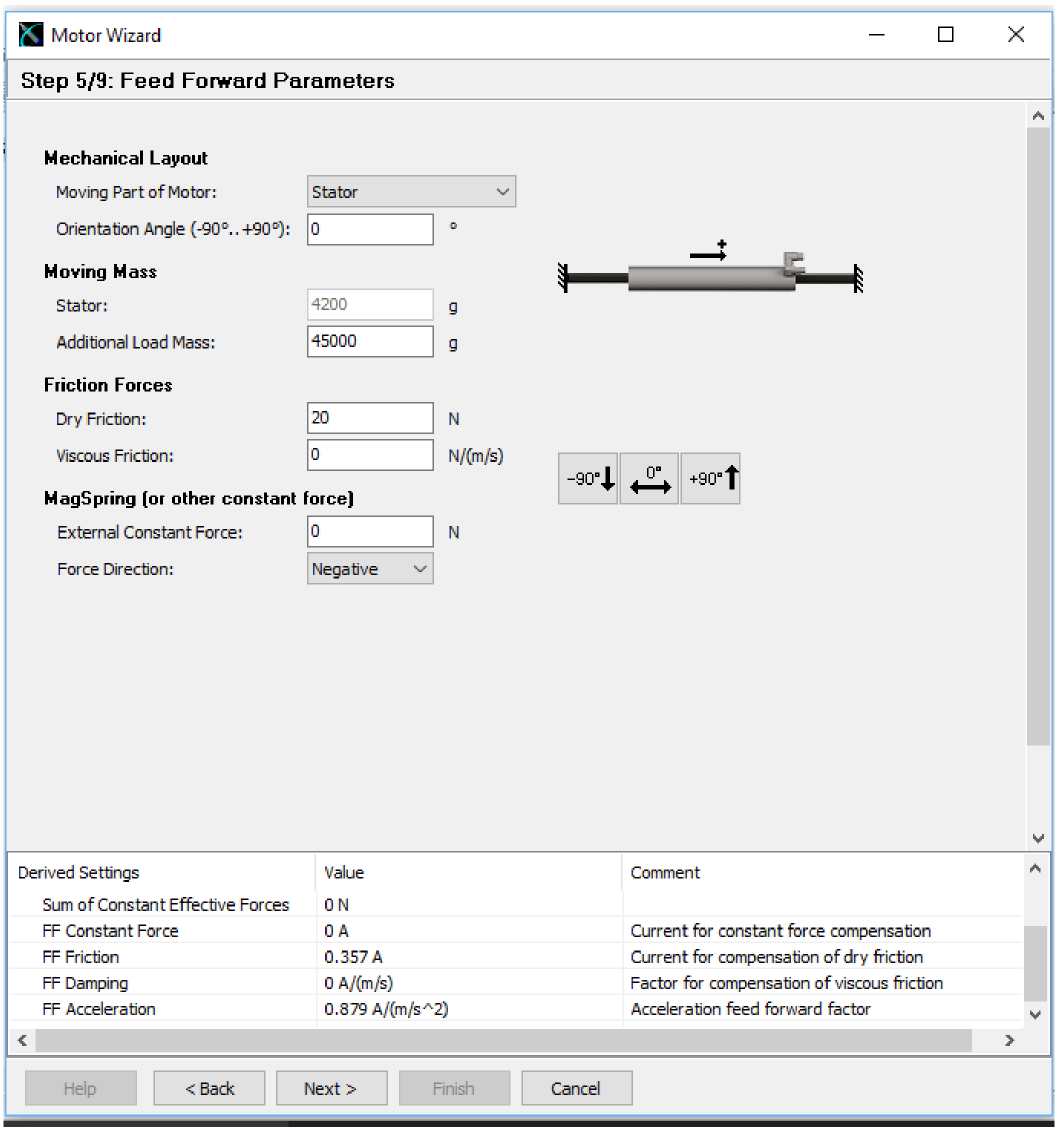
If low motor voltage is being used, for example 24V on a higher voltage motor, the warning and error detection levels must be adjusted accordingly:





Motor Wizard

When using the Motor Wizard it is recommended that the "Additional Load Mass" and "Dry Friction" be set to 0. Failure to do so can cause the LinMot drive not to track closely with the EtherCAT commanded position.

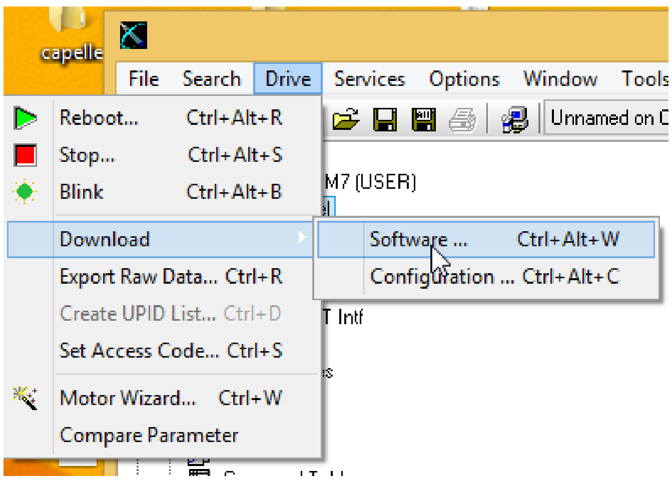


Control Parameter A/B

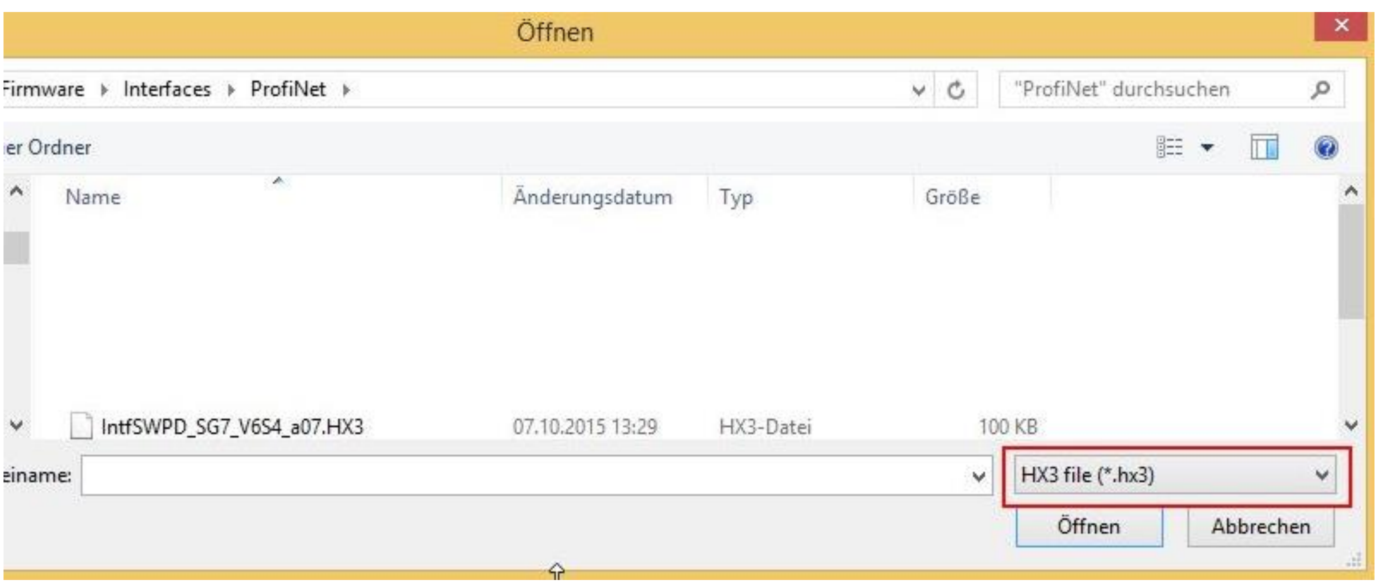
LinMot provides two sets of tuning parameters. By default Control Parameter A is used. The variable 'tmax' is used to set the Maximum Current, UPID 0x13A6 & 0x13BA. You may select Control Parameter B by writing a 1 to the msb variable 'IAI_control'.

INTF file upgrade (.HX3)

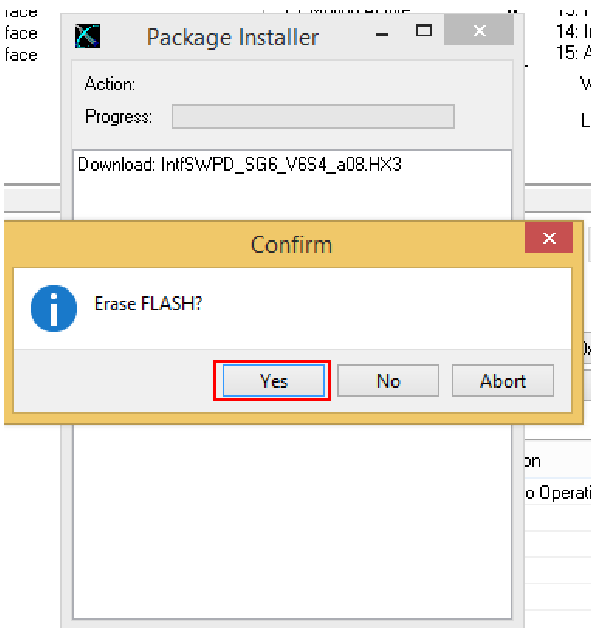
Log into the individual drive with LinMot talk then go to the Drive Toolbar and select download/Software.



A dialogue box will open, and in this window you will need to locate the file named "IntfSWEC_SG6_V6S5_a01.HX3". Make sure to select HX3 files from the dropdown menu outlined below:



The next popup will prompt you to erase the FLASH program, and you will select Yes.



After installation is complete cycle power on the LinMot Drive.

Test MSB

```
// This is a background MSB.  Make sure inposw is set for the drive, typically
// .001 for 1048576 ppr.  Also set the ppr and mppr.  This program will set
// the ppr and mppr to a value commonly used, it may have to be changed based
// on the user setup.

// Enable the drive, turning power on to the amplifier.  The current position
// will be constantly updated so the drive does not move.

// Activate DC Sync0 each cycle time.
// dcsync <slave node or -1 for current>,
//      <Sync0 Cycle Time in nanoseconds, ns>,
//      <Sync1 shift from Sync0, ns>,
//      <Sync0 shift from Cycle Time, ns>,
//      <Sync start delay in ns>
// Set all parameters to 0 except the slave node to deactivate.
// Below is a 1ms Sync0 cycle time with no Sync1.
// Sync0 starts at cycle time and is not shifted and there is a
// 100ms delay before it all starts the first sequence.

// Set dec, acc, and max velocity.
dec = 100;
acc = 100;
vmax = 50;
driveenable = 0;  // disable output control since LINMOT does not need it

drive_type = eCAT_driveType;
// Adjust the ppr and mppr based on the drive/encoder we have installed.
// This overrides that of the property sheet.
if eCAT_driveType == 2 goto Copley;
if eCAT_driveType == 3 goto Yaskawa;
if eCAT_driveType == 4 goto Elmo;
if eCAT_driveType == 5 goto Kollmorgen;
if eCAT_driveType == 6 goto Sanyo_Denki;
if eCAT_driveType == 7 goto Emerson;
if eCAT_driveType == 8 goto AMC;
if eCAT_driveType == 9 goto Virtual;
if eCAT_driveType == 12 goto ABB;
```

```
if eCAT_driveType == 15 goto LinMot;
goto AMC;
// by default assume 1048576
[Yaskawa]
[Kollmorgen]
mppr = 1048576;
ppr = 1048576;
goto beginTest;

[ABB]
mppr = 524288;
ppr = 524288;
goto beginTest;

[Virtual]
mppr = 65536;
ppr = 65536;
// After each control loop fposc = tposc so it appears as though the axis moved during
// a move command
//set simulated feedback on;
goto beginTest;

[Copley]
[Elmo]
mppr = 8000;
ppr = 8000;
goto beginTest;

[Emerson]
mppr = 65536;
ppr = 65536;
goto beginTest;

[Sanyo_Denki]
mppr = 131072;
ppr = 131072;
goto beginTest;

[AMC]
mppr = 12000;
ppr = 12000;
goto beginTest;

[LinMot]
mppr = 10000;
ppr = 10000;
inposw = .1;   // This can be tightened up after tuning, C1250 about .01, E1450 was not tuned so .1
goto beginTest;

[beginTest]
cmode = $CYCLIC_SYNC_POSITION_MODE;
//inposw=.001;

// Initialize distributed clocks
delay 3000 ms;
dcsync -1, 1000000, 0, 0, 100000000;
delay 200 ms;  // starts 100 milliseconds into the future
drive enable;
delay 1000 ms;
zero feedback position;  // assume current position is home and zero our position
[run]
// Begin the move, 20 mm/second for 30mm
move at 20 for 30;
wait for in position;
//setout 1,2,3,4,5,6,7,8;
// Delay 100ms once in position
delay 100 ms;
// Do a relative move back 30mm  at 20 mm/second
move at 20 for -30;
wait for in position;
// Delay 100ms second once in position
```

```
//clrout 1,2,3,4,5,6,7,8;
delay 100 ms;
// Do it again, forever...

goto run;
```

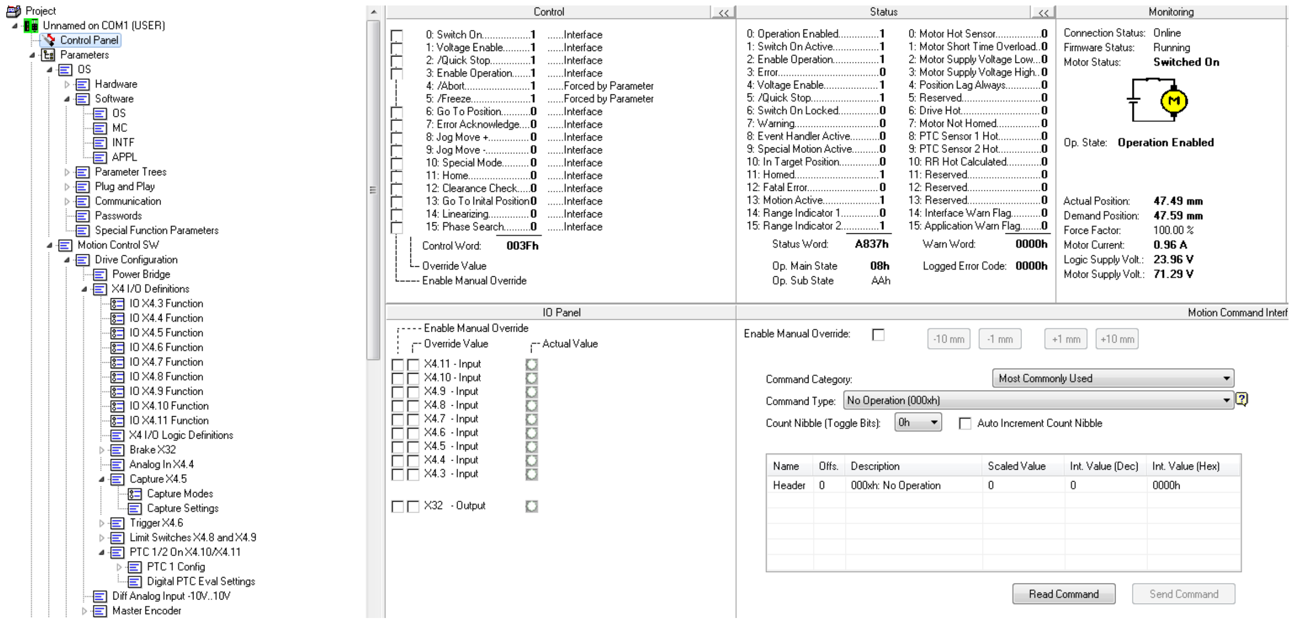
Homing MSB

When homing based on torque a position error will build up. This error is generated by the LinMot drive as the slider is stalled. Proper homing consists of moving in the direction of home while monitoring torque. Once the desired torque level is reached ‘stop’ the motor and then move in the opposite direction the amount of ‘perr’. Since the error was accumulated by the drive while it was stalled the move of the amount of ‘perr’ simply syncs the position reported by the drive with that given by the program as the target postion, removing the error. Once at that position you can clear the feedback position and begin normal moves.

```
//**********************************HomeRoutine********************************
[HomeRoutine]
// A setting to define which direction we're traveling in to home the axis.
Direction = -1;

// Move the axis based on commands fed from main program for homing.
// We move some very large amount based on direction until torque is large.
move at Maxspeed to (1000000 * Direction) using Accel,Decel;
delay 10 ms;

/*** Scan for excessive torque indicating we hit something
*     ...hopefully it's the end-stop
*/
[HomeMoveLoop]
// Check for user to end the move early.
if command == 0 goto StopMove;
// Check for the end-stop using accumulated torque (rmstrq).
// This value may have to be adjusted based upon the application.
if rmstrq >= .9 goto HomeStopAndZero;
if rmstrq <= -.9 goto HomeStopAndZero;
goto HomeMoveLoop; // loop until something happens

[HomeStopAndZero]
stoprate = Decel;// Set stopping decel rate
delay 10 ms;
//stop the axis because we've reached our defined torque level from above.
stop;
delay 500 ms;

// We've theoretically gone beyond the end-stop causing following-error
// to accumulated during the decel transition.
// Move back to the point where we began accumuating following-error along
// with an additional short 1mm move to us get off the end-stop.
// Direction is defined previously.
move at Maxspeed for ((perr * Direction) + (1 * -Direction));
wait for in position;
delay 500 ms;
zero feedback position; //Clear the feedback position
delay 10 ms;
goto MoveToOffsetPosition;  // All set we are home, do any move needed...
```

LinMot-Talk Control Panel

The LinMot-Talk Control Panel can be used during drive operation to monitor operation by attaching an RS232 port from the PC to X19, Config RS232 port. You must then use their Login function prior to communications.



EtherCAT Explorer View

Manuf	LinMot
Grp	Drive
Name	C1250-DS-XC-0S
Out	176 bits (22 bytes)
In	208 bits (26 bytes)
Axis #	1
pstate	COMPLETE (2)
tracking_pstate	COMPLETE (2)
inpos	0
fpos	29.933399
tpos	30.000000
perr	0.066601
vel	0.000000
DRV MODE	Cyclic Sync Position (8)
PDO STATUS	0x1237
PDO CNTLWORD	0x000F
PDO ACT VEL	0x00000000
PDO ACT TORQ	0x000000BD
PDO ACT ERR	0x00000000
PDO HOME PWRUP	0xFFFFFE3D
PDO ACT POS	0x00048F83
PDO TARG POS	0x0004921D
PDO TARG VEL	0x00000000
PDO DIG INP	0x00000000
State	8 (OPERATIONAL)
Delay	0 ns
Has DC	true (64 bits)
DC Parent	0
DC Active	true, Cyc time: 1000000 ns, Shft: 0
Parent	0
Config addr	0x1001 (4097)
Station Alias	0
Vndr	0x4c4e5449 (1280201801)
Product Code	0x0096096f (9832815)
Rev	0x00010008

Manuf	LinMot
Grp	Drive
Name	E1450-DS-QN-0S
Out	176 bits (22 bytes)
In	208 bits (26 bytes)
Axis #	1
pstate	RUNNING (1)
tracking_pstate	COMPLETE (2)
inpos	0
fpos	13.985600
tpos	13.859999
perr	-0.105601
vel	-19.409000
DRV MODE	Cyclic Sync Position (8)
PDO STATUS	0x1237
PDO CNTLWORD	0x000F
PDO ACT VEL	0xFFFFB42F
PDO ACT TORQ	0x0000FD48
PDO ACT ERR	0x00000000
PDO HOME PWRUP	0xFFFAC2FE
PDO ACT POS	0xFFFCE54E
PDO TARG POS	0xFFFCE066
PDO TARG VEL	0x00000000
PDO DIG INP	0x00000000
State	8 (OPERATIONAL)
Delay	0 ns
Has DC	true (64 bits)
DC Parent	0
DC Active	true, Cyc time: 1000000 ns, Shft: 0
Parent	0
Config addr	0x1001 (4097)
Station Alias	0
Vndr	0x4c4e5449 (1280201801)
Product Code	0x0096096b (9832811)
Rev	0x00010007

Appendix

L1

[L1] Mitsubishi J4/J5

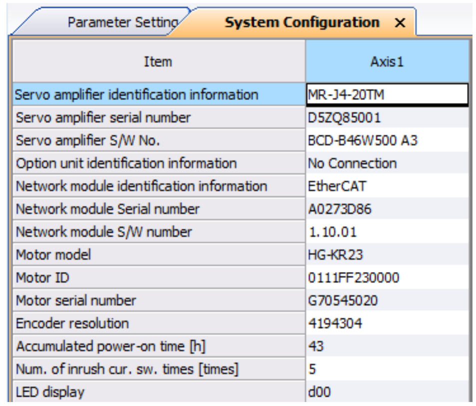
Mitsubishi manufactures a number of drives. That currently supported by the M3-41A is the J4 and J5. For test purposes two drives in particular were tested: MR-J4-20TM-ECT, MR-J4-40TM1-ETP and the MR-J5-20G-RJN1. This section provides information that may be specific to this manufacturer.

J4 eCAT_driveType – 13 (ppr – 4,194,304)
J5 eCAT_driveType – 28 (ppr – 67,108,864)

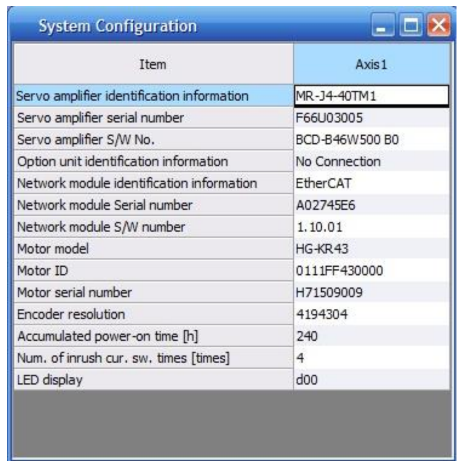
Overview

Only CSP and Homing modes are supported with the Mitsubishi drives. Additionally, the DC SYNC motion control instruction is ignored since it is automatically enabled prior to network operation. Thus, it is automatically enabled by the EtherCAT Master with the SYNC0 always having an offset from the network cycle time, typically 1 mS cycle time, with a shift of 'cycle time/4' or 250uS for 1 mS, whichever is smaller. SYNC1 is not supported by the drive. Additionally, early versions of Mitsubishi firmware had jitter in their clock when used as the master slave device. If a synchronization problem occurs it is recommended to replace the first slave device with a different device, such as Wago, or Omron switch, otherwise contact Mitsubishi to ensure you have the latest firmware.

For proper operation you must verify the firmware revision levels installed in the drive using their MR Configurator2 PC software. The 'Control mode selection' of the drive must also be set to 'Cyclic synchronous mode'. Below is information on the drives used for testing, note the servo amplifier software revision of 'BCD-B46W500 A3', this firmware contains clock jitter. The later revision 'B46W500 B0' has corrected the problem:



Item	Axis1
Servo amplifier identification information	MR-J4-20TM
Servo amplifier serial number	D5ZQ85001
Servo amplifier S/W No.	BCD-B46W500 A3
Option unit identification information	No Connection
Network module identification information	EtherCAT
Network module Serial number	A0273D86
Network module S/W number	1.10.01
Motor model	HG-KR23
Motor ID	0111FF230000
Motor serial number	G70545020
Encoder resolution	4194304
Accumulated power-on time [h]	43
Num. of inrush cur. sw. times [times]	5
LED display	d00

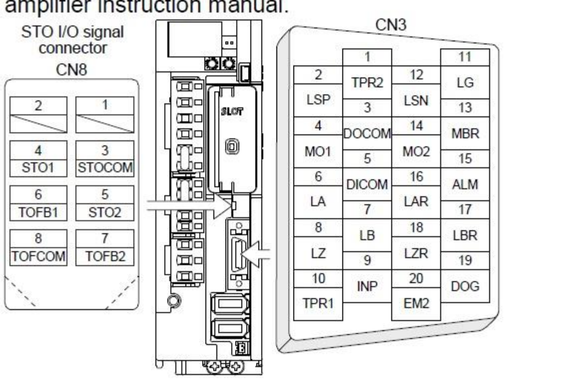


Item	Axis1
Servo amplifier identification information	MR-J4-40TM1
Servo amplifier serial number	F66U03005
Servo amplifier S/W No.	BCD-B46W500 B0
Option unit identification information	No Connection
Network module identification information	EtherCAT
Network module Serial number	A02745E6
Network module S/W number	1.10.01
Motor model	HG-KR43
Motor ID	0111FF430000
Motor serial number	H71509009
Encoder resolution	4194304
Accumulated power-on time [h]	240
Num. of inrush cur. sw. times [times]	4
LED display	d00

CN3 Connector

For test purposes there are three signals that must be enabled either internally using the MR Configurator2 PC based software or pins jumpered on the CN3 Connector. This consists of LSP (Limit Positive), LSN (Limit Negative), EM2 (Enable), and DICOM. DICOM must be connected to +24V, the other signals to 24 Volt Return or proper controlling switch.

The following shows MR-J4-10TM signals as a typical example. For other servo amplifiers, refer to each servo amplifier instruction manual.



Input device

Symbol	Device	Connector	Pin No.
EM2	Forced stop 2	CN3	20
STOCOM	Common terminal for input signals STO1/STO2	CN8	3
STO1	STO1 state input		4
STO2	STO2 state input		5

Output device

Symbol	Device	Connector	Pin No.
TOFCOM	Common terminal for monitor output signal in STO state	CN8	8
TOFB1	Monitor output signal in STO1 state		6
TOFB2	Monitor output signal in STO2 state		7

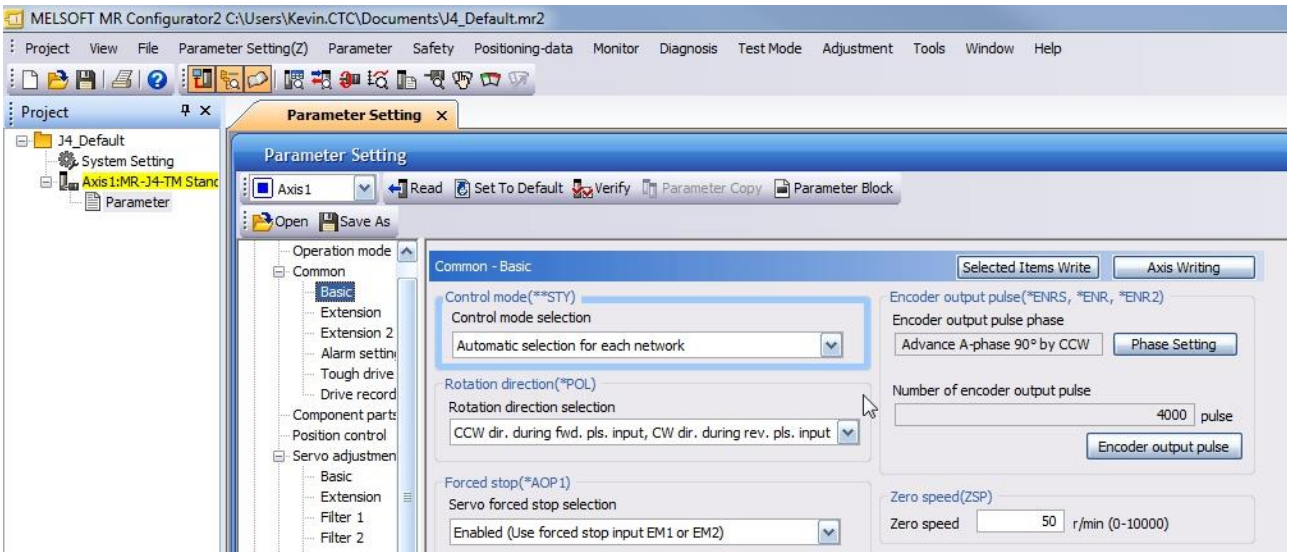
Power supply

Symbol	Device	Connector	Pin No.
DICOM	Digital I/F power supply input	CN3	5
DOCOM	Digital I/F common		3
SD	Shield		Plate

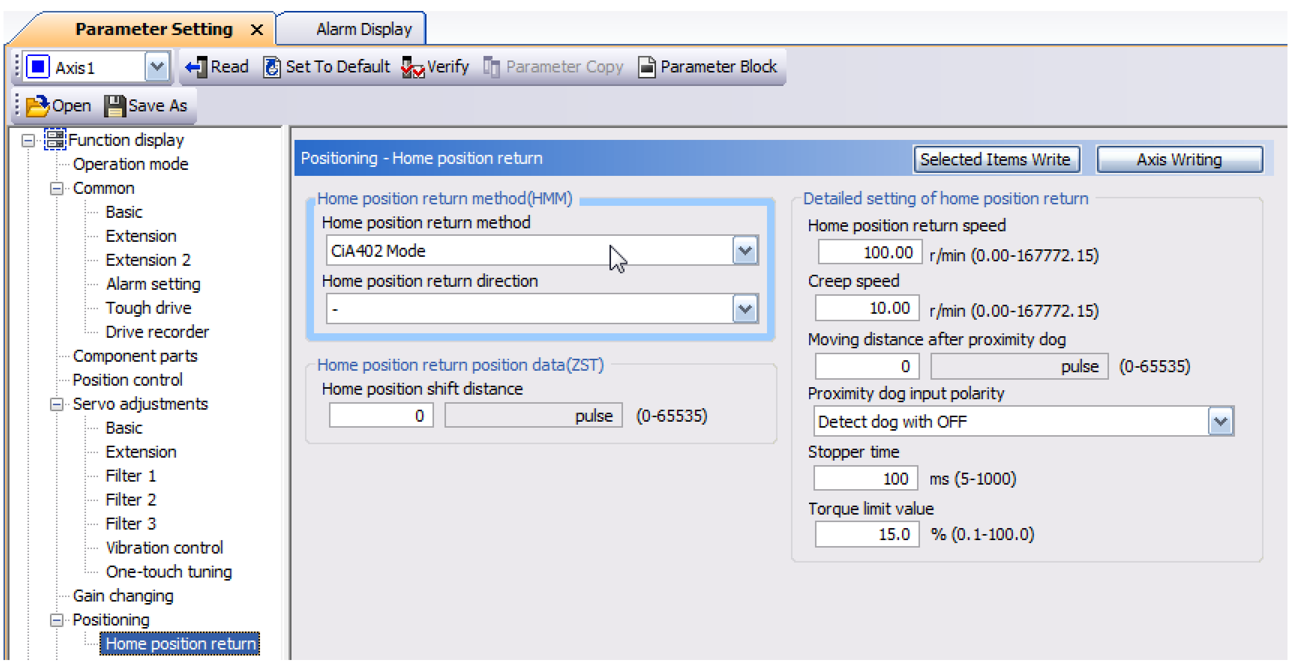
MR Configurator2 J4

Connect CN5 to a USB port on your PC and then connect to the drive using the Mitsubishi MR Configurator2 (MR2) software. For the purposes of this manual version V1.51D of MR2 was used.

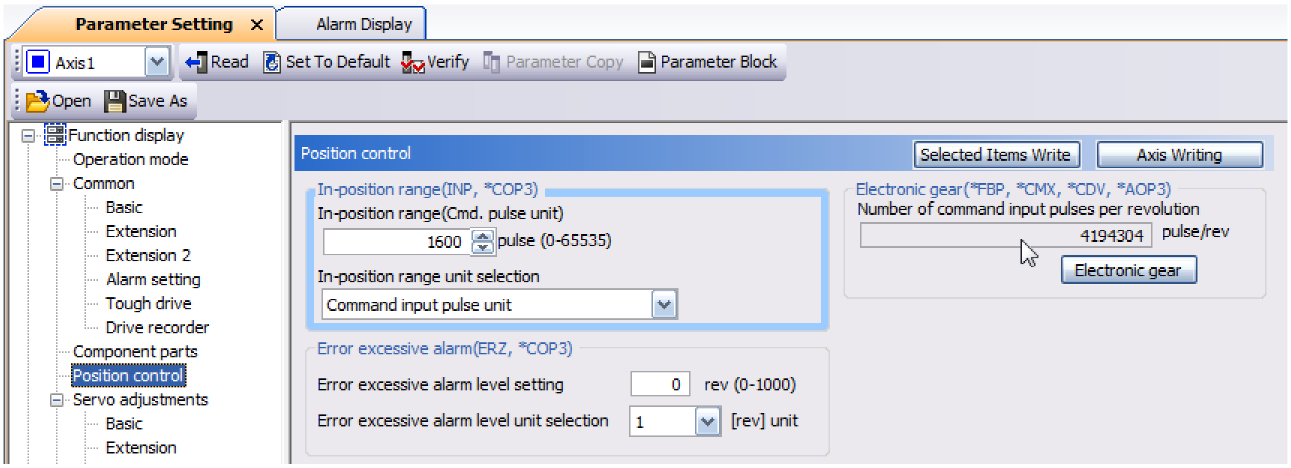
Make sure 'Automatic selection for each network' is selected.



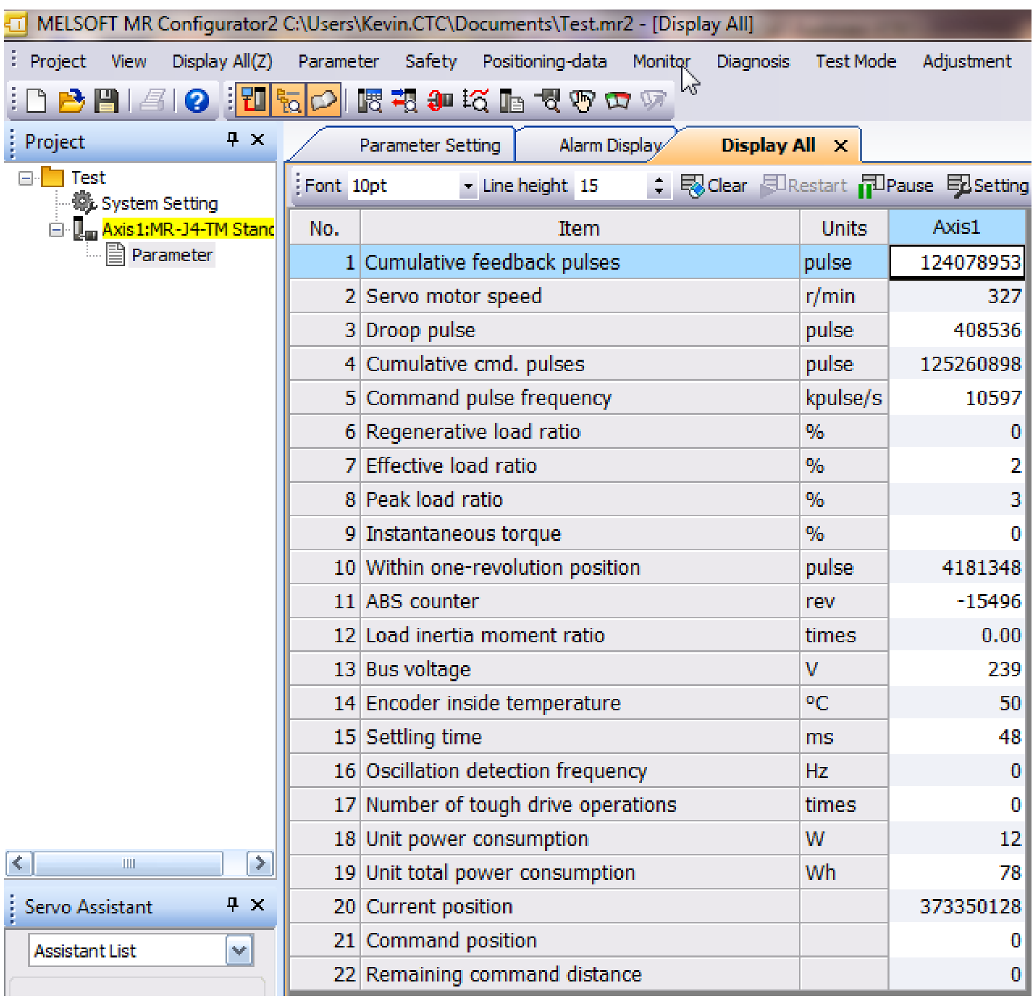
Homing mode by the drive using Cia402 standard is supported. You may also use CSP mode and zero the feedback when homing manually using MSB instructions. The drive should still be configured as below.



Check your ppr for MSB programming, this must be set in the properties section of the drive when using QuickBuilder. Below shows a ppr of 419430 pulse/rev. Also look at the "Error excessive alarm level setting". This typically needs to be increased to prevent 35.1 and 52.03 alarm errors. The more the drive lags the commanded position the larger the value would need to be (2 to 3 revs would not be unreasonable).

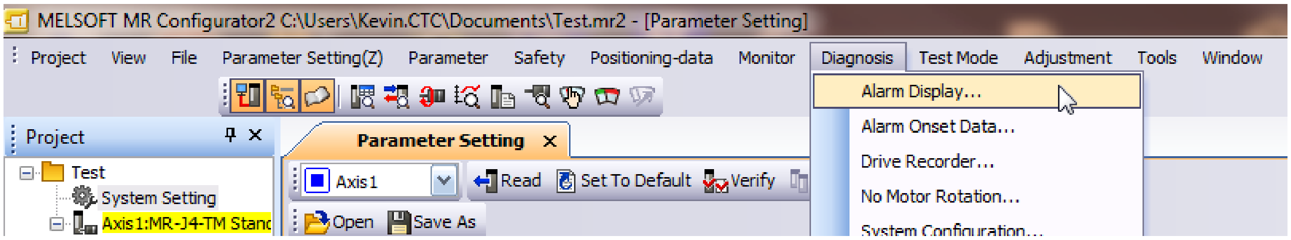


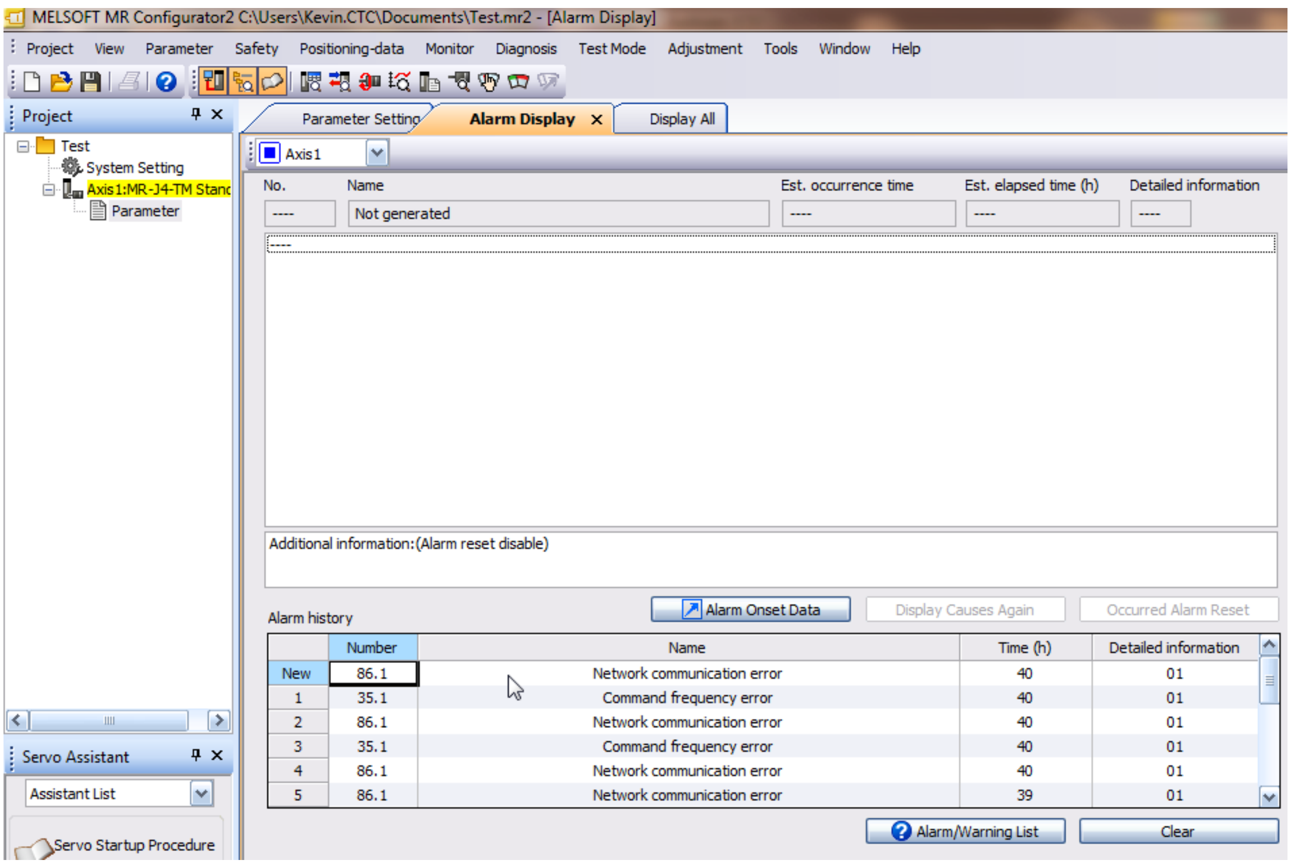
You may monitor operation using the Monitor->Display All selection:



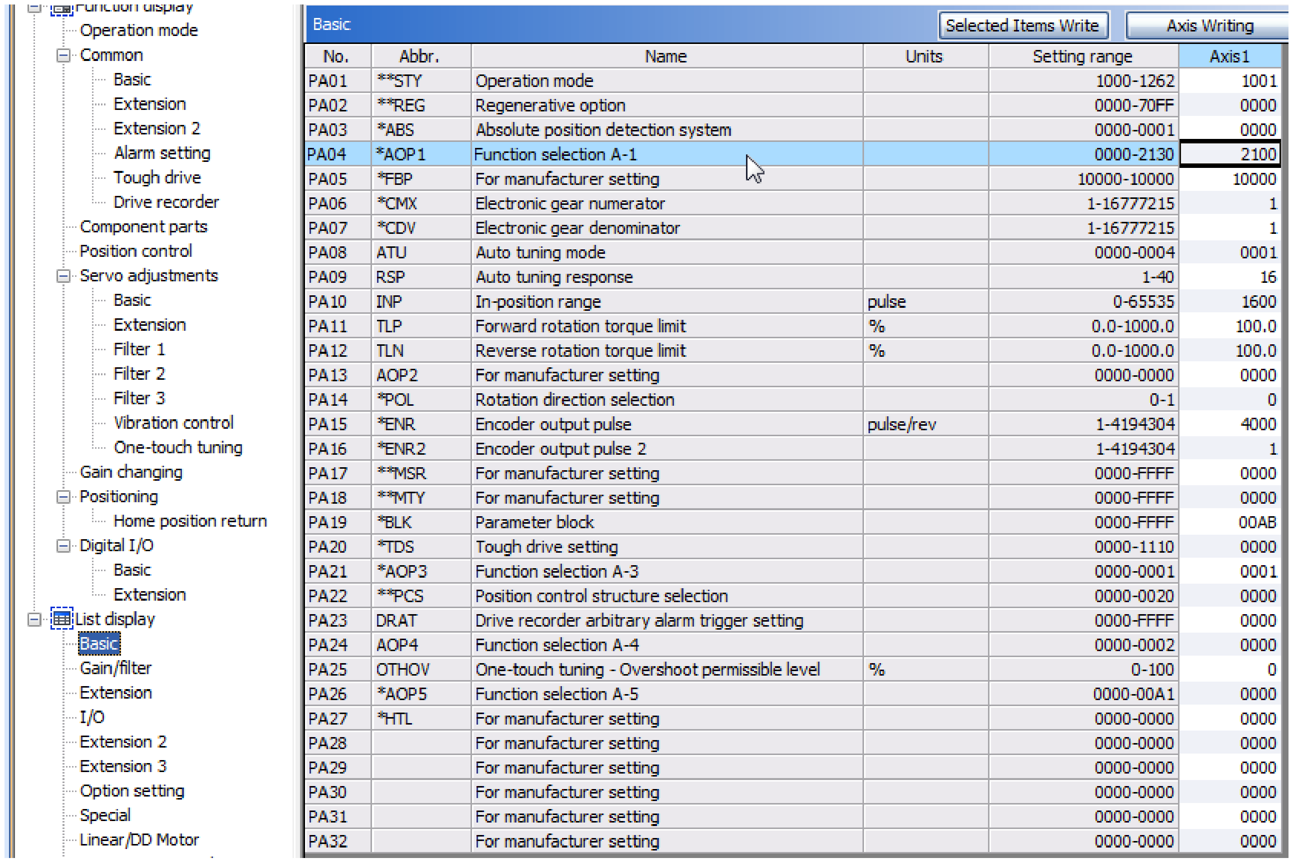
No.	Item	Units	Axis1
1	Cumulative feedback pulses	pulse	124078953
2	Servo motor speed	r/min	327
3	Droop pulse	pulse	408536
4	Cumulative cmd. pulses	pulse	125260898
5	Command pulse frequency	kpulse/s	10597
6	Regenerative load ratio	%	0
7	Effective load ratio	%	2
8	Peak load ratio	%	3
9	Instantaneous torque	%	0
10	Within one-revolution position	pulse	4181348
11	ABS counter	rev	-15496
12	Load inertia moment ratio	times	0.00
13	Bus voltage	V	239
14	Encoder inside temperature	°C	50
15	Settling time	ms	48
16	Oscillation detection frequency	Hz	0
17	Number of tough drive operations	times	0
18	Unit power consumption	W	12
19	Unit total power consumption	Wh	78
20	Current position		373350128
21	Command position		0
22	Remaining command distance		0

Alarm display is also useful. Typically an 86.1 will appear, this is normal as the EtherCAT network is restarted and is automatically cleared by the EtherCAT Master during initialization.

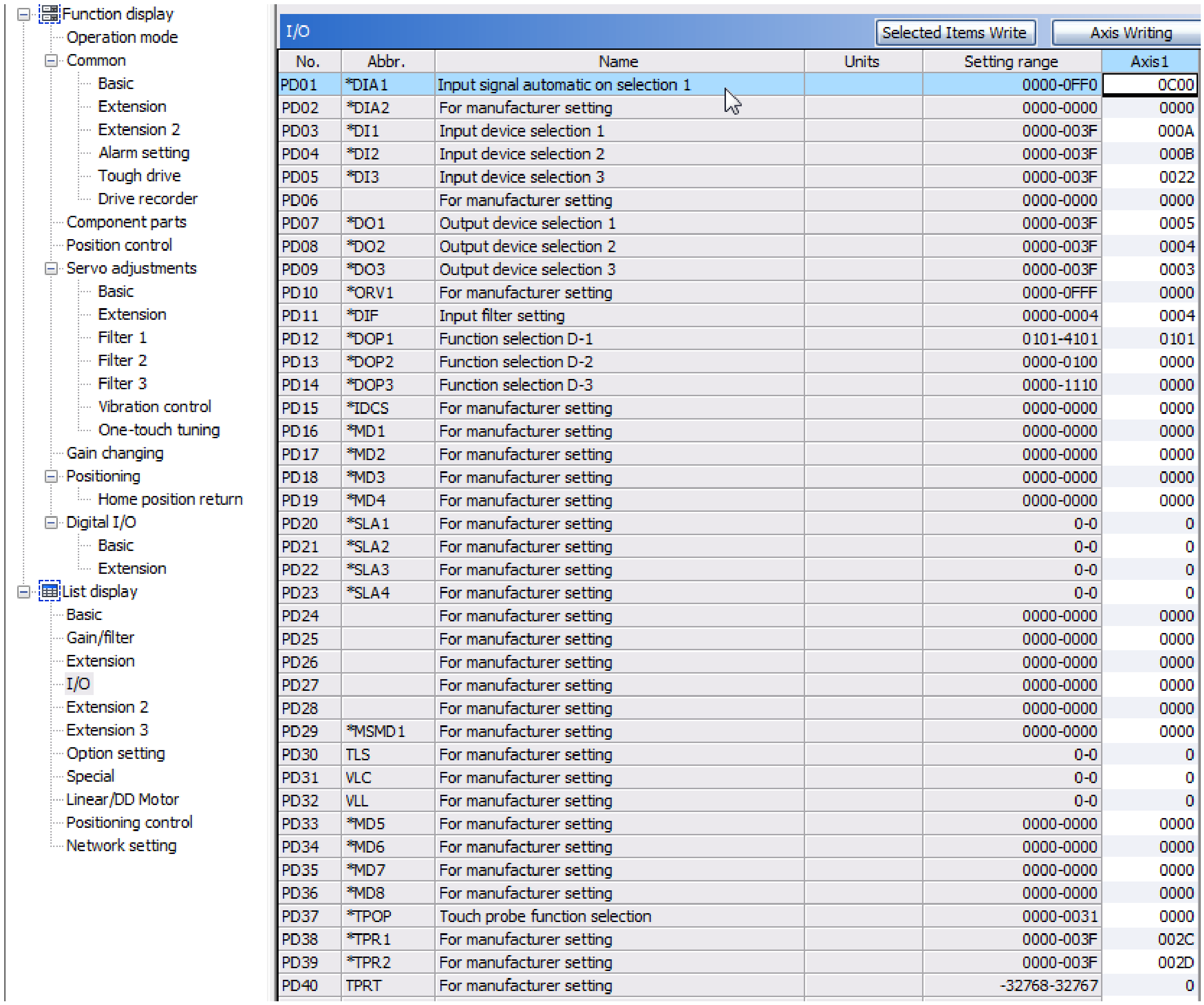




If you need to override the CN3 inputs, servo parameter PA04 can be set to 2100H to disable forced stop input and PD01 to 0C00 to disable the external limit switches.



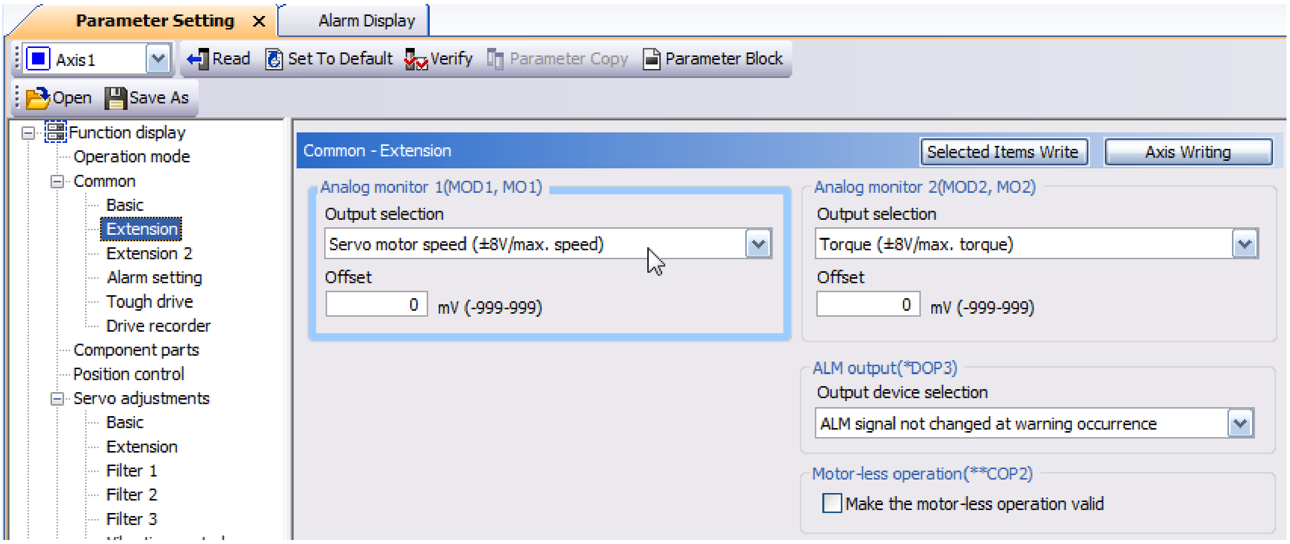
No.	Abbr.	Name	Units	Setting range	Axis1
PA01	**STY	Operation mode		1000-1262	1001
PA02	**REG	Regenerative option		0000-70FF	0000
PA03	*ABS	Absolute position detection system		0000-0001	0000
PA04	*AOP1	Function selection A-1		0000-2130	2100
PA05	*FBP	For manufacturer setting		10000-10000	10000
PA06	*CMX	Electronic gear numerator		1-16777215	1
PA07	*CDV	Electronic gear denominator		1-16777215	1
PA08	ATU	Auto tuning mode		0000-0004	0001
PA09	RSP	Auto tuning response		1-40	16
PA10	INP	In-position range	pulse	0-65535	1600
PA11	TLP	Forward rotation torque limit	%	0.0-1000.0	100.0
PA12	TLN	Reverse rotation torque limit	%	0.0-1000.0	100.0
PA13	AOP2	For manufacturer setting		0000-0000	0000
PA14	*POL	Rotation direction selection		0-1	0
PA15	*ENR	Encoder output pulse	pulse/rev	1-4194304	4000
PA16	*ENR2	Encoder output pulse 2		1-4194304	1
PA17	**MSR	For manufacturer setting		0000-FFFF	0000
PA18	**MTY	For manufacturer setting		0000-FFFF	0000
PA19	*BLK	Parameter block		0000-FFFF	00AB
PA20	*TDS	Tough drive setting		0000-1110	0000
PA21	*AOP3	Function selection A-3		0000-0001	0001
PA22	**PCS	Position control structure selection		0000-0020	0000
PA23	DRAT	Drive recorder arbitrary alarm trigger setting		0000-FFFF	0000
PA24	AOP4	Function selection A-4		0000-0002	0000
PA25	OTHOV	One-touch tuning - Overshoot permissible level	%	0-100	0
PA26	*AOP5	Function selection A-5		0000-00A1	0000
PA27	*HTL	For manufacturer setting		0000-0000	0000
PA28		For manufacturer setting		0000-0000	0000
PA29		For manufacturer setting		0000-0000	0000
PA30		For manufacturer setting		0000-0000	0000
PA31		For manufacturer setting		0000-0000	0000
PA32		For manufacturer setting		0000-0000	0000

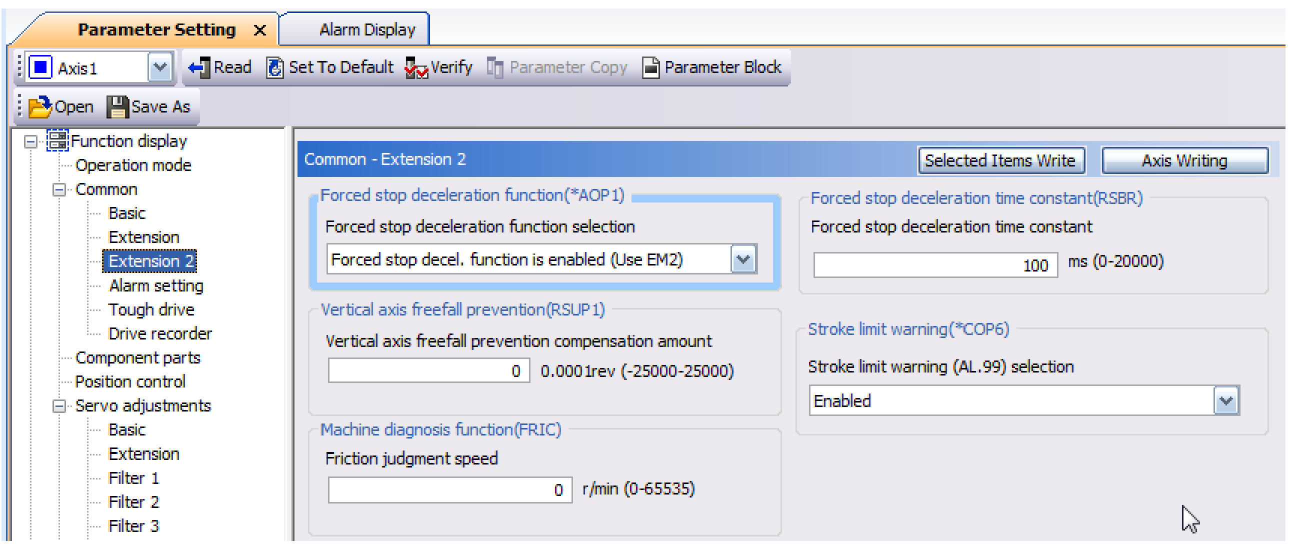


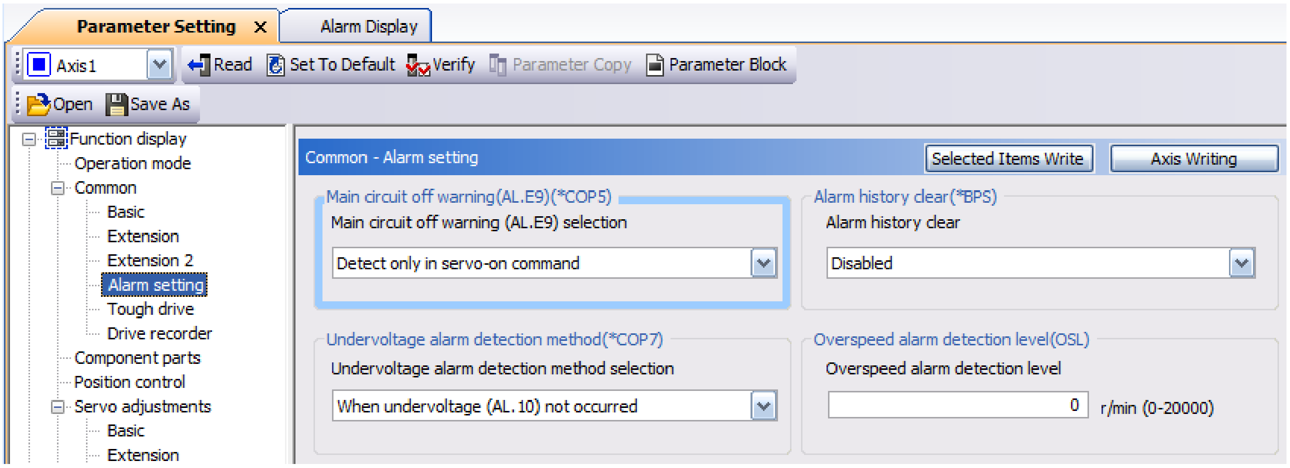
No.	Abbr.	Name	Units	Setting range	Axis1
PD01	*DIA1	Input signal automatic on selection 1		0000-0FF0	0C00
PD02	*DIA2	For manufacturer setting		0000-0000	0000
PD03	*DI1	Input device selection 1		0000-003F	000A
PD04	*DI2	Input device selection 2		0000-003F	000B
PD05	*DI3	Input device selection 3		0000-003F	0022
PD06		For manufacturer setting		0000-0000	0000
PD07	*DO1	Output device selection 1		0000-003F	0005
PD08	*DO2	Output device selection 2		0000-003F	0004
PD09	*DO3	Output device selection 3		0000-003F	0003
PD10	*ORV1	For manufacturer setting		0000-0FFF	0000
PD11	*DIF	Input filter setting		0000-0004	0004
PD12	*DOP1	Function selection D-1		0101-4101	0101
PD13	*DOP2	Function selection D-2		0000-0100	0000
PD14	*DOP3	Function selection D-3		0000-1110	0000
PD15	*IDCS	For manufacturer setting		0000-0000	0000
PD16	*MD1	For manufacturer setting		0000-0000	0000
PD17	*MD2	For manufacturer setting		0000-0000	0000
PD18	*MD3	For manufacturer setting		0000-0000	0000
PD19	*MD4	For manufacturer setting		0000-0000	0000
PD20	*SLA1	For manufacturer setting		0-0	0
PD21	*SLA2	For manufacturer setting		0-0	0
PD22	*SLA3	For manufacturer setting		0-0	0
PD23	*SLA4	For manufacturer setting		0-0	0
PD24		For manufacturer setting		0000-0000	0000
PD25		For manufacturer setting		0000-0000	0000
PD26		For manufacturer setting		0000-0000	0000
PD27		For manufacturer setting		0000-0000	0000
PD28		For manufacturer setting		0000-0000	0000
PD29	*MSMD1	For manufacturer setting		0000-0000	0000
PD30	TLS	For manufacturer setting		0-0	0
PD31	VLC	For manufacturer setting		0-0	0
PD32	VLL	For manufacturer setting		0-0	0
PD33	*MD5	For manufacturer setting		0000-0000	0000
PD34	*MD6	For manufacturer setting		0000-0000	0000
PD35	*MD7	For manufacturer setting		0000-0000	0000
PD36	*MD8	For manufacturer setting		0000-0000	0000
PD37	*TPOP	Touch probe function selection		0000-0031	0000
PD38	*TPR1	For manufacturer setting		0000-003F	002C
PD39	*TPR2	For manufacturer setting		0000-003F	002D
PD40	TPRT	For manufacturer setting		-32768-32767	0

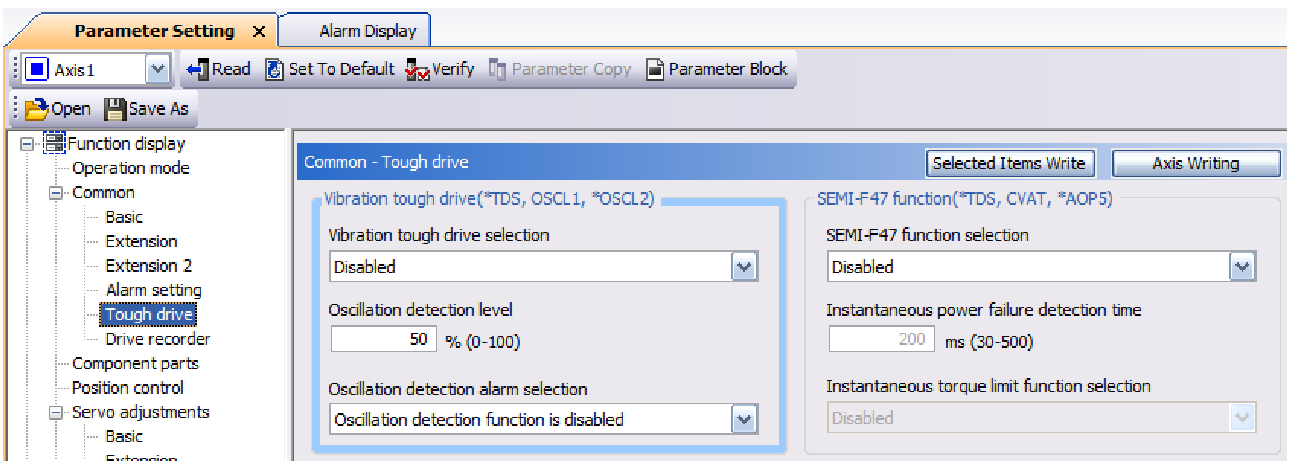
MR Configurator2 Miscellaneous

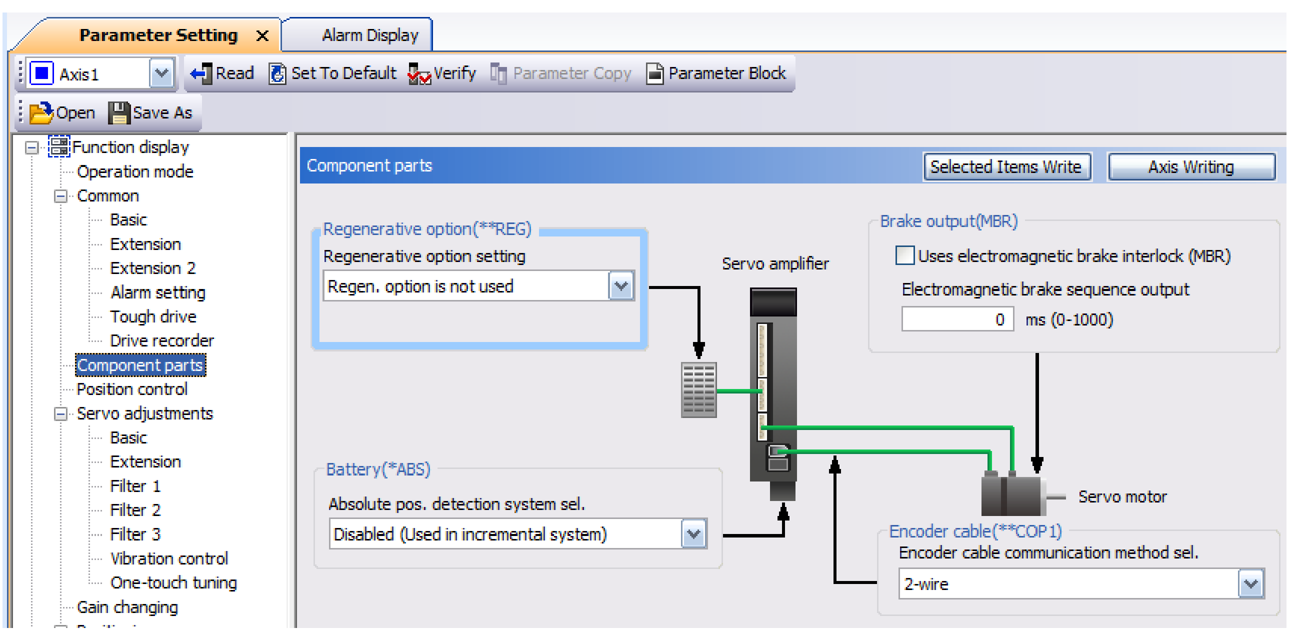
Other screens of interest and their settings during testing:

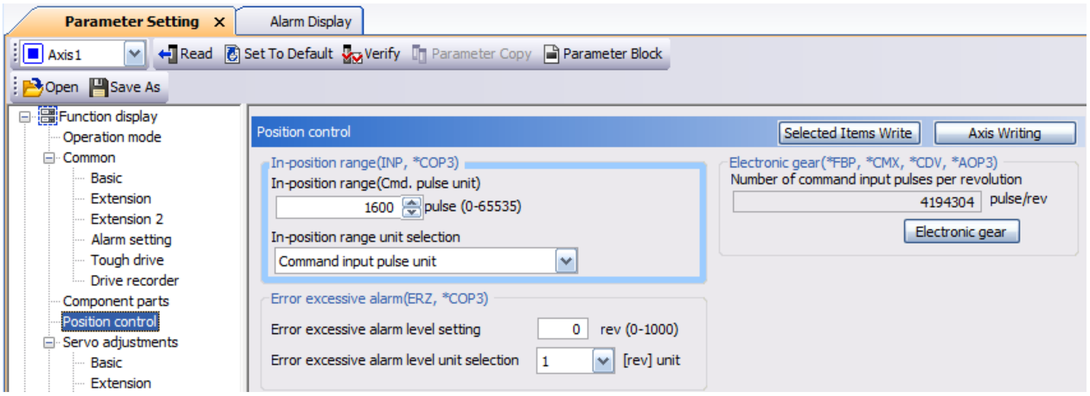




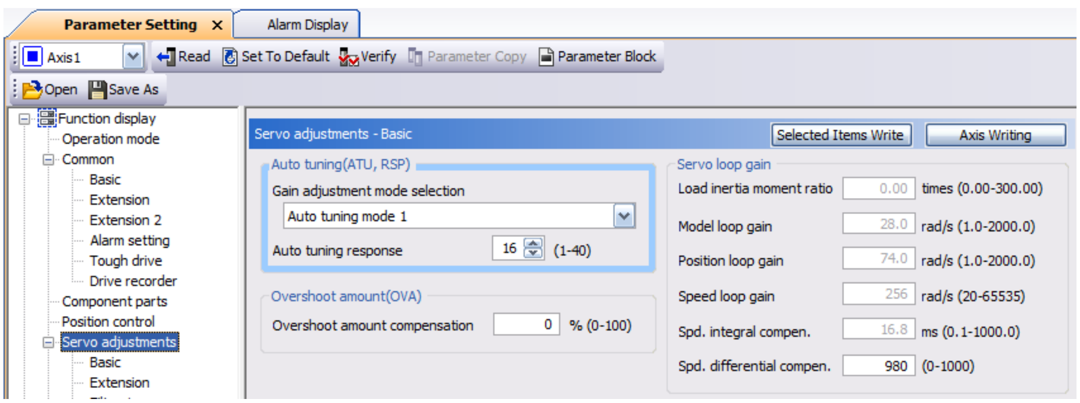

Parameter Setting
Alarm Display
Axis1
Read
Set To Default
Verify
Parameter Copy
Parameter Block
Open
Save As
Function display
Operation mode
Common
Basic
Extension
Extension 2
Alarm setting
Tough drive
Drive recorder
Component parts
Position control
Servo adjustments
Basic
Extension
Common - Alarm setting
Selected Items Write
Axis Writing
Main circuit off warning(AL.E9)(*COP5)
Main circuit off warning (AL.E9) selection
Detect only in servo-on command
Alarm history clear(*BPS)
Alarm history clear
Disabled
Undervoltage alarm detection method(*COP7)
Undervoltage alarm detection method selection
When undervoltage (AL.10) not occurred
Overspeed alarm detection level(OSL)
Overspeed alarm detection level
0
r/min (0-20000)


Parameter Setting
Alarm Display
Axis1
Read
Set To Default
Verify
Parameter Copy
Parameter Block
Open
Save As
Function display
Operation mode
Common
Basic
Extension
Extension 2
Alarm setting
Tough drive
Drive recorder
Component parts
Position control
Servo adjustments
Basic
Common - Tough drive
Selected Items Write
Axis Writing
Vibration tough drive(*TDS, OSCL1, *OSCL2)
Vibration tough drive selection
Disabled
Oscillation detection level
50
% (0-100)
Oscillation detection alarm selection
Oscillation detection function is disabled
SEMI-F47 function(*TDS, CVAT, *AOP5)
SEMI-F47 function selection
Disabled
Instantaneous power failure detection time
200
ms (30-500)
Instantaneous torque limit function selection
Disabled

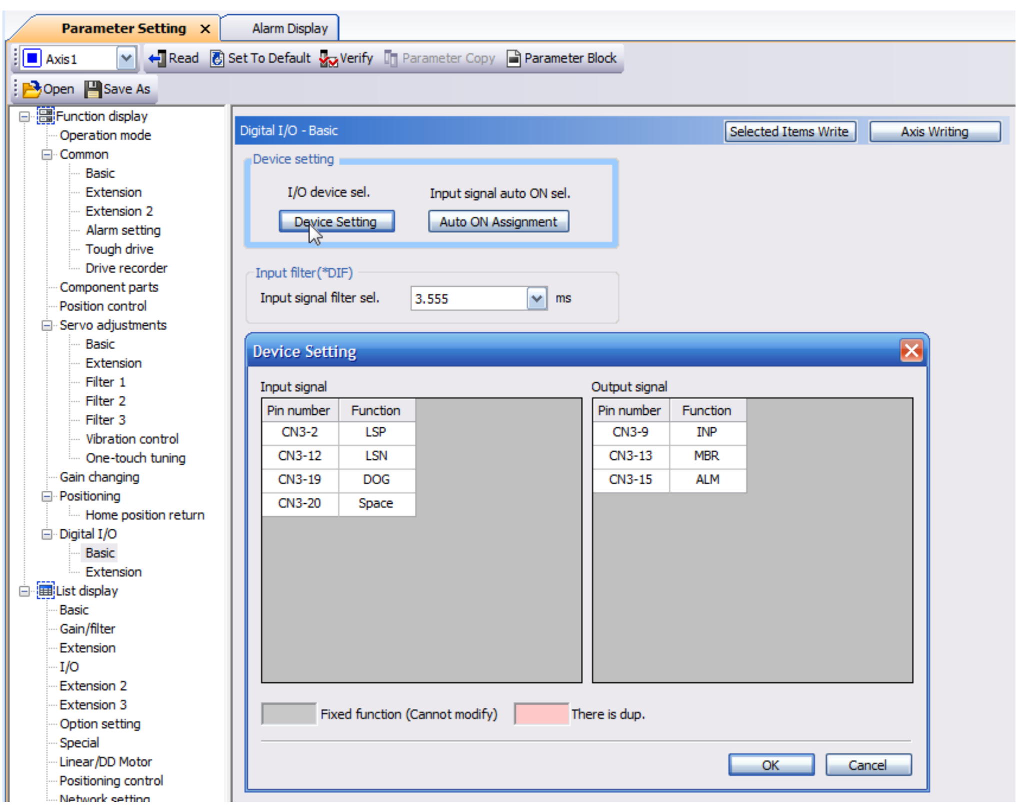

Parameter Setting
Alarm Display
Axis1
Read
Set To Default
Verify
Parameter Copy
Parameter Block
Open
Save As
Function display
Operation mode
Common
Basic
Extension
Extension 2
Alarm setting
Tough drive
Drive recorder
Component parts
Position control
Servo adjustments
Basic
Extension
Filter 1
Filter 2
Filter 3
Vibration control
One-touch tuning
Gain changing
Component parts
Selected Items Write
Axis Writing
Regenerative option(**REG)
Regenerative option setting
Regen. option is not used
Servo amplifier
Brake output(MBR)
Uses electromagnetic brake interlock (MBR)
Electromagnetic brake sequence output
0
ms (0-1000)
Battery(*ABS)
Absolute pos. detection system sel.
Disabled (Used in incremental system)
Servo motor
Encoder cable(**COP1)
Encoder cable communication method sel.
2-wire



Parameter Setting
Alarm Display
Axis1
Read
Set To Default
Verify
Parameter Copy
Parameter Block
Open
Save As
Function display
Operation mode
Common
Basic
Extension
Extension 2
Alarm setting
Tough drive
Drive recorder
Component parts
Position control
Servo adjustments
Basic
Extension
Position control
Selected Items Write
Axis Writing
In-position range(INP, *COP3)
In-position range(Cmd. pulse unit)
1600 pulse (0-65535)
In-position range unit selection
Command input pulse unit
Error excessive alarm(ERZ, *COP3)
Error excessive alarm level setting
0 rev (0-1000)
Error excessive alarm level unit selection
1 [rev] unit
Electronic gear(*FBP, *CMX, *CDV, *AOP3)
Number of command input pulses per revolution
4194304 pulse/rev
Electronic gear



Parameter Setting
Alarm Display
Axis1
Read
Set To Default
Verify
Parameter Copy
Parameter Block
Open
Save As
Function display
Operation mode
Common
Basic
Extension
Extension 2
Alarm setting
Tough drive
Drive recorder
Component parts
Position control
Servo adjustments
Basic
Extension
Servo adjustments - Basic
Selected Items Write
Axis Writing
Auto tuning(ATU, RSP)
Gain adjustment mode selection
Auto tuning mode 1
Auto tuning response
16 (1-40)
Overshoot amount(OVA)
Overshoot amount compensation
0 % (0-100)
Servo loop gain
Load inertia moment ratio
0.00 times (0.00-300.00)
Model loop gain
28.0 rad/s (1.0-2000.0)
Position loop gain
74.0 rad/s (1.0-2000.0)
Speed loop gain
256 rad/s (20-65535)
Spd. integral compen.
16.8 ms (0.1-1000.0)
Spd. differential compen.
980 (0-1000)

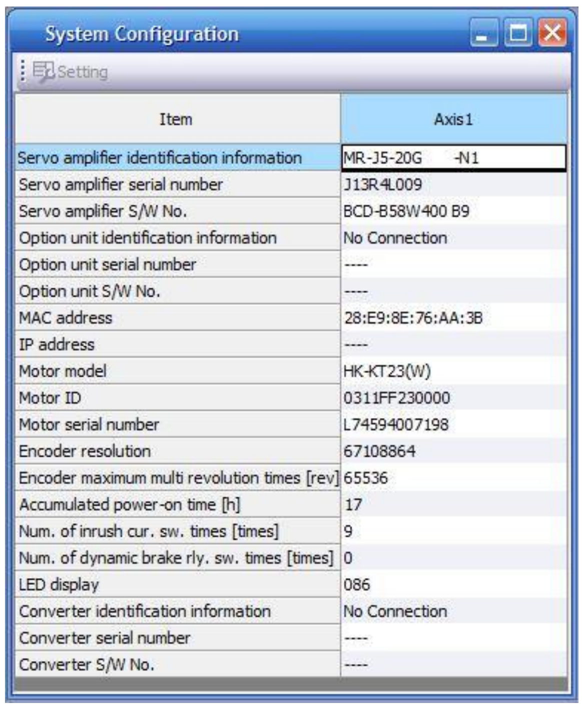


Parameter Setting
Alarm Display
Axis1
Read
Set To Default
Verify
Parameter Copy
Parameter Block
Open
Save As
Function display
Operation mode
Common
Basic
Extension
Extension 2
Alarm setting
Tough drive
Drive recorder
Component parts
Position control
Servo adjustments
Basic
Extension
Filter 1
Filter 2
Filter 3
Vibration control
One-touch tuning
Gain changing
Positioning
Home position return
Digital I/O
Basic
Extension
List display
Basic
Gain/filter
Extension
I/O
Extension 2
Extension 3
Option setting
Special
Linear/DD Motor
Positioning control
Network setting
Digital I/O - Basic
Selected Items Write
Axis Writing
Device setting
I/O device sel.
Input signal auto ON sel.
Device Setting
Auto ON Assignment
Input filter(*DIF)
Input signal filter sel.
3.555 ms
Device Setting
Input signal
Pin number Function
CN3-2 LSP
CN3-12 LSN
CN3-19 DOG
CN3-20 Space
Output signal
Pin number Function
CN3-9 INP
CN3-13 MBR
CN3-15 ALM
Fixed function (Cannot modify)
There is dup.
OK
Cancel

MR Configurator2 V1.125F J5

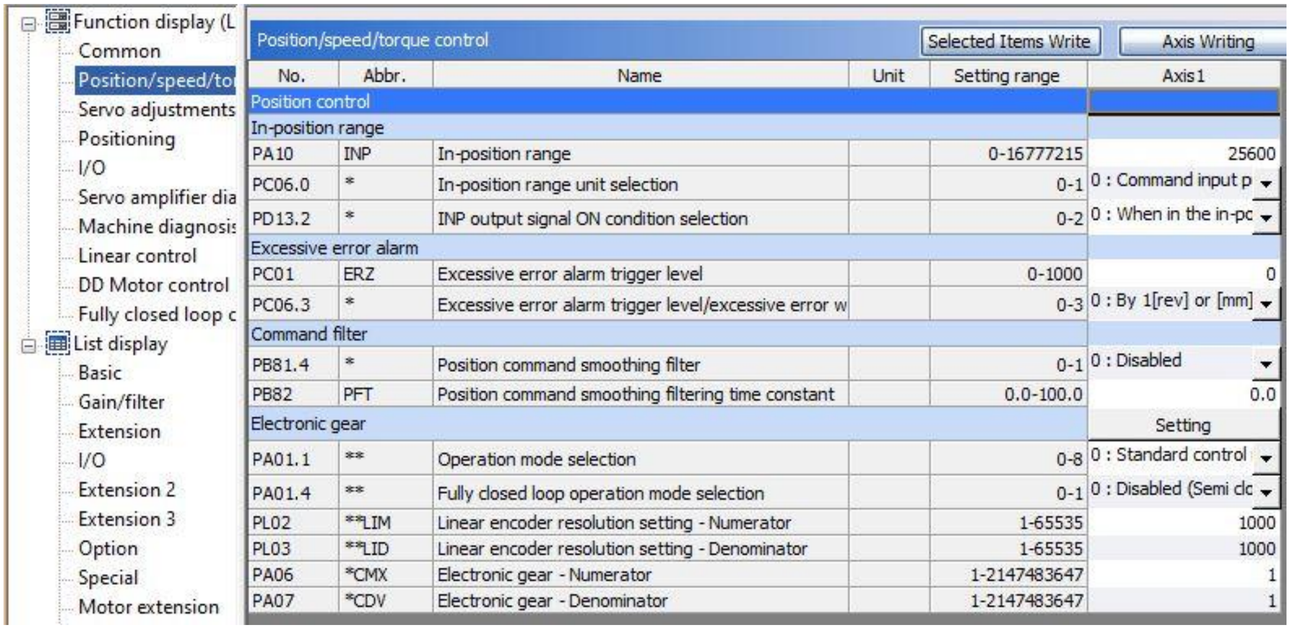
Connect CN5 to a USB port on your PC and then connect to the drive using the Mitsubishi MR Configurator2.

Some Parameters of interest:
PT01.1 = 0 (r/min, required for QuickBuilder 'vel' parameter to report correct speed)
PT01.2 = 3 (pulse)
PT45 = 37 (homing mode, current position)
PT15 = 0
PT17 = 0
PC03.2 = 0



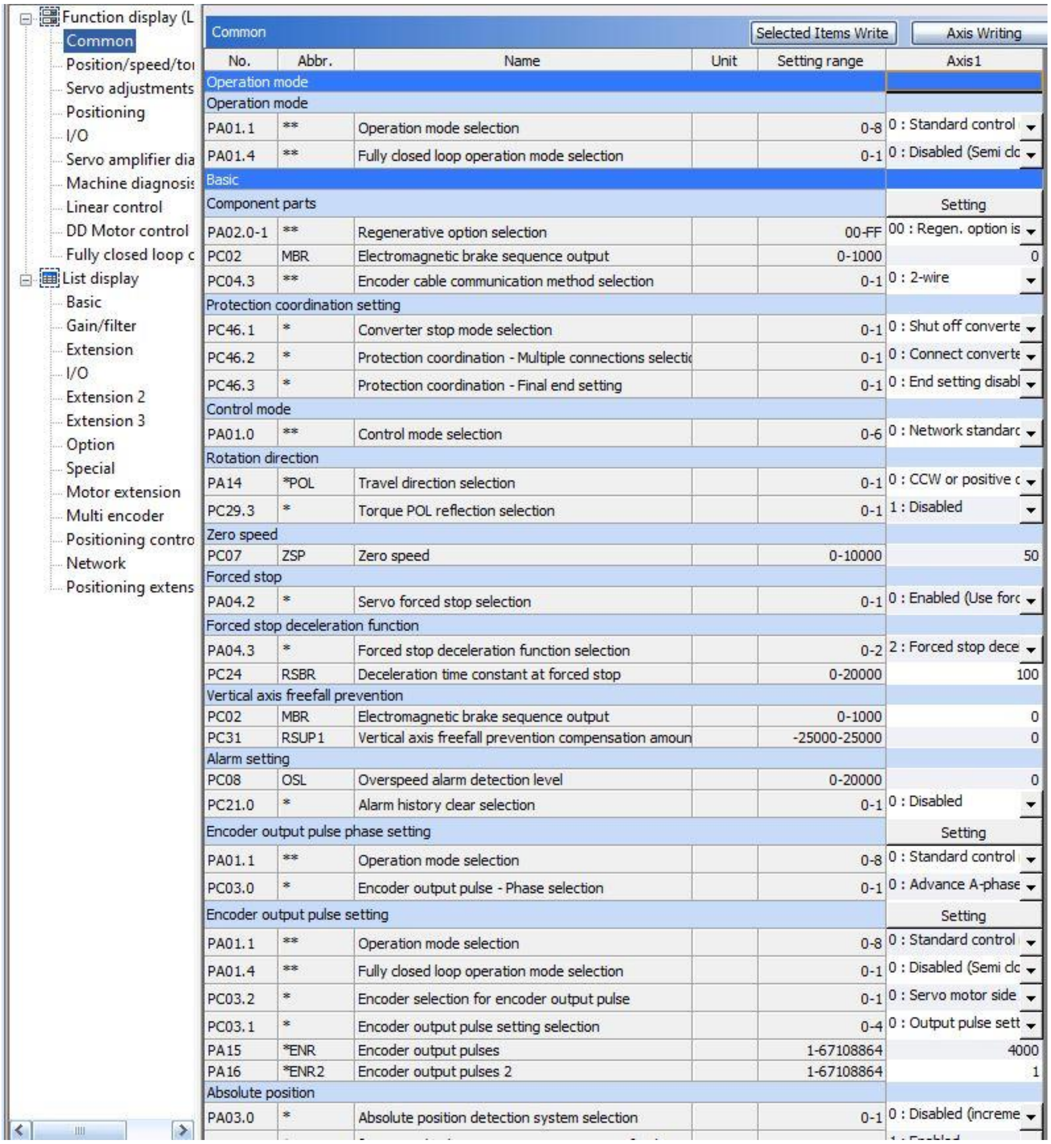
Item	Axis1
Servo amplifier identification information	MR-J5-20G -N1
Servo amplifier serial number	J13R4L009
Servo amplifier S/W No.	BCD-B58W400 B9
Option unit identification information	No Connection
Option unit serial number	----
Option unit S/W No.	----
MAC address	28:E9:8E:76:AA:3B
IP address	----
Motor model	HK-KT23(W)
Motor ID	0311FF230000
Motor serial number	L74594007198
Encoder resolution	67108864
Encoder maximum multi revolution times [rev]	65536
Accumulated power-on time [h]	17
Num. of inrush cur. sw. times [times]	9
Num. of dynamic brake rly. sw. times [times]	0
LED display	086
Converter identification information	No Connection
Converter serial number	----
Converter S/W No.	----

Common Parameters:



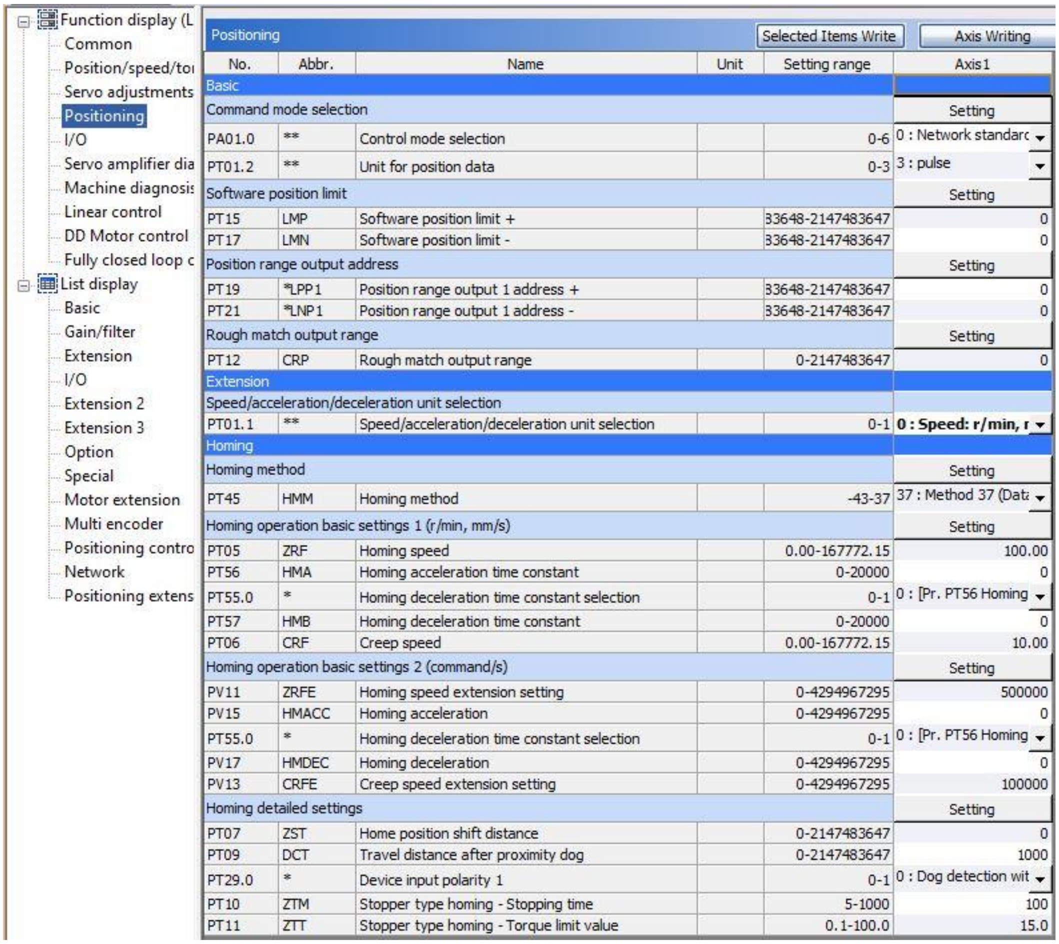
No.	Abbr.	Name	Unit	Setting range	Axis1
Position control					
In-position range					
PA10	INP	In-position range		0-16777215	25600
PC06.0	*	In-position range unit selection		0-1	0 : Command input p
PD13.2	*	INP output signal ON condition selection		0-2	0 : When in the in-pc
Excessive error alarm					
PC01	ERZ	Excessive error alarm trigger level		0-1000	0
PC06.3	*	Excessive error alarm trigger level/excessive error w		0-3	0 : By 1[rev] or [mm]
Command filter					
PB81.4	*	Position command smoothing filter		0-1	0 : Disabled
PB82	PFT	Position command smoothing filtering time constant		0.0-100.0	0.0
Electronic gear					Setting
PA01.1	**	Operation mode selection		0-8	0 : Standard control
PA01.4	**	Fully closed loop operation mode selection		0-1	0 : Disabled (Semi clc
PL02	**LIM	Linear encoder resolution setting - Numerator		1-65535	1000
PL03	**LID	Linear encoder resolution setting - Denominator		1-65535	1000
PA06	*CMX	Electronic gear - Numerator		1-2147483647	1
PA07	*CDV	Electronic gear - Denominator		1-2147483647	1

Position/Speed/Torque Parameters:



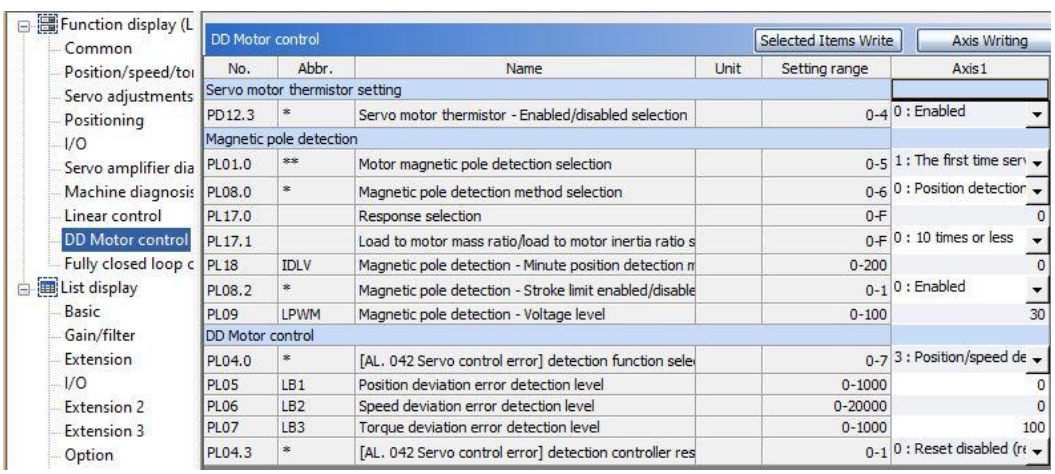
No.	Abbr.	Name	Unit	Setting range	Axis1
Operation mode					
Operation mode					
PA01.1	**	Operation mode selection		0-8	0 : Standard control
PA01.4	**	Fully closed loop operation mode selection		0-1	0 : Disabled (Semi clc
Basic					
Component parts					Setting
PA02.0-1	**	Regenerative option selection		00-FF	00 : Regen. option is
PC02	MBR	Electromagnetic brake sequence output		0-1000	0
PC04.3	**	Encoder cable communication method selection		0-1	0 : 2-wire
Protection coordination setting					
PC46.1	*	Converter stop mode selection		0-1	0 : Shut off converte
PC46.2	*	Protection coordination - Multiple connections selectic		0-1	0 : Connect converte
PC46.3	*	Protection coordination - Final end setting		0-1	0 : End setting disabl
Control mode					
PA01.0	**	Control mode selection		0-6	0 : Network standarc
Rotation direction					
PA14	*POL	Travel direction selection		0-1	0 : CCW or positive c
PC29.3	*	Torque POL reflection selection		0-1	1 : Disabled
Zero speed					
PC07	ZSP	Zero speed		0-10000	50
Forced stop					
PA04.2	*	Servo forced stop selection		0-1	0 : Enabled (Use forc
Forced stop deceleration function					
PA04.3	*	Forced stop deceleration function selection		0-2	2 : Forced stop dece
PC24	RSBR	Deceleration time constant at forced stop		0-20000	100
Vertical axis freefall prevention					
PC02	MBR	Electromagnetic brake sequence output		0-1000	0
PC31	RSUP1	Vertical axis freefall prevention compensation amoun		-25000-25000	0
Alarm setting					
PC08	OSL	Overspeed alarm detection level		0-20000	0
PC21.0	*	Alarm history clear selection		0-1	0 : Disabled
Encoder output pulse phase setting					Setting
PA01.1	**	Operation mode selection		0-8	0 : Standard control
PC03.0	*	Encoder output pulse - Phase selection		0-1	0 : Advance A-phase
Encoder output pulse setting					Setting
PA01.1	**	Operation mode selection		0-8	0 : Standard control
PA01.4	**	Fully closed loop operation mode selection		0-1	0 : Disabled (Semi clc
PC03.2	*	Encoder selection for encoder output pulse		0-1	0 : Servo motor side
PC03.1	*	Encoder output pulse setting selection		0-4	0 : Output pulse sett
PA15	*ENR	Encoder output pulses		1-67108864	4000
PA16	*ENR2	Encoder output pulses 2		1-67108864	1
Absolute position					
PA03.0	*	Absolute position detection system selection		0-1	0 : Disabled (increme

Positioning Parameters:



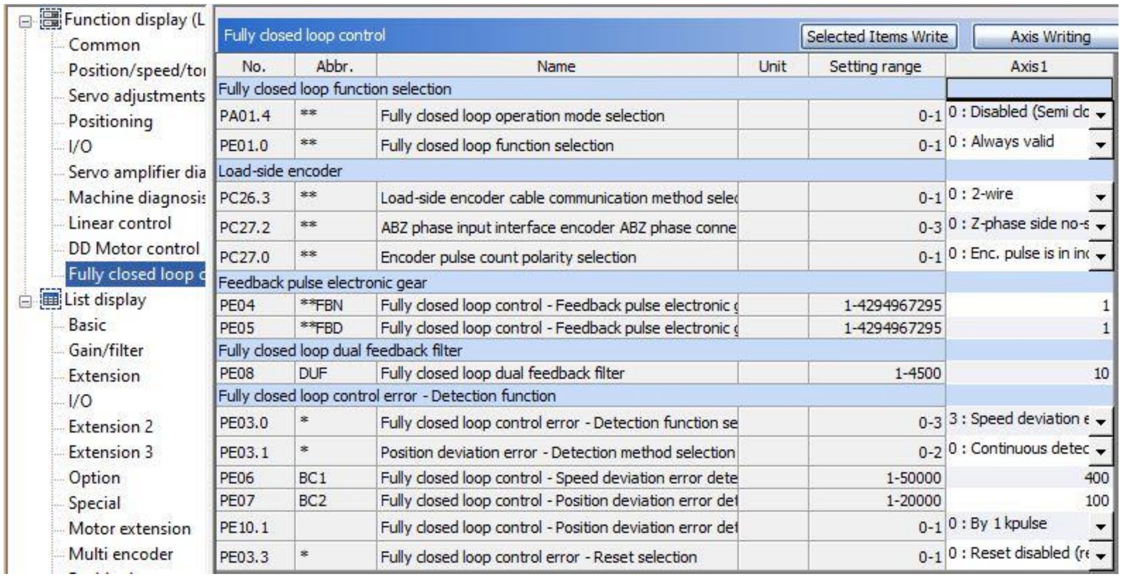
No.	Abbr.	Name	Unit	Setting range	Axis1
Basic					
Command mode selection					Setting
PA01.0	**	Control mode selection		0-6	0 : Network standarc
PT01.2	**	Unit for position data		0-3	3 : pulse
Software position limit					Setting
PT15	LMP	Software position limit +		33648-2147483647	0
PT17	LMN	Software position limit -		33648-2147483647	0
Position range output address					Setting
PT19	*LPP1	Position range output 1 address +		33648-2147483647	0
PT21	*LNP1	Position range output 1 address -		33648-2147483647	0
Rough match output range					Setting
PT12	CRP	Rough match output range		0-2147483647	0
Extension					
Speed/acceleration/deceleration unit selection					
PT01.1	**	Speed/acceleration/deceleration unit selection		0-1	**0 : Speed: r/min, r**
Homing					
Homing method					Setting
PT45	HMM	Homing method		-43-37	37 : Method 37 (Data
Homing operation basic settings 1 (r/min, mm/s)					Setting
PT05	ZRF	Homing speed		0.00-167772.15	100.00
PT56	HMA	Homing acceleration time constant		0-20000	0
PT55.0	*	Homing deceleration time constant selection		0-1	0 : [Pr. PT56 Homing
PT57	HMB	Homing deceleration time constant		0-20000	0
PT06	CRF	Creep speed		0.00-167772.15	10.00
Homing operation basic settings 2 (command/s)					Setting
PV11	ZRFE	Homing speed extension setting		0-4294967295	500000
PV15	HMACC	Homing acceleration		0-4294967295	0
PT55.0	*	Homing deceleration time constant selection		0-1	0 : [Pr. PT56 Homing
PV17	HMDEC	Homing deceleration		0-4294967295	0
PV13	CRFE	Creep speed extension setting		0-4294967295	100000
Homing detailed settings					Setting
PT07	ZST	Home position shift distance		0-2147483647	0
PT09	DCT	Travel distance after proximity dog		0-2147483647	1000
PT29.0	*	Device input polarity 1		0-1	0 : Dog detection wit
PT10	ZTM	Stopper type homing - Stopping time		5-1000	100
PT11	ZTT	Stopper type homing - Torque limit value		0.1-100.0	15.0

DD Motor Control Parameters:



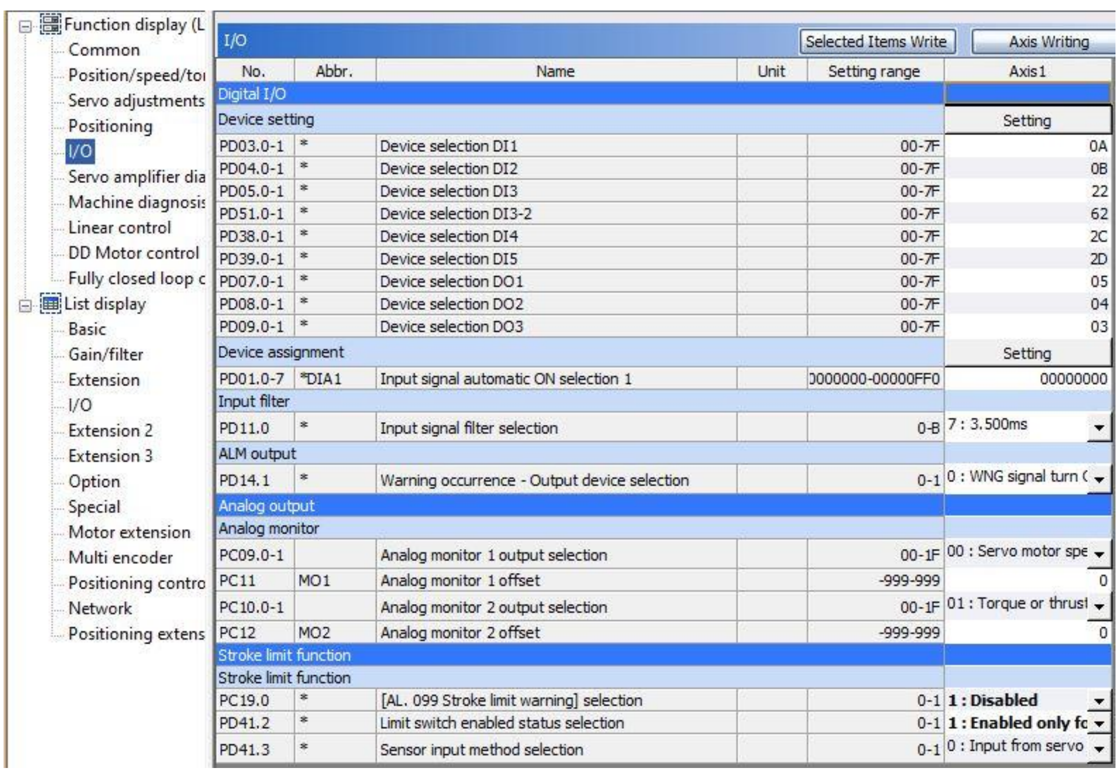
No.	Abbr.	Name	Unit	Setting range	Axis1
Servo motor thermistor setting					
PD12.3	*	Servo motor thermistor - Enabled/disabled selection		0-4	0 : Enabled
Magnetic pole detection					
PL01.0	**	Motor magnetic pole detection selection		0-5	1 : The first time serv
PL08.0	*	Magnetic pole detection method selection		0-6	0 : Position detection
PL17.0		Response selection		0-F	0
PL17.1		Load to motor mass ratio/load to motor inertia ratio s		0-F	0 : 10 times or less
PL18	IDLV	Magnetic pole detection - Minute position detection m		0-200	0
PL08.2	*	Magnetic pole detection - Stroke limit enabled/disable		0-1	0 : Enabled
PL09	LPWM	Magnetic pole detection - Voltage level		0-100	30
DD Motor control					
PL04.0	*	[AL. 042 Servo control error] detection function sele		0-7	3 : Position/speed de
PL05	LB1	Position deviation error detection level		0-1000	0
PL06	LB2	Speed deviation error detection level		0-20000	0
PL07	LB3	Torque deviation error detection level		0-1000	100
PL04.3	*	[AL. 042 Servo control error] detection controller res		0-1	0 : Reset disabled (re

Fully Closed Loop Parameters:



No.	Abbr.	Name	Unit	Setting range	Axis1
Fully closed loop function selection					
PA01.4	**	Fully closed loop operation mode selection		0-1	0 : Disabled (Semi clc
PE01.0	**	Fully closed loop function selection		0-1	0 : Always valid
Load-side encoder					
PC26.3	**	Load-side encoder cable communication method selec		0-1	0 : 2-wire
PC27.2	**	ABZ phase input interface encoder ABZ phase conne		0-3	0 : Z-phase side no-s
PC27.0	**	Encoder pulse count polarity selection		0-1	0 : Enc. pulse is in inc
Feedback pulse electronic gear					
PE04	**FBN	Fully closed loop control - Feedback pulse electronic g		1-4294967295	1
PE05	**FBD	Fully closed loop control - Feedback pulse electronic g		1-4294967295	1
Fully closed loop dual feedback filter					
PE08	DUF	Fully closed loop dual feedback filter		1-4500	10
Fully closed loop control error - Detection function					
PE03.0	*	Fully closed loop control error - Detection function se		0-3	3 : Speed deviation e
PE03.1	*	Position deviation error - Detection method selection		0-2	0 : Continuous detec
PE06	BC1	Fully closed loop control - Speed deviation error dete		1-50000	400
PE07	BC2	Fully closed loop control - Position deviation error det		1-20000	100
PE10.1		Fully closed loop control - Position deviation error det		0-1	0 : By 1 kpulse
PE03.3	*	Fully closed loop control error - Reset selection		0-1	0 : Reset disabled (re

I/O Parameters:



No.	Abbr.	Name	Unit	Setting range	Axis1
Digital I/O					
Device setting					Setting
PD03.0-1	*	Device selection DI1		00-7F	0A
PD04.0-1	*	Device selection DI2		00-7F	0B
PD05.0-1	*	Device selection DI3		00-7F	22
PD51.0-1	*	Device selection DI3-2		00-7F	62
PD38.0-1	*	Device selection DI4		00-7F	2C
PD39.0-1	*	Device selection DI5		00-7F	2D
PD07.0-1	*	Device selection DO1		00-7F	05
PD08.0-1	*	Device selection DO2		00-7F	04
PD09.0-1	*	Device selection DO3		00-7F	03
Device assignment					Setting
PD01.0-7	*DIA1	Input signal automatic ON selection 1		0000000-00000FF0	00000000
Input filter					
PD11.0	*	Input signal filter selection		0-B	7 : 3.500ms
ALM output					
PD14.1	*	Warning occurrence - Output device selection		0-1	0 : WNG signal turn (
Analog output					
Analog monitor					
PC09.0-1		Analog monitor 1 output selection		00-1F	00 : Servo motor spe
PC11	MO1	Analog monitor 1 offset		-999-999	0
PC10.0-1		Analog monitor 2 output selection		00-1F	01 : Torque or thrust
PC12	MO2	Analog monitor 2 offset		-999-999	0
Stroke limit function					
Stroke limit function					
PC19.0	*	[AL. 099 Stroke limit warning] selection		0-1	**1 : Disabled**
PD41.2	*	Limit switch enabled status selection		0-1	**1 : Enabled only fc**
PD41.3	*	Sensor input method selection		0-1	0 : Input from servo

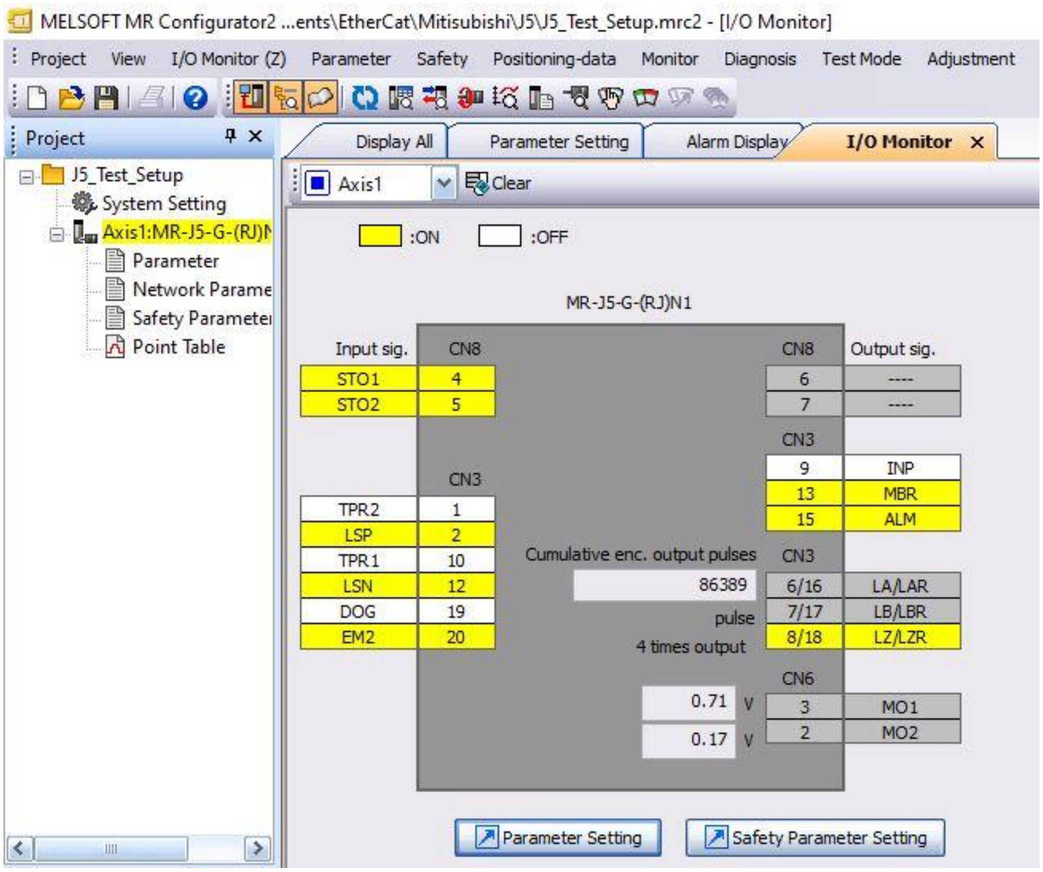
Safety Parameters:

Function display (L
Safety sub functio
List display
Safety sub functio
Safety sub functio
Network
Safety I/O device
Extension

Safety sub function — Selected Items Write — Single Axis Write

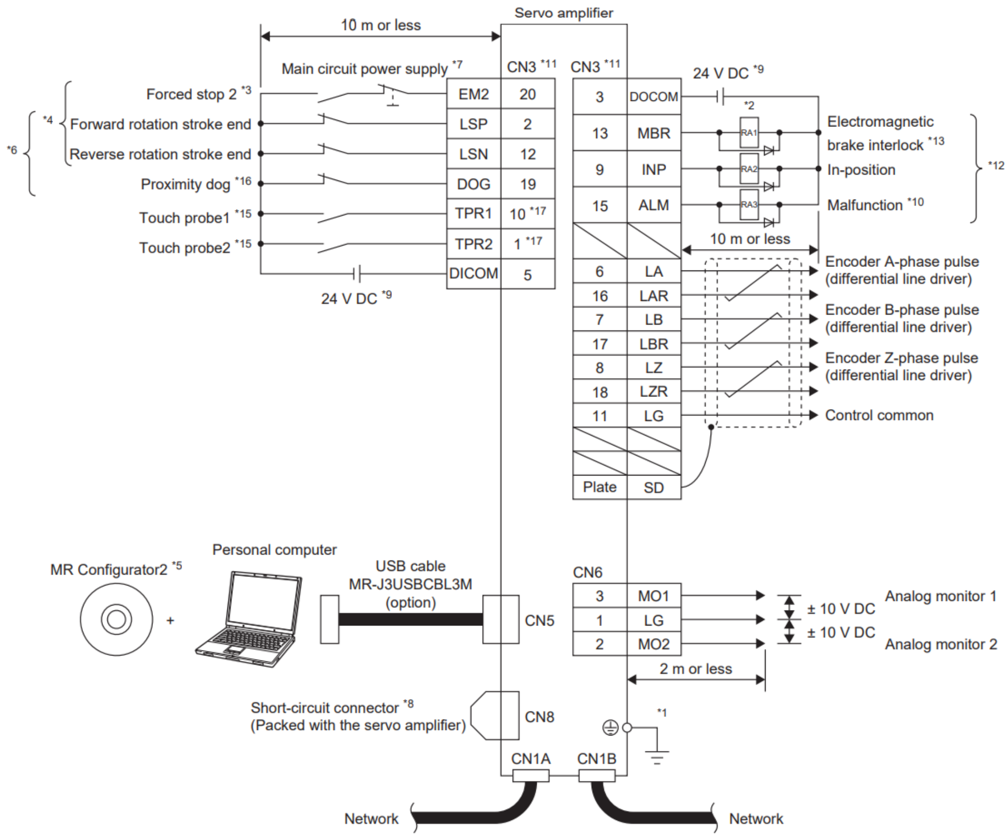
No.	Abbr.	Name	Unit	Setting range	Axis1
Safety sub-function activation setting					
Startup procedure					Setting
PSA01.0	**	Safety sub-function activation setting		0-1	0 : Disabled
Safety sub-function					
Basic					
PSA01.1	**	Input mode selection		0-2	0 : Safety observatic
PSA02.1	**	Position/Speed monitor setting		0-2	0 : Do not perform p
PSC03.0	**	Rotation direction sel./travel direction sel.		0-1	0 : CCW or positive c
Safety communication					
PSC01	**SNC	Transmission interval monitor time		16-1000	32
SS1					
PSA03	**SST	SS1/SS2 deceleration monitor time		0-60000	1000
SS2/SOS					
PSA03	**SST	SS1/SS2 deceleration monitor time		0-60000	1000
PSA04	**SSS	Safety sub-function - Stop speed		0-10000	50
PSA05	**SSDP	SOS permissible travel distance		0-1000	3
PSA06.0	**	SOS permissible travel distance unit sel.		0-2	0 : 1 [rev]
PSA15	**SLST	Safety sub-function - Speed detection delay time		5-2000	10
PSA17	**SOSPT	Safety sub-function - Position detection delay time		0-2000	0
PSA24	**SSTC	SS1/SS2 deceleration monitor time constant		0-20000	100
PSA25	**SSOF	SS1/SS2 deceleration monitor speed offset		0-10000	0
PSA26	**SSDT	SS1/SS2 deceleration monitor delay time		0-60000	100
SLS					
PSA07	**SLSDT1	SLS deceleration monitor time 1		0-60000	1000
PSA08	**SLSDT2	SLS deceleration monitor time 2		0-60000	1000
PSA09	**SLSDT3	SLS deceleration monitor time 3		0-60000	1000
PSA10	**SLSDT4	SLS deceleration monitor time 4		0-60000	1000
PSA11	**SLSS1	SLS speed 1		0-10000	50
PSA12	**SLSS2	SLS speed 2		0-10000	50
PSA13	**SLSS3	SLS speed 3		0-10000	50
PSA14	**SLSS4	SLS speed 4		0-10000	50
PSA15	**SLST	Safety sub-function - Speed detection delay time		5-2000	10
SSM					
PSA18	**SSMS	SSM speed		0-10000	50
PSA19	**SSMHW	SSM hysteresis width		0-10000	20
SDI					
PSA04	**SSS	Safety sub-function - Stop speed		0-10000	50
PSA15	**SLST	Safety sub-function - Speed detection delay time		5-2000	10
PSA27	**SDIDTP	SDI positive direction monitor delay time		0-60000	1000
PSA28	**SDIDTN	SDI negative direction monitor delay time		0-60000	1000
SLI					
PSA17	**SOSPT	Safety sub-function - Position detection delay time		0-2000	0
PSB01.0	**	SLI permissible travel distance - Unit selection 1		0-2	0 : 1 [rev]
PSB02	**SLIPP1	SLI permissible travel distance - Positive direction 1		0-1000	3
PSB06	**SLIPN1	SLI permissible travel distance - Negative direction 1		0-1000	3
SLT					
PSB10	**SLTP1	SLT torque upper limit value 1		-1000.0-1000.0	15.0
PSB11	**SLTP2	SLT torque upper limit value 2		-1000.0-1000.0	15.0
PSB12	**SLTP3	SLT torque upper limit value 3		-1000.0-1000.0	15.0

Axis I/O Monitor for Testing:

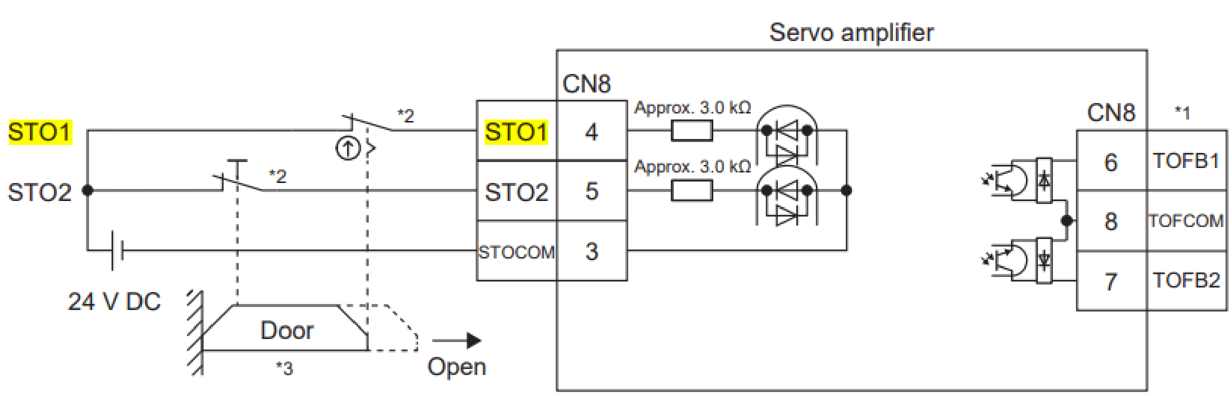


MR-J5-_G_

Sink I/O interface



For test operation –
DICOM - +24V
EM2, LSP, LSN – GND
STO1 and STO2 - + 24V



EtherCAT Explorer View

Manuf	Mitsubishi
Grp	
Name	MR-J4-20TM
Out	144 bits (18 bytes)
In	256 bits (32 bytes)
Axis #	1
pstate	RUNNING (1)
tracking_pstate	COMPLETE (2)
inpos	0
fpos	1.237558
tpos	1.044708
perr	-0.188275
vel	-5.048000
DRV MODE	Cyclic Sync Position (8)
PDO STATUS	0x1237
PDO CNTLWORD	0x000F
PDO ACT VEL	0xFFFF89B0
PDO ACT TORQ	0x0000FFF2
PDO ACT ERR	0xFFF4FB07
PDO HOME PWRUP	0xD6CCABC2
PDO ACT POS	0xD71BDFEA
PDO TARG POS	0xD70F8840
PDO TARG VEL	0x00000000
PDO DIG INP	0x00000000
State	8 (OPERATIONAL)
Delay	0 ns
Has DC	true (64 bits)
DC Parent	0
DC Active	true, Cyc time: 1000000 ns, Shft: 0
Parent	0
Config addr	0x1001 (4097)
Station Alias	0
Vndr	0x00000a1e (2590)
Product Code	0x00000201 (513)
Rev	0x00010000

Note: Rev shows 0x00020001 on the later model MR-J4-40TM1 and should be the firmware of choice.

Blank

Appendix
L2

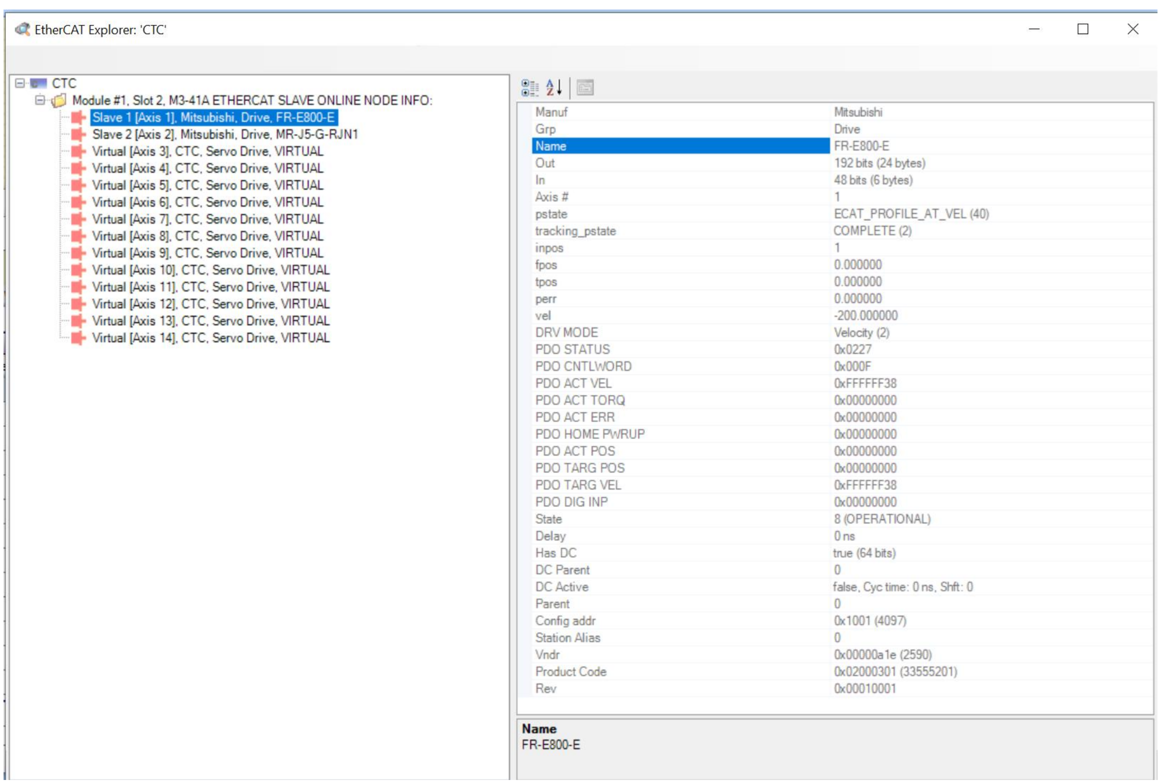
[L2] Mitsubishi FR-E800 VFD

Maxon manufactures a number of drives. That currently supported by the M3-41A is the EPOS4. For test purposes the EC-max brushless motor with hall sensors was tested, part number 283828. This section provides information that is specific to this manufacturer.

eCAT_driveType – 29

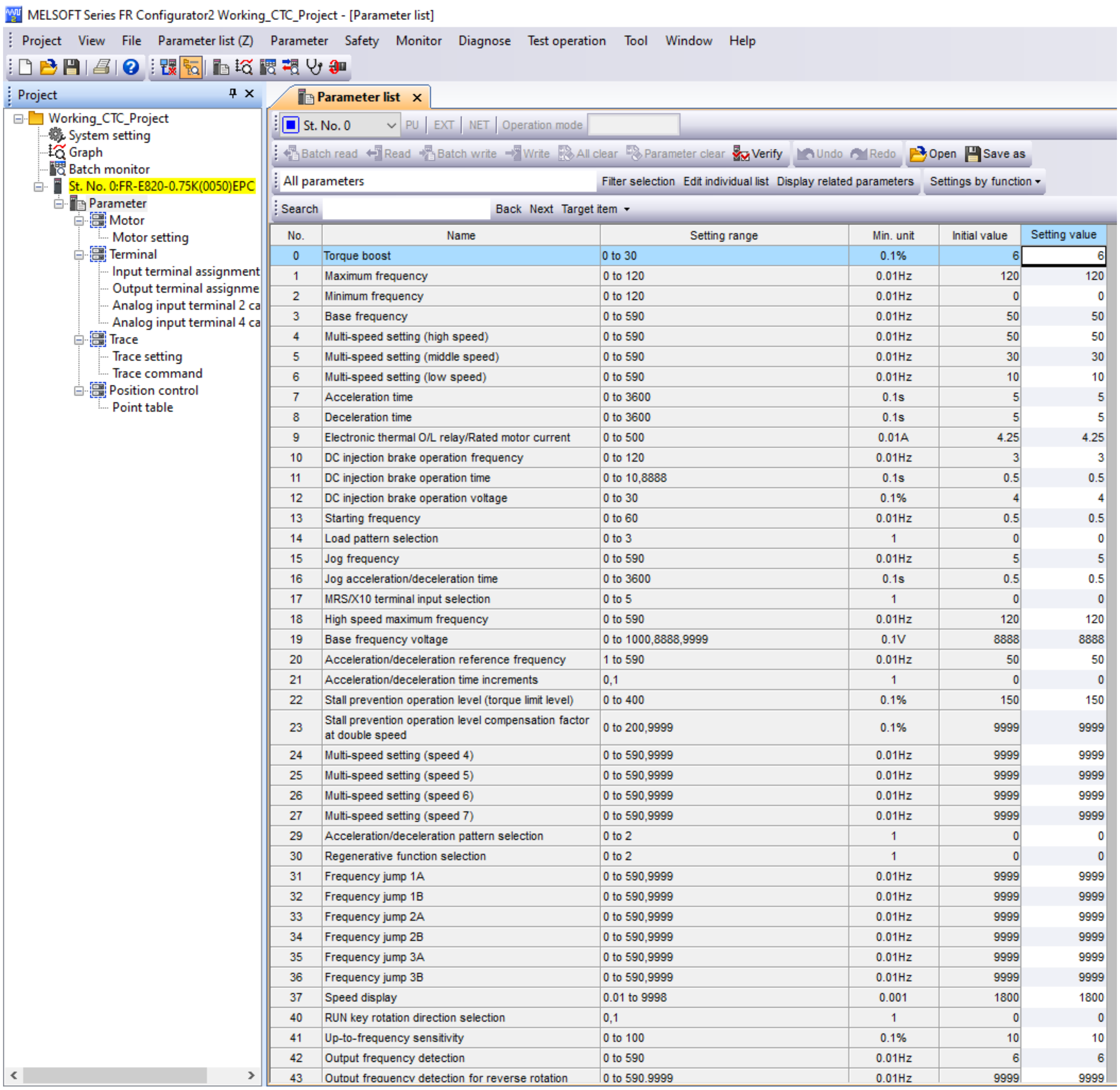
Overview

The E800 is a VFD drive whose velocity is controlled by Incentive. Its motion parameters are in rpm instead of rps used by servo's. The programming is identical to that of the Yaskawa V1000 and you should reference that section for examples. The Homing mode of the E800 is not supported, just velocity.



FR Configurator 2

V1.26 or greater of the FR Configurator is needed to support the E800 series of VFD drives. Below are some of the parameter settings used during testing.



No.	Name	Setting range	Min. unit	Initial value	Setting value
0	Torque boost	0 to 30	0.1%	6	6
1	Maximum frequency	0 to 120	0.01Hz	120	120
2	Minimum frequency	0 to 120	0.01Hz	0	0
3	Base frequency	0 to 590	0.01Hz	50	50
4	Multi-speed setting (high speed)	0 to 590	0.01Hz	50	50
5	Multi-speed setting (middle speed)	0 to 590	0.01Hz	30	30
6	Multi-speed setting (low speed)	0 to 590	0.01Hz	10	10
7	Acceleration time	0 to 3600	0.1s	5	5
8	Deceleration time	0 to 3600	0.1s	5	5
9	Electronic thermal O/L relay/Rated motor current	0 to 500	0.01A	4.25	4.25
10	DC injection brake operation frequency	0 to 120	0.01Hz	3	3
11	DC injection brake operation time	0 to 10,8888	0.1s	0.5	0.5
12	DC injection brake operation voltage	0 to 30	0.1%	4	4
13	Starting frequency	0 to 60	0.01Hz	0.5	0.5
14	Load pattern selection	0 to 3	1	0	0
15	Jog frequency	0 to 590	0.01Hz	5	5
16	Jog acceleration/deceleration time	0 to 3600	0.1s	0.5	0.5
17	MRS/X10 terminal input selection	0 to 5	1	0	0
18	High speed maximum frequency	0 to 590	0.01Hz	120	120
19	Base frequency voltage	0 to 1000,8888,9999	0.1V	8888	8888
20	Acceleration/deceleration reference frequency	1 to 590	0.01Hz	50	50
21	Acceleration/deceleration time increments	0,1	1	0	0
22	Stall prevention operation level (torque limit level)	0 to 400	0.1%	150	150
23	Stall prevention operation level compensation factor at double speed	0 to 200,9999	0.1%	9999	9999
24	Multi-speed setting (speed 4)	0 to 590,9999	0.01Hz	9999	9999
25	Multi-speed setting (speed 5)	0 to 590,9999	0.01Hz	9999	9999
26	Multi-speed setting (speed 6)	0 to 590,9999	0.01Hz	9999	9999
27	Multi-speed setting (speed 7)	0 to 590,9999	0.01Hz	9999	9999
29	Acceleration/deceleration pattern selection	0 to 2	1	0	0
30	Regenerative function selection	0 to 2	1	0	0
31	Frequency jump 1A	0 to 590,9999	0.01Hz	9999	9999
32	Frequency jump 1B	0 to 590,9999	0.01Hz	9999	9999
33	Frequency jump 2A	0 to 590,9999	0.01Hz	9999	9999
34	Frequency jump 2B	0 to 590,9999	0.01Hz	9999	9999
35	Frequency jump 3A	0 to 590,9999	0.01Hz	9999	9999
36	Frequency jump 3B	0 to 590,9999	0.01Hz	9999	9999
37	Speed display	0.01 to 9998	0.001	1800	1800
40	RUN key rotation direction selection	0,1	1	0	0
41	Up-to-frequency sensitivity	0 to 100	0.1%	10	10
42	Output frequency detection	0 to 590	0.01Hz	6	6
43	Output frequency detection for reverse rotation	0 to 590,9999	0.01Hz	9999	9999

No.	Name	Setting range	Min. unit	Initial value	Setting value
43	Output frequency detection for reverse rotation	0 to 590,9999	0.01Hz	9999	9999
44	Second acceleration/deceleration time	0 to 3600	0.1s	5	5
45	Second deceleration time	0 to 3600,9999	0.1s	9999	9999
46	Second torque boost	0 to 30,9999	0.1%	9999	9999
47	Second V/F (base frequency)	0 to 590,9999	0.01Hz	9999	9999
48	Second stall prevention operation level	0 to 400,9999	0.1%	9999	9999
51	Second electronic thermal O/L relay/Rated second motor current	0 to 500,9999	0.01A	9999	9999
52	Operation panel main monitor selection	0,5 to 14,17 to 20,22 to 33,35,38,40 to 42,44,45,50 to 57,61,62,64,65,67,68,91,97,100	1	0	0
53	Frequency / rotation speed unit switchover	0,1,4	1	0	0
57	Restart coasting time	0,0.1 to 30,9999	0.1s	9999	9999
58	Restart cushion time	0 to 60	0.1s	1	1
59	Remote function selection	0 to 3,11 to 13	1	0	0
60	Energy saving control selection	0,9	1	0	0
61	Reference current	0 to 500,9999	0.01A	9999	9999
62	Reference value at acceleration	0 to 400,9999	1%	9999	9999
63	Reference value at deceleration	0 to 400,9999	1%	9999	9999
65	Retry selection	0 to 5	1	0	0
66	Stall prevention operation reduction starting frequency	0 to 590	0.01Hz	50	50
67	Number of retries at fault occurrence	0 to 10,101 to 110	1	0	0
68	Retry waiting time	0.1 to 600	0.1s	1	1
69	Retry count display erase	0	1	0	0
70	Special regenerative brake duty	0 to 100	0.1%	0	0
71	Applied motor	0,3,5,6,10,13,15,16,20,23,30,33,40,43,50,53,70 ,73,540,1140,1800,1803,8090,8093,9090,9093	1	0	0
72	PWM frequency selection	0 to 15	1	1	1
73	Analog input selection	0,1,6,10,11,16	1	1	1
74	Input filter time constant	0 to 8	1	1	1
75	Reset selection/disconnected PU detection/PU stop selection	0 to 3,14 to 17	1	14	14
77	Parameter write selection	0 to 2	1	0	0
78	Reverse rotation prevention selection	0 to 2	1	0	0
79	Operation mode selection	0 to 4,6,7	1	0	0
80	Motor capacity	0.1 to 30,9999	0.01kW	9999	9999
81	Number of motor poles	2,4,6,8,10,12,9999	1	9999	9999
82	Motor excitation current	0 to 500,9999	0.01A	9999	9999
83	Rated motor voltage	0 to 1000	0.1V	200	200
84	Rated motor frequency	10 to 400,9999	0.01Hz	9999	9999
89	Speed control gain (Advanced magnetic flux vector)	0 to 200,9999	0.1%	9999	9999
90	Motor constant (R1)	0 to 50,9999	0.001Ω	9999	9999
91	Motor constant (R2)	0 to 50,9999	0.001Ω	9999	9999

No.	Name	Setting range	Min. unit	Initial value	Setting value
92	Motor constant (L1)/d-axis inductance (Ld)	0 to 6000,9999	0.1mH	9999	9999
93	Motor constant (L2)/q-axis inductance (Lq)	0 to 6000,9999	0.1mH	9999	9999
94	Motor constant (X)	0 to 100,9999	0.1%	9999	9999
95	Online auto tuning selection	0,1	1	0	0
96	Auto tuning setting/status	0,1,11	1	0	0
117	PU communication station number	0 to 31	1	0	0
118	PU communication speed	48,96,192,384,576,768,1152	1	192	192
119	PU communication stop bit length / data length	0,1,10,11	1	1	1
120	PU communication parity check	0 to 2	1	2	2
121	Number of PU communication retries	0 to 10,9999	1	1	1
122	PU communication check time interval	0,0.1 to 999.8,9999	0.1s	0	0
123	PU communication waiting time setting	0 to 150,9999	1ms	9999	9999
124	PU communication CR/LF selection	0 to 2	1	1	1
125	Terminal 2 frequency setting gain frequency	0 to 590	0.01Hz	50	50
126	Terminal 4 frequency setting gain frequency	0 to 590	0.01Hz	50	50
127	PID control automatic switchover frequency	0 to 590,9999	0.01Hz	9999	9999
128	PID action selection	0,20,21,40 to 43,50,51,60,61,1000,1001,1010,1011,2000,2001,2010,2011	1	0	0
129	PID proportional band	0.1 to 1000,9999	0.1%	100	100
130	PID integral time	0.1 to 3600,9999	0.1s	1	1
131	PID upper limit	0 to 100,9999	0.1%	9999	9999
132	PID lower limit	0 to 100,9999	0.1%	9999	9999
133	PID action set point	0 to 100,9999	0.01%	9999	9999
134	PID differential time	0.01 to 10,9999	0.01s	9999	9999
136	MC switchover interlock time	0 to 100	0.1s	1	1
139	Automatic switchover frequency from inverter to bypass operation	0 to 60,9999	0.01Hz	9999	9999
145	PU display language selection	0 to 7	1	-	1
147	Acceleration/deceleration time switching frequency	0 to 590,9999	0.01Hz	9999	9999
150	Output current detection level	0 to 400	0.1%	150	150
151	Output current detection signal delay time	0 to 10	0.1s	0	0
152	Zero current detection level	0 to 400	0.1%	5	5
153	Zero current detection time	0 to 10	0.01s	0.5	0.5
154	Voltage reduction selection during stall prevention operation	1,11	1	1	1
156	Stall prevention operation selection	0 to 31,100,101	1	0	0
157	OL signal output timer	0 to 25,9999	0.1s	0	0
160	User group read selection	0,1,9999	1	0	0
161	Frequency setting/key lock operation selection	0,1,10,11	1	0	0
162	Automatic restart after instantaneous power failure selection	0,1,10,11	1	0	0
165	Stall prevention operation level for restart	0 to 400	0.1%	150	150

No.	Name	Setting range	Min. unit	Initial value	Setting value
166	Output current detection signal retention time	0 to 10,9999	0.1s	0.1	0.1
167	Output current detection operation selection	0,1,10,11	1	0	0
168	Parameter for manufacturer setting		1		8654
169	Parameter for manufacturer setting		1		5
170	Watt-hour meter clear	0,10,9999	1	9999	9999
171	Operation hour meter clear	0,9999	1	9999	9999
172	User group registered display/batch clear	(0 to 16),9999	1	0	0
173	User group registration	0 to 1999,9999	1	9999	9999
174	User group clear	0 to 1999,9999	1	9999	9999
178	STF terminal function selection / DI0 terminal function selection	0 to 5,7,8,10,12 to 16,18,22 to 27,30,37,42,43,46,47,50,51,60,62,65 to 67,72,74,76,84,87 to 89,92,9999	1	60	60
179	STR terminal function selection / DI1 terminal function selection	0 to 5,7,8,10,12 to 16,18,22 to 27,30,37,42,43,46,47,50,51,61,62,65 to 67,72,74,76,84,87 to 89,92,9999	1	61	61
180	RL terminal function selection	0 to 4,8,13 to 15,18,22 to 24,26,27,30,37,42,43,46,47,50,51,72,74,76,84, 87 to 89,92,9999	1	0	0
181	RM terminal function selection	0 to 4,8,13 to 15,18,22 to 24,26,27,30,37,42,43,46,47,50,51,72,74,76,84, 87 to 89,92,9999	1	1	1
182	RH terminal function selection	0 to 4,8,13 to 15,18,22 to 24,26,27,30,37,42,43,46,47,50,51,72,74,76,84, 87 to 89,92,9999	1	2	2
183	MRS terminal function selection	0 to 4,8,13 to 15,18,22 to 24,26,27,30,37,42,43,46,47,50,51,72,74,76,84, 87 to 89,92,9999	1	24	24
184	RES terminal function selection	0 to 4,8,13 to 15,18,22 to 24,26,27,30,37,42,43,46,47,50,51,72,74,76,84, 87 to 89,92,9999	1	9999	9999
185	NET X1 input selection	0 to 4,8,13 to 15,18,22 to 24,26,27,30,37,42,43,46,47,50,51,72,74,76,84, 87 to 89,92,9999	1	9999	9999
186	NET X2 input selection	0 to 4,8,13 to 15,18,22 to 24,26,27,30,37,42,43,46,47,50,51,72,74,76,84, 87 to 89,92,9999	1	9999	9999
187	NET X3 input selection	0 to 4,8,13 to 15,18,22 to 24,26,27,30,37,42,43,46,47,50,51,72,74,76,84, 87 to 89,92,9999	1	9999	9999
188	NET X4 input selection	0 to 4,8,13 to 15,18,22 to 24,26,27,30,37,42,43,46,47,50,51,72,74,76,84, 87 to 89,92,9999	1	9999	9999
189	NET X5 input selection	0 to 4,8,13 to 15,18,22 to 24,26,27,30,37,42,43,46,47,50,51,72,74,76,84, 87 to 89,92,9999	1	9999	9999
		0,1,3,4,7,8,11 to 16,18 to 20,24 to 28,30 to 36,38 to 41,44 to 48,56,57,60 to 66,70,80,81,84,90 to 93,95,96,98,99,100,101,103,104,107,108,111			

No.	Name	Setting range	Min. unit	Initial value	Setting value
198	Display corrosion level	(1 to 3)	1	1	1
232	Multi-speed setting (8 speed)	0 to 590,9999	0.01Hz	9999	9999
233	Multi-speed setting (9 speed)	0 to 590,9999	0.01Hz	9999	9999
234	Multi-speed setting (10 speed)	0 to 590,9999	0.01Hz	9999	9999
235	Multi-speed setting (11 speed)	0 to 590,9999	0.01Hz	9999	9999
236	Multi-speed setting (12 speed)	0 to 590,9999	0.01Hz	9999	9999
237	Multi-speed setting (13 speed)	0 to 590,9999	0.01Hz	9999	9999
238	Multi-speed setting (14 speed)	0 to 590,9999	0.01Hz	9999	9999
239	Multi-speed setting (15 speed)	0 to 590,9999	0.01Hz	9999	9999
240	Soft-PWM operation selection	0,1	1	1	1
241	Analog input display unit switchover	0,1	1	0	0
244	Cooling fan operation selection	0,1	1	1	1
245	Rated slip	0 to 50,9999	0.01%	9999	9999
246	Slip compensation time constant	0.01 to 10	0.01s	0.5	0.5
247	Constant output range slip compensation selection	0,9999	1	9999	9999
249	Earth (ground) fault detection at start	0,1	1	1	1
250	Stop selection	0 to 100,1000 to 1100,8888,9999	0.1s	9999	9999
251	Output phase loss protection selection	0,1	1	1	1
255	Life alarm status display	(0 to 15,32 to 111,256 to 879)	1	0	0
256	Inrush current limit circuit life display	(0 to 100)	1%	100	100
257	Control circuit capacitor life display	(0 to 100)	1%	100	100
258	Main circuit capacitor life display	(0 to 100)	1%	100	100
259	Main circuit capacitor life measuring	0,1	1	0	0
260	PWM frequency automatic switchover	0,10	1	10	10
261	Power failure stop selection	0 to 2	1	0	0
267	Terminal 4 input selection	0 to 2	1	0	0
268	Monitor decimal digits selection	0,1,9999	1	9999	9999
269	Parameter for manufacturer setting				
270	Stop-on contact control selection	0,1,11	1	0	0
275	Stop-on contact excitation current low-speed scaling factor	0 to 300,9999	0.1%	9999	9999
276	PWM carrier frequency at stop-on contact	0 to 9,9999	1	9999	9999
277	Stall prevention operation current switchover	0,1	1	0	0
278	Brake opening frequency	0 to 30	0.01Hz	3	3
279	Brake opening current	0 to 400	0.1%	130	130
280	Brake opening current detection time	0 to 2	0.1s	0.3	0.3
281	Brake operation time at start	0 to 5	0.1s	0.3	0.3
282	Brake operation frequency	0 to 30	0.01Hz	6	6
283	Brake operation time at stop	0 to 5	0.1s	0.3	0.3
284	Deceleration detection function selection	0,1	1	0	0
285	Overspeed detection frequency/Speed deviation excess detection frequency	0 to 30,9999	0.01Hz	9999	9999

No.	Name	Setting range	Min. unit	Initial value	Setting value
286	Droop gain	0 to 100	0.1%	0	0
287	Droop filter time constant	0 to 1	0.01s	0.3	0.3
289	Inverter output terminal filter	5 to 50,9999	1ms	9999	9999
290	Monitor negative output selection	0,1,4,5,8,9,12,13	1	0	0
292	Automatic acceleration/deceleration	0,1,7,8,11	1	0	0
293	Acceleration/deceleration separate selection	0 to 2	1	0	0
295	Frequency change increment amount setting	0,0.01,0.1,1,10	0.01	0	0
298	Frequency search gain	0 to 32767,9999	1	9999	9999
299	Rotation direction detection selection at restarting	0,1,9999	1	0	0
338	Communication operation command source	0,1	1	0	0
339	Communication speed command source	0 to 2	1	0	0
340	Communication startup mode selection	0,1,10	1	10	10
342	Communication EEPROM write selection	0,1	1	0	0
343	Communication error count		1	0	0
349	Communication reset selection	0,1	1	0	0
374	Overspeed detection level	0 to 590,9999	0.01Hz	9999	9999
375	Faulty acceleration rate detection level	0 to 400,9999	0.01Hz	9999	9999
414	PLC function operation selection	0 to 2,11,12	1	0	0
415	Inverter operation lock mode setting	0,1	1	0	0
420	Command pulse scaling factor numerator (electronic gear numerator)	1 to 32767	1	1	1
421	Command pulse multiplication denominator (electronic gear denominator)	1 to 32767	1	1	1
422	Position control gain	0 to 150	1s^-1	10	10
423	Position feed forward gain	0 to 100	1%	0	0
425	Position feed forward command filter	0 to 5	0.001s	0	0
426	In-position width	0 to 32767	1pulse	100	100
427	Excessive level error	0 to 400,9999	1Kpulse	40	40
430	Pulse monitor selection	0 to 5,100 to 105,1000 to 1005,1100 to 1105,8888,9999	1	9999	9999
446	Model position control gain	0 to 150	1s^-1	25	25
450	Second applied motor	0,3,5,6,10,13,15,16,20,23,30,33,40,43,50,53,70,73,540,1140,1800,1803,8090,8093,9090,9093,9999	1	9999	9999
451	Second motor control method selection	10 to 14,20,40,9999	1	9999	9999
453	Second motor capacity	0.1 to 30,9999	0.01kW	9999	9999
454	Number of second motor poles	2,4,6,8,10,12,9999	1	9999	9999
455	Second motor excitation current	0 to 500,9999	0.01A	9999	9999
456	Rated second motor voltage	0 to 1000	0.1V	200	200
457	Rated second motor frequency	10 to 400,9999	0.01Hz	9999	9999
458	Second motor constant (R1)	0 to 50,9999	0.001Ω	9999	9999
459	Second motor constant (R2)	0 to 50,9999	0.001Ω	9999	9999
460	Second motor constant (L1) / d-axis inductance (Ld)	0 to 6000,9999	0.1mH	9999	9999

No.	Name	Setting range	Min. unit	Initial value	Setting value
461	Second motor constant (L2) / q-axis inductance (Lq)	0 to 6000,9999	0.1mH	9999	9999
462	Second motor constant (X)	0 to 100,9999	0.1%	9999	9999
463	Second motor auto tuning setting/status	0,1,11	1	0	0
464	Digital position control sudden stop deceleration time	0.01 to 360	0.01s	0.01	0.01
465	First target position lower 4 digits	0 to 9999	1	0	0
466	First target position upper 4 digits	0 to 9999	1	0	0
467	Second target position lower 4 digits	0 to 9999	1	0	0
468	Second target position upper 4 digits	0 to 9999	1	0	0
469	Third target position lower 4 digits	0 to 9999	1	0	0
470	Third target position upper 4 digits	0 to 9999	1	0	0
471	Fourth target position lower 4 digits	0 to 9999	1	0	0
472	Fourth target position upper 4 digits	0 to 9999	1	0	0
473	Fifth target position lower 4 digits	0 to 9999	1	0	0
474	Fifth target position upper 4 digits	0 to 9999	1	0	0
475	Sixth target position lower 4 digits	0 to 9999	1	0	0
476	Sixth target position upper 4 digits	0 to 9999	1	0	0
477	Seventh target position lower 4 digits	0 to 9999	1	0	0
478	Seventh target position upper 4 digits	0 to 9999	1	0	0
495	Remote output selection	0,1,10,11	1	0	0
496	Remote output data 1	0 to 4095	1	0	0
497	Remote output data 2	0 to 4095	1	0	0
498	PLC function flash memory clear	0,9696(0 to 9999)	1	0	0
502	Stop mode selection at communication error	0 to 2,6	1	0	0
503	Maintenance timer	0(1 to 9998)	1	0	0
504	Maintenance timer alarm output set time	0 to 9998,9999	1	9999	9999
505	Speed setting reference	1 to 590	0.01Hz	50	50
506	Display estimated main circuit capacitor residual life	(0 to 100)	1%	100	100
507	Display/reset ABC relay contact life	(0 to 100)	1%	100	100
509	Display power cycle life	(0 to 100)	0.01%	100	100
510	Rough match output range	0 to 32767	1	0	0
511	Home position return shifting speed	0 to 400	0.01Hz	0.5	0.5
514	Emergency drive dedicated retry waiting time	0.1 to 600,9999	0.1s	9999	9999
515	Emergency drive dedicated retry count	1 to 200,9999	1	1	1
523	Emergency drive mode selection	100,111,112,121,122,200,211,212,221,222,300 ,311,312,321,322,400,411,412,421,422,9999	1	9999	9999
524	Emergency drive running speed	0 to 590,9999	0.01Hz	9999	9999
538	Current position retention selection	1,2,11,12,9999	1	9999	9999
541	Frequency command sign selection	0,1	1	0	0
544	CC-Link extended setting	0,1,12,14,18,38,100,112,114,118,138	1	0	0
547	USB communication station number	0 to 31	1	0	0
548	USB communication check time interval	0 to 999.8,9999	0.1s	9999	9999
549	Protocol selection	0,1	1	0	0

No.	Name	Setting range	Min. unit	Initial value	Setting value
550	NET mode operation command source selection	0,5,9999	1	9999	9999
551	PU mode operation command source selection	3,4,9999	1	9999	9999
552	Frequency jump range	0 to 30,9999	0.01Hz	9999	9999
553	PID deviation limit	0 to 100,9999	0.1%	9999	9999
554	PID signal operation selection	0 to 3,10 to 13	1	0	0
555	Current average time	0.1 to 1	0.1s	1	1
556	Data output mask time	0 to 20	0.1s	0	0
557	Current average value monitor signal output reference current	0 to 500	0.01A	5	5
560	Second frequency search gain	0 to 32767,9999	1	9999	9999
561	PTC thermistor protection level	0.5 to 30,9999	0.01kΩ	9999	9999
563	Energization time carrying-over times	(0 to 65535)	1	0	0
564	Operating time carrying-over times	(0 to 65535)	1	0	0
569	Second motor speed control gain	0 to 200,9999	0.1%	9999	9999
570	Multiple rating setting	1,2	1	2	2
571	Holding time at a start	0 to 10,9999	0.1s	9999	9999
574	Second motor online auto tuning	0,1	1	0	0
575	Output interruption detection time	0 to 3600,9999	0.1s	1	1
576	Output interruption detection level	0 to 590	0.01Hz	0	0
577	Output interruption cancel level	900 to 1100	0.1%	1000	1000
592	Traverse function selection	0 to 2	1	0	0
593	Maximum amplitude amount	0 to 25	0.1%	10	10
594	Amplitude compensation amount during deceleration	0 to 50	0.1%	10	10
595	Amplitude compensation amount during acceleration	0 to 50	0.1%	10	10
596	Amplitude acceleration time	0.1 to 3600	0.1s	5	5
597	Amplitude deceleration time	0.1 to 3600	0.1s	5	5
600	First free thermal reduction frequency 1	0 to 590,9999	0.01Hz	9999	9999
601	First free thermal reduction ratio 1	1 to 100	1%	100	100
602	First free thermal reduction frequency 2	0 to 590,9999	0.01Hz	9999	9999
603	First free thermal reduction ratio 2	1 to 100	1%	100	100
604	First free thermal reduction frequency 3	0 to 590,9999	0.01Hz	9999	9999
607	Motor permissible load level	110 to 250	1%	150	150
608	Second motor permissible load level	110 to 250,9999	1%	9999	9999
609	PID set point/deviation input selection	2 to 5	1	2	2
610	PID measured value input selection	2 to 5	1	3	3
611	Acceleration time at a restart	0 to 3600,9999	0.1s	9999	9999
631	Inverter output fault enable/disable selection	0,1	1	0	0
639	Brake opening current selection	0,1	1	0	0
640	Brake operation frequency selection	0,1	1	0	0
653	Speed smoothing control	0 to 200	0.1%	0	0
654	Speed smoothing cutoff frequency	0 to 120	0.01Hz	20	20
[illegible]	Increased magnetic excitation deceleration operation	[illegible]	[illegible]	[illegible]	[illegible]

No.	Name	Setting range	Min. unit	Initial value	Setting value
791	Acceleration time in low-speed range	0 to 3600,9999	0.1s	9999	9999
792	Deceleration time in low-speed range	0 to 3600,9999	0.1s	9999	9999
800	Control method selection	0 to 5,9 to 14,19,20,40	1	40	40
801	Output limit level	0 to 400,9999	0.1%	9999	9999
802	Pre-excitation selection	0,1	1	0	0
803	Constant output range torque characteristic selection	0 to 2,10	1	0	0
804	Torque command source selection	0,1,3 to 6	1	0	0
805	Torque command value (RAM)	600 to 1400	1%	1000	1000
806	Torque command value (RAM, EEPROM)	600 to 1400	1%	1000	1000
807	Speed limit selection	0,1	1	0	0
808	Speed limit	0 to 400	0.01Hz	50	50
809	Reverse-side speed limit	0 to 400,9999	0.01Hz	9999	9999
810	Torque limit input method selection	0 to 2	1	0	0
811	Set resolution switchover	0,10	1	0	0
812	Torque limit level (regeneration)	0 to 400,9999	0.1%	9999	9999
813	Torque limit level (3rd quadrant)	0 to 400,9999	0.1%	9999	9999
814	Torque limit level (4th quadrant)	0 to 400,9999	0.1%	9999	9999
815	Torque limit level 2	0 to 400,9999	0.1%	9999	9999
816	Torque limit level during acceleration	0 to 400,9999	0.1%	9999	9999
817	Torque limit level during deceleration	0 to 400,9999	0.1%	9999	9999
820	Speed control P gain 1	0 to 1000	1%	60	60
821	Speed control integral time 1	0 to 20	0.001s	0.333	0.333
822	Speed setting filter 1	0 to 5,9999	0.001s	9999	9999
824	Torque control P gain 1 (current loop proportional gain)	0 to 500	1%	100	100
825	Torque control integral time 1 (current loop integral time)	0 to 500	0.1ms	5	5
826	Torque setting filter 1	0 to 5,9999	0.001s	9999	9999
828	Model speed control gain	0 to 1000	1rad/s	100	100
830	Speed control P gain 2	0 to 1000,9999	1%	9999	9999
831	Speed control integral time 2	0 to 20,9999	0.001s	9999	9999
832	Speed setting filter 2	0 to 5,9999	0.001s	9999	9999
834	Torque control P gain 2 (current loop proportional gain)	0 to 500,9999	1%	9999	9999
835	Torque control integral time 2 (current loop integral time)	0 to 500,9999	0.1ms	9999	9999
836	Torque setting filter 2	0 to 5,9999	0.001s	9999	9999
840	Torque bias selection	0 to 3,9999	1	9999	9999
841	Torque bias 1	600 to 1400,9999	1%	9999	9999
842	Torque bias 2	600 to 1400,9999	1%	9999	9999
843	Torque bias 3	600 to 1400,9999	1%	9999	9999
844	Torque bias filter	0 to 5,9999	0.001s	9999	9999
845	Torque bias operation time	0 to 5,9999	0.01s	9999	9999

No.	Name	Setting range	Min. unit	Initial value	Setting value
845	Torque bias operation time	0 to 5,9999	0.01s	9999	9999
846	Torque bias balance compensation	0 to 100,9999	0.1%	9999	9999
847	Fall-time torque bias terminal 4 bias	0 to 400,9999	1%	9999	9999
848	Fall-time torque bias terminal 4 gain	0 to 400,9999	1%	9999	9999
849	Analog input offset adjustment	0 to 200	0.1%	100	100
850	Brake operation selection	0 to 2	1	0	0
853	Speed deviation time	0 to 100	0.1s	1	1
854	Excitation ratio	0 to 100	1%	100	100
858	Terminal 4 function assignment	0,4,6,9999	1	0	0
859	Torque current/Rated PM motor current	0 to 500,9999	0.01A	9999	9999
860	Second motor torque current/Rated PM motor current	0 to 500,9999	0.01A	9999	9999
864	Torque detection	0 to 400	0.1%	150	150
865	Low speed detection	0 to 590	0.01Hz	1.5	1.5
866	Torque monitoring reference	0 to 400	0.1%	150	150
870	Speed detection hysteresis	0 to 15	0.01Hz	0	0
872	Input phase loss protection selection	0,1	1	1	1
874	OLT level setting	0 to 400	0.1%	150	150
877	Speed feed forward control/model adaptive speed control selection	0 to 2	1	0	0
878	Speed feed forward filter	0.01 to 1	0.01s	0.01	0.01
879	Speed feed forward torque limit	0 to 400	0.1%	150	150
880	Load inertia ratio	0 to 200	0.1	7	7
881	Speed feed forward gain	0 to 1000	1%	0	0
882	Regeneration avoidance operation selection	0 to 2	1	0	0
883	Regeneration avoidance operation level	300 to 1200	0.1V	400	400
885	Regeneration avoidance compensation frequency limit value	0 to 45,9999	0.01Hz	6	6
886	Regeneration avoidance voltage gain	0 to 200	0.1%	100	100
888	Free parameter 1	0 to 9998,9999	1	9999	9999
889	Free parameter 2	0 to 9998,9999	1	9999	9999
891	Cumulative power monitor digit shifted times	0 to 4,9999	1	9999	9999
892	Load factor	30 to 150	0.1%	100	100
893	Energy saving monitor reference (motor capacity)	0.1 to 30	0.01kW	0.75	0.75
894	Control selection during commercial power-supply operation	0 to 3	1	0	0
895	Power saving rate reference value	0,1,9999	1	9999	9999
896	Power unit cost	0 to 500,9999	0.01	9999	9999
897	Power saving monitor average time	0 to 1000,9999	1h	9999	9999
898	Power saving cumulative monitor clear	0,1,10,9999	1	9999	9999
899	Operation time rate (estimated value)	0 to 100,9999	0.1%	9999	9999
902	Terminal 2 frequency setting bias frequency	0 to 590	0.01Hz	0	0
(902)	Terminal 2 frequency setting bias	0 to 300	0.1%	0	0

No.	Name	Setting range	Min. unit	Initial value	Setting value
1196	PLC function user parameter 47	0 to 65535	1	0	0
1197	PLC function user parameter 48	0 to 65535	1	0	0
1198	PLC function user parameter 49	0 to 65535	1	0	0
1199	PLC function user parameter 50	0 to 65535	1	0	0
1220	Direct command mode selection	0,4	1	0	0
1222	First positioning acceleration time	0.01 to 360	0.01s	5	5
1223	First positioning deceleration time	0.01 to 360	0.01s	5	5
1225	First positioning sub-function	0,1,10,11,100,101,110,111	1	10	10
1226	Second positioning acceleration time	0.01 to 360	0.01s	5	5
1227	Second positioning deceleration time	0.01 to 360	0.01s	5	5
1229	Second positioning sub-function	0,1,10,11,100,101,110,111	1	10	10
1230	Third positioning acceleration time	0.01 to 360	0.01s	5	5
1231	Third positioning deceleration time	0.01 to 360	0.01s	5	5
1233	Third positioning sub-function	0,1,10,11,100,101,110,111	1	10	10
1234	Fourth positioning acceleration time	0.01 to 360	0.01s	5	5
1235	Fourth positioning deceleration time	0.01 to 360	0.01s	5	5
1237	Fourth positioning sub-function	0,1,10,11,100,101,110,111	1	10	10
1238	Fifth positioning acceleration time	0.01 to 360	0.01s	5	5
1239	Fifth positioning deceleration time	0.01 to 360	0.01s	5	5
1241	Fifth positioning sub-function	0,1,10,11,100,101,110,111	1	10	10
1242	Sixth positioning acceleration time	0.01 to 360	0.01s	5	5
1243	Sixth positioning deceleration time	0.01 to 360	0.01s	5	5
1245	Sixth positioning sub-function	0,1,10,11,100,101,110,111	1	10	10
1246	Seventh positioning acceleration time	0.01 to 360	0.01s	5	5
1247	Seventh positioning deceleration time	0.01 to 360	0.01s	5	5
1249	Seventh positioning sub-function	0,10,100,110	1	10	10
1282	Home position return method selection	2 to 4,6,103,106,203,206	1	4	4
1283	Home position return speed	0 to 400	0.01Hz	2	2
1285	Home position shift amount lower 4 digits	0 to 9999	1	0	0
1286	Home position shift amount upper 4 digits	0 to 9999	1	0	0
1289	Home position return stopper torque	0 to 200	0.1%	40	40
1290	Home position return stopper waiting time	0 to 10	0.1s	0.5	0.5
1292	Position control terminal input selection	0,1,10,11,100,101,110,111	1	0	0
1293	Roll feeding mode selection	0 to 2	1	0	0
1294	Position detection lower 4 digits	0 to 9999	1	0	0
1295	Position detection upper 4 digits	0 to 9999	1	0	0
1296	Position detection selection	0 to 2	1	0	0
1297	Position detection hysteresis width	0 to 32767	1	0	0
1298	Second position control gain	0 to 150	1s^-1	10	10
1299	Second pre-excitation selection	0,1	1	0	0
1305	EtherCAT node address setting	0 to 65535	1	0	0

Appendix
L3

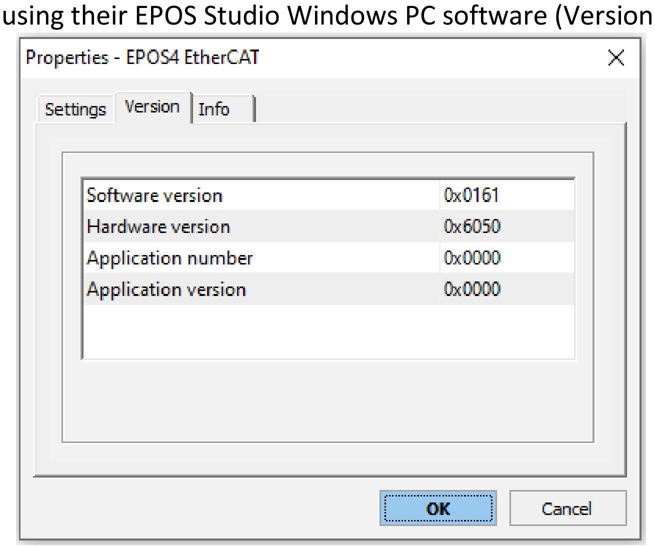
[L3] Maxon EPOS4

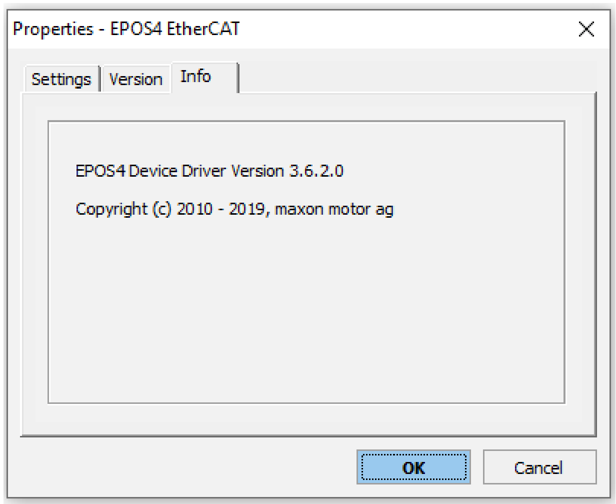
Maxon manufactures a number of drives. That currently supported by the M3-41A is the EPOS4. For test purposes the EC-max brushless motor with hall sensors was tested, part number 283828. This section provides information that is specific to this manufacturer.

eCAT_driveType – 24

Overview

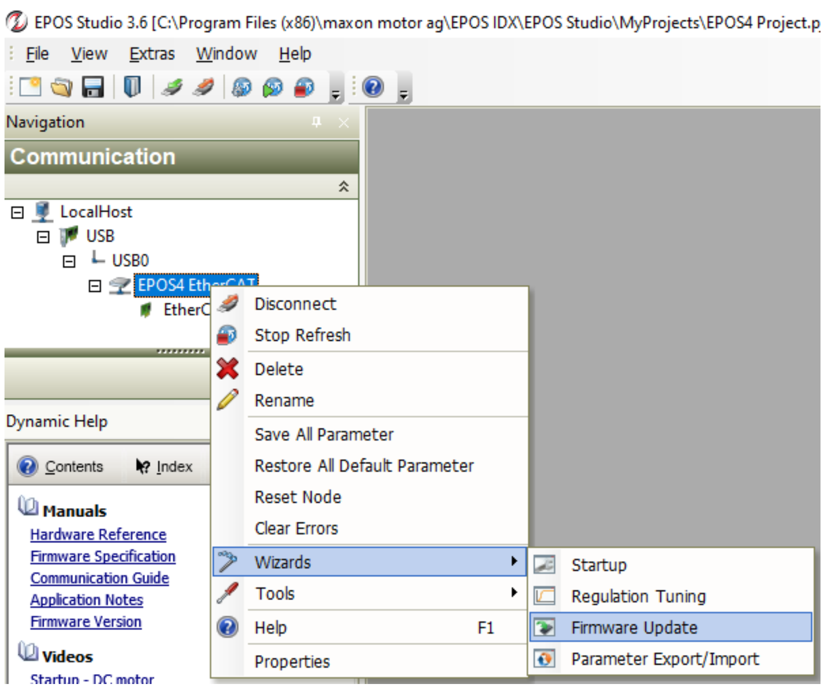
Only CSP and Homing mode are supported with the Maxon EPOS4 drive, Profile position mode is functional but not recommended. Additionally the DC SYNC motion control instruction is ignored since it is automatically enabled prior to network operation. Thus it is automatically enabled by the EtherCAT Master with the SYNC0 always having an offset from the network cycle time, typically 2 mS cycle time, with a shift of ‘cycle time/4’ or 500uS for 2 mS, whichever is smaller. One millisecond cycle time is not supported in CSP mode by the drive. SYNC1 is not supported. If a synchronization problem occurs it is recommended to replace the first slave device with a different device, such as Wago, or Omron EtherCAT junction, otherwise contact Maxon to ensure you have the latest firmware. At the time of testing firmware version 0x150 was shipped with the drive but was updated to 0x161. For proper operation you verify the firmware revision level installed in the drive using their EPOS Studio Windows PC software (Version 3.6 was used for testing).



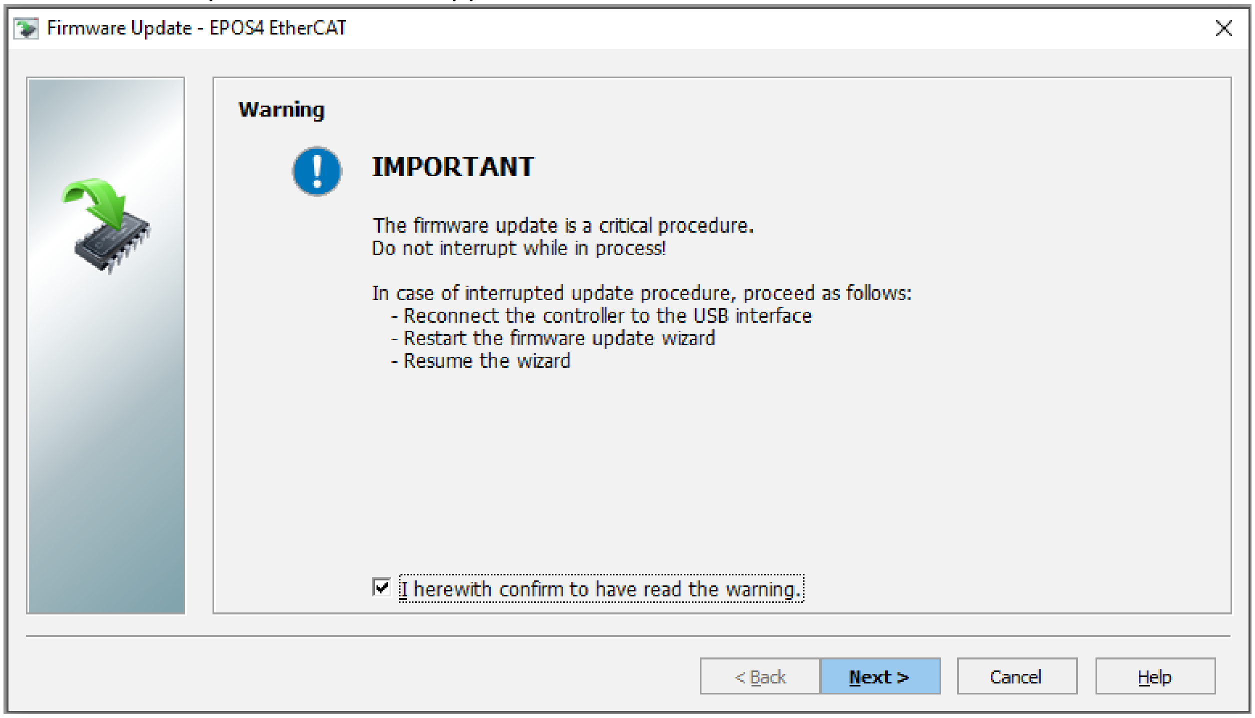


Firmware Upgrade

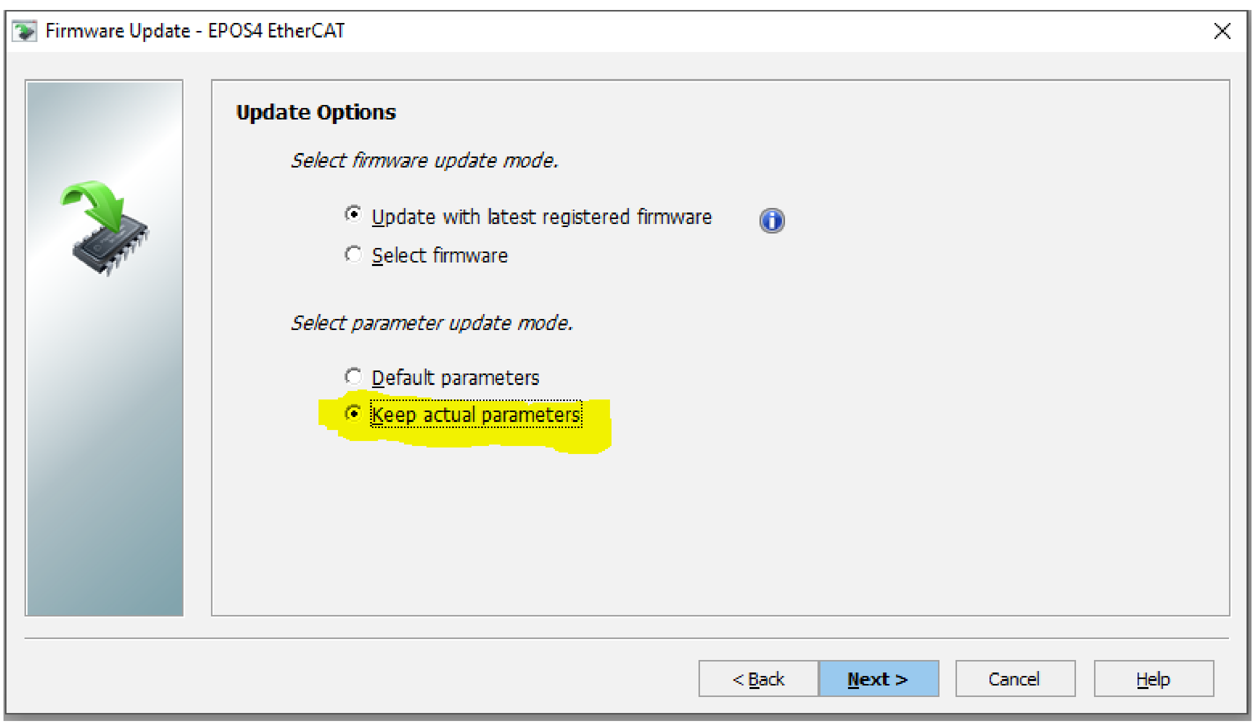
In order to upgrade the firmware of the drive you must be connected to it via a USB cable and running EPOS4 Studio. Select 'Firmware Update' under the Wizard menu selections.



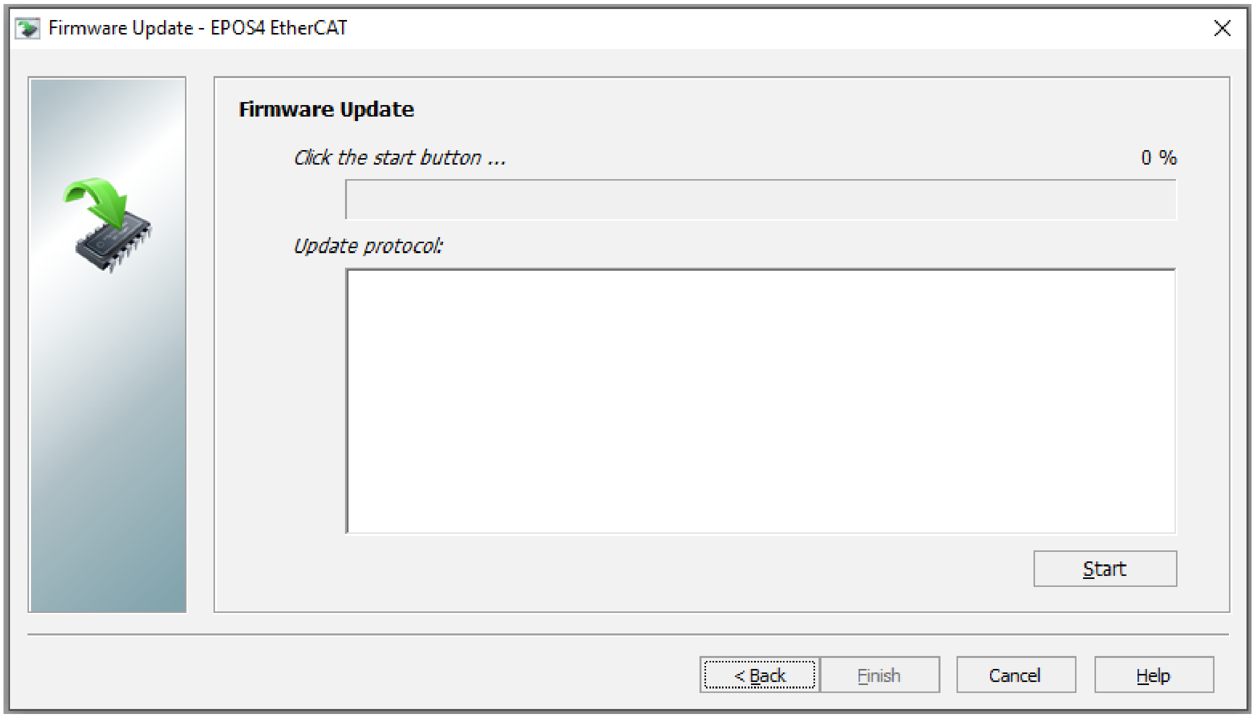
The Firmware update screen will appear:



By default updating the firmware will delete any saved parameters and substitute a default that can cause motor damage due to overcurrent. To avoid this select “Keep actual parameters”, this is highlighted below.

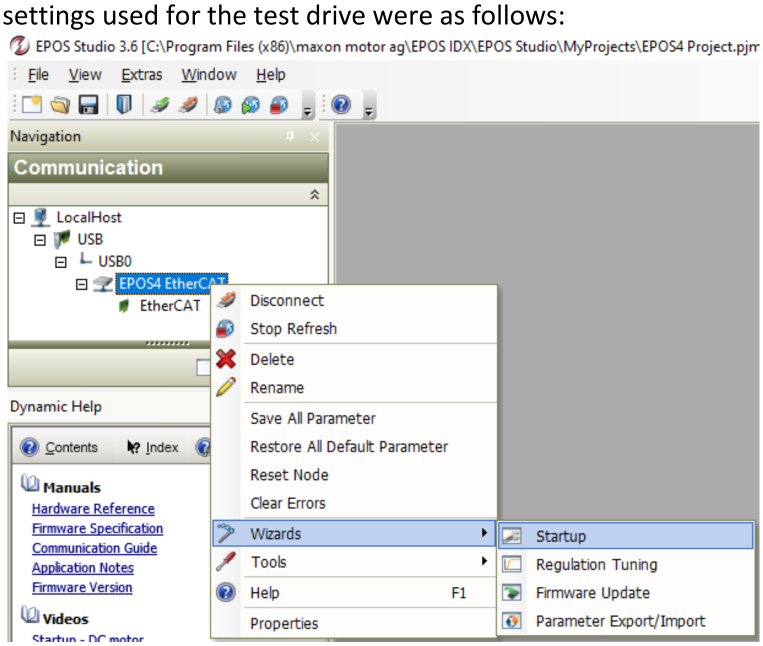


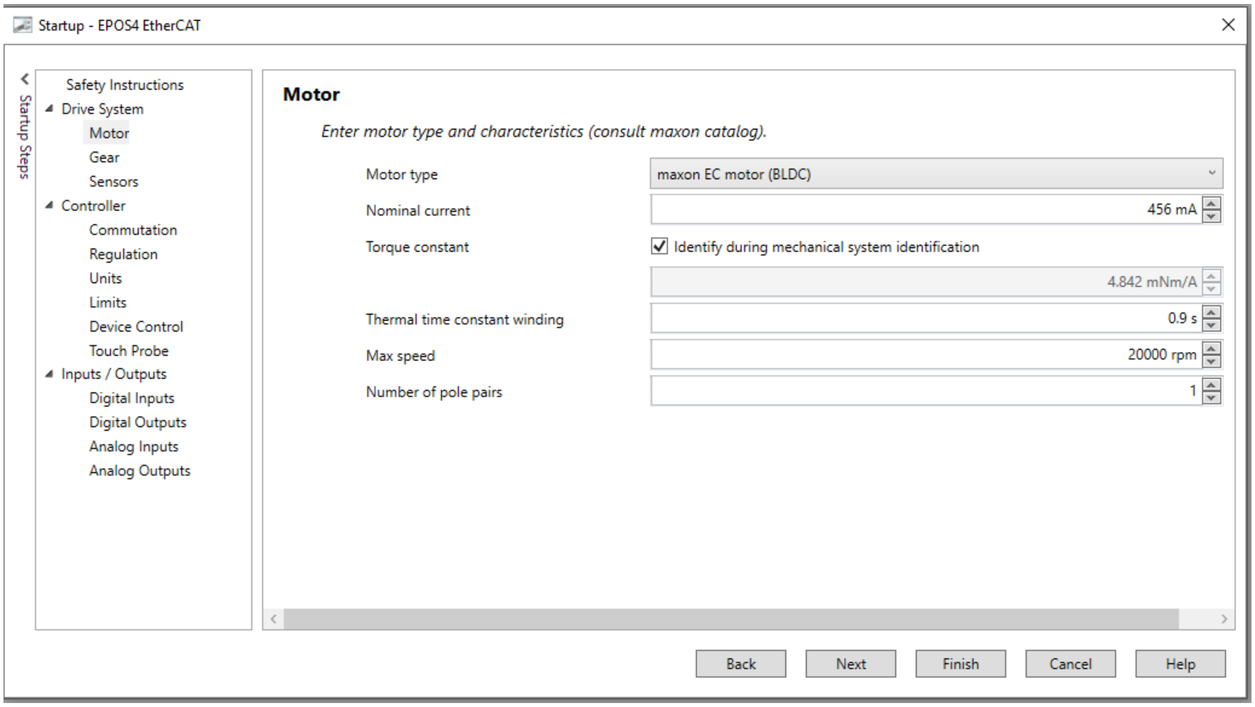
Click the 'Start' button after 'Next' to begin the update:

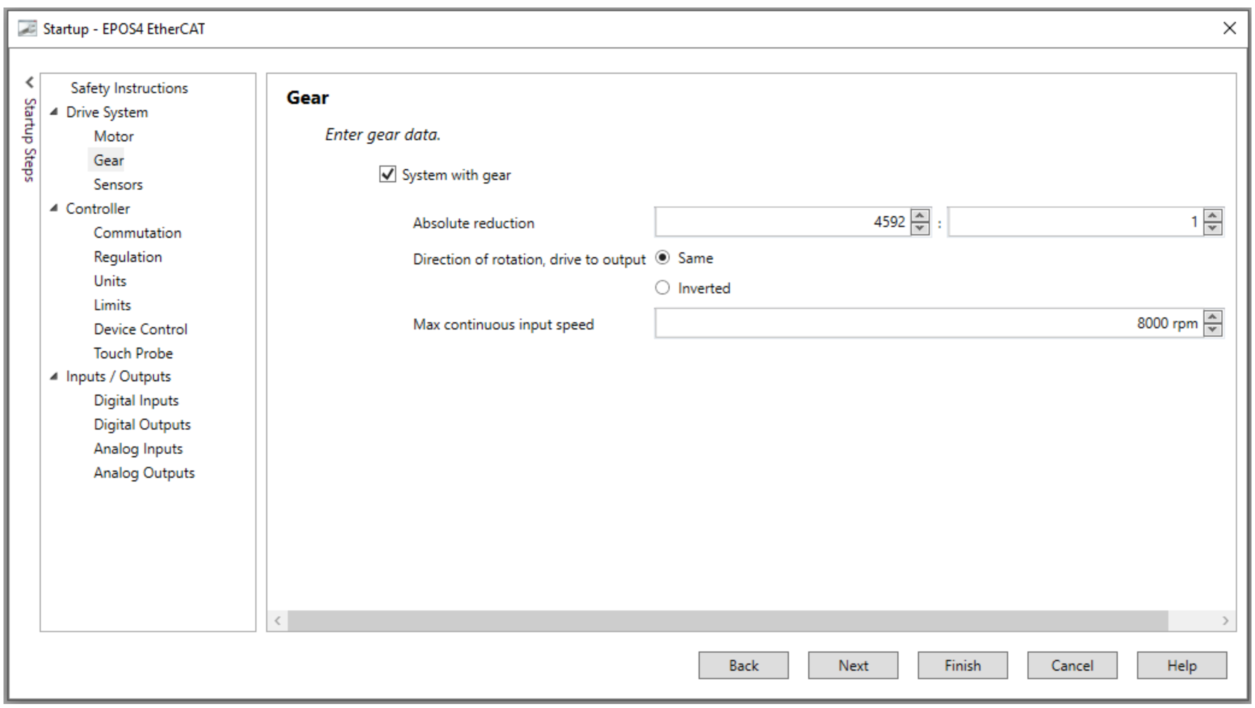


Startup Wizard Settings

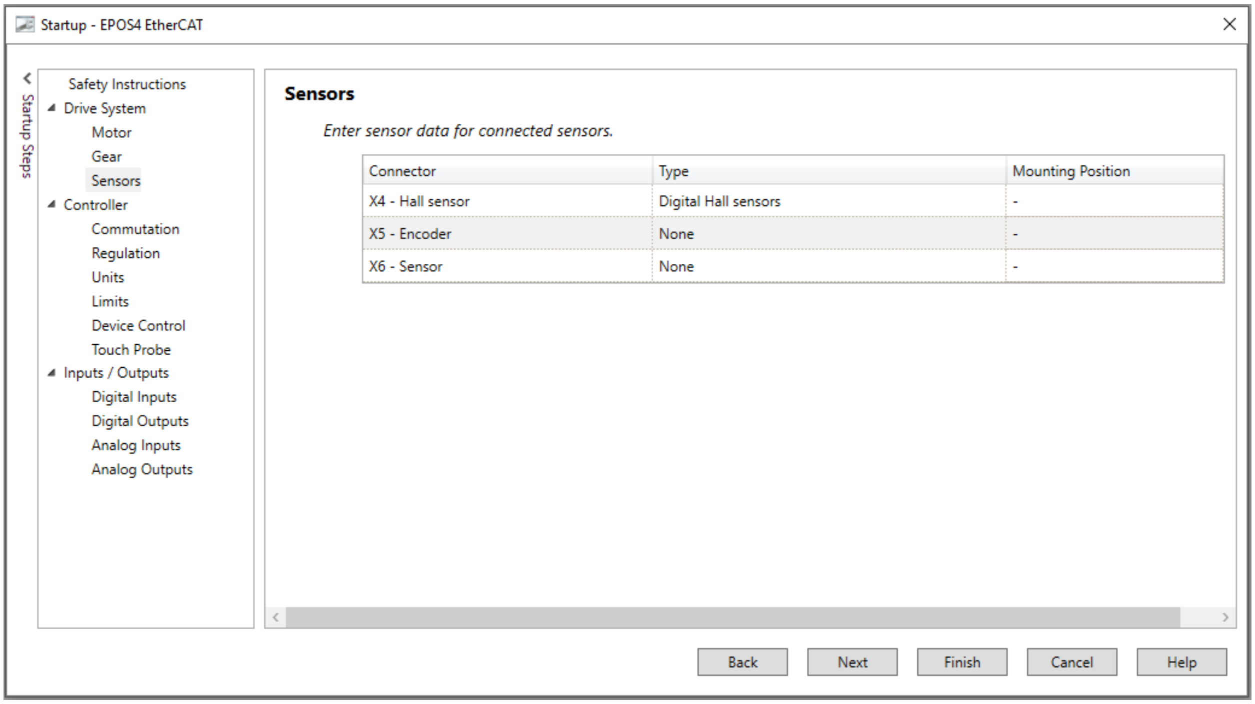
Prior to using a new drive the proper motor parameters must be set and the motor tuned. The data needed for setup can be found in the motor specification sheet. Startup allows you to tell the drive what is available with regards to encoder, current settings, digital and analog IO, etc. Note the velocity units must be set to .1 rpm. The settings used for the test drive were as follows:

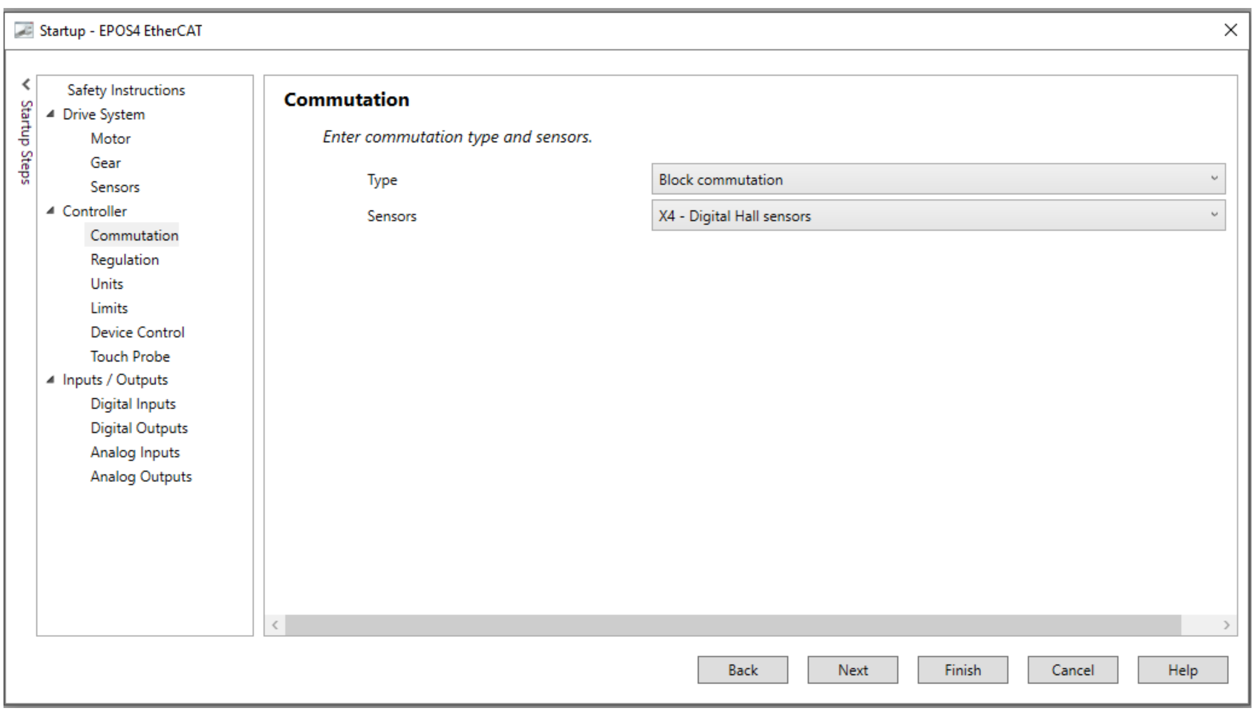


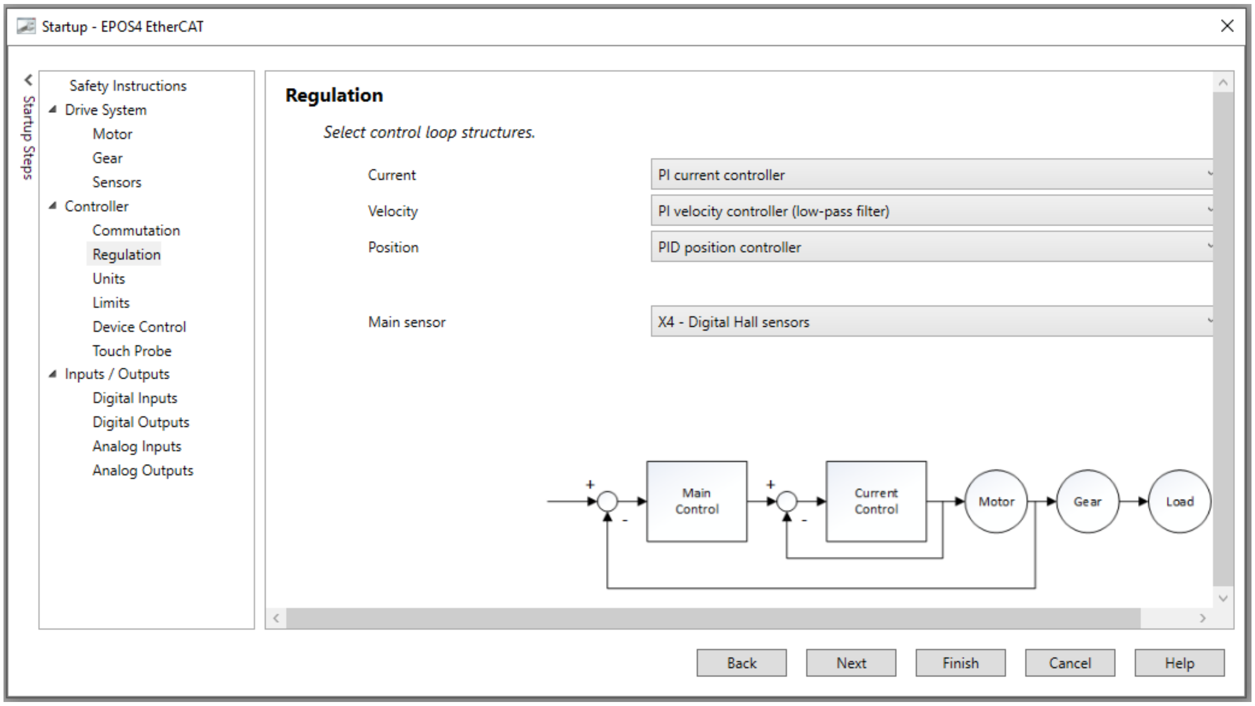

Startup - EPOS4 EtherCAT
Startup Steps
Safety Instructions
Drive System
Motor
Gear
Sensors
Controller
Commutation
Regulation
Units
Limits
Device Control
Touch Probe
Inputs / Outputs
Digital Inputs
Digital Outputs
Analog Inputs
Analog Outputs
Motor
Enter motor type and characteristics (consult maxon catalog).
Motor type
maxon EC motor (BLDC)
Nominal current
456 mA
Torque constant
Identify during mechanical system identification
4.842 mNm/A
Thermal time constant winding
0.9 s
Max speed
20000 rpm
Number of pole pairs
1
Back
Next
Finish
Cancel
Help


Startup - EPOS4 EtherCAT
Startup Steps
Safety Instructions
Drive System
Motor
Gear
Sensors
Controller
Commutation
Regulation
Units
Limits
Device Control
Touch Probe
Inputs / Outputs
Digital Inputs
Digital Outputs
Analog Inputs
Analog Outputs
Gear
Enter gear data.
System with gear
Absolute reduction
4592
:
1
Direction of rotation, drive to output
Same
Inverted
Max continuous input speed
8000 rpm
Back
Next
Finish
Cancel
Help

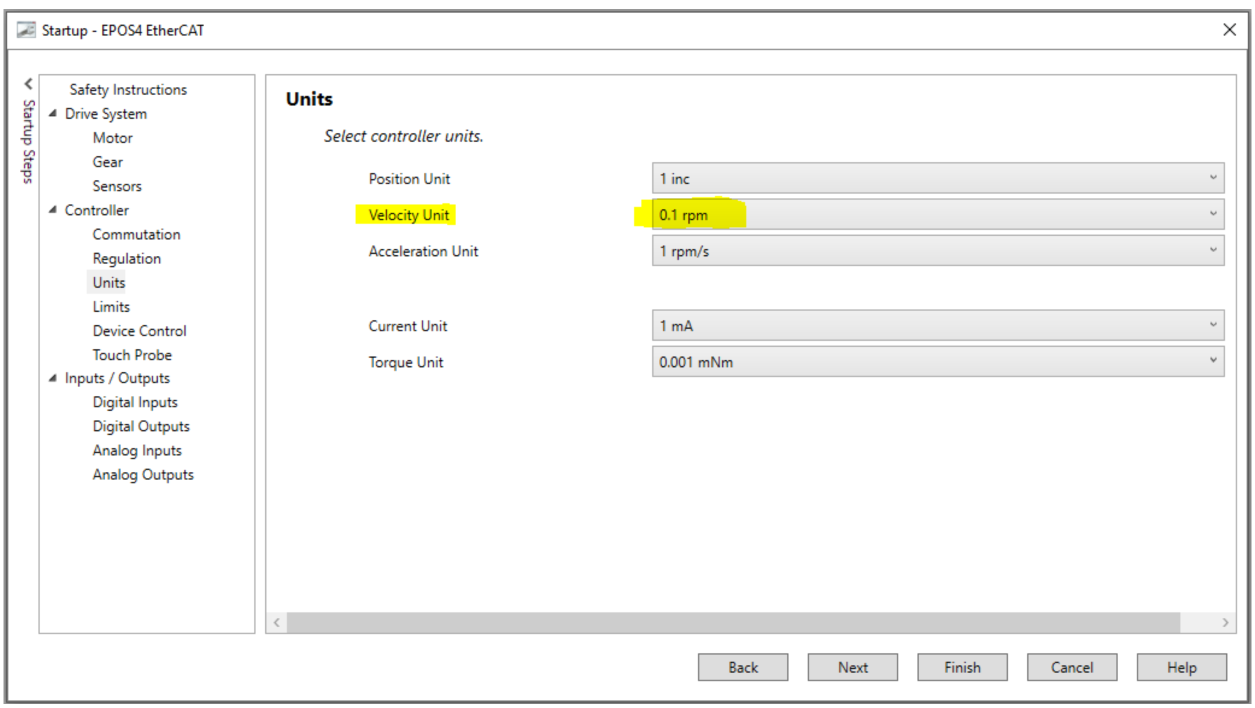
Note that no encoder was available on this drive so the Digital Hall sensors were used yielding a course ppr (pulses per revolution) of 6.



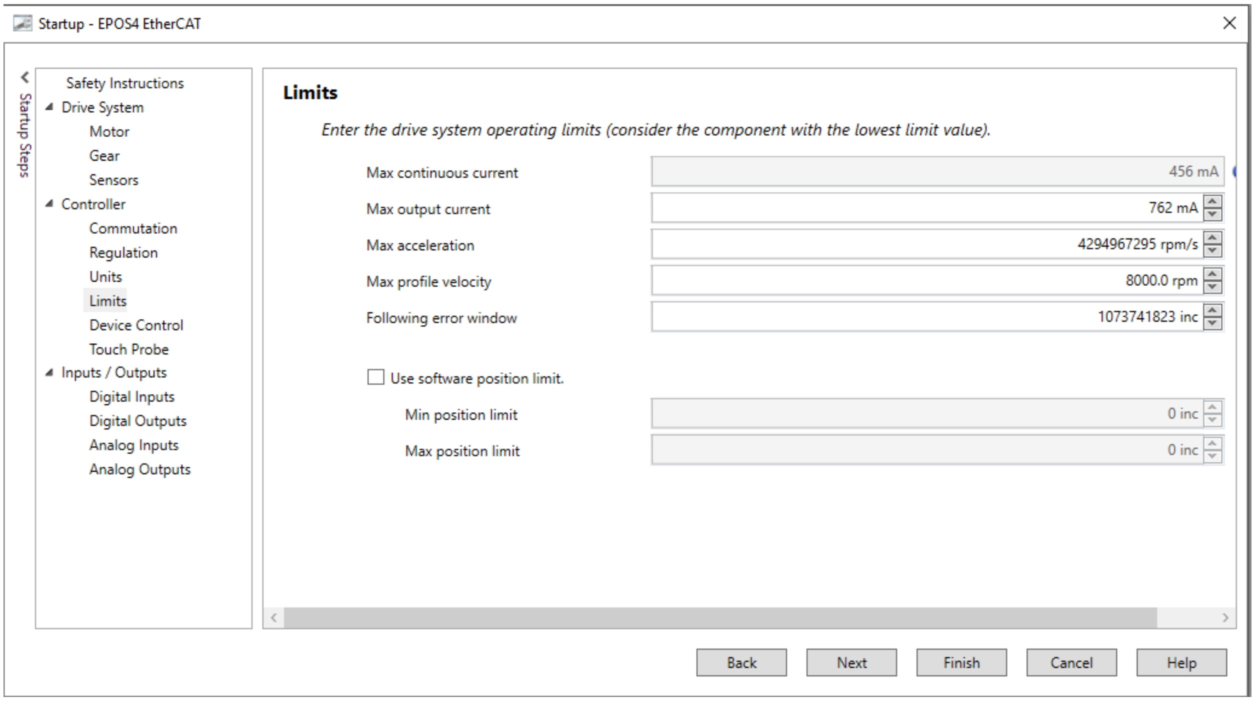


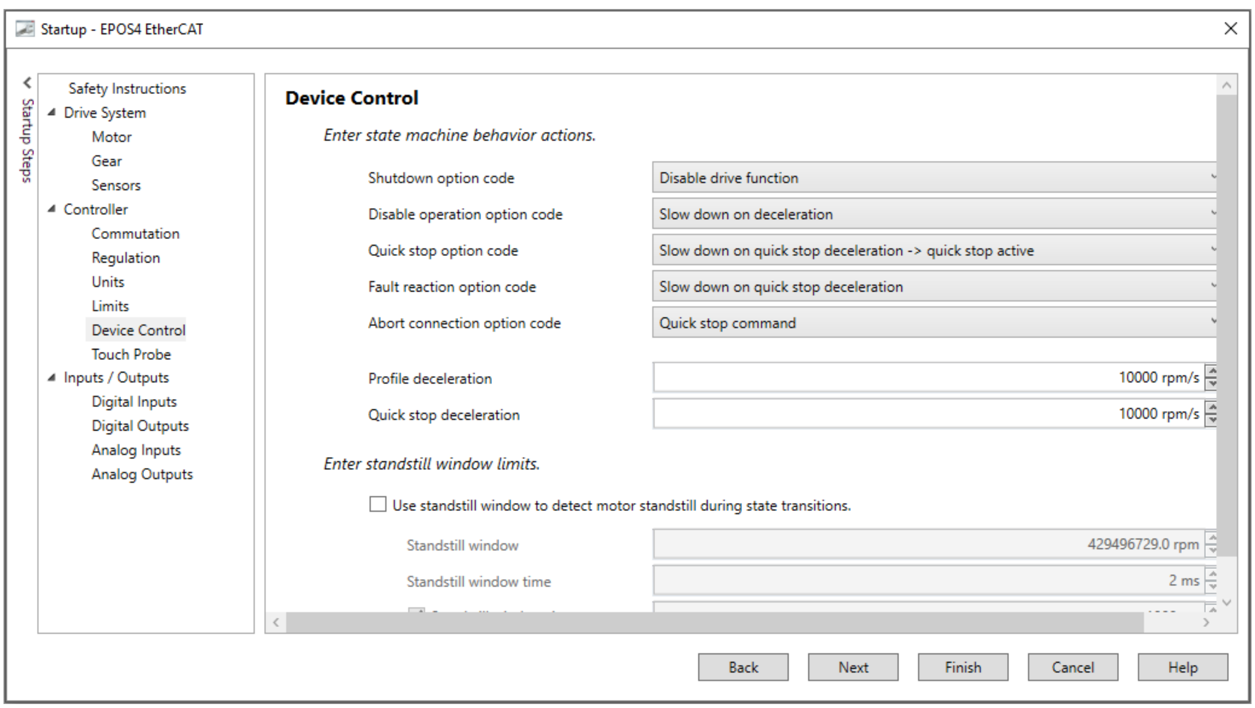


Note that the Velocity units must be set to .1 rpm, the default is 1 rpm:

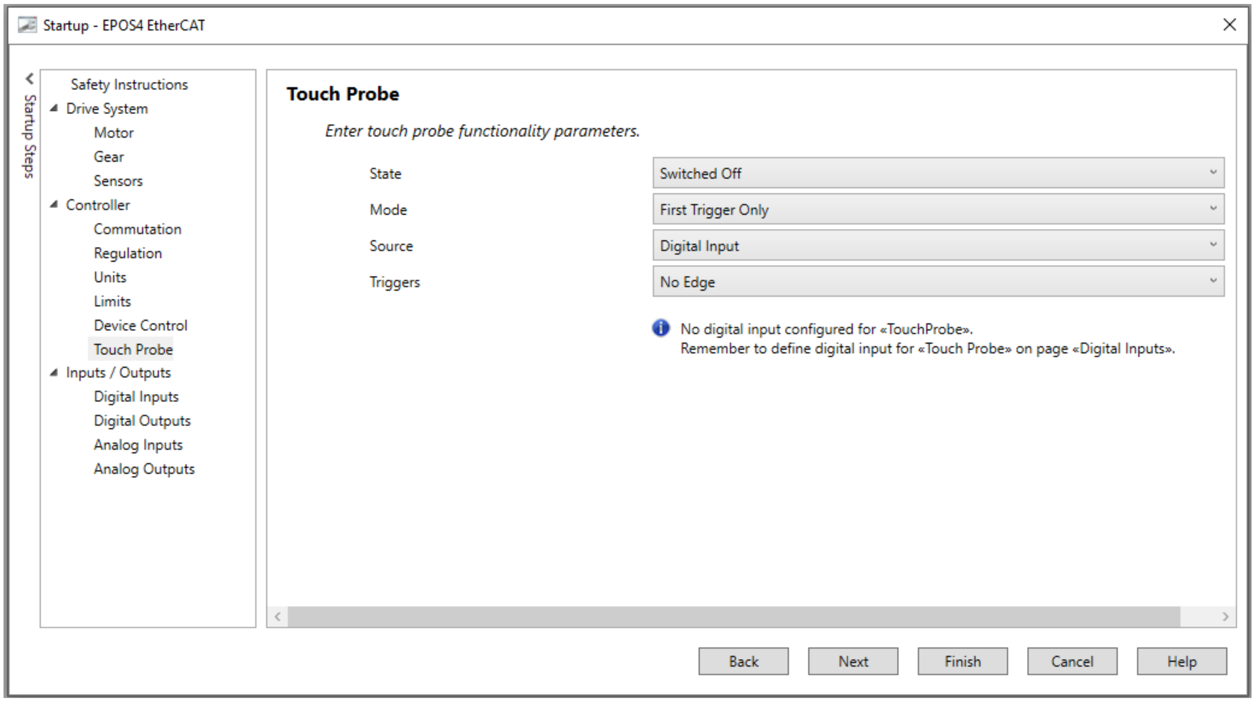


Note the 'Max output current' is listed as the 'Stall current' in the specification at the time of this documentation:

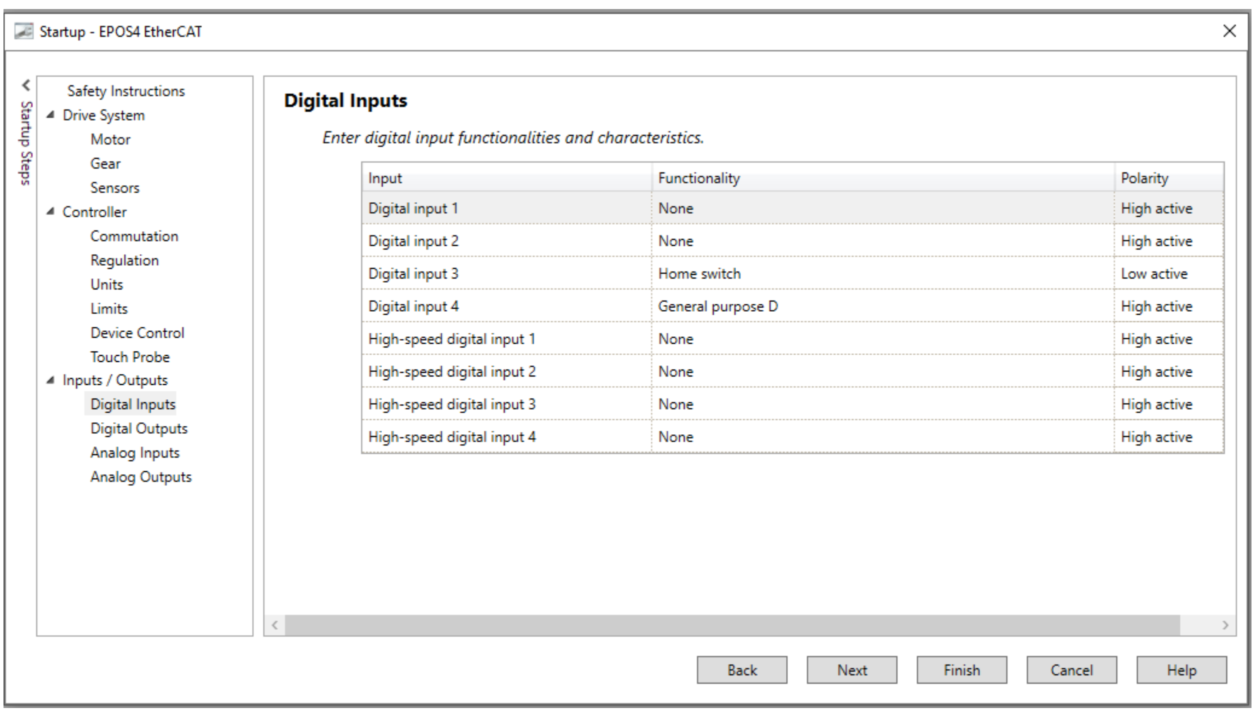




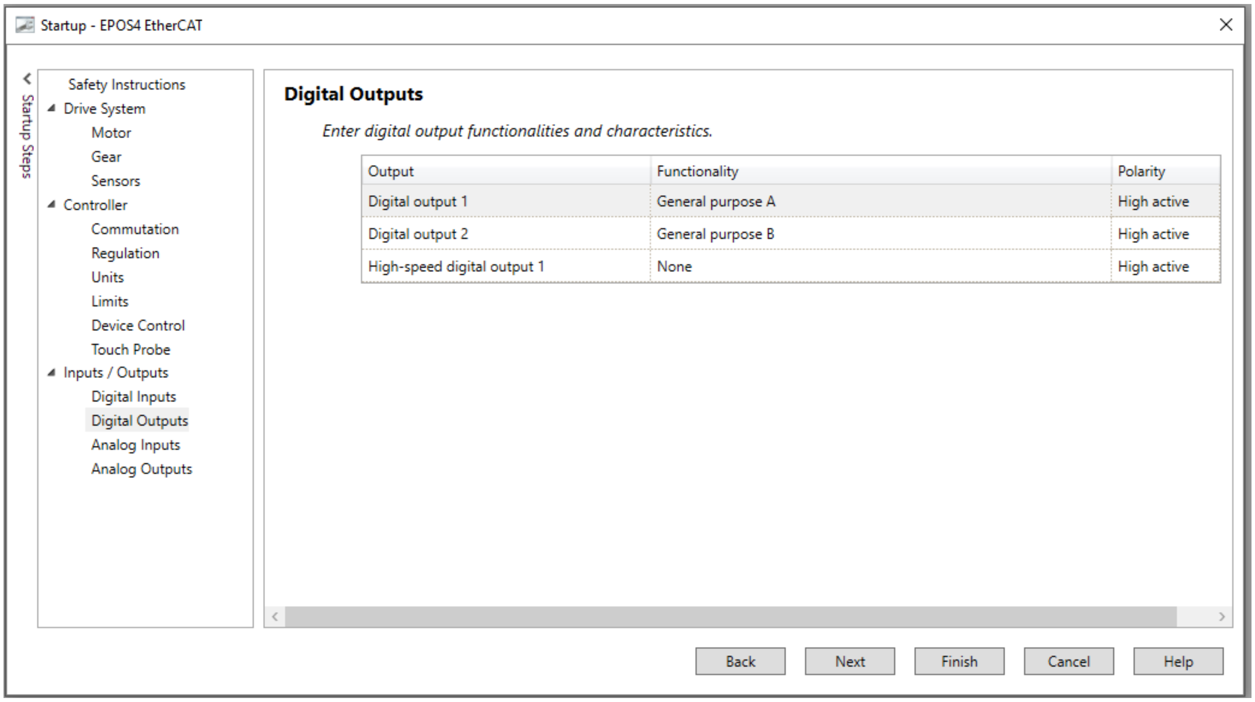
The Touch Probe is not supported:



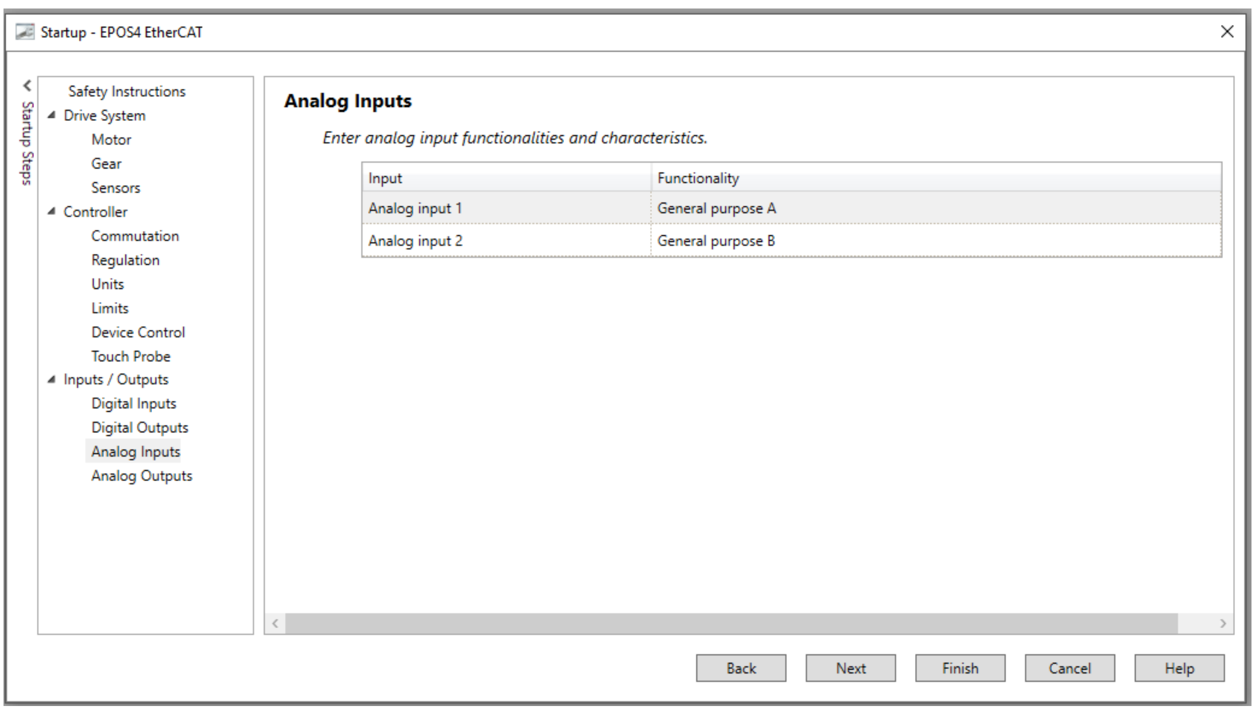
The digital inputs will appear in QuickBuilder as the dins. In this case the default polarity for the Home Switch was inverted for test purposes but can be set to whatever the requirements are:

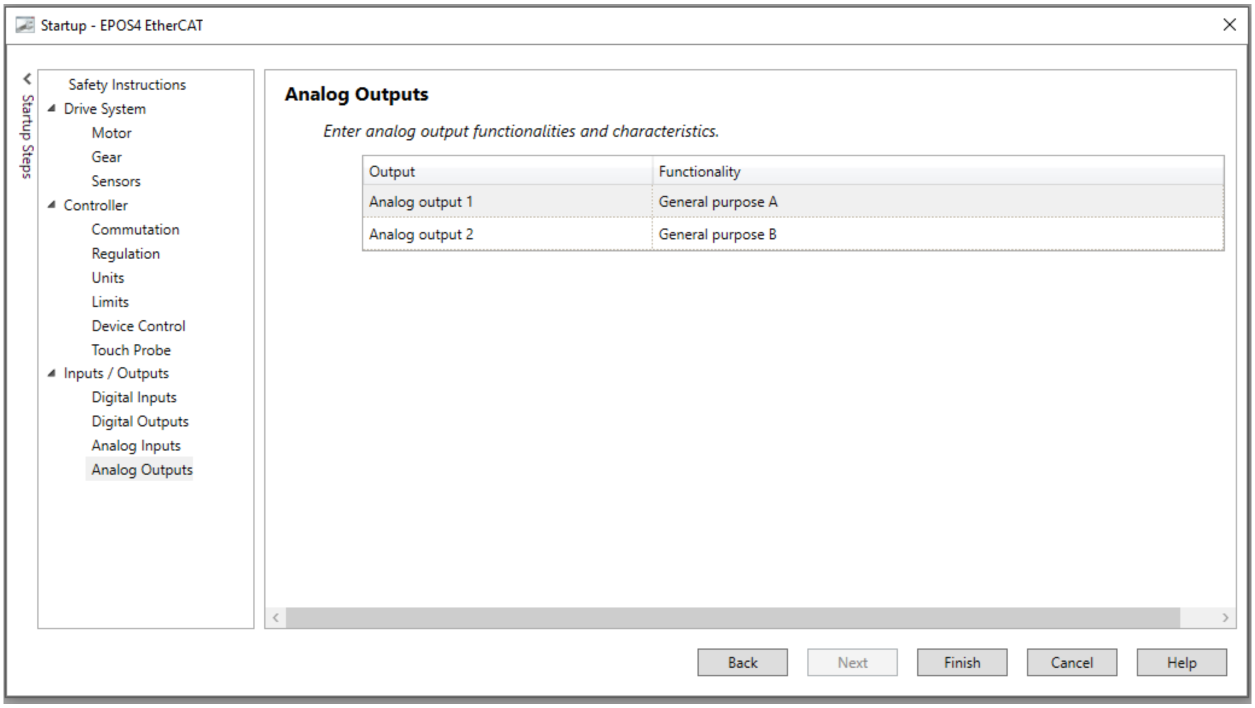


The QuickBuilder drive douts are mapped to the digital outputs:



The Analog inputs and outputs are not supported by QuickBuilder but can be used as needed:

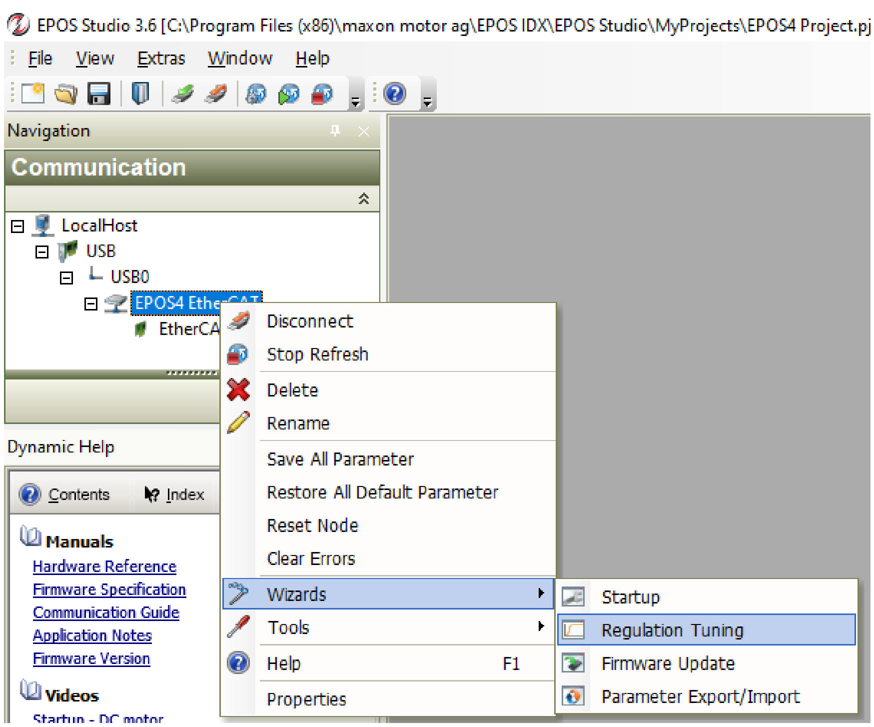




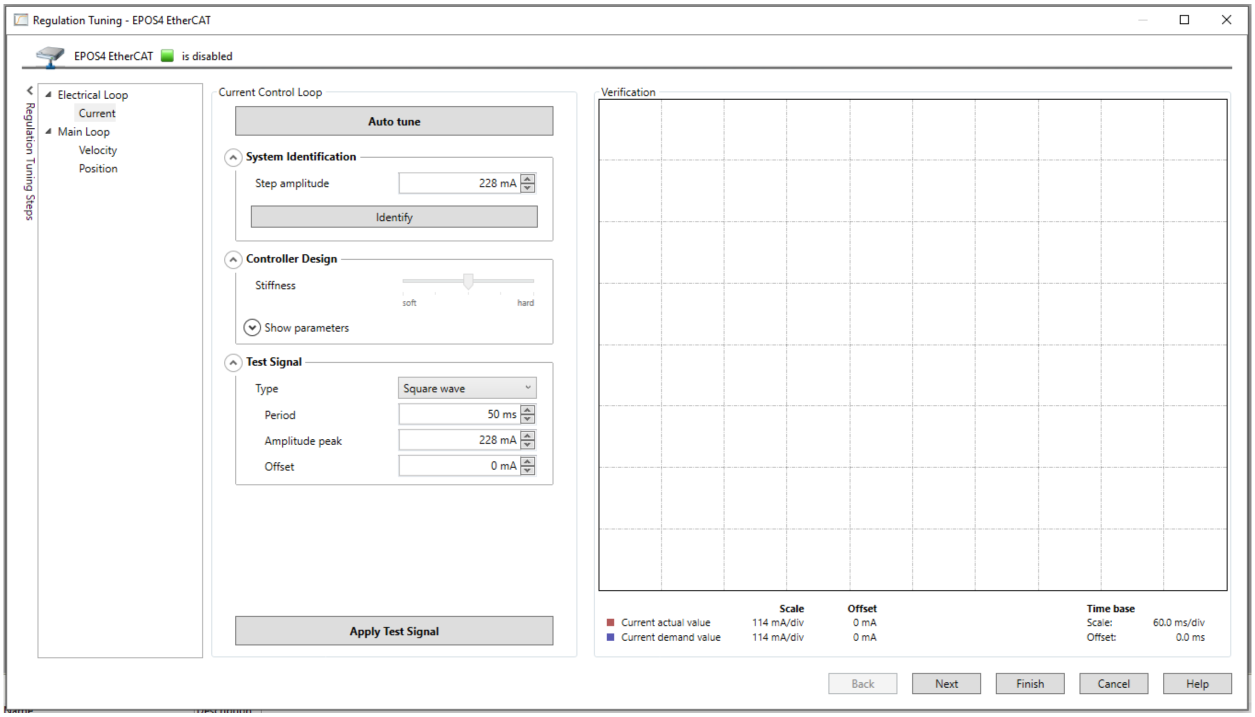
Make sure to save everything to a project before existing the EPOS Studio so it can be used on the next drive that has the same settings.

Tuning

The motor must be tuned prior to use using the Tuning Wizard.



Press “Auto tune”, wait for the tuning to occur and then press “Apply Test Signal” once done. Click ‘Next’ to tune for Velocity and Position following the same procedure.



Appendix

M

[M] MOTIONLINX-AI (Pulseroller)

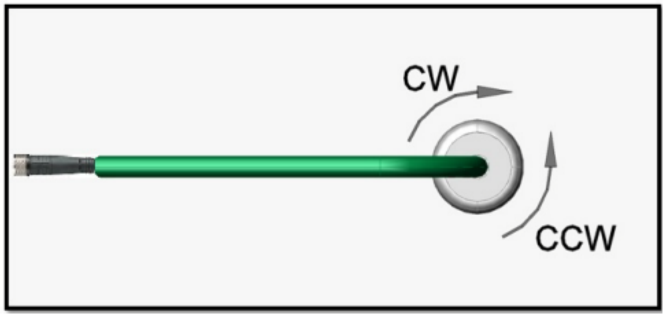
Pulseroller manufactures a number of conveyor controllers; one in particular is the MOTIONLINX-AI. This device can control 2 motors in velocity mode and has 4 inputs. Only ECO Plus motor is supported which is the default configuration.

eCAT_driveType – 17, $DRIVE_MOTIONLINX

Configuration

Only the standard configuration, as received, is currently supported. Individual SDO writes are not supported since this cannot be handled in Operational Mode by the MotionLinx product with revision 1.4 or 1.5 firmware. Future firmware changes may allow configuration changes in Operational Mode so the inclusion of the possibility has been implemented within Incentive through the use of the 'IAI_control' axis property detailed at the end of this section.

CCW is the positive directional move (position increments in a positive amount) and CW is the negative direction as referenced by the position variable fpos. The tested MotionLinx-AI firmware revision was V1.4. EtherCAT configuration file referenced was "Industrial Software MotionLinx-Ai V1_5.xml".



Customization of settings is done using configuration object 0x4000, but is currently fixed at the below values by default (MotionLinx documentation is not correct for SubIndex 2 to 8, SubIndex 2 is actually Debounce and all the following indexes are incremented by one, thus Push-Pull Sensor Type would be SubIndex 3, not 2 as displayed from the manufactures manual below):

SubIndex 1 **Sensor Polarity**	**INT**	Bitwise Value bit 00 ▪ Left Sensor Port – Aux Input bit 01 ▪ Reserved bit 02 ▪ Right Sensor Port – Aux Input bit 03 ▪ Reserved bit 04 ▪ Left Sensor Port – Sensor Input bit 05 ▪ Reserved bit 06 ▪ Right Sensor Port – Sensor Input bit 07 thru bit 15 ▪ Reserved See *Note* ②
SubIndex 2 **Push-Pull Sensor Type**	**INT**	Bitwise Value bit 00 ▪ Left Sensor Port Inputs bit 01 ▪ Right Sensor Port Inputs bit 02 thru bit 15 ▪ Reserved bit ▪ ON: Both Inputs on the port are set to Push-Pull bit ▪ OFF: Both inputs on the port are set to NPN/PNP Auto-Detect
SubIndex 3 **Motor Type Left**	**INT**	Integer value to set the motor type See *Note* ③
SubIndex 4 **Motor Type Right**	**INT**	

Default Values as tested, SubIndex:

4. Sensor Polarity = 0x0000
5. Debounce = 0x14 (20)
6. Push-Pull Sensor Type = 0x0000
7. Motor Type Left = 0x0000 (ECO Plus)
8. Motor Type Right = 0x0000 (ECO Plus)

SubIndex	Data Type	Data Description
SubIndex 5 **Brake Mode Left**	**INT**	Integer Value to set the motor braking method See Note ④
SubIndex 6 **Brake Mode Right**	**INT**	
SubIndex 7 **Motor Speed Left**	**INT**	Integer value to set the motor speed • For MDR the value is in mm/sec • For PGD the value is in motor RPM
SubIndex 8 **Motor Speed Right**	**INT**	

NOTE ②

In this SDO register, when a given bit is set to 1; the logical state of the corresponding bit in the All Sensors object of PDO0 is inverted. This also inverts the LED state displayed on the module.
This is provided as a convenience for the master controller programmer. For example, the master controller could have re-useable code or routines that expect a photo sensor to have its electrical signal ON when it is "blocked". Let's say on a given *MotionLinx-Ai* module, you need to connect a photo sensor whose electrical signal is OFF when "blocked". Instead of modifying the program logic in each and every place this input is used; you can simply change its logical polarity with this SDO register. This logical polarity setting also affects the LED state.

For example, by default when the Left Sensor input is electrically energized, its corresponding LED illuminates green and bit 4 of the All Sensors PDO register is a 1. If you set bit 4 in the SDO Sensor Polarity register; when the Left Sensor input is electrically energized, bit 4 of the All Sensors PDO is 0 and its corresponding LED is off.

NOTE ③

The values for motor type are as follows

Value	Motor Type	Description
0	ECO Plus	See *Appendix C – Power Supply Loading* on page 47 for details on Motor Types
1	ECO	
2	Boost	
3	Boost 8	

NOTE ④

The following table defines the MDR Braking Methods available:

Value	Method	Description
0	**Normal**	Standard Dynamic braking - MDR power circuit in *MotionLinx-Ai* is internally connected during motor stop sequence to provide backward energy to bring rotor to a stop. When *MotionLinx-Ai* has detected that the motor has stopped; all winding current is shut off from the motor windings. This is the industry standard braking method and is the default factory setting for all *MotionLinx-Ai* modules
1	**Free**	MDR power circuit in *MotionLinx-Ai* is internally disconnected to allow rotor to "free spin" until its mechanical load brings it to a stop.
2	**Servo Brake**	When a motor is commanded to stop; the *MotionLinx-Ai* utilizes the motor's Hall Effect sensors to determine the position of the rotor and will inject current into the motor windings to maintain rotor position.

Please note that for Servo Brake modes, the motor circuitry supplies power to the motor to keep it in position. The more torque required to hold the motor rotor position will result in more current being supplied. Prolonged braking at higher torque values can result in motor over current and/or over heating conditions.

Default Values as tested, SubIndex:

9. Brake Mode Left = 0x0000 (Normal, standard dynamic braking)
10. Brake Mode Right = 0x0000 (Normal, standard dynamic braking)
11. Motor Speed Left = 0x320 (800) mm/sec (MDR mode)
12. Motor Speed Right = 0x320 (800) mm/sec (MDR mode)

The 'IAI_control' axis property contains some of the above configuration information that was the present settings during system initialization. If you wish to change that configuration prior to enabling the drive you may modify 'IAI_control' and the appropriate SDO writes will be initiated to the MotionLinx device. The bits of 'IAI_control' map as is detailed below:

Bits 0 - 7	Debounce in milliseconds
Bits 8 - 9	Motor_type
Bits 10 - 11	Brake_mode
Bits 12 - 13	Pin 2 is input or output, '0' default being input, '1' left motor pin 2 is output, '2' right motor output, '3' both pin 2's are outputs. When set axis property 'output' bit 0 will turn the output on/off.

Directional Detection

There are two ways to determine what direction the motors are currently rotating. The first is direct from the motor status bits and represented by bits 8 and 9 of the wStatus property for the axis (under diagnostics in watch window)

Motor Status bit 0 and bit 1		
Bit 1	*Bit 0*	*Description*
0	0	Motor not running, standard or servo braking applied
0	1	Motor running in CCW Direction
1	0	Motor running in CW Direction
1	1	Motor not running and no braking applied (free to spin)

Thus bit 0 above is bit 8 of wStatus, bit 1 is bit 0. Typically you will see it change between 0x0100 (CCW) and 0x0200 (CW).

Other method is looking at your velocity, this does not represent information directly from the motor (since we have no feedback) but does show what you commanded. Direction is changed by positive (CCW) or negative (CW) velocity.

Data Input Mapping

Data inputs of the drive are mapped as follows:

```
[System Diagnostics Bits 24 to 31][Diagnostic Bits 23 to 16][All Sensors Bits 15 to 0]
```

SubIndex 1 All Sensors	Index 0x3000 SubIndex 0	INT	Bitwise Value - Read only bit 00 ▪ Left Sensor Port – Aux Input bit 01 ▪ Reserved bit 02 ▪ Right Sensor Port – Aux Input bit 03 ▪ Reserved bit 04 ▪ Left Sensor Port – Sensor Input bit 05 ▪ Reserved bit 06 ▪ Right Sensor Port – Sensor Input bit 07 thru bit 14 ▪ Reserved bit 15 ▪ 2 sec on / 2 sec off heartbeat See *Appendix B - IOX-2Breakout Module* for details on how to connect to Aux input

SubIndex 1 Module Diagnostic Left Motor	Index 0x3001 SubIndex 0	INT	Bitwise Value – See *Note* ①	
SubIndex 2 Module Diagnostic Right Motor	Index 0x3005 SubIndex 0	INT	bit 00 ▪ Motor Status bit 01 ▪ Motor Status bit 02 ▪ Port in Digital Mode bit 03 ▪ Reserved bit 04 ▪ Reserved bit 05 ▪ Board Overheat bit 06 ▪ Over-Voltage bit 07 ▪ Low Voltage	bit 08 ▪ Overheated bit 09 ▪ Over Current bit 10 ▪ Short Circuit bit 11 ▪ Motor Not Connected bit 12 ▪ Overloaded bit 13 ▪ Motor Stalled bit 14 ▪ Hall Sensor Error bit 15 ▪ Motor Not Used

NOTE ①

Bits 0 and 1 are used in combination to provide 4 possible states for motor status. The following chart defines the bit values for these states:

Motor Status bit 0 and bit 1		
Bit 1	*Bit 0*	*Description*
0	0	Motor not running, standard or servo braking applied
0	1	Motor running in CCW Direction
1	0	Motor running in CW Direction
1	1	Motor not running and no braking applied (free to spin)

System Diagnostics bits:

Bit 0 – Set if logic voltage below 14V
Bit 1 – Set if speed for left MDR is higher than the max speed of that MDR.
Bit 2 – Set if speed for the left MDR is lower than the lowest speed of that MDR.
Bit 3 – Set if speed for right MDR is higher than the max speed of that MDR.
Bit 4 – Set if speed for the right MDR is lower than the lowest speed of that MDR.

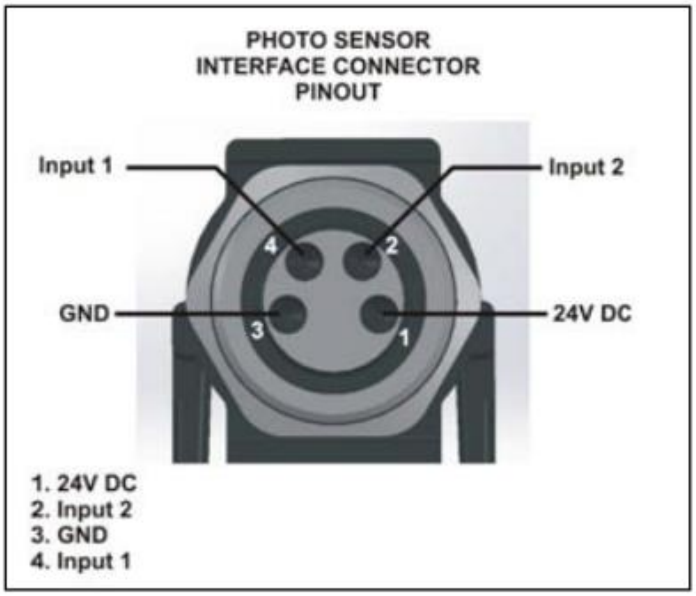
Sensor Port

Inputs	• PNP/NPN Auto-sensing (Default) • 4 per module - 2 inputs per sensor port • Programmable Push-Pull selectable
Minimum ON current	1.5mA
Maximum OFF current	0.4mA

Sensor port inputs are auto-sensing for the connected circuit type of either PNP or NPN. Please note that both sourcing and sinking current will activate the input.

Both Left and Right Sensor ports are protected by a single 200mA resettable fuse. Combined current between Left and Right ports cannot exceed 200mA.

Both Left and Right Sensor ports utilize load detection circuits between the +24V (Pin 1) and Gnd (Pin 3). This circuit provides input to the processor such that sensor detected status can be known. Short circuit between Pin 1 and Pin 3 can cause damage to the load detection circuit and render this function inoperable.



Power Connections

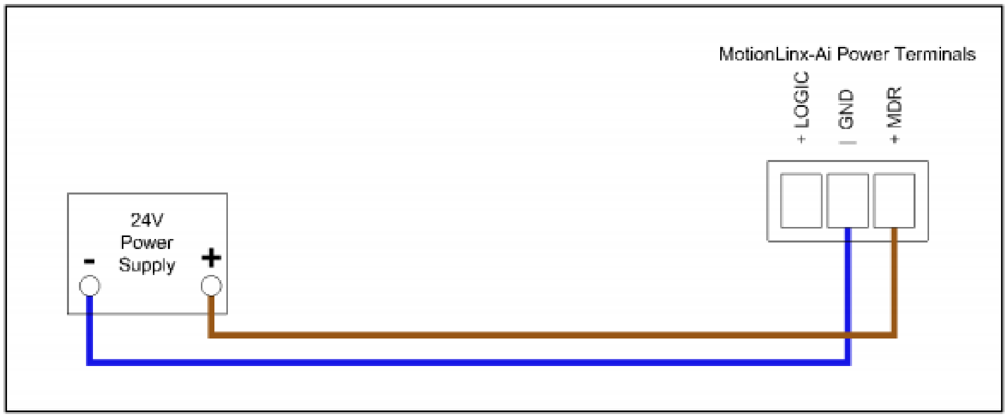


FIGURE 14 - CONNECTION FOR SINGLE MDR AND LOGIC POWER SUPPLY

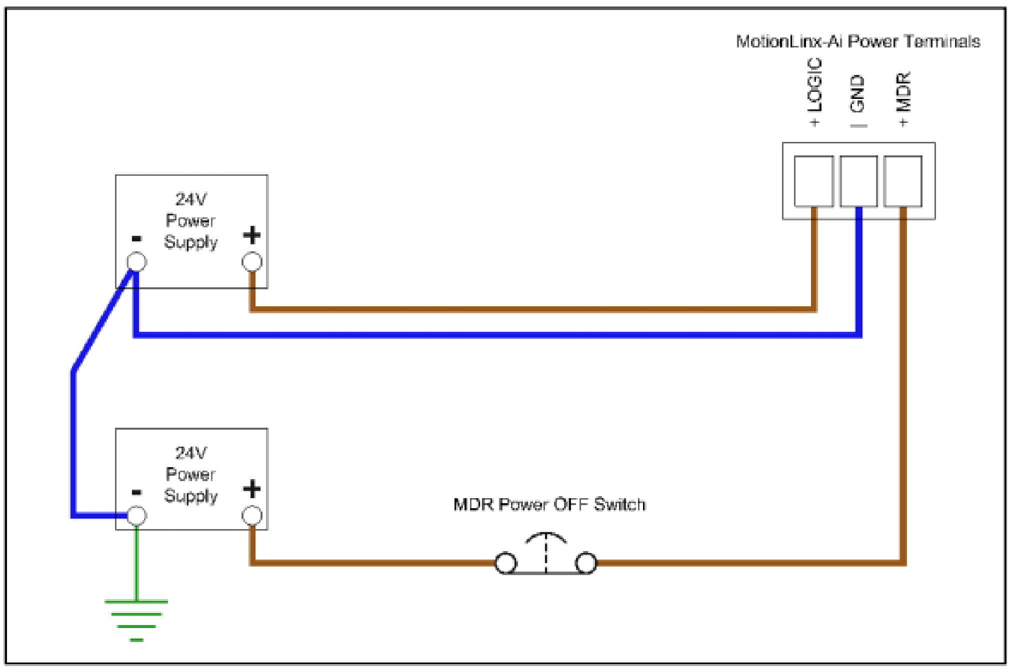


FIGURE 15 - TYPICAL CONNECTION FOR SEPARATE MDR AND LOGIC POWER SUPPLIES

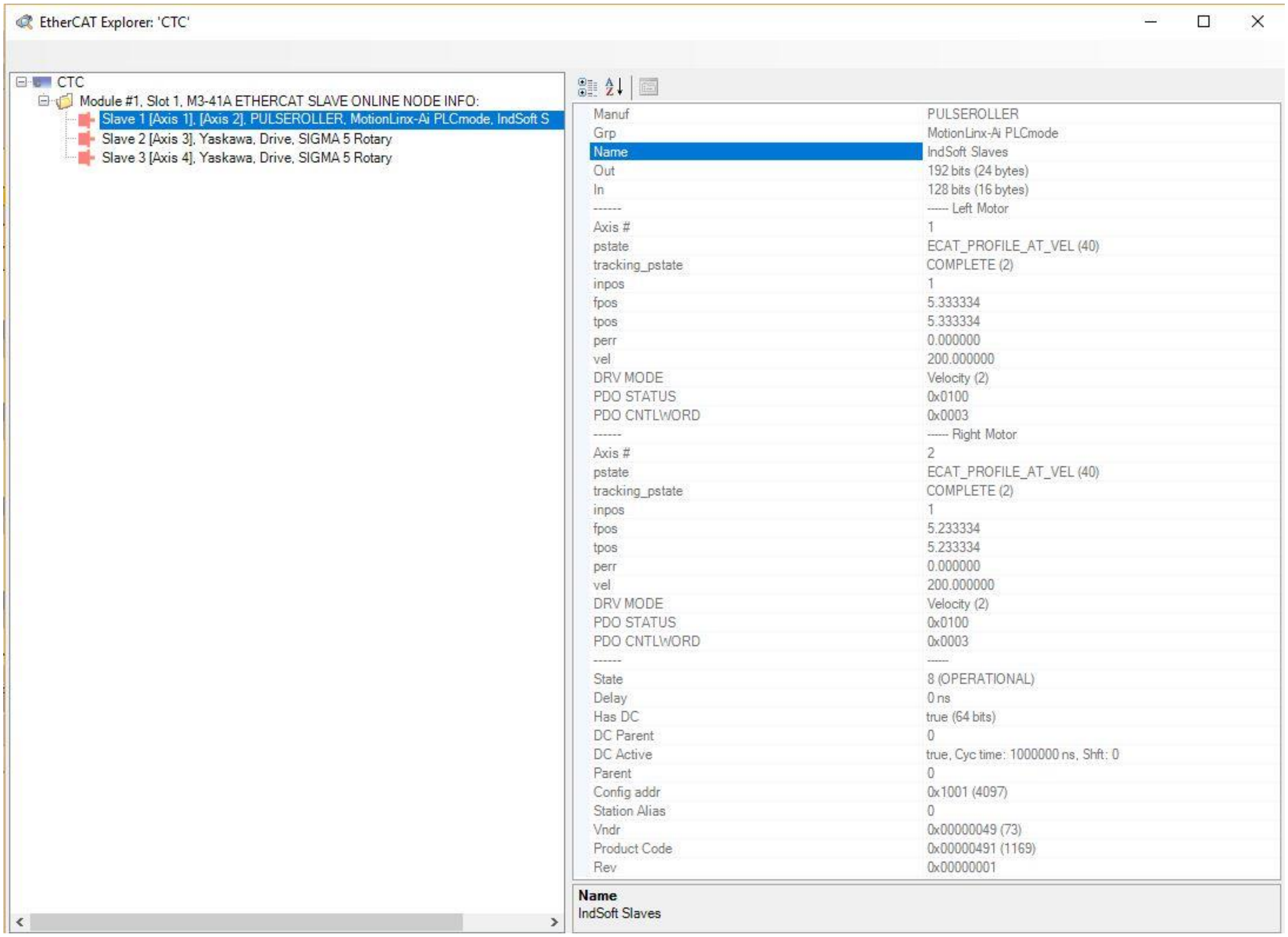
	ECO	*ECO Plus*	*Boost*	*Boost 8*
Power supply load per Motor Port at rated torque at maximum speed	2.5 A	2.5 A	3.5 A	3.5 A
Power supply load per Motor Port during motor starting period	3.0 A	4.1 A	5.0 A	8.0 A
Duration of motor starting period	5.0 sec.	No time limit	1.5 sec	3.0 sec

Programming Example

The below example will obtain a speed of 200 for about 8 seconds, then reverse direction at 200 and then stop at a position of ‘fpos’ + 300 using the stop command.

```
acc = 50;
dec = 10;
cmode = $VELOCITY_MODE;  // Set type of motion control to do
ever = 0;  // Assign 0 to a variable to make code more readable

[begin]
move at 200 for ever;
wait for in position;  // Wait till at velocity if VELOCITY_MODE, with
MOTIONLINX we have no velocity feedback but know is moving...
delay 8000 ms;
move at -200 for ever;     // Now go the opposite direction
wait for in position;
delay 6000 ms;
stop;
wait for in position;  // wait for motor to fully stop
savefpos = fpos;
targetx = fpos + 300-dec;  // This will stop at exactly fpos + 300 since dec is
set to 10.
move at 10 for ever;  // Move slowly so can stop it when at targetx
[loop]
if fpos >= targetx goto haltit;
goto loop;
[haltit]
stop;   // Issue a stop command since at targetx
wait for in position;  // Wait for command to take effect and motion to stop
[stall]
delay 10 ms;
goto stall;
```

EtherCAT Explorer View



Blank

Appendix
N

[N] Parker Hannifin

Parker Hannifin manufactures a number servo drives. Incentive currently supports the PD-xxC series of drives. This section provides information that may be specific to this manufacturer. V1.02 Firmware was used for testing with a PD-04C Drive.

I/O

Inputs and Outputs of the drive are mapped into the incentive variables din and dout as follows:

Inputs

Parker Hannifin Manual Assignment:

Bit	Details
0	NOT (negative limit switch)
1	POT (positive limit switch)
2	HOME (origin sensor input)
3 to 15	Reserved
16	DI #1 (I/O pin 2), 0: Open, 1: Close
17	DI #2 (I/O pin 3), 0: Open, 1: Close
18	DI #3 (I/O pin 4), 0: Open, 1: Close
19	DI #4 (I/O pin 5), 0: Open, 1: Close
20	DI #5(I/O pin 13), 0:Open, 1:Close
21	DI #6(I/O pin 14), 0:Open, 1:Close
22	DI #7(I/O pin 9), 0:Open, 1:Close
23	DI #8(I/O pin 10), 0:Open, 1:Close
24~30	Reserved
31	STO (Safe Torque Off), 0: Close, 1: Open

Incentive remapped Assignment:

Bit 0 - NOT (negative limit Switch)
Bit 1 - POT (positive limit Switch)
Bit 2 - HOME (origin sensor input)
Bit 3- DI #1 (I/O pin 2) 0:Open, 1:Close
Bit 4 - DI #2 (I/O pin 3)
Bit 5 - DI #3 (I/O pin 4)
Bit 6 - DI #4 (I/O pin 5)
Bit 7 - DI #5 (I/O pin 13)
Bit 8 - DI #6 (I/O pin 14)
Bit 9 - DI #7 (I/O pin 9)
Bit 10 - DI #8 (I/O pin 10)
Bit 11 to 17 Reserved
Bit 18 -STO (Safe Torque Off), 0:Close, 1 Open

Outputs

Parker Hannifin Manual Assignment:

Bit	Details
0 to 15	Reserved
16	Forced output (0: OFF, 1: ON) of DO #1 (I/O pins 1 and 2) Provided that the relevant bit mask (0x60FE:02.16) is set to 1.
17	Forced output (0: OFF, 1: ON) of DO #2 (I/O pins 17 and 18) Provided that the relevant bit mask (0x60FE:02.17) is set to 1.
18	Forced output (0: OFF, 1: ON) of DO #3 (I/O pins 3 and 4) Provided that the relevant bit mask (0x60FE:02.18) is set to 1.
19	Forced output (0: OFF, 1: ON) of DO #4 (I/O pins 19 and 20) Provided that the relevant bit mask (0x60FE:02.19) is set to 1.
20 to 23	Reserved
24	Output status of DO #1 (0: OFF, 1: ON)
25	Output status of DO #2 (0: OFF, 1: ON)
26	Output status of DO #3 (0: OFF, 1: ON)
27	Output status of DO #4 (0: OFF, 1: ON)
28 to 31	Reserved

Incentive Remapped Assignment:

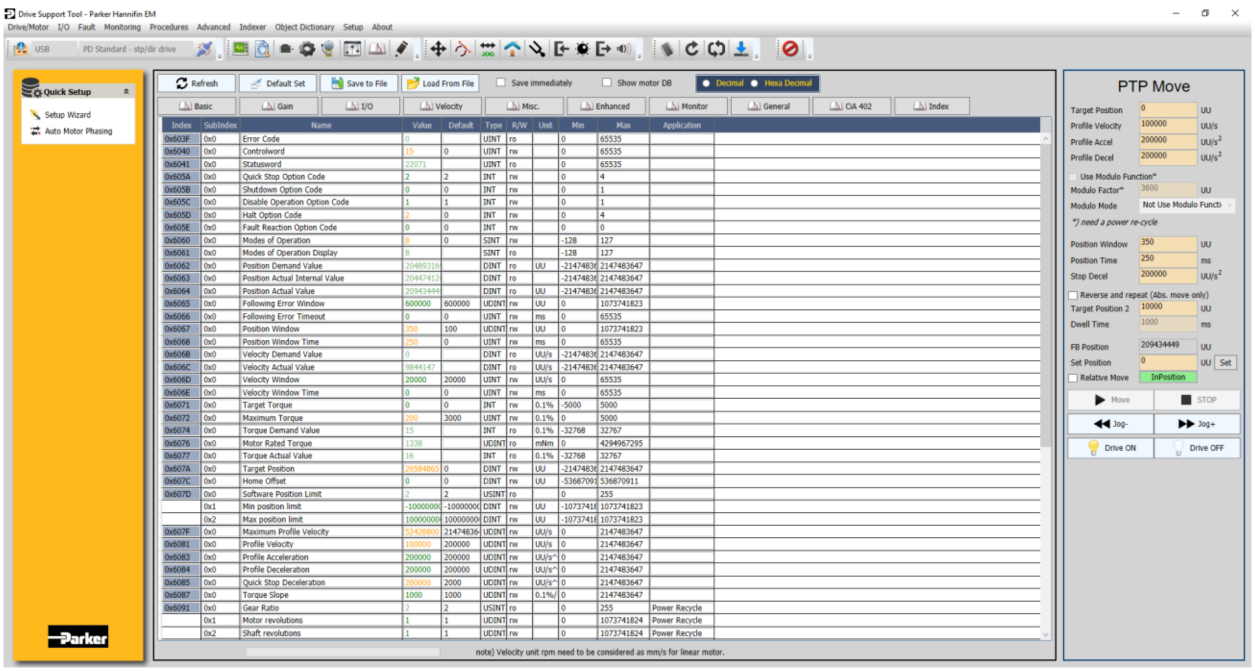
Bit 0 – DO #1
Bit 1 – DO #2
Bit 2 – DO #3
Bit 3 – DO #4

Drive Support Tool

The Parker Hannifin Drive support tool can be downloaded from their website and connected to the drive via USB for extern configuration and monitoring. It is also useful to jog the drive while offline to ensure wiring is correct.

https://www.parkermotion.com/products/Servo_Drives__7319__30_32_80_567_29.html

http://divapps.parker.com/divapps/emn/download/Drives/Drive_Support_Tool_0.85.0_64bit.zip



It is useful for monitoring various EtherCAT objects during operation:

Refresh | Default Set | Save to File | Load From File | Save immediately | Show motor DB

Basic | Gain | I/O | Velocity | Misc. | Enhanced | Monitor

Index	SubIndex	Name	Value	Default	Type	R/W	Unit	Min	Max	Application
0x603F	0x0	Error Code	0		UINT	ro		0	65535	
0x6040	0x0	Controlword	15	0	UINT	rw		0	65535	
0x6041	0x0	Statusword	22071		UINT	ro		0	65535	
0x605A	0x0	Quick Stop Option Code	2	2	INT	rw		0	4	
0x605B	0x0	Shutdown Option Code	0	0	INT	rw		0	1	
0x605C	0x0	Disable Operation Option Code	1	1	INT	rw		0	1	
0x605D	0x0	Halt Option Code	2	0	INT	rw		0	4	
0x605E	0x0	Fault Reaction Option Code	0	0	INT	rw		0	0	
0x6060	0x0	Modes of Operation	8	0	SINT	rw		-128	127	
0x6061	0x0	Modes of Operation Display	8		SINT	ro		-128	127	
0x6062	0x0	Position Demand Value	20489316		DINT	ro	UU	-21474836	2147483647	
0x6063	0x0	Position Actual Internal Value	20447412		DINT	ro		-21474836	2147483647	
0x6064	0x0	Position Actual Value	20943444		DINT	ro	UU	-21474836	2147483647	
0x6065	0x0	Following Error Window	600000	600000	UDINT	rw	UU	0	1073741823	
0x6066	0x0	Following Error Timeout	0	0	UINT	rw	ms	0	65535	
0x6067	0x0	Position Window	350	100	UDINT	rw	UU	0	1073741823	
0x6068	0x0	Position Window Time	250	0	UINT	rw	ms	0	65535	
0x606B	0x0	Velocity Demand Value	0		DINT	ro	UU/s	-21474836	2147483647	
0x606C	0x0	Velocity Actual Value	9844147		DINT	ro	UU/s	-21474836	2147483647	
0x606D	0x0	Velocity Window	20000	20000	UINT	rw	UU/s	0	65535	
0x606E	0x0	Velocity Window Time	0	0	UINT	rw	ms	0	65535	
0x6071	0x0	Target Torque	0	0	INT	rw	0.1%	-5000	5000	
0x6072	0x0	Maximum Torque	200	3000	UINT	rw	0.1%	0	5000	
0x6074	0x0	Torque Demand Value	15		INT	ro	0.1%	-32768	32767	
0x6076	0x0	Motor Rated Torque	1338		UDINT	ro	mNm	0	4294967295	
0x6077	0x0	Torque Actual Value	16		INT	ro	0.1%	-32768	32767	
0x607A	0x0	Target Position	20584865	0	DINT	rw	UU	-21474836	2147483647	
0x607C	0x0	Home Offset	0	0	DINT	rw	UU	-53687091	536870911	
0x607D	0x0	Software Position Limit	2	2	USINT	ro		0	255	
	0x1	Min position limit	-10000000	-10000000	DINT	rw	UU	-10737418	1073741823	
	0x2	Max position limit	10000000	10000000	DINT	rw	UU	-10737418	1073741823	
0x607F	0x0	Maximum Profile Velocity	52428800	21474836	UDINT	rw	UU/s	0	2147483647	
0x6081	0x0	Profile Velocity	100000	200000	UDINT	rw	UU/s	0	2147483647	
0x6083	0x0	Profile Acceleration	200000	200000	UDINT	rw	UU/s^	0	2147483647	
0x6084	0x0	Profile Deceleration	200000	200000	UDINT	rw	UU/s^	0	2147483647	
0x6085	0x0	Quick Stop Deceleration	200000	2000	UDINT	rw	UU/s^	0	2147483647	
0x6087	0x0	Torque Slope	1000	1000	UDINT	rw	0.1%/	0	2147483647	
0x6091	0x0	Gear Ratio	2	2	USINT	ro		0	255	Power Recycle
	0x1	Motor revolutions	1	1	UDINT	rw		0	1073741824	Power Recycle
	0x2	Shaft revolutions	1	1	UDINT	rw		0	1073741824	Power Recycle

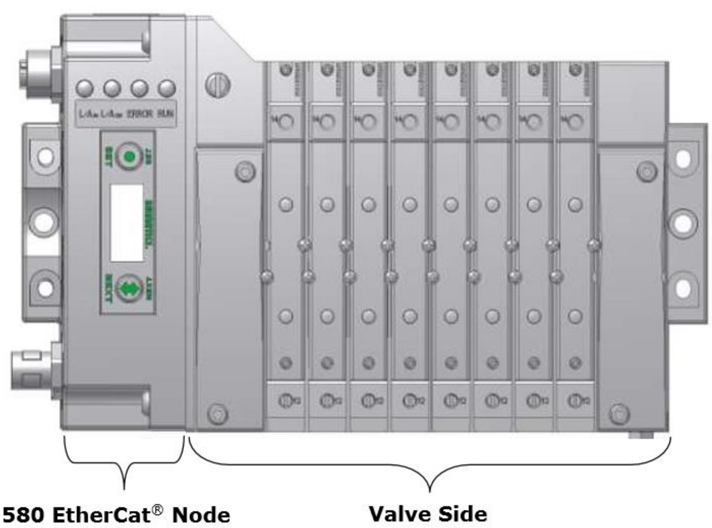
Appendix O

[O] Numatics (Emerson)

Numatics, a division of Emerson, manufactures a number solenoid valves and manifolds as well as I/O modules. That currently supported is the 501 & G3 Series as well as the 580. This section provides information that may be specific to this manufacturer.

G3:

580:



Valve Modules

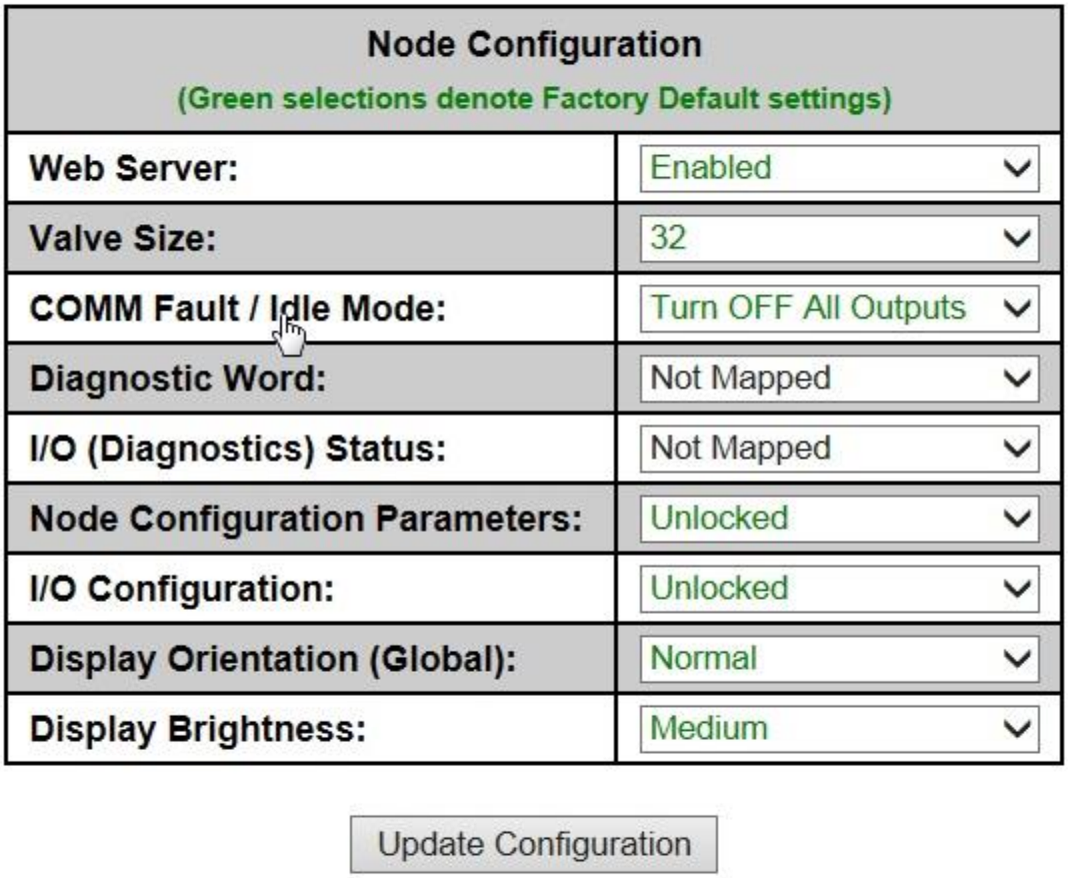
The Valve Driver Module uses the first 32 outputs regardless of how many actual valve drivers are installed. This IO space is reserved by the Numatics controller and must be considered when mapping the 5300 IO. Ensure that the firmware is at the proper level, 1.1.42194 for the G3 and 1.1.42656 for the 580, newer firmware may work but is not guaranteed beyond that verified by CTC. This firmware has special features and firmware enhancements for proper EtherCAT operation. Older firmware will not work properly. Station Alias should be set to 0.

The G3 must have outputs to operate properly, inputs are optional. Thus a G3 with only input modules is not currently supported.

Input Mapping (Required for New Controllers)

Prior to operation all Diagnostic Word & Status input bytes must be disabled or network inputs will not be aligned. This is done using the built-in web server on the Numatics unit. Numatics has documentation for this in chapter 10 of the "G3 Series EtherCAT Technical Manual" (reference the "580 EtherCAT Technical Manual" for that product) but in summary:

13. Make sure the Numatics controller EtherCAT Input port is connected to a network switch, not directly to a PC, unless a crossover cable is used.
14. Set a dedicated PC Ethernet adapter port with a static IP address of 192.168.3.100 and subnet of 255.255.255.0. Nothing is listed for the DNS server.
15. On the Numatics controller you must disable EtherCAT and enable the Web Server. This will enable TCP only on the controller and allow communications to a PC browser. This is done using the SET/NEXT buttons on the EtherCAT controller. Cycle power after the changes.
16. On the PC open a web browser and enter the IP address of the controller, 192.168.3.200. A web page should appear.
17. Set the Diagnostic Word and I/O (Diagnostics) Status to "Not Mapped" and select "Update Configuration".



18. When complete set the controller back to EtherCAT Enabled and Web Server Disabled, cycle power.

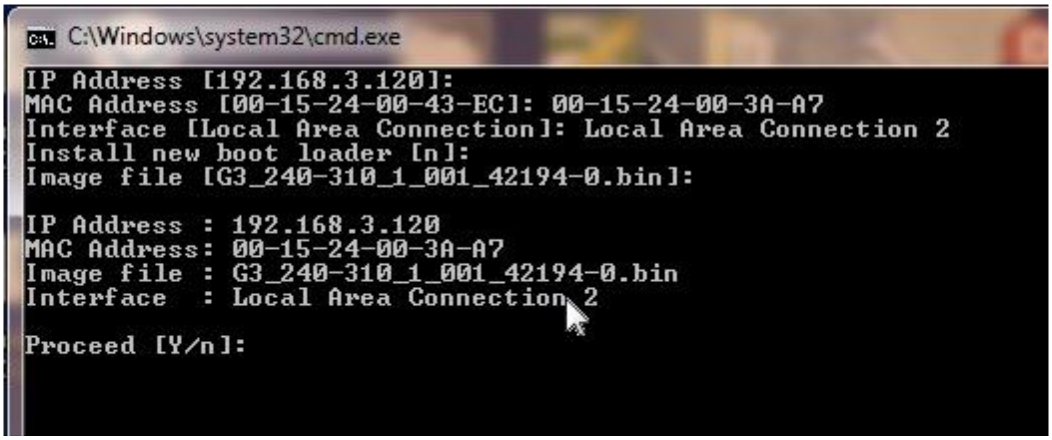
Above is for the G3, the 580 is slightly simpler but presents much of the same information. Note the Diagnostic Words and I/O Status should not be mapped. Presently it appears that the current firmware still maps 2 bytes of information regardless of the settings which our software ignores. Future firmware changes on their part may impact this, thus the importance of firmware revisions. When in doubt set the 580 to factory defaults and re-configure from a known state.

Firmware Updates

To update firmware the tftp protocol is used. A PC is required with a dedicated Ethernet port. That port should be connected to a network switch, the Numatics controller should also be connected to the same switch from the EtherCAT module. EtherCAT must be disabled and the Web Server enabled using the SET/NEXT buttons. Numatics provides documentation on the proper update of their firmware and it can also be provided by CTC if needed. In summary:

19. Set a dedicated PC Ethernet adapter port with a static IP address of 192.168.3.100 and subnet of 255.255.255.0. Nothing is listed for the DNS server.
20. The tftp client protocol must be enabled as a Windows Feature.

21. Cycle power on the Numatics controller with the previously described option changes to the Numatics EtherCAT module.
22. Scroll through the EtherCAT module settings and note the MAC address of the unit.
23. Run the Numatics tftp-load.bat program.
24. Enter information for the MAC Address, you LAN Interface used, similar to below:



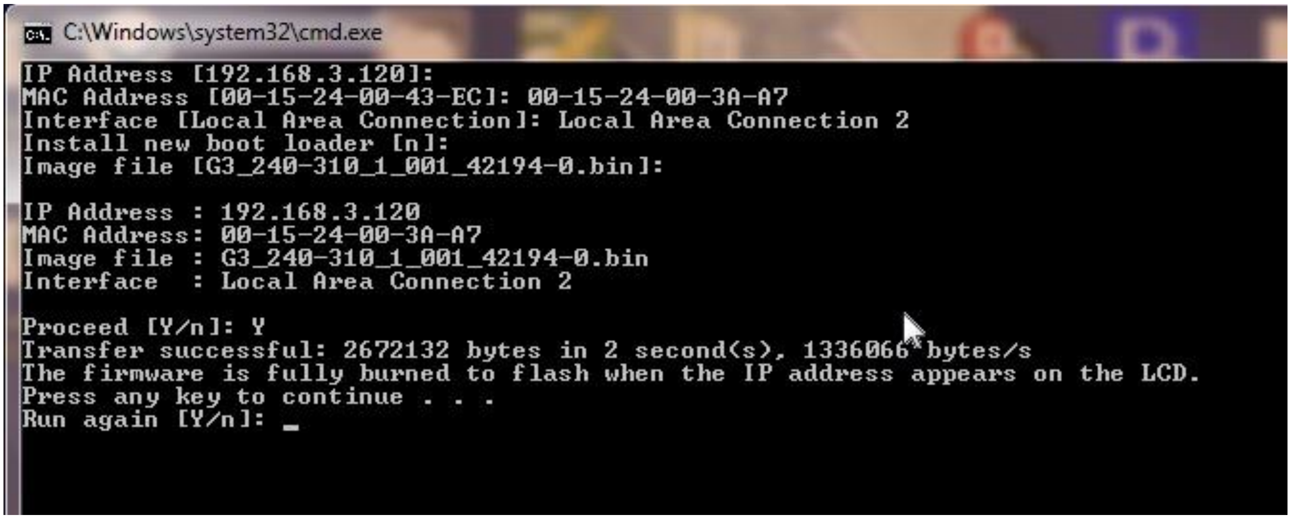
25. On the Numatics controller:

- Push NEXT on the graphic display of the G3 Node until Diagnostics is shown

- Push SET to enter the menu
- Push NEXT to move through the Diagnostics menu until "LOAD FIRMWARE" is shown
- Push SET to enter the menu
- Select yes by pushing SET and then NEXT until YES is highlighted.
- Push SET to select YES
- Push SET again to confirm choice
- Display should start flashing "WAITING FOR FIRMWARE LOAD"
- Now select Y

- You will see a transfer successful message in DOS if everything transferred OK
- Wait for the G3 node to reboot **DO NOT CYCLE POWER ON THE NODE!**
  You may confirm the successful firmware download by verifying the firmware revision on the G3 web page or through the display.

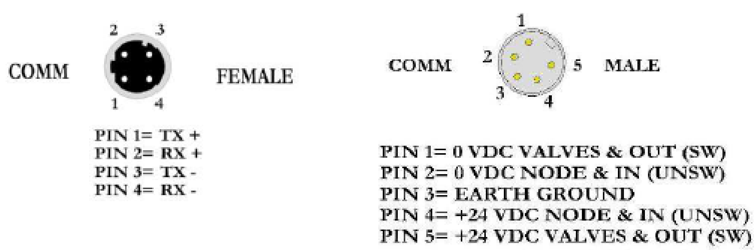
Firmware updates to the proper revision are critical to correct some anomalies in early EtherCAT Numatics firmware on both the G3 and 580. The above applies to the G3 only and is assumed to be similar for the 580, if not please contact Numatics or your distributor for additional details.



G3 Power Connector

Power Connector Pin-Out

Pin No.	*Function*	*Description*
1	0 VDC Common (Valves and Outputs)	0 VDC Voltage used to power outputs (valve coils and discrete outputs) SW
2	0 VDC Common (Node and Inputs)	0 VDC Voltage used to power discrete inputs and node electronics UNSW
3	Earth Ground	Protective Earth
4	+24 VDC (Node and Inputs)	Voltage used to power discrete inputs and node electronics UNSW
5	+24 VDC (Valves and Outputs)	Voltage used to power outputs (valve coils and discrete outputs) SW



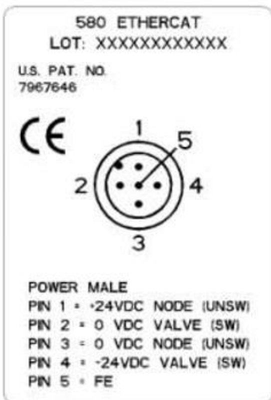
- *Power common (0 VDC) pins 1 and 2 are isolated from each other to allow separate (isolated) power supply connection if required. However, they can be tied together if a single common, non-isolated, application is preferred.*
- *The combined draw of the +24VDC Valves and Outputs and +24VDC Node and Inputs pins cannot exceed 8 Amps, at any given moment in time.*
- *The Node and Inputs pin supplies power to the node electronics. This pin must be powered at all times for communication node to be functional.*

580 Power Connector

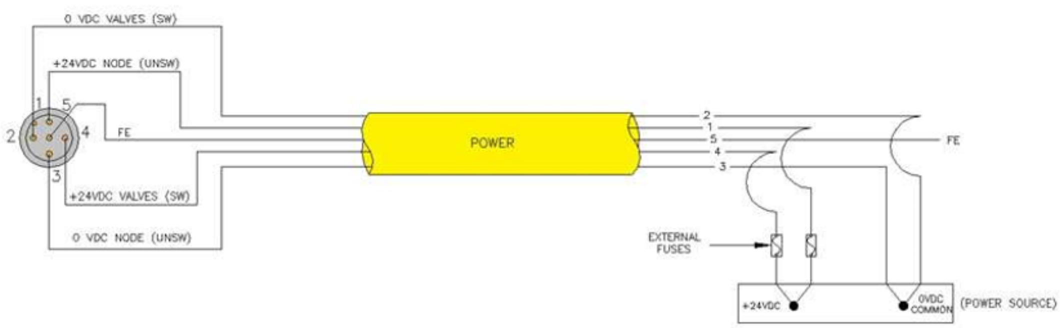
Industry standard M12 connectors are used for communication and power.
The EtherCAT® Communication connector is a D-Code 4 pin Female connector.
The Power connector is a single keyway 4 pin male connector.

Power Connector

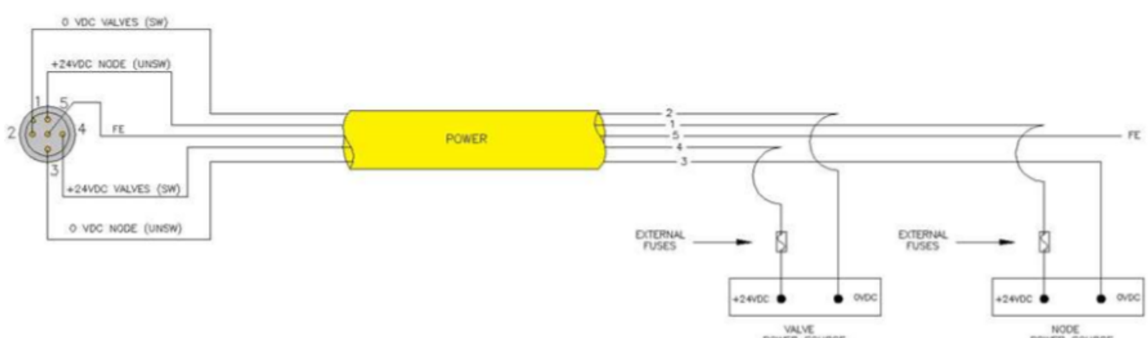
Pin No.	Function	Description (MALE)
1	+24 VDC (Node)	Voltage used to power node electronics UNSW
2	0 VDC Common (Valves)	0 VDC Voltage used to power outputs (Valves) SW
3	0 VDC Common (Node)	0 VDC Voltage used to power node electronics UNSW
4	+24 VDC (Valves)	Voltage used to outputs (Valves) SW
5	FE	Functional Earth



- ***Power common (0 VDC) pins 3 and 4 are isolated from each other to allow separate (isolated) power supply connection if required. However, they can be tied together if a single common, non-isolated, application is preferred.***
- ***The draw of the +24VDC Valves and +24VDC Node pins must not exceed 4 Amps.***
- ***The Node pins supplies power to the node electronics. These pins must be powered at all times for communication node to be functional.***
- ***To be connected to Class 2 power source only***



Separate Power Supply Example (Isolated commons)



580 EtherCAT Connector Issues

It has been discovered that on many 580's the EtherCAT IN/OUT connectors are documented and labeled backwards. Below is a mislabeled unit. Use the lower port as the IN and the upper port as the OUT. Hopefully the manufacturer will update their documentation as well as device labeling to avoid confusion. Improperly plugging the EtherCAT cable into the wrong port will cause other devices not to be identified properly as network packets are not routed properly.

Blank

Appendix

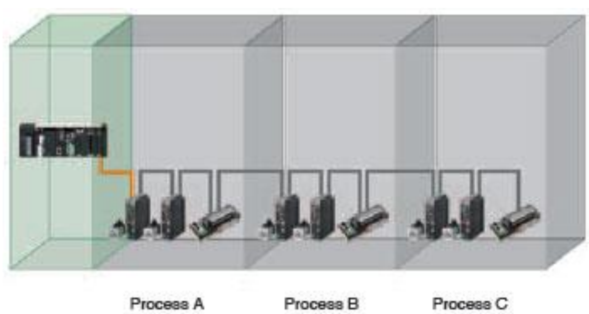
P

[P] Omron GX-JC06 EtherCAT Junction Slaves

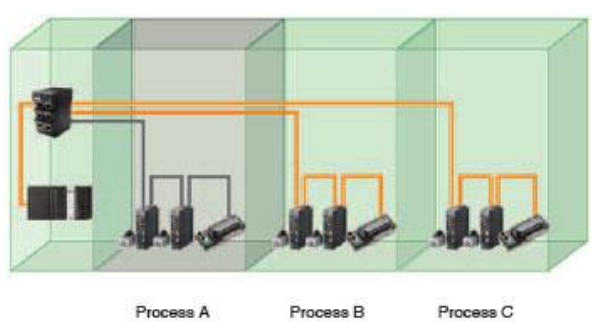
EtherCAT devices are typically wired in a daisy chain, where an IN/OUT port exist on each device. The Omron EtherCAT Junction Slave is similar to an Ethernet switch, allowing for flexible cabling. Detailed information can be found on their web site: http://www.ia.omron.com/products/family/3079/feature.html.

Cabling

Typical EtherCAT cabling as detailed by Omron:

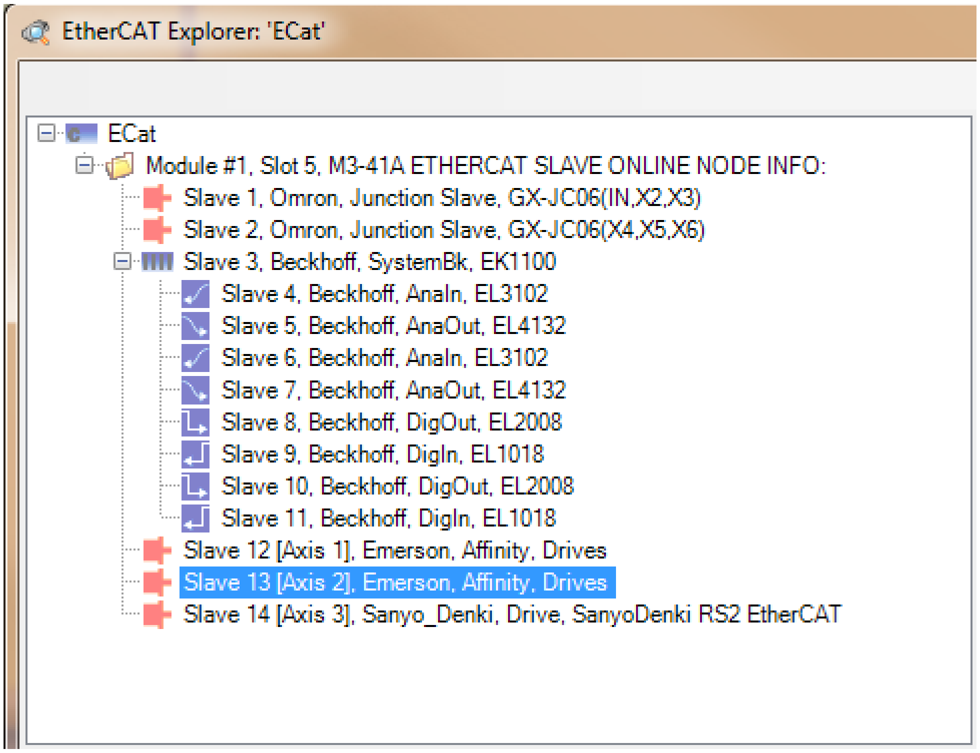


Cabling with an EtherCAT Junction Slave:



Example with M3-41 Module

The following example shows how address assignment is done using the Junction Slave. In this example the M3-41 master is plugged into the Junction Slave IN port (1). The Beckhoff IO is plugged into port 4, Emerson drives into port 5, and Sanyo Denki into port 6.



Note that the ports that are not used have no effect on the address assignment. All of the devices on a lower number port are assigned addresses first, followed by those on the next higher port number. The Junction Slave device itself appears as one or more slave nodes to the EtherCAT master and can operate as a 64 bit distributed clock reference.

Appendix

Q

[Q] Omron NX Series Controllers

Incentive supports a number of the NX Series Omron Controllers and modules. One module in particular, the Temperature Controller module, requires a special interface since it is not a typical IO device. This section details the support of Temperature Controllers from within the Incentive language.

TC-XXXX Temperature Controllers (TC)

Incentive currently supports the TC-2406 and TC-3406 temperature controller modules (TC-24/3405, 07, and 08 are also available but not tested although they look identical). The default factory EtherCAT mapping as provided by Omron is used. These modules consume a very large block of data within the EtherCAT packet of around 250 bytes per module thus it should only be used for small installations until proper packet segmentation has been verified. The controller variables have been mapped into the Incentive language in such a way that programming is done by referring to the Omron Temperature Controller manual.

The interface is identical to that of RFID where a virtual axis is run on the EtherCAT Master and the MSB language accesses the TC_ variables. Each TC channel is selected and multiplexed through the interface.

TC Property Variables

TC_totalChannels – Read only, represents the total number of Temperature Controller (TC) channels available in the system.

TC_channel – Read/write, allows selection of the TC channel to be operated on by the properties that follow. Entries of 1 to *TC_totalChannels* are the valid selections, with 0 disabling access. Write access is done by copying variables that may be written to the EtherCAT packet window for the TC. This is activated by the TC_write property as detailed below. This prevents accidental writes when changing the selected channel. Allow about 5 milliseconds delay to allow EtherCAT packet to propagate (takes 1 ms) before reading or writing any values.

TC_write – Read/write, when nonzero the bit assigned will cause that value to be copied to the temperature controller. This is needed since you are multiplexing channels and do not want to copy another channels setpoint and alarm data during a new channel selection.

Bit 0 - write command
Bit 1 - write setpoint
Bit 2 - write alarmValue1
Bit 3 - write alarmValue2
Bit 4 - write alarmLimitUpper1
Bit 5 - write alarmLimitLower1
Bit 6 - write alarmLimitUpper2
Bit 7 - write alarmLimitLower2
Bit 8 - write MV_manual

When done TC_write will be cleared to 0. A write only happens when non-zero and this does not include the separate command that the temperature command needs itself, just moving it into the proper EtherCAT packet.

TC_command - Read/write, see the TC_response for response flag verification of write operation. The operation command is executed starting with bit 0 and in ascending order. When the detection method is edge (rising), the operation command is executed when the value changes from "0" to "1". Therefore, when issuing the operation command again, set it to "0" and then change it to "1". If the execution conditions are not satisfied when the operation command is issued, the operation command is not executed. The operation command is issued when the execution conditions are satisfied. If the execution conditions are not satisfied when the operation command is issued, the operation command is not executed, but is set as pending. The operation command is executed when the execution conditions are satisfied. Example: If "40% AT" is executed during the execution of "100% AT", "40% AT" will be executed after the execution of "100% AT" is completed. If the operation command is "1" when the power supply is turned ON, the edge (rising) is detected and the operation command is issued.

Operation Command bit definitions shown below.

Bit 0 - Run (0)/Stop (1), level triggered (EtherCAT initializes to 1, stop)
Bit 1 - 100% at Execution 0 -> 1, Edge (rising)
Bit 2 - 40% at Execution 0 ? 1: Edge (rising)
Bit 3 - AT Cancel 0 ? 1: Edge (rising)
Bit 4 - Automatic Filter Adjustment Execute, 0 -> 1: Executing Edge (rising)
Bit 5 - Automatic Filter Adjustment Cancel 0 ? 1: Cancel Edge (rising)
Bit 6 - Water Cooling Output Adjustment 0: Disable 1: Enable (Level)
Bit 7 - Adaptive Control PID Update 0 ? 1: PID Update by adaptive control Edge (rising)
Bit 8 - Auto or Manual. If you switch to manual mode during AT execution, AT is canceled.
0: Auto mode 1: Manual mode (Level)
Bit 9 - Reflect Manual MV 0: Do not reflect 1: Reflect (Level)
Bit 10- Direct/Reverse Operation 0: Not inverting 1: Inverting (Level)
Bit 11- Reflect Data for Adjustment 0: Do not reflect 1: Reflect (Level)
Bit 12 to 15 Reserved

TC_status – Read only, from the Omron manual, Operating Status is bits 0 to 15 and Output and Alarm status are bits 16 to 31

Bit 0 - RUN or STOP Status 0: Run, 1: Stop
Bit 1 - 100 Percent AT Status 0: 100% AT Stopping, 1: 100% AT Executing
Bit 2 - 40 Percent AT Status 0: 40% AT Stoppin,g 1: 40% AT Executing
Bit 3 - Automatic Filter Adjustment Status 0: Stopping, 1: Executing
Bit 4 - Water Cooling Output Adjustment Proportional Band Increase 0: Not increasing, 1: Increasing
Bit 5 - Water Cooling Output Adjustment Proportional Band Decrease 0: Not decreasing, 1: Decreasing
Bit 6 - Adaptive Control System Performance Evaluation State 0: Performance evaluation is not in progress, 1: Performance evaluation is in progress
Bit 7 - Adaptive Control Notification in Progress 0: No notification, 1: Notification
Bit 8 - Adaptive Control PID Update Enabled 0: Without updatable PID constants, 1: With updatable PID constants
Bit 9 - Auto or Manual Status 0: Auto mode, 1: Manual mode
Bit 10 - Channel Reflect Manual MV Status 0: Not reflected, 1: Reflected
Bit 11 - Inverting Direct or Reverse Operation Status 0: Not inverting, 1: Inverting
Bit 12 - Tuning Parameter Updated 0: Tuning parameter is not updated, 1: Tuning parameter is updated
Bit 13 to 15 Reserved
Bit 16- Heating Control Output 0: OFF, 1: ON
Bit 17- Cooling Control Output 0: OFF, 1: ON
Bit 18- Sensor Disconnected Error 0: No errors occurred, 1: Error occurred
Bit 19- Cold Junction Error 0: No errors occurred, 1: Error occurred
Bit 20- AD Converter Error 0: No errors occurred, 1: Error occurred
Bit 21- Heater Burnout Detection 0: Not detected, 1: Detected
Bit 22- SSR Failure Detection 0: Not detected, 1: Detected
Bit 23- Heater Current Hold 0: Updated, 1: Not updated
Bit 24- Heater Current Exceeded 0: The measurement range is not exceeded, 1: The measurement range is exceeded
Bit 25- Alarm 1 Detection 0: Not detected, 1: Detected
Bit 26- Alarm 2 Detection 0: Not detected, 1: Detected
Bit 27 to 31 Reserved

TC_setpoint – Read/write, temperature set point register.

TC_value – Read only, current temperature value.

TC_response - Read only, the response flag which changes settings based upon result of TC_command. The reflection result when the 'Reflect Data for Adjustment' of Operation Command is executed.

When writing is normal.
- 0000 hex: Not reflected
- 0001 hex: Reflected

When writing is abnormal.
- High-order byte: Low-order byte of the index number of the I/O data for adjustment in

which an error has occurred.
- Low-order byte: Sub-index number of the I/O data for adjustment in which an error has occurred.

If an error occurs during writing in multiple data, the largest number assigned in the I/O data is reflected in the response flag. For example if Ch2 Proportional Band (Index: 0x7005, Sub-index: 0x02) is assigned after Ch4 PV Input Shift (Index: 0x701A, Sub-index: 0x04) in the I/O data, and a write error due to surpassing of the setting range occurs in both, the response flag becomes 0x0502.

TC_moduleChannels – Read only, the total number of channels available on this particular module. For example a TC3406 would be 4.

TC_moduleChannel – Read only, the channel being reference on this particular module, for example a TC3406 would be 1 to 4 since a 4 channel module.

TC_moduleCode – Read only, the product code of the module this channel is accessing. May be used if anything specific to a module is needed. 0x00313406 is Omron TC3406 and 0x00312406 is Omron TC2406.

For below variables reference the NX series Omron TC2406 and TC3406 Temperature Control Unit manual.

TC_alarmValue1 (read/write)
TC_alarmValue2 (read/write
TC_alarmLimitUpper1 (read/write)
TC_alarmLimitUpper2 (read/write)
TC_alarmLimitLower1 (read/write)
TC_alarmLimitLower2 (read/write)
TC_MV_monitorHeating (read only)
TC_MV_manual (read/write)

Note that any timing between commands must be done by the application. Incentive merely gives the programmer a window into the Omron Temperature Control Unit data window, updated at each scanning interval.

Example logic for writing a command:

1. Set TC_channel to the desired channel.
2. Load TC_command with the desired command as per the Omron manual.
3. Load any properties the command will reference, for example TC_setpoint.
4. Make sure TC_write is 0 and if so then set the bits needed for the command to be copied to the EtherCAT packet. If TC_command and TC_setpoint then set bits 0 and 1.
5. Wait for TC_write to become 0. Which means the command has been copied to the EtherCAT packet and is being transmitted.
6. Delay enough time for the packet to be sent to the Omron controller. Typically 5 mS is plenty of time.
7. Check TC_response for result.

Appendix

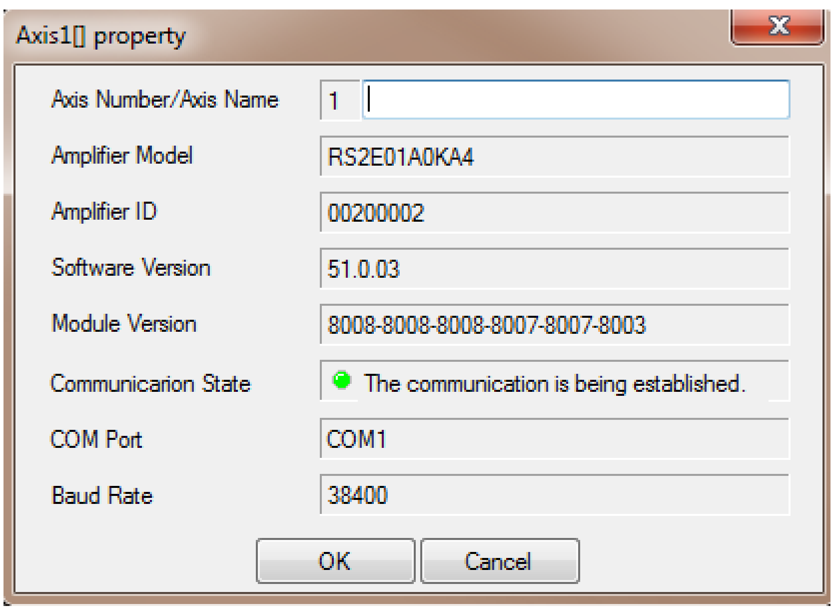
R

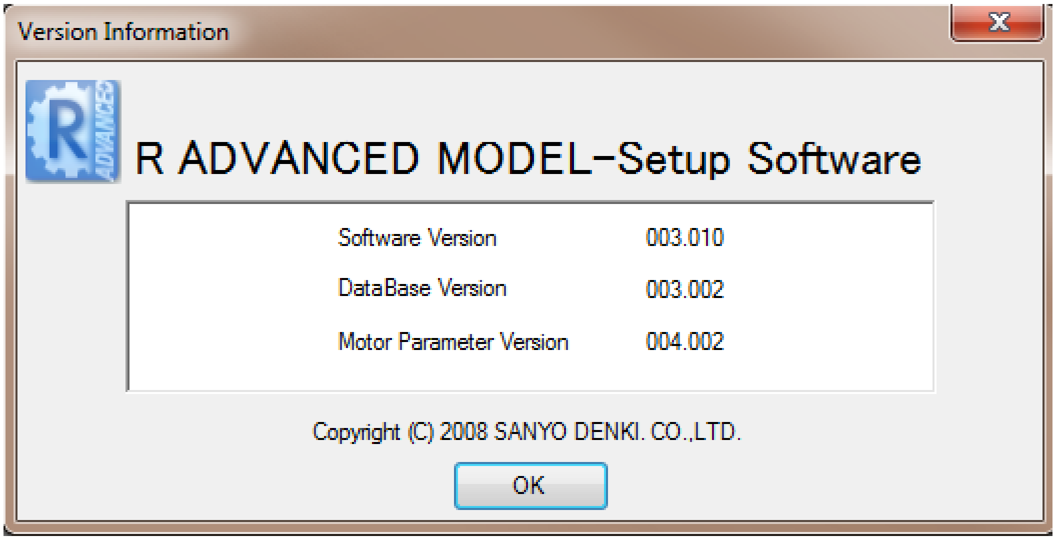
[R] Sanyo Denki

Sanyo Denki manufactures a number of drives. The drive currently supported is the single axis RS2E. The configuration utility used is “R Advanced Model – Setup Software”. This section provides information that may be specific to this manufacturer.

eCAT_driveType – 6

Drive Information & Firmware





Station Alias

In an EtherCAT network, slaves are automatically assigned addresses based on their position in the bus. When a device, such as a drive, must have a fixed assigned identification that is independent of cabling, a Station Alias is needed. Sanyo Denki provides a single 16-position rotary switch with hexadecimal encoding for this purpose. This 4-bit switch is used to set the lower bits 3 to 0 of the alias, while bits 15 to 4 are written using the Setup Software, with a default of 0. Since the M3-41 only supports up to 16 drives, the single switch will suffice using the setup software defaults for the upper bits. A switch setting of 0 will default to automatic addressing.

Sanyo Denki Drive Mode Transitions

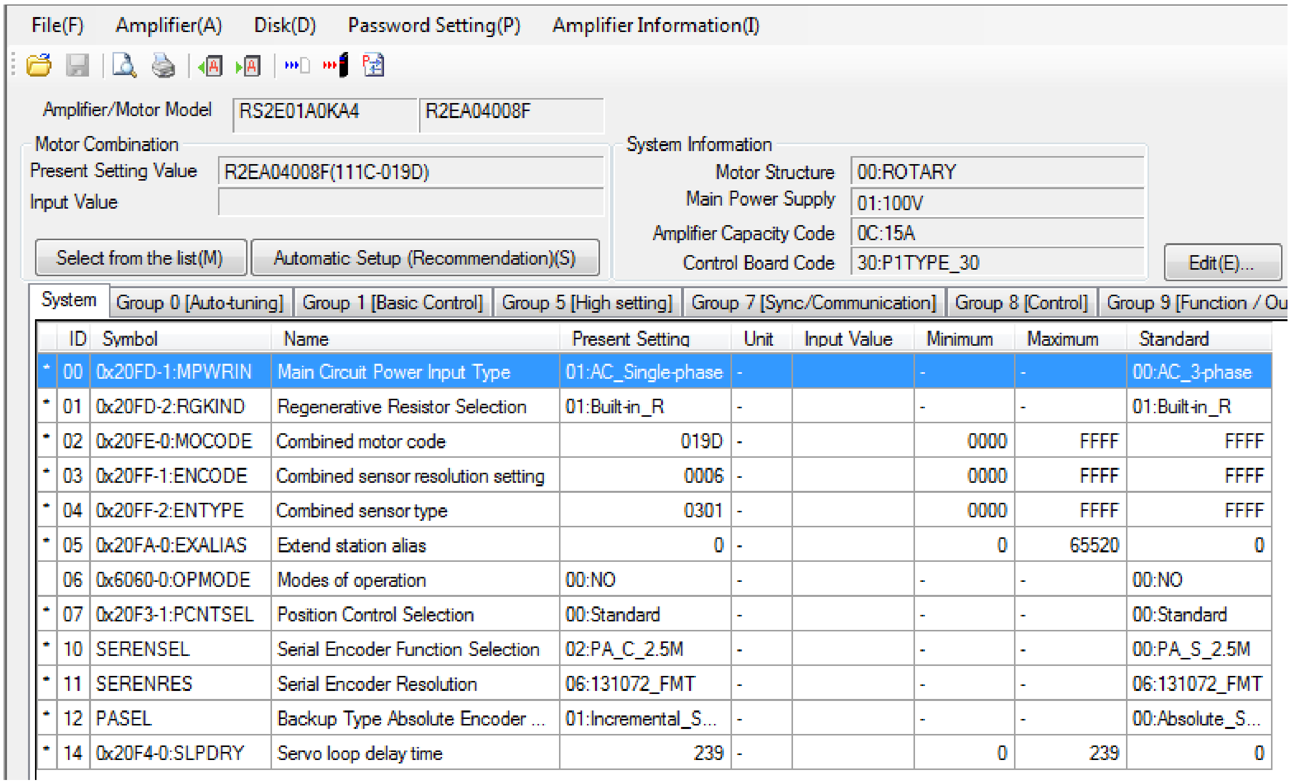
Unlike other drive manufacturers, Sanyo Denki requires that the Operation Enabled bit be disabled in the control word prior to changing a drive mode (such as cyclic sync position mode to profile position mode). To allow this to operate correctly, a 500-millisecond delay will occur during the first move instruction after changing a mode. This will only happen once, until a mode is changed again. Additional move instructions using the same drive mode will not be delayed.

Sanyo Denki Power-Up Delay

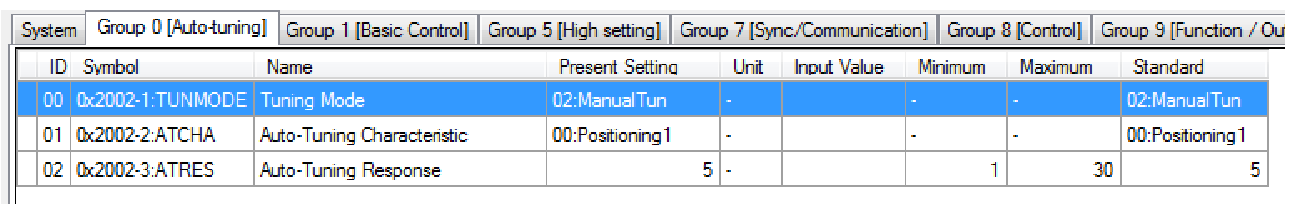
As just discussed the Sanyo Denki drive requires a 500-millisecond delay after a drive mode change. This also includes transitioning to Cyclic Sync Position mode at power up. The 500-millisecond delay will occur with the first move. Additional commands following the first move will not be delayed.

R Advanced Model – Setup Software

CTC used R Advanced Model Setup Software, on a Windows 7 64-bit computer to test and configure the drives. Typically R Advanced Model will be needed for tuning and limited setup. Below are some setup screens and their typical parameters at power up (offline).



	ID	Symbol	Name	Present Setting	Unit	Input Value	Minimum	Maximum	Standard
*	00	0x20FD-1:MPWRIN	Main Circuit Power Input Type	01:AC_Single-phase	-		-	-	00:AC_3-phase
*	01	0x20FD-2:RGKIND	Regenerative Resistor Selection	01:Built-in_R	-		-	-	01:Built-in_R
*	02	0x20FE-0:MOCODE	Combined motor code	019D	-		0000	FFFF	FFFF
*	03	0x20FF-1:ENCODE	Combined sensor resolution setting	0006	-		0000	FFFF	FFFF
*	04	0x20FF-2:ENTYPE	Combined sensor type	0301	-		0000	FFFF	FFFF
*	05	0x20FA-0:EXALIAS	Extend station alias	0	-		0	65520	0
	06	0x6060-0:OPMODE	Modes of operation	00:NO	-		-	-	00:NO
*	07	0x20F3-1:PCNTSEL	Position Control Selection	00:Standard	-		-	-	00:Standard
*	10	SERENSEL	Serial Encoder Function Selection	02:PA_C_2.5M	-		-	-	00:PA_S_2.5M
*	11	SERENRES	Serial Encoder Resolution	06:131072_FMT	-		-	-	06:131072_FMT
*	12	PASEL	Backup Type Absolute Encoder ...	01:Incremental_S...	-		-	-	00:Absolute_S...
*	14	0x20F4-0:SLPDRY	Servo loop delay time	239	-		0	239	0



ID	Symbol	Name	Present Setting	Unit	Input Value	Minimum	Maximum	Standard
00	0x2002-1:TUNMODE	Tuning Mode	02:ManualTun	-		-	-	02:ManualTun
01	0x2002-2:ATCHA	Auto-Tuning Characteristic	00:Positioning1	-		-	-	00:Positioning1
02	0x2002-3:ATRES	Auto-Tuning Response	5	-		1	30	5

System | Group 0 [Auto-tuning] | Group 1 [Basic Control] | Group 5 [High setting] | Group 7 [Sync/Communication] | Group 8 [Control] | Group 9 [Function / Ou

ID	Symbol	Name	Present Setting	Unit	Input Value	Minimum	Maximum	Standard
00	0x2003-0:PCSMT	Position Command Smoothing Co...	0.5	ms		0.0	500.0	0.5
01	0x2004-0:PCFIL	Position Command Filter	0.0	ms		0.0	2000.0	0.0
02	0x2005-1:KP1	Position Loop Proportional Gain 1	28	1/s		1	3000	30
03	0x2006-1:TPI1	Position Loop Integral Time Const...	1000.0	ms		0.3	1000.0	1000.0
04	0x2007-0:TRCPGN	Higher Tracking Control Position ...	0	%		0	100	0
05	0x2008-1:FFGN	Feed Forward Gain	0	%		0	100	0
06	0x2008-2:FFFIL	Feed Forward Filter	4000	Hz		1	4000	4000
10	0x2009-0:VCFIL	Velocity Command Filter	4000	Hz		1	4000	4000
11	0x200A-0:VDFIL	Velocity Feedback Filter	1500	Hz		1	4000	1500
12	0x200B-1:KVP1	Velocity Loop Proportional Gain 1	27	Hz		1	2000	50
13	0x200C-1:TVI1	Velocity Loop Integral Time Const...	37.3	ms		0.3	1000.0	20.0
14	0x200D-1:JRAT1	Load Inertia Moment Ratio 1	100	%		0	15000	100
15	0x200E-0:TRCVGN	Higher Tracking Control Velocity ...	0	%		0	100	0
16	0x200F-1:AFBK	Acceleration Feedback Gain	0.0	%		-100.0	100.0	0.0
17	0x200F-2:AFBFIL	Acceleration Feedback Filter	500	Hz		1	4000	500
20	0x2011-1:TCFIL1	Torque Command Filter 1	369	Hz		1	4000	600
21	0x202B-0:TCFILOR	Torque Command Filter Order	2	Order		1	3	2

System | Group 0 [Auto-tuning] | Group 1 [Basic Control] | Group 5 [High setting] | Group 7 [Sync/Communication] | Group 8 [Control] | Group 9 [Function / Ou

ID	Symbol	Name	Present Setting	Unit	Input Value	Minimum	Maximum	Standard
00	0x2015-1:ACCC0	Acceleration Compensation	0	x50 ...		-9999	9999	0

System | Group 0 [Auto-tuning] | Group 1 [Basic Control] | Group 5 [High setting] | Group 7 [Sync/Communication] | Group 8 [Control] | Group 9 [Function / Ou

	ID	Symbol	Name	Present Setting	Unit	Input Value	Minimum	Maximum	Standard
	00	0x1C32-1:SM2TYP	SM2 Sync mode	0002	-		0000	0003	0002
	01	0x1C32-2:SYCLE	SM2 Sync cycle time	500000	nsec		500000	64000000	500000
	02	0x1C33-1:SM3TYP	SM3 Sync mode	0002	-		0000	0022	0002
*	03	0x20FD-3:COMBA...	Serial Communication Baud Rate	05:38400bps	-		-	-	05:38400bps

System | Group 0 [Auto-tuning] | Group 1 [Basic Control] | Group 5 [High setting] | Group 7 [Sync/Communication] | Group 8 [Control] | Group 9 [Function / Ou

ID	Symbol	Name	Present Setting	Unit	Input Value	Minimum	Maximum	Standard
00	0x607E-0:CMDPOL	Polarity	00:P+_V+_T+	-		-	-	00:P+_V+_T+
01	0x607F-0:VCLM	Max Profile Velocity	4294967295	pps		0	4294967295	4294967295
02	0x6081-0:PROVEL	Profile Velocity	4294967295	pps		0	4294967295	4294967295
03	0x201C-0:VCMMAX	Velocity Limit Command	65535	min-1		1	65535	65535
04	0x6083-0:TVCACC	Profile Acceleration	4294967295	pps/s		0	4294967295	4294967295
05	0x6084-0:TVCDEC	Profile Declaration	4294967295	pps/s		0	4294967295	4294967295
06	0x6072-0:MAXTRQ	Max torque	500.0	%		0.0	500.0	500.0
07	0x60E0-0:TCLM-F	Forward Direction Internal Torque...	500.0	%		0.0	500.0	500.0
08	0x60E1-0:TCLM-R	Reverse Direction Internal Torqu...	500.0	%		0.0	500.0	500.0
09	0x201E-0:SQTCLM	Sequence Operation Torque Limi...	120.0	%		10.0	500.0	120.0
0...	0x201F-0:NEAR	Near Range	500	Pulse		1	2147483647	500
0...	0x6067-0:INP	In-Position Window	100	Pulse		1	2147483647	100
0...	0x2020-0:ZV	Speed Zero Range	50	min-1		50	500	50
0...	0x6087-0:TSLOPE	Torque Slope	429496729.5	%/s		0.0	42949672...	429496729.5
0...	0x2021-0:LOWV	Low Speed Range	50	min-1		0	65535	50
0F	0x2022-0:VA	Speed Attainment Setting (High S...	1000	min-1		0	65535	1000
10	0x606D-0:VCMP	Speed Matching Range	50	min-1		0	65535	50
11	0x202A-0:VCMPR	Speed Matching Range Ratio	5.0	%		0.0	100.0	5.0
12	0x607D-1:SMINLIM	Software minimum position limit	80000000	Pulse		80000000	7FFFFFFF	00000000
13	0x607D-2:SMAXLIM	Software maximum position limit	7FFFFFFF	Pulse		80000000	7FFFFFFF	00000000

System | Group 0 [Auto-tuning] | Group 1 [Basic Control] | Group 5 [High setting] | Group 7 [Sync/Communication] | Group 8 [Control] | Group 9 [Function / Ou

ID	Symbol	Name	Present Setting	Unit	Input Value	Minimum	Maximum	Standard
00	0x20F8-1:PLIMSW	Positive Limit Switch Function	00:Always_Disable	-		-	-	00:Always_Dis...
01	0x20F8-2:NLIMSW	Negative Limit Switch Function	00:Always_Disable	-		-	-	00:Always_Dis...
02	0x20F8-3:EXT-E	External Trip Input Function	00:Always_Disable	-		-	-	00:Always_Dis...
03	0x20F8-4:DISCHA...	Main Power Discharge Function	01:Always_Enable	-		-	-	01:Always_En...
04	0x20F8-5:EMR	Emergency Stop Function	00:Always_Disable	-		-	-	00:Always_Dis...
05	0x20F0-1:ACTOT	Limit Switch Action	06:CMDACK_VCL...	-		-	-	06:CMDACK_...
* 06	0x20F0-2:EDGEPOS	Positioning Methods	00:Pulse_Interval	-		-	-	00:Pulse_Inter...
07	0x20F0-3:PDEVMON	In-Position Signal/ Position Devia...	00:After_Filter	-		-	-	00:After_Filter
08	0x20F0-4:VCMPUS	Speed Matching Unit Selection	00:min-1	-		-	-	00:min-1
09	0x20F0-5:CLR	Deviation Clear Selection	00:Type1	-		-	-	00:Type1
0...	0x20F9-1:OUT1	General Purpose Output 1 Selecti...	42:FOUT1_ON	-		-	-	42:FOUT1_ON
0...	0x20F9-2:OUT2	General Purpose Output 2 Selecti...	44:FOUT2_ON	-		-	-	44:FOUT2_ON
10	0x2023-1:MON1	Analog Monitor Select Output 1	05:VMON_2mV/m...	-		-	-	00:RESERVE
11	0x2023-2:MON2	Analog Monitor Select Output 2	02:TCMON_2V/TR	-		-	-	02:TCMON_2...
12	0x2023-3:MONPOL	Analog Monitor Output Polarity	00:MON1+_MON2+	-		-	-	00:MON1+_M...
21	JOGVC	JOG Velocity Command	50	min-1		0	32767	50

System | Group 0 [Auto-tuning] | Group 1 [Basic Control] | Group 5 [High setting] | Group 7 [Sync/Communication] | Group 8 [Control] | Group 9 [Function / Ou

ID	Symbol	Name	Present Setting	Unit	Input Value	Minimum	Maximum	Standard
01	0x605C-0:DISOP	Disable operation option code	0	-		-5	0	0
02	0x605D-0:HALTCD	Halt option code	1	-		1	3	1
03	0x6065-0:OFLV	Deviation Counter Overflow Value	5000000	Pulse		1	2147483647	5000000
04	0x2024-0:BONDLY	Delay Time of Engaging Holding ...	300	ms		0	1000	300
05	0x2025-0:BOFFDLY	Delay Time of Releasing Holding ...	300	ms		0	1000	300
06	0x2026-0:BONBGN	Brake Operation Beginning Time	10000	ms		0	65535	10000
07	0x2027-0:PFDDLY	Power Failure Detection Delay Ti...	32	ms		20	1000	32
08	0x20F5-0:CPETLSEL	Torque Limit at power supply shor...	00:Typ_Limit	-		-	-	00:Typ_Limit
09	0x2028-0:OFWLV	Excessive Deviation Warning Le...	2147483647	Pulse		1	2147483647	2147483647
0...	0x2029-0:OLWLV	Overload Warning Level	90	%		20	100	90
0...	0x2103-2:WARMSK	Warning mask	4C8D	-		0000	FFFF	4C8D

System | Group 0 [Auto-tuning] | Group 1 [Basic Control] | Group 5 [High setting] | Group 7 [Sync/Communication] | Group 8 [Control] | Group 9 [Function / Ou

ID	Symbol	Name	Present Setting	Unit	Input Value	Minimum	Maximum	Standard
00	0x20F1-1:ECLRFU...	Encoder Clear Function Selection	00:Status_MultiTurn	-		-	-	00:Status_Mult...
04	0x20F2-1:MPESEL	Main Power Error Selection	01:MPE_ENA	-		-	-	01:MPE_ENA
05	0x20F2-2:VCALM	Velocity Control Alarm (ALM_C2) ...	00:Disabled	-		-	-	00:Disabled
06	0x20F2-3:VFBALM	Velocity Feedback Alarm (ALM_C...	01:Enabled	-		-	-	01:Enabled
07	0x20F2-4:CRCSET	Frame error filter	0	-		0	8	0
08	0x20F2-5:COTOUT	Comunication timeout filter	0	-		0	255	0
09	0x201D-0:OVFSET	Position Command Error 1 Level	4294967295	pps		0	4294967295	4294967295

System | Group 0 [Auto-tuning] | Group 1 [Basic Control] | Group 5 [High setting] | Group 7 [Sync/Communication] | Group 8 [Control] | Group 9 [Function / Ou

ID	Symbol	Name	Present Setting	Unit	Input Value	Minimum	Maximum	Standard
00	0x6098-0:HOMETYP	Homing method	35	-		0	35	35
01	0x6099-1:SSVCM	Homing speed during search for s...	655360	pps		0	2147483647	655360
02	0x6099-2:ZSVCMD	Homing speed during search for z...	32768	pps		0	2147483647	32768
03	0x609A-1:HOMEACC	Homing Acceleration	4294967295	pps/s		1	4294967295	4294967295
04	0x607C-0:HOFFSET	Home offset	0	Pulse		-214748...	2147483647	0
05	0x605A-0:QSTOP	Quick stop option code	2	-		-2	7	2
06	0x6085-0:QSDEC	Quick stop deceleration	4294967295	pps/s		1	4294967295	4294967295

EtherCAT Explorer View

Manuf	Sanyo_Denki
Grp	Drive
Name	SanyoDenki RS2 EtherCAT
Out	160 bits (20 bytes)
In	240 bits (30 bytes)
Axis #	8
pstate	COMPLETE (2)
tracking_pstate	COMPLETE (2)
inpos	0
fpos	0.000000
tpos	0.000000
perr	0.000000
vel	0.000000
DRV MODE	Cyclic Sync Position (8)
PDO STATUS	0x0450
PDO CNTLWORD	0x0002
PDO ACT VEL	0x00000000
PDO ACT TORQ	0x00000000
PDO ACT ERR	0x00000000
PDO HOME PWRUP	0x0027D093
PDO ACT POS	0x0027D093
PDO TARG POS	0x0027D093
PDO TARG VEL	0x00000000
PDO DIG INP	0x00000008
State	8 (OPERATIONAL)
Delay	4850 ns
Has DC	true (64 bits)
DC Parent	1
DC Active	false, Cyc time: 0 ns, Shft: 0
Parent	6
Config addr	0x1007 (4103)
Station Alias	0
Vndr	0x000001b9 (441)
Product Code	0x00000002 (2)
Rev	0x00000000

Appendix
S

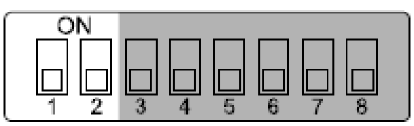
[S] SMC Corporation

SMC Corporation manufactures a number solenoid valves and manifolds. That currently supported is the EX600. This section provides information that may be specific to this manufacturer.

EX600 Fieldbus System

The valves module allows for 8 to 32 valves. A dip switch must be set to configure the number of valves present or the wrong number of valves will be reported to EtherCAT and placed in the 5300 Output section.

•V_SEL switch: A function to select the number of occupied valve outputs.
Select the number of outputs (size) occupied by the SI unit.



Settings

Settings		Content	SI unit output data size
1	2		
OFF	OFF	Number of valves = 32 outputs (Default setting)	4 byte
OFF	ON	Number of valves = 24 outputs	3 byte
ON	OFF	Number of valves = 16 outputs	2 byte
ON	ON	Number of valves = 8 outputs	1 byte

*: Set the number of occupied valve outputs to at least the number of valves used.

Also note that diagnostics are not supported for this device.

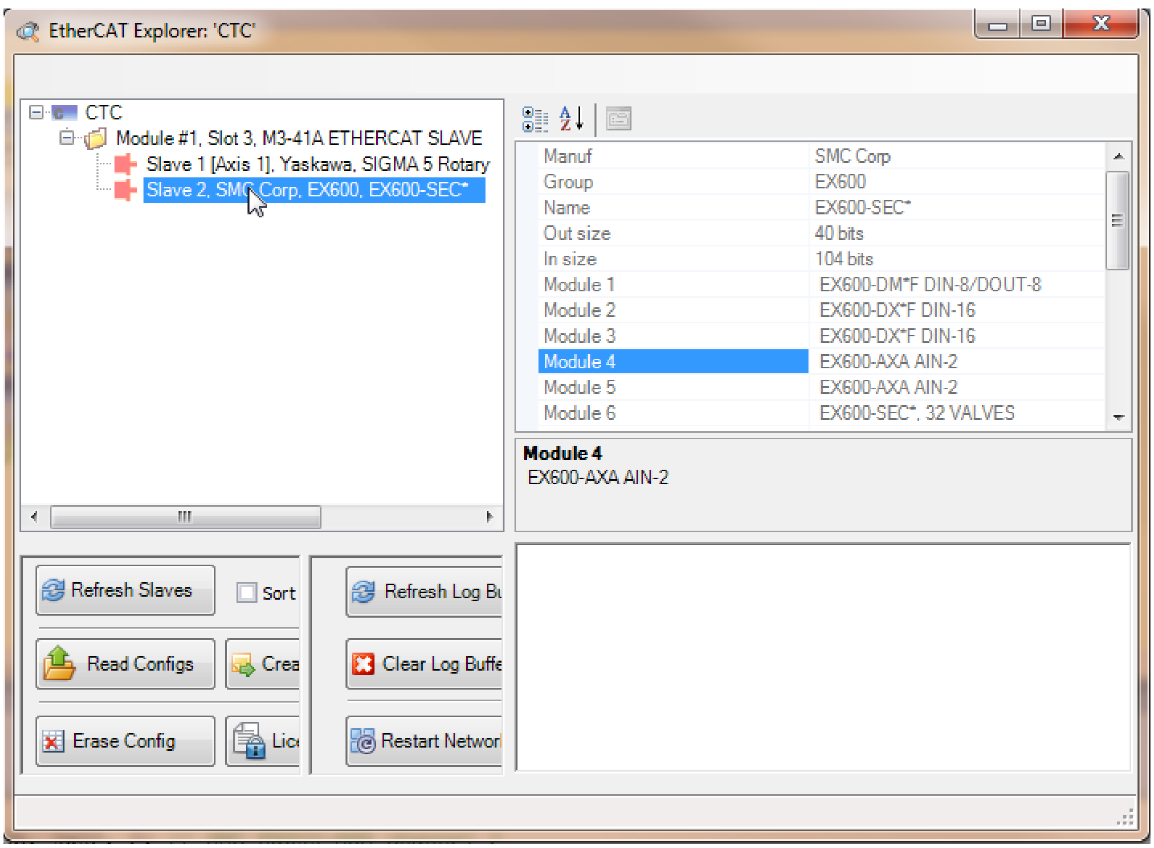
DC Sync

The EX600 reports within the QuickBuilder Explorer that it is capable of operating as a 64 bit distributed clock reference. **Upon testing it has been found that the device does not support the ARMW packet that is required to distribute the clock amongst the slave devices and therefore must not be used as the first device on the network, in a distributed clock environment.**

SDO Configuration

By default, QuickBuilder uses the SMC EX600 in its default configuration. Each module has numerous option settings which can be customized offline using an EtherCAT Configurator, such as Beckhoff's. SMC provides no non-EtherCAT method of configuration. Reference SMC's Operation Manual, EX##-OMO0027 (page 68), for detailed object mapping information. Page 53 describes configuration using the Beckhoff EtherCAT Configurator (simplified version of TwinCAT).

An alternative to using an EtherCAT Configurator would be to write to the individual objects that require customization. This would normally be done once during initialization using the 'sdo write' MSB instruction. Since the EX600 has no dedicated MSB one of the drives MSB's must be used to issue the sdo writes. Prior to doing any sdo writes the EX600 slave address and module slot must be determined:



Referencing the EtherCAT Explorer image, slave #2 is the SMC unit. In our example we will configure the analog modules (EX600-AXA) for 4-20 ma loop, given the default is +/- 10V. In the EX600 rack modules 4 and 5 are analog input modules, each with 2 channels. According to the SMC Operations Manual, page 83, the EX600-AXA configuration objects (in hex) start at 80x0.00 and end at 80xB.02 where the x is the 'Module # - 1' and the .## is the index. From the manual 80xB is the Analog Input Range setting whose contents are defined as follows:

0 = -10…+10 V
1 = -5…+5 V
2 = -20…+20 mA
3 = 0…10 V
4 = 0…5 V
5 = 1…5 V
6 = 0…20 mA
7 = 4…20 mA

Substituting the Module #, the objects that must be written are 0x803b and 0x804b, for modules 4 and 5 respectively. The default value of 0 must be changed to 7 for 4-20 mA. A sample piece of code appears below which is run on Slave 1, the Yaskawa drive MSB.

```
[beginTest]
// Setup the SMC for 4-20ma, it is slave device 2, slot 4/5 are AXA modules
// according to the EtherCAT Explorer.  According to SMC slot configuration
// address for a module is 0x80#0, zero based. Thus 0x803X & 0x804X are the
// configuration blocks for the EX600-AXA modules.  According to their
// Operation Manual, page 83, range parameter:
// 0x803B.01 = channel 1
// 0x803B.02 = channel 2
// 0x804B.01 = channel 3
// 0x804B.02 = channel 4
// Enumerated values that can be written and their meaning:
//    0=-10…+10 V
//    1=-5…+5 V
//    2=-20…+20 mA
//    3=0…10 V
//    4=0…5 V
//    5=1…5 V
//    6=0…20 mA
//    7=4…20 mA
// A range value of 0 is +/- 10V, 7 is for 4 - 20 ma.  The configuration seems
// to be non-volatile so can also be done using the Beckhoff EtherCAT
// Configurator.  Other parameters such as Monitoring Over/Under range,
// limits, etc., may be of interest and are available in their manual.
//
// Must write to SMC from a running MSB controlling a drive since it
// has no MSB itself.

// Configure all channels for 4-20ma
value = 7;
sdo write value, 2, 0x803b, 0x01, 2; // SMC EX600-AXA channel 1
delay 10ms;
sdo write value, 2, 0x803b, 0x02, 2; // SMC EX600-AXA channel 2
delay 10ms;
sdo write value, 2, 0x804b, 0x01, 2; // SMC EX600-AXA channel 3
delay 10ms;
```

```
sdo write value, 2, 0x804b, 0x02, 2; // SMC EX600-AXA channel 4
delay 10ms;

[run]

// Begin the move, 1 rev/second for 2 revolutions
move at 1 for 2;
wait for in position;
// Delay 1 second once in position
delay 1000 ms;
// Do a relative move back 2 revolutions at 1 rev/second
move at 1 for -2;
wait for in position;
// Delay 1 second once in position
delay 1000 ms;
// Do it again, forever...
goto run;
```

Appendix

T

[T] Turck

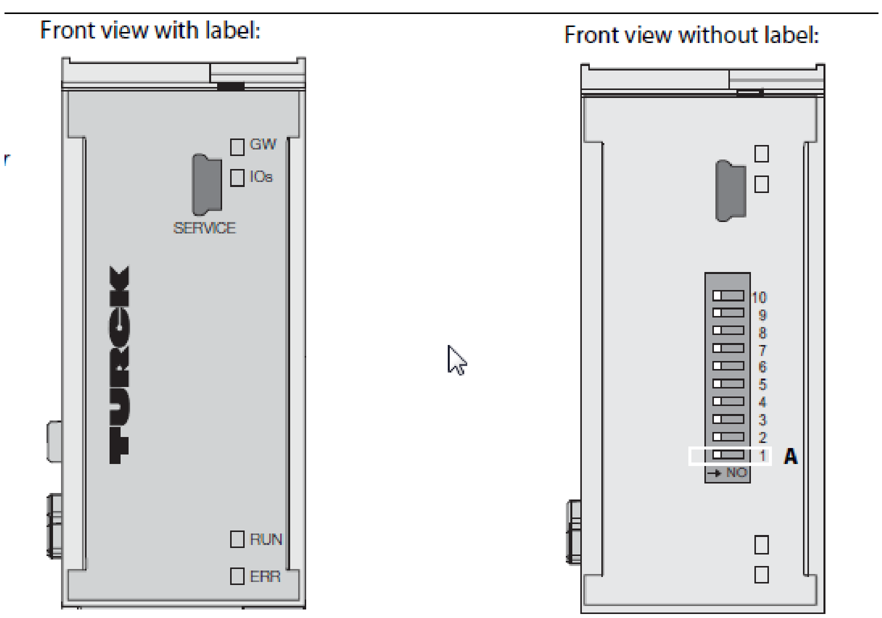
Turck manufactures an IO controller which can operate as a slave device with the M3-41 EtherCAT network. This section discusses the implementation of the RFID reader option within this controller. Much of the information was derived from the Turck “BLident RFID-S Startup Guide”.

Synchronization via hardware using the CFG-Switch

Prior to operation the Turck BL-20 EtherCAT Gateway must have its detected module configuration saved to its configured module list or an error will result preventing proper online operation. This is done by installing the desired modules, powering the unit, removing its plastic label, and setting dip switch #1 ON.

Thh DIP-switches are located under the gateway’s upper label.

For setting the DIP-switch pull out the label.



Switching to ON starts the storage of the Current Configuration as the Required Configuration (Reference configuration).

Procedure:

Switching the DIP-switch no. 1 to ON

→ Starting of storage process

→ LED IOs flashes green (1 Hz)

→ LED IOs shortly lits up orange

→ storage process active

→ set back the DIP-switch

→ storage process terminated successfully, if the LEDs IOs and GW are constant green.

Note

If the DIP-switch is not set back, the gateway will continuously restart the storage process. Only setting the switch back will terminate this process.

Diagnostic LEDs

LED	Status	Meaning	Remedy
GW	OFF	No power supply of the CPU.	Check the system power supply at the gateway.
	green	Firmware active, gateway ready	-
	green flashing, 1 Hz	Firmware not active	If LED "**IOs**" red, then firmware-download necessary
	green flashing, 4 Hz	Firmware active. gateway-hardware-failure	Replace the gateway.
	red	hardware-failure, no communication possible	Replace the gateway.
	red/green flashing, 4 Hz	WINK	WINK-Command active (serves for the identification of the device)
IOs	OFF	No power supply of the CPU.	Check the system power supply at the gateway.
	green	Module bus is running if LED MS green	Configured modules match plugged modules
	green flashing, 1 Hz	Station is in the Force Mode of I/O-ASSISTANT.	Deactivate the Force Mode of the I/O-ASSISTANT
	red	Hardware error	Replace the gateway.
	red flashing, 1 Hz	The actual and the config-ured module list do not match, no communication	Check the physical station for pulled or new but not planned modules.
	red flashing, 4 Hz	No communication via the module bus.	At least one module has to be plugged and has to be able to communicate with the gateway.

LED	Status	Meaning	Remedy
IOs	red/green flashing, 1 Hz	The current and configured module list do not match but the data exchange proceeds as normal.	Check the physical station for pulled or new but not planned modules.
RUN	OFF	The device is in state INITIALIZATION	see EtherCAT®- State Machine (page 3-3)
	green, flashing 200 ms on/ 200 ms off (Blinking)	The device is in state PRE-OPERATIONAL	
	green, flashing 200 ms on/ 1000 ms off (Single Flash)	The device is in state SAFE-OPERATIONAL	
	green	The device is in state OPERATIONAL	
ERR	OFF	Process data exchange	
	red	Critical communication error or controller error occurred	Execute a power-rest, eventually the device has to be changed.
	red, flashing: 200 ms on/ 200 ms off (Blinking)	Invalid configuration	Check if the hardware configuration of your device matches the configured
	red, flashing: 200 ms on/ 1000 ms off (Single Flash)	local error The device switches to the SAFE-OPERATIONAL state due to an internal error (see EtherCAT®- State Machine (page 3-3)).	
LNK/ ACT (left LED)	green	Link established,100 Mbps	
	green, flashing	Data exchange (Ethernet-Traffic 100 Mbps)	
	OFF	no link	Check the Ethernet-connection.

EtherCAT Connector

ETH1 is the EtherCAT input connector (closest to the power connector), ETH2 is the output.

RFID

The M3-41 module is capable of supporting 16 channels of RFID. The Turck BL-2RFID-S supports 2 channels per module; therefore up to 8 modules can be supported. The purpose of the RFID interface is to be able to read and write RFID tags at high speed. Any size RFID tag is supported with a read/write burst of 256 bytes available with MSB programming, offset by a modifiable address register.

The interface to the RFID channel consists of a number of properties which are mapped to the reader interface. The M3-41 adds some unique high speed options such as ensuring the tag goes away and becomes present before reading as well as the option to verify that the tag ID (8 byte license plate unique to all tags) is different on the next tag seen, prior to a read or write operation.

When interfacing with the RFID reader transfers are done in groups of up to 8 bytes per access. This is loaded into two integer arrays that are 32 deep (4 bytes per integer X 2 arrays X 32 deep = 256 bytes). Integer arrays are used since MSB's do not support strings. String manipulation can be done by QuickBuilder using a high speed transfer mechanism built into the MSB 'host read' & 'host write' instructions. The 'host read' & 'host write' instructions have direct byte wide access to the integer arrays and can transfer QuickBuilder strings to/from the RFID buffers as needed. MSB's may also manipulate data themselves at the integer level.

RFID Property Variables

RFID_totalChannels – Read only, represents the total number of RFID channels available in the system.

RFID_channel – Read/write, selects the RFID channel to be operated on by the properties that follow. Entries of 1 to *RFID_totalChannels* are the valid selections, with 0 disabling access. All properties should be initialized to their proper values before setting the RFID_channel to a non-zero value.

RFID_state – Read only, represents the current state of the RFID interface logic state machine as it executes any requests issued by the RFID_control property variable. Possible values are as follows:

```
RFID_OFF                  0
RFID_IDLE                 1

RFID_READING_1            2
RFID_READING_2            3
RFID_READING_WAIT_DONE    4

RFID_WRITING_1            10
RFID_WRITING_2            11
RFID_WRITING_3            12
RFID_WRITING_4            13
RFID_WRITE_DONE           15

RFID_WRITE_WAITTAG        20
RFID_READ_WAITTAG         21

RFID_WAITNOTAG            22
RFID_ERROR                30
```

RFID_error – Read only, Turck specific error where bits 7 to 0 represent the category and bits 15 to 8 are the description. Any time the *RFID_error* property is non-zero an error is present. To clear the error the RFID reader must be reset using the *RFID_control* property RESET bit.

Error_Cat	Error_Desc	Meaning
1	1	Tag memory error (e.g. CRC error)
1	2	Presence error , tag has left the transmission window
1	3	Address or command does not fit the tag characteristics (e.g. memory size)
1	4	Tag is defective, replace tag
1	5	Tag memory overflow
1	6	Unformatted tag
1	7	Inconsistent tag data structure, reformat tag
1	8	Tag within the transmission window does not have the expected UID
1	9	Command not supported by the tag
1	10	Access violation (e.g. block locked) refer to ISO18000-x
1	11-127	Reserved for future profile use
1	128-255	Vendor specific
2	1	Communication timeout at air interface
2	2	More tags within transmission window than allowed
2	3..127	Reserved for future profile use
2	128	CRC error in air interface
2	129..255	Vendor specific
3	1	Incorrect file name
3	2	File does not exist
3	3	The tag type is incorrect or unsuitable for the selected mode of operation, no file system available on tag
3	4	Create command; no more directory entries available.

RFID_address – Read/write, this address is forwarded to the RFID reader to determine where in its memory block to begin accessing data for read and write operations. A value of 0 is the first address. *RFID_address* is auto-incremented after any read or write by the amount in *RFID_bytesTransferred* therefore set it back to the desired start location after each read or write.

RFID_index – Read/write, the index is used to select which RFID_data_readl/RFID_data_readh or RFID_data_writel/RFID_data_writeh array item is to be operated on. Where 0 is the first item, up to 31 (32 array items for 256 bytes total possible).

```
int RFID_data_readl[32];
int RFID_data_readh[32];
int RFID_data_writel[32];
int RFID_data_writeh[32];
```

RFID_data_readl – Read/write, the first 32 bit integer or 4 bytes of data transferred from the RFID tag. This property is an array of 32 deep, indexed by the *RFID_index* property.

int RFID_data_readl[32];

RFID_data_readh – Read/write, the second 32 bit integer or 4 bytes of data transferred from the RFID tag. This property is an array of 32 deep, indexed by the *RFID_index* property in parallel to *RFID_data_readl*.

int RFID_data_readh[32];

RFID_data_writel – Read/write, the first 32 bit integer or 4 bytes of data transferred to the RFID tag. This property is an array of 32 deep, indexed by the *RFID_index* property.

int RFID_data_writel[32];

RFID_data_writeh – Read/write, the second 32 bit integer or 4 bytes of data transferred to the RFID tag. This property is an array of 32 deep, indexed by the *RFID_index* property in parallel to *RFID_data_writel*.

int RFID_data_writeh[32];

RFID_status – Read/write, 32 bit integer with only the first 8 bits reflecting the status as returned by the Turck RFID reader.

	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	0	Done	Busy	Error	Trans_Conn	Trans_On	TP	TFR	Reserved

- *Done* – Slice is ready to receive command. This bit will be off until previous command bit is turned off.
- *Busy* – Slice is currently processing command. This is normally on when transceiver is waiting for a tag to be presented.
- *Error* – Slice has encountered an error during last command. Refer to Error_Cat and Error_Desc for details. This bit is not always set so check RFID_error for nonzero.
- *Trans_Conn* – Transceiver is correctly connected and communicating with the slice.
- *Trans_On* – Transceiver has been turned on by slice.
- *TP* – Tag present; Tag is present in transceiver field. LED on transceiver will blink rapidly.
- *TFR* – Tag Fully Read; Tag has been present in transceiver field long enough so that entire tag memory has been stored in buffer. This bit does not need to be on to indicate a command has been completed.

RFID_control – Read/write, 32 bit integer which is used to request RFID transactions to occur, read and writing different aspects of the tag and transceiver. Some of the bits are defined by Turck but have been enhanced by CTC for additional features.

From Turck:

	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	0	Transceiver	Next	Tag_ID	Read	Write	Tag_Info	Trans_Info	Reset

- *Transceiver* – Turns on and off transceiver. Used only if two transceivers are close enough to cross talk. Otherwise this bit should be set to an "Always On". If transceiver is off, LED will blink slowly (default), and be solid if transceiver is on. It is suggested to always leave this bit set unless low power operation is required.

- *Next* – If turned on while command is processed, the next command run will require a new tag id to enter the field. This bit should not be used instead reference the upper bits supplied by CTC for similar functionality
- *Tag_ID* – Turn on to read the Unique Identifier (UID) from tag. These are always unique for every tag in the world. The bit is cleared automatically when the operation is complete and data is present or error occurred.
- *Read* – Turn on to read data from a tag. The bit is cleared automatically when the operation is complete and data is present or error occurred.
- *Write* – Turn on to write data to a tag. The bit is cleared automatically when the operation is complete and data is present or error occurred.
- *Tag_Info* – Turn on to read information about tag in field, including tag manufacturer and memory available in tag. The bit is cleared automatically when the operation is complete and data is present or error occurred.
- *Trans_Info* – Turn on to read information about transceiver connected to the channel. It can return data such as type of transceiver, hardware and software revisions. The bit is cleared automatically when the operation is complete and data is present or error occurred.
- *Reset* – Turning this bit on will reset any in-process or queued commands. Use this bit to clear any errors that occur. This bit must be cleared manually to remove the device from reset.

CTC bit enhancements for RFID control property:

USER_NO_TAGFIRST – Bit 15, set this bit if no tag present is to be detected prior to starting the requested read or write operation.

USER_READ_TAGID_FIRST – Bit 14, set this bit if the unique tag id is to be read prior to the requested read or write operation. This bit automatically sets bit 5, Tag_ID, during operation.

USER_NEW_TAGID – Bit 12, set this bit in conjunction with Bit 14 when the tag id is to be different than that previously read. The tag id is a unique 8 byte value present on all RFID tags. Prevents mistakenly reading and writing the same tag. Prior to any requested read or write the tag id is read and if different from that previously (RFID_lasttagIDl & RFID_lasttagIDh) it is stored to the properties RFID_tagIDl and RFID_tagIDh, the requested read or write operation will then automatically be completed.

RFID_controlActive – Read only, represents the value actually being transferred to the Turck RFID controller at any moment. During operation bits are set/cleared automatically by the M3-41 module, especially when transferring multiple blocks of data. Useful for diagnostic purposes.

RFID_count – Read only, represents the value actually being transferred to the Turck RFID controller as the needed byte count. This property is automatically set based upon the *RFID_bytesTotal* required. Useful for diagnostic purposes.

	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	0	Transceiver	Next	Tag_ID	Read	Write	Tag_Info	Trans_Info	Reset
	1	Reserved					ByteCount2	ByteCount1	ByteCount0

- ByteCount – These 3 bits represent the number of bytes to Read or Write
  - 000 = 1 byte
  - 001 = 2 bytes
  - 010 = 3 bytes
  - 011 = 4 bytes
  - 100 = 5 bytes
  - 101 = 6 bytes
  - 110 = 7 bytes
  - 111 = 8 bytes

RFID_bytesTotal – Read/write, this property must be set to the total number of bytes to be transferred to/from the RFID reader. If to the reader RFID_data_writel/h array is used, if from the reader RFID_data_readl/h is used. For a single transfer this is typically set to 8. This property is also used in conjunction with the 'host read' and 'host write' commands when transfers are done with the RFID_data_readl and RFID_data_writel properties. These properties can be used to transfer strings to/from Quickbuilder variants.

RFID_bytesTransferred - Read/write, this property represents the number of bytes that have been transferred during a read/write RFID operation as well as 'host read' and 'host write'. If a 'host read' is used to read a QuickBuilder variant string this property will represent the length of the string after the access.

RFID_tagIDl – Read/write, 32 bit integer, this property is automatically set when a TAG ID read operation is performed. This property represents the lower 4 bytes.

RFID_tagIDh – Read/write, 32 bit integer, this property is automatically set when a TAG ID read operation is performed. This property represents the upper 4 bytes.

RFID_lasttagIDl – Read/write, 32 bit integer, this property is automatically set when a TAG ID read operation is performed and a unique id is found, different than that previously read. This property represents the lower 4 bytes. The RFID_lasttagIDl/h is checked against the latest tag id read to ensure no duplicates are found, if that option is enabled.

RFID_lasttagIDh – Read/write, 32 bit integer, this property is automatically set when a TAG ID read operation is performed and a unique id is found, different than that previously read. This property represents the upper 4 bytes. The RFID_lasttagIDl/h is checked against the latest tag id read to ensure no duplicates are found, if that option is enabled.

TAG INFO

This command is initiated on the rising edge of the input bit. The command is executed when a tag is present in the interface. The command returns 8 bytes of information to the "Read_Data" bytes.

- Byte 0: Number of Memory blocks
- Byte 1: Number of Bytes per block
- Byte 2: DSFID
- Byte 3: AFI
- Byte 4: ICID
- Bytes 5-7: Always "0"

TRANS INFO

This command is initiated on the rising edge of the input bit. The command is executed immediately. The command returns 8 bytes of information to the *RFID_data_readl/h* array, with the content dependent on the value of *RFID_address*. Below is an example supplied by Turck.

Address	Data Represents		Example
0xF0	First 8 bytes of transceiver part number		"TNLR-Q80"
0xF1	Second 8 bytes of transceiver part number		"-H1147\0\0"
0xF2	Third 8 bytes of transceiver part number		""
0xF3	Fourth 8 bytes of transceiver part number		""
0xF4	Hardware and Firmware Rev of transceiver		
	Byte 0	x digit of HW Rev (x.y)	
	Byte 1	y digit of HW Rev (x.y)	
	Byte 2	Character "V"	0x56
	Byte 3	x digit of FW Rev (Vx.y.z)	
	Byte 4	y digit of FW Rev (Vx.y.z)	
	Byte 5	z digit of FW Rev (Vx.y.z)	
	Byte 6	Blank	
	Byte 7	Blank	

Strings

MSB's cannot operate directly on strings but QuickBuilder can. Thus to manipulate string data it is first read from the Tag as 1 or more bytes of data, stored in the RFID_data_readl/h data array, and then transferred to QuickBuilder for further processing. This is done using the 'host write' command referencing the RFID_data_readl or RFID_data_writel storage locations. RFID_bytesTotal are the number of bytes to transfer.

Assume a tag was read with 128 bytes (up to 223 bytes may be transferred with 'host read/write' commands):

```
RFID_bytesTotal = 128;
```

```
host write RFID_data_readl,36103;    // Write the integer array as bytes of a
                                     // string, to QB automatically null
                                     // terminated.
```

QB and MSB handshake as needed as the data is manipulated…

```
RFID_bytesTotal = 0;
host write RFID_data_writel,36103;   // Read the modified string, from QB
                                     // and store to tag write array.
                                     // RFID_bytesTotal is set to size.
```

Programming Examples

```
// TURCK RFID TEST Application

/*
RFID_control (Read/Write)

        Bit 0 - Reset: Turning this on will reset any in-process or queued commands.
        Bit 1 - Trans_Info: Turn on to read information about transceiver connected to the
                channel. It can return data such as type of transceiver, hardware and software
                revisions. See section 2.1.5 for details.
        Bit 2 - Tag_Info: Turn on to read information about tag in field, including tag
                manufacturer and memory available in tag. See section 2.1.4 for details.
        Bit 3 - Write: Turn on to write data to a tag.
        Bit 4 - Read: Turn on to read data from a tag.
        Bit 5 - Tag_ID: Turn on to read the Unique Identifier (UID) from tag. These are always
                unique for every tag in the world.
        Bit 6 - Next: If turned on while command is processed, the next command run will require
                a new tag id to enter the field.  Not used.
        Bit 7 - Transceiver: Turns on and off transceiver. Used only if two transceivers are
                close enough to cross talk. Otherwise this bit should be set to an "Always
                On". If transceiver is off, LED will blink slowly (default), and be solid if
                transceiver is on.
        Bit 12, USER_NEW_TAGID, set this bit in conjunction with Bit 14 when the tag id is
                to be different than that previously read.  The tag id is a unique 8 byte value
                present on all RFID tags.  Prevents mistakenly reading and writing the same tag.
                Prior to any requested read or write the tag id is read and if different from that
                previously (RFID_lasttagIDl & RFID_lasttagIDh) it is stored to the properties
                RFID_tagIDl and RFID_tagIDh, the requested read or write operation will then
                automatically be completed.
        Bit 14, USER_READ_TAGID_FIRST, set this bit if the unique tag id is to be read prior
                to the requested read or write operation.  This bit automatically sets bit 5,
                Tag_ID, during operation.
        Bit 15, USER_NO_TAGFIRST, set this bit if no tag present is to be detected prior to
                starting the requested read or write operation.

RFID_count (Read only)
        // ByteCount: These 3 bits represent the number of bytes to Read or Write per block
        // o 000 = 1 byte
        // o 001 = 2 bytes
        // o 010 = 3 bytes
        // o 011 = 4 bytes
        // o 100 = 5 bytes
        // o 101 = 6 bytes
        // o 110 = 7 bytes
        // o 111 = 8 bytes

RFID_status (Read only)

        Bit 0 - Reserved
        Bit 1 - TFR ? Tag Fully Read; Tag has been present in transceiver field long enough
                so that entire tag memory has been stored in buffer. This bit does not need
                to be on to indicate a command has been completed.
        BIT 2 - TP ? Tag present; Tag is present in transceiver field. LED on transceiver
                will blink rapidly.
```

```
        BIT 3 - Trans_On ? Transceiver has been turned on by slice.
        BIT 4 - Trans_Conn ? Transceiver is correctly connected and communicating with the
               slice.
        BIT 5 - Error ? Slice has encountered an error during last command. Refer to
               Error_Cat and Error_Desc for details.
        BIT 6 - Busy ? Slice is currently processing command. This is normally on when
               transceiver is waiting for a tag to be presented.
        BIT 7 - Done ? Slice is ready to receive command. This bit will be off until previous
               command bit is turned off.

RFID_error – Read only, Turck specific error where bits 7 to 0 represent the category
        and bits 15 to 8 are the description.  Any time the RFID_error property is non-zero
        an error is present.  To clear the error the RFID reader must be reset using the
        RFID_control property RESET bit.

// RFID_data_readl - data[0] - data[3]
// RFID_data_readh - data[4]-data[7]

*/

// Example to read QB variant 36102 and then write it back to 36103, row 0, column 0.
RFID_bytesTotal = 0;          // Clear the total bytes to 0, it will be set to actual
                              // size after the read is complete.
// Read the string "This is a test string for RFID." from QB variant 36102
host read RFID_data_writel, 36102;
host write RFID_data_writel,36103;    // Write the string back again, note RFID_bytesTotal was
                                      // set to the total number of bytes to write by prior
                                      // command.  You can verify with Debugger Watch Window.
[restart]
// Disable RFID control prior to activating a channel
RFID_control = 0;      // Init to nothing, which is default.
RFID_state = 1;        // By default the state is RFID_OFF so set to RFID_IDLE for operation.
RFID_bytesTotal = 8;   // Number of bytes to tranfer each time to/from the tag

// Set address of tag to start transfer on
RFID_address = 0;

RFID_channel = 1;      // Set to first channel, this activates logic and scanning

delay 50 ms;           // Delay a bit to let EtherCAT scan update information to reader.

// Turn Transceiver on
RFID_control = RFID_control | 0x0080;
// Wait for acknowledgement back that both the Transceiver is turned on and connected.
[wait0]
// Check to see if device was offlined so we don't hang.
if (RFID_state == 0) goto restart;
// Trans_On/Trans_Conn bits will be set when transceiver is on.
if ((RFID_status & 0x0018)!=0x0018) goto wait0;

// Reset it the Transceiver
RFID_control = RFID_control | 0x0001;
delay 250 ms;                                 // Allow reset to probagate the network
RFID_control = RFID_control & ~0x0001;        // Disable the Reset
delay 250 ms;                                 // Allow Transceiver to come out of reset

// Wait until ready to receive a command
[wait1]
// Check to see if device was offlined so we don't hang.
if (RFID_state == 0) goto restart;
if ((RFID_status & 0x0080)==0) goto wait1;

// Read Trans Info for diagnostics reasons, not needed in actual operation.
RFID_control = RFID_control | 0x0002;

// Wait until done with command
[wait2]
// Check to see if device was offlined so we don't hang.
if (RFID_state == 0) goto restart;
if (RFID_control & 0x0002) goto wait2;
if (RFID_error !=0) goto processTransError;          // Ensure no errors
```

```
// Load the data for possible later use
TransInfo_data_low = RFID_data_readl;
TransInfo_data_high = RFID_data_readh;

/*---------------------------------------------*/
/*
       Test below will wait for a tag to be present then read the Tag ID
       when the tag becomes present, making sure the Tag ID is not the
       same as the previous tag.  If it is go back to waiting for a new
       tag, otherwise write an incrementing number.  Once written, read
       the tag back again.
*/

// Monitor for tag available to read
loopCounter = 1;

// ***** WAIT FOR TAG ******
[writeTag]
// Write the counter value when the tag becomes present
RFID_data_writel = loopCounter;
loopCounter = loopCounter + 1;
// ***** WRITE TAG ******
RFID_address = 0;      // Address is auto-incremented so reset to 0
// Set wait for no tag first, then read tag id and make sure not previous before write
RFID_control= RFID_control | 0x8000 | 0x4000 | 0x1000 | 0x0008;

// Wait for write flag to turn off, meaning writing complete, then check for error.
[waitWrite1]
// Check to see if device was offlined so we don't hang.
if (RFID_state == 0) goto restart;
if (RFID_control & 0x0008) goto waitWrite1;
if RFID_error != 0 goto processWriteErr;

// ***** READ TAG ******
// Read data back now
[readTag]
RFID_address = 0;  // Address is auto-incremented so reset to 0
// Read same tag data back so must disable net Tag flags.
RFID_control = (RFID_control & ~(0x8000 | 0x4000 | 0x1000)) | 0x0010;
// Wait for read flag to turn off, meaning reading complete, then check for error.
[waitRead1]
// Check to see if device was offlined so we don't hang.
if (RFID_state == 0) goto restart;
if (RFID_control & 0x0010) goto waitRead1;  // Wait for command to be accepted.
if RFID_error != 0 goto processReadErr;

// Save the read data away for diagnostic use.
Read_data_low = RFID_data_readl;
Read_data_high = RFID_data_readh;
// Go wait for the next tag
goto writeTag;

[processReadErr]
// Issue reset and save error off
readError = readError + 1;    // Bump number of times read error occured.
goto waitErr;       // GO issue reset and wait for error to clear.

[processWriteErr]
writeError = writeError + 1;  // Bump number of times write error occured.

lastError = RFID_error; // Save error code off.
[waitErr]
RFID_control = 0x0001;   // Transceiver on and reset bit set
[waitClr]
delay 10 ms;
if (RFID_error != 0) goto waitClr;
delay 10 ms;             // Allow to probagate over the network.
RFID_control = 0x0080;   // Transceiver on and reset bit clear
delay 10 ms;             // Allow to probagate over the network.
// Wait for error to clear
```

```
[waitErr1]
// Check to see if device was offlined so we don’t hang.
if (RFID_state == 0) goto restart;
if (RFID_status & 0x0020) goto waitErr;  // Wait for error to go away
if (RFID_status & 0x0040) goto waitErr1;  // Wait for busy to go away
if ((RFID_status & 0x0080) == 0) goto waitErr1;  // Wait for command to be accepted
delay 10 ms;
RFID_state = 1;       // Enable read/write cycles since when error occurs state
                      // machine will hang at RFID_ERROR state for processing.
delay 10 ms;
goto writeTag;       // Go wait for next tag and begin writing again.

// Enter here if have error during Transceiver Info read
[processTransError]
lastError = RFID_error;
[stall]
goto stall;
```

Blank

Appendix

U

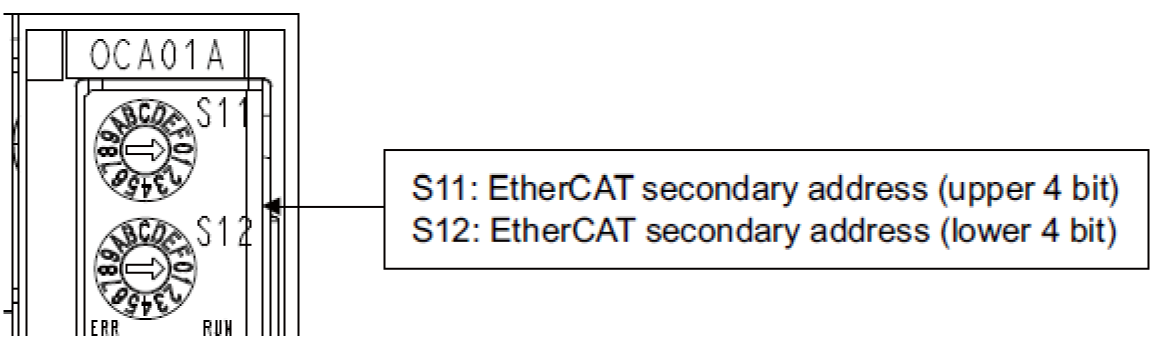
[U] Yaskawa

Yaskawa manufactures a number of drives. That currently supported is the single axis Sigma 5 & 7 rotary and linear drive. This section provides information that may be specific to this manufacturer.

eCAT_driveType – 3

Station Alias

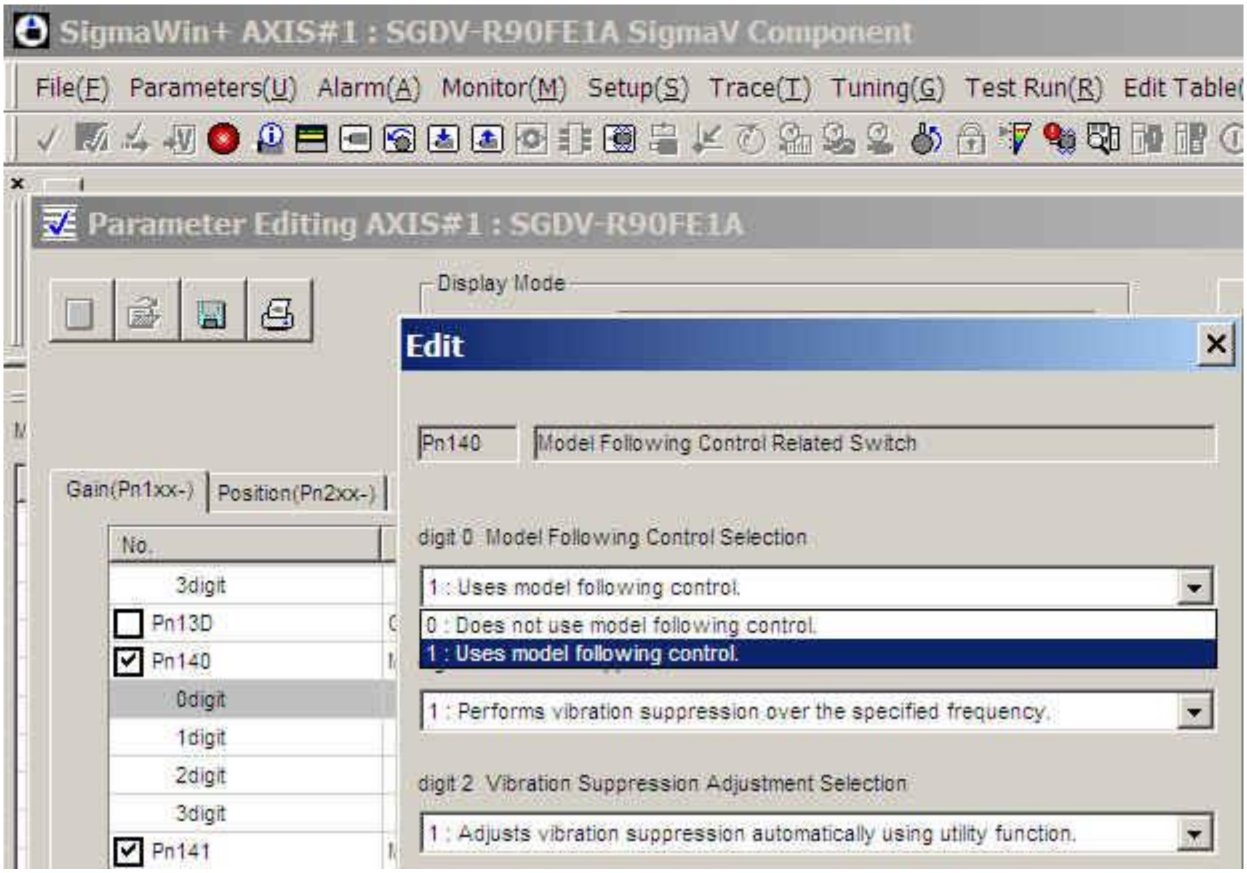
In an EtherCAT network, slaves are automatically assigned addresses based on their position in the bus. When a device, such as a drive, must have a fixed assigned identification that is independent of cabling, a Station Alias is needed. Yaskawa provides two 16-position rotary switches with hexadecimal encoding for this purpose. This allows for a setting of 0 to 255 (FFh), where 0 defaults to the automatic address assignment. As an example, if S11 is set to a 1 and S12 to an A this would be 1Ah or 1 X 16 + 10 = 26. Since the M3-41 only supports up to 16 drives S11 would always be set to 0 and only S12 used.



Yaskawa Position Lag & perr

By default Yaskawa uses what it terms 'Model Following Control' in both CSP and interpolated moves. This causes the actual position of the drive to lag from the desired target more than expected (typically 10X that of other drives). This actually provides for a smoother move. Yaskawa delays the move on purpose to better figure out the profile the Master wants to execute and then provides smoothing. When not gearing to dissimilar drives this is a good thing, since at the end of the move you will always be at the correct position, although it takes a bit longer and 'perr' will build up.

In some applications, it is desirable to reduce the lag between the commanded position and the actual position reported back by the drive. Do this by disabling 'Model Following Control' using SigmaWin. This is parameter Pn140.0 as shown below.



Note that Pn141, Model Following Gain may also need additional adjustment from its default value.

Drive IO Connector Mapping

The Yaskawa Sigma 5 drive has a number of inputs and outputs available for MSB control on the drive. The following table defines their connector mapping to MSB drive input/outputs.

Name	Yaskawa Pin	MSB Assignment
General Purpose Input	SI0	Din1 (inputs[1])
Forward Run Prohibit Input	SI1	Din2
Reverse Run Prohibit Input	SI2	Din3
General Purpose Input	SI3	Din4
Probe 1 Latch Signal Input	SI4	Din5
Probe 2 Latch Signal Input	SI5	Din6
Home Switch Input	SI6	Din7
Brake Output (option)	SO1	Out1 (outputs[1])
General Purpose Output	SO2	Out2
General Purpose Output	SO3	Out3

MSB ‘errorRegister’ Variable Value Definitions

Reference the specific drive Manufacture for object 0x1001 definitions. For Yaskawa the definitions are as follows:

Bit 0 – Generic Error, 0: No error, 1: Error
Bit 1 to 7 – Reserved – 0: Always.

MSB ‘errorCode’ Variable Value Definitions

Reference the specific drive Manufacture for specific errorCode definitions. For Yaskawa object 0x603f is referenced with values given in hexidecimal:

errorCode	Description
0x0A10	The Sync0 event and the SERVOPACK cannot be synchronized.
0x0A11	The EtherCAT AL state became not 'Operational' while the DS402 drive state is in 'Operation enabled.
0x0A12	The events, receive process data and sync0, do not synchronize. (Failed to receive the process data.)
0x0A20	The parameter setting is out of range.
0x0A40	The initialization at power on sequence was

errorCode	Description
	failed.
0x0EA2	The data exchange between the EtherCAT (CoE) Network Module and the SERVOPACK was not synchronized.
0x0510	The servomotor speed is excessively high.
0x0511	The motor speed upper limit of the set encoder output pulse (pulse unit) (Pn212) is exceeded.
0x910	The motor was operating continuously under a torque largely exceeding ratings.
0x0720	The motor was operating continuously under a torque largely exceeding ratings.
0x710	The motor was operating for several seconds to several tens of seconds under a torque largely exceeding ratings.
0x911	Vibration at the motor speed was detected.
0x520	Vibration at the motor speed was detected.
0x0d00	Position error pulses exceeded parameter (Pn520).
0x0d01	Position error pulses accumulated too much.
0x0d30	Position data exceeded +/- 1879048192.
0x0cc0	Different multi-turn limits have been set in the encoder and the SERVOPACK.
0x0f10	With the main power supply ON, voltage was low for more than 1 second in phase-R, -S or -T.
0x????	Undefined, see manual.

EtherCAT Explorer View

Manuf	Yaskawa
Grp	Drive
Name	SIGMA 5 Rotary
Out	192 bits (24 bytes)
In	240 bits (30 bytes)
Axis #	1
pstate	RUNNING (1)
tracking_pstate	COMPLETE (2)
inpos	0
fpos	9.519889
tpos	9.556000
perr	0.035112
vel	1.683844
DRV MODE	Cyclic Sync Position (8)
PDO STATUS	0x1237
PDO CNTLWORD	0x000F
PDO ACT VEL	0x001AF106
PDO ACT TORQ	0x00000005
PDO ACT ERR	0x00008280
PDO HOME PWRUP	0x000E5D2D
PDO ACT POS	0x00A6AEA4
PDO TARG POS	0x00A7428D
PDO TARG VEL	0x00000000
PDO DIG INP	0x00000006
State	8 (OPERATIONAL)
Delay	0 ns
Has DC	true (32 bits)
DC Parent	0
DC Active	true, Cyc time: 1000000 ns, Shft: 0
Parent	0
Config addr	0x1001 (4097)
Station Alias	0
Vndr	0x00000539 (1337)
Product Code	0x02200001 (35651585)
Rev	0x00030005

Blank

Appendix

V

[V] Yaskawa V1000 VFD

Yaskawa manufactures a number of drives. This section discusses the V1000 VFD drive and provides information that may be specific to this manufacturer.

eCAT_driveType – 16

Do not run as the first drive for EtherCAT system clock if servo drives are also in the network.

Sample Program

The V1000 is a velocity only drive with units as rpm

```
[Yaskawa_V1000]
cmode = $VELOCITY_MODE;  // The V1000 only supports velocity mode
dec = 1000;  // Deceleration speed in rpm, usually time is 10 seconds and set
with DriveWizard.  Deceleration = speed/time
acc = 1000;  // Acceleration speed in rpm, usually time is 10 seconds and set
with DriveWizard.  Acceleration = speed/time.
vmax = 1000; // Max Velocity in rpm only set when enable drive, default drive
dependent.
vmin = 0;    // Min Velocity in rpm only set when enable drive
ever = 0;    // Variable to make move code more readable
goto beginTest;

[beginTest]
// Initialize distributed clocks
delay 3000 ms;
dcsync -1, 1000000, 0, 0, 100000000;
delay 200 ms;     // starts 100 milliseconds into the future
drive enable;
zero feedback position;
ever = 0;

[run]

if eCAT_driveType == $DRIVE_YASKAWA_V1000 goto vfdMotion;
goto normalMotion;

[vfdMotion]
```

```
// Begin the move at 100 rpm
move at 100 for ever;
wait for in position;   // Wait for the motor to reach the desired velocity.
delay 6000 ms;          // Maintain speed for 6 seconds.
move at -200 for ever;  // Decelerate to 0 and move other direction at 200 rpm.
wait for in position;   // Wait for the motor to reach the desired velocity.
delay 4000 ms;          // Maintain speed for 6 seconds.
stop;                   // Stop the drive and decelerate to 0 rpm.
wait for in position;   // Wait for drive to stop.
delay 3000 ms;    // Stay stopped for 3 seconds.
goto run;         // Loop test…

[normalMotion]
```
… other drive motion code …

Note: MSB Variables 'v_accel_time' and 'v_decel_time' are used to set the number of seconds before the 'acc' and 'dec' speeds take, thus generating the acceleration and deceleration ramps. They are typically initialized to the defaults of the drive at power up but may be changed as desired since they are constantly updated to the drive with each PDO cyclic data transfer.

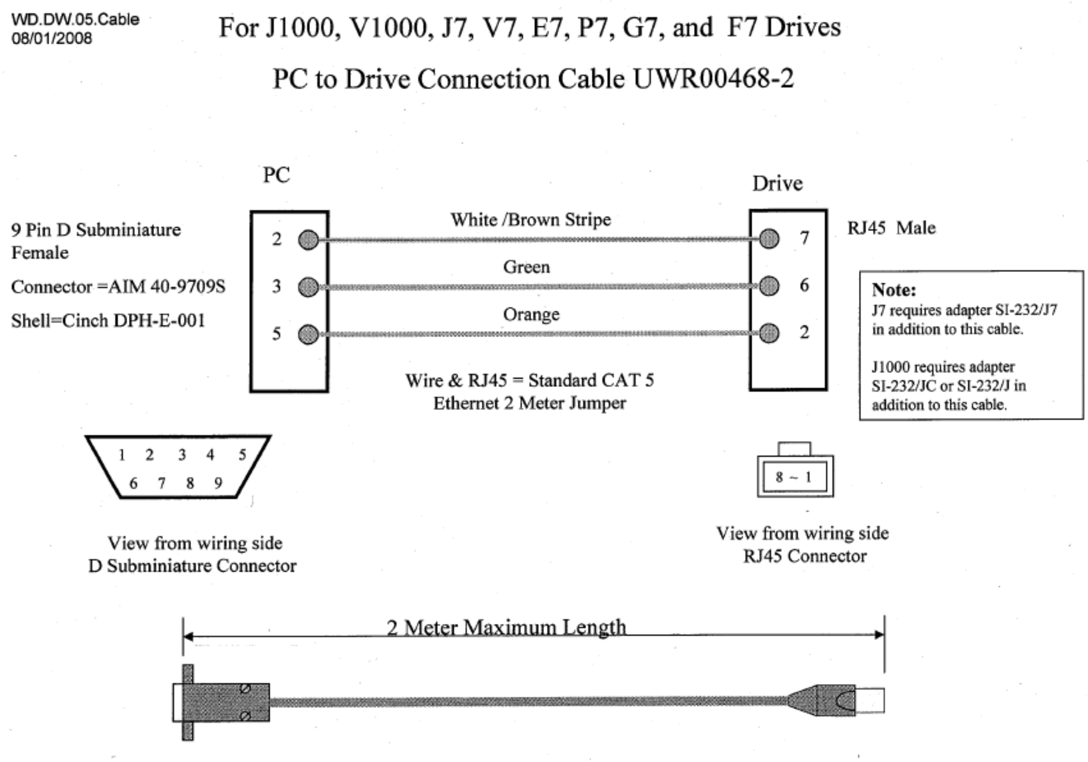
DriveWizard Windows Software

DriveWizard is a software program available from Yaskawa which is used to view and modify the parameters within a V1000 drive.

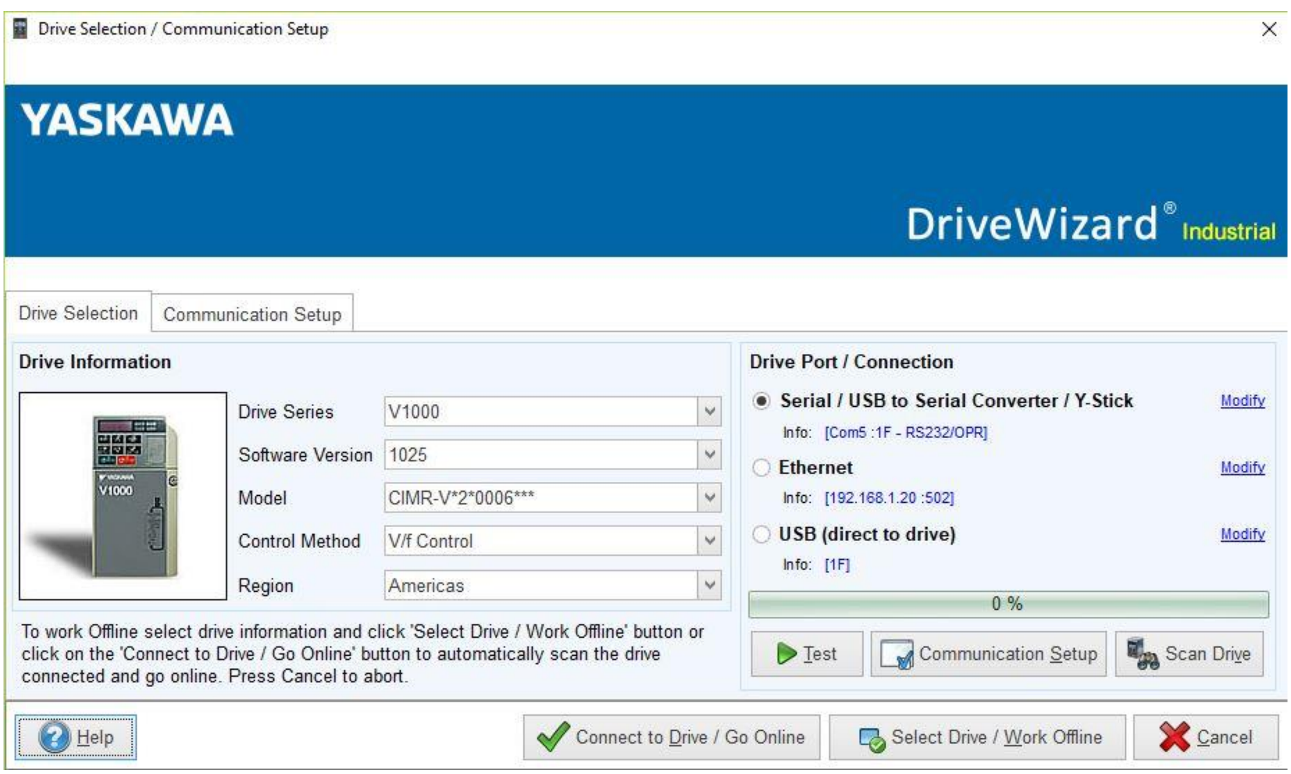
Click on the link to download: DriveWizard Industrial - A1000, P1000, U1000, V1000, J1000, D1000, R1000, F7, G7, P7
or copy/paste the URL into your browser window:
https://www.yaskawa.com/documents/20197/890819/sw.dw.30.zip/63d30433-5004-4533-83a9-5015513b3688

VIDEO: DriveWizard Industrial Installation and Connection

Communications to the drive can be by a serial port with either a cable purchased from Yaskawa or pinned as below:



In order to go online the communications setup must be configured similar to:



Reference the Yaskawa documentation for proper use of the DriveWizard program. The default settings that were used in the drive tested have been included in the following section for your reference.

Yaskawa V1000 Parameter Settings As Tested

Below are the settings used for initial testing as viewed with DriveWizard. That highlighted in yellow are specific to EtherCAT. Note b1-01 and b1-02 must be set to ‘option card’ (3), e2-04 to the number of poles of for the motor and o1-03 to (2). Other items highlighted should also be set as they may have values left over from prior standalone configurations.

Drive Type / Model:	V1000 CIMR-V*2*0006***
Software:	1025

Parameter		Value	Information	Default Setting
A1 Initialization				
A1-01	Access level	2	Advanced Level Access	2
A1-02	Control method	0	V/f control	0
A1-04	Enter password	0		0
A1-05	Select password	0		0
A1-06	Select application	0	General	0
A1-07	DWEZ Function Selection	0	Disabled	0
A2 User Parameters				
*A2-01	User parameter 1... **[M]**	Not Mapped		A1-02
*A2-02	User parameter 2... **[M]**	Not Mapped		b1-01
*A2-03	User parameter 3... **[M]**	Not Mapped		b1-02
*A2-04	User parameter 4... **[M]**	Not Mapped		b1-03
*A2-05	User parameter 5... **[M]**	Not Mapped		C1-01
*A2-06	User parameter 6... **[M]**	Not Mapped		C1-02
*A2-07	User parameter 7... **[M]**	Not Mapped		C6-01
*A2-08	User parameter 8... **[M]**	Not Mapped		C6-02
*A2-09	User parameter 9... **[M]**	Not Mapped		d1-01
*A2-10	User parameter 10... **[M]**	Not Mapped		d1-02
*A2-11	User parameter 11... **[M]**	Not Mapped		d1-03
*A2-12	User parameter 12... **[M]**	Not Mapped		d1-04
*A2-13	User parameter 13... **[M]**	Not Mapped		d1-17
*A2-14	User parameter 14... **[M]**	Not Mapped		E1-01
*A2-15	User parameter 15... **[M]**	Not Mapped		E1-03
*A2-16	User parameter 16... **[M]**	Not Mapped		E1-04
*A2-33	User parameter Automatic registr... **[M]**	1	Automatic registration is available (The constants used in A2-17~A2-32 are memorized Automatically)	0
b1 Operation Mode Selection				
*b1-01	Reference selection... **[M]**	3	Option card	1
*b1-02	Operation method	3	Option card	1
b1-03	Stopping method	0	Deceleration to stop	0
b1-04	Reverse operation	0	Reverse enabled	0
b1-07	Local/remote run selection	0	Cycle External Run - If the run command is closed	0

b1-08	Run command at	0	Cannot operate	0
b1-14	Phase Turn Selection	0	Normal	0
b1-15	Reference Selection in	0	Digital Operator	0

Parameter		Value	Information	Default Setting
b1-16	Operation method selection in Local case	0	Digital Operator	0
b1-17	Operation permission with power ON/OFF	0	Prohibition	0
b2 DC Injection Braking				
b2-01	DC injection start frequency	0.5 Hz		0.5 Hz
b2-02	DC injection current	50 %		50 %
b2-03	DC injection time at start	0.00 sec		0.00 sec
b2-04	DC injection time at stop	0.50 sec		0.50 sec
b3 Speed Search				
b3-01	Speed search at start	0	Disabled	0
b3-02	Speed search current	120 %		120 %
b3-03	Speed search deceleration time	2.0 sec		2.0 sec
b3-05	Search wait time	0.2 sec		0.2 sec
b3-06	Output current	0.5 A		0.5 A
b3-08	ACR Gain for Speed Search	0.50		0.50
b3-10	Speed Search Detection Compensation	1.05		1.05
b3-14	Bidirectional search selection	0	Disabled	0
b3-17	Retry current level	150 %		150 %
b3-18	Retry detection time	0.10 sec		0.10 sec
b3-19	Retry count	3		3
b3-24	Speed search method selection	0	Current detection type	0
b3-25	Speed search retry interval time	0.5 sec		0.5 sec
b4 Timer Function				
b4-01	Delay-ON timer	0.0 sec		0.0 sec
b4-02	Delay-OFF timer	0.0 sec		0.0 sec
b5 PID Control				
b5-01	PID control mode selection	0	Disabled	0
b5-02	PID P gain	1.00		1.00
b5-03	PID I time	1.0 sec		1.0 sec
b5-04	PID I limit	100.0 %		100.0 %
b5-05	PID D time	0.00 sec		0.00 sec
b5-06	PID limit	100.0 %		100.0 %
b5-07	PID offset adjustment	0.0 %		0.0 %
b5-08	PID primary delay time constant	0.00 sec		0.00 sec
b5-09	Output level selection	0	PID output is forward.	0
b5-10	PID output gain	1.00		1.00
b5-11	PID output reverse selection	0	0 limit when PID putput is negative	0
b5-12	Feed back loss detection selection	0	Disabled. No detection PID feedback	0
b5-13	Feed back loss detection level	0 %		0 %
b5-14	Feed back loss detection time	1.0 sec		1.0 sec
b5-15	PID sleep level	0.0 Hz		0.0 Hz
b5-16	PID sleep time	0.0 sec		0.0 sec
b5-17	Accel/Decel time for PID reference	0 sec		0 sec
b5-18	PID Setpoint Selection	0	Disabled	0

b5-19	PID Setpoint Value	0.00 %		0.00 %
b5-20	PID Setpoint Scaling	1	0.01% units	1
b5-34	PID Output Lower Limit	0.0 %		0.0 %
b5-35	PID Input Limit	1000.0		1000.0 %
b5-36	PID Feedback High Detection Level	100 %		100 %
b5-37	PID Feedback High Detection Time	1.0 sec		1.0 sec
b5-40	PID Fref monitor Selection	0	Fref Monitor with PID	0
b5-47	PID Output Reverse Selection2	1	Reverses when PID output is negative.	1
b6 Dwell Function				
b6-01	Dwell frequency at start	0.0 Hz		0.0 Hz

Parameter		Value	Information	Default Setting
b6-02	Dwell time at start	0.0 sec		0.0 sec
b6-03	Dwell frequency at stop	0.0 Hz		0.0 Hz
b6-04	Dwell time at stop	0.0 sec		0.0 sec
b8 Energy Saving				
b8-01	Energy-saving mode selection	0	Disabled	0
b8-04	Energy-saving coefficient	196.6		196.6
b8-05	kW filter time	20 ms		20 ms
b8-06	Search V limit	0 %		0 %
C1 Acceleration and Deceleration Times				
C1-01	Acceleration time 1	10.0 sec		10.0 sec
C1-02	Deceleration time 1	10.0 sec		10.0 sec
C1-03	Acceleration time 2	10.0 sec		10.0 sec
C1-04	Deceleration time 2	10.0 sec		10.0 sec
C1-05	Acceleration time 3 for motor2	10.0 sec		10.0 sec
C1-06	Deceleration time 3 for motor2	10.0 sec		10.0 sec
C1-07	Acceleration time 4 for motor2	10.0 sec		10.0 sec
C1-08	Deceleration time 4 for motor2	10.0 sec		10.0 sec
C1-09	Emergency stop time	10.0 sec		10.0 sec
C1-10	Accel/Decel time setting unit	1	0.1-second units	1
C1-11	Accel/Decel switching frequency	0.0 Hz		0.0 Hz
C1-14	Picking up speed and slowing down rate	0.0 Hz		0.0 Hz
C2 S-Curve Characteristics				
C2-01	S-curve acceleration at start	0.20 sec		0.20 sec
C2-02	S-curve acceleration at end	0.20 sec		0.20 sec
C2-03	S-curve deceleration at start	0.20 sec		0.20 sec
C2-04	S-curve deceleration at end	0.00 sec		0.00 sec
C3 Slip Compensation				
C3-01	Slip compensation gain	0.0		0.0
C3-02	Slip compensation time	2000 ms		2000 ms
C3-03	Slip compensation limit	200 %		200 %
C3-04	Slip compensation regeneration	0	Disabled	0
C4 Torque Compensation				
C4-01	Torque compensation gain	1.00		1.00
C4-02	Torque compensation time	200 ms		200 ms
C6 Carrier Frequency				
C6-01	Duty Cycle	1	ND(VT)	1

Parameter		Value	Information	Default Setting
C6-02	Carrier frequency selection	07	Swing-PWM1	07
d1 Frequency Reference				
d1-01	Frequency reference 1	0 rpm		0 rpm
d1-02	Frequency reference 2	0 rpm		0 rpm
d1-03	Frequency reference 3	0 rpm		0 rpm
d1-04	Frequency reference 4	0 rpm		0 rpm
d1-05	Frequency reference 5	0 rpm		0 rpm
d1-06	Frequency reference 6	0 rpm		0 rpm
d1-07	Frequency reference 7	0 rpm		0 rpm
d1-08	Frequency reference 8	0 rpm		0 rpm
d1-09	Frequency reference 9	0 rpm		0 rpm
d1-10	Frequency reference 10	0 rpm		0 rpm
d1-11	Frequency reference 11	0 rpm		0 rpm
d1-12	Frequency reference 12	0 rpm		0 rpm
d1-13	Frequency reference 13	0 rpm		0 rpm
d1-14	Frequency reference 14	0 rpm		0 rpm
d1-15	Frequency reference 15	0 rpm		0 rpm
d1-16	Frequency reference 16	0 rpm		0 rpm
*****d1-17	Jog frequency reference... **[M]**	180 rpm		600 rpm
d2 Frequency Upper and Lower				
d2-01	Frequency reference upper limit	100.0 %		100.0 %
d2-02	Frequency reference lower limit	0.0 %		0.0 %
d2-03	Master speed reference lower	0.0 %		0.0 %
d3 Jump Frequency				
d3-01	Jump frequency 1	0.0 Hz		0.0 Hz
d3-02	Jump frequency 2	0.0 Hz		0.0 Hz
d3-03	Jump frequency 3	0.0 Hz		0.0 Hz
d3-04	Jump frequency width	1.0 Hz		1.0 Hz
d4 Frequency Reference Hold				
d4-01	MOP reference memory	0	Disabled	0
d4-03	Frequency reference bias	0.00 Hz		0.00 Hz
d4-04	Accel/Decel rate	0	Bias value Accel/Decel at the rate of current	0
d4-05	Bias mode selection(up/down2)	0	Hold bias value while both of UP/DOWN	0
d4-06	Bias value(up/down2)	0.0 %		0.0 %
d4-07	Analog change limit	1.0 %		1.0 %
d4-08	Frequency Reference Bias	100.0 %		100.0 %
d4-09	Frequency Reference Bias	0.0 %		0.0 %
d4-10	Up/Down Frequency Reference	0	The lower limit is determined by d2-02 or analog	0
d4-11	Bi-directional Output Selection	0	Disabled	0
d4-12	Stop Position Gain	1.00		1.00
d7 Offset Frequency				
d7-01	Offset frequency 1	0.0 %		0.0 %
d7-02	Offset frequency 2	0.0 %		0.0 %
d7-03	Offset frequency 3	0.0 %		0.0 %
E1 Motor 1 V/f Pattern				

E1-01	Input voltage setting	230 VAC		230 VAC
E1-03	V/F pattern selection	0F	User-defined V/f pattern	0F
E1-04	Max. output frequency	60.0 Hz		60.0 Hz
E1-05	Max. voltage	230.0 VAC		230.0 VAC
E1-06	Base frequency	60.0 Hz		60.0 Hz
E1-07	Mid output frequency	3.0 Hz		3.0 Hz
E1-08	Mid output frequency voltage	18.4 VAC		18.4 VAC
E1-09	Min. output frequency	1.5 Hz		1.5 Hz
E1-10	Min. output frequency voltage	13.8 VAC		13.8 VAC
E1-11	Mid output frequency 2	0.0 Hz		0.0 Hz
E1-12	Mid output frequency voltage 2	0.0 VAC		0.0 VAC
E1-13	Base voltage	0.0 VAC		0.0 VAC
E2 Motor 1 Setup				
E2-01	Motor rated current	4.90 A		4.90 A
E2-02	Motor rated slip	2.60 Hz		2.60 Hz
E2-03	Motor no-load current	2.30 A		2.30 A
E2-04	Number of motor poles	4		4
E2-05	Motor line-to-line resistance	3.577 Ohm		3.577 Ohm
E2-06	Motor leak inductance	18.5 %		18.5 %
E2-10	Torque compensation iron loss	38 W		38 W
E2-11	Motor rated output	1.10 kW		1.10 kW
E5 PM Motor Setup				
E5-39	Current Detection Delay Time	0 us		0 us
F6 Option Card 1 Serial				
F6-01	Operation selection after communications error	1	Coast to stop	1

Parameter		Value	Information	Default Setting
F6-02	Selection of External Fault from Communication	0	Always detect	0
F6-03	Stopping Method for External Fault from	1	Coast to Stop	1
F6-04	BUS error detection delay	2.0 sec		2.0 sec
F6-07	Fref Priority Selection	1	MultiStep	1
F6-08	Comm Parameter Initialize Selection	0	Comm	0
F6-10	Node Address	0		0
F6-11	Communication Speed	0	156 Kbps	0
F6-14	BUS Error Auto Reset	0	Disabled	0
F6-20	M-II Station Address	21		21
F6-21	M-II Frame length	0	32byte mode	0
F6-22	M-II Link Speed	0	10 Mbps	0
F6-23	M-II Mon E register	0000		0000
F6-24	M-II Mon F register	0000		0000
F6-25	M-II WDT error selection	1	Coast to stop	1
F6-26	M-II Number of bUS error detection	2		2
F6-30	Node Address	0		0
F6-31	Clear Mode Selection	0	Resets back to	0
F6-32	ProfibusMap Selection	0	PPO Type	0
F6-35	CanOpen Node Address	0		0
F6-36	Communication Speed	6	500 Kbps	6

F6-40	Node Address	0		0
F6-41	Communication Speed	1	Auto-adjust	1
F6-50	MAC Address	64		64
F6-51	Baud Rate	4	Detect	4
F6-52	PCA Setting	21		21
F6-53	PPA Setting	71		71
F6-54	Idle Mode Fault Detection	0	Stop	0
F6-56	Speed Scaling	0		0
F6-57	Current Scaling	0		0
F6-58	Torque Scaling	0		0
F6-59	Power Scaling	0		0
F6-60	Voltage Scaling	0		0
F6-61	Time Scaling	0		0
F6-62	Heart Beat	0		0
F6-64	Dynamic Output Assembly 109 Parameter 1	0000		0000
F6-65	Dynamic Output Assembly 109 Parameter 2	0000		0000
F6-66	Dynamic Output Assembly 109 Parameter 3	0000		0000
F6-67	Dynamic Output Assembly 109 Parameter 4	0000		0000
F6-68	Dynamic Input Assembly 159 Parameter 1	0000		0000
F6-69	Dynamic Input Assembly 159 Parameter 2	0000		0000
F6-70	Dynamic Input Assembly 159 Parameter 3	0000		0000
F6-71	Dynamic Input Assembly 159 Parameter 4	0000		0000
F6-72	Node Address	0		0
F7 Option Card 2				
F7-01	IP Address1	192		192
F7-02	IP Address2	168		168
F7-03	IP Address3	1		1
F7-04	IP Address4	20		20
F7-05	Subnet Mask1	255		255
F7-06	Subnet Mask2	255		255
F7-07	Subnet Mask3	255		255
F7-08	Subnet Mask4	0		0
F7-09	Gateway Address1	192		192

Parameter		Value	Information	Default Setting
F7-10	Gateway Address2	168		168
F7-11	Gateway Address3	1		1
F7-12	Gateway Address4	1		1
F7-13	Address Startup Mode	2		2
F7-14	Duplex Mode Setting	1	Auto Negotiate	1
F7-15	Speed Mode Setting	10	10 Mbps speed setting	10
F7-16	Timeout Value	0 sec		0 sec
F7-17	Speed Scaling	0		0
F7-18	Current Scaling	0		0
F7-19	Torque Scaling	0		0
F7-20	Power Scaling	0		0
F7-21	Voltage Scaling	0		0

F7-22	Time Scaling	0	0
F7-23	Dynamic Output Assembly 116 Parameter 1	0000	0000
F7-24	Dynamic Output Assembly 116 Parameter 2	0000	0000
F7-25	Dynamic Output Assembly 116 Parameter 3	0000	0000
F7-26	Dynamic Output Assembly 116 Parameter 4	0000	0000
F7-27	Dynamic Output Assembly 116 Parameter 5	0000	0000
F7-28	Dynamic Output Assembly 116 Parameter 6	0000	0000
F7-29	Dynamic Output Assembly 116 Parameter 7	0000	0000
F7-30	Dynamic Output Assembly 116 Parameter 8	0000	0000
F7-31	Dynamic Output Assembly 116 Parameter 9	0000	0000
F7-32	Dynamic Output Assembly 116 Parameter 10	0000	0000
F7-33	Dynamic Input Assembly 166 Parameter 1	0000	0000
F7-34	Dynamic Input Assembly 166 Parameter 2	0000	0000
F7-35	Dynamic Input Assembly 166 Parameter 3	0000	0000
F7-36	Dynamic Input Assembly 166 Parameter 4	0000	0000
F7-37	Dynamic Input Assembly 166 Parameter 5	0000	0000
F7-38	Dynamic Input Assembly 166 Parameter 6	0000	0000
F7-39	Dynamic Input Assembly 166 Parameter 7	0000	0000
F7-40	Dynamic Input Assembly 166 Parameter 8	0000	0000
F7-41	Dynamic Input Assembly 166 Parameter 9	0000	0000
F7-42	Dynamic Input Assembly 166 Parameter 10	0000	0000
F7-60	PZD1 Write	0000	0000
F7-61	PZD2 Write	0000	0000
F7-62	PZD3 Write	0000	0000
F7-63	PZD4 Write	0000	0000
F7-64	PZD5 Write	0000	0000
F7-65	PZD6 Write	0000	0000
F7-66	PZD7 Write	0000	0000
F7-67	PZD8 Write	0000	0000
F7-68	PZD9 Write	0000	0000
F7-69	PZD10 Write	0000	0000
F7-70	PZD1 Read	0000	0000
F7-71	PZD2 Read	0000	0000
F7-72	PZD3 Read	0000	0000
F7-73	PZD4 Read	0000	0000
F7-74	PZD5 Read	0000	0000
F7-75	PZD6 Read	0000	0000
F7-76	PZD7 Read	0000	0000
F7-77	PZD8 Read	0000	0000
F7-78	PZD9 Read	0000	0000
F7-79	PZD10 Read	0000	0000

Parameter		Value	Information	Default Setting
H1 Multi-Function				
H1-01	Terminal S1 function selection	40	Forward Run Command	40
H1-02	Terminal S2 function selection	41	Reverse Run Command	41
H1-03	Terminal S3 function selection	24	External fault, N/O Detect always, coast	24
H1-04	Terminal S4 function selection	14	Fault reset	14
H1-05	Terminal S5 function selection	03	Multi-step speed reference 1	03

H1-06	Terminal S6 function selection	04	Multi-step speed reference 2	04
H1-07	Terminal S7 function selection	06	Jog frequency reference	06
H2 Multi-Function				
H2-01	Terminal MA/MB-MC Selection	000E	Fault	000E
H2-02	Terminal P1 Selection	0000	During run	0000
H2-03	Terminal P2 Selection	0002	Frequency agree 1	0002
H2-06	Output unit selection	0	0.1kWh unit	0
H3 Multi-Function				
H3-01	Terminal A1 Signal Level Selection	0	0 to +10 V	0
H3-02	Terminal A1 Function Selection	00	Frequency bias	00
H3-03	Terminal A1 Gain Setting	100.0 %		100.0 %
H3-04	Terminal A1 Bias Setting	0.0 %		0.0 %
H3-09	Terminal A2 Signal Level Selection	2	4 to 20 mA (9 bit input)	2
H3-10	Terminal A2 Function Selection	00	Frequency bias	00
H3-11	Terminal A2 Gain Setting	100.0 %		100.0 %
H3-12	Terminal A2 Bias Setting	0.0 %		0.0 %
H3-13	Filter Avg time	0.03 sec		0.03 sec
H3-14	Terminal Analog Input Selection(when	7	Terminal A1 and A2 Enable	7
H3-16	A1 Offset	0		0
H3-17	A2 Offset	0		0
H4 Multi-Function Analog Outputs				
H4-01	Multi-Function Analog 1(Analog	102	Output frequency	102
H4-02	Multi-Function Analog 1(Analog	100.0 %		100.0 %
H4-03	Multi-Function Analog 1(Analog	0.0 %		0.0 %
H5				
H5-01	Station address	1F		1F
H5-02	Communication speed selection	3	9600 bps	3
H5-03	Communication parity selection	0	No parity	0
H5-04	Serial fault selection	3	Operation continued	3
H5-05	CE Detection Selection	1	Enabled	1
H5-06	Send wait time	5 ms		5 ms
H5-07	RTS control ON/OFF	1	Enabled	1
H5-09	CE detection time	2.0 sec		2.0 sec
H5-10	Output voltage monitor(Register No.25)	0	0.1V unit	0
H5-11	ENTER function selection of transmission	1	In the case of changing a constant, constant is reflected and be memorized by inputting ENTER	1
H5-12	Run Command Method Selection	0	FWD/STOP,REV/STOP type	0
H6 Pulse Train				
H6-01	Pulse train input function selection	0	Frequency reference	0
H6-02	Pulse train input scaling	1440 Hz		1440 Hz
H6-03	Pulse train input gain	100.0 %		100.0 %
H6-04	Pulse train input bias	0.0 %		0.0 %
H6-05	Pulse train input filter time	0.10 sec		0.10 sec
H6-06	Pulse train monitor selection	102	Output frequency	102
H6-07	Pulse train monitor scaling	1440 Hz		1440 Hz
H6-08	Pulse Train Min Input Frequency	0.5 Hz		0.5 Hz
L1 Protection				
L1-01	Motor protection selection	1	General-purpose motor protection	1

Parameter		Value	Information	Default Setting
L1-02	Motor protection time constant	1.0 min		1.0 min
L1-03	MOL thermistor input	3	Operation continued	3
L1-04	MOL filter time	1	Coast to stop	1
L1-05	MOL reserved 1	0.20 sec		0.20 sec
L1-13	Electronic thermal continues	1	Continue Electronic thermal	1
L1-22	Leakage Current Filter Time Constant 1	0.0 sec		0.0 sec
L1-23	Leakage Current Filter Time Constant 2	0.0 sec		0.0 sec
L2 Momentary				
L2-01	Momentary power loss detection	0	Disabled	0
L2-02	Momentary power loss ridethru time	0.2 sec		0.2 sec
L2-03	Min. baseblock time	0.4 sec		0.4 sec
L2-04	Voltage recovery time	0.3 sec		0.3 sec
L2-05	Undervoltage detection level	190 VDC		190 VDC
L2-06	KEB deceleration time	0.0 sec		0.0 sec
L2-07	Momentary recovery time	0.0 sec		0.0 sec
L2-08	KEB Frequency	100 %		100 %
L2-11	Desired DC Bus Voltage during KEB	240 V		240 V
L3 Stall Prevention				
L3-01	StallP acceleration selection	1	Enabled	1
L3-02	StallP acceleration level	120 %		120 %
L3-03	StallP CHP level	50 %		50 %
L3-04	StallP deceleration selection	1	General Purpose - The deceleration will be stop when the Vdc reaches the stall prevention level. And decelerates again when the Vdc recovered.	1
L3-05	StallP running selection	1	Deceleration time 1	1
L3-06	StallP running level	120 %		120 %
L3-11	OV Inhibit selection	0	Disable	0
L3-17	Overvoltage Suppression and Deceleration Stall and	370 V		370 V
L3-20	Adjustment Gain	1.00		1.00
L3-21	Deceleration rate operation gain	1.00		1.00
L3-23	StallP running level output auto reduction	0	Sets the stall prevention level	0
L3-24	Inertia conversion motor acceleration time	0.142 sec		0.142
L3-25	Load inertia P	1.0		1.0
L4 Frequency				
L4-01	Speed agreement level	0.0 Hz		0.0 Hz
L4-02	Speed agreement width	2.0 Hz		2.0 Hz
L4-03	Speed agreement level +-	0.0 Hz		0.0 Hz
L4-04	Speed agreement width +-	2.0 Hz		2.0 Hz
L4-05	Reference loss selection	0	Stop	0
L4-06	Frequency reference at floss	80.0 %		80.0 %
L4-07	Conditions of Frequency detection	0	No detection during baseblock.	0
L5 Fault Reset				
L5-01	Number of auto restart attempts	0 time(s)		0 time(s)
L5-02	Auto restart operation selection	0	Not output(Fault contact is not	0
L5-04	Fault Reset Interval Time	10.0 sec		10.0 sec
L5-05	Auto restart selection	0	Count the number of restart which	0
L6 Overtorque				

L6-01	Torque detection selection 1	0	Overtorque/undertorque detection disabled	0
L6-02	Torque detection level 1	150 %		150 %
L6-03	Torque detection time 1	0.1 sec		0.1 sec
L6-04	Torque detection selection 2	0	Overtorque/undertorque detection	0
L6-05	Torque detection level 2	150 %		150 %
L6-06	Torque detection time 2	0.1 sec		0.1 sec
L6-08	Machine degradation detection selection	0	Machine degradation detection	0
L6-09	Machine degradation detection speed level	110.0 %		110.0 %

Parameter		Value	Information	Default Setting
L6-10	Machine degradation detection	0.1 sec		0.1 sec
L6-11	Machine degradation detection	0		0
L8 Hardware Protection				
L8-01	DB resistor protect	0	Disabled	0
L8-02	Overheat pre-alarm level	100 deg		100 deg
L8-03	Overheat pre-alarm selection	3	Operation continued (monitor display only)	3
L8-05	Phase loss input selection	1	Enabled	1
L8-07	Phase loss output selection	1	1PH loss detection	1
L8-09	Ground protection selection	0	Disabled	0
L8-10	Cooling fan control selection	0	Operates when inverter is running	0
L8-11	Cooling fan control delay time	60 sec		60 sec
L8-12	Ambient temperature	40 deg		40 deg
L8-15	OL characteristics at low speeds	1	L8-16,17 setting enable	1
L8-18	Software CLA selection	1	Software CLA enable	1
L8-19	Frequency Reduction Rate during	0.8		0.8
L8-38	Carrier frequency decel selection	1	Frequency of overload decreases gradually when the	1
L8-40	Reduction carrier frequency time	0.50 sec		0.50 sec
L8-41	Current warning	0	Disable	0
n1 Hunting Prevention				
n1-01	Hunting-prevention selection	1	Enabled	1
n1-02	Hunting-prevention gain	1.00		1.00
n1-03	Hunting-prevention time	10 ms		10 ms
n1-05	Reverse hunting prev gain	0.00		0.00
n3 High-Slip Braking				
n3-01	HSB deceleration frequency	5 %		5 %
n3-02	HSB current	150 %		150 %
n3-03	HSB DWELL time	1.0 sec		1.0 sec
n3-04	HSB OL time	40 sec		40 sec
n3-13	Over exciting deceleration gain	1.10		1.10
n3-21	Over slip inhibit current level	100 %		100 %
n3-23	Over Excitation Run	0	Disable	0
o1 Display Settings				
o1-01	Monitor selection	106	Output voltage	106
o1-02	Monitor selection after power up	1	Frequency reference	1
*o1-03	Display scaling... **[M]**	2	min-1 unit	0
o1-05	LCD Contrast	3		3
o2 Key Selections				
o2-01	LOCAL/REMOTE key	1	Enabled	1

Parameter		Value	Information	Default Setting
o2-02	Operation STOP key	1	Always enabled	1
o2-04	KVA selection	63	CIMR-V*2*0006***	63
o2-05	Frequency Reference Setting	0	Enter key needed	0
o2-06	Operator detection	0	Operation continues even if the dig. Operator is	0
o2-07	Bidirectional when the inverter	0	Forward run	0
o2-09	Initialize mode selection	1	American spec	1
o3 Copy Function				
o3-02	Copy Allowed Selection	0	Disabled - No digital operator copy functions are	0
q1 DWEZ Function Parameters				
q1-01	Custom Parameter 1	0.00 %		0.00 %
q1-02	Custom Parameter 2	0.00 %		0.00 %
q1-03	Custom Parameter 3	0.00 %		0.00 %
q1-04	Custom Parameter 4	0.00 %		0.00 %
q1-05	Custom Parameter 5	0.00 %		0.00 %
q1-06	Custom Parameter 6	0.00 %		0.00 %
q1-07	Custom Parameter 7	0.0 %		0.0 %
q1-08	Custom Parameter 8	0.0 %		0.0 %
q1-09	Custom Parameter 9	0.0 %		0.0 %
q1-10	Custom Parameter 10	00		00
q2 DWEZ Drive I/O				
q2-11	Drive U1 Monitor Select 1	0		0
Drive U1 Monitor Select 2		0		0
q3 DWEZ Logic Functions				
q3-01	Timer #1 Delay	0.0 sec		0.0 sec
q3-02	Timer #2 Delay	0.0 sec		0.0 sec
q3-03	Timer #3 Delay	0.0 sec		0.0 sec
q3-04	1 Shot On Time	0.0 sec		0.0 sec
q3-05	Interval Timer Off Time	0.0 sec		0.0 sec
q3-06	Interval Timer On Time	0.0 sec		0.0 sec
q4 DWEZ Ramp Time				
q4-01	Ramp Time	0.0 sec		0.0 sec
q4-02	Delay Filter Time	0.5 sec		0.5 sec
q4-03	Scale #1 Multiplier	1		1
q4-04	Scale #1 Divisor	1		1
q4-05	Scale #1 Bias	0.00		0.00
q4-06	Upper Limit 1	0.00 %		0.00 %
q4-07	Lower Limit 1	0.00 %		0.00 %
q4-08	Upper Limit 2	0.00 %		0.00 %
q4-09	Lower Limit 2	0.00 %		0.00 %
q4-10	Dead Zone	0.00 %		0.00 %
q4-11	Limit/Dead Zone Properties	00		00
q4-12	Scale #2 Multiplier	1		1
q4-13	Scale #2 Divisor	1		1
q4-14	Scale #2 Bias	0.00		0.00
q5 DWEZ Compare - MOP Functions				
q5-01	MOP Time	0.0 %		0.0 %
q5-02	MOP Min Value	0.0 %		0.0 %
q5-03	MOP Max Value	0.0 %		0.0 %
q5-04	MOP Reset Value	0.0 %		0.0 %

q5-05	Step MOP Amount	0.0 %	0.0 %
q5-06	Step MOP Min Value	0.0 %	0.0 %
q5-07	Step MOP Max Value	0.0 %	0.0 %
q5-08	Step MOP Reset Value	0.0 %	0.0 %
q5-09	Compare <= Hysteresis Level	0.00 %	0.00 %
q5-10	Compare = Bandwidth	0.00 %	0.00 %
q5-11	Compare >= Hysteresis Level	0.00 %	0.00 %
q6 DWEZ Application Functions			
q6-01	PI Properties	0	0
q6-02	PI Log Input Sel	10	10
q6-03	PI Prop Gain	1.00	1.00
q6-04	PI Int Time	1.0 sec	1.0 sec
q6-05	PI Int Limit	100.0 %	100.0 %
q6-06	PI Limit	100.0 %	100.0 %
q6-07	PI Output Gain	1.00	1.00

Blank

Appendix

W

[W] General Anomalies and Tips

This section is meant to discuss information generic to EtherCAT and aid in the installation.

Network Switches

Network Switches – Network switches should not be used with EtherCAT. Network switches may cause single packet losses. If a switch is required, make sure it is specifically recommended for EtherCAT operation, such as the Omron EtherCAT Junction Slave. Loss of even a single packet may cause a drive error in some drives. This has been noted especially with Kollmorgen devices.

MSBs

MSBs – Do not use foreground MSBs. They typically are not needed and will slow down the control loop. Background MSBs operate outside the control loop and are more efficient. They are supported for legacy M3-40 applications but are not needed with the M3-41 unless limited to a very small MSB.

ppr/mppr

'ppr/mppr' (pulses/rev, and master pulses/rev) must be set properly for the type of encoder used, or improper motion will occur. Set these parameters using QuickBuilder and the properties section of the drive.

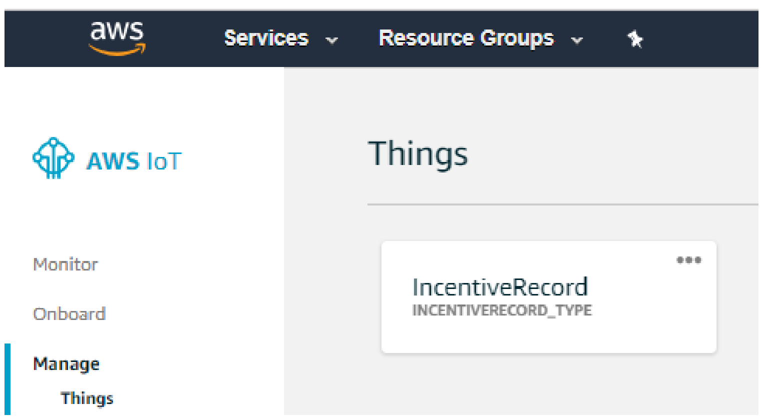
Blank

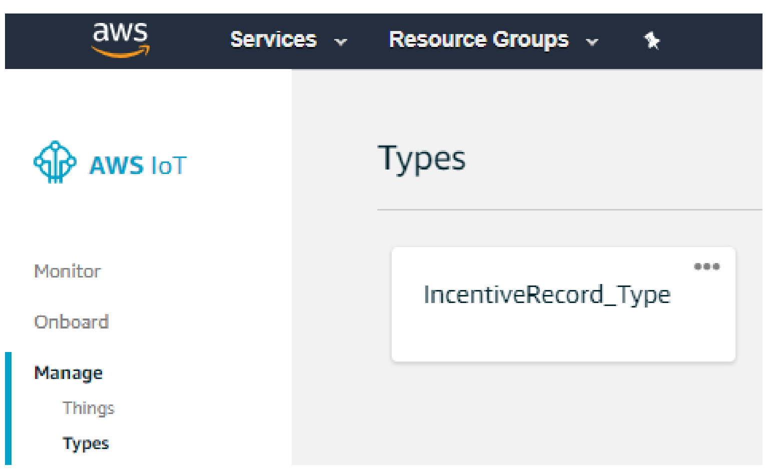
Appendix

X

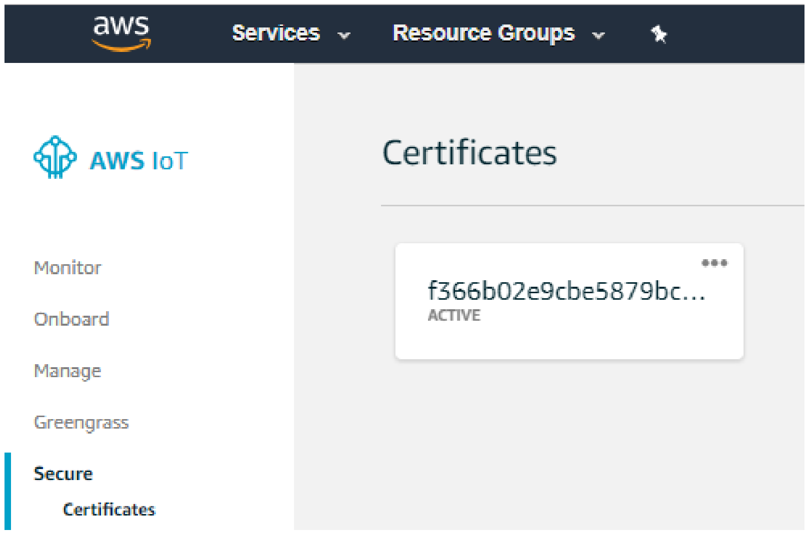
[X] Amazon AWS IoT

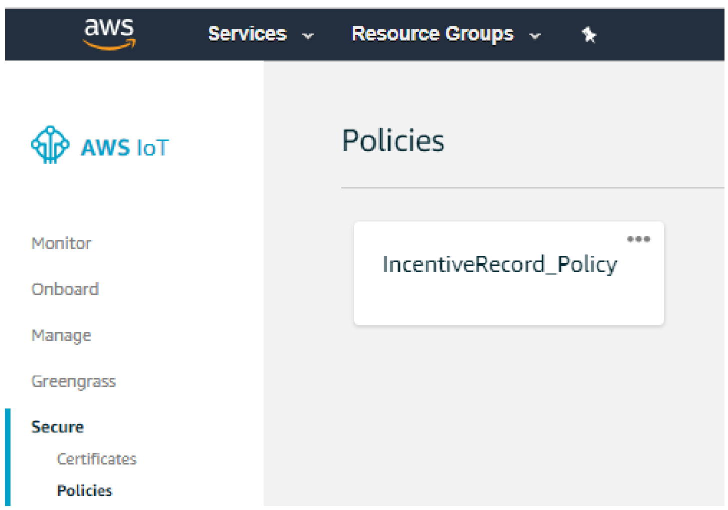
This section contains screen shots from the AWS IoT setup used for the C# example included with Incentive. Should you be creating this for the first time Amazon has a lot of good documentation available onine. One example can be found here: https://docs.aws.amazon.com/iot/latest/developerguide/iot-sdk-setup.html.

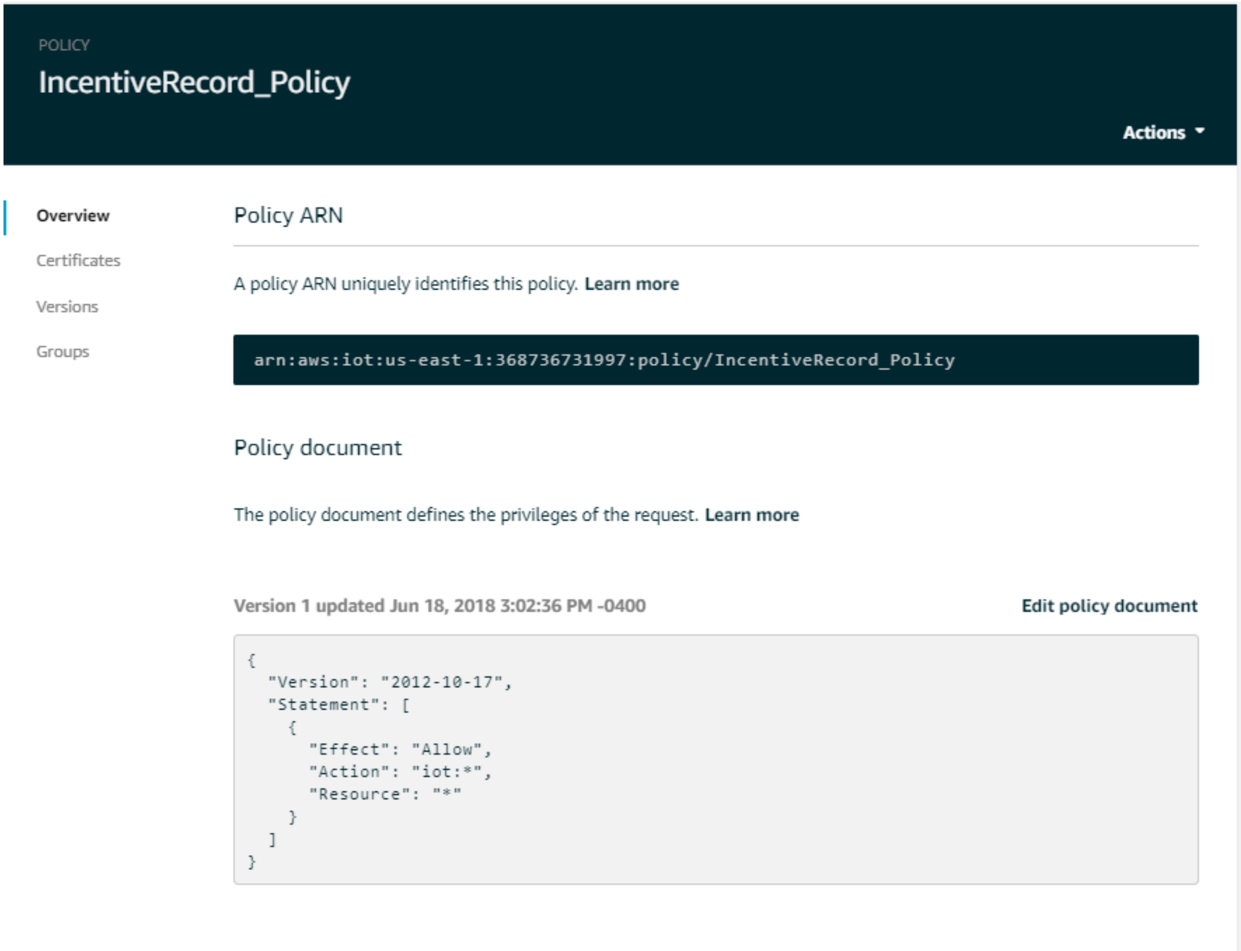


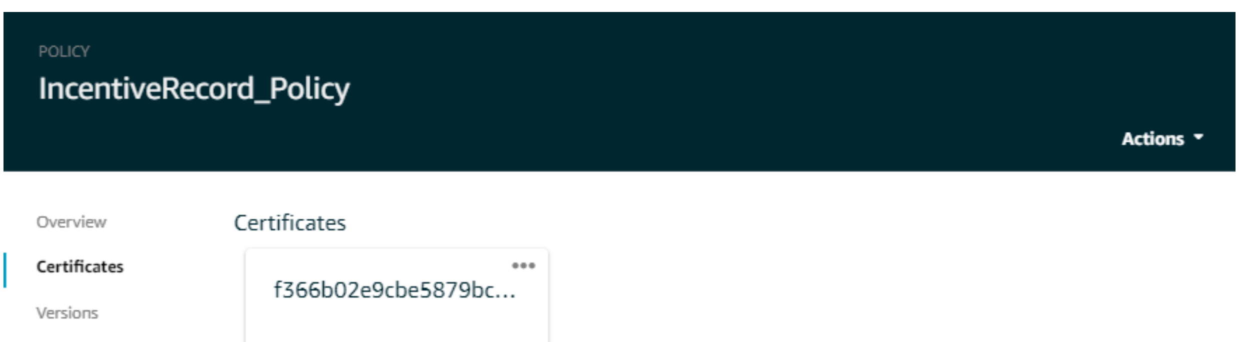


Security is done with X509 certificates which must be created and activated as well as attached.

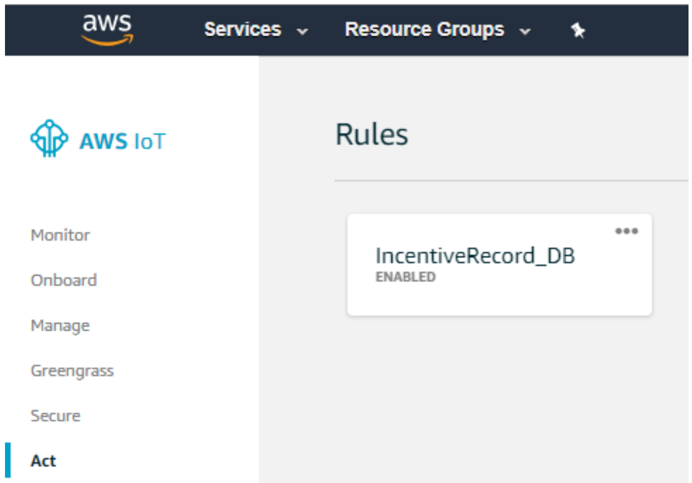


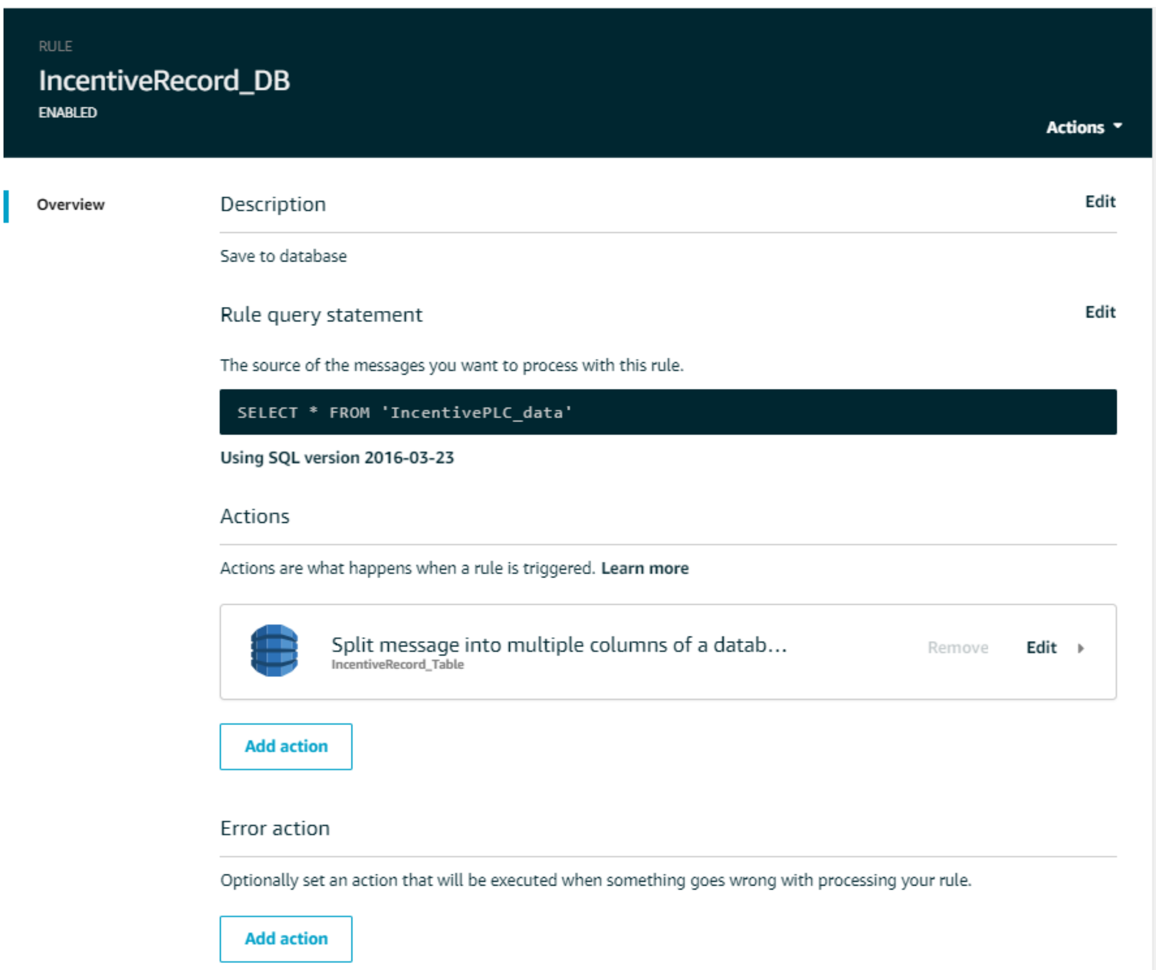


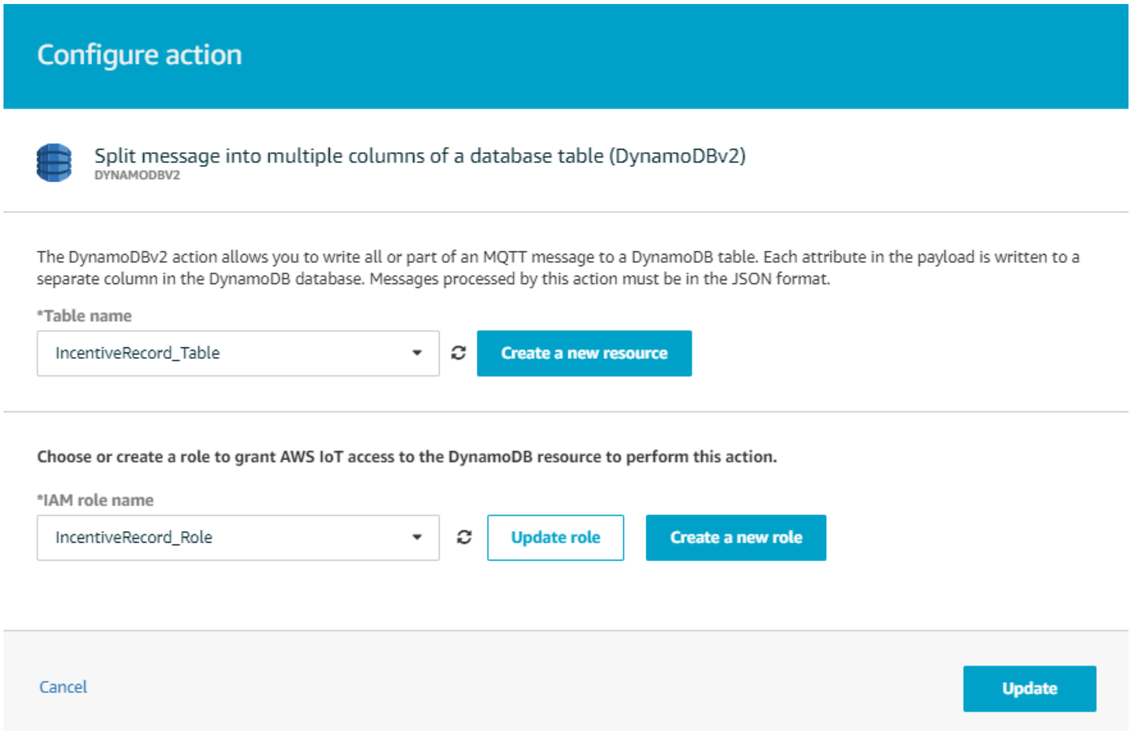

POLICY
IncentiveRecord_Policy
Actions
Overview
Certificates
Versions
Groups
Policy ARN
A policy ARN uniquely identifies this policy. Learn more
arn:aws:iot:us-east-1:368736731997:policy/IncentiveRecord_Policy
Policy document
The policy document defines the privileges of the request. Learn more
Version 1 updated Jun 18, 2018 3:02:36 PM -0400
Edit policy document
{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Action": "iot:*",
"Resource": "*"
}
]
}


POLICY
IncentiveRecord_Policy
Actions
Overview
Certificates
Versions
Groups
Certificates
f366b02e9cbe5879bc...

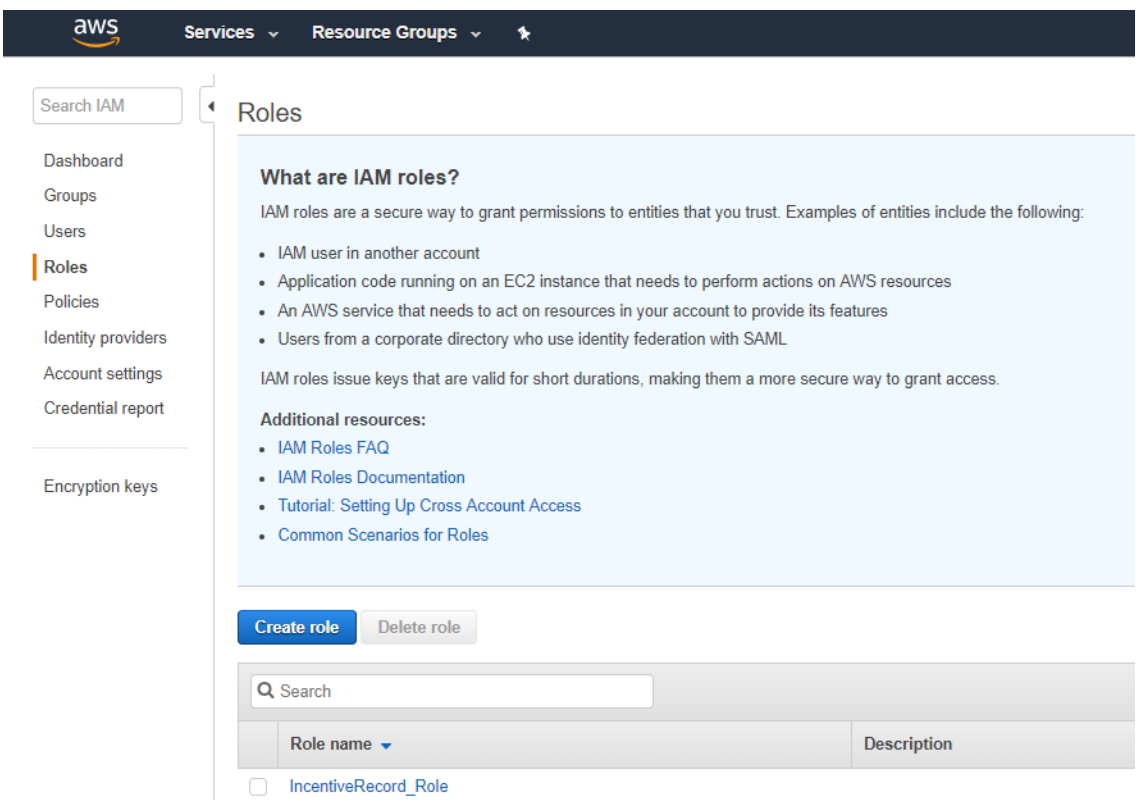
Optional rules on what to do with the IoT data once it reaches the cloud. In this example it is sent to a DynamoDBv2 database for storage. Subscribers can also receive the data in parallel.

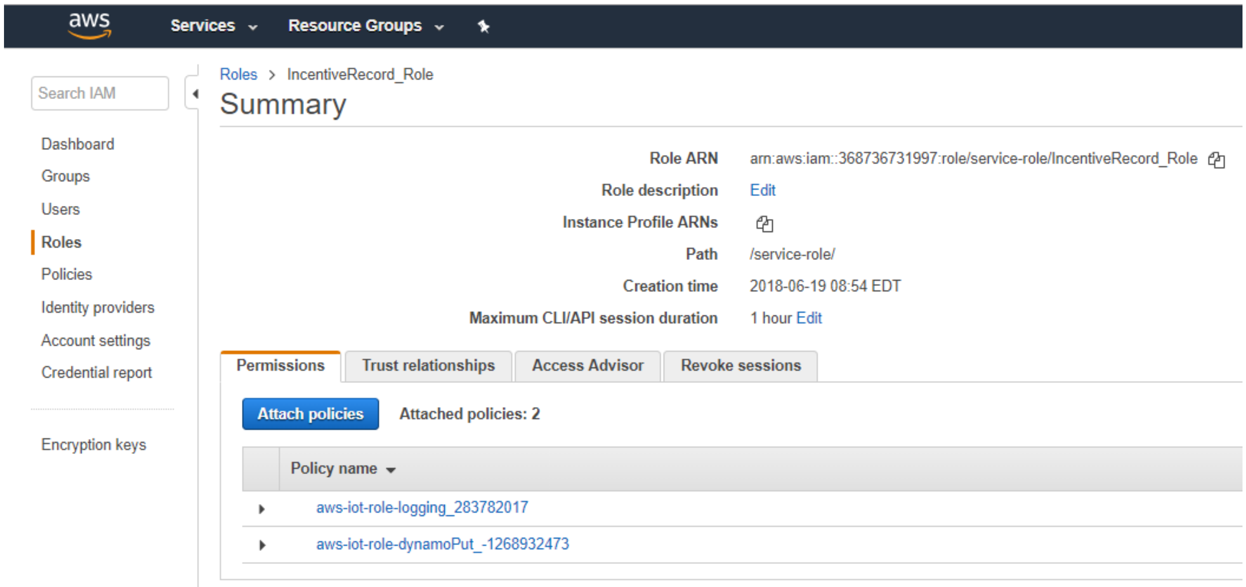


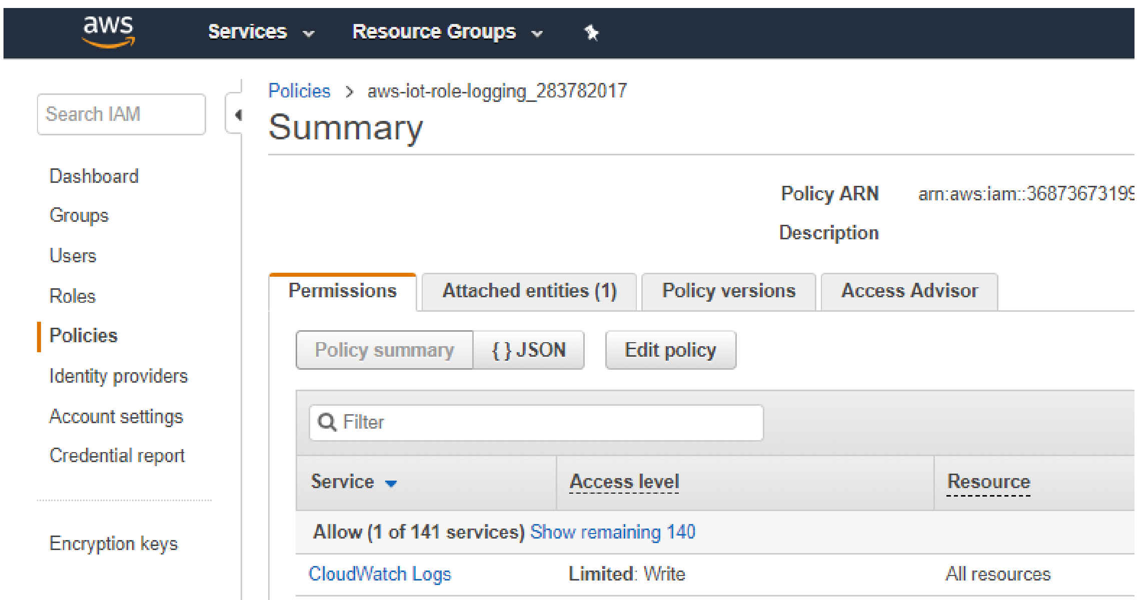


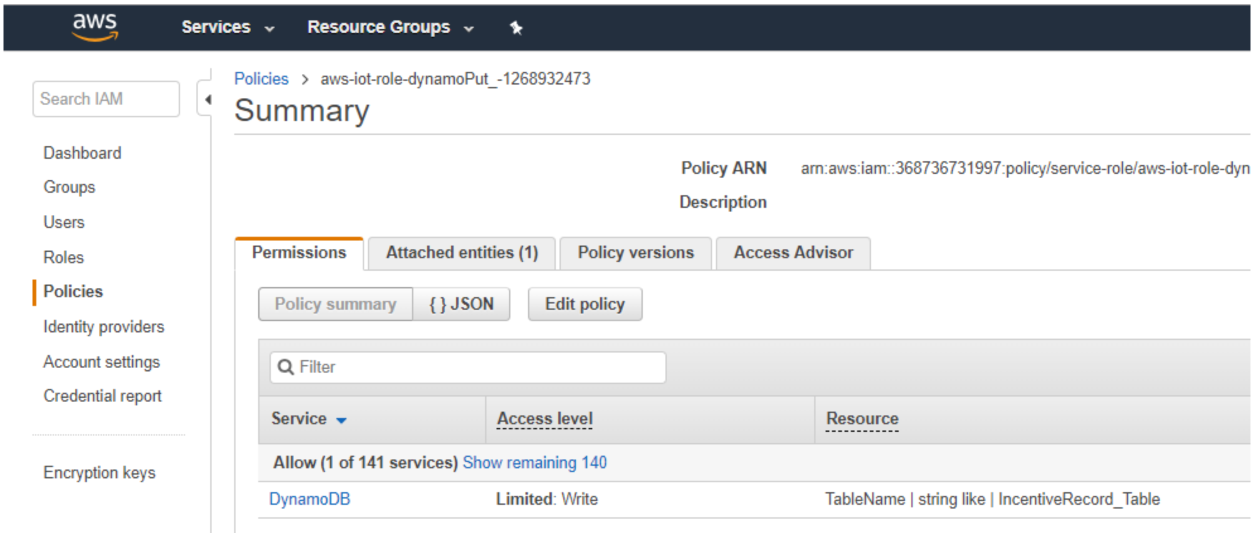


Roles and Policies:

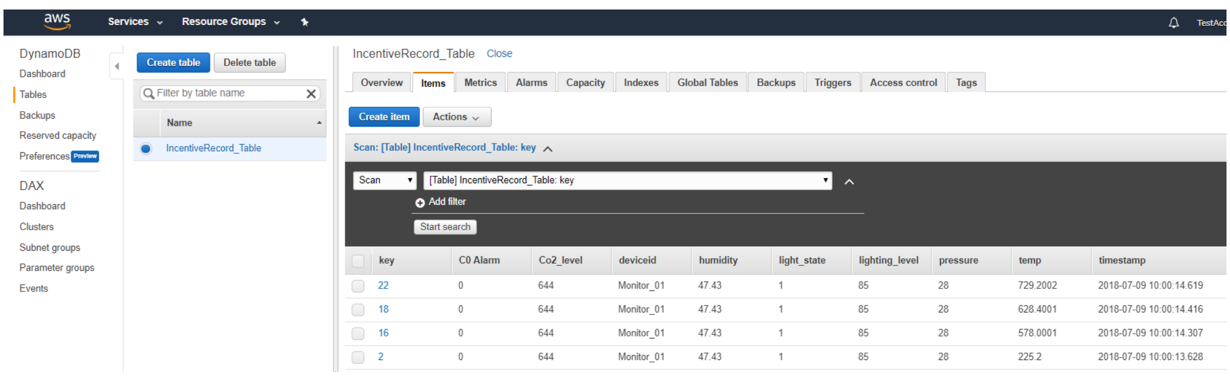



aws
Services
Resource Groups
Search IAM
Dashboard
Groups
Users
Roles
Policies
Identity providers
Account settings
Credential report
Encryption keys
Roles > IncentiveRecord_Role
Summary
Role ARN
arn:aws:iam::368736731997:role/service-role/IncentiveRecord_Role
Role description
Edit
Instance Profile ARNs
Path
/service-role/
Creation time
2018-06-19 08:54 EDT
Maximum CLI/API session duration
1 hour Edit
Permissions
Trust relationships
Access Advisor
Revoke sessions
Attach policies
Attached policies: 2
Policy name
aws-iot-role-logging_283782017
aws-iot-role-dynamoPut_-1268932473

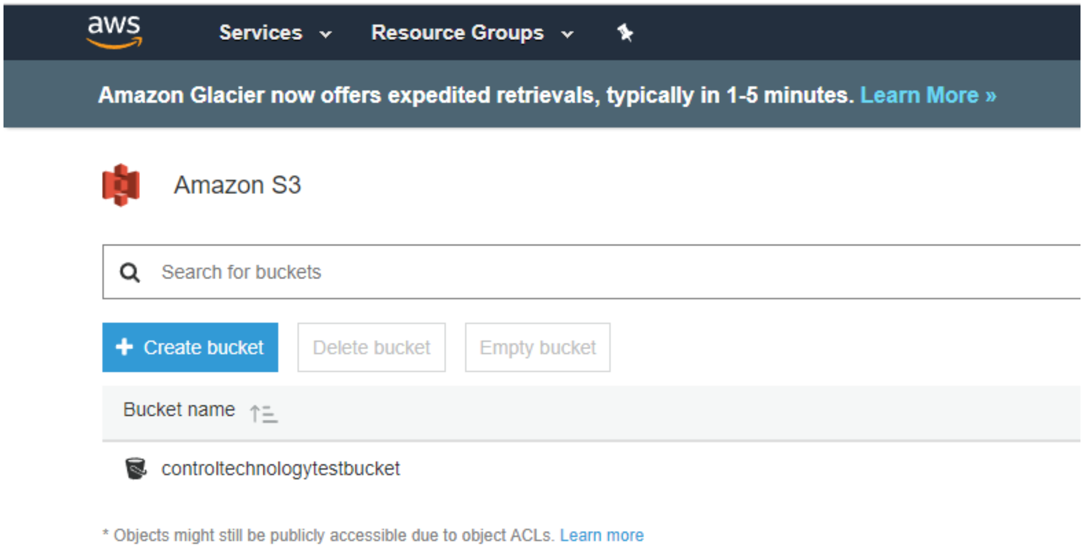

aws
Services
Resource Groups
Search IAM
Dashboard
Groups
Users
Roles
Policies
Identity providers
Account settings
Credential report
Encryption keys
Policies > aws-iot-role-logging_283782017
Summary
Policy ARN
arn:aws:iam::36873673199
Description
Permissions
Attached entities (1)
Policy versions
Access Advisor
Policy summary
{ } JSON
Edit policy
Filter
Service
Access level
Resource
Allow (1 of 141 services) Show remaining 140
CloudWatch Logs
Limited: Write
All resources

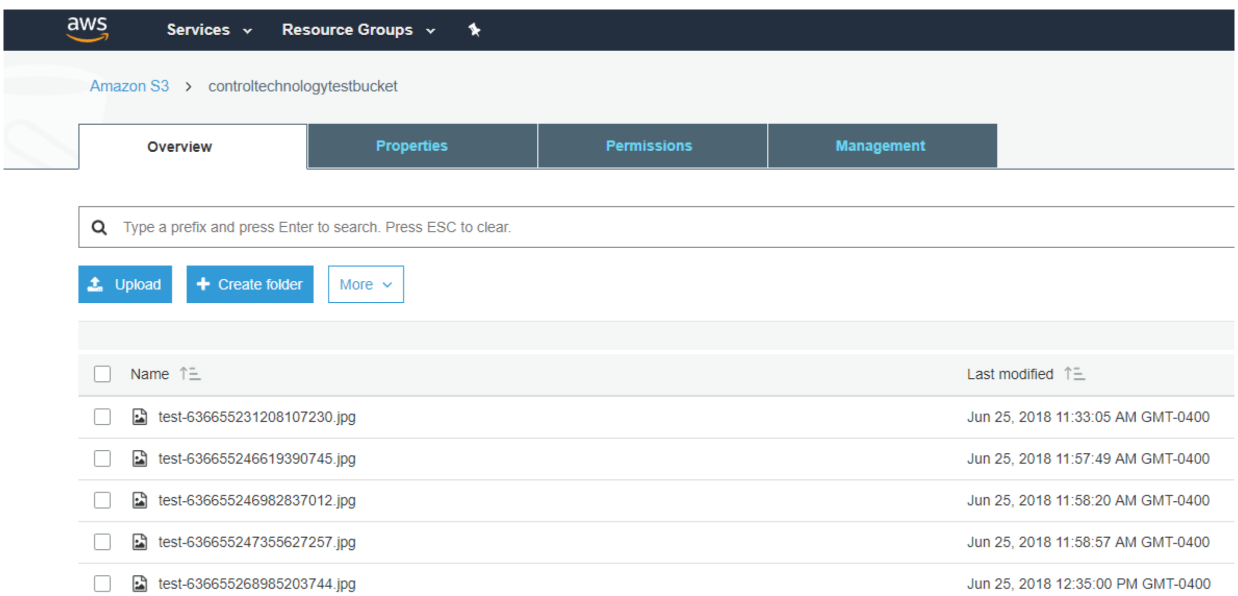


DynamoDB IncentiveRecord_Table sample records (fields other than key are dynamically created when using DynamoDBv2):



Amazon S3 bucket used for file upload and download examples:





Appendix

Y

[Y] Software and Firmware Revision History

IncentiveECAT & M3-41A Revisions

V1.113 (Incentive PC & M3-41A Synced)

1. **Mitsibushi FR-E800E Series Support:** Support for Mitsibushi VFD drive in velocity mode. E820 tested.
2. **Advanced Motion Control:** Updates for AMC newer drives with sinusoidal encoder. Scaling needed adjustment. EEPROM Esi xml file creation.

V1.111 (Incentive PC & M3-41A Synced)

1. **Mitsibushi J5 Support:** Support for CSP and Homing added running DC Sync.
2. **5300 EtherCAT Slave:** Support for 5300 Slave updated for conformance testing as well as new object availability.
3. **Applied Motion STF:** Support added.
4. **Remote Server Licensing:** Added support for remote server user licensing. Prior licensing using QuickBuilder is still supported.
5. **Advanced Motion Control:** Updates for AMC newer drives. Removed check Communications user event enable at initialization, object 0x2065 index 0x21.
6. **CSP Mode:** Added check to make sure the drive did not go offline just prior to sending CSP mode request to the drive.
7. **Initialization Mode:** Minor change to broadcast 0's to DCtime at init and all of System time. Clear enable flag if Sync Manager length is 0. Modified PDO slave routines to return slave number that fails.
8. **Omron Controller:** Modified data alignment to account for 8 bits digital inputs and outputs to occupy 16 bits of space.
9. **Yaskawa VFD Drive:** Added support for product ID 0x53455333.
10. **QuickBuilder Digital Output:** Resolved possible race condition when QuickBuilder does write while updating PDO operational data.
11. **Offline:** Improve offline monitoring of slave devices.

12. **Program Logic:** Improved logic such that if attempt a move that is delayed a bit and then the drive is disabled, program will fault.
13. **On the fly Motion Changes:** Improved performance when making changes to motion control without stopping.

V1.100 (Incentive PC & M3-41A Synced)

1. **New Position:** SCorrected problem where if new motion command was issued while another was still active user units were 1:1 instead of what the user set them to.
2. **Delta Electronics**: Correct problem when disable drive under program control and then enable again Delta firmware had a bug that required working around to maintain CSP mode sync.
3. **Homing**: Corrected a problem where there was a 1 mS window where if the drive disable was done immediately after a homing operation a deadlock could occur waiting for the inpos flag to be set even though the homing was complete. A slight delay would also fix the problem.

V1.99 (Incentive PC & M3-41A Synced)

1. **Delta Electronics: ADSA**-A2 Support added, torque control supported.
2. **Maxon EPOS4**: Support added, torque control not supported. 2mS max cycle time.
3. **Interpolation Bug**: Fixed bug when moving from arc to linear interpolation units were left in radians.
4. **Triangular Motion Bug**: Fixed bug that when was doing a trapezoidal move whose velocity could not be met there would be a pause until the next move of whatever the deceleration time was.
5. **CAM Table Variant Access**: Added for 5300 enhancements to allow variant access to axis tables as well as being able to set the type of precompute to be done on the table to Linear, Cubic, or Quadratic to speed up precompute time when all not needed.
6. **Virtual Drive Feature for Faulted Drive**: Added feature for when two servos are controlling the same device and one is meant to be a fall back in case of failure for redundancy. The failed drive can automatically come up as a virtual drive and drive type checked so the redundant drive can be used.
7. **CTC EtherCAT Slave Support**: Added code for future 5300 EtherCAT slave board.
8. **Kollmorgan**: Newer firmware on Kollmorgan drives now support setting 0x6065 to 0 to disable position tracking window. This is set to 0 now by default where previously it was set to a large number.
9. **Copley**: Added product code for Argus-GEM drive and assumed to be initialized like the Xenus.
10. **QS2_Status**: Fixed status variable to reflect proper state during initialization.
11. **rmstrq**: This variable now is an absolute value instead of reflecting negative torque.

V1.95 (Incentive PC & M3-41A Synced)

1. **Added Support for Parker Hannifin Drive**: Parker Hannifin PD-xxC series drive supported.

V1.94 (Incentive PC & M3-41A Synced)

1. **SMC I/O Reporting**: Changed all versions to report proper DIN, DOUT, AIN, and AOUT in EtherCAT Explorer for SMC I/O.

V1.93 (Incentive PC & M3-41A Synced)

1. **Yaskawa Multiturn/Software Reset support**: If bit 7 is set in encoder_mode then when a drive enable is done the drive enable will not happen and a bit is set such that when the network is reset a Multi-turn and Software Reset will be done during PREOP stage of initialization. This is needed sometimes when trying to use an absolute encoder as well as their resident homing functions.
2. **New Licensing Support (PC Only)**: The Incentive EtherCAT Master run-time will now first check for a legacy license and if found reference that, if not present it will attempt to contact the Windows Incentive License Montior Service for authorization, if not found the product is placed in demo mode and will automatically timeout and shutdown in about 3 hours.

V1.92 (Incentive PC & M3-41A Synced)

1. **Dualport Axis Property Refresh Rate**: On Incentive PC changed dualport update rate for some axis property variables from 10 mS to 5 mS when greater than 7 axis (each access delay time for update).
2. **Virtual Axis Filler:** Added option to automatically define number of axis desired and if the axis is not present to automatically startup a virtual axis. One example use is when a backup servo is installed and fixed station addressing is used. Requires updated QuickBuilder
3. **Diagnostic Messages:** Added diagnostic messages for restarting network to identify source of request.

V1.91 (Incentive PC & M3-41A Synced)

1. **Yaskawa V1000 VFD** – Added support for Yaskawa V1000 VFD drive..
2. **EtherCAT Device Count -** Now monitor for more devices on network than what began with and if changes fault. Protects against a device plugged into the network while it is running.

V1.90 (Incentive PC Only)

1. **Diagnostic Resource Naming** – Added additional global names on some threads and mutexes for debugging purposes with INtime Explorer.
2. **INtime Version** - Added console log message as to which INtime version is running at startup.
3. **INtime 6.3** - Rebuilt IncentiveECAT with 6.3 TenAsys INtime libraries which can resolve some upgrade issues. Included alternate .rta files during installation built for 6.3.18220.1 and the older version, 6.1.17004.
4. **Thread Pool** – Corrected mutex issue with free thread pool. No problem has ever been observed.

V1.89 (Incentive PC & M3-41A Synced)

1. **Omron Temperature Controllers** – Support added for Omron Temperature Control modules TC-2406 and TC-3406. Other Temperature controller product codes added but not tested due to lack of availability.
2. **WAGO 750-630/000-008** – Support added for Wago SSI transmitter module.
3. **Festo Drive** – Support added for EMCA-EC Festo drives, CSP mode only with free run.
4. **Additional ABB Support** – Added support for higher voltage models, MicroFlex e180/e190.
5. **M31-41A SOM Support** – Drivers added for updated OMAPL138 SOM serial 16M flash chip, 1603 series SOM to replace older 1602 series. Change is backward compatible and applies to the 5300 product line only.
6. Added PCAP_BACKTRACE option to command line to always collect and log the last 512 packets of EtherCAT sent/received in Wireshark capture format.
7. Wireshark compatible packet captures now increments from 1 to 16 for file number before overwriting the first file.

V1.88 (Incentive PC Only)

1. **Diagnostic Packet Collection** – Startup parameters of PCAPSTARTUPONLY and PCAP####, where #### is the number of seconds to collect packets after operational and including startup. A value of PCAP0 will collect forever but be careful since at a 1mS scan rate this is typically 5.55 megabytes/10 seconds of collection time with a base of about 3.5 megabytes for startup initialilzation. The capture file is called Incentive_ecat_trace.pcapng and is compatible with Wireshark for viewing.

V1.87 (Incentive PC & 5300 M3-41A Synced)

1. **Diagnostic messages** - A number of the drives were listing the axis that had an error as one less than what it really was.
2. **PLC FOE Requests** - Added message ability for 5300PLC process to request FOE transfers.
3. **Fixed Diagnostic Logging Message** - Found logging message on EtherCAT network interface open that printed out the link speed was using a stack variable that would

disappear sometimes causing junk to print on the screen or be logged to a logging file. Now allocate memory.

4. **Added FOE Protocol** - Added FOE protocol for file upload/download to slave.
5. **Mitsubishi Homing** - Fixed homing with Mitsubishi, broke it when added support for Applied Motion.
6. **Fixed Tracking Creep** - Fixed tracking when ppr and mppr are not evenly divisible or large difference with small ppr on slave, had remainder that would cause drive to creep. Now if master does not move will hold position.
7. **FOE API** - Added API functions for FOE protocol.
8. **Stop EtherCAT** – Added Incentive API stopEtherCAT() function to AxisSupervisor class.

Note FOE is only supported by the Incentive PC, not the 5300 due to limited file system space.

V1.85 (Incentive PC & 5300 M3-41A Synced)

1. **Copley Drive Reset –** Added ability to reset Copley drive when there is a possibility of a latched encoder error. Warning is logged for the user as well.
2. **Applied Motion drive enable -** Fixed Applied Motion stepper drive enable so don't set CSP mode until power fully applied and operational, then just set once in PDO. If don't do this the drive vibrates for a moment before first move from something to do with holding position. Only drive that has this issue so is probably a bug in their EtherCAT implementation.

V1.84 (Incentive PC & 5300 M3-41A Synced)

1. **Support file expansion of device Product Codes –** Supported product codes can be field expandable by adding a file with the new product codes and mapping them to an existing device driver.
2. **Support Applied Motion Products SS-**EC – Added support for stepper drive.
3. **Added Arc Stop Command –** Added command such that if you do a ‘stop’ during an arc or vector move it will stop both X, Y, and Z axis and clear out the axisY and axisZ variables upon execution. This command should also be used when exiting from drawing arcs to normal linear moves.
4. **Support for Additional Copley Drives –** New XE2, XEL, SE2, 800-1849, Stepnet.
5. **New Variables** - Added variables for manufacturer and product EtherCAT codes: eCAT_manufID and eCAT_productCode.
6. **New Variable and units ratio feature** - Added variables for units_ratio (r/w) and tr (translation ratio being used, read only). If uun is set to 0 then units_ratio will be used which is the double version of uun/uud. Had an issue where uun was an integer and a 64 bit value was needed by user. tr can be viewed in a QB diagnostics window and is the result of uun/uun or units_ratio so you can know what the MSB's are using for a value.
7. **Automatically Add EtherCAT Devices Option** – Option to not automatically add EtherCAT devices at the start of a QB program and allow the program to do it.
8. **5300 Code Sync –** Code synced to run on 5300 M3-41A.

9. **SDO read** – Found potential lockup if power on a drive was shutdown while reading diagnostic information to report an offline error. Would not exit the loop.

V1.82 (Incentive PC only)

1. **Redundant Master (Ring/Loop) -** Support added to be able to recover from a broken cable within the slave loop. Two Ethernet ports are needed for a single EtherCAT Master. One connects the primary Master port to the first IN of the slave device. The second Ethernet port is connected to the OUT of the last slave device.
2. **WAGO Incremental Encoder** – Support added for Wago 750-631 (16 bit) and 750-637 (32 bit) incremental encoders. Each encoder is added as a virtual axis with the encoder being that axis feedback encoder (fpos).
3.

V1.81 (Incentive PC & 5300 M3-41A Synced)

1. **Secondary Master -** Support a secondary backup master on the same master network for fail safe operation. Modified both Primary and Secondary Masters to monitor for each other's communications and take control when the other fails or stops communicating. The network will be reset and the backup master, running on a different embedded PC, will begin a new application program. It is up to the user to sync the applications..
2. **Memory Diagnostics -** Added ability to add DIAG2 to the parameters when the .rta file is loaded to display the amount of total memory allocated each time a message is logged to the IO console. Note that extensive use of memory logging is not recommended since it can cause EtherCAT timing problems given the overhead to process and fact that all threads are locked from executing that requires memory during calculation.

V1.80 (Incentive PC & 5300 M3-41A Synced)

1. **Tracking** – Fixed a problem where if you were tracking the master for a long period the slave would suddenly stop if using large ratios and large ppr values. When value exceeded the size of an integer, 32 bits, the slave would stop tracking. Now 64 bits is used.
2. **DOUT –** The number of digital outputs was not being reported properly to Quickbuilder EtherCAT Explorer, it has been corrected.

V1.79 (Incentive PC & 5300 M3-41A Synced)

1. **Thread Priority** – Decrease thread priority by 1 for highest priory task disabling so not running at the same time as an INtime internal thread. Highest priority for EtherCAT Master was 129 and is still 129, for PLC Logic it is now 131 instead of 130. Should have no effect.
2. **Packet Timing Warning** – Modified error recovery and message when an EtherCAT packet exceeds 1.2 mS when timing is set for 1 mS. Now faster processing of diagnostic message so can continue running and recover better. Previously a simple printf was used to the IO Console which took considerable time and hung the transmission thread, making

things worse. Now a message is sent to a background thread to process the error. Timing has been tightened up to 1.1 mS for reporting a warning.

3. **Wago –** Added support for Analog Input module; 750-481, 750-483, 750-485, 750-486, 750-487, 750-491, 750-496, 750-497
4. **DIAG1** – As with the Incentive PLC R70.08 release, if DIAG1 is specified on the EtherCAT_Master.rta parameter line additional diagnostic messages will appear on the display and file logs for each process.
5. **Homing Stop –** Added ability to abort a Homing command and stop. Previously there was no good way to abort a drive internal Homing command once started.
6. **Diagnostics –** Added more descriptive diagnostic messages for Homing errors when attempt to initialize drive objects and it fails.
7. **Synchronized Code –** Unified source code for Incentive PC EtherCAT Master and M3-41A ARM based 5300.
8. **Tracking –** Fixed a problem where if you were tracking the master for a long period the slave would suddenly stop if using large ratios and large ppr values. When value exceeded the size of an integer, 32 bits, the slave would stop tracking. Now 64 bits is used.

IncentivePLC and 5300 Revisions

R70.33 Incentive PC & 5300 Sync'ed

1. **EtherCAT Slave**: 5300 EtherCAT Slave passes EtherCAT Technology Group Conformance Test Tool V2.2.1.0, Test Library 1.1.0.0.

R70.32 Incentive PC & 5300 Sync'ed

1. **EtherCAT Slave**: EtherCAT slave capability added along with supporting registers 13056 to 13065.
2. **Expansion Rack** – Expansion rack timing adjustment.
3. **M3-41A Timing** – Increase power up scan delay from 100ms to 200 ms for ID settling time between slots and extend dualport quiet time by 1 second. Resolves 41A identifier issues at boot.
4. **M3-41A** – Added support for QuickBuilder to send a reset network request.
5. **Licensing** – Added Remote Server based licensing support.
6. **QuickBuilder Loading Issue** – Resolved problem where a new QuickBuilder program would attempt to disable drives that may have been offline upon reloading. Was previously done prior to loading now done afterwards.

R70.26 Incentive PC & 5300 Sync'ed

1. **MD5**: Added diagnostic MD5 file check that can be used via telnet when debugging is enabled: MD5 'filename'. Helps to verify it matches that on another computer. This should not be run while running a system as it can take a while to calculate the MD5 checksum.

R70.25 Incentive PC & 5300 Sync'ed

1. **EtherCAT Camming Table Variants**: Support added to allow interface to all camming tables used by EtherCAT through variants 36830 – 36837. Allows table display through Quickbuilder watch windows as well.
2. **Modbus TCP 32 bit Swap Register**: 20099 added to allow upper and lower 16 bits of 32 bit register contents to be swapped with Modbus TCP connection when we are the server. Default is 0, set to 1 to activate on all Modbus TCP Server connections.
3. **Future EtherCAT Slave**: Preliminary EtherCAT Slave code synced with Incentive PC, not released.

R70.22 Incentive PC & 5300 Sync'ed

1. **EtherCAT Driver**: Fixed a problem on the 5300 when using multiple EtherCAT modules as well as had AIN modules in the rack the analog input start number for the EtherCAT module was being overwritten with a 0. Incentive PC was not effected but still had the parameter being overwritten.
2. **Modbus Client (Master)** - Fixed Incentive PLC Logic so that Modbus Client (Master) works properly on serial ports.
3. **Sync Source** - Merge PC and 5300 software.

R70.21 Incentive PC & 5300

1. **QuickBuilder Version -** Added register 13046/13047 to show QuickBuilder versions that is running (Major/Minor).
2. **Sync Source -** Synced Incentive PC and 5300 PLC Logic source code.

R70.20 Incentive PC Only

1. **New Licensing Support**: The Incentive PLC Logic run-time will now first check for a legacy license and if found reference that, if not present it will attempt to contact the Windows Incentive License Montior Service for authorization, if not found the product is placed in demo mode and will automatically timeout and shutdown in about 3 hours.
2. **Expanded Register 13464** – For both 5300 and Incentive writing 0x5500 clears faults and turns VBIAS on, 0x5501 sets fault and disabled VBIAS.
3. **CTPLC_1 IO Console Window**: Changed the software to wait for INtime to be fully operational prior to writing information to the diagnostic console. On a low end, slow system the console could fail to appear if Incentive was started quickly after login and services were still starting.

R70.19 Incentive PC Only

1. **TCP Serial Communications** – Added change when TCP serial socket is deleted during transmission to clear the TX buffer counter so looks like message is sent and port not stuck on transmit.

2. **Diagnostics -** Added diagnostic messages for TCP sockets, low level. Also recommended increasing loader.cfg file kern.workers_max parameter to 64 from default of 32.
3. **Added support for Common CRC –** When QuickBuilder sets the CRC type to ‘common’ and includes the project source file (.qbp) the CRC is generated on the project file, not the compiled program. This allows the debugger to match the program without having to download a new image each time since the prior default was ‘unique’ and the Microsoft compiler always generates a unique image.

R70.18 Incentive PC Only

1. **TCP Serial Communications** – Updates to TCPSerial TCP protocol to speed up connect and recv functions timeout so don't timeout while waiting for a user request to shutdown the connection (Applied Motion devices in particular).
2. **Cached Thread Resume -** Updates for resuming a cached thread only after it is in the self suspended state and if not wait for it. Problem was never observed but better to be safe.
3. **Thread Suspended Detection** - Added feature for QB programs to detect a thread is in self suspended state before using it. Used primarily for function completion detection.

R70.16 Incentive PC Only

1. **INtime 6.3** – Rebuilt with INtime 6.3 libraries. Now shipping 6.1 and 6.3 .rta IncentivePLC and IncentiveECAT runtimes. Slowly moving away from 6.1 and towards latest 6.3 environments.
2. **INtime Version** - Added console log message as to which INtime version is running at startup.
3. **Telnet** - Added INtime version to telnet 'get versions' command so QuickBuilder can detect the INtime runtime version that is installed and warn users to download the correct QB application program.
4. **File Sync** – Added code to FTP to flush buffers to disk after a file download. There was an intermitant problem where the CRC on a newly downloaded QB program would fail but if retried again was fine. Also attempt to load a second time on failure before report.
5. **DIAG3** – Added DIAG3 to startup arguments of 5300PLC.rta node for TCPSerial diagnostics to monitor thread TX/RX starting and stopping for connection monitoring.
6. **Thread Pool** – Corrected mutex issue with free thread pool.

R70.14 Incentive PC Only

1. **TCP Serial Protocol** – Fixed a bug in client mode where the RX thread was being invoked twice when a connection was closed and then re-opened. Cleaned up shutting down of connections. Problem probably does not exist in 5300 due to differences in the way the operating system processes the waking of a thread from a suspended state.
2. **MODBUS Client** – Resolved a problem where if a user writes to a remapped register at high speed without checking status that the write is complete other threads can be locked out of execution. This is not a valid operation but regardless now we ignore the write, sleep 50mS and log an error message telling the programmer to check for completion status before writing.

R70.13 Incentive PC & 5300

1. **TCP Serial Protocol** - Added 3 new protocols and types to the TCP serial redirection. It now supports UDP as well as a feature to strip a null from the front of a received message as well as add it to the front of a transmitted one. A NULL is used as the header for Applied Motion ASCII protocol and since not able to handle it in a string is now done automatically when register 22XX1 client/server mapping is set properly.
   0 – Client TCP
   1 – Server TCP
   2 – Client Applied Motion TCP (precede transmissions with 0x00, strip leading 0x00 from received messages)
   3 – Client Applied Motion UDP (precede transmissions with 0x00, strip leading 0x00 from received messages)
   4 – Client UDP
2. **TCP Stack KEEPALIVE** (PC ONLY) - It was noticed that the TenAsys INtime stack does not send KEEPALIVE very often on idle tcp connections so it was shortened to about 6 seconds with retries up to 8 times every 200 milliseconds afterwhich the connection will be dropped if no reply. Needed since a PC bridged network can not detect the network cable being unplugged, hence loss of link.

R70.12 Incentive PC & 5300

1. Added telnet interface for FOE upload and download (PC only).

R70.11 Incentive PC & 5300

1. **Automatically Add EtherCAT Devices Option** – Option to not automatically add EtherCAT devices at the start of a QB program and allow the program to do it. Telnet feature command also added: "`disable EtherCAT automatic device installation`", "`enable EtherCAT automatic device installation`" and "`get EtherCAT automatic device installation`".
2. Fix raw sockets with serial port session. Worked on 5300 but on IncentivePC byte order for socket and IP address were backwards due to processor difference.
3. Added ability to write a -1 to serial port connection register to shutdown current connection.

R70.10 Incentive PC & 5300

1. **Memory Diagnostics** – Added ability to add DIAG2 to the parameters when the .rta file is loaded to display the amount of total memory allocated each time a message is logged to the IO console (PC version only).
2. **Secondary Master** – Added option to allow setting secondary master when writing user options from EtherCAT Explorer screen in QB.

R70.09 Incentive PC & 5300

1. **Synchronized with 5300 –** Code synchronization with 5300, no changes from R70.08 except for some additions of conditionals for compiling so can use same source code for 5300 ARM based PLC.
2. **Diagnostic Messages –** Enhanced diagnostic messages with regards to startup and interaction with the EtherCAT Master so more information for the user.

R70.08

1. **Diagnostics –** Added ability for API diagnostics when DIAG1 is added to the startup parameters list of 5300PLC.rta. Additional diagnostics will appear on the IO console windo and also be logged to _logPLC.txt. Also fixed Mutex on diagnostic logging.
2. **Transaction Sequence Numbers –** Added the ability to monitor for duplicate API write packets and not duplicate the write but to acknowledge it since it occurred previously and the API RX queue must have had issues causing a re-connect to occur.
3. **Reconnect –** Added the API ‘RCLOSE CONNECTION’ so signal a queue reconnect is needed from the Windows API and that the last transaction sequence number should be used when restarted since it is not a new connection. This prevents duplicate writes if the original one had been received properly.

R70.05

1. **Standalone Operation –** Added ability for CTPLC_1 to run without the EtherCAT Master. Useful for operation where just serial ports or network protocols such as Modbus are required.

R70.04

1. **Variant Table Access –** Corrected a problem where if reading a table or vector for display in a QB watch window and a different data type was encountered it would fail to return the proper length of the table causing the table not to be displayed properly up to the point of the differing data type.
2. **Telnet Register Range Display –** There was a problem if you requested a dump of registers within a range and they were variants, only the first value returned was correct. Example: BlueFusion> 36701-36705

R70.03

1. **Shutdown –** Corrected problem in conjunction with Windows registry change to extend the time it takes to shut the system down when using the INtime stop menu item. Default time was 500 ms which did not allow the disk buffers to flush to SSD causing file corruption of nvars in larger programs.