

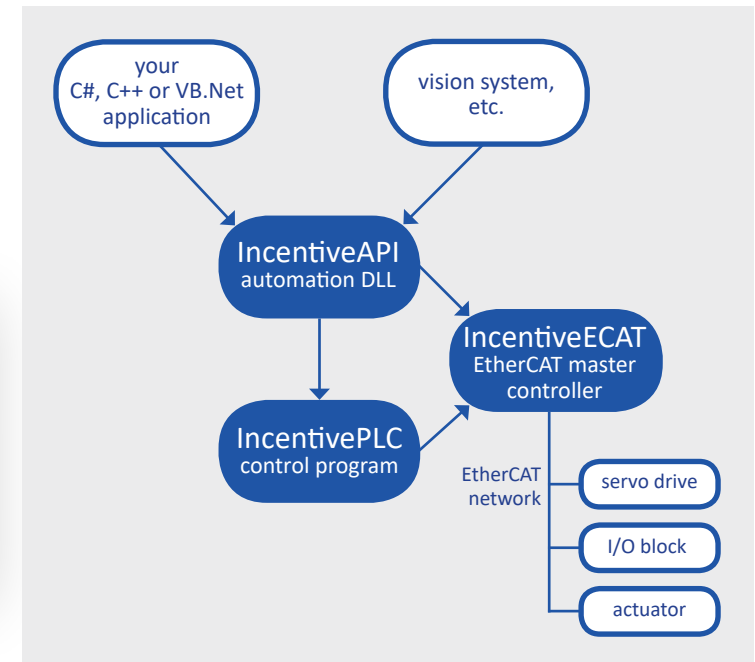
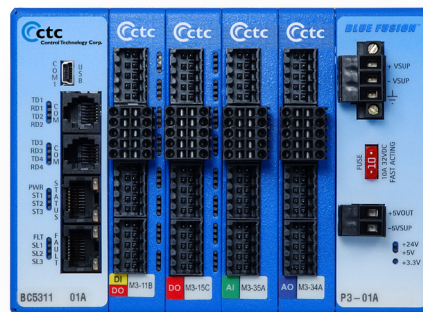


*Something new for
Automation Engineers: **Choice.***

With CTC's Incentive™ and a broad range of high-performance control hardware, you are no longer tied to a single platform, language, network or architecture to implement your designs.

Your Application – Your Choice

- Desktop, embedded or PLC.
- C#, C++, VB.Net, or QuickBuilder.
- EtherCAT, Modbus, DeviceNet, or BACnet.
- Distributed or standalone.



Advanced automation control for technology companies.



Signature Technologies

For manufacturers needing the advantages that advanced technologies can bring, CTC offers choices of platform, programming, networks and architectures. Explore further using the links at left to learn more.

Incentive™ Software

Running on standard PC hardware, Incentive software components give you unprecedented flexibility in control system design. [Click to learn more.](#)

Series 5300 Hardware

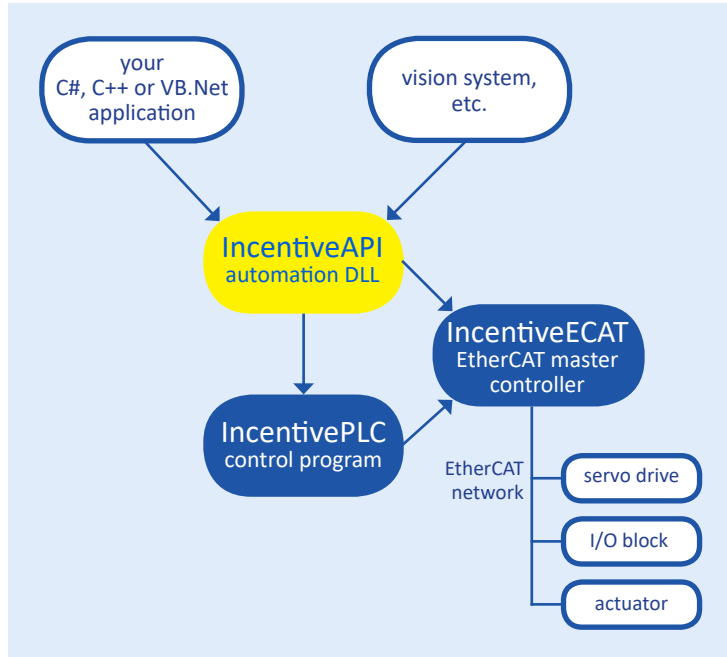
Comprehensive control in a small package – programmable automation controllers for the 21st century, with integrated motion, data handling and a powerful high-level language, all with full integration and portability with Incentive software. [Click to learn more.](#)



IncentiveAPI

Connecting Your Code to Your Process

IncentiveAPI is a Windows DLL that provides a high-level control API for your code written in C#, C++ or VB.Net. It's also useful in interfacing third-party systems such as vision systems to your process. The IncentiveAPI library can be used to interact with IncentivePLC control programs, either locally or remotely. You can also use it to directly send commands to EtherCAT devices through the IncentiveECAT component.



Principal Object Classes

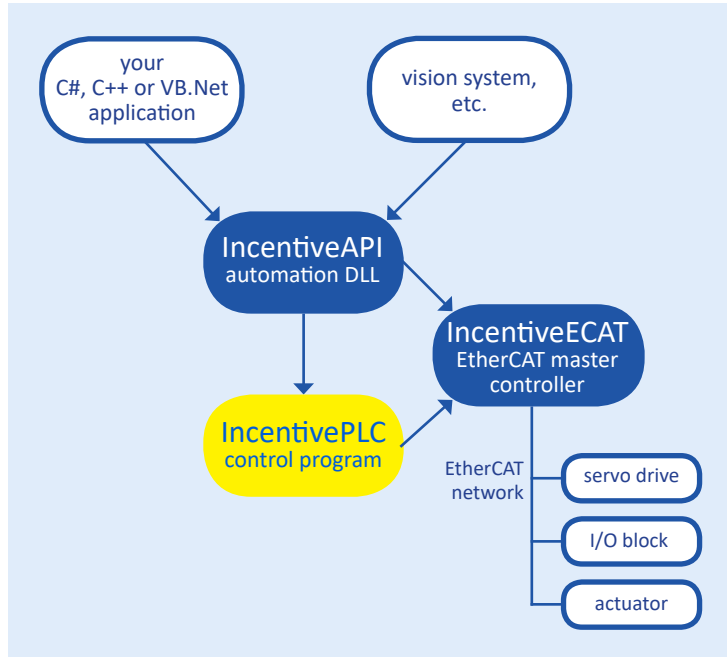
- **Controller::Axis** – primarily for motion control.
- **Controller::PLCLogic** – supports access to IncentivePLC registers and I/O

Example Methods

- **move_at_for**
- **segmove_slew_until_position**
- **gear_at_in_after**
- **set_master**
- **get_register**
- **input32**

More Information

For more information, or to download the **IncentiveAPI help file** for complete documentation of all supported objects and methods, refer to the **CTC website**.

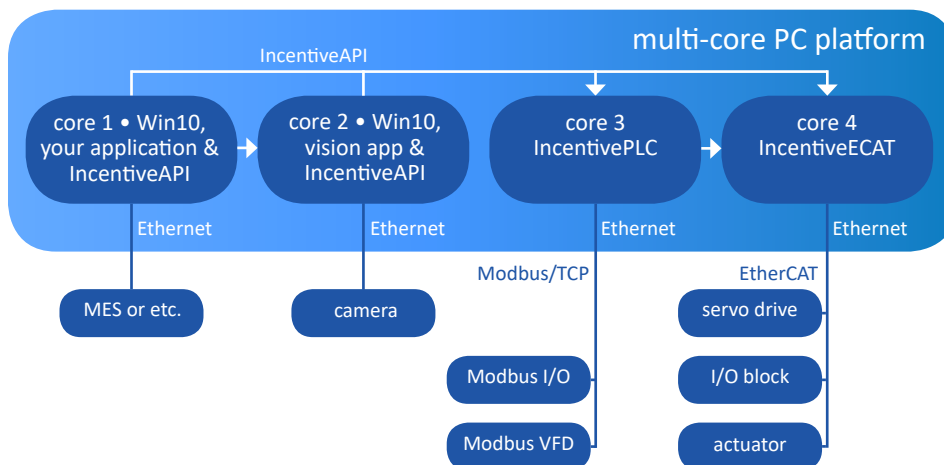


IncentivePLC

High-performance PC-based Control Software

With IncentivePLC, you have a powerful, multitasking programmable controller supporting function calls, asynchronous event triggers, motion commands and much more, all running on a standard PC platform.

IncentivePLC runs via a supplied RTOS and supports multicore CPU execution, allowing you to simultaneously run Windows 10 (or 7) on the same system. This provides high-speed execution of control functions, while at the same time allowing you to run Windows-based applications such as HMIs and vision software on the same machine.



An example of a PC with a multicore CPU showing Incentive components running alongside Windows applications

More Information

For more information about the QuickBuilder language supported by IncentivePLC, [click here](#) to go to the QuickBuilder page.

To view tutorial videos, to download the **QuickBuilder Reference Guide**, or to receive a trial version of IncentivePLC, refer to the [CTC website](#).

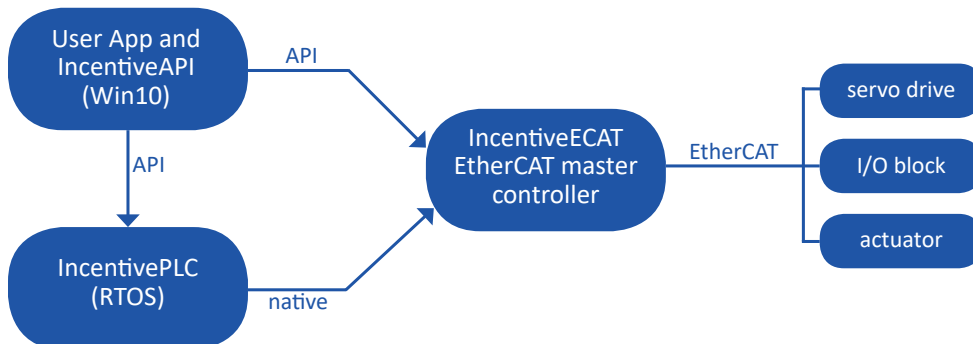
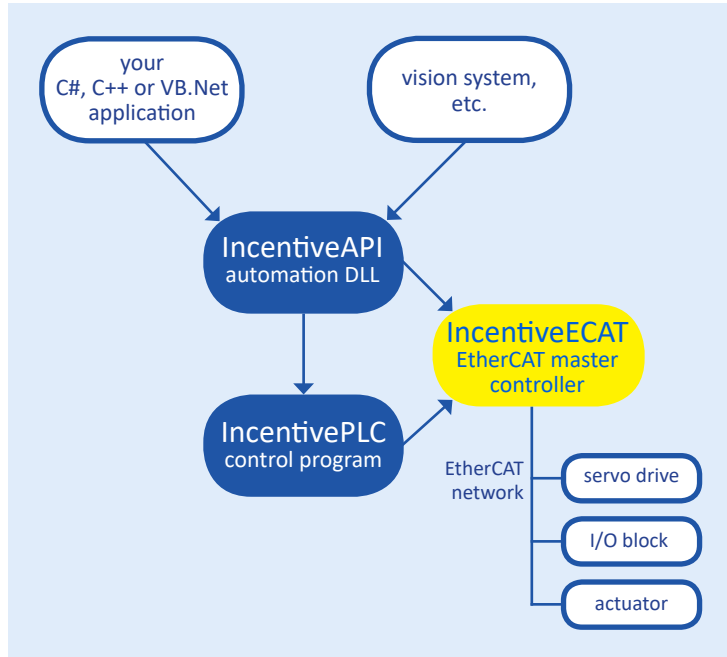


IncentiveECAT

Vendor-independent Motion & I/O

IncentiveECAT is a software-based EtherCAT master controller that runs on a PC platform. By assigning a CPU core to the IncentiveECAT process using the supplied RTOS, you can achieve great performance without dedicated hardware.

Just like CTC's hardware-based EtherCAT master controller, IncentiveECAT automatically scans the network and configures the attached devices, so you don't have to. By automatically handling many of the fine distinctions among vendors' products, we're helping fulfill the promise of open systems to make life a bit easier.

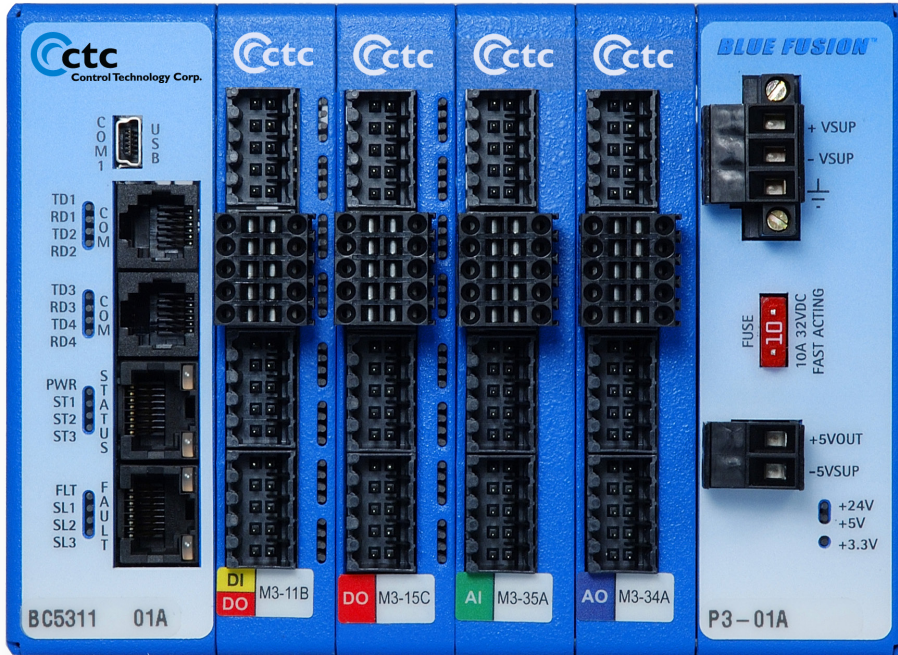


User and third-party code can communicate with IncentiveECAT via the IncentiveAPI, while IncentivePLC can communicate natively

More Information

To learn more about CTC support for EtherCAT:

- See the [EtherCAT page](#) of this document.
- Refer to the [EtherCAT Application Guide](#), available on the [CTC website](#).



Series 5300 systems are available in a wide variety of capacities and configurations – consult a CTC support specialist for information

Series 5300

Programmable Automation Controllers

Featuring program portability with the IncentivePLC control platform, Series 5300 controllers offer a panel-mountable hardware solution for both standalone and distributed control applications.

With a wide array of high-density digital, analog and motion control modules available, and extensive data communications and networking capabilities, these systems can serve either as the principal control system on even the most complex equipment designs, or as intelligent remotes.

More Information

To learn more about Series 5300 modules and capabilities, refer to the following pages:

- **Local and Remote Digital I/O**
- **Analog Sensing and Control**
- **High-performance Servo and Stepper Control**
- **EtherCAT master controller**



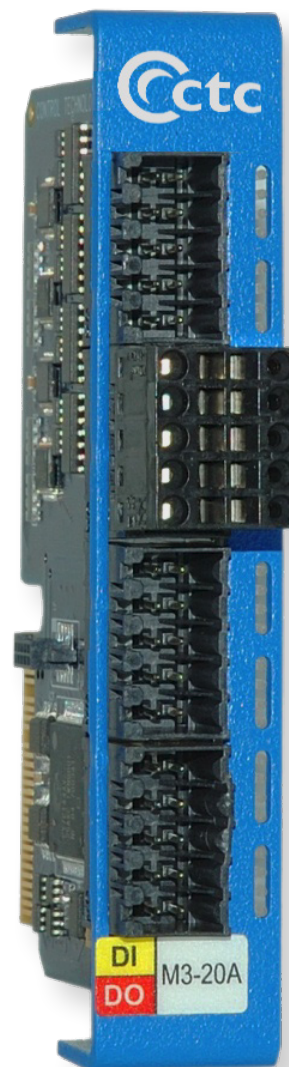
Series 5300: Local and Remote Digital I/O

High density and complete local/remote flexibility using a variety of device-level protocols are all supported by 5300 Series systems.

Local Digital I/O Modules

Model	Inputs	Type	Outputs	Type
M3-10A	16	sinking +V _s	16	500mA sinking +V _s
M3-10B	16	sinking +5V	16	500mA sinking +V _s
M3-20A*	16	sinking +V _s	16	500mA sinking +V _s
M3-20C*	16	sinking +5V	16	500mA sinking +V _s
M3-11A	16	sourcing +V _s	16	375mA sourcing +V _s
M3-11B	16	sourcing +5V	16	375mA sourcing +5V
M3-21A*	16	sourcing +V _s	16	375mA sourcing +V _s
M3-21C*	16	sourcing +5V	16	375mA sourcing +5V
M3-16A	32	sinking +V _s		
M3-16B	16	sinking +V _s		
M3-16C	32	sinking +5V		
M3-16D	32	sourcing +5V		
M3-16E	32	sourcing +V _s		
M3-16F	16	sourcing +V _s		
M3-14A			32	500mA sinking +V _s
M3-14B			16	500mA sinking +V _s
M3-15A			32	375mA sourcing +V _s
M3-15B			16	375mA sourcing +V _s
M3-15C			32	375mA sourcing +5V
M3-12A			16	4A form A relay

All modules are optically isolated (exc. relay outputs). V_s refers to supply voltage of controller (10-32 VDC, typically 24 VDC nominal). See data sheets for per-module and per-controller total output current limitations, and other specifications.



Levels of Support

- Remote I/O via Modbus/TCP and RTU, Ethernet/IP, DeviceNet and EtherCAT are programmed identically to local I/O.
- EtherCAT and DeviceNet master and slave modules available.
- Native Modbus master and slave operation.
- High I/O densities support large, complex systems.
- Smart modules (denoted *) support advanced local features: local high-speed counting with set-point outputs, PWM.

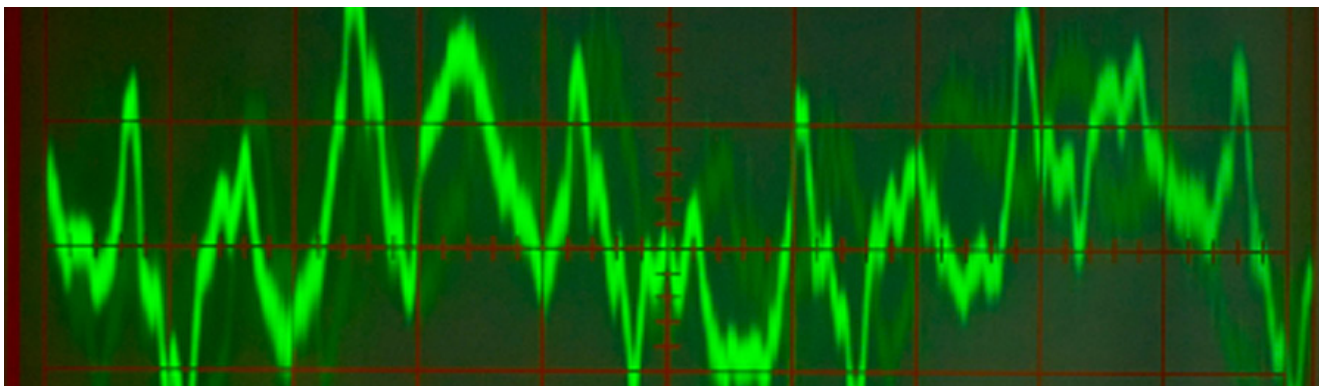
More Information

- **5300 Series Datasheets**
- **QuickBuilder Reference Guide**



Series 5300: Analog Sensing and Control

Because the real world is (mostly) analog, Series 5300 systems can accept a variety of analog I/O modules to interface to a diverse range of sensors and actuators.



Input Modules

Model	Inputs	Type	Differential
M3-31A	16	±10V	
M3-31B	8	±10V	
M3-31C	16	4-20 mA	
M3-31D	8	4-20 mA	
M3-33A	8	±20 mV	•
M3-33B	4	±20 mV	•
M3-33C	8	±100 mV TC	•
M3-33D	4	±100 mV TC	•
M3-33E	8	±10V	•

All input modules have 16-bit resolution, optically isolated inputs, digital filtering. TC=thermocouple inputs supporting types E, K, J, R, S, T.

Output Modules

Model	Outputs	Type
M3-32A	32	±10V
M3-32B	16	±10V
M3-34A	8	4-20 mA
M3-34B	4	4-20 mA
	4	0-10VDC

All output modules have 16-bit resolution and optically isolated outputs

Levels of Support

- High I/O densities support large, complex systems.
- 16-bit resolution.
- Optically isolated.
- Support for high- and low-level voltage signals, current and thermocouple signals.
- Input modules support individually-programmable digital filtering.

More Information

- **5300 Series Datasheets**
- **QuickBuilder Reference Guide**



Series 5300: High-performance Servo and Stepper Control with Full Integration

As an alternative to networked motion control, Series 5300 systems also support servo and stepping motor control modules that can drive inexpensive external amplifiers, yet can achieve high levels of performance and integration.

Local Motion Control Modules

	Model M3-40A Dual Axis Servo	Model M3-40B Dual Axis Stepper	Model M3-40C Dual Axis Stepper
Output Type	Analog torque or velocity	Step & Direction ±4M steps/sec.	Step & Direction ±4M steps/sec.
Output Voltage	±10V, 16-bit	24 V	5 V
Velocity Resolution	1 count/sec.	1 step/sec.	1 step/sec.
Common Specifications			
Encoder Rate	17.5 MHz		
Loop Update	800 µS		
Registration Accuracy	1 encoder count		

Levels of Support

- Closed-loop servo or stepping motor control.
- Linear, S-curve, Cam, Spline, Gear, Move on a gear, Virtual master and Segmented move modes.
- Numerous additional functions integrated in modules.

Additional Functions Supported via Module I/O points

- *Programmable Limit Switch outputs based on encoder position.*
- *High-speed counting to 10 MHz.*
- *Pulse outputs with 1.0 mS accuracy.*
- *Frequency output to 500 Hz.*

More Information

- [Series 5300 Data Sheets](#)
- [QuickBuilder Reference Guide](#)



Series 5300: EtherCAT Master Controller

EtherCAT support in the Series 5300 is available via the model M3-41A EtherCAT Master Controller module. It provides the same capabilities as the IncentiveECAT software component, but in a dedicated hardware configuration.



Levels of Support

- Local execution of Motion Sequence Blocks for fast response.
- Up to 16 axes per network.
- Digital output support for PLS, PWM and calibrated pulse output.
- Supports all MSB and IncentiveAPI instructions.

More Information

- [EtherCAT information page](#)
- [EtherCAT Applications Guide](#)

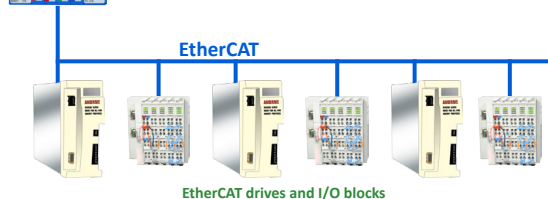


EtherCAT: Simplified Wiring for High-performance Motion Systems and I/O

CTC offers fully-integrated EtherCAT support with IncentiveECAT, as well as in the 5300 Series controllers where EtherCAT axes are programmed identically to local motion axes. Network configuration is automatic.

A Simple EtherCAT Network

Series 5300 controller with M3-41A or PC with IncentiveECAT software



video (2:26)

An Introduction to EtherCAT

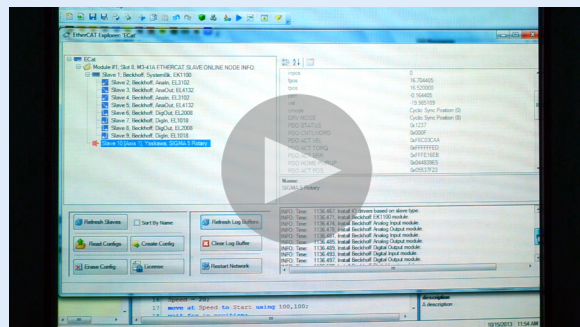


Levels of Support

- Automatic configuration.
- Up to 64 axes per network (platform dependent).
- Master/slave clock syncing with simultaneous motion trigger.
- Linear, circular interpolation.
- Modes include cyclic sync position, interpolated position, profiled velocity, and position.
- Segmented, geared, position, velocity and cammed moves.

video (6:05)

Setting up an EtherCAT Network



video (6:56)

A multi-vendor EtherCAT Network



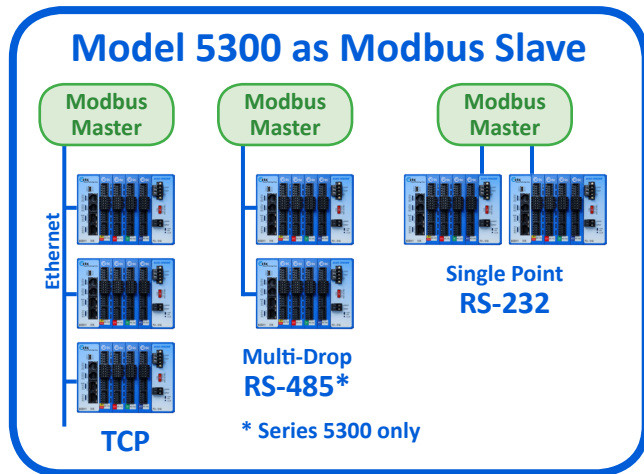
More Information

- **Series 5300 EtherCAT Master**
- **IncentiveECAT page**
- **CTC website**
- **EtherCAT Applications Guide**

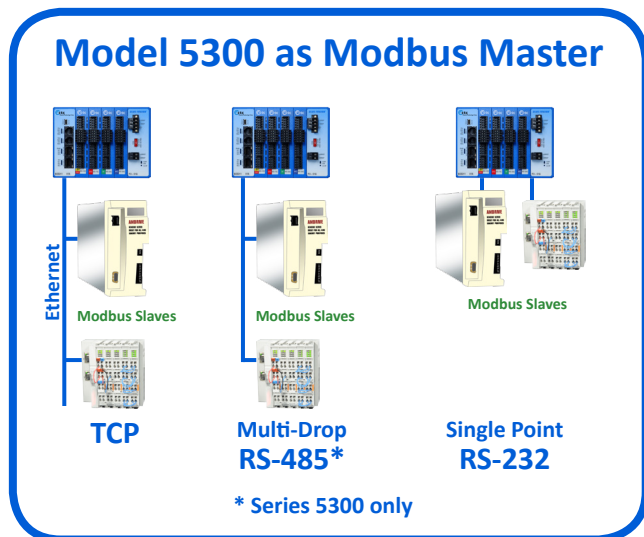


Modbus® Master/Slave Operation – TCP or RTU

CTC provides robust, native support for Modbus protocol in its various formats. With integrated support for a variety of protocols, CTC controllers can even be used for protocol translation.



CTC supports Modbus Slave protocol commands as built-in functions on Ethernet or RS-232 ports with IncentivePLC and Series 5300 controllers. The Series 5300 also supports RS-485 multi-drop operation.



As a Modbus Master, you set up a connection block in your program to configure your slave devices based on IP address (Ethernet) or node address (RTU). You can set **unique poll rates** for each slave device and map slave device data to internal registers in your master.

Levels of Support

- Master or Slave operation.
- Ethernet or Serial connection.
- Data maps to controller registers for easy access.
- Configurable polling rates per slave device.
- Various levels of support in other CTC platforms.

More Information

- [CTC website](#)

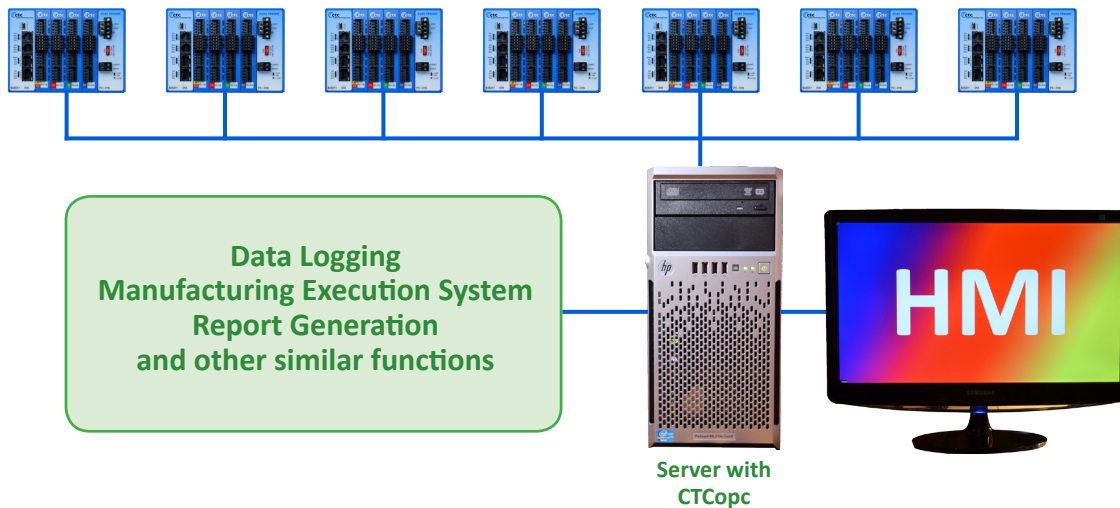


OPC Connectivity with CTCopc Simplifies Integration

Connecting to HMI, data logging, reporting and similar applications from a variety of vendors is greatly simplified through the use of CTCopc, a 32-bit Windows application providing a native interface to CTC controllers.

CTCopc can be used with Series 5300 controllers as well as with IncentivePLC. In the case of an IncentivePLC implementation, CTCopc can be installed locally on the same system or on a separate remote system.

Example Model 5300 Network with OPC Connectivity



Levels of Support

- Features include scaling, floating point (float32 and float64), string, and data table support.
- Connect up to 500 systems.
- “Scatter/Gather” packet greatly improves performance.

More Information

- [CTC website](#)
- [CTCopc Reference Guide \(PDF\)](#)

OPC performance is greatly enhanced through the use of CTC’s “Scatter/Gather” technology. Users can define up to 50 selected items to track in the controller, which can include analog values, registers, servo positions and other numeric parameters. These are refreshed in a single transaction, instead of requiring separate transactions for each data source, thus allowing much faster refresh rates.



QuickBuilder™ State Language Programming

State languages offer the most direct, intuitive path for converting design ideas into a working program. The QuickBuilder is a high-level language with automation-specific commands that greatly accelerate the design, debug and modification cycles in commissioning a new machine or process.

A Simple QuickBuilder Step

```
// This is a QuickBuilder step.  
// We'll turn on a named output, and send a value  
//   to an analog output controlling a heater.  
// Once the correct temperature has been reached,  
//   we'll proceed to the next step of the program.  
set CLAMP on;  
store STARTING_TEMP to HEAT_CONTROL;  
if TEMP >= READY_TEMP goto next;
```

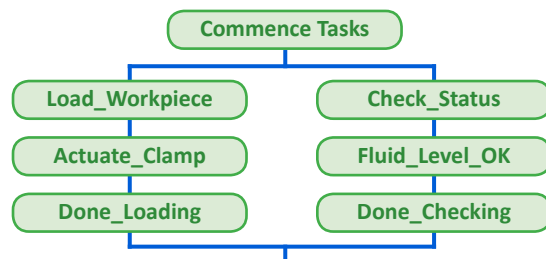
A Sequence of Steps

A “step” defines the only instructions active within a task at that moment, and corresponds to the physical state of the machine or process being controlled.

Levels of Support

- High-level, visually-obvious coding.
- Parallel sequences using multitasking.
- High performance due to lack of “scan-time” constraints.
- More suited to today’s computationally-intensive applications.

Multitasking for Parallel Events



Up to 96 parallel, asynchronous tasks may be accommodated.

The QuickBuilder™ programming environment offers multiple languages in a comprehensive development tool for IncentivePLC and Series 5300 controllers.

More Information

- [CTC website](#)
- [QuickBuilder Reference Guide](#)



QuickBuilder: Mathematical Functions for Computationally-Intensive Applications

In applications requiring intensive mathematical computation (for example, dynamically-generated complex motion vectors), QuickBuilder supports a range of math functions to 64-bit precision.

Available Math Functions and their return values

max(a,b) - maximum of two values
min(a,b) - minimum of two values
sin(x) - sine of (x in radians)
cos(x) - cosine of (x in radians)
tan(x) - tangent of (x in radians)
asin(x) - arc sine of x in radians
acos(x) - arc cosine of x in radians
atan(x) - arc tangent of x in radians
atan2(y,x) - arc tangent of y/x in radians
exp(x) - Euler's constant e to the power of x
log(x) - natural logarithm of x
log10(x) - common logarithm of x
pow(x,y) - x raised to the power of y
sqrt(x) - square root of x
hypot(x,y) - square root of (x² + y²)
sinh(x) - hyperbolic sine of x
cosh(x) - hyperbolic cosine of x
tanh(x) - hyperbolic tangent of x
abs(x) - absolute value of x
ceil(x) - x rounded up to nearest integer
floor(x) - x rounded down to nearest integer
frac(x) - fractional part of x
sign(x) - sign of x
sign2(x,y) - sign of x applied to y
fmod(x,y) - remainder of x/y

Other Functions

Bitwise Functions

bit(x,n) - value of nth bit in x
_rol(x,n) - x with bits shifted left n times
_ror(x,n) - x with bits shifted right n times

String Functions

trim(x,y) - string x with leading/trailing characters y removed
left(x,n) - leftmost n characters of string x
right(x,n) - rightmost n characters of string x
mid(x,s,n) - n characters of x starting at s
padl(x,n,p) - pad left of string x with character p to length n
padr(x,n,p) - pad right of string x with character p to length n
len(x) - length of string x
compare(x,y) - lexical comparison of strings
find(x,y) - position of string y in string x

Other Functions

addr(var) - address of variable var
isdone(tasks(s)) - returns 1 if specified task(s) are complete

Constants

pi - trigonometric constant π

Levels of Support

- Math functions are double-precision (64-bit floats).
- Some are type-independent (refer to Reference Guide).

*Please refer to the
QuickBuilder Reference Guide
for more detailed information
about the use of these functions.*

More Information

- QuickBuilder Reference Guide



QuickBuilder Motion Sequence Blocks optimize performance

Motion Sequence Blocks (MSBs) are blocks of specialized motion control instructions, automatically downloaded and executed locally either on a servo or stepper control modules or in the IncentiveECAT core. This alleviates the main CPU of the burden of executing time-critical instructions, especially important in large, sophisticated systems.

A Simple MSB

```
// This MSB initiates a trapezoidal move.  
// Once the end position is reached, a delay of  
// 250mS is triggered, after which another MSB  
// called GoBack is triggered in the background.  
move trap for short_move using alarming_speed;  
wait for in position;  
delay 250;  
end and start GoBack BG;
```

Separate Execution but Logically Integrated

MSBs are executed directly on the related motion module, although extensive bidirectional communication with the main program is supported.

Levels of Support

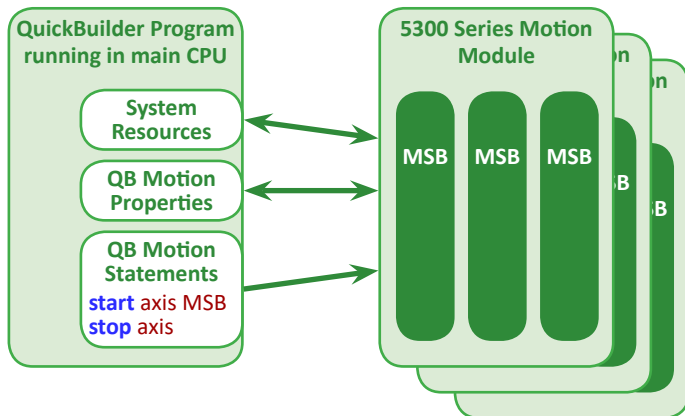
- High-level motion command set.
- Support for gearing, spline/camming, slewing, trapezoidal and absolute/relative position.
- Segmented moves.
- Linear/circular interpolation (EtherCAT).
- Programmable Limit Switch.

More Information

- **MSB Instruction Set**
- **CTC website**
- **QuickBuilder Reference Guide**

How MSBs relate to the main program...

MSBs are preloaded into the appropriate axes at power-up. They run in parallel to the QuickBuilder program in the main CPU or core.





QuickBuilder Motion Sequence Blocks: Instruction Set*

Simple Motion

move to position { using *acc, dec* }
move at maxvelocity to position { using *acc, dec* }
move trap to position using rate
move in time to position { mode *n* }
move for displacement { using *acc, dec* }
move at maxvelocity for displacement { using *acc, dec* }
move trap for displacement using rate
move in time for displacement { mode *n* }
wait for in position
new endposition position using rate
new endposition relative displacement using rate
slew begin
slew at velocity in time
slew for displacement
slew end

Utility

stop { slewed using *rate* }
drive enable
drive disable
delay time ms
variable = expression
zero feedback position
zero target position
zero following error
reset
if condition then *variable = expression*
wait until condition

Host Register

host read *variable, register* { *row, column* }
host write *variable, register* { *row, column* }

Segmented Move

segmove table clear
segmove table accdec to vel using rate
segmove table accdec to vel for displacement
segmove table slew until position
segmove table stop at position using rate
segmove table start relative

Loading Tables

table *n* clear
table *n* addpair *xexpression, yexpression*
table *n* addseries pairs
table *n* copy from rowOffset1 to table *m* *rowOffset2 numRows*
table *n* loadoffset *rowOffsetFile, numPairs, rowOffsetTable*
table *n* loadseries *source fileName*

Spline/CAM

table *n* continue
table *n* precompute
table *n* start imethod *tscale, rpscale, repeatcount*
table *n* start imethod cam *mpscale, spscale, repeatcount*
stop table

Position and Capture

set capture transition of input *input* { *gate input gateinput gatestate* }
set capwin range *start, end* using *reference* { *arm* }
wait capture { *if limit of limit goto limitlabel* }

Virtual Master

move master at rate for limit { using *ramp* }

Gearing

gear at numerator : denominator
gear at numerator : denominator in counts
gear at numerator : denominator in counts after accounts
gear for slavecounts in mastercounts
gear for slavecounts in mastercounts after accounts
offset slave by slavecounts in time
wait master counts
wait slave counts
wait source within *start, end*
wait source outside *start, end*
zero masslv counters

I/O

setout outputlist
clout outputlist
pulse output for *n*
pls output using reference definitions
pls output state
wait for transition of input { *or condition* }
generate output output rate freq
generate *n* steps on pair
variable = ctr[n]
ctr[n] = expression
ctr[n] = offset
generate alternate mode

Set

set common bit number state
set common var number value
set looperperiod value
set mode positioning
set mode tracking
set timeout ticks
set target position value
set feedback position value
set target position counts vcounts
set feedback position counts vcounts
set simulated feedback on/off
offset position value
offset position counts vcounts
set master mode { using *global* }

Program Flow

[label]
start MSB mode
end { and start MSB mode }
abort MSB
goto label
if condition goto label
on asynchevent asynchhandler

Common bits and variables

set common bit number state
wait common bit number state
set common var number value
wait common var number range

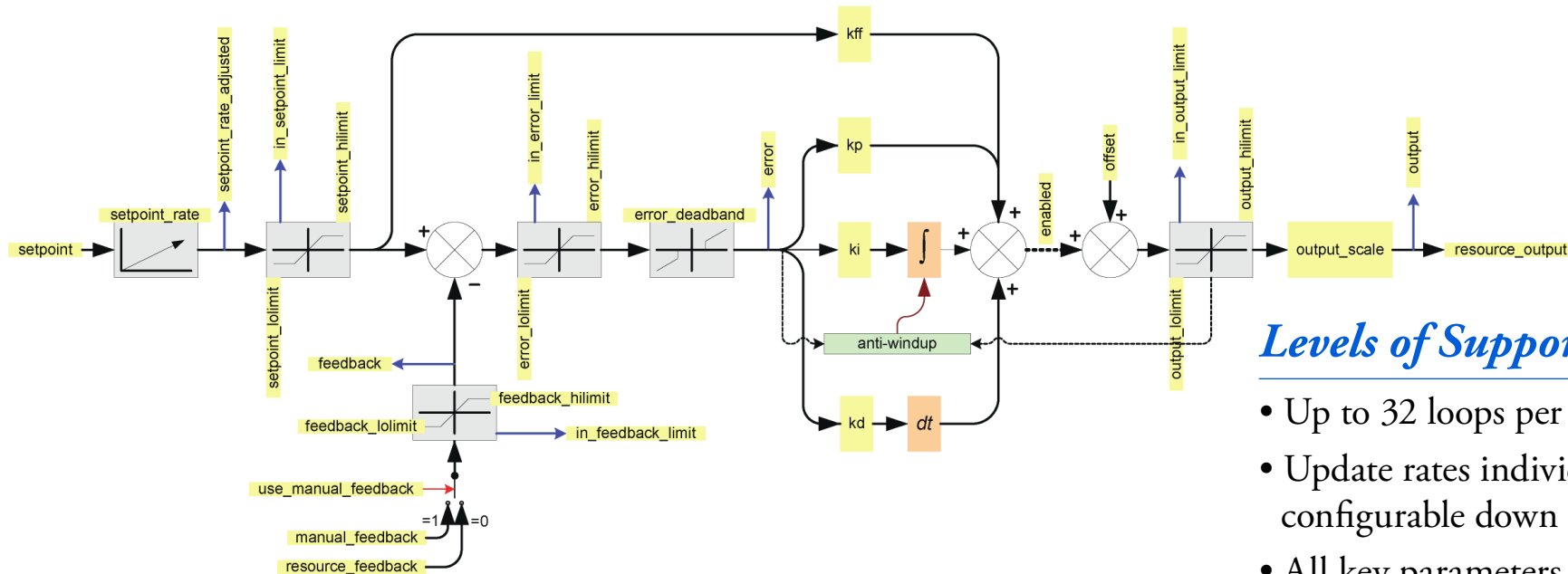
*equivalent functionality is available through the IncentiveAPI library.

For detailed information on these instructions, refer to the **QuickBuilder Reference Guide**.



QuickBuilder: PID Loops as Independent, but Integrated Functions

PID loops may be configured in QuickBuilder to run in the background, independent of the main program flow. The controller program still has complete access to all PID parameters for tuning and setpoint adjustment.



Levels of Support

- Up to 32 loops per CPU.
- Update rates individually configurable down to 1 mS.
- All key parameters are programmatically tunable.
- 64-bit floating point calculations, with 200 nS timing accuracy for improved response.
- Programmable deadband.

More Information

- **QuickBuilder Reference Guide**

PID Loop Properties

resource_feedback, resource_output - these specify the controller resources (analog I/O, registers, etc.) to use for input and output.

manual_feedback, use_manual_feedback - allows for alternative feedback source.

kp, ki, kd, kff - the proportional, integral, derivative and feedforward gain parameters.

integrator_unwind_constant - supports self-discharge of integrator.

derivative_form - supports additional D after PID loop.

tick_multiplier - determines update rate.

enabled - activates/de-activates PID loop.

setpoint, setpoint_hilimit, setpoint_lolimit, setpoint_rate - initial setpoint value and limits.

feedback_hilimit, feedback_lolimit - limits range of feedback applied to loop.

error_deadband, error_hilimit, error_lowlimit - controls behavior of error signal.

output_hilimit, output_lolimit, output_scale, offset - controls behavior of output signal.



QuickBuilder: Options for C Language Programming

Typically used only in Series 5300 installations for applications requiring more complex data manipulation, QuickBuilder supports two options for programming in C.

The 'C' Step

A step of a Quickstep program may contain C code, as in this example of a data conversion routine...

```
// Convert a 32 bit binary value to an ASCII string
// storing in successive integer registers starting at
// _CTC_destRegister

char buffer[32]; // scratch buffer for conversion
int i, j, val;

// Convert 32 bit binary value to ascii bytes locally
sprintf(buffer,"%0.6f",_CTC_floatValue.value);

// Find out length of all characters
i = strlen(buffer);

for (j = 0;j!=i;j++,_CTC_destRegister.value++)
{
    // Get a character
    val = buffer[j];
    // Store it in the next desired sequential register
    regWrite(_CTC_destRegister.value, val);
}

// Set number of converted characters
_CTC_length.value = i;
```

After execution, with a _CTC_floatValue of 3.567, the registers beginning at _CTC_destRegister would contain:

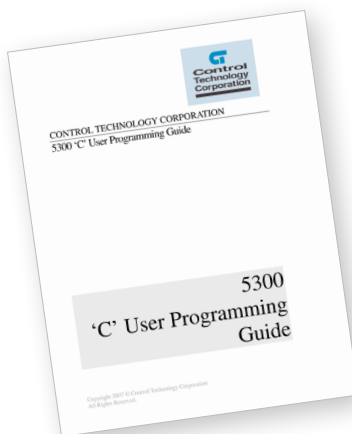
0x33 (ascii '3')
0x2e (ascii '.' - period)
0x35 (ascii '5')
0x36 (ascii '6')
0x37 (ascii '7')

'C' Tasks or Programs

Alternatively, independently-compiled C functions may be stored in a controller.

These may be called as:

- Initialization routines
- Periodic functions
- Independent threads (in parallel to the main controller program)
- The sole exclusive control program



free download:
*'C' User Programming
Guide*

Levels of Support

- Access to all internal resources.
- Access to Comm/Enet ports.
- Resource filters may be implemented.
- gcc-based compiler.
- User functions can be called upon initialization, periodically, or as a Quickstep step or separate thread.

More Information

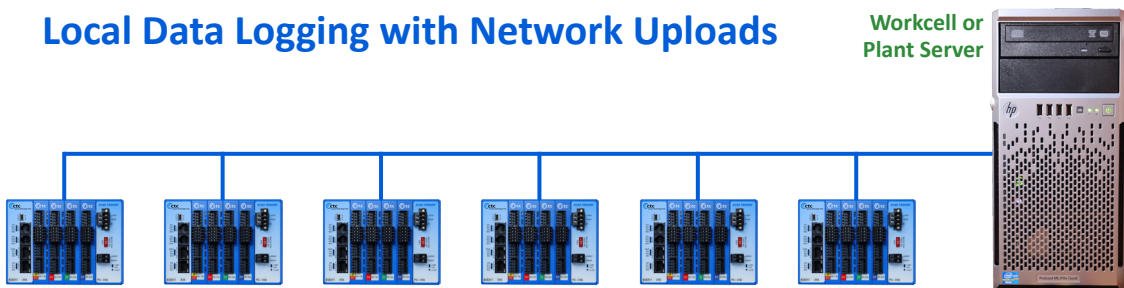
- **5300 'C' User Programming Guide**
- **QuickBuilder Reference Guide**



Comprehensive Local Data Logging

Data logging allows the periodic storing of data within the controller in a format pre-determined by a configuration file. Storage is in nonvolatile memory (NV RAM in Series 5300, SSD when using IncentivePLC), and data may be transferred cleanly at any time via a “snapshot” provision.

Local Data Logging with Network Uploads



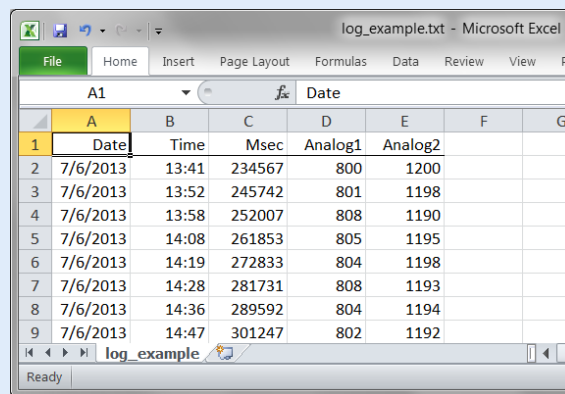
Data uploads to server may be performed periodically (end of shift, daily, etc.) or on demand.

Levels of Support

- Logging from registers, data tables, timestamps or fixed strings.
- Data may be transferred as a CSV file.
- FTP file transfer supported.
- “Snapshot” capability.

Data-Logging Example

07/06/2013, 13:41, 234567, 800, 1200
 07/06/2013, 13:52, 245742, 801, 1198
 07/06/2013, 13:58, 252007, 808, 1190
 07/06/2013, 14:08, 261853, 805, 1195
 07/06/2013, 14:19, 272833, 804, 1198
 07/06/2013, 14:28, 281731, 808, 1193
 07/06/2013, 14:36, 289592, 804, 1194
 07/06/2013, 14:47, 301247, 802, 1192



	A	B	C	D	E	F	G
1	Date	Time	Msec	Analog1	Analog2		
2	7/6/2013	13:41	234567	800	1200		
3	7/6/2013	13:52	245742	801	1198		
4	7/6/2013	13:58	252007	808	1190		
5	7/6/2013	14:08	261853	805	1195		
6	7/6/2013	14:19	272833	804	1198		
7	7/6/2013	14:28	281731	808	1193		
8	7/6/2013	14:36	289592	804	1194		
9	7/6/2013	14:47	301247	802	1192		

...brought into Excel as a CSV file.

More Information

- **Communications & Logging Guide**
- **Data Logging TechNote**



iPad- and iPhone-based Monitoring using qMon App

The qMon app allows you to remotely monitor and change a variety of controller resources, and includes graphing capability with snapshots.



Resource Monitoring Supported

- Registers
- Digital inputs
- Digital outputs
- Flags
- Analog inputs
- Analog outputs
- Motor position/error
- Program status

Levels of Support

- Dynamic monitoring of most controller resources.
- Chart generation and saving to photo stream.
- Monitoring of program status.
- iPad and iPhone optimizations.

**Click for Free
Download
from App Store**

video (5:21)

Introduction to qMon



More Information

- [CTC website](#)
- [qMon Users Guide](#)
- [qMon in iTunes App Store](#)



Compatible Touchscreens for rapid system development

CTC sells and supports a range of compatible touchscreens, along with the companion EasyBuilder Pro configuration software.



Available Models

Model	Size	Resolution	Notes
44150A	15	1024x768	AL
44120A	12.1	1024x768	AL
44105P	10.4	800x600	262K
44100	10.0	800x480	262K
4470A	7.0	800x480	AL
4470	7.0	800x480	
4470NE	7.0	800x480	NE
4443	4.3	480x272	

All models are 16M-color and have plastic IP65 housings and are equipped with Ethernet, USB and Serial ports, except as noted: 262K=262K-color, AL=aluminum housing, NE=no Ethernet port

EasyBuilder Object Summary

Bit lamp – indicator follows bit status
Word lamp – indicator follows word status
Set bit – event (e.g., button press) sets bit
Set word – event sets word
Toggle switch – switch toggles bit
Multistate switch – increments/decrements word
Slider – change word value by dragging cursor
Function button – perform a system operation
Numeric input/display – change/display numeric data
ASCII input/display – change/display alphanumeric data
X, Y move animation – move an object via pixel coordinates
Spot move animation – move an object on a predefined path
Direct window – popup window on bit status change
Indirect window – popup window depending on Word's value
Bar graph – display data as changing rectangle
Alarm display – display predefined text strings when bit changes
Trend graph – plot active data on a trend
Data history display – show contents of data logger
Meter – display one of four different types of meters
Scrolling alarm – display predefined text strings when bit changes
Event display – display and log predefined text strings when bit or word changes
Data logging – store data to USB or CF card memory
Utility – change window, backlight control, macro execution, data transfer



Levels of Support

- 16M and 262K-color TFT LCD displays.
- IP65 / NEMA 4 Faceplates.
- Supports Modbus/TCP and Ethernet IP (Ethernet models) as well as DF1 protocols.
- Supported by free EasyBuilder Pro configuration software.

More Information

- **4400 Series Datasheets**



In this video... CTC customer
*Jim Gallagher describes his experience
working with CTC on a system to fabricate
large-scale composite aircraft components.
CTC customized a data interface to
SolidWorks for this project.*

A Control Partner with a Difference

We take a personal interest in our customers' goals. This is demonstrated in an exceptional level of support and a willingness to create tailored solutions to unique application requirements.



Although the breadth of the CTC hardware and software product line fulfills most applications, the availability of a customized solution ensures that, for our customers, there are no show-stoppers.



The Things We Do for our Customers...

We take our customers' interests to heart, in a way that many find surprising. Today, some of our most advanced features are generalizations of specific needs of a customer, which we accommodated. The result is unsurpassed value in an automation controller... powerful, tailored, supported.

A large **packaging machinery** OEM was spending over \$1000 per machine for external PLS hardware. We incorporated this functionality in their controller, resulting in a substantial reduction in cost and complexity.

A well-known **golf ball manufacturer** needed precisely repeatable pulses for a precision spray paint application. We incorporated pulse outputs with 1 mS repeatability as a standard feature.

An advanced supplier of **semiconductor CVD equipment** needed to transfer vast quantities of data from their controller at high speed. We developed technology allowing a 12 mS total transfer time via Ethernet.

One of the world's largest makers of **computer disks** needed to transfer blocks of data to their SCADA system serially and via Ethernet. We were able to achieve data turnaround times in the 3-5 mS range.

An OEM manufacturer of **plastic bag machines** needed highly accurate, very high speed registration for a cut-and-perf operation. We developed hardware with firmware support for registration accuracies of 1 encoder count.

A volume manufacturer of **industrial robots** needed a controller in a pendant format. We repackaged and value engineered a solution that precisely met their needs while reducing their costs.

A **semiconductor equipment** manufacturer needed a high-speed, repeatable approach to controlling mass-flow controllers. We developed technology allowing group updates of these controllers in a fraction of the time.

A leading innovator in the **weighing field** needed a fast way to select best-of-n-bins for a net weighing system. We integrated the selection algorithm into firmware, greatly improving the machine's cycle time.

More Information

- **CTC website**