25 South Street
Hopkinton, MA 01748
Phone: 508.435.9595
Fax: 508.435.2373
www.ctc-control.com

Model 5100/5200 Quick Reference Register Guide

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
5100 series	> B	>4.05
5200 series	> C	>= 5.00.31

Blank Page

Table of Contents

Blank Page

General Registers

Register Number	Description

General Purpose Registers

Register Number	Description
	General Purpose registers are 32-bit. They may be accessed for a variety of reasons.
1-8	Internal counters: R/W, counters may also be used as general purpose registers.
9-125	General purpose registers: R/W, data in these registers is stored in volatile memory
129-130	General purpose registers: R/W, (volatile)
133-500	General purpose registers: R/W, (volatile)
501-1000	General purpose registers: R/W, data is stored in nonvolatile memory.
32001-36000	General purpose registers: R/W, data is stored in nonvolatile memory.

Data Table Registers

Register Number	Description
126	Data table pointer: R/W, used with column reference.
131-132	Data table row and column pointers: R/W, Used with register 9000
9000	Access to the data table: R/W, phantom - works with 131 & 132.

Phantom Register

Register Number	Description
127-128	Pointer for phantom register and phantom register

Alternate Access to Resources

Register Number	Description
1001-1999	Alternate Access to Outputs 1-999: R/W, 0 = off, 1 = on.
2001-3024	Alternate Access to Inputs 1-1024: R only, 0 = open, 1 = closed.
10001-10032	Access Outputs as a 32-bit number: R/W
10101-10164	Access Outputs as a 16-bit number: R/W
10201-10328	Access Outputs as an 8-bit number: R/W
11001-11032	Access Inputs as a 32-bit number: R only
11101-11164	Access Inputs as a 16-bit number: R only
11201-11328	Access inputs as an 8-bit number: R only

Flags

Register Number	Description
13005	Flag Access as a 32-bit number: R/W, Alternate access to Flags 1-32.
13021-13024	Flag Group Registers
13201-13232	Flag Bit Access as a 1-bit number: R/W, 0 = off, 1 = on.

Access to Analog Input and Output Points

Register Number	Description
8001-8256	Analog Outputs Alternate Access: R/W
8501-8756	Analog Inputs Alternate Access: R only
9001-9256	Analog Input Conversion Type: R/W.

Value	*Description*
0	**No Conversion**
1	Cold Junction Temperature Algorithm
2	B1-50A RTD Algorithm
10	K-Type Thermocouple Linearization Algorithm
11	J-Type Thermocouple Linearization Algorithm
12	T-Type Thermocouple Linearization Algorithm
13	E-Type Thermocouple Linearization Algorithm
14	R-Type Thermocouple Linearization Algorithm
15	S-Type Thermocouple Linearization Algorithm

<table>
<tr><th>Register Number</th><th>Description</th></tr>
<tr><td>9501-9756</td><td>Analog Input Units: R/W
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td><b>0</b></td><td><b>+/- 10,000 (All modules, Also See Register 13010)</b></td></tr>
<tr><td>1</td><td>+/- 10,000,000 (All modules, Also See Register 13010)</td></tr>
<tr><td>2</td><td>+/- 20,000 (alternate setting for +/- 20 mV)</td></tr>
<tr><td>3</td><td>+/- 100,000 (alternate setting for +/- 100 mV)</td></tr>
<tr><td>4</td><td>4,000 – 20,000 (alternate setting for 4-20 mA)</td></tr>
<tr><td>11</td><td>Celsius (for CJC,RTD, and Thermocouple Conversions Only)</td></tr>
<tr><td>12</td><td>Fahrenheit (for CJC, RTD, and Thermocouple Conversions Only)</td></tr>
<tr><td>13</td><td>Kelvin (for CJC, RTD and Thermocouple Conversions Only)</td></tr>
</table></td></tr>
<tr><td>9991-9994</td><td>Controller Local Temperature Sensors, 9991 is local (on the CPU/Base Module), 9992 to 9994 are on the I/O Expansion Racks when they are present.</td></tr>
<tr><td>9996</td><td>Analog Input Cold Junction Virtual IO index: R/W, Selects the data that is available in Register 9997 where 0 = local (default) 1,2 and 3 are the I/O Expansion racks connected to the local system. 4-7 are the Base Module and Expansion Racks from the first virtual I/O system, references 21500 block</td></tr>
<tr><td>9997</td><td>Analog Input Cold Junction Temperature in Celsius: R/W
There are multiple copies of this value, selected by the value of Register 9996. This provides the separate Cold Junction Compensation values for the CPU/Base module and the different I/O Expansion racks.
Valid values are 0 to 700 (70.0°C).
When this is set to zero then the local temperature sensor (Registers 9991 to 9994) will be used for cold junction compensation.
<b>Default = 250 (i.e. 25.0°C)</b></td></tr>
<tr><td>13010</td><td>Analog Input Scaling: R/W
0 = uVolts (i.e. +/-10000000)
<b>1 = mVolts (i.e.+/-10000)</b></td></tr>
<tr><td>13022</td><td>Analog Output Scaling: R/W
0 = uVolts
<b>1 = mVolts</b></td></tr>
<tr><td>17508-17756</td><td>Analog Input Cold Junction Compensation selection, 0 indicates that register 9997 is used for the Cold Junction compensation (which will use the local temperature sensors if the register is zero), a value from 1 to 256 indicates which analog channel is providing the CJC reference for this channel.</td></tr>
<tr><td>18501-18756</td><td>Analog Input Digital Running Filter Length (Default = 1): R/W, range 1 to 255.</td></tr>
<tr><td>19001-19256</td><td>Analog Input Minimum Threshold Set Point: R/W</td></tr>
<tr><td>19501-19756</td><td>Analog Input Maximum Threshold Set Point: R/W</td></tr>
</table>

Script and Flash Disk Registers

12311	Script Register: R/W; writing a numeric to this register will cause the corresponding script to be executed. For example: writing a 4 to this register will cause `Script004.ini` to be executed.
12312	Script Result Register: R, 0 = busy, 1 = successfully executed, else error code (TBD).
12313 reserved (not implemented)	Data Table Transfer Register: R/W; writing a numeric to this register will cause the current data table to be saved to the corresponding script. Writing the numeric within the filename + 1000 will cause an existing file within the flash disk to be loaded into memory, becoming the current data table. For example: writing a 4 to this register will load Datatable004.datand its values will become the current table. (to be implemented in a future release)
12314	Flash Disk Selection Register: (R/W), 0 = root, 1 – n = drive mounted in sequence. Determines volume in which Flash Disk Space Register (12315) returns information. (5200 Only, 5100 = root).
12315	Flash Disk Space Register: (R), contains the approximate free space available on the flash disk.

Register Number	Description
12324	Script Line Result Register: R, contains the last error code that caused a foreground script to stop executing (Script #001 to 999). (5200 only).
12325	Log Selection Register: R/W, writing a numeric to this register will define the Log file name to be used, *Log###.log*. 1-999 for normal operation, 1001 to 1999 for background threads. (5222 only).
12326	Log String Transfer Register: R/W, write record number of 'log.ini' format file to reference and begin writing formatted record to *Log###.log* file. (5222 only).
12327	Log String Result Register: R, result of logging operation, 0 = success, -1 = busy. (5222 only).
12328	Log Deletion Register: R/W, write the numeric value of the *Log###.log* file to delete. (5222 only).
12329	Snap Execution Register: R/W, write the numeric value of the *Log###.log* file to rename to *Snap###.log*. (5222 Only).
12320	Snap Result Register: R, result of logging operation, 0 = success, 53 = failed, or not exist. (5222 only).
12331	Snap Deletion Register: R/W, write the numeric value of the *Snap###.log* file to delete. (5222 only).

Serial Communications Registers

12000	Select Controller Communications Port: W access, 1 = COM1, 2 = COM2, 3 – 7 = TCP raw virtual socket connections (see 22XX0 register descriptions).
12000	Message Transmission Status for Controllers: R access, 0 = not busy, 1 = busy.
12001	Transmit Message from Data Table: W only, Store row number to transmit.
12001-12255	Controller Receive Buffer Access, R only, 1 character per location.
12300	Protocol Variation: R/W, Controls RS-232 terminal protocol modes. 0 = computer, 1 = terminal (default)
12301	Serial Baud Rate Selection: R/W, 2 = 1200, 3=2400, 4 = 4800, 5 = 9600, 6 = 19.2K (default), 7 = 38.4K, 8 = 57.6K, 9 = 115.2K (57.6K/115.2K only supported on 5200 series).
12302	Serial Input Buffer Counter: (R) number of characters available. (W) any value to clear buffer and zero count.
12303	Disable Automatic Parsing: R/W, 0 = inhibits response, 1 = resumes normal response to incoming messages.
12304	Extract Number from RS-232 Receive Buffer: R only, Automatically assembles ASCII strings into a numeric value. The result is a signed 32-bit number. Automatically assembles strings of ASCII characters containing numeric information into a numeric value. Number multiplied by 10,000, allowing decimal points to 4 places.
12305	Communications Priority: R/W, when running multiple tasks. 0 = normal, 1 = priority.
12308	Serial Parity: R/W, 0=None (default), 1=Odd, 2= Even
12309	Serial Stop Bits: R/W, 1 (default) or 2
12310	Serial Data Bits: R/W, 7 or 8 (default)
12316	Message String Transfer Register: R/W, write records number of message.ini file to send out serial port selected in 12000 register, read returns status with 0 = success. See the Model 5200 Script Configuration Guide.
12320	Serial Active Protocol Selection: R/W; by default the protocol is set to CTC (0). Write to this port last after setting up any relevant parameters in other register, since this register enables the selected protocol immediately. (5100 only supports on COM1). CTC Binary & ASCII – 0 Modbus Master RTU – 1 (max of 120 16 bit Modbus Registers/block read; do not set manually, as it will be set when configuring the Modbus Master Register Control Block. Up to 256 may be read using automatic de-blocking feature of the Control Block) Modbus Master ASCII – 2 (max of 56 16 bit Modbus Registers/block read; do not set manually, as it will be set when configuring the Modbus Master Register Control Block. Up to 256 may be read using automatic de-blocking feature of the Control Block) Modbus Slave RTU – 3 (max of 120 16 bit Modbus Registers or 60 32 bit 5200 registers) Modbus Slave ASCII – 4 (max of 56 16 bit Modbus Registers or 28 32 bit 5200 registers)

Register Number	Description
12321	Serial Active Address: R/W, address to be used by the controller, based upon the enabled protocol. By default the Global Serial Address is used unless overridden by writing a different one for the enabled port (12000 register) to this register. Currently on Modbus Slave protocols use this address. Modbus Master uses the Modbus Master Register Control Block, 21000 – 21299.
12322	Global Serial Address: R/W, Address to be used as the 5200 power up default for Modbus Slave Serial Protocols unless overridden by a write to register 12321. To save this value permanently a 1 must be written to register 20096.

Ethernet Communications Registers

Register Number	Description
20000	CTNet Node Number: R/W, Controller's node number, valid numbers are 1 - 32767. Requires power cycle after change.
20007	Specify Connection Type: R, 1 = 10baseT.
20025-20028	SNTP Server IP Address: R/W, with 20025 being the first Octet, X.0.0.0, 20026 the second, 0.X.0.0, 20027 is the third, 0.0.X.0, and 20028 is the fourth, 0.0.0.X. Default is 192.43.244.18 (standard).
20041	SNTP Server Port: R/W, default is 123.
20042	SNTP Update Time: R/W; this register contains the number of seconds before the next synchronization request with the SNTP server. For example 3600 would be an hour, 86400 would be 24 hours. Default is 86400. When a change in time is made to this value it typically takes about 1 minute before the new value will take effect. Power cycling of the controller is not required.
20043	SNTP Offset from GMT: R/W, number of seconds to add or subtract from GMT, default is 0.
20048-51	Controller IP Address: R/W, with 20048 being the first Octet, X.0.0.0, 20049 the second, 0.X.0.0, 20050 is the third, 0.0.X.0, and 20051 is the fourth, 0.0.0.X. Set IP address to 0.0.0.0 to enable DHCP.
20064-67	Controller Subnet Mask: R/W, with 20064 being the first Octet, X.0.0.0, 20065 the second, 0.X.0.0, 20066 is the third, 0.0.X.0, and 20067 is the fourth, 0.0.0.X.
20080-83	Controller Gateway Address: R/W, with 20080 being the first Octet, X.0.0.0, 20081 the second, 0.X.0.0, 20082 is the third, 0.0.X.0, and 20083 is the fourth, 0.0.0.X.
20096	**Commit to Nonvolatile memory network settings:** (W) IP address, gateway, subnet, ctc node, sntp address, system name, password, etc…
20097	**Delete `5100.ini or _startup.ini` File:** W only. Writing a 1 to this register will cause the `5100.ini (5100) or _startup.ini (5200)` file residing in `/_system/Scripts` directory to be deleted. Typically used to recover from a file that sets Security parameters in such a way as to totally restrict access.
20102	**Millisecond Timer:** R only. Range is -2,147,483,648 to +2,147,483,647

Register Number	Description
21000-21299	**TCP Peer to Peer and Modbus Master Parameters:** R/W, starts at 21000 and repeated every 10 blocks as follows: **21XX0 – First Octet IP Address Register (Most Significant) – R/W** This is the first octet of the IP address (XXX.000.000.000) to connect to. **21XX1 – Second Octet IP Address Register – R/W** This is the second octet of the IP address (000.XXX.000.000) to connect to. **21XX2 – Third Octet IP Address Register – R/W** This is the third octet of the IP address (000.000.XXX.000) to connect to. **21XX3 – Fourth Octet IP Address Register (Least Significant) – R/W** This is the fourth octet of the IP address (000.000.000.XXX) to connect to. **21XX4 – Start Register – R/W** This register stores the starting register address that is to be read from the remote device. **21XX5 – Sequential Number Register – R/W** This register stores the number of sequential registers (starting with Register 21XX4) you want to read during a polling session. The value 1 represents a single register and the maximum number of registers allowed is 100 for Peer to Peer and 256 for Modbus. Configure this register before setting up any other registers. Do not change this value during a transaction or all data will be lost and new values will have to be entered. If you modify this register, it lets you reset the connection. All register reads from remote devices will be the same block size. For Modbus Master this register is the default; see 21XXX8, 1011 to change when polling differing devices. **21XX6 – Poll Timer Register – R/W** Set this register to 0 for a single read request else set the scan rate in milliseconds. The minimum value allowed is 50 ms for Peer to Peer and 10 ms for Modbus. You can write to this register at any time. **21XX7 – Status Flag Register – R** This register reflects the current status of the data registers. Its value is based on any requested operations. Typically, you initiate an operation and then wait for a status of 1. Possible values are: 0 - Offline; no connection is present. 1 - Last request is successful and completed. Data is available in the data registers if requested. Read or Write may now be done. -1 - Requested operation has failed; typically a Modbus Exception error -2 - Busy; connecting to the desired host. -3 - Busy; reading data. -4 - Busy; writing data. -5 – Timed out; poll timeout on a device by Modbus Master. -10 - Aborted operation; out of local memory or resources.

Register Number	Description
21000-21299 Cont'd	**21XX8 – Index Offset Register – R/W**

This register lets you access each of the requested sequential data registers. It works in conjunction with Register 21XX9 and acts as its array pointer. You can store the number of a general or special purpose register in 21XX8 and 21XX9 can then access the resource contained in the pointer. By default 0, this register points to the very first data element read from the remote device. This would be equivalent to what you set the Start Register to begin with (21XX4). Incrementing this register allows you to access other data elements, like an array. Register 21XX9 can then be read or written accordingly. The index register also has a few special features when you set it to 1000 or above.

1000 – Peer Request Time-Out Register – (R/W) The timer starts when a peer node request is initiated and stops (times out) if no response is received within the time specified by this register. Retries only occur if automatic updates are active (Register 21XX6 is set to a value other than 0). Defaults are 500 ms for single register reads and time-out value*2.5 for automatically updated register read transactions.

1001 – Peer Request Failed Index Register –(R) This register indicates when a peer transaction fails and an error occurs. The Status Flag Register (21XX7) is set to a value other than 1. Any data that was read or written when the error occurred has an offset value that is stored in 1001. If you read the data register, it returns the offset failure value. Data written before this offset value is valid. For example, if your process continuously updates 50 registers and the register returns a value of 25, it means the process failed while trying to write the 25th element of data. All data written before this element was written correctly.

1002 – Peer Request Retry Counter Index Register – (R/W) This debugging register points the data register to the retry counter. Quickstep can set this register to any value. The register is incremented by 1 when a time-out occurs because of waiting for data from a peer node.

1003 – Protocol Index Register – (R/W) This register tells the data register what protocol to use for setting the peer block registers. You must set this register before setting the Start Register (21XX4). Default mode is 0 for UDP Peer-to-Peer protocol. 2 is used for Modbus TCP Master mode, 3 for Modbus Master RTU Serial on COM1 (TBD).

1004 – TCP Client Port Index Register – (R/W) This register points the data register to the destination TCP Port address for your connection. You must set this register before setting the Start Register (21XX4). 1004 is currently used for Modbus TCP and serial Master mode with a default port number of 502 for TCP and 1 for serial (COM1).

1005 – Modbus Master Unit ID Index Register – (R/W) This register points the data register to the Device/Unit ID field value used in the Modbus Master request packet. The default ID is '1' but you can set it to any desired value. This ID affects all subsequent transmissions and allows multiplexed devices to be addressed in a Modbus environment.

1006 – Modbus Master Exception Index Register – (R) This register allows you to interrogate the last Modbus Exception error code received from the data register (21XX9). Referencing this register helps to interpret failure types. Typically you would reference this register if a '-1' appears as the current status in register 21XX7.

1007 – Register Remapping Start Index Register – (R/W) This option allows remote registers to be mapped into the 23000 to 24999 consecutive memory space. Previously an index register at 21XX8 needed to be set then data read from 21XX9. This can result in slow operation if a lot of data needs to be transferred. Setting 21XX8 to 1007 and then writing the register value from 23000 to 24999 will allow all data to be remapped to that register block area, consecutively, based upon the block size (21XX5). A write to the remapped area will result in a remote write. By default re-mapping is not active.

Register Number	Description
21000-21299 Cont'd	**1008 – Modbus Master MAX Retries Register** – (R/W) This register allows you to change the maximum number of retry attempts on a Unit ID before giving up. Default is 2. **1009 – Modbus Master Retry Counter Register** – (R/W) This register allows you to observe and change the current number of message retries to the current Unit ID. **1010 – Modbus Master Timeout Register** – (R/W) This register allows you to change the default Unit ID timeout from 50 milliseconds to that desired, in milliseconds. **1011 – Modbus Master Block Size Register** – (R/W) This register sets the number of Holding Registers to be accessed. Must be the same or smaller than the Sequential Number Register, defaults to the same. Used to access Unit ID's with varying block sizes when manually changing the Unit ID under program control. **1999 – Peer Request Initiate Write Block Index Register** – (R/W) This register writes a block of registers (beginning with the Start Register) to a destination at the other end of the connection. The number of registers written is defined by Register 21XX5. To write to a single register, set the index to the desired offset and then write to the data register. Do not attempt to access the data register during a block write operation. **2000 – 2099 Peer Request Write Block Index Registers** – (R/W) This register points the data register to a temporary storage array and makes 2000 equivalent to an index offset of 0. Because data is written locally, you can write multiple values before initiating the write sequence by setting this register to 1999. Automatic polled updates cannot overwrite data values until the register is set to a non-2XXX value. **21XX9 – Data Registers – R/W** This phantom register contains peer data that is read or written in a peer transaction. It is a "window" into a register array in the controller. The array size is set by Register 21XX5 and the offset is specified by Register 21XX8. Data integrity is indicated in Register 21XX7. Array data can be mapped to the 23000 block instead of using this register (21XX8, 1007).
21600 – 21999 (5100 ONLY)	**Virtual IO Registers – R/W** Starts at 21600 and repeated every 100 blocks as follows: 21600 – 21699 – Virtual I/O Block 1 (Note that the following blocks are available, but not supported at this time) 21600 – 21799 – Virtual I/O Block 2 21700 – 21899 – Virtual I/O Block 3 21800 – 21999 – Virtual I/O Block 4 **21X00 VIRTUALIO_IPA – R/W** First Octet of the IP dot address (XXX.000.000.000) to connect to. **21X01 VIRTUALIO_IPB – R/W** Second Octet of the IP dot address (000.XXX.000.000) to connect to. **21X02 VIRTUALIO_IPC – R/W** Third Octet of the IP dot address (000.000. XXX.000) to connect to. **21X03 VIRTUALIO_IPD – R/W** Fourth Octet of the IP dot address (000.000.000. XXX) to connect to.

Register Number

Description

21600 – 21999
Cont'd

21X04 VIRTUALIO_REMAP_STARTREG – R/W

This register defines where in the 23000 - 24999 block all the Virtual IO registers are to be viewed. There are 192 registers, the first 96 volatile, and the second block, 96 nonvolatile. This same block is also used for peer-to-peer register mapping also so be careful not to overlap areas. This register must be set. Typically the first connection would be set to 23000, the second to 23200, third to 23400, etc… Setting this value to 0, at initialization, will disable Virtual IO register updates, thereby decreasing network traffic.

21X05 VIRTUALIO_STATUS – R

-1, not initialized or never connected to a server.

-2, initialized only.

0, offline

1, connecting

2, getting module bus information

3, getting first complete set of system data

10, ready for operation

21X06 VIRTUALIO_WINDOWSIZE – R/W

This register may be used to reduce the number of registers that will appear in the 23000 block based upon the VIRTUALIO_REMAP_STARTREG setting. By default this value is 192, but can be made smaller. Note that registers 1 to 96 are the volatile registers and registers 97 to 192 are the non-volatile registers, starting at 501 in the remote controller. Setting this value to 0 at anytime will disable all register scanning updates.. To notify the server of the change a VIRTUALIO_REQUEST_SCANUPDATE command must be issued.

21X07 VIRTUALIO_COMMAND – R/W

1 - VIRTUALIO_COMMAND_CREATE_SERVER

First command initiated to create a server block prior to connection, once IP address and register viewing blocks are defined.

2 - VIRTUALIO_COMMAND_INITIATE_CONNECTION

This is the second command, used to actually connect to a Virtual IO server and add its IO to your controller. This command only needs to be initiated once,; retries are automatic.

The following commands are used to send the current mask configuration to the server, changing its high priority mask accordingly or setting new scan rates.

9 - VIRTUALIO_REQUEST_SCANUPDATE

Sets the scan rates of the server to that defined in the VIRTUALIO_SCANIO_RATE and VIRTUALIO_SCANREGS_RATE registers. No change is made to the server until this command is initiated.

10 - VIRTUALIO_REQUEST_MASK_DIGITALINPUT_UPDATE

This command sets the server to the high priority mask defined in the VIRTUALIO_DI_MASK block; there are 32 registers, representing 1024 bits or input points.

11 - VIRTUALIO_REQUEST_MASK_DIGITALOUTPUT_UPDATE

This command sets the server to the high priority mask defined in the VIRTUALIO_DO_MASK block; there are 32 registers, representing 1024 bits or input points.

Register Number | Description

21600 – 21999
Cont'd

21X07 VIRTUALIO_COMMAND – R/W (Cont'd)

12 - VIRTUALIO_REQUEST_MASK_REG_UPDATE

This command sets the server to the high priority mask defined in the VIRTUALIO_VOLATILE_REG_MASK block; there are 3 registers, representing 96 bits or input points.

13 - VIRTUALIO_REQUEST_MASK_NVREG_UPDATE

This command sets the server to the high priority mask defined in the VIRTUALIO_NONVOLATILE_REG_MASK block; there are 3 registers, representing 96 bits or input points.

14 - VIRTUALIO_REQUEST_MASK_FULL_UPDATE

This command sets the server to the high priority mask and scan rates defined in all the masks, all at once. Note that this is automatically done upon initial connection but this is a fast way to make massive changes, all at once, during operation.

21X08 VIRTUALIO_SCANIO_RATE – R/W

This register displays/sets the current rate at which all IO data points available in the server are transferred to the client, by default 100 milliseconds. Note that all Analog Inputs are transmitted at this rate and there is no high priority mask available for them. The VIRTUALIO_REQUEST_SCANUPDATE command must be initiated to activate any changes.

21X09 VIRTUALIO_SCANREGS_RATE – R/W

This register displays/sets the current rate at which all public data registers available in the server are transferred to the client, by default 100 milliseconds. The VIRTUALIO_REQUEST_SCANUPDATE command must be initiated to activate any changes.

21X10 VIRTUALIO_WRITEREG_INDEX – R/W

This register can be set to point to any valid register in the server. The data written to the VIRTUALIO_WRITEREG_VALUE register will be written to that register. This register is not limited to the public registers. Note that you can only write the register, not read it back. There are no limitations as to what this register is.

21X11 VIRTUALIO_WRITEREG_VALUE – R/W

This value written to this register will be written to the server register whose index is contained in the VIRTUALIO_WRITEREG_INDEX.

21X12 to 21X15 Reserved for future use

Note: The appropriate MASK command must be initiated for any of the following MASK register changes to take effect. Changes may be made in multiple registers and then a single command used to activate them all at once.

21X16 – 21X47 VIRTUALIO_DI_MASK – R/W

These 32 registers consist of the high priority digital input mask registers. To activate, set one of 1024 bits, representing an IO point in the remote controller. Bit 0 in the first register represents IO point 1. When a bit is set, any change of state will be sent immediately to the client. Note the more bits you set the higher the load on the controller given the additional network and monitoring traffic. It is advised that this be used for change of states that are important to know about quickly but that do not change that often.

Register Number	Description
21600 – 21999 Cont'd	21X48 – 21X79 VIRTUALIO_DO_MASK - R/W These 32 registers consist of the high priority digital output mask registers. To activate, set one of 1024 bits, representing an IO point in the remote controller. Bit 0 in the first register represents IO point 1. When a bit is set, any change of state will be sent immediately to the client. Note the more bits you set the higher the load on the controller given the additional network and monitoring traffic. It is advised that this be used for change of states that are important to know about quickly but that do not change that often. 21X80 – 21X82 VIRTUALIO_VOLATILE_REG_MASK - R/W These 3 registers consist of the high priority volatile register mask registers. To activate, set one of 96 bits, representing a volatile register point in the remote controller (register 1 to 96). Bit 0 in the first register represents remote register 1. When a bit is set, any change of state will be sent immediately to the client. Note the more bits you set the higher the load on the controller given the additional network and monitoring traffic. It is advised that this be used for change of states that are important to know about quickly but that do not change that often. 21X83 – 21X85 VIRTUALIO_NONVOLATILE_REG_MASK - R/W These 3 registers consist of the high priority non-volatile register mask registers. To activate, set one of 96 bits, representing a non-volatile register point in the remote controller (register 501 to 596). Bit 0 in the first register represents remote register 501. When a bit is set, any change of state will be sent immediately to the client. Note the more bits you set the higher the load on the controller given the additional network and monitoring traffic. It is advised that this be used for change of states that are important to know about quickly but that do not change that often.
22000-22049	TCP Raw Socket Session Parameters: R/W, starts at 22000 and repeated every 10 blocks as follows: 22XX0 – Serial port ID register, offset 0, range 3 – 7. 22XX1 – Client/Server register, offset 1, to initiate connection set to a 0, if controller is a server set to a 1. 22XX2 – Most significant octet of IP Address to connect to, IPA, offset 2 22XX3 – IP Address octet, IPB, offset 3 22XX4 – IP Address octet, IPC, offset 4 22XX5 – Least significant octet of IP Address to connect to, IPD, offset 5 22XX6 – Port to connect to (client) or listen on (server), offset 6 22XX7 – Connection status register, offset 7, on read, -1 = not initialized, 0 = offline, 1 = online, write a 1 to initiate connection or start server thread. 22XX8 – Index register to offset to data, offset 8. Recommend using serial port buffer, not this interface but available to mimic the peer to peer interface. 22XX9 – Data array, offset 9. Recommend using serial port buffer commands, not this interface but available to mimic the peer to peer interface.
23000-24999	Remapped Register Block – (R/W) Peer and Modbus data defined in registers 21000-21999 may be remapped to this consecutive data block starting at the register number designated in 21XX8, index 1007.

Diagnostic Registers

13030	Debug Slot Register: R/W
13031	Debug Stack Limit: R only
30000-31023	Debug Board Registers: R/W
13049	Boot Sequence Status Register: R only
13050	Total Tasks Active: R only
13051	Task Cycle Time: R only
13052	Task Cycle Minimum Time: R only
13053	Task Cycle Maximum time: R only

Special Functions

Pulse Width Modulated Outputs

5901, 5905	Pulse Output Count: R/W, Number of pulses to send out of outputs 1 and 2 respectively. Storing 65535 sends pulses continuously. Any other number of pulses stored here will count down as they are output. Note: Set this register last as it will initiate the pulse upon a non-zero value.
5902, 5906	Pulse Time Output: R only, Current time output has been outputting pulses in ms. For output 1 and 2 respectively.
5903, 5907	Pulse ON Time: R/W, PWM pulse "on time" in mSec. 1 mSec minimum. Set this register prior to the Pulse Output Count register.
5904, 5908	Pulse Period: R/W, PWM pulse period or interval in mSec. 2 mSec minimum. Set this register prior to the Pulse Output Count register.

Real-Time Clock

13002	Continuous millisecond counter: R/W, increments every 1 millisecond. Range is -2,147,483,648 to +2,147,483,647.
13013	RTC Lock Register: R/W, 0 = locked from write operations, 1 = unlocked, OK to write
13014	Seconds: R/W
13015	Minutes: R/W
13016	Hours: R/W, 24 hour clock
13017	Day of Month: R/W
13018	Month of Year: R/W, 1-12
13019	Year: R/W, Two fields
13020	Day of Week: R/W, 1-7, where Monday = 1

Tasks

13011	Task Priority: R/W, Specifies Super Task serviced on a priority basis. 1000 = Currently executing task
13012	Current Task Number: R only
13032	Task Register Fault Status: R only
13033	Task Register Fault Step: R only
13034	Task Register Fault Task: R only
13035	Task Register Fault Data: R only
13036	Performance Adjustment Register (PAR): R/W, Number of milliseconds till Quickstep opens network service window. 10 <= PAR <= 250 (smaller PAR > Network Performance)
13037	Network Service Window (NSW): R/W, 5 ms. X this value = number of ms. Network window allowed to be open. If no service is required, or is complete the window is closed immediately and Quickstep uses the extra time for its own execution. 2 <= NSW <= 14 (larger NSW > Network Performance).
13101-13164	Delay Timer Registers
13038	Fault Step Register – (R/W) Step to branch to when fault occurs. Write a 0 to disable.
13039	Fault Task Register – (R) Task number that is the active Fault Handler, 0 means none.
13040	Fault Mask Register – (R/W) Bit OR types of fault that will invoke the handler, by default all enabled (-1) when the Fault Step Register is written

Tasks (cont'd)

13041	Fault Clear Register – (W) Used to write the recovery state when done processing the Fault.

Program State	Description
1	RESET – Reset the controller only and then stop..
5	RESTART – Reset the controller and begin running again at step 1.
6	STOPPED – Stop the controller but do not reset.
8	RUNNING – Ignore the fault and continue running.
9	FAULT – Continue to fault as usual.
10	SHUTDOWN – Reset the controller and shutdown, requires a power cycle to exit.

High Speed Counters

5001	5102/5202 Internal High Speed Counter 1 (for Dual counters, only Counter for Quadrature mode): R/W
5002	5102/5202 Internal High Speed Counter 2 (for Dual counters, not used in Quadrature mode): R/W
5011	5102/5202 Internal High Speed Counter 1 Speed: R
5100	5102/5202 Internal High Speed Counter Mode, 0 = Quadrature, 1 = Dual Counters: R/W
5101	5102/5202 Internal High Speed Counter Frequency Period: R/W
5102	5102/5202 Internal High Speed Counter Frequency Value: R
5103	51/5202 Internal Registration 1 Start Value (Compare to counter 1): R/W
5104	5102/5202 Internal Registration 1 Window Size: R/W
5105	5102/5202 Internal Registration 1 Capture Value (from Counter 1): R
5106	5102/5202 Internal Registration 1 Status, 0 = Armed, 1 = Captured: R/W
5107	5102/5202 Internal Registration 2 Start Value (Compare to counter 1): R/W
5108	5102/5202 Internal Registration 2 Window Size : R/W
5109	5102/5202 Internal Registration 2 Capture Value (from Counter 1): R
5110	5102/5202 Internal Registration 2 Status, 0 = Armed, 1 = Captured: R/W

Miscellaneous Special Functions

5801-5808	Output Trigger Thresholds for Counters: R/W, Triggers an output when a counter reaches a specified value. Register 5801 is associated with counter 1 and output 1, register 5802 is associated with counter 2 and output 2, etc.
6500	Snapshot of Controller's Step Status: W only, Writing any value triggers snapshot.
6500	Number of Active Tasks: R only, Must write to this register before reading it. (See above.)
6501-6564	Step Number of Active Tasks: R only, Lists the step numbers of active tasks.
13001	Compile Option: R only
13003	Revision level of Firmware: R only, Multiplied x 100.
13004	Controller Architecture: R only, 1 indicates CTC's expanded architecture.
13007	Serial Number: R only
13008	Controller Model Code: R/W, Must be set to 3 to use CT Utilities. (DOS version)
13009	Automatically Turn Off Output at Software Fault: R/W, Storing an output number to this register and then turning that output ON in your program will cause that output to turn OFF in the event of any program software fault. This is commonly used to control a relay circuit that will drop out field power if a software fault occurs, for any reason.
13010	Analog Input Range: R/W, Storing 1 sets all analog inputs to 1 millivolt level.
29999	System Delay (R/W), causes processing to stop for value written milliseconds.

Motion Control Functions

Motion Registers Grouped by function then axis

<table>
<tr><td></td><td>The 5200 firmware is designed to access up to 16 axes. For the 14XXX register values below substitute the axis number for 'ax' to get the correct register. <b>Axis #1 = 1</b>; For example the position of axis #1 is stored in register 14001.</td></tr>
<tr><td>140ax</td><td><b>Position</b> (counts), R only</td></tr>
<tr><td>141ax</td><td><b>Error</b> (counts), R only</td></tr>
<tr><td>142ax</td><td><b>Velocity</b> (counts / sec), R only</td></tr>
<tr><td>143ax</td><td><b>Status</b>, R only:
<table>
<tr><th>Status</th><th>Description</th></tr>
<tr><td>0</td><td>Axis not initialized</td></tr>
<tr><td>1</td><td>Stopped and ready</td></tr>
<tr><td>2</td><td>Motion imminent: waiting for start</td></tr>
<tr><td>3</td><td>Accelerating</td></tr>
<tr><td>4</td><td>At max speed.</td></tr>
<tr><td>5</td><td>Decelerating to new max speed</td></tr>
<tr><td>6</td><td>Decelerating to stop</td></tr>
<tr><td>7</td><td>Soft stop</td></tr>
<tr><td>8</td><td>Registration move</td></tr>
<tr><td>9</td><td>Home</td></tr>
<tr><td>10</td><td>Following</td></tr>
<tr><td>128-255</td><td>Error</td></tr>
</table></td></tr>
<tr><td>144ax</td><td><b>Integral Error</b> (count-seconds), R only</td></tr>
<tr><td>145ax</td><td><b>Velocity Feedforward</b></td></tr>
<tr><td>146ax</td><td><b>Deceleration</b> (counts/sec^2)</td></tr>
<tr><td>147ax</td><td><b>Dedicated Inputs</b>, R only<b>:</b>
This is a bit map of the input signals
<table>
<tr><th>Bit Number</th><th>Description</th><th>Bit Number</th><th>Description</th></tr>
<tr><td>0 (lsb)</td><td>Reg.</td><td>4</td><td>Rev EOT</td></tr>
<tr><td>1</td><td>Home</td><td>5</td><td>Fwd EOT</td></tr>
<tr><td>2</td><td>Start</td><td>6</td><td>Z/Index</td></tr>
<tr><td>3</td><td>Kill</td><td>7</td><td>Not Used</td></tr>
</table></td></tr>
<tr><td>148ax</td><td><b>Acceleration Feedforward</b></td></tr>
<tr><td>149ax</td><td><b>Analog Output</b>, R only -32,767 = -10.000V; 32,767 = 10.000V</td></tr>
</table>

Motion Registers Grouped by axis then function

<table>
<tr><td></td><td>For the 15xxx, 16xxx, 17xxx register values below substitute the axis number for 'bx' to get the correct register. Axis #1 = 0; For example the position of axis #1 is stored in register 15000.</td></tr>
<tr><td>15bx0</td><td>Position (counts), R only</td></tr>
<tr><td>15bx1</td><td>Error (counts), R only</td></tr>
<tr><td>15bx2</td><td>Velocity (counts / sec), R only</td></tr>
<tr><td>15bx3</td><td>Status, R only:
<table>
<tr><th>Value</th><th>Description</th><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Axis not initialized</td><td>6</td><td>Decelerating to stop</td></tr>
<tr><td>1</td><td>Home</td><td>7</td><td>Soft stop</td></tr>
<tr><td>2</td><td>Motion imminent: waiting for start</td><td>8</td><td>Registration move</td></tr>
<tr><td>3</td><td>Accelerating</td><td>9</td><td>Home</td></tr>
<tr><td>4</td><td>At MAX speed</td><td>10</td><td>Following</td></tr>
<tr><td>5</td><td>Decelerating to new MAX speed</td><td>128-255</td><td>Error</td></tr>
</table></td></tr>
<tr><td>15bx4</td><td>Integral Error (count-seconds), R only</td></tr>
<tr><td>15bx5</td><td>Velocity Feedforward, also used to specify the Output in Direct mode</td></tr>
<tr><td>15bx6</td><td>Deceleration (counts/sec^2), R only</td></tr>
<tr><td>15bx7</td><td>Dedicated Inputs, R only:
This is a bit map of the input signals
<table>
<tr><th>Bit Number</th><th>Description</th><th>Bit Number</th><th>Description</th></tr>
<tr><td>0 (lsb)</td><td>Reg.</td><td>4</td><td>Rev EOT</td></tr>
<tr><td>1</td><td>Home</td><td>5</td><td>Fwd EOT</td></tr>
<tr><td>2</td><td>Start</td><td>6</td><td>Z/Index</td></tr>
<tr><td>3</td><td>Kill</td><td>7</td><td>Not Used</td></tr>
</table></td></tr>
<tr><td>15bx8</td><td>Acceleration Feedforward</td></tr>
<tr><td>15bx9</td><td>Analog Output, R only -32,767 = -10.000V; 32,767 = 10.000V</td></tr>
<tr><td>16bx0</td><td>Reg. Start, R/W – Position at which the registration will be enabled</td></tr>
<tr><td>16bx1</td><td>Reg. Window, R/W – The range that the registration will be enabled</td></tr>
<tr><td>16bx2</td><td>Reg. Position, R only – The position at which the registration was detected, when Reg status is 1</td></tr>
<tr><td>16bx3</td><td>Reg. Offset, R/W – The distance to be moved after the registration input</td></tr>
<tr><td>16bx4</td><td>Reg. Status – 0 = Armed, 1 = Detected, can only set to 0</td></tr>
<tr><td>16bx5</td><td>Numerator, R/W – For following the master axis</td></tr>
<tr><td>16bx6</td><td>Denominator, R/W – For following the master axis</td></tr>
<tr><td>16bx7</td><td>Leader Position, R only – Only valid when following a master axis</td></tr>
<tr><td>16bx8</td><td>Leader Velocity, R only – Only valid when following a master axis</td></tr>
<tr><td>16bx9</td><td>Reserved</td></tr>
<tr><td>17bx0</td><td>Firmware Revision, R only</td></tr>
</table>

<table>
<tr><td>17bx1</td><td><b>Filter & Mode, R/W:</b>
In Direct mode the Feedforward Velocity gain specifies the output value (0 to 32767) with a value of 32767 = 10V (sign depends on the Filter type).
<table>
<tr><th>Description</th><th colspan="2">Value</th></tr>
<tr><td>Lower 3 bits (0x07)
Filter type</td><td>0 or 3 = PID
1 = + Direct (CW)
4 = PAVff</td><td>
2 = - Direct (CCW)
5 = PAV</td></tr>
<tr><td>Bits 4 &amp; 5
Accel/Decel Type</td><td>0 = Linear
2 = Parabolic</td><td>1 = S Curve
3 = Inverse Parabolic</td></tr>
<tr><td>Bit 7 (0x80)</td><td colspan="2">0=Trajectory Following
1 (value 128) = Encoder Following</td></tr>
</table></td></tr>
<tr><td>17bx2</td><td><b>Input Polarity, R/W:</b>
This is a bit map that controls the active level of the input signals,. When the bit is 0 then the input is active when it is On; if the bit is 0 then the input is active Off.
<table>
<tr><th>Bit Number</th><th>Description</th><th>Bit Number</th><th>Description</th></tr>
<tr><td>0 (lsb)</td><td>Reg.</td><td>4</td><td>Rev EOT</td></tr>
<tr><td>1</td><td>Home</td><td>5</td><td>Fwd EOT</td></tr>
<tr><td>2</td><td>Start</td><td>6</td><td>Z/Index</td></tr>
<tr><td>3</td><td>Kill</td><td>7</td><td>Not Used</td></tr>
</table></td></tr>
<tr><td>17bx3</td><td><b>Home Direction, R/W</b>
<table>
<tr><th>Direction</th><th>Description</th></tr>
<tr><td>CCW</td><td>0 or –1 = Home & Index
-2 = Home Only
-3 = Index Only</td></tr>
<tr><td>CW</td><td>1 = Home & Index
2 = Home Only
3 = Index Only</td></tr>
</table></td></tr>
<tr><td>17bx4</td><td><b>Options,</b> R only</td></tr>
<tr><td>17bx5</td><td><b>Reserved</b></td></tr>
<tr><td>17bx6</td><td><b>Maximum Following Error, R/W</b> default = 30000</td></tr>
<tr><td>17bx7</td><td><b>Speed Limit, R/W</b> – overrides maximum velocity, default = 4194303 steps/sec</td></tr>
<tr><td>17bx8</td><td><b>Maximum Position, R/W</b> – Used as a Software EOT when it is larger than the Minimum Position</td></tr>
<tr><td>17bx9</td><td><b>Minimum Position, R/W</b> – Used as a Software EOT when it is smaller than the Maximum Position</td></tr>
</table>

Extended Motion Registers Grouped by axis then function

	For the extended motion registers below (15xxx, 16xxx, 17xxx), substitute the axis number for **'cx'** to get the correct register. In this series Axis **#1 =16**; for example the Start Input of axis #1 is stored in register 15160.
15cx0	**Start Input, R/W** – System input used as Start Input, 0 = disabled, 1-16 specifies input number
15cx1	**Shutdown Input, R/W** – System input used as Shutdown Input, 0 = disabled, 1-16 specifies input number
15cx2	**Home Input, R/W** – System input used as Home Input, 0 = disabled, 1-16 specifies input number
15cx3	**Positive (CW) EOT Input, R/W** – System input used as CW EOT Input, 0 = disabled, 1-16 specifies input number
15cx4	**Negative (CCW) EOT Input, R/W** – System input used as CCW EOT Input, 0 = disabled, 1-16 specifies input number
15cx5	--------- Reserved ------------
15cx6	--------- Reserved ------------
15cx7	--------- Reserved ------------
15cx8	--------- Reserved ------------
15cx9	**Master Axis Select**
16cx0	--------- Reserved ------------
16cx1	--------- Reserved ------------
16cx2	**Index Position,** R only – Latched when using Index input (5V level signal) as a Registration input
16cx3	--------- Reserved ------------
16cx4	**Index Status,** can only write 0 – When using Index input (5V level signal) as a Registration input
16cx5	--------- Reserved ------------
16cx6	--------- Reserved ------------
16cx7	--------- Reserved ------------
16cx8	--------- Reserved ------------
16cx9	**Gain Scaling,** default = 8. Used when converting PID gains from the PROFILE command to the internal values. Increasing by 1 allows the PROFILE command values to be divided by 2 without changing the actual performance. Suggested when the values used in the command are large (near 255) and need to be increased to get acceptable performance. Decreasing by 1 allows the PROFILE command values to be doubled without affecting performance. Suggested when the values used in the PROFILE command are small and don't allow fine enough control to get acceptable performance.
17cx0	--------- Reserved ------------
17cx1	--------- Reserved ------------
17cx2	--------- Reserved ------------
17cx3	--------- Reserved ------------
17cx4	--------- Reserved ------------
17cx5	--------- Reserved ------------
17cx6	--------- Reserved ------------
17cx7	--------- Reserved ------------
17cx8	**Module Serial Number**
17cx9	**Module Monitor Version**

Notes

1. R indicates read, W indicates write.
2. Default settings are in **bold** print.

The information in this document is subject to change without notice. Any software described in this document is provided under license agreement and may be used or copied only in accordance with the terms of the license agreement.

The information, drawings, and illustrations contained herein are the property of Control Technology Corporation. No part of this manual may be reproduced or distributed by any means, electronic or mechanical, for any purpose other than the purchaser’s personal use, without the express written consent of Control Technology Corporation.