MotionWorks IEC Configuration Manual - Rev. B: 5/28/2008

Table Of Contents

1. MotionWorks IEC Configuration

1.1 MotionWorks IEC Configuration Overview

MotionWorks IEC Configuration provides a means of setting hardware and communication information with which the application program operates. This information consists of items such as: MECHATROLINK Axis and Remote I/O configuration, parameters for devices on the Ethernet network, and Option Slot configuration. For each of these elements, the Configuration software automatically enters variable groups and default names in the Global Data Definition for use with the application program.

MotionWorks IEC Configuration also provides a graphical motor tuning interface, which allows the user to change parameters in the ServoPacks in real time and view motor response.

1.2 Accessing the Configuration

Open a project before launching the Configuration, otherwise the Open Project dialog box will appear so a project can be selected.

Click the Icon on the toolbar  to launch the Configuration.

1.3 Closing the Configuration

The Configuration will automatically close when the programming environment is closed, or if another project is opened. It is not necessary to close the Configuration while in the MotionWorks IEC programming environment.

1.4 Online vs. Offline

When the Configuration is offline, all data displayed, modified and saved is written to XML files in the project directory.

Upon connection with a controller, a search for configuration data previously stored by MotionWorks IEC Configuration is performed.

If no configuration is found in controller (factory default):

The auto discovered hardware is compared to the offline hardware configuration. If the hardware matches, the parameters for each component are compared. If the parameters match, the controller's configuration is displayed. If the configurations are different at either the hardware or parameter level, a side-by-side comparison will be displayed. The user must select one of the two configurations. When the Save function is executed, the user configuration is stored in the controller.

If a previously stored configuration is found:

The previously stored hardware configuration is compared to the offline hardware. If the hardware matches, the parameters for each component are compared. If the parameters match, the controller's configuration is displayed. If the configurations are different at either the hardware or parameter level, a side-by-side comparison will be displayed. The user must select one of the two configurations. A backup copy of the ServoPack parameters is stored in the controller.

1.5 Connecting to the Controller

The project IP address is located under the TCP/IP Settings in the Configuration Tree. The IP Address is cross-linked with the IP Address in the Hardware Tab of the Project. All programming tools can communicate simultaneously with the controller (MotionWorks IEC, Configuration, and web server.) A color-coded indicator in the upper right corner of the window indicates the connection status with a red or green background and displaying the text “Online” or “Offline.”

Setup Information for remote connections via router or firewall

MotionWorksIEC uses the following Ethernet ports to communicate with the controller hardware. Certain firewall implementations may block these ports, and prohibit communication. An easy way to test for this is a network Ping, or if communication is possible via the web server, but not MotionWorks IEC.

Configuration: TCP Port	4040
Ethernet/IP	2222
Modbus/TCP	502
MotionWorks IEC Debug Mode	20547
MotionWorks IEC Program Download	20547
OPC Server	20547
Web Server	80

1.6 Saving Configuration Data

Configuration files are stored in a sub directory of the application’s project directory when the SAVE operation is invoked. When online with the controller, the save function also downloads parameters to the controller and all Mechatrolink ServoPack devices.

A red status messages on certain parameters will alert the user when power cycle is necessary for updated values to become effective.

The Save operation saves all configuration for all axes and every other component no matter what component in the tree is currently open.

1.7 Saving While Online

When the save operation is performed, XML data files are written to the project directory on the computer and controller. Servopack Parameters are written to FLASH memory. Notice the red text indicators will inform the user when a parameter requires power cycle to become effective.

The Save operation saves all configuration for all axes and every other component no matter what component in the tree is currently open.

1.8 MECHATROLINK Configuration

1.8.1 Adding a Servo Axis

There are a couple ways to add axes to the configuration.

Offline:

1) Right click on the Mechatrolink-II item in the configuration tree.

2) Select Add Device

3) Select a ServoPack model.

4) Enter a hardware node number. This must match the rotary switch of the Mechatrolink-II address on the device. Each device must have a unique hardware and logical address.

Online:

If Self Configuration is selected, the configuration will be automatically loaded into the configuration tree if the auto discovered is selected.

ServoPack configuration is divided into the following areas:

Limits	Test Move	Hardware
Configuration	Function	Alarm
I/O	Absolute	Brake
Tuning	Encoder	Dual Encoder

Note: Do not to delete automatically created variables or groups. If you must delete a group, be sure to delete the group header (an all of its variables) and the group name as listed under IO_Configuration in the Hardware tab. These two items must remain in sync for project to compile properly

1.8.2 Limits

Set the Position (Software limits), Torque, and Speed limits for the application.

1.8.3 Setting User Units

When one of the servo axes is selected on the configuration tree, click on the Configuration tab to set the user unit parameters. Changes to user units will only take effect after a power cycle.

Warning: If user unit parameters are changed after MC_SetPosition has been executed to store an absolute encoder position offset, the position value will be incorrect after power cycle. Use MC_SetPosition again to calibrate the axis after changing the User Unit parameters.

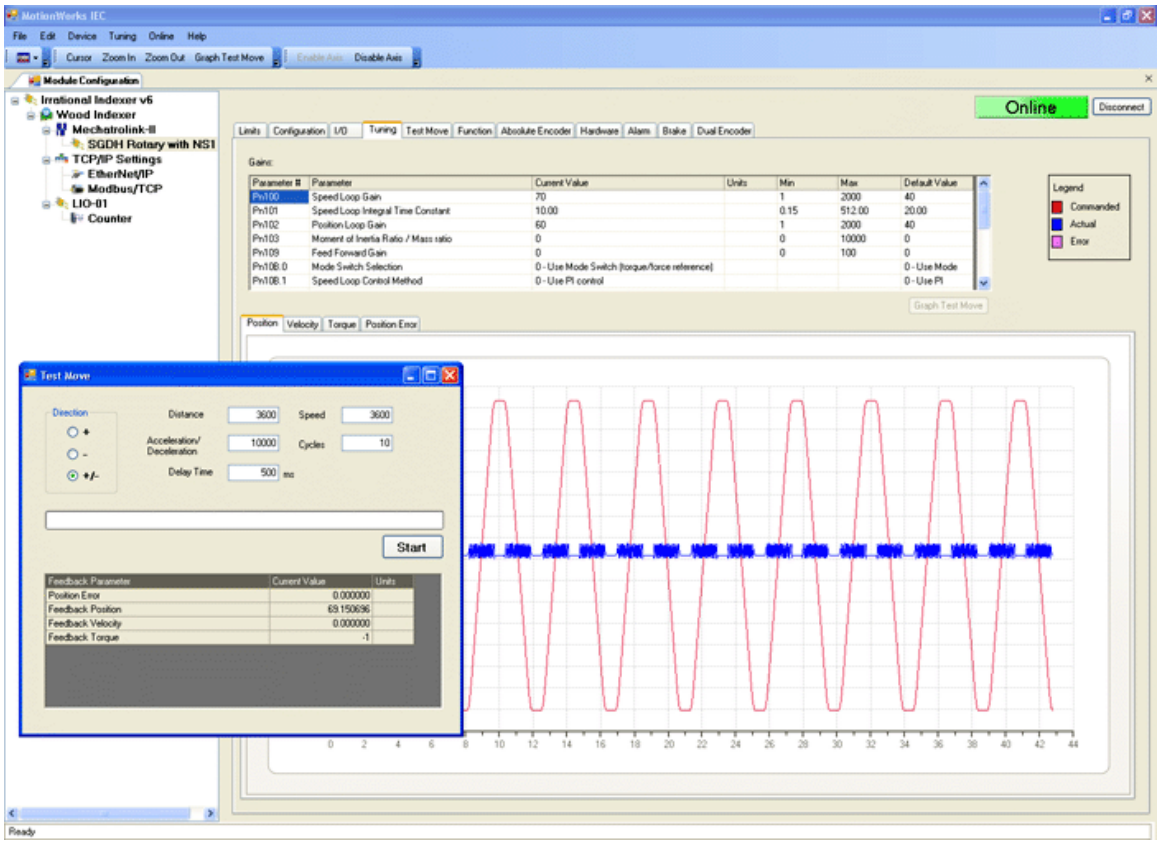
1.8.4 Servo Tuning

Please refer to these documents related to tuning on www.yaskawa.com:

> What should the inertia ratio parameter, Pn103, be set to in the Sigma II amplifier when the inertia ratio is not known?

1.8.5 Performing a Test Move

The tuning tab provides access to the ServoPack's tuning parameters with changes taking effect in real time. All basic and advanced tuning parameters are listed. Warning: Be sure to safeguard the machine during the Test Move operation! Use a hardwired E-Stop function in case of unexpected operation.



Direction

Select from forward only, reverse only, and forward & reverse motion.

Distance	User Units
Accel/Decel	User Units/s^2
Delay Time	ms
Speed	User Units/s
Cycles	Quantity

1.8.6 Servopack Alarms

A tab is provided to show alarm history. The alarms displayed here are the same alarms available from the MC_ReadAxisAlarm function block. For more information regarding Servopack alarms, refer to the following manuals:

Sigma II with NS115: SIEP C710800 01, see section 9.3

Sigma III: YEA-SIA-S800-11, see section 10.1.4

Sigma V with rotary motor: SIEPS8000043, see Section 6.1

Sigma V with linear motor: SIEPS8000044, see Section 6.1

1.8.7 Remote I/O Devices

Global variables for the following remote I/O devices with a fixed number of I/O will automatically be entered in the Global Variables window of the IEC Programming Environment when the user presses 'Save' in the configuration:

LIO-01 LIO-02 LIO-04 LIO-05 LIO-06 CNTR AO-01

AI-01

Note: Do not to delete automatically created variables or groups. If you must delete a group, be sure to delete the group header (an all of its variables) and the group name as listed under IO_Configuration in the Hardware tab. These two items must remain in sync for project to compile properly

1.9 Ethernet Connections

1.9.1 Ethernet Connections Overview

The controller can operate as an EtherNet/IP scanner and adapter, a Modbus/TCP master and slave, and deliver OPC data simultaneously. The "New Project" template will automatically create communication drivers and allocate global memory for external devices to read & write information in the controller. These memory areas are generic: no variables are automatically created, the user can decide on the arrangement and type of data to populate within the communications memory region.

Name	Type	Usage	Description
⊞ System			
⊟ EIP Input Instance #111, Qty: 128 Bytes, Address Range: %IB0-%IB127			
⊟ EIP Input Instance #112, Qty: 256 Bytes, Address Range: %IB128 - %IB383			
⊟ EIP Input Instance #113, Qty: 128 Bytes, Address Range: %IB384 - %IB511			
⊟ EIP Input Instance #114, Qty: 256 Bytes, Address Range: %IB512 - %QI767			
⊟ EIP Input Instance #115, Qty: 128 Bytes, Address Range: %IB768 - %IB895			
⊟ EIP Input Instance #116, Qty: 256 Bytes, Address Range: %IB896 - %IB1151			
⊟ EIP Output Instance #101, Qty: 128 Bytes, Address Range: %QB0 - %QB127			
⊟ EIP Output Instance #102, Qty: 256 Bytes, Address Range: %QB128 - %QB383			
⊟ EIP Output Instance #103, Qty: 128 Bytes, Address Range: %QB384 - %QB511			
⊟ EIP Output Instance #104, Qty: 256 Bytes, Address Range: %QB512 - %QB767			
⊟ EIP Output Instance #105, Qty: 128 Bytes, Address Range: %QB768 - %QB895			
⊟ EIP Output Instance #106, Qty: 256 Bytes, Address Range: %QB896 - %QB1151			
⊟ Modbus FC#02 Qty: 128 Inputs, Address Range: %QX1152.0 - %QX1167.7			
⊟ Modbus FC#04 Qty: 1024 Registers, Address Range: %QB1168 - %QB3215			
⊟ Modbus FC#05 Qty: 128 Coils, Address Range: %IX1152.0 - %IX1167.7			
⊟ Modbus FC#06,16 Qty: 1024 Registers, Address Range: %IB1168 - %IB3215			

Figure 1: Global Variable groups created when selecting the New Project template.

1.9.2. Using an HMI (Master) to Communicate to the Controller Via Modbus/TCP

The New Project template includes global I/O groups and drivers to allow the controller to respond to incoming Modbus messages. Figure 2 shows the Modbus memory map, and how it relates to the Global Variables in MotionWorks IEC. Note that function codes 01 and 03 return data that was sent to the controller from the master and do not reflect data from the Global Variables in the application program.

- The Modbus data memory is copied to the Global Variables at the task update rate.
- Modbus coil 0 equates to the Global Variable at %IX1152. 128 coils are available.
- Modbus register 40000 equates to the Global Variable at %IW1168. 1024 registers are available.
- Modbus input 10000 equates to the Global Variable at %QX1152. 128 inputs are available.
- Modbus register 30000 equates to the Global Variable at %QW1168. 1024 registers are available.

As shown in Figure 2 below, the input and output memory is in a different location even though they have the same numerical addressing.

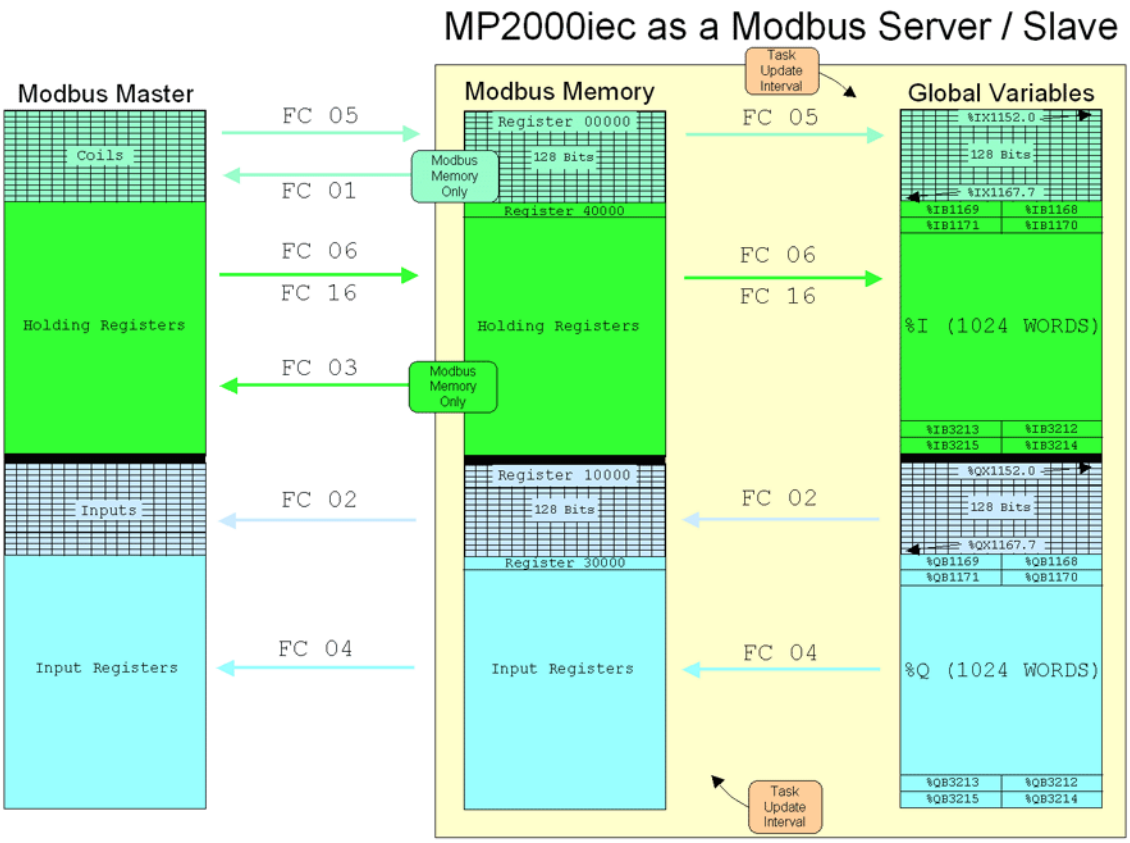


Figure 2: Memory map for Modbus data when the controller is a server / slave.

1.9.3 Adding a Modbus Server/Slave Device

The MP2000iec controller can communicate with up to 10 unique IP addresses simultaneously using Modbus TCP. A maximum of 20 blocks of memory or various function codes can be configured for each device if necessary.

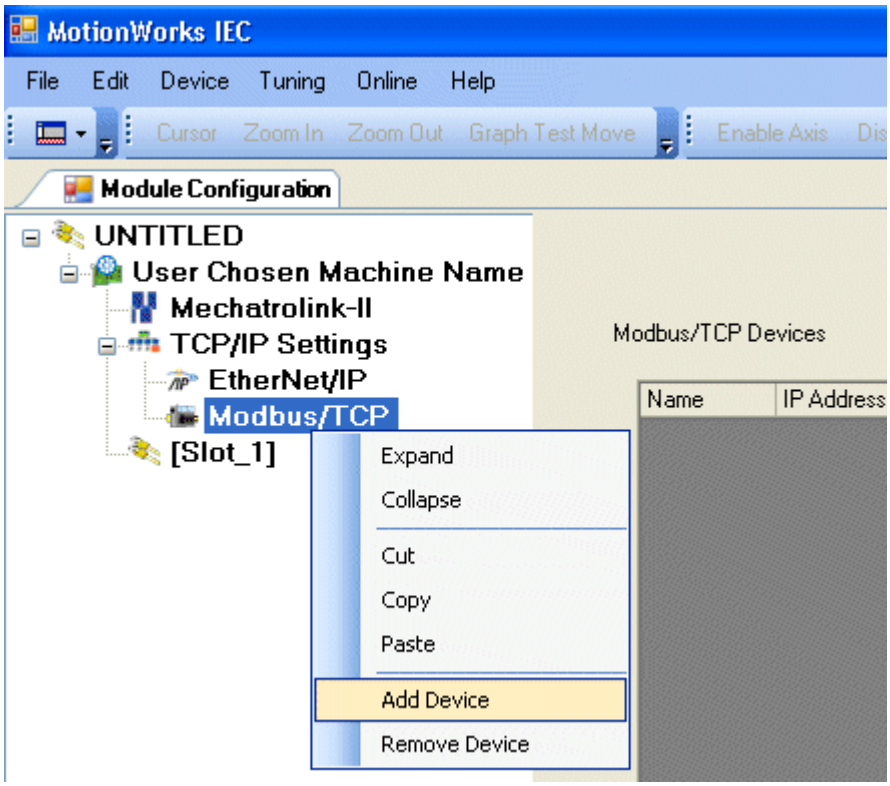
Supported Modbus Function Codes:

Function Code	Description
1	Read Coils
2	Read Inputs
3	Read Holding Registers
4	Read Input Registers
5	Write Single Coil
6	Write Single Register
16	Write Multiple Registers

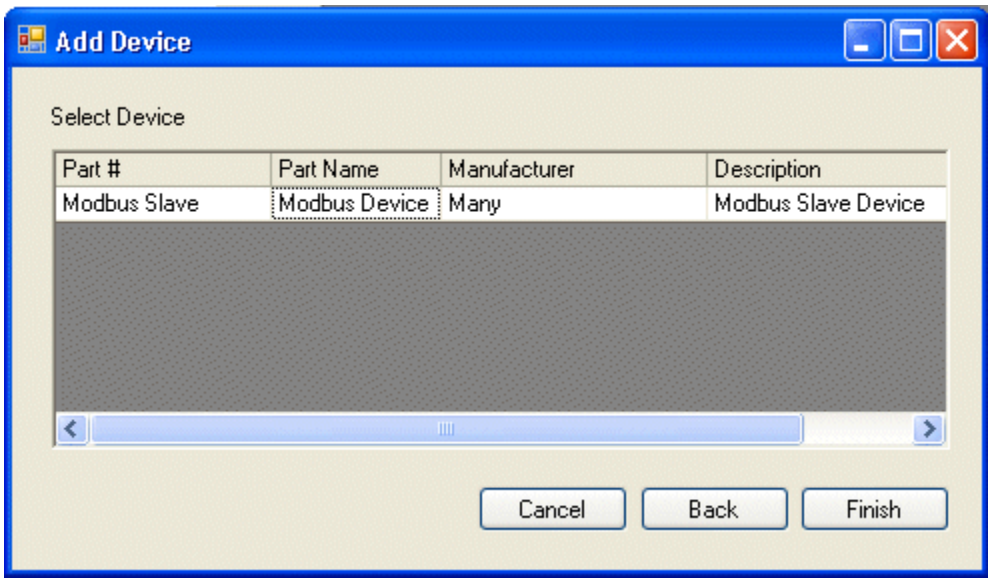
Modbus Server devices must be added to the configuration offline, then sent to the controller. Before the new configuration will become effective, power on the controller must be cycled.

Step 1: Launch the Configuration 

Step 2: Right click on the Configuration tree on the Modbus/TCP item



Step 3: Click Finish on the Add Device Dialog Box.



Step 4: The Add Modbus Device window is shown below in Figure 3. Select a name for the device to be added. This name will be displayed in the Global Variables list. The Status Variable will be automatically entered in the Global Variables section of the Development Environment. The Status Variable data type is WORD. This variable will indicate the status of the connection. See Figure 7 for details regarding the status variable. The minimum update time is 20 mSec.

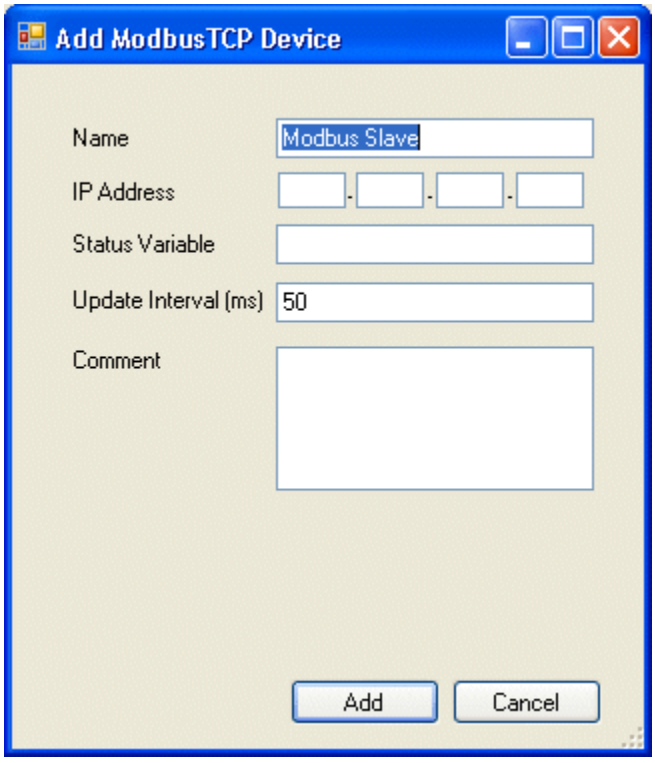


Figure 3: Add Modbus Device

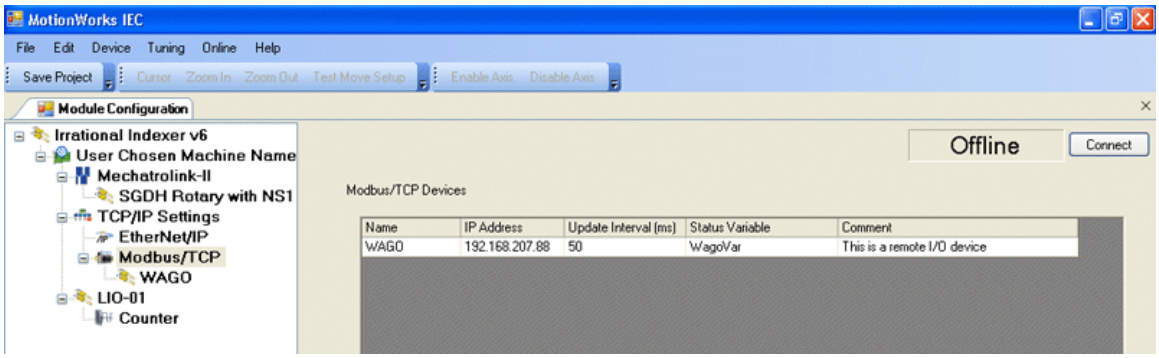
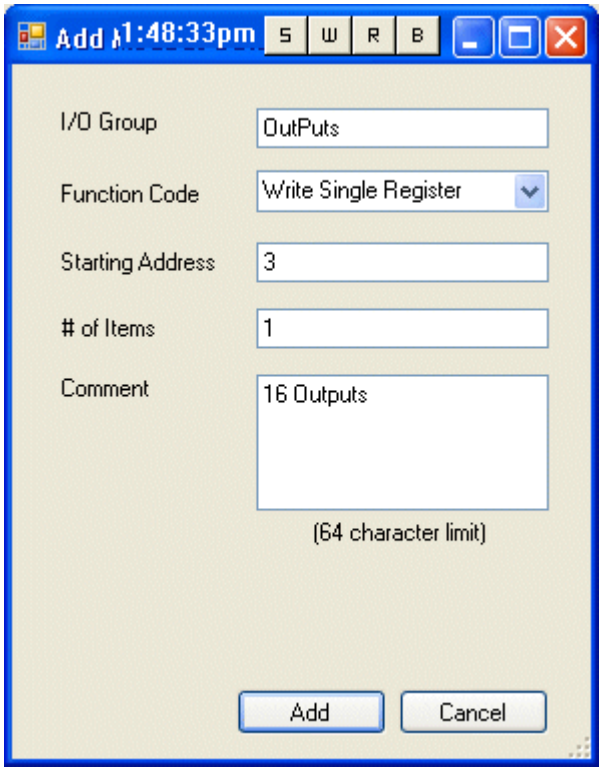


Figure 4: Configuration as shown with one Modbus server / slave configured.

Step 5: Click on the new Modbus Device in the Configuration tree at the left of the screen. Locate the hyperlink at the lower right of the screen to Add Data Blocks to this device. Select a name to be associated with the function code. For example, if the device is remote I/O, name the I/O Group "Output" for example. There is a seven-character limitation on the I/O Group name. The Starting Address is dependant on the remote device. Consult the documentation for the remote device to understand the register offset required to access its functions.



Step 6: When finished entering data blocks, Save the Configuration. This will create the global memory I/O Group in the Development Environment.

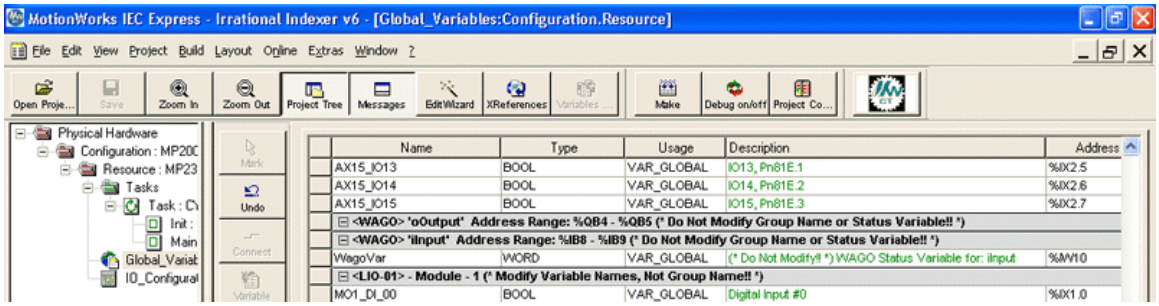


Figure 5: View of the Global Variables list. Click the Hardware tab in the Project Tree or use the 'View' menu to access. Note the status variable has been created under an input section for this Modbus device even though no input-type function codes were configured.

Step 7: Open the Global Variables list. Right click on the I/O group header to “Insert Variable.” This variable can either be a BOOL, WORD, or any other data type that fits the usage within the program. For example, 16 outputs can be defined as one WORD, and in the program, the bits can be accessed as follows: MYWORD.X3. Another example would be 16 individual BOOL variables with unique names.

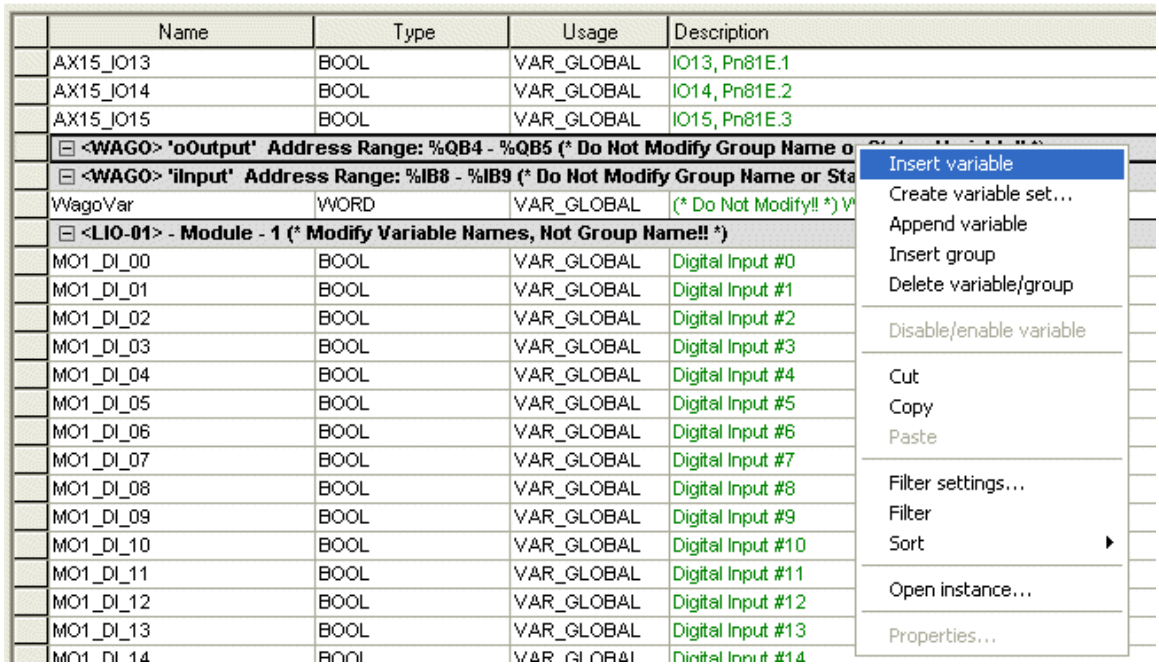


Figure 6: Inserting a variable into the Modbus group.

The memory area for this Modbus device is shown in the Group Heading. In Figure 5, bytes %B1 and %QB2 are allocated for the Modbus output register. Enter the proper memory address for the Modbus memory. If bits are required, enter %IX1.0 for example, or if creating a WORD, enter %IW1.

Any IEC data type can be transmitted or requested from the remote device, provided the data is interpreted as the same type on the other side.

To determine the memory area allocated for the Modbus connection, look in the MotionWorks IEC’s IO_Configuration window in the hardware section.

Modbus TCP Status Variable

This variable reflects the connection status. It can be used in the application program to determine if the connection to the remote device is active and the data is valid. The status WORD can be compared to 16#1000, which means that the connection is good. If bits 0 or 1 are on, the connection is not active, and the controller is attempting to reconnect to the remote device.

The status variable is only available when the MP2000iec controller is the client or master of remote devices.

NOTE: If the status variable has a value of zero as observed in the Global Variables list, the controller may not be running the application program.

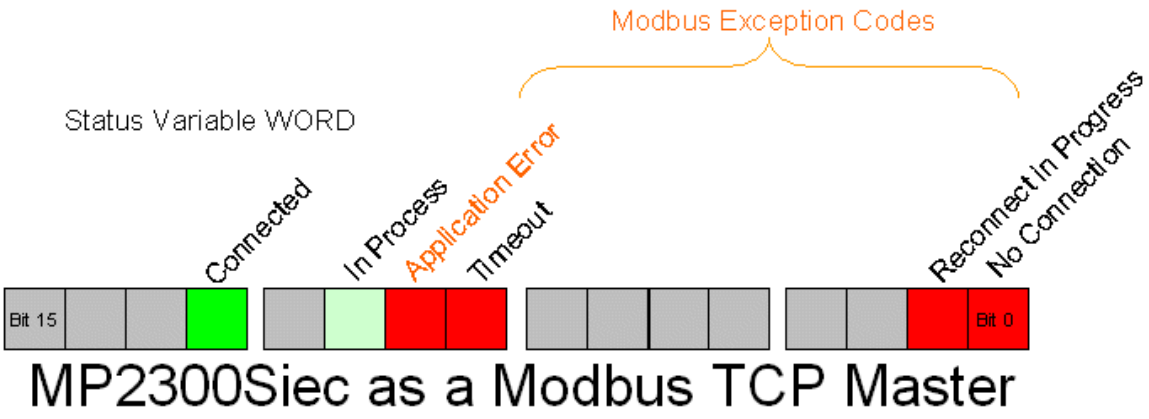


Figure 7: Modbus status WORD

Modbus errors in the lower byte of the status variable ONLY if the Application Error bit is TRUE.

Modbus Exception Codes		
Code (In lower byte of the status variable)	Name	Meaning
01	Illegal Function	The function code received in the query is not an allowable action for the server (or slave). This may be because the function code is only applicable to newer devices, and was not implemented in the unit selected. It could also indicate that the server (or slave) is in the wrong state to process a request of this type, for example because it is not configured and is being asked to return register values.
02	Illegal Data Address	The data address received in the query is not an allowable address for the server (or slave). More specifically, the combination of reference number and transfer length is invalid. For a controller with 100 registers, the PDU addresses the first register as 0, and the last one as 99. If a request is submitted with a starting register address of 96 and a quantity of registers of 4, then this request will successfully operate (address-wise at least) on registers 96, 97, 98, and 99. If a request is submitted with a starting register address of 96 and a quantity of registers of 5, then this request will fail with Exception Code 0x02 “Illegal Data Address” since it attempts to operate on registers 96, 97, 98, 99 and 100, and there is no register with address 100.

03	Illegal Data Value	A value contained in the query data field is not an allowable value for server (or slave). This indicates a fault in the structure of the remainder of a complex request, such as that the implied length is incorrect. It specifically does NOT mean that a data item submitted for storage in a register has a value outside the expectation of the application program, since the MODBUS protocol is unaware of the significance of any particular value of any particular register.
04	Slave Device Failure	An unrecoverable error occurred while the server (or slave) was attempting to perform the requested action.
05	Acknowledge	Specialized use in conjunction with programming commands. The server (or slave) has accepted the request and is processing it, but a long duration of time will be required to do so. This response is returned to prevent a timeout error from occurring in the client (or master). The client or master) can next issue a Poll Program Complete message to determine if processing is completed.
06	Slave Device Busy	Specialized use in conjunction with programming commands. The server (or slave) is engaged in processing a long–duration program command. The client (or master) should retransmit the message later when the server (or slave) is free.
08	Memory Parity Error	Specialized use in conjunction with function codes 20 and 21 and reference type 6, to indicate that the extended file area failed to pass a consistency check. The server (or slave) attempted to read record file, but detected a parity error in the memory. The client (or master) can retry the request, but service may be required on the server (or slave) device.
0A	Gateway Path Unavailable	Specialized use in conjunction with gateways. It indicates that the gateway was unable to allocate an internal communication path from the input port to the output port for processing the request. Usually means that the gateway is misconfigured or overloaded.
0B	Gateway Target Device Failed to Respond	Specialized use in conjunction with gateways. It indicates that no response was obtained from the target device. Usually means that the device is not present on the network.

Note: Do not to delete automatically created variables or groups. If you must delete a group, be sure to delete the group header (an all of its variables) and the group name as listed under IO_Configuration in the Hardware tab. These two items must remain in sync for project to compile properly

1.9.4 Using an EtherNet/IP Scanner to Communicate to the MP2000iec Controller as an Adapter

As previously mentioned, the controller will be configured to accept select EtherNet/IP instances when the New Project template is selected. Six pre-configured Instances are defined in the controller for input and output. The following diagram shows these instances and their Global Variable mapping.

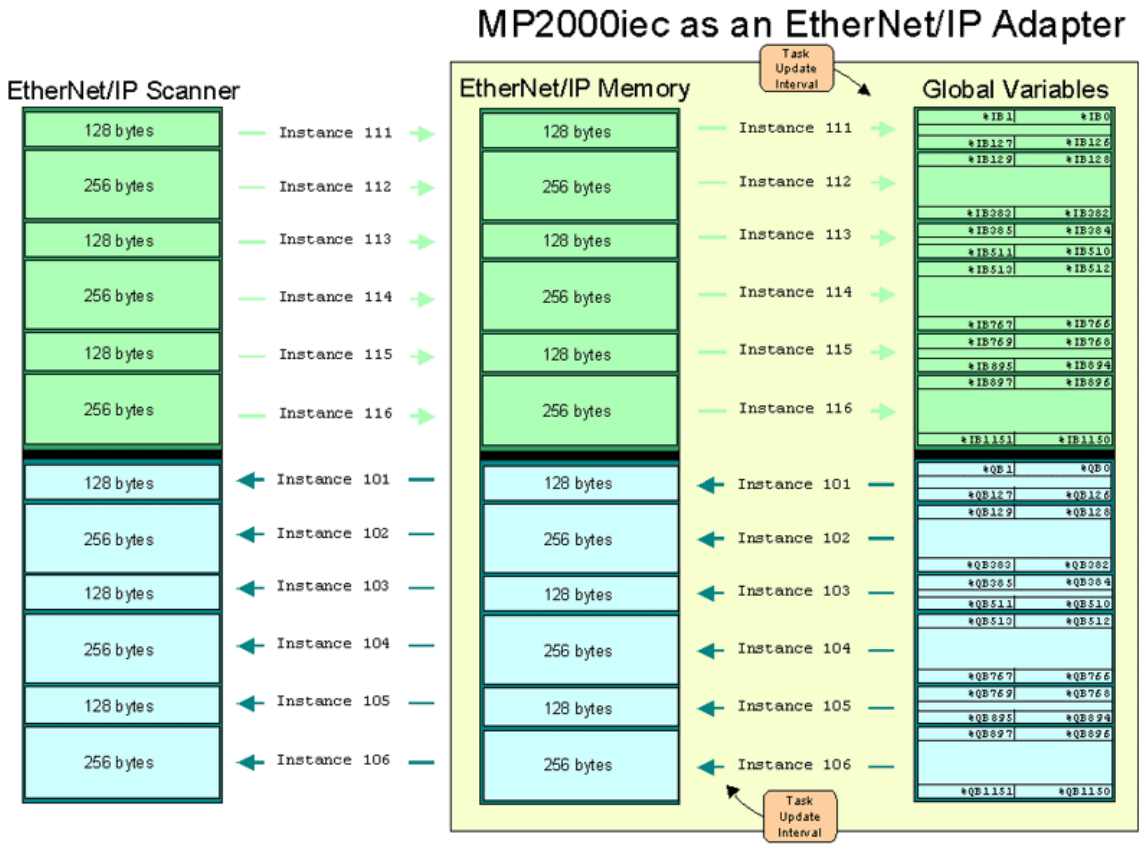
Note: when communicating to the MP2000iec controller to the available instances, the exact byte size of the instance must be configured on the Scanner side. (The entire 128-byte or 256-byte block must be transferred even if less data is required.)

The controller will automatically respond to incoming EtherNet/IP messages from Scanners requesting to read or write data into the pre-defined instances. No other instance numbers can be used.

Note: The scanner must configure both an input and output assembly. If the scanner is not required to receive any dat from the MP2000 controller, use assembly 128 with a size of zero.

Name	Type	Usage	Description
⊞ System			
⊟ EIP Input Instance #111, Qty: 128 Bytes, Address Range: %IB0-%IB127			
⊟ EIP Input Instance #112, Qty: 256 Bytes, Address Range: %IB128 - %IB383			
⊟ EIP Input Instance #113, Qty: 128 Bytes, Address Range: %IB384 - %IB511			
⊟ EIP Input Instance #114, Qty: 256 Bytes, Address Range: %IB512 - %QI767			
⊟ EIP Input Instance #115, Qty: 128 Bytes, Address Range: %IB768 - %IB895			
⊟ EIP Input Instance #116, Qty: 256 Bytes, Address Range: %IB896 - %IB1151			
⊟ EIP Output Instance #101, Qty: 128 Bytes, Address Range: %QB0 - %QB127			
⊟ EIP Output Instance #102, Qty: 256 Bytes, Address Range: %QB128 - %QB383			
⊟ EIP Output Instance #103, Qty: 128 Bytes, Address Range: %QB384 - %QB511			
⊟ EIP Output Instance #104, Qty: 256 Bytes, Address Range: %QB512 - %QB767			
⊟ EIP Output Instance #105, Qty: 128 Bytes, Address Range: %QB768 - %QB895			
⊟ EIP Output Instance #106, Qty: 256 Bytes, Address Range: %QB896 - %QB1151			
⊟ Modbus FC#02 Qty: 128 Inputs, Address Range: %QX1152.0 - %QX1167.7			
⊟ Modbus FC#04 Qty: 1024 Registers, Address Range: %QB1168 - %QB3215			
⊟ Modbus FC#05 Qty: 128 Coils, Address Range: %IX1152.0 - %IX1167.7			
⊟ Modbus FC#06,16 Qty: 1024 Registers, Address Range: %IB1168 - %IB3215			

Figure 8: Global Variable Groups created when selecting the "New Project" template



1.9.5 Adding an Ethernet/IP Adapter

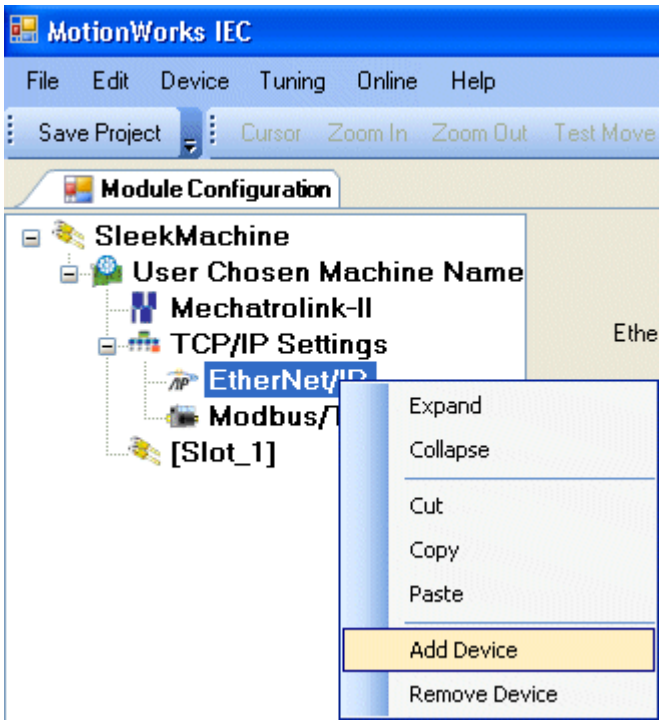
Configuring the controller to communicate with an EtherNet/IP adapter device is quite similar to configuring a Modbus server or slave device. The major difference is the type of data to be transferred. Modbus data is either bits or registers. EtherNet/IP data is referred to as an Instance. Each device manufacturer defines the type of data contained and the instances supported. Refer to the adapter documentation for details regarding MP2000iec configuration for successful communication.

The MP2000iec controller can communicate with up to 10 unique IP addresses simultaneously using the EtherNet/IP protocol. One input and one output instance can be configured per adapter.

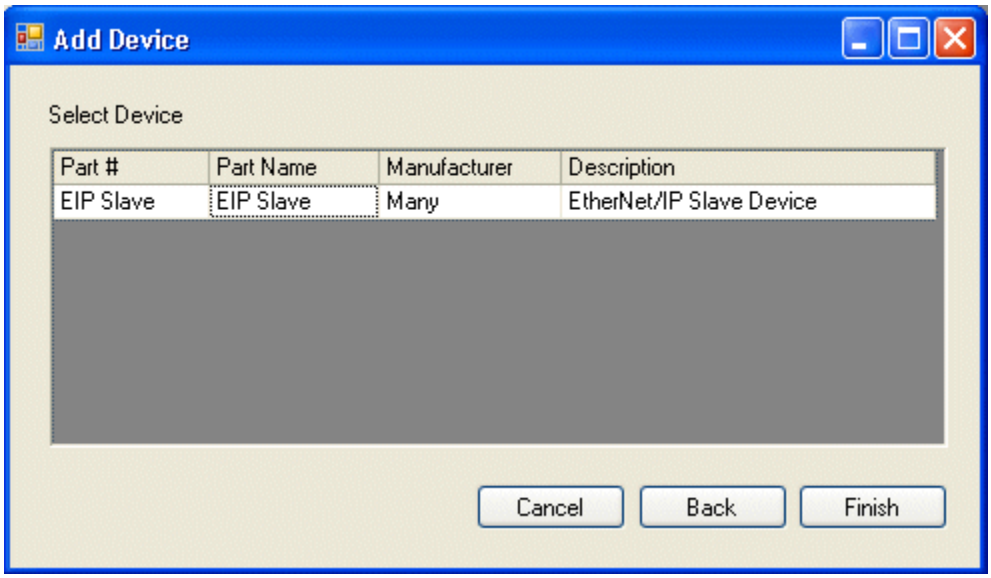
Explicit Messaging is not supported on the MP2300Siec in firmware version 1.0.

Step 1: Launch the Configuration 

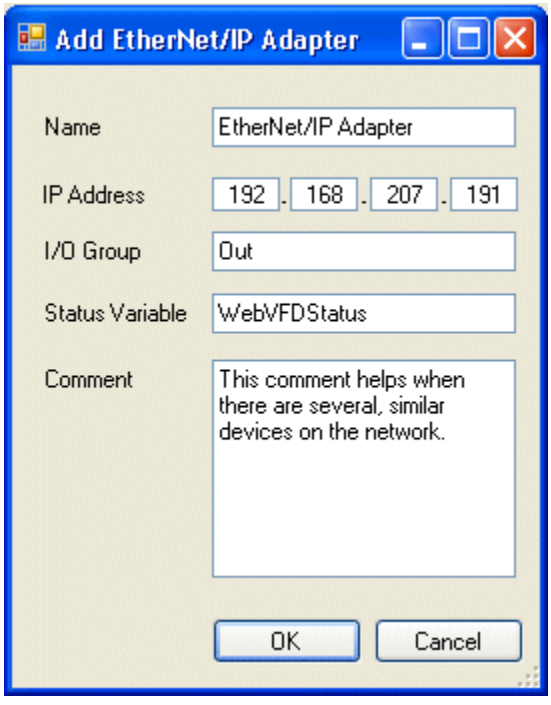
Step 2: Right click on the Configuration tree on the Modbus/TCP item



Step 3: Click Finish on the Add Device Dialog Box.

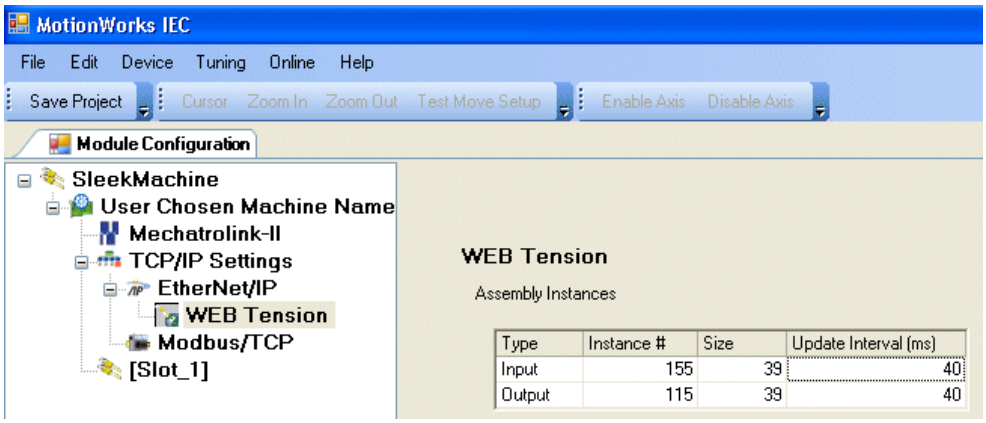


Step 4: The Add EtherNet/IP Adapter window is shown below. Select a name for the device to be added. This name will be displayed in the Global Variables list as a variable group. The Status Variable will be automatically entered in a variable group associated with this device. The Status Variable data type is WORD. This variable indicates the connection status. See ??? for details regarding the status variable. The minimum update time is 20 mSec.



Step 5: Click on the new Adapter Device in the Configuration tree at the left of the screen. Enter the assembly Instances required to communicate with the remote device. For example, in the documentation for Yaskawa's V7 EtherNet/IP communication module, the CM093, Instance 155 contains 27h bytes of data as shown in Figure 9. This predefined data will be output to a scanner requesting Assembly Instance 155.

It's very important to think about the definition of input and output. The output of one device is the input to another device. Notice that the V7 documentation indicates instance 155 is an output, but in the MotionWorksIEC configuration, Instance 155 is listed as in input.



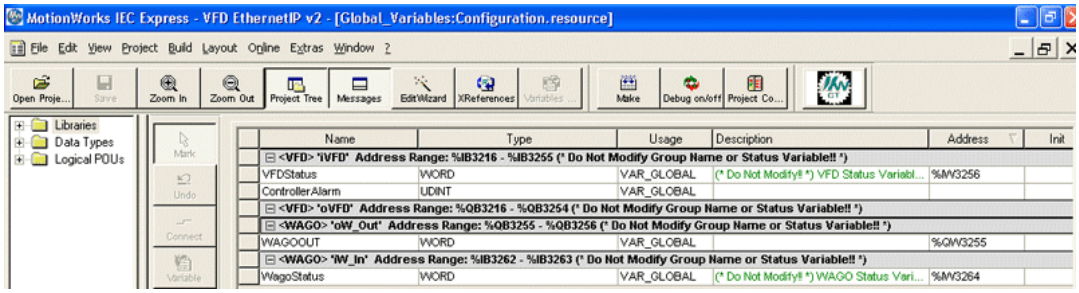
Yaskawa Standard Speed/Torque Output Instance 155 (9Bh)

	Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Input Instance 155	0	FAULT	ALARM	READY	Speed Agree	Reset	Running REV	-	Running FWD
	1	-	-	Terminal PC-PC	Terminal P1-PC	Terminal MA/MB-MC	Local / Remote	UV	OPE
	2	Output Frequency (Low Byte) [n035 Scaled]							
	3	Output Frequency (High Byte) [n035 Scaled]							
	4	Reserved							
	5	Reserved							
	6	Reserved							
	7	Reserved							
	8	Frequency Command (Low Byte)							
	9	Frequency Command (High Byte)							
	Ah	Output Frequency (Low Byte)							
	Bh	Output Frequency (High Byte)							
	Ch	Output Current (Low Byte)							
	Dh	Output Current (High Byte)							
	Eh	Reserved							
	Fh	Reserved							
	10h	Main Circuit DC Voltage (Low Byte)							
	11h	Main Circuit DC Voltage (High Byte)							
	12h	Error Alarm 1 (Low Byte)							
	13h	Error Alarm 1 (High Byte)							
	14h	Reserved							
	15h	Reserved							
	16h	Reserved							
	17h	Reserved							
	18h	Reserved							
	19h	Reserved							
	1Ah	Input Terminals (Low Byte)							
	1Bh	Input Terminals (High Byte)							
	1Ch	Reserved							
	1Dh	Reserved							
	1Eh	Reserved							
	1Fh	Reserved							
	...	Reserved							
	27h	Reserved							

Figure 9: Example of Assembly Instance documentation for an EtherNet/IP adapter. Taken from Yaskawa manual TM.V7.26

Step 6: When finished configuring the Assembly Instances, Save the Configuration. This will create the global memory I/O Group in the Development Environment. Switch to the Online mode and "Send the Offline Configuration," then Save the configuration to the controller and Cycle Power and the controller will start to communicate with the adapter device.

Step 7: Open the Global Variables list. Right click on the I/O group header to "Insert Variable." This variable can be any data type that fits the usage within the program. For example, a structure can be created to match the data size of the instance as described in the documentation for the adapter device.



The following is an example STRUCT created to match instance 155 of the Yaskawa V7 VFD.

NOTE: When creating STRUCTS, the minimum amount of space a single data element will occupy is a BYTE, so it is not possible to create BOOL type data in the STRUCT that will match the memory map of the EtherNet/IP instance.

```
YaskawaVFDOutputInstance155:
STRUCT
            (*            Byte 0                    *)
                (*  RunningForward          : BOOL; *)
                (*  UnusedByte0_1           : BOOL; *)
                (*  RunningReverse          : BOOL; *)
                (*  Reset                   : BOOL; *)
                (*  SpeedAgree              : BOOL; *)
                (*  Ready                   : BOOL; *)
                (*  Alarm                   : BOOL; *)
                (*  Fault                   : BOOL; *)
                 ControlByte0               : BYTE;

            (*            Byte 1                    *)
                (*  OPE                     : BOOL; *)
                (*  UV                      : BOOL; *)
                (*  LocalRemote             : BOOL; *)
                (*  TerminalMA_MC           : BOOL; *)
                (*  TerminalP1_PC           : BOOL; *)
                (*  TerminalP2_PC           : BOOL; *)
                (*  ReservedByte1_6         : BOOL; *)
                (*  ReservedByte1_7         : BOOL; *)
                ControlByte1                : BYTE;

            (*    Bytes  3 and 2                    *)
                OutputFrequency_Scaled      : UINT;

            (*    Bytes  5 and 4                    *)
                MotorTorque                 : UINT;

            (*    Bytes  7 and 6                    *)
                ReservedBytes7_6            : UINT;

            (*    Bytes  9 and 8                    *)
                FrequencyCommand            : UINT;

            (*    Bytes  B and A                    *)
                OutputFrequency             : UINT;

            (*    Bytes  D and C                    *)
                OutputCurrent               : UINT;
```

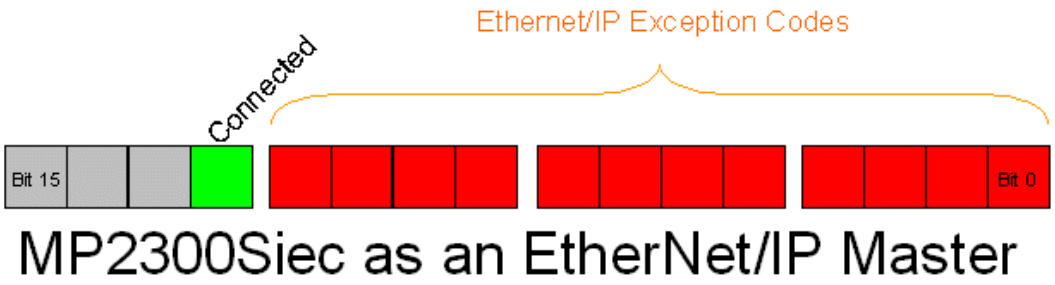
```
    (*    Bytes  F and E                    *)
       ReservedINT6                 : UINT;

    (*    Bytes 11 and 10                   *)
       Main_DC_Voltage              : UINT;

    (*    Bytes 13 and 12                   *)
       ErrorAlarm1                  : UINT;

    (*    Bytes 15 and 14                   *)
       ErrorAlarm2                  : UINT;

    (*    Bytes 17 and 16                   *)
       ErrorAlarm3                  : UINT;

    (*    Bytes 19 and 18                   *)
       ReservedBytes19_18           : UINT;

    (*    Bytes 1B and 1A                   *)
       InputTerminals               : WORD;

    (*    Bytes 1D and 1C                   *)
       ReservedBytes1D_1C           : UINT;

    (*    Bytes 1F and 1E                   *)
       ReservedBytes1F_1E           : UINT;

    (*    Bytes 21 and 20                   *)
       ReservedBytes21_20           : UINT;

    (*    Bytes 23 and 22                   *)
       ReservedBytes23_22           : UINT;

    (*    Bytes 25 and 24                   *)
       ReservedBytes25_24           : UINT;

    (*    Bytes 27 and 26                   *)
       ReservedBytes27_26           : UINT;

END_STRUCT;
```

Ethernet/IP Status Variable

This variable reflects the connection status. It can be used in the application program to determine if the connection to the remote device is active and the data is valid. The status WORD can be compared to 16#1000, which means that the connection is good. If bits 0 or 1 are on, the connection is not active, and the controller is attempting to reconnect to the remote device.

This status variable is only available when the MP2000iec controller is the client or master of a remote device.

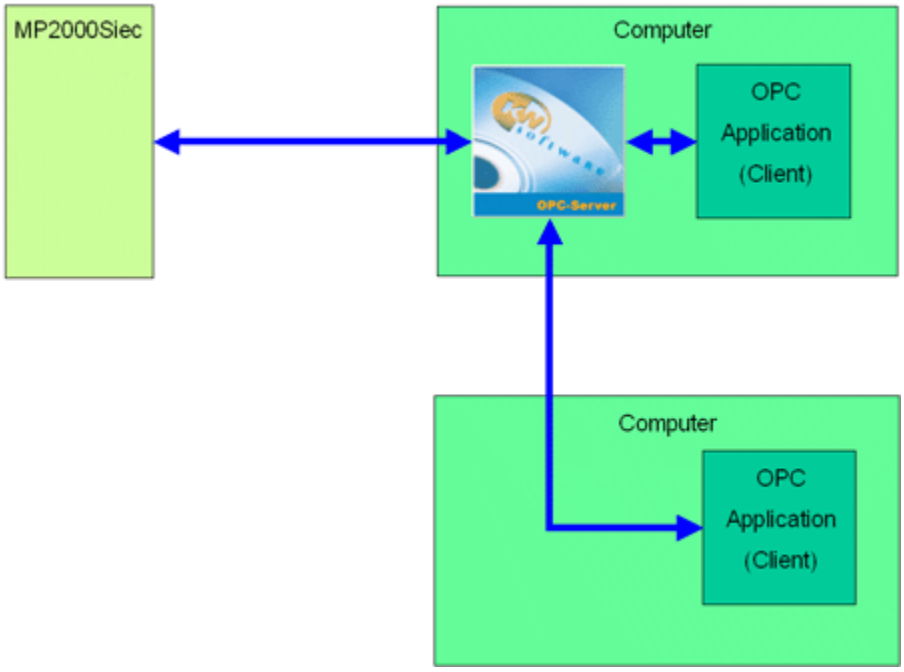
NOTE: If the status variable has a value of zero as observed in the Global Variables list, the controller may not be running the application program.



Lower 3 Nibbles	Description
0x001	Unable to connect (timeout)
0x002	I/O Timeout
0x003	Reconnect In Progress
0x004	Connection Failed (invalid response from server)
0x005	Connection Failed (out of resources)

Note: Do not to delete automatically created variables or groups. If you must delete a group, be sure to delete the group header (an all of its variables) and the group name as listed under IO_Configuration in the Hardware tab. These two items must remain in sync for project to compile properly

1.9.6 OPC Server



By default, all global variables are transmitted as OPC data. To disable this feature, click the hardware tab, right click the resource folder, and select Settings. (See Figure 10)

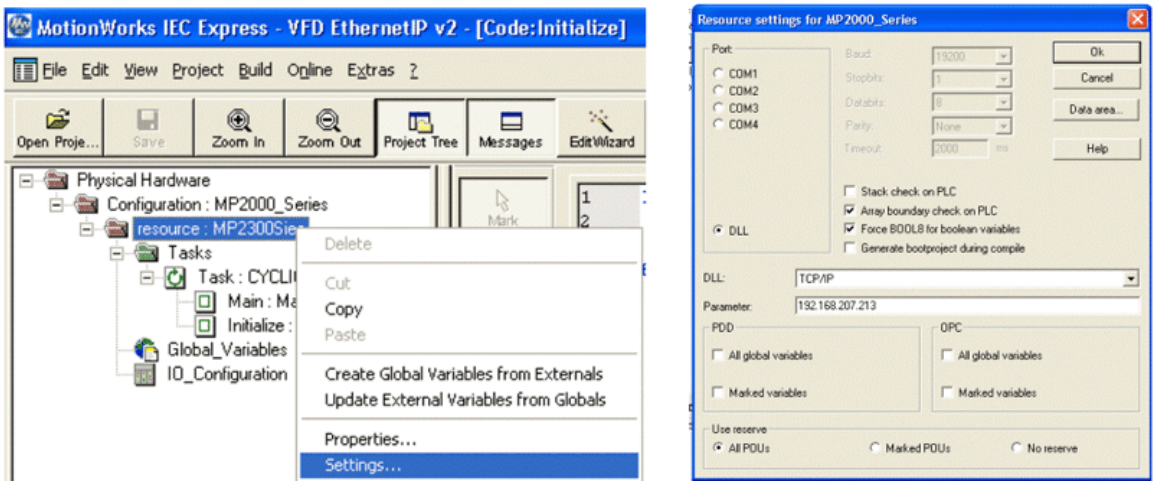


Figure 10: OPC Configuration

Any data in the application program can be set as OPC data by simply clicking the checkbox in the variable definition screen in the IEC development environment.

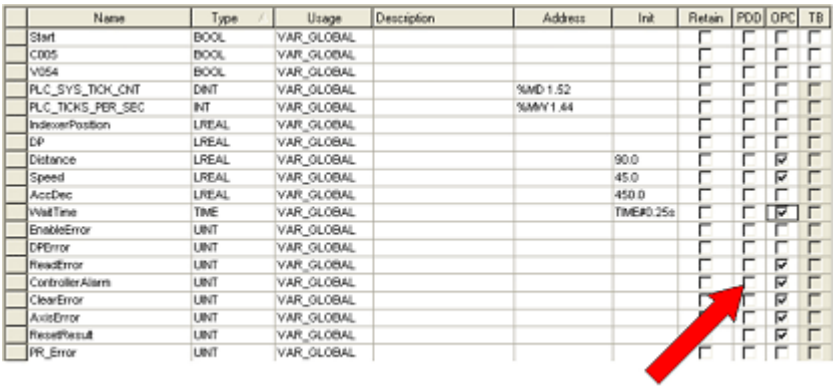


Figure 11: Variable Definition Window

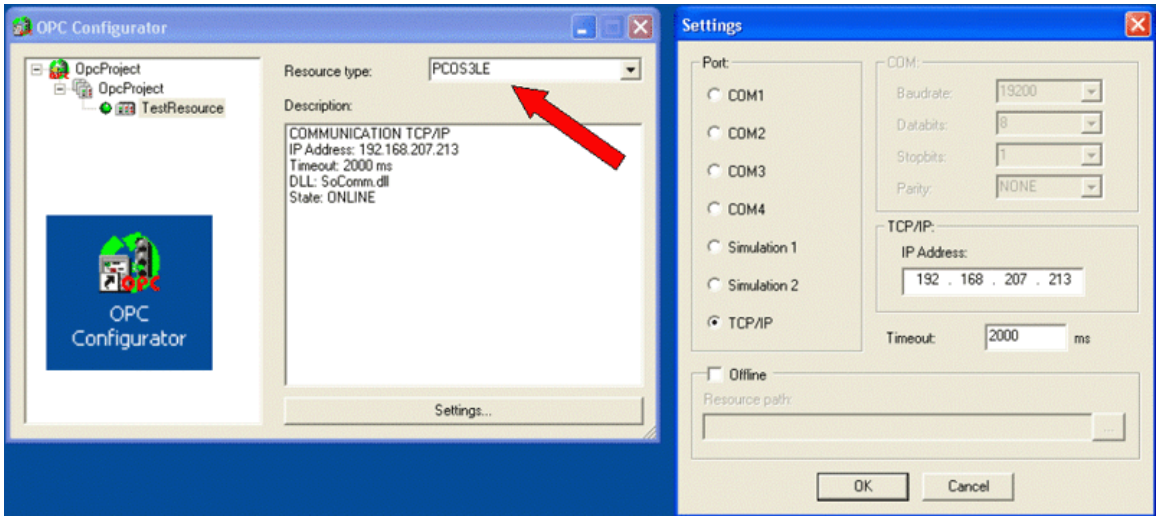
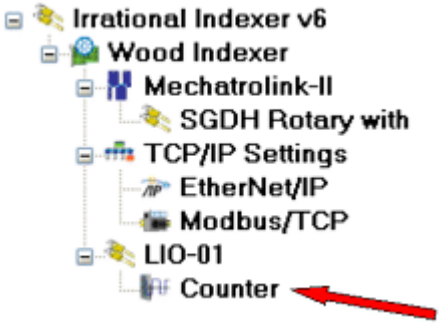


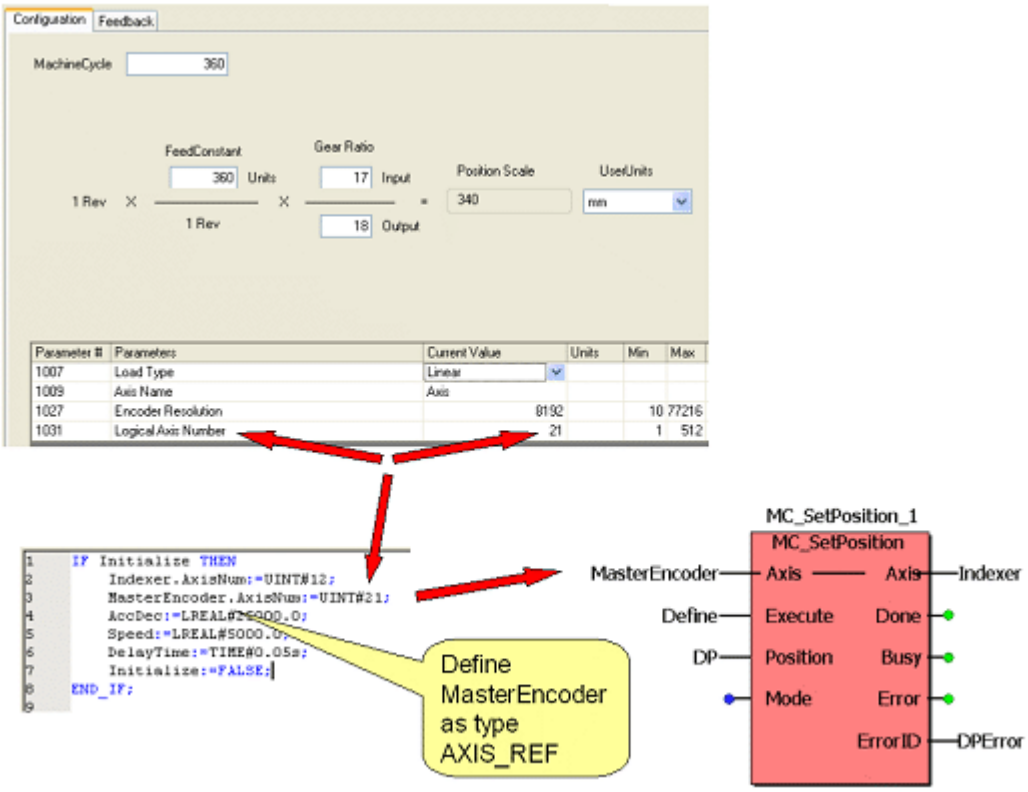
Figure 12: OPC Server Configurator

1.10 External Encoders

Option cards LIO-01, LIO-02, CNTR-01, and LIO-06 have an encoder input available for use with the PLCopen function blocks. When any of these cards are configured, external encoder functionality is enabled.



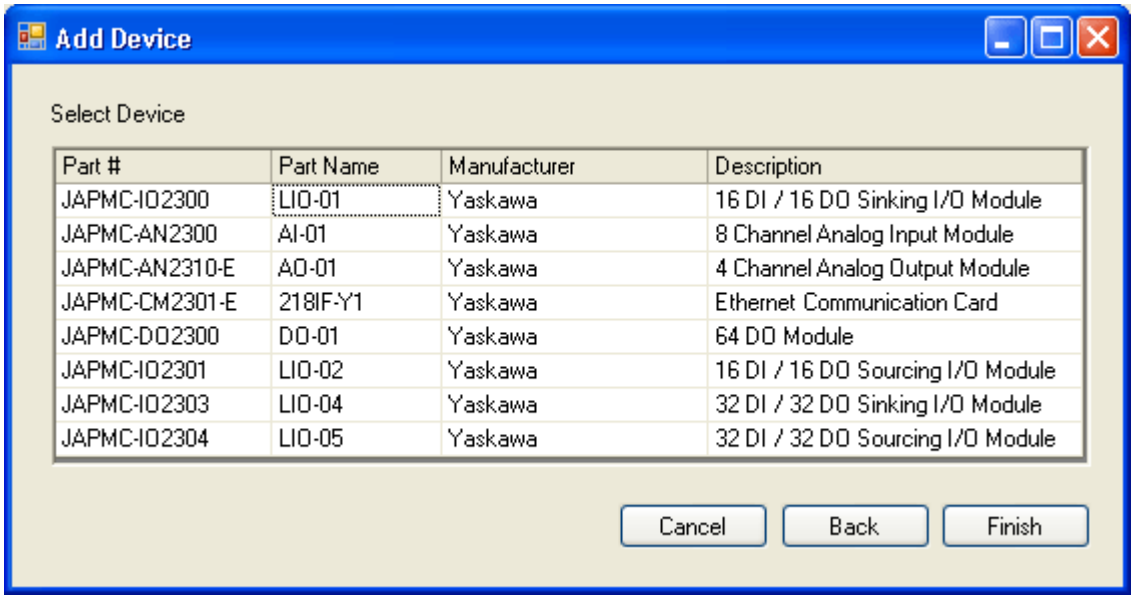
To use the encoder, look in the configuration to identify the logical axis number. Changes to the user unit parameters will not take effect until power is cycled. Encoder operation can be verified by checking the Feedback tab.



1.11 Option Slots

Each slot in the controller can optionally contain a module, or remain empty.

The following option cards are supported:



Note: Do not to delete automatically created variables or groups. If you must delete a group, be sure to delete the group header (an all of its variables) and the group name as listed under IO_Configuration in the Hardware tab. These two items must remain in sync for project to compile properly

2. Web Server

2.1 Web Server Overview

The Web Server is built into the controller firmware and allows the user to locally or remotely perform various activities. There are two access levels to the controller via the web server; the second level requires a password to access features such as firmware upgrade utilities.

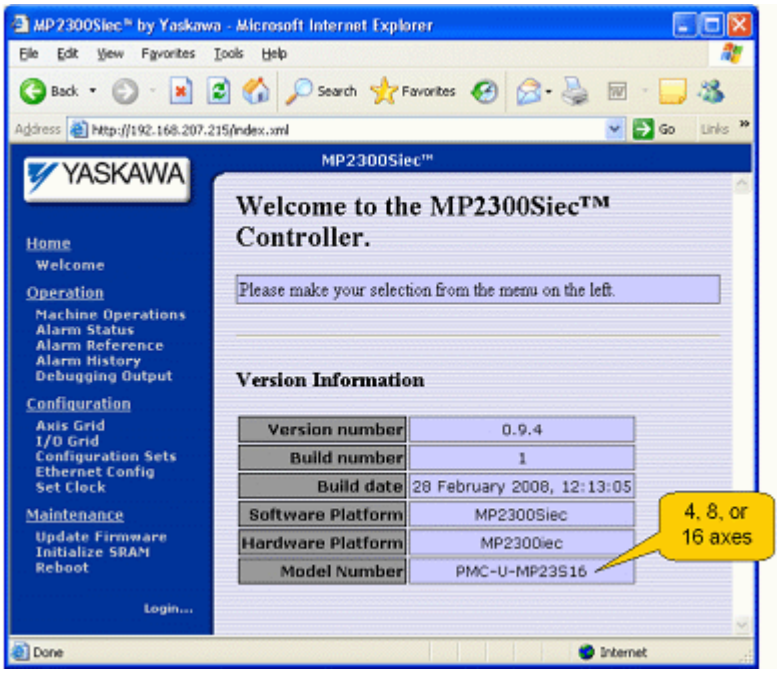
2.1 Web Server Overview

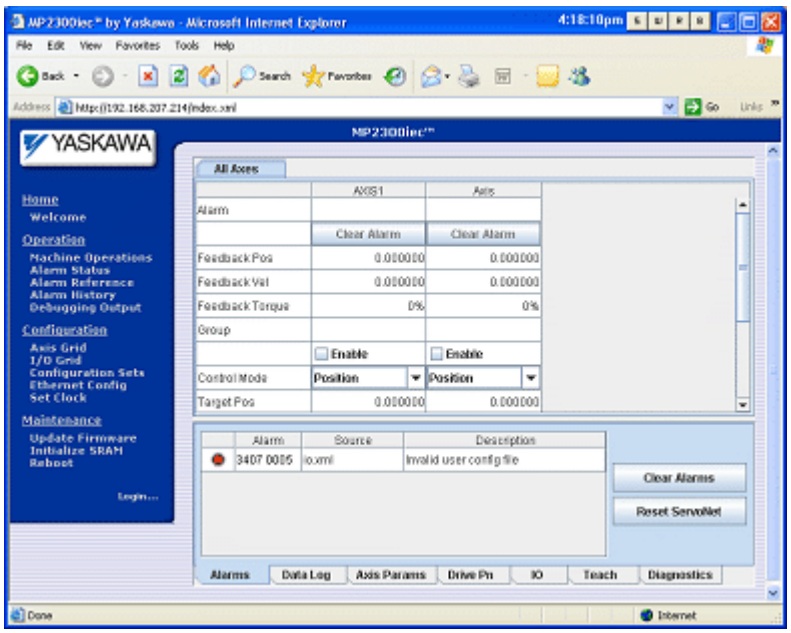
The Web Server is built into the controller firmware and allows the user to locally or remotely perform various activities. There are two access levels to the controller via the web server; the second level requires a password to access features such as firmware upgrade utilities.

2.2 Web Server Requirements

You must use Internet Explorer and have Java Virtual machine v.xx installed on your computer.

2.3 Web Server Homepage

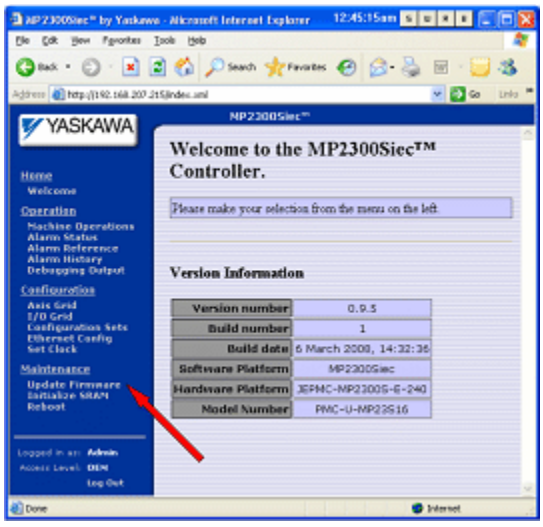




2.4 Firmware Upgrade

The firmware should not be upgraded unless recommended by an authorized Yaskawa representative.

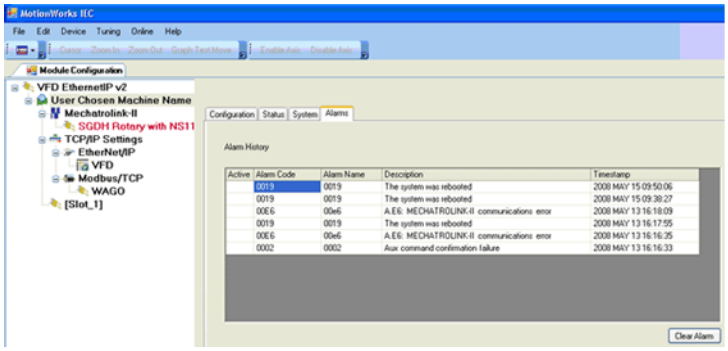
The controller firmware may be upgraded only if the controller is powered up in Supervisor mode. This is accomplished by setting the SUP DIP switch before turning the power on. To gain access to the controller's firmware upgrade function login to the controller via the web server with the Login and Password supplied by Yaskawa.

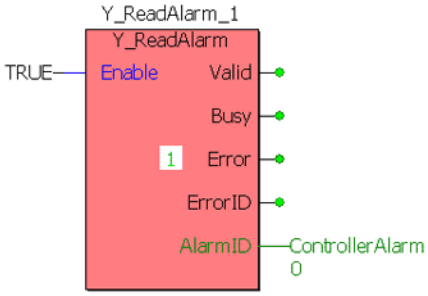


The firmware upgrade page will provide instructions regarding the upgrade process.

3. Controller AlarmID List

The following is a list of alarm codes that are reported in the Configurations Controller Alarms tab or via the Y_ReadAlarm function block.





Code		Type	Description
Hex	Decimal		
14030002	335740930	systemFailure	The task responsible for publishing events to a remote client failed to stop cleanly, which may result in unreleased system resources. Error recovery requires the controller be reset.
14030003	335740931	systemFailure	The task responsible for replying to remote clients failed to stop cleanly, which may result in unreleased system resources. Error recovery requires the controller be reset.
14030004	335740932	systemFailure	The task responsible starting and stoping connections to remote clients failed to stop cleanly, which may result in unreleased system resources. Error recovery requires the controller be reset.
14070001	336003073	systemFailure	The file system on which the configuration file directory resides could not be read and may be unmounted or corrupted. The system has booted in a minimal configuration mode, and most functionality is limited. If possible, the file system should be recovered or reformatted and new config files uploaded if applicable.
14070103	336003331	systemFailure	The watchdog timer expired.
14070108	336003336	systemFailure	A CPU exception occurred.

14070109	336003337	systemFailure	The firmware files on the controller do not match the expected checksums.
23010001	587268097	servoNetFailure	The drive returned an invalid watch dog code indicating a possible dropped communication packet.
23010002	587268098	servoNetFailure	The drive failed to return confirmation of last aux command within the default timeout period.
23010003	587268099	servoNetFailure	An unrecoverable error occurred during auto configuration. As a result, one or more drives are excluded from the servo network.
23010004	587268100	servoNetFailure	Overriding the auto configured axes parameters failed. As a result, one or more drives are excluded from the servo network.
23010005	587268101	servoNetFailure	Two or more nodes have the same ID. As a result, all servo network communication has been suspended.
23010006	587268102	servoNetFailure	The controller must be the root node on the servo network. All servo network communication has been suspended
23010007	587268103	servoNetFailure	The servo network communication device failed to initialize. Servo network communication is not possible.
23010008	587268104	servoNetFailure	An error occurred sending command to a node during initialization. The node may not support the configured communications rate. Communication with this node has been prohibited, but communication with other nodes may be possible.
23010010	587268112	servoNetFailure	It receives response with the same channel at the same Iso cycle.
23010011	587268113	servoNetFailure	The ID in the response packet is not same to ID of AxisNode.
23010012	587268114	servoNetFailure	The data length in the response packet is not same to value of CSR register(SEND_DSP_DATA_LENGTH) of drive.
23010013	587268115	servoNetFailure	The packet type in the response packet is not same S-DSP.

23010014	587268116	servoNetFailure	Invalid cycle time has passed with configuration file 'servonet.xml'. As a result, all servo network communication has been suspended.
23010015	587268117	servoNetFailure	Node is not found on 1394 network.
23010016	587268118	servoNetFailure	Invalid node.
23010017	587268119	servoNetFailure	Error matching node IDs.
33010009	855703561	alarm	Some motor properties, such as encoder resolution, maximum speed, and maximum torque, could not be determined for the attached motor. The serial encoder may be malfunctioning, incorrectly programmed, or unplugged.
33010018	855703576	alarm	The command position specified an instantaneous jump too large relative to the current position.
33010019	855703577	alarm	Setting of Pn002 digit 4 specifies torque feed-forward, but the SERVOPACK model does not support torque FF in position mode.
34070002	872873986	alarm	The base directory for configuration files was missing and has been created automatically. The system has booted in a minimal configuration mode, and most functionality is limited. Please upload a new complete configuration file set.
34070003	872873987	alarm	A required default configuration file was missing. A minimal configuration for the corresponding component has been loaded, and some functionality may be limited.
34070004	872873988	alarm	A required default configuration file was incorrectly formatted. A minimal configuration for the corresponding component has been loaded, and some functionality may be disabled.
34070005	872873989	alarm	A configuration file specified by the user configuration file set was incorrectly formatted. The corresponding default configuration file is being used instead.
34070006	872873990	alarm	The file describing which configuration set to use was corrupted. The default configuration set is being used.

34070007	872873991	alarm	An error occurred while writing a config file. The file system may be full or damaged.
34070101	872874241	alarm	The configured RAM disk on the controller was unable to be created.
34070102	872874242	alarm	Detected an unsupported card.
34070104	872874244	alarm	Data in the controller SRAM did not match the expected value. It should be treated as corrupted until it is re-initialized.
34070106	872874246	alarm	The SRAM battery backup power failed. SRAM data should be treated as corrupted until it is re-initialized.
34070107	872874247	alarm	The controller's time-of-day clock detected a voltage decrease in the backup battery. The current time and date is likely to be incorrect. This alarm can be cleared, but will recur when the controller is powered ON until the time and day is reset and the battery is replaced.
44030001	1141047297	warning	The event queue for the remote client was full, and an event was dropped. This is generally caused either by exceeding the network bandwidth or exceeding the general system processing power (starving the connection). When an event is dropped in this manner, the connection is terminated.
44030005	1141047301	warning	An RMI connection was attempted by an external client and rejected due to the concurrent connection limit.
44070001	1141309441	warning	The configuration file directory is read-only or resides on a read-only file system. Attempts to update the configuration or create directories will fail.
44070002	1141309442	warning	An attempt was made to write to a read-only configuration file. The write failed.
44070105	1141309701	warning	There was an indication that the SRAM battery backup power may have failed temporarily. SRAM data may have been compromised.

44080001	1141374977	warning	The alarm history was configured to use NVRAM storage, but either the available NVRAM was not sufficient to contain the configured buffer size, or the configured buffer size was not large enough to contain the configured number of records. The alarm history will contain fewer records than configured.
44080002	1141374978	warning	The alarm history was configured to use NVRAM storage and the data was found to be corrupted. The alarm history has been lost. NOTE: this alarm also occurs if the configured size of the alarm history has been changed.
1407010a	336003338	systemFailure	The manufacturing procedure failed. The controller probably could not fetch the current time from the network.
140a0009	336199689	systemFailure	Network reset detected multiple Axes connected to the same servo network node.
140a000a	336199690	systemFailure	Network reset detected multiple I/O connected to the same network node.
140a0015	336199701	systemFailure	Controller memory was corrupted during network reset resulting in a lost logical Axis data structure.
140a0016	336199702	systemFailure	Controller memory was corrupted during network reset resulting in a lost logical I/O data structure.
140a0018	336199704	systemFailure	An Abort input specified in the configuration could not be found. The abort condition is considered permanently asserted. No motion is possible until the I/O configuration can be matched to the abort inputs (restart required).
140a0021	336199713	systemFailure	Too many events were posted from the system ISR. The motion scan and servo net loop have been shut down.
140b0002	336265218	systemFailure	The controller ran out of free memory, possibly resulting in an unrecoverable failure. Please reboot the controller.
140b0004	336265220	systemFailure	The largest free memory block is too small, possibly resulting in an unrecoverable failure. Please reboot the controller.

140c0201	336331265	systemFailure	The EtherNet/IP task was shut down but failed to stop cleanly within 5 seconds. Some system resources may be unable to be reclaimed.
140c1026	336334886	systemFailure	The controller firmware consists of two components: the executable code image and the data file package. The two components' versions do not match. Re-install the firmware.
140c1027	336334887	systemFailure	The controller firmware file package was damaged or could not be verified. Re-install the firmware.
140c1100	336335104	systemFailure	Motion kernel assertion failed indicating the controller is in an invalid state.
2301000E	587268110	servoNetFailure	The drive does not return response packet.
2301000F	587268111	servoNetFailure	Bus reset generation that controller is not demanding.
3301000B	855703563	alarm	Setting of Pn002, digits 3 and 4, disables torque limit and/or velocity limit in velocity and/or torque control modes. Set Pn002 = xx11 to initialize.
3301000D	855703565	alarm	The servo network does not support this motion control mode.
3301001A	855703578	alarm	An error occurred while initializing a Mechatrolink node.
340a0001	873070593	alarm	The source for the logical input was not found, so the configured input will not be available.
340a0002	873070594	alarm	The source for the logical output was not found, the the configured output will not be available.
340a0003	873070595	alarm	Two or more axis in the configuration file had the same axis ID.
340a0004	873070596	alarm	The servo network axis node for the axis specified in the configuration file was not found.
340a0005	873070597	alarm	The axis group specified in the configuration file could not be created because either one or more of its axes are invalid or the group name is already being used.

340a0006	873070598	alarm	The type of AtTargetAgent specified in the configuration file is unknown. This is because AtTargetAgent could not be created.
340a0007	873070599	alarm	The number of constraints for axis group soft limit must be the same as the number of axes in the axis group.
340a0008	873070600	alarm	The axis group doesn't have the configured frame.
340a000b	873070603	alarm	A continuous-wrap range for an axis causes its position to automatically wrap around between two user-specified numbers. Generally these numbers evaluate to full revolutions of the encoder but other ranges are permitted. However, all ranges specified in user units must map exactly to an integral number of encoder pulses. This alarm indicates that the mapping from user units to encoder ticks was inexact. Use more precise numbers to describe the range or choose a different range that evaluates to an integral number of encoder pulses. When this alarm occurs at startup or servo-net reset, it indicates that the axis has not been connected to an axis node and cannot be servoed on. Otherwise, this alarm indicates that the specified continuous-wrap range was not put into effect.
340a000d	873070605	alarm	Two or more logical outputs specified in the I/O configuration file use the same physical bit. This can cause writes to not correctly generate value-change events on logical outputs for the shared bits. The configuration file should be fixed.
340a000e	873070606	alarm	One or more of the data parameters in the axis configuration file were out-of-range or otherwise incorrectly specified for the axis. The axis was not created and is not available.
340a0010	873070608	alarm	After servo network reset, the Axis failed to reconnect to the servo network. The drive might have been removed from the network, the node ID of the drive might have changed or there might be a communication problem.

340a0012	873070610	alarm	After servo network reset, the network I/O failed to reconnect to the servo network. The network I/O module might have been removed from the network, the node ID of the network I/O module might have changed or there might be a network communication problem.
340a0013	873070611	alarm	After servo network reset, a new axis node was discovered. This axis node is not associated with any existing axes and will not be available. To make this node available, update the configuration and power cycle the controller.
340a0014	873070612	alarm	After servo network reset, a new I/O node was discovered. This I/O node is not associated with any existing I/O and will not be available. To make this node available, update the configuration and power cycle the controller.
340a0017	873070615	alarm	One or more of the axis data or configuration parameters were inconsistent or incompatible with the axis node specified. The axis was created but was not connected to the servo node.
340a001b	873070619	alarm	Two or more LogicalInput have the same ID. The configuration file should be fixed.
340a001c	873070620	alarm	Two or more LogicalOutput have the same ID. The configuration file should be fixed.
340a001d	873070621	alarm	Two or more AnalogInput have the same ID. The configuration file should be fixed.
340a001e	873070622	alarm	Two or more AnalogOutput have the same ID. The configuration file should be fixed.
340a001f	873070623	alarm	Analog I/O configuration is missing the 'hardwareConfig' element, and configuration could not be resolved by the physical hardware. The configuration file should be fixed by adding this element to the analog I/O element.

340a0020	873070624	alarm	One or more axes failed to respond to a servo-off command during a system I/O initiated abort. This is normally the result of communication problems with the drive, which also causes an automatic servo-off.
340a0022	873070626	alarm	Reset of a servo node failed.
340a0023	873070627	alarm	The axis position may not be valid because the persistent axis data was corrupted. SRAM should be reinitialized and the axis should be homed.
340c0100	873201920	alarm	Invalid parameter passed to ProConOS I/O Driver
340c0101	873201921	alarm	Invalid structure for reading or writing ProConOS I/O Driver
340c0102	873201922	alarm	I/O memory area is not aligned to the correct byte to accommodate reading and writing.
340c0200	873202176	alarm	Common causes of invalid configuration include duplicate t2o/o2t assembly instances or invalid client connection parameters.
340c1020	873205792	alarm	The controller battery voltage has dropped, indicating it has failed or is about to fail. While the controller is powered on, the battery should be replaced as soon as possible or a prolonged power-down state will cause various static data to be lost.
340c1028	873205800	alarm	The driver parameter specified in the axis configuration caused an exception
340c1029	873205801	alarm	The driver parameter did not match the axis configuration
340c1030	873205808	alarm	The configured axis count exceeded the allowable limit.
340c1031	873205809	alarm	The axis count exceeded the allowable limit due to an auto-detected axis.
4301000A	1124139018	warning	The SERVOPACK model type was unable to be determined. This can indicate that some parameters may be incorrect.

4301000C	1124139020	warning	The controller was unable to send the drive command because servo network resources were allocated to motion. Brake on, brake off, absolute encoder initialization and alarm clear can only be sent when not moving.
440a000c	1141506060	warning	The position and torque scales specified in the configuration file have different signs. As a result, a positive acceleration results in a negative torque, and position limits are opposite in sign as the torque limits.
440a000f	1141506063	warning	The axis was temporarily disconnected from the servo network during reset. During this time, the feedback data is not valid and the axis cannot be moved.
440a0011	1141506065	warning	The network I/O was temporarily disconnected from the servo network during reset. During this time, any network I/O state change will be unobservable to the controller.
440a0019	1141506073	warning	The system was rebooted by the user.
440a001a	1141506074	warning	The system failed to shut down gracefully during a reboot, although the reboot did occur. This does not necessarily indicate that the software is damaged.
440b0001	1141571585	warning	The controller is running out of memory. Memory should be freed as soon as possible. Try closing connections to the controller or stopping scripts.
440b0003	1141571587	warning	The largest free memory block is approaching the critical level. Memory should be freed as soon as possible. Try closing connections to the controller or stopping scripts.