Yaskawa Memobus Plus (SA85) Driver Help

Table of Contents

Yaskawa Memobus Plus (SA85) Driver Help

Help version 1.009

CONTENTS

Overview

The Yaskawa Memobus Plus (SA85) Driver provides an easy and reliable way to connect Yaskawa Memobus Plus (SA85) devices to OPC Client applications, including HMI, SCADA, Historian, MES, ERP and countless custom applications. It is intended for use with a Modicon SA85 Network card. The driver also requires the Modicon MBPLUS.SYS driver for Windows NT or the Modicon MBPLUS.VXD driver for Windows 95. These drivers can be acquired from Modicon. These low level Modicon MBPLUS drivers must be installed and configured before this driver will work.

The driver can poll multiple devices (PLCs) on a Memobus Plus network and also act as a single slave device on the Memobus Plus network for other devices to poll. Currently the driver is limited to 2048 devices and is only designed to use Adapter 0. Up to eight channels can now be defined. Refer to the Server Help documentation for operating system (OS) requirements.

The Yaskawa Memobus Plus (SA85) Driver now supports the use of additional channel definitions for improved performance. These additional channels allow users to configure applications to take full advantage of the SA85 card. For more information on using multiple channels, refer to **Optimizing your Yaskawa Memobus Plus (SA85) Communications**.

Device ID (PLC Network Address)

The Device ID specifies the path to a node on the network, and consists of a path type designator (either slave or master) and five consecutive routing bytes. The **Data Master** address paths start with the prefix DM and are used to communicate with another node on the network. The host PC acts as the master in conversations of this type. A DM path can identify a PLC or any other devices that can respond to Memobus read and write commands, including another host PC running the Memobus Plus driver. The format of a DM path is **DM.r1.r2.r3.r4.r5**.

One **Data Slave** path is allowed per project and has the format **DS.1.0.0.0.0**. A slave path will enable the host PC running the Memobus Plus driver to simulate a PLC device on the network capable of responding to read/write requests from other devices. Other devices can communicate with this simulated device by opening a Data Master path to it. Addresses 1 to 65536 are implemented for output coils, input coils, internal registers and holding registers. The driver will respond to any valid request to read or write these values from external devices. These locations can also be accessed locally by the host PC as tags assigned to the slave device.

If a slave device is not defined in the project, the driver will ignore any unsolicited read or write requests it receives.

Note: A TIO Module device does not support a slave network address.

Addressing Example
Suppose that the single network consists of four nodes such that nodes 1 and 4 are host PCs running software that uses the Memobus Plus driver and nodes 2 and 3 are PLCs. The following table identifies the addressing for the network as seen from each node.

From	To Node 1	To Node 2	To Node 3	To Node 4
Node 1	---------------	DM.2.0.0.0.0	DM.3.0.0.0.0	DM.4.1.0.0.0
Node 2	DM.1.1.0.0.0	---------------	DM.3.0.0.0.0	DM.4.1.0.0.0
Node 3	DM.1.1.0.0.0	DM.2.0.0.0.0	---------------	DM.4.1.0.0.0
Node 4	DM.1.1.0.0.0	DM.2.0.0.0.0	DM.3.0.0.0.0	---------------

Note: To access the simulated device on a host PC, the last non-zero number in the path must always be one. This indicates the slave path the driver uses.

Block Sizes

Coil Block Sizes

Coils can be read from 8 to 800 points (bits) at a time. The default is 512 coils.

Register Block Sizes

Registers can be read from 1 to 120 locations (words) at a time. The default is 120 registers.

Reasons to Change the Default Block Sizes

1. The device may not support block read/write operations of the default size. Smaller Yaskawa PLCs and non-Yaskawa devices may not support the maximum data transfer lengths supported by the Memobus Plus network.

2. The device may contain non-contiguous addresses. If this is the case and the driver attempts to read a block of data that encompasses undefined memory, the device will probably reject the request.

Settings

Adapter Number
This parameter specifies the adapter number that will be used by the Memobus Plus card. Up to eight channels may be assigned the same adapter number. Valid adapter numbers are 0 to 3.

Timeout
This parameter specifies the amount of time that the driver will wait on a response from the device before giving up and going on to the next request. The timeout will be rounded up to the nearest half second. Longer timeouts only affect performance if a device is not responding. The driver polls the MBplus system driver for the device response at 10 ms intervals.

Zero vs One Based Addressing
If the address numbering convention for the device starts at one as opposed to zero, this parameter can be specified when defining the device's parameters. By default, user entered addresses will have one subtracted from them when frames are constructed to communicate with a Yaskawa Memobus device. This means that if holding register 40001 is requested, the driver will actually ask for the zero's holding register in the device.

If the device doesn't follow this convention, then uncheck **Use zero based addressing** in Device Properties. The default behavior follows the convention of the Yaskawa PLCs.

First Word Low in 32 Bit Data Types
Two consecutive registers addresses in a Yaskawa Memobus device are used for 32 bit data types. Users can specify whether the driver should assume the first word is the low or the high word of the 32 bit value. The default, first word low, follows the convention of the Yaskawa Memosoft programming software.

Optimizing Your Yaskawa Memobus Plus (SA85) Communications

The Yaskawa Memobus Plus (SA85) driver has been redesigned to provide better throughput and take full advantage of the SA85 card. In the past, the Yaskawa Memobus Plus (SA85) driver restricted users to configuring a single channel in the OPC Server project. All Memobus Plus devices were defined under this one channel. In simple terms, this meant that the driver had to move between devices one at a time making requests. The OPC Server was already designed to be efficient and for most applications the single channel scheme provides more than enough performance. With the advent of OPC as an enabling technology, the size of projects have increased dramatically and have also caused a decrease in performance. To remedy this, the Yaskawa Memobus Plus (SA85) driver operates at a new level of efficiency and performance.

Note: Before implementing any changes, it is suggested that users make a backup of their existing OPC Server project directory in order to return to the previous current settings quickly, if needed.

An example of how the application may appear when configured using a single channel is shown below.
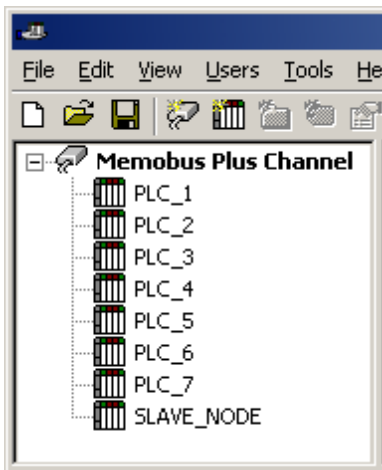
Each device appears under a single Yaskawa MemobusPlus channel. In this configuration, the driver must move from one device to the next as quickly as possible in order to gather information at an effective rate. As more devices are added or more information is requested from a single device, the overall update rate begins to suffer.

If the Yaskawa Memobus Plus (SA85) driver could only define one single channel, then the example shown above would be the only option available; however, the Yaskawa Memobus Plus (SA85) driver can define up to 8 channels. Using multiple channels distributes the data collection workload by simultaneously issuing multiple requests to the network. An example of how the same application may appear when configured using multiple channels to improve performance is shown below.
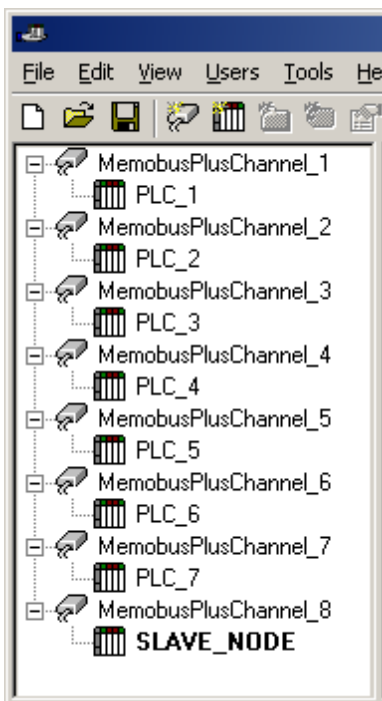
Each device has now been defined under its own channel. In this new configuration, a single path of execution is dedicated to the task of gathering data from each device. If the application has 8 or fewer devices, it can be optimized exactly how it is shown here.

The performance will improve even if the application has more than 8 devices. While 8 or fewer devices may be ideal, the application will still benefit from additional channels. Although by spreading the device load across all channels will cause the server to move from device to device again, it can now do so with far less devices to process on a single channel.

The 8-channel limit matches the multi-path limitations of the SA85 card as set by Modicon.

This application can support multiple channels even if it contains a large number of tags defined under each device. To do so, open the existing OPC Server project while it is still configured using a single channel. Next, click Channels|Add Channel in ordet to add a new Memobus Plus channel. Name the channel as desired. In this example, NewMemobusPlus is used.

Next, use the cut and paste options available by right-clicking on a device or by using the Edit menu. In this example, PLC_2 is cut from the MemobusPlus channel and then pasted under the NewMemobusPlus channel.
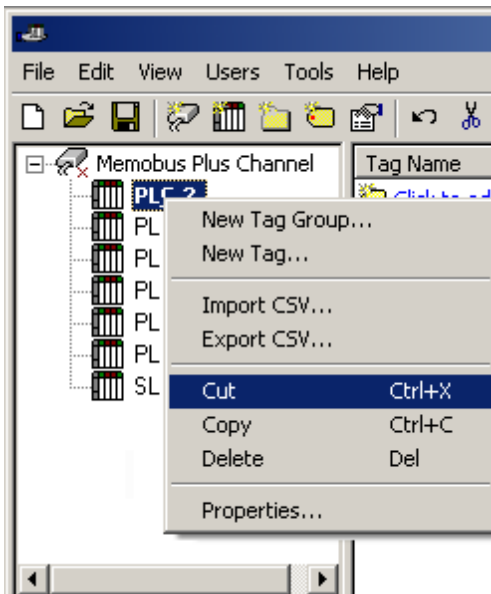
PLC_2 being cut from the Memobus Plus Channel.
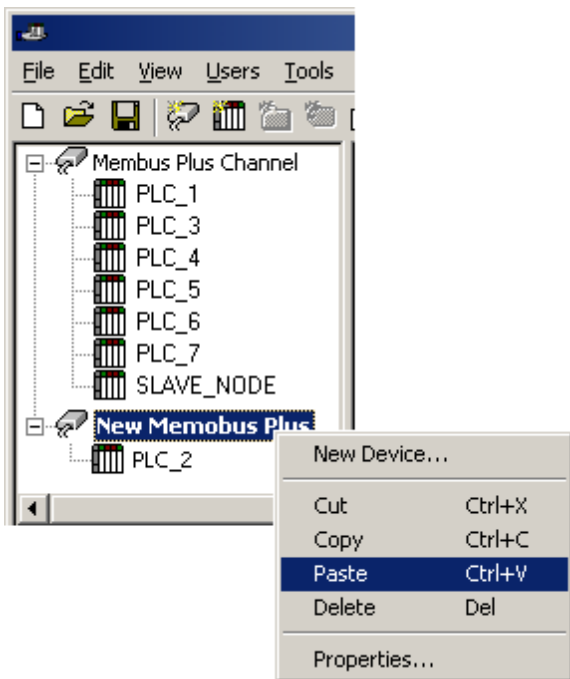
PLC_2 being pasted beneath the New Memobus Plus channel.

Other possible Memobus Plus Driver optimizations include dedicating a single channel to just Global Data in order to receive fast Global Data updates. To do so, define a new set of device names for each device whose Global Data will be accessed under that new channel. The important thing to remember in this case is to only access Global Data from these newly defined device names. There are many possible configurations that may benefit the application; the new Memobus Plus driver gives users the opportunity to investigate them.

Data Types Description

Data Type	Description
Boolean	Single bit
Word	Unsigned 16 bit value bit 0 is the low bit bit 15 is the high bit
Short	Signed 16 bit value bit 0 is the low bit bit 14 is the high bit bit 15 is the sign bit
DWord*	Unsigned 32 bit value bit 0 is the low bit bit 31 is the high bit
Long*	Signed 32 bit value bit 0 is the low bit bit 30 is the high bit bit 31 is the sign bit
BCD	Two byte packed BCD Value range is 0-9999. Behavior is undefined for values beyond this range.
LBCD*	Four byte packed BCD Value range is 0-99999999. Behavior is undefined for values beyond this range.
Float*	32 bit floating point value. The driver interprets two consecutive registers as a floating point value by making the second register the high word and the first register the low word.
Float Example	If register 40001 is specified as a float, bit 0 of register 40001 would be bit 0 of the 32 bit word, and bit 15 of register 40002 would be bit 31 of the 32 bit word.
String	Null terminated ASCII string. Includes Hi-Lo Lo-Hi byte order selection.

*The descriptions of the bit significance assume default, 'first word low' setting for 32 bit data types.

Yaskawa Memobus Plus (SA85) Address Descriptions

Address specifications vary depending on the model in use. Select a link from the following list to obtain specific address information for the model of interest.

Output Coils
Input Coils
Internal Registers
Holding Registers
Global Data
Constant Registers
Link Coils
Link Registers
MC Coils
MC Control Coils
MC Relays
MC Control Relays
MC Code Relays

Output Coils

	Decimal Addressing
Type	Boolean
Format	0xxxxx
Security	Read/Write
Range	1-65536

Array Support
Arrays are also supported for the output coil addresses. The syntax for declaring an array (shown using decimal addressing) is as follows:

0xxxxx[cols] with assumed row count of 1
0xxxxx[rows][cols]

The number of coils requested cannot exceed the coil block size set for the device.

Examples
1. The 255th output coil would be addressed as '0255' using decimal addressing.
2. The 35005th output coil would be addressed as '035005' using decimal addressing.

Input Coils

	Decimal Addressing
Type	Boolean
Format	1xxxxx
Security	Read
Range	1-65536

*These locations are Read/Write for Slave Devices.

Array Support
Arrays are also supported for the input coil addresses. The syntax for declaring an array (shown using decimal addressing) is as follows:

1xxxxx[cols] with assumed row count of 1
1xxxxx[rows][cols]

The number of coils requested cannot exceed the coil block size set for the device.

Examples
1. The 127th input coil would be addressed as '10127' using decimal addressing.
2. The 35005th output coil would be addressed as '135005' using decimal addressing.

Internal Registers

The default data types are shown in **bold**.

	Decimal Addressing
Type	**Word**, Short, BCD
Format	3xxxxx
Security	Read Only*
Range	1-65536
Type	**Boolean**
Format	3xxxxx.bb
Security	Read Only*
Range	xxxxx.0-xxxxx.15
Type	Float, DWord, Long, LBCD
Format	3xxxxx
Security	Read Only*
Range	1-65535

*These locations are Read/Write for Slave Devices.

Array Support

Arrays are also supported for the internal register addresses. The syntax for declaring an array (shown using decimal addressing) is as follows:

3xxxxx[cols] with assumed row count of 1
3xxxxx[rows][cols]

For Word, Short and BCD arrays, the base address + (rows * cols) cannot exceed 65536.
For Float, DWord, Long and Long BCD arrays, the base address + (rows * cols * 2) cannot exceed 65536.
For all arrays, the total number of registers being requested cannot exceed the register block size that was specified for this device.

Examples
1. To access Internal Register 1000, enter 301000.
2. To access Internal Register 11000, enter 311000.
3. To access Internal Register 300 bit 11, enter 300300.11.

Holding Registers

The default data types are shown in **bold.**

	Decimal Addressing
Type	**Word**, Short, BCD
Format	4xxxxx
Security	Read/Write
Range	1-65536
Type	**Boolean**
Format	4xxxxx.bb
Security	Read/Write
Range	xxxxx.0-xxxxx.15
Type	Float, DWord, Long, LBCD
Format	4xxxxx
Security	Read/Write
Range	1-65535
Type	**String**
String with HiLo byte order	400001.2H-465536.240H .Bit is string length, range 2 to 240 bytes.
String with LoHi byte order	400001.2L-465536.240L

	.Bit is string length, range 2 to 240 bytes.
Security	Read/Write

String Support

This driver supports reading and writing holding register memory as an ASCII string. When using holding registers for string data, each register will contain two bytes of ASCII data. The order of the ASCII data within a given register can be selected when the string is defined. The length of the string can be from 2 to 240 bytes and is entered in place of a bit number. The length must be entered as an even number. Appending either an "H" or "L" to the address specifies the byte order.

Examples

1. To access a string starting at 40200 with a length of 100 bytes and Hi-Lo byte order, enter: 40200.100H.

2. To access a string starting at 40500 with a length of 78 bytes and Lo-Hi byte order, enter: 40500.78L.

Array Support

Arrays are also supported for the holding register addresses. The syntax for declaring an array (using decimal addressing) is as follows:

4xxxx[cols] with assumed row count of 1
4xxxx[rows][cols]

For Word, Short and BCD arrays, the base address + (rows * cols) cannot exceed 65536.
For Float, DWord, Long and Long BCD arrays, the base address + (rows * cols * 2) cannot exceed 65535.
For all arrays, the total number of registers being requested cannot exceed the register block size that was specified for this device.

Examples

1. To access Holding Register 5000, enter 405000.
2. To access Holding Register 14000, enter 414000.
3. To access Holding Register 500 bit 11, enter 400500.11.

Global Data

The default data types are shown in **bold**.

Note: Global Data is not supported for the slave device.

	Decimal Addressing
Type	**Word**, Short, BCD
Format	Gxx
Security	Read/Write
Range	1-32
Type	Float, DWord, Long, LBCD
Format	Gxx
Security	Read/Write
Range	1-31

Array Support

Arrays are also supported for Global Data. The syntax for declaring an array (shown using decimal addressing) is as follows:

Gxx[cols] with assumed row count of 1
Gxx[rows][cols]

For Word, Short and BCD arrays, the base address + (rows * cols) cannot exceed 32.
For Float, DWord, Long and Long BCD arrays, the base address + (rows * cols * 2) cannot exceed 32.

Examples

1. To access Global Register 11, enter G11.
2. To access Global Register 30, enter G30.

Constant Registers

The default data types are shown in **bold**.

	Decimal Addressing
Type	**Word**, Short, BCD
Format	7xxxxx
Security	Read/Write
Range	1-65536
Type	**Boolean**
Format	7xxxxx.bb
Security	Read/Write
Range	xxxxx.0-xxxxx.15
Type	Float, DWord, Long, LBCD
Format	7xxxxx
Security	Read/Write
Range	1-65535
Type	**String**
String with HiLo byte order	700001.2H-765536.240H .Bit is string length, range 2 to 240 bytes.
String with LoHi byte order	700001.2L-765536.240L .Bit is string length, range 2 to 240 bytes.
Security	Read/Write

String Support

This driver supports reading and writing constant register memory as an ASCII string. When using constant registers for string data, each register will contain two bytes of ASCII data. The order of the ASCII data within a given register can be selected when the string is defined. The length of the string can be from 2 to 240 bytes and is entered in place of a bit number. The length must be entered as an even number. Appending either an "H" or "L" to the address specifies the byte order.

Examples

1. To access a string starting at 70200 with a length of 100 bytes and Hi-Lo byte order, enter: 70200.100H.

2. To access a string starting at 70500 with a length of 78 bytes and Lo-Hi byte order, enter: 70500.78L.

Array Support

Arrays are also supported for the constant register addresses. The syntax for declaring an array (using decimal addressing) is:

7xxxxx[cols] with assumed row count of 1
7xxxxx[rows][cols]

For Word, Short and BCD arrays, the base address + (rows * cols) cannot exceed 65536.
For Float, DWord, Long and Long BCD arrays, the base address + (rows * cols * 2) cannot exceed 65535.
For all arrays, the total number of registers being requested cannot exceed the register block size that was specified for this device.

Examples

1. To access Constant Register 300, enter 703000.
2. To access Constant Register 1400, enter 714000.
3. To access Constant Register 500 bit 11, enter 700500.11.

Link Coils

	Decimal Addressing
Type	Boolean

Format	D1xxxx or D2xxxx
Security	Read/Write
Range	1-9999

Array Support

Arrays are also supported for the link coil addresses. The syntax for declaring an array (shown using decimal addressing) is as follows:

D1xxxx[cols] with assumed row count of 1
D1xxxx[rows][cols]

The number of coils requested cannot exceed the coil block size set for the device.

Examples
1. The 10255th Link coil would be addressed as 'D10255' using decimal addressing.
2. The 20005th Link coil would be addressed as 'D20005' using decimal addressing.

Link Registers

The default data types are shown in **bold.**

	Decimal Addressing
Type	**Word**, Short, BCD
Format	R1xxxx or R2xxxx
Security	Read/Write
Range	1-9999
Type	**Boolean**
Format	R1xxxx.bb or R2xxxx.bb
Security	Read/Write
Range	xxxx.0-xxxx.15
Type	Float, DWord, Long, LBCD
Format	R1xxxx or R2xxxx
Security	Read/Write
Range	1-9998
Type	**String**
String with HiLo byte order	R10001.2H-R19999.240H R20001.2H-R29999.240H .Bit is string length, range 2 to 240 bytes.
String with LoHi byte order	R10001.2L-R19999.240L R20001.2L-R29999.240L .Bit is string length, range 2 to 240 bytes.
Security	Read/Write

String Support

This driver supports reading and writing link register memory as an ASCII string. When using link registers for string data, each register will contain two bytes of ASCII data. The order of the ASCII data within a given register can be selected when the string is defined. The length of the string can be from 2 to 240 bytes and is entered in place of a bit number. The length must be entered as an even number. Appending either an "H" or "L" to the address specifies the byte order.

Examples
1. To access a string starting at R10200 with a length of 100 bytes and Hi-Lo byte order, enter: R10200.100H.

2. To access a string starting at R10500 with a length of 78 bytes and Lo-Hi byte order, enter: R10500.78L.

Array Support

Arrays are also supported for the holding register addresses. The syntax for declaring an array (using decimal addressing) is as follows:

R1xxxx[cols] with assumed row count of 1
R1xxxx[rows][cols]

For Word, Short and BCD arrays, the base address + (rows * cols) cannot exceed 9999.
For Float, DWord, Long and Long BCD arrays, the base address + (rows * cols * 2) cannot exceed 9998.
For all arrays, the total number of registers being requested cannot exceed the register block size that was specified for this device.

Examples
1. To access Link Register 10500, enter R10500.
2. To access Link Register 21400, enter R21400.
3. To access Link Register 10500 bit 11, enter R10500.11

MC Coils

	Decimal Addressing
Type	Boolean
Format	Y1xxxx or Y2xxxx
Security	Read/Write
Range	1-9999

Array Support
Arrays are also supported for the MC coil addresses. The syntax for declaring an array (shown using decimal addressing) is as follows:

Y1xxxx[cols] with assumed row count of 1
Y1xxxx[rows][cols]

The number of coils requested cannot exceed the coil block size set for the device.

Examples
1. The 10255th MC coil would be addressed as 'Y10255' using decimal addressing.
2. The 20005th MC coil would be addressed as 'Y20005' using decimal addressing.

MC Control Coils

	Decimal Addressing
Type	Boolean
Format	Q1xxxx or Q2xxxx
Security	Read/Write
Range	1-9999

Array Support
Arrays are also supported for the MC Control coil addresses. The syntax for declaring an array (shown using decimal addressing) is as follows:

Q1xxxx[cols] with assumed row count of 1
Q1xxxx[rows][cols]

The number of coils requested cannot exceed the coil block size set for the device.

Examples
1. The 10255th MC Control coil would be addressed as 'Q10255' using decimal addressing.
2. The 20005th MC Control coil would be addressed as 'Q20005' using decimal addressing.

MC Relays

	Decimal Addressing
Type	Boolean
Format	X1xxxx or X2xxxx
Security	Read Only
Range	1-9999

Array Support

Arrays are also supported for the MC Relay addresses. The syntax for declaring an array (shown using decimal addressing) is as follows:

X1xxxx[cols] with assumed row count of 1
X1xxxx[rows][cols

The number of relays requested cannot exceed the coil block size set for the device.

Examples
1. The 10255th MC Relay would be addressed as 'X10255' using decimal addressing.
2. The 20005th MC Relay coil would be addressed as 'X20005' using decimal addressing.

MC Control Relays

	Decimal Addressing
Type	Boolean
Format	P1xxxx or P2xxxx
Security	Read Only
Range	1-9999

Array Support

Arrays are also supported for the MC Control Relay addresses. The syntax for declaring an array (shown using decimal addressing) is as follows:

P1xxxx[cols] with assumed row count of 1
P1xxxx[rows][cols]

The number of relays requested cannot exceed the coil block size set for the device.

Examples
1. The 10255th MC Control Relay would be addressed as 'P10255' using decimal addressing.
2. The 20005th MC Control Relay coil would be addressed as 'P20005' using decimal addressing.

MC Code Relays

	Decimal Addressing
Type	Boolean
Format	M1xxxx or M2xxxx
Security	Read Only
Range	1-9999

Array Support

Arrays are also supported for the MC Code Relay addresses. The syntax for declaring an array (shown using decimal addressing) is as follows:

M1xxxx[cols] with assumed row count of 1
M1xxxx[rows][cols]

The number of relays requested cannot exceed the coil block size set for the device.

Examples
1. The 10255th MC Code Relay would be addressed as 'M10255' using decimal addressing.
2. The 20005th MC Code Relay coil would be addressed as 'M20005' using decimal addressing.

Error Descriptions

The following error/warning messages may be generated. Click on the link for a description of the message.

Address Validation
Missing address
Device address '<address>' contains a syntax error
Address '<address>' is out of range for the specified device or register
Data Type '<type>' is not valid for device address '<address>'
Device address '<address>' is Read Only
Array size is out of range for address '<address>'
Array support is not available for the specified address: '<address>'

Device Status Messages
Started MBPLUS.SYS device
Device '<device name>' is not responding
Unable to write to '<address>' on device '<device name>'

Device Specific Messages
Unable to start MBPLUS.SYS device
Unable to communicate with MBPLUS.VXD
Unable to open MBPLUS slave path
Error opening MBPLUS path: <ID>
Bad address in block [<start address> to <end address>] on device '<device name>'

Address Validation

The following error/warning messages may be generated. Click on the link for a description of the message.

Address Validation
Missing address
Device address '<address>' contains a syntax error
Address '<address>' is out of range for the specified device or register
Data Type '<type>' is not valid for device address '<address>'
Device address '<address>' is Read Only
Array size is out of range for address '<address>'
Array support is not available for the specified address: '<address>'

Missing address

Error Type:
Warning

Possible Cause:
A tag address that has been specified dynamically has no length.

Solution:
Re-enter the address in the client application.

Device address '<address>' contains a syntax error

Error Type:
Warning

Possible Cause:
A tag address that has been specified dynamically contains one or more invalid characters.

Solution:
Re-enter the address in the client application.

Address '<address>' is out of range for the specified device or register

Error Type:
Warning

Possible Cause:

A tag address that has been specified dynamically references a location that is beyond the range of supported locations for the device.

Solution:

Verify the address is correct; if it is not, re-enter it in the client application.

Data Type '<type>' is not valid for device address '<address>'

Error Type:
Warning

Possible Cause:

A tag address that has been specified dynamically has been assigned an invalid data type.

Solution:

Modify the requested data type in the client application.

Device address '<address>' is Read Only

Error Type:
Warning

Possible Cause:

A tag address that has been specified dynamically has a requested access mode that is not compatible with what the device supports for that address.

Solution:

Change the access mode in the client application.

Array size is out of range for address '<address>'

Error Type:
Warning

Possible Cause:

A tag address that has been specified dynamically is requesting an array size that is too large for the address type or block size of the driver.

Solution:

Re-enter the address in the client application to specify a smaller value for the array or a different starting point.

Array support is not available for the specified address: '<address>'

Error Type:
Warning

Possible Cause:

A tag address that has been specified dynamically contains an array reference for an address type that doesn't support arrays.

Solution:

Re-enter the address in the client application to remove the array reference or correct the address type.

Device Status Messages

The following error/warning messages may be generated. Click on the link for a description of the message.

Device Status Messages
Started MBPLUS.SYS device
Device '<device name>' is not responding
Unable to write to '<address>' on device '<device name>'

Started MBPLUS.SYS device

Error Type:
Information

Possible Cause:
Posted by the driver when it successfully starts the MBPLUS.SYS device driver. This is a Windows NT only message and will not be seen if the MBPLUS.SYS driver is already running when this driver starts.

Solution:
N/A

Device '<device name>' is not responding

Error Type:
Serious

Possible Cause:
1. The PLC network card may not be correctly installed in the host PC.
2. The named device may not be connected to the PLC network.
3. The named device may have been assigned an incorrect Network ID.
4. The driver cannot open a path on the specified adapter.
5. The response from the device took longer to receive than the amount of time specified in the "Request Timeout" device setting.

Solution:
1. Verify the network card installation using the supplied utility software.
2. Check the PLC network connections.
3. Verify that the Network ID given to the named device matches that of the actual device.
4. Verify that no more than eight channels are assigned the same adapter number.
5. Increase the Request Timeout setting so that the entire response can be handled.

Unable to write to '<address>' on device '<device name>'

Error Type:
Serious

Possible Cause:
1. The PLC network card may not be correctly installed in the host PC.
2. The named device may not be connected to the PLC network.
3. The named device may have been assigned an incorrect Network ID.

Solution:
1. Verify the network card installation using the supplied utility software.
2. Check the PLC network connections.
3. Verify that the Network ID given to the named device matches that of the actual device.

Device Specific Messages

The following error/warning messages may be generated. Click on the link for a description of the message.

Device Specific Messages
Unable to start MBPLUS.SYS device
Unable to communicate with MBPLUS.VXD
Unable to open MBPLUS slave path
Error opening MBPLUS path: <ID>
Bad address in block [<start address> to <end address>] on device '<device name>'

Unable to start MBPLUS.SYS device

Error Type:
Fatal

Possible Cause:
Probable cause is that the MBPLUS.SYS driver was not properly configured.

Solution:
Verify that the MBPLUS device can be started and stopped manually using the Control Panel|Devices applet. Once the MBPLUS.SYS driver can be started manually, the Yaskawa_Memobus_Plus.dll driver will also be able to start the driver.

Unable to communicate with MBPLUS.VXD

Error Type:
Fatal

Possible Cause:
Probable cause is that the MBPLUS.VXD driver was not properly configured or not installed.

Solution:
Install or properly setup the MBPLUS.VXD before running this driver. Proper configuration can be checked using the test programs that come with the VXD driver.

Unable to open MBPLUS slave path

Error Type:
Fatal

Probable Cause:
The driver was unable to open a slave path with the MBPLUS.SYS driver on Windows NT or the MBPLUS.VXD driver on Windows 95. Probable cause is that the MBPLUS driver is not properly installed.

Solution:
Install or properly setup the MBPLUS.VXD before running this driver. Proper configuration can be checked using the test programs that come with the VXD driver.

Error opening MBPLUS path: <ID>

Error Type:
Serious

Possible Cause:
1. The MBPLUS.SYS driver for Windows NT or the MBPLUS.VXD driver for Windows 95 has not been properly configured.
2. The driver cannot open a path on the specified adapter.

Solution:
1. Follow the instructions for installing and configuring the MBPLUS driver.
2. Verify that no more than eight channels are assigned the same adapter number.

Bad address in block [<start address> to <end address>] on device '<device name>'

Error Type:
Serious

Possible Cause:
An attempt has been made to reference a nonexistent location in the specified device.

Solution:
Verify the addresses of all tags assigned to the device and eliminate ones that reference invalid locations.

Index