Motor Logic® Plus

Network Programming Guide

Instruction Bulletin

Retain for future use.

a brand of Schneider Electric

SQUARE D

SECTION 1— INTRODUCTION TO PROGRAMMING THE MOTOR LOGIC PLUS® SSOLR

This instruction bulletin describes how to program the Motor Logic® Plus solid-state overload relay (SSOLR) for use on Modbus® and DeviceNet™ networks.

RELATED DOCUMENTATION

The following documents contain additional information about the Motor Logic Plus SSOLR and the DeviceNet system:

- 30072-450-76*, Motor Logic® Plus Display, 9999MLPD,* provides installation information for the 9999MLPD display and instructions for programming Motor Logic Plus and Motor Logic Plus II solid-state overload relays with the display. **This document is also available in Spanish (30072-451-23) and French (30072-451-22).**
- 30072-451-24*, Motor Logic® Plus Motor Management System 9999MMS*, describes how to display and manage the settings and motor history of a Motor Logic Plus programmable solid-state overload relay (SSOLR).**This document is also available in Spanish (30072-451-25) and French (30072-451-26).**
- 30072-450-95*, Motor Logic® Plus DeviceNet Communication Module, 9999DN and 9999DN2*, contains the installation instructions for the Motor Logic Plus network communication module.
- 30072-450-20, *Motor Logic® Plus Network Communication Module Class 9999 Type MB11 Installation Guide.*
- 30072-013-102, *Motor Logic® Plus Network Communication Module Class 9999 Type MB11 and Type MB22 Programming Guide.*

For copies of these documents, contact your local Schneider Electric field office or obtain them from www.us.SquareD.com (under Technical Library).

IMPORTANT USER INFORMATION

Only qualified persons, as defined in the National Electrical Code (NEC), who are familiar with the installation, maintenance, and operation of this device and the equipment onto which it is to be installed—as well as applicable local, state, and national regulations, industry standards, and accepted practices regarding the safety of personnel and equipment—should be permitted to install, maintain, or operate this device.

These instructions are provided only as a general guide to qualified persons and are not all-inclusive. They do not cover every application or circumstance that may arise in the installation, maintenance, or operation of this equipment. Users are advised to comply with all local, state, and national regulations, industry standards, and acceptable practices regarding safety of personnel and equipment.

ABBREVIATIONS

The following abbreviations are used in this document:

- CT(s): Current Transformer(s)
- DN: DeviceNet Communication Module for Motor Logic Plus
- FVR: Full Voltage Reversing
- FVNR: Full Voltage Non-reversing
- MLP: Motor Logic Plus
- MLP SSOLR: Motor Logic Plus Solid-state Overload Relay
- MMS-D: DeviceNet Motor Management System Display
- MMS-RD: Reverser DeviceNet Motor Management System Display
- SSOLR: Solid-state Overload Relay

PRODUCT SUPPORT

For support and assistance, contact the Product Support Group. The Product Support Group is staffed from 8:00 am until 6:00 pm Eastern time to assist with product selection, start-up, and diagnosis of product or application problems. Emergency phone support is available 24 hours a day, 365 days a year.

Toll Free: 1-888-SquareD (888-778-2733) USA
Toll Free: 1-800-565-6699 Canada

E-mail: control.products.support@us.schneider-electric.com
E-mail: canadian.pss@ca.schneider-electric.com

SECTION 2— PROGRAMMABLE PARAMETERS

NETWORK CONNECTIONS AND HARDWARE REQUIREMENTS

Table 1 describes the network connections possible for the Motor Logic® Plus SSOLR, the hardware requirements for the connections, and associated reference material. Refer to the reference material specified for your application in Table 1 before programming the Motor Logic® Plus SSOLR.

Table 1: Network Connections and Hardware Requirements

Network Connection Type	Programming Hardware Requirements	Reference Material
Modbus network connection	Motor Logic® Plus SSOLR connects directly to a ModBus master through either a 9999MB11 or 9999 MB22 communication module to connect to a ModBus network.	30072-450-96, Section 3
	Motor Logic® Plus SSOLR connects 9999MMS and 9999MMS-R through the use of either a 9999MB11 or 9999 MB22 communication module.	
DeviceNet network connection	9999DN DeviceNet Communication Module mounts directly on a Motor Logic® Plus SSOLR	30072-450-95 30072-450-96, Section 4 30072-451-24
	9999MB22 Communication Module with a 9999MMS-D and 9999MMS-RD	

PARAMETER SETTINGS

⚠ WARNING

IMPROPER OR INADVERTENT SETTINGS

- Take care when modifying parameter settings while the motor is running.
- Disable network programming to prevent unauthorized changes.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Table 2: Parameters Definitions and Default Values

Description	Range/Values	Units	Default
Current multiplier (MULT)		Integer	
SPB4 or SPB6	1		1
SPC4 or SPC6	1		1
SP14 or SP16	1		1
SP24 or SP26	1		1
SP34 or SP36	1		1
SP44 or SP46	10–200		15
SP54 or SP56	10–200		30
SP64 or SP66	10–200		60
Voltage threshold (LV, HV, VUB)			
LV: Low Voltage			
200 – 480 Vac	170 Vac to High Voltage (HV) setting	V	450 Vac
600 Vac	450 Vac to High Voltage (HV) setting	V	550 Vac
HV: High Voltage			
200 – 480 Vac	Low Voltage (LV) setting to 528 V	V	500 V
600 Vac	Low Voltage (LV) setting to 660 V	V	630 V
VUB: Voltage unbalance	2-15%, OFF (999)		5%

Table 2: Parameters Definitions and Default Values *(continued)*

Description	Range/Values	Units	Default
Overcurrent (OC)	Current range of SSOLR	A	Minimum rating
SPB4 or SPB6	0.5–2.3		0.5
SPC4 or SPC6	2–9		2
SP14 or SP16	6–27		6
SP24 or SP26	10–45		10
SP34 or SP36	20–90		20
SP44 or SP46	60–135		60
SP54 or SP56	120–270		120
SP64 or SP66	240–540		240
Undercurrent (UC)	0.5 x OC Min. to OC Max., Off	A	0.5 x Min rating
SPB4 or SPB6	0.25–2.3, 0 = Off		0.25
SPC4 or SPC6	1–9, 0 = Off		1
SP14 or SP16	3–27, 0 = Off		3
SP24 or SP26	5–45, 0 = Off		5
SP34 or SP36	10–90, 0 = Off		10
SP44 or SP46	30–135, 0 = Off		30
SP54 or SP56	60–270, 0 = Off		60
SP64 or SP66	120–540, 0 = Off		120
Current unbalance (CUB)	2–25, or OFF	%	6
Trip class (TC)	5, J5, 10, J10, 15, J15, 20, J20, 30, J30	—	20
Rapid cycle timer (RD1)	0–500	Seconds	10
Cool down timer (RD2)	2–500	Minutes	8
Dry well recovery timer (RD3)	2–500	Minutes	20
Number of restarts after UC (#RU)	0, 1, 2, 3, 4, A	—	0
Number of restarts after other faults (#RF)	0, 1, oc1, 2, oc2, 3, oc3, 4, oc4, A, ocA	—	0
Undercurrent trip delay (UCTD)	2–60	Seconds	5
Ground fault (GF)	1–10 x MULT, Off = 999	A	2
Modbus address	1–99	Integer	1
Modbus communication parameters	C01, C02, C03, C04, C05, C06	—	C01

SETUP

Multiplication Factor (MULT) Register 40441

The multiplication factor (MULT) calculates the true current settings.

For Motor Logic® Plus SSOLRs SPC4, SPC6, SP14, SP16, SP24, SP26, SP34, and SP36, the MULT is always 1 and cannot be changed.

For Motor Logic® Plus SSOLRs SP44, SP46, SP54, SP56, SP64 and SP66, set the MULT equal to the effective turns ratio of the current transformers (CTs) used. For example, with 300:5 CTs and 2 passes through each window on the SSOLR, set MULT as follows:

$$\text{MULT} = 300/5/2 = 30$$

Low Voltage and High Voltage Thresholds (LV/HV)

The low voltage and high voltage settings determine the input voltage range. These setpoints are set to within +/- 10% of the nameplate rating of the motor. The low voltage setpoint cannot be set above the high voltage setpoint.

The appropriate settings for low voltage and high voltage depend on:

- Application
- Environment
- Motor size
- Voltage tolerance of the motor

Consult the motor manufacturer for the high voltage and low voltage setpoints. However, the NEMA MGI standards recommends that the low voltage and high voltage should not be more than +/- 10% of the motor's nameplate voltage. The setting can be determined by multiplying the motor's nameplate voltage by the recommended percent over and under voltage. For example, if the motor nameplate voltage is 230V, set the low voltage to 0.9 x 230 V = 207 V and set the high voltage to 1.10 x 230 V = 253 V. The low voltage setpoint cannot be set higher than the high voltage thresold setting.

Low Voltage Threshold (LV) Register 40438

The low voltage setpoint is determined by the input voltage range of 170 V (450 V for 600 V application) up to the high voltage threshold setting.

High Voltage Threshold (HV) Register 40439

The high voltage setpoint is determined by the low voltage threshold setpoint through 528 V (660 V for 600 V application).

Overcurrent Threshold (OC) Register 40442

⚠ WARNING

IMPROPER PROTECTION

- Set overcurrent threshold to the maximum service factor amperage of the motor.
- Refer to the motor manufacturer's instructions before setting this parameter.

Failure to follow these instructions can result in serious injury or equipment damage.

The overcurrent threshold (OC) sets the trip point of the SSOLR. Set the OC to the maximum service factor amperage of the motor. This value is typically between 110% and 120% of the full load amperage (FLA).

Undercurrent Threshold (UC) Register 40443

The undercurrent threshold (UC) sets the level of acceptable undercurrent. Typically, the UC is set to 80% of the motor's FLA. At 80%, the SSOLR will detect a loss of load for various pumps and motors. To disable the undercurrent protection, set the UC to 0, where 0 = OFF.

Current Unbalance Threshold (CUB) Register 40444

⚠ CAUTION

MOTOR OVERHEATING

- The current unbalance level must be set to protect the wiring and motor equipment.
- The setting must conform to national and local safety regulations and codes.
- Refer to the motor manufacturer's instructions before setting this parameter.

Failure to follow these instructions can result in injury or equipment damage.

This function sets the percentage of the phase current unbalance (CUB) allowed by the motor. Most motor manufacturers recommend restricting motor operation to a CUB of 5% or less, at the service factor amperage. Use the formula below to calculate the setting of the CUB:

$$CUB = \left(\frac{I_{dmax}}{I_{ave}}\right) \times 100$$

CUB = current unbalance (%)
I_{ave} = average current
I_{dmax} = maximum deviation from the average current

NOTE: When CUB = 999, the current unbalance and single-phase current detection are disabled.

Voltage Unbalance Trip Point (VUB) Register 40440

This function sets the percentage of voltage unbalance (VUB) threshold allowed by the motor. The voltage unbalance threshold range is between 2 – 15 % or can be disabled by setting the setpoint to 999. Any motor that is operated above a 1% VUB should be derated. Use the formula below to calculate the setting of VUB.

$$VUB = \left(\frac{V_{dmax}}{V_{ave}}\right) \times 100$$

VUB = voltage unbalance (%)
V_{ave} = average voltage
V_{dmax} = maximum deviation from the average voltage

NOTE: Although Voltage Unbalance Protection is disabled when VUB = 999, single-phase voltage detection remains active. This prevents the motor from starting if the voltage unbalance is greater than 25%.

Trip Class (TC) Register 40445

⚠ CAUTION

MOTOR OVERHEATING

- The trip class adjustment must be set to match the thermal heating characteristics of the motor.
- Refer to the motor manufacturer's instructions before setting this parameter.

Failure to follow these instructions can result in injury or equipment damage.

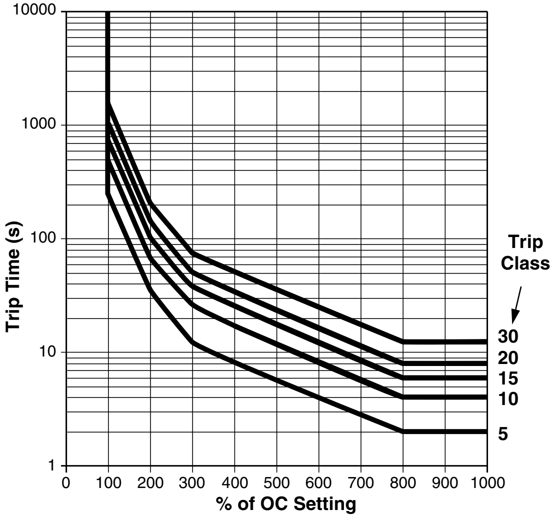
This function selects the trip class (TC) of the SSOLR.[1] The motor and application determine the setting (see Table 3 on page 16). Consult the motor manufacturer if you are unsure of the trip class.

Table 3: Trip Class Definitions

Class	Related Application
5	For small, fractional horsepower motors with almost instantaneous acceleration. An extremely short trip time is required.
10 (fast)	For hermetic refrigerant motors, compressors, submersible pumps, and general-purpose IEC-rated motors that reach rated speed in less than 4 seconds
15	Available for specialized applications
20 (standard)	For most NEMA-rated, general-purpose motors
30 (slow)	For motors with acceleration times longer than 10 seconds, or with high inertia loads
J	For applications that require jam protection. Jam protection is enabled 1 minute after the motor starts. For motors that exceed 400% of the overcurrent limit, the trip time is always 2 seconds, regardless of the assigned trip class. To enable the jam protection feature, see Table 18 on page 40 and Table 64 on page 96.

[1] The trip class is determined by the time (in seconds) it takes for the overload relay to trip when carrying 600% of the overcurrent threshold setting (OC). *UL508, NEMA ICS 2 Part 4.*

Figure 1: Trip Class Curves



Rapid Cycle Timer (RD1) Register 40446

The rapid cycle timer (RD1) controls the elapsed time (in seconds) between motor restarts. To protect the motor from rapid, successive power outages or short cycling caused by the motor controls, set RD1 between 20 and 30 seconds. To allow the motor to start immediately after power-up or after a normal shutdown, set RD1 to 0 seconds.

Motor Cool-Down Timer/Restart Delay (RD2) Register 40447

⚠ CAUTION

MOTOR OVERHEATING

- The thermal reset parameter settings must allow sufficient time for the motor to cool after a thermal fault.
- Refer to the motor manufacturer's instructions before setting this parameter.

Failure to follow these instructions can result in injury or equipment damage.

The setting of RD2 determines the motor cooling time (in minutes) after a fault condition, such as

- A current unbalance (CUB). See page 16.
- A single-phasing fault (SP)
- An overload (when #RF is set to oc1 through ocA). See page 21

Consult the motor manufacturer for the delay required for your motor. Normally, 5 to 10 minutes is sufficient.

NOTE: This SSOLR has a thermal memory feature and an associated thermal capacity available reset level that can inhibit a start after a fault.

For more information, see the "Network Communications Setup Word (Register 40455)" section on page 23.

Dry Well Recovery Timer/Restart Delay after Undercurrent (RD3) Register 40448

This timer determines the motor restart delay (in minutes) after an undercurrent fault. See page 15. It is often used in submersible pumping applications. The setting of RD3 depends on the recovery time of the water well, and varies from application to application.

Number of Restarts after Undercurrent (#RU) Register 40449

⚠ DANGER

AUTOMATIC RESTART HAZARD

- Automatic Restart may only be used for machines or applications that present no danger in the event of automatically restarting, either to personnel or equipment.
- Equipment operation must conform to national and local safety regulations and codes.

Failure to follow these instructions will result in death, serious injury, or equipment damage.

#RU determines the number of successive restart attempts after an undercurrent fault.

- The SSOLR allows up to four successive restart attempts after an undercurrent fault.
- If the #RU is exceeded, the SSOLR trips and a manual restart is required.
- The #RU counter resets after the motor has been running for 1 minute.
- For manual restarts only, set #RU = 0.
- For continuous automatic restarts, set #RU = A.

Number of Restarts after Faults Other Than Undercurrent (#RF) Register 40450

#RF allows you to program the SSOLR for:

- Continuous automatic restarts (#RF = A)
- The number of restarts allowed after a CUB, or SP fault (#RF = 1, 2, 3, or 4). When the #RF counter is greater than this setting, the SSOLR requires a manual or network restart.
- The number of restarts allowed after a CUB, SP, or OC fault (#RF = oc1, oc2, oc3, oc4, or ocA)
- Manual restart only (#RF = 0)

The #RF counter resets after the motor has been running for more than 1 minute.

Undercurrent Trip Delay (UCTD) Register 40451

The undercurrent trip delay (UCTD) determines how long the SSOLR will allow an undercurrent condition before it trips on an undercurrent fault (UC). Typically, UCTD is set to 5 seconds.

Ground Fault Current Threshold (Register 40452)

⚠ DANGER

IMPROPER FAULT PROTECTION

- Ground fault trip levels must be set to protect the wiring and motor equipment.
- Ground fault settings must conform to national and local safety regulations and codes.

Failure to follow these instructions will result in death or serious injury.

*NOTE: **In case of short circuit to ground,** the overload relay will trip on ground fault.*

Table 4: Ground Fault Current Set Points

SSOLR Type	Minimum Set Point	Maximum GF Current
SPB4 or SPB6	0.075 A	1.0 A
SPC4 or SPC6	0.3 A	2.5 A
SP14 or SP16	0.9 A	12 A
SP24 or SP26	1.5 A	16 A
SP34 or SP36	3.0 A	25 A
SP44 or SP46	9.0 A	75 A
SP54 or SP56	18.0 A	150 A
SP64 or SP66	36.0 A	300 A

Modbus Network Watchdog (Register 40199)

To enable or disable the network watchdog on the Motor Logic® Plus, send commands to the COMLINE register (Location 0xC6, Register 40199).

Table 5: COMLINE Network Watchdog Commands

07H	Network Watchdog Enable
08H	Network Watchdog Disable

If an Motor Logic® Plus Display or MMS is used, be sure to enable the watchdog feature on the keypad as well. See the keypad user manuals for details.

Network Communications Setup Word (Register 40455)

⚠ WARNING

IMPROPER COMMUNICATION PORT USAGE

- Communication ports should be used for non-critical data transfers only.
- Contactor status and current level values reported over the communication ports are delayed by transmission time and should not be used for critical control decisions.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

The Motor Logic® Plus SSOLR Modbus port can be configured to the settings described in Table 6. When configuring network communication, the values shown in the first column are written to the register to program the corresponding baud rate, parity, and stop bit.

Table 6: Network Communication Setup Word Bit Description

Value	Setting	Baud Rate	Parity	Stop Bit
1	C01	19200	Even	1
2	C02	19200	Odd	1
3	C03	19200	None	1
4	C04	9600	Even	1

Fault Relay (Register 40436)

The COMLINE register can be used to open or close the fault relay by writing the following values:

- 01H: Start/Reset
- 02H: Stop

Raw Average Current (Register 40418)

The Raw Average Current (Register 40418) contains the average value of the three phase currents A, B, and C.

Current Unbalance (Register 40420)

This register contains the calculated value of the current unbalance between the three phase currents.

Raw Current Phase C (Register 40430)

This register contains the value of the raw current flowing in phase C.

Raw Current Phase B (Register 40431)

This register contains the value of the raw current flowing in phase B.

Raw Current Phase A (Register 40432)

This register contains the value of the raw current flowing in phase A.

Raw Average Voltage (Register 40417)

This register contains the average value of the three-phase line voltages: A-B, B-C, C-A.

Voltage Unbalance (Register 40419)

This register contains the calculated value of the voltage unbalance between the three-phase voltages as previously defined in this document.

Raw Line Voltage VA-C (Register 40427)

This register contains the value of the line voltage between A-C.

Raw Line Voltage VB-C (Register 40428)

This register contains the value of the line voltage between B-C.

Raw Line Voltage VA-B (Register 40429)

This register contains the value of the line voltage between A-B.

SECTION 3— MODBUS NETWORK COMMUNICATION

INTRODUCTION

The Motor Logic® Plus family of products supports reading and writing to holding registers. Input points, output points, data, and setpoints are all treated as holding registers. 03 block read command, 06 single register write command, and a modified 16 multiple register write command are supported. The 16 multiple register write command can only be used to write to a single register.

MODBUS MEMORY ADDRESSING / REGISTER MAPPING

The Modbus uses two memory numbering schemes. The Modicon Modbus standard refers to registers and addresses, whereas the standard available from Modbus.org only refers to addresses.

If you are writing Ladder Logic for a PLC or using other Modbus tools that are written around the Modicon standard, register numbering is typical. The prefixes (references) are often used to automatically select which command set is used to read or write a memory location (coil, input or holding register).

If you are writing embedded code for a micro controller or your own Modbus code, use the address numbering system. The Address is used by high level tools such as PLC programming software.

Modicon Modbus Standard

The Modicon Modbus standard uses two numbering systems for memory.

The register map consists of different register types, such as coil (0X references), input (1X references) and holding registers (4X references). Each register type uses a different set of commands for reading and writing.

The actual address sent in the bus message is one less than the register number minus the reference prefix (see the following discussion). Register numbering starts at 1 plus a reference prefix (for example, 400001 for holding registers) and addresses start at 0.

Data Addresses in Modbus Messages (from Modicon Modbus Protocol Reference Guide PI-MBUS-300 Rev J)

All data addresses in Modbus messages are referenced to zero. The first occurrence of a data item is addressed as item number zero. For example, the coil known as “coil 1” in a programmable controller is addressed as “coil 0000” in the data address field of a Modbus message. Coil 127 decimal is addressed as “coil 007E” hex (126 decimal).

Holding register 40001 is addressed as “register 0000” in the data address field of the message. The function code field already specifies a holding register operation. Therefore, the “4XXXX” reference is implicit. Holding register 40108 is addressed as “register 006B” hex (107 decimal).

Modbus.org Standard

Another version of the Modbus standard is available at Modbus.org. This standard only uses one memory addressing system and does not contain references to registers as distinct from addresses or references to different register types.

SUPPORTED MODBUS COMMANDS

03 Read Holding Registers (from Modicon Modbus Standard)

This command reads the binary contents of the holding registers (4X references) in the slave. Broadcast is not supported. Appendix B lists the maximum parameters supported by various controller models.

Query

The query message specifies the starting register address and the quantity of registers to be read. Registers are addressed starting at zero, so registers 1–16 are addressed as “0-15”.

Table 7 on page 29 provides an example of a request to read registers 40108–40110 from a slave device.

Table 7: Read Holding Registers Query

Field Name	Example (Hex)
Slave Address	11
Function	03
Starting Address Hi	00
Starting Address Lo	6B
No. of Points Hi	00
No. of Points Lo	03
Error Check (LRC or CRC)	—

Response

The register data in the response message are packed as two bytes per register, with the binary contents right justified within each byte. For each register, the first byte contains the high order bits and the second contains the low order bits.

Data is scanned in the slave at the rate of 125 registers per scan for 984–X8X controllers (984–685), and at the rate of 32 registers per scan for all other controllers. The response is returned when the data is completely assembled.

Table 8 provides an example of a response to the query in Table 7.

Table 8: Read Holding Registers—Response

Field Name	Example (Hex)
Slave Address	11
Function	03
Byte Count	06
Data Hi (Register 40108)	02
Data Lo (Register 40108)	2B
Data Hi (Register 40109)	00
Data Lo (Register 40109)	00
Data Hi (Register 40110)	00
Data Lo (Register 40110)	64
Error Check (LRC or CRC)	—

The contents of register 40108 are shown as the two byte values of 02 2B hex, or 555 decimal. The contents of registers 40109–40110 are 00 00 and 00 64 hex, or 0 and 100 decimal.

03 (0x03) Read Holding Registers (from Modbus.org Standard)

This function code is used to read the contents of a contiguous block of holding registers in a remote device. The Request PDU specifies the starting register address and the number of registers to be read. Registers are addressed starting at zero, so registers 1–16 are addressed as “0-15”.

The register data in the response message are packed as two bytes per register, with the binary contents right justified within each byte.

For each register, the first byte contains the high order bits and the second contains the low order bits.

Request

Function Code	1 Byte	**0x03**
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Registers	2 Bytes	1 to 125 (0x7D)

Response

Function Code	1 Byte	**0x03**
Byte Count	2 Bytes	2 x **N**[1]
Register Value	**N**[1] x 2 bytes	

[1] N = Quantity of registers.

Error

Error Code	1 Byte	**0x83**
Exception Code	1 Byte	01 or 02 or 03 or 04

Table 9 illustrates a request to read registers 108–110.

Table 9: Request to Read Registers 108–110

Request		Response	
Field Name	**(Hex)**	**Field Name**	**(Hex)**
Function	03	Function	03
Starting Address Hi	00	Byte Count	06
Starting Address Lo	6B	Register Value Hi (108)	02
No. of Registers Hi	00	Register Value Lo (108)	2B
No. of Registers Lo	03	Register Value Hi (109)	00
		Register Value Lo (109)	00
		Register Value Hi (110)	00
		Register Value Lo (110)	64

The contents of register 108 are shown as the two byte values of 02 2B hex, or 555 decimal. The contents of registers 109–110 are 00 00 and 00 64 hex, or 0 and 100 decimal, respectively.

06 Preset (Write) Single Register (From Modicon Modbus Standard)

This function code preset writes a value into a single holding register (4XXXX reference). When broadcast, the function code presets the same register reference in all attached slaves.

NOTE: This function overrides the controller's memory protect state. The preset value remains valid in the register until the next time the controller's logic solves the register contents. The register's value will remain if it is not programmed in the controller's logic.

Query

The query message specifies the register reference to be preset. Registers are addressed starting at zero. For example, register 1 is addressed as “0”.

Table 10 on page 32 provides an example of a request to preset (write) register 40002 to 00 03 hex, in slave device 11 hex.

Table 10: Preset (Write) Single Register—Query

Field Name	Example (hex)
Slave Address	11
Function	06
Register Address Hi	00
Register Address Lo	01
Preset Data Hi	00
Preset Data Lo	03
Error Check (LRC or CRC)	—

Response

The normal response is an echo of the query, returned after the register contents have been preset. Table 11 provides an example of a response to the query in Table 10.

Table 11: Preset (Write) Single Register—Response

Field Name	Example (hex)
Slave Address	11
Function	06
Register Address Hi	00
Register Address Lo	01
Preset Data Hi	00
Preset Data Lo	03
Error Check (LRC or CRC)	—

06 (0x06) Preset (Write) Single Register (from Modbus.org Standard)

This function code is used to write a single holding register in a remote device. The Request PDU specifies the address of the register to be written. Registers are addressed starting at zero. For example, register 1 is addressed as “0.”

The normal response is an echo of the request, returned after the register contents have been written.

Request

Function Code	1 Byte	**0x06**
Register Address	2 Bytes	0x0000 to 0xFFFF
Registers Value	2 Bytes	0x0000 to 0xFFFF

Response

Function Code	1 Byte	**0x06**
Register Address	2 Bytes	0x0000 to 0xFFFF
Registers Value	2 Bytes	0x0000 to 0xFFFF

Error

Error Code	1 Byte	**0x86**
Exception Code	1 Byte	01 or 02 or 03 or 04

Table 12 provides an example of a request to write register 2 to 00 03 Hex.

Table 12: Request to Preset (Write) Register 2 to 00 03 Hex

Request		Response	
Field Name	**(Hex)**	**Field Name**	**(Hex)**
Function	06	Function	06
Register Address Hi	00	Register Address Hi	00
Register Address Lo	01	Register Address Lo	01
Register Value Hi	00	Register Value Hi	00
Register Value Lo	03	Register Value Lo	03

16 (10 Hex) Preset (Write) Multiple Registers (from Modicon Modbus Standard)

This function code preset writes values to a sequence of holding registers (4XXXX references). When broadcast, the function presets the same register references in all attached slaves.

NOTE: This function overrides the controller's memory protect state. The preset values remain valid in the registers until the next time the controller's logic solves the register contents. The register values will remain if they are not programmed in the controller's logic.

Appendix B, beginning on page 93 lists the maximum parameters supported by various controller models.

Query

The query message specifies the register references to be preset. Registers are addressed starting at zero. For example, register 1 is addressed as "0".

The requested preset (write) values are specified in the query data field.

Table 13 provides an example of a request to preset two registers starting at 40002 to 00 0A and 01 02 hex, in slave device 17.

Table 13: Preset (Write) Multiple Registers—Query

Field Name	Example (Hex)
Slave Address	11
Function	10
Starting Address Hi	00
Starting Address Lo	01
No. of Registers Hi	00
No. of Registers Lo	02
Byte Count	04
Data Hi	00
Data Lo	0A
Data Hi	01
Data Lo	02
Error Check (LRC or CRC)	—

Response

The normal response returns the slave address, function code, starting address, and quantity of registers preset.

Table 14 provides an example of a response to the query shown in Table 13 on page 32.

Table 14: Preset (Write) Multiple Registers

Field Name	Example (Hex)
Slave Address	11
Function	10
Starting Address Hi	00
Starting Address Lo	01
No. of Registers Hi	00
No. of Registers Lo	02
Error Check (LRC or CRC)	—

16 (0x10) Preset (Write) Multiple Registers (from Modbus.org Standard)

This function code is used to write a block of contiguous registers (approximately 1 to 120) in a remote device. However, in the Motor Logic Plus SSOLR only a single register may be written at a time.

The requested written values are specified in the request data field. Data is packed as one byte per register, with the binary contents right justified within each byte.

The normal response returns the function code, starting address, and quantity of registers written.

Request

Function Code	1 Byte	**0x10**
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Registers	2 Bytes	0x0001 to 0x0078
Byte Count	1 Byte	2 x **N**[1]
Registers Value	N x 2 Bytes	Value

[1] N = Quantity of registers.

Response PDU

Function Code	1 Byte	**0x10**
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Registers	2 Bytes	(0x7B)

Error

Error Code	1 Byte	**0x90**
Exception Code	1 Byte	01 or 02 or 03 or 04

Table 15 on page 37 provides an example of a request to write two registers starting at 2 to 00 0A and 01 02 Hex.

Table 15: Request to Write Two Registers

Request		Response	
Field Name	**Example (Hex)**	**Field Name**	**Example (Hex)**
Function	**10**	Function	**10**
Starting Address Hi	**00**	Starting Address Hi	**00**
Starting Address Lo	**01**	Starting Address Lo	**01**
Quantity of Registers Hi	**00**	Quantity of Registers Hi	**00**
Quantity of Registers Lo	**02**	Quantity of Registers Lo	**02**
Byte Count	**04**	—	—
Registers Value Hi	**00**	—	—
Registers Value Lo	**0A**	—	—
Registers Value Hi	**01**	—	—
Registers Value Lo	**02**	—	—

REGISTER MAP

Table 16: Read Only Registers[2]

<table>
<tr><th colspan="2">16 Bit Memory</th><th rowspan="2">Code and Description</th><th rowspan="2" colspan="4">Notes[1]</th></tr>
<tr><th>Hex</th><th>Register</th></tr>
<tr><td>19E</td><td>40415</td><td>MMSCMDSTAT
MMS command status</td><td colspan="4">See Appendix B beginning on page 93 for MMS Memory Map</td></tr>
<tr><td>19F</td><td>40416</td><td>MMSOPRSTAT
MMS operating status</td><td colspan="4">See Appendix B beginning on page 93 for MMS Memory Map</td></tr>
<tr><td>1A1</td><td>40418</td><td>IAVE
Raw average current</td><td colspan="4">A (x100, x10, x1) multiplied by scale factor for unit display</td></tr>
<tr><td>1A3</td><td>40420</td><td>CUB
Current unbalance</td><td colspan="4">0–100%</td></tr>
<tr><td>1A5</td><td>40422</td><td>THERM CAP
Thermal capacity remaining</td><td colspan="4">0–100%</td></tr>
<tr><td>1A6</td><td>40423</td><td>GF
Ground fault current</td><td colspan="4">A (x100, x10, x1), multiplied by scale factor for unit display</td></tr>
<tr><td rowspan="9">1A7</td><td rowspan="9">40424</td><td rowspan="9">TRIPRN/ERCODE
Bit real-time errors and trip indicator</td><td>Bit #</td><td>TRIPRN</td><td>Bit #</td><td>ERCODE</td></tr>
<tr><td>0</td><td>Fault lockout</td><td>8</td><td>Low voltage</td></tr>
<tr><td>1</td><td>Remote stop</td><td>9</td><td>High voltage</td></tr>
<tr><td>2</td><td>Contactor failure</td><td>10</td><td>Unbalance voltage</td></tr>
<tr><td>3</td><td>Undercurrent</td><td>11</td><td>Undercurrent</td></tr>
<tr><td>4</td><td>Overcurrent</td><td>12</td><td>Phase reversal</td></tr>
<tr><td>5</td><td>Ground fault</td><td>13</td><td>CUB</td></tr>
<tr><td>6</td><td>CUB</td><td>14</td><td>CUB > 25%</td></tr>
<tr><td>7</td><td>CUB >50%</td><td>15</td><td>CUB >50%</td></tr>
</table>

1 For registers that have bit information, reading a “1” in any bit means that the statement for that bit alone is true.

2 Do not write to reserved bit locations.

3 The four-fault history is based on the scheme where Bits 0–3 = Last Fault, Bits 4–7 = 2nd from the last fault, Bits 8–11 = 3rd from the last fault, and Bits 12–15 = 4th from the last fault.

4 Can be viewed via network

Table 16: Read Only Registers[2] *(continued)*

16 Bit Memory Hex	16 Bit Memory Register	Code and Description	Notes[1]	
1A8	40425	FH NIBBLE_CODED 4-fault history[3] These four nibble-bits indicate a hex value that corresponds to the following faults.	Hex Value	Fault
			01	*High voltage*
			02	*Low voltage*
			03	N/A
			04	*Contactor failure*
			05	N/A
			06	Single phase
			07	Ground fault
			08	Current unbalance
			09	N/A
			A	Overcurrent
			B	Undercurrent
1A9	40426	PID 4-bit multiplier (scale factor), and 4-bit product ID	Bits 0–3 = Motor Logic® Plus product ID, which is broken down below; Bits 4–7 = Multiplier (1 = 100; 2 = 10; 4 = 1); Bit 15 = 0 = MLP, SPXX	
			Value of bits 3–0 (decimal/binary)	SSOLR Model
			1/0001	SPB4
			2/0010	SPB6
			3/0011	SPC4
			4/0100	SPC6
			5/0101	SP14
			6/0110	SP16
			7/0111	SP24
			8/1000	SP26
			9/1001	SP34
			10/1010	SP36
			11/1011	SP44, SP54, and SP64
			12/1100	SP46, SP56, and SP66
1A0	40417	VOLTAV Average voltage L-L	Volts	

1 For registers that have bit information, reading a “1” in any bit means that the statement for that bit alone is true.

2 Do not write to reserved bit locations.

3 The four-fault history is based on the scheme where Bits 0–3 = Last Fault, Bits 4–7 = 2nd from the last fault, Bits 8–11 = 3rd from the last fault, and Bits 12–15 = 4th from the last fault.

4 Can be viewed via network

Table 16: Read Only Registers[2] *(continued)*

16 Bit Memory Hex	Register	Code and Description	Notes[1]
1A2	40419	VUB[4] Voltage unbalance	0-100%
1A4	40421	PFANGLE[4] Power factor angle	Degrees
1AA	40427	VA-C Line voltage A-C	Volts
1AB	40428	VB-C Line voltage B-C	Volts
1AC	40429	VA-B Line voltage A-B	Volts
1AD	40430	IC Raw current phase C	A (x100, x10, x1) multiplied by scale factor for unit display
1AE	40431	IB Raw current phase B	A (x100, x10, x1) multiplied by scale factor for unit display
1AF	40432	IA Raw current phase A	A (x100, x10, x1) multiplied by scale factor for unit display
1B0	40433	RD1 Remaining restart delay RD1	Seconds
1B1	40434	RD2 Remaining restart delay RD2	Minutes
1B2	40435	RD3 Remaining restart delay RD3	Minutes
1B4	40437	Scale PowerLogic scale parameter	0, 1, 1 2; 16-bit signed word (2's complement) Default is model dependent 0000 = 1 FFFF = 10 FFFE = 100 Default is model dependent SPB4 or SPB6 = 100x SPC4 or SPC6 = 100x SP14 or SP16 = 10x SP24 or SP26 = 10x SP34 or SP36 = 10x SP44 or SP46 = 1x SP54 or SP56 = 1x SP64 or SP66 = 1x

1 For registers that have bit information, reading a "1" in any bit means that the statement for that bit alone is true.
2 Do not write to reserved bit locations.
3 The four-fault history is based on the scheme where Bits 0–3 = Last Fault, Bits 4–7 = 2nd from the last fault, Bits 8–11 = 3rd from the last fault, and Bits 12–15 = 4th from the last fault.
4 Can be viewed via network

CAUTION

EXCEEDING THE STORAGE LIMIT

- Set points in Motor Logic® Plus SSOLRs are stored in a nonvolatile EEPROM. The EEPROM retains the set points in its memory for years without power applied.
- The network may read the EEPROM register locations an unlimited number of times. Limit the network register writes to set point changes only.
- Do not exceed 100,000 network register write cycles. This will permanently damage the Motor Logic® Plus EEPROM.

Failure to follow these instructions can result in equipment damage.

Table 17: Read/Write Registers

16 Bit Memory		Code and Description	Range	Default
Hex	Register			
1B3	40436	COM-LINE Command line	01H: Start/reset 02H: Stop 03H: Display lock 04H: Display unlock 05H: Network program enable 06H: Network program enable 07H: Network program enable 08H: Network program enable 09H: Clear motor run timer 0AH: Clear last fault	
1B5	40438	LV Low voltage threshold	HV Setting - 170 V (450 V for 600 V application)	435 (550)
1B6	40439	HV High voltage threshold	LV setting - 528 V (660 V for 600 V application)	500 (630)
1B7	40440	VUB Voltage unbalance threshold	2-15% or (999)	5%

Table 17: Read/Write Registers *(continued)*

16 Bit Memory		Code and Description	Range	Default
Hex	Register			
1B8	40441	MULT CT/turns effective ratio	1 or 10–200	Sizes B–3: 1 Size 4: 15 Size 5: 30 Size 6: 60
1B9	40442	OC Overcurrent threshold	OC current range	Min. rating
1BA	40443	UC Undercurrent threshold	0.5 x OC Min. to OC Max., Off	0.5 x OC Min.
1BB	40444	CUB Current unbalance threshold	2–25% or Off (999)	6%
1BC	40445	TC Overcurrent trip class	5, J5, 10, J10, 15, J15, 20, J20, 30, J30 *(J = Jam protection enabled)* **Class** / **Decimal** 5 / 5 J5 / 133 10 / 10 J10 / 138 15 / 15 J15 / 143 20 / 20 J20 / 148 30 / 30 J30 / 158	20
1BD	40446	RD1 Rapid cycle timer	0–500 seconds	10
1BE	40447	RD2 Restart delay after all faults except undercurrent	2–500 minutes	8
1BF	40448	RD3 Restart delay after undercurrent	2–500 minutes	20

Table 17: Read/Write Registers *(continued)*

16 Bit Memory		Code and Description	Range	Default
Hex	Register			
1C0	40449	#RU Number of restarts after undercurrent	0, 1, 2, 3, 4, A (automatic) **#RU Values** 0, 1, 2, 3, 4 or A = 5	0
1C1	40450	#RF Number of restarts after all faults except undercurrent	0, 1, oc1, 2, oc2, 3, oc3, 4, oc4, A, ocA 0 = manual, A = continuous, oc = automatic restart after RD2 expires **RF Value** / **Decimal Value** 0 / 1 1 / 2 oc1 / 3 2 / 4 oc2 / 5 3 / 6 oc3 / 7 4 / 8 oc4 / 9 A / 10 ocA / 11	0
1C2	40451	UCTD Undercurrent trip delay	2–60 seconds	5
1C3	40452	GF Ground fault current threshold	1–10 x MULT, 999 = Off	2A
1C4	40453	MbADDR RS-485 slave address	1–99	1
1C5	40454	MTIM Motor run timer	0–65,000 hours	0

MODBUS COMMUNICATION TROUBLESHOOTING

Refer to Tables 18 and 19 (pages 41–42) for Modbus network communication troubleshooting.

Table 18: Modbus Network System Using Solutions Software or PowerLogic SMS-3000 Software

Symptom	Problem	Solution
Cannot find one or more nodes on the Modbus network.	The address is set improperly on the SSOLR. Duplicate addresses cause communication errors.	Set the address correctly on the SSOLR.
	The wiring to the SSOLR is incorrect.	Check the wiring. Refer to the SSOLR instruction bulletin, 30072-013-98.
	The RS-232 comm port is not sending data.	Ensure that the RS-232 comm port settings match the network settings (baud rate, parity, stop bits, and data bits). Test the RS-232 comm port on the PC, following the procedures below. You will need a voltmeter and the 9999MLPS Solutions software. 1. Exit the Solutions software. Leave the RS-485 converter plugged into the RS-232 port on your PC, but unplug the communication module from the SSOLR. Halt all communication on the RS-232 port. 2. Set the voltmeter to the 10 V ⎓ range. Measure the voltage on terminals A and B of the communication module. Be sure to ground the (–) probe of the voltmeter. No voltage should be present at terminals A or B on the communication module. If voltage is present on either terminal, another program may be using the RS-232 port on the PC, preventing communication to the network. 3. Start the Solutions software and wait for the Comm Port Configuration screen to display. The Comm Port status should show that the port is closed (there still should be no voltage at terminals A or B on the communication module). 4. Click the **OK** button to open the RS-232 port. If the Solutions software cannot find the RS-232 port, run-time error 8002 (Invalid port number) is displayed. If the port is busy, run-time error 8005 (Port already open) is displayed. 5. After the RS-232 port is opened, the Main Menu is displayed. Use the positive (+) probe of the voltmeter to verify that the voltages are: • 4 V ⎓ at terminal RD (B) of the RS-485 converter and at terminal A of the communication module. • 0 V ⎓ at terminal B of the communication module. If the voltages are reversed on the communication module (A = 0 V ⎓ and B = 4 V ⎓), the polarity is not correct. Switch the transmit pair and the receive pair. If the measurement on terminal A of the communication module is not 4 V ⎓ , the Solutions software will display a time-out error and you will not be able to communicate with the SSOLR.

Table 18: Modbus Network System Using Solutions Software or PowerLogic SMS-3000 Software *(continued)*

Symptom	Problem	Solution
Cannot communicate with one or more nodes on the Modbus network.	Modbus port settings are set incorrectly on the PC.	Check the PC serial port. Refer to the SSOLR bulletin, 30072-013-98. Ensure that the communication parameters are set correctly.
	The wiring connection for the RS-232 to RS-485 converter is incorrect.	Check the control wiring scheme. Refer to the SSOLR bulletin, 30072-013-98.
	The communications parameter settings are incorrect.	Verify that the Modbus master device settings match the network settings (baud rate, parity, stop bits, and data bits).

Table 19: Modbus Network Connected to a PLC

Symptom	Problem	Solution
Cannot find one or more nodes on the Modbus network.	The address is set improperly on the SSOLR. Duplicate addresses cause communication errors.	Set the address correctly on the SSOLR.
	The wiring to the SSOLR is incorrect.	Check the wiring. Refer to the SSOLR bulletin, 30072-013-98.
	The network cabling is disconnected or broken.	Verify that the wires are securely connected to the network adapters. If the connections are OK, check the network wires. If necessary, use an ohmmeter to verify the continuity of each wire.
	The polarity in the connection between the SSOLR and the RS-485 network is incorrect.	Verify the network connections based on the wiring schematic. If you have a custom-designed installation that does not use the RS-232 to RS-485 converter, try switching the connections at terminals A and B. If possible, send a Start or Stop command from the Modbus master device to the SSOLR; if the transmit pair polarity is correct, the Start and Stop commands will be executed whether or not the SSOLR returns an acknowledgement.
Cannot communicate with one or more nodes on the Modbus network.	Incorrect communication parameter settings.	Configure the network parameters for the Modbus port of the PLC to match the Motor Logic Plus network parameters.
	The wiring connections are incorrect.	Check the wiring. Refer to the SSOLR bulletin, 30072-013-98.
	Software parameters within the PLC program are incorrect.	Ensure that the communication parameters are set correctly.

SECTION 4— DEVICENET NETWORK COMMUNICATION[1]

DeviceNet Overview (DeviceNet Specification, Volume 1, Chapter 1)

The DeviceNet low-level controller-area network (CAN) provides a communication link between simple industrial devices (such as actuators and sensors) and controlling devices. The network transports the control information and the properties of the device being controlled. It permits operation in either master/slave or peer-to-peer mode.

The DeviceNet powered four-wire network operates in a trunkline dropline configuration and supports up to 64 nodes.

Object Modeling (DeviceNet Specification, Volume 2, Chapter 1)

A DeviceNet node is modeled as a collection of **objects.** An object represents a product component. Objects are organized as follows:

- **Classes** are groups of objects with shared characteristics.
- **Instances** are the objects within the class**.** Each instance within a class has the same set of **behaviors** and **attributes,** but a unique set of attribute values.
- **Attributes** are characteristics of objects represented by values, which can vary. Attributes may provide status information or determine the object's **behaviors.**
- **Behaviors** are actions that an object takes in response to particular events.
- **Services** are operations that an object performs.
- **Instantiations** are creations of new object instances. Attribute values default to 0 unless otherwise specified.

Object Identification (DeviceNet Specification, Volume 2, Chapter 1)

Objects are identified using the following codes:

- The **MAC ID** distinguishes each node from all other nodes on the same network.
- The **Class ID** distinguishes each object class from all other classes accessible from the network.

[1] Contact the Open DeviceNet Vendor Association, Inc. for information and complete specifications.

- The **Instance ID**, assigned to an object instance when it is created, distinguishes it from all other objects within the same class.
- The **Attribute ID** distinguishes each attribute from all other attributes of the same object.

Accessing Variables

Variables on the node are accessed using a key, called a Path, which is composed of:

- The Class ID
- The Instance ID
- The Attribute ID

The classes available in the SSOLR are grouped into three parts:

- Classes required for all equipment connected to the DeviceNet network, whatever their functionality
- Classes relating to the overload relay profile, as defined by the ODVA
- Classes relating to the Motor Logic® Plus SSOLR, allowing access to all internal variables: configuration, adjustment, monitoring, etc.

Definition of Overload Profile (DeviceNet Specifications, Volume 2, Chapter 3-9)

The ODVA designates SSOLRs as device type 03_{hex}.

In accordance with this profile, the SSOLRs are interchangeable on the DeviceNet network with respect to their basic control services. However, the SSOLR must be configured through some means outside the DeviceNet interface, such as a network configuration tool or the Motor Logic® Plus Display.

SOFTWARE CONFIGURATION

Software Configuration Using EDS Files and ICO Files

Software configuration of the DeviceNet network and the associated DeviceNet master requires use of the DeviceNet electronic data sheet (EDS) files, listed in Table 20.

The icon (ICO) files personalize the configuration software with a picture representation of the Motor Logic® Plus SSOLR.

The EDS and ICO files are available from Schneider Electric's Intelligent Motor Control Center web site, www.us.SquareD.com. Upload the EDS files to your system to access relevant files.

Table 20: EDS and ICO File References

SSOLR Catalog Number	EDS File Name	ICO File Name	Description
SPB4	SPB4_xx-xx-xx.eds	SPB4_xx-xx-xx.ico	MLP relay
SPC4	SPC4_xx-xx-xx.eds	SPC4_xx-xx-xx.ico	MLP relay
SP14	SP14_xx-xx-xx.eds	SP14_xx-xx-xx.ico	MLP relay
SP24	SP24_xx-xx-xx.eds	SP24_xx-xx-xx.ico	MLP relay
SP34	SP34_xx-xx-xx.eds	SP34_xx-xx-xx.ico	MLP relay
SP44	SP44_xx-xx-xx.eds	SP44_xx-xx-xx.ico	MLP relay
SP54	SP54_xx-xx-xx.eds	SP54_xx-xx-xx.ico	MLP relay
SP64	SP64_xx-xx-xx.eds	SP64_xx-xx-xx.ico	MLP relay
SPB6	SPB6_xx-xx-xx.eds	SPB6_xx-xx-xx.ico	MLP relay
SPC6	SPC6_xx-xx-xx.eds	SPC6_xx-xx-xx.ico	MLP relay
SP16	SP16_xx-xx-xx.eds	SP16_xx-xx-xx.ico	MLP relay
SP26	SP26_xx-xx-xx.eds	SP26_xx-xx-xx.ico	MLP relay
SP36	SP36_xx-xx-xx.eds	SP36_xx-xx-xx.ico	MLP relay
SP46	SP46_xx-xx-xx.eds	SP46_xx-xx-xx.ico	MLP relay
SP56	SP56_xx-xx-xx.eds	SP56_xx-xx-xx.ico	MLP relay
SP66	SP66_xx-xx-xx.eds	SP66_xx-xx-xx.ico	MLP relay
MMS-D w/SPB4	SPB4_xx-x-xx.eds	SPB4_xx-xx-xx.ico	MLP with MMS-D
MMS-D w/SPC4	SPC4_xx-x-xx.eds	SPC4_xx-xx-xx.ico	MLP with MMS-D
MMS-D w/SP14	SP14_xx-x-xx.eds	SP14_xx-xx-xx.ico	MLP with MMS-D
MMS-D w/SP24	SP24_xx-x-xx.eds	SP24_xx-xx-xx.ico	MLP with MMS-D
MMS-D w/SP34	SP34_xx-x-xx.eds	SP34_xx-xx-xx.ico	MLP with MMS-D

Table 20: EDS and ICO File References *(continued)*

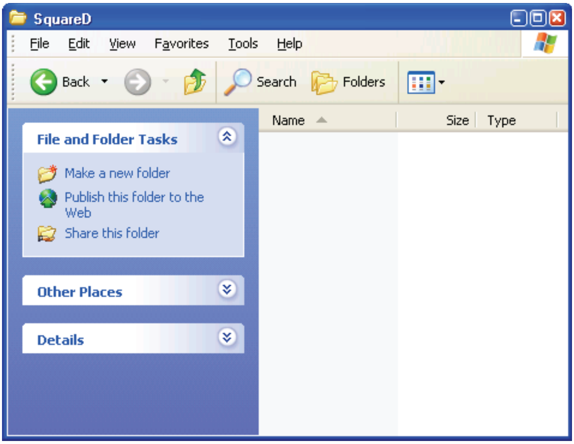
SSOLR Catalog Number	EDS File Name	ICO File Name	Description
MMS-D w/SP44	SP44_xx-x-xx.eds	SP44_xx-xx-xx.ico	MLP with MMS-D
MMS-D w/SP54	SP54_xx-x-xx.eds	SP54_xx-xx-xx.ico	MLP with MMS-D
MMS-D w/SP64	SP64_xx-x-xx.eds	SP64_xx-xx-xx.ico	MLP with MMS-D
MMS-D w/SPB6	SPB6_xx-x-xx.eds	SPB6_xx-xx-xx.ico	MLP with MMS-D
MMS-D w/SPC6	SPC6_xx-x-xx.eds	SPC6_xx-xx-xx.ico	MLP with MMS-D
MMS-D w/SP16	SP16_xx-x-xx.eds	SP16_xx-xx-xx.ico	MLP with MMS-D
MMS-D w/SP26	SP26_xx-x-xx.eds	SP26_xx-xx-xx.ico	MLP with MMS-D
MMS-D w/SP36	SP36_xx-x-xx.eds	SP36_xx-xx-xx.ico	MLP with MMS-D
MMS-D w/SP46	SP46_xx-x-xx.eds	SP46_xx-xx-xx.ico	MLP with MMS-D
MMS-D w/SP56	SP56_xx-x-xx.eds	SP56_xx-xx-xx.ico	MLP with MMS-D
MMS-D w/SP66	SP66_xx-x-xx.eds	SP66_xx-xx-xx.ico	MLP with MMS-D
MMS-RD w/SPB4	SPB4_xx-x-xx.eds	SPB4_xx-xx-xx.ico	MLP with MMS-RD
MMS-RD w/SPC4	SPC4_xx-x-xx.eds	SPC4_xx-xx-xx.ico	MLP with MMS-RD
MMS-RD w/SP14	SP14_xx-x-xx.eds	SP14_xx-xx-xx.ico	MLP with MMS-RD
MMS-RD w/SP24	SP24_xx-x-xx.eds	SP24_xx-xx-xx.ico	MLP with MMS-RD
MMS-RD w/SP34	SP34_xx-x-xx.eds	SP34_xx-xx-xx.ico	MLP with MMS-RD
MMS-RD w/SP44	SP44_xx-x-xx.eds	SP44_xx-xx-xx.ico	MLP with MMS-RD
MMS-RD w/SP54	SP54_xx-x-xx.eds	SP54_xx-xx-xx.ico	MLP with MMS-RD
MMS-RD w/SP64	SP64_xx-x-xx.eds	SP64_xx-xx-xx.ico	MLP with MMS-RD
MMS-RD w/SPB6	SPB6_xx-x-xx.eds	SPB6_xx-xx-xx.ico	MLP with MMS-RD
MMS-RD w/SPC6	SPC6_xx-x-xx.eds	SPC6_xx-xx-xx.ico	MLP with MMS-RD
MMS-RD w/SP16	SP16_xx-x-xx.eds	SP16_xx-xx-xx.ico	MLP with MMS-RD
MMS-RD w/SP26	SP26_xx-x-xx.eds	SP26_xx-xx-xx.ico	MLP with MMS-RD
MMS-RD w/SP36	SP36_xx-x-xx.eds	SP36_xx-xx-xx.ico	MLP with MMS-RD
MMS-RD w/SP46	SP46_xx-x-xx.eds	SP46_xx-xx-xx.ico	MLP with MMS-RD
MMS-RD w/SP56	SP56_xx-x-xx.eds	SP56_xx-xx-xx.ico	MLP with MMS-RD
MMS-RD w/SP66	SP66_xx-x-xx.eds	SP66_xx-xx-xx.ico	MLP with MMS-RD

Registering EDS Files

Follow the procedure in this section to register the EDS and ICO files listed in Table 20 on page 49.

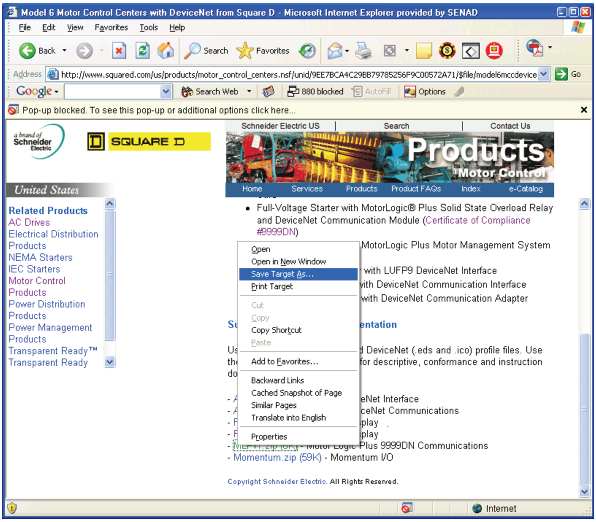
1. Using Microsoft® Windows® Explorer and Microsoft Internet Explorer, create a directory on your hard drive named “SquareD” for downloading the EDS and ICO files. The remaining instructions of this procedure assume that the new directory has the path “C:\SquareD,” however it is acceptable to locate the SquareD directory in any folder structure.

Figure 2: SquareD Directory



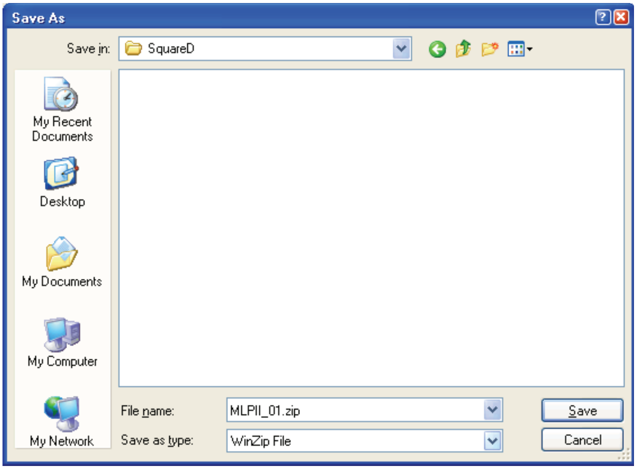
2. Select the appropriate *.zip file from the Model 6 Motor Control Center with DeviceNet area of the Square D web site (www.us.SquareD.com) and save it to the SquareD directory. Click on the desired file name with the right mouse button to access a pop-up menu that contains a “Save Target As...” feature.

Figure 3: Model 6 Motor Control Center with DeviceNet



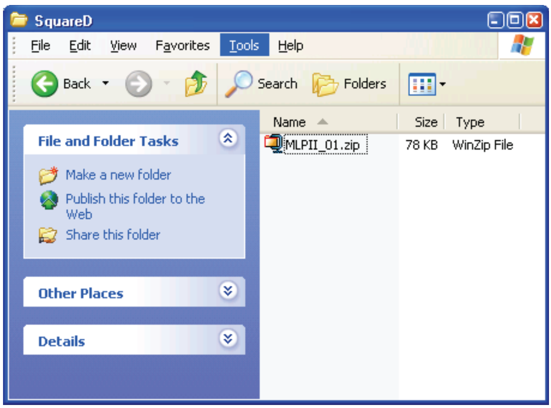
3. Browse to the SquareD directory and click the Save button.

Figure 4: Saving the *.zip File



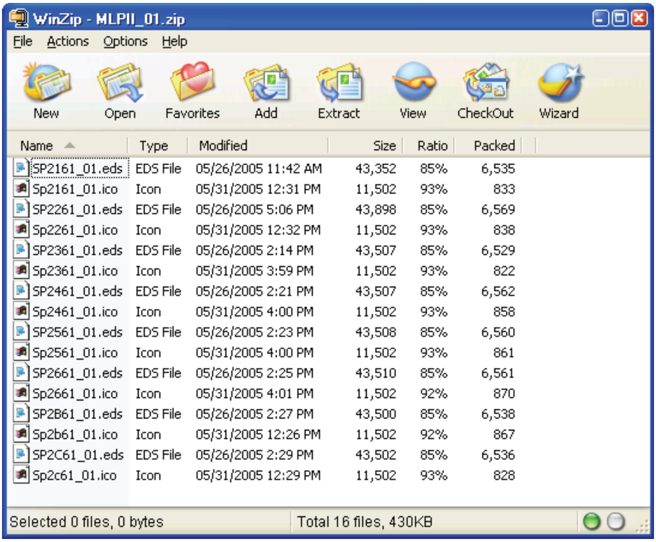
4. Navigate to the SquareD directory and verify that the save operation was successful.

Figure 5: The *.zip File in the SquareD Directory



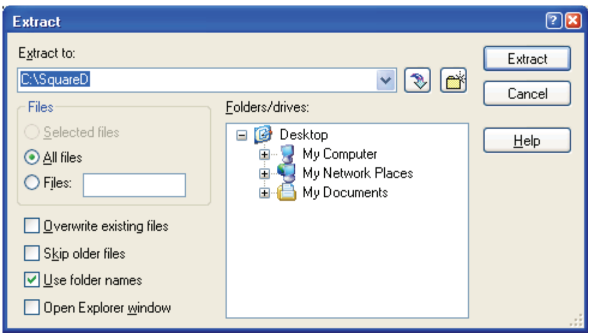
5. Open the *.zip file and ensure that the desired EDS and ICO files are present.

Figure 6: Open the *.zip File



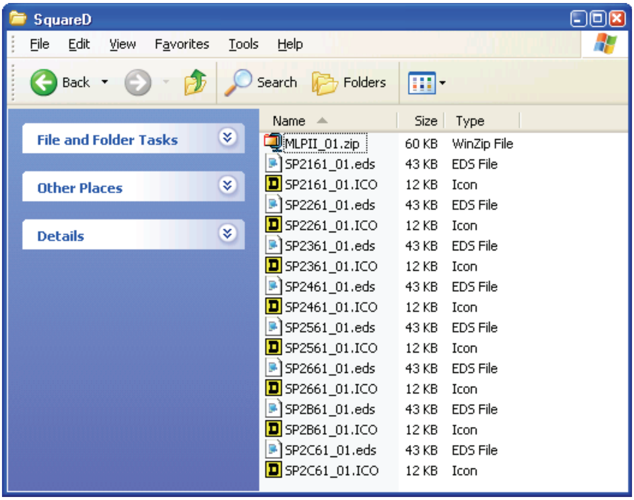
6. Extract the contents of the *.zip file to the SquareD directory.

Figure 7: Extracting the Contents of the *.zip File



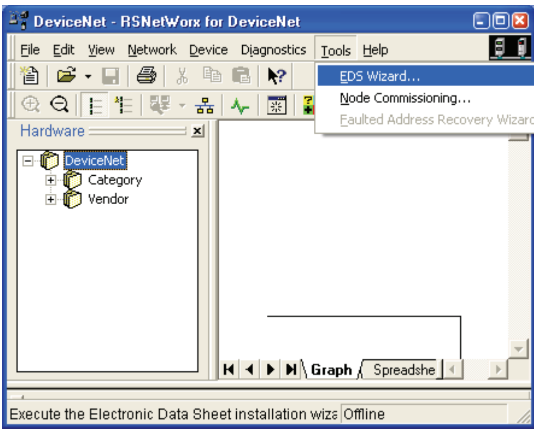
All Motor Logic® Plus SSOLR EDS and ICO files should now be visible in the SquareD directory.

Figure 8: EDS and ICO Files in the SquareD Directory



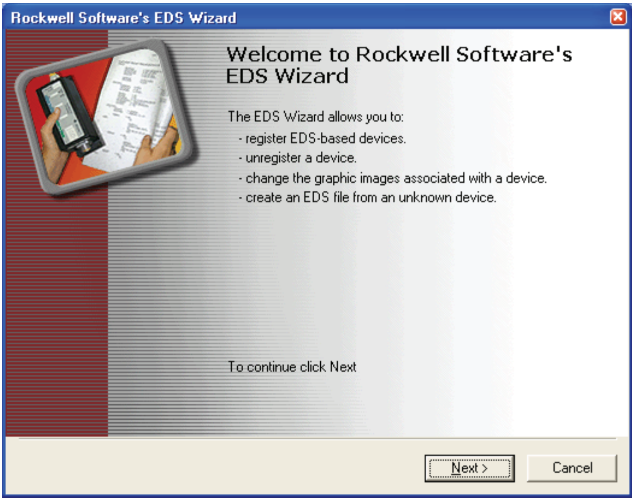
7. Using RSNetworx, select the EDS Wizard from Tools>EDS Wizard.

Figure 9: Selecting the EDS Wizard



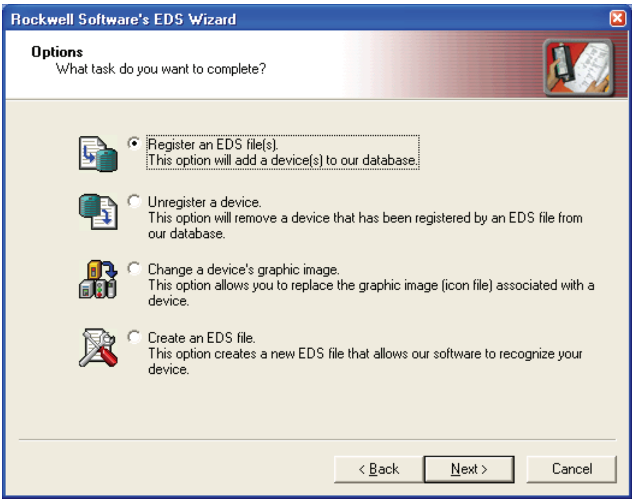
8. Click the Next button from the Welcome screen.

Figure 10: EDS Wizard Welcome Screen



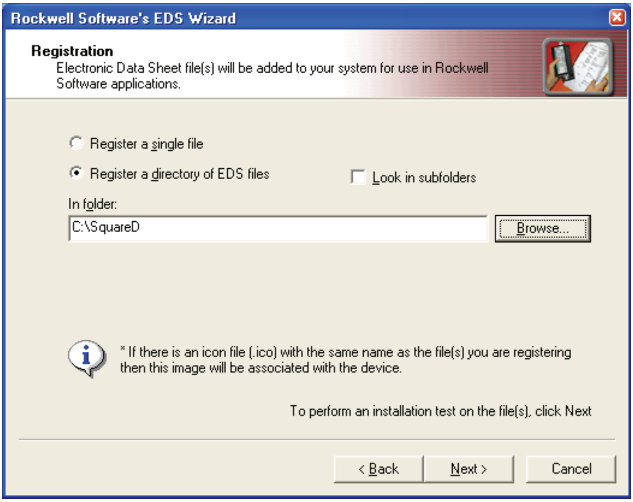
9. Select the “Register an EDS file(s)” option. Then click the Next button.

Figure 11: EDS Wizard Options



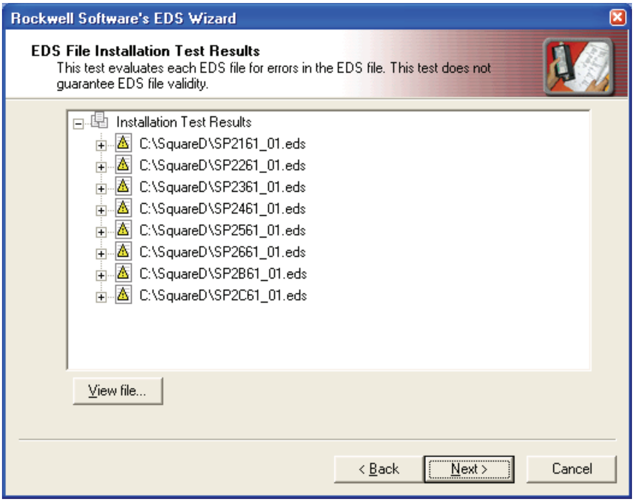
10. Select the “Register a directory of EDS files” option and browse to C:\SquareD. Then click the Next button.

Figure 12: EDS Wizard Registration



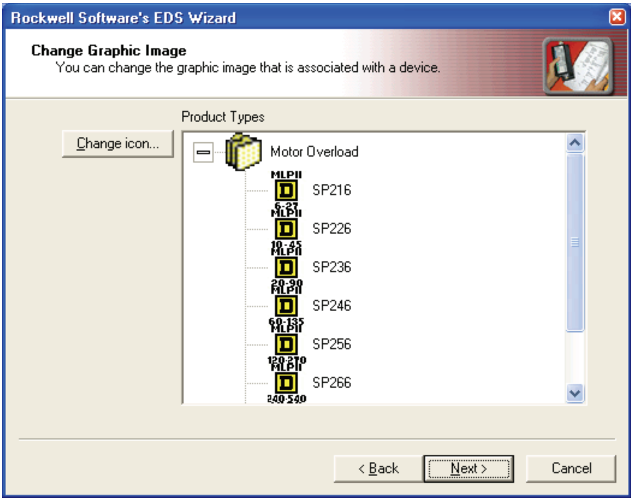
11. The EDS file evaluation screen appears. The EDS file installation screen evaluates the EDS file for errors. The test does not guarantee the EDS file validity. When the evaluation is complete, click the Next button.

Figure 13: EDS File Installation Test Results



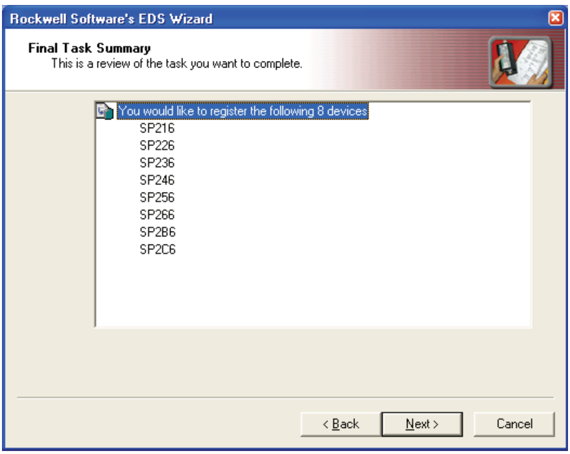
12. The graphic image screen appears displaying all of the icons for the Motor Logic® Plus SSOLRs. Click the Next button.

Figure 14: Change Graphic Image Screen



13. The summary screen confirms the devices that will be registered. If any of the devices you want to register do not appear in this list, perform this procedure again. Click the Next button if the list is complete.

Figure 15: Final Task Summary Screen

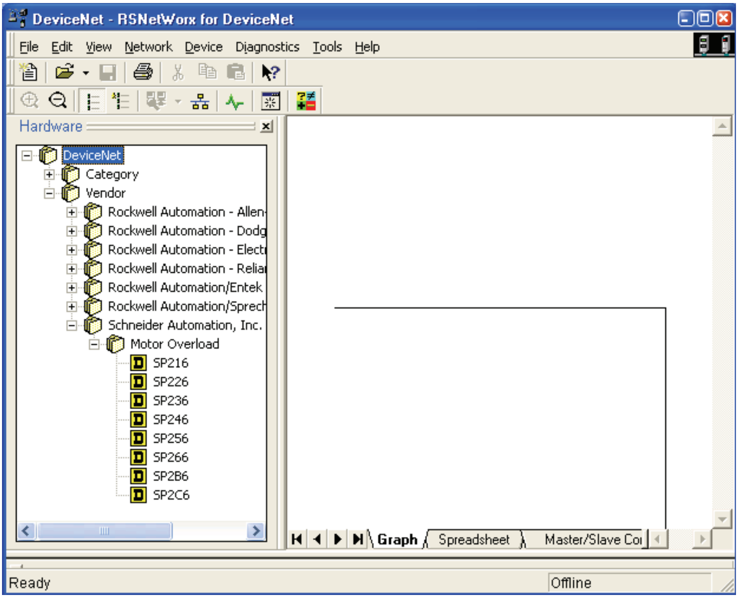


14. You have successfully registered the Motor Logic® Plus SSOLRs using the EDS Wizard. Click the Finish button.

Figure 16: Completing the EDS Wizard

15. For final verification, navigate to DeviceNet > Vendor > Schneider Automation, Inc. > Motor Overload on the left pane of RSNetWorx to see all of the devices.

Figure 17: Final Verification



Faulted Node Recovery

Devices with shared addresses will stop communicating and go into the communication faulted mode. The Faulted Address Recovery Wizard (see Figure 18 on page 63) searches the network for devices with conflicting addresses, then lists the serial numbers of the faulted devices.

When commissioning multiple devices, you can record all serial numbers along with the device's location, panel number, or other identifying characteristics. You can use the serial number list to assign network addresses.

If serial numbers are not used, faulted devices display Faulted Node Recovery (FNR) indicators. The Identify Communication Faulted (ICF) indicator appears when the device receives an ICF request message. Table 21 lists the indicators for the Motor Logic® Plus SSOLR.

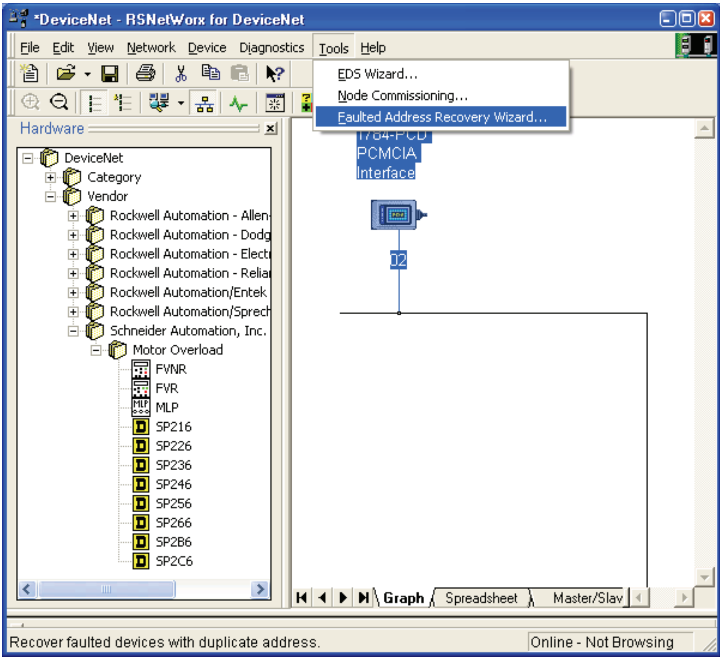
Table 21: Motor Logic® Plus SSOLR, MMS-D, and MMS-RD Indicators

Product	Display Indicator	Performance
MLP Series A and B	oFF	The SSOLR goes into the Off state. oFF is displayed and the relay opens. The SSOLR stays in the off state even when ICF requests are no longer sent.
MLP Series C	ICF	The SSOLR goes into the Off state. ICF is displayed while the ICF request is present. The SSOLR remains in the Off state and displays off when the ICF is no longer sent.
MMS-D and MMS-RD Displays	Identify Comm Faulted	Message displayed while request is present. Returns to normal state after message is removed. If connected and energized, the SSOLR operates as described above.

Using the Faulted Address Recovery Wizard

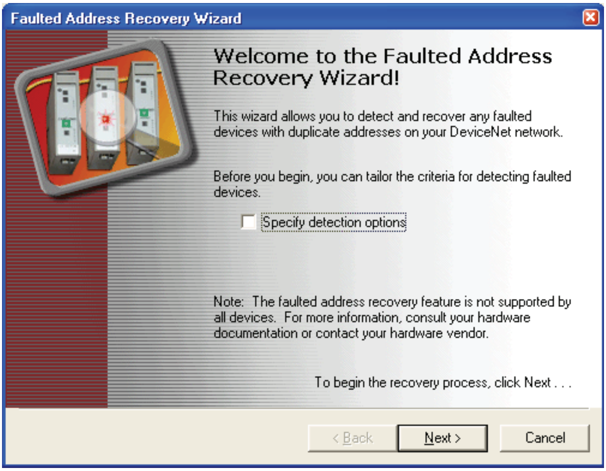
1. Select Tools > Faulted Address Recovery Wizard.

Figure 18: Selecting Faulted Address Recovery Wizard



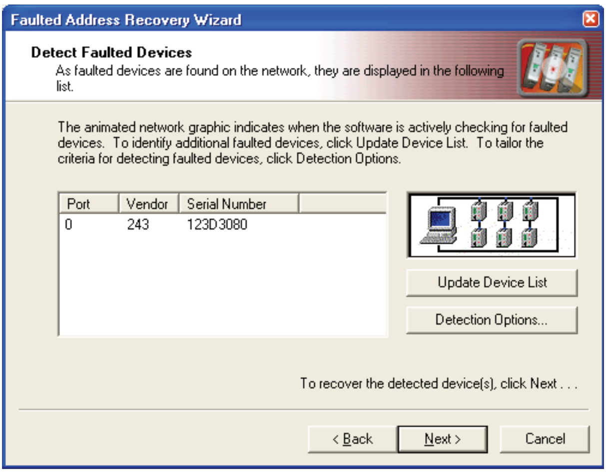
2. On the welcome screen, click the Next button.

Figure 19: Faulted Address Recovery Wizard Welcome Screen



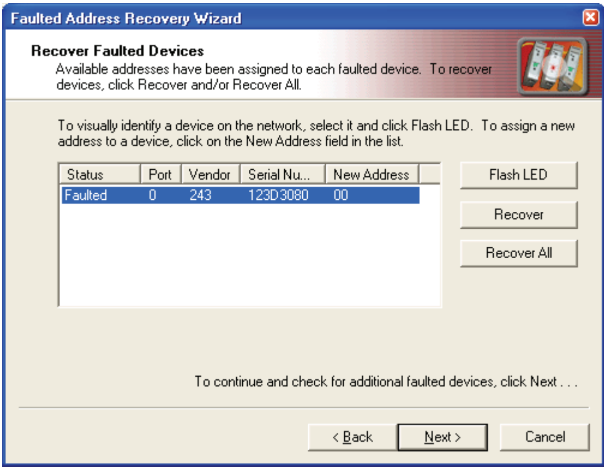
3. The Wizard scans the entire network and lists the devices that are faulted. After the Wizard is finished scanning the network, click the Next button.

Figure 20: Scan for Faulted Devices



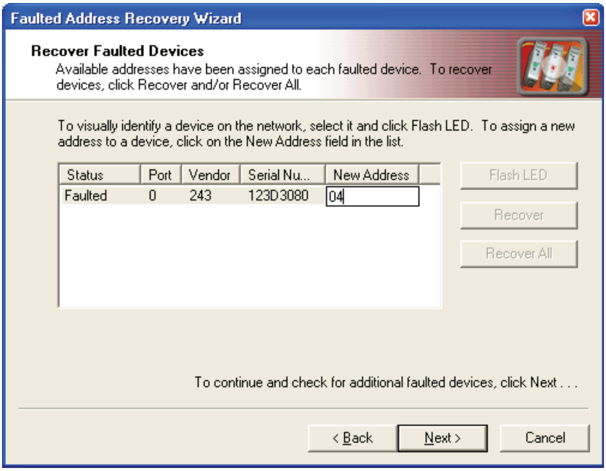
4. The recover faulted devices screen allows you to perform several actions on the faulted device. You can send an ICF request to the faulted device by clicking the Flash LED button to ensure that the Wizard has detected the correct device. The faulted device's actions will match those listed in Table 21, depending on the specific type of faulted device.

Figure 21: Recover Faulted Devices



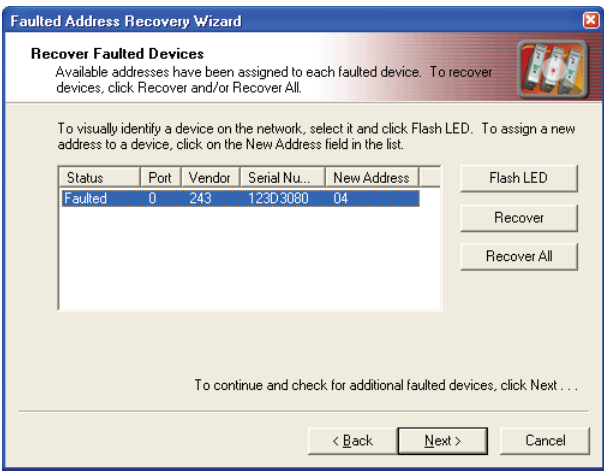
5. After confirming the location of the faulted device, you can change its address from the one automatically generated by the Wizard to another one. In this example, the address is changed from “00” to “04.”

Figure 22: Change Device Address



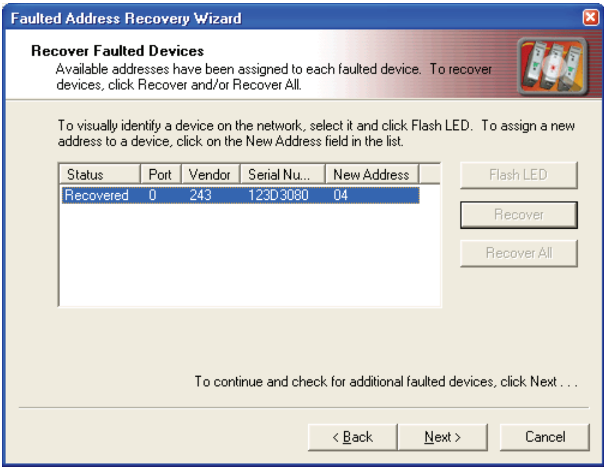
6. Once the identity of the faulted device and its address are known, the device may be recovered by clicking on the Recover button.

Figure 23: Recover Faulted Devices



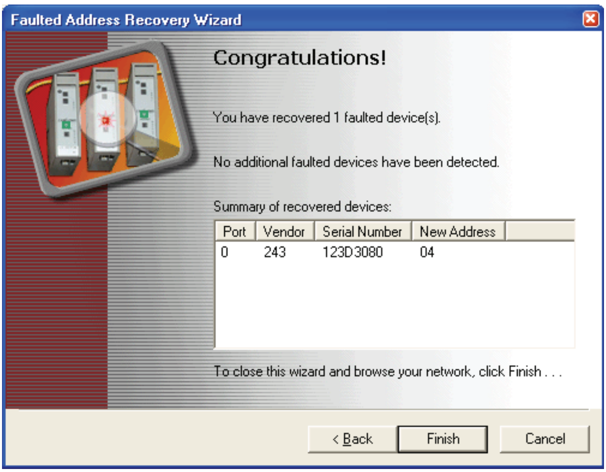
7. The status of the device will change from faulted to recovered. After all devices are recovered, click the Next button to go to the final summary screen.

Figure 24: Recover Faulted Devices (cont.)



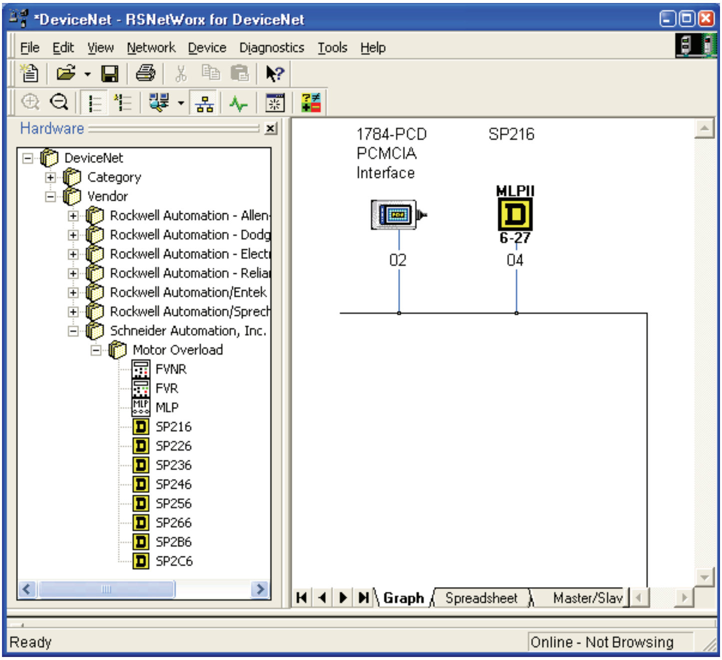
8. The final summary screen lists all devices that were recovered. Click the Finish button to close the Wizard and return to the main RSNetWorx network panes.

Figure 25: Recover Faulted Devices (cont.)



9. The recovered faulted devices will now appear in the network pane.

Figure 26: Network Pane with Recovered Devices



Configuration of the DeviceNet Parameters

A node's address and data must be programmed into the device before it can be added to a DeviceNet network.

Commissioning Nodes

Devices can be individually commissioned by:

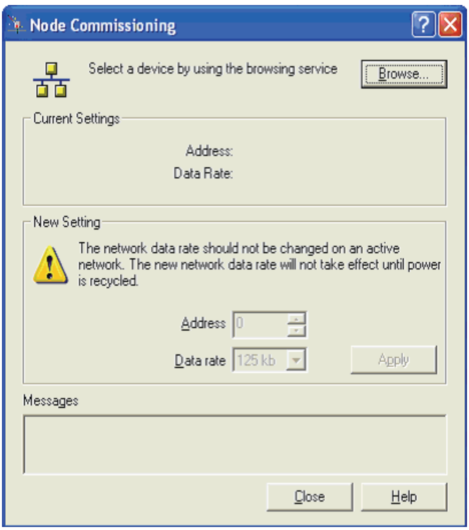
- Placing one unit on the network at a time, or
- Placing multiple units on the network that have individual default addresses

Use the Node Commissioning tool to configure the address and data (baud) rate of an individual device. If all of the units have default addresses and are placed on the network simultaneously, use the Faulted Address Recovery Wizard to commission the group of nodes.

To commission nodes:

1. From your desktop, click RSNetworx > Tools > Node Commissioning...

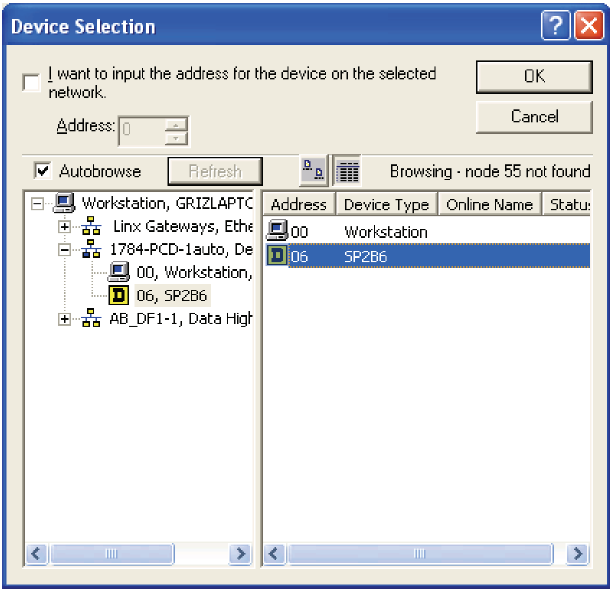
Figure 27: Node Commissioning Window



2. Click Browse, then select a node to change its address and data (baud) rate settings.

3. Click Apply > Close to confirm the changes and close the window.

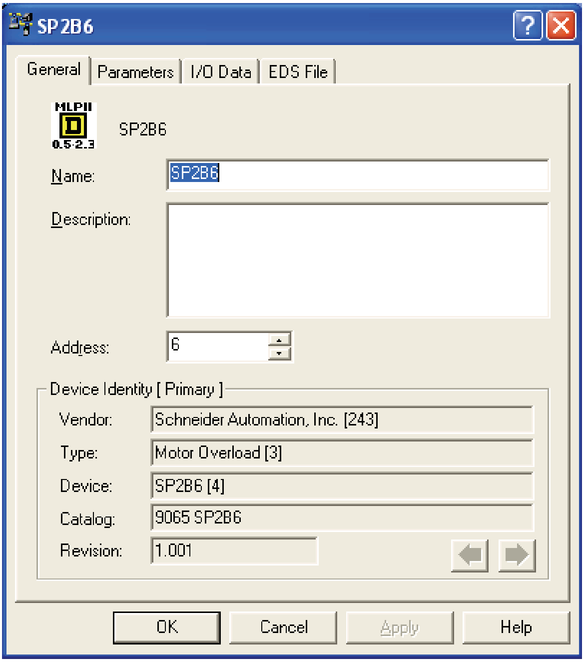
Figure 28: Device Selection Window



4. In the Device Selection window, select the network from the left pane. Then, to select a device, do one of the following:
   - Check the “I want to input the address for...” check box, then manually enter the address.
   - Select the node from the right pane.

5. Click OK to confirm the changes and close the window. The Motor Logic Plus SSOLR window displays the General properties of the SSOLR.

Figure 29: Motor Logic® Plus General Properties Window



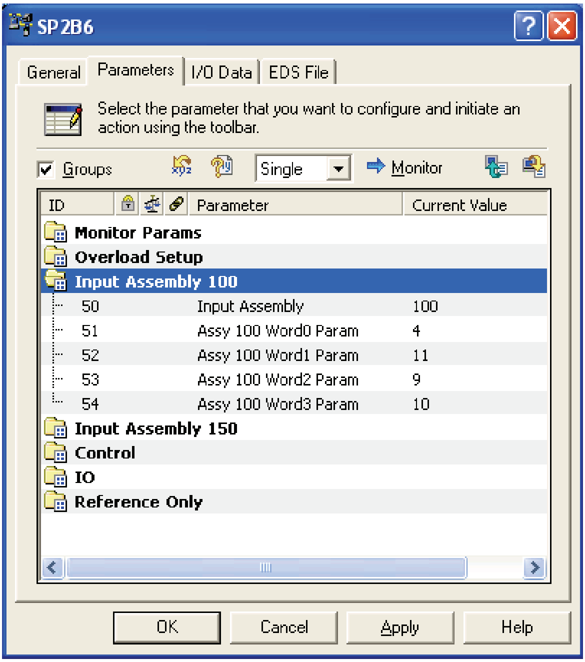
6. Change the address of this device to the new network address. (The address must be changed for each device placed on the network.)
7. Click OK to confirm the changes and close the window.

Reading and Configuring the SSOLR Parameters

To read and configure the parameters of the devices:

1. Select Network > Online > Properties to open the properties window for the SSOLRs that you want to read or configure.
2. Select the Parameters tab and upload the parameters from the SSOLR when prompted.

Figure 30: Motor Logic® Plus SSOLR Parameters Properties Window



3. To configure a parameter, double-click the group name.

   The Parameters tab allows get/set access to the unit's parameters as specified by the EDS file registered in RSNetWorx.

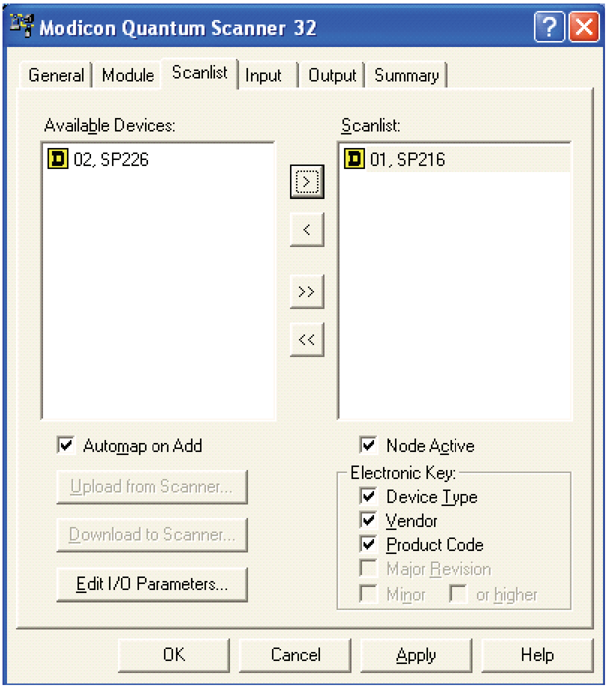
   Table 22 on page 73 lists and describes the Motor Logic Plus groups that you can configure. Each group contains the parameters needed for typical actions such as monitoring and SSOLR setup.

Table 22: Motor Logic® Plus SSOLR Groups

Group	Description
Monitor Params	Real time parameters or data to be monitored. For example, currents, timers, voltages, and status.
SSOLR Setup	SSOLR set points.

4. Select the option from the Scanlist, or left-click the current value to change set points.
5. Click the Apply button or the Download to Scanner button to download the change.

Figure 31: Scanlist Window for Scanner



6. Use the right and left arrows to add or remove the SSOLR from the Scanlist.

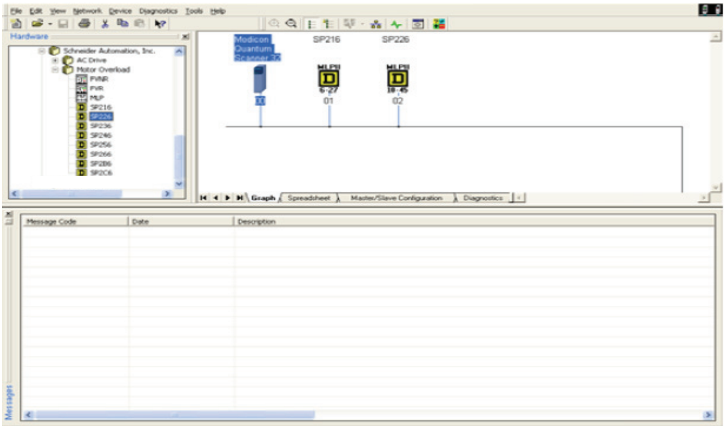
Configure the Network

Follow the procedure below to configure the DeviceNet scanner using RSNetworx™ software. Figure 32 on page 75 shows a simple network with several Motor Logic Plus SSOLRs.

1. Use the EDS Wizard to register the relevant EDS files: RSNetwork > Tools > EDS Wizard.
2. After the EDS files are registered, select Network > Single Pass Browse to load the DeviceNet products onto the network.

   With Single Pass Browse, the recovered nodes are automatically placed on the network.
3. Select the device from the Hardware list to display its icon in the right pane.
4. To change the default settings of a device, double-click on its icon, then do the following:
   a. In the General tab, change the default address to the preferred address.
   b. In the Parameters tab, change the assembly parameters.
   c. In the EDS File, change the EDS file (optional).
   d. Click Apply > OK.
5. Repeat Step 4 for each device that was added to the network or that requires configuring.

Figure 32: EDS Wizard



Configuring the DeviceNet Scanner

1. Open the Properties window for the scanner. See Figure 33 on page 76.
2. Select the Scanlist tab.
3. Move the desired devices from the Available Devices list to the Scanlist list. See Figure 34 on page 76.

Figure 33: DeviceNet Scanner Properties Window

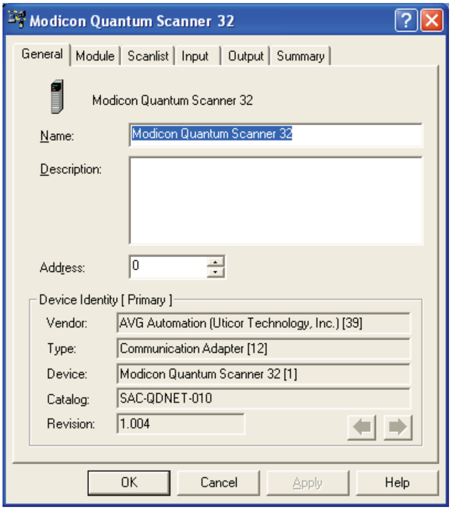
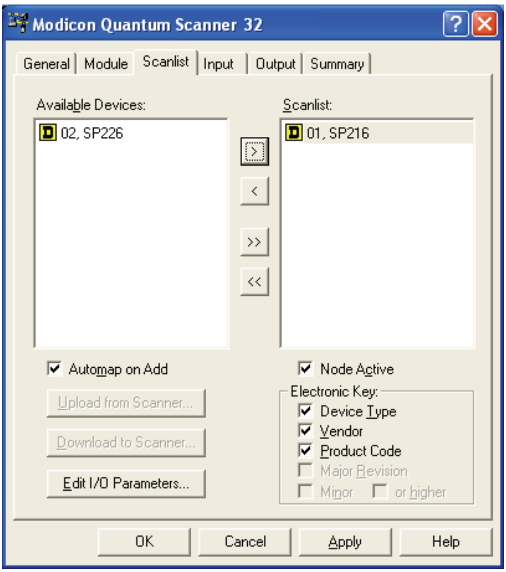


Figure 34: DeviceNet Scanner: Scanlist Tab



OVERLOAD CLASSES—DEVICENET SPECIFICATION VOLUME 2, CHAPTERS 3-9

Table 23 identifies the device classes for SSOLRs.

Table 23: Object Model Definitions for SSOLRs

Object Class	Class ID	Need	No. of Instances	Effect on Behavior	Interface
Identity	01_{hex}	Req.	1	Supports the reset service	Message Router
Message Router	02_{hex}	Opt.	1	No effect	Explicit Message Connection
DeviceNet	03_{hex}	Req.	1	Configures node attributes	Message Router
Assembly	04_{hex}	Req.	3	Defines I/O data format	Message Router, Assembly, or Parameter Object
DeviceNet Connection	05_{hex}	Req.	2	Logical ports into or out of the device	I/O Connection or Message Router
Control Supervisor	29_{hex}	Req.	1	Manages SSOLR functions, operational states, and control	Message Router, Assembly, or Parameter Object
Overload	$2C_{hex}$	Req.	1	Provides SSOLR configuration	Message Router, Assembly, or Parameter Object
Acknowledge Handler	$2B_{hex}$	Opt.	1	——	I/O Connection or Message Router

Network Module Objects

Identity Object (Class ID 01_{hex})

The Identity Object provides general, identifying information about the device.

Table 24: Identity Object Class Code 0x01

Attribute ID	Access Rule	Name	Data Type	Value
1	GET	Revision	UINT	1

Table 25: Identity Object Class Code 0x01, Instance 1

Attribute ID	Access Rule	Name	Data Type	Value
1	GET	Vendor ID	UINT	243
2	GET	Device Type	UINT	3
3	GET	Product Code, SPB4	UINT	12
		Product Code, SPC4		13
		Product Code, SP14		14
		Product Code, SP24		15
		Product Code, SP34		16
		Product Code, SP44		17
		Product Code, SP54		18
		Product Code, SP64		19
3	GET	Product Code, SPB6	UINT	20
		Product Code, SPC6		21
		Product Code, SP16		22
		Product Code, SP26		23
		Product Code, SP36		24
		Product Code, SP46		25
		Product Code, SP56		26
		Product Code, SP66		27
3	GET	Product Code, MMS-D w/SPB4	UINT	28
		Product Code, MMS-D w/SPC4		29
		Product Code, MMS-D w/SP14		30
		Product Code, MMS-D w/SP24		31
		Product Code, MMS-D w/SP34		32
		Product Code, MMS-D w/SP44		33
		Product Code, MMS-D w/SP54		34
		Product Code, MMS-D w/SP64		35
3	GET	Product Code, MMS-D w/SPB6	UINT	36
		Product Code, MMS-D w/SPC6		37
		Product Code, MMS-D w/SP16		38
		Product Code, MMS-D w/SP26		39
		Product Code, MMS-D w/SP36		40
		Product Code, MMS-D w/SP46		41
		Product Code, MMS-D w/SP56		42
		Product Code, MMS-D w/SP66		43

Table 25: Identity Object Class Code 0x01, Instance 1 *(continued)*

Attribute ID	Access Rule	Name	Data Type	Value
3	GET	Product Code, MMS-RD w/SPB4	UINT	44
		Product Code, MMS-RD w/SPC4		45
		Product Code, MMS-RD w/SP14		46
		Product Code, MMS-RD w/SP24		47
		Product Code, MMS-RD w/SP34		48
		Product Code, MMS-RD w/SP44		49
		Product Code, MMS-RD w/SP54		50
		Product Code, MMS-RD w/SP64		51
		Product Code, MMS-RD w/SPB6		52
		Product Code, MMS-RD w/SPC6		53
		Product Code, MMS-RD w/SP16		54
		Product Code, MMS-RD w/SP26		55
		Product Code, MMS-RD w/SP36		56
		Product Code, MMS-RD w/SP46		57
		Product Code, MMS-RD w/SP56		58
		Product Code, MMS-RD w/SP66		59
4	GET	Revision Major Minor	STRUCT of: USINT USINT	— 1 1
5	GET	Status	WORD	—
6	GET	Serial Number	UDINT	4
7	GET	9065 SPB4 or 9065 SPB6	SHORT_STRING	MLP
		9065 SPC4 or 9065 SPC6		MLP
		9065 SP14 or 9065 SP16		MLP
		9065 SP24 or 9065 SP26		MLP
		9065 SP34 or 9065 SP36		MLP
		9065 SP44 or 9065 SP46		MLP
		9065 SP54 or 9065 SP56		MLP
		9065 SP64 or 9065 SP66		MLP
		Product Name, MMS-D		FVNR
		Prodcut Name, MMS-RD		FVR

Table 26: Attribute 5—Status (Class ID 01_{hex})

Bit	Definition
0	Owned by master (predefined master/slave connection)
2	Configured
8	Minor recoverable fault
9	Minor unrecoverable fault
10	Major recoverable fault
11	Major unrecoverable fault
Others	Reserved 0

Table 27: Class Service (Class ID 01_{hex})

Service Code	Service Name	Description
$0E_{hex}$	Get_Attribute_Single	Read an attribute

Table 28: Instance Service (Class ID 01_{hex})

Service Code	Service Name	Description
$0E_{hex}$	Get_Attribute_Single	Read an attribute
10_{hex}	Set_Attribute_Single	Write an attribute
05_{hex}	Reset	Reset DeviceNet module

Table 29: Message Router Object (Class ID 02_{hex})

Attribute ID	Access Rule	Name	Data Type	Value
1	GET	Revision 1	UINT	1

DeviceNet Object (Class ID 03_{hex})

The DeviceNet Object provides the status and configuration of a DeviceNet node.

Table 30: Class Attributes (Class ID 03_{hex})

Attribute ID	Access Rule	Name	Data Type	Value	Details
1	GET	Revision	UINT	1	—
2	GET	Max Instances	UINT	1	1 defined instance

Table 31: Instance Attributes (Class ID 03_{hex})

Attribute ID	Access Rule	Name	Data Type	Value	Details
1	GET/SET	MAC ID	USINT	0–63	Ref = 63
2	GET/SET	Baud rate	USINT	0–2	0 = 125 k 1 = 250 k 2 = 500 k
3	GET/SET	BOI (BusOff interrupt)	BOOL	—	Upon BusOff event: 0: CAN component remains in BusOff 1: Component is reset—communication resumes
4	GET/SET	BusOff counter	USINT	0–255	Number of occurrences of BusOff state
5	GET	Allocation information	BYTE USINT	— 0–63	Allocation choice Master address (255 not allocated)

Table 32: Class Service (Class ID 03_{hex})

Service Code	Service Name	Description
$0E_{hex}$	Get_Attribute_Single	Read an attribute

Table 33: Instance Service (Class ID 03_{hex})

Service Code	Service Name	Description
$0E_{hex}$	Get_Attribute_Single	Read an attribute
10_{hex}	Set_Attribute_Single	Write an attribute
$4B_{hex}$	Allocate Master/Slave Connection Set	Allocation connection master/slave
$4C_{hex}$	Release Master/Slave Connection Set	Release connection master/slave

Assembly Object Class Code 0x04

Output Assemblies[1]

Output Assemblies allow control of the SSOLR using a polled message. These assemblies allow the SSOLR with the appropriate display module (9999MMS-D or 9999MMS-RD) to be reset and the output relays to be opened and closed.

Table 34: Output Assemblies

Fault Reset	0	No change
	1	Reset fault relay
Forward	0	Open Forward relay
	1	Close Forward relay
Reverse	0	Open Reverse relay
	1	Close Reverse relay

Table 35: Assembly Object Class Instance 2

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
—	—	—	—	—	Fault Reset	—	—

Table 36: Assembly Object Class Instance 101

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
—	—	—	—	—	—	—	Forward

Table 37: Assembly Object Class Instance 103

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
—	—	—	—	—	Fault Reset	—	Forward

Table 38: Assembly Object Class Instance 104

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
—	—	—	—	—	—	Reverse	Forward

[1] When using displays 9999MMS-D or 9999MMS-RD, Out A refers to "Forward" Out B refers to "Reverse".

Table 39: Assembly Object Class Instance 104

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
—	—	—	—	—	Fault Reset	Reverse	Forward

Table 40: Trip Log 0–3, Control Supervisor Attribute 116–119, (0 x 29, 1, 0 x 74–0x77)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—	—	—	—	—	—	—	CUB	—	UC	—	—	Gnd Flt	Phase Loss	OC	—

Table 41: Trip Log 0–3, Control Supervisor Attr. 116–119, Bit Description

Bit	Setting
Bit 1	1 = Overcurrent
Bit 2	1 = Phase loss
Bit 3	1 = Ground fault
Bit 4	—
Bit 5	—
Bit 6	1 = Undercurrent
Bit 8	1 = Current unbalance

Table 42: Connection Object Class Code 0x05

Attribute ID	Access Rule	Name	Data Type	Value
1	GET	Revision	UINT	1

Table 43: Connection Object—Class Code 0x05, Explicit

Attribute ID	Access Rule	Name	Data Type	Value
1	GET	State	USINT	0 = Nonexistent 1 = Configuring 3 = Established 4 = Timed out
2	GET	Instance type	USINT	0 = Explicit
3	GET	Transport class trigger	BYTE	—
4	GET	Produced connection ID	UINT	10xxxxxx011 xxxxxx = Node address

Table 43: Connection Object—Class Code 0x05, Explicit *(continued)*

Attribute ID	Access Rule	Name	Data Type	Value
5	GET	Consumed connection ID	UINT	10xxxxxx100 xxxxxx = Node address
6	GET	Initial comm. characteristics	BYTE	—
7	GET	Produced connection size	UINT	8
8	GET	Consumed connection size	UINT	7
9	GET/SET	Expected packet rate	UINT	—
12	GET/SET	Watchdog timeout action	USINT	—
13	GET	Produced connection path length	UINT	—
14	GET	Produced connection path	—	—
15	GET	Consumed connection path length	UINT	—
16	GET	Consumed connection path	—	—

Table 44: Connection Object—Class Code 0x05, Polled

Attribute ID	Access Rule	Name	Data Type	Value
1	GET	State	USINT	0 = Nonexistent 1 = Configuring 3 = Established 4 = Timed out
2	GET	Instance type	USINT	1 = I/O
3	GET	Transport class trigger	BYTE	—
4	GET	Produced connection id	UINT	01111xxxxxx xxxxxx = Node address
5	GET	Consumed connection ID	UINT	10xxxxxx101 xxxxxx = Node address
6	GET	Initial comm. characteristics	BYTE	—
7	GET	Produced connection size	UINT	1–50
8	GET	Consumed connection size	UINT	0–8
9	GET/SET	Expected packet rate	UINT	—
12	GET/SET	Watchdog timeout action	USINT	—
13	GET	Produced connection path length	UINT	—
14	GET	Produced connection path	—	—
15	GET	Consumed connection path length	UINT	—
16	GET	Consumed connection path	—	—

Table 45: Discrete Input Object—Class 0x08

Attribute ID	Access Rule	Name	Data Type	Value
1	GET	Revision	UINT	1

Table 46: Discrete Input Object—Class 0x08, Instances 1 & 2

Attribute ID	Access Rule	Name	Data Type	Value
3	GET	Value	BOOL	0 = Off 1 = On

Table 47: Discrete Output Object—Class 0x09

Attribute ID	Access Rule	Name	Data Type	Value
1	GET	Revision	UINT	1

Table 48: Discrete Output Object—Class 0x09, Instances 1, 2, 3

Attribute ID	Access Rule	Name	Data Type	Value
3	GET	Value, instance 1 Motor Logic® Plus fault relay	BOOL	0 = Off 1 = On
	GET/SET	Value, instance 2 Out A relay (MMS-D)	BOOL	0 = Off 1 = On
	GET/SET	Value, instance 3 Out B relay (MMS-RD)	BOOL	0 = Off 1 = On

Table 49: Parameter Object—Class 0x0F

Attribute ID	Access Rule	Name	Data Type	Value
1	GET	Revision	UINT	1

Table 50: Parameter Object—Class 0x0F, All Instances

Attribute ID	Access Rule	Name	Data Type	Value
1	GET	Value	—	Actual value of parameter
2	GET	Link path size	USINT	6
3	GET	Link path	UINT	Depends on instance
4	GET	Descriptor	UINT	Depends on instance
5	GET	Data type	UINT	Depends on instance
6	GET	Data size	UINT	2

Table 51: Parameter Object—Class 0x0F, Instances 1–25

Instance	Access Rule	Name	Data Type	Value
1	GET	L1 current	UINT	A
2	GET	L2 current	UINT	A
3	GET	L3 current	UINT	A
4	GET	Average current	UINT	A
5	GET	L1 current (%FLA)	UINT	—
6	GET	L2 current (%FLA)	UINT	—
7	GET	L3 current (%FLA)	UINT	—
8	GET	Avg. current(%FLA)	UINT	—
9	GET	% Thermal capacity used	UINT	—
10	GET	Ground fault current	UINT	A
11	GET	Current imbalance (%)	UINT	—
12	GET	OL time to reset	UINT	—
13	GET	Trip status bits	UINT	Bit 1: 1 = Overcurrent Bit 2: 1 = Phase loss Bit 3: 1 = Ground fault Bit 6: 1 = Undercurrent Bit 8: 1 = CUB
14	GET	Warning status bits	UINT	
15	GET	Trip log 0	UINT	
16	GET	Trip log 1	UINT	
17	GET	Trip log 2	UINT	
18	GET	Trip log 3	UINT	
19	GET	Device status	UINT	Bit 0: Tripped Bit 8: Motor is running Bit 9: Ground fault
20	GET	RD1 timer	UINT	0–500 seconds
21	GET	RD2 timer	UINT	2–500 minutes
22	GET	RD3 timer	UINT	2–500 minutes
23	GET	MULT (CT ratio)	UINT	—
24	GET	Run hours	UINT	0–65,000 hours
25	GET	Voltage 1		V
26	GET	Voltage 2		V
27	GET	Voltage 3		U
28	GET	Average voltage		V
29	GET	Voltage unbalance (%)		%
30	GET	Power factor		viewable through MMS-D or MMS-RD display

Table 52: Control Supervisor Object—Class Code 0x29

Attribute ID	Access Rule	Name	Data Type	Value
1	GET	Revision	UINT	1
7	GET	Forward running indicator	UINT	0,1
8	GET	Reverse running indicator	UINT	0,1
10	GET	Fault status	UINT	—
12	GET	Fault reset	UINT	—
13	GET	Fault code	UINT	—
14	GET	Warning code	UINT	—
114	GET	Trip status	UINT	Bit 1: 1 = Overcurrent Bit 2: 1 = Phase loss Bit 3: 1 = Ground fault Bit 6: 1 = Undercurrent Bit 8: 1 = CUB Bit 14: 1 = AC power fault
115	GET	Warning status	UINT	
116	GET	Trip log 0	UINT	
117	GET	Trip log 1	UINT	
118	GET	Trip log 2	UINT	
119	GET	Trip log 3	UINT	
121	GET	Device status	UINT	Bit 0: Tripped Bit 8: Motor is running Bit 9: Ground fault
126	GET/SET	Fault reset	BOOL	0 = reset, 1 = faulted
155	GET/SET	Modbus address		1–99
156	GET/SET	Modbus comm settings		C01, C02, C03, C04, C05, C06
169	GET	Motor run hours	UINT	0–65,000 hours
170	GET	RD1 timer	UINT	0–500 seconds
171	GET	RD2 timer	UINT	2–500 minutes
172	GET	RD3 timer	UINT	2–500 minutes
173	GET/SET	RD1 setting	UINT	0–500 seconds
174	GET/SET	RD2 setting	UINT	2–500 minutes
175	GET/SET	RD3 setting	UINT	2–500 minutes
176	GET/SET	#RU setting	UINT	0 = manual 1 = 1 auto reset 2 = 2 auto resets 3 = 3 auto resets 4 = 4 auto resets 5 = continuous auto reset

Table 52: Control Supervisor Object—Class Code 0x29 *(continued)*

Attribute ID	Access Rule	Name	Data Type	Value
177	GET/SET	#RF setting	UINT	0 = manual 1 = manual 2 = 1 auto reset 3 = oc1 4 = 2 auto resets 5 = oc2 6 = 3 auto resets 7 = oc3 8 = 4 auto resets 9 = oc4 10 = continuous auto reset 11 = ocA

[1] Do not write to reserved bits.

Table 53: Acknowledge Handler Object—Class 0x2B

Attribute ID	Access Rule	Name	Data Type	Value
1	GET	Revision	UINT	1

Table 54: Acknowledge Handler Object—Class 0x2B, Instance 1

Attribute ID	Access Rule	Name	Data Type	Value
1	GET/SET	Acknowledge timer	UINT	in milliseconds
2	GET/SET	Retry limit	USINT	0 or 1
3	GET	COS producing conn instance	UINT	[1]

[1] Value changes in real time.

Table 55: Overload Object—Class 0x2C

Attribute ID	Access Rule	Name	Data Type	Value
1	GET	Revision	UINT	1

Table 56: Overload Object—Class Code 0x2C, Instance 1

Attribute ID	Access Rule	Name	Data Type	Value
3	GET/SET	Trip FLA setting	INT	—
4	GET/SET	Trip class	UINT	5, 10, 15, 20, 30
5	GET	Avg current	UINT	—
6	GET	%PhImbal	UINT	—
7	GET	%Thermal cap utilized	UINT	0–99%

Table 56: Overload Object—Class Code 0x2C, Instance 1 *(continued)*

Attribute ID	Access Rule	Name	Data Type	Value
8	GET	L1 current	INT	—
9	GET	L2 current	INT	—
10	GET	L3 current	INT	—
11	GET	GF current x Attr 100	INT	—
100	GET/SET	Current multiplier	USINT	—
101	GET	L1 current	UINT	—
102	GET	L2 current	UINT	—
103	GET	L3 current	UINT	—
104	GET	Avg current	UINT	—
105	GET	L1%FLA	UINT	—
106	GET	L2%FLA	UINT	—
107	GET	L3%FLA	UINT	—
108	GET	Average%FLA	UINT	—
109	GET	% Thermal cap utilized	UINT	0–99%
110	GET	CUB measured	INT	—
113	GET	OL time to reset	UINT	—
129	GET/SET	Trip class	INT	5, 10, 15, 20, 30
130	GET/SET	Jam protection	USINT	1 = Jam enabled
137	GET/SET	GF trip level	INT	—
146	GET/SET	UC trip delay	USINT	2–60 seconds
147	GET/SET	UC trip level	UINT	[1]
151	GET/SET	CUB trip level	USINT	2–25%
177	GET/SET	CT ratio[2]	USINT	—

[1] Refer to the appropriate table for current limits by SSOLR model number.

[2] Sizes 4, 5, and 6.

Table 57: Voltage Monitor Object—Class 0x77

Attribute ID	Access Rule	Name	Data Type	Value
1	GET	Revision	UINT	1

Table 58: Voltage Monitor Object—Class 0x77, Instance 1

Attribute ID	Access Rule	Name	Data Type	Value
3	GET	Average Voltage	UINT	
4	GET	Voltage 1	UINT	
5	GET	Voltage 2	UINT	

Table 58: Voltage Monitor Object—Class 0x77, Instance 1 *(continued)*

Attribute ID	Access Rule	Name	Data Type	Value
6	GET	Voltage 3	UINT	
7	GET	Voltage unbalance	UINT	
8	GET	Power factor	UINT	
20	GET/SET	Low voltage limit	UINT	
21	GET/SET	High voltage limit	UINT	
22	GET/SET	Voltage unbalance limit		

Table 59: DeviceNet Interface Object—Class 0xB4

Attribute ID	Access Rule	Name	Data Type	Value
1	GET	Revision	UINT	1

Table 60: DeviceNet Interface Object—Class 0xB4, Instance 1

Attribute ID	Access Rule	Name	Data Type	Value
7	GET/SET	Assembly 100, Word 1	UINT	—
8	GET/SET	Assembly 100, Word 2	UINT	—
9	GET/SET	Assembly 100, Word 3	UINT	—
10	GET/SET	Assembly 100, Word 4	UINT	—
16	GET/SET	Polled Output Assembly Instance	UINT	0, 2, 101, 103, 104, 105
17	GET/SET	Polled Input Assembly	UINT	0, 100, 150
20	GET/SET	Control Supervisor Model Type	UINT	—
22	GET/SET	DeviceNet Watchdog	UINT	0 = disable, 1 = enable
25	GET/SET	Assembly 150, Word 1	UINT	—
26	GET/SET	Assembly 150, Word 2	UINT	—
27	GET/SET	Assembly 150, Word 3	UINT	—
28	GET/SET	Assembly 150, Word 4	UINT	—
29	GET/SET	Assembly 150, Word 5	UINT	—
30	GET/SET	Assembly 150, Word 6	UINT	—
31	GET/SET	Assembly 150, Word 7	UINT	—
32	GET/SET	Assembly 150, Word 8	UINT	—
33	GET/SET	Assembly 150, Word 9	UINT	—
34	GET/SET	Assembly 150, Word 10	UINT	—
35	GET/SET	Assembly 150, Word 11	UINT	—
36	GET/SET	Assembly 150, Word 12	UINT	—
37	GET/SET	Assembly 150, Word 13	UINT	—

Table 60: DeviceNet Interface Object—Class 0xB4, Instance 1 *(continued)*

Attribute ID	Access Rule	Name	Data Type	Value
38	GET/SET	Assembly 150, Word 14	UINT	—
39	GET/SET	Assembly 150, Word 15	UINT	—
40	GET/SET	Assembly 150, Word 16	UINT	—
41	GET/SET	Assembly 150, Word 17	UINT	—
42	GET/SET	Assembly 150, Word 18	UINT	—
43	GET/SET	Assembly 150, Word 19	UINT	—
44	GET/SET	Assembly 150, Word 20	UINT	—
45	GET/SET	Assembly 150, Word 21	UINT	—
46	GET/SET	Assembly 150, Word 22	UINT	—
47	GET/SET	Assembly 150, Word 23	UINT	—
48	GET/SET	Assembly 150, Word 24	UINT	—
49	GET/SET	Assembly 150, Word 25	UINT	—
50	GET/SET	Assembly 150, Word 26	UINT	—
51	GET/SET	Assembly 150, Word 27	UINT	—
52	GET/SET	Assembly 150, Word 28	UINT	—
53	GET/SET	Assembly 150, Word 29	UINT	—
54	GET/SET	Assembly 150, Word 30	UINT	—
55	GET/SET	Assembly 150, Word 31	UINT	—
56	GET/SET	Assembly 150, Word 32	UINT	—
57	GET/SET	Assembly 150, Word 33	UINT	—
58	GET/SET	Assembly 150, Word 34	UINT	—
59	GET/SET	Assembly 150, Word 35	UINT	—
60	GET/SET	Assembly 150, Word 36	UINT	—
61	GET/SET	Assembly 150, Word 37	UINT	—
62	GET/SET	Assembly 150, Word 38	UINT	—
63	GET/SET	Assembly 150, Word 39	UINT	—
64	GET/SET	Assembly 150, Word 40	UINT	—
65	GET/SET	Assembly 150, Word 41	UINT	—
66	GET/SET	Assembly 150, Word 42	UINT	—
67	GET/SET	Assembly 150, Word 43	UINT	—
68	GET/SET	Assembly 150, Word 44	UINT	—
69	GET/SET	Assembly 150, Word 45	UINT	—
70	GET/SET	Assembly 150, Word 46	UINT	—
71	GET/SET	Assembly 150, Word 47	UINT	—
72	GET/SET	Assembly 150, Word 48	UINT	—

Table 60: DeviceNet Interface Object—Class 0xB4, Instance 1 *(continued)*

Attribute ID	Access Rule	Name	Data Type	Value
73	GET/SET	Assembly 150, Word 49	UINT	—
74	GET/SET	Assembly 150, Word 50	UINT	—

APPENDIX A—8-BIT AND 16-BIT ADDRESS CROSS REFERENCE

The 8-bit memory addresses are provided solely for legacy installations currently using Motor Logic Plus® II SSOLR. Modbus commmands to the Motor Logic Plus® II SSOLR should be sent to the 16-bit memory addresses.

The following table provides a cross-reference between the 8-bit and 16-bit memory addresses. The 8-bit memory addresses increment the address for each byte used. Therefore, 2 byte location addresses increment by 2 rather than 1. A Modbus command can request data starting anywhere in the 8-bit space, allowing the low and high bytes of neighboring locations to be combined in a message.

Table 61: 8-Bit and 16-Bit Address Cross Reference

8-Bit Memory		16-Bit Memory		Code
Hex	Decimal	Hex	Decimal	
84	40133	192	40403	WARNSTAT
86	40135	193	40404	MLP2STAT2
88	40137	194	40405	INFOPWRUP
8A	40139	195	40406	INSTFLT
8C	40141	196	40407	PENDFLT
8E	40143	197	40408	FAULTST
90	40145	198	40409	PTCOHMS
92	40147	199	40410	HOLDOFF
94	40149	19A	40411	MANRESET
96	40151	19B	40412	NETWKSTAT
98	40153	19C	40413	MLP2CMDSTAT
9A	40155	19D	40414	MLP2OPRSTAT
9C	40157	19E	40415	MMSCMDSTAT
9E	40159	19F	40416	MMSOPRSTAT
A2	40163	1A1	40418	IAVE
A6	40167	1A3	40420	CUB
AA	40171	1A5	40422	THERM CAP
AC	40173	1A6	40423	GF
AE	40175	1A7	40424	TRIPRN/ ERCODE
B0	40177	1A8	40425	FH
B2	40179	1A9	40426	PID

Table 61: 8-Bit and 16-Bit Address Cross Reference *(continued)*

8-Bit Memory		16-Bit Memory		Code
Hex	Decimal	Hex	Decimal	
BA	40187	1AD	40430	IC
BC	40189	1AE	40431	IB
BE	40191	1AF	40432	IA
C0	40193	1B0	40433	RD1
C2	40195	1B1	40434	RD2
C4	40197	1B2	40435	RD3
C6	40199	1B3	40436	COM-LINE
C8	40201	1B4	40437	Scale
D0	40209	1B8	40441	MULT
D2	40211	1B9	40442	OC
D4	40213	1BA	40443	UC
D6	40215	1BB	40444	CUB
D8	40217	1BC	40445	TC
DA	40219	1BD	40446	RD1
DC	40221	1BE	40447	RD2
DE	40223	1BF	40448	RD3
E0	40225	1C0	40449	#RU
E2	40227	1C1	40450	#RF
E4	40229	1C2	40451	UCTD
E6	40231	1C3	40452	GF
E8	40233	1C4	40453	MBAddress
EA	40235	1C5	40454	MTIM
EC	40237	1C6	40455	MLP2COMS
EE	40239	1C7	40456	MLP2INPUT
		257	40600	DnADDR
		258	40601	DnBaudRt
		259	40602	ThermCapAvail
		263	40612	WarnUC
		264	40613	WarnOC
		265	40614	WarnGF
		266	40615	WarnCUB
		267	40616	MBAssembly103
		269	40618	WarnEnableMask

APPENDIX B—MOTOR MANAGEMENT SYSTEM DISPLAY (9999MMS) MEMORY MAP

This memory map applies to the Motor Management System Display (9999MMS) beginning with revision 3.11.

Operation: MMSCMDSTAT and the Setup Parameters for the MMS can be read and written; all other registers are Read Only. All registers are 16-bit words.

Address: Relative ONLY (addresses increment by 1 for each 16-bit word).

Table 62: : Relative Address

(Hex)	(Decimal)	Code	Description	Size (words)
19EH	40415	MMSCMDSTAT	Command State/Status (bits)	(1)
19FH	40416	MMSOPRSTAT	Operating Status (bits)	(1)
	...			
400H	41025	PRODID	Product Identifier	(1)
401H	41026	SFTREV	Software Revision (Major * 100 + Minor)	(1)
402H	41027	PWRFCTR	Power Factor * 256 (0–256)	(1)
403H	41028	POWRKW	Power * 256 (KW)	(2)
405H	41030	KVARS	KVA reactive * 256	(2)
407H	41032	ENRGUSE	Energy Usage (KWH * 10)	(2)
409H	41034	NSTARTS	Number of Starts	(1)
40AH	41035	NTRIPS	Number of Trips (Fault trips)	(1)
40BH	41036	DTLSTART	Date/Time of Last Start	(4)
40FH	41040	DTLOFF	Date/Time of Last Off	(4)
413H	41044	DTMINOFF	Date/Time of Minimum Off Time	(2)
417H	41048	MINOFF	Minimum Off Time (Seconds)	(2)
419H	41050	RTC	Real Time Clock - current reading	(4)
41DH	41054	WRNGSTAT	Warning Status (bits)	(1)
41EH	41055	LSTFLT1	Most Recent Fault	(1)
41FH	41056	LSTFLT2	2nd Most Recent Fault	(1)
420H	41057	LSTFLT3	3rd Most Recent Fault	(1)
421H	41058	LSTFLT4	4th Most Recent Fault	(1)
	...			
422H			(Reserved for future Use)	
	...			

Table 62: : Relative Address *(continued)*

(Hex)	(Decimal)	Code	Description	Size (words)
430H	41073	FLTCNT	Number of Faults (0–6)	(1)
431H	41074	FLTPTR	Pointer to Most Recent Fault (0–5)	(1)
432H	41075	FLTDATA0	Data for Fault in Position 0	(12)
43EH	41087	FLTDATA1	Data for Fault in Position 1	(12)
44AH	41099	FLTDATA2	Data for Fault in Position 2	(12)
456H	41111	FLTDATA3	Data for Fault in Position 3	(12)
462H	41123	FLTDATA4	Data for Fault in Position 4	(12)
46EH	41135	FLTDATA5	Data for Fault in Position 5	(12)
47AH	41147	EVTCNT	Number of Events	(1)
47BH	41148	EVTPTR	Pointer to Most Recent Event (0–31)	(1)
47CH	41149	EVTDATA0	Date/Time & Code for Event in Position 1	(4)
480H	41153	EVTDATA1	Date/Time & Code for Event in Position 2	(4)
...				...
4F4H	41269	EVTDATA30	Date/Time & Code for Event Position 30	(4)
4F8H	41273	EVTDATA31	Date/Time & Code for Event Position 31	(4)
...				...
		SETUP PARAMETERS FOR MMS		
5C0H	41473	SLVCOMS	Comm Setting	(1)
5C1H	41474	SLVBAUD	Baud Rate	(1)
5C2H	41475	SLVPAR	Parity bits	(1)
5C3H	41476	SLVSTP	Stop bits	(1)
5C4H	41477	MMSADDR	Network Address of MMS	(1)
5C5H	41478	MMSCOMS	Comm Setting for MMS (1–6)	(1)
5C6H	41479	MMSBAUD	Baud Rate for MMS	(1)
5C7H	41480	MMSPAR	Parity bits for MMS	(1)
5C8H	41481	MMSSTP	Stop bits for MMS	(1)
5C9H	41482	PCFGOPT	Configuration Options	(1)
5CAH	41483	EVTSLOG	Type of Events to Log	(1)
5CBH	41484	LOCLRMT	Local/Remote Control	(1)
5CCH	41485	STRTOPT	Startup Option	(1)

MMS Command State / Status Word

The MMSCMDSTAT register contains 16 bits that control the MMS. The bits are numbered from 15 (most significant bit) to 0 (least significant bit).

Table 63: MMSCMDSTAT Register

Bit 0	1 = MMS - Forward Run	Activates the Foward Run relay
Bit 1	1 = MMS - Reverse Run	Activates the Reverse Run relay
Bit 2	1 = MMS - Stop	Deactivates both Fwd & Rev Run relays
Bit 3	1 = MMS - Clear History Data	
Bit 4	1 = MMS - Clear Last Start Data	
Bit 5	1 = MMS - Clear Fault Data	
Bit 6	1 = MMS - Clear Event Data	
Bit 7	---	
Bit 8	1 = Start / Reset	(See Command 1)[1]
Bit 9	1 = Stop / Off	(See Command 2)[1]
Bit 10	1 = Display Lock 0 = Display Unlock	(See Command 3)[1] (See Command 4)[1]
Bit 11	1 = Network Program Disable 0 = Network Program Enable	(See Command 6)[1] (See Command 5)[1]
Bit 12	1 = Network Watch Dog Enable 0 = Network Watch Dog Disable	(See Command 7)[1] (See Command 8)[1]
Bit 13	1 = Clear Run Time Hours	(See Command 9)[1]
Bit 14	1 = Clear Last Fault	(See Command 10)[1]
Bit 15	1 = MMS - ShutDown State to Hold Last Values 0 = MMS - ShutDown State to Manual Restart	(Auto Restart)

1 See SSOLR Command Processing section on page 102.

Writing a 1 to bits 3–9 and bits 13–14 will not latch a 1 value, but it will cause an action; then the bit will revert to 0. Bits 0–2 will be changed depending on the state of the Forward and Reverse Control relays. Bit 15 will be changed to reflect any change in the ShutDown state due to operator change at the MMS keypad.

Bits 10–12 will not accurately reflect the operating state of the SSOLR until the states are changed. In other words, for a newly installed SSOLR, the controller may need to send all three bits as 1’s (ones), then send all three bits as 0’s (zeroes), and finally, send the desired operating state for each bit.

MMS Operating Status Format

The MMSOPRSTAT register contains 16 bits that describe the Operating Status of the MMS. The bits are numbered from 15 (most significant bit) to 0 (least significant bit).

Table 64: MMSOPRSTAT Register

Bit 0	1 = MMS Forward Run Relay is On or Closed	
Bit 1	1 = MMS Reverse Run Relay is On or Closed	
Bit 2	1 = Motor is drawing current	
Bit 3	---	
Bit 4	1 = SSOLR is not in communication with the MMS display	(1 = Data Fault)
Bit 5	1 = Fault Condition	(RD2 or RD3 is not zero)
Bit 6	1 = ROM checksum failure	
Bit 7	1 = Power Problem or Voltage Error HoldOff	
Bit 8	1 = MMS is in Local Mode	
Bit 9	1 = MMS is in Full Remote mode	
Bit 10	1 = MMS is in Semi-Remote mode	
Bit 11	1 = EEPROM checksum failure	
Bit 12	1 = Input1 is high. See PCFGOPT, bit 7 for definition	
Bit 13	1 = Input2 is high. See PCFGOPT, bit 7 for defintion	
Bit 14	---	
Bit 15	1 = MMS ShutDown State is in Hold Last Value (Auto Restart) 0 = MMS All Relays Off (Manual Restart)	

Format of PWRFCTR variable

The power factor variable at location 41027 is the cosine of the power factor angle (at location 40421). The power factor is multiplied by 256. If PWRFCTR reads 181, the power factor is 181 / 256 or 0.707, which corresponds to the power factor for a 45 degree power factor angle. If the power factor variable reads 256, the power factor is 256 / 256 or 1.00, which corresponds to a power factor angle of 0 degrees.

Warning Status

The WRNGSTAT register at location 41054 contains 16 bits that describe the Instantaneous Warning Status of the SSOLR. The bits are numbered from 15 (Most Significant bit) to 0 (Least Significant bit).

Table 65: WRNGSTAT Register

Bit 0	1 = Pending Trip Condition	(OC, UC, CUB, GF, Current SP)
Bit 1	1 = Voltage Error Detected	If not drawing current, indicates HOLDOFF (VHI, VLO, VUB, VSP, RP)

Date Formats

Date and Time information is stored in the following format:

1st word	Century 256	+	Year
2nd word	Month 256	+	Day
3rd word	Hour 256	+	Minute
4th word	Second 256	+	(Optional Code)

Fault Data

The 12 words of data for each fault are stored in the following format:

1st word	Line Voltage A–C
2nd word	Line Voltage B–C
3rd word	Line Voltage A–B
4th word	Current in Phase C
5th word	Current in Phase B
6th word	Current in Phase A
7th word	Ground Fault Current
8th word	PID (for scale factor for currents)

9th–12th words contain Date and Time information and Fault Code			
9th word	Century 256	+	Year
10th word	Month 256	+	Day
11th word	Hour 256	+	Minute
12th word	Second 256	+	Fault Code

Fault Codes

The Fault Codes stored in the 12th word of the Fault Data Block are the same as the SSOLR units:

4	Contactor Failure
6	Single Phase
7	Ground Fault
8	Unbalance
10	Overcurrent
11	Undercurrent

Event Data

The 4 words of data for each fault are stored in the following format:

1st word	Century 256	+	Year
2nd word	Month 256	+	Day
3rd word	Hour 256	+	Minute
4th word	Second 256	+	Event Code

Event Codes

The EVENT CODE data is stored in the low byte of the 4th word:

1	Motor Start
2	Motor Stop
3	9999MMS Power on
4	9999MMS Power loss
5	Communications with SSOLR unit established
6	Communcations with SSOLR unit lost

Processing Fault and Event Data

The MMS uses ring buffers for the fault data and the event data. The EVTPTR and the FLTPTR always point to the next free data block. If the EVTCNT or the FLTCNT is less than the maximum, the oldest data will be in data block 0. Once the counters are set to their maximum values, the oldest data is the block where the pointer is currently pointing.

In order to calculate the starting address of a block of fault data, multiply the block number by 12, and add the starting address of block 0 (41075 or 432 hexadecimal). For example, with FLTPTR set to 3, the starting address of the most recent fault data will be ((3 - 1) * 12) + (41075); or 41099 (44A hexadecimal).

In order to calculate the starting address of a block of Event data, multiply the block number by 4, and add the starting address of block 0 (41149 or 47C hexadecimal). For example, with EVTPTR set to 8, the starting address of the most recent event data will be ((8 - 1) * 4) + (41149); or 4177 (498hex).

Example of Reading Event Data

As a concrete example, assume there have been 40 motor starts and stops recorded as events since the Event data was cleared. The Event buffer will only store the last 32 events. The first step is to read 2 words starting at location 41147 (47Ahex). The value of EVTCNT will be 32. The value of EVTPTR will be 8. This means that the most recent event is 1 less than the value of the EVTPTR, or position 7.

To read the most recent event, read 4 words starting at location (41149 + 7*4) or location 41177 (498 hex).

For this example, assume that the most recent event was a motor stop on February 17, 2003, at 5:10:25 pm. A motor stop is Event Code 2. The data returned from the Modbus READ command will be:

Register 1:	5123	Century & Yr	(20 * 256 + 03)
Register 2:	529	Month & Day	(2 * 256 + 17)
Register 3:	4362	Hour & Minute	(17 * 256 + 10)
Register 4:	6402	Second & Evt. Code	(25 * 256 + 2)

The 2nd most recent event data is at position 6 or starting at location (41149 + 6*4) or location 41173 (494 hex). The 8th most recent event is at position 0, and the 9th most recent event will wrap back around to position 31. The 10th most recent event is at position 30.

Normally, the MMS can supply 40 words of data in a single 03 (Read) command, but in order to read the 10 most recent events, it will be necessary to do two separate reads. The first read will be 32 words starting at location 41149 (47Chex). This would read the 8th most recent event data through the most recent event data. The second read will be be 8 words starting at location 41269 (4F4hex). This would read the 10th and 9th most recent events.

In the above example, the value of EVTCNT is at its maximum (32). This means that the oldest event, which is the 32nd oldest event data, will be at the current value of EVTPTR or position 8. If EVTCNT were less than its maximum, the oldest event data would be at data block 0.

Clearing Events or Fault History

Clearing the events or clearing the fault history only resets the pointers and counters to zero. The FLTCNT or EVTCNT values will only be zero after clearing the Fault History or Event History.

SSOLR Command Processing

For the normal SSOLR unit, RAM location C6 (or relative location 1B3) is the command line code (COMLINE). The MMS will pass all of the commands down to the SSOLR unit.

Table 66: COMLINE Register

01H	Start / Reset (SSOLR)
02H	Stop / OFF (SSOLR)
03H	Display Lock
04H	Display Unlock
05H	Network Program Enable
06H	Network Program Disable
07H	Enable network comm. loss watchdog timer
08H	Disable network comm loss watchdog timer
09H	Clear Run Time Hours
0AH	Clear Last Fault
25H	Enable Network WD
26H	Disable Network WD

MMS Configuration Options

The PCFGOPT register at location 41482 contains 6 bits that enable or disable several features of the MMS. The bits are numbered from 15 (most significant bit) to 0 (least significant bit).

Table 67: PCFGOPT Register

Bit 0	RESERVED
Bit 1	1 = Display time as AM/PM 0 = Display time in 24 hour format
Bit 2	1 = Automatically dim LCD backlighting after 2 minutes of no key press
Bit 3	RESERVED
Bit 4	1 = Treat under current as Fault and record conditions at time of trip
Bit 5	1 = Enable Modbus host watchdog timer
Bit 6	1 = Enable contactor proof relay follower option
Bit 7	1 = Report input status (bits 12 and 13 in MMSOPRSTAT) as 1 when connections are open 0 = Report as 1 when connections are shorted

Events to Log

The EVTSLOG register at location 41483 controls which events will be logged.

Table 68: EVTSLOG Register

Bit 0	1 = Log motor starts
Bit 1	1 = Log motor stops
Bit 2	1 = Log MMS power on
Bit 3	1 = Log MMS power loss
Bit 4	1 = Log MMS-to-slave communications established
Bit 5	1 = Log MMS-to-slave communications loss

Local or Remote Control of MMS Relays

The LOCLRMT register at location 41484 determines how the output relays of the MMS are controlled. Local control means the RUN buttons on the MMS can be usd to turn on the control relays. Remote control means that only a PLC is allowed to control the relays. In all cases, the STOP / RESET button is active.

Table 69: LOCLRMT Register

0 = Semi-remote control (Both Local and Network control are permitted)
1 = Local control only (Network control is not permitted)
2 = Full remote control only (Local control is not permitted)

Startup Control of MMS Relays

Register STRTOPT at location 41485 determines whether or not the relays automatically restart.

Table 70: STRTOPT Register

Bit 0	1 = Hold last state of relays 0 = Manual start

Schneider Electric USA
8001 Knightdale Boulevard
Knightdale, NC, 27545 USA
1-888-SquareD (1-888-778-2733)
www.us.SquareD.com

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

02/2007