For the mobile version of this guide, see:
animatics.com/docs/guides-html/c56_mbus_rtu/

SMART Motor™

Modbus® RTU Guide

Class 5 & 6 SmartMotor Technology

MOOG ANIMATICS

Copyright Notice

Moog Animatics Class 5 & 6 SmartMotor™ Modbus RTU Guide, Rev. C, PN: SC80100014-001.

Please let us know if you find any errors or omissions in this manual so that we can improve it for future readers. Such notifications should contain the words "Modbus RTU Guide" in the subject line and be sent by e-mail to: animatics_marcom@moog.com. Thank you in advance for your contribution.

Contact Us:

Americas - West
Moog Animatics
2581 Leghorn Street
Mountain View, CA 94043
USA
Tel: 1 650-960-4215

Americas - East
Moog Animatics
750 West Sproul Road
Springfield, PA 19064
USA
Tel: 1 610-328-4000 x3999
Fax: 1 610-605-6216

Support: 1 (888) 356-0357

Website: www.animatics.com

Email: animatics_sales@moog.com

Table Of Contents

Introduction

This chapter provides information on the purpose and scope of this manual. It also provides information on safety notation, related documents and additional resources.

Purpose

This Modbus® guide describes the Modbus RTU protocol support provided by the Moog Animatics Class 5 & 6 SmartMotor™. It describes the major concepts that must be understood to integrate a SmartMotor slave with a PLC or other Modbus RTU (Remote Terminal Unit) master. However, it does not cover all the low-level details of the Modbus RTU protocol.

> **NOTE:** The feature set described in this version of the manual refers to motor firmware 5.x.4.31 or higher and 6.0.2.26 or higher.

This manual is intended for programmers or system developers who have read and understand the *Modbus Serial Line Protocol and Implementation Guide V1.02*, which is published and maintained by Modbus.org. Therefore, this manual is not a tutorial on that specification or the Modbus RTU protocol. Instead, it should be used to understand the specific implementation details for the Moog Animatics SmartMotor. For a general overview of Modbus RTU, see the FAQ page and other resources at www.modbus.org.

Safety Information

This section describes the safety symbols and other safety information.

Safety Symbols

The manual may use one or more of the following safety symbols:

WARNING: This symbol indicates a potentially nonlethal mechanical hazard, where failure to follow the instructions could result in serious injury to the operator or major damage to the equipment.

CAUTION: This symbol indicates a potentially minor hazard, where failure to follow the instructions could result in slight injury to the operator or minor damage to the equipment.

NOTE: Notes are used to emphasize non-safety concepts or related information.

Other Safety Considerations

The Moog Animatics SmartMotors are supplied as components that are intended for use in an automated machine or system. As such, it is beyond the scope of this manual to attempt to cover all the safety standards and considerations that are part of the overall machine/system design and manufacturing safety. Therefore, the following information is intended to be used only as a general guideline for the machine/system designer.

It is the responsibility of the machine/system designer to perform a thorough "Risk Assessment" and to ensure that the machine/system and its safeguards comply with the safety standards specified by the governing authority (for example, ISO, OSHA, UL, etc.) for the locale where the machine is being installed and operated. For more details, see Machine Safety on page 8.

Motor Sizing

It is the responsibility of the machine/system designer to select SmartMotors that are properly sized for the specific application. Undersized motors may: perform poorly, cause excessive downtime or cause unsafe operating conditions by not being able to handle the loads placed on them. The *System Best Practices* document, which is available on the Moog Animatics website, contains information and equations that can be used for selecting the appropriate motor for the application.

Replacement motors must have the same specifications and firmware version used in the approved and validated system. Specification changes or firmware upgrades require the approval of the system designer and may require another Risk Assessment.

Environmental Considerations

It is the responsibility of the machine/system designer to evaluate the intended operating environment for dust, high-humidity or presence of water (for example, a food-processing environment that requires water or steam wash down of equipment), corrosives or chemicals that may come in contact with the machine, etc. Moog Animatics manufactures specialized

IP-rated motors for operating in extreme conditions. For details, see the *Moog Animatics Product Catalog*.

Machine Safety

In order to protect personnel from any safety hazards in the machine or system, the machine/system builder must perform a "Risk Assessment", which is often based on the ISO 13849 standard. The design/implementation of barriers, emergency stop (E-stop) mechanisms and other safeguards will be driven by the Risk Assessment and the safety standards specified by the governing authority (for example, ISO, OSHA, UL, etc.) for the locale where the machine is being installed and operated. The methodology and details of such an assessment are beyond the scope of this manual. However, there are various sources of Risk Assessment information available in print and on the internet.

> **NOTE:** The following list is an example of items that would be evaluated when performing the Risk Assessment. Additional items may be required. The safeguards must ensure the safety of all personnel who may come in contact with or be in the vicinity of the machine.

In general, the machine/system safeguards must:

- Provide a barrier to prevent unauthorized entry or access to the machine or system. The barrier must be designed so that personnel cannot reach into any identified danger zones.
- Position the control panel so that it is outside the barrier area but located for an unrestricted view of the moving mechanism. The control panel must include an E-stop mechanism. Buttons that start the machine must be protected from accidental activation.
- Provide E-stop mechanisms located at the control panel and at other points around the perimeter of the barrier that will stop all machine movement when tripped.
- Provide appropriate sensors and interlocks on gates or other points of entry into the protected zone that will stop all machine movement when tripped.
- Ensure that if a portable control/programming device is supplied (for example, a hand-held operator/programmer pendant), the device is equipped with an E-stop mechanism.

  > **NOTE:** A portable operation/programming device requires *many* additional system design considerations and safeguards beyond those listed in this section. For details, see the safety standards specified by the governing authority (for example, ISO, OSHA, UL, etc.) for the locale where the machine is being installed and operated.

- Prevent contact with moving mechanisms (for example, arms, gears, belts, pulleys, tooling, etc.).
- Prevent contact with a part that is thrown from the machine tooling or other part-handling equipment.
- Prevent contact with any electrical, hydraulic, pneumatic, thermal, chemical or other hazards that may be present at the machine.
- Prevent unauthorized access to wiring and power-supply cabinets, electrical boxes, etc.

- Provide a proper control system, program logic and error checking to ensure the safety of all personnel and equipment (for example, to prevent a run-away condition). The control system must be designed so that it does not automatically restart the machine/system after a power failure.
- Prevent unauthorized access or changes to the control system or software.

Documentation and Training

It is the responsibility of the machine/system designer to provide documentation on safety, operation, maintenance and programming, along with training for all machine operators, maintenance technicians, programmers, and other personnel who may have access to the machine. This documentation must include proper lockout/tagout procedures for maintenance and programming operations.

It is the responsibility of the operating company to ensure that:

- All operators, maintenance technicians, programmers and other personnel are tested and qualified before acquiring access to the machine or system.
- The above personnel perform their assigned functions in a responsible and safe manner to comply with the procedures in the supplied documentation and the company safety practices.
- The equipment is maintained as described in the documentation and training supplied by the machine/system designer.

Additional Equipment and Considerations

The Risk Assessment and the operating company's standard safety policies will dictate the need for additional equipment. In general, it is the responsibility of the operating company to ensure that:

- Unauthorized access to the machine is prevented at all times.
- The personnel are supplied with the proper equipment for the environment and their job functions, which may include: safety glasses, hearing protection, safety footwear, smocks or aprons, gloves, hard hats and other protective gear.
- The work area is equipped with proper safety equipment such as first aid equipment, fire suppression equipment, emergency eye wash and full-body wash stations, etc.
- There are no modifications made to the machine or system without proper engineering evaluation for design, safety, reliability, etc., and a Risk Assessment.

Safety Information Resources

Additional SmartMotor safety information can be found on the Moog Animatics website; open the file "109_Controls, Warnings and Cautions.pdf" located at:

http://www.animatics.com/support/moog-animatics-catalog.html

OSHA standards information can be found at:

https://www.osha.gov/law-regs.html

ANSI-RIA robotic safety information can be found at:

http://www.robotics.org/robotic-content.cfm/Robotics/Safety-Compliance/id/23

UL standards information can be found at:

http://ulstandards.ul.com/standards-catalog/

ISO standards information can be found at:

http://www.iso.org/iso/home/standards.htm

EU standards information can be found at:

http://ec.europa.eu/growth/single-market/european-standards/harmonised-standards/index_en.htm

Additional Documents

The Moog Animatics website contains additional documents that are related to the information in this manual. Please refer to the following list.

Related Guides

- *Class 5 SmartMotor™ Installation & Startup Guide*

  http://www.animatics.com/cl-5-install-startup-guide

- *Class 6 SmartMotor™ Installation & Startup Guide*

  http://www.animatics.com/cl-6-install-startup-guide

- *SmartMotor™ Developer's Guide*

  http://www.animatics.com/smartmotor-developers-guide

Other Documents

- *SmartMotor™ System Best Practices*

  http://www.animatics.com/system-best-practices-application-note

- *SmartMotor™ Product Certificate of Conformance*

  http://www.animatics.com/download/Declaration of Conformity.pdf

- *SmartMotor™ UL Certification*

  http://www.animatics.com/download/MA_UL_online_listing.pdf

- *SmartMotor Developer's Worksheet*
  (interactive tools to assist developer: Scale Factor Calculator, Status Words, CAN Port Status, Serial Port Status, RMODE Decoder and Syntax Error Codes)

  http://www.animatics.com/tools

- *Moog Animatics Product Catalog*

  http://www.animatics.com/support/moog-animatics-catalog.html

Additional Resources

The Moog Animatics website contains useful resources such as product information, documentation, product support and more. Please refer to the following addresses:

- General company information:

  http://www.animatics.com

- Product information:

  http://www.animatics.com/products.html

- Product support (Downloads, How To videos, Forums, Knowledge Base, and FAQs):

  http://www.animatics.com/support.html

- Sales and distributor information:

  http://www.animatics.com/sales-offices.html

- Application ideas (including videos and sample programs):

  http://www.animatics.com/applications.html

Modbus Resources

Modbus is a common standard maintained by Modbus.org:

- Modbus.org website:

  http://www.modbus.org

Motor Pinouts, Connections and Status LEDs

The following sections describe the motor pinouts, system connections and the status LEDs.

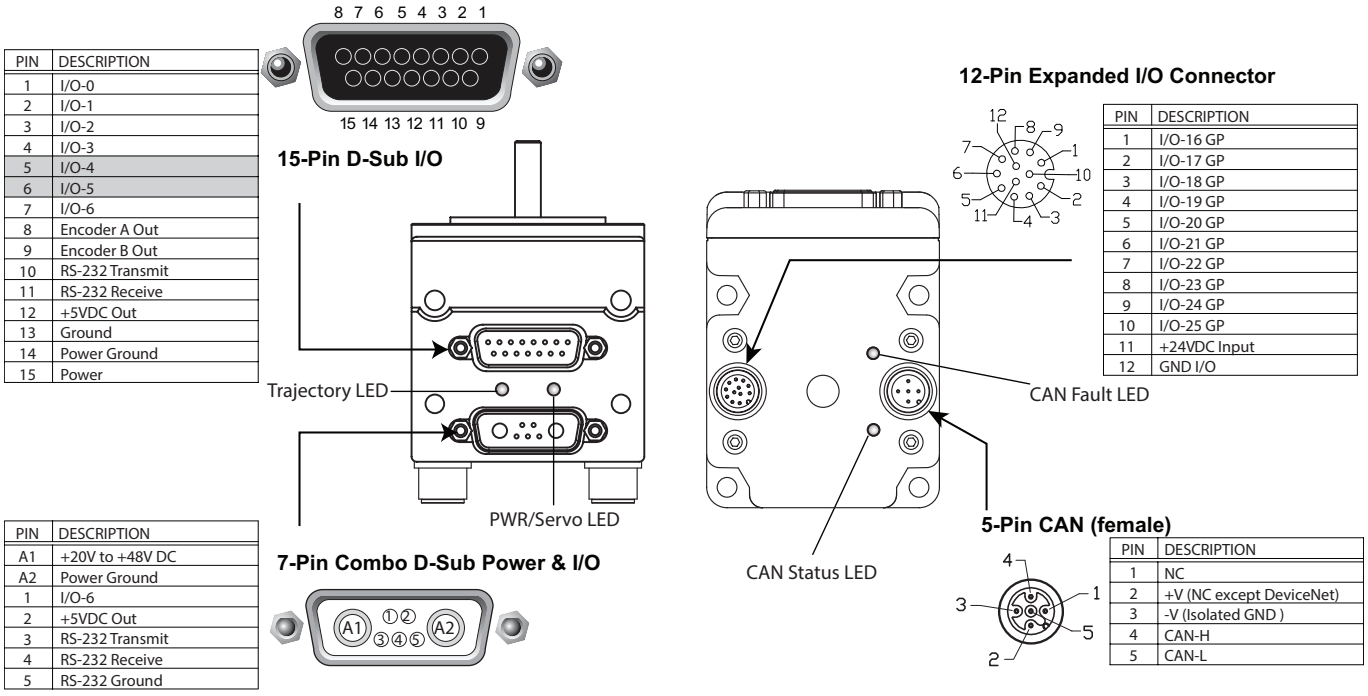
Connecting the System (RTU)

The following sections show system connections and cable diagrams for typical Class 5 D-style and M-style motors and Class 6 M-style motors.

Class 5 D-Style Motors: Connectors and Pinouts

The following figure provides a brief overview of the connectors and pinouts available on the D-style SmartMotors. Additional connector specifications are available in the Class 5 *SmartMotor™ Installation & Startup Guide*.

I/O 4 and 5 (pins 5 and 6 on the 15-pin D-sub I/O connector) are the RS-485 connections for Modbus RTU.



PIN	DESCRIPTION
1	I/O-0
2	I/O-1
3	I/O-2
4	I/O-3
5	I/O-4
6	I/O-5
7	I/O-6
8	Encoder A Out
9	Encoder B Out
10	RS-232 Transmit
11	RS-232 Receive
12	+5VDC Out
13	Ground
14	Power Ground
15	Power

PIN	DESCRIPTION
1	I/O-16 GP
2	I/O-17 GP
3	I/O-18 GP
4	I/O-19 GP
5	I/O-20 GP
6	I/O-21 GP
7	I/O-22 GP
8	I/O-23 GP
9	I/O-24 GP
10	I/O-25 GP
11	+24VDC Input
12	GND I/O

PIN	DESCRIPTION
A1	+20V to +48V DC
A2	Power Ground
1	I/O-6
2	+5VDC Out
3	RS-232 Transmit
4	RS-232 Receive
5	RS-232 Ground

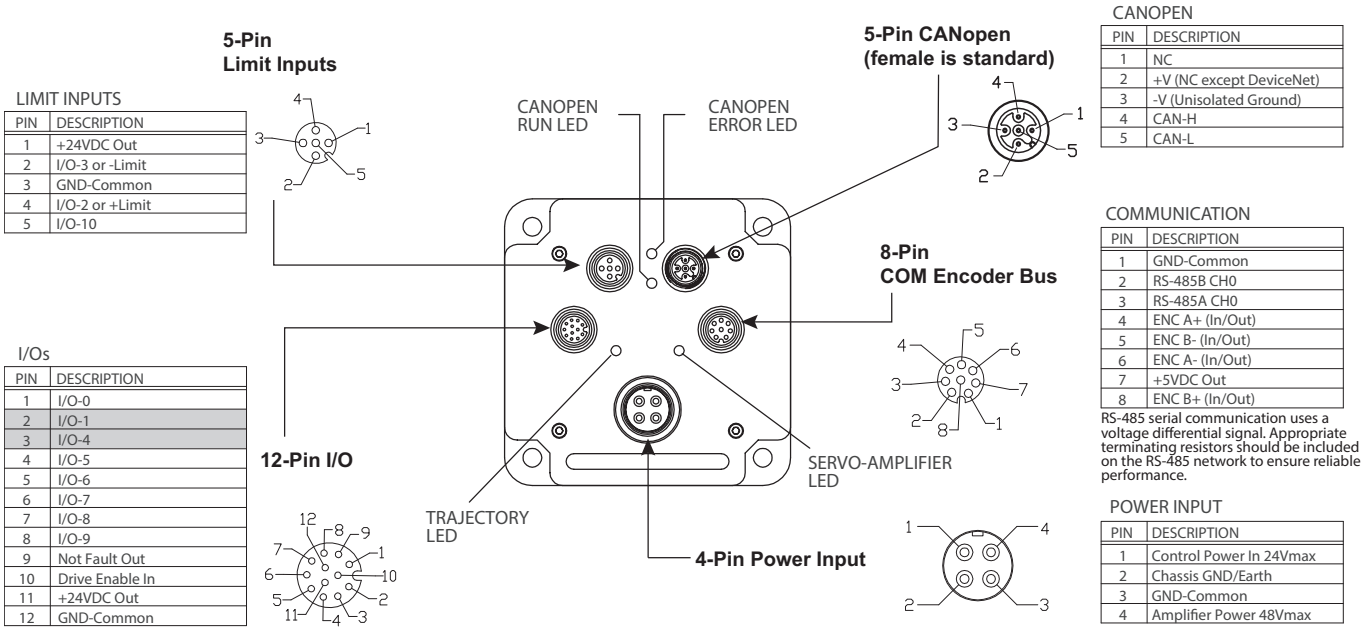
PIN	DESCRIPTION
1	NC
2	+V (NC except DeviceNet)
3	-V (Isolated GND)
4	CAN-H
5	CAN-L

NOTE: The DE power option is recommended. For details, see the Class 5 *SmartMotor™ Installation & Startup Guide*.

Class 5 M-Style Motors: Connectors and Pinouts

The following figure provides a brief overview of the connectors and pinouts available on the M-style SmartMotors. Additional connector specifications are available in the Class 5 *SmartMotor™ Installation & Startup Guide*.

RS-485 channel 0 (pins 2 and 3 on the Communication connector) is the RS-485 connection for Modbus.



LIMIT INPUTS

PIN	DESCRIPTION
1	+24VDC Out
2	I/O-3 or -Limit
3	GND-Common
4	I/O-2 or +Limit
5	I/O-10

I/Os

PIN	DESCRIPTION
1	I/O-0
2	I/O-1
3	I/O-4
4	I/O-5
5	I/O-6
6	I/O-7
7	I/O-8
8	I/O-9
9	Not Fault Out
10	Drive Enable In
11	+24VDC Out
12	GND-Common

CANOPEN

PIN	DESCRIPTION
1	NC
2	+V (NC except DeviceNet)
3	-V (Unisolated Ground)
4	CAN-H
5	CAN-L

COMMUNICATION

PIN	DESCRIPTION
1	GND-Common
2	RS-485B CH0
3	RS-485A CH0
4	ENC A+ (In/Out)
5	ENC B- (In/Out)
6	ENC A- (In/Out)
7	+5VDC Out
8	ENC B+ (In/Out)

RS-485 serial communication uses a voltage differential signal. Appropriate terminating resistors should be included on the RS-485 network to ensure reliable performance.

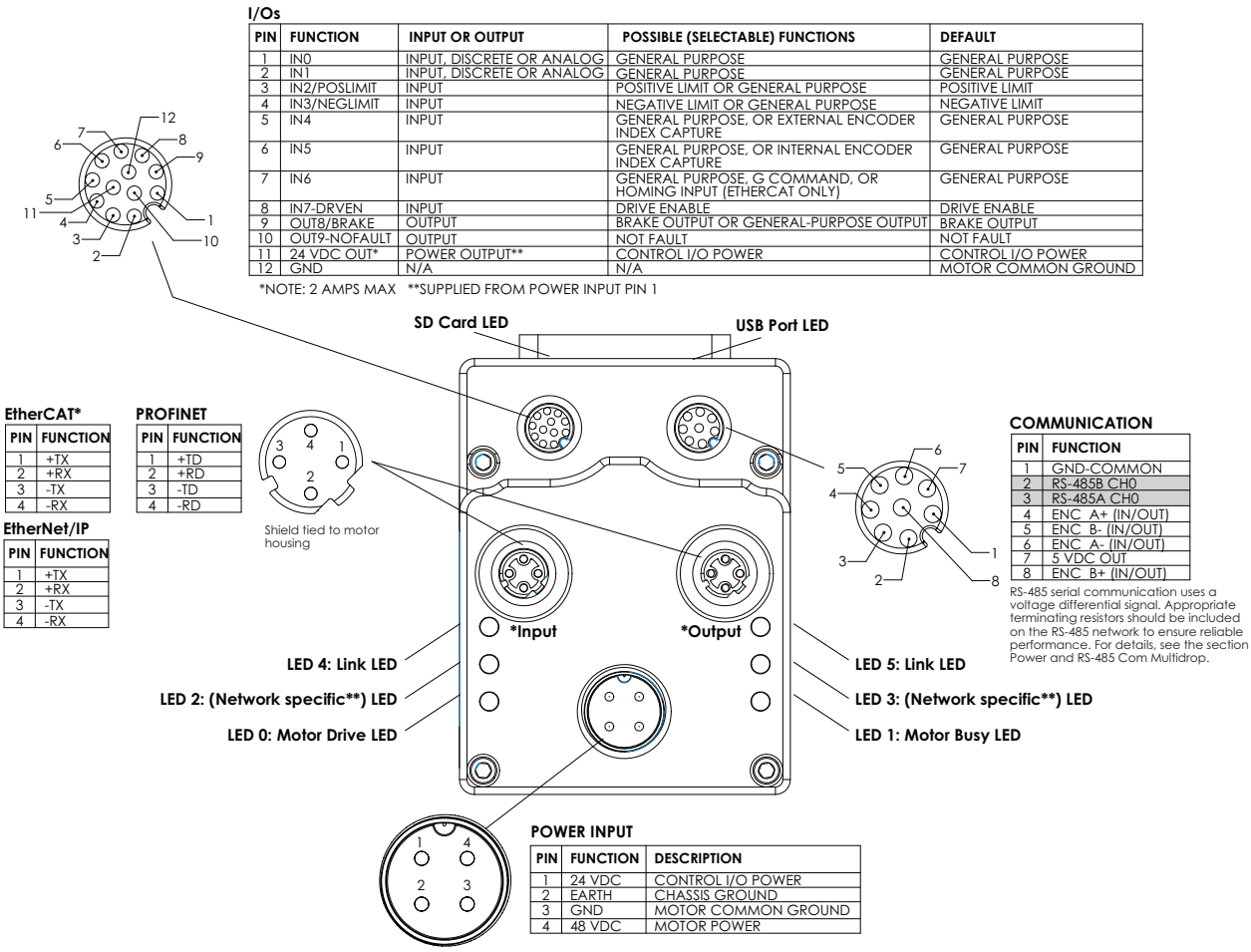
POWER INPUT

PIN	DESCRIPTION
1	Control Power In 24Vmax
2	Chassis GND/Earth
3	GND-Common
4	Amplifier Power 48Vmax

Class 6 M-Style Motors: Connectors and Pinouts

The following figure provides a brief overview of the connectors and pinouts available on the Class 6 M-style SmartMotors. Additional connector specifications are available in the *Class 6SmartMotor™ Installation & Startup Guide*.

RS-485 channel 0 (pins 2 and 3 on the Communication connector) is the RS-485 connection for Modbus.

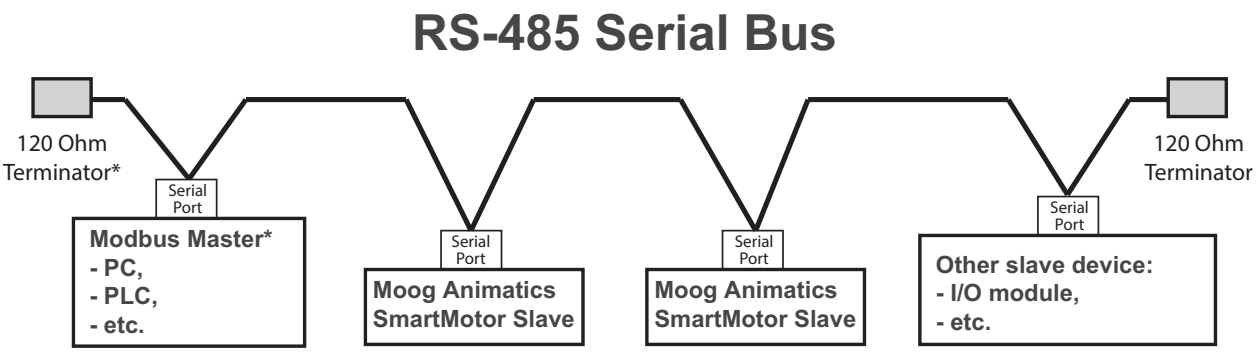


**The Input/Output applies to these networks. Refer to the following CAUTIONS.*
***For LED Status information, refer to the corresponding SmartMotor fieldbus guide.*

Cable Diagram

The following figure shows a Modbus RTU master connected to a serial daisy chain of slave devices. Although different network topologies are possible, the daisy chain provides the most reliable performance. If drops are made from the main trunk line, they should be kept as short as possible.

> **NOTE:** When calculating the overall (total) cable length, you must account for all cable segments in the network.



***Master may have termination option; see master's documentation for details.**

> **NOTE:** RS-485 serial communications uses a voltage differential signal that requires proper termination with a 120 ohm resistor at both ends of the network cable. This follows RS-485 standards for biasing to ensure reliable performance.

Maximum Cable Length

> **NOTE:** When calculating the overall (total) cable length, you must account for all cable segments in the network.

Moog Animatics recommends a maximum cable length of 1000 meters or a maximum baud rate of 115200. As baud rate increases, the maximum cable length decreases. The maximum cable length allowed depends on the baud rate, gauge and other physical properties of the cable, operating environment and other factors.

For more details, see the *Modbus Serial Line Protocol and Implementation Guide V1.02*

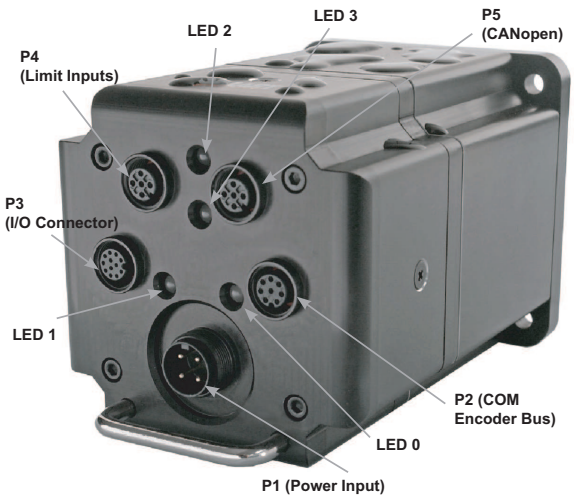
Understanding the Status LEDs

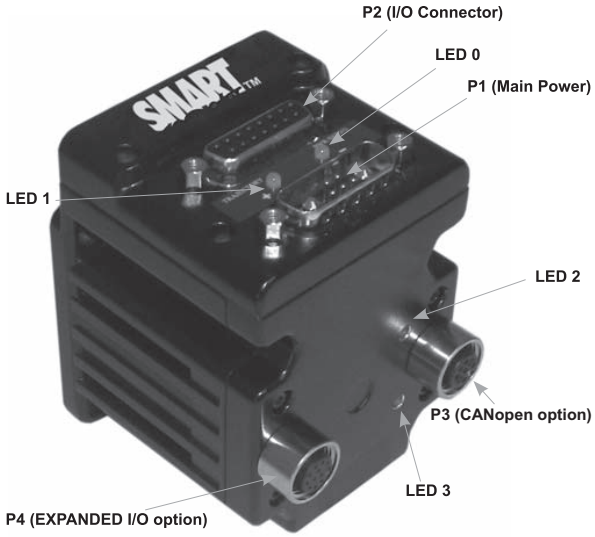
This section describes the Modbus RTU status LEDs for the Class 5 and Class 6 motors.

Class 5 Motors

The Status LEDs provide the same functionality for the D-style and M-style (including IP-sealed) SmartMotors.

> **NOTE:** If the motor is equipped with the CANopen option, LEDs 2 and 3 do not apply to Modbus RTU.





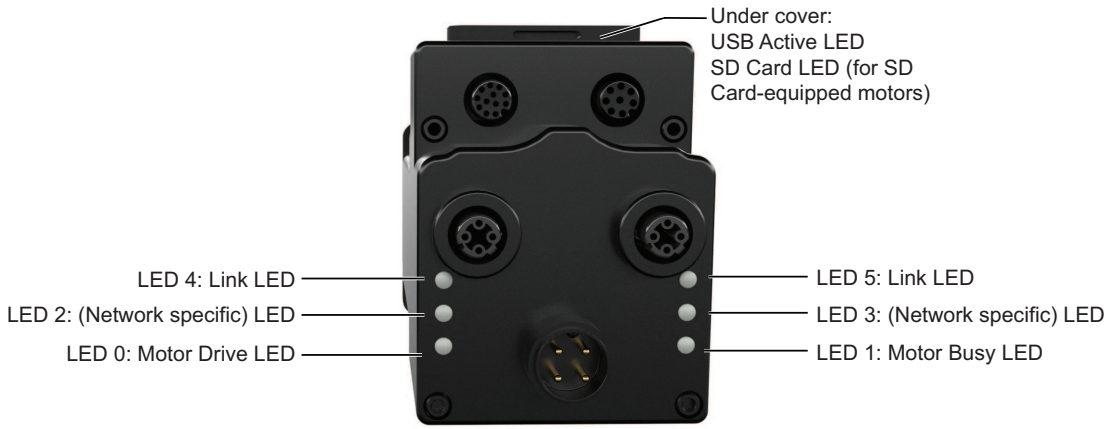
LED 0:	Drive Status Indicator	
	Off	No power
	Solid green	Drive on
	Flashing green	Drive off
	Flashing red	Watchdog fault
	Solid red	Major fault
	Alt. red/green	In boot load; needs firmware
LED 1:	Trajectory Status Indicator	
	Off	Not busy
	Solid green	Drive on, trajectory in progress

LED Status on Power-up:

- With no program and the travel limit inputs are low:
  LED 0 will be solid red indicating the motor is in a fault state due to travel limit fault.
  LED 1 will be off.
- With no program and the travel limit inputs are high:
  LED 0 will be solid red for 500 milliseconds and then begin flashing green.
  LED 1 will be off.
- With a program that disables only travel limits and nothing else:
  LED 0 will be solid red for 500 milliseconds and then begin flashing green.
  LED 1 will be off.

Class 6 Motors

The following figure and tables describe the functionality of the Status LEDs on the Class 6 SmartMotor.



Flickering = On/Off in 0.1 sec; Blinking = On/Off in 0.5 sec; Flashing = separated by 1 sec for EtherCAT LEDs and 2 sec for Fault Codes

SD Card LED (for SD Card-equipped motors)

Off	No card, bad or damaged card
Blinking green	Busy, do not remove card
Solid green	Card detected
Solid red	Card with no SmartMotor data

USB Active LED

Flashing green	Active
Flashing red	Suspended
Solid red	USB power detected, no configuration

LED 0: Motor Drive LED

Off	No power
Solid green	Drive on
Blinking green	Drive off, no faults
Triple red flash	Watchdog fault
Solid red	Faulted or no drive enable input

LED 2: (Network specific) LED
Refer to the corresponding SmartMotor fieldbus guide

LED 4: Link LED
Refer to the corresponding SmartMotor fieldbus guide

LED 1: Motor Busy LED

Off	Not busy
Solid green	Drive on, trajectory in progress
Flashing # red	Flashes fault code* (see below) when Drive LED is solid red

LED 3: (Network specific) LED
Refer to the corresponding SmartMotor fieldbus guide

LED 5: Link LED
Refer to the corresponding SmartMotor fieldbus guide

LED Status on Power-up:

- With no program and the travel limit inputs are low:
  LED 0 solid red; motor is in fault state due to travel limit fault
  LED 1 off
- With no program and the travel limits are high:
  LED 0 solid red for 500 milliseconds then flashing green
  LED 1 off
- With a program that only disables travel limits:
  LED 0 red for 500 milliseconds then flashing green
  LED 1 off

LED 1 Fault Codes:

Flash	Description
1	NOT Used
2	Bus Voltage
3	Over Current
4	Excessive Temperature
5	Excessive Position
6	Velocity Limit
7	dE/Dt - First derivative of position error is excessive
8	Hardware Positive Limit Reached
9	Hardware Negative Limit Reached
10	Software Positive Travel Limit Reached
11	Software Negative Travel Limit Reached

*Busy LED pauses for 2 seconds before flashing the code

Using Modbus

The following sections describe how to enable Modbus communications with your SmartMotor, along with information on supported function codes, input registers and holding registers.

Modbus RTU Description

Modbus RTU is a standard that allows industrial devices to communicate over serial connections. The Moog Animatics Class 5 & 6 SmartMotors support communication to a PLC, HMI, or other host device over serial RS-485 only.

> **NOTE:** The Moog Animatics Class 6 SmartMotor also supports the Modbus TCP/IP protocol over Ethernet TCP/IP connections. Refer to that guide for details.

OCHN Command

The OCHN command is used to open the serial port with Modbus RTU support. Refer to the following examples. By default, SmartMotor serial ports are not open in this mode.

The Class 5 & 6 SmartMotors will act as a slave devices in such a network. User integer variables have read/write access as word (16-bit), or long (32-bit values). Status words can be read as 16-bit words. Also, subroutines may be called via GOSUB(value).

> **NOTE:** In this guide, hexadecimal numbers are prefixed by 0x. Therefore 0x0001 is a hexadecimal one in 16-bit representation, and 0x00000001 is a hexadecimal one in 32-bit representation.

M-style Motor Example

```
OCHN(MB4,0,N,115200,1,8,D) ' Modbus Com0 M-series circular connector.
```

> **NOTE:** Because the M-style SmartMotor has only one serial port, it is mutually exclusive with applications that need to interact with the SMI software over serial communications. If the above code is used in an M-style motor, it may be necessary to use the SMI software Communication Lockup Wizard to disable to program when SMI is needed to configure the motor.

D-style Motor Example

```
OCHN(MB4,1,N,115200,1,8,D) ' Modbus Com1 D-series 15-pin connector.
SADDR1 ' Modbus uses this address to identify itself.
```

For more details on the OCHN command, see the *SmartMotor™ Developer's Guide*.

Legacy Modbus RTU Discussion

Memory/registers designated as being in the 4X space are referred to as read/write space. This is an association consistent with legacy Modbus RTU.

Memory/registers designated as being in the 3X space are referred to as read-only space. This is an association consistent with legacy Modbus RTU.

For Class 5 & 6 SmartMotor access to resources described in this document, note that zero-based addressing is used. At the Modbus network level, this is the address that is transmitted. Some controllers may ask the user to specify addresses that are not zero-based. The address in the 4X space of 40001 is actually 0000(hex) on the network. Further, because legacy space is small, a network address of 8000(hex) is converted to a 4X reference as follows:

32768 + 40001 = 72769 (some Modbus RTU tools may do this)

Likewise, in the 3X space, the network address of 8000(hex) is converted to legacy format as follows:

32768 + 30001 = 62769

Note that address offsets are actually separated by function codes in the 4X and 3X legacy reference spaces. Therefore, what looks like an overlap in controller memory is not. The controller memory has been separated to respect the convention of read/write memory and read-only memory, 4X and 3X, respectively.

Some controllers may handle this differently. Therefore, it is the responsibility of the system programmer to be aware of the method used in the host controller. The SmartMotor simply expects zero-based addresses and is not aware of any translation that the host may conduct.

Supported Function Codes

A small set of Modbus function codes are supported for simple access to variables and status words. The GOSUB feature of the AniBasic language can be accessed through register write as well.

16-Bit Access

The following table shows the codes, descriptions and functions for 16-bit access.

Code	Description	Function
03	Read Holding Registers (4X space)	Read 16-bit value or values.
04	Read Input Registers (3X space)	Read 16-bit (read-only) value or values.
06	Write Single Register (4X space)	Write 16-bit value or values.
16	Write Multiple Registers (4X space)	Write 16-bit value or values.

32-Bit Access

The following table shows the codes, descriptions and functions for 32-bit access.

Code	Description	Function
03	Read Holding Registers (4X space)	Read 32-bit value or values.
16	Write Multiple Registers (4X space)	Write 32-bit value or values.
NOTE: Low word of 32-bit values is stored at lower Modbus address.		

Input Registers - 3X

The Modbus 3X input registers are 16-bit registers used to read data to the PLC (i.e., they are read only). Regarding the SmartMotor, the set of data that can be read includes the Moog Animatics AniBasic "RW(x)" status words — the physical I/O state inputs RW(16) and, optionally, RW(17), and other RW(x) status words. Refer to the following table.

3X Mapping

The following table describes the 3X mapping.

Address (hex)	Byte #	Description	Comments
0x0000	2	Status Register 0	Drive state and hardware limits
0x0001	2	Status Register 1	Index capture and software limits
0x0002	2	Status Register 2	Programs and communications
0x0003	2	Status Register 3	PID and motion
0x0004	2	Status Register 4	Timers
0x0005	2	Status Register 5	Interrupts
0x0006	2	Status Register 6	Commutation and bus
0x0007	2	Status Register 7	Trajectory details
0x0008	2	Status Register 8	Cam and interpolation user bits
0x0009	2	Status Register 9	N/A
0x000a	2	Status Register 10	N/A
0x000b	2	Status Register 11	N/A
0x000c	2	Status Register 12	User bits word 0
0x000d	2	Status Register 13	User bits word 1
0x000e	2	Status Register 14	N/A
0x000f	2	Status Register 15	N/A
0x0010	2	Status Register 16	I/O state, word 0
0x0011	2	Status Register 17	I/O state, word 1 (D-style with AD1 option only)
NOTES: 1. Addresses shown are zero-based. Legacy Modbus addresses may be translated differently by the host controller. 2. Refer to the *SmartMotor Developer's Guide* for a full description of status word functionality.			

LIMITATIONS: up to 29 words can be read at a time (for the purposes of the input registers, reading is only meaningful up to the index shown in the previous table.)

Holding Registers - 4X

The Modbus 4X holding registers are 16-bit registers used to read data to and write data from the PLC. Regarding the SmartMotor, the set of data that can be read/written includes the Moog Animatics AniBasic variables a-zzz, ab, aw and al, and the GOSUB command. Refer to the following table.

4X Mapping

The following table describes the 4X mapping.

Address (hex)	Byte #	AniBasic Command Description	Comments
0x2000-2033	-	a to z	User memory
0x2034-2067	-	aa to zz	User memory
0x2068-209B	-	aaa to zzz	User memory, includes zzz
0x209C-0x2101		ab[0]-ab[203] al[0]-al[50] aw[0]-aw[101]	User memory array
0x8004		GOSUB(label)	Execute subroutine specified by label
NOTES: 1. Addresses shown are zero-based. Legacy Modbus addresses may be translated differently by the host controller. 2. User memory is word-addressable only. The low-addressed word is the lower half of a 32-bit number in the controller.			

LIMITATIONS: up to 29 words can be read at a time. However, if accessing SmartMotor variables a, b, c, etc. (which are 2 words each as 32-bit variables), then 14 variables can be accessed in a read operation. Writing multiple registers has a restriction of up to 27 words (13 variables that are 32-bits each).

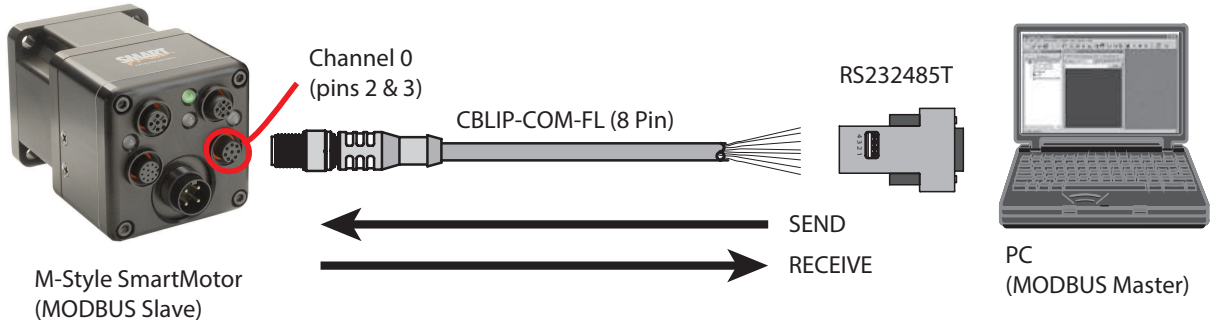
Modbus RTU Communications Example

This topic contains Modbus communications examples.

Modbus RTU Communication Setup

This section describes a typical setup for Modbus RTU (serial) communications.

- Connect the SmartMotor's RS-485 pins to the PLC, HMI or other device that is serving as the Modbus master. For details, refer to Connecting the System (RTU) on page 13.
    - For D-style motors, I/O 4 and 5 (pins 5 and 6 on the 15-pin D-sub I/O connector) are the RS-485 connections for Modbus RTU.
    - For M-style motors, RS-485 channel 0 (pins 2 and 3 on the Communication connector) is the RS-485 connection for Modbus.
- Verify that the OCHN (Open Channel) command is in a user program in the connected SmartMotor. For details, see OCHN Command on page 20.
- Verify that the SmartMotor's serial address is also used for the Modbus slave ID (i.e., both the motor address and Modbus slave ID must match). The SADDR= command is used in the program to set the SmartMotor serial address. Refer to the SADDR example in OCHN Command on page 20.
- For testing, you can use a PC as the Modbus master along with the free utility program QModBus (http://qmodbus.sourceforge.net/). Refer to the following diagram.



Modbus RTU Communication Test Example

Although this wouldn't be used for a real application, it allows you to communicate with a SmartMotor as the Modbus RTU slave. For examples, see Modbus RTU Sample Command Sequences on page 25.

Modbus RTU Sample Command Sequences

This topic contains some sample Modbus RTU (serial) command sequences. These examples show the data sent from and received by the Modbus master communicating with a SmartMotor. For these examples, a utility software, QModBus, is used to simulate the master, and the SmartMotor uses Slave ID 5.

For each of the following sections:

- Section title = action being performed
- SEND to motor = formatted byte stream sent from master to the SmartMotor
- RECV from motor = formatted byte stream received by the master from the SmartMotor

For each of the following tables:

> **NOTE:** A table is provided to illustrate the parts of the byte sequence only. The byte sequence must be transmitted as a stream of bytes shown in the SEND/RECEIVE strings above the table (i.e., no pause or null for the blank cells).

- Slave ID = device node address
- Funct = function code (see Supported Function Codes on page 21)
- Start Addr = start address in memory or single register address (see Input Registers - 3X on page 22 and Holding Registers - 4X on page 23)
- No. of Reg. = number of coils or number of registers
- Byte Cnt =byte count
- Data low word = data (low word)
- Data high word = data (high word)
- CRC = cyclic redundancy check

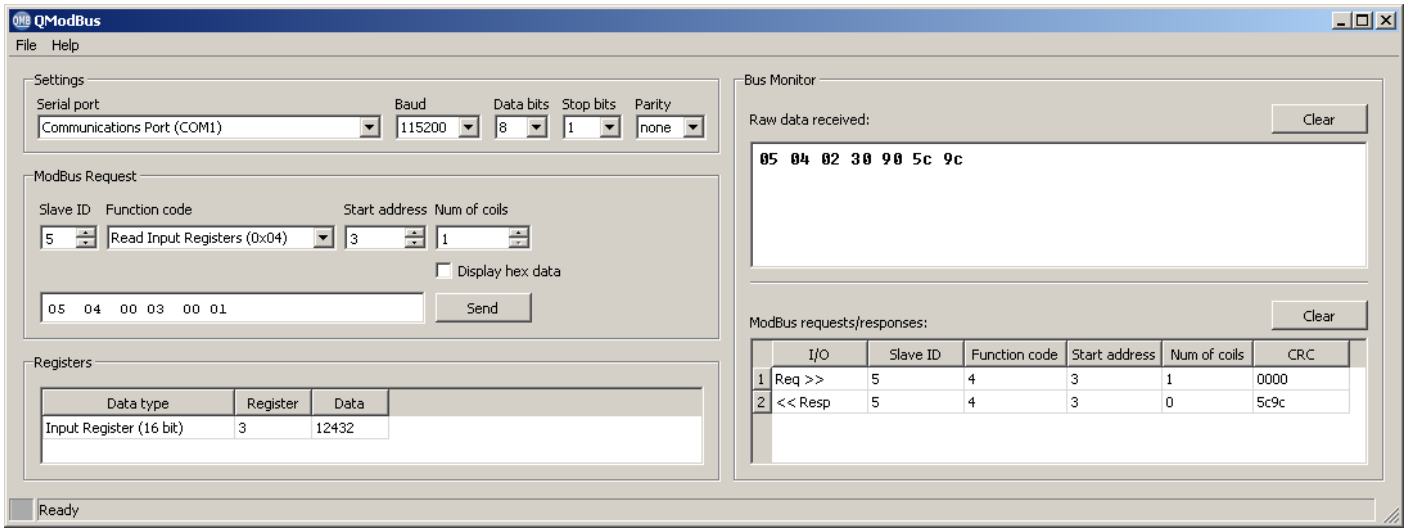
Read input registers (status word 3)

> **NOTE:** For information on input registers, see Input Registers - 3X on page 22.

SEND to motor: 05 04 00 03 00 01 C0 4E

RECV from motor: 05 04 02 30 90 5C 9C

	Slave ID	Funct	Start Addr	No. of Reg.	Byte Cnt	Data low word	Data high word	CRC
SEND	05	04	00 03	00 01				C0 4E
RECV	05	04			02	30 90		5C 9C
A table is provided to illustrate the parts of the byte sequence only. The byte sequence must be transmitted as a stream of bytes shown in the SEND/RECEIVE strings above the table (i.e., no pause or null for the blank cells).								



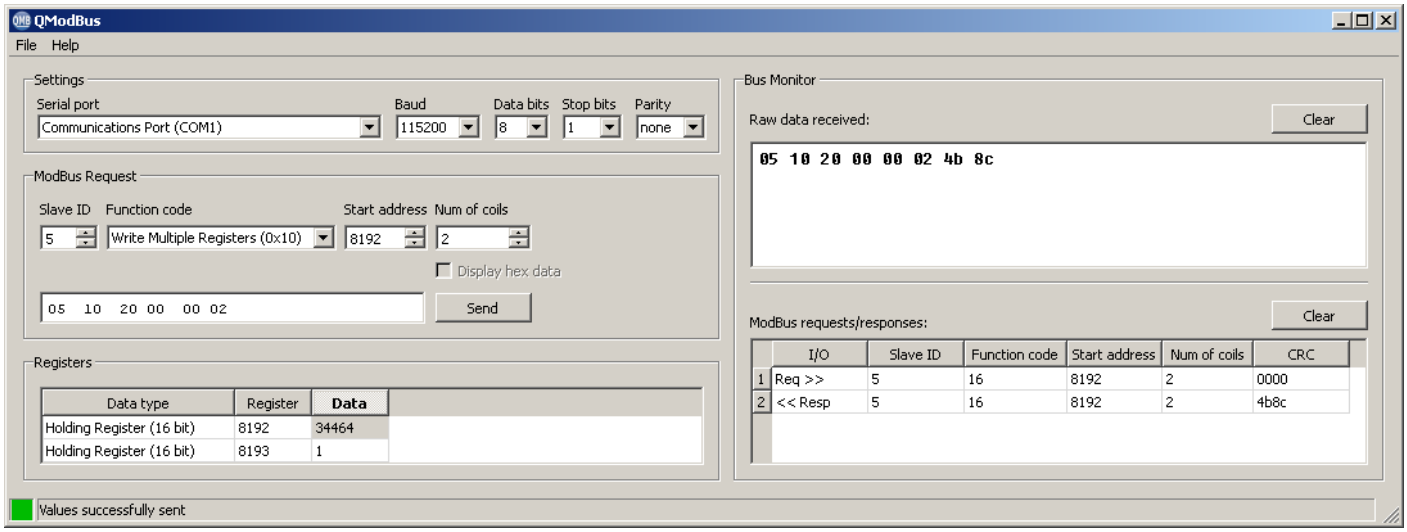
QModBus Utility Showing SEND / RECEIVE Data

Write variable "a" (a=100000)

SEND to motor: 05 10 20 00 00 02 04 86 A0 00 01 97 F4

RECV from motor: 05 10 20 00 00 02 4B 8C

	Slave ID	Funct	Start Addr	No. of Reg.	Byte Cnt	Data low word	Data high word	CRC
SEND	05	10	20 00	00 02	04	86 A0	00 01	97 F4
RECV	05	10	20 00	00 02				4B 8C
A table is provided to illustrate the parts of the byte sequence only. The byte sequence must be transmitted as a stream of bytes shown in the SEND/RECEIVE strings above the table (i.e., no pause or null for the blank cells).								



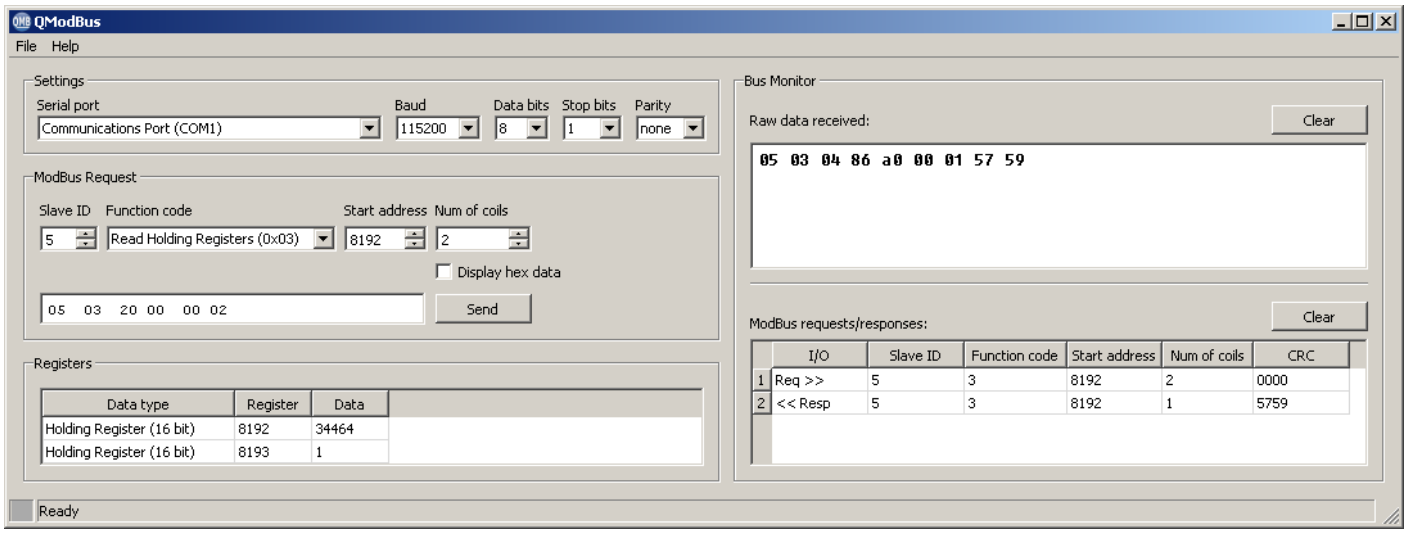
QModBus Utility Showing SEND / RECEIVE Data

Read variable "a" (value returned is 100000)

SEND to motor: 05 03 20 00 00 02 CE 4F

RECV from motor: 05 03 04 86 A0 00 01 57 59

	Slave ID	Funct	Start Addr	No. of Reg.	Byte Cnt	Data low word	Data high word	CRC
SEND	05	03	20 00	00 02				CE 4F
RECV	05	03			04	86 A0	00 01	57 59
A table is provided to illustrate the parts of the byte sequence only. The byte sequence must be transmitted as a stream of bytes shown in the SEND/RECEIVE strings above the table (i.e., no pause or null for the blank cells).								



QModBus Utility Showing SEND / RECEIVE Data

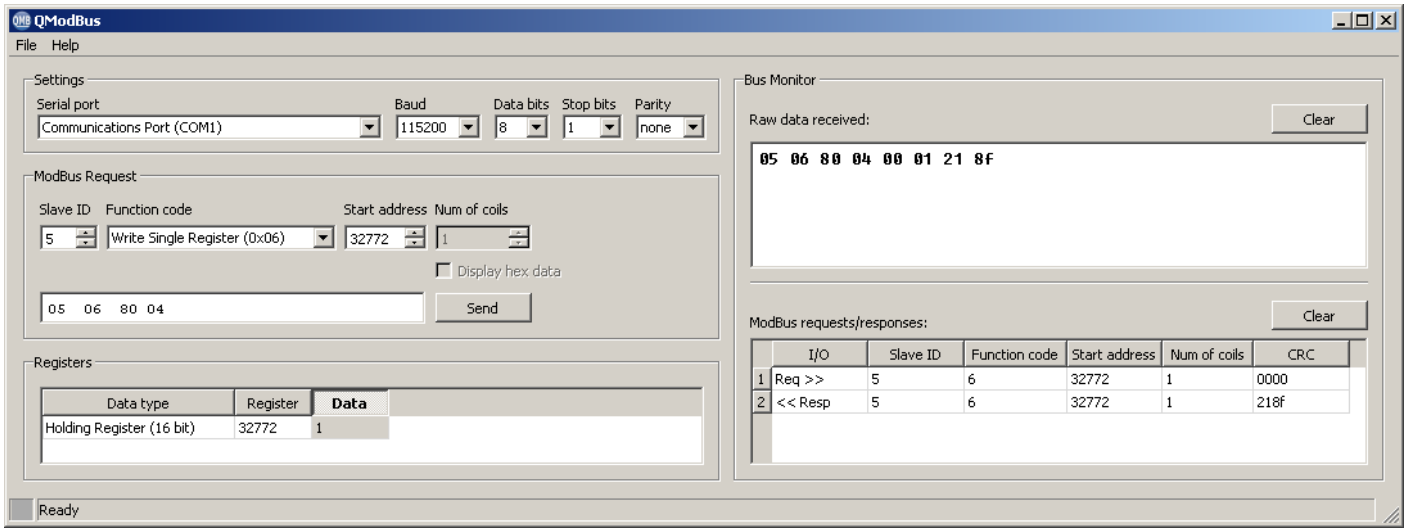
Call GOSUB(1) (Success)

> **NOTE:** If the program label doesn't exist (it must be loaded as a user program in the motor), then the SmartMotor will return exception code 0x86 instead of the function code 0x06.

SEND to motor: 05 06 80 04 00 01 21 8F

RECV from motor: 05 06 80 04 00 01 21 8F

	Slave ID	Funct	Start Addr	No. of Reg.	Byte Cnt	Data low word	Data high word	CRC
SEND	05	06	80 04			00 01		21 8F
RECV	05	06	80 04			00 01		21 8F
A table is provided to illustrate the parts of the byte sequence only. The byte sequence must be transmitted as a stream of bytes shown in the SEND/RECEIVE strings above the table (i.e., no pause or null for the blank cells).								



QModBus Utility Showing SEND / RECEIVE Data

Troubleshooting

The following table provides troubleshooting information for solving common problems. For additional support resources, see the Moog Animatics Support page at:

http://www.animatics.com/support.html

Issue	Cause	Solution
Communication and Control Issues		
Motor control power light does not illuminate.	Control power is off, disconnected or incorrectly wired.	Check that control power is connected to the proper pins and turned on. For connection details, see Connecting the System (RTU) on page 13.
	Motor has routed drive power through drive-enable pins.	Ensure cabling is correct and drive power is not being delivered through the 15-pin connector.
	Motor is equipped with the DE option.	To energize control power, apply 24-48 VDC to pin 15 and ground to pin 14.
Motor does not communicate with SMI.	Transmit, receive or ground pins are not connected correctly.	Ensure that transmit, receive and ground are all connected properly to the host PC.
	Motor program is stuck in a continuous loop or is disabling communications.	To prevent the program from running on power up, use the Communications Lockup Wizard located on the SMI software Communications menu.
Motor does not communicate with Modbus RTU.	No OCHN command in program.	Verify that the OCHN command is used in program to set communication parameters. Modbus RTU does not have default settings.
	Incorrect baud rate.	Check the settings used for the OCHN command.
	Incorrect Modbus RTU address.	Use SADDR or ADDR= command in program to set the correct address at startup.
Motor stops communicating after power reset, requires re-detection.	Motor does not have its address set in the user program. NOTE: Serial addresses are lost when motor power is off or reset.	Use the SADDR or ADDR= command within the program to set the motor address.

Issue	Cause	Solution
Motor disconnects from SMI sporadically.	COM port buffer settings are too high.	Adjust the COM port buffer settings to their lowest values.
	Poor connection on serial cable.	Check the serial cable connections and/or replace it.
	Power supply unit (PSU) brownout.	PSU may be too high-precision and/or undersized for the application, which causes it to brown-out during motion. Make moves less aggressive, increase PSU size or change to a linear unregulated power supply.
Red PWR SERVO light illuminated.	Critical fault.	To discover the source of the fault, use the Motor View tool located on the SMI software Tools menu.
Common Faults		
Bus voltage fault.	Bus voltage is either too high or too low for operation.	Check servo bus voltage. If motor uses the DE power option, ensure that both drive and control power are connected.
Overcurrent occurred.	Motor intermittently drew more than its rated level of current. Does not cease motion.	Consider making motion less abrupt with softer tuning parameters or acceleration profiles.
Excessive temperature fault.	Motor has exceeded temperature limit of 85°C. Motor will remain unresponsive until it cools down below 80°C.	Motor may be undersized or ambient temperature is too high. Consider adding heat sinks or forced air cooling to the system.
Excessive position error.	The motor's commanded position and actual position differ by more than the user-supplied error limit.	Increase error limit, decrease load or make movement less aggressive.
Historical positive/negative hardware limit faults.	A limit switch was tripped in the past.	Clear errors with the ZS command.
	Motor does not have limit switches attached.	Configure the motor to be used without limit switches by setting their inputs as general use.
Programming and SMI Issues		
Several commands not recognized during compiling.	Compiler default firmware version set incorrectly.	Use the Compiler default firmware version option in the SMI software Compile menu to select a default firmware version closest to the motor's firmware version. In the SMI software, view the motor's firmware version by right-clicking the motor and selecting Properties.

PN: SC80100014-001
Rev. C