SureFire

Burner Management Systems

MODBUS OPERATIONS MANUAL

Programming Information for Surefire BMS300 with Modbus Interface (Firmware Rev 1.7 and 1.8)

6 May 2013

This document describes the programming interface for the BMS-300 burner controller using the Modbus interface board. The Modbus registers, their contents, command sequencing and examples of command execution over Modbus are described.

This document applies to firmware revisions 1.7 and 1.8. Version 1.8 has some extended registers (40020 thru 40035) that are described later in the document. In order to use Modbus 1.7 or 1.8, the BMS-300 must be upgraded to firmware package 1.7.

1.0 Introduction

The Modbus interface to the BMS-300 is accomplished via an intermediary processor board, the BMS Modbus board. **The function of this board is to serve as a Modbus RTU slave**, handling requests from the Modbus master to read information and relay command data to the BMS-300 board.

The Modbus board behaves as a specialized “mailbox”; a set of Modbus holding registers is available in the Modbus board, any of which can be read or written by either the BMS board or the Modbus master. This arrangement relieves the BMS board of the job of hosting the Modbus and relaxes many of the timing constraints that would overtax the limited hardware resources on the BMS board MCU.

A number of the registers are constantly updated by the BMS board with information such as temperatures, modes, output states, ignition attempts, and other important data. These should be treated as read-only by the Modbus master. Other registers are defined as command and parameter registers to be written by the Modbus master to cause the BMS board to perform an action or set an operational variable (such as high temperature limit).

Both RS-485 and RS-232 interfaces are available for use by the Modbus master. Only one of these interfaces can be selected for use at any given time via the configuration DIP switch. The configuration DIP switch also sets the Modbus address and the baud rate.

LEDs are present on the Modbus board to indicate processing of Modbus packets (from the Modbus side) and BMS300 packets (from the BMS300 side). The **LEDs illuminate when intact packets are received and are being processed**. Under normal operation the LED on the BMS side should show regular activity as it updates the holding registers on the Modbus board and queries for command data. The LED on the Modbus side will only show activity if the Modbus master is reading from or writing to the Modbus board.

2.0 Amber LED Indication

There are two LEDs in the Modbus circuit board that indicates different operations

COM LED:

This LED indicates that the Modbus circuit board is communicating and sending data packages to the

BMS-300 circuit board. This LED is located between the MCU chip and the large terminal block and is labeled COM.

BMS COM LED:

This LED indicates that the BMS-300 is communicating and sending data packages to the Modbus circuit board. This LED is located between the MCU chip and the small terminal block and is labeled BMS COM.

3.0 Basic read/write operation

Basic operation of the BMS-300 with Modbus is as follows:

For reading a register (or registers) the Modbus master sends a holding register read request to the BMS Modbus board using **Modbus Function 03** (see "MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b", www.modbus.org for more detailed information about the Modbus protocol and functions). The Modbus board will respond with the contents of the requested registers. There are currently 256 registers defined (Modbus addresses 40001 thru 40256), but not all are used. Attempts to read registers outside that address space will return an error according to the Modbus protocol.

While the Modbus board is servicing a read request from the master, it is unable to service simultaneous read requests from the BMS board for command data. This may result in the BMS board waiting for access and may result in a blinking LED display on the BMS board. For this reason it is best for the master to refrain from reading a large number of registers in a single request, and also to avoid issuing rapid read requests. It is suggested that read requests be limited to about 10 registers or less at a time, with a maximum read rate of a few Hz.

For writing a register, the Modbus master sends a holding register write request to the BMS Modbus board using **Modbus Function 6** (write single register) or **Modbus Function 16** (write multiple registers). The supplied data will be written to the specified register.

The following table is the register map for the BMS-300. Bits are in MSB first order.

Modbus address	**Register name Modbus master does R (Read) or RW (Read or Write)**	**Data type**	**Notes**
40001	System Mode(R)	unsigned int 16	See mode table
40002	Average temperature(R)	unsigned int 16	Bit 9..0 is 0-511 degrees
40003	High temperature limit(R)	unsigned int 16	Bit 9..0 is 0-511 degrees
40004	Low temperature limit(R)	unsigned int 16	Bit 9..0 is 0-511 degrees
40005	Configuration and status(R)	unsigned int 16	Bit 0 – Light status 0 : RED light is on solid 1 : GREEN light is on solid Bit 1 – Temperature units 0 : Degrees C

Modbus address	Register name Modbus master does R (Read) or RW (Read or Write)	Data type	Notes
			1 : Degrees F Bit 2 – Input sense 0 : RTD 1 : ALT (pressure switch) Bit 3 - Igniter volts 0 : 14V 1 : 13V Bit 4 – Pilot mode 0 : Intermittent pilot 1 : Standing pilot Bit 5 – Flame sensed 0 : No flame sensed 1 : Flame sensed Bit 6 – Factory use only Bit 7 – Factory use only Bit 8 – Spare standby #1 0 : Clear 1 : Activated (GREEN light blinking) Bit 9 – 2pare standby #2 0 : Clear 1 : Activated (GREEN light blinking) Bit 10 – Shutdown interlock 0 : Clear 1 : Activated (RED light blinking) Bit 11 – ESD status 0 : Clear 1 : Activated (RED light blinking) Bit 12 – Standby interlock 0 : Clear 1 : Activated (GREEN light blinking) Bit 13 – Factory use only Bit 14 – Factory use only Bit 15 – Factory use only
40006	Average battery voltage(R)	unsigned int 16	Bit 7..0 is 0 to 25.5 volts
40007	Burn time(R)	unsigned int 16	Hours, rolls over at 9999
40008	Number of attempts(R)	unsigned int 16	Rolls over at 9999
40009	Number of successes(R)	unsigned int 16	Rolls over at 9999

Modbus address	Register name Modbus master does R (Read) or RW (Read or Write)	Data type	Notes
40010	Command status(R)	unsigned int 16	0x55 IDLE - Ready for new command 0x01 Command executed OK 0x02 Bad command, nothing done 0x03 BMS read of exec reg from Modbus board timed out 0x04 BMS read of exec packet from Modbus board had bad checksum 0x05 BMS read of cmd reg from Modbus board timed out 0x06 BMS read of cmd packet had bad checksum 0x07 Command number not recognized 0x08 BMS read of param reg from Modbus board timed out 0x09 BMS read of param packet had bad checksum 0x0A Parameter out of limits for command 0x0B BMS 300 Controller is locked 0x0C Unknown status code returned from Modbus board read attempt
40011	Command number(RW)	unsigned int 16	0x01 – Unlock – same as pressing unlock sequence on keypad. Display function and timeout exactly the same as if done from keypad. No parameter required. 0x02 – Set low temperature limit. Behaves as if done from the keypad. Requires parameter in the format of register 40004. 0x03 – Set high temperature limit. Behgaves as if done from the keypad. Requires parameter in the format of register 40003. 0x04 – Turn ON. Behaves as if keypad ON button were pressed. No parameter required. 0x05 – Turn OFF. Behaves as if keypad OFF button were pressed. No parameter required. 0x06 – Set flame strength threshold. Behaves as if done from the keypad. Requires a parameter in the form of an 11-bit unsigned integer.
40012	Command parameter(RW)	unsigned int 16	Parameter for command (if required, ignored if not required). May also be interpreted as “command value”.
40013	Command execute(RW)	unsigned int 16	Set to non-zero to request that the command in register 40011 be executed with the parameter in register 40012(if needed). Will continue to execute until set to zero.

Modbus address	Register name Modbus master does R (Read) or RW (Read or Write)	Data type	Notes
40014 thru 40019	Unallocated	unsigned int 16	Read as zero, can be written but will be ignored
40020 (Ver 1.8)	Bit 0 – Light status 0:RED light is on solid 1:GREEN light is on solid	unsigned int 16	Bit #0 of register 40005 unpacked into a single register for use by controllers with primitive bit manipulation capabilities.
40021 (Ver 1.8)	Bit 1 – Temperature units 0 : Degrees 1 : Degrees F	unsigned int 16	Bit #1 of register 40005 unpacked into a single register for use by controllers with primitive bit manipulation capabilities.
40022 (Ver 1.8)	Bit 2 – Input sense 0 : RTD 1 : ALT (pressure switch)	unsigned int 16	Bit #2 of register 40005 unpacked into a single register for use by controllers with primitive bit manipulation capabilities.
40023 (Ver 1.8)	Bit 3 - Igniter volts 0 : 14V 1 : 13V	unsigned int 16	Bit #3 of register 40005 unpacked into a single register for use by controllers with primitive bit manipulation capabilities.
40024 (Ver 1.8)	Bit 4 – Pilot mode 0 : Intermittent pilot 1 : Standing pilot	unsigned int 16	Bit #4 of register 40005 unpacked into a single register for use by controllers with primitive bit manipulation capabilities.
40025 (Ver 1.8)	Bit 5 – Flame sensed 0 : No flame sensed 1 : Flame sensed	unsigned int 16	Bit #5 of register 40005 unpacked into a single register for use by controllers with primitive bit manipulation capabilities.
40026 (Ver 1.8)	Bit 6 – Factory use only	unsigned int 16	Bit #6 of register 40005 unpacked into a single register for use by controllers with primitive bit manipulation capabilities.
40027 (Ver 1.8)	Bit 7 – Factory use only	unsigned int 16	Bit #7 of register 40005 unpacked into a single register for use by controllers with primitive bit manipulation capabilities.
40028 (Ver 1.8)	Bit 8 – Spare standby #1 0 : Clear 1 : Activated (GREEN light blinking)	unsigned int 16	Bit #8 of register 40005 unpacked into a single register for use by controllers with primitive bit manipulation capabilities.

Modbus address	Register name Modbus master does R (Read) or RW (Read or Write)	Data type	Notes
40029 (Ver 1.8)	Bit 9 – 2pare standby #2 0 : Clear 1 : Activated (GREEN light blinking)	unsigned int 16	Bit #9 of register 40005 unpacked into a single register for use by controllers with primitive bit manipulation capabilities.
40030 (Ver 1.8)	Bit 10 – Shutdown interlock 0 : Clear 1 : Activated (RED light blinking)	unsigned int 16	Bit #10 of register 40005 unpacked into a single register for use by controllers with primitive bit manipulation capabilities.
40031 (Ver 1.8)	Bit 11 – ESD status 0 : Clear 1 : Activated (RED light blinking)	unsigned int 16	Bit #11 of register 40005 unpacked into a single register for use by controllers with primitive bit manipulation capabilities.
40032 (Ver 1.8)	Bit 12 – Standby interlock 0 : Clear 1 : Activated (GREEN light blinking)	unsigned int 16	Bit #12 of register 40005 unpacked into a single register for use by controllers with primitive bit manipulation capabilities.
40033 (Ver 1.8)	Bit 13 – Factory use only	unsigned int 16	Bit #13 of register 40005 unpacked into a single register for use by controllers with primitive bit manipulation capabilities.
40034 (Ver 1.8)	Bit 14 – Factory use only	unsigned int 16	Bit #14 of register 40005 unpacked into a single register for use by controllers with primitive bit manipulation capabilities.
40035 (Ver 1.8)	Bit 15 – Factory use only	unsigned int 16	Bit #15 of register 40005 unpacked into a single register for use by controllers with primitive bit manipulation capabilities.
40036 thru 40249	Unallocated	unsigned int 16	Read as zero, can be written but will be ignored
40250	BMS diagnostics register #1 (R)	unsigned int 16	Factory use only
40251	BMS diagnostics register #1(R)	unsigned int 16	Factory use only
40252	BMS read count (R)	unsigned int16	Number of packets sent by the BMS board to the Modbus board to read the contents of a holding register

Modbus address	Register name Modbus master does R (Read) or RW (Read or Write)	Data type	Notes
40253	Modbus board firmware revision (R)	unsigned int16	High byte: uint8 – major release number Low byte: uint8 – minor release number Ex: Version 1.7 will exhibit 0x17
40254	Modbus read count (R)	unsigned int16	Number of function 03 requests from the Modbus master
40255	BMS write count (R)	unsigned int16	Number of packets sent by the BMS board to the Modbus board to update the contents of a holding register

4.0 Command operation

Command operation on the BMS300 utilizes the “mailbox” concept. The general sequence is as follows:

1. MASTER reads Command Status register (Modbus 40010). If it reads as 0x55 (IDLE) then proceed. Otherwise wait a short time (a few hundred ms would be reasonable) and poll that register again. Continue to do so until the register reads as 0x55 (IDLE).
2. MASTER writes the Command Number register (Modbus 40011) with the desired command number. Note that the first command to be issued is likely 0x01 (UNLOCK) unless the BMS300 has already been unlocked and the unlock has not timed out.
3. If the desired command requires a parameter, MASTER writes the parameter data to the Command Parameter register (Modbus 40012).
4. MASTER writes a non-zero value to the Command Execute register (Modbus 40013).

The next poll of the Command Execute register by the BMS board will tell it that the MASTER wants to execute a command. The BMS board will read the command and parameter, execute the command if possible, and write a status code to the Command Status register.

5. MASTER polls the Command Status register (Modbus 40010). If the command executed without error, status code 0x01 (OK) will be returned. If an error occurred during reading or executing the command, the relevant status code will be returned. If the status code is 0x55 (IDLE), then wait a short time (a few hundred ms would be reasonable) and poll that register again. Continue to do so until the register returns a status code other than 0x55.
6. MASTER writes zero (0) to the Command Execute register (Modbus 40013).

The next poll of the Command Execute register by the BMS board will tell it that the MASTER no longer needs to have the command executed. The BMS board will write 0x55 (IDLE) to the Command Status register.

7. MASTER polls the Command Status register (Modbus 40010). If it reads 0x55 (IDLE) then proceed. Otherwise wait a short time (a few hundred ms would be reasonable) and poll that register again. Continue to do so until the register reads as 0x55 (IDLE).

Note that when the Modbus master has written a non-zero value to the Command Execute register, the BMS300 will continue to execute the command repeatedly at its internal command processing loop rate until the Modbus master writes a zero to the Command Execute register

5.0 Example

In the following example the Modbus master attempts to change the high temperature limit and then turn ON the BMS300. The attempt to change the high temperature limit fails initially because the BMS300 had not been unlocked. After the BMS300 is unlocked, the temperature limit is changed and the unit is turned ON.

Time step	Master's action	BMS300 action	Registers
1	Read Command Status register. Is 0x55 (IDLE). Proceeds to next step.		40010 (status) 0x55 40011 (cmd) 0x00 40012 (param) 0x00 40013 (exec) 0x00
2	Write command to set high temp limit (0x03) to command number reg		40010 (status) 0x55 40011 (cmd) 0x03 40012 (param) 0x00 40013 (exec) 0x00
3	Write parameter (125F) to parameter reg		40010 (status) 0x55 40011 (cmd) 0x03 40012 (param) 0x7D 40013 (exec) 0x00
4	Request command execute by writing 0x01 to command exec reg		40010 (status) 0x55 40011 (cmd) 0x03 40012 (param) 0x7D 40013 (exec) 0x01
4		Sees command exec register with non-zero value. Reads command and parameter. Attempts to execute command. Fails since unit is locked. Writes LOCKED error code to command status reg.	40010 (status) 0x0B 40011 (cmd) 0x03 40012 (param) 0x7D 40013 (exec) 0x01
5	Polls command status register. Sees LOCKED (0x0B) error code. Looks up error code and finds it must unlock the BMS300 before changing the temp limit.		40010 (status) 0x0B 40011 (cmd) 0x03 40012 (param) 0x7D 40013 (exec) 0x01
6	Write 0x00 to command exec reg.		40010 (status) 0x0B 40011 (cmd) 0x03 40012 (param) 0x7D 40013 (exec) 0x00
7		Sees command exec register with zero value. Writes IDLE (0x55) to status register. Returns to idle state.	40010 (status) 0x55 40011 (cmd) 0x03 40012 (param) 0x7D 40013 (exec) 0x00

Time step	Master's action	BMS300 action	Registers
8	Reads command status register. Sees IDLE (0x55). Proceeds to next step.		40010 (status) 0x55 40011 (cmd) 0x03 40012 (param) 0x7D 40013 (exec) 0x00
9	Write command to unlock unit.		40010 (status) 0x55 40011 (cmd) 0x01 40012 (param) 0x7D 40013 (exec) 0x00
10	Request command execute by writing 0x01 to command exec reg		40010 (status) 0x55 40011 (cmd) 0x01 40012 (param) 0x7D 40013 (exec) 0x01
11		Sees command exec register with non-zero value. Reads command. Attempts to execute command. Succeeds in unlocking BMS300 Writes OK code to command status reg.	40010 (status) 0x01 40011 (cmd) 0x01 40012 (param) 0x7D 40013 (exec) 0x01
12	Polls command status register. Sees OK (0x01). Proceeds to next step.		40010 (status) 0x01 40011 (cmd) 0x01 40012 (param) 0x7D 40013 (exec) 0x01
13	Write 0x00 to command exec reg.		40010 (status) 0x01 40011 (cmd) 0x01 40012 (param) 0x7D 40013 (exec) 0x00
14		Sees command exec register with zero value. Returns to idle state.	40010 (status) 0x55 40011 (cmd) 0x01 40012 (param) 0x7D 40013 (exec) 0x00
15	Polls command status register. Sees IDLE (0x55). Proceeds to next step.		40010 (status) 0x55 40011 (cmd) 0x01 40012 (param) 0x7D 40013 (exec) 0x00
16	Write command to set high temp limit (0x03) to command number reg. Desired parameter is still in parameter register, so won't write it again.		40010 (status) 0x55 40011 (cmd) 0x03 40012 (param) 0x7D 40013 (exec) 0x00
17	Request command execute by writing 0x01 to command exec reg		40010 (status) 0x55 40011 (cmd) 0x03 40012 (param) 0x7D 40013 (exec) 0x01

Time step	Master's action	BMS300 action	Registers
18		Sees command exec register with non-zero value. Reads command and parameter. Attempts to execute command. Succeeds. Writes OK status code to command status reg.	40010 (status) 0x01 40011 (cmd) 0x03 40012 (param) 0x7D 40013 (exec) 0x01
19	Polls command status register. Sees OK (0x01). Proceeds to next step.		40010 (status) 0x01 40011 (cmd) 0x03 40012 (param) 0x7D 40013 (exec) 0x01
20	Write 0x00 to command exec reg.		40010 (status) 0x01 40011 (cmd) 0x03 40012 (param) 0x7D 40013 (exec) 0x00
21		Sees command exec register with zero value. Returns to idle state.	40010 (status) 0x55 40011 (cmd) 0x03 40012 (param) 0x7D 40013 (exec) 0x00
22	Polls command status register. Sees IDLE (0x55). Proceeds to next step.		40010 (status) 0x55 40011 (cmd) 0x03 40012 (param) 0x7D 40013 (exec) 0x00
23	Write command to turn ON (0x04) to command number reg. No parameter needed. Existing parameter value will be ignored.		40010 (status) 0x55 40011 (cmd) 0x04 40012 (param) 0x7D 40013 (exec) 0x00
24	Request command execute by writing 0x01 to command exec reg		40010 (status) 0x55 40011 (cmd) 0x04 40012 (param) 0x7D 40013 (exec) 0x01
25		Sees command exec register with non-zero value. Reads command and parameter. Attempts to execute command. Succeeds. Begins turn-on sequence as if it were a keypad press. Writes OK status code to command status reg.	40010 (status) 0x01 40011 (cmd) 0x04 40012 (param) 0x7D 40013 (exec) 0x01
26	Polls command status register. Sees OK (0x01). Proceeds to next step.		40010 (status) 0x01 40011 (cmd) 0x04 40012 (param) 0x7D 40013 (exec) 0x01
27	Write 0x00 to command exec reg.		40010 (status) 0x01 40011 (cmd) 0x04 40012 (param) 0x7D 40013 (exec) 0x00

Time step	Master's action	BMS300 action	Registers
28		Sees command exec register with zero value. Returns to idle state.	40010 (status) 0x55 40011 (cmd) 0x04 40012 (param) 0x7D 40013 (exec) 0x00
29	Polls command status register. Sees IDLE (0x55). All done!		40010 (status) 0x55 40011 (cmd) 0x04 40012 (param) 0x7D 40013 (exec) 0x00

6.0 DIP switch configuration and settings

Switch SW1 on the Modbus board configures the Modbus address, baud rate, and selects the serial interface to be used. The notation “ON” and “OFF” follows from the direction of the arrow and the word “ON” located on the left side of the switch body near SW1-1. The switches are as follows:

Switch	Selects	Description
SW1-1	Modbus address bit 0	“ON” position is “1”, “OFF” is 0
SW1-2	Modbus address bit 1	“ON” position is “1”, “OFF” is 0
SW1-3	Modbus address bit 2	“ON” position is “1”, “OFF” is 0
SW1-4	Modbus address bit 3	“ON” position is “1”, “OFF” is 0
SW1-5	Modbus address bit 4	“ON” position is “1”, “OFF” is 0
SW1-6	Baud rate for communication with Modbus master	“ON” is 19200, “OFF” is 9600
SW1-7	Unused	Unused
SW1-8	Interface for communication with Modbus master	“ON” is RS232, “OFF” is RS485

The following table shows the switch settings for SW1-1 through SW1-5 required to obtain the desired Modbus address:

Desired Modbus address	SW1-1	SW1-2	SW1-3	SW1-4	SW1-5
1	**ON**	Off	Off	Off	Off
2	Off	**ON**	Off	Off	Off
3	**ON**	**ON**	Off	Off	Off
4	Off	Off	**ON**	Off	Off
5	**ON**	Off	**ON**	Off	Off
6	Off	**ON**	**ON**	Off	Off
7	**ON**	**ON**	**ON**	Off	Off
8	Off	Off	Off	**ON**	Off
9	**ON**	Off	Off	**ON**	Off

Desired Modbus address	SW1-1	SW1-2	SW1-3	SW1-4	SW1-5
10	Off	**ON**	Off	**ON**	Off
11	**ON**	**ON**	Off	**ON**	Off
12	Off	Off	**ON**	**ON**	Off
13	**ON**	Off	**ON**	**ON**	Off
14	Off	**ON**	**ON**	**ON**	Off
15	**ON**	**ON**	**ON**	**ON**	Off
16	Off	Off	Off	Off	**ON**
17	**ON**	Off	Off	Off	**ON**
18	Off	**ON**	Off	Off	**ON**
19	**ON**	**ON**	Off	Off	**ON**
20	Off	Off	**ON**	Off	**ON**
21	**ON**	Off	**ON**	Off	**ON**
22	Off	**ON**	**ON**	Off	**ON**
23	**ON**	**ON**	**ON**	Off	**ON**
24	Off	Off	Off	**ON**	**ON**
25	**ON**	Off	Off	**ON**	**ON**
26	Off	**ON**	Off	**ON**	**ON**
27	**ON**	**ON**	Off	**ON**	**ON**
28	Off	Off	**ON**	**ON**	**ON**
29	**ON**	Off	**ON**	**ON**	**ON**
30	Off	**ON**	**ON**	**ON**	**ON**
31	**ON**	**ON**	**ON**	**ON**	**ON**

7.0 Troubleshooting information

SYMPTOM: The LEDs on the BMS Modbus board do not flash.

DISCUSSION: The LEDs only flash when intact (complete and correct) packets are received from their respective interfaces. If the data arrives garbled, or if no data is sent, then the corresponding LED will not flash. The COM LED is between the MCU and the large terminal blocks, and is associated with activity on the Modbus. The BMS COM LED is between the MCU and the small terminal blocks, and is associated with activity on the BMS board.

POSSIBLE SOLUTIONS:

1. Verify that the cables or wires to the BMS Modbus board are connected.
2. Verify that the wires are connected to the correct pins.
3. Verify that the baud rate for data transmission is in agreement on both sides (for example,

if the BMS Modbus board is set for 9600 baud, then the Modbus master must also be set for 9600 baud).

4. Verify that power is present at the BMS Modbus board.

5. If the Modbus COM LED doesn't flash, verify that the Modbus master is sending read or write requests.
6. If the BMS COM LED doesn't flash, verify that the BMS board has power. Under normal operation this LED should always be flashing if both boards are powered.

SYMPTOM: BMS-300 display flickers during Modbus operation

DISCUSSION: Due to the hardware constraints on the BMS-300, the LED display will flicker slightly during normal operation when the Modbus is in use. If the Modbus master polls the BMS300 for data at a rapid rate, or transfers many registers during each poll, the flickering becomes worse. If the Modbus is being queried on a continuous basis, the display and keypad may become difficult to operate.

POSSIBLE SOLUTIONS:

1. Limit the number of registers being read from the BMS-300 during each transfer to about 10 or less.
2. Limit the polling interval to about twice per second.

SYMPTOM: Command status always reads 0x0B when attempting to execute a command

DISCUSSION: The BMS-300 board must be "unlocked" before any commands that modify the operational variables will succeed. This is similar to the front panel operation which requires a special sequence to unlock access to operational variables. A special command is used to allow the Modbus master to remotely unlock the BMS-300, command 0x01. The unlock will eventually timeout in the same manner as if done from the keypad.

POSSIBLE SOLUTIONS:

1. Execute an UNLOCK command (0x01) before attempting to execute any command that changes an operational variable (e.g. sets a low temperature limit). See the example in Section 4.

MODBUS OPERATIONS MANUAL

5/6/2013
Last Update: 1/5/2016

SureFire Farmington, NM Office:
2405 W. Aztec Blvd. Aztec, NM 87410
P: 505-333-2878
F: 505-333-2879

SureFire Houston, TX Office:
12510 Cutten Rd. Houston, TX 77066
P: 281-377-9756
F: 281-379-1081

Tech Support:
505- 333-2876

For SureFire Product Updates Please Visit:
www.surefire-controls.com