



PF2200-SB

MODBUS CONFIGURATION GUIDE

PROFIRE



STATUS		READY	
BATH (°F)	186.5 [197]	STACK (°F)	557.8 [580]
OUTLET (°F)	107.5 [128]	LEVEL (L)	246.7 [208]
		PRESSURE (psi)	45.6 [70]

●
STATUS

START

STOP



OK



PF2200



TABLE OF CONTENTS

INTRODUCTION.....	4
CONFIGURING MODBUS.....	4
MODBUS COMMANDS.....	6
REGISTER DATA FORMAT.....	7
REGISTER ADDRESS vs OFFSET.....	7
LATCHED vs UNLATCHED REGISTERS.....	8
SYSTEM UNITS.....	8
PF2100 BACKWARD COMPATIBILITY.....	8
COMMUNICATION LOSS.....	8
MODBUS REGISTER MAP.....	9
TROUBLESHOOTING.....	9

INTRODUCTION

This document outlines the Modbus Register Map available, as well as configuration details on the PF2200-SB. The physical communication implementation used is a half-duplex RS-485 with the Modbus RTU as a slave device.

The Modbus protocol is set up on the PF2200 user interface. To enable the Modbus protocol, you will need to enter Commissioner Mode by pressing the Commissioner key and answering "Yes" to the confirmation Prompt. Once in Commissioner Mode, navigate to the Settings screen and highlight Modbus, press the "OK" key to enter the Modbus configuration screen.

For a complete listing of all Modbus Registers, see Common Registers, Coil & Discrete Input Registers, or Read Only Input Holding Registers.

CONFIGURING MODBUS

The table below is a description of each of the configuration settings found under the Modbus settings screen on the PF2200.

Setting	Description
Modbus RTU Communication	Enables or disables the Modbus port on the user interface. This must be enabled if Modbus will be used.
Modbus Termination	When enabled a 100R resistor will be automatically terminated across the A and B signal lines. When disabled, no termination resistor is used. Often the termination is enabled if this device is the last drop on the Modbus line.
Slave Address	The slave address of the BMS device may be selected by the Commissioner. If other devices are on the same Modbus line their addresses must be different.
Baud Rate	The baud rate of the communication protocol may be selected as 9600, or 19200. It is recommended to use the slowest baud rate available for noisy or long run lengths.
Stop Bits	One or two stop bits may be selected. Default configuration is one stop bit.
Parity	Parity may be selected as None, Odd, or Even. The default configuration is no parity.

The following steps are the recommended configuration for the Modbus master to the PF2200 system.

1. Configure the PF2200 Modbus settings, located under commissioner mode in the settings screen of the PF2200.
 - a. Enable Modbus RTU Communication.
 - b. If the PF2200 is the last drop on the Modbus line, or it is the only device, set the Modbus termination setting to enabled. Some Modbus Master drivers may have a weak drive strength and will work better with not termination. Both of the cases can be tested to ensure the optimum result.
 - c. Set the slave address to a unique value and unused for any other devices on the Modbus Line. Match the slave address between the Modbus master and the slave device.
 - d. The baud rate of the PF2200 must match the Modbus Master. The recommended baud rate is 9600bps.
 - e. Set the Stop bits to 1 and the parity to none. These are the standard configurations for the Modbus protocol.
2. Configure the Modbus Master.
 - a. Minimum delay between Modbus poll packets should be longer than 20 milliseconds. The recommended inter-packet delay is 20 milliseconds.
 - b. The recommended response timeout is 1 second and should not exceed 500 milliseconds.
 - c. Writing setting values via Modbus will overwrite any local user settings on the PF2200. It is recommended to keep setting writes to a minimum. If continually writing settings is necessary, keep the write rate to a minimum of 5 seconds or longer.
 - d. When reading settings, it is recommended to have a scan rate of greater than 1 second.
 - e. Confirm the Modbus Master is in RTU mode and not in ASCII mode.
3. Determine which settings to write.
 - a. Settings may be written one by one, or multiple settings may be written all at once if they are sequential in the register table (shown below).
 - b. Some settings such as floats and uint32_t span multiple registers, and therefore must be written at the same time, or the write request will fail.
4. Determine which registers to read.
 - a. Registers may be read one at a time, or multiple registers can be requested in one packet.
 - b. If an individual register is requested, that does not exist, and an exception code will be returned.
 - c. When it is desired to have multiple register requests, as long as the first register has a valid address, the following registers, regardless of validity, will return successfully. This avoids the breaking up of the register table reads. Any registers with an invalid address will return as 0.
5. Configuring the Modbus register units.

- a. Many of the registers use units that match the user interface units. For example, unless otherwise noted all temperatures will either be reported in Celsius or Fahrenheit depending on the temperature unit selection when commissioning the PF2200. Units for Registers read or written should be confirmed on the user interface and converted on the Modbus master side if required.
- b. Many of the registers use a factor of x10 to increase resolution of data. For example, a register with a value of 12.1 may be multiplied by a factor of 10 to return a value of 121; this allows one decimal place to be retained over the Modbus read/write. Check each register description to see if it is multiplied by this factor. The Modbus master will be required to convert the value back to a decimal format by dividing it by 10.

For Trouble shooting the Modbus configuration, see the Troubleshooting section below.

MODBUS COMMANDS

Only the following Modbus RTU commands are supported. All other Modbus RTU commands are not supported and will return an exception code for the invalid command. Modbus TCP is not directly supported but can be used if a third-party bridge is purchased or if the Modbus TCP/IP expansion card is installed.

Name	Command	Description
Read Input Registers	4 = 0x04	Two bytes per register are returned.
Read Coil	1 = 0x01	Bits pack the response.
Read Holding Registers	3 = 0x03	Two bytes per register are returned.
Read Discrete Input	2 = 0x02	Bits pack the response.
Write Multiple Holding Registers	16 = 0x10	Two bytes per register must be sent.
Write Single Holding Register	6 = 0x06	Two bytes per register must be sent.
Write Multiple Coils	15 = 0x0F	This function code is NOT SUPPORTED.
Write Single Coil	5 = 0x05	This function code is NOT SUPPORTED.

REGISTER DATA FORMAT

The PF2200 Modbus register map supports multiple data formats in addition to the standard Modbus definitions. These include floats, uint32_t, and arrays. These types require multiple (16bit) registers for representation and described as follows:

General:

- Big-Endian format where the most significant byte and the most significant word are sent first (in the lower register).
- All registers must be read/written in one request. Hence a multi-read or multi-write command must be used with a minimum length of the size of the data type.

Long words and arrays (uint32_t, uint64_t, arrays):

- uint32_t are held in two sequential registers: (ABCD) = Reg 1: AB, Reg 2: CD
- uint64_t are held in four sequential registers: (ABCDEFGH) = Reg 1: AB, Reg 2: CD, Reg 3: EF, Reg 4: GH
- Arrays are held in sequential registers with 2 bytes per register. The number of registers to read/write will be the length of the array divided by 2. If the size of the array is odd the last byte is extended to a full word.

Floating Point numbers (ABCD):

- Held in two sequential registers: Reg 1 - AB, Reg 2 CD
- Floating point numbers are represented in IEEE-754 Standard format.

REGISTER ADDRESS vs OFFSET

Some Modbus configuration software requires the 5-digit Register Address to be entered while other software uses the 1 to 4-digit Register Offset. This guide provides only the offset number. Consult your software documentation to determine which is required in your case.

To understand why this is, it is useful to know a bit about how Modbus RTU packets are structured. Modbus RTU packets encode the Register Address using a Command and a Register Offset. Each Command represents a Base Address, which is the same as the first address listed in the table above next to each command. The Register Offset is calculated by subtracting the Base Address from the Register Address as follows:

Register Offset = Register Address - Base Address

For example, the Base Address for Command 0x03 is 40001. If you want to read register 40003, then the Register Offset encoded in the Modbus RTU packet would be $40003 - 40001 = 2$.

LATCHED vs UNLATCHED REGISTERS

Latched registers have the same function as their corresponding unlatched registers, but once set will remain set until the system is stopped and then restarted. All registers are unlatched unless explicitly listed as latched.

SYSTEM UNITS

Settings and status registers may be displayed in the same unit as the user interface is configured for. If a register follows a UI unit, it will be mentioned in the register description. Commonly the temperatures, pressures, levels and auxiliary inputs will use the UI display unit. Registers that show the span of an input (such as pressure, level, or auxiliary span min and max) when set to a unit of % or ma will display as 0 from Modbus. The reason for this is the span cannot be mapped back into its own base unit (of ma or %). In these cases, the span will always be 4 - 20mA as 0 - 100% of the span of the input.

PF2100 BACKWARD COMPATIBILITY

The PF2200 Modbus register map has been substantially expanded over the PF2100 to include registers for all settings and system status information. Register mapping from the PF2100 has been included as a subset to maintain backward compatibility for Profire products. These registers are labeled with the clause: *PF2100 compatible register in their descriptions. This allows for PF2200 units to be a drop-in replacement for PF2100 units without requiring an update to the Modbus Master on most sites. Some register formats from the PF2100 are not supported identically in this map as hardware IO may be different between platforms. These are indicated as a PF2100 Unsupported register. When starting a new project, it is recommended to use only PF2200 registers as these will be fully supported going forward.

COMMUNICATION LOSS

The PF2200 user interface communicates to the BMS via a proprietary communication protocol called PFN. With the Slave Modbus port being accessible on the user interface, data must be transferred from the BMS to the user interface over the PFN link. If the user interface loses communication to the BMS it can no longer retrieve Modbus setting and status information. In this case the Modbus registers will return all zeros except for the Modbus communication loss register (which will indicate a 1) and the communication loss counter (which will increment every second that comms are lost).

MODBUS REGISTER MAP

For a complete list of all the register maps see the Modbus Register Map found [here](#).

TROUBLESHOOTING

The following section outlines some common issues with Modbus configuration and installation. Check the Modbus Diagnostics page to provide detailed information on attempted transactions.

If the device is not responding:

Problem	Diagnosis
Device is not responding	<ol style="list-style-type: none">1. Check that configuration parameters match between the Master and Slave device such as baud rate, stop bits and parity.2. Check that the slave address matches.3. Check the Modbus response timeout is greater than 1000ms.4. Check that the RS-485 Lines are not connected backwards. A wire connects to A or D-. B wire connects to B or D+.5. Confirm a signal ground wire is connected between the Modbus Master and Slave device.6. Confirm the Slave device is enabled in settings.7. Try communication with termination enabled or disabled. Sometimes termination enabled can cause the Modbus Master to be incorrectly biased.8. Confirm if the Modbus master has internal pullup and pull-down termination on the data lines. Some Master devices require biasing resistors to be installed externally.
If there are CRC errors	<ol style="list-style-type: none">1. Check that configuration parameters match between the Master and Slave device such as baud rate, stop bits and parity.2. Check if there is noise on the line. This can be caused by external equipment or long run lengths.3. Check if the data lines are reversed.
If data returned is always zero	<ol style="list-style-type: none">1. Check that the modbus port is enabled on the user interface.2. Check that the user interface is communicating with the BMS. The modbus communication register will return a value of 1 if comms have been lost to the BMS.3. Check that the correct register is addressed. If the register is invalid it will return an exception code or a zero.
My BMS shuts down when	<ol style="list-style-type: none">1. The set-points may not be formatted correctly. Check the register definition.

Problem	Diagnosis
writing set-points to it	<ol style="list-style-type: none"> <li data-bbox="505 289 1357 352">2. The units may not be configured as expected. Check the temperature units in the UI units, configuration. <li data-bbox="505 359 1357 491">3. The set-points may be written out of range. Check the PF2200 user manual for set-point configuration errors. For example, if the process set-point is written to be higher than the high temperature set-point the system will shutdown.
My BMS won't restart when i request a start	<ol style="list-style-type: none"> <li data-bbox="505 552 1195 583">1. Confirm the correct value is written to the start register. <li data-bbox="505 590 1333 653">2. Confirm the lockout code is cleared prior to attempting to start the system.
The values read such as pressure, level and temperature don't make sense	<ol style="list-style-type: none"> <li data-bbox="505 709 1344 814">1. Check the pressure, level and temperature units in the UI units configuration. The modbus representation matches the UI Units. For example, pressure can be in PSI or kPa, etc. <li data-bbox="505 821 1360 919">2. Check if the register is using a multiplier such as x10. If this is the case then the Modbus Master will need to divide by 10 to get the actual value.
I don't know if my Modbus read or write is working	<ol style="list-style-type: none"> <li data-bbox="505 987 1369 1157">1. Read the available Test Registers and confirm the results match as described. These registers always return a specific value when read so that addressing and formatting can be confirmed. These registers may also be written. If the write is incorrect they will return an exception code. <li data-bbox="505 1163 1357 1333">2. Commonly the incorrect register address is used. Each Modbus Master is different, it may require the address offset (index starting at 0) or the address itself (index starting at 1). Attempt to read the test registers at both the address and the offset to see which result provides the correct response.



UNITED STATES

1.801.796.5127
321 South, 1250 West Suite 1
Lindon, UT 84042, USA
support@profireenergy.com

CANADA

1.780.960.5278
9671 - 283 Street
Acheson, Alberta T7X 6J5, Canada
support@profireenergy.com