Prairie Wind Systems, LLC

# PWS-400
# MODBUS Data Module


# User's Manual

14-10400
10/20/2013

The information contained in this document is subject to change without notice. Prairie Wind Systems has made a reasonable effort to ensure that the information contained in this document is accurate as of the date of publication.

Prairie Wind Systems makes no warranty of any kind with regards to this material, including, but not limited to, its fitness for a particular application. Prairie Wind Systems will not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material. In no event shall Prairie Wind Systems be liable for any claim for direct, incidental, or consequential damages arising out of, or in connection with, the sale, manufacture, delivery, or use of any product.

# 1 Regulatory Information

The PWS-400 has been tested and approved to be compliant to the following regulatory standards.

- EN61326-1: 2006, for immunity in industrial locations (CE)
- EN55011: 2009, Class A, Group 1, for emissions (CE)
- CISPR 11, Ed. 5.0, 2009-05, Class A, Group 1, for emissions
- CFR Title 47: FCC Part 15, Class A, for emissions

Any changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

## 1.1 FCC Statement

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with this manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

## 1.2 Battery Statement

This device contains a poly-carbonmonoflouride lithium coin cell battery to preserve the real-time clock when power is not applied. The Department of Transportation requires that the outside of each package that contains primary lithium batteries, regardless of the size or number of batteries, be labeled with the following statement: "PRIMARY LITHIUM BATTERIES – FORBIDDEN FOR TRANSPORT ABOARD PASSENGER AIRCRAFT". The labeling requirement covers shipping via highway, rail, vessel or cargo-only aircraft and covers all shipments into or out of the United States. The label must be in contrasting color and the letters must be 12 mm (0.5 in) in height for packages weighing more than 30 Kg and 6 mm (0.25 in) in height for packages weighing less than 30 Kg.

The lithium battery does not contain enough lithium to qualify as a reactive hazardous waste. The battery is safe for disposal in the normal municipal waste stream.

## 2 Warranty and Assistance

The PWS-400 is warranted by Prairie Wind Systems, LLC to be free from defects in materials and workmanship under normal use and service for twelve (12) months from the date of shipment unless specified otherwise. Prairie Wind Systems' obligation under this warranty is limited to repairing or replacing, at Prairie Wind Systems' option, defective products. The customer shall assume all costs of removing, reinstalling, and shipping defective products to Prairie Wind Systems. Prairie Wind Systems will return such products by surface carrier prepaid. This warranty shall not apply to any product which has been subjected to modification, misuse, neglect, accidents of nature, or shipping damage. This warranty is in lieu of all other warranties, expressed or implied, including warranties of merchantability or fitness for a particular purpose. Prairie Wind Systems is not liable for special, indirect, incidental, or consequential damages.

Products may not be returned without prior authorization. To obtain a Returned Materials Authorization (RMA) number, contact Prairie Winds Systems at the phone number below. Please write the RMA number clearly on the outside of the shipping container. Prairie Wind Systems' shipping address is:

Prairie Wind Systems, LLC
RMA #_____
7784 Big Sky Court
Windsor, CO 80550

support@prairiewindsystems.com
Phone: 970.460.6066
Fax: 970.692.2434

# 3 Overview

The PWS-400 is a MODBUS slave device that adds data logging capability to the master device of a MODBUS RS485 network. Standard write and read register commands are used to store and recall data in non-volatile memory. Data records are free-format and are automatically time-stamped and check-summed. Data records can be encrypted for maximum data security. When the memory is full, recording automatically wraps around to record the newest data over the oldest data.

In addition to the primary data logging feature, the PWS-400 extends the capabilities of a system with these additional features.

- Data can be recorded and retrieved securely using the Advanced Encryption Standard with a user-defined key.
- The PWS-400 includes an accurate, temperature-compensated, real-time clock. Every data record logged to the device is date and time stamped. The clock is also available to the system master device via simple register reads.
- The device's ambient temperature sensor is also available via register access. In addition to providing temperature compensation for the real-time clock, the ambient temperature is also logged with every data record for enhanced data tracking and system troubleshooting.
- The PWS-400 measures and records its system power input voltage. This value enhances data tracking and system troubleshooting, particularly in battery operated systems. The user can track battery and recharging performance in standalone and solar charged systems.

The Prairie View Software that comes with the device provides a convenient means to configure the device and manage its data log.

# 4 Specifications

| | |
|---|---|
| **Case** | |
| Description | ABS plastic with integral mounting flanges |
| Dimensions | 4.61"L x 2.32"W x 1.30"H (11.7 x 5.9 x 3.3 cm) including flanges and connector |
| Weight | 3.5 oz (100 g) |
| Ratings | IP 66 and NEMA 4X |
| Connector | 9-pin D-sub socket |
| **Environmental** | |
| Operating Temperature | -40 to 70 °C (-40 to 158 °F) |
| Operating Humidity | 5 to 95 %RH non-condensing |
| Storage Temperature | -40 to 80 °C (-40 to 176 °F) |
| **Power Input** | |
| Input Voltage | 8 to 32 Vdc |
| Transient Protection | 250 Watts Peak |
| Sleep Current | 20 µA typical at 13.5 Vdc, network inactive |
| Idle Current | 100 µA typical at 13.5 Vdc, network inactive |
| Active Current | 10 mA typical at 13.5 Vdc |
| **RS485 Transceiver** | |
| Unit Load | 1/8 |
| Termination | None |
| Common Mode Range | +/- 7 Vdc |
| Transient Protection | 250 Watts Peak |
| Protocol | Half-Duplex, MODBUS over Serial Line |
| Transmission Modes | RTU, ASCII |
| Addressing | 1 to 247 plus broadcast |
| Baud Rates | 1200, 2400, 4800, 9600, 19200, 38400, 57600 |
| Data Bits | 7, 8 |
| Parity | Odd, Even, None |
| Stop Bits | 1, 2 |

| | |
|---|---|
| **Data Storage** | |
| Memory Type | 4 MB non-volatile flash |
| Data Record Size | 1 to 64 registers (2 to 128 bytes) |
| Number of Data Records | 131,008 (1 register per record, no encryption)<br>to 25,216 (64 registers per record, with encryption) |
| Recording Rate | 1 record per second with no wraparound<br>1 record per 5 seconds after wraparound |
| Recording Endurance | 50,000 wraparound and erase cycles |
| Data Security | AES-128 Advanced Encryption Standard |
| Data Retention | 10 years |
| **General-Purpose Registers** | |
| Configuration Registers | 32 (64 bytes) non-volatile flash |
| **Real-Time Clock** | |
| Resolution | 0.01 seconds |
| Accuracy | +/- 1 minute per month, temperature compensated |
| Backup Battery Life | 10 years |
| **Temperature Sensor** | |
| Resolution | 0.1 °C |
| Accuracy | +/- 3 °C |
| **Voltage Sensor** | |
| Resolution | 0.01 Vdc |
| Accuracy | +/- 0.05 Vdc |

# 5 Installation

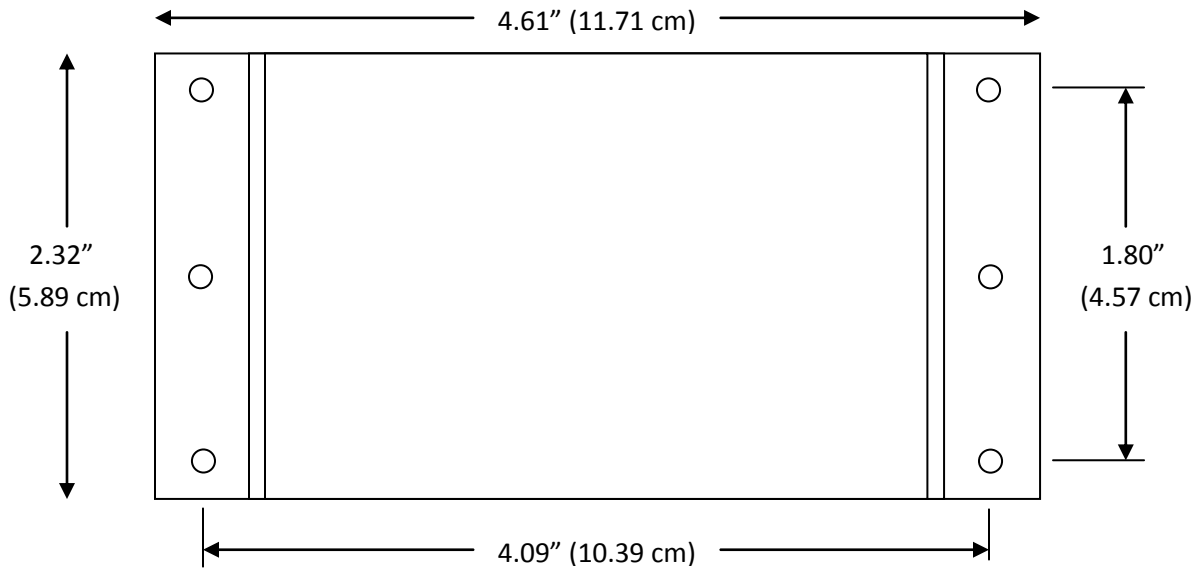
## 5.1 Environmental Considerations

The PWS-400 is specified for operation in a non-condensing humidity environment. When temperature and/or humidity tolerances are exceeded, damage to internal components and/or measurement inaccuracies due to condensation may result.

The device must be housed in an enclosure suited for field use. The environmental ratings of the device's enclosure are intended as a backup to the primary field housing. The field housing should contain desiccant that is replaced or dried frequently enough to control the humidity. Effective control of the humidity is the user's responsibility.

## 5.2 Mounting the Device

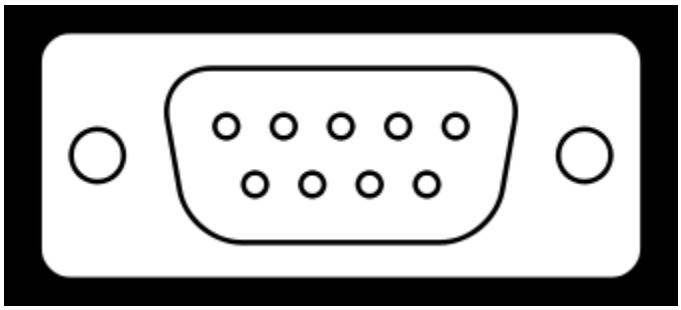
Mounting flanges are integral to the device's enclosure. Six mounting holes sized for #4 screws are provided. Two or more screws should be used to mount the enclosure.



## 5.3 Connector Pin-Out

The PWS-400 uses a standard 9-pin D-subminiature socket. The sockets on the face of the connector are numbered as shown below. This is also the pin-out looking at the wiring side of the mating 9-pin D-sub plug.

**5 4 3 2 1**



**9 8 7 6**

The function of each pin is described below. The pin-out follows the MODBUS specification for RS485 connections using this style of connector.

| Pin | Name | Description |
|---|---|---|
| 1 | GROUND | Signal and power supply common |
| 2 | POWER | Positive power input, 8 to 32 Vdc |
| 3 | MODE | No connection |
| 4 | A | No connection |
| 5 | RS485B(+) | RS485 transceiver positive terminal |
| 6 | D | No connection |
| 7 | C | No connection |
| 8 | B | No connection |
| 9 | RS485A(-) | RS485 transceiver negative terminal |

For reliability of the connection, the mating connector should utilize the mounting standoffs provided. The cable length should be kept as short as practical. The maximum length of cable for an RS485 network is 4000 feet (1200 meters).

### 5.3.1 Power Requirements

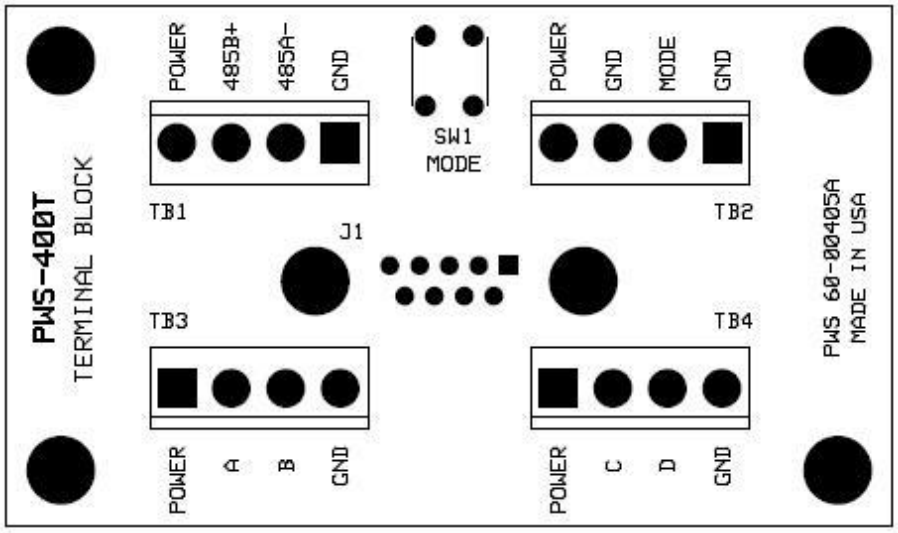
The wide input voltage range of the device permits operation on a 12 volt or 24 volt DC supply. The device will automatically shut down if the input voltage falls below 7 volts and will not resume normal operation until the input voltage rises above 8 volts. Sustained operation above 32 volts will damage the device. The input is reverse polarity and transient protected. Apply the positive side of the supply to pin 2 (POWER), the negative side to pin 1 (GROUND).

### 5.3.2 RS485 Signals

Pin 5 of the connector is the positive transceiver signal (RS485B). Pin 9 is the negative transceiver signal (RS485A). No signal termination resistance is provided internally. In most low power applications, no additional termination resistor is necessary. If required, a resistor may be added externally. The maximum offset of the signals from the GROUND at pin 1 is 7 volts. Both signals are transient protected.

## 5.4 Terminal Block Accessory

The optional PWS-400T Terminal Block accessory board attaches to the 9-pin connector of the PWS-400 and provides terminal block connections for wiring all signals.




*Note: Signals A, B, C, D and MODE are not used on the PWS-400 and must be left unconnected.*

# 6 Getting Started

The PWS-400 communicates as a slave device on an RS485 network using the industry standard MODBUS protocol for serial devices. This section provides guidelines to getting the device up and running on a network. If the device is not already wired, follow the installation instructions provided in the previous section.

Section 7 describes the device register map. The PWS-400 uses the MODBUS Holding Registers data model exclusively. All of the device functionality is accessed using the MODBUS read and write holding register commands.

Section 8 provides the details of the MODBUS protocol for users that are not familiar with MODBUS. Also refer to Section 8 for device-specific implementation information, such as the organization of registers into data types.

## 6.1 Device Configuration

Configuring the device consists of writing user-specific settings to one or more of the device registers. The Prairie View Software supplied with the device provides a convenient, step-by-step means to configure the device and is the recommended place to start. Refer to the software manual for instructions on installing the software on your desktop computer, laptop computer, or mobile device. It is also possible to configure the device by directly writing to the registers using a master device with pass-through capability, or using a third-party software tool.

## 6.2 Network Communication Settings

Before the device is placed on an active RS485 network, it must be configured to match the network's communication settings and be assigned a unique device address. The device is preconfigured with the MODBUS standard settings of 19200 baud, 8 data bits, even parity, 1 stop bit, RTU transmission mode, and device address 1. Refer to the Device Configuration section of the device register map to change the device address. Refer to the RS485 Configuration Registers section of the device register map if any of the communication settings need to be changed.

## 6.3 Site Identification

If the user is collecting data from multiple devices located at different sites, it is recommended that the Site Id register be written with a unique value for each site. The Site Id is recorded with every data record, making it easier to track the data once it has been retrieved from a device.

# 7 Register Map

The PWS-400 uses the MODBUS Holding Registers data model exclusively. All of the device functionality is accessed using the MODBUS read and write holding register commands. The organization of the register map is kept consistent among Prairie Wind System devices as much as practical.

## 7.1 Device Configuration Registers

The device configuration registers identify the device, its site location, and its basic configuration.

| Device Configuration Registers | | | | | |
|---|---|---|---|---|---|
| **Register Number** | **Size (Registers)** | **Data Type** | **Access** | **Name** | **Default Value** |
| 1000 | 1 | USHORT | R | Register Map Version | 2 |
| 1001 | 1 | USHORT | R | Device Id | 400 |
| 1002 | 2 | ULONG | R | Serial Number | |
| 1004 | 1 | USHORT | R | Firmware Version | |
| 1005 | 1 | USHORT | R | Boot Code Version | |
| 1006 | 1 | USHORT | R | Hardware Version | |
| 1007 | 1 | USHORT | R/W | Site Id | 0 |
| 1008 | 16 | STRING | R/W | Site Name | 0 |
| 1024 | 32 | USHORT | R/W | Site Information | 0 |
| 1056 | 1 | USHORT | R/W | Device Address | 1 |
| 1057 | 1 | USHORT | R/W | Low Voltage Warning Threshold | 9600 mV |

### 7.1.1 Register Map Version

This is the version of the register map supported by the device. This provides for the modification of the register map at a future date and detection of the difference by the master device.

### 7.1.2 Device Id

This is the model number of the device. It can be used to validate the system configuration and to identify the available feature set.

### 7.1.3 Serial Number

This is the serial number of the device. The master device can read the serial number of the device to provide system traceability.

### 7.1.4 Firmware Version

This register identifies the firmware version of the device.

### 7.1.5 Boot Code Version

This register identifies the boot code version of the device. The boot code supports field upgrades of the device firmware. The boot code on a given device will not change with firmware upgrades. Its version is made available to support field upgrade utilities.

### 7.1.6 Hardware Version

This register contains the hardware version of the device.

### 7.1.7 Site Id

The Site Id is a general-purpose, non-volatile, read/write register. The contents of this register are recorded with each logged data record, permitting data to be traced to a particular site or system.

### 7.1.8 Site Name

The Site Name identifies the site to a user. The 16-register, non-volatile string holds up to 32 characters.

### 7.1.9 Site Information

Site Information is comprised of 32, general-purpose, non-volatile, read/write registers. Use these registers to hold system configuration and/or calibration data. The format of the data contained in the registers is defined by the user. They can be used as individual registers or combined to form ULONG, FLOAT or STRING data types. These registers should not be used to hold data that changes frequently.

### 7.1.10 Device Address

The valid range for the device address is 1 to 247. The default value is 1. When this register is written, the response will be returned with the previous address. All subsequent commands must be sent with the new address. The register is non-volatile.

### 7.1.11 Low Voltage Warning Threshold

If the input voltage to the device is less than or equal to this threshold, the low voltage warning status bit is set in the device status register. The threshold is set in millivolts and can range from 8000 to 32000 mV (8 to 32 volts). Attempting to write a value outside this range will generate an exception response with the ILLEGAL WRITE VALUE exception code.

## 7.2 Device Command Register

The device command register is used to issue commands to the device. The register is isolated from other registers in the register map to help prevent accidental writing.

| Device Command Register | | | | | |
|---|---|---|---|---|---|
| **Register Number** | **Size (Registers)** | **Data Type** | **Access** | **Name** | **Default Value** |
| 1065 | 1 | USHORT | R/W | Device Command | 0 |

Writing a value shown in the following table will issue the corresponding device command. Attempting to write any other value will generate an exception response with the ILLEGAL WRITE VALUE exception code. Reading the register always returns zero.

| Device Commands | | |
|---|---|---|
| **Value** | **Name** | **Description** |
| 56573 (0xDCFD) | Factory Defaults | Resets device configuration registers to their factory default values. The encryption key is erased and encryption is disabled. Does not affect the communication settings, the data log or the diagnostic registers. |
| 56557 (0xDCED) | Erase Data Log | Erases the data log. Data cannot be recovered after issuing this command. The response message is sent after erasure is complete. Typical response time is 30 seconds. The maximum response time is 90 seconds. Erasing the data log does not affect the current encryption key. |
| 56541 (0xDCDD) | Download Data Log | Starts the fast data download process. The maximum response time to this command is 1000 milliseconds. Refer to the Data Retrieval Registers section for a description of the command. |
| 56320 (0xDC00) | Security Mode | Places the device in the security mode, requiring a master device to log in with a password to regain access. The command has no effect if password security is disabled. |

## 7.3 Device Status Registers

The device status registers allow the master device to obtain the current operating status of the device.

| Device Status Registers | | | | | |
|---|---|---|---|---|---|
| **Register Number** | **Size (Registers)** | **Data Type** | **Access** | **Name** | **Default Value** |
| 1070 | 1 | USHORT | R/W | Device Status | 0 |
| 1071 | 1 | SHORT | R | Ambient Temperature | °C x 10 |
| 1072 | 1 | USHORT | R | Input Voltage | mV |
| 1073 | 1 | USHORT | R | Charge Voltage | 0 mV |
| 1074 | 4 | TIME | R/W | Date and Time | UTC |

### 7.3.1 Device Status

This register contains device operational status. Each bit in the register represents a status value as defined in the following table.

| Device Status Register | | | |
|---|---|---|---|
| **Bit** | **Mask** | **Name** | **Description** |
| 0 | 0x0001 | Power Outage | A power outage caused a hardware reset and recovery |
| 1 | 0x0002 | Low Voltage | The input voltage is below the warning threshold |
| 2 | 0x0004 | Clock Battery | Timekeeping backup battery is low |
| 3 | 0x0008 | Clock Fault | Timekeeping restarted at 2000-01-01 00:00:00.00 |
| 4 | 0x0010 | Clock Adjusted | Date and Time were changed |
| 5 | 0x0020 | Device Fault | A device fault caused a reset and recovery |
| 6 | 0x0040 | Temperature | Device operating temperature range was exceeded |
| 7 | 0x0080 | Reserved | Always returns 0 |
| 8 | 0x0100 | Encryption Enabled | Encryption key is non-zero, data records will be encrypted |
| 9 | 0x0200 | Reserved | Always returns 0 |
| 10 | 0x0400 | Reserved | Always returns 0 |
| 11 | 0x0800 | Reserved | Always returns 0 |
| 12 | 0x1000 | Reserved | Always returns 0 |
| 13 | 0x2000 | Reserved | Always returns 0 |
| 14 | 0x4000 | Reserved | Always returns 0 |
| 15 | 0x8000 | Reserved | Always returns 0 |

The status register is non-volatile: information is retained across a power outage. The status represents events that occurred since the last time the register was cleared. The device status is recorded with each logged data record. The status register can be cleared at any time by writing zero to it. Any conditions that persist or reoccur after being cleared will cause those bits to be set again. Attempting to

write any value other than zero to clear the register will generate an exception response with the ILLEGAL WRITE VALUE exception code.

### 7.3.2 Ambient Temperature

This register returns the ambient temperature in degrees Celsius (°C) multiplied by 10, providing a temperature resolution of 0.1 °C. For example, an ambient temperature of 23.7 °C will be read as 237.

### 7.3.3 Input Voltage

This register returns the input voltage applied to the device in millivolts (mV). For example, an input voltage of 13.54 volts will be read as 13540.

### 7.3.4 Charge Voltage

This register returns the charging voltage applied to the device in millivolts (mV). The PWS-400 does not support battery charging and will always return zero.

### 7.3.5 Date and Time

These registers hold the current date and time as a TIME data type. The device is programmed to Universal Coordinated Time (UTC) at the factory and generally does not need to be adjusted. If the clock should require adjustment, it is recommended that UTC continue to be used as the clock does not provide for automatic adjustments for local Daylight Savings Time. The time stamps of stored data may be post-processed to any local time as needed for reporting or analysis.

When setting the clock, the year must be in the range 2000 to 2399. Leap years in this range are correctly handled. Attempting to write an invalid date or time will generate an exception response with the ILLEGAL WRITE VALUE exception code. Setting the clock will clear the clock fault bit and set the clock adjusted bit in the status register.

## 7.4 RS485 Configuration Registers

These registers specify the communication configuration of the RS485 port.

| RS485 Configuration Registers | | | | | |
|---|---|---|---|---|---|
| **Register Number** | **Size (Registers)** | **Data Type** | **Access** | **Name** | **Default Value** |
| 1101 | 1 | USHORT | R/W | RS485 Communication Settings | 276 |
| 1102 | 1 | USHORT | R/W | RS485 Message Timeout (ms) | 0 |
| 1103 | 1 | USHORT | R/W | RS485 Sleep Timeout (ms) | 0 |
| 1104 | 1 | USHORT | R/W | RS485 Good Message Counter | 0 |
| 1105 | 1 | USHORT | R/W | RS485 Bad Message Counter | 0 |
| 1106 | 1 | USHORT | R/W | RS485 Exception Response Counter | 0 |

### 7.4.1 RS485 Communication Settings

This register sets the communication parameters of the RS485 port. The default value is 20 (0x14) for RTU mode at 19200 baud, 8 bits, even parity, and 1 stop bit. The register is non-volatile.

| Register Bits | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | Mode | Stop | Parity | | Data | Baud Rate | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **(default) RTU transmission mode** | | | | | | **0** | **1** | | | | | | | | |
| ASCII transmission mode | | | | | | 1 | 0 | | | | | | | | |
| **(default) 1 stop bit** | | | | | | | | **0** | | | | | | | |
| 2 stop bits | | | | | | | | 1 | | | | | | | |
| **(default) Even parity** | | | | | | | | | **0** | **0** | | | | | |
| Odd parity | | | | | | | | | 0 | 1 | | | | | |
| No parity | | | | | | | | | 1 | 0 | | | | | |
| 7 data bits | | | | | | | | | | | 0 | | | | |
| **(default) 8 data bits** | | | | | | | | | | | **1** | | | | |
| 1200 baud | | | | | | | | | | | | 0 | 0 | 0 | 0 |
| 2400 baud | | | | | | | | | | | | 0 | 0 | 0 | 1 |
| 4800 baud | | | | | | | | | | | | 0 | 0 | 1 | 0 |
| 9600 baud | | | | | | | | | | | | 0 | 0 | 1 | 1 |
| **(default) 19200 baud** | | | | | | | | | | | | **0** | **1** | **0** | **0** |
| 38400 baud | | | | | | | | | | | | 0 | 1 | 0 | 1 |
| 57600 baud | | | | | | | | | | | | 0 | 1 | 1 | 0 |

When this register is written, the response will be returned at the previous communication settings. All subsequent commands must be sent at the new settings. Attempting to write any combination of bits not listed in the above table will result in an exception response with the ILLEGAL WRITE VALUE exception code. Note that 7 data bits in the RTU transmission mode is an invalid combination; the device will respond with an exception and the ILLEGAL WRITE VALUE exception code.

### 7.4.2 RS485 Message Timeout

If RTU mode is selected, this register specifies the amount of idle time that must elapse before the device recognizes the end of a message. A value of zero specifies a timeout that is selected automatically based on the selected baud rate as described in the RTU Message Format section. If the master device is unable to meet the default timing requirements, a fixed timeout from 5 to 50 milliseconds may be specified. The register is non-volatile.

If ASCII mode is selected, this register specifies the maximum time that may elapse between characters within a message when the ASCII transmission mode is selected. Intervals exceeding the timeout value will cause the device to assume an error has occurred and discard the message. The valid range for the timeout in ASCII mode is 1000 to 60000 milliseconds (1 to 60 seconds). The recommended setting for most applications is 1000 milliseconds.

### 7.4.3 RS485 Sleep Timeout

This register specifies the amount of time allowed to elapse with no network activity before the device enters its low power sleep mode. The valid range for the timeout is 0, or 1000 to 60000 milliseconds (1 to 60 seconds). A value of zero disables the low power sleep mode. The register is non-volatile.

After the device enters its sleep mode, any activity detected on the network will cause the device to wake; however, the contents of the first message will most likely be missed. If this first message was a command addressed to the device, the master device will receive no response and must retry the command. Afterwards, as long as there is network activity more frequent than the sleep timeout setting, the device will remain awake and process commands.

> *Note: The sleep timeout takes precedence over the message timeout. If the sleep timeout is used (non-zero) it should be set greater than or equal to the message timeout.*

### 7.4.4 RS485 Message Counters

The message counter registers provide diagnostic information for troubleshooting communication problems. Each counter is reset by writing it to zero. The counters roll back to zero after the maximum count value of 65535 is reached. The registers are volatile and will reset to zero if power is removed.

The Good Message Counter counts properly formatted messages that are addressed to the device. The Bad Message Counter tracks the number of improperly formatted messages, such as those with a bad CRC. The Exception Response Counter counts the number of messages received that were rejected with an exception response.

## 7.5 Data Log Recording Registers

These registers configure data log security and allow the master device to record registers in the data log.

| Data Log Recording Registers | | | | | |
|---|---|---|---|---|---|
| **Register Number** | **Size (Registers)** | **Data Type** | **Access** | **Name** | **Default Value** |
| 1300 | 8 | STRING | W | Encryption Key | |
| 1320 | 1 | USHORT | W | Register Count | |
| 1321 | 64 | USHORT | W | Data Registers | |

### 7.5.1 Encryption Key

Write this 8-register string to set a 128-bit encryption key. Afterwards, all data records written to the data log will be encrypted using this key. The registers are write-only so that the key is protected. The registers are non-volatile, so the key need only be written once prior to recording data. When shipped from the factory or after a factory defaults command is issued, the key is erased and encryption is disabled. Issuing a factory defaults command does not remove encryption from data already stored in the data log.

It is recommended that the user not mix data recorded with different encryption keys, or mix encrypted and unencrypted data in the same log, as this will make decrypting the log difficult. Changing the encryption key does not change the encryption of previously recorded data. The data log should be erased prior to, or immediately after, changing the encryption key.

> ***Note: It is the responsibility of the user to manage encryption keys. Once written, the encryption key cannot be read out of the device. Once data is recorded with encryption, it cannot be decrypted without the original key.***

### 7.5.2 Recording Data

Data is recorded in the data log using the following procedure.

1. Write the number of registers to be recorded (1 to 64) to register 1320. Attempting to write a value outside this range will result in an exception response with the ILLEGAL WRITE VALUE exception code.

2. Write data registers beginning at register 1321. Only the number of data registers specified by the register count can be written. Attempting to write more data registers than specified by the register count will result in an exception response with the ILLEGAL ADDRESS VALUE exception code. A single Write Multiple Registers function can be used to write the register count and the data registers in one transaction.

3. When the last data register is written as determined by the register count, the device will write the record to the data log along with the time, site id, device id, device serial number, device status, ambient temperature, and input voltage. The typical response time to this write is less than 100 milliseconds. The maximum response time is 3 seconds.

The register count and data registers are write-only to protect unencrypted data from being read.

### 7.5.3 Calculating Data Record Capacity

The data record capacity of the data log varies with the number of data registers in each record. The capacity is calculated as follows.

1. Determine R, the number of data registers in each record. This is the value that is written to the Register Count.

2. The number of bytes in a record is B = 2 x R + 30.

3. If encryption is enabled, set E = B – 6. If E is not evenly divisible by 16, increase E so that E is an even multiple of 16. Then set B = E + 6.

4. The number of records in a block is the integer part of N = 65532 / B. A block is the smallest part of the log memory that can be erased when the log begins to wrap around.

5. The total number of records is 64 * N.

The calculation assumes that all records will be of equal size. For example, if 4 registers are written per record, R = 4. With no encryption, the number of bytes in a record is B = 2 x 4 + 30 = 38. The number of records in a block is the integer part of N = 65532 / 38 = 1,724. The total number of records is 1,724 x 64 = 110,336. Once the log becomes full, the oldest 1,724 data records will be erased to make room for new data.

The total number of records can be used to determine how long it will take to fill the log memory at a given recording interval. Continuing with the previous example, if the recording interval is 5 minutes, it will take 383 days to fill the log memory: 110,366 x 5 minutes / 1440 minutes per day.

The total number of records can also be used to determine the fastest recording interval that can be used to fill the log memory in a given time. Again, using the example above, if data will be retrieved once per month, a recording interval as fast as 30 seconds can be used: 31 days * 86,400 seconds per day / 110,336 (the data log will actually fill in 38 days at this interval).

The following tables provide some examples of the recording time available for various data record sizes, with and without encryption, before the log begins to wrap around. The tables assume that all data records are the same size.

| Register Count | Record Capacity | Recording Interval without Encryption | | | |
|---|---|---|---|---|---|
| | | 30 seconds | 1 minute | 5 minutes | 15 minutes |
| 1 | 131,008 | 45 days | 90 days | 454 days | 1,364 days |
| 2 | 123,328 | 42 days | 85 days | 428 days | 1,284 days |
| 4 | 110,336 | 38 days | 76 days | 383 days | 1,149 days |
| 8 | 91,136 | 31 days | 63 days | 316 days | 949 days |
| 16 | 67,584 | 23 days | 46 days | 234 days | 704 days |
| 32 | 44,608 | 15 days | 30 days | 154 days | 464 days |
| 48 | 33,280 | 11 days | 23 days | 115 days | 346 days |
| 64 | 26,496 | 9 days | 18 days | 92 days | 276 days |

| Register Count | Record Capacity | Recording Interval with Encryption | | | |
|---|---|---|---|---|---|
| | | 30 seconds | 1 minute | 5 minutes | 15 minutes |
| 1 | 110,336 | 38 days | 76 days | 383 days | 1,149 days |
| 2 | 110,336 | 38 days | 76 days | 383 days | 1,149 days |
| 4 | 110,336 | 38 days | 76 days | 383 days | 1,149 days |
| 8 | 77,632 | 26 days | 53 days | 269 days | 808 days |
| 16 | 59,904 | 20 days | 41 days | 208 days | 624 days |
| 32 | 41,088 | 14 days | 28 days | 142 days | 428 days |
| 48 | 31,296 | 10 days | 21 days | 108 days | 326 days |
| 64 | 25,216 | 8 days | 17 days | 87 days | 262 days |

### 7.5.4 Recording Speed

The PWS-400 is capable of recording data as fast as 1 record per second provided that the data log has not wrapped around. Once the data log has wrapped around, the recording rate drops to 1 record every 5 seconds. This is necessary in order to provide time for block erasures when the oldest data is erased.

If the slower rate is undesirable, log data should be retrieved before the data log fills, then the data log erased and a new log started.

## 7.6 Data Log Retrieval Registers

These registers allow a master device to retrieve data records from the data log.

| Data Log Retrieval Registers | | | | | |
|---|---|---|---|---|---|
| **Register Number** | **Size (Registers)** | **Data Type** | **Access** | **Name** | **Default Value** |
| 1400 | 2 | ULONG | R | Data Log Size (bytes) | 4,194,304 |
| 1402 | 2 | ULONG | R | Data Log Used (bytes) | 0 |
| 1404 | 2 | ULONG | R | Lowest Record Number | 0 |
| 1406 | 2 | ULONG | R | Highest Record Number | 0 |
| 1408 | 1 | USHORT | R/W | Download Record Count | 0 |
| 1409 | 2 | ULONG | R/W | Record Number | 0 |
| 1411 | 1 | USHORT | R | Record Size in Registers | 0 |
| 1412 | 4 | TIME | R | Date and Time | 0 |
| 1416 | 1 | USHORT | R | Site Id | 0 |
| 1417 | 1 | USHORT | R | Device Id | 0 |
| 1418 | 2 | ULONG | R | Device Serial Number | 0 |
| 1420 | 1 | USHORT | R | Device Status | 0 |
| 1421 | 1 | SHORT | R | Ambient Temperature | °C x 10 |
| 1422 | 1 | USHORT | R | Input Voltage | mV |
| 1423 | 1 | USHORT | R | Register Count | 0 |
| 1424 | 64 | USHORT | R | Data Registers | 0 |
| 1488 | 4 | USHORT | R | Encryption Padding Registers | |

### 7.6.1 Data Log Size

This register returns the data log memory size in bytes.

### 7.6.2 Data Log Used

This register returns the number of data log memory bytes currently used. Combined with the data log size, the register can be used to calculate the percent of memory used and percent of memory remaining.

### 7.6.3 Lowest and Highest Record Numbers

If no records have been logged, both the lowest and highest record numbers will be zero. Records are numbered beginning at record one. The highest record number will increment with each record logged (the highest record number is also the total number of records recorded since the log was last erased). As long as the data log hasn't wrapped, the lowest record number will remain at one. Once the log wraps around, the lowest record number will begin to advance as the oldest data is deleted. Every record will have a unique and always increasing number. If the user keeps track of the last record number that was retrieved, only higher record numbers will need to be retrieved at the next download.

### 7.6.4 Download Record Count

This register specifies the maximum number of records to transfer when the Data Log Download command is used. The default value of zero permits continuous streaming of data records. This record is non-volatile. The content of this register has no effect when reading data records via the Data Log Retrieval registers.

### 7.6.5 Record Number

Write this register to specify the data record number to read. If the write is successful, the remaining registers are filled in with the record data and are ready to be read. The average response time is 500 milliseconds seconds. The maximum response time is 1000 milliseconds.

Attempting to write a record number less than the lowest record number or greater than the highest record number will result in an exception response with the ILLEGAL WRITE VALUE exception code.

If a data record is found to be corrupt the device will return an exception response with the CORRUPT DATA RECORD exception code. As much of the data record will be retrieved as possible; however, the register count and/or contents of the record are suspect. If the register count returns zero, the record could not be retrieved. A corrupt record can occur if power fails during the time the data record is being written.

### 7.6.6 Record Size

After writing the record number, this register provides the size of the record in registers. The size includes all registers from the first register of the time stamp to the last data register.

### 7.6.7 Record Data

After writing the record number, the record data registers can be read. All data registers may be read, including those beyond the record size. This permits the master device to read all of the data record registers with one command regardless of the record size. Only the data registers specified by the register count will contain valid information.

If the data is unencrypted, the register numbers and sizes shown in the table above can be read to extract portions of the data record if not all of the information is needed.

If the data is encrypted, reading individual registers from the table will not be useful. A read registers command must be used to read the number of registers specified by the record size. All of the registers in the record are required in the decryption process.

### 7.6.8 Data Decryption

The Prairie View Software that comes with the device provides data decryption and the capability to view and export data from encrypted logs.

The Advanced Encryption Standard is supported by many operating systems, so it is possible for user applications and third-party software to decrypt logs, providing that the original encryption key is known.

### 7.6.9 Data Retrieval Procedure

The following procedure describes data retrieval using conventional MODBUS register read and write commands.

1. The master device reads the lowest and highest record number registers to determine the valid record number range and selects the range of records to be downloaded.

2. The master device writes a record number to the record number register. The PWS-400 retrieves the record from the data log.

3. The master device reads the record size register to determine the number of registers in the record.

4. The master device reads the number of registers specified by the size registers beginning with the time register.

5. Optionally, after the master device has written the record number, it can simply read all record registers from the register size through the encryption padding registers and decode the record later.

6. The master device repeats steps 2 through 6 until all selected records have been retrieved.

### 7.6.10 Data Log Download Command

The following procedure describes the use of the data log download command. This command provides considerably faster downloads as compared to conventional register access.

1. The master device reads the lowest and highest record number registers to determine the valid record number range.

2. The master device writes the first record number to be downloaded to the record number register. To download all data, set the record number register equal to the lowest record number. The PWS-400 has a maximum response time of 1000 milliseconds to this command.

3. The master device writes the data log download command to the command register. The PWS-400 will return a standard response to this write register command with a maximum response time of 1000 milliseconds.

4. After a 500 millisecond delay, the PWS-400 will begin to stream the number of records specified by the download record count, beginning at the record specified by the record number register. If the download record count is set to zero, the download will continue uninterrupted to the highest record number. Each record is constructed as a standard MODBUS read registers response that begins at the record number register and contains all registers in the record. There will be little or no delay between records.

5. After all records have been sent, the PWS-400 will send an exception response with the END DOWNLOAD exception code.

6. The master device must be capable of receiving records back-to-back until it detects the END DOWNLOAD exception.

The user should consider the following when using this command to download data.

- There is no handshaking. The master device must be capable of receiving records back-to-back with little or no delay between records. Since the record number is included in each record sent and each record contains a CRC or LRC, the master device can determine if any records were missed.

- On the RS485 port, there is no way to cancel the command. The PWS-400 will stream data until it is done. The network is unavailable for any other communication until the download is finished.

- If the PWS-400 encounters a corrupt record during the download, it will send an exception response with the CORRUPT DATA RECORD exception code in place of the corrupt record and the download will continue.

## 7.7 Password Security Registers

Additional module security is available by enabling password protection. When a security password is written to the PWS-420, the master device must log in with the correct password in order to access any of the device registers. When the device is secure, only the read slave id function and the write to the login password registers are possible.

| Data Log Retrieval Registers | | | | | |
|---|---|---|---|---|---|
| **Register Number** | **Size (Registers)** | **Data Type** | **Access** | **Name** | **Default Value** |
| 3000 | 10 | STRING | W | Security Password | 0 |
| 3010 | 10 | STRING | W | Login Password | 0 |

### 7.7.1 Security Password

This ten-register, non-volatile, register string sets the password for the device. By default, the registers are set to zero, disabling password protection. The device slave id and all device registers are open and accessible.

Writing a 10-register, non-zero password to these registers enables password security. If no communication port activity is detected after a period of 60 seconds, or if the Security Mode device command (56320) is issued, the device will enter its secure mode. In the security mode, only the slave id and the login password registers are available. Attempting to perform any other function while in the security mode will result in an exception response with the SECURITY MODE exception code.

> ***Note: It is the responsibility of the user to manage passwords. Once written, the security password cannot be read out of the device. Once the device is secured with a password, it cannot be accessed without the original password.***

### 7.7.2 Login Password

Writing a 10-register password that matches the Security Password will temporarily take the device out of the security mode. Logging in with a password that does not match the security password will result in an exception response with the ILLEGAL WRITE VALUE exception code and, if the device is in the security mode, the device will remain in the security mode.

> *Note: When password security is in effect, the login password is required independently from the RS485 network master device and the Bluetooth master device.*

## 7.8 Diagnostic Registers

These registers provide device diagnostic and troubleshooting information.

| Device Description | | | | | |
|---|---|---|---|---|---|
| **Register Number** | **Size (Registers)** | **Data Type** | **Access** | **Name** | **Default Value** |
| 9000 | 1 | USHORT | R | Configuration Flash Writes. | < 50,000 |
| 9001 | 1 | USHORT | R | Last Reset Type | 0 |
| 9002 | 1 | USHORT | R | Fault Count | 0 |
| 9003 | 1 | USHORT | R | Last Fault Type | 0 |
| 9004 | 1 | USHORT | R | Last Fault Information | 0 |
| 9005 | 1 | SHORT | R | High Temperature | °C x 10 |
| 9006 | 1 | SHORT | R | Low Temperature | °C x 10 |
| 9007 | 1 | USHORT | R | Data Log Chip Id | 8214 or 9538 |
| 9008 | 1 | USHORT | R | Data Log Erasure Count | < 50,000 |

### 7.8.1 Configuration Flash Writes

This non-volatile register records the number of times the device configuration registers have been written to. Since these registers are intended for occasional configuration of the device, a high number indicates frequent changes and possibly improper use of the registers.

### 7.8.2 Last Reset Type

This register records information about the last reset event that caused the power outage bit to be set in the device status register.

| Reset Types | |
|---|---|
| **Type** | **Description** |
| 0 | None recorded |
| 1 | Power outage detected by the CPU – power lost and restored |
| 2 | Brownout detected by CPU – marginal power source (e.g. very low battery) |
| 3 | Brownout detected by the timekeeping circuit – marginal power source |

### 7.8.3 Fault Information

These registers record information about the last fault event that caused the device fault bit to be set in the device status register. The fault count records the number of fault events. An occasional fault may indicate a normal recovery from an unusual power loss sequence or transient event. Frequent faults may indicate hardware damage or a firmware defect.

| Fault Types | |
|---|---|
| **Type** | **Description** |
| 0 | None recorded |
| 4 | Invalid interrupt, last fault information register updated with interrupt number |
| 5 | Memory access violation |
| 6 | Watchdog timeout |

### 7.8.4 High and Low Temperatures

These registers record the ambient temperature extremes (in degrees Celsius multiplied by 10) experienced by the device.

### 7.8.5 Data Log Chip Id

This register reports the identification of the internal data log chip. The value may be any one of the values listed in the table.

### 7.8.6 Data Log Erasure Count

This non-volatile register records the number of times the data log has wrapped around and/or has been erased.

# 8 MODBUS Protocol

The PWS-400 utilizes the MODBUS over Serial Line protocol for communication over an RS485 network.

The MODBUS Serial Line protocol is a half-duplex, master-slave protocol. One master device controls the network or link, and is connected to one or more slave devices. A MODBUS transaction is always initiated by the master device. Slave devices never transmit data without first receiving a request from the master device. Slave devices never communicate with each other. The master device only initiates one MODBUS transaction with one slave device at a time. The PWS-400 implements the functions of a slave device in the protocol.

The MODBUS standard defines two serial transmission modes: RTU mode and ASCII mode. All devices connected to a network must be configured to use the same transmission mode and communication parameters. This device supports both transmission modes.

## 8.1 RTU Transmission Mode

The MODBUS RTU (Remote Terminal Unit) transmission mode is an 8-bit byte-oriented binary protocol with timing-based message framing. The RTU mode is the default transmission mode for the device.

### 8.1.1 RTU Character Format

The standard format for each character in an RTU message is as follows.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|
| Start | Bit 0 | Bit 1 | Bit 2 | Bit 3 | Bit 4 | Bit 5 | Bit 6 | Bit 7 | Parity | Stop |

The 11-bit character is sent least-significant bit first, left to right as shown, with 1 start bit, 8 data bits, a parity bit, and one stop bit. The default parity of the device is even parity. Odd parity and no parity are also supported.

When no parity is specified, 2 stop bits should be used so that the 11-bit character size is maintained:

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|
| Start | Bit 0 | Bit 1 | Bit 2 | Bit 3 | Bit 4 | Bit 5 | Bit 6 | Bit 7 | Stop | Stop |

### 8.1.2 RTU Message Format

An RTU message is comprised of a sequence of 4 to 256 characters transmitted in a continuous stream. No more than 1.5 character times should be permitted between the characters in a message. The format of a message is as follows.

| Slave Address | Function Code | Data | CRC |
|---|---|---|---|
| 1 byte | 1 byte | 0 to 252 bytes | 2 bytes |

By default, for baud rates less than or equal to 19200, an idle time of 3.5 character times signifies the end of a message. For baud rates higher than 19200, an idle time of 1.75 milliseconds signifies the end of a message.

The RTU mode uses a Cyclical Redundancy Check (CRC) to provide error checking across a message, from the slave address to the last data byte. Only the eight data bits of each character are used in generating the CRC; start and stop bits and the parity bit do not apply. The CRC field is a 16-bit value. The low-order byte is appended first, followed by the high-order byte. The high-order byte of the CRC is the last byte in the message. A slave device will not act on or respond to a message that has an invalid CRC or parity errors. The master device must discard a slave response with an invalid CRC or parity errors and either resend the message or generate an error.

Techniques for calculating the CRC can be found in the document *MODBUS over Serial Line Specification and Implementation Guide V1.02*, available at modbus-ida.org.

## 8.2 ASCII Transmission Mode

In the MODBUS ASCII transmission mode, each 8-bit byte in a message is sent as two hexadecimal ASCII characters. The message is also framed with unique ASCII characters. This mode is preferred when the communication link cannot guarantee the inter-character timing required by the RTU mode, such as a packet-based wireless link.

### 8.2.1 ASCII Character Format

The standard format for each character in an RTU message is as follows.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|
| Start | Bit 0 | Bit 1 | Bit 2 | Bit 3 | Bit 4 | Bit 5 | Bit 6 | Parity | Stop |

The 10-bit character is sent least-significant bit first, left to right as shown, with 1 start bit, 7 data bits, a parity bit, and one stop bit. The default parity of the device is even parity; odd parity and no parity are also supported.

When no parity is specified, 2 stop bits should be used so that the 10-bit character size is maintained:

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|
| Start | Bit 0 | Bit 1 | Bit 2 | Bit 3 | Bit 4 | Bit 5 | Bit 6 | Stop | Stop |

Although not specified by the MODBUS standard, the PWS-400 will also support the ASCII mode using 8-bit characters with or without parity as described for the RTU mode.

### 8.2.2 ASCII Message Format

An ASCII message is comprised of a sequence of 9 to 513 characters. The format of a message is as follows.

| Start | Slave Address | Function Code | Data | LRC | End |
|---|---|---|---|---|---|
| 1 char ':' | 2 chars<br>(1 byte) | 2 chars<br>(1 byte) | 0 to 504 chars<br>(0 to 254 bytes) | 2 chars | 2 chars CR, LF |

The ASCII colon character ":" begins a message. The message ends with the ASCII control characters carriage return (CR) and line feed (LF). The byte information of an ASCII message from the slave address to the end of the data is identical to that of an RTU message. Each byte of the message is transmitted as two ASCII hexadecimal characters 0-9, A-F.

By default, intervals up to one second may elapse between characters within the message. The device supports a programmable timeout interval up to 60 seconds. Intervals exceeding the timeout value will cause the device to assume an error has occurred and discard the message. Each reception of a colon character signals the beginning of a new message. If a message was in the process of being received when a colon is received, the previous message will be discarded.

The ASCII mode uses a Longitudinal Redundancy Check (LRC) to provide error checking across a message, from the slave address to the last data byte. The beginning colon and ending CR-LF pair are not included in the calculation. Only the data bits of each character are used in generating the LRC; start and stop bits and the parity bit do not apply. The LRC is an 8-bit value, encoded with two ASCII characters in the same manner as a data byte. A slave device will not act on or respond to a message that has an invalid LRC or parity errors. The master device must discard a slave response with an invalid LRC or parity errors and either resend the message or generate an error.

Techniques for calculating the LRC can be found in the document *MODBUS over Serial Line Specification and Implementation Guide V1.02*, available at modbus-ida.org.

## 8.3 Device Addressing

The MODBUS master device has no specific address, only the slave devices have an address. Each slave device on a network must be assigned a unique address. Slave addresses can range from 1 to 247. Address 0 is reserved as the broadcast address. The device will recognize the broadcast address as well as its own address. No response is returned to broadcast requests.

## 8.4 Data Types

The MODBUS Protocol defines four primary data models: Discrete Inputs, Coils, Input Registers, and Holding Registers. The PWS-400 uses the Holding Registers model exclusively.

MODBUS uses big endian representation for register numbers and data items – when a numerical quantity larger than a single byte is transmitted, the most significant byte is sent first. Register data are packed as two bytes per register, the first byte contains the high order bits and the second byte contains the low order bits.

The device extends the big endian representation to data types requiring multiple registers. The first register contains the high order bytes and subsequent registers contain the lower order bytes.

### 8.4.1 USHORT: Unsigned Short

An unsigned short is a 16-bit unsigned integer value in the range 0 to 65535. The value is contained in a single register. This is the basic MODBUS holding register data type.

| **Register** | |
|---|---|
| Byte 1 | Byte 2 |
| Bits 15 - 8 | Bits 7 - 0 |

### 8.4.2 SHORT: Signed Short

A signed short is a 16-bit two's complement signed integer value in the range -32768 to +32767. The value is contained in a single register.

| **Register** | |
|---|---|
| Byte 1 | Byte 2 |
| Sign, Bits 14 - 8 | Bits 7 - 0 |

### 8.4.3 ULONG: Unsigned Long

An unsigned long is a 32-bit unsigned integer value contained in two consecutive registers. Values can range from 0 to 4,294,967,295.

| **Register** | | **Register + 1** | |
|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 |
| Bits 31 - 24 | Bits 23 - 16 | Bits 15 - 8 | Bits 7 - 0 |

### 8.4.4 LONG: Signed Long

A signed long is a 32-bit two's complement signed integer value contained in two consecutive registers. Values can range from -2147483648 to +2147483647.

| **Register** | | **Register + 1** | |
|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 |
| Sign, Bits 30 - 24 | Bits 23 - 16 | Bits 15 - 8 | Bits 7 - 0 |

### 8.4.5 FLOAT: Floating Point

A float is a 32-bit IEEE-754 floating point value contained in two consecutive registers.

| Register | | Register + 1 | |
|---|---|---|---|
| Byte 1 | Byte 2 | Byte 3 | Byte 4 |
| SXXXXXXX | XMMMMMMM | MMMMMMMM | MMMMMMMM |

S is the sign bit, X is the 8-bit exponent, and M is the 23-bit mantissa.

### 8.4.6 STRING: Character String

A string is a sequence of consecutive registers where each byte within a register represents an 8-bit ASCII-encoded character. A string of N characters requires N / 2 registers. When reading or writing a string, all characters in the string must be transmitted. If the string to be written does not require the full available length, the unused characters must be padded with zeroes.

| Register | |
|---|---|
| Byte 1 | Byte 2 |
| Character 1 | Character 2 |

| Register + 1 | |
|---|---|
| Byte 1 | Byte 2 |
| Character 3 | Character 4 |

.
.
.

| Register + (N / 2) – 1 | |
|---|---|
| Byte 1 | Byte 2 |
| Character N - 1 | Character N |

### 8.4.7 TIME: Date and Time

Time is represented by 16 packed BCD digits contained in 4 consecutive registers. The registers are organized with the most significant temporal digit first, ranging from 1000 years to 0.01 seconds. Only BCD digits '0' to '9' are used. Time is in the 24-hour format.

| **Register** | | | |
|---|---|---|---|
| Byte 1 | | Byte 2 | |
| 1000 years | 100 years | 10 years | Year |

| **Register + 1** | | | |
|---|---|---|---|
| Byte 1 | | Byte 2 | |
| 10 Months | Month | 10 Days | Day |

| **Register + 2** | | | |
|---|---|---|---|
| Byte 1 | | Byte 2 | |
| 10 Hours | Hour | 10 Minutes | Minutes |

| **Register + 3** | | | |
|---|---|---|---|
| Byte 1 | | Byte 2 | |
| 10 Seconds | Seconds | 0.1 Seconds | 0.01 Seconds |

## 8.5 Function Codes

The MODBUS Protocol defines a variety of standard function codes for reading and writing data to a device. Only the function codes described in this section are supported by the device. All other function codes will generate an exception response with the ILLEGAL FUNCTION exception code.

### 8.5.1 Report Slave Id

This function code is used to read the device description. All of the device-specific information is also available in the register map and is detailed in that section.

| Command | | | |
|---|---|---|---|
| Byte Offset | Field Description | Type | Value |
| 0 | Device Address | BYTE | 1-247 |
| 1 | Function Code | BYTE | 17 (0x11) |

| Response | | | |
|---|---|---|---|
| Byte Offset | Field Description | Type | Value |
| 0 | Device Address | BYTE | 1-247 |
| 1 | Function Code | BYTE | 17 (0x11) |
| 2 | Byte Count | BYTE | 18 |
| 3 | Slave Id | BYTE | 80 (0x50) 'P' |
| 4 | Run Status Indicator | BYTE | 255 (0xFF) = ON |
| 5 | Slave Id Version | USHORT | 1 |
| 7 | Device Id | USHORT | 400 |
| 9 | Serial Number | ULONG | |
| 13 | Firmware Version | USHORT | |
| 15 | Boot Code Version | USHORT | |
| 17 | Hardware Version | USHORT | |
| 19 | Register Map Version | USHORT | |

The response to a slave id command is typically returned in less than 50 milliseconds.

### 8.5.2 Read Registers

This function code is used to read the contents of a contiguous block of registers.

| Command | | | |
|---|---|---|---|
| Byte Offset | Field Description | Type | Value |
| 0 | Device Address | BYTE | 1-247 |
| 1 | Function Code | BYTE | 3 (0x03) |
| 2-3 | Starting Register Address | USHORT | 0-65535 (0xFFFF) |
| 4-5 | Number of Registers (N) | USHORT | 1-125 (0x007D) |
| 6 | CRC Low | BYTE | |
| 7 | CRC High | BYTE | |

| Response | | | |
|---|---|---|---|
| Byte Offset | Field Description | Type | Value |
| 0 | Device Address | BYTE | 1-247 |
| 1 | Function Code | BYTE | 3 (0x03) |
| 2 | Byte Count | BYTE | 2 x N |
| 3 | Register Data | 2 x N BYTES | |

The starting register address is one less than the starting register number. The byte count returned will be two times the number of registers read.

- If the Read Registers function code is used to read data from a single register, the single register must be a single-register data type such as a USHORT. An attempt to read a single register from within a multiple-register data type will result in an exception response with the ILLEGAL DATA ADDRESS exception code.

- If the Read Registers function code is used to read a multiple-register data type, the start register address must be of the first register in the field and the number of registers must include all registers in the field. If one or both parameters are invalid, the device will return an exception response with the ILLEGAL DATA ADDRESS exception code.

- The Read Registers function code can be used to read data from multiple registers of the same or different data types. In this case, the starting register address must be of either a single-register data type or the first register in a multiple-register data type. The last register read must be either a single-register data type or the last register in a multiple-register data type. Attempting to start or end the read within a multiple-register data type will result in an exception response with the ILLEGAL DATA ADDRESS exception code.

- Some registers are designated as write-only (W versus R/W in the register map). Attempting to read a write-only register, the device will result in an exception response with the WRITE-ONLY REGISTER exception code.

The response to a read registers command is typically returned in less than 50 milliseconds.

### 8.5.3 Write Multiple Registers

This function code is used to write the contents of a contiguous block of registers.

| Command | | | |
|---|---|---|---|
| Byte Offset | Field Description | Type | Value |
| 0 | Device Address | BYTE | 1-247 |
| 1 | Function Code | BYTE | 16 (0x10) |
| 2 | Starting Register Address | USHORT | 0-65535 (0xFFFF) |
| 4 | Number of Registers (N) | USHORT | 1-123 (0x007B) |
| 6 | Byte Count | BYTE | 2 x N |
| 7 | Register Data | 2 x N BYTES | |

| Response | | | |
|---|---|---|---|
| Byte Offset | Field Description | Type | Value |
| 0 | Device Address | BYTE | 1-247 |
| 1 | Function Code | BYTE | 16 (0x10) |
| 2 | Starting Register Address | USHORT | 0-65535 (0xFFFF) |
| 4 | Number of Registers | USHORT | N |

The starting register address is one less than the starting register number. The byte count must be two times the number of registers to be written.

- If the Write Multiple Registers function code is used to write data to a single register, the register must be a single-register data type such as a USHORT. An attempt to write a single register within a multiple-register data type will result in an exception response with the ILLEGAL DATA ADDRESS exception code.

- If the Write Multiple Registers function code is used to write a multiple-register data type, the start register address must be of the first register in the field and the number of registers must include all registers in the field. If one or both parameters are invalid, the device will return an exception response with the ILLEGAL DATA ADDRESS exception code.

- The Write Multiple Registers function code can be used to write data to multiple registers of the same or different data types. In this case, the starting register address must be of either a single-register data type or the first register in a multiple-register data type. The last register written must be either a single-register data type or the last register in a multiple-register data type. Attempting to start or end the write within a multiple-register data type will result in an exception response with the ILLEGAL DATA ADDRESS exception code.

- Some registers are designated as read-only (R versus R/W in the register map). Attempting to write data to a read-only register will result in an exception response with the READ-ONLY REGISTER exception code.

- If an attempt is made to write an out-of-range value to a register, an exception response with the ILLEGAL WRITE VALUE exception code will be returned.
- If the function returns an exception response with an exception code of ILLEGAL DATA ADDRESS or READ-ONLY REGISTER, no data was written to any of the registers.
- If the function returns an exception response with an exception code of ILLEGAL WRITE VALUE or DEVICE FAILURE, data was written to registers up to but not including the register that caused the exception. No data was written to registers after the register that caused the exception. There is no provision in the exception response to isolate the problem register.

The response to a write multiple registers command is typically returned in less than 100 milliseconds.

### 8.5.4 Write Single Register

This function code is used to write the contents of a single register. Only single-register data types can be written with this function. Writing to a multiple-register data type requires the Write Multiple Registers function code.

| **Command** | | | |
|---|---|---|---|
| Byte Offset | Field Description | Type | Value |
| 0 | Device Address | BYTE | 1-247 |
| 1 | Function Code | BYTE | 6 (0x06) |
| 2 | Register Address | USHORT | 0-65535 (0xFFFF) |
| 4 | Register Data | USHORT | 0-65535 (0xFFFF) |

| **Response** | | | |
|---|---|---|---|
| Byte Offset | Field Description | Type | Value |
| 0 | Device Address | BYTE | 1-247 |
| 1 | Function Code | BYTE | 6 (0x06) |
| 2 | Register Address | USHORT | 0-65535 (0xFFFF) |
| 4 | Register Data | USHORT | 0-65535 (0xFFFF) |

The register address is one less than the register number.

- An attempt to write a single register within a multiple-register data type will result in an exception response with the ILLEGAL DATA ADDRESS exception code.
- Some registers are designated as read-only (R versus R/W in the register map). Attempting to write data to a read-only register will result in an exception response with the READ-ONLY REGISTER exception code.
- If an attempt is made to write an out-of-range value to a register, an exception response with the ILLEGAL WRITE VALUE exception code will be returned.

The response to a write single register command is typically returned in less than 100 milliseconds.

### 8.5.5 Exception Response

When the device encounters an error condition in a command from the Master, it will reply to the command with the standard MODBUS exception response.

| Exception Response | | | |
|---|---|---|---|
| Byte Offset | Field Description | Type | Value |
| 0 | Device Address | BYTE | 1-247 |
| 1 | Function Code + 128 (0x80) | BYTE | 128-255 (0x80-0xFF) |
| 2 | Exception Code | BYTE | 0-255 (0xFF) |

#### *8.5.5.1 Standard Exception Codes*

The following exception codes are defined by the MODBUS protocol.

| Exception Code | Name | Description |
|---|---|---|
| 1 (0x01) | ILLEGAL FUNCTION | The function code received in the command is not valid for the device. |
| 2 (0X02) | ILLEGAL DATA ADDRESS | The register address or the register range received in the command is not valid. |
| 3 (0x03) | ILLEGAL DATA VALUE | A value contained in the command is not valid, such as a mismatch between the number of registers and the byte count. This exception indicates an error in the structure of a command. |
| 4 (0x04) | DEVICE FAILURE | An error occurred in the device while attempting to perform the command. |

#### 8.5.5.2 Custom Exception Codes

The following exception codes are device-specific extensions to the standard codes to help speed command/response troubleshooting.

| Exception Code | Name | Description |
|---|---|---|
| 128 (0x80) | READ-ONLY REGISTER | The command attempted to write data to a read-only register. |
| 129 (0x81) | WRITE-ONLY REGISTER | The command attempted to read data from a write-only register. |
| 130 (0x82) | ILLEGAL WRITE VALUE | The command attempted to write an illegal or out of bounds value. |
| 131 (0x83) | CORRUPT DATA RECORD | The requested data record is corrupted. |
| 132 (0x84) | DOWNLOAD END | End of fast data log download. |
| 133 (0x85) | SECURITY MODE | The requested function is not available in security mode. |