



# **Power Xpert® Gateway 400 Modbus Communications and Register Mapping**

## ***Introduction***

Eaton's Power Xpert Gateway 400 (PXG 400) can connect to and communicate with a large number of different devices. These can include, for example, meters, transfer switches, and starters. Modbus TCP register maps for the current collection of supported devices are included on this CD. Each register map (.pdf file) contains the list of device data that can be monitored by a Modbus TCP connection to the PXG 400, including voltages, currents and status. The register map files are contained in the Modbus Register Map folder. For example, the register map file for the ATC400 can be found in the MOD\_ATC400.pdf file.

## ***Locating Modbus TCP Data Registers***

If the device is connected to the PXG 400 Modbus RTU/485 interface, the data can be obtained by reading the respective Modbus registers over Modbus TCP. The available Modbus registers are listed in the register map document for the specific device and their locations can be found in the "Modbus Register" column.

Continuing with the ATC400 example, the files show that the Vendor Name (description) is contained in Modbus register 1001 for 32 registers (Register count). Also the Units column shows that there are no units (volts, current, etc) associated with the Vendor Name registers. The Type ID column indicates that the data is in ASCII format. This is two ASCII bytes per register for a total of 64 bytes.

## ***Configuring the PXG 400 for Modbus***

The PXG 400 has two interfaces for attaching devices – RTU/485 and INCOM. Each interface can support up to 32 devices. The attached devices on the Modbus RTU/485 interface must have unique individual addresses. Up to 32 devices can be attached to this interface. Refer to the specific device operator's instructions for entering a Modbus address into the device. The device address can be any number from 1 to 32.

Ensure that the address is also entered into the PXG 400. This can be accomplished from the web configuration page. When the addresses are entered, a set of corresponding addresses are automatically created and cached in the PXG 400 and have addresses from 33 to 64. These are identified in the "Modbus TCP Unit ID" field on the PXG 400 device list configuration page. Refer to the PXG 400 User's Guide for details on this operation.



## ***Reading the Modbus TCP Registers***

The registers can be read by addressing the PXG 400's IP address and port (default port is 502<sup>1</sup>) using a Modbus TCP analyzer or software application. In general, set the analyzer program for RTU protocol and MODICON as the mod type. If you have the option, select 0-based addressing.

When polling the device use the Modbus TCP Unit ID field address specified in the device list configuration. The number of registers should correspond to the register count on the spread sheet. Use function code 04 for reading the cached values in the PXG 400.

## ***Interpreting the Individual Modbus TCP Register Maps***

The register maps describe the characteristics of individual registers for a specific device connected to the PXG 400.

### **Modbus Register**

This is simply the number of the starting register for the Modbus data. When the data spans several registers begin reading the data at this register.

### **Description**

This is the basic name of the data contained in the register.

### **Register Count**

This is the number of registers that contain the data. Begin reading the registers at the addresses specified in the Modbus Register column.

### **Type ID (Data Formats)**

Modbus analyzer or application programs usually provide an option for viewing various data types. Refer to your Modbus program documentation for detailed information on viewing data options.

Not all data formats are available in all devices.

A register is 16 bits (two bytes). The PXG 400 supports the following data types:

- **STRING or ASCII**– A string of ASCII characters (two per register). Consult the Size parameter to find the string's length for a given entry (it is a multiple of two so that entries are register-aligned).
- **FLOAT** – A 32-bit IEEE754 floating point number. FLOATs are always two registers.

---

<sup>1</sup> The default port 502 can be changed via the PXG 400's web page.



- **UINT** – An unsigned integer. **UINT16** - a 16 bit unsigned integer. Consult the Size parameter to find the integer's size for a given entry (it is a multiple of two so that entries are register-aligned).
- **TIME or DATE** – The Time type (*ymdhms*) consists of six bytes (three registers) specifying the year, month, day, hour, minutes, and seconds. The data is stored in hexadecimal format and is best viewed by using the binary option and converting each byte to decimal.
- **BOOL** – A binary (Boolean) value of 0 or 1. BOOL is usually assigned to discrete input alarms.

### **Data Addressing**

If a data type spans multiple registers (such as a FLOAT), lower addressed registers map to higher-order parts of the value. Within each register, data is in the most significant bit first (MSB first) format.

*NOTE A Modbus register contains two bytes. Therefore, the number of registers containing the information can be obtained by dividing the bytes by two. For example, the vendor name can be obtained by reading 32 Modbus registers starting at register 1032.*

*NOTE If a register is not supported in a particular device, an exception response is returned.*

## **UNITS**

This column is just the description of the value in the register. If the value is numeric the units usually describe volts, amps, or power. Most string values have no units.

### ***A note on devices that support the Status register***

Many of the supported devices return a numeric status that depends on the specific device. But generally, the status follows the following format:

- 1 means unknown status
- 0 means off or open
- 1 means on or closed
- 2 means fault or tripped
- 3 means an alarm condition exists