ELKOR Technologies Inc.

6 Bainard Street
London, Ontario N6P 1A8 CANADA
tel(519)652-9959 fax(519)652-1057

Application Note AN0303

ModBus RTU Troubleshooting for Elkor ETA3.RTU and WattsOn

Associated Product: ETA3.RTU & WattsOn

Summary

Modbus communications allows a large amount of data to be collected using only two wires. By using this digital communications standard, available on most PLCs, HMIs, SCADA systems and PCs, it saves the need for complex wiring and costly analog conversion peripherals. Although the initial setup is slightly more involved, once configured, digital communications offers a wide variety of advantages.

Background

Both the Elkor WattsOn and ETA3.RTU devices offer Modbus RTU communications over the RS-485 bus. It is important to understand that "RS-485" defines the hardware layer of the communications (the signal voltage levels and wiring type), while "Modbus RTU" defines the language used over the RS-485 bus. The Modbus protocol may be implemented over Ethernet, RS-232, RS-422, etc, however the WattsOn and ETA3.RTU natively communicate over RS-485, and will require an interface converter to communicate over other network types. For example, most PCs have only RS-232 Serial connections, therefore when communicating with a PC, an RS-232-to-485 converter must be employed

Modbus is a query/response protocol. This means that a Modbus network requires one master, and may contain many slaves. Each slave must be assigned a unique address. Every slave on the network will listen to the commands from the master but will only reply if the packet contains its address.

A Modbus query packet is composed of the following:

Device Address (1 byte)	Function (1 byte)	Starting Register (2 bytes)	# of Registers Required (2 bytes)	CRC (2 bytes)

The slave response will be composed of the following:

Device Address (1 byte)	Function (1 byte)	Byte Count	Integer Data (2 bytes per Register)	CRC (2 bytes)

Byte Count: number of bytes received (2 times the total number of registers requested).

The WattsOn and ETA3.RTU use "function 3" data registers. Depending on the Modbus master, the register referencing method may be different. It is best to read the users manual of the master device in question to obtain an understanding on how the manufacturer incorporated the syntax and register mapping.

To fully define a register location, it is necessary to specify the function which it resides in as well as the offset of the register within that function. This corresponds to the "function" and "starting register" bytes in the query packet. Many PLCs will "map" their own data memory registers to the Slave data memory, in which case the definitions may change. A common method is to refer to the function/offset locations as an absolute number. In many cases this number is a "V" memory location such as "V30001". The function "4" registers are often mapped on to "30000" series registers, while the function "3" registers are mapped to "40000" series registers. The following table gives some examples of common function/offset mappings.

Register/Offset	"V" memory mapping
Function 4, Offset 0x00	V30001
Function 4, Offset 0x0A	V30011
Function 3, Offset 0x05	V40006

The offsets are given as hex digits. Notice that when using the register/offset notation "0" defines the first register, while the memory mapping noation defines the first register at "1".

This example applies to only some of the PLC devices available, however it is recommended that the programmer obtains a thorough understanding of the programming, syntax and notations used by the PLC with which he/she is intending to implement.

Troubleshooting Steps

Step 1) Make sure your device either has a built in RS-485 port or you are using the proper interface converter (ie: RS-232/RS-485). Elkor recommends that an interface converter with "automatic send data" control be used. This feature eliminates the need for RTS/CTS handshaking by detecting when the master is sending data and enables/disables the RS-485 output automatically. This eliminates one level of complexity, and allows the use of software that may not have a precise method of controlling the RTS/CTS line.

Step 2) Make sure your device is set up for the correct COM parameters. Both the WattsOn and the ETA3.RTU are configured for operation at 9600 baud, No parity, 8 Data bits and 1 stop bit (9600, N, 8, 1).

Step 3) Make sure your device is properly connected to the WattsOn/ETA3.RTU. For short runs, only two wires (+/-) are required, but proper grounding should be observed over longer runs/noisy environments. Many times devices will specify the terminal connections as A/B instead of +/-. Usually the case is that A is connected to – and B is connected to +, however there may be an instance where this is reversed. Check the user manual of the PLC/converter for details.

Step 4) Make sure that each slave is assigned a unique address, and that the address of the unit in question is identical to the configuration in the master. The address configuration differs from device to device. Check the device manual, and Elkor's application note AN0302 – RS485 Addressing.

Step 5) Make sure your outgoing packet requests contain the correct data/syntax. Ideally some means to view the serial data (there are some freeware programs that allow viewing of incoming/outgoing serial data). A sample query/response packet is shown below (each set of brackets contains one hex digit byte). This sample is from a query/response to an ETA3.RTU, reading all 5 registers. The values of the data registers were all "0".

```
Query -> (01)(03)(00)(00)(00)(05)(85)(C9)
Response <- (01)(03)(0A)(00)(64)(00)(00)(00)(00)(00)(00)(00)(00)(BE)(74)
```

Sources

Modbus Organization:
http://www.modbus.org

Freeware Modbus Master Program (for testing purposes):
http://www.modbus.pl/downloads.htm

Modbus Protocol and CRC Generation:
http://www.modicon.com/techpubs/toc7.html

Serial Port Monitor:
http://www.sysinternals.com/ntw2k/freeware/portmon.shtml