

APPLICATION NOTE:

NOT ALL MODBUS WERE BORN EQUAL



Nowadays practically any industrial device supports the well established Modbus communication protocol. It is a "must have" for any specification. When it comes to implementation, integrators find huge differences between products that are not necessarily reflected in the product's data sheet but severely affect the use of the product. This document will emphasize some of the important functionality that will help ensuring that the Modbus project will be successful.

There are several factors that affect the quality of the Modbus communication:

- Availability of information about the protocol, including file structure and more
- Maximum baud rate (and the reliability at that baud rate)
- Communication response time
- Multi address support
- Data compression

Is It Really an Open Protocol?

The protocol includes various formats, such as integer and floating point. This means that the protocol description includes table of addresses and parameters with their format.

The amount of information that different equipment manufacturers provide in this table varies. While some provide the full details of the device, others hide some of the information. In this way they can force their customers to use their own software tools for complete functionality. This cause to the customers increased cost and reduced performance, as other software suppliers may have better functionality and lower price.

It is recommended to verify that the protocol description includes at least the following: real time data, average data, waveform data, harmonic information (totals and individual), configuration information, configuration modification (write commands), event log, data log and harmonic log (including detailed description how each log is organized). By following this checklist, the user will know that he has the choice of software provider and that his software team can maximize the benefits from his assets.

Baud Rate

Originally Modbus worked in baud rates of 4800, 9600 and 19200 kbps (kilo bit per second). As all computers today supports baud rate of up to 115,200 Kbps, manufacturers adapted their products to the high speed serial port. In short communication cables it is possible to utilize the higher baud rate to reduce the length of the data packet (which equal to the number of bytes, multiply by 10 or 11 divided by the baud rate). The benefit of shorter packet length is higher commutation update rate.

Communication Response Time

When a master sends a request to the device, the device needs to parse the packet, look for the required parameters, build the return packet and send it back. It takes time. This time is very important for the communication throughput and varies from single network cycle (20 ms) in quality meters to 1 second in low cost products. As in many times there are few meters on every RS-485 communication link, it multiplies the frequency that new information can be retrieved.

Multi Address Support

Each Modbus “read packet” includes address and length - the device returns the continuous values between the address and the (address+length-1). It is common that the master requires parameters that are not continuous. In this case the master has to send another read packet for each none continuous address which causes two problems:

- Longer communication cycle due to few read commands for single device
- Inconsistent reading - since the data is from another cycle, the values may have changes, which means no calculation between the parameters is allowed. For example, you read voltage, current and power factor and would like to calculate active power ($kW = V \times I \times PF$) but the current and PF are not from the same time and the result will differ from active power. The solution is to read additional parameters and longer communication cycle.

Some manufacturers offer dynamic Modbus mapping. It means that regardless of the static Modbus address map of the required parameters, it will be possible to read all the required at single read command. This function solves both issues described above. An example of dynamic mapping is available on the following pages.

Data Compression

In order to provide wide range of values for both low voltage and high voltage applications, parameters are usually stored in 64 bits - as long integer or a floating point. However, since the accuracy of high end meters is typically 0.1%, the actual value can be represented in 16 bit or less. Some manufacturers offer various types of data compression, such as RLE compression or aligning the data into 16 bit integers. This reduces the packet length by half, resulting with twice faster communication.

SATEC Modbus Communication Protocol

All SATEC meters include the industry's most advanced Modbus protocol implementation that includes:

- Complete open protocol - all information, including data logs, configuration and more, is comprehensively explained in the manual
- Baud rate of up to 115,200 bps, which is the highest baud rate that is supported by personal computers, to maximize communication speed
- Communication response time of less than 20 ms ensures real time communication
- Dynamic Modbus mapping feature allows configuring of up to 120 parameters for continuous reading (the maximum for single Modbus read command - see details below)
- Data alignment of 32 bit parameters into 16 bit to reduce communication type by half.

SATEC Modbus Mapping

SATEC allows mapping of any one of the thousands of parameters in the protocol into dynamic map address. The first 120 addresses are dynamically mapped, while the following 120 addresses include the mapping information. For example, the address of L1 voltage is 256 and the address of L1 current is 259. By writing the value 256 to address 120 and 259 to address 121, a dynamic mapping is created. The result is that address 0 has the voltage and address 1 has the current, allowing reading of both of them with only a single Modbus read command. The dynamic mapping can be configured by writing these addresses or using the PAS software (the free software that is provided for all SATEC devices).

SATEC Modbus Mapping Example

The following table shows the address and parameters without using the dynamic mapping:

Address	Parameter	Value (decoded)
256	L1 Voltage	400
259	L1 Current	20

SATEC Modbus Mapping

The following table shows the mapping of parameters (written once):

Address	Parameter	Value
120	Parameter #0 Map	256
121	Parameter #1 Map	259

SATEC Dynamic Modbus Content

The following table shows the values in continuous address map that is constantly updated:

Address	Meaning	Value (decoded)
0	L1 Voltage	400
1	L1 Current	20

Example

A SCADA system needs to read 10 parameters from each meter (e.g., voltages, currents, powers and system statuses). Since usually the mapping is per phase and then average, it requires 10 separated read cycles with static address mapping. In our example we shall assume that each RS-485 cable has 16 devices connected to it (50% of the full capacity).

Case 1: a device with communication response time of 750 ms that does not support dynamic mapping. Even when ignoring the communication speed and data compression, the minimum read cycle of all the devices will be 0.75 seconds, multiplied by 10 parameters and 16 devices, equals 120 seconds – a full two minutes!

Case 2: SATEC device with dynamic mapping, 20 ms communication response time, 16 bit data adjustment and fast baud rate - packet communication will be $10 \times 2 + 6 = 26$ bytes (data length plus header and CRC) which is 2 ms in 115,200 baud rate. Allowing 1ms for the request from the master to the slave, will give a total read cycle of 23 ms for complete reading from one device or 368 ms from 16 devices (less than half of second).

This means that in case 1 the communication is not even close to real time, cannot make internal calculation or virtual devices (a meter which is calculation of two or more physical meters) while case 2 provides real time operation as well as maximizing the number of units on single RS-485 line, which saves money and other resources.

Summary

Different products have different Modbus capabilities, not necessary indicated on their specifications. A thorough analysis of them allows us to select between a slow response basic communication only and a real time solution that can achieve the goal of supplying reliable data in a timely manner to our software

