Modbus and ION® Technology

Modicon Modbus is a communications protocol widely used in process control industries such as manufacturing. ACCESS meters are compatible with Modbus networks as both slaves and masters, and can communicate easily with WinPM.Net software or other third-party software.

This technical note describes the Modbus protocol in general, how to use ACCESS meters and software with Modbus devices, and how ACCESS meters fit into a Modbus network.

In This Document

- **Introduction** .................................................. 2
  - The Modbus Message ........................................ 3
  - Modbus Maps for ACCESS Meters ......................... 5

- **ION and Modbus in Networks** ................................. 6
  - Scenario 1: ACCESS Meters in a Modbus Network ........ 6
  - Scenario 2: Modbus Devices in a WinPM.Net Network .... 7
  - Scenario 3: ACCESS Meter as Modbus Master ............. 9
  - Scenario 4: EtherGate and Modbus RTU .................. 10

- **ACCESS Meters as a Modbus Master** ...................... 11
  - ION Modules for Modbus Mastering ....................... 11
  - Reading Data Using the Modbus Master Device Module ... 11
  - Reading Data Using the Modbus Import Module ........... 12
  - Writing Data Using the Modbus Export Module .......... 13

- **ACCESS Meters as a Modbus Slave** ....................... 14
  - Using the ACCESS Meter as a Modbus Slave .............. 14
  - Modbus Slave Module Settings .......................... 14

Additional Information

- **ION Reference**
- **ION/Modbus Register Map** document that applies to your meter
- www.sea.siemens.com (ACCESS meter and Modbus/WinPM.Net software documentation)
- www.ioreference.com (devices that support Enhanced Modbus/TCP)
- www.modicon.com (Modbus/TCP standard, Modicon Modbus Serial Communications Protocol documentation)
- www.modbus.org (SEMI E54-0997 standard)
Introduction

Modbus is a protocol used in industrial manufacturing. It was created to transfer control data between controllers and sensors using RS-232 serial ports. Today, the Modbus protocol is widely used and supported on serial (RS-232 and RS-485) and Ethernet connections.

Modbus is a master/slave protocol where the master initiates transactions and the slaves respond with the requested data or action. Modbus masters are usually software programs, such as Distributed Control Systems (DCS). However, masters can also be devices such as remote terminal units (RTU), programmable logic controllers (PLC) or ACCESS meters. Modbus slaves are devices such as PLCs, I/O monitoring devices, relays, software and ACCESS meters.

ACCESS meters, when used as Modbus master or slave devices, provide sensing for your Modbus network so you can quickly view and respond to power data and equipment status information.

Modbus RTU and Modbus ASCII

The Modbus protocol has two basic forms. The RTU or Binary form and the ASCII form. The ASCII form transmits all bytes using the ASCII character set (‘0-9’ and ‘A-F’). The RTU form transmits all bytes in a binary format making the protocol faster and more efficient. Both forms are serial (RS-232/RS-485) protocols.

Modbus Plus

Modbus Plus is a proprietary Modicon protocol used in industrial networking systems. It uses token-passing peer-to-peer communications at a data transfer rate of one megabit per second (high-speed passing of groups of bits within a layer). The network medium is shielded twisted-pair cable.

Modbus/TCP

Modbus/TCP is the open Modbus protocol variant (formerly called MBAP). It defines the packet structure and connection port for the industry standard TCP/IP protocol. The structure of Modbus/TCP is very similar to the Modbus RTU packet except that it has an extra six-byte header and does not use the cyclic redundancy check (CRC). Some ION firmware supports Modbus/TCP for direct communications with the meter.

Modbus/TCP retains the Modbus RTU limit of 256 bytes to a packet. It is suggested that higher throughput is possible if this limitation is removed. This variant is called Enhanced Modbus/TCP but, so far, few devices have moved to support it.

Modbus RTU over Ethernet

Your ACCESS meter can use Modbus RTU to transport direct to a Modbus device over Ethernet.
The Modbus Message

There are a few variants of the Modbus protocol for use in different networks and situations but all use the same message structure. The most common Modbus variants are Modbus RTU, Modbus ASCII, Modbus Plus and Modbus/TCP. The Modbus message that a master sends to a slave contains these main pieces of information:

<table>
<thead>
<tr>
<th>Type of Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slave Address</td>
<td>Unit ID (from 1 to 247)</td>
</tr>
<tr>
<td>Function Code</td>
<td>Read from or write to registers</td>
</tr>
<tr>
<td>Data</td>
<td>16-bit words of various formats</td>
</tr>
<tr>
<td>Checksum</td>
<td>CRC-cyclical redundancy check</td>
</tr>
</tbody>
</table>

ACCESS meters are preconfigured with a unit ID that you can change from the front panel of the meter or via software. This unit ID serves as the slave address that the master uses to determine which slave to communicate with.

The function code used is based on the variable type that a particular ACCESS meter supports, and whether the ACCESS meter is acting as a master or a slave.

Variable Types
The basic Modbus message building blocks are called variable types. Variable types group data into address ranges that perform certain functions (using function codes). For example, Coil registers have an address range from 00001 to 09999 and the data that makes up the message can be read or written to (Function 1, 5 and 15).

There are four main classes of Modbus data, or variable types, that support different function codes. Coil and input registers are single-bit registers used to indicate ON or OFF conditions. Input and holding registers are 16-bit registers used to store and retrieve data.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Supported Function Code</th>
<th>Address Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coils</td>
<td>Digital bits that can be read and written to</td>
<td>Function 1, 5, 15</td>
<td>00001 to 09999</td>
</tr>
<tr>
<td>Input Status</td>
<td>Digital bits that can be read</td>
<td>Function 2</td>
<td>10001 to 19999</td>
</tr>
<tr>
<td>Input Registers</td>
<td>16-bit integers that can be read</td>
<td>Function 4</td>
<td>30001 to 39999</td>
</tr>
<tr>
<td>Holding Registers</td>
<td>16-bit integers that can be read and written to</td>
<td>Function 3, 6 and 16</td>
<td>40001 to 49999</td>
</tr>
</tbody>
</table>
Function Codes

ACCESS meters (acting as master or slave) use ION modules (Modbus Master Device, Modbus Master Mapping, Modbus Import, Modbus Export and Modbus Slave modules) to support the following function codes:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>ION master or slave function</th>
<th>ION module that supports the function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function 1</td>
<td>Read Coil Status</td>
<td>Master</td>
<td>Modbus Import, Modbus Master Device, and Modbus Master Map¹</td>
</tr>
<tr>
<td>Function 2</td>
<td>Read Input Status</td>
<td>Master</td>
<td>Modbus Import, Modbus Master Device, and Modbus Master Map¹</td>
</tr>
<tr>
<td>Function 3</td>
<td>Read Holding Registers</td>
<td>Master and slave</td>
<td>Modbus Import, Modbus Master Device, and Modbus Master Map¹</td>
</tr>
<tr>
<td>Function 4</td>
<td>Read Input Registers</td>
<td>Master</td>
<td>Modbus Import, Modbus Master Device, and Modbus Master Map¹</td>
</tr>
<tr>
<td>Function 5</td>
<td>Write to a Coil</td>
<td>Master</td>
<td>Modbus Export</td>
</tr>
<tr>
<td>Function 6</td>
<td>Write to a Holding Register</td>
<td>Master</td>
<td>Modbus Export</td>
</tr>
<tr>
<td>Function 15</td>
<td>Write to Multiple Coils</td>
<td>Master</td>
<td>Modbus Export</td>
</tr>
<tr>
<td>Function 16</td>
<td>Write to Multiple Holding Registers</td>
<td>Master and slave</td>
<td>Modbus Export and Modbus Slave</td>
</tr>
</tbody>
</table>

¹ The Modbus Master Map module and the Modbus Master Device module work together; you need both configured properly for this feature to work.

The maximum number of Modbus registers the module can read at once (per request) depends on the Modbus format and data register type. Modbus supports a combination of 16-bit data types. These data formats use one or two 16-bit words to encode signed/unsigned, or integer/floating point numbers:

<table>
<thead>
<tr>
<th>Format</th>
<th>Type</th>
<th># of Modbus registers used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unsigned 16-bit</td>
<td>Integer</td>
<td>1</td>
</tr>
<tr>
<td>Signed 16-bit</td>
<td>Integer</td>
<td>1</td>
</tr>
<tr>
<td>Unsigned 32-bit</td>
<td>Integer</td>
<td>2</td>
</tr>
<tr>
<td>Signed 32-bit</td>
<td>Integer</td>
<td>2</td>
</tr>
<tr>
<td>Unsigned 32-bit M10k</td>
<td>Integer</td>
<td>2</td>
</tr>
<tr>
<td>Signed 32-bit M10k</td>
<td>Integer</td>
<td>2</td>
</tr>
<tr>
<td>IEEE Float¹</td>
<td>Floating point</td>
<td>2</td>
</tr>
<tr>
<td>Packed Boolean</td>
<td>Integer</td>
<td>1</td>
</tr>
</tbody>
</table>

¹ ACCESS meters acting as Modbus masters support two versions of IEEE Float with different word orders: high/low = Big Endian, low/high = Little Endian. Big Endian/Little Endian is supported for all 32-bit formats by ACCESS meters acting as Modbus masters.
Modbus Maps for ACCESS Meters

There are two types of Modbus mappings available on ACCESS meters: one is a fixed map and the second is a flexible map. The fixed map is useful for configuring parameters on the ACCESS meter (slave), or allowing third-party software to access other ION modules. For more information on the Modbus fixed map see your meter’s user guide.

When a master sends a request to a fixed map on the ACCESS meter (slave), the value is written to the corresponding mapped register on the ACCESS meter. For example, the Modbus master sends “5” to register “44590” which changes the baud rate on COM2 to 19200—the baud rate is an enumerated parameter and the fifth enumeration is 19200.

ACCESS meters also provide a flexible map that you can use for reading data. The flexible Modbus map lets you read various ION registers with Modbus master devices by correlating the Modbus register number with the address of the ION register (on the slave) you want to read. When a Modbus master requests to read a register, the corresponding ION register data is returned.

For more information on the Modbus flexible map, see your meter’s Modbus map document.

NOTE

If you have Advanced security enabled (ION 8000 series meter), you must configure the Modbus Map Access register to allow write access for specific users. See the Security Options module description in the ION Reference. If you have Standard or Advanced security enabled on the meter, you can only read Modbus information from a slave module. You must disable Standard security to write data to a slave module. See the ION System Security technical note for more information.
ION and Modbus in Networks

An ACCESS meter acting as a slave can make real-time data (and in some cases, historical data) available to Modbus masters using third-party software. Modbus masters can also write data to ACCESS meters to change device configurations or initialize control actions.

You can use WinPM.Net software to collect data from and send data to Modbus devices directly. Similarly, the ACCESS meter with Modbus master capability can collect data from a Modbus network and send it to WinPM.Net software for display and analysis. Using WinPM.Net software, you can configure these meters to read data from and write data to Modbus slave devices.

The following section describes four different scenarios where ACCESS meters and WinPM.Net software integrate with Modbus devices and networks.

Scenario 1: ACCESS Meters in a Modbus Network

If you have third-party Modbus software you can communicate with ACCESS meters over Ethernet and serial connections. Most ACCESS meters with Ethernet capability support Modbus/TCP on TCP/IP port 502. There is no meter configuration required for a third-party master to communicate with an ACCESS meter over Ethernet.
You can also make a serial connection to an ACCESS meter using third-party software. Set the communication port on the meter using the front panel of the meter.

**Setting the serial communication from the ACCESS meter to third-party software**
1. Using the front panel of the ACCESS meter, navigate to the communications port you want to use to connect to third-party software.
2. Set the protocol for this port to “Modbus RTU” and ensure the baud rate, unit ID, serial port and RS-232 or RS-485 settings are correct.

**Scenario 2: Modbus Devices in an WinPM.Net Network**

You can add a Modbus slave device to your WinPM.Net network using RS-232, RS-485 or Ethernet (shown below) connections.

Use the WinPM.Net Modbus Device Importer utility to configure your Modbus device (mapping information) and add it as a Device Type to the Network Configuration database (NOM).

**Note**

See the Modbus Device Importer technical note for more information on adding Modbus device types. You can also contact Power Measurement about pre-made Device Integration Packages, which will add a device directly into your NOM.

Once the device type is added, you can see your Modbus device in a drop-down menu in the Management Console (see below).
Adding a Modbus device to the WinPM.Net network

1. Launch the Management Console. Click the Devices button on the System Setup pane.

2. Right-click in the display window and choose Add>New Serial Device or Add>New Ethernet Device depending on your type of connection.

3. Use the drop-down menus or type the required information in the fields. Your Modbus device appears in the Device Type drop-down menu.
Scenario 3: ACCESS Meter as Modbus Master

As a master, the ACCESS meter can aggregate data from multiple devices and then pass it to WinPM.Net software for analysis. The ACCESS meter as a master also gives you control of devices on a Modbus network with its write capability.

If the ACCESS meter you are using is acting as a master, there are two communication links to consider when you communicate from a Modbus network (or device) to WinPM.Net software: first, there is a link between the hardware communications port on the ACCESS meter to the Modbus network. This link is a one-to-one relationship between the communications port on the meter (COM1, COM2, etc.) and the wiring that connects the meter to a Modbus network (RS-232 or RS-485).

The second communications link occurs between the meter and WinPM.Net software. For Ethernet communications, the meter allows multiple TCP/IP “virtual ports” (i.e. 502, 7700, etc.) to connect to a single hardware port. On the meter, TCP/IP virtual ports indicate to the software what type of data is being communicated.

Configuring the ACCESS meter to act as Modbus Master

1. Using the front panel of the ACCESS meter, navigate to the communications port you want to use to connect to the Modbus Slave devices.

2. Set the protocol for this port to “Modbus Master” and ensure the baud rate, unit ID, serial port and RS-232 or RS-485 settings are appropriate.

The image below shows how WinPM.Net software and an ACCESS meter (communicating over Ethernet) can integrate with a serial Modbus network:
Scenario 4: EtherGate and Modbus RTU

EtherGate™ gateway capability is built into all ACCESS meters with Ethernet cards (except the 9300 meter). With EtherGate, users can communicate over a network with any ACCESS or Modbus-compatible device connected to a RS-485 loop on the EtherGate meter.

The EtherGate meter acts as an intermediary between the RS-485 devices (Modbus slaves) and the Modbus master, allowing the Modbus master to send and receive data over TCP/IP. In WinPM.Net, users can create an EtherGate site and communicate with a series of ACCESS devices on the RS-485 loop. Or, with the WinPM.Net VIP’s Modbus master capability users can communicate with a series of Modbus slave devices on another RS-485 loop.
ACCESS Meters as a Modbus Master

ACCESS meters that have Modbus master capability let you write data to (export), and read data from (import) Modbus slave devices. The ACCESS meter (acting as a Modbus master) can read data on a Modbus network and act as a point of data aggregation. The data can be processed by the meter and sent out using other communications methods (email, WinPM.Net software, etc.). The meter can also send control commands or data directly to other devices on a Modbus network.

NOTE

Only one meter on a Modbus network can have the communications channel set to the Modbus Master protocol.

ION Modules for Modbus Mastering

Several ION modules work together to create Modbus mastering functionality on the meter. Your meter will have some or all of these modules, depending on the model and firmware version. See the ION Reference for more information on these and other ION modules:

- **Modbus Master Device Module (currently only available on 9510 RTU meters)**: provides read functionality when used in conjunction with the Modbus Master Map module. This imported data can be used by other ION modules.

- **Modbus Master Map Module (currently only available on 9510 RTU meters)**: provides a common place to hold setup information for decoding a Modbus response. This information can be used by multiple Modbus Master Device modules.

- **Modbus Export Module**: provides write functionality.

- **Modbus Import Module**: provides read functionality. This data can then be used by other ION modules.

NOTE

There are many Modbus Import and Modbus Export modules available. The modules behave like Modbus controllers. However, each module can communicate to only one Modbus slave device.

Reading Data Using the Modbus Master Device Module

The Modbus Master Device module reads data from a Modbus slave, which can be an ACCESS meter or third-party device. A Modbus Master Device module uses settings from a Modbus Master Map module setup register to specify the Modbus format, scaling, and base address settings.

**Reading from Modbus devices**

1. Launch Designer (ensure that Options>Show Toolbox is checked).

   Drag out a new grouping object from the toolbox, name it appropriately and double-click on your new grouping object. A blank workspace displays where you can keep your master configuration.
2. Drag out a new Modbus Master Device module to use with your Modbus slave device. Right-click the module and configure the following registers:

<table>
<thead>
<tr>
<th>Register Name</th>
<th>Register Type</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable</td>
<td>Boolean Input</td>
<td>Set this register to ON.</td>
<td>Set to OFF to make the outputs N/A.</td>
</tr>
<tr>
<td>Read Now</td>
<td>Pulse Input</td>
<td>Link this register to set the module to Read Now mode.</td>
<td>If this is unlinked, module operates in Polling Mode and will request every second.</td>
</tr>
<tr>
<td>Slave Addr</td>
<td>Numeric Setup</td>
<td>Enter the address of the Modbus slave device (1-247 accepted).</td>
<td>Make sure each device has a unique address.</td>
</tr>
<tr>
<td>Device Type</td>
<td>String Setup</td>
<td>Enter the type of Modbus slave device. For example, 6200.</td>
<td>This register value must match the corresponding Device Type register in the Modbus Master Map module, for the map to be available to this module.</td>
</tr>
<tr>
<td>Slave Name</td>
<td>String Setup</td>
<td>Enter the name of the Modbus slave device. For example, Submeter1.</td>
<td>This value will be appended to the output registers.</td>
</tr>
<tr>
<td>Outputs 1-32</td>
<td>Numeric Output</td>
<td>Link these to a variety of ION modules. For example, a Data Recorder.</td>
<td>The output value is in the format: &lt;label&gt;@&lt;slave name&gt;[value]</td>
</tr>
</tbody>
</table>

3. Drag out a new Modbus Master Map module. Right-click the module and configure the following registers:

<table>
<thead>
<tr>
<th>Register Name</th>
<th>Register Type</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Type</td>
<td>String Setup</td>
<td>Enter the type of Modbus slave device. For example, 6200.</td>
<td>This register value must match the corresponding Device Type register in the Modbus Master Device module, for the map to be available to the module.</td>
</tr>
<tr>
<td>Device Map</td>
<td>String Setup</td>
<td>Enter the mapping information for this device. See the module description in the ION Reference for details.</td>
<td>This map (label, address, format and scaling information) can be used by multiple Modbus Master Device modules.</td>
</tr>
</tbody>
</table>

4. Link the output registers to desired modules (such as Data Recorders). The output values from the Modbus Master Device module are appended with a label and device name. The Data Recorder module understands these names, and will populate the ION database using them (e.g. kW@Submeter1).

5. In Vista, create and link an object that best displays your configuration.

**Reading Data Using the Modbus Import Module**

The Modbus Import module reads data from a Modbus slave, which can be an ACCESS meter or third-party device. A Modbus Import module’s setup registers specify the Modbus format, scaling, and base address settings.

**Reading from Modbus devices**

1. Launch Designer (ensure that Options>Show Toolbox is checked).

   Drag out a new grouping object from the toolbox, name it appropriately and double-click on your new grouping object. A blank workspace displays where you can keep your master configuration.

2. Drag out a Modbus Import module.
3. Use the ReadNow input of the Modbus Import module if you want to setup a trigger source that activates a read (i.e. a pulse). If you do not link ReadNow the module polls Modbus devices continuously.

4. Right-click the Modbus Import module to configure register settings.
   Configure the following setup registers as needed: Slave Address, Register Address, Number of Registers read by the module, Format and scaling requirements. The supported Slave Address range (Unit ID on ACCESS meters) for a Modbus device is from 1 to 247.

5. Link the output registers to desired modules (such as Data Recorders).
6. In Vista, create and link an object that best displays your configuration.

**Writing Data Using the Modbus Export Module**

The Modbus Export module writes data to a Modbus device. A Modbus master can send data (numeric, pulse and Boolean register classes) to an ACCESS meter via Modbus. You can use this data to enable, disable, and reset metering functions, as well as change setup register values to configure the meter's operation.

**Writing data to ACCESS meters**

1. Launch Designer (ensure that Options>Show Toolbox is checked).
   You can drag out a new grouping object from the toolbox, name it appropriately and double-click on your new grouping object. A blank workspace displays where you can keep your master configuration.

2. Drag out a Modbus Export module.

3. Link the Source inputs of the Export module to values that you want to export. You must link WriteNow to initiate a write.

4. Right-click on the Modbus Export module to configure the setup register settings.
   Set the Slave Address, Register Address, Request Type, Format, and any scaling requirements. A Slave Address of zero (0) is reserved for broadcast messages. Broadcast messages are useful for control actions, since slaves do not respond to a broadcast. A typical use of a broadcast is to synchronize all the devices (only an approximate synchronization is possible due to signal latencies).

   **Tip**

   Use Designer to determine the ION module/register that you want to write and then use the ION/Modbus Register Map document to find the corresponding Modbus register number.

5. In Vista, create and link an object that best displays your configuration.
ACCESS Meters as a Modbus Slave

All ACCESS meters have a unique unit ID. When acting as a slave, the ACCESS meter’s unit ID serves as the slave address that the master uses to connect. These addresses must be in the range 1 to 247.

NOTE

A Modbus slave always responds to requests; it never initiates them.

Most ACCESS meters have pre-configured Modbus Slave modules. These can be accessed using Designer, inside the ACCESS meter’s “3rd-Party Protocols” folder. If these pre-configured modules do not suit your needs you can drag out new Modbus Slave modules or change the configuration of the existing ones.

Using the ACCESS Meter as a Modbus Slave

You can use the Modbus protocols for real-time data communications on the meter’s serial, Ethernet, modem and Infrared ports. These hardware ports are controlled by ACCESS communications (COMM) modules that exist in the meter’s firmware. First, choose the communications port(s) through which you want to transport the Modbus data. The meter can then transmit data from ION Modbus Slave modules. The Modbus Slave modules convert an ACCESS meter’s data into Modbus format for Modbus master devices. Most meters that support Modbus Slave modules already have some pre-configured modules for common parameters such as power and energy values and power quality information.

Modbus Slave Module Settings

Once the communications port is configured to use Modbus, the data is available to a Modbus master device. The Modbus Slave modules are pre-configured to communicate specific power system parameters in a particular format. A sample configuration is shown below:

Sample of Factory Configuration
When there are no input links, on some ACCESS meters, the output registers show the contents of the fixed Modbus address map as defined by the setup registers of the Slave Module. The values at the map address are copied to the output registers, starting at the address specified in the BaseAddr setup register, until each output register is filled. You must change the Format setup register to “Unsigned 16B Input Mode.”

**Changing default Modbus configuration**

1. Open the ACCESS meter in Designer.
2. Re-link a new, or existing Modbus Slave module to any parameters.
3. Right-click near the center of the module to edit any of the registers. You also edit the setup registers of the Slave module if your Modbus master device requires data in a format different than the factory configuration or requires special addressing. Modbus Slave modules have the following setup registers:

<table>
<thead>
<tr>
<th>Setup Register</th>
<th>Default Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format</td>
<td>Unsigned 16B</td>
</tr>
<tr>
<td>Base Address</td>
<td>40011</td>
</tr>
<tr>
<td>Scaling</td>
<td>No</td>
</tr>
<tr>
<td>IONInMinScale</td>
<td>0</td>
</tr>
<tr>
<td>IONInMaxScale</td>
<td>65535</td>
</tr>
<tr>
<td>ModbusOutMinScale</td>
<td>0</td>
</tr>
<tr>
<td>ModbusOutMaxScale</td>
<td>65535</td>
</tr>
</tbody>
</table>

See the Modbus Slave module description in the ION Reference for more information on using the Modbus Slave module.