

MODBUS NETWORKS

Introduction

This document provides a general introduction to industrial networking, and in particular to the Modbus RTU protocol, on which the Connect network is based.

Physical Interconnection Standards

RS232

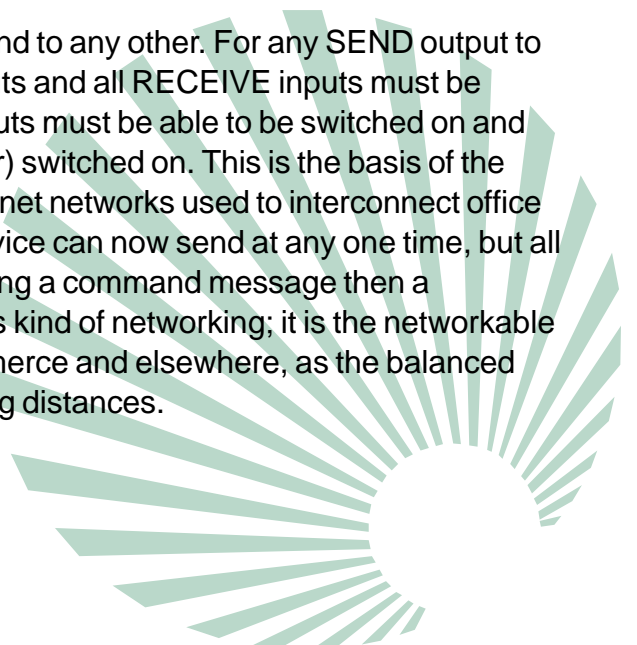
RS232 is a very old standard which still persists as it is very popular and generally adequate. It specifies the physical interconnection between just two devices for point-to-point serial communications over a distance up to 10m. It's separate send and receive wiring paths generally allow simultaneous communications in both directions (full duplex).

RS422

While RS232 connections use 1 unbalanced wire for each signal circuit with a common ground return, the better RS422 standard specifies more sensitive circuitry and two wires per signal circuit, arranged as a balanced pair (the two wires carry opposite voltages) to reduce both incoming and outgoing interference, and to increase the useable transmission distance up to more than 1km at low speeds. Like RS232, RS422 is a point-to-point full duplex system to interconnect only two devices.

RS485

Modern networks require that any device be able to send to any other. For any SEND output to be able to access any RECEIVE input, all SEND outputs and all RECEIVE inputs must be connected together electrically, and all the SEND outputs must be able to be switched on and off on demand, leaving one one (the current transmitter) switched on. This is the basis of the common party-line network, including the coaxial Ethernet networks used to interconnect office PCs. These are HALF-DUPLEX because only one device can now send at any one time, but all network protocols work this way, each transaction having a command message then a response message. The RS485 standard supports this kind of networking; it is the networkable version of RS422, and is widely used in industry, commerce and elsewhere, as the balanced format provides a reliable physical connection over long distances.



CONNECT Network

CONNECT uses a half duplex RS485 physical network, so a PC running CONNECT or SCADA software will need either an automatic commercial RS232-RS485 converter, or the ANP1 power supply (which also converts RS232 to RS485), or any plug-in RS485 PC card supported by the operating system. These components enable the electrical connections between devices. They do not ensure compatibility, as it is also necessary that the devices talk the same language, and to this end, CONNECT supports the well known Modbus protocol.

CONNECT's Modbus implementation uses an RS485 network at 9600 bits per second, or about 48 basic transactions per second. Theoretically it is possible for 24 network devices to be polled every 500ms, but with a PC host, 600ms would be more practical due to operating system timing errors.

Modbus Protocol

History

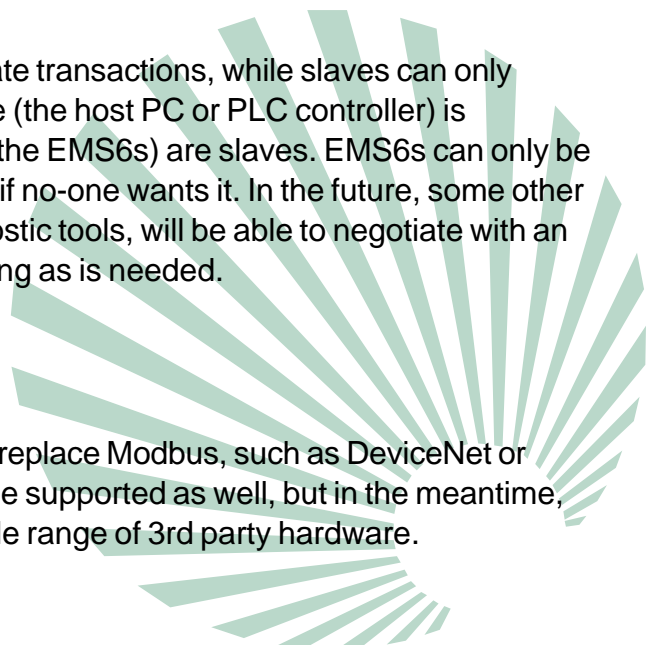
Modbus is one of the oldest serial communications protocols for industrial applications, and because of its reliability and initial widespread application, it has gained very wide acceptance as an open standard ever since its inception. Further, unlike all modern standards, it can be implemented easily by any PC or small microprocessor, so even today, it is supported by an increasing number of manufacturers for simple network communications, while virtually all other legacy standards have fallen into disuse. Modbus was devised by the company Modicon, who were amongst the most successful of the early manufacturers of programmable logic controllers (PLCs). Many other manufacturers of sensors, controllers and PLCs also support the Modbus protocol, however, Modicon's rivals all have their own protocols (which are less widely supported), but even they all offer Modbus options as well.

Master-Slave Operation

Modbus is a master-slave network. Masters can initiate transactions, while slaves can only respond. In a traditional Modbus network, one device (the host PC or PLC controller) is permanently assigned as the master, and all others (the EMS6s) are slaves. EMS6s can only be slaves, as there is no point in an EMS6 sending data if no-one wants it. In the future, some other devices, such as hand-held configuration and diagnostic tools, will be able to negotiate with an existing bus master to share the mastership for as long as is needed.

Future Directions

In the future, another faster protocol will undoubtedly replace Modbus, such as DeviceNet or Fieldbus H1, and at that time that protocol will likely be supported as well, but in the meantime, Modbus provides the easiest way to connect to a wide range of 3rd party hardware.



PCs and PLCs

Like PCs, PLCs can be programmed in many languages, although the most popular PLC language is still the traditional Ladder Logic which allows the programmer to simulate a collection of relays, timers, etc. PCs can handle larger volumes of data, but for many reasons (mostly to do with their Microsoft operating systems), timing is imprecise and communications are inefficient. A PC/SCADA is ideal to create historical logs, display plant diagrams, values, status and trends. A PLC is used for all the logic, interlocking, safety systems and general control of industrial plant, and usually, to report back an overall picture to the PC/SCADA system.

PC Modbus Masters

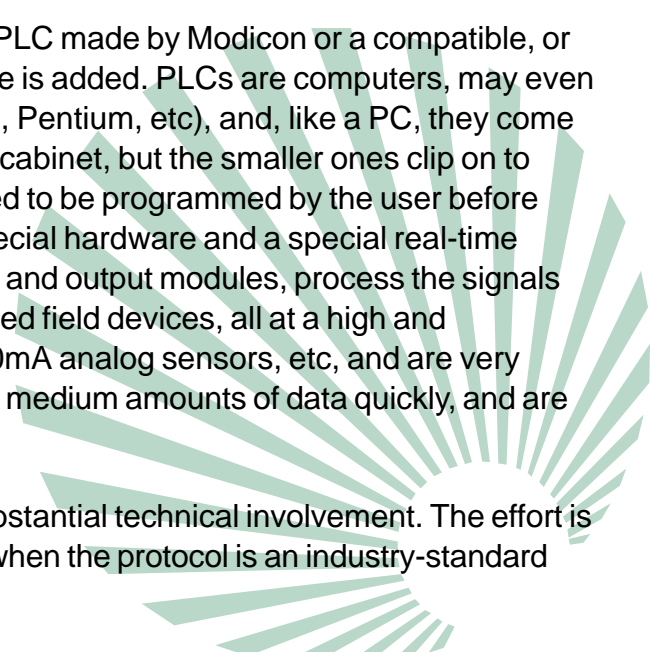
The network is controlled by any Modbus Master, which is responsible for polling (asking) each separate node in turn for it's reading. The master can be a personal computer (PC) with data logging software. Some SCADA systems (like CITECT) support Modbus out-of-the-box, while others may need the addition of an optional installable driver, which can be written or purchased separately. Alternatively, the CONNECT software can be used. All these PC based options run on current hardware under 32 bit Microsoft Windows operating systems, typically Windows 95, 98, NT4.0, 2000, or, in the future, Windows CE (which runs on small hand-held computers).

To program a PC to poll network nodes usually requires the purchase and installation of a Data Logging software package (either CONNECT, or a larger third party SCADA system). A SCADA system will be easier to configure than a PLC, but still needs considerable technical effort to set up the communications system, to set up the database to accept the desired EMS6 signals, and to set up display screens to display the data. CONNECT is easiest of all, as it will generally work directly out-of-the-box with a minimum of setting up.

PLC Modbus Masters

The Modbus master can alternatively be nearly any PLC made by Modicon or a compatible, or most other makers too, if an optional Modbus module is added. PLCs are computers, may even use the same microprocessor as a desktop PC (486, Pentium, etc), and, like a PC, they come in a wide range of sizes (most are bolted to a wall or cabinet, but the smaller ones clip on to standard DIN or G-type terminal rails). Both also need to be programmed by the user before they become useful, but PLCs are equipped with special hardware and a special real-time operating system so they can scan their plug-in input and output modules, process the signals and send control commands back out to the connected field devices, all at a high and dependable speed. PLCs read switch contacts, 4-20mA analog sensors, etc, and are very efficient at repetitively handling communications and medium amounts of data quickly, and are generally very reliable.

To program a PLC to poll network nodes requires substantial technical involvement. The effort is often reduced and the results more generally useful when the protocol is an industry-standard like Modbus, but it is still substantial.



PLC Polling Parameters

With small networks of relatively simple devices such as EMS6s, each device mostly supplies only one value in response to each poll (request), even though the Modbus protocol allows many values to be sent in each message. If a PLC acts as the Modbus master, it can maintain tight control of the polling, and can easily be programmed to react to exception conditions, or to provide corrections (eg using temperature and pressure data from other sensors to correct mg/m³ data into normalised mg/Nm³ data at NTP), or to mask the data when it is known to be inaccurate due to other influences, etc. The PLC can then assemble into a single table all the data needed by the SCADA system from the EMS6s and all its other inputs and calculations, so the SCADA system can get this data easily (often the whole table can be returned in a single reply). Generally, it is recommended to scan EMS6s at 500ms to 2000ms. If the PLC polls the EMS6s at 500ms, it can also present much steadier data to the SCADA system if it keeps a running average of, say, 10 samples, and the SCADA system polls the PLC every 5s, as is quite common for SCADA systems.

Network Theory

Network Termination

Every cable has a characteristic impedance, which is the impedance which would be measured if the cable was infinitely long. When a waveform propagates along a cable (eg a Modbus message along Connect's RS485 network), it does so smoothly until it reaches an end. If the end is connected to another similar cable, the waveform just continues unhindered from one cable into the other. If a resistor of the same value as the cable's characteristic impedance is substituted for the second cable, the waveform is still unhindered, and is simply absorbed by the resistor. On the other hand, if the resistor is the wrong value, or absent, the waveform is reflected back from the end of the cable, and interferes with itself, like water waves reflecting from the edge of a pool.

Network Characteristic Impedance

The cable is a continuous circuit element with small values of series resistance and inductance, parallel (shunt) capacitance, and almost no shunt conductance. The series elements raise the effective impedance of the cable and cause the current to lag the voltage. The shunt elements reduce the effective impedance and cause the current to lead the voltage, so the characteristic impedance is the result of those two counteracting effects. If the cable is designed for communications, the inductive and capacitive elements are balanced so the characteristic impedance is resistive only, or at least a good approximation to it.

