

# EVERYTHING YOU WANTED TO KNOW ABOUT MODBUS BUT WERE AFRAID TO ASK

For the majority of the applications we are involved with, our data loggers, such as the [dataTaker DT80](#), are wired directly to sensors or equipment that provide an analog signal such as a 0-10V or 4-20 mA that the logger measures and scales to the correct units such as pressure, force, temperature, etc. However, we are finding that there is an increasing need to connect to newer types of sensors that provide a digital output or other devices like displays or intelligent controller.

In some cases, the customer may have a number of sensors distributed around a facility where it is not practical to use long cable runs to bring the signal from each individual sensor back to the logger. In these situations, we have found that it is very convenient to use [Modbus](#) communications to move the data between devices. So, I thought it would be a good idea to tell you a little bit about Modbus and how you can use it.



## WHAT IS MODBUS?

Modbus is an industrial standard communications protocol for exchanging data between devices that was developed in the late 1970s. Initially designed for communicating between [programmable logic controllers \(PLCs\)](#), it has grown to support a wide variety of devices including sensors, valves, drives, data acquisition equipment, and PCs. Modbus uses a client/server model (previously known as master/slave) where the client will either request or send data to the server and, in return, the server will respond with the requested data. For example, to request a measured value like temperature or humidity or to perform some action like closing a relay or setting the speed of a motor.

In the Modbus protocol, each server has a unique device address and one or more address registers. Data transfer begins with the client (master) sending a request message to one and only one server (slave). The server will never initiate a data exchange. The Modbus protocol defines a number of request types such as reading or writing an individual bit or coil that normally corresponds to a relay or switch state of on or off. Another request type is to read or write one or more registers containing numeric values. The numeric values can be in the form of 16 or 32-bit integers or as floating point numbers. The server will define the format of the data and the client must be flexible enough to interpret it correctly.

## HOW DO MODBUS DEVICES CONNECT TO EACH OTHER?

Now you may be asking yourself, “how do the modbus devices physically connect?” It turns out that Modbus can run on several different electrical interconnections. Historically, the most common type is what is known as Modbus RTU, which uses a serial interface. If you have only two devices, a standard RS-232 serial interface could be used. The main drawback of RS-232 is that it is only for point-to-point communication, so you can only have two devices. Modbus RTU also works with RS-485 serial communications, that typically allows up to 32 devices to be connected to the bus. Another advantage of RS-485 over RS-232 is that it can allow for a very long bus – up to 4,000 feet with the proper cable and termination. There is also Modbus TCP, which runs over standard Ethernet networks, so it is limited only by the standard Ethernet constraints of IP address space and cable lengths. In addition, Modbus TCP allows for multiple clients and servers; in fact, the same device can be a client and server.

In the last few years, like everything else, Modbus has also gone wireless. A number of companies, like [Novus Automation](#) and [LumenRadio](#), offer wireless gateways that can extend Modbus RS-485 networks thousands of feet using point-to-point, star, tree, and mesh topologies. Likewise, vendors are also offering Modbus devices that use Wi-Fi to wirelessly move data using the Modbus TCP protocol.

## HOW IS MODBUS USED IN REAL APPLICATIONS?

### **1. For connection to a display.**

We have built many systems with data loggers connected via Modbus to LCD screens from [Maple Systems](#) to allow users to view real-time data collected by the data logger on dashboards and graphs. This eliminates the need to have a separate PC and allows the use of small portable enclosures to hold the whole system.

### **2. To bring data from multiple locations in a plant back to a data logger.**

A customer had multiple furnaces within a large manufacturing plant. They wanted to measure the temperature at several spots in each furnace and then bring all of the data back to a single data logger, which would save the data for record keeping. However, because of the size and layout of the facility, it was not convenient to hardwire all of the sensors to a single logger. We built 5 remote transmitter boxes each with a wireless Modbus RTU gateway and several Modbus thermocouple input modules which could be located near the individual furnaces. A data logger with its own wireless Modbus RTU gateway was located at an easily accessible location. The logger was programmed to sequentially request the temperature from each of the thermocouples via Modbus and then save the data in its internal non-volatile memory.

### **3. To bring data from multiple locations within a manufacturing campus back to a data logger.**

We have a customer that has a manufacturing campus with multiple buildings within an area of several acres. They wanted to get environmental data from each of the buildings along with the temperature of freezers located in several of the buildings back into their quality control system. Similar to #2 above, we built data collection boxes for each building with a mix of Modbus input modules to suit the different types of sensors in the building, plus a wireless Modbus RTU gateway. Because of the layout of the facility, we used a remote mounted antenna on the gateway and positioned it outside the building located

high enough to get a good line of sight to a wireless gateway attached to the data logger. In this application, not only did the data logger sample and store the data from each of the remote buildings, but it was also connected to their Ethernet network so their QC system could periodically pull the most recent measurement via Modbus TCP. In essence, the logger acted as a Modbus gateway between the wireless RTU network and the TCP network for their plant.

#### **4. To Simplify Calibration**

We just provided a solution for a government customer with multiple laboratories in a large research facility. They need to monitor the temperature and humidity in the labs and they had a need to be able to easily calibrate the temp/RH sensors. If analog sensors were used, they would have to take the loggers and sensors offline and calibrate them as a unit. Instead, we provided them data loggers with two sets of sensors with a Modbus RTU interface. Since the data exchanged between the sensors and the loggers is digital and not analog, the sensors could easily be calibrated by themselves as they had built-in offset and gain correction factors for the Modbus measurements. And, with redundant sensors for each logger, one set could be sent out for calibration without interrupting the monitoring of the labs. Exchanging sensors was a simple matter of unplugging one set of connectors and plugging in the other set resulting in virtually zero downtime.

In conclusion, Modbus is a mature communications protocol that simplifies data exchange between many types of industrial measurement and control devices, PLCs, HMIs, and data loggers. It offers the flexibility to use different communications methods – serial, Ethernet or wireless, and to transfer different kinds of data including on/off state, integer, and floating point values. Modbus is especially useful in distributed data acquisition projects to capture data from multiple pieces of equipment.