Technical Note 48

Modbus Tips and Troubleshooting

Overview

This technical note provides solutions to common problems encountered implementing Modbus networks. It is intended as a supplement to the MBX-MDB help, and not as a replacement. Wherever possible, the appropriate sections of the help system are referenced, and the reader is strongly encouraged to review these sections. The Orion help system also contains a number of programming examples, as well as hardware configuration information. Refer to the “Modbus Communications” page for details (select “Help Topics”, then select the “Index” tab).

Common Mistakes

• Make sure that the network type and baud rate parameters specified with the MOD.OPEN statement match the configuration for the remainder of the network.

• Before a variable can be accessed over the network, it must be “mapped”. Refer to the MBX-MAP documentation for details.

• After initializing the Modbus interface with the MOD.OPEN statement, the MotionBASIC program must also set MODBUS@ equal to TRUE. This enables access to “mapped” variables via the network.

• Make sure that the MODBUS network has only one “master” device, and all remaining devices are configured as “slaves”. The “master” is the device that initiates data transfers, and “slaves” respond to data transfer requests from the “master”.

• If a PLC is performing reads and writes on another device on the network, the message block must perform the read/write function, and then wait for a reply to the message before another read/write is attempted. A message block that is executed each scan of the PLC program can cause all sorts of network errors.

• If the Orion is performing reads and/or writes on another device on the network, the program must not perform a second read/write until MOD.MSG@ becomes true. After MOD.MSG@ becomes true, the application should also check MOD.STS@ to verify that the operation completed successfully. If MOD.STS@ is zero, this indicates that no errors were detected. A non-zero value indicates a problem. Refer to the documentation for MOD.STS@ for additional details.
• If possible, program the PLC such that there is a time interval between message block executions. This helps to relieve network traffic. MotionBASIC® program execution speed may also be slowed down because the CPU must process all characters sent and received over the serial interface.

• Electrical noise can cause problems on any high-speed network. Make sure that proper grounding and shielding techniques are followed. An application note detailing the recommended grounding and shielding techniques can be found at http://www.ormec.com.

**Network Problems**

The assumption is made here that the “Common Mistakes” section of this document has already been referenced, and the network still does not function properly. There are a number of cabling examples in the Orion help system under “Modbus Cabling Examples”.

• For a direct RS-232 connection between a PC and an Orion, or between two Orions, a “null modem” connection must be made.

• If possible, reduce the network to the Orion, and one other device (preferably the “master”, if the Orion is a “slave”). Test the network operation with these two devices. If it works, start adding devices to see which one is causing problems. If it doesn’t work with these two devices, get it working before adding additional devices.

**Register Access Problems**

This section assumes that the network is functional, but there is a problem with register access to/from the Orion.

• Make sure MODBUS@ is set ON.

• If registers are being accessed by another device on the network (a PLC, for example), make sure that the appropriate registers are “mapped” in the Orion. There are a number of examples of MotionBASIC code, and PLC ladder diagrams in the help system.

• There is an offset in register numbers between Modicon PLCs, and the Orion. For example register 0 in an Orion is register 40001 in a Modicon 984. This may also be true for other devices such as MMI software packages, etc. Refer to the “Modicon / Modbus Register Number” section in the help system.

• If the PLC is reading and writing registers, watch out for entering decimal numbers where octal or hex numbers are required in the message block definitions.

• Use MOD.TRACE@ to enable more debugging information.

• Use MOD.STS@ to determine the type of errors that are detected.

• Use MOD.DUMP to get a detailed listing of useful debug information.