Application Note for
OCS Modbus Communications
Using Data-Linc SRM6000 Radio Modems

Scope: This document covers how to make one OCS, acting as a Modbus Master, talk to another OCS, acting as a Modbus Slave, using two Data-Linc SRM6000 Radio Modems as a means of communications between the two.

Purpose: The purpose of this document is to show that, indeed, it can be done.

Equipment: 2 OCS units with Firmware 8.15 or higher
2 Data-Linc SRM6000 Radio Modems (Firmware Ver. 5.51 used here)
2 Null-Modem 9-pin Male-Male cables
1 Straight-Through 9-pin Male-Female cable
1 Computer with CScape 3.1 or higher installed

Wiring: The COM port on the computer is an RS-232 DCE device with the following pinout:

Computer RS-232 COM Port Pins			
Pin	**Pin Name**	**Description**	**Direction**
2	RXD	Received Data	In
3	TXD	Transmitted Data	Out
5	GND	Ground	-
7	RTS	Request to Send	Out
8	CTS	Clear to Send	In

Both the OCS and the Data-Linc Radio Modems are RS-232 DTE devices with the following pinouts:

OCS and Radio Modem RS-232 Port Pins			
Pin	**Pin Name**	**Description**	**Direction**
2	RXD	Received Data	Out
3	TXD	Transmitted Data	In
5	GND	Ground	-
7	RTS	Request to Send	In
8	CTS	Clear to Send	Out

Note: The notation used here is the RS-232 EIA standard, in which the pin names are based on the DCE device. The Receive (RXD) line on a DTE device is actually a signal Out to the Receive (RXD) line on the DCE device, where it is a signal In. This goes for the Transmit (TXD) line as well. The TXD line on the DTE device is actually a signal In from the TXD line on the DCE device, where it is a signal Out.

Connections between the computer COM port and the OCS 9-pin port for programming purposes are made with the Straight-Through 9-pin cable. The same cable is used to configure the Data-Linc Radio Modem. The cable is as follows:

Computer COM Port to OCS or Radio Modem Port			
Pin Name	**COM Port Pin**	**Wire to Use**	**OCS/RM Pin**
RXD	2	Twisted	2
TXD	3	Pair	3
GND	5	Single	5
RTS	7	Twisted	7
CTS	8	Pair	8

Connections between the OCS port and the Radio Modem port for Modbus communications purposes are made with a Null-Modem cable. The cable is as follows:

OCS Port to Radio Modem Port				
OCS Pin Name	**OCS Port Pin**	**Wire to Use**	**RM Port Pin**	**RM Pin Name**
RXD	2	Twisted	3	TXD
TXD	3	Pair	2	RXD
GND	5	Single	5	GND
RTS	7	Twisted	8	CTS
CTS	8	Pair	7	RTS

Note: For this application, the handshaking was set to NONE and the handshaking lines (RTS and CTS) were not used. Therefore, it is possible to only use pins 2, 3, and 5.

Data-Linc Radio Modem Configuration: It is necessary to make sure the Radio Modems are configured for Modbus communications, as well as making sure the baud rate and other communications settings are the same as for the OCS. Both Radio Modems will need to be configured.

To configure the Radio Modem, connect the Computer COM port to the Radio Modem 9-pin port using the Straight-Through 9-pin cable and connect power to the modem through its adapter. You should see the red 'P' LED light up. On the Computer, a dumb terminal program, such as Term.exe or HyperTerminal in Win95/Win98, should be run. Configure it to talk at 19,200 Baud, 8 Data bits, No parity, 1 stop bit, and No Handshaking. Once this is accomplished, you will need to put the Radio Modem into Configuration mode. Do this by inserting a paper clip into the small hole in the back panel of the Radio Modem and pressing the button inside. The Radio Modem's Main Menu should appear on the terminal program.

Press 0 for Set Operation Mode. One Radio Modem should be set to Point to Point Master, the other to Point to Point Slave. **It makes no difference whether the Radio Modem set to Master is connected to the OCS acting as Modbus Master or to the OCS acting as Modbus Slave. This setting is completely separate from anything relating to Modbus.** Once this is set, press Esc to exit back to the Main Menu.

Next, press 1 for Set Baud Rate. Select one of the baud rates that will match the baud rate you set for the OCS in the Open Port Function Block. For this test, both 9600 Baud and 57.6 KBaud worked fine. (115.2 KBaud pushes the OCS limits too far and errors occurred regularly.) This setting only affects the rate of transfer between the OCS and the Radio Modem. Communications between the Radio Modems is fixed at 144 KBaud. After selecting a baud rate, the "Data, Parity" (option A) should be set to 0. This corresponds to a setting of 8 Data bits, No parity, and 1 Stop bit. After setting the Data and Parity, the MODBus RTU (option B) should be set to 1. This enables the Radio Modem to transmit the Modbus protocol. Once these options are set, press Esc to exit back to the Main Menu.

Next, press 3 for Edit Radio Transmission Characteristics. The first option within this menu is the FreqKey. This setting MUST be the same on both Radio Modems. Valid values are from 0 to E (hex, of course). The value can be changed if interference is suspected or if communications seem slower than normal, but it must be changed in both Radio Modems. **Options 1, 2, 3, and 4 should not be changed!** Option 5 sets the transmission power of the Radio Modem. Valid values are from 1 to 9. The higher the number, the further apart the Radio Modems can be placed. Within a building and within 100 feet of each other, a setting of 3 should be sufficient. A setting of 9 may enable the Radio Modems to be placed as far as 20 miles apart, though conditions would have to be nearly perfect. (A test was conducted where one Radio Modem was left here at Horner Electric, APG, and the other taken to Horner Electric, ISS, about 2 miles as the crow flies. With the transmit power set to full, communications could not be established. However, within the Horner Electric, APG building, a distance of a few hundred feet was easily accomplished.) **Options 6, 7, and 8 should not be changed!** Once these options are set, press Esc to exit back to the Main Menu.

Press Esc from the Main Menu to ready the Radio Modem for communications.

OCS Configuration: For this particular test, an OCS100, with two DIQ611 modules as I/O cards, was used as the Modbus Master. An OCS200, with no I/O cards installed, was used as the slave. Which OCS units are used and what I/O cards, if any, are used makes no difference.

Slave

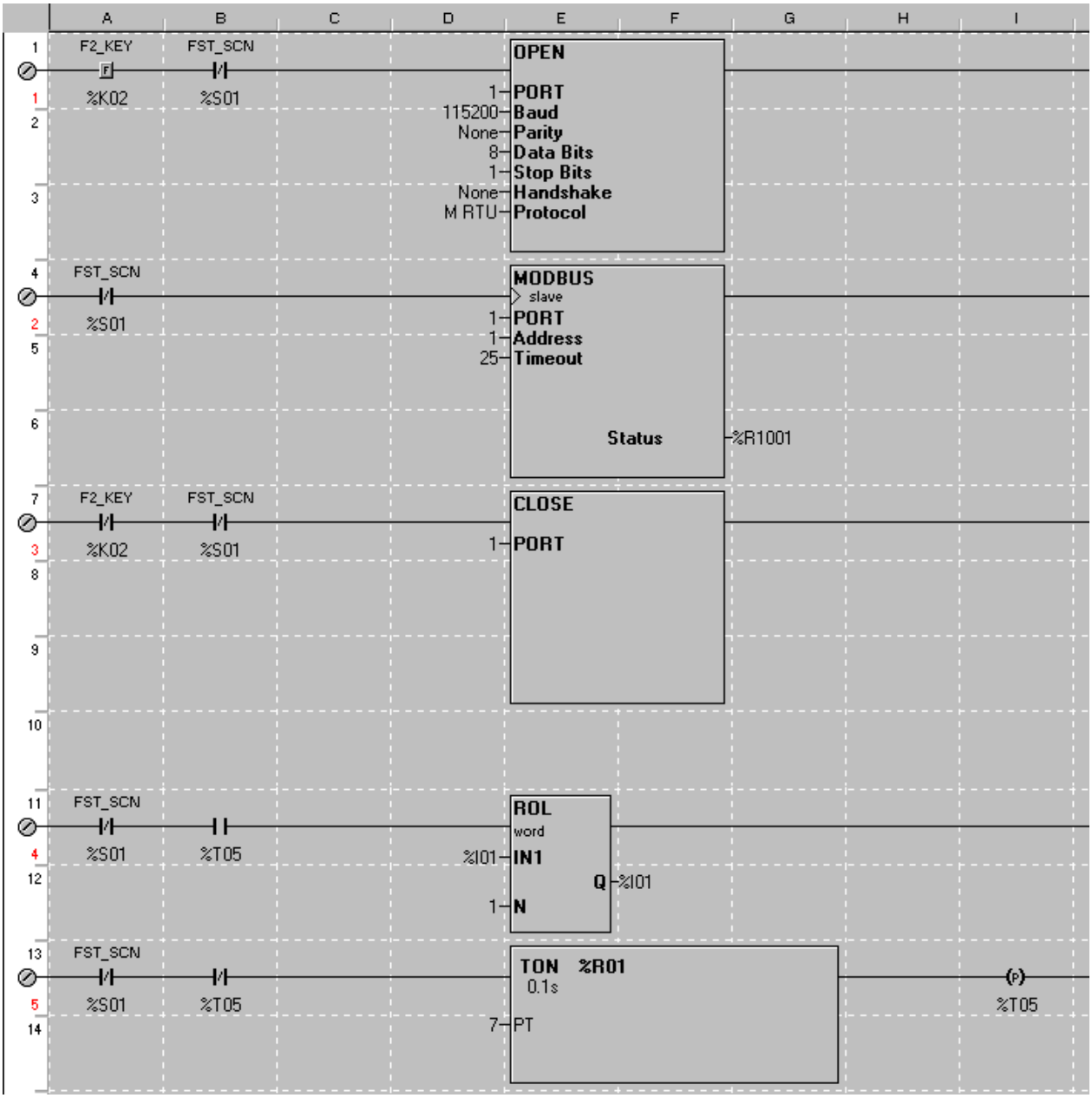
The program written and downloaded to the Modbus Slave OCS200 for this test is shown in Appendix A. It is simple, as the Slave does not do anything except for what the Master tells it to do. In this program, the Function Keys are set to Toggle mode. The port is opened by pressing F2, thereby toggling the N.O. %K02 contact on and opening the OCS port for Modbus communications. Once the port is opened, the Modbus Slave Function Block is executed, specifying the OCS port 1 as Modbus Slave 1. The timeout is set to 25/10ths of a second, or 2.5 seconds. The Status Register is set to %R1001. When the F2 key is pressed again, %K02 is toggled off and the N.C. %K02 contact passes power to the Close Port Function Block. The only other function this ladder program executes is a bit rotator, which rotates the 16 bits from %I01 to %I16 every 0.7 seconds. These registers are what the Master reads in this test.

Master

The program written and downloaded to the Modbus Master OCS100 for this test is shown in Appendix B. The F-Key mode should be set to Momentary. The basic operation of the program is to first initialize the Message Control Block (MCB) register with the first MCB values. When the user presses F1, the port is opened for Modbus Communications. The port is defined as a Modbus Master with a timeout of 25/10ths of a second, or 2.5 seconds. The Status Register is set for %R1001. The rest of the program does an assortment of things. There are two MCBs for this test. The first MCB writes the contents of the Master OCS's registers %I01 through %I16 to the Slave OCS's registers %Q01 through %Q16. The second MCB reads the Slave OCS's registers %I01 through %I16 and places those values in the Master OCS's registers %Q1 through %Q16. Upon power up, the first MCB is loaded into the MCB Register. When the Status Register turns out a 1, meaning the command was successful, the second MCB is loaded into the MCB Register and told to execute. The Status Register is zeroed. When the Status Register turns out a 1, the first MCB is loaded into the MCB Register and told to execute. This continues to cycle until a fault is reported (Status Register turns out something higher than 1). Upon seeing a value of anything higher than 1 in the Status Register, Screen 5 is triggered and the Status Register is shown in Binary format. To clear the fault and close the port, F2 is pressed. To restart the port, F1 is pressed again.

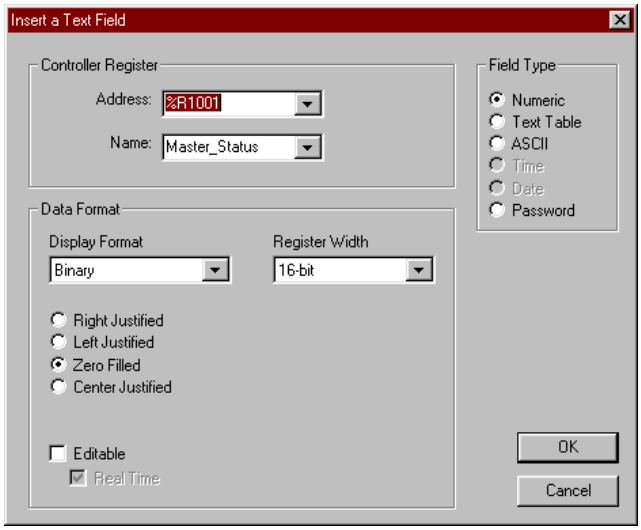
Appendix A: Slave OCS CScape Program

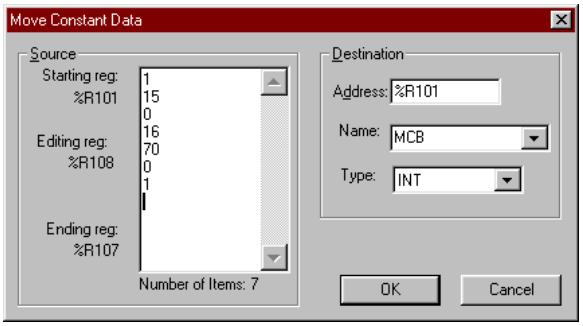
The program on this page is for the OCS acting as the Modbus Slave. Its function is simple: Open the port for Modbus Communications and wait for the Master to tell it what to do. The bit rotating logic (the last two rungs) rotates the bits in %I01 through %I16 for the Master to read back.

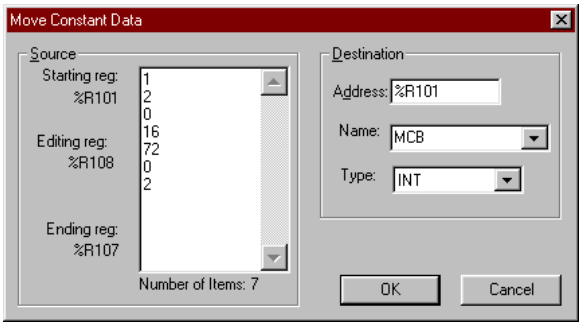


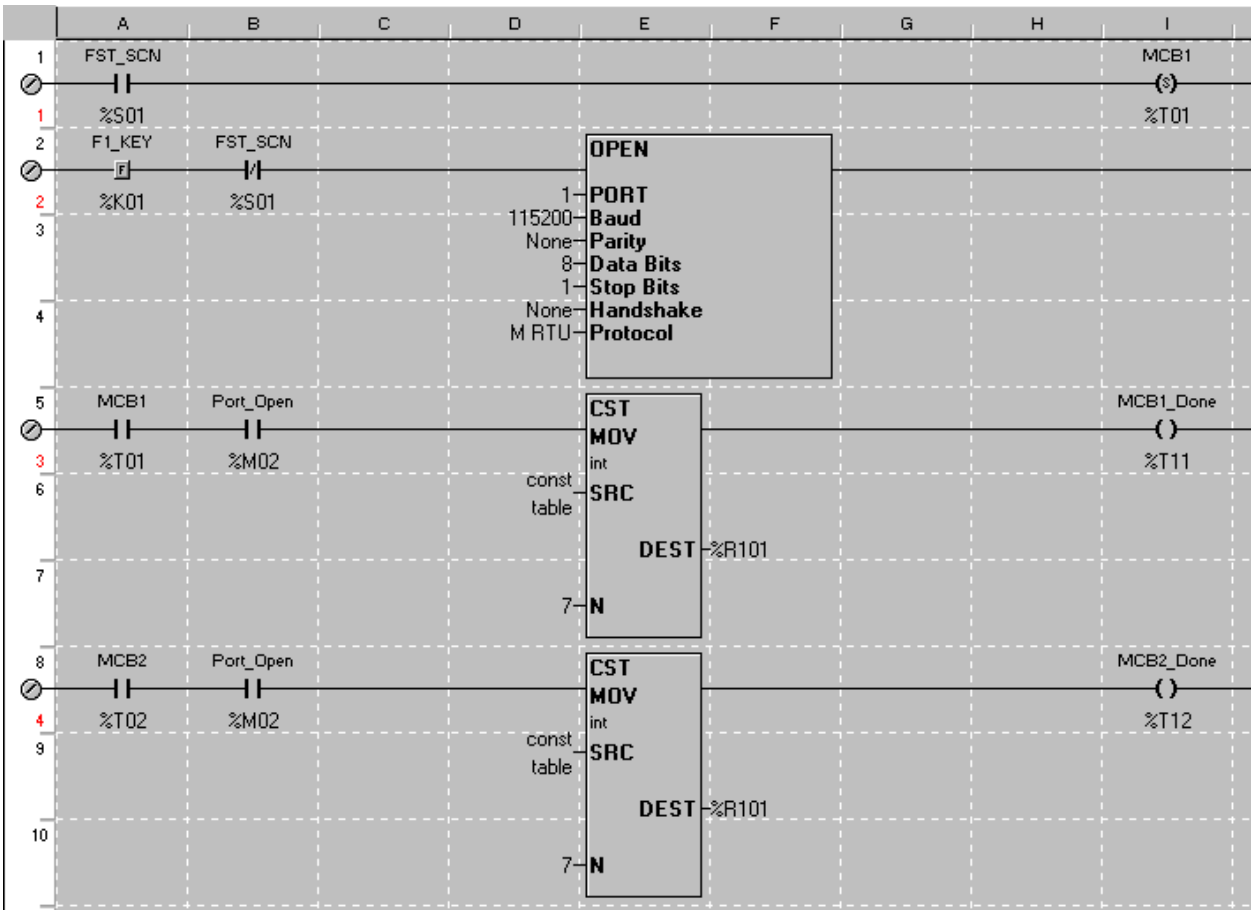
Appendix B: Slave OCS CScape Program

The program on this and the following pages is for the OCS acting as the Modbus master. The data field on Screen 5 is setup as shown here to the right. The screen will trigger if a fault is logged and communications will stop until the fault is cleared, by pressing the F2 key, and the port is restarted, by pressing F1. The Constant Move function blocks are set up as shown below. They act as the Message Control Blocks (MCBs) for Modbus communications. The seventh entry is only to let the ladder logic know which MCB is currently being executed. Entries 1-6 are as according to the CScape documentation.









Appendix B: Slave OCS CScape Program (Cont.)