



Setting Up an HSQ RTU as a Modbus Slave

Introduction

MODSLV and MODTCPSLV are programs which implement the Slave part of the Modbus protocol in RTU mode, allowing an HSQ RTU to respond to queries sent from a Master device. MODSLV is the program used to send data over a serial connection and MODTCPSLV is the program used to send data over an Ethernet connection. The programs send 8-bit binary characters and use a CRC-16 error check sequence to ensure the quality of the data transmission. When the Modbus Master requests it, the RTU responds with the status or value for one or more points of the same type. For example, the Modbus Master sends a Code 2 message and the RTU responds with the status for all of its digital inputs (DIs).

The implemented Modbus function codes are:

- 1) **Code 1** — Read Coil Status (DO).
- 2) **Code 2** — Read Input Status (DI).
- 3) **Code 3** — Read Output Registers (AO).
- 4) **Code 4** — Read Input Registers (AI).
- 5) **Code 5** — Force Single Coil (DO).
- 6) **Code 6** — Preset Single Register (AO).
- 7) **Code 15** — Write Multiple Coils (DO).
- 8) **Code 16** — Write Multiple Registers (AO).

More information on the Modbus protocol is available in the *Network Communication Controller (NCC) Protocol List*.

Overview

The file, ***MNET\$DATA:MODBUS.DAT***, defines the configuration for the MODSLV/MODTCPSLV program. The settings on the ***PARAMETERS*** line define:

- 1) PLC Slave address
- 2) Name of the communication port (serial line or TCP)
- 3) Maximum number for each type of point.

IMPORTANT! The line detailing the maximum number of points must be at the beginning of the file. Any line prior to this one will be treated like a comment line, as will any line starting with an exclamation point (!).

HSQ RTU AS A MODBUS SLAVE

The **MODBUS.DAT** file also defines the correspondence between the Modbus address and the point acronym. The format of the address is:

- 1) **0XXXX** — COIL (Output Status) DO
- 2) **1XXXX** — Input Status DI
- 3) **3XXXX** — Input Register AI
- 4) **4XXXX** — Holding (Output) Register AO

The maximum query and response data parameters are:

- 1) **MAX BIN** (maximum query binary points) — 2000.
- 2) **MAX ANA** (maximum query analog points) — 125.

The **MODBUS.DAT** file is read when the MODSLV/MODTCPSLV program is started or anytime when the point “MODBUS-READ” is started. After reading the file and setting all of the parameters, the “MODBUS-READ” point is turned off.

A second instance of the MODSLV (not MODTCPSLV) program can be run with the name MODSL2. This allows two slave units to work on two different serial ports. A line in the **MODBUS.DAT** file specifies the second PLC address and second serial port name. This line starts with **MODSL2** and is placed before the **PARAMETERS** line. MODSL2 uses the “MODBUS-READ-2” point for reading the **MODBUS.DAT** file and “MODBUS-STATUS-2” for indicating the status of communications.

Example of MODTCPSLV MODBUS.DAT

```
! MODBUS      PLC Address      Port      MAX_AI  MAX_AO  MAX_DI  MAX_DO
! Process
PARAMETERS    1                502       320     1000    256     1000
!
! DEMVSA Source points:
!ADDR  ACRONYM                RTU ID      AO
!
40001  MODTCP-AC-1             ! 151        1,2
40003  MODTCP-AO-3             ! 151        3
!
!
!
30001  MODTCP-AI-1             ! 151        AI points
30002  MODTCP-AI-2             ! 151        1
!
!
!
10001  MODTCP-DI-1             ! 151        2 ab cd ef gh ij
10002  MODTCP-DI-2             ! 151        DI points
!
!
!
00001  MODTCP-DO-1             ! 151        DO points
00002  MODTCP-DO-2             ! 151
```

HSQ RTU AS A MODBUS SLAVE

NOTES:

- All connections must have the same PLC address.
- There is a maximum of five simultaneous Masters.
- All Masters must have the same point mapping.
- Accumulators – 32-bit floating point numbers require two registers each (e.g., 40001 MODTCP-AC-1).

Example of MODSLV MODBUS.DAT

```
! MODBUS      PLC Address      Port      MAX_AI      MAX_AO      MAX_DI      MAX_DO
! Process
MODSL2        3                          LNMB2
PARAMETERS    1                          LNMB1      320         320         256         256
!
! DEMVSA Source points:
!ADDR        ACRONYM                      RTU ID      AO
!
40001F      MODSER-AC-1                      ! 151        1,2
40003      MODSER-AO-3                      ! 151        3
!
!
!
30001      MODSER-AI-1                      ! 151        1
30002      MODSER-AI-2                      ! 151        2
!
!
!
10001      MODSER-DI-1                      ! 151        DI points
10002      MODSER-DI-2                      ! 151
!
!
00001      MODSER-DO-1                      ! 151        DO points
00002      MODSER-DO-2                      ! 151
```

NOTES:

- There is a limit of two ports (e.g., LNMB1, LNMB2). Each must be its own instance (MODSLV, MODSL2).
- Accumulators – 32-bit floating point numbers require two registers each (e.g., 40001 MODSER-AC-1).

Operation Errors

Operation errors involving illegal data in a message, result in an exception response from the MODSLV/MODTCP-AC-1 program. The exception response codes are:

- 1) **Code=01** — ILLEGAL FUNCTION (function received is not allowable action for the addressed slave).
- 2) **Code =02** — ILLEGAL DATA ADDRESS (address referenced in the data field is not an allowable address for the slave location).
- 3) **Code=3** — ILLEGAL DATA VALUE (value referenced in the data field is not an allowable address for the slave location).
- 4) **Code=4** — FAILURE DEVICE (Slave has failed to respond to a message or an abortive error occurred).

HSQ RTU AS A MODBUS SLAVE

Modbus Status Point

The "MODBUS-STATUS" point shows the status of the connection between MODSLV and the Modbus Master. It can be either UP or DOWN. The point should be defined as follows:

BINARY									15535
ACRONYM	MODBUS-STATUS					INPUT	SUBTYPE	CALC	
SEGMENT	0	NODE	DEMVSA	NCC	ID	OUTPUT	SUBTYPE	NONE	
NAME	MODSLV STS			RTU	ID				
AREA				MUX	ID			COS REPORTING	Y
BUILDING			IN	ADD			INTERVAL	TYPE	NONE
UNIT			OUT	ADD			AUTO	LAST	COMMAND
								RE-ISSUE	N
ON CODE	UP		OFF CODE	DOWN		INTERMEDIATE	CODE	MID	
ON STATUS	1		OFF STATUS	0		INTERMEDIATE	STATUS	0	
START VERB	START		STOP VERB	STOP					
MINIMUM ON TIME			MINIMUM OFF TIME						
VERIFICATION DELAY			RUN TIME LIMIT			POWER	DEMAND		
ON ALARM N	OFF ALARM Y		INTERMEDIATE ALARM N			UNCOMMANDED	COS	ALARM N	
ALARM DELAY			CRITICAL ALARM N			UNDEFINE	STATE	ALARM N	
ALARM PRINTERS	0		EVENT PRINTERS	0		MESSAGE	NUMBER		
ACK CATEGORY	INTO		PRINT CATEGORY	ALARMS		RELATED	TASK		
POINT ASSOCIATION	NONE		ASSOC POINT						
POINT ACCESS LEVEL	100		POINT CONTROL LEVEL	100		SLIDE	NUMBER		

HSQ RTU AS A MODBUS SLAVE

Modbus Line Monitor Point

The "MODBUS-LINMON" point enables and disables printing of trace messages received from the MODSLV program. The point should be defined as follows:

BINARY									15536
ACRONYM	MODBUS-LINMON					INPUT	SUBTYPE	NONE	
SEGMENT	0	NODE	DEMVSA	NCC	ID	OUTPUT	SUBTYPE	NORMAL	
NAME	LINMON				RTU	ID			
AREA					MUX	ID		COS REPORTING	Y
BUILDING			IN	ADD			INTERVAL	TYPE	NONE
UNIT			OUT	ADD			AUTO	LAST	COMMAND
								RE-ISSUE	N
ON CODE	ON		OFF CODE	OFF		INTERMEDIATE	CODE	MID	
ON STATUS	1		OFF STATUS	0		INTERMEDIATE	STATUS	0	
START VERB	START		STOP VERB	STOP					
MINIMUM ON TIME			MINIMUM OFF TIME						
VERIFICATION DELAY			RUN TIME LIMIT			POWER	DEMAND		
ON ALARM N	OFF ALARM N		INTERMEDIATE ALARM N			UNCOMMANDED	COS ALARM N		
ALARM DELAY			CRITICAL ALARM N			UNDEFINE	STATE ALARM N		
ALARM PRINTERS	0		EVENT PRINTERS	0		MESSAGE	NUMBER		
ACK CATEGORY	INTO		PRINT CATEGORY	ALARMS		RELATED	TASK		
POINT ASSOCIATION	NONE		ASSOC POINT						
POINT ACCESS LEVEL	100		POINT CONTROL LEVEL	100		SLIDE	NUMBER		

HSQ RTU AS A MODBUS SLAVE

Modbus Read Point

The "MODBUS-READ" point instructs MODSLV to read the **MODBUS.DAT** file. The point should be defined as follows:

```
BINARY 15537
ACRONYM MODBUS-READ INPUT SUBTYPE NONE
SEGMENT 0 NODE DEMVSA NCC ID OUTPUT SUBTYPE NORMAL
NAME MODSLV READ RTU ID
AREA MUX ID COS REPORTING Y
BUILDING IN ADD INTERVAL TYPE NONE INTERVAL
UNIT OUT ADD AUTO LAST COMMAND RE-ISSUE N
ON CODE ON OFF CODE OFF INTERMEDIATE CODE MID
ON STATUS 1 OFF STATUS 0 INTERMEDIATE STATUS 0
START VERB START STOP VERB STOP
MINIMUM ON TIME MINIMUM OFF TIME
VERIFICATION DELAY RUN TIME LIMIT POWER DEMAND
ON ALARM N OFF ALARM N INTERMEDIATE ALARM N UNCOMMANDED COS ALARM N
ALARM DELAY CRITICAL ALARM N UNDEFINE STATE ALARM N
ALARM PRINTERS 0 EVENT PRINTERS 0 MESSAGE NUMBER
ACK CATEGORY INTO PRINT CATEGORY ALARMS RELATED TASK
POINT ASSOCIATION NONE ASSOC POINT
POINT ACCESS LEVEL 100 POINT CONTROL LEVEL 100 SLIDE NUMBER
```