QSI HMI

Associated Files:

- QCI-AN057_QSI_HMI.qcp
- Modbus.qly (Written with Qlarity Foundry Version 2.50)

Overview

This application demonstrates how to successfully interface a SilverDust controller and a QSI QTERM G55 HMI with the Modbus RTU Protocol. Modbus® allows for communication between many devices connected to the same network. Modbus® is a registered trademark of Modicon.

Depending on the QSI HMI model used, there may be additional settings or options required. The basic description in this document should allow for successful operation. The following application will describe how to use the Profile Move Continuous command with the QSI QTERM G55 HMI as the host.

For support on the QSI device please contact:
info@qsicorp.com.

For support on SilverDust controller contact:
support@quicksilvercontrols.com.

It is assumed the reader is familiar with programming QuickSilver's servo controllers. If not familiar with our controllers, read Chapter 1 of the SilverLode User Manual.

Hardware and Software Setup

The QSI QTERM G55 HMI DB-9 communication ports are used to program the HMI from the PC and communicate with the SilverDust controller. The application uses COM port 1 to program the HMI and COM port 2 to communicate to the SilverDust controller.

RS-232 Cable Connection

Signal Type	SilverDust SMI Port	QSI G55 Com 1 Port
Tx	Pin 12	Pin 2
Rx	Pin 2	Pin 3
Logic Ground	Pin 8	Pin 5

RS-485 Cable Connection

Signal Type	SilverDust SMI Port	QSI G55 Com 2 Port
A	Pin 12	Pin 2
B	Pin 2	Pin 1
Logic Ground	Pin 8	Pin 5

*RS-485 may vary refer to the HMI Terminal Hardware Manual

SilverDust Programming

Download QuickControl 4.63 (or higher-available from the QCI website http://www.quicksilvercontrols.com) to program the SilverDust controller. The "Profile Move Continuous-QSI.qcp" program is bundled with this application note. The controller's communication protocol and the QTERM G55 HMI protocol require the same communication settings for the two devices to communicate. In this application, the SilverDust controller is initialized with the Modbus® protocol, 57.6K-baud rate, 2-stop bit and no parity bit.
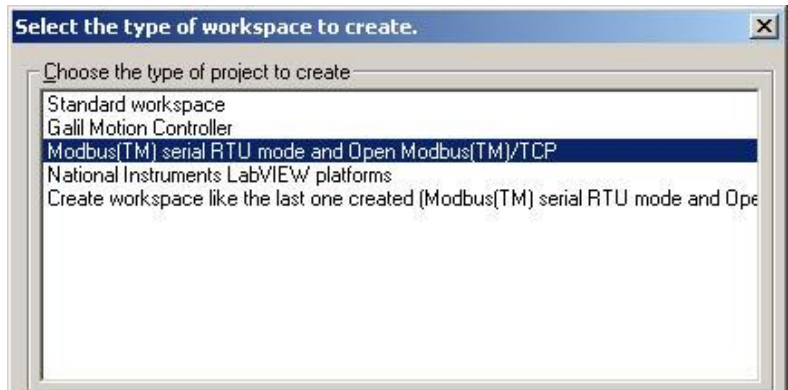
The program uses the ACR command to filter the torque value stored in the lower word of register 9 and copy the value to register 24. Register 24 is then displayed by the HMI. The PMC command uses five designated register for position, acceleration, velocity, deceleration and offset. Enable Multi-Task command allows the program to run the loop condition located at the bottom of the program. If the Enable Multi-Task command is not inserted then the program will remain at line 14 until the PMC command has reached an exit condition. The loop will take the incoming data from the QTERM G55 HMI stored in register 25 and multiple the values by 8000. The value in register 25 will then be copied into the position register 20 and the

Line# Oper	Label	Command
1:EMT		Enable Multi-Tasking
2:REM		filter torque value and store in register 24
3:ACR		Analog Continuous Read: "User or Profile Move Offset[24]" = Torque
4:REM		Writes "0" to the Position Register
5:WRP		Write 0 counts to "User or Profile Move Pos[20]" Register
6:REM		Writes to Acceleration Register
7:WRP		Write 99999 cps/s to "User or Profile Move Acc[21]" Register
8:REM		Writes to the Velocity Register
9:WRP		Write 10000 cps to "User or Profile Move Vel[22]" Register
10:REM		Writes to the Deceleration Register
11:WRP		Write 99999 cps/s to "User or Profile Move Dec[23]" Register
12:REM		Put the servo into a Continuous Profile Move.
13:PMC		Profile Move Continuous:
14:REM		Go into a loop that changes the Postion every second.
15:CLD	LOOP	User or Profile Move Pos[20] = User[25] * 8000
16:JMP		Jump to "LOOP"

motor will move the requested revolution value sent from the HMI.

Qlarity Foundry Modbus Setup

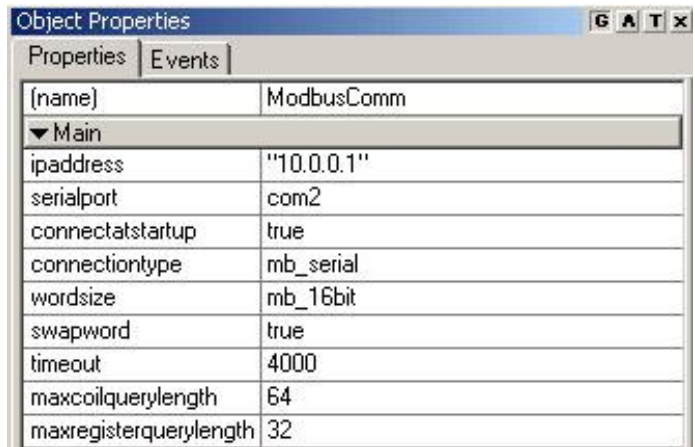The Qlarity Foundry Version 2.5 software can be download from the QSI website http://www.qsicorp.com. The Qlarity program requires a Modbus® application workspace to be created for use with the Modbus® protocol. The program can be created by selecting file-new workspace and selecting the Modbus(TM) serial RTU mode workspace. The example program "Sample_modbus.qly" has the setup already completed. To download the program to the G55 HMI it requires the unit to be set into a download mode. When the HMI is powered down hold any three buttons on the G55 HMI and then power on while holding down the buttons. The HMI will then powered up in a power on mode. The App Mode under the Flash Memory section needs to be changed to download and then Save and Exit. The Qlarity program should be complied and then downloaded to the HMI. For further information please refer to the QSI website for the QTERM-G55 datasheet or contact QSI support at info@qsicorp.com.
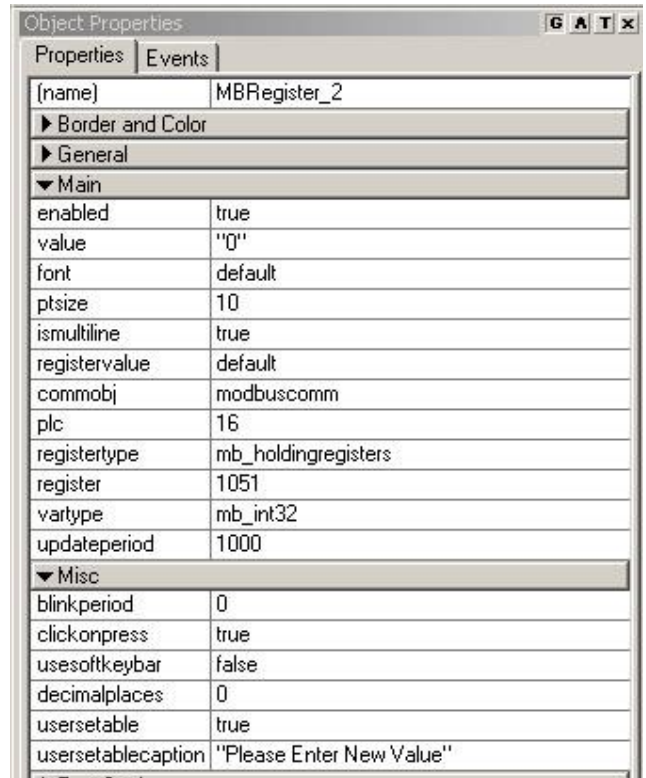
Qlarity Communication Object

The MBComV2 object defines the communication parameters between the G55 HMI and the SilverDust controller. The G55 HMI Modbus® configuration can communicate in either Modbus RTU or Modbus® TCP/IP. In this example, the application uses the Modbus® RTU protocol. The serial port setting is set for COM 2 of the HMI. The word size is set to 16 bits even through the SilverDust registers are 32-bit values. The object receives the 32-bit value as two separate registers. The object defines the low word of the SilverDust register as the high word so the swap word condition is set to true in order for the data to be read and written correctly.
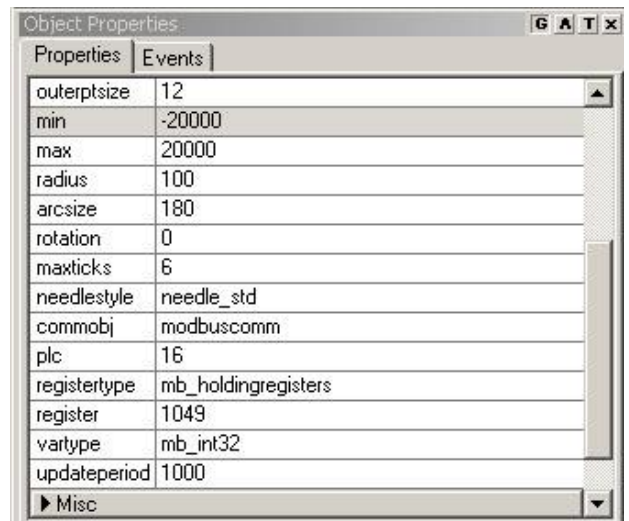
Qlarity Register Object

The MBResgisterV2 object defines the Modbus communication parameters, PLC addressing, register mapping and register size. The Modbus communication object "mobbuscomm" defined previously is called to establish communication parameters for the holding register. The PLC category will be the unit ID number of the SilverDust controller. The register type holding register allows an object to read and write to the SilverDust register. The register is set to read and write to Modbus® logic address 1051. The Modbus® address represents user register 25 of the SilverDust controller. Refer to Application Note QCI-AN038 Modbus® Protocol for additional details about address translation. Since the controller register is a 32-bit register, the variable type is defined as an integer 32-bit value. The object requires the user interface to send numeric values to register 25 so the usersettable object needs to be set to true. The usersettable can be set to false if only a read register is needed.
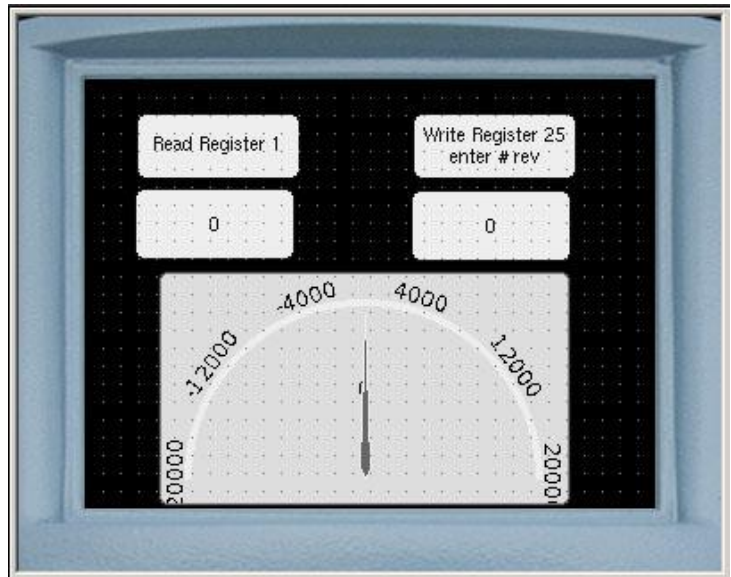
Object Properties	G A T x
Properties	Events	
(name)	MBRegister_2
▶ Border and Color	
▶ General	
▼ Main	
enabled	true
value	"0"
font	default
ptsize	10
ismultiline	true
registervalue	default
commobj	modbuscomm
plc	16
registertype	mb_holdingregisters
register	1051
vartype	mb_int32
updateperiod	1000
▼ Misc	
blinkperiod	0
clickonpress	true
usesoftkeybar	false
decimalplaces	0
usersetable	true
usersetablecaption	"Please Enter New Value"

Qlarity Tachometer Object

The MBTachometerV2 object graphs the value stored in register 24 of the SilverDust controller. The SilverDust controller program uses the ACR command to filter the torque value and stores the value in register 24. The torque value can range from –20,000 to 20,000 SilverLode Torque Value (STU). The max ticks limit the values displayed around the gauge. The update period can be change to either increase or slow the rate the gauge checks for new data.

Object Properties	G A T x
Properties	Events	
outerptsize	12
min	-20000
max	20000
radius	100
arcsize	180
rotation	0
maxticks	6
needlestyle	needle_std
commobj	modbuscomm
plc	16
registertype	mb_holdingregisters
register	1049
vartype	mb_int32
updateperiod	1000
▶ Misc	

Qlarity User Interface

The user interface shows basic functions using Modbus® objects. Using the communication and the register objects the program is able to display the actual position, graph the torque register and write a value to user register 25 of the SilverDust controller. The user interface allows a value to be entered to move the motor a certain amount of revolutions. The move is an absolute move and can be moved back to the original position by typing in zero. The torque meter will change according to the direction and the load on the shaft of the motor.

Troubleshooting Tips

Make sure both the HMI and the SilverDust controller have been setup for the same communication parameters. See the "Hardware and Software Setup" section of this document for the recommended settings. When the HMI communicates with the SilverDust controller the red LED on the SilverDust should flash constantly verifying communication. If this is not the case, double-check the wiring. The SilverDust controller can only communicate with one device at a time, make sure not to communicate with the QuickControl software and have the QTERM G55 HMI simultaneously.