

Application Note #5442

Connecting and Programming a QSI HMI to a Galil Controller

Introduction and Description:

One of the many benefits to Galil controllers is the ability to communicate to 3rd party devices. This is possible because Galil products send and receive commands as standard 8 bit ASCII characters. The standard communication methods that Galil's stand alone controllers support include RS-232, Ethernet and USB. This allows for communication between Galil controllers and a variety of Human Machine Interfaces or HMI's. One popular HMI is the QSI QTERM-G124 or GS124, RS-232/10Base-T touch screen. This HMI is very easy to use, inexpensive and has been customized to interface to Galil stand alone controllers by offering pre-built programming and communication objects. This Application note will describe how to communicate to a Galil stand alone controller through the QSI GS124 HMI. Although this application note describes using the GS124 specifically, any of the QSI's Qlarity based terminals can be used with the predefined Galil programming and communication objects.

The QSI GS124 is a 320x240, 256 color touch screen, which offers RS-232 and Ethernet communication ports. The GS124 has custom firmware designed to work with Galil stand-alone controllers.



Figure 1: QSI GS124 Touch Screen HMI, connected to a Galil DMC-4040 Accelerera Series controller.

The Qlarity Foundry software package is the QSI programming environment. Qlarity Foundry is a Visual Basic-like programming environment which makes programming the interface between the GS124 and a Galil motion controller as simple as configuring and adding predefined objects. Figure 2 shows the basic Qlarity Foundry Layout.

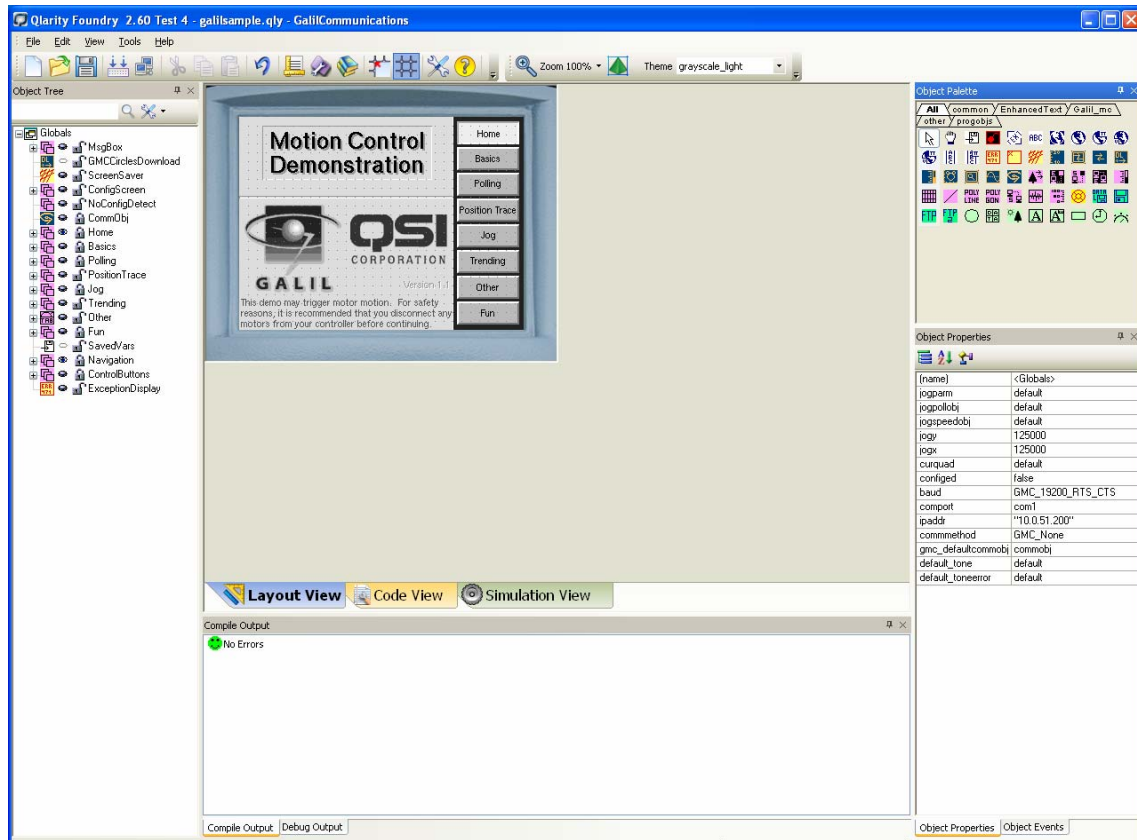


Figure 2: Qlarity Foundry Programming Environment

Connection Procedure and Brief Programming Overview:

The following procedure will review the connection and programming process for a Galil stand alone controller to a QSI GS124. Before starting the connection process it is recommended that the user review the GS70 User Manual and Qlarity Programming guide. The GS124 contains the same hardware as the GS70, but has the additional features used to communicate specifically to Galil controllers. On top of offering instruction and description on how to operate the GS70, the manual will offer many tutorials to help the user get familiarized with the QSI QTERM and Qlarity Foundry environment 1.

Contents

1	Installation of Qlarity Foundry software.....	4
2	Developing and HMI Application Program.....	4
2.1	Running the Galil Demo Program.....	4
2.2	Starting a New Application.....	5
2.2.1	From “new workspace” prompt.....	5
2.2.2	From a “standard workspace”	5
3	Establishing PC to HMI Connection.....	6
3.1	Serial.....	6
3.2	Ethernet.....	7
3.2.1	Direct connection between PC and HMI.....	7
3.2.2	Network Connection.....	8
4	Setup Galil for Communication.....	8
4.1	RS-232.....	8
4.2	Ethernet.....	8
4.2.1	Software.....	9
4.2.1.1	Galiltools/Galiltools Lite.....	9
4.2.1.1.1	No IP Address.....	9
4.2.1.1.2	Existing IP Address.....	9
4.2.1.2	Smart Terminal.....	10
4.2.1.2.1	No IP Address.....	10
4.2.1.2.2	Existing IP Address.....	10
	**Existing IP address conflict, review both options.....	10
5	Connecting QTERM to Galil.....	11

1 Installation of Qlarity Foundry Software

- a. Download the latest version of the Qlarity Foundry package found on the QSI web site at: <http://www.qsicorp.com/support/files/> and install it.
- b. Download the file galil_demo.zip that can be found on the Galil web site at: <http://www.galilmc.com/ftp/pub/appnotes/qsi/> .

Unzip these files into a folder located onto the hard drive of the PC in a location that is easy to find. A recommended location is: C:\Program Files\QSI Corporation\Qlarity Foundry\Galil

2 Developing an HMI Application Program

2.1 Running the Galil Demo Program:

- a. Run the Qlarity Foundry software and open the Galil demo program through the Qlarity terminal, galil_demo.qly, which was downloaded from the Galil site. A screen similar to the one below should be displayed.

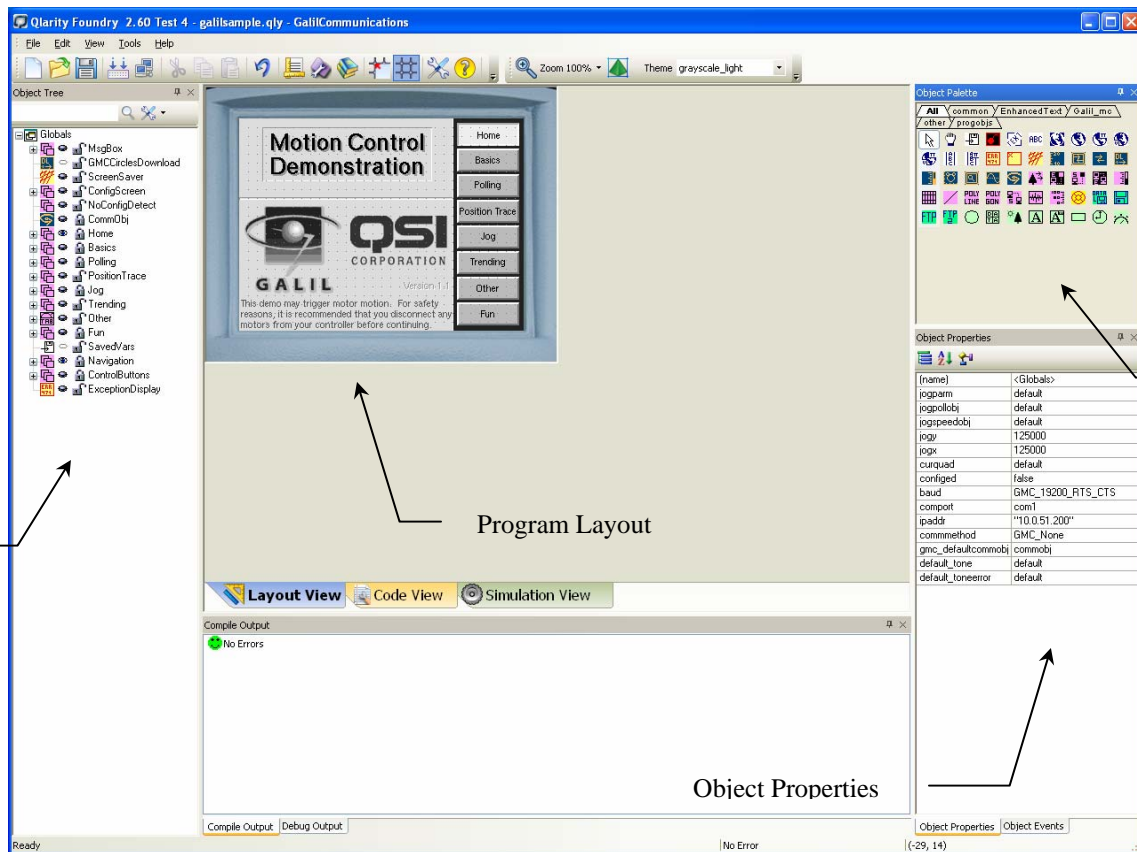


Figure 3: Details of Qlarity Foundry terminal screen.

2.2 Starting a new application:

2.2.1 From a “new workspace”

Open a “new workspace;” either by choosing “new workspace” upon starting Qlarity Foundry software, or by choosing “new workspace” from an open Qlarity window. This can be done by **following**: file>new workspace

- a. Select “Galil Motion Controller” and click finish.

This will automatically add the Galil object libraries installed with the Qlarity software to the Object Palette (black and yellow icons) and add a Galil connection object to the object tree (GalilComm). By highlighting this object in the object tree the connection parameters can be set within the Object Properties window. See Figure 4 for a picture detailing the GalilComm Object Properties Window.

2.2.2 From a “standard workspace”

Starting from “standard workspace” or if additional libraries are to be added upon opening new workspace.

- a. Add libraries
Follow: Edit>Edit Libraries>Add Existing Libraries>[choose specific library and click okay, include GalilComm library]
- b. Select “GalilComm” object from the Object Pallet. Drop this object on the workspace. This will place a new object in the object tree called “GalilComm.” Highlight GalilComm in the object tree and set communication parameters under Object Properties. For details see Figure 4.
- c. A new sheet can be added to the workspace and program development can take place following Qlarity programming instructions.

Object Properties	
(name)	CommObj
commmethod	gmc_ethernettcp
ipaddress	"10.0.51.195"
serialport	com2
serialformat	gmc_19200_rts_cts
maxretries	0
timeout	3000
displayerrors	gmc_displaycommerrors
unsolicitedaware	true

Figure 4: Details of GalilComm Object Properties

For a description of all Galil-specific objects that are available in the Qlarity Foundry software, see Appendix I

3 Establish PC to HMI connection

3.1 Serial

- a. Connect a straight through RS-232 cable from the COM1 dongle provided by the QSI cable to the RS-232 port on the PC
- b. Power down the HMI, if not done so already. Depress the upper left corner (see figure 5) of the HMI screen and power up.
- c. Use the “next” arrow to toggle over to the COM1 settings.
Follow: Flash Memory>Display>Calibration>Network and review the COM1 settings
- d. If the default settings are not appropriate for this application select COM1 and change settings as desired. If the settings are acceptable, ignore.
- e. Configure the HMI to accept the program by changing the “App Mode” setting, under “Flash Memory” to “Download.”
Follow: COM1<Network<Calibration<Display<Flash Memory
- f. Select “Save and Exit” under the “Done” menu
Follow: Flash<Done
- g. The HMI should be ready to accept the program and should display:
 “Firmware Version...”
 “Built...”
 “Download Application...”

3.2 Ethernet

3.2.1 Direct connection between PC and HMI

- a. Connect a cross-over Ethernet cable from the PC Ethernet port to Ethernet port on the QTERM HMI. ^{III}
- b. Note the PC's IP address and subnet mask (this can be done by running ipconfig:

Follow: Start>run>[type]cmd[enter]>[type]ipconfig[enter].

Note: for every field of the subnet mask that is set to 255 the corresponding field of the IP address of the external device eg. the QTERM HMI, must match the IP address of the PC. Entering '0' allows differences in the IP addresses. ^{IV}

Ex. PC; Subnet Mask: 255.255.255.0 IP Address: 169.169.169.1
Device IP Address: 169.169.169.2

PC; Subnet mask: 255.0.0.0 IP Address: 169.169.169.1
Device IP Address: 169.1.1.2

- c. Power down the HMI, if not done so already. Depress the upper left corner (see figure 5) of the HMI screen and power up.
- d. Use the "next" arrow to toggle over to the NETWORK setting.
Follow: Flash Memory>Display>Calibration>Network
- e. Note or set the Subnet Mask on the HMI, following the same rules as above with PC.
- f. Configure the HMI to accept the program by changing the "App Mode" setting, under "Flash Memory" to "Download."
Follow: Network<Calibration<Display<Flash Memory
- g. Select "Save and Exit" under the "Done" menu
Follow: Flash<Done
- h. The HMI should be ready to accept the program and should display:
"Firmware Version..."
"Built..."
"Download Application..."

3.2.2 Network Connection II

- a. It is possible to have the HMI connected to the PC and other devices through a network hub or switch. For this configuration follow the same instructions as in step 3.2.1.a as above except use straight through Ethernet cables and note the IP addresses of all devices on the network. No two devices should have the same IP addresses and all subnet mask(s) must accept all of the relevant IP addresses on the network to communicate. v

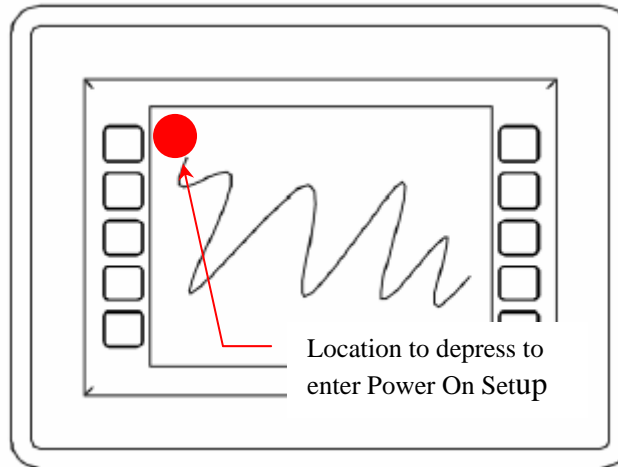


Figure 5: Illustration of where to depress the screen of the QTERM upon power up to reach the 'Power On Setup'.

4 Set up Galil for Communication

This step will require the Galil WSDK/SmartTerminal, or Galiltools/Galiltools Lite software. If not already installed on the PC, these programs are available for download on the Galil website. Be sure to verify software compatibility with the specific controller to be used in this system. <http://www.galilmc.com/support/download.html#windows2k>

4.1 RS-232

If connecting between the HMI and the Galil controller via RS-232 no initialization is necessary. However, note the connection parameters for standard Galil serial connection; Baud rate (set on jumpers or controller default), 8 data bits, no parity, 1 stop bit, hardware flow control.

4.2 Ethernet

1. Note the PC's IP address and subnet mask, if unknown, open the command prompt and run ipconfig.

Follow: Start>run>[type]cmd[enter]>[type]ipconfig[enter]

Note IP address and subnet mask

4.2.1 Software

4.2.1.1 Galiltools/Galiltools Lite

Note: Please see Application Note: 4422 for more details as to how to communicate to Ethernet stand alone controllers.

4.2.1.1.1 No IP address.

- a. Open Galiltools and wait for the connection window to appear.
- b. Choose the “No IP Address” tab at the top of the window and select desired controller, click assign, then wait under the available controllers tab for the controller to appear next to the new IP address.
- c. Establish connection and burn IP address into EEPROM by issuing a “burn parameters” command by entering BN into the command line.

4.2.1.1.2 Existing IP address.

- a. Open Galiltools and wait for connection window to appear.
- b. Note IP address listed for particular controller under “available” tab.
- c. Be sure it does not conflict with any other controllers on the network and will be accepted by all subnet masks. If it is not, then proceed to step **. If this IP address is acceptable highlight particular controller and click “connect” button to establish connection.
- d. If the IP address is not already burned into the controller’s EEPROM, establish communication and burn in IP address with “burn parameters” command or by entering the BN command in the command line,

4.2.1.2 SmartTerminal

4.2.1.2.1 No IP address.

- a. Register controller
Follow: Tools>Controller Registration>Find Ethernet Controllers>[choose controller] click okay> assign IP address.
- b. Return to main screen and establish connection to controller by selecting controller
Follow: Tools>Select Controller>[choose specific controller]>okay
Burn in IP address into EEPROM by issuing a “burn parameters” command or BN in command line.

4.2.1.2.2 Existing IP address.

- a. Open SmartTerminal and select controller
Follow: Tools> Select Controller, highlight the particular controller and note IP address listed for particular controller in Controller Details window with specific controller highlighted.
- b. Be sure it does not conflict with any other controllers on the network and will be accepted by all subnet masks.
- c. If acceptable, choose controller and click okay.
- d. If IP address is not already burned into the controller’s EEPROM, establish communication and burn in IP address with “burn parameters” command or by entering the BN command in the command line.

If the controller has an IP address that will conflict with another component in the system or will be denied by any subnet mask choose one of the two below options.

1.Master Reset the controller by installing MRST jumper and cycling power (once the power has been applied and the red LED has turned off, the jumper can be removed). At this point the controller will be in the “factory default” condition and no IP address will exist in memory. Follow the steps in 4 to assign an IP address.

Warning: A Master Reset clears all parameters, programs and user variables. All data must be backed up before Master Resetting the controller.

2. Connect straight through RS-232 cable from PC to the controller.

- a. Galiltools: open Galiltools and choose appropriate com port and connection speed from the connections window to the specific controller, go to step c below.
- b. SmartTerminal: Open SmartTerminal. **If registered** choose Tools> Select Controller, [highlight specific controller]> [click] okay. **If not registered** select: Tools>Register Controller> [Follow instructions to set up Serial Communication]>[return to main screen], then follow Tools> Select Controller>[select specific controller] >[click] okay
- c. Establish connection and set desired IP address by issuing an IA command. See the controller's specific command reference for a complete description. Burn in IP address into EEPROM by issuing a "burn parameters" command or BN in command line.

5 Connect QTERM to Galil

At this point the QTERM HMI should be able to connect and communicate to the Galil controller and the developed installed program run.

I. One feature of the QSI GS124 worth noting is the Simulation mode with "virtual" ports. This mode will emulate the presence of a controller to allow for preliminary programming and debugging prior to actually being connected to a controller. Another important feature worth reviewing in the user manual is the Object Tree. Much like Visual Basic, this feature allows the user to view and navigate through all of the interface's screens, buttons and objects.

II. The Qlarity Foundry software package contains the Galil Object Library (Galil.mg.lib) and sample program (galil_demo.qly)

III. The Galil Accelera series stand alone controllers (DMC-40x0) offer an "auto-recognition" feature that will automatically configure the communication signals, regardless of the cable type. This means that the DMC-40x0 can use either straight-through or crossover Ethernet cables.

IV. A good rule of thumb is to use the same subnet masks on all components on the network and to simply add 1 to the last sector of the IP address. If the last sector of the IP address is 254 leave it the same and add 1 to the 3rd sector.

V. If the components are connected to a network with a DHCP server the IP addresses may be assigned automatically.

Conclusion

Communication between Galil stand alone controllers and external devices can be done by issuing standard ASCII commands. This allows for communication between many 3rd party HMI's and Galil stand alone controllers. One HMI that takes advantage of this communication protocol is the QSI QTERM-G127 HMI. Using the -G127, coupled with the Qlarity Foundry software package, offers many pre-built, Galil-specific communication objects. This eases the communication and motion programming through a -G127 and a Galil stand alone controller. The above steps offer instruction as to how to communicate to a Galil stand alone controller and develop a motion program through the Qlarity Foundry software package. For more information or assistance with setting up a QSI GS124 with a Galil controller, please contact a Galil Motion Control Applications Engineer at 1.800.377.6329.

Appendix I– Pre-Built Object Descriptions



Object: **GalilCommV2**

Description: Galil motion controller communications object

This object forms the core of the Galil library. This object controls all interaction between the motion controller and the QTERM-G70. It is capable of communicating with a motion controller either serially or via ethernet. To communicate with a motion controller, create an instance of this object and set up its properties as appropriate. Then create other GMC objects and set their commObject property to refer to this object. You can communicate with multiple motion controllers by creating one GalilCommunication object per controller. There is a limit of one controller per serial port on the G70 and a limit of 8 Ethernet controllers that can be used at once.



Object: **GMCCommandDisplayV2**

Description: Executes a command (or series of commands) on the motion controller when you push this button
When the command(s) complete execution, the results are optionally displayed.

This object provides a simple interface to execute an arbitrary command (or series of commands) on your Galil Motion Controller. This object appears as a push button which when pushed, executes the commands defined by the command property, then optionally displays the results in the button.



Object: **GMCCommandExecV2**

Description: Executes a command (or series of commands) on the motion controller

This object provides a simple interface to execute an arbitrary command (or series of commands) on your Galil Motion Controller. Unlike the GMCCommandDisplay object, this object has no graphical interface. It is used to programmatically execute commands on the controller. The ResponseReceived event can be used to process any data returned by the controller in response to your commands.



Object: **GMCGaugeV2**

Description: a vertical/horizontal bar gauge for visual representation

This object implements a flexible bar graph to graphically indicate the level of a floating-point value. The bar graph may be horizontal or vertical. The color of the bar and background are user settable. The value of the gauge may optionally be displayed as text in front of the graph.

 **Object: GMCPATHTraceV2**

Description: Galil motion controller Path tracing object

This object implements a type of 2D graph that can be used to plot the recent positions of the motor. Generally you would be interested in polling the motor position along two of the motor axis (such as the X and the Y axis). This object is robust and can consume a lot of the processing power available. As such, you should minimize the use of other polling type objects on the same screen as you place this path trace object. You can also improve performance, if necessary, by increasing pollInterval, and decreasing numTracePoints.

 **Object: GMCPOLLV2**

Description: Galil motion controller polling object

This object polls the controller with an arbitrary command at regular intervals. It then displays the controller's response. It is useful for displaying values that will change over time, such as the motor position. Many properties are provided which control the appearance of this object, the polling frequency and what is being polled.

 **Object: GMCSimpleTerminal**

Description: A simple terminal for entering interactive commands to the controller.

This object is a simple terminal window used to send arbitrary commands to the controller. You can use this object to compose commands to send to the motion controller using a simple on-screen keypad. Recent commands are saved so that they may be resent. The results of the commands are displayed in the terminal window at the bottom of the object.

 **Object: GMCTrendChartV2**

Description: This object draws a graph of data received

This object implements a line graph that can be used to chart data. Data is assigned to the “newdatapoint” property one point at a time. As new data is assigned, the chart shifts to the left, presenting a rolling history of the last N data points. The number of points on the plot is determined by the width of the chart and the “pointseparation” property.

 **Object: GMCDownloadV2**

Description: An object used to easily download a program to the Galil motion controller

This is a simple object which can either download a program to a Galil Motion Controller or upload an existing program from the controller. To download a program, set the DLProgram property (either in Layout View, or at runtime) then call the Download method at runtime.

The easiest way to add a motion controller program to your workspace, is to include the program as a Binary Resource in Qlarity Foundry. You then use the LoadFromResource method to load the specified binary resource into the DLProgram property. You can then call Download() to send the program to the controller. You can also set the DLProgram property by calling the LoadFromFile method at runtime to load a program that has been saved to the terminal's flash file system.

To upload a file, simply call the Upload method at runtime. This will set the ULProgram property. You can then either examine the value of this property, or call the SaveToFile method to save the uploaded program to a file.



Object: GMCTachometerV2

Description: Analogue gauge for displaying data.

This meter is an analog dial gauge which allows the user a way to quickly glance at the status of a value.