



## Application Note #5493

---

### Connecting an RIO to a Fuji HMI via Modbus TCP/IP

---

#### Introductions:

The RIO Pocket PLC is a versatile device that can be used to control a variety of processes, perform multi-threaded operation, or simply extend the I/O capabilities of a machine. In this application note, we will describe how to connect an RIO controller as a Modbus slave to a Fuji HMI and how to establish basic communication between the two products.

#### Materials:

- RIO-47122 Pocket PLC (any RIO model will work)
- Fuji Monitouch V8 series HMI
- V Series Editor Software for Windows (V-SFT 5V)\*
- Galiltools software\*
- Ethernet hub for connection of a PC to the HMI and RIO \*

*\* For development work only. Final system can connect the HMI directly to the RIO*

#### RIO Setup:

Start by powering up the RIO and connecting it and the PC together using the Ethernet hub. Assign the RIO to an IP address using the Connection window of Galiltools. In this example, we assign it an IP address of 192.168.2.115 on a static network. Other values may be appropriate depending on the IP settings of your computer, and whether it is a static network or a DHCP enabled network. See the RIO user manual for more instructions on setting the RIO IP address. <http://www.galilmc.com/support/manuals.php>

Once you have done this, open a connection to the controller. Now, send the following commands to the Galil controller to initialize it to communicate properly to the Fuji HMI:

```
MV1  
MW1  
MI1  
ME1
```

Once this is completed, enter the BN command to burn in these parameters to the Galil controller, and close Galiltools.

#### HMI Setup:

Start by powering up the HMI and setting it to a correct IP address. Please refer to the HMI user manuals for instructions on assigning the IP address directly on the HMI setup screen. In our example, we set the HMI to the IP address of 192.168.2.100. Connect the HMI to the PC via the Ethernet Hub.

Next, install the V-SFT 5V software from Fuji. Once installed, start the software and choose to start a New Project. Select the model HMI you are using for your application in the Edit Model Selection window (in our case, we are using the V808iC). Now click OK to be brought to the PLC selection window. Under 'Device' select PLC for the device we are connecting to. Under the 'Maker' tab select Others at the bottom. Under 'Series', select Modbus TCP/IP(Ethernet). Select OK once you are finished.

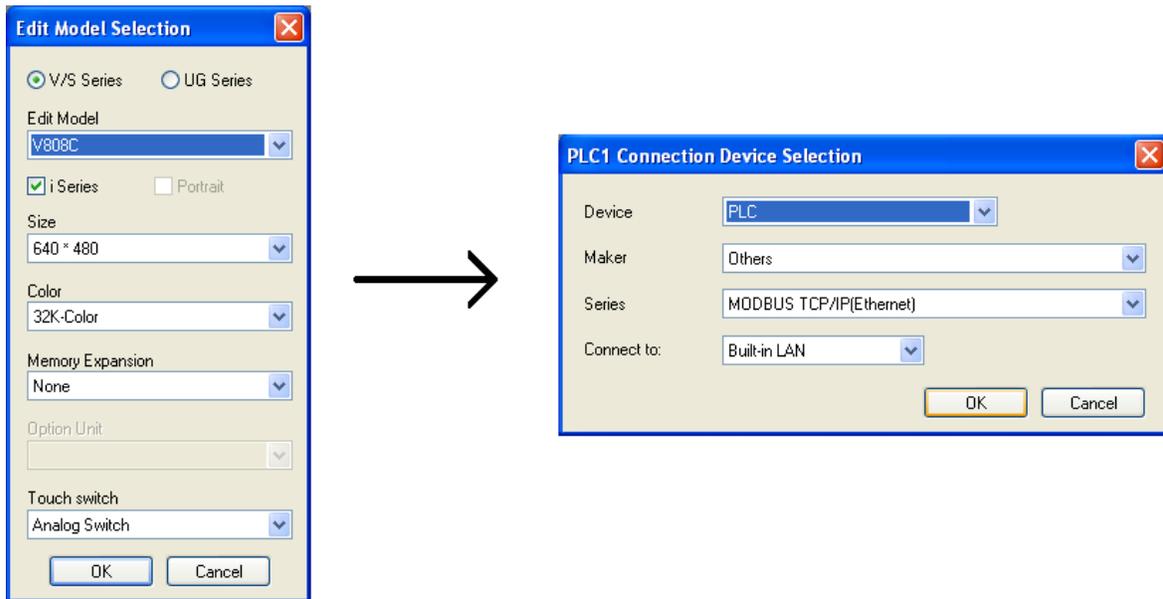


Figure 1: Setup windows selection used

Once you finish selecting the device, you will be brought to the 'Device Connection Setting' windows. This sets up the details of communication between the HMI and the RIO device. Click on the 'Target Settings' tab and double-click on the PLC table under 'Port Name' and name the port 'RIO'. Set the IP address to the desired IP address of the RIO (in this case, we will set it to 192.168.2.115 as that is what we set the IP address of the RIO to). Leave the default Port number of 502 as this is the standard Modbus port.

Next click on the 'Format Setting' tab. The table row marked as 'Modbus Free' defines how the Modbus commands are sent. In order to set this up correctly, we need to adjust how the HMI sends register commands. Scroll the table window to the right until you see the 'Write Holding Register' column. Double click on the first element in that column and change the window to '2-word'. See the Format Setting screen below for an example of this setting.

Once that is changed, close that window. You should now see a blank screen depicting your HMI screen. On this we will begin to add components to visualize the data being received from the RIO.

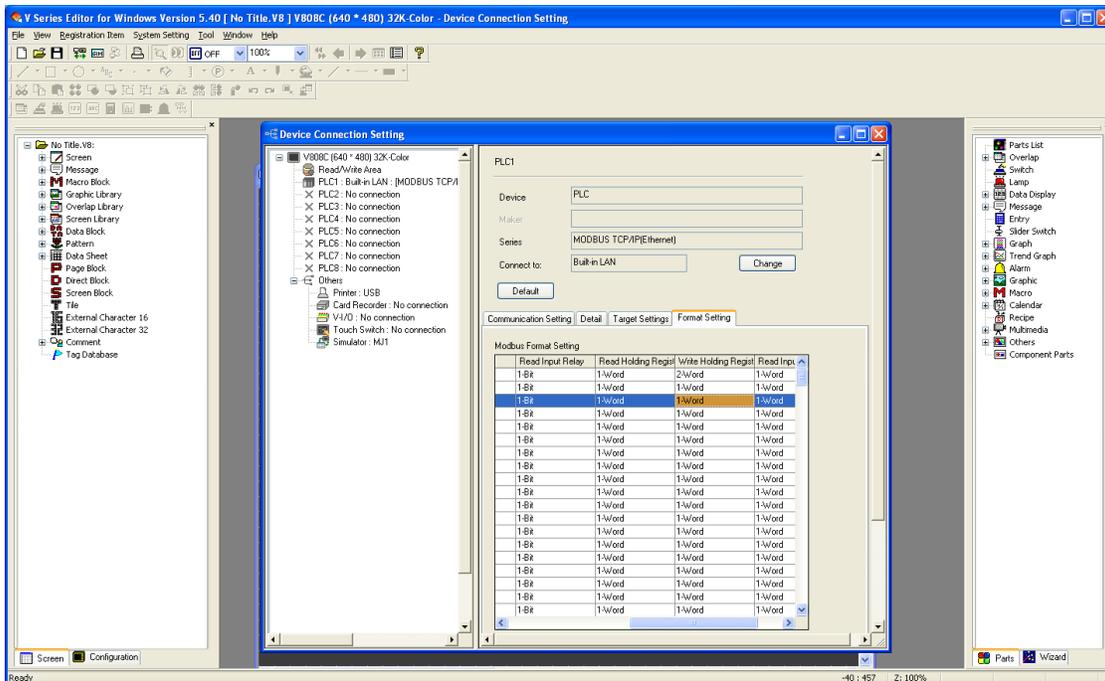


Figure 2: Device Connection Settings window

### Setup of IO:

We will setup the IO reading on the HMI in the following way for each type of IO point

**Digital Outputs:** The Parts List should be on the right-hand side of your window. Click and drag 'Switch' and place a switch on your screen space. Double click on that switch to bring up the Main switch options. In this window, you define what the switch is going to control. In this case, we want to have the switch toggle Output 0 on the RIO.

To do this, first make sure to check the 'Output Memory' box. Then under the first dropdown box, select our 'PLC1'. This is the RIO address. Leave the first arrow box as '0'. Go to the second dropdown box and select '0' to reference the coil addressing. Go to the second arrow box and type in '1'. For output action, select 'Alternate'. Leave everything else at the default value. The box should look as shown below.

Now, go to the Part List and drag and drop a 'Lamp'. Double-click on the lamp to bring up the Main Lamp options. Make the Lamp memory match the address given to the output memory of our switch we just created to have the lamp indicate the status of the output. See below to see how they look.

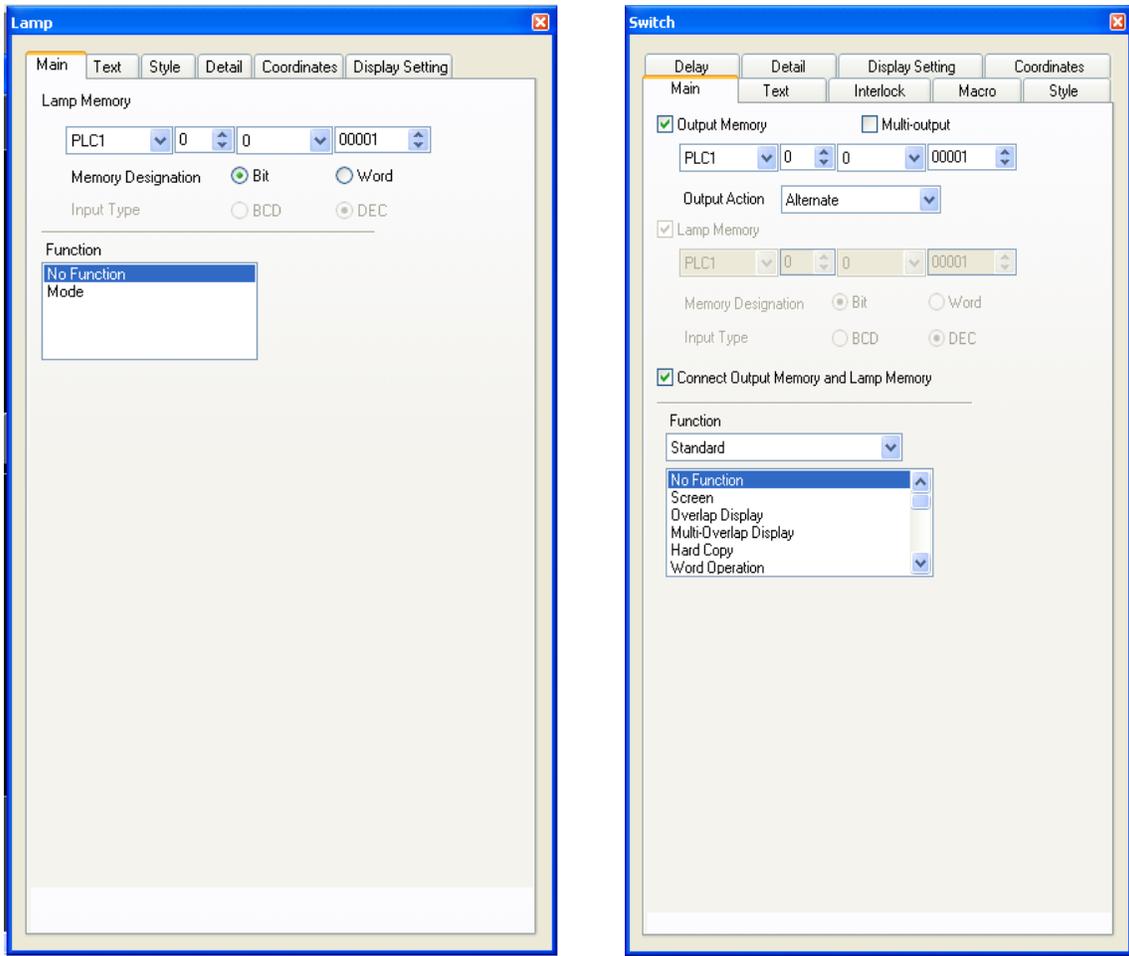


Figure 3: Setting up the Digital output lamp (left) and switch (right)

**Digital Inputs:** Drag and drop another 'Lamp' part onto the screen and double-click it. This will bring up the new Lamp's Main options. Here we are going to display the status of input 0 of the RIO. Since there is no setting the input via a command, all we need is a lamp for display purposes. For addressing, follow the same steps as above, but for the second drop-down menu, select '1' instead of '0'. This indicates to read address memory.

After this is done, click the 'Text' icon on the top toolbar of the window and go ahead and place a label next to your two lamps labeling them as 'Output 0' and 'Input 0' in order to differentiate them. Your screen should look something like the following figure.

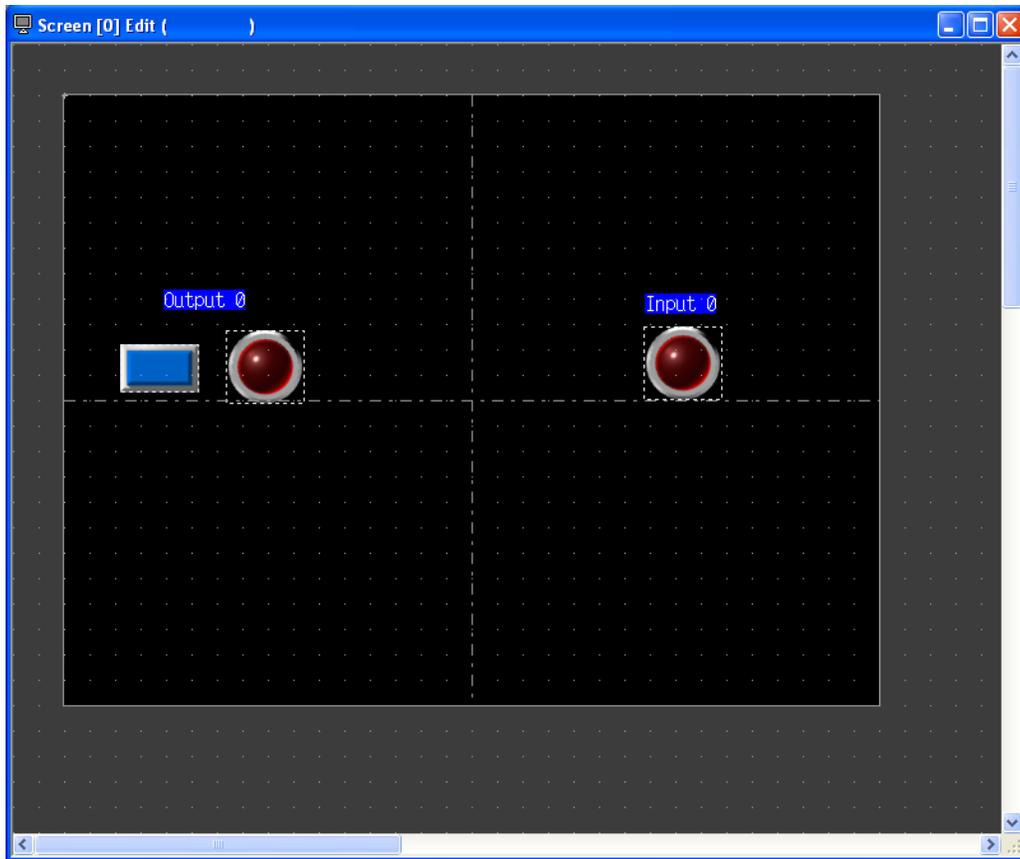


Figure 4: Initial screen for digital inputs and outputs

Analog Inputs: Analog inputs can be read on the HMI screen easily using a numerical display box to show the value. Click on the Data Display -> Num Display in the Parts list and drag-and-drop one onto your HMI development screen. Once it is in place, double click the box to bring up the Num. Display Main options. Under Memory, select 'PLC1' from the 1<sup>st</sup> dropdown box. Leave the port number alone. In the second dropdown box, select 3 to denote that we are accessing input registers. We are going to read analog input 0, so in the right-most box and type in 1 for the input address. Leave all other settings alone and close the window.

Analog Outputs: We are going to follow the same steps as we did for setting up an analog input box. However, once we reach the second dropdown address box, we will select a 4 instead to indicate we are accessing a holding register. Type a 1 into the rightmost box to indicate that we are reading analog output 0. At the 'Function' window, select 'Entry Target'. Click the checkbox that appears labeled 'With Entry Key'. You will need to register a keypad to the overlap library if you have not done so previously. Click on Register and select a keypad to use for entering analog output data on the HMI screen. Once you select one, double click it to set it in the Overlap Library number you have in your setup window (it should be 0 by default).

Check the box for 'Display Position' and select 'Specify with Mouse'. This will allow you to place the keypad that will display when you press the data box on your screen. Select a spot for the top-left corner of the display. Once you are done, close the screen. The setup windows for Analog inputs and outputs should look like this:

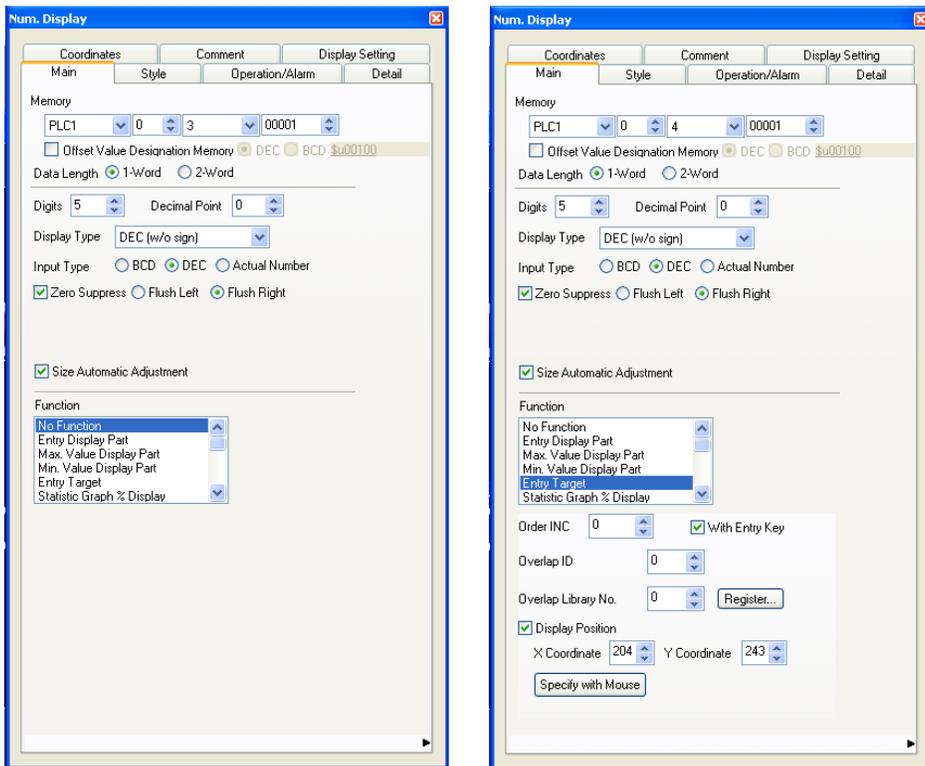


Figure 5: Analog Input (left) and Analog Output (right) setup

Place a label each data box with the 'Text' button from the top toolbar labeled 'Analog Input 0' and 'Analog Output 0' in order to distinguish your data boxes apart. Your screen should look something like this.

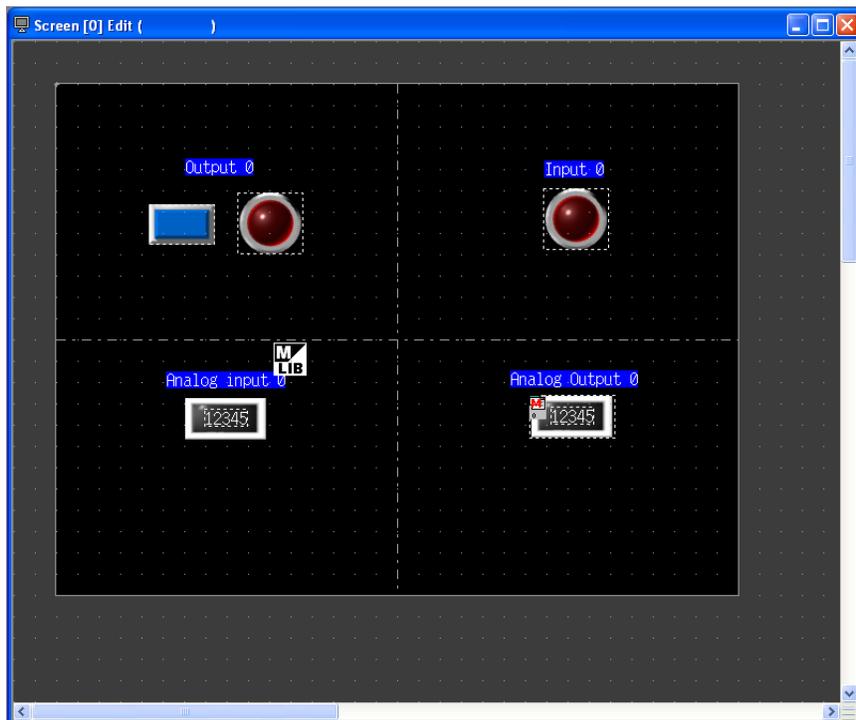


Figure 6: Screen Example with analog and digital I/O displayed

### Array Read/Write:

Modbus Array Read/Write is available on RIO modules with rev. 1.0d firmware or newer. In order to enable this feature, ME1 must be set on the controller. Once that is set, array values can be read and written to the RIO in the same manner as analog outputs. The addressing for array reads on the RIO start at address 1000 and count from there. So, if you want to read and write from array space 0, in the setup window in the right-hand address box for a data display, type in the address 1001. All other setup values remain the same as for other analog outputs.

For more information on reading and writing to array space in the RIO using Modbus commands from a Modbus master, see the ME command in the RIO command reference. <http://www.galilmc.com/support/manuals.php>

### Transferring the Project:

Once you are done setting up the screen how you would like, select File -> Transfer. Make sure to select the communication port that you would like to use to transfer the project using the 'Communication Setting..' box. *You must ensure that the subnet mask for both the HMI and the PC allow communication to one another. Confirm that the PC and the HMI share the same subnet mask to ensure communication success.*

Once you have set the communication up, select the 'PC ->' button to transfer the project to the HMI. If everything is correct, the software will indicate that the download is complete. At this point, the HMI should reset and start running the application you created. Both the RIO and the HMI should at this point be configured correctly and sitting on the network. If everything was configured correctly, the screen should display on the HMI with data updating on the screen.

For additional support questions regarding connection of an RIO to a Fuji HMI, please contact Applications support at Galil Motion Control.

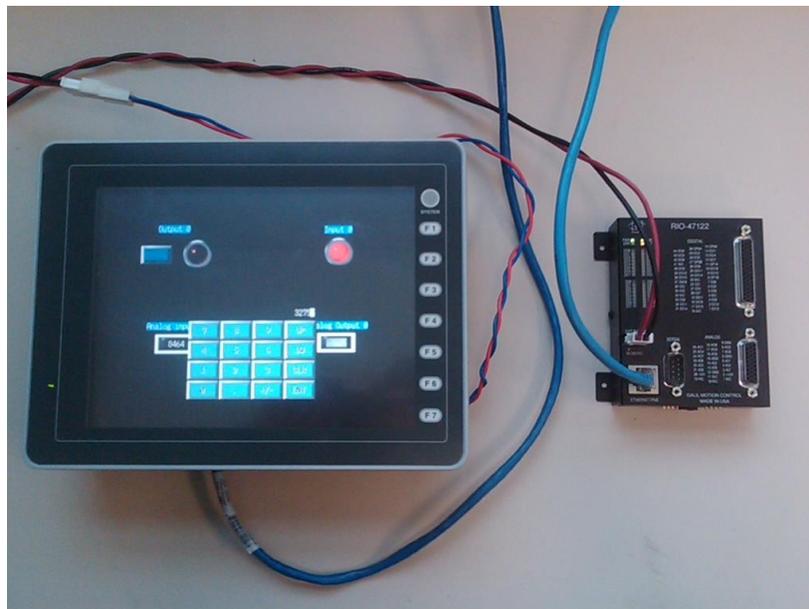


Figure 7: HMI and RIO in operation from example application