

QUICKSTART GUIDE:

THE RIO:

RIO is a programmable remote I/O controller. The RIO is programmed using the same logic commands as any Galil DMC (Digital Motion Controller) with the exception of a few revised commands.

Communication with the RIO utilizes the same software programs needed to communicate to any DMC, such as the DMC Smart Terminal or Galiltools. Interrogation commands have been included to allow a user to instantly view the entire I/O status, thread execution, or Ethernet handle availability.

WHAT IT NEEDS TO WORK:

To begin using the RIO unit, a communication cable (Ethernet or serial), a power source (PoE, or external DC supply), and a computer are required. Proper software should be installed on the computer, however other communication programs such as HyperTerminal or Telnet work with limited functionality.

Installation of a complete, operational RIO system consists of 4 steps:

1. Configure Jumpers:

AUX vs PoE – this selection allows power to be drawn from an external 18-36 volt power supply, or from power over the Ethernet (PoE) cable.

Baud Selection – This jumper allows the user to select the serial communication baud rate.

OPT – This jumper forces the RIO to bypass the Ethernet auto-negotiate feature and forces 10 base-T with half duplex.

2. Connect power to the RIO

AUX: 18-36VDC power input is the default configuration. The power supply should be capable of delivering up to 4 Watts.

PoE: The controller will derive its power directly from the Ethernet cable. A PoE style switch can be used such as the FS108P from Netgear.

3. Install the communications software on host PC

GalilTools: It is strongly recommended to use the Galil software “GalilTools” when communicating to the RIO unit. Please see the GalilTools Manual for a complete description of how to install and connect to Serial or Ethernet controllers.

4. Establish communications between the RIO and the host PC

RS-232: To connect the RIO and a computer via serial RS-232, a 9pin straight-through RS-232 cable (CABLE-9-PIND) is required

Ethernet: Connect the RIO Ethernet port to your computer via a crossover Ethernet cable, or to a network hub by a straight through Ethernet cable.

I/O:

The standard RIO controller has 16 digital inputs, 16 digital outputs, 8 analog inputs and 8 analog outputs. Access to I/O points is made through the 44pin and 26pin High Density D-Sub connectors on the top of the unit. The I/O channels are broken into 5 different categories:

I/O Type	I/O	Description
High Power Sourcing Outputs	0-7	Opto-isolated sourcing power outputs capable of delivering 500mA per channel
Low Power Sinking Outputs	8-15	Digital Outputs 8-15 are opto-isolated (25mA max) sinking outputs
Digital Inputs	0-15	Digital inputs 0-15 are opto-isolated inputs with a range of 5-24VDC.
Analog Outputs	0-7	Analog Outputs have a voltage range of 0-5VDC, and can sink or source up to 5mA of current.
Analog Inputs	0-7	Analog Inputs 0-7 have a voltage range of 0-5VDC.

CONNECTING TO THE RIO:

The RIO supports both TCP/IP and UDP/IP protocols. The RIO can be assigned an IP address three different ways:

Galil Software: It is recommended that GalilTools be used when connecting to the RIO unit. Galil tools will automatically detect the RIO controller on the network. If the network is unmanaged, then the user must assign an IP address.

DHCP Server: The RIO defaults to DHCP method of IP address assignment. If a DHCP server is not present, GalilTools may be used to assign an IP address over Ethernet. Alternatively the IP address may be assigned over serial port with the IA command.

IA: A user can set the IP address by using the IA command through the RS-232 port.

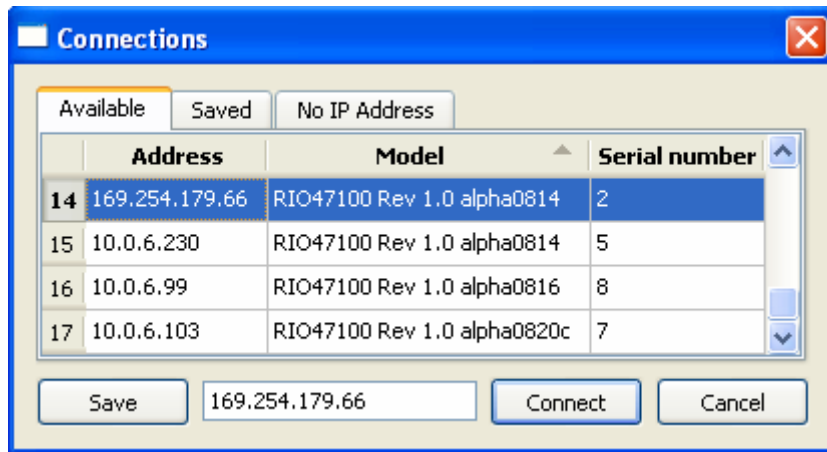


Figure 1: Connection dialog for managed network. The controller has already been assigned an IP address.

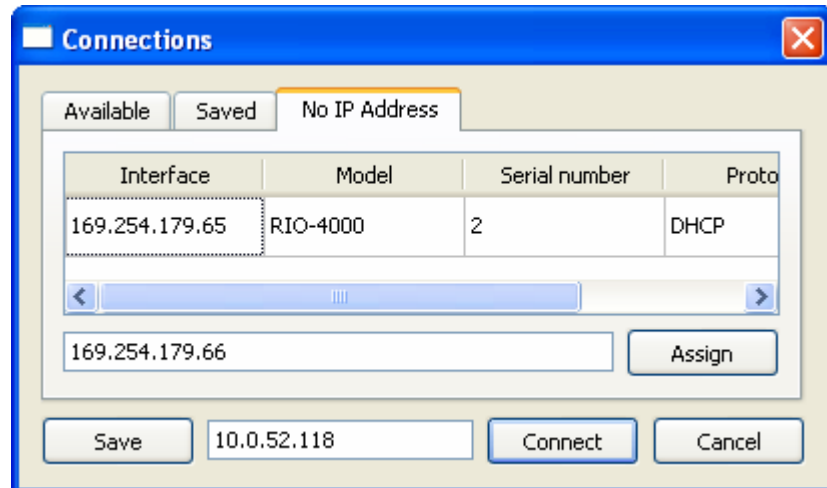


Figure 2: Connection dialog for unmanaged network. The user is prompted to assign an IP address

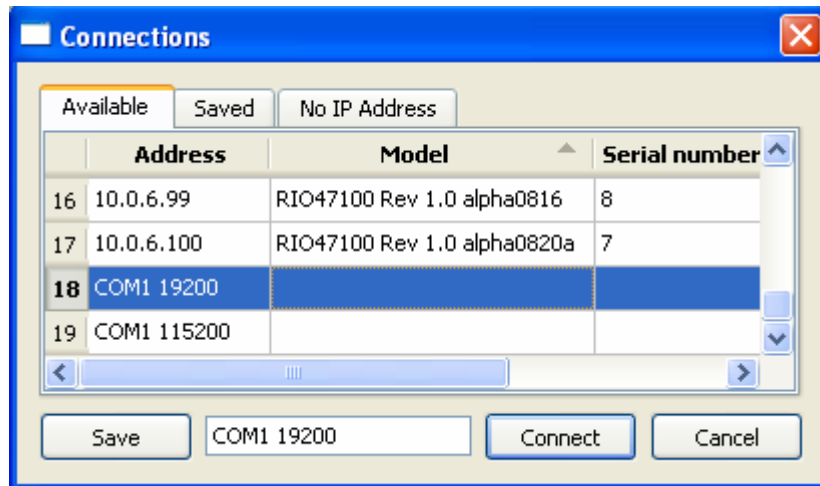


Figure 3: Connection is made over serial and IA is used to assign the IP address.

WATCH WINDOW:

The watch window contains parameters available from the RIO's data record. Boolean values are colored uniquely; a 1 is green, and a 0 is red. By double-clicking any of the outputs (@OUT), the output points can be toggled. Each column can be sorted by clicking on the title header.

Name	Value	Units	Description
@IN[06]	1	boolean	Digital input 6
@IN[07]	1	boolean	Digital input 7
@IN[08]	1	boolean	Digital input 8
@IN[09]	1	boolean	Digital input 9
@IN[10]	1	boolean	Digital input 10
@IN[11]	1	boolean	Digital input 11
@IN[12]	1	boolean	Digital input 12
@IN[13]	1	boolean	Digital input 13
@IN[14]	1	boolean	Digital input 14
@IN[15]	1	boolean	Digital input 15
@OUT[00]	0	boolean	Digital output 0
@OUT[01]	0	boolean	Digital output 1
@OUT[02]	0	boolean	Digital output 2
@OUT[03]	0	boolean	Digital output 3
@OUT[04]	0	boolean	Digital output 4
@OUT[05]	0	boolean	Digital output 5
@OUT[06]	0	boolean	Digital output 6
@OUT[07]	0	boolean	Digital output 7
@OUT[08]	0	boolean	Digital output 8
@OUT[09]	0	boolean	Digital output 9
@OUT[10]	0	boolean	Digital output 10
@OUT[11]	0	boolean	Digital output 11
@OUT[12]	0	boolean	Digital output 12
@OUT[13]	0	boolean	Digital output 13
@OUT[14]	0	boolean	Digital output 14
@OUT[15]	0	boolean	Digital output 15
IN	0	boolean	IN command waiting for user input
TIME	54618	samples	Sample counter (TIME & \$FFFF)
TR	0	boolean	Trace on
XQ	0	boolean	Program running
_EO	1	boolean	Echo on
_OP0	0		Digital outputs 0-7
_OP1	0		Digital outputs 8-15
_PC	0	edges	Pulse counter on digital input 3
_TC	0		Error code
_TIO	255		Digital inputs 0-7

Figure 4: The watch window

TERMINAL WINDOW:

The terminal allows commands to be sent to the controller from the input textbox, and can be recalled from the drop down menu at the bottom. The terminal window is shown in figure 5.

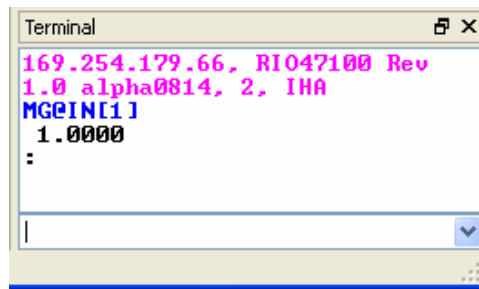
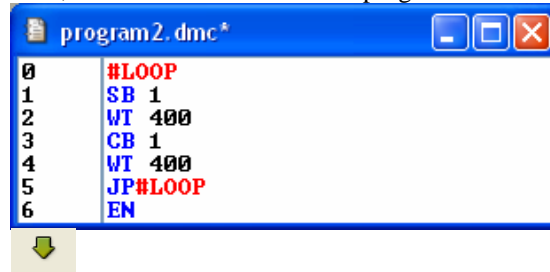


Figure 5: The Terminal window

Example program: Blinking LED

To blink an LED, enter the shown text in the program editor window:



Then hit the Execute button, which will download and execute the program. This will toggle output 1, and the corresponding LED will blink