Features:

- Remote Analog Input Module with Web Access AJAX and Modbus TCP
- One 10/100 Ethernet port
- 8 channels 16-bit A/D
- Isolation up to 2500Vrms
- One channel relay output port
- Form A or form B relay with contact rating 30VDC@1A or 125VAC@0.5A
- Support Web-based I/O control
- DIN Rail mounting
- Windows configuration utility included

Specification:

Ethernet:

10/100 Mbps, RJ45
Protection: 1500V Magnetic isolation
Protocol: Modbus/TCP, UDP, HTTP, DHCP

Isolation analog input:

Channel number: 8
Input type: Differential input
Input mode: Voltage/Current (0~20mA)
Resolution: 16-bit
Input range:
Unipolar: 0~150mV, 0~500 mV, 0~1V, 0~5V,0~10V
Bipolar: +/- 150mV, +/-500mV, +/-1V, +/- 5V, +/- 10V
Current: 0~20mA
Input impedance: 20M Ohm (voltage), 120 Ohm (current)
Accuracy: +/- 1% FSR
Isolation: 1500VDC

Relay output:

Channel number: 1
Contact rating: 30VDC@1A or 125VAC@0.5A

Power: 9~48 VDC terminal block

Protection: Auto polarity and surge protect

Dimension: **108x78x25mm (HxWxD)**

Operating Temperature: 0~70°C

Storage Temperature: -20~85°C

Packing List

1. RIO-2070
2. Software utility download from Artila Web

Layout

Power Connector

Connecting 9~48VDC power line to the Power in terminal block. If the power is properly supplied, the Power LED will keep solid green color and a beep will be heard.

LED Status

The LED provides the RIO-2070 operation information. The LED status is described as follow:

Power LED: Power LED keeps ON if power (+9VDC to +48VDC) is correct.

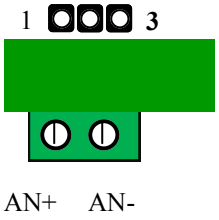
Ready LED: Ready LED keeps ON when RIO-2070 firmware is ready for operation.

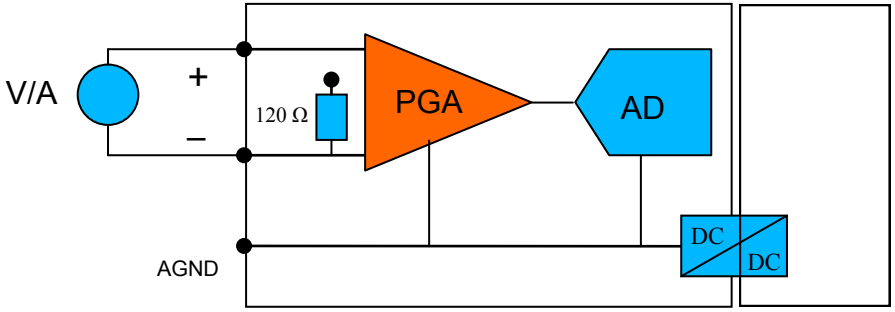
LAN LED: Link and Activity LED will turn ON when the Ethernet cable is connected. When there is network data traffic, this LED will flash.

Input mode selection jumper (JP4~JP11)

To configure the voltage or current input, users need to open the metal case to set the jumper to proper position.

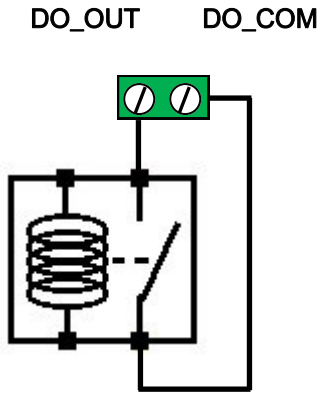
Voltage Input: Short 1-2 (Default setting)
Current Input: Short 2-3 (a 120 Ohm resistor in shunt with +/-)





Relay output connector (DO_OUT, DO_COM)

The relay provides normal open output as shown. It can switch voltage source up to 30VDC@1A or 125VAC@0.5A

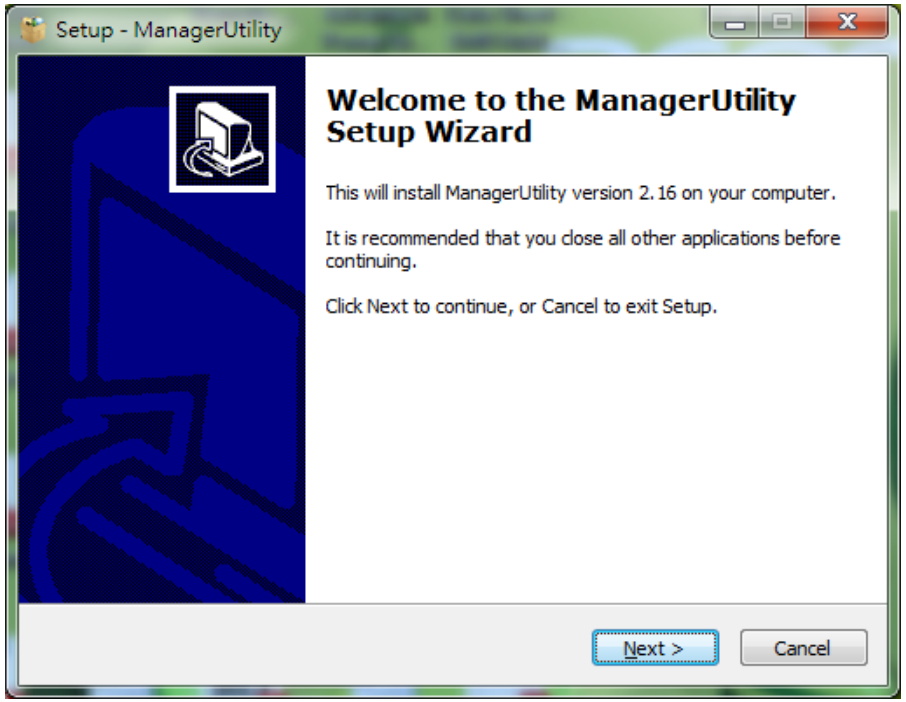


Factory default setting

IP Address: 192.168.2.127
Netmask: 255.255.255.0
Modbus mode: Modbus TCP
Port number: 502
Relay output: Normal open
Web server: enabled
Home page: http://192.168.2.127:5003
Telnet console: telnet 192.168.2.127 5001

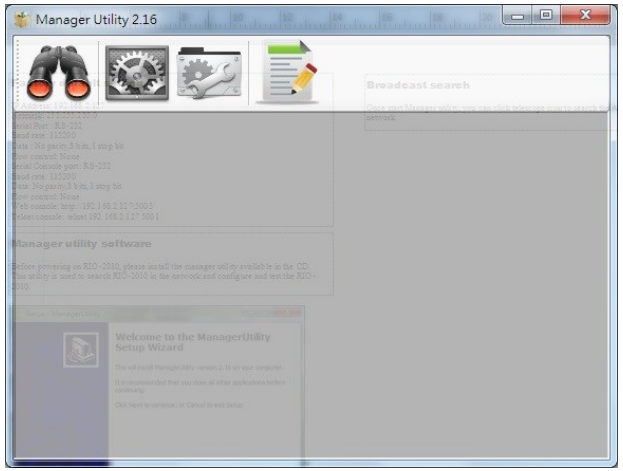
Manager utility software

Before powering on RIO-2017, please install the manager utility available from Artila Website download section. This utility is used to search RIO-2017 in the network and configure and test the RIO-2017.



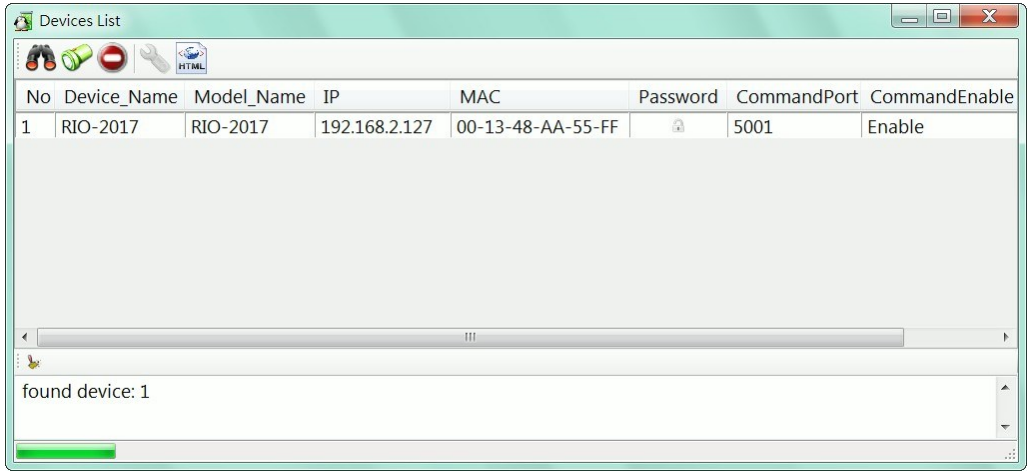
Broadcast search

Once start Manager utility, you can click telescope icon to search the RIO-2017 in the network.



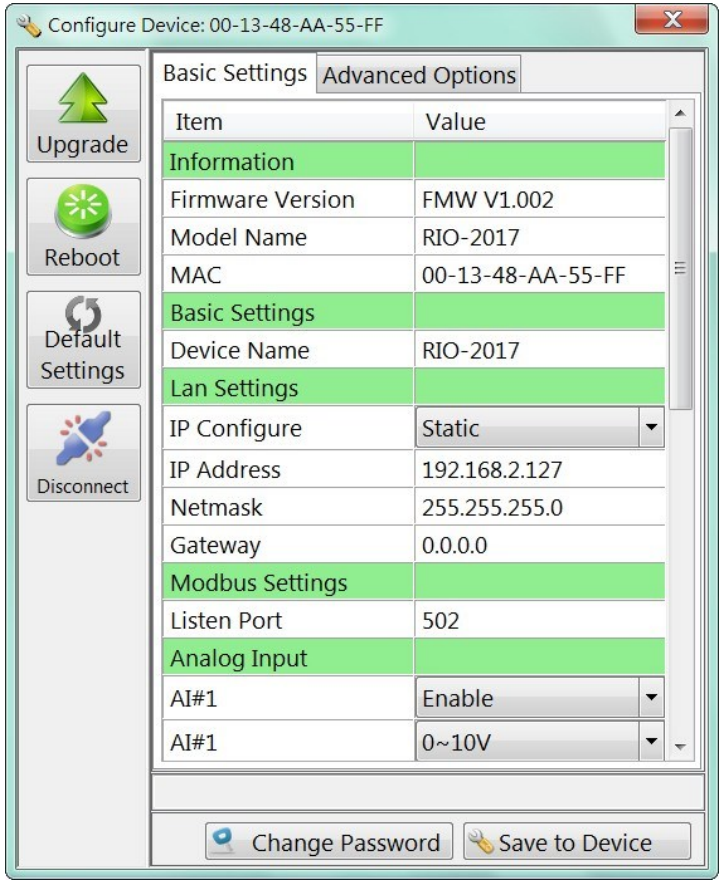
Configure RIO-2017

Once RIO-2017 is discovered, Manager will show following information.



Basic settings

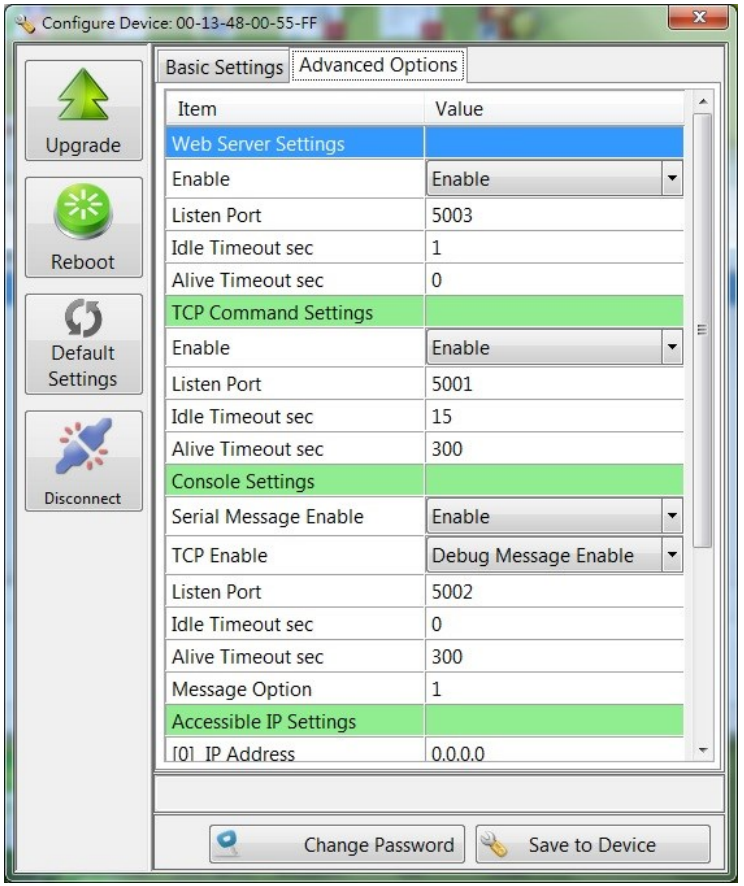
Click the RIO-2017 will open the windows to configure. The Basic settings allows user to configure following settings:



Device name: user configurable device name
IP Configure: Static IP or DHCP
IP Address: specify IP address
Netmask: Netmask settings
Gateway: Gateway IP address
Listen Port: Modbus Port number
AI#: Analog input range setting

Advanced options

The Advanced options allow user to configure following settings:



Web Server Settings
Enable: Enable or Disable Web server
Listen Port: Web server port
Idle Timeout sec: disconnect connection while no data on line and time out occur
Alive Timeout sec: disconnect connection while no data on line, time out and no response to Ack signal

TCP Command Settings
Enable: Enable or Disable TCP command port
Listen Port: TCP command port number
Idle Timeout sec: disconnect connection while no data on line and time out occur
Alive Timeout sec: disconnect connection while no data on line, time out and no response to Ack signal

Console Settings
Console setting is used for RIO-2017 designer to perform system debug. Currently it is not open for user's application

Accessible IP Settings
RIO-2017 provides access control functionality. User can configure the IP address and Netmask range and masters only with these IP address can access RIO-2017.

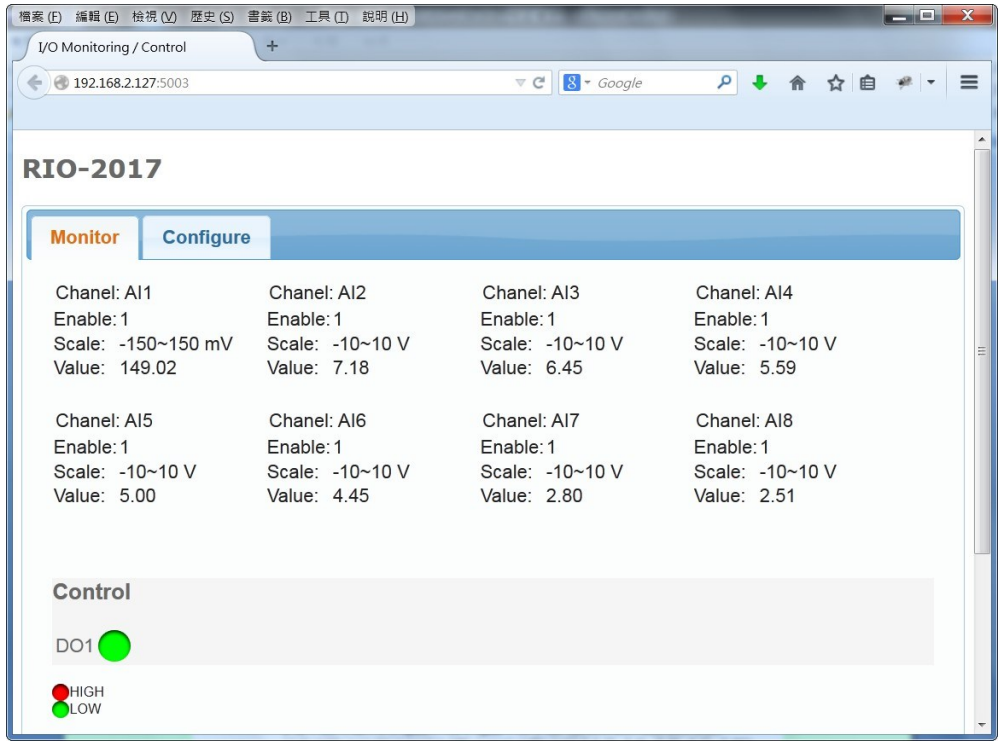
How to access the data of RIO-2017

There are three way to access RIO-2017

1. **Modbus**: user can use Modbus TCP to access RIO-2017. Using the holding register of RIO-2017, user can read the data of analog channels and control the relay On/Off. The register format of RIO-2017 is available in the appendix
2. **Web application**: RIO-2017 support AJAX interface. It is designed for user to develop Web based application. A demo web page is available for your reference.
3. **X86 and Matrix ARM Linux API**: For users who want to develop their own application software using C, they can use AIO library which is bundled with RIO-2017. Please refer the on line help of the API for the information of using the AIO library.

Web based I/O control

In addition to Modbus TCP, user can also use Web port to access data and information of RIO-2017. RIO-2017 uses AJAX scheme to read and write I/O of the RIO-2017. Use GET request together with command parameter, you can retrieve data and information from the web server of RIO-2017. Use mouse to click the DO icon can trigger DO on/off.



AJAX Command

1. **GET ALL Value:** To get all value and settings of analog and digital channels of RIO-2017, you can use

GET URL:port/action/fetchValue?all_val=0

For example

GET http://192.168.2.127:5003/action/fetchValue?all_val=0

The response data in **JSON** format as follow

GET http://192.168.2.127:5003/action/fetchValue?all_val=0&_=1402648090538 200 OK 211ms

參數 標頭 回應 XML JSON


```
{
        "DOVal":[
                {"name":"DO1","val":"L"}
        ],
        "AIVal":[
                {"id":"AI1","en":"1","rgmin":"-150","rgmax":"150","val":"148.99"},
                {"id":"AI2","en":"1","rgmin":"-10","rgmax":"10","val":"7.18"},
                {"id":"AI3","en":"1","rgmin":"-10","rgmax":"10","val":"6.43"},
                {"id":"AI4","en":"1","rgmin":"-10","rgmax":"10","val":"5.58"},
                {"id":"AI5","en":"1","rgmin":"-10","rgmax":"10","val":"4.99"},
                {"id":"AI6","en":"1","rgmin":"-10","rgmax":"10","val":"4.44"},
                {"id":"AI7","en":"1","rgmin":"-10","rgmax":"10","val":"2.78"},
                {"id":"AI8","en":"1","rgmin":"-10","rgmax":"10","val":"2.50"}
        ]

}
```


DOVal:

name: DO channel number
val: H (High), L (Low)

AIVal:

id: AI channel number
en: 1 (Enable), 0 (Disable)
rgmin: minimum value
rgmax: maximum value
val: current value

2. **GET Analog Input Range** settings can be done by command

GET URL:port/action/fetchValue?all_val=0

For example:

GET http://192.168.2.127:5003/action/fetchAIcfg?all_val=0

The response data in **JSON** format as follow

GET http://192.168.2.127:5003/action/fetchAIcfg?all_val=0&_=1402648360426

參數 標頭 回應 XML JSON


```
{
        "CfgSelect":[
                {"name":"en_0","val":"1"},
                {"name":"rg_0","val":"0"},
                {"name":"en_1","val":"1"},
                {"name":"rg_1","val":"8"},
                {"name":"en_2","val":"1"},
                {"name":"rg_2","val":"8"},
                {"name":"en_3","val":"1"},
                {"name":"rg_3","val":"8"},
                {"name":"en_4","val":"1"},
                {"name":"rg_4","val":"8"},
                {"name":"en_5","val":"1"},
                {"name":"rg_5","val":"8"},
                {"name":"en_6","val":"1"},
                {"name":"rg_6","val":"8"},
                {"name":"en_7","val":"1"},
                {"name":"rg_7","val":"8"}
        ]

}
```


CfgSelect:

en_X: 1 (Enable), 0 (Disable)
rg_X: 0 (input range: -150mV~150mV)
1 (input range: 0mV~150mV)
2 (input range: -500mV~500mV)
3 (input range: 0mV~500mV)
4 (input range: -1V~1V)
5 (input range: 0V~1V)
6 (input range: -5V~5V)
7 (input range: 0V~5V)
8 (input range: -10V~10V)
9 (input range: 0V~10V)
10(input range: -0~20mA)

3. **SET Analog Input Range** settings can be done by command

POST URL:port/action/CfgAI

For example:

POST http://192.168.2.127:5003/action/fetchAIcfg?all_val=0

POST http://192.168.2.127:5003/action/CfgAI 200 OK 212ms

標頭 Post 回應 XML

參數 application/x-www-form-urlencoded

en_0	1
en_1	1
en_2	1
en_3	1
en_4	1
en_5	1
en_6	1
en_7	1
rg_0	0
rg_1	8
rg_2	8
rg_3	8
rg_4	8
rg_5	8
rg_6	8
rg_7	8
save	1

原始碼

```
en_0=1&rg_0=0&en_1=1&rg_1=8&en_2=1&rg_2=8&en_3=1&rg_3=8&en_4=1&rg_4=8&en_5=1&rg_5=8&en_6=1&rg_6=8&en_7
=1&rg_7=8&save=1
```

4. Set Relay Output:

POST URL:port/action/CtrlDO1

POST http://192.168.2.127:5003/action/CtrlDO 200 OK 213ms

標頭 Post 回應 XML

參數 application/x-www-form-urlencoded

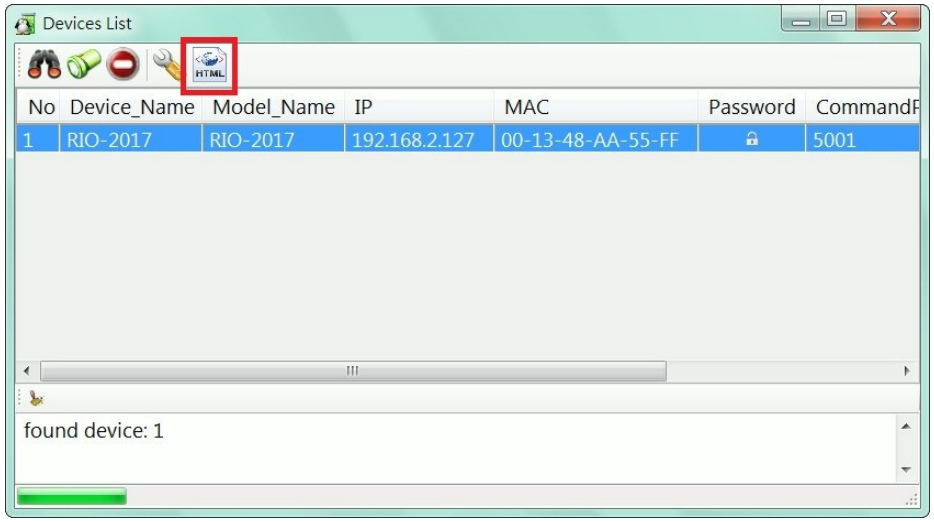
DO1 L

原始碼

```
DO1=L
```

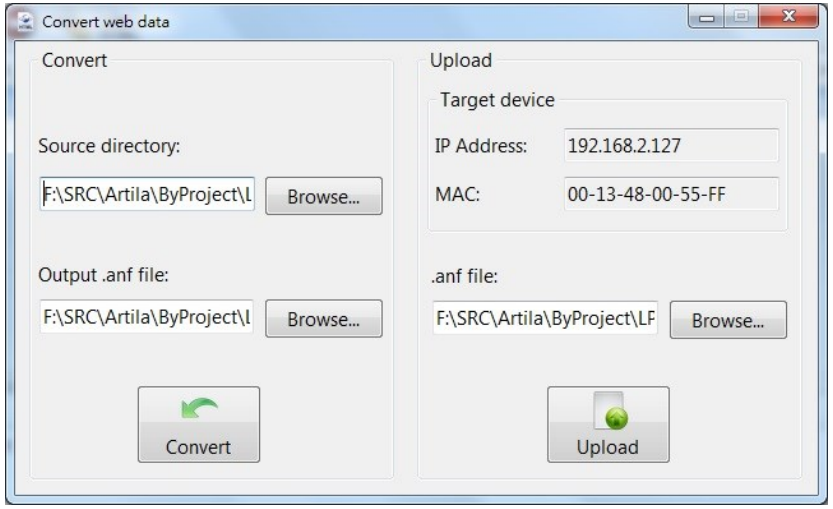
Convert HTML file to anf binary

User can customize web page to access RIO-2017. Once ready, you can use Manager utility to convert the web files to binary file (.anf) to upload to RIO-2017. First to create a folder to save all the Web files and then click the HTML icon to convert web data to .anf file and upload to RIO-2017.



The default web page source is available at Artila Web for download

Click the HTML icon and use following tool to convert the folder of Web page files to a binary file with extension of anf and upload it to web server of RIO-2017.



Access RIO-2017 via Modbus TCP

RIO-2017 supports Modbus TCP access. The Holding register is as follow:

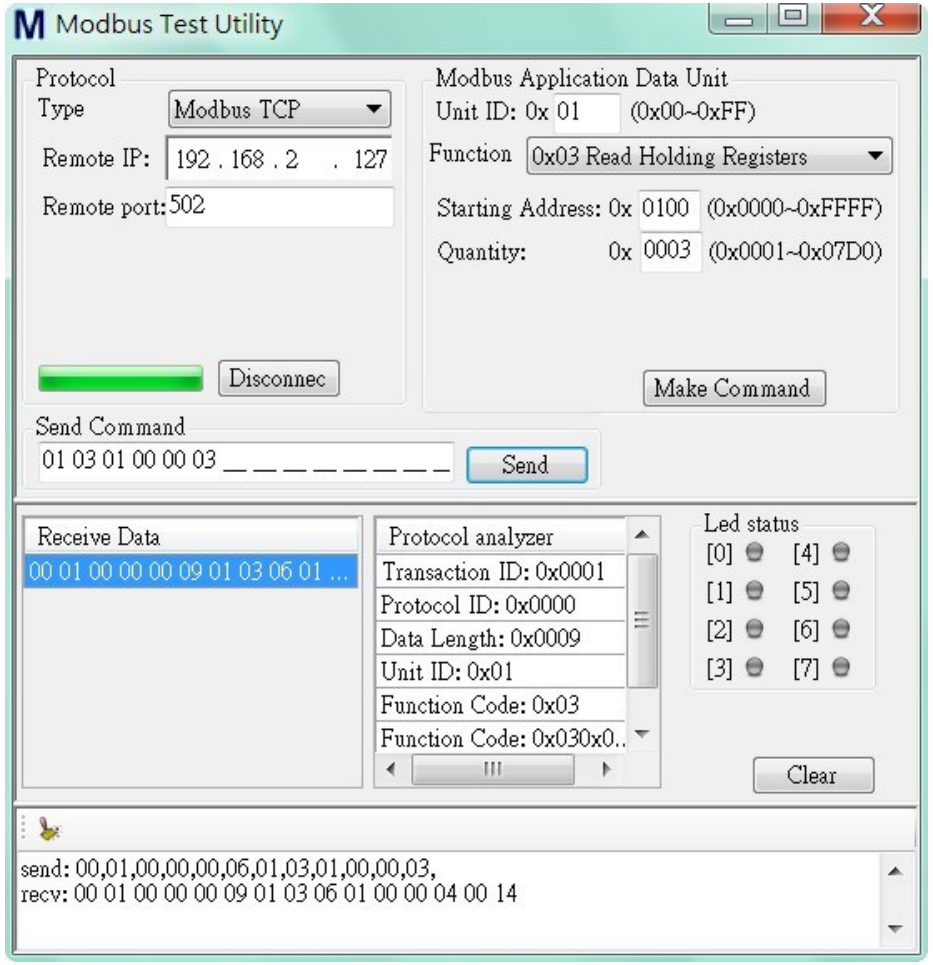
Register map

Starting address	Stopping address		
0x0000	0x000F	Temp sensor 1	RIO-2010 only
0x0010	0x001F	Temp sensor 2	RIO-2010 only
0x0020	0x002F	Temp sensor 3	RIO-2010 only
0x0100	0x0105	AI1	RIO-2017 only
0x0106	0x010B	AI2	RIO-2017 only
0x010C	0x0111	AI3	RIO-2017 only
0x0112	0x0117	AI4	RIO-2017 only
0x0118	0x011D	AI5	RIO-2017 only
0x011E	0x0123	AI6	RIO-2017 only
0x0124	0x012B	AI7	RIO-2017 only
0x012A	0x012F	AI8	RIO-2017 only

Analog Input holding register

Register[0] Hi	AI enable/disable	0x01:enable 0x00:disable	
Register[0] Lo	AI Value flag	0x00: + 0x01: -	
Register[1] Hi	AI integer Hi		
Register[1] Lo	AI integer Lo		
Register[2] Hi	AI decimal Hi		
Register[2] Lo	AI decimal Lo		

Example:
Read AI1: Starting:0x0100, Quantity: 0x0003
Response: enable, +4.20



Example:
Read AI4 and AI5: Starting:0x0112, Quantity: 0x0006
Response: AI4:enable, +70.15, AI5:enable, -0.15