Korenix JetI/O 6512
Industrial Intelligent Ethernet I/O Server

User Manual

Sept. 2009 (V1.5)

korenix

www.korenix.com

Korenix JetI/O 6512
Industrial Intelligent Ethernet I/O Server
User Manual

Copyright Notice

Index

1 Introduction

Welcome to Korenix *JetI/O 6500* Series Industrial Managed Ethernet I/O Module User Manual. Following topics are covered in this chapter:

1.1 Overview of JetI/O 6500 Series

JetI/O 6500 series is a series of Managed Ethernet I/O Server for distributive monitoring and controls. The JetI/O 6500 series equipped with one Ethernet port and multiple channels Analog Input/Output, Digital Input/Output and temperature measurement connectors. Thus users can easily perform I/O data collecting, status changing, automatically activate events… through the Ethernet network. JetI/O 6500 series provides Windows Utilities, and SNMP for configuration. And support Modbus/TCP protocol, OPC Server for Modbus/TCP, thus user can easily monitor and control the remote I/O devices and combine the JetI/O with existed HMI/SCADA package.

Naming Rule: JetI/O 65AB

A: Major Feature

1: Analog Input Series. Includes the RTD input, Thermocouple Input

2: Analog Output Series

3: Digital Input Series

4: Digital Output Series

5: Digital Input and Digital Output Series

B: Sequence Number

JetI/O 6500 Series includes:

JetI/O 6510: Intelligent 8-CH Analog Input Ethernet I/O Server

JetI/O 6511: Intelligent 8-CH Thermocouple Input Ethernet I/O Server

JetI/O 6512: Intelligent 4-CH RTD Input Ethernet I/O Server

JetI/O 6520: Intelligent 4-CH Analog Output Ethernet I/O Server

JetI/O 6550: Intelligent 14-CH DI and 8-CH DO Ethernet I/O Server

1.2 Package Checklist

Korenix JetI/O 6500 Series products are shipped with following items:

- One Ethernet I/O Server
- One attached DIN-Rail clip
- Terminal Blocks for I/O and Power Input
- Documentation and Software CD
- Quick Installation Guide

If any of the above items are missing or damaged, please contact your local sales representative.

1.3 JetI/O 6512 Introduction

JetI/O 6512 is an intelligent I/O Server equipped with 4 channels RTD Input connectors. JetI/O 6512 provides 16 bit resolution and high accuracy for temperature measurement. The supported RTD types include PT100 and Ni120 types.
JetI/O 6512 provides Windows Utilities, SNMP for configuration. Industrial Modbus/TCP protocol and OPC Server for integrating JetI/O with existed HMI/SCADA. Robust aluminum case with good heat dispersing and IP31 protection. With JetI/O users can easily perform status monitoring and control the remote I/O devices.

1.4 JetI/O 6512 Product Specification

System

CPU: 100MHZ, RISC-Based

SDRAM: 32K bytes

Flash ROM: 512K bytes

EEPROM: 256 bytes

Watchdog Timer: 1.0 sec H/W

LED:

PWR: Power Input plugged and On (Red)

RDY: System startup ready (Green)

Network Interface

Ethernet: IEEE 802.3 10Base-T, IEEE 802.3u 100Base-TX

Connector: 1 * RJ-45, Auto MDI/MDI-X

Protection: Built-in 1.5 KV magnetic isolation protection

LED:

Upper (LAN Activity): Orange ON & Blinking

Lower (10M/100M): 10M: Green OFF, 100M: Green ON

PWR: Power On (Green)

RDY: System boot up Ready (Red), system booting (No LED)

RTD Input

Input Channels: 4 Channels

Resolution: 16 bits

Input Type:
Pt100 and Ni120 RTD

Input Connections: 3, 4 or 5 wire

Input Range:

Pt100 (-100℃~100℃ α=0.00385)
Pt100 (0℃~100℃ α=0.00385)
Pt100 (0℃~200℃ α=0.00385)
Pt100 (0℃~600℃ α=0.00385)
Pt100 (-100℃~100℃ α=0.00392)
Pt100 (0℃~100℃ α=0.00392)
Pt100 (0℃~200℃ α=0.00392)
Pt100 (0℃~600℃ α=0.00392)
Nickel 120 (-80℃ ~ 260℃ α=0.00672)

Accuracy: ±0.05% of FSR ±1LSB

Sampling Rate: 10 samples/sec (total)

Input Impedance: 10M ohm

Calibration: On Board EEPROM

Isolation Voltage: 2500Vrms

Feature

Network Protocols: IP, TCP, UDP, SNMP, HTTP, BOOTP, DHCP, Modbus/TCP, OPC Server

Configuration: Windows Utility, SNMP, DHCP Client, TFTP Server for firmware update

Windows Utility: Block I/O Utility

OPC Server Utility: OPC Server for Modbus/TCP

SNMP: MIB-II: System, SNMP Trap and Private MIB

SNMP Trap Server: Up to 4 SNMP Trap Server

I/O Rules: High-/Low- Temperature alarms

Power Requirements

System Power: external unregulated +24V (18-32V)

Power Consumption: Max. 2.88W

Mechanical

Dimensions : 120 (H) x 55 (W) x 75 (D)mm

Mounting: Din-Rail

Material: Aluminum

Environmental

Regulatory Approvals: CE, FCC Class A

Operating Temperature: -25 ~ 70°C

Operating Humidity: 0 ~ 95% non-condensing

Storage Temperature: -40 ~ 80°C

Warranty: 3 years

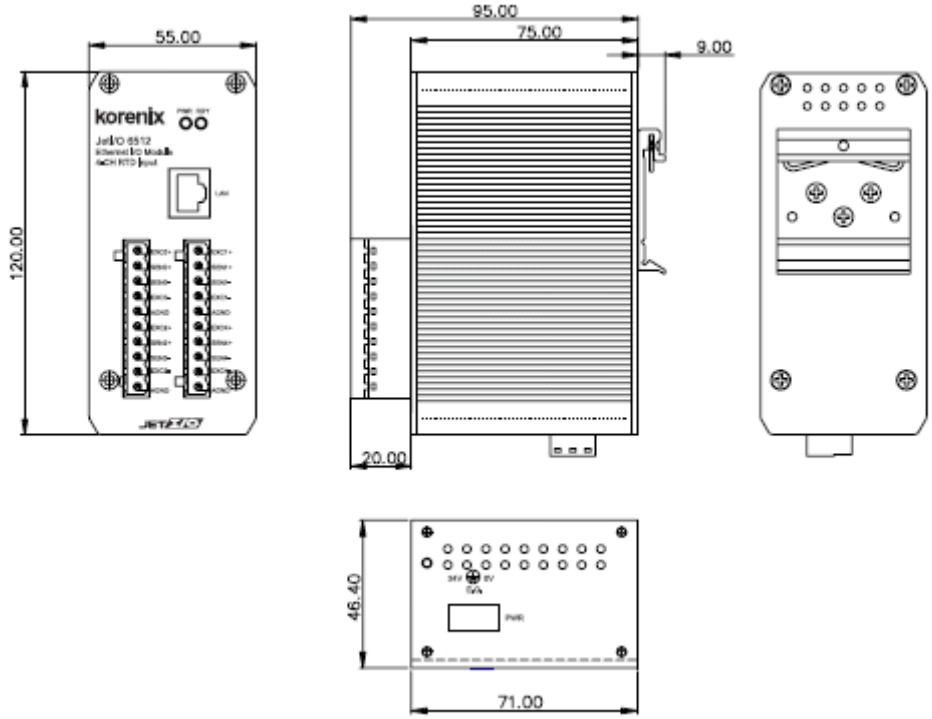
2 Hardware Installation

This chapter includes hardware introduction, installation and configuration information.

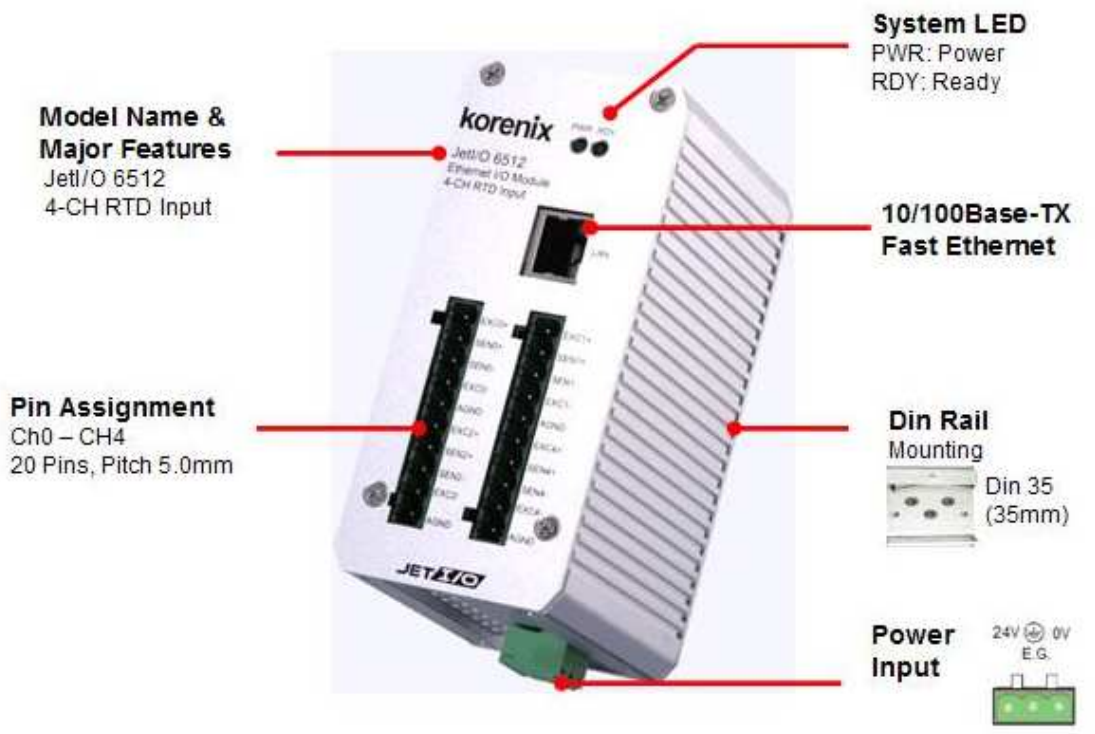
Following topics are covered in this chapter:

2.1 Hardware Introduction

Dimensions: 120 (H) x 55 (W) x 75 (D) mm



JetI/O 6512 Appearance:



LED Indicators:

System LED	
PWR	Power Input plugged and On (Green)
RDY	System startup ready (Red)
Ethernet LED	
Upper (LAN Activity)	Orange On & Blinking
Lower(10M/100M)	10M (Green Off) /100M(Green ON

2.2 Wiring Power Input

Follow below steps to wire JetI/O DC power inputs.

1. Follow the pin assignment to insert the wires into the contacts on the terminal block connector.
2. Tighten the wire-clamp screws to prevent DC wires from being loosened.
3. Connect to and turn on the power source. The suitable working voltage is 24VDC.
4. When the unit is ready, the PWR LED turns Greed, the RDY LED turns Red.

Note1: It is a good practice to turn off input and load power, and to unplug power terminal block before making wire connections. Otherwise, your screwdriver blade can

inadvertently short your terminal connections to the grounded enclosure.

Note 2: The range of the suitable electric wire is from 12 to 24 AWG.

2.3 Wiring I/O Connectors

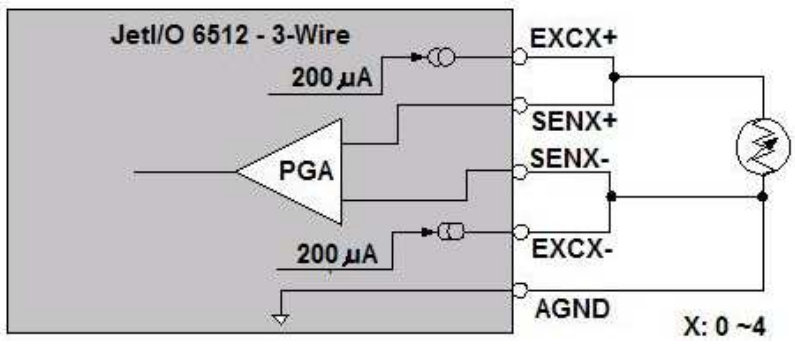
Follow the pin assignment to insert the wires into the front contacts on the terminal block connector. Tighten the wire-clamp screws to prevent the I/O wires from being loosened.

The wiring diagram of the JetI/O 6512 is as below:

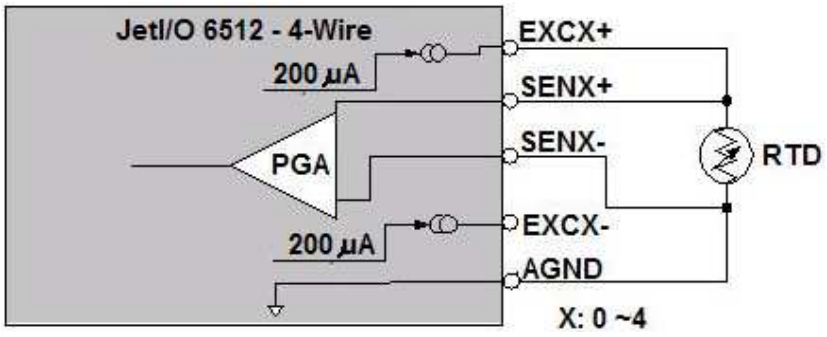
Pin No	Description	Pin No	Description
EXC0+	CH0 RTD Excitation Current +	EXC1+	CH1 RTD Excitation Current +
SEN0+	CH0 Analog Input +	SEN1+	CH1 Analog Input +
SEN0	CH0 Analog Input -	SEN1-	CH1 Analog Input -
EXC0-	CH0 RTD Excitation Current -	EXC1-	CH1 RTD Excitation Current -
AGND	Analog GND	AGND	Analog GND
EXC2+	CH2 RTD Excitation Current +	EXC3+	CH3 RTD Excitation Current +
SEN2+	CH2 Analog Input +	SEN3+	CH3 Analog Input +
SEN2-	CH2 Analog Input -	SEN3-	CH3 Analog Input -
EXC2-	CH2 RTD Excitation Current -	EXC3-	CH3 RTD Excitation Current -
AGND	Analog GND	AGND	Analog GND

2.4 JetI/O 6512 Wiring Example

2.4.1 JetI/O 6512 3-wire RTD input wiring example



2.4.2 JetI/O 6512 4-wire RTD input wiring example



2.5 Wiring Earth Ground

To ensure the system will not be damaged by noise or any electrical shock, we suggest you to make exact connection with JetI/O products with Earth Ground.

On the bottom side of JetI/O 6500 Series, there is one power earth ground pin in the Power Input terminal block.

Pin No	Description
1(+24V)	DC+24V Power Input
2(FGND)	Power Earth Ground
3(0V)	Referenced Ground for Power Input

2.6 Wiring Fast Ethernet Ports

JetI/O 6500 series includes 1 RJ45 Fast Ethernet ports. The fast Ethernet ports support 10Base-T and 100Base-TX, full or half duplex modes. The fast Ethernet port will auto-detect the signal from connected devices to negotiate the link speed and duplex mode. Auto MDI/MDIX allows users to connect another switch, hub or workstation without changing straight through or crossover cables.

Connect one side of an Ethernet cable into the Ethernet port and connect the other side to the attached switch or host. The link LED will light up when the cable is correctly connected. Refer to the **LED Indicators** section for descriptions of each LED indicator. Always make sure that the cable length between the 2 ends is less than 100 meters (328 feet).

The wiring cable types are as below.

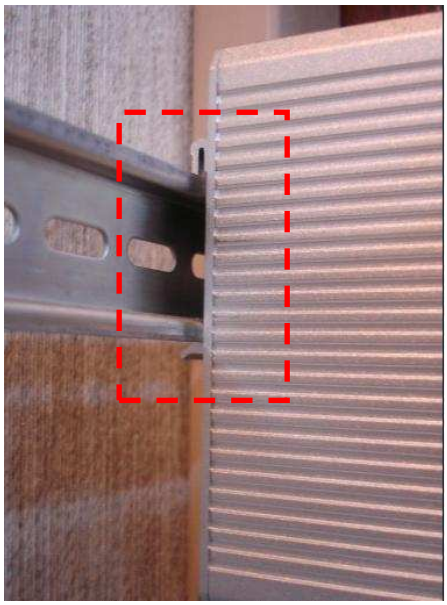
10Base-T: 2-pair UTP/STP Cat. 3, 4, 5 cable, EIA/TIA-568 100-ohm (100m)
100 Base-TX: 2-pair UTP/STP Cat. 5 cable, EIA/TIA-568 100-ohm (100m)
1000 Base-TX: 4-pair UTP/STP Cat. 5 cable, EIA/TIA-568 100-ohm (100m)

2.7 Din-Rail Mounting Installation

The DIN-Rail clip is already attached to the JetI/O 6500 Series when packaged. If the DIN-Rail clip is not screwed on the JetI/O, follow the instructions and the figure below to attach DIN-Rail clip to JetI/O.

Figure 1. Insert the upper end of DIN-Rail clip into the back of DIN-Rail track from its upper side.

b. Lightly push the bottom of DIN-Rail clip into the track.



c. Check if DIN-Rail clip is tightly attached on the track.

d. Korenix suggests reserve at least 5mm interval distance between the JetI/O devices. This is good for heat dispersing.

e. To remove JetI/O 6500 from the track, reverse the steps above.

3 Preparation for Management

Before you start to configure the JetI/O, you need to know the system architecture of the JetI/O products, configure the device's IP address, and then you can remotely manage the Ethernet I/O via the network. This chapter introduces the basic knowledge of the related technologies.

Following topics are covered in this chapter:

3.1 Understand the Ethernet I/O Server Architecture

The Figure 1 shows the JetI/O Intelligent Ethernet I/O Server Architecture. In the top level shows the typical applications run in the remote I/O environment. The middle level is the Ethernet infrastructure. The low level, gray block include the software agent, signal types of the JetI/O 6500 series intelligent Ethernet I/O Server.

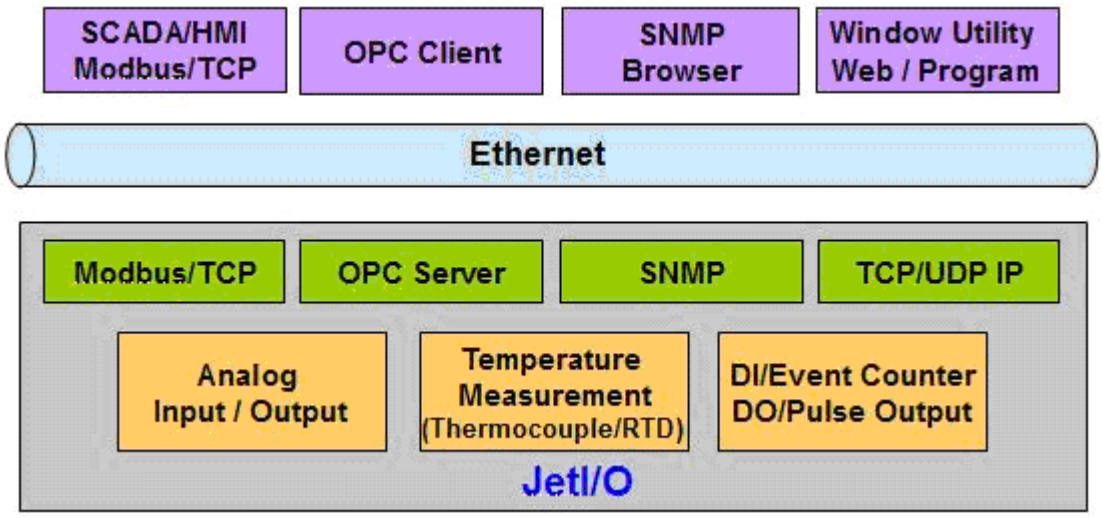


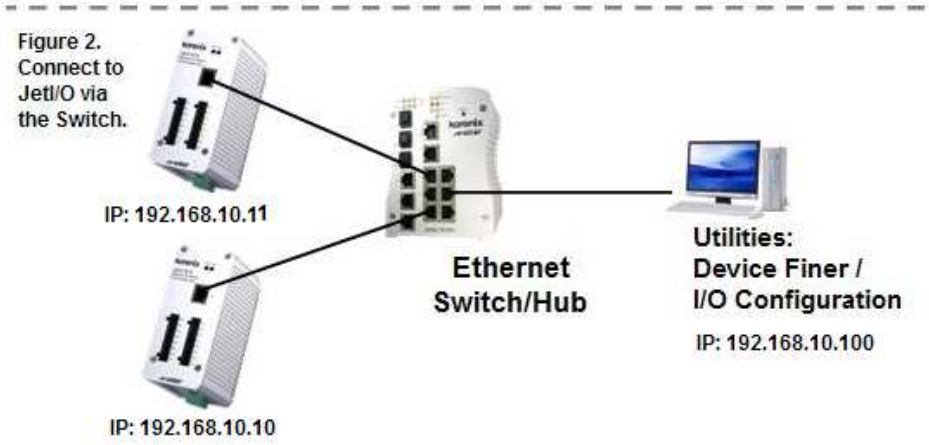
Figure 1. The JetI/O Intelligent Ethernet I/O Server Architecture.

3.2 Preparation for Remote Management

JetI/O 6500 series Intelligent I/O Server provides several types remote management methods. You can configure the JetI/O via the Ethernet network. You just need to know the device's IP address and then you can remotely control or monitor the I/O channels' information.

JetI/O provides several ways for users to configure the IP address. The default IP address is 192.168.10.3. You can directly connect the JetI/O one after one to change its

IP address. Or connect the JetI/Os to the same switch or network, then the host PC can modify the IP address via the switch or network.



If you purchase several JetI/Os and connect them to the same network before change their IP address. They must have the same default IP address, and you may not control them well due to the IP conflict. At this time, you should change their IP address first. The JetI/O' Block I/O configuration utility and its Device Finder Popup Window can help you to do this.

Note 1: Device Finder Popup Window allows you to discover the JetI/Os which have the same IP address. Change the IP address of the JetI/O one after one. After you configured the new IP address for the unit, please notice whether the ARP table of the device is flashed or not. If not, you can choose "Start -> Run", type "cmd" to open the DOS prompt. Use "arp –d" to clear the ARP cache.

Note 2: After changed IP address or changed the DHCP client mode in Block I/O Configuration utility, the utility will automatically reboot the unit. Please rescan the devices after around 5 seconds.

Note 3: You can find the detail progress in the next chapters.

4 Feature Configuration

JetI/O 6500 series Industrial Managed Ethernet I/O module provides several configuration methods. This chapter introduces the configuration steps.

Following topics are covered in this chapter:

4.1 Block I/O Configuration Utility

Block I/O Utility is the major JetI/O Configuration Utility. With this tool, you can browse the available units, view the status of each channel, configure the I/O settings, configure active alarms and Conditions&Go logic rule.

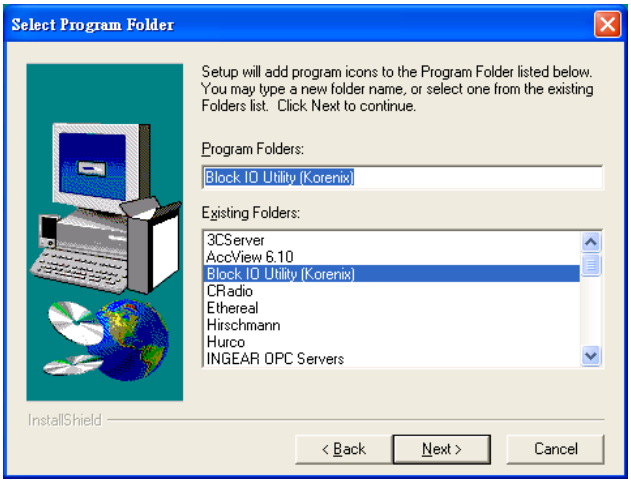
4.1.1 Installation

1. Go to the “Utility -> IO Configuration” folder. Click “Setup.exe” to run the setup progress.

2. Click “Next” and type the Name and Company in the “User Information” window. Then click “Next”.

3. Choose the Destination Directory in the “Choose Installation Location” window. Then click “Next”.

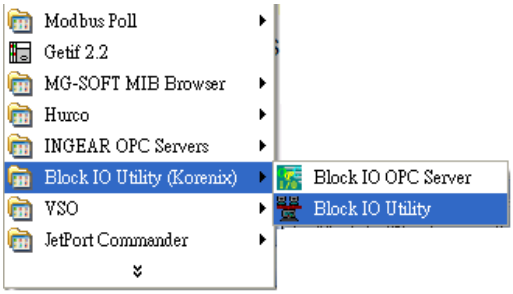
4. Type the name for the Block I/O Configuration Utility or use the default name, Block IO Utility (Korenix) for the program in the “Program Folder” field of the “Select Program Folder” window. Then click “Next”.



5. Click “Next” in the “Starting Copying File” window to continue the setup progress.

6. As long as you see the “Setup Complete” window that means the progress is finished. Click “Finish” to exit the setup progress.

7. Go to “Start” -> “Program”, and then you can see the “Block IO Utility (Korenix) folder. There are 2 utilities are installed, Block IO OPC Server and Block IO Utility.

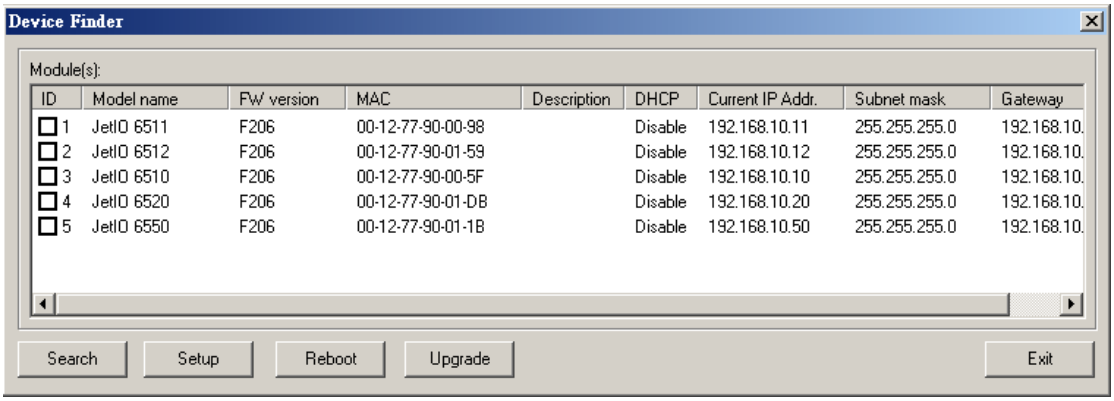


4.1.2 Device Finder

Device Finder helps you search JetIO 6500 devices on the same physical subnet, even if their IP addresses are conflit or if their IP address setting are not on the same subnet with your host PC. Device Finder also helps you configure the IP address and upgrade firmware of the found devices.

Select the Device Finder icon  from toolbar or select "Tools"→"Device Finder" to launch Device Finder.

Figure II. Click "Search" to search the JetI/O devices. You can see the available devices in the list.

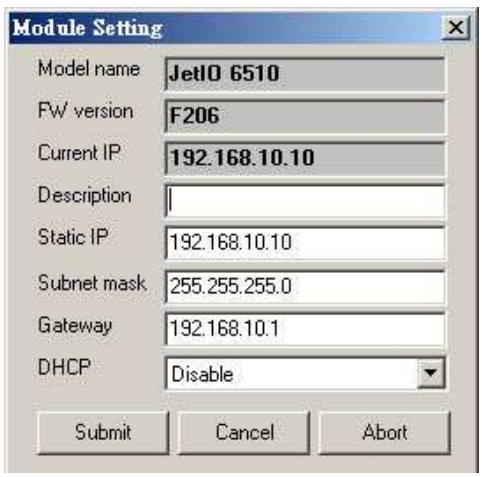


The following information is displayed:

Item	Description
ID	index of the list
Model Name	JetIO model name
FW version	The firmware version
MAC	MAC address
Description	A short description of the device, max 16 characters
DHCP	DHCP client function status: Enable or disable
Current IP Addr	If DHCP is enabled, dynamic IP is acquired from the DHCP server, Else, static IP is assigned as dynamic IP.
Subnet mask	Subnet mask
Gateway	Gateway IP address

2. Select the target unit and click "Setup" button to configure the device. Click "Submit" to apply the new setting.

Note: When changing IP address, the new IP address and the NIC's (Network Interface Card) IP address should be located within the same subnet.

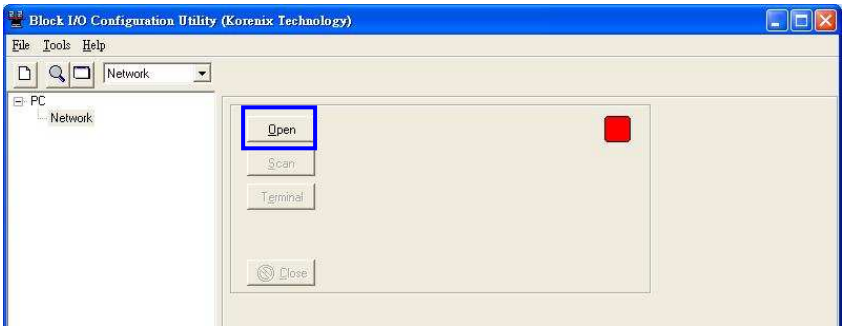


3. Select the target units and click “Reboot” to reboot the device. You can reboot one or multiple units in one time.
4. Select the target units and click “Upgrade” to upload the new firmware. Please refer to the section 4.4 to know the detail step by step progress.
5. Click “Exit” to exit the device finder tool.

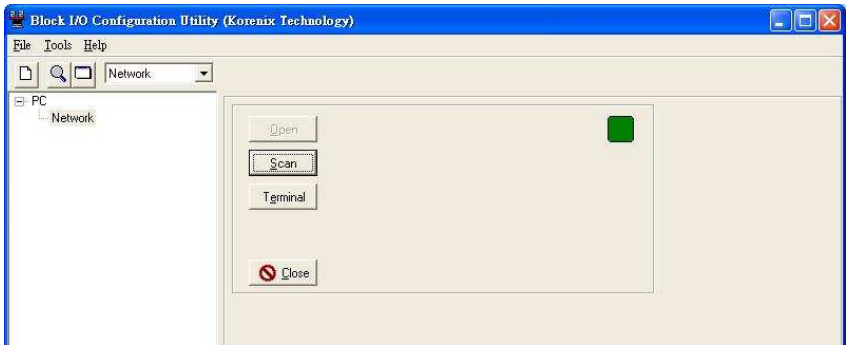
Note: *Clear the ARP cache (arp –d in DOS prompt) if you can’t change the second unit’s IP address. In DOS prompt, “arp –a” can help you to see the ARP table. “arp –d” can help you to clear all the ARPs in your host PC.*

4.1.3 Device Scan

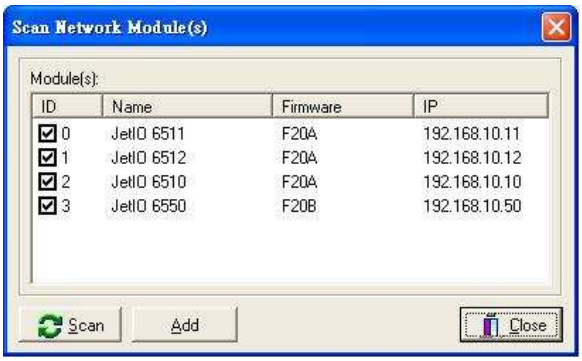
1. Lunch the Block IO Utility and then press “Open” to enable the network Interface.



The square LED shows “Green” after you opened the interface. Click “Close” can close the network interface.



2. Click “Scan” to open the “Scan Network Module(s)” popup window. Click Scan to start the searching.



Note: Please modify the IP address of your target devices. The scan feature can't browse the devices which have the same IP address. Only one of the devices which have the same IP address can be found. This is the current restriction. Please modify the IP address first. You can use Block I/O Utility or Device Finder to do the IP modification.

3. Click "Add" to add the available JetI/O units. Then you can see the JetI/O units are listed in the left column.

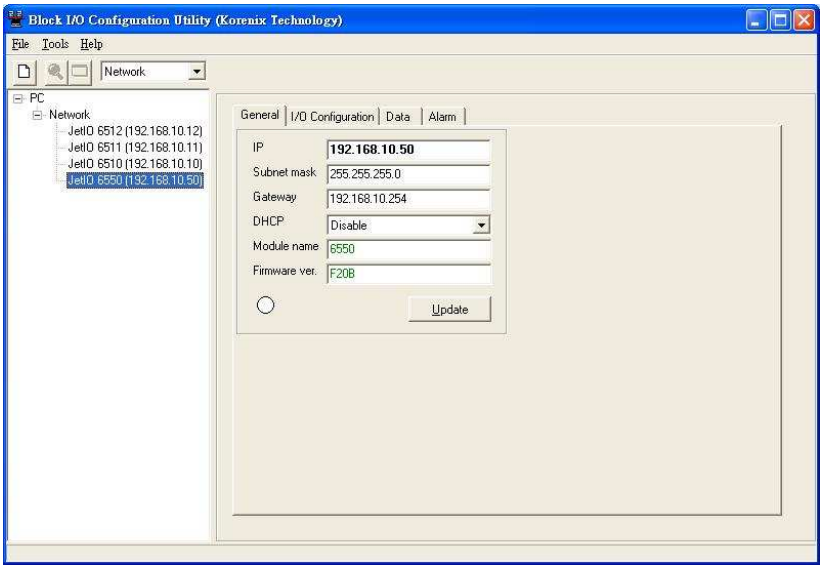
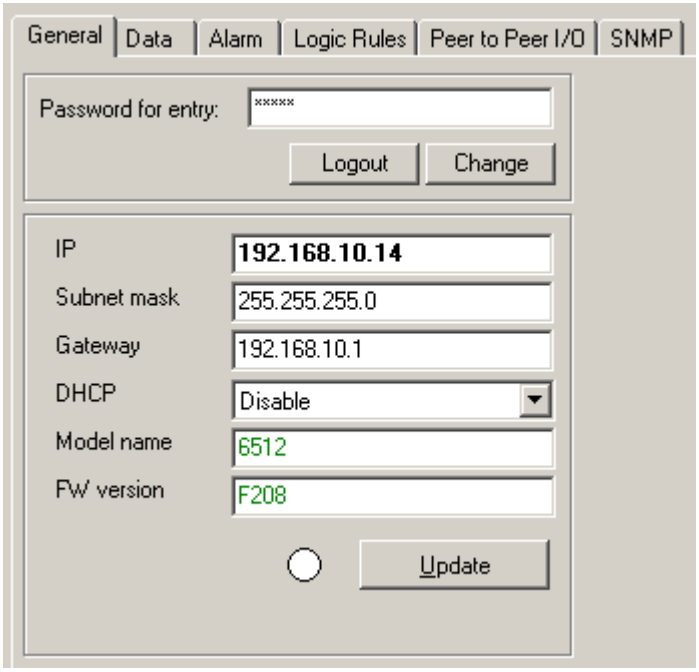


Figure 1. Move the mouse over to one of the JetI/O units. Select the unit then you can configure and monitor the configurations of the JetI/O. The features Block I/O Configuration utility provides are similar. Please find your model name and go to its configuration introduction chapter in below.

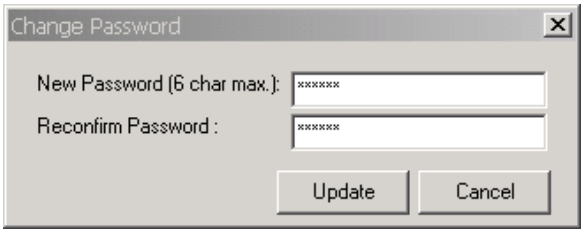
4.1.4 JetI/O 6512 Configuration

4.1.4.1 Go to **"General"** page.

Once you select one of the existing I/O modules in the Network interface tree, the main window defaults to the **"**General**"** page.



It should be noticed that the privilege setting is required to gain access to do the further configuration options. When you install the I/O module in first time, the default password is "**admin**". You just need to click on "Login**"** button and type the password of "Admin" on "password for entry" field. If you want to make password changes, click "Change**"** button and then the dialog of "Chang Password" prompts you to update the new password with up to six characters.

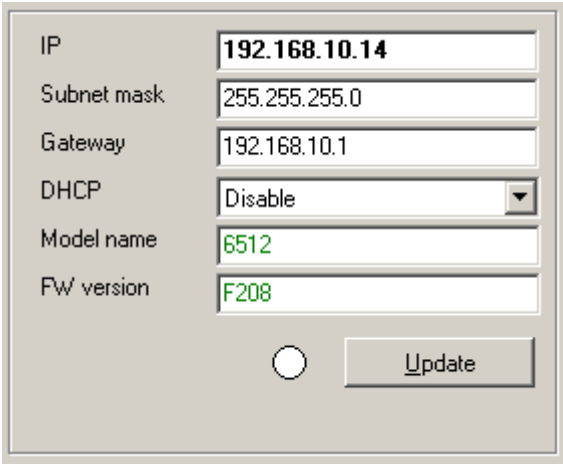


Note 1: *The password protection is the new feature provided in firmware F206 and Utility V1.3 or later.*

Note 2: *When you upgrade firmware from F204 to F206, the default password is disabled, please change new password for your device.*

Note 3: *If you forget the password, you may need to reset the module via Modbus/TCP 'RESET' command to clear the password and load factory defaults. This will result in the clear all the configuration settings as you assigned.*

Each module has its own page to display and configure the TCP/IP parameters. If the function of DHCP is disabled, user can type the IP address, Gateway and Subnet Mask information on the page. When everything is ok, user can push the "Update**"** button. This setting will be affected after restarting the module.

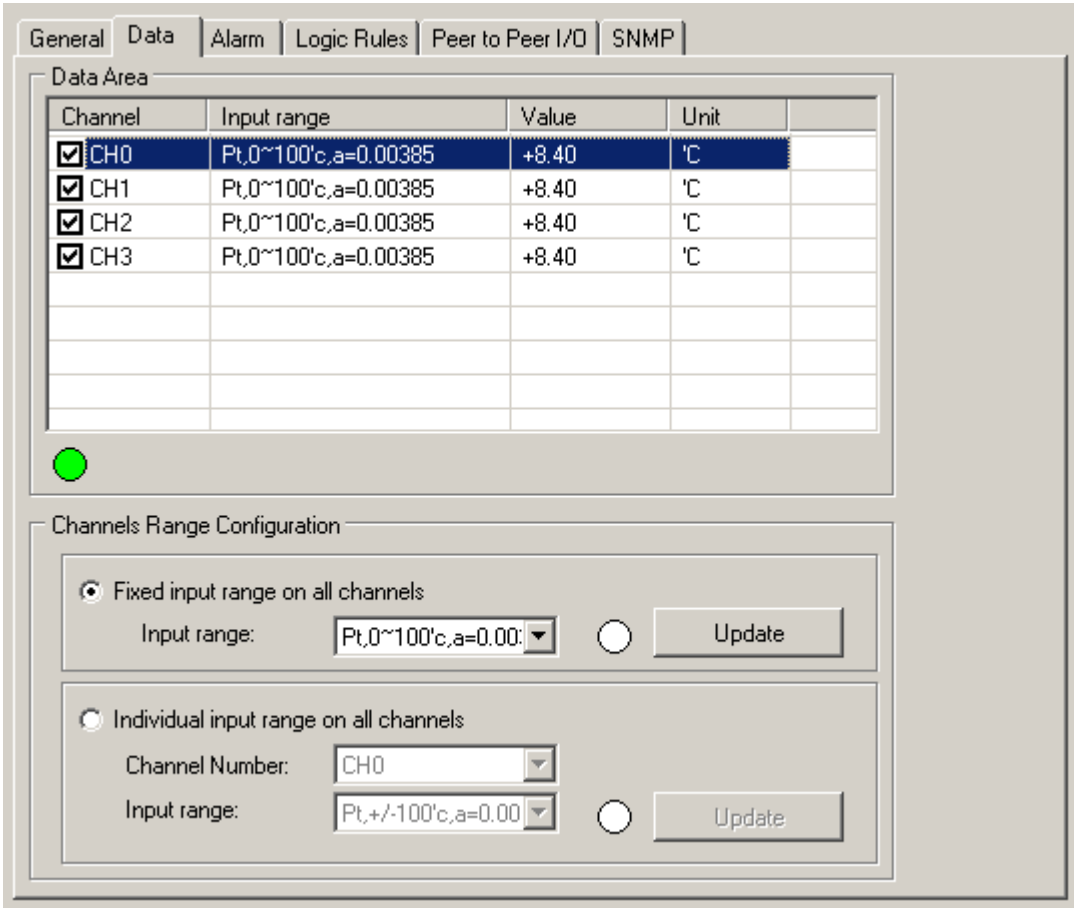


Note: *After changed IP address or changed the DHCP client mode, the utility will automatically reboot the unit. Please rescan the devices after around 5 seconds.*

4.1.4.2 Go to **"Data"** page.

In this page, you could monitor the current working status of each channel on AI as well as the channels range configuration.

Check Box	Select the check box to monitor the info of the channel. Unselect the check box when you don't want to monitor it.
Blinking	The indicator in the bottom of the Data Area means the utility is monitoring the status of the channels. If there is error occurred, the color become to red or not light.

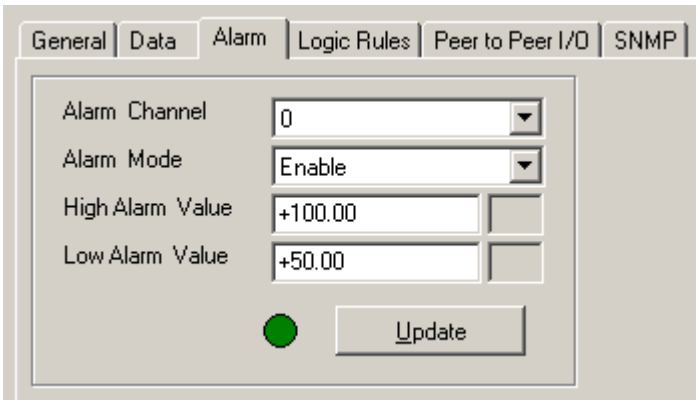


The Channels Range Configuration area, you can choose either “Fixed input range on all channels” or “Individual input range on all channels”. The factory default setting is “Fixed input range on all channels”. Just select the appropriate range from the “Input range” combo box and then click “Update” to take effect. If you want to set a different range for each channel, select the “Individual input range on all channels”, set the channel number in the Channel index combo box, and then select the range in the Input range combo box. After selecting appropriate range, click the Apply button. For example, channel 1 with Ni Type meanwhile the others with Pt type.

Note: *The* individual input range on the different channels *is the new feature provided in hardware version Rev. B or later*

4.1.4.3 Go to **“Alarm”** page.

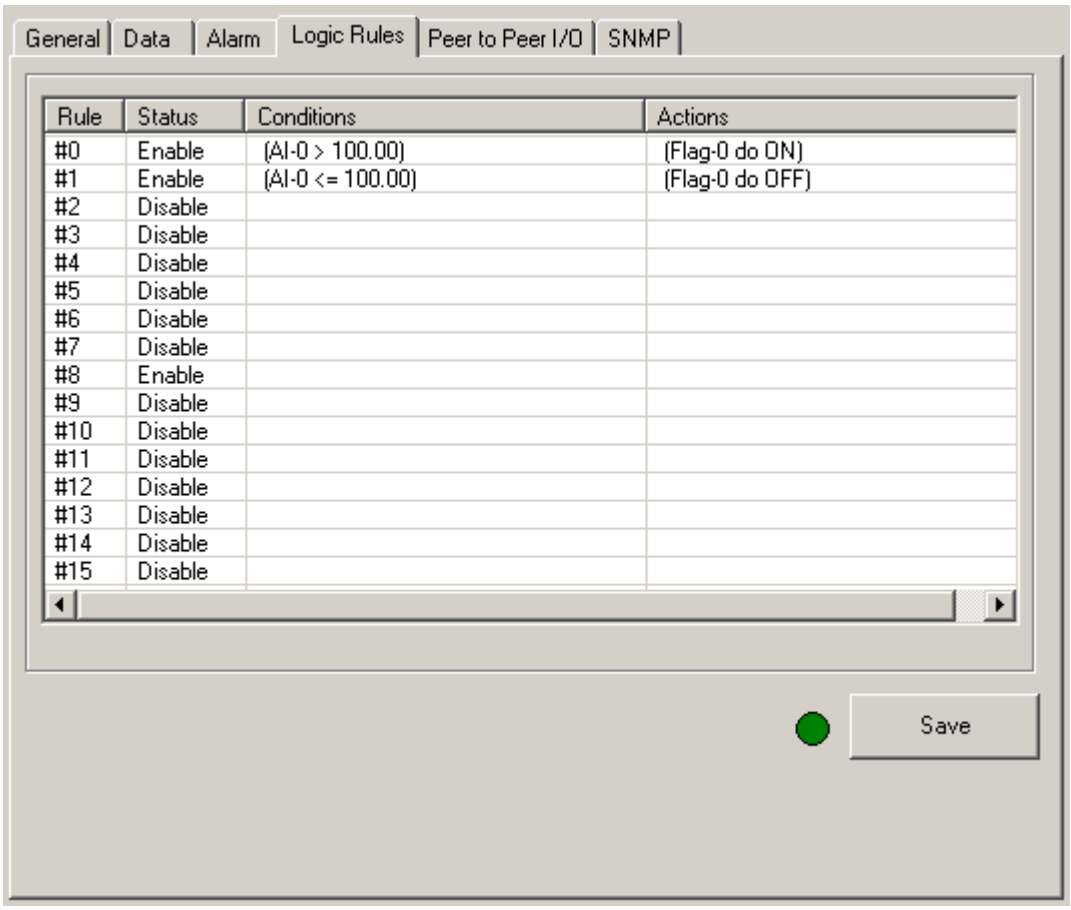
In this page, you can setup the High/Low alarm value (Voltage or Current) for each channel and enable generic alarms (Device Cold Reboot, LAN Link Up).



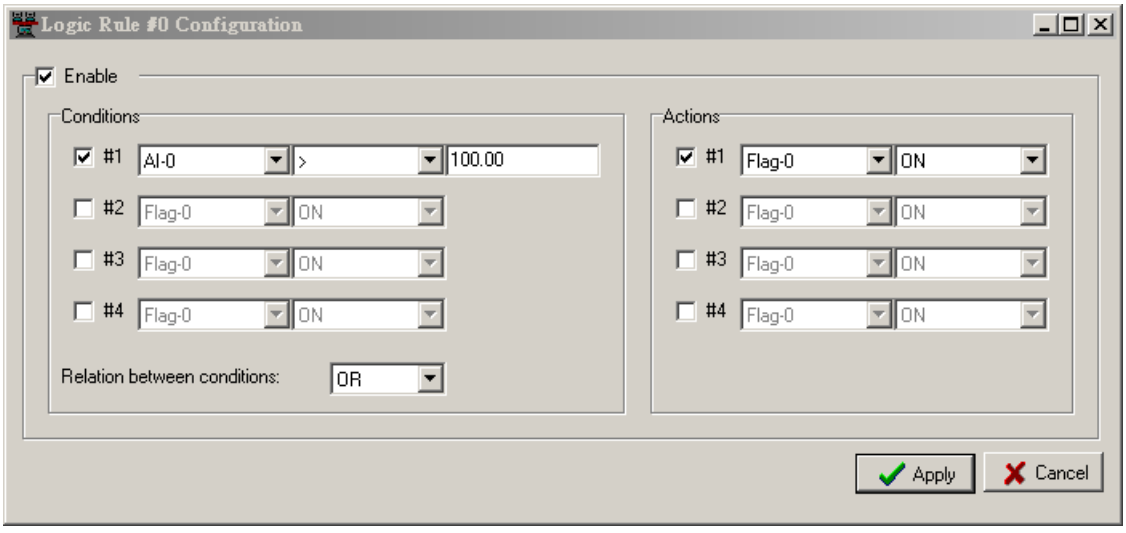
Alarm Channel: Select the channel.
Alarm Mode: Enable or Disable
High Alarm Value, Low Alarm Value: Type the value here.
Update: Activate the new setting.

4.1.4.4 Go to **"Logic Rules"** page.

In this page, you can configure the I/O logic rule. It allows you to define the logic operation and process rules in this utility and then download the rules to the I/O module. The module will automatically execute the logic rules to process different action depending on the input conditions as you defined. The theory is the same as the "IF-Then" rule. It's easy to understand rules, no need know extra program script to configure this.



Double click the Rule ID and then you can go to the Logic Rulle#ID Configuration page. Select “Enable” and configure the Condition and Actions then press “Apply” to enable the rule.



Thus, when the “Conditions” is reached, the system automatically activates the “Actives”. For example:

Rule	Status	Condition	Action
#1	Enable	(AI-0 > 100)	Flag-0 = ON
#2	Enable	(AI-0 ≤ 100)	Flag-0 = OFF

Rule #1: If condition is equal to “(AI-0 > 100 degree)”, the “Flag-0” is automatically “ON”. Again, configure the reverse way in the rule #2: If condition is equal to “(AI-0 ≤ 100 degree)”, the “Flag-0” is automatically “OFF”. In this meantime, the local internal flag, that says Flag-0, on the module can be mapping to remote digital outputs on different destination modules via. peer-to-peer I/O activity.

The maximum I/O logic rules support up to 16. Each rule can support up to 4 different conditions and 4 different actions. There are 16 internal flags (i.e., auxiliary channels) on each module. The data type of internal flags is digital, meaning its value is either logic True or logic False. It allows you to choose these internal flags as input for condition rule as well as these flags as output for action state. Therefore, you could easy to use these internal flags to implement logic rule in cascade mode and mirror these flags to/from the remote modules via peer-to-peer I/O activity.

The supported conditions and actions are as below:

Condition		**Action**	
Internal Flag	ON, OFF, ON to OFF, OFF to ON	Internal Flag	ON, OFF
Analog Input	=, >, <, ≥, ≤ (Number)	SNMP	(Trap Server IP)
Relation between Conditions	OR, AND		

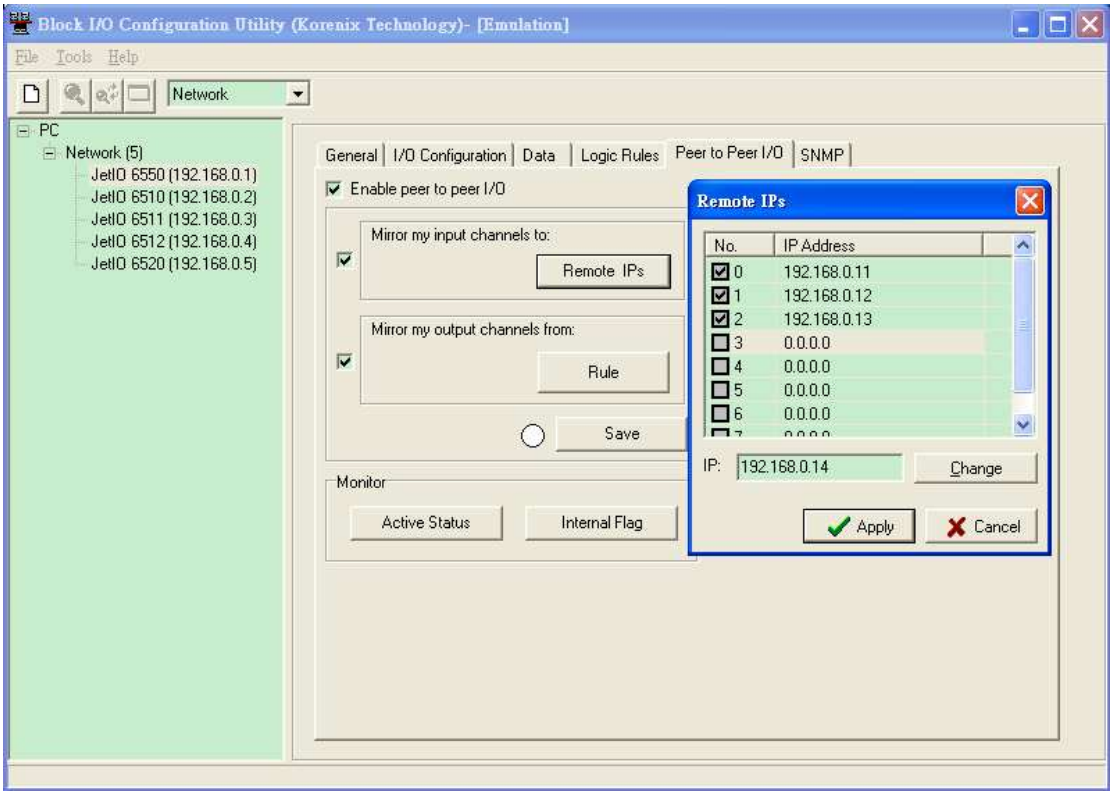
After completing all configurations for I/O logic rule, click “Save” button in the bottom of the Logic Rules area. All the mapping configurations will be flushed into flash memory on the module. In order to extend the flash memory life, it is strongly recommended that you should save all configurations together at one time instead of saving individual setting many times.

4.1.4.5 Go to **“Peer to Peer I/O”** page.

With peer to peer I/O activity, input channel status on one module could be actively updated to specific output channel on another module over the existing Ethernet connection. The above data exchange will be transferred automatically without any controller or programming needed.

Each channel including all the internal flags on the source module can be mapping to channel including all the internal flags on different destination

modules. The Peer-to-Peer I/O activity on all the modules not only supports the **"one-to-one"** mapping but also **"multiple-to-one/from-one"** and **"one-to/from-multiple"** mapping simultaneously.



The below will guide you on how to configure peer-to-peer functions step-by-step walkthrough.

(1) Enable Peer to Peer I/O activity

The Peer to Peer I/O activity is disabled by default. You could enable this function by checking "Enable peer to peer I/O" on this page.

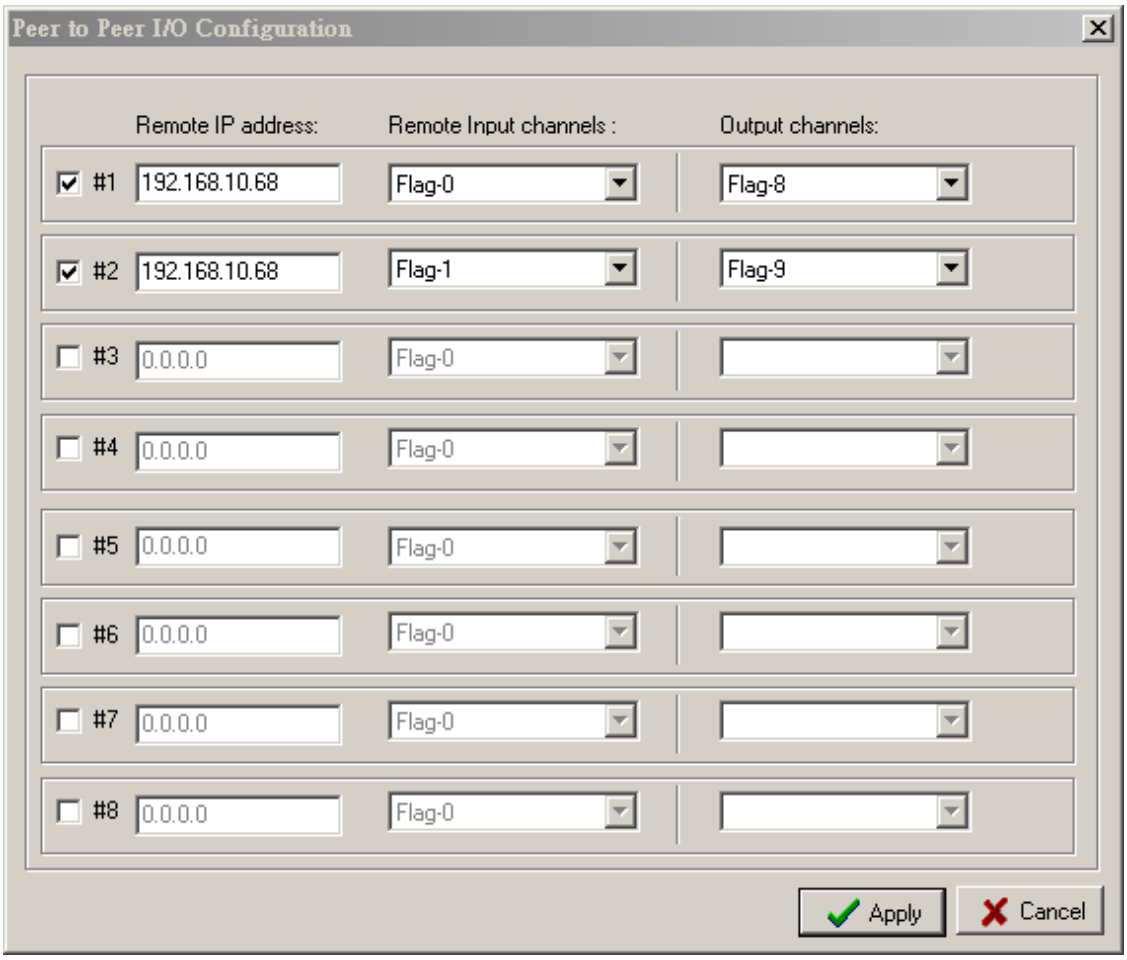
(2) Configure module's input channels mirrored to remote IP

Select "Mirror my input channels to:", the input channel status on the input module could be assigned to send those input data including all the internal flags to one remote destination IP address as defined on the field of "Remote IP", e.g., 192.168.10.51 as the above figure.

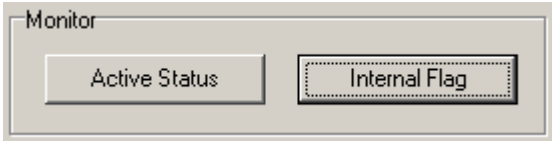
(3) Configure module's output channels mirrored from remote IP

Select "Mirror my output channels from:", the output channel state on the output module could be mirrored from one remote input channels or internal flags on different destination modules. Click "Rule" and do the further assignments for those internal flags which will mirror the remote internal flags on the corresponding remotely input module as defined on the

field of “Remote IP address”. It could support up to 8 rules assignment. Press “Apply” to activate the new setting.

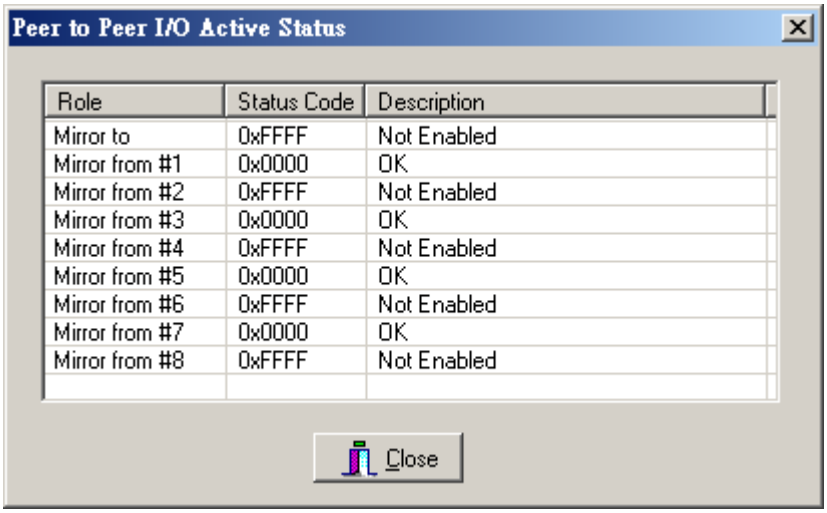


After completing all configurations for Peer-to-Peer I/O activity, click “Save” button, all the mapping configurations will be flushed into flash memory on the module. In order to extend the flash memory life, it is strongly recommended that you should save all configurations together at one time instead of saving individual setting many times. The indicator in the left of the “Save” button shows the status of being flushed into flash memory. If there was error occurred, the color become to red. Otherwise, the system will automatically restart the module and prompt you to rescan the module on the network again.



Refer to above Figure as below, in the bottom field of this “Peer to Peer I/O” page is the motoring area. It is monitoring the status both on peer-to-peer I/O activity and internal flags. Click “Active Status” button, the dialog of

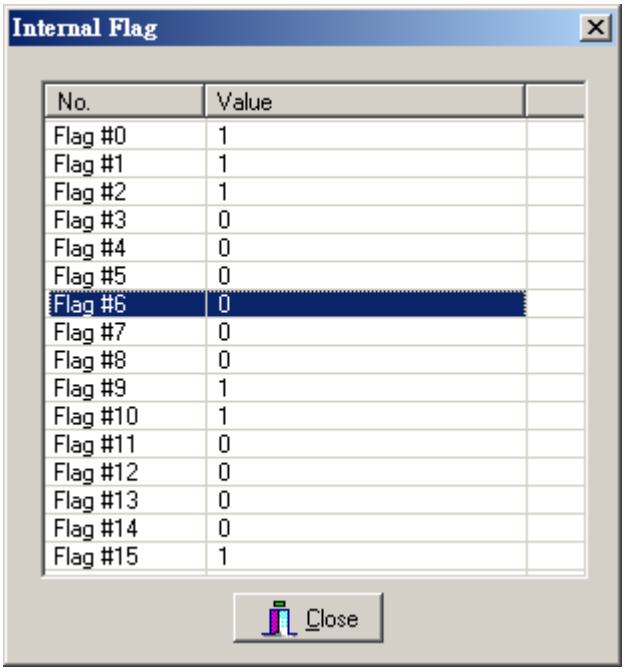
peer-to-peer I/O active status will popup to represent the status code on peer-to-peer I/O activity.



If there is error occurred, you could directly browse the latest activity from status code of peer-to-peer I/O activity. The status code is defined as below.

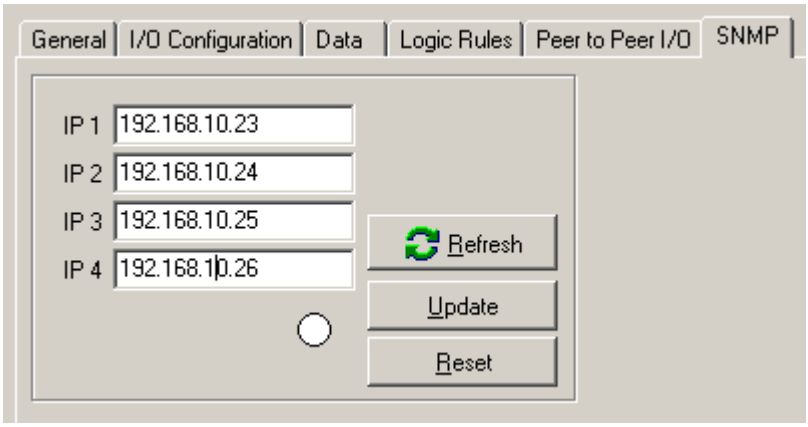
Status Code	Descriptions
0x0000	NO Errors
0x0001	Requested Content Not Satisfiable. That is, The content was well-formed but was unable to be followed due to non-satisfiable data. For example, module type associated with data format was NOT consistent.
0x0002	Remote Module Not Found
0xFFFF	Not Enabled the Peer-to-Peer I/O Activity

Click “Internal Flag” button, the internal flag dialog will prompt for you to browse all the internal flags. If you use internal flags as the inputs of logic rules and/or peer-to-peer I/O activity, you can dynamically change the flag values in the monitoring by double clicking the Flag# row as you selected, and then the flag values will be changed from “True” to “False”, or from “False” to “True”.



4.1.4.6 Go to **"SNMP"** page.

IP Setting: You can configure up to 4 SNMP Trap Server's IP here. Type the IP address and press "Update" to activate the new setting. Click "Refresh" to reload the current SNMP trap server's IP from registers. Once the SNMP trap has been activated by I/O login rule, you need to press "Reset" to acknowledge it and then allow the next SNMP trap activity.

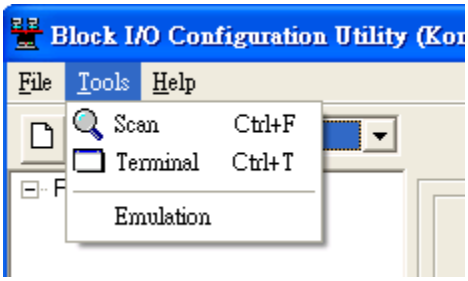


The round LED shows green when press update and the setting is correct.

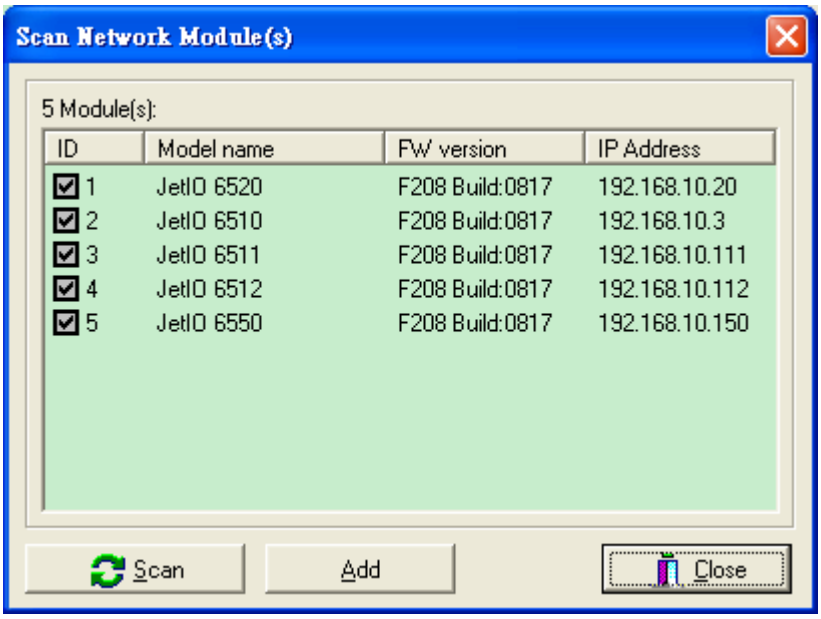
4.1.5 Emulation Mode

Block I/O Configuration Utility provides Emulation mode that allows users to know the functions it supports, and good practice for users to know how to operate block I/O configuration utility even when users don't have physical devices on hand.

4.1.4.1 Select "Tools -> Emulation".



4.1.4.2 Follow the 4.1.3 to scan the network, you'll find the models JetI/O currently supported. Click "Add" to add the models.

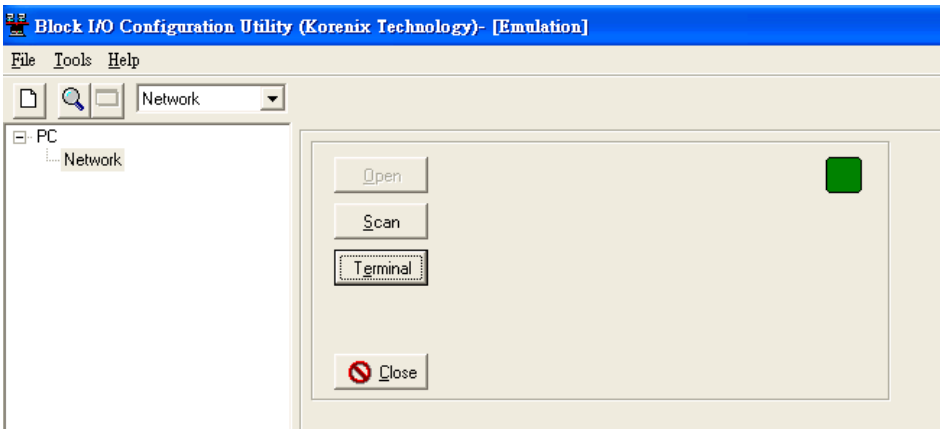


4.1.4.3 Follow the 4.1.4 to practice JetI/O configuration. Select the model and read or write status and configurations. As to how to operate the JetI/O configuration of other model, please refer to its manual.

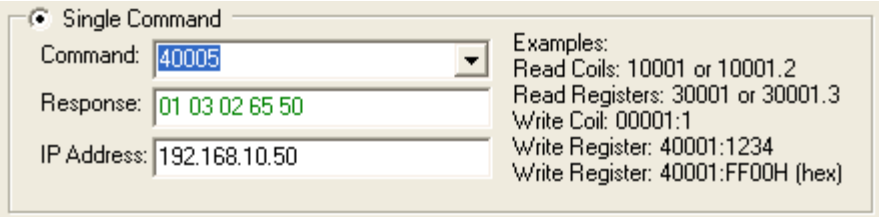
4.1.6 Terminal Mode

Block I/O Configuration Utility provides "Terminal" mode for user to read and write Modbus/TCP registers, thus users do not need additional tools but still can practice Modbus/TCP protocol well.

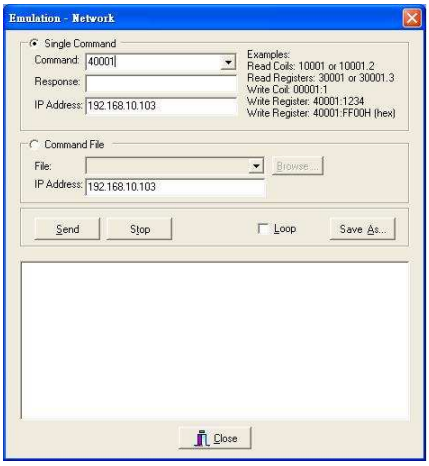
4.1.6.1 Open Terminal Mode. Click Open and then select "Terminal". The terminal emulation popup screen appears.



4.1.6.2 Single Command mode. Type the correct IP address of target unit in IP Address field, PLC address in the Command field. Then click “Enter” key. You can read the Response of the PLC address.



Note: If you type the wrong IP address, the utility will re-try the connection few times. This may take few seconds, please wait and close the popup alert screen and type



the correct IP address again.

4.1.6.3 To read register, just type the PLC address. To write register, user needs to type the new value behind the PLC address. The example is 40001:1234 (ASCII word) or 40001:FF00h (16x Hex).

4.1.6.4 Command File mode. Type PLC address you want read or write in the text file. Type the correct IP address of target unit in IP Address field. Browse the text file to load the file.

Example: Write below commands in Modbus test.txt file and browse it.

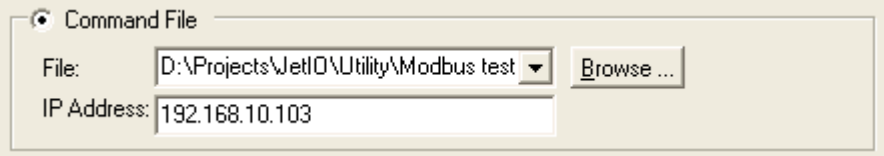
40001

40002

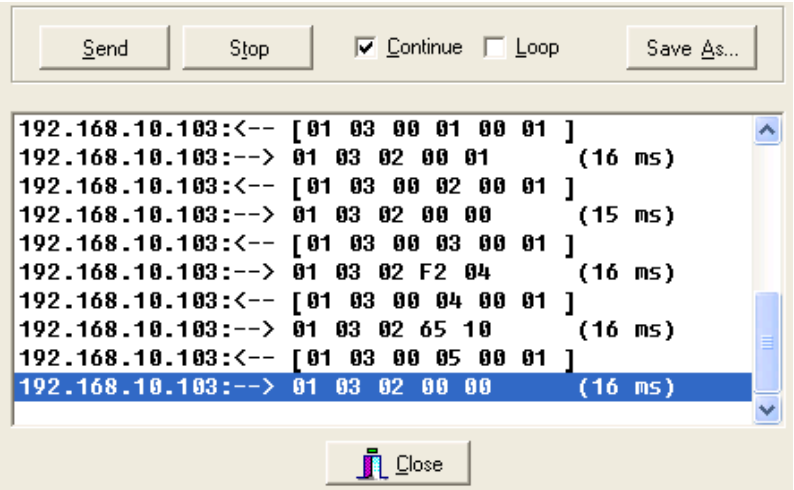
40003

40004

40005



4.1.6.5 Commands: Run Send to run the multiple commands. Run “Stop” to stop the program. Select “Continue” to run all commands once. Select “Loop” to continuously run all commands. The commands can be applied to Single Command mode and Command File mode.



The above screen shows you the result of running “Modbus test.txt” example in 4.1.6.4.

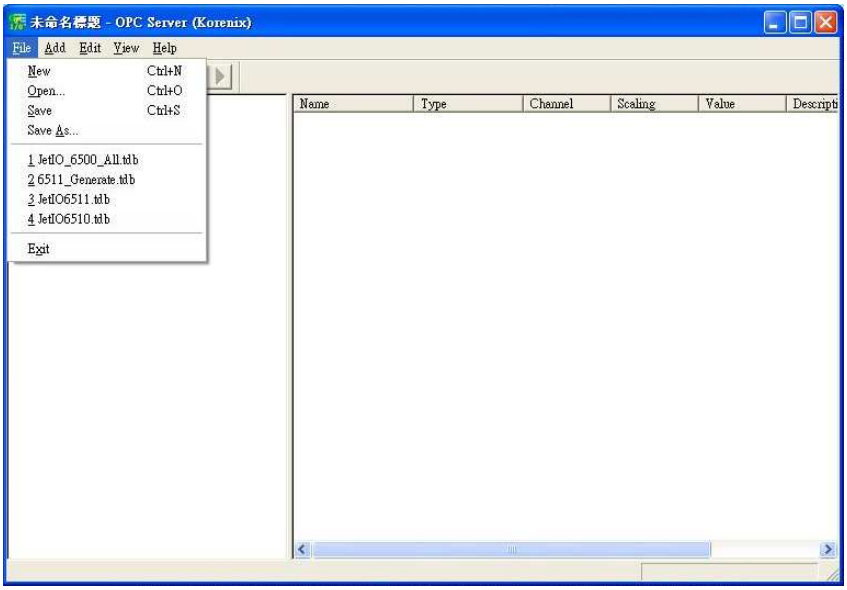
4.2 Block I/O OPC Server Utility

4.2.1 OPC Server Utility

1. Go to “Start” -> “Program”, and then you can see the “Block IO Utility (Korenix) folder. There are 2 utilities are installed, Block IO OPC Server and Block IO Utility.

2. Open the “Block IO OPC Server”.

3. Select “File -> New” to create new profile. Or select “File -> Open” to open profile you saved.



4. Select “Add -> New Device”, the popup window “Driver Selection” will appear. (Only appear in the first time you add new device). Click “Add” and type the driver name and correct IP address. Click “OK” to next popup windows for Driver Selection. Use “Edit -> Comm Setting” can modify the parameters.

Note: *Different model should have different Driver Name. We recommend user add the entire driver for all the available models you connected first.*

Figure 4.1 “Add” the “New Device”.

Figure 4.2 “Driver Selection Window. Click “Add…” to next popup window.

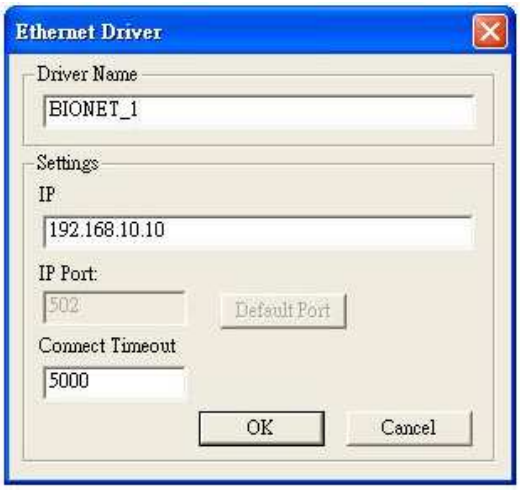


Figure 4.3 “Ethernet Driver” popup Window. Type the Driver Name and IP address for the device.

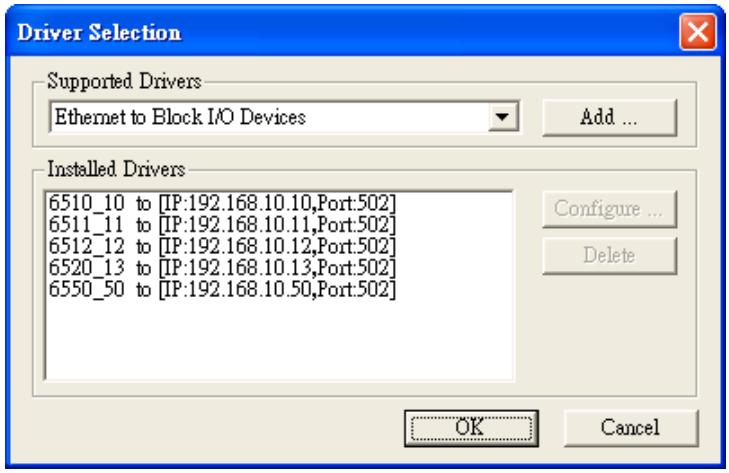
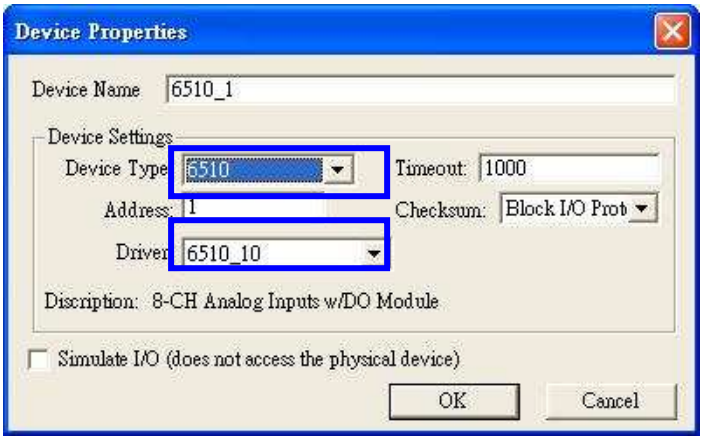
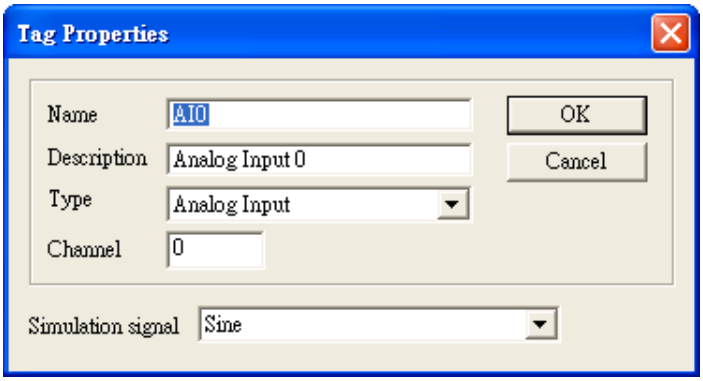


Figure 5: Example: Add all the drivers for available models. If you have 5 models over the same network, add them and give them different name for identification in next steps. Like: (Model Name/6510)_(IP address/10).

5. Type the “Device Name” and select the “Device Type” and the “Driver” in the “Device Properties” window. Device Type means the JetI/O model name. Driver is the name you configured in last step.

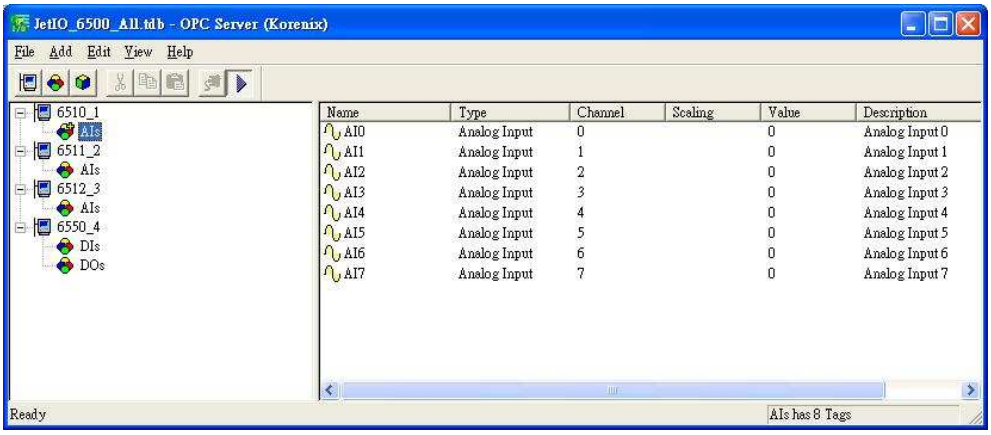


6. Select “Add -> New Group” to create new group for the later new tags you’ll create. Select “Add -> New Tag” and fill the “Tag Properties” in the popup window. Select the tag and “Edit –> Properties“, you can modify the tag properties.



*Note: The **Simulation Signal** is used when choosing Simulate I/O mode. Simulate I/O mode is selected in “Device Properties”. This feature allows you to generate simulation values and run testing when you operate the OPC client. You can see the value is continuously changed. The Sine, Ramp and Random are the different type’s simulation signal.*

7. Select the device in the device list. Choose “Add -> Generate Tags”, the utility generate all the channels’ tags for the device you choose. This step can save the time to create all channels’ tags.



Name: The name of the channel. You can manually change this value.

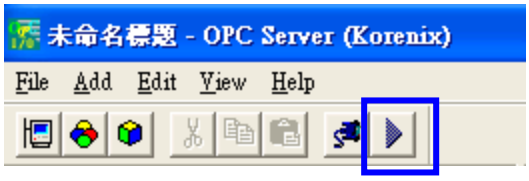
Type: The input type of the channel.

Channel: The channel ID.

Value: The value of the channel, you can use “Monitor” to read them.

Description: The description of this channel, you can manually change this value.

8. Select “View -> Monitor” to monitor the status of the tags. Or you can click the “Monitor” icon in the UI.



9. Select “File -> Save” to save the profile, then your OPC Client can monitor the Jet I/O status.

4.3 SNMP

Simple Network Management Protocol (SNMP) is a protocol used for exchanging management information between network devices. SNMP is a member of the TCP/IP protocol suite. JetI/O 6500 series acts as SNMP node, SNMP browser can discover and read/write channels' information.

An SNMP managed network consists of two main components: agents and manager. An agent is a management software module that resides in a managed switch. An agent translates the local management information from the managed device into a SNMP compatible format. The manager is the console through the network.

JetI/O 6500 series supports Public MIB: MIB II-System. This is for SNMP browser discovering. Private MIB includes channels' information. Please refer to the appendix 1 (6.1).

SNMP Trap allows the JetI/O to send the active alarm to trap servers. The SNMP Trap supports Device Cold Start, LAN interface Link Up trap (Common), Low and High Voltage/Current/Temperature (651x) and Logic Rules' traps (655x). You can configure this through Modbus/TCP registers or I/O Configuration utility.

4.4 Web UI

Type the IP address of the device. Then you can access the embedded web browser of the I/O server. The web browser allows you monitor the information/status of each channels.

4.5 How to Upgrade Firmware

The JetI/O server allows you remotely upgrade the firmware to fix the known issues or to update the new software features. Device Finder provides a user-friendly environment for firmware upgrade, which includes two modes:

1. Mode A (Firmware Upgrade): Used to upgrade the firmware of a JetI/O module which is with valid firmware and workable. Device Finder supports batch upgrade in this mode. User can upgrade more than one JetI/O device (the same model) at the same time.
2. Mode B (Firmware Rescue): Used to reload the firmware of a JetI/O without firmware.

When user starts the progress of the firmware upgrade, the JetI/O runs as the DHCP client mode to get the IP from DHCP Server and download the firmware from the server.

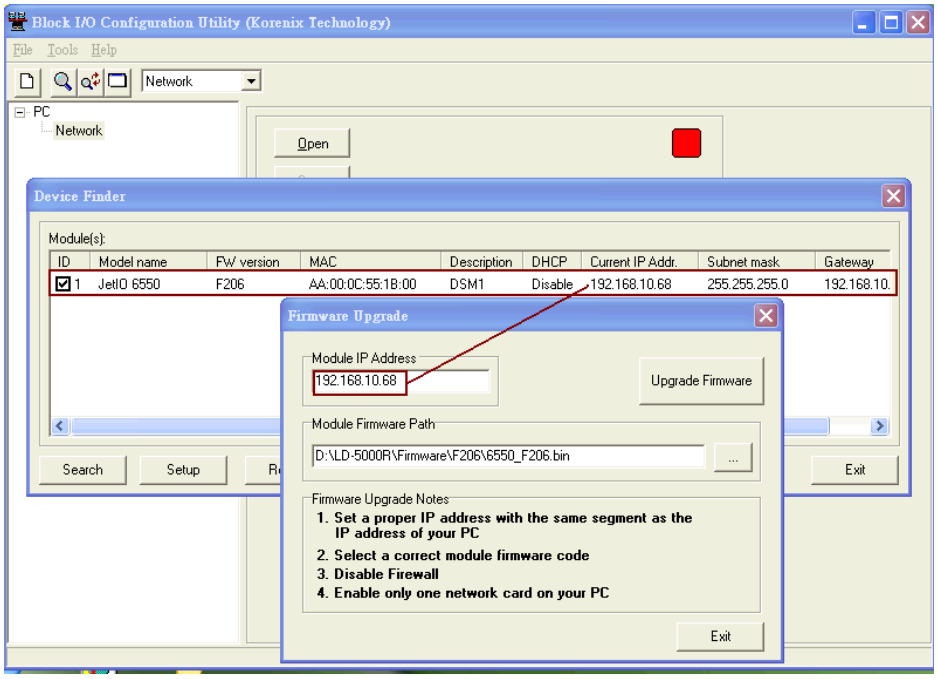
Note 1: The progress is also known as BootP, Get IP address and upgrade new firmware in the same progress. Please note that there is only one DHCP server available over the same network. Otherwise the device may get the wrong IP. Since Device Finder builds in a BOOTP server, Korenix suggests you make sure there is only one DHCP/BOOTP server on the network when you upgrade the JetIO firmware.

Upgrade Procedure for Mode A (Firmware Upgrade):

1. Lunch the Block I/O utility and then select the 'Device Finder'  from toolbar or select Device Finder from Tools menu to enable the Device Finder tool.
2. Press "Search" button to search all JetIO modules on the network and check the IP address (e.g. 192.168.10.68) of the JetIO target module.

Notes:
(a). Disable Firewall
(b). Enable only one network card on your PC
I. Set a proper IP address with the same segment as the IP address of your PC
(d). 'ON'T configure more than one IP address on the network interface.
(e). Select a correct module firmware code (i.e., 6550_Fxxx.bin for JetIO 6550)

3. Select the target module and click "Upgrade" to upload the new firmware, and then the 'Firmware Upgrade' dialog prompt you to do the further setting.
4. Select the JetIO target module from the device list of Device Finder console. You can select one more modules with the same model name to do batch upgrade.
5. Press "Upgrade" button to pop up the Firmware Upgrade dialog.
6. The default value of the "Module IP Address" field is the current IP address of the device. For batch upgrade YOU do not need to change this field.
7. Press the browser button ... to select a correct firmware code. Please do not modify the filename. Device Finder uses the filename of the firmware to identify if the firmware matches the model of the JetI/O device.
8. Press "Upgrade Firmware" button to start upgrading the new firmware code.
9. The JetIO target module should be rebooted automatically after the new JetIO firmware code was upgraded successfully.



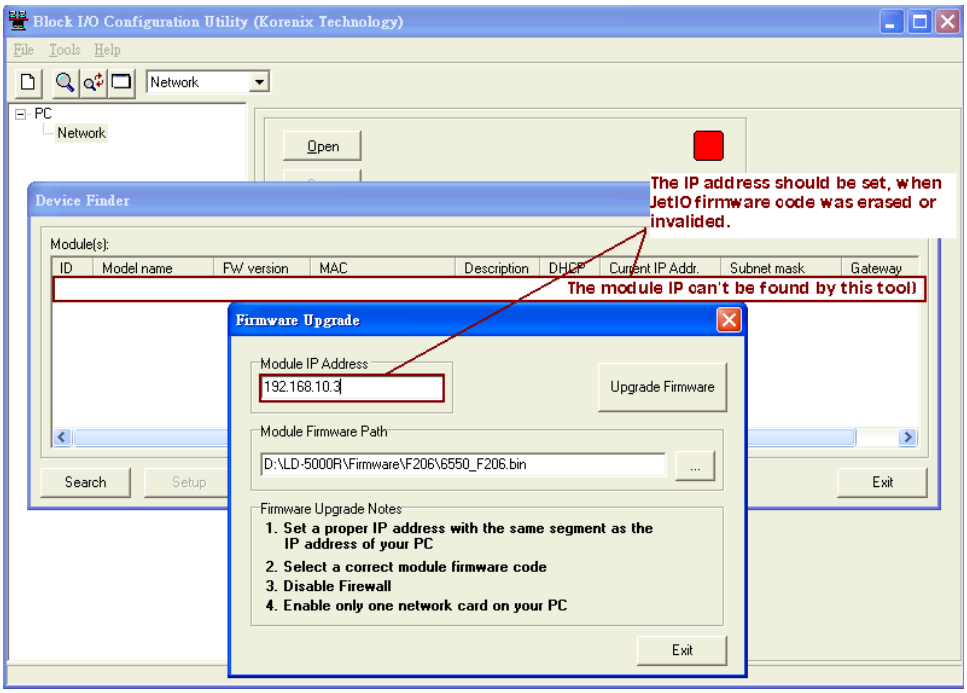
Upgrade Procedure for Mode B (Firmware Resure):

1. Lunch the Block I/O utility and then select the 'Device Finder' from toolbar or select Device Finder from Tools menu to enable the Device Finder tool.
2. A JetI/O with invalid firmware can not be found by search.

Notes:
(a). Disable Firewall
(b). Enable only one network card on your I(c). Set a proper IP address with the same segment as the IP address of your PC
(d'. DON'T configure more than one IP address on the network interface.

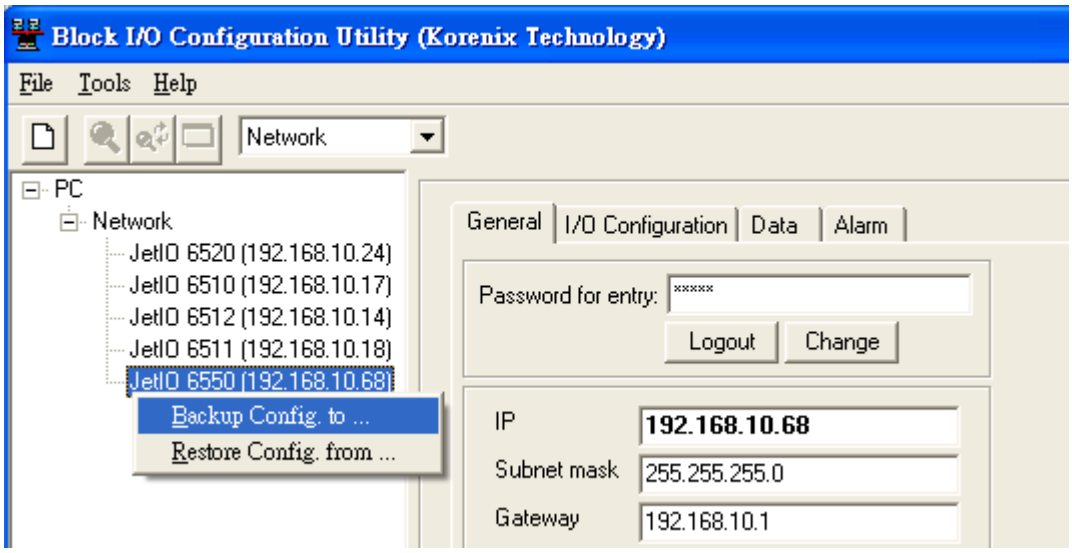
(e). Select a correct module firmware code (i.e., 6550_Fxxx.bin for JetIO 6550)

3. Press “Upgrade” button to pop up the Firmware Upgrade console.
4. Set a proper IP address for JetIO module boot loader. Please note that the IP address should be set to the same network segment of your PC.
5. Press the browser button ... to select a correct JetIO firmware code. Please do not modify the filename. Device Finder uses the filename of the firmware to identify if the firmware matches the model of the JetI/O device.
6. Press “Upgrade Firmware” button to start upgrading the new JetIO firmware code.
7. Press “Yes” to start the progress when seeing the upgrading information popup window. Press “No” to stop the progress.
8. The JetIO target module should be rebooted automatically after the new JetIO firmware code was upgraded successfully.

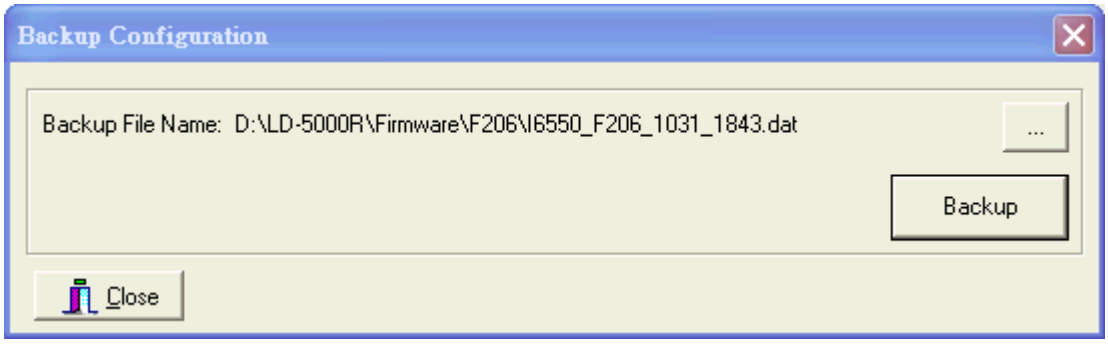


4.6 Configuration Backup/Restore, Reset Default, and Reboot

The backup/restore configuration functions are accessed by right clicking on a JetIO module in the network interface tree.



Select “Backup Config. to …” and “Restore Config. from …” command to backup and restore the configuration of the JetIO to/from a text file. It should be noticed that you will need to login first and then gain a privilege to do these functions.



Click “Reset to default” and “Reboot” in the popup menu to set factory default or reboot the device. It should be noticed that you need to “rescan network” after completing the “Reset to default” and “Reboot” activity.

Below is the related configuration information corresponding to JetIO’s Modbus registers as well as module IP configuration for your reference (i.e., internally script command to implement backup and restore configuration activity).

JetIO-6512 Module

[Backup]

0=irange.conf

1=snmp.trap.ips

2=alarm.conf

3=mis

4=#ip.conf

5=#alarm.limit.conf

[mis.PLCAddr]

0=40001:40002

1=41000:41003

[irange.conf.PLCAddr]
0=40011
1=40022
[snmp.trap.ips.PLCAddr]
0=40024:40032
[alarm.conf.PLCAddr]
0=40023
[ip.conf.Program]
key=ip.conf
var0=ip.ip
var1=ip.mask
var2=ip.gateway
var3=ip.dhcp
[alarm.limit.conf.Program]
key=alm.limit
var0=40014:40021

5 Modbus/TCP Command Set

This chapter introduces the Modbus/TCP command set JetI/O provided. When you creating application for your SCADA/HMI or coding your own programs. The command set is helpful for you to find the value of each registers.

Following topics are covered in this chapter:

5.1 Introduction of Modbus/TCP Protocol

5.2 JetI/O 6512 Modbus/TCP Address Mapping

5.1 Introduction of Modbus/TCP Protocol

5.1.1 Modbus/TCP Protocol

The Modbus protocol, developed by Gould-Modicon, is widely used in industrial communications to integrate PLC's, computer, terminals and other various I/O devices. Intelligent JetI/O Server equipped with communication interface provides an Ethernet communication links with Modbus/TCP protocol support.

Modbus/TCP is a variant of the Modbus family of communication protocol. Modbus/TCP is a Master/Slave communication protocol. A master (a host PC) initiates queries, a slave (one of the JetI/O servers) then responds by supplying the requested data to the master by using Modbus/TCP commands.

5.1.2 Function Code (FC)

The Intelligent JetI/O Server uses a subset of the standard Modbus/TCP function code to access device-dependent information. Modbus/TCP function code is defined as below.

FC	Name	Usage
01	Read Coils	Read the state of a digital output
02	Read Input Status	Read the state of a digital input
03	Read Holding Register	Read holding register in 16-bits register format
04	Read Input Registers	Read data in 16-bits register format
05	Write Coil	Write data to force a digital output ON/OFF
06	Write Single Register	Write data in 16-bits register format
15	Force Multiple Coils	Write data to force multiple consecutive coils

5.1.3 Error Checking

The utilization of the error checking will help eliminate errors caused by noise in the communication link. In Modbus/TCP mode, messages include an error-checking field that is based on a Cyclical Redundancy Check (CRC) method. The CRC filed checks the contents of the entire message. It applied regardless of any parity check method used for the individual BYTE actors of the message. The CRC value is calculated by the transmitting device, which appends the CRC to the message. The receiving device recalculates a CRC during receipt of the message, and compares the calculated value to the actual value it received in the CRC filed.

5.1.4 Exception Response

If an error occurs, the slave sends an exception response message to master consisting of the slave address, function code, exception response code and error check field. In an exception response, the slave sets the high-order bit (MSB) of the response function code to one. The exception response codes are listed below.

Code	Name	Descriptions
01	Illegal Function	The message function received is not allowable action.
02	Illegal Data Address	The address referenced in the data field is not valid.
03	Illegal Data Value	The value referenced at the addressed device location is no within range.
04	Slave Device Failure	An unrecoverable error occurred while the slave was attempting to perform the requested action.
05	Acknowledge	The slave has accepted the request and processing it, but a long duration of time will be required to do so.
06	Slave Device Busy	The slave is engaged in processing a long-duration program command.
07	Negative Acknowledge	The slave cannot perform the program function received in the query.
08	Memory Parity Error	The slave attempted to read extended memory, but detected a parity error in the memory.

5.2 JetI/O 6512 Modbus/TCP Address Mapping

Note: Some of the results are using Hex (Ex: FF00H) mode, Signed (-1Iode... or others mode. This result depends on the value of the address. There are two types Modbus/TCP addresses. The protocol address is an address from 0000 to the last address of the function code you choose. The PLC address is the fixed address number of the function code. This is also defined in Modbus/TCP protocol. Please choose the type your application uses.

Should you encounter problem on reading this, please contact our technical support engineer, korecare@korenix.com

JetI/O 6512 Common Register Map (Holding Registers, Function Code =03)

Protocol Address (Hex)	PLC Address (Decimal)	Access	Description
0000	40001	R/W	(Read/Write) Host Watch-dog enable/disable **R/W:AABB** **AABB:0000H** (disable) **AABB:FF00H** (enable)
0001	40002	R/W	(Read/Write) Host Watch-dog cycle count **R/W:AABB**

			AABB:0001H~00FFH
			(0.1*AABB)=Cycle Time (sec)
0002	40003	R/W	**R:** Read the host-watchdog status
			W: Reset the host-watchdog status
			R:AABB
			AABB:0000H (remote module OK)
			AABB:FF00H (host-watchdog fail)
			W: AABB
			AABB:FF00H(reset)
0003	40004	R	Read the firmware version
			R:AAAA
			AAAA: F203 (HEX)
0004	40005	R	Read module name
			R:AAAA
			AAAA: 6512 (HEX)
0005	40006	R	Read reset status
			R:AABB
			AABB:0000H (after using this read command)
			AABB:0001H(The value is equal to0001H after reset module)
0006	40007	R	Read AD offset Calibration Coefficients
			R:AABB(bit 16~23)
0007	40008	R	Read AD offset Calibration Coefficients
			R:00AA(bit 0~15)
0008	40009	R	Read AD span Calibration Coefficients
			R:AABB(bit 16~23)
0009	40010	R	Read AD span Calibration Coefficients
			R:AABB(bit 0~15)
000A	40011	R/W	Input code (address low)
			Code:20H~27H
			20H: pt100 (-100℃~ 100℃ α=0.00385)
			21H: pt100 (0℃~ 100℃ α=0.00385)
			22H: pt100 (0℃~ 200℃ α=0.00385)
			23H: pt100 (0℃~ 600℃ α=0.00385)
			24H: pt100 (-100℃~ 100℃ α=0.00392)
			25H: pt100 (0℃~ 100℃ α=0.00392)
			26H: pt100 (0℃~ 200℃ α=0.00392)
			27H: pt100 (0℃~ 600℃ α=0.00392)
			28H: Nickel 120Ω(-80℃ ~ 260℃

			α=0.00672)
000B	40012	R/W	Offset (Zero) calibration (R: no used) **W: AABB** **AABB:FF00H** **Note: timeout time 300 ms at least** (Warning: You should calibrate the value for the selected input range by the certificated calibrator when need.)
000C	40013	R/W	Span calibration (R: no used) **W: AABB** **AABB:FF00H** **Note: timeout time 300 ms at least** (Warning: You should calibrate the value for the selected input range by the certificated calibrator when need.)

JetI/O 6512 Special Register Map (Holding Registers)

000D	40014	R/W	Read/write Channel 0 Low alarm value R/W:AABB
000E	40015	R/W	Read/write Channel 0 High alarm value R/W:AABB
000F	40016	R/W	Read/write Channel 1 Low alarm value R/W:AABB
0010	40017	R/W	Read/write Channel 1 High alarm value R/W:AABB
0011	40018	R/W	Read/write Channel 2 Low alarm value R/W:AABB
0012	40019	R/W	Read/write Channel 2 High alarm value R/W:AABB
0013	40020	R/W	Read/write Channel 3 Low alarm value R/W:AABB
0014	40021	R/W	Read/write Channel 3 High alarm value R/W:AABB
0015	40022	R/W	Read/write masked AD-channels of the module R/W:AABB AA:00 BB:0000XXXX(Binary) X: 1 Enable X: 0 Disable
0016	40023	R/W	Enable / Disable alarm status R/W: 0000xxxx (Binary)

			X : 1 Enable X: 0 Disable Least bit means the channel 0
0017	40024	R/W	Read/write SNMP Trap Number **R/W:AAAA** AAAA:0000~0004 0000: close SNMP trap
0018	40025	R/W	(Read/Write)SNMP Trap IP1 Lo-Word **R: AABB(hex)** **W: AABB(hex)** IP=X.X.AA.BB
0019	40026	R/W	(Read/Write)SNMP Trap IP1 Hi-Word **R: AAB** R/W **B(hex)** **W: AABB(hex)** IP=AA.BB.X.X
001A	40027	R/W	(Read/Write)SNMP Trap IP2 Lo-Word **R: AABB(hex)** **W: AABB(hex)** IP=X.X.AA.BB
001B	40028	R/W	(Read/Write)SNMP Trap IP2 Hi-Word **R: AABB(hex)** **W: AABB(hex)** IP=AA.BB.X.X
001C	40029	R/W	(Read/Write)SNMP Trap IP3 Lo-Word **R: AABB(hex)** **W: AABB(hex)** IP=X.X.AA.BB
001D	40030	R/W	(Read/Write)SNMP Trap IP3 Hi-Word **R: AABB(hex)** **W: AABB(hex)** IP=AA.BB.X.X
001E	40031	R/W	(Read/Write)SNMP Trap IP4 Lo-Word **R: AABB(hex)** **W: AABB(hex)** IP=X.X.AA.BB
001F	40032	R/W	(Read/Write)SNMP Trap IP4 Hi-Word **R: AABB(hex)** **W: AABB(hex)** IP=AA.BB.X.X

0020	40033	R/W	(Write) Repeat enable SNMP Trap **W: FF00(hex)**
0021	40034	R	Read AD offset Calibration Coefficients **R:AABB(bit 16~23)** (factory calibration value)
0022	40035	R	Read AD offset Calibration Coefficients **R:00AA(bit 0~15)** (factory calibration value)
0023	40036	R	Read AD span Calibration Coefficients **R:AABB(bit 16~23)** (factory calibration value)
0024	40037	R	Read AD span Calibration Coefficients **R:00AA(bit 0~15)** (factory calibration value)
01F4	40501	R/W	(Read/Write) RuleEnable Logic 0~15 R: xxxxxxxxxxxxxxxx(bit) W: xxxxxxxxxxxxxxxx(bit) 0: disable 1: enable (write to Flash RAM)
01F4	40501	R/W	(Read/Write) RuleEnable Logic 0~15 R: xxxxxxxxxxxxxxxx(bit) W: xxxxxxxxxxxxxxxx(bit) 0: disable 1: enable (write to Flash RAM)
01F5	40502	R/W	(Read/Write) Select “OR” or “AND” R:AAAA(hex) W: AAAA (hex) AAAA:xxxxxxxxxxxxxxxx X: 0 “OR” Logic 1 “AND” Logic (write to Flash RAM)
01F6	40503	R/W	(Read/Write) Select condition AI channel (or Auxflag channel) for Rule 0 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic AI Channel B: condition #2 Logic AI Channel C: condition #1 Logic AI Channel D: condition #0 Logic AI Channel(write to Flash RAM)
01F7	40504	R/W	(Read/Write) Select condition AI channel (or Auxflag channel) for Rule

			1 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic AI Channel B: condition #2 Logic AI Channel C: condition #1 Logic AI Channel D: condition #0 Logic AI Channel(write to Flash RAM)
01F8	40505	R/W	(Read/Write) Select condition AI channel (or Auxflag channel) for Rule 2 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic AI Channel B: condition #2 Logic AI Channel C: condition #1 Logic AI Channel D: condition #0 Logic AI Channel(write to Flash RAM)
01F9	40506	R/W	(Read/Write) Select condition AI channel (or Auxflag channel) for Rule 3 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic AI Channel B: condition #2 Logic AI Channel C: condition #1 Logic AI Channel D: condition #0 Logic AI Channel(write to Flash RAM)
01FA	40507	R/W	(Read/Write) Select condition AI channel (or Auxflag channel) for Rule 4 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic AI Channel B: condition #2 Logic AI Channel C: condition #1 Logic AI Channel D: condition #0 Logic AI Channel(write to Flash RAM)

01FB	40508	R/W	(Read/Write) Select condition AI channel (or Auxflag channel) for Rule 5 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic AI Channel B: condition #2 Logic AI Channel C: condition #1 Logic AI Channel D: condition #0 Logic AI Channel(write to Flash RAM)
01FC	40509	R/W	(Read/Write) Select condition AI channel (or Auxflag channel) for Rule 6 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic AI Channel B: condition #2 Logic AI Channel C: condition #1 Logic AI Channel D: condition #0 Logic AI Channel(write to Flash RAM)
01FD	40510	R/W	(Read/Write) Select condition AI channel (or Auxflag channel) for Rule 7 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic AI Channel B: condition #2 Logic AI Channel C: condition #1 Logic AI Channel D: condition #0 Logic AI Channel(write to Flash RAM)
01FE	40511	R/W	(Read/Write) Select condition AI channel (or Auxflag channel) for Rule 8 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic AI Channel B: condition #2 Logic AI Channel C: condition #1 Logic AI Channel D: condition #0 Logic AI

			Channel(write to Flash RAM)
01FF	40512	R/W	(Read/Write) Select condition AI channel (or3Auxflag channel) for Rule 9 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic AI Channel B: condition #2 Logic AI Channel C: condition #1 Logic AI Channel D: condition #0 Logic AI
			Channel(write to Flash RAM)
0200	40513	R/W	(Read/Write) Select condition AI channel (or Auxflag channel) for Rule 10 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic AI Channel B: condition #2 Logic AI Channel C: condition #1 Logic AI Channel D: condition #0 Logic AI
			Channel(write to Flash RAM)
0201	40514	R/W	(Read/Write) Select condition AI channel (or Auxflag channel) for Rule 11 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic AI Channel B: condition #2 Logic AI Channel C: condition #1 Logic AI Channel D: condition #0 Logic AI
			Channel(write to Flash RAM)
0202	40515	R/W	(Read/Write) Select condition AI channel (or Auxflag channel) for Rule 12 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic AI Channel B: condition #2 Logic AI Channel

			C: condition #1 Logic AI Channel D: condition #0 Logic AI Channel(write to Flash RAM)
0203	40516	R/W	(Read/Write) Select condition AI channel (or3Auxflag channel) for Rule 13 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic AI Channel B: condition #2 Logic AI Channel C: condition #1 Logic AI Channel D: condition #0 Logic AI Channel(write to Flash RAM)
0204	40517	R/W	(Read/Write) Select condition AI channel (or Auxflag channel) for Rule 14 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic AI Channel B: condition #2 Logic AI Channel C: condition #1 Logic AI Channel D: condition #0 Logic AI Channel(write to Flash RAM)
0205	40518	R/W	(Read/Write) Select condition AI channel (or Auxflag channel) for Rule 15 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic AI Channel B: condition #2 Logic AI Channel C: condition #1 Logic AI Channel D: condition #0 Logic AI Channel(write to Flash RAM)
0206	40519	R/W	(Read/Write) Select Operators condition for Rule 0 R: ABCD(hex) W: ABCD(hex) A: Operators condition #3 Logic

			B: Operators condition #2 Logic C: Operators condition #1 Logic D: Operators condition #0 Logic Value: 0 : ON (Auxflag) 1:OFF (Auxflag) 2:Lo-Hi (Auxflag) 3:Hi-Lo (Auxflag) 4: “=” (AIValue = SetAIValue) 5:”>” (AIValue > SetAIValue) 6:”<” (AIValue < SetAIValue) 7:”>=” (AIValue >= SetAIValue) 8:”<=” (AIValue <= SetAIValue) (write to Flash RAM)
0207	40520	R/W	(Read/Write) Select Operators condition for Rule 1 R: ABCD(hex) W: ABCD(hex) A: Operators condition #3 Logic B: Operators condition #2 Logic C: Operators condition #1 Logic D: Operators condition #0 Logic Value: 0 : ON (Auxflag) 1:OFF (Auxflag) 2:Lo-Hi (Auxflag) 3:Hi-Lo (Auxflag) 4: “=” (AIValue = SetAIValue) 5:”>” (AIValue > SetAIValue) 6:”<” (AIValue < SetAIValue) 7:”>=” (AIValue >= SetAIValue) 8:”<=” (AIValue <= SetAIValue) (write to Flash RAM)
0208	40521	R/W	(Read/Write) Select Operators condition for Rule 2 R: ABCD(hex) W: ABCD(hex) A: Operators condition #3 Logic

			B: Operators condition #2 Logic C: Operators condition #1 Logic D: Operators condition #0 Logic Value: 0 : ON (Auxflag) 1:OFF (Auxflag) 2:Lo-Hi (Auxflag) 3:Hi-Lo (Auxflag) 4: “=” (AIValue = SetAIValue) 5:”>” (AIValue > SetAIValue) 6:”<” (AIValue < SetAIValue) 7:”>=” (AIValue >= SetAIValue) 8:”<=” (AIValue <= SetAIValue) (write to Flash RAM)
0209	40522	R/W	(Read/Write) Select Operators condition for Rule 3 R: ABCD(hex) W: ABCD(hex) A: Operators condition #3 Logic B: Operators condition #2 Logic C: Operators condition #1 Logic D: Operators condition #0 Logic Value: 0 : ON (Auxflag) 1:OFF (Auxflag) 2:Lo-Hi (Auxflag) 3:Hi-Lo (Auxflag) 4: “=” (AIValue = SetAIValue) 5:”>” (AIValue > SetAIValue) 6:”<” (AIValue < SetAIValue) 7:”>=” (AIValue >= SetAIValue) 8:”<=” (AIValue <= SetAIValue) (write to Flash RAM)
020A	40523	R/W	(Read/Write) Select Operators condition for Rule 4 R: ABCD(hex) W: ABCD(hex) A: Operators condition #3 Logic

			B: Operators condition #2 Logic C: Operators condition #1 Logic D: Operators condition #0 Logic Value: 0 : ON (Auxflag) 1:OFF (Auxflag) 2:Lo-Hi (Auxflag) 3:Hi-Lo (Auxflag) 4: “=” (AIValue = SetAIValue) 5:”>” (AIValue > SetAIValue) 6:”<” (AIValue < SetAIValue) 7:”>=” (AIValue >= SetAIValue) 8:”<=” (AIValue <= SetAIValue) (write to Flash RAM)
020B	40524	R/W	(Read/Write) Select Operators condition for Rule 5 R: ABCD(hex) W: ABCD(hex) A: Operators condition #3 Logic B: Operators condition #2 Logic C: Operators condition #1 Logic D: Operators condition #0 Logic Value: 0 : ON (Auxflag) 1:OFF (Auxflag) 2:Lo-Hi (Auxflag) 3:Hi-Lo (Auxflag) 4: “=” (AIValue = SetAIValue) 5:”>” (AIValue > SetAIValue) 6:”<” (AIValue < SetAIValue) 7:”>=” (AIValue >= SetAIValue) 8:”<=” (AIValue <= SetAIValue) (write to Flash RAM)
020C	40525	R/W	(Read/Write) Select Operators condition for Rule 6 R: ABCD(hex) W: ABCD(hex) A: Operators condition #3 Logic

			B: Operators condition #2 Logic C: Operators condition #1 Logic D: Operators condition #0 Logic Value: 0 : ON (Auxflag) 1:OFF (Auxflag) 2:Lo-Hi (Auxflag) 3:Hi-Lo (Auxflag) 4: “=” (AIValue = SetAIValue) 5:”>” (AIValue > SetAIValue) 6:”<” (AIValue < SetAIValue) 7:”>=” (AIValue >= SetAIValue) 8:”<=” (AIValue <= SetAIValue) (write to Flash RAM)
020D	40526	R/W	(Read/Write) Select Operators condition for Rule 7 R: ABCD(hex) W: ABCD(hex) A: Operators condition #3 Logic B: Operators condition #2 Logic C: Operators condition #1 Logic D: Operators condition #0 Logic Value: 0 : ON (Auxflag) 1:OFF (Auxflag) 2:Lo-Hi (Auxflag) 3:Hi-Lo (Auxflag) 4: “=” (AIValue = SetAIValue) 5:”>” (AIValue > SetAIValue) 6:”<” (AIValue < SetAIValue) 7:”>=” (AIValue >= SetAIValue) 8:”<=” (AIValue <= SetAIValue) (write to Flash RAM)
020E	40527	R/W	(Read/Write) Select Operators condition for Rule 8 R: ABCD(hex) W: ABCD(hex) A: Operators condition #3 Logic

			B: Operators condition #2 Logic C: Operators condition #1 Logic D: Operators condition #0 Logic Value: 0 : ON (Auxflag) 1:OFF (Auxflag) 2:Lo-Hi (Auxflag) 3:Hi-Lo (Auxflag) 4: “=” (AIValue = SetAIValue) 5:”>” (AIValue > SetAIValue) 6:”<” (AIValue < SetAIValue) 7:”>=” (AIValue >= SetAIValue) 8:”<=” (AIValue <= SetAIValue) (write to Flash RAM)
020F	40528	R/W	(Read/Write) Select Operators condition for Rule 9 R: ABCD(hex) W: ABCD(hex) A: Operators condition #3 Logic B: Operators condition #2 Logic C: Operators condition #1 Logic D: Operators condition #0 Logic Value: 0 : ON (Auxflag) 1:OFF (Auxflag) 2:Lo-Hi (Auxflag) 3:Hi-Lo (Auxflag) 4: “=” (AIValue = SetAIValue) 5:”>” (AIValue > SetAIValue) 6:”<” (AIValue < SetAIValue) 7:”>=” (AIValue >= SetAIValue) 8:”<=” (AIValue <= SetAIValue) (write to Flash RAM)
0210	40529	R/W	(Read/Write) Select Operators condition for Rule 10 R: ABCD(hex) W: ABCD(hex) A: Operators condition #3 Logic

			B: Operators condition #2 Logic C: Operators condition #1 Logic D: Operators condition #0 Logic Value: 0 : ON (Auxflag) 1:OFF (Auxflag) 2:Lo-Hi (Auxflag) 3:Hi-Lo (Auxflag) 4: “=” (AIValue = SetAIValue) 5:”>” (AIValue > SetAIValue) 6:”<” (AIValue < SetAIValue) 7:”>=” (AIValue >= SetAIValue) 8:”<=” (AIValue <= SetAIValue) (write to Flash RAM)
0211	40530	R/W	(Read/Write) Select Operators condition for Rule 11 R: ABCD(hex) W: ABCD(hex) A: Operators condition #3 Logic B: Operators condition #2 Logic C: Operators condition #1 Logic D: Operators condition #0 Logic Value: 0 : ON (Auxflag) 1:OFF (Auxflag) 2:Lo-Hi (Auxflag) 3:Hi-Lo (Auxflag) 4: “=” (AIValue = SetAIValue) 5:”>” (AIValue > SetAIValue) 6:”<” (AIValue < SetAIValue) 7:”>=” (AIValue >= SetAIValue) 8:”<=” (AIValue <= SetAIValue) (write to Flash RAM)
0212	40531	R/W	(Read/Write) Select Operators condition for Rule 12 R: ABCD(hex) W: ABCD(hex) A: Operators condition #3 Logic

			B: Operators condition #2 Logic C: Operators condition #1 Logic D: Operators condition #0 Logic Value: 0 : ON (Auxflag) 1:OFF (Auxflag) 2:Lo-Hi (Auxflag) 3:Hi-Lo (Auxflag) 4: “=” (AIValue = SetAIValue) 5:”>” (AIValue > SetAIValue) 6:”<” (AIValue < SetAIValue) 7:”>=” (AIValue >= SetAIValue) 8:”<=” (AIValue <= SetAIValue) (write to Flash RAM)
0213	40532	R/W	(Read/Write) Select Operators condition for Rule 13 R: ABCD(hex) W: ABCD(hex) A: Operators condition #3 Logic B: Operators condition #2 Logic C: Operators condition #1 Logic D: Operators condition #0 Logic Value: 0 : ON (Auxflag) 1:OFF (Auxflag) 2:Lo-Hi (Auxflag) 3:Hi-Lo (Auxflag) 4: “=” (AIValue = SetAIValue) 5:”>” (AIValue > SetAIValue) 6:”<” (AIValue < SetAIValue) 7:”>=” (AIValue >= SetAIValue) 8:”<=” (AIValue <= SetAIValue) (write to Flash RAM)
0214	40533	R/W	(Read/Write) Select Operators condition for Rule 14 R: ABCD(hex) W: ABCD(hex) A: Operators condition #3 Logic

			B: Operators condition #2 Logic C: Operators condition #1 Logic D: Operators condition #0 Logic Value: 0 : ON (Auxflag) 1:OFF (Auxflag) 2:Lo-Hi (Auxflag) 3:Hi-Lo (Auxflag) 4: “=” (AIValue = SetAIValue) 5:”>” (AIValue > SetAIValue) 6:”<” (AIValue < SetAIValue) 7:”>=” (AIValue >= SetAIValue) 8:”<=” (AIValue <= SetAIValue) (write to Flash RAM)
0215	40534	R/W	(Read/Write) Select Operators condition for Rule 15 R: ABCD(hex) W: ABCD(hex) A: Operators condition #3 Logic B: Operators condition #2 Logic C: Operators condition #1 Logic D: Operators condition #0 Logic Value: 0 : ON (Auxflag) 1:OFF (Auxflag) 2:Lo-Hi (Auxflag) 3:Hi-Lo (Auxflag) 4: “=” (AIValue = SetAIValue) 5:”>” (AIValue > SetAIValue) 6:”<” (AIValue < SetAIValue) 7:”>=” (AIValue >= SetAIValue) 8:”<=” (AIValue <= SetAIValue) (write to Flash RAM)
0216	40535	R/W	(Read/Write) Select THEN Auxflag action for Rule 0 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic Auxflag

			Channel B: condition #2 Logic Auxflag Channel C: condition #1 Logic Auxflag Channel D: condition #0 Logic Auxflag Channel(write to Flash RAM)
0217	40536	R/W	(Read/Write) Select THEN Auxflag action for Rule 1 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic Auxflag Channel B: condition #2 Logic Auxflag Channel C: condition #1 Logic Auxflag Channel D: condition #0 Logic Auxflag Channel(write to Flash RAM)
0218	40537	R/W	(Read/Write) Select THEN Auxflag action for Rule 2 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic Auxflag Channel B: condition #2 Logic Auxflag Channel C: condition #1 Logic Auxflag Channel D: condition #0 Logic Auxflag Channel(write to Flash RAM)
0219	40538	R/W	(Read/Write) Select THEN Auxflag action for Rule 3 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic Auxflag Channel B: condition #2 Logic Auxflag

			Channel C: condition #1 Logic Auxflag Channel D: condition #0 Logic Auxflag Channel(write to Flash RAM)
021A	40539	R/W	(Read/Write) Select THEN Auxflag action for Rule 4 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic Auxflag Channel B: condition #2 Logic Auxflag Channel C: condition #1 Logic Auxflag Channel D: condition #0 Logic Auxflag Channel(write to Flash RAM)
021B	40540	R/W	(Read/Write) Select THEN Auxflag action for Rule 5 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic Auxflag Channel B: condition #2 Logic Auxflag Channel C: condition #1 Logic Auxflag Channel D: condition #0 Logic Auxflag Channel(write to Flash RAM)
021C	40541	R/W	(Read/Write) Select THEN Auxflag action for Rule 6 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic Auxflag Channel B: condition #2 Logic Auxflag Channel C: condition #1 Logic Auxflag

			Channel D: condition #0 Logic Auxflag Channel(write to Flash RAM)
021D	40542	R/W	(Read/Write) Select THEN Auxflag action for Rule 7 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic Auxflag Channel B: condition #2 Logic Auxflag Channel C: condition #1 Logic Auxflag Channel D: condition #0 Logic Auxflag Channel(write to Flash RAM)
021E	40543	R/W	(Read/Write) Select THEN Auxflag action for Rule 8 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic Auxflag Channel B: condition #2 Logic Auxflag Channel C: condition #1 Logic Auxflag Channel D: condition #0 Logic Auxflag Channel(write to Flash RAM)
021F	40544	R/W	(Read/Write) Select THEN Auxflag action for Rule 9 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic Auxflag Channel B: condition #2 Logic Auxflag Channel C: condition #1 Logic Auxflag Channel D: condition #0 Logic Auxflag

			Channel(write to Flash RAM)
0220	40545	R/W	(Read/Write) Select THEN Auxflag action for Rule 10 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic Auxflag Channel B: condition #2 Logic Auxflag Channel C: condition #1 Logic Auxflag Channel D: condition #0 Logic Auxflag Channel(write to Flash RAM)
0221	40546	R/W	(Read/Write) Select THEN Auxflag action for Rule 11 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic Auxflag Channel B: condition #2 Logic Auxflag Channel C: condition #1 Logic Auxflag Channel D: condition #0 Logic Auxflag Channel(write to Flash RAM)
0222	40547	R/W	(Read/Write) Select THEN Auxflag action for Rule 12 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic Auxflag Channel B: condition #2 Logic Auxflag Channel C: condition #1 Logic Auxflag Channel D: condition #0 Logic Auxflag Channel(write to Flash RAM)
0223	40548	R/W	(Read/Write) Select THEN Auxflag

			action for Rule 13 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic Auxflag Channel B: condition #2 Logic Auxflag Channel C: condition #1 Logic Auxflag Channel D: condition #0 Logic Auxflag Channel(write to Flash RAM)
0224	40549	R/W	(Read/Write) Select THEN Auxflag action for Rule 14 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic Auxflag Channel B: condition #2 Logic Auxflag Channel C: condition #1 Logic Auxflag Channel D: condition #0 Logic Auxflag Channel(write to Flash RAM)
0225	40550	R/W	(Read/Write) Select THEN Auxflag action for Rule 15 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic Auxflag Channel B: condition #2 Logic Auxflag Channel C: condition #1 Logic Auxflag Channel D: condition #0 Logic Auxflag Channel(write to Flash RAM)
0226	40551	R/W	(Read/Write) Select Operators Action for Rule 0

			R: ABCD(hex) W: ABCD(hex) A: Operators Action #3 Logic B: Operators Action #2 Logic C: Operators Action #1 Logic D: Operators Action #0 Logic Value: 0 : ON (Auxflag) 1: OFF (Auxflag) 2: reserved 3: reserved 4: reserved 5: SNMP Trap (write to Flash RAM)
0227	40552	R/W	(Read/Write) Select Operators Action for Rule 1 R: ABCD(hex) W: ABCD(hex) A: Operators Action #3 Logic B: Operators Action #2 Logic C: Operators Action #1 Logic D: Operators Action #0 Logic Value: 0 : ON (Auxflag) 1: OFF (Auxflag) 2: reserved 3: reserved 4: reserved 5: SNMP Trap (write to Flash RAM)
0228	40553	R/W	(Read/Write) Select Operators Action for Rule 2 R: ABCD(hex) W: ABCD(hex) A: Operators Action #3 Logic B: Operators Action #2 Logic C: Operators Action #1 Logic D: Operators Action #0 Logic Value: 0 : ON (Auxflag)

			1: OFF (Auxflag) 2: reserved 3: reserved 4: reserved 5: SNMP Trap (write to Flash RAM)
0229	40554	R/W	(Read/Write) Select Operators Action for Rule 3 R: ABCD(hex) W: ABCD(hex) A: Operators Action #3 Logic B: Operators Action #2 Logic C: Operators Action #1 Logic D: Operators Action #0 Logic Value: 0 : ON (Auxflag) 1: OFF (Auxflag) 2: reserved 3: reserved 4: reserved 5: SNMP Trap (write to Flash RAM)
022A	40555	R/W	(Read/Write) Select Operators Action for Rule 4 R: ABCD(hex) W: ABCD(hex) A: Operators Action #3 Logic B: Operators Action #2 Logic C: Operators Action #1 Logic D: Operators Action #0 Logic Value: 0 : ON (Auxflag) 1: OFF (Auxflag) 2: reserved 3: reserved 4: reserved 5: SNMP Trap (write to Flash RAM)
022B	40556	R/W	(Read/Write) Select Operators Action for Rule 5 R: ABCD(hex)

			W: ABCD(hex) A: Operators Action #3 Logic B: Operators Action #2 Logic C: Operators Action #1 Logic D: Operators Action #0 Logic Value: 0 : ON (Auxflag) 1: OFF (Auxflag) 2: reserved 3: reserved 4: reserved 5: SNMP Trap (write to Flash RAM)
022C	40557	R/W	(Read/Write) Select Operators Action for Rule 6 R: ABCD(hex) W: ABCD(hex) A: Operators Action #3 Logic B: Operators Action #2 Logic C: Operators Action #1 Logic D: Operators Action #0 Logic Value: 0 : ON (Auxflag) 1: OFF (Auxflag) 2: reserved 3: reserved 4: reserved 5: SNMP Trap (write to Flash RAM)
022D	40558	R/W	(Read/Write) Select Operators Action for Rule 7 R: ABCD(hex) W: ABCD(hex) A: Operators Action #3 Logic B: Operators Action #2 Logic C: Operators Action #1 Logic D: Operators Action #0 Logic Value: 0 : ON (Auxflag) 1: OFF (Auxflag)

			2: reserved 3: reserved 4: reserved 5: SNMP Trap (write to Flash RAM)
022E	40559	R/W	(Read/Write) Select Operators Action for Rule 8 R: ABCD(hex) W: ABCD(hex) A: Operators Action #3 Logic B: Operators Action #2 Logic C: Operators Action #1 Logic D: Operators Action #0 Logic Value: 0 : ON (Auxflag) 1: OFF (Auxflag) 2: reserved 3: reserved 4: reserved 5: SNMP Trap (write to Flash RAM)
022F	40560	R/W	(Read/Write) Select Operators Action for Rule 9 R: ABCD(hex) W: ABCD(hex) A: Operators Action #3 Logic B: Operators Action #2 Logic C: Operators Action #1 Logic D: Operators Action #0 Logic Value: 0 : ON (Auxflag) 1: OFF (Auxflag) 2: reserved 3: reserved 4: reserved 5: SNMP Trap (write to Flash RAM)
0230	40561	R/W	(Read/Write) Select Operators Action for Rule 10 R: ABCD(hex) W: ABCD(hex)

			A: Operators Action #3 Logic B: Operators Action #2 Logic C: Operators Action #1 Logic D: Operators Action #0 Logic Value: 0 : ON (Auxflag) 1: OFF (Auxflag) 2: reserved 3: reserved 4: reserved 5: SNMP Trap (write to Flash RAM)
0231	40562	R/W	(Read/Write) Select Operators Action for Rule 11 R: ABCD(hex) W: ABCD(hex) A: Operators Action #3 Logic B: Operators Action #2 Logic C: Operators Action #1 Logic D: Operators Action #0 Logic Value: 0 : ON (Auxflag) 1: OFF (Auxflag) 2: reserved 3: reserved 4: reserved 5: SNMP Trap (write to Flash RAM)
0232	40563	R/W	(Read/Write) Select Operators Action for Rule 12 R: ABCD(hex) W: ABCD(hex) A: Operators Action #3 Logic B: Operators Action #2 Logic C: Operators Action #1 Logic D: Operators Action #0 Logic Value: 0 : ON (Auxflag) 1: OFF (Auxflag) 2: reserved

			3: reserved 4: reserved 5: SNMP Trap (write to Flash RAM)
0233	40564	R/W	(Read/Write) Select Operators Action for Rule 13 R: ABCD(hex) W: ABCD(hex) A: Operators Action #3 Logic B: Operators Action #2 Logic C: Operators Action #1 Logic D: Operators Action #0 Logic Value: 0 : ON (Auxflag) 1: OFF (Auxflag) 2: reserved 3: reserved 4: reserved 5: SNMP Trap (write to Flash RAM)
0234	40565	R/W	(Read/Write) Select Operators Action for Rule 14 R: ABCD(hex) W: ABCD(hex) A: Operators Action #3 Logic B: Operators Action #2 Logic C: Operators Action #1 Logic D: Operators Action #0 Logic Value: 0 : ON (Auxflag) 1: OFF (Auxflag) 2: reserved 3: reserved 4: reserved 5: SNMP Trap (write to Flash RAM)
0235	40566	R/W	(Read/Write) Select Operators Action for Rule 15 R: ABCD(hex) W: ABCD(hex) A: Operators Action #3 Logic

			B: Operators Action #2 Logic C: Operators Action #1 Logic D: Operators Action #0 Logic Value: 0 : ON (Auxflag) 1: OFF (Auxflag) 2: reserved 3: reserved 4: reserved 5: SNMP Trap (write to Flash RAM)
0236	40567	R/W	(Read/Write) Condition Enable for Rule 0~3 R: ABCD(hex) W: ABCD(hex) A: xxxx x= 0:disable 1:enable rule 3 B: xxxx x= 0:disable 1:enable rule 2 C: xxxx x= 0:disable 1:enable rule 1 D: xxxx x= 0:disable 1:enable rule 0 (write to Flash RAM)
0237	40568	R/W	(Read/Write) Condition Enable for Rule 4~7 R: ABCD(hex) W: ABCD(hex) A: xxxx x= 0:disable 1:enable rule 7 B: xxxx x= 0:disable 1:enable rule 6 C: xxxx x= 0:disable 1:enable rule 5 D: xxxx x= 0:disable 1:enable rule 4 (write to Flash RAM)
0238	40569	R/W	(Read/Write) Condition Enable for Rule 8~11 R: ABCD(hex) W: ABCD(hex) A: xxxx x= 0:disable 1:enable rule 11

			B: xxxx x= 0:disable 1:enable rule 10 C: xxxx x= 0:disable 1:enable rule 9 D: xxxx x= 0:disable 1:enable rule 8 (write to Flash RAM)
0239	40570	R/W	(Read/Write) Condition Enable for Rule 12~15 R: ABCD(hex) W: ABCD(hex) A: xxxx x= 0:disable 1:enable rule 15 B: xxxx x= 0:disable 1:enable rule 14 C: xxxx x= 0:disable 1:enable rule 12 D: xxxx x= 0:disable 1:enable rule 12 (write to Flash RAM)
023A	40571	R/W	(Read/Write) Action Enable for Rule 0~3 R: ABCD(hex) W: ABCD(hex) A: xxxx x= 0:disable 1:enable rule 3 B: xxxx x= 0:disable 1:enable rule 2 C: xxxx x= 0:disable 1:enable rule 1 D: xxxx x= 0:disable 1:enable rule 0 (write to Flash RAM)
023B	40572	R/W	(Read/Write) Action Enable for Rule 4~7 R: ABCD(hex) W: ABCD(hex) A: xxxx x= 0:disable 1:enable rule 7 B: xxxx x= 0:disable 1:enable rule 6 C: xxxx x= 0:disable 1:enable rule 5

			D: xxxx x= 0:disable 1:enable rule 4 (write to Flash RAM)
023C	40573	R/W	(Read/Write) Action Enable for Rule 8~11 R: ABCD(hex) W: ABCD(hex) A: xxxx x= 0:disable 1:enable rule 11 B: xxxx x= 0:disable 1:enable rule 10 C: xxxx x= 0:disable 1:enable rule 9 D: xxxx x= 0:disable 1:enable rule 8 (write to Flash RAM)
023D	40574	R/W	(Read/Write) Action Enable for Rule 12~15 R: ABCD(hex) W: ABCD(hex) A: xxxx x= 0:disable 1:enable rule 15 B: xxxx x= 0:disable 1:enable rule 14 C: xxxx x= 0:disable 1:enable rule 13 D: xxxx x= 0:disable 1:enable rule 12 (write to Flash RAM)
023E	40575	R/W	(Read/Write) set “if logic” AI value for rule 0 #0 R/W: ABCD (write to Flash RAM)
023F	40576	R/W	(Read/Write) set “if logic” AI value for rule 0 #1 R/W: ABCD (write to Flash RAM)
0240	40577	R/W	(Read/Write) set “if logic” AI value for rule 0 #2 R/W: ABCD (write to Flash RAM)
0241	40578	R/W	(Read/Write) set “if logic” AI value for

			rule 0 #3
0242	40579	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 1 #0
0243	40580	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 1 #1
0244	40581	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 1 #2
0245	40582	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 1 #3
0246	40583	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 2 #0
0247	40584	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 2 #1
0248	40585	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 2 #2
0249	40586	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 2 #3
024A	40587	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 3 #0
024B	40588	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 3 #1
024C	40589	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 3 #2
024D	40590	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 3 #3
			R/W: ABCD (write to Flash RAM)

024E	40591	R/W	(Read/Write) set “if logic” AI value for rule 4 #0 R/W: ABCD (write to Flash RAM)
024F	40592	R/W	(Read/Write) set “if logic” AI value for rule 4 #1 R/W: ABCD (write to Flash RAM)
0250	40593	R/W	(Read/Write) set “if logic” AI value for rule 4 #2 R/W: ABCD (write to Flash RAM)
0251	40594	R/W	(Read/Write) set “if logic” AI value for rule 4 #3 R/W: ABCD (write to Flash RAM)
0252	40595	R/W	(Read/Write) set “if logic” AI value for rule 5 #0 R/W: ABCD (write to Flash RAM)
0253	40596	R/W	(Read/Write) set “if logic” AI value for rule 5 #1 R/W: ABCD (write to Flash RAM)
0254	40597	R/W	(Read/Write) set “if logic” AI value for rule 5 #2 R/W: ABCD (write to Flash RAM)
0255	40598	R/W	(Read/Write) set “if logic” AI value for rule 5 #3 R/W: ABCD (write to Flash RAM)
0256	40599	R/W	(Read/Write) set “if logic” AI value for rule 6 #0 R/W: ABCD (write to Flash RAM)
0257	40600	R/W	(Read/Write) set “if logic” AI value for rule 6 #1 R/W: ABCD (write to Flash RAM)
0258	40601	R/W	(Read/Write) set “if logic” AI value for rule 6 #2 R/W: ABCD (write to Flash RAM)
0259	40602	R/W	(Read/Write) set “if logic” AI value for rule 6 #3 R/W: ABCD (write to Flash RAM)
025A	40603	R/W	(Read/Write) set “if logic” AI value for rule 7 #0

025B	40604	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 7 #1
025C	40605	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 7 #2
025D	40606	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 7 #3
025E	40607	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 8 #0
025F	40608	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 8 #1
0260	40609	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 8 #2
0261	40610	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 9 #3
0262	40611	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 9 #0
0263	40612	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 9 #1
0264	40613	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 9 #2
0265	40614	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 9 #3
0266	40615	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 10 #0
0267	40616	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for

			rule 10 #1
0268	40617	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 10 #2
0269	40618	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 10 #3
026A	40619	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 11 #0
026B	40620	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 11 #1
026C	40621	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 11 #2
026D	40622	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 11 #3
026E	40623	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 12 #0
026F	40624	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 12 #1
0270	40625	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 12 #2
0271	40626	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule12 #3
0272	40627	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 13 #0
0273	40628	R/W	R/W: ABCD (write to Flash RAM) (Read/Write) set “if logic” AI value for rule 13 #1
			R/W: ABCD (write to Flash RAM)

0274	40629	R/W	(Read/Write) set “if logic” AI value for rule 13 #2 R/W: ABCD (write to Flash RAM)
0275	40630	R/W	(Read/Write) set “if logic” AI value for rule 13 #3 R/W: ABCD (write to Flash RAM)
0276	40631	R/W	(Read/Write) set “if logic” AI value for rule 14 #0 R/W: ABCD (write to Flash RAM)
0277	40632	R/W	(Read/Write) set “if logic” AI value for rule 14 #1 R/W: ABCD (write to Flash RAM)
0278	40633	R/W	(Read/Write) set “if logic” AI value for rule 14 #2 R/W: ABCD (write to Flash RAM)
0279	40634	R/W	(Read/Write) set “if logic” AI value for rule 14 #3 R/W: ABCD (write to Flash RAM)
027A	40635	R/W	(Read/Write) set “if logic” AI value for rule 15 #0 R/W: ABCD (write to Flash RAM)
027B	40636	R/W	(Read/Write) set “if logic” AI value for rule 15 #1 R/W: ABCD (write to Flash RAM)
027C	40637	R/W	(Read/Write) set “if logic” AI value for rule 15 #2 R/W: ABCD (write to Flash RAM)
027D	40638	R/W	(Read/Write) set “if logic” AI value for rule 15 #3 R/W: ABCD (write to Flash RAM)
027E	40639	R/W	(Read/Write) select condition channel for PTP R/W: ABCD (write to Flash RAM) A: condition #3 B: condition #2 C: condition #1 D: condition #0
027F	40640	R/W	(Read/Write) select condition channel

			for PTP R/W: ABCD (write to Flash RAM) A: condition #7 B: condition #6 C: condition #5 D: condition #4
0280	40641	R/W	(Read/Write) select action channel for PTP R/W: ABCD (write to Flash RAM) A: condition #3 B: condition #2 C: condition #1 D: condition #0
0281	40642	R/W	(Read/Write) select action channel for PTP R/W: ABCD (write to Flash RAM) A: condition #7 B: condition #6 C: condition #5 D: condition #4
0282	40643	R/W	(Read/Write) select PTP rule enable/disable R/W: 0000 0000 xxxx xxxx(write to Flash RAM) X:0 disable X:1 enable
0283	40644	R/W	(Read/Write) Peer to Peer Enable/Disable R: 000A(hex) W: 000A(hex) A: 0:disable 1:enable(write to Flash RAM)
0284	40645	R/W	(Read/Write) Mirror to(client) or Mirror from(server) R: 000A(hex) W: 000A(hex) A: 0: Mirror to(client)

			1: Mirror from(server) 2: “Mirror to” and “ Mirror from” simultaneously (write to Flash RAM)
0285	40646	R/W	(Read/Write) PTP IP address ip[0] low word for server R: AABB W: AABB
0286	40647	R/W	(Read/Write) PTP IP address ip[0] hi word for server R: AABB W: AABB
0287	40648	R/W	(Read/Write) PTP IP address ip[1] low word for server R: AABB W: AABB
0288	40649	R/W	(Read/Write) PTP IP address ip[1] hi word for server R: AABB W: AABB
0289	40650	R/W	(Read/Write) PTP IP address ip[2] low word for server R: AABB W: AABB
028A	40651	R/W	(Read/Write) PTP IP address ip[2] hi word for server R: AABB W: AABB
028B	40652	R/W	(Read/Write) PTP IP address ip[3] low word for server R: AABB W: AABB
028C	40653	R/W	(Read/Write) PTP IP address ip[3] hi word for server R: AABB W: AABB
028D	40654	R/W	(Read/Write) PTP IP address ip[4] low word for server R: AABB

			W: AABB
028E	40655	R/W	(Read/Write) PTP IP address ip[4] hi word for server R: AABB W: AABB
028F	40656	R/W	(Read/Write) PTP IP address ip[4] low word for server R: AABB W: AABB
0290	40657	R/W	(Read/Write) PTP IP address ip[4] hi word for server R: AABB W: AABB
0291	40658	R/W	(Read/Write) PTP IP address ip[6] low word for server R: AABB W: AABB
0292	40659	R/W	(Read/Write) PTP IP address ip[6] hi word for server R: AABB W: AABB
0293	40660	R/W	(Read/Write) PTP IP address ip[7] low word for server R: AABB W: AABB
0294	40661	R/W	(Read/Write) PTP IP address ip[7] hi word for server R: AABB W: AABB
0295	40662	R/W	(Read/Write) PTP IP address ip[8] low word for client R: AABB W: AABB
0296	40663	R/W	(Read/Write) PTP IP address ip[8] hi word for client R: AABB W: AABB
0297	40664	R/W	(Read/Write) internal flags

			R/W: xxxxxxxxxxxxxxxx X:0 or 1
0298	40665	R	(Read) error code of peer to peer for client Error Code=0; OK Error Code=1; Requested Content Not Satisfiable Error Code=2; Remote Module Not Found Error Code=FFFF; Not Enabled
0299	40666	R	(Read) error code of peer to peer #1 for server Error Code=0; OK Error Code=1; Requested Content Not Satisfiable Error Code=2; Remote Module Not Found Error Code=FFFF; Not Enabled
029A	40667	R	(Read) error code of peer to peer #2 for server Error Code=0; OK Error Code=1; Requested Content Not Satisfiable Error Code=2; Remote Module Not Found Error Code=FFFF; Not Enabled
029B	40668	R	(Read) error code of peer to peer #3 for server Error Code=0; OK Error Code=1; Requested Content Not Satisfiable Error Code=2; Remote Module Not Found Error Code=FFFF; Not Enabled

029C	40669	R	(Read) error code of peer to peer #4 for server Error Code=0; OK Error Code=1; Requested Content Not Satisfiable Error Code=2; Remote Module Not Found Error Code=FFFF; Not Enabled
029D	40670	R	(Read) error code of peer to peer #5 for server Error Code=0; OK Error Code=1; Requested Content Not Satisfiable Error Code=2; Remote Module Not Found Error Code=FFFF; Not Enabled
029E	40671	R	(Read) error code of peer to peer #6 for server Error Code=0; OK Error Code=1; Requested Content Not Satisfiable Error Code=2; Remote Module Not Found Error Code=FFFF; Not Enabled
029F	40672	R	(Read) error code of peer to peer #7 for server Error Code=0; OK Error Code=1; Requested Content Not Satisfiable Error Code=2; Remote Module Not Found Error Code=FFFF; Not Enabled
02A0	40673	R	(Read) error code of peer to peer #8 for server

Error Code=0; OK
Error Code=1; Requested Content Not Satisfiable
Error Code=2; Remote Module Not Found
Error Code=FFFF; Not Enabled

JetI/O 6512 Channel Number Register Map (Input Registers, Function Code = 04)

Protocol Address (Hex)	PLC Address (Decimal)	Access	Description
0000	30001	R	Analog input signal (Channel 0) Units: signed Input code: 20H ~23H (DinRTD) Return Value: (HEX) F830 ~ 2134(-200.0℃ ~ 850.0℃) Input code: 24H ~27H (JisRTD) Return Value: (HEX) F830 ~ 206C(-200.0℃ ~ 830.0℃) I put code : 28H(Ni 120Ω) Return Value: (HEX) FFB0 ~ 0A28(-80.0℃ ~ 260.0℃) Temperature signal = xxx.x ℃
0001	30002	R	Analog input signal (Channel 1) the same data format as Channel 0
0002	30003	R	Analog input signal (Channel 2) the same data format as Channel 0
0003	30004	R	Analog input signal (Channel 3) the same data format as Channel 0

Calibrated-voltage input for zero calibration and span calibration

Input Code (Hex)	Zero Calibration Resistor	Span Calibration Resistor
20	0Ω	390Ω
21	0Ω	390Ω
22	0Ω	390Ω
23	0Ω	390Ω
24	0Ω	390Ω
25	0Ω	390Ω
26	0Ω	390Ω
27	0Ω	390Ω
28	0Ω	390Ω

6 Appendix

6.1 SNMP MIB

An SNMP to I/O MIB file that can help you monitor I/O status with SNMP software. You can find the MIB file on the package.

(I). Public- System MIB:

Object ID (OID)	Description	Community, R/W Access
sysDescr	The *sysDescr* directive is used to define the system description of the host on which the SNMP agent (server) is running. This description is used for the *sysDescr* object instance of the MIB-II. SYNTAX: *DisplayString* (SIZE (0..31))	Public, Read Only
sysObjectID	The vendor's authoritative identification of the network management subsystem contained in the entity. This value is allocated within the SMI enterprises sub tree. SYNTAX: *DisplayString* (SIZE (0..31))	Public, Read Only
sysUpTime	The *sysUpTime* directive is used to measures the time, in hundredths of a second, since the last system restart. SYNTAX: *DisplayString* (SIZE (0..31))	Public, Read Only
sysContact	The *sysContact* directive is used to define the system contact address used for the *sysContact* object instance of the MIB-II. SYNTAX: *DisplayString* (SIZE (0..31))	Public, Read Only
sysName	The *sysName* directive is a string containing an administratively-assigned name for the system running the SNMP agent. By convention, this should be its fully-qualified domain name.	Public, Read Only

	SYNTAX: *DisplayString* (SIZE (0..31))	
sysLocation	The *sysLocation* directive is used to define the location of the host on which the SNMP agent (server) is running. This directive is used for the *sysLocation* object instance of the MIB-II. SYNTAX: *DisplayString* (SIZE (0..31))	Public, Read Only

(II).Private MIB – Intelligent I/O Server – 6512

Object ID (OID)	Description	Community, R/W Access
ch0- Temperature	Analog input signal (Channel 0) SYNTAX: *INTEGER* (0..65535)	Private, Read Only
ch1- Temperature	Analog input signal (Channel 1) SYNTAX: *INTEGER* (0..65535)	Private, Read Only
ch2- Temperature	Analog input signal (Channel 2) SYNTAX: *INTEGER* (0..65535)	Private, Read Only
ch3- Temperature	Analog input signal (Channel 3) SYNTAX: *INTEGER* (0..65535)	Private, Read Only

6.2 Revision History

Version	Description	Date
1.5	● Update “General” page ● Update “Data” page ● Update “Alarm” page ● Add “Logic Rules” page ● Add “Peer to Peer I/O” page ● Change “Alarm” page to “SNMP” page ● Add new Modbus/TCP registers in FW208	Sept. 1, 2009
1.4	● Add configuration backup/restore 1. Simply firmware upgrade procedure	Dec. 24, 2008
V1.21	● Add description for Input Range of Current mode ● Change Pin No. Table… ● Change Device Finder Utility to Device Finder Popup Window due to Device Finder Utility is merged. ● Remove Web Configuration. ● Add steps and screen for Password Login and Change ● Add examples for Terminal mode ● Update Tag Properties popup window ● Add more description for firmware upgrading progress ● Change Watch dog timer to host watch dog timer in Register 40001/40002 ● Add description for Register 40034 to 40037 and 41000 to 41003.	Jun. 25, 2008
1.2	Add Emulation mode operation description. Add Terminal mode operation description. Add description of the AI channels' floating state in "Data" page. Add Notes for the Simulation mode - OPC Server Utility Add Notes for Device IP change - Device Finder Utility Add Notes in How to upgrade firmware. Add description for Modbus/TCP address mapping. Add warning for calibration registers. Correct Wordings: Logic rule ->I/O rule, I/O module -> I/O server, pin table in appearance and	Apr. 17, 2008

	description, Description of the SNMP and Trap.	
1.11	Update Modbus/TCP Reset Status register.	Feb. 20, 2008
1.1	Add Note for IP changed, Trap types, update latest datasheet info, correct some wordings, add calibration table.	Feb. 12, 2008
1.0	First Release Change V0.2 to V1.0	Jan. 30, 2008