Korenix JetI/O 6550
Industrial Intelligent Ethernet I/O Server

User Manual

Sept. 2009 (V1.6)

www.korenix.com

Korenix JetI/O 6550
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 JeI/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: Industrial Intelligent 8-CH Analog Input Ethernet I/O Server

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

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

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

JetI/O 6550: Industrial 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 Module
- 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 6550 Introduction

JetI/O 6550 is a Block I/O module equipped with 14 channels Digital Input, 8 channels Digital Output connectors. JetI/O 6550 supports digital input and event counter for input mode, digital output and pulse output for output mode. The flexible Condition-&-Go (IF-Then) rules can help users to define intelligent logic rule for remote I/O control.

JetI/O 6550 provides Windows Utilities, SNMP for configuration. Industrial Modbus/TCP protocol and OPC Server driver 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 6550 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)

Network Protocols: IP, TCP, UDP, SNMP, HTTP, Telnet, BOOTP, DHCP

Digital Input

Input Channels: 14 Channels

Input Type: source type

Input Mode: D/I or event counting with input frequency of 100 Hz max

DC Input: 30V max

Threshold Voltage: 4V

Responding Time to Host PC Request: <2ms

Isolation Voltage: 2500Vrms

Digital Output

Output Channels: 8 Channels

Output Type: SSR output, sink type

Output Mode: Level or pulse output with programmable pulse width

Working Range: 5-40VDC

Driving Capacity: 250mA max

Responding Time to Host PC Request: <2ms

Output Initial State: Programmable

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 3 SNMP Trap Server

Logic Condition&Go Rules: Conditions of the DI/Counter values, Actions include DO/Pulse, Counter Reset and Trap

Peer-to-Peer I/O: Mapping DI channel events from one JetIO 6500 to the DO channels of another JetI/O 6500

Power Requirements

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

Power Consumption: Max. 1.92W

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

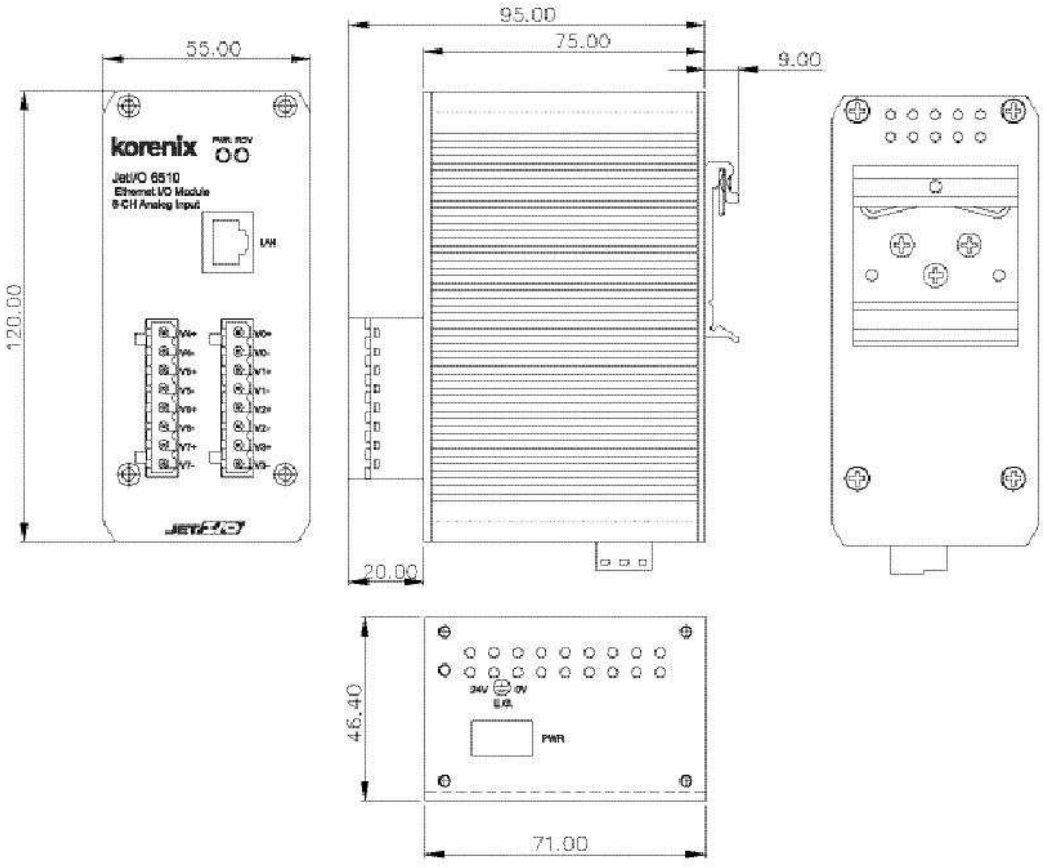
2 Hardware Installation

This chapter includes hardware introduction, installation and configuration information.

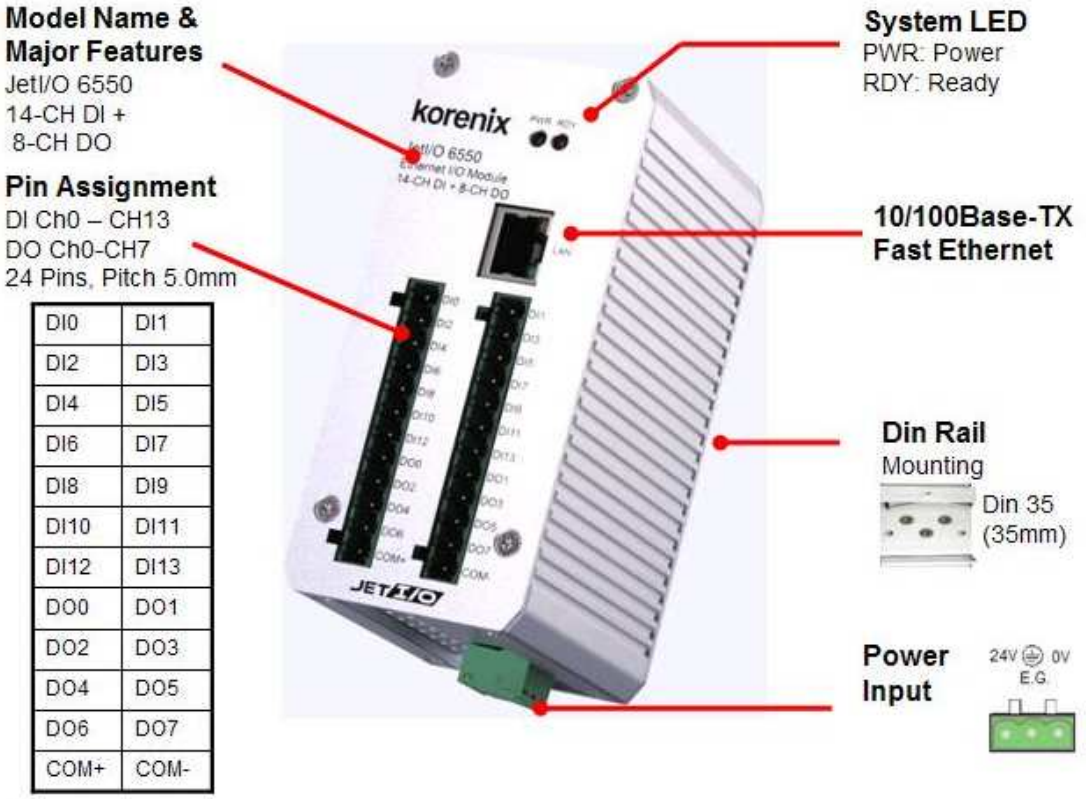
Following topics are covered in this chapter:

2.1 Hardware Introduction

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



JetI/O 6550 Appearance:



DI0	DI1
DI2	DI3
DI4	DI5
DI6	DI7
DI8	DI9
DI10	DI11
DI12	DI13
DO0	DO1
DO2	DO3
DO4	DO5
DO6	DO7
COM+	COM-

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

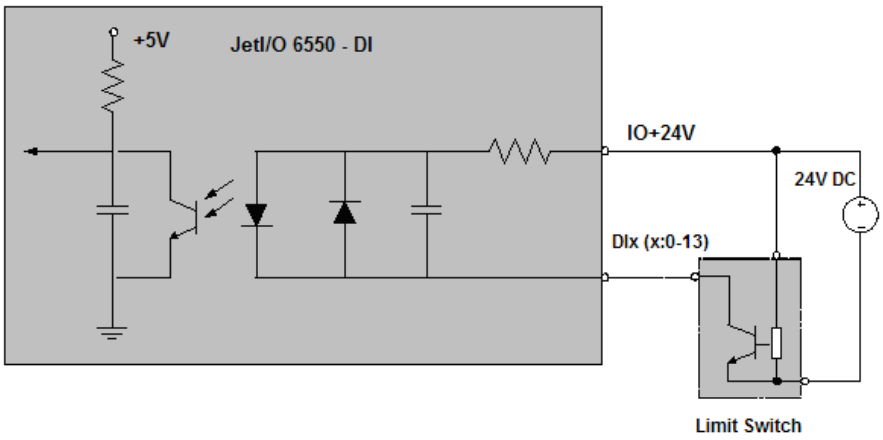
Follow the pin assignment to insert the wires into the front contacts on the terminal block connector. Tighten the wire-clamp screws to prevent the I/O wires from being loosened.

The wiring diagram of the JetI/O 6550 is as below:

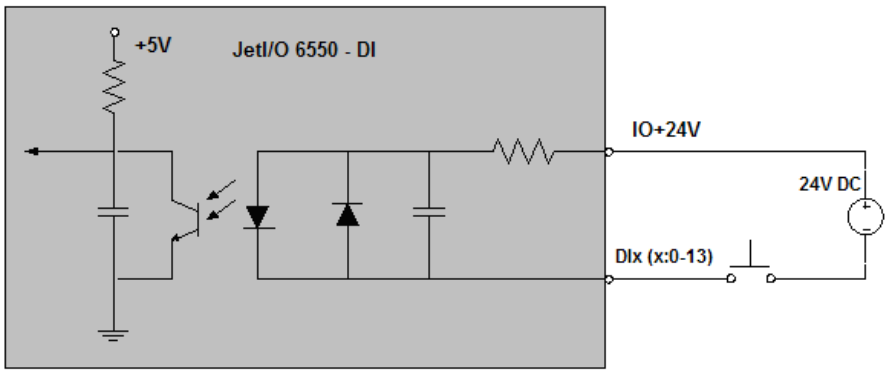
Pin No	Description	Pin No	Description
DI0	Digital input Ch0	DI1	Digital input Ch1
DI2	Digital input Ch2	DI3	Digital input Ch3
DI4	Digital input Ch4	DI5	Digital input Ch5
DI6	Digital input Ch6	DI7	Digital input Ch7
DI8	Digital input Ch8	DI9	Digital input Ch9
DI10	Digital input Ch10	DI11	Digital input Ch11
DI12	Digital input Ch12	DI13	Digital input Ch13
DO0	Digital Output 0	DO1	Digital Output 1
DO2	Digital Output 2	DO3	Digital Output 3
DO4	Digital Output 4	DO5	Digital Output 5
DO6	Digital Output 6	DO7	Digital Output 7
COM+	IO Power--COM+	COM-	IO Power--COM-

2.4 JetI/O 6550 Wiring Example

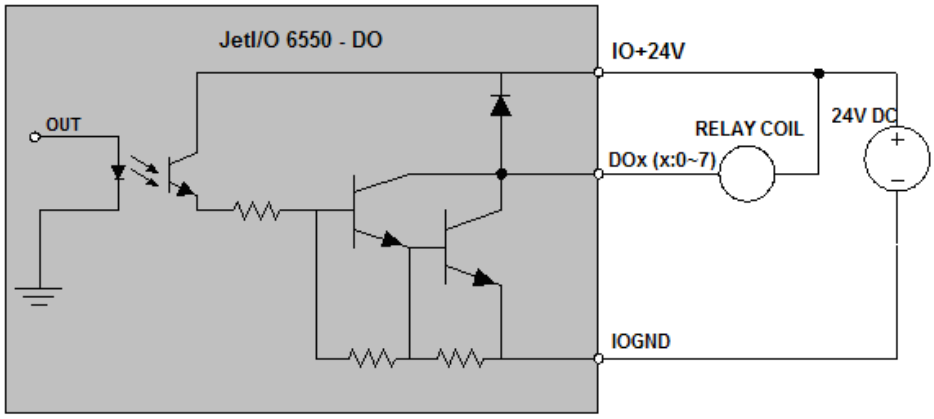
2.4.1 JetI/O 6550 Digital Input Wiring Example (Limit Switch)



2.4.2 JetI/O 6550 Digital Input Wiring Example (Push Button)



2.4.3 JetI/O 6550 Digital Output Wiring Example (Push Button)



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 cables length between the 2 ends is less than 100 meters (328 feet).

The wiring cable types are as below.

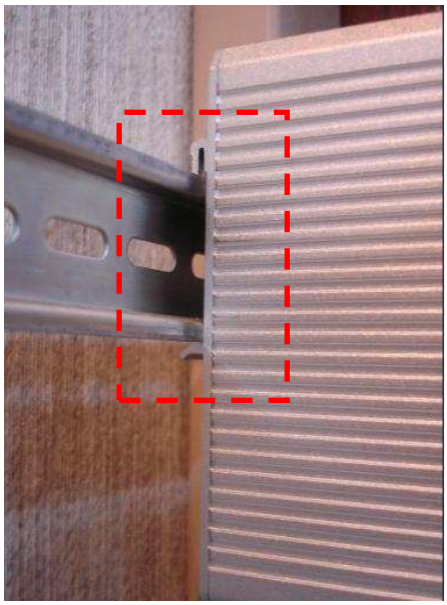
10Base-T: 2-pair UTP/STP Cat. 3, 4, 5 cable, EIA/TIA-568 100-ohm (100m)
100 Base-TX: 2-pair UTP/STP Cat. 5 cable, EIA/TIA-568 100-ohm (100m)
1000 Base-TX: 4-pair UTP/STP Cat. 5 cable, EIA/TIA-568 100-ohm (100m)

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.

a. 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 Architecture

The Figure 1 shows the JetI/O Intelligent Ethernet I/O Server Architectures. 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.

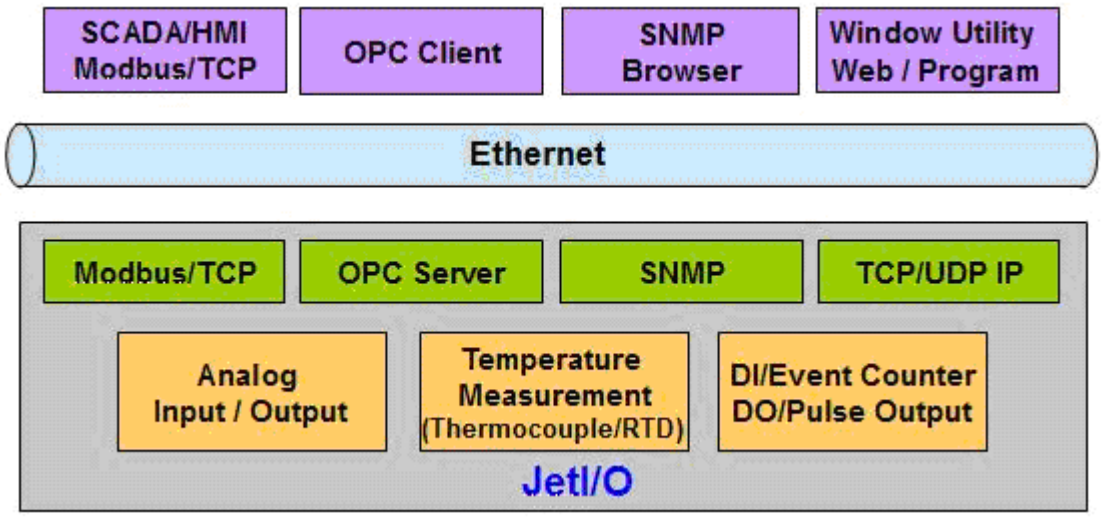
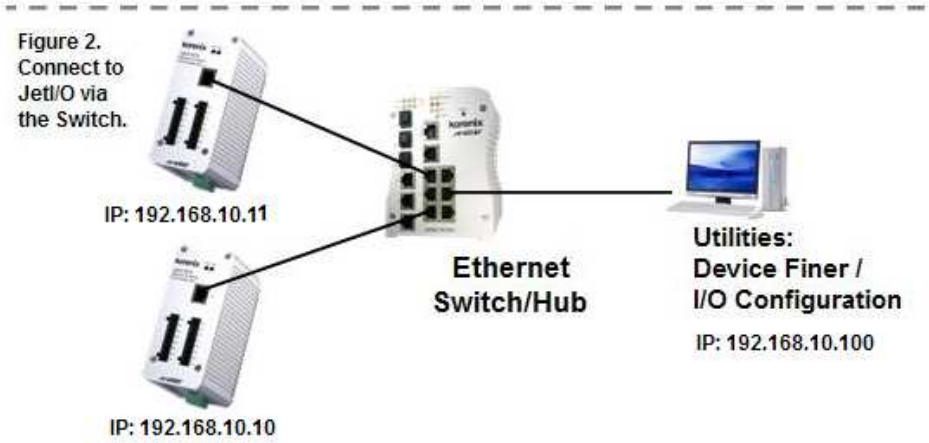


Figure 1. The JetI/O Intelligent Ethernet I/O Server Architecture.

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.

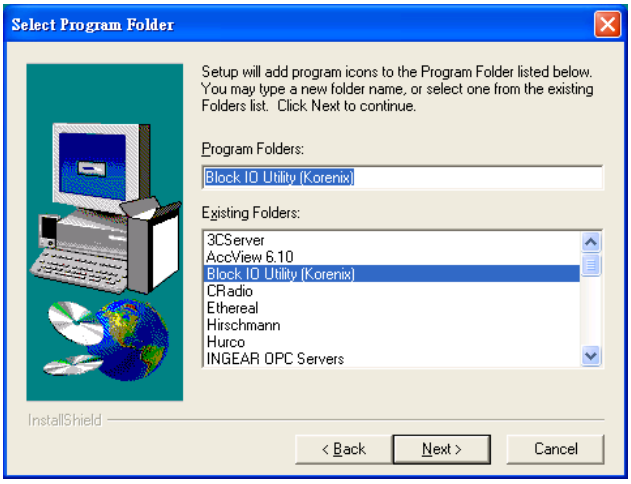
4.1.1 Installation

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”.

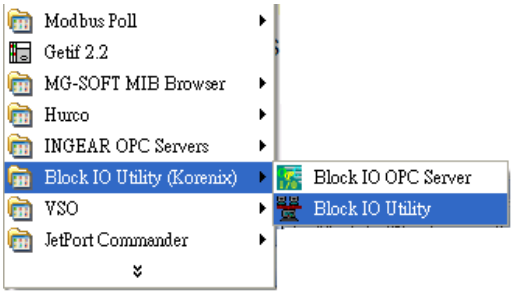
4. Type the name for the Block I/O Configuration Utility or use the default name, Block IO Utility (Korenix) for the program in the “Program Folder” field of the “Select Program Folder” window. Then click “Next”.



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.

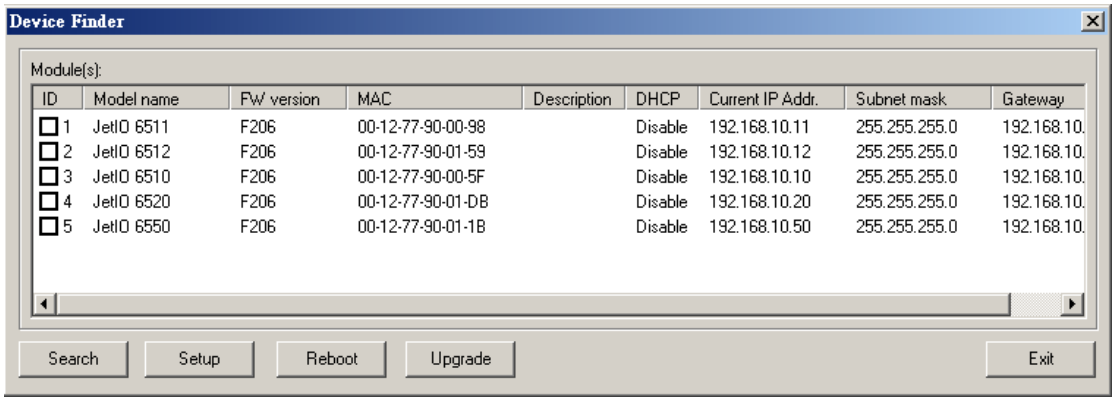


4.1.2 Device Finder

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.

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

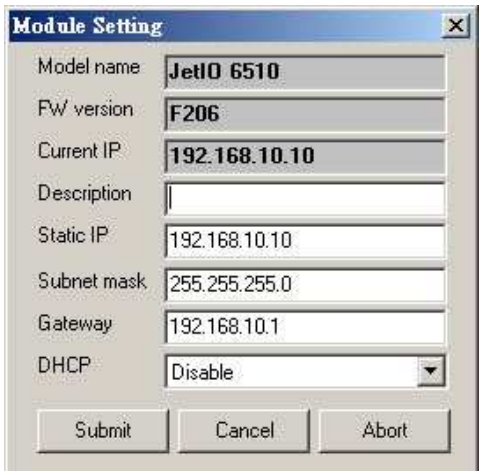


The following information is displayed:

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.

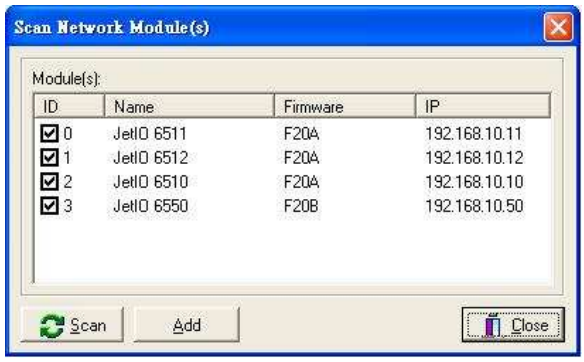
Note: *Clear the ARP cache (arp –d in DOS prompt) if you can’t change the second unit’s IP address. In DOS prompt, “arp –a” can help you to see the ARP table. “arp –d” can help you to clear all the ARPs in your host PC.*

4.1.3 Device Scan

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

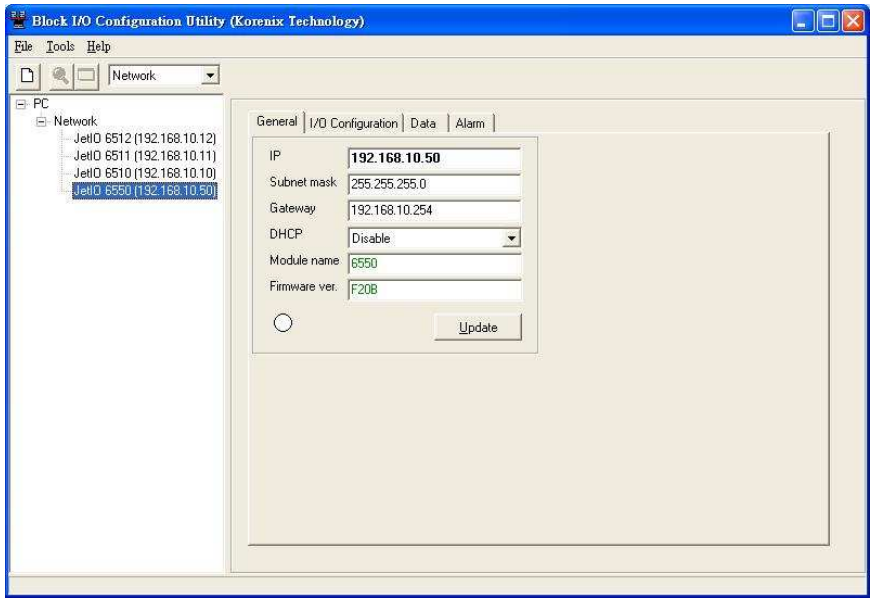
The right indicator will show “Green” after you opened the interface. Click “Close” can close the network interface.

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.

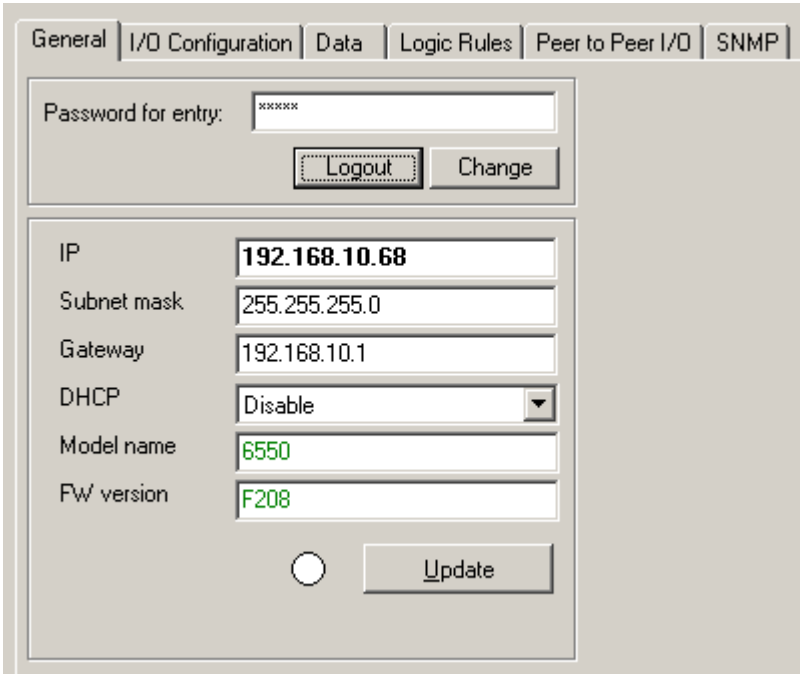
3. Click "Add" to add the available JetI/O units. Then you can see the JetI/O units are listed in the left column.



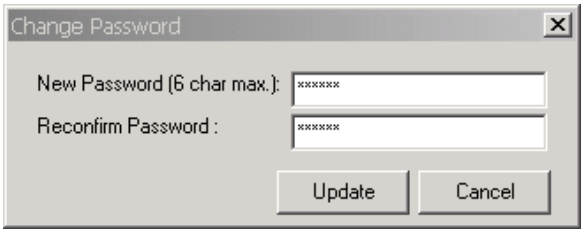
4. Move the mouse over to one of the JetI/O units. Select the unit then you can configure and monitor the configurations of the JetI/O. The features Block I/O Configuration utility provides are similar. Please find your model name and go to its configuration introduction chapter in below.

4.1.4 JetI/O 6550 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.

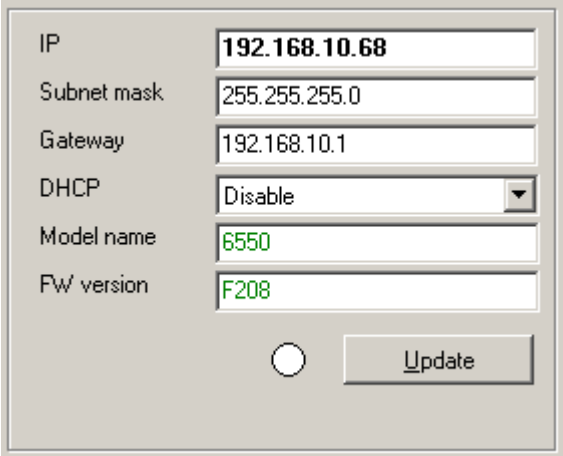


Note 1: The password protection is the new feature provided in firmware F206 and Utility V1.3 or later.

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 'Reset to default' command, referred to section 2.2.6, to clear the password and load factory defaults. This will result in the clear all the configuration settings as you assigned.*

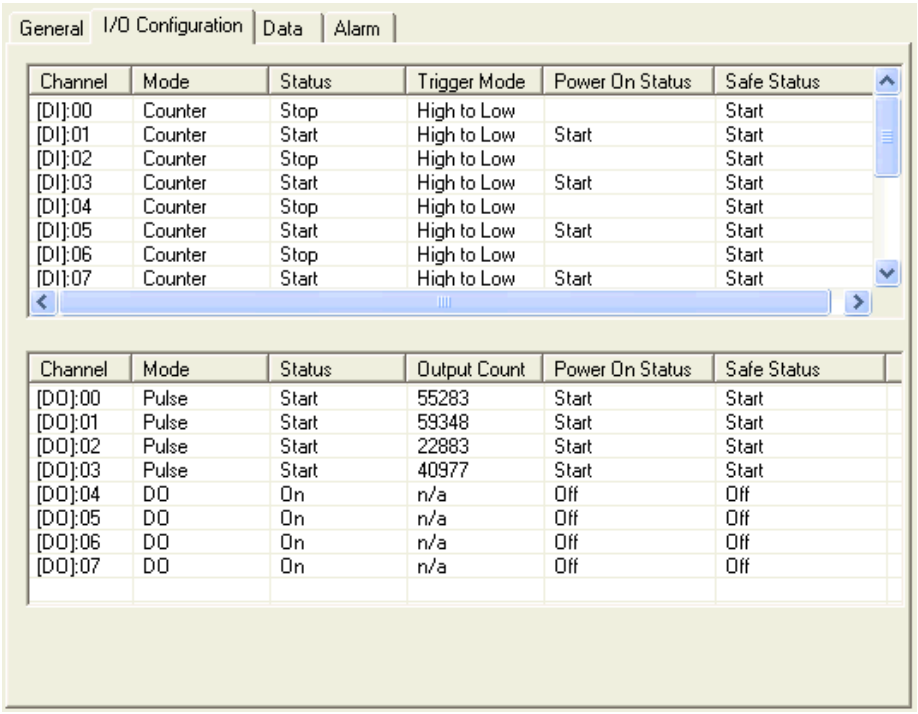
Each module has its own page to display and configure the TCP/IP parameters. If the function of DHCP is disabled, user can type the IP address, Gateway and Subnet Mask information on the page. When everything is ok, user can push the "Update" button. This setting will be affected after restarting the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

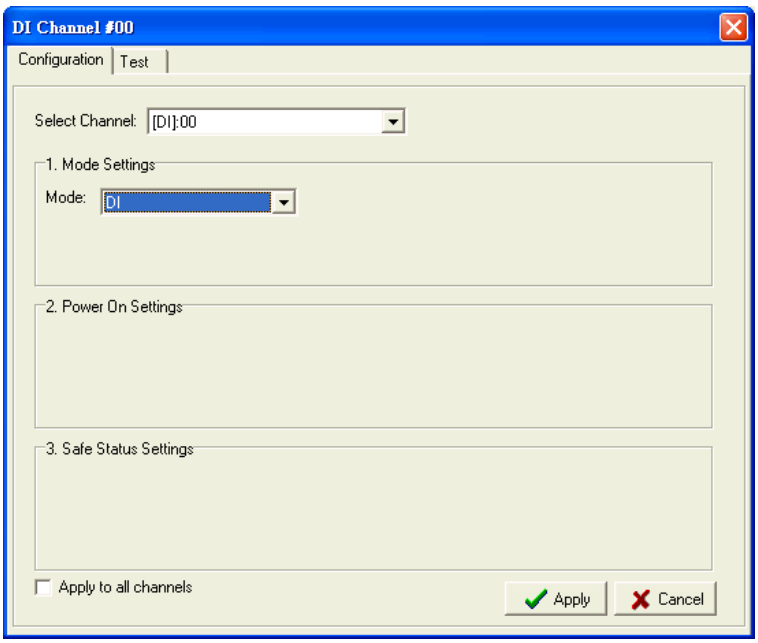
4.1.4.2 Go to "**I/O Configuration**" page.

In "I/O Configuration" page, you can configure the DI/DO mode and check status for each channel.



Select the entry of the DI channel you want configure and double click. The below popup window of the DI Channel's setting appears.

DI Channel Configuration – DI (Digital Input):



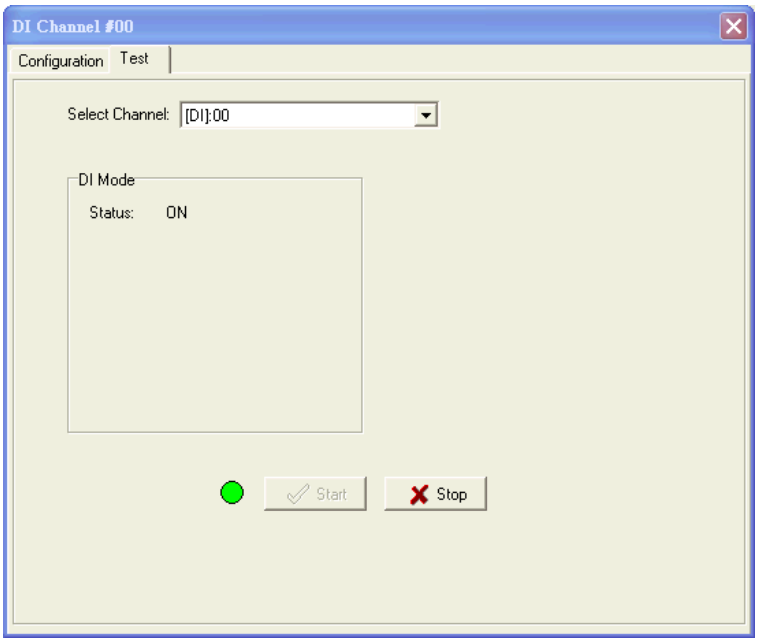
Select Channel- [DI]:00 ~ [DI]:13 or select “Apply to all channels”

Mode Settings- Mode: DI

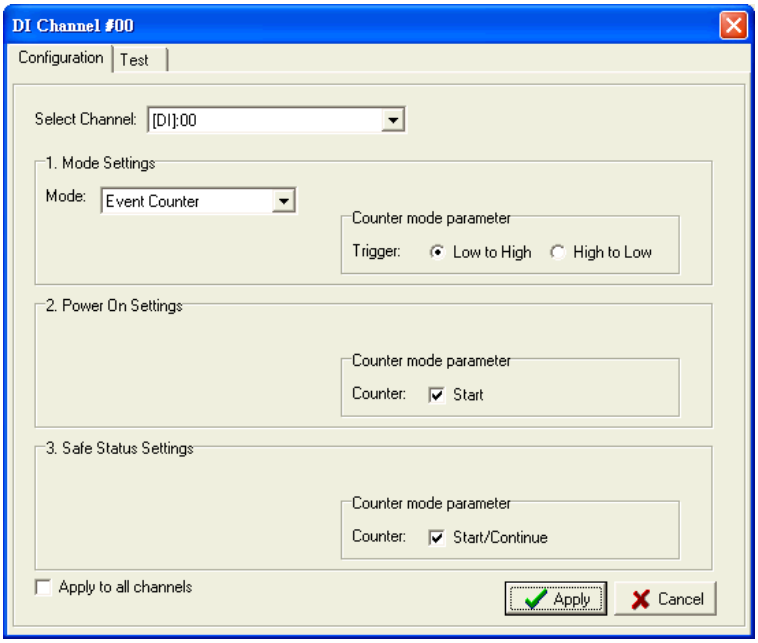
Press “Apply” to assign the mode for the channel.

DI Channel Test – DI (Digital Input):

Select the target channel and press “Start” to monitor the status of the DI mode.



DI Channel Configuration – Event Counter:



Select Channel- [DI]:00 ~ [DI]:13 or select “Apply to all channels”

- Mode Settings- Mode: Event Counter

  Counter mode parameter- “Low to High” or “High to Low”
- Power On Settings: Start or Disable

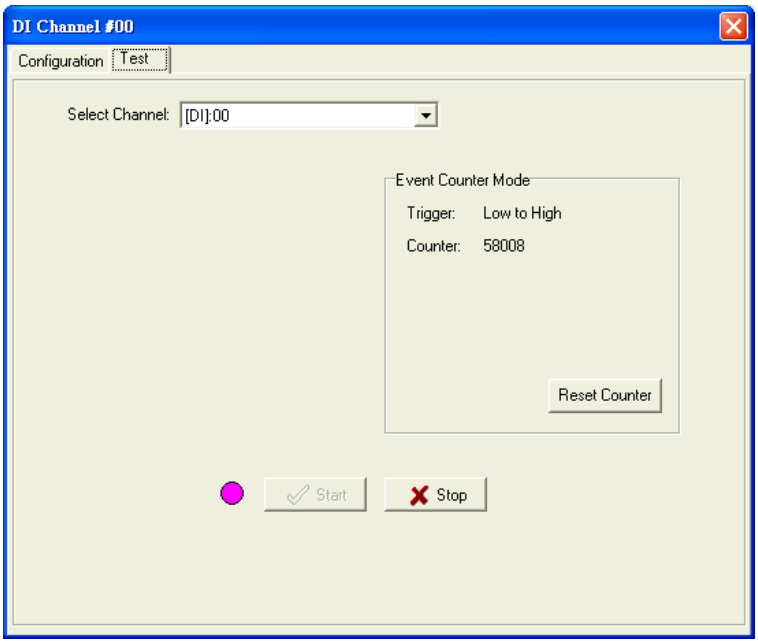
  Enable Power On selection to automatically count the events when startup the device
- Safe Status Settings: Start/Continue or Disable

  Enable Safe Status selection to keep counting the events when disconnected the LAN interface

Press “Apply” to assign the mode for the channel

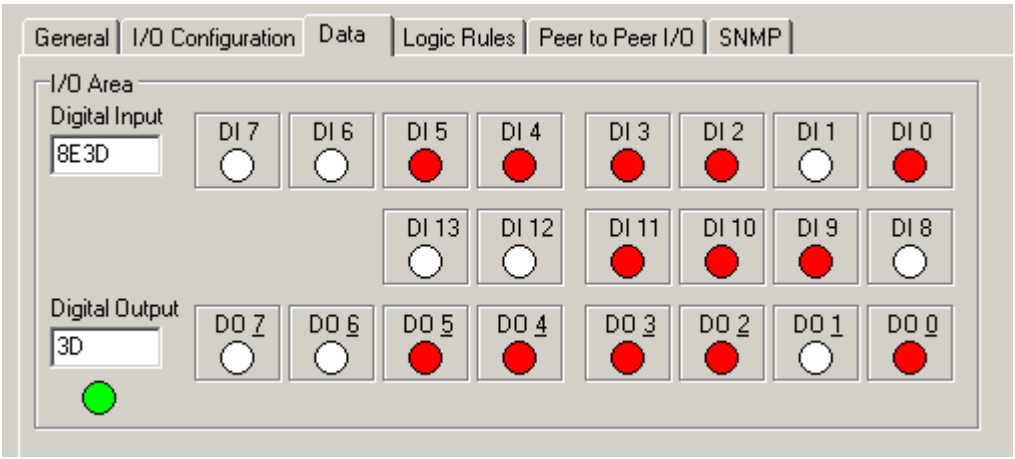
DI Channel Test –Event Counter:

Select the target channel and press “Start” to monitor the counter of the trigger mode. Press “Reset Counter” to reset the counter to zero.



4.1.4.3 Go to **"Data"** page.

In this page, you could monitor the current working status of each channel both on DI and DO.

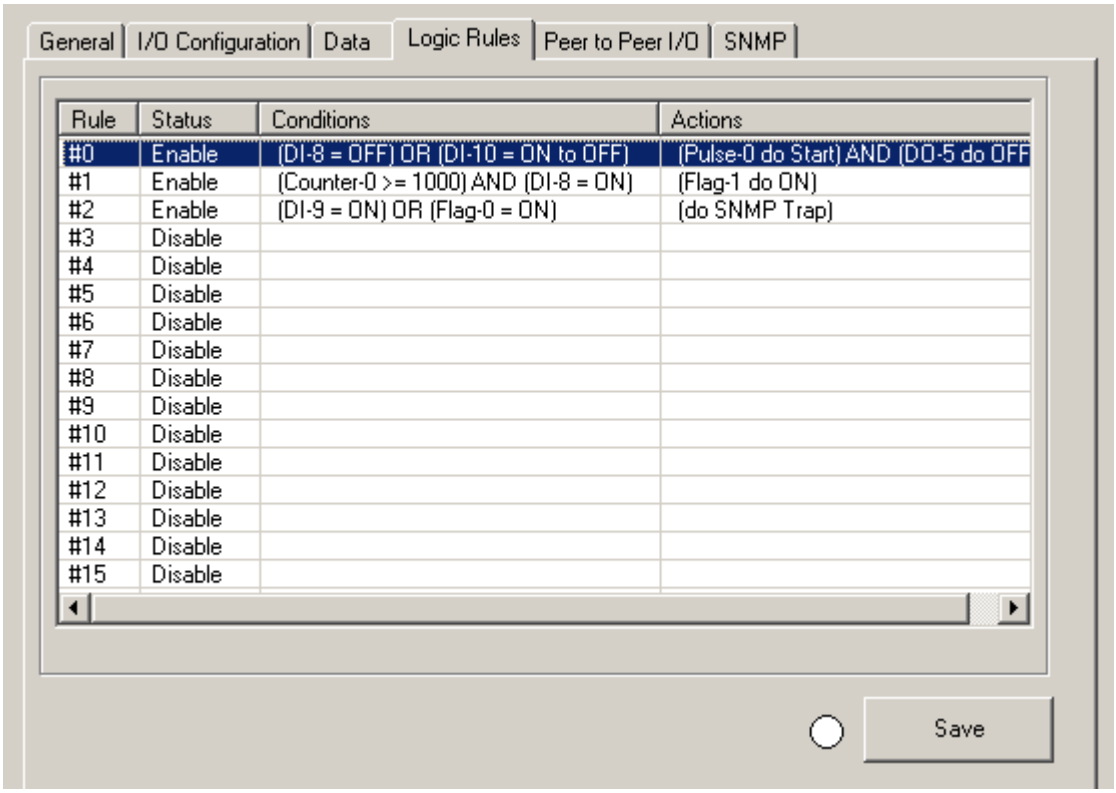


DI0-DI13/ DO0-DO8	In DI 0-13, you can read the DI status. In DO 0-7, you can read the status or control the ON/OFF state.
○	After turned off the DI/DO, the round LED is not light, thus it keeps in white color.
●	After activated the DI/DO, the round LED becomes to red color.
Digital Input 2DB6	In DI 0-13, you can read the DI status. The hex number displayed in the Digital Input field shows you which DI channels are under ON status. The value is the same as the

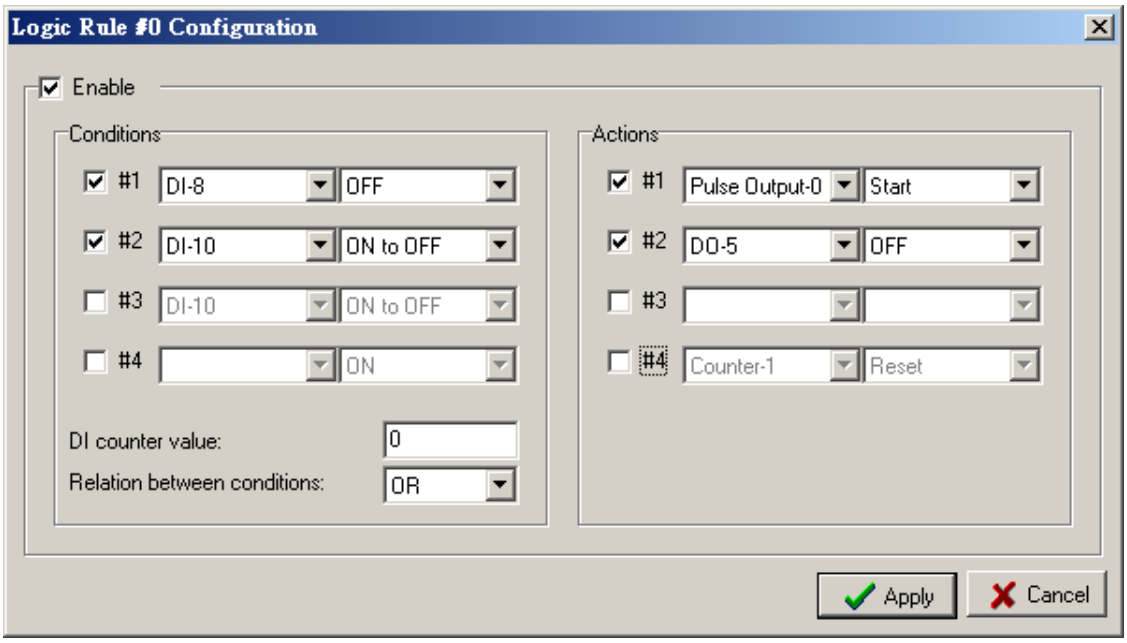
	value you can access via Modbus/TCP register. For example, 2DB6 (0b'0010 1101 1011 0110) represent those DI channels of CH13, CH11, CH10, CH8, CH7, CH5, CH4, CH2, CH1 turn ON in 'RED' LED indication as well as the rest channels are 'OFF' in 'WHITE' LED indication.
	In DO 0-7, you can read the DO status or activate the ON/OFF state. The hex number displayed in the Digital Output field shows you which DO channels are under ON status. The value is the same as the value you can access by Modbus/TCP register. For example, B6 (0b' 1011 0110) represent those DO channels of CH7, CH5, CH4, CH2, CH1 turn ON in 'RED' LED indication as well as the rest channels are 'OFF' in 'WHITE' LED indication.
 Blinking	The indicator in the bottom of the I/O Area means the utility is monitoring the status of the channels. If there is error occurred, the color become to red or not light.

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 Block 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	DI = ON	DO = ON

#2	Enable	DI = OFF	DO = OFF

Rule #1: If condition is equal to “DI0=ON”, the “DO 0” is automatically “ON”. Should you want to run the reverse behavior, configure the reverse way in the rule #2: If condition is equal to “DI0=OFF”, the “DO 0” is automatically “OFF”.

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**	
DI-Channel	ON, OFF, ON to OFF, OFF to ON	DO- Channel	ON, OFF
Internal Flag	ON, OFF, ON to OFF, OFF to ON	Internal Flag	ON, OFF
Event Counter -Channel	=, >, <, ≥, ≤	Event Counter-Channel	Reset
Counter Value	(Number)	Pulse Output	Start or Stop
Relation between Conditions	OR, AND	SNMP	(Trap Server IP)

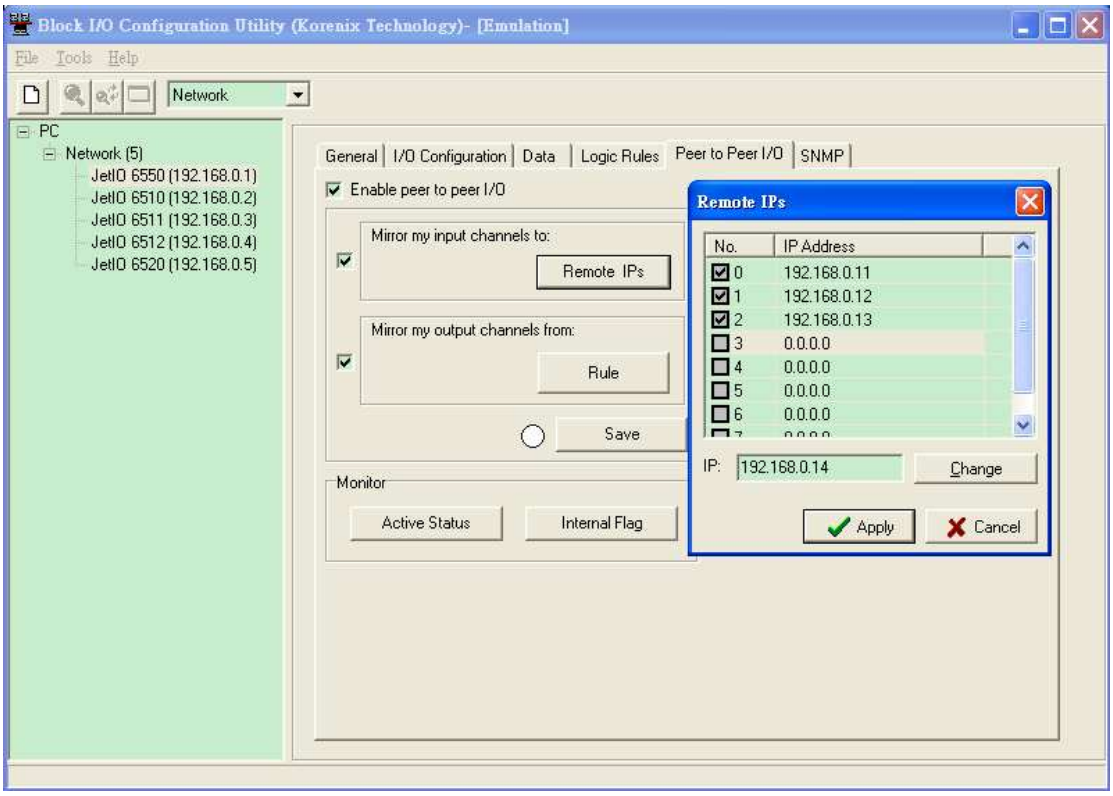
After completing all configurations for I/O logic rule, click “Save” button in the bottom of the Logic Rules area. All the mapping configurations will be flushed into flash memory on the module. In order to extend the flash memory life, it is strongly recommended that you should save all configurations 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 digital inputs D/Is and/or internal flags on the

source module can be mapping to channel including all the digital outputs as well as 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.

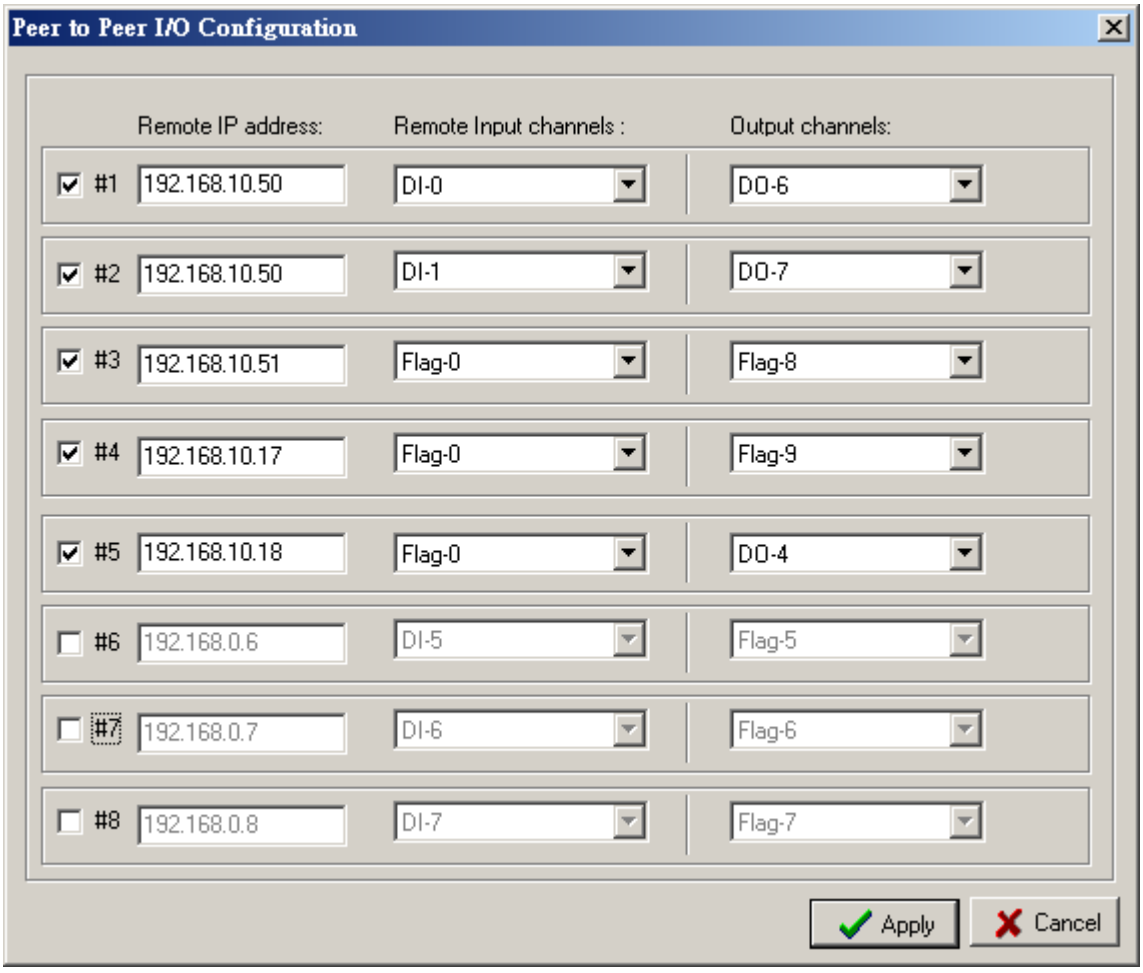
(2) Configure module's input channels mirrored to remote IP

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 digital inputs D/Is and/or 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.

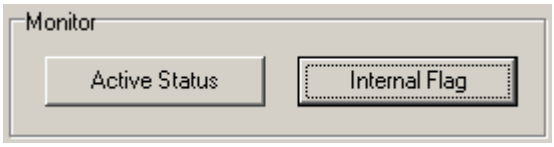
(3) Configure module’s output channels mirrored from remote IP

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 DO channels or internal flags which will mirror the remote input channels or 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.

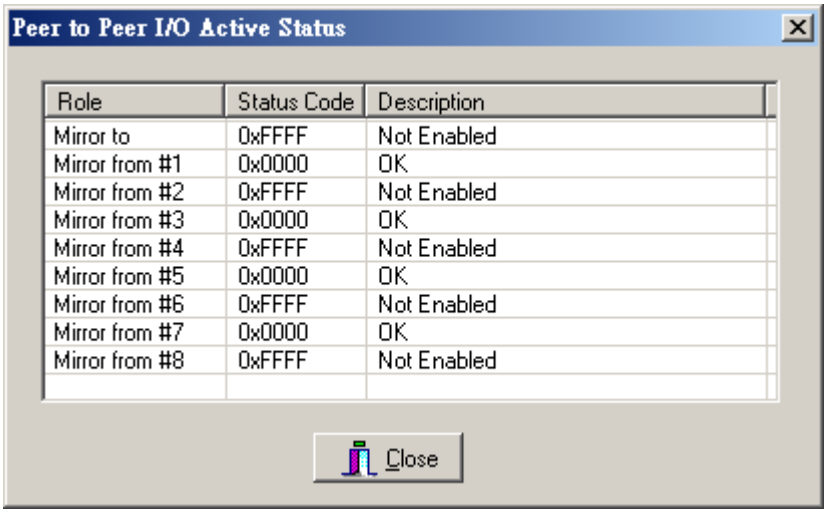


After completing all configurations for Peer-to-Peer I/O activity, click “Save” button, all the mapping configurations will be flushed into flash memory on

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.

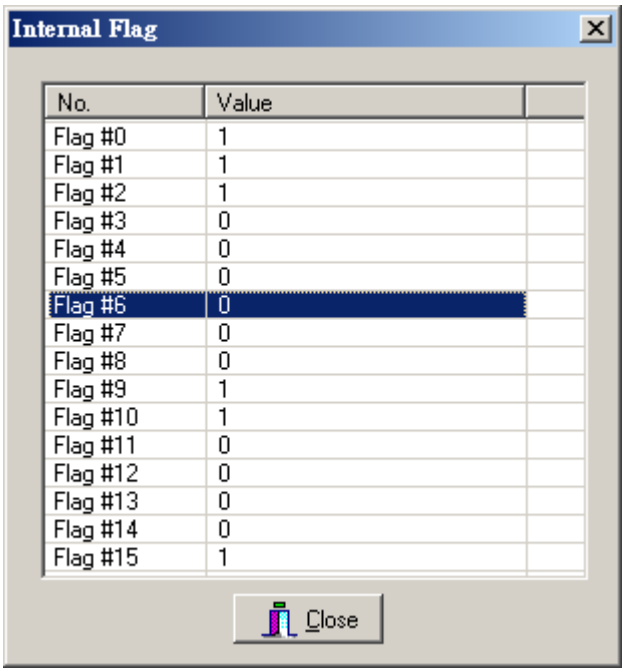


If there is error occurred, you could directly browse the latest activity from status code of peer-to-peer I/O activity. The status code is defined as below.

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

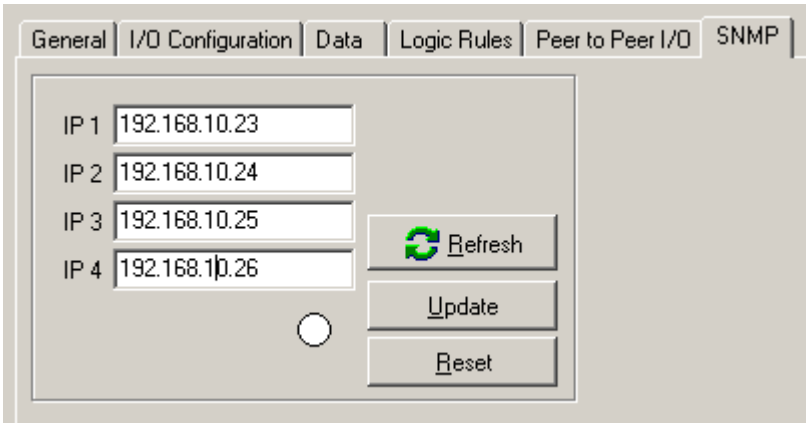
Click "Internal Flag" button, the internal flag dialog will prompt for you to

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.

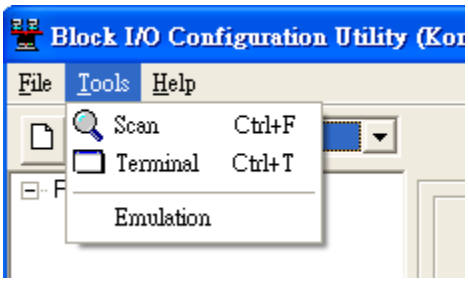
IP Settings: You can configure up to 4 SNMP Trap Server’s IP here. Type the IP address and press “Update” to activate the new setting. Click “Refresh” to reload the current SNMP trap server’s IP from registers. Once the SNMP trap has been activated by I/O login rule, you need to press “Reset” to acknowledge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



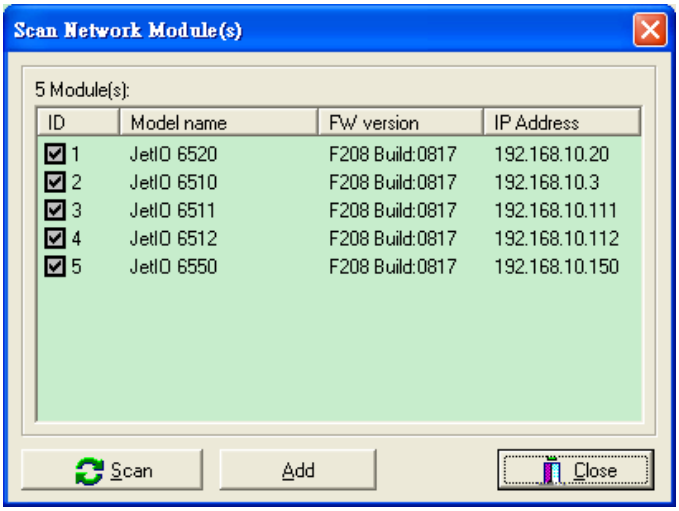
4.1.5 Emulation Mode

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.

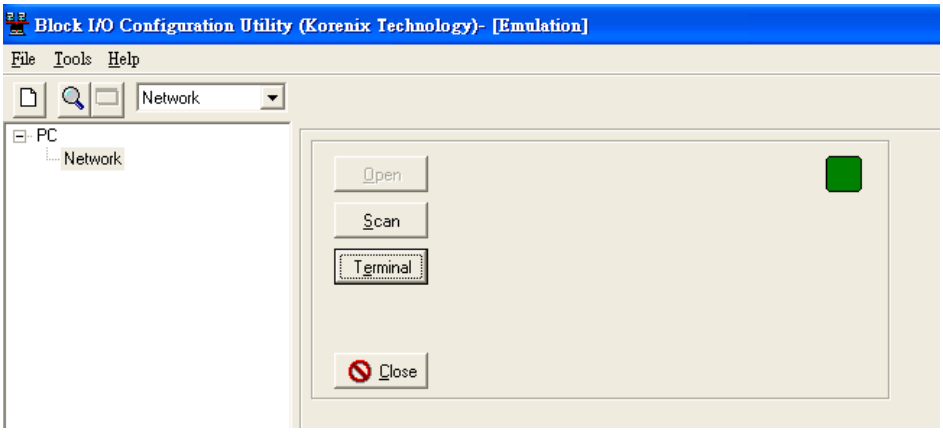


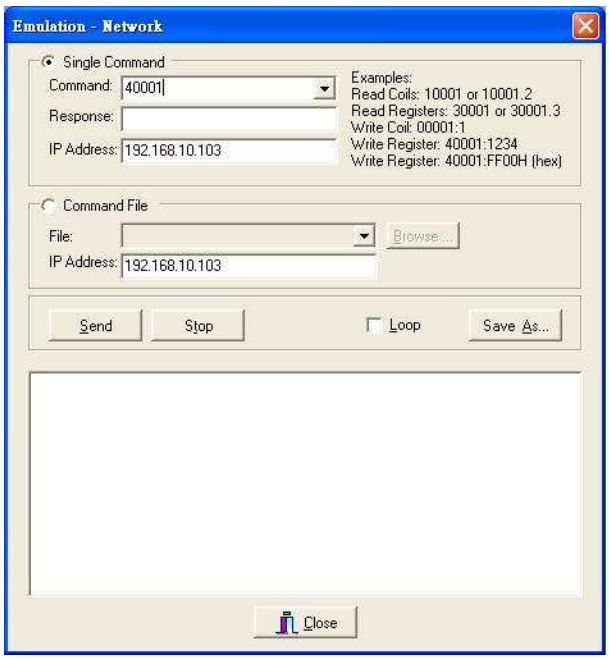
4.1.4.3 Follow the 4.1.4 to practice JetI/O configuration. Select the model and read or write status and configurations. As to how to operate the JetI/O configuration of other model, please refer to its manual.

4.1.6 Terminal Mode

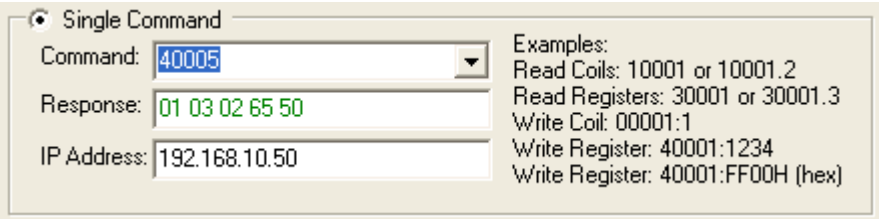
Block I/O Configuration Utility provides "Terminal" mode for user to read and write Modbus/TCP registers, thus users do not need additional tools but still can practice Modbus/TCP protocol well.

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.

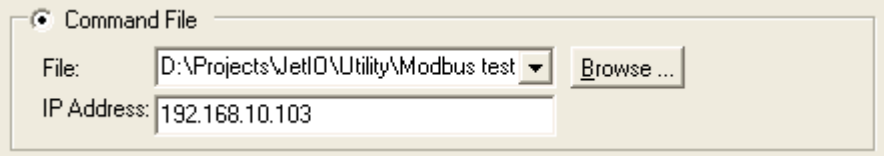
40001

40002

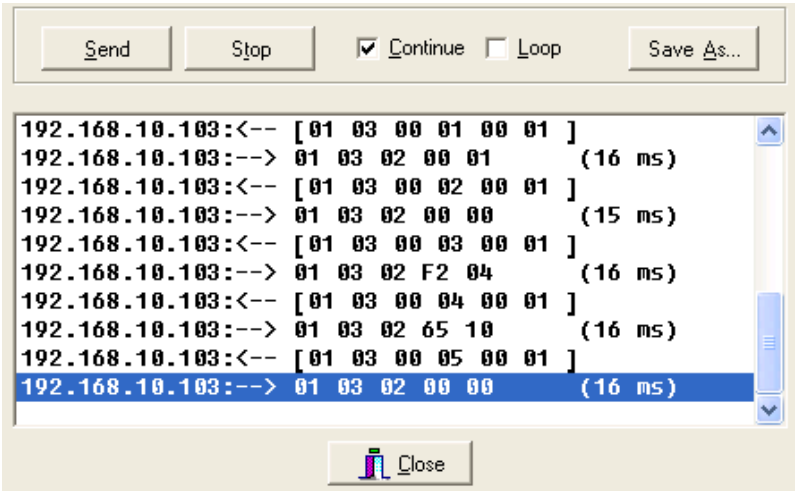
40003

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.

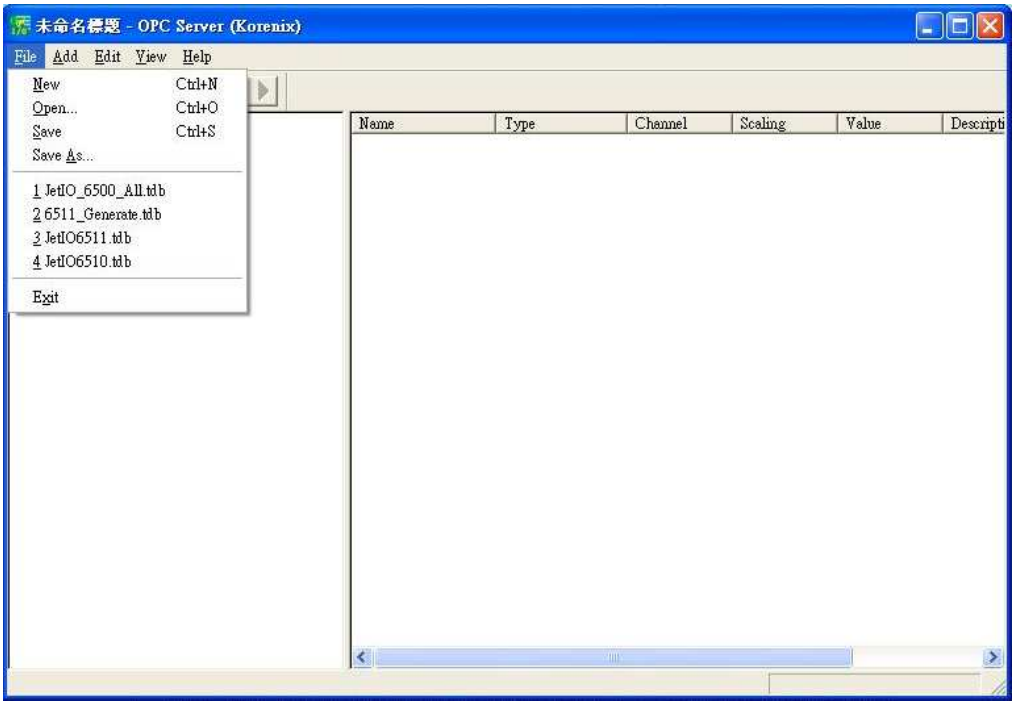
4.2 Block I/O OPC Server Utility

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.

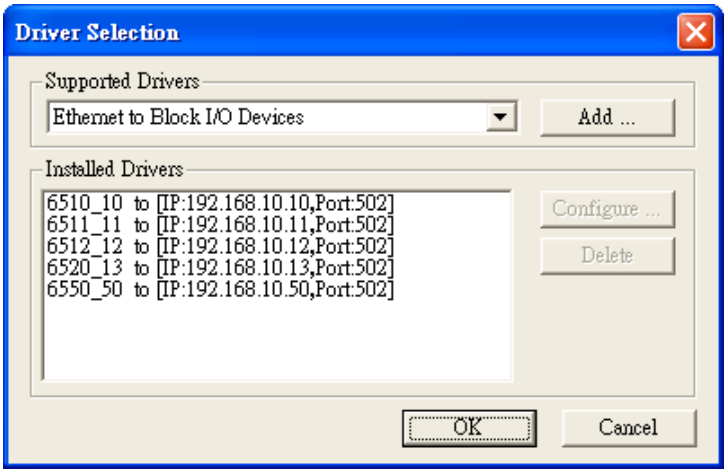
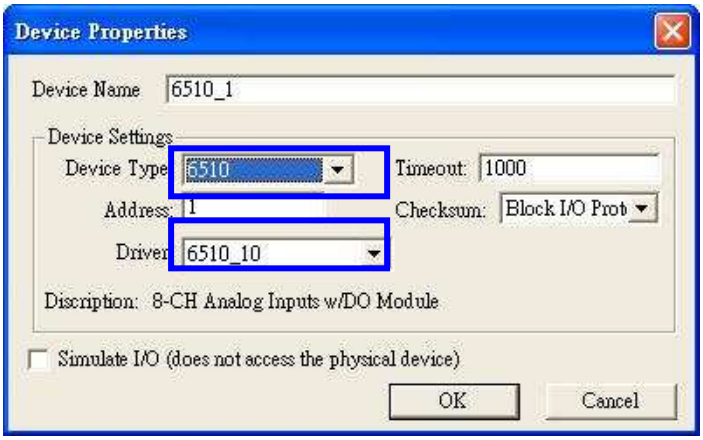
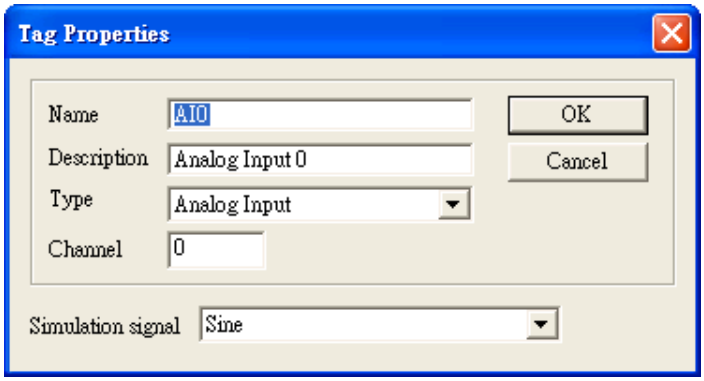


Figure 4.4 Example: Add all the drivers for available models. If you have 5 models over the same network, add them and give them different name for identification in next steps. Like: (Model Name/6510)_(IP address/10).

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.

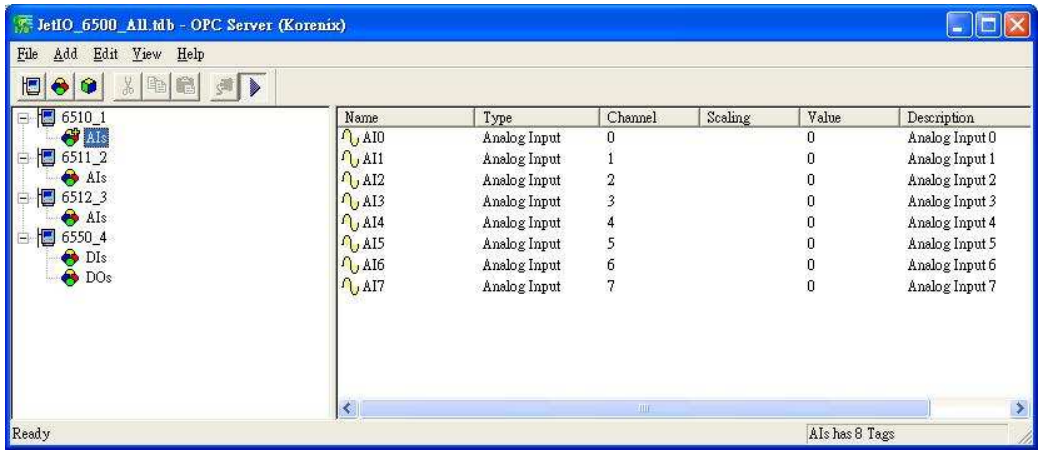


6. Select “Add -> New Group” to create new group for the later new tags you’ll create. Select “Add -> New Tag” and fill the “Tag Properties” in the popup window. Select the tag and “Edit –> Properties“, you can modify the tag properties.



*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.

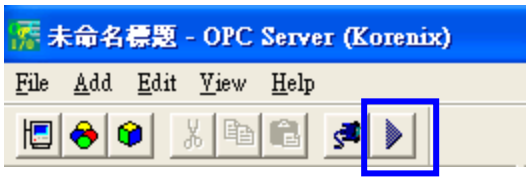
Type: The input type of the channel.

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.

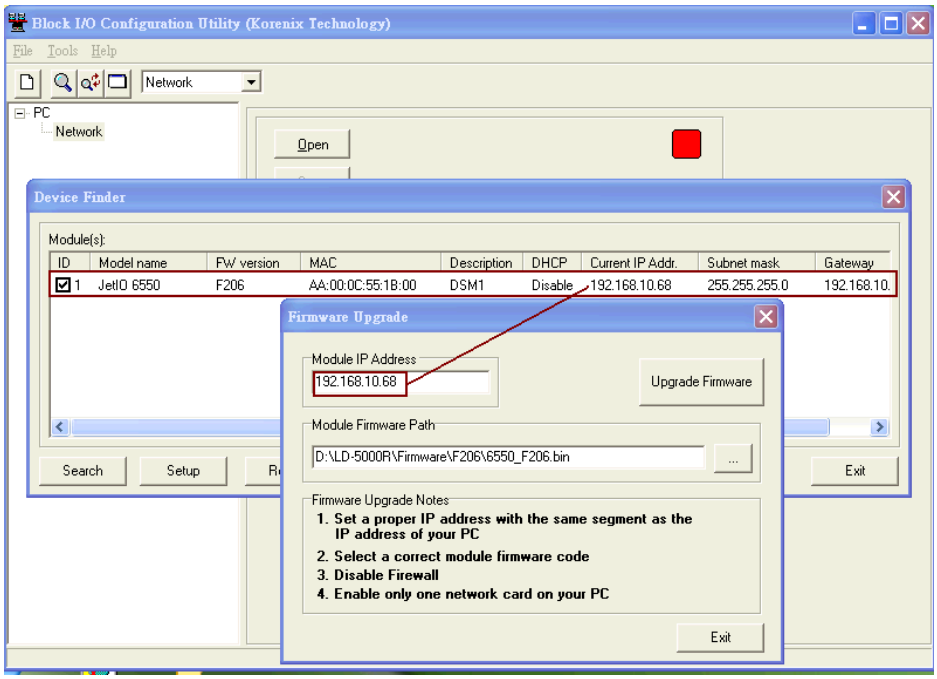
Note 1: The progress is also known as BootP, Get IP address and upgrade new firmware in the same progress. Please note that there is only one DHCP server available over the same network. Otherwise the device may get the wrong IP. Since Device Finder builds in a BOOTP server, Korenix suggests you make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

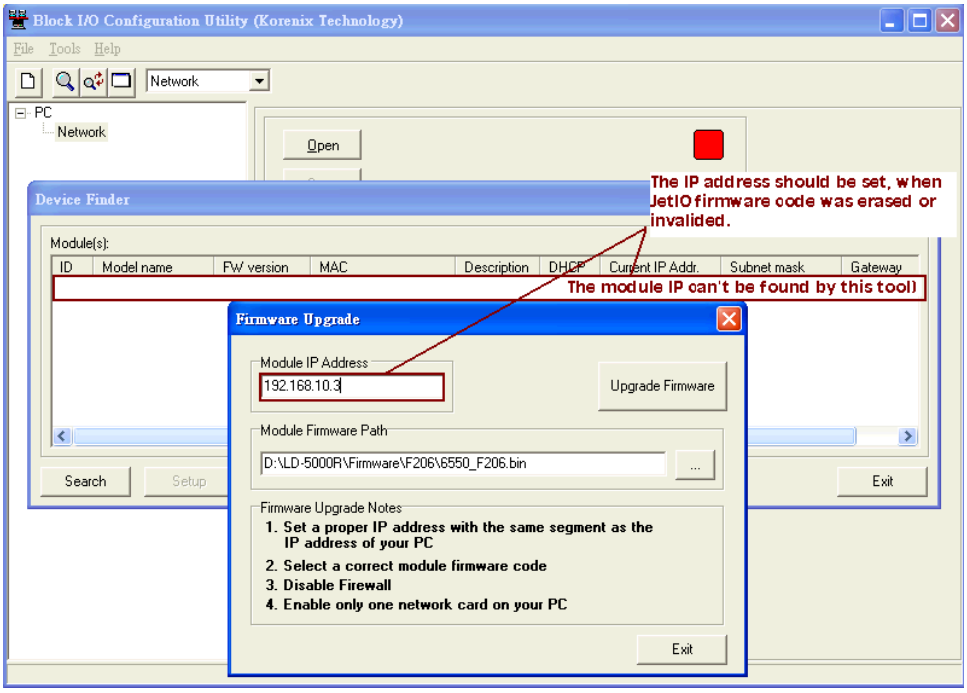


Upgrade Procedure for Mode B (Firmware Resure):

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.

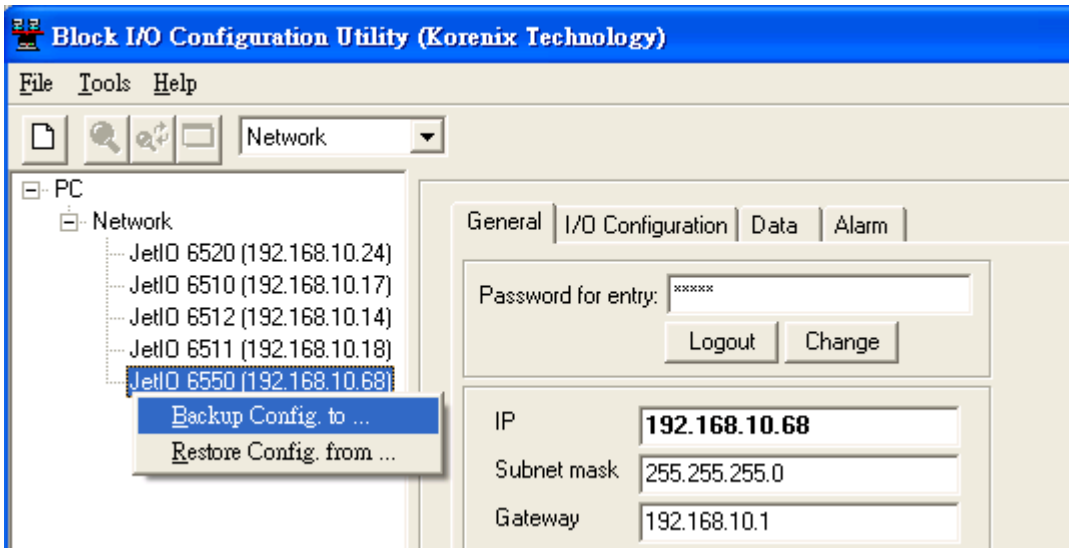
Notes:
(a). Disable Firewall
(b). Enable only one network card on your PC
(c). Set a proper IP address with the same segment as the IP address of your PC
(d). DON'T configure more than one IP address on the network interface.
(e). Select a correct module firmware code (i.e., 6550_Fxxx.bin for JetIO 6550)

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.

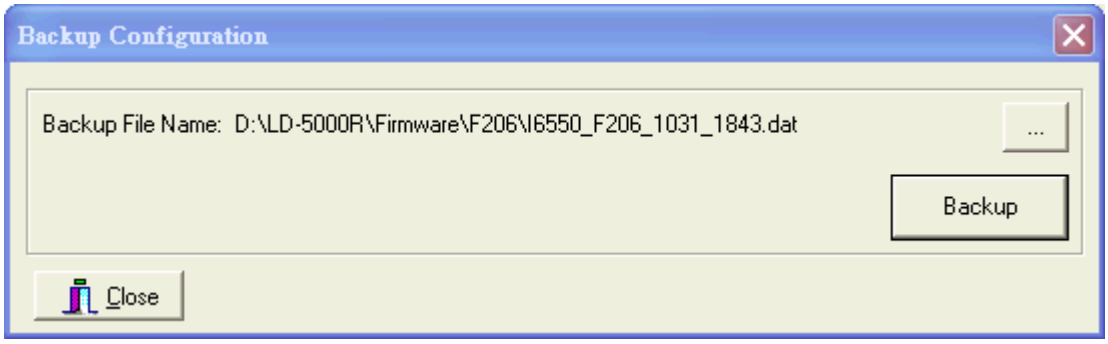


4.6 Configuration Backup/Restore, Reset Default, and Reboot

The backup/restore configuration function 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-6550 Module

[Backup]

0=di.conf

1=do.conf

2=snmp.trap.ips

3=logic.rules

4=ppio.conf

5=mis

6=#ip.conf
[mis.PLCAddr]
0=40001:40002
1=41000:41003
[di.conf.PLCAddr]
0=40011:40014
[do.conf.PLCAddr]
0=40051
1=40017:40018
2=#do.poweron.value
3=#do.safe.value
4=40035:40050
5=40019:40034
[do.poweron.value.Action]
r0=40007
w0=00001.8:@r0
w1=40007:FF00
[do.safe.value.Action]
r0=40008
w0=00001.8:@r0
w1=40008:FF00
[snmp.trap.ips.PLCAddr]
0=40098:40106
[logic.rules.PLCAddr]
0=40052:40087
1=40096:40097
2=40108:40131
[ppio.conf.PLCAddr]
0=40094:40095
[ip.conf.Program]
key=ip.conf
var0=ip.ip
var1=ip.mask
var2=ip.gateway
var3=ip.dhcp

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 6550 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 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 6550 Modbus/TCP Address Mapping

Note: Some of the results are using Hex (Ex: FF00H) mode, Signed (-1) mode... 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 6550 Common Register Map (Holding Registers, Register = 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** **BB:1 ~255** BB x 0.1 (sec)=count cycle (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:F10A (HEX)**
0004	40005	R	Read module name **R:AAAA** **AAAA: 6550(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/W	**R:** Read power on value **W:** Setting present coil value is POWERON value Read: Write: **R:AABB** **W:FF00(action value)** **AABB:0000H~00FFH** (write to EEprom)
0007	40008	R/W	**R:** Read safe value **W:** Setting present coil value is SAFE value Read: Write: **R:AABB** **W:FF00 (action value)** **AABB:0000H~00FFH** (write to EEprom)
0008	40009	R/W	(Read/Write) DI Counter status **R/W:** 00xxxxxxxxxxxxxx (bin) Bit 0 is channel 0 0: Off 1: On (start DI count)
0009	40010	R/W	(Read/Write) DI Counter Overflow status **R/W:** 00xxxxxxxxxxxxxx (bin) Bit 0 is channel 0 Read: 0 : Normal 1 : Overflow Write: 0 : clear overflow status 1 : return illegal Data Value
000A	40011	R/W	(Read/Write) DI Counter Trigger Mode **R/W:** 00xxxxxxxxxxxxxx (bin) Bit 0 is channel 0 0 : Low to High 1 : High to Low (write to EEprom)
000B	40012	R/W	(Read/Write) DI Mode **R/W:** 00xxxxxxxxxxxxxx (bin) Bit 0 is channel 0 0 : DI 1 : Counter (write to EEprom)
000C	40013	R/W	(Read/Write) DI Counter Power-On Status **R/W:** 000xxxxxxxxxxxxx (bin) Bit 0 is channel 0 0 : Off 1 : On (write to EEprom)
000D	40014	R/W	(Read/Write) DI Counter Safe Status **R/W:** 00xxxxxxxxxxxxxx (bin) Bit 0 is channel 0 0 : Off 1 : On (write to EEprom)
000E	40015	R/W	(Read/Write) DI Clear Counter Value **R:** always 0

			W: 00xxxxxxxxxxxxx (bin) Bit 0 is channel 0 Write 1: Clear Counter Value 0: Return illegal data value
000F	40016	R/W	(Read/Write) DO Pulse Operate Status **R/W:** 00000000xxxxxxxx (bin) Bit 0 is channel 0 0 : Off 1 : On (start DO pulse output)
0010	40017	R/W	(Read/Write) DO Power-On Pulse Operate Status **R/W:** 00000000xxxxxxxx (bin) Bit 0 is channel 0 0 : Off 1 : On (write to EEprom)
0011	40018	R/W	(Read/Write) DO Safe Pulse Operate Status **R/W:** 00000000xxxxxxxx (bin) Bit 0 is channel 0 0 : Off 1 : On (write to EEprom)
0012	40019	R/W	(Read/Write) Channel 0 DO Pulse Output Count Value Hi-Word
0013	40020	R/W	(Read/Write) Channel 0 DO Pulse Output Count Value Low-Word 0: Continuous count 1~ FFFFFFFF : number of pulse (write to EEprom)
0014	40021	R/W	(Read/Write) Channel 1 DO Pulse Output Count Value Hi-Word
0015	40022	R/W	(Read/Write) Channel 1 DO Pulse Output Count Value Low-Word 0: Continuous count 1~ FFFFFFFF : number of pulse(write to EEprom)
0016	40023	R/W	(Read/Write) Channel 2 DO Pulse Output Count Value Hi-Word
0017	40024	R/W	(Read/Write) Channel 2 DO Pulse Output Count Value Low-Word 0: Continuous count 1~ FFFFFFFF : number of pulse(write to EEprom)
0018	40025	R/W	(Read/Write) Channel 3 DO Pulse Output Count Value Hi-Word
0019	40026	R/W	(Read/Write) Channel 3 DO Pulse Output Count Value Low-Word 0: Continuous count 1~ FFFFFFFF : number of pulse (write to EEprom)
001A	40027	R/W	(Read/Write) Channel 4 DO Pulse Output Count Value Hi-Word
001B	40028	R/W	(Read/Write) Channel 4 DO Pulse Output Count Value Low-Word 0: Continuous count 1~ FFFFFFFF : number of pulse (write to EEprom)
001C	40029	R/W	(Read/Write) Channel 5 DO Pulse Output Count Value Hi-Word
001D	40030	R/W	(Read/Write) Channel 5 DO Pulse Output Count Value Low-Word 0: Continuous count 1~ FFFFFFFF : number of pulse (write to EEprom)
001E	40031	R/W	(Read/Write) Channel 6 DO Pulse Output Count Value Hi-Word

001F	40032	R/W	(Read/Write) Channel 6 DO Pulse Output Count Value Low-Word 0: Continuous count 1~ FFFFFFFF : number of pulse (write to EEprom)
0020	40033	R/W	(Read/Write) Channel 7 DO Pulse Output Count Value Hi-Word
0021	40034	R/W	(Read/Write) Channel 7 DO Pulse Output Count Value Low-Word 0: Continuous count 1~ FFFFFFFF : number of pulse (write to EEprom)
0022	40035	R/W	(Read/Write) Channel 0 DO Pulse Low Signal Width **R/W:** (1 ~ FFFF)* 1ms (write to EEprom)
0023	40036	R/W	(Read/Write) Channel 1 DO Pulse Low Signal Width **R/W:** (1 ~ FFFF)* 1ms (write to EEprom)
0024	40037	R/W	(Read/Write) Channel 2 DO Pulse Low Signal Width **R/W:** (1 ~ FFFF)* 1ms (write to EEprom)
0025	40038	R/W	(Read/Write) Channel 3 DO Pulse Low Signal Width **R/W:** (1 ~ FFFF)* 1ms (write to EEprom)
0026	40039	R/W	(Read/Write) Channel 4 DO Pulse Low Signal Width **R/W:** (1 ~ FFFF)* 1ms (write to EEprom)
0027	40040	R/W	(Read/Write) Channel 5 DO Pulse Low Signal Width **R/W:** (1 ~ FFFF)* 1ms (write to EEprom)
0028	40041	R/W	(Read/Write) Channel 6 DO Pulse Low Signal Width **R/W:** (1 ~ FFFF)* 1ms (write to EEprom)
0029	40042	R/W	(Read/Write) Channel 7 DO Pulse Low Signal Width **R/W:** (1 ~ FFFF)* 1ms (write to EEprom)
002A	40043	R/W	(Read/Write) Channel 0 DO Pulse High Signal Width **R/W:** (1 ~ FFFF)* 1ms (write to EEprom)
002B	40044	R/W	(Read/Write) Channel 1 DO Pulse High Signal Width **R/W:** (1 ~ FFFF)* 1ms (write to EEprom)
002C	40045	R/W	(Read/Write) Channel 2 DO Pulse High Signal Width **R/W:** (1 ~ FFFF)* 1ms (write to EEprom)
002D	40046	R/W	(Read/Write) Channel 3 DO Pulse High Signal Width **R/W:** (1 ~ FFFF)* 1ms (write to EEprom)
002E	40047	R/W	(Read/Write) Channel 4 DO Pulse High Signal Width **R/W:** (1 ~ FFFF)* 1ms (write to EEprom)
002F	40048	R/W	(Read/Write) Channel 5 DO Pulse High Signal Width **R/W:** (1 ~ FFFF)* 1ms (write to EEprom)
0030	40049	R/W	(Read/Write) Channel 6 DO Pulse High Signal Width **R/W:** (1 ~ FFFF)* 1ms (write to EEprom)
0031	40050	R/W	(Read/Write) Channel 7 DO Pulse High Signal Width **R/W:** (1 ~ FFFF)* 1ms (write to EEprom)
0032	40051	R/W	(Read/Write) DO Mode **R/W:** 00000000xxxxxxxx (bin) Bit 0 is channel 0 0 : DO 1 : Pulse (write to EEprom)
0033	40052	R/W	(Read/Write) Rule Enable Logic 0~7 **R: 00000000xxxxxxxx(bit)** **W: 00000000xxxxxxxx(bit)** 0: disable 1: enable (write to EEprom)
0034	40053	R/W	(Read/Write) Select “OR” or “AND” **R:AAAA(hex)** **W: AAAA (hex)** AAAA:00000000xxxxxxxx X: 0 “OR” Logic 1 “AND” Logic (write to EEprom)

0035	40054	R/W	(Read/Write) Select condition DI for Rule 0 **R: ABCD(hex)** **W: ABCD(hex)** A: condition #3 Logic DI Channel B: condition #2 Logic DI Channel B: condition #1 Logic DI Channel D: condition #0 Logic DI Channel(write to EEprom)
0036	40055	R/W	(Read/Write) Select condition DI for Rule 1 **R: ABCD(hex)** **W: ABCD(hex)** A: condition #3 Logic DI Channel B: condition #2 Logic DI Channel B: condition #1 Logic DI Channel D: condition #0 Logic DI Channel(write to EEprom)
0037	40056	R/W	(Read/Write) Select condition DI for Rule 2 **R: ABCD(hex)** **W: ABCD(hex)** A: condition #3 Logic DI Channel B: condition #2 Logic DI Channel B: condition #1 Logic DI Channel D: condition #0 Logic DI Channel(write to EEprom)
0038	40057	R/W	(Read/Write) Select condition DI for Rule 3 **R: ABCD(hex)** **W: ABCD(hex)** A: condition #3 Logic DI Channel B: condition #2 Logic DI Channel B: condition #1 Logic DI Channel D: condition #0 Logic DI Channel(write to EEprom)
0039	40058	R/W	(Read/Write) Select condition DI for Rule 4 **R: ABCD(hex)** **W: ABCD(hex)** A: condition #3 Logic DI Channel B: condition #2 Logic DI Channel B: condition #1 Logic DI Channel D: condition #0 Logic DI Channel(write to EEprom)
003A	40059	R/W	(Read/Write) Select condition DI for Rule 5 **R: ABCD(hex)** **W: ABCD(hex)** A: condition #3 Logic DI Channel B: condition #2 Logic DI Channel B: condition #1 Logic DI Channel D: condition #0 Logic DI Channel(write to EEprom)
003B	40060	R/W	(Read/Write) Select condition DI for Logic 6 **R: ABCD(hex)** **W: ABCD(hex)** A: condition #3 Logic DI Channel B: condition #2 Logic DI Channel B: condition #1 Logic DI Channel D: condition #0 Logic DI Channel(write to EEprom)
003C	40061	R/W	(Read/Write) Select condition DI for Rule 7 **R: ABCD(hex)** **W: ABCD(hex)** A: condition #3 Logic DI Channel B: condition #2 Logic DI Channel B: condition #1 Logic DI Channel D: condition #0 Logic DI Channel(write to EEprom)
003D	40062	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 B: Operators condition #1 Logic D: Operators condition #0 Logic Value: 0 : ON (DI) 1:OFF (DI) 2:Lo-Hi (DI) 3:Hi-Lo (DI) 4: “=” (DICounterValue = SetCounterValue) 5:”>” (DICounterValue > SetCounterValue) 6:”<” (DICounterValue < SetCounterValue) 7:”>=” (DICounterValue >= SetCounterValue) 8:”<=” (DICounterValue <= SetCounterValue) (write to EEprom)
003E	40063	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 B: Operators condition #1 Logic D: Operators condition #0 Logic Value: 0 : ON (DI) 1:OFF (DI) 2:Lo-Hi (DI) 3:Hi-Lo (DI) 4: “=” (DICounterValue = SetCounterValue) 5:”>” (DICounterValue > SetCounterValue) 6:”<” (DICounterValue < SetCounterValue) 7:”>=” (DICounterValue >= SetCounterValue) 8:”<=” (DICounterValue <= SetCounterValue) (write to EEprom)
003F	40064	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 B: Operators condition #1 Logic D: Operators condition #0 Logic 0 : ON (DI) 1:OFF (DI) 2:Lo-Hi (DI) 3:Hi-Lo (DI) 4: “=” (DICounterValue = SetCounterValue) 5:”>” (DICounterValue > SetCounterValue) 6:”<” (DICounterValue < SetCounterValue) 7:”>=” (DICounterValue >= SetCounterValue) 8:”<=” (DICounterValue <= SetCounterValue) (write to EEprom)
0040	40065	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

			B: Operators condition #1 Logic D: Operators condition #0 Logic Value: 0 : ON (DI) 1:OFF (DI) 2:Lo-Hi (DI) 3:Hi-Lo (DI) 4: “=” (DICounterValue = SetCounterValue) 5:”>” (DICounterValue > SetCounterValue) 6:”<” (DICounterValue < SetCounterValue) 7:”>=” (DICounterValue >= SetCounterValue) 8:”<=” (DICounterValue <= SetCounterValue) (write to EEprom)
0041	40066	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 B: Operators condition #1 Logic D: Operators condition #0 Logic Value: 0 : ON (DI) 1:OFF (DI) 2:Lo-Hi (DI) 3:Hi-Lo (DI) 4: “=” (DICounterValue = SetCounterValue) 5:”>” (DICounterValue > SetCounterValue) 6:”<” (DICounterValue < SetCounterValue) 7:”>=” (DICounterValue >= SetCounterValue) 8:”<=” (DICounterValue <= SetCounterValue) (write to EEprom)
0042	40067	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 B: Operators condition #1 Logic D: Operators condition #0 Logic Value: 0 : ON (DI) 1:OFF (DI) 2:Lo-Hi (DI) 3:Hi-Lo (DI) 4: “=” (DICounterValue = SetCounterValue) 5:”>” (DICounterValue > SetCounterValue) 6:”<” (DICounterValue < SetCounterValue) 7:”>=” (DICounterValue >= SetCounterValue) 8:”<=” (DICounterValue <= SetCounterValue) (write to EEprom)
0043	40068	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 B: Operators condition #1 Logic D: Operators condition #0 Logic Value:

			0 : ON (DI) 1:OFF (DI) 2:Lo-Hi (DI) 3:Hi-Lo (DI) 4: “=” (DICounterValue = SetCounterValue) 5:”>” (DICounterValue > SetCounterValue) 6:”<” (DICounterValue < SetCounterValue) 7:”>=” (DICounterValue >= SetCounterValue) 8:”<=” (DICounterValue <= SetCounterValue) (write to EEprom)
0044	40069	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 B: Operators condition #1 Logic D: Operators condition #0 Logic Value: 0 : ON (DI) 1:OFF (DI) 2:Lo-Hi (DI) 3:Hi-Lo (DI) 4: “=” (DICounterValue = SetCounterValue) 5:”>” (DICounterValue > SetCounterValue) 6:”<” (DICounterValue < SetCounterValue) 7:”>=” (DICounterValue >= SetCounterValue) 8:”<=” (DICounterValue <= SetCounterValue) (write to EEprom)
0045	40070	R/W	(Read/Write) Select THEN DO for Rule 0 **R: ABCD(hex)** **W: ABCD(hex)** A: condition #3 DO Channel B: condition #2 DO Channel B: condition #1 DO Channel D: condition #0 DO Channel(write to EEprom)
0046	40071	R/W	(Read/Write) Select THEN DO for Rule 1 **R: ABCD(hex)** **W: ABCD(hex)** A: condition #3 DO Channel B: condition #2 DO Channel B: condition #1 DO Channel D: condition #0 DO Channel(write to EEprom)
0047	40072	R/W	(Read/Write) Select THEN DO for Rule 2 **R: ABCD(hex)** **W: ABCD(hex)** A: condition #3 DO Channel B: condition #2 DO Channel B: condition #1 DO Channel D: condition #0 DO Channel(write to EEprom)
0048	40073	R/W	(Read/Write) Select THEN DO for Rule 3 **R: ABCD(hex)** **W: ABCD(hex)** A: condition #3 DO Channel B: condition #2 DO Channel B: condition #1 DO Channel D: condition #0 DO Channel(write to EEprom)
0049	40074	R/W	(Read/Write) Select THEN DO for Rule 4

			R: ABCD(hex) **W: ABCD(hex)** A: condition #3 DO Channel B: condition #2 DO Channel B: condition #1 DO Channel D: condition #0 DO Channel(write to EEprom)
004A	40075	R/W	(Read/Write) Select THEN DO for Rule 5 **R: ABCD(hex)** **W: ABCD(hex)** A: condition #3 DO Channel B: condition #2 DO Channel B: condition #1 DO Channel D: condition #0 DO Channel(write to EEprom)
004B	40076	R/W	(Read/Write) Select THEN DO for Rule 6 **R: ABCD(hex)** **W: ABCD(hex)** A: condition #3 DO Channel B: condition #2 DO Channel B: condition #1 DO Channel D: condition #0 DO Channel(write to EEprom)
004C	40077	R/W	(Read/Write) Select THEN DO for Rule 7 **R: ABCD(hex)** **W: ABCD(hex)** A: condition #3 DO Channel B: condition #2 DO Channel B: condition #1 DO Channel D: condition #0 DO Channel(write to EEprom)
004D	40078	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 B: Operators Action #1 Logic D: Operators Action #0 Logic Value: 0 : ON 1: OFF 2: Counter reset 3: DO pulse output start 4: DO pulse output stop 5: SNMP Trap(write to EEprom)
004E	40079	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 B: Operators Action #1 Logic D: Operators Action #0 Logic Value: 0 : ON 1: OFF 2: Counter reset 3: DO pulse output start 4: DO pulse output stop 5: SNMP Trap(write to EEprom)
004F	40080	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 B: Operators Action #1 Logic D: Operators Action #0 Logic Value: 0 : ON 1: OFF 2: Counter reset 3: DO pulse output start 4: DO pulse output stop 5: SNMP Trap(write to EEprom)
0050	40081	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 B: Operators Action #1 Logic D: Operators Action #0 Logic Value: 0 : ON 1: OFF 2: Counter reset 3: DO pulse output start 4: DO pulse output stop 5: SNMP Trap(write to EEprom)
0051	40082	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 B: Operators Action #1 Logic D: Operators Action #0 Logic Value: 0 : ON 1: OFF 2: Counter reset 3: DO pulse output start 4: DO pulse output stop 5: SNMP Trap(write to EEprom)
0052	40083	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 B: Operators Action #1 Logic D: Operators Action #0 Logic Value: 0 : ON 1: OFF 2: Counter reset 3: DO pulse output start 4: DO pulse output stop 5: SNMP Trap(write to EEprom)
0053	40084	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 B: Operators Action #1 Logic D: Operators Action #0 Logic Value: 0 : ON 1: OFF 2: Counter reset 3: DO pulse output start 4: DO pulse output stop 5: SNMP Trap(write to EEprom)
0054	40085	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 B: Operators Action #1 Logic D: Operators Action #0 Logic Value: 0 : ON 1: OFF 2: Counter reset 3: DO pulse output start 4: DO pulse output stop 5: SNMP Trap(write to EEprom)
0055	40086	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 EEprom)
0056	40087	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 EEprom)
0057	40088	R/W	Reserved
0058	40089	R/W	Reserved
0059	40090	R	Read device type of peer to peer **R: 0x6550 (hex)**
005A	40091	R	Read status of peer to peer **R: 0x000A (bhex)** **A=1 connect active** **A=0 not connect**
005B	40092	R	Read output coil of client / server(peer to peer) **R: xxxxxxxx(bin)** **X= 1** **X= 0**
005C	40093	R	Read input coil of client/server (peer to peer) **R: 0000xxxxxxxxxxxx(bin)** **X= 1** **X= 0**
005D	40094	R/W	(Read/Write) Peer to Peer Enable/Disable

			R: 000A(hex) **W: 000A(hex)** A: 0:disable 1:enable
005E	40095	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)
005F	40096	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 EEprom)
0060	40097	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 EEprom)
0061	40098	R/W	(Read/Write) SMNP Trap Number **R: 000X(hex)** **W: 000X(hex)** X:1~4 Trap Number (write to EEprom)
0062	40099	R/W	(Read/Write)SNMP Trap IP1 Lo-Word **R: AABB(hex)** **W: AABB(hex)** IP=X.X.AA.BB (write to EEprom)
0063	40100	R/W	(Read/Write)SNMP Trap IP1 Hi-Word **R: AABB(hex)** **W: AABB(hex)** IP=AA.BB.X.X (write to EEprom)
0064	40101	R/W	(Read/Write)SNMP Trap IP2 Lo-Word **R: AABB(hex)** **W: AABB(hex)** IP=X.X.AA.BB (write to EEprom)
0065	40102	R/W	(Read/Write)SNMP Trap IP2 Hi-Word **R: AABB(hex)** **W: AABB(hex)** IP=AA.BB.X.X (write to EEprom)
0066	40103	R/W	(Read/Write)SNMP Trap IP3 Lo-Word **R: AABB(hex)** **W: AABB(hex)** IP=X.X.AA.BB (write to EEprom)
0067	40104	R/W	(Read/Write)SNMP Trap IP3 Hi-Word **R: AABB(hex)** **W: AABB(hex)** IP=AA.BB.X.X (write to EEprom)
0068	40105	R/W	(Read/Write)SNMP Trap IP4 Lo-Word **R: AABB(hex)** **W: AABB(hex)** IP=X.X.AA.BB (write to EEprom)

0069	40106	R/W	(Read/Write)SNMP Trap IP4 Hi-Word **R: AABB(hex)** **W: AABB(hex)** IP=AA.BB.X.X (write to EEprom)
006A	40107	R/W	(Write) Repeat (Reset) enable SNMP Trap **W: FF00(hex)**
006B	40108	R/W	(Read/Write) Which DI channel is reset for Rule 0 **R/W: ABCD(hex)** A: Reset #3 Logic DI Channel B: Reset #2 Logic DI Channel B: Reset #1 Logic DI Channel D: Reset #0 Logic DI Channel(write to EEprom)
006C	40109	R/W	(Read/Write) Which DI channel is reset for Rule 1 **R/W: ABCD(hex)** A: Reset #3 Logic DI Channel B: Reset #2 Logic DI Channel B: Reset #1 Logic DI Channel D: Reset #0 Logic DI Channel(write to EEprom)
006D	40110	R/W	(Read/Write) Which DI channel is reset for Rule 2 **R/W: ABCD(hex)** A: Reset #3 Logic DI Channel B: Reset #2 Logic DI Channel B: Reset #1 Logic DI Channel D: Reset #0 Logic DI Channel(write to EEprom)
006E	40111	R/W	(Read/Write) Which DI channel is reset for Rule 3 **R/W: ABCD(hex)** A: Reset #3 Logic DI Channel B: Reset #2 Logic DI Channel B: Reset #1 Logic DI Channel D: Reset #0 Logic DI Channel(write to EEprom)
006F	40112	R/W	(Read/Write) Which DI channel is reset for Rule 4 **R/W: ABCD(hex)** A: Reset #3 Logic DI Channel B: Reset #2 Logic DI Channel B: Reset #1 Logic DI Channel D: Reset #0 Logic DI Channel(write to EEprom)
0070	40113	R/W	(Read/Write) Which DI channel is reset for Rule 5 **R/W: ABCD(hex)** A: Reset #3 Logic DI Channel B: Reset #2 Logic DI Channel B: Reset #1 Logic DI Channel D: Reset #0 Logic DI Channel(write to EEprom)
0071	40114	R/W	(Read/Write) Which DI channel is reset for Rule 6 **R/W: ABCD(hex)** A: Reset #3 Logic DI Channel B: Reset #2 Logic DI Channel B: Reset #1 Logic DI Channel D: Reset #0 Logic DI Channel(write to EEprom)
0072	40115	R/W	(Read/Write) Which DI channel is reset for Rule 7 **R/W: ABCD(hex)** A: Reset #3 Logic DI Channel B: Reset #2 Logic DI Channel B: Reset #1 Logic DI Channel D: Reset #0 Logic DI Channel(write to EEprom)
0073	40116	R/W	(Read/Write) set “if logic” DI counter value for rule 0 **R/W: ABCD(hex) low word**
0074	40117	R/W	(Read/Write) set “if logic” DI counter value for rule 0

			R/W: ABCD(hex) hi word
0075	40118	R/W	(Read/Write) set “if logic” DI counter value for rule 1 **R/W: ABCD(hex) low word**
0076	40119	R/W	(Read/Write) set “if logic” DI counter value for rule 1 **R/W: ABCD(hex) hi word**
0077	40120	R/W	(Read/Write) set “if logic” DI counter value for rule 2 **R/W: ABCD(hex) low word**
0078	40121	R/W	(Read/Write) set “if logic” DI counter value for rule 2 **R/W: ABCD(hex) hi word**
0079	40122	R/W	(Read/Write) set “if logic” DI counter value for rule 3 **R/W: ABCD(hex) low word**
007A	40123	R/W	(Read/Write) set “if logic” DI counter value for rule 3 **R/W: ABCD(hex) hi word**
007B	40124	R/W	(Read/Write) set “if logic” DI counter value for rule 4 **R/W: ABCD(hex) low word**
007C	40125	R/W	(Read/Write) set “if logic” DI counter value for rule 4 **R/W: ABCD(hex) hi word**
007D	40126	R/W	(Read/Write) set “if logic” DI counter value for rule 5 **R/W: ABCD(hex) low word**
007E	40127	R/W	(Read/Write) set “if logic” DI counter value for rule 5 **R/W: ABCD(hex) hi word**
007F	40128	R/W	(Read/Write) set “if logic” DI counter value for rule 6 **R/W: ABCD(hex) low word**
0080	40129	R/W	(Read/Write) set “if logic” DI counter value for rule 6 **R/W: ABCD(hex) hi word**
0081	40130	R/W	(Read/Write) set “if logic” DI counter value for rule 7 **R/W: ABCD(hex) low word**
0082	40131	R/W	(Read/Write) set “if logic” DI counter value for rule 7 **R/W: ABCD(hex) hi word**
01F4	40501	R/W	(Read/Write) Select DI channel or auxiliary flag for rule 0~3 R: xxxx xxxx xxxx xxxx(bit) Rule 3 Rule 2 Rule 1 Rule 0 W: xxxx xxxx xxxx xxxx(bit) Rule 3 Rule 2 Rule 1 Rule 0 0: select DI channel 1: select auxiliary flag channel (write to Flash RAM)
01F5	40502	R/W	(Read/Write) Select DI channel or auxiliary flag for rule 4~7 R: xxxx xxxx xxxx xxxx(bit) Rule 7 Rule 6 Rule 5 Rule 4 W: xxxx xxxx xxxx xxxx(bit) Rule 7 Rule 6 Rule5 Rule4 0: select DI channel 1: select auxiliary flag channel (write to Flash RAM)
01F6	40503	R/W	(Read/Write) Select DI channel or auxiliary flag for rule 8~11

			R: xxxx xxxx xxxx xxxx(bit) Rule11 Rule10 Rule9 Rule8 W: xxxx xxxx xxxx xxxx(bit) Rule11 Rule10 Rule9 Rule8 0: select DI channel 1: select auxiliary flag channel (write to Flash RAM)
01F7	40504	R/W	(Read/Write) Select DI channel or auxiliary flag for rule 12~15 R: xxxx xxxx xxxx xxxx(bit) Rule15 Rule14 Rule13 Rule12 W: xxxx xxxx xxxx xxxx(bit) Rule15 Rule14 Rule13 Rule12 0: select DI channel 1: select auxiliary flag channel (write to Flash RAM)
01F8	40505	R/W	(Read/Write) Select DO channel or auxiliary flag for rule 0~3 R: xxxx xxxx xxxx xxxx(bit) W: xxxx xxxx xxxx xxxx(bit) 0: select DO channel 1: select auxiliary flag channel (write to Flash RAM)
01F9	40506	R/W	(Read/Write) Select DO channel or auxiliary flag for rule 4~7 R: xxxx xxxx xxxx xxxx(bit) W: xxxx xxxx xxxx xxxx(bit) 0: select DO channel 1: select auxiliary flag channel (write to Flash RAM)
01FA	40507	R/W	(Read/Write) Select DO channel or auxiliary flag for rule 8~11 R: xxxx xxxx xxxx xxxx(bit) W: xxxx xxxx xxxx xxxx(bit) 0: select DO channel 1: select auxiliary flag channel (write to Flash RAM)
01FB	40508	R/W	(Read/Write) Select DO channel or auxiliary flag for rule 12~15 R: xxxx xxxx xxxx xxxx(bit) W: xxxx xxxx xxxx xxxx(bit) 0: select DO channel 1: select auxiliary flag channel (write to Flash RAM)

01FC	40509	R/W	(Read/Write) Select PTP Enable/disable condition flag R: 0000 0000 xxxx xxxx(bit) W: 0000 0000 xxxx xxxx(bit) x: 0 disable 1 enable(write to Flash RAM)
01FD	40510	R/W	(Read/Write) PTP select condition channel for 4 condition R: ABCD W: ABCD A: condition #3 B: condition #2 A: condition #1 B: condition #0 (write to Flash RAM)
01FE	40511	R/W	(Read/Write) PTP select condition channel for 4 condition R: ABCD W: ABCD A: condition #7 B: condition #6 A: condition #5 B: condition #4 (write to Flash RAM)
01FF	40512	R/W	(Read/Write) PTP select action channel for 4 condition R: ABCD W: ABCD A: action #3 B: action #2 A: action #1 B: action #0 (write to Flash RAM)
0200	40513	R/W	(Read/Write) PTP select action channel for 4 condition R: ABCD W: ABCD A: action #7 B: action #6 A: action #5 B: action #4 (write to Flash RAM)
0201	40514	R/W	(Read/Write) Select DI channel or auxiliary flag R: 0000 0000 xxxx xxxx(bit) W: 0000 0000 xxxx xxxx(bit)

			x:0: select DI channel 1: select auxiliary flag channel (write to Flash RAM)
0202	40515	R/W	(Read/Write) Select DO channel or auxiliary flag R: 0000 0000 xxxx xxxx(bit) W: 0000 0000 xxxx xxxx(bit) x:0: select DO channel 1: select auxiliary flag channel (write to Flash RAM)
0203	40516	R/W	(Read/Write) PTP IP address ip[0] low word for server R: AABB W: AABB(write to Flash RAM)
0204	40517	R/W	(Read/Write) PTP IP address ip[0] hi word for server R: AABB W: AABB(write to Flash RAM)
0205	40518	R/W	(Read/Write) PTP IP address ip[1] low word for server R: AABB W: AABB(write to Flash RAM)
0206	40519	R/W	(Read/Write) PTP IP address ip[1] hi word for server R: AABB W: AABB(write to Flash RAM)
0207	40520	R/W	(Read/Write) PTP IP address ip[2] low word for server R: AABB W: AABB(write to Flash RAM)
0208	40521	R/W	(Read/Write) PTP IP address ip[2] hi word for server R: AABB W: AABB(write to Flash RAM)
0209	40522	R/W	(Read/Write) PTP IP address ip[3] low word for server R: AABB W: AABB(write to Flash RAM)
020A	40523	R/W	(Read/Write) PTP IP address ip[3] hi word for server R: AABB W: AABB(write to Flash RAM)
020B	40524	R/W	(Read/Write) PTP IP address ip[4] low word for server R: AABB W: AABB(write to Flash RAM)
020C	40525	R/W	(Read/Write) PTP IP address ip[4] hi word for server R: AABB W: AABB(write to Flash RAM)
020D	40526	R/W	(Read/Write) PTP IP address ip[4] low word for server

			R: AABB W: AABB(write to Flash RAM)
020E	40527	R/W	(Read/Write) PTP IP address ip[4] hi word for server R: AABB W: AABB(write to Flash RAM)
020F	40528	R/W	(Read/Write) PTP IP address ip[6] low word for server R: AABB W: AABB(write to Flash RAM)
0210	40529	R/W	(Read/Write) PTP IP address ip[6] hi word for server R: AABB W: AABB(write to Flash RAM)
0211	40530	R/W	(Read/Write) PTP IP address ip[7] low word for server R: AABB W: AABB(write to Flash RAM)
0212	40531	R/W	(Read/Write) PTP IP address ip[7] hi word for server R: AABB W: AABB(write to Flash RAM)
0213	40532	R/W	(Read/Write) PTP IP address ip[8] low word for client R: AABB W: AABB(write to Flash RAM)
0214	40533	R/W	(Read/Write) PTP IP address ip[8] hi word for client R: AABB W: AABB(write to Flash RAM)
0215	40534	R/W	(Read/Write) RuleEnable Logic 0~7 R: 00000000xxxxxxxx(bit) W: 00000000xxxxxxxx(bit) 0: disable 1: enable (write to Flash RAM)
0216	40535	R/W	(Read/Write) Select “OR” or “AND” R:AAAA(hex) W: AAAA (hex) AAAA:00000000xxxxxxxx X: 0 “OR” Logic 1 “AND” Logic (write to Flash RAM)
0217	40536	R/W	(Read/Write) Select condition DI for Rule 8 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic DI Channel

			B: condition #2 Logic DI Channel C: condition #1 Logic DI Channel D: condition #0 Logic DI Channel(write to Flash RAM)
0218	40537	R/W	(Read/Write) Select condition DI for Rule 9 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic DI Channel B: condition #2 Logic DI Channel C: condition #1 Logic DI Channel D: condition #0 Logic DI Channel(write to Flash RAM)
0219	40538	R/W	(Read/Write) Select condition DI for Rule 10 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic DI Channel B: condition #2 Logic DI Channel C: condition #1 Logic DI Channel D: condition #0 Logic DI Channel(write to Flash RAM)
021A	40539	R/W	(Read/Write) Select condition DI for Rule 11 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic DI Channel B: condition #2 Logic DI Channel C: condition #1 Logic DI Channel D: condition #0 Logic DI Channel(write to Flash RAM)
021B	40540	R/W	(Read/Write) Select condition DI for Rule 12 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic DI Channel B: condition #2 Logic DI Channel C: condition #1 Logic DI Channel D: condition #0 Logic DI Channel(write to Flash RAM)
021C	40541	R/W	(Read/Write) Select condition DI for Rule 13 R: ABCD(hex)

			W: ABCD(hex) A: condition #3 Logic DI Channel B: condition #2 Logic DI Channel C: condition #1 Logic DI Channel D: condition #0 Logic DI Channel(write to Flash RAM)
021D	40542	R/W	(Read/Write) Select condition DI for Rule 14 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic DI Channel B: condition #2 Logic DI Channel C: condition #1 Logic DI Channel D: condition #0 Logic DI Channel(write to Flash RAM)
021E	40543	R/W	(Read/Write) Select condition DI for Rule 15 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic DI Channel B: condition #2 Logic DI Channel C: condition #1 Logic DI Channel D: condition #0 Logic DI Channel(write to Flash RAM)
021F	40544	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 (DI) 1:OFF (DI) 2:Lo-Hi (DI) 3:Hi-Lo (DI) 4: “=” (DICounterValue = SetCounterValue) 5:”>” (DICounterValue > SetCounterValue) 6:”<” (DICounterValue < SetCounterValue) 7:”>=” (DICounterValue >= SetCounterValue)

			8:”<=” (DICounterValue <= SetCounterValue) (write to Flash RAM)
0220	40545	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 (DI) 1:OFF (DI) 2:Lo-Hi (DI) 3:Hi-Lo (DI) 4: “=” (DICounterValue = SetCounterValue) 5:”>” (DICounterValue > SetCounterValue) 6:”<” (DICounterValue < SetCounterValue) 7:”>=” (DICounterValue >= SetCounterValue) 8:”<=” (DICounterValue <= SetCounterValue) (write to Flash RAM)
0221	40546	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 (DI) 1:OFF (DI) 2:Lo-Hi (DI) 3:Hi-Lo (DI) 4: “=” (DICounterValue = SetCounterValue) 5:”>” (DICounterValue > SetCounterValue) 6:”<” (DICounterValue < SetCounterValue) 7:”>=” (DICounterValue >= SetCounterValue) 8:”<=” (DICounterValue <= SetCounterValue) (write to Flash RAM)

0222	40547	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 (DI) 1:OFF (DI) 2:Lo-Hi (DI) 3:Hi-Lo (DI) 4: “=” (DICounterValue = SetCounterValue) 5:”>” (DICounterValue > SetCounterValue) 6:”<” (DICounterValue < SetCounterValue) 7:”>=” (DICounterValue >= SetCounterValue) 8:”<=” (DICounterValue <= SetCounterValue) (write to Flash RAM)
0223	40548	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 (DI) 1:OFF (DI) 2:Lo-Hi (DI) 3:Hi-Lo (DI) 4: “=” (DICounterValue = SetCounterValue) 5:”>” (DICounterValue > SetCounterValue) 6:”<” (DICounterValue < SetCounterValue) 7:”>=” (DICounterValue >= SetCounterValue) 8:”<=” (DICounterValue <= SetCounterValue) (write to Flash RAM)
0224	40549	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 (DI) 1:OFF (DI) 2:Lo-Hi (DI) 3:Hi-Lo (DI) 4: “=” (DICounterValue = SetCounterValue) 5:”>” (DICounterValue > SetCounterValue) 6:”<” (DICounterValue < SetCounterValue) 7:”>=” (DICounterValue >= SetCounterValue) 8:”<=” (DICounterValue <= SetCounterValue) (write to Flash RAM)
0225	40550	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 (DI) 1:OFF (DI) 2:Lo-Hi (DI) 3:Hi-Lo (DI) 4: “=” (DICounterValue = SetCounterValue) 5:”>” (DICounterValue > SetCounterValue) 6:”<” (DICounterValue < SetCounterValue) 7:”>=” (DICounterValue >= SetCounterValue) 8:”<=” (DICounterValue <= SetCounterValue) (write to Flash RAM)
0226	40551	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 (DI) 1:OFF (DI) 2:Lo-Hi (DI) 3:Hi-Lo (DI) 4: “=” (DICounterValue = SetCounterValue) 5:”>” (DICounterValue > SetCounterValue) 6:”<” (DICounterValue < SetCounterValue) 7:”>=” (DICounterValue >= SetCounterValue) 8:”<=” (DICounterValue <= SetCounterValue) (write to Flash RAM)
0227	40552	R/W	(Read/Write) Select THEN DO for Rule 8 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic DO Channel B: condition #2 Logic DO Channel C: condition #1 Logic DO Channel D: condition #0 Logic DO Channel(write to Flash RAM)
0228	40553	R/W	(Read/Write) Select THEN DO for Rule 9 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic DO Channel B: condition #2 Logic DO Channel C: condition #1 Logic DO Channel D: condition #0 Logic DO Channel(write to Flash RAM)
0229	40554	R/W	(Read/Write) Select THEN DO for Rule 10 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic DO Channel B: condition #2 Logic DO Channel C: condition #1 Logic DO Channel D: condition #0 Logic DO Channel(write to Flash RAM)

022A	40555	R/W	(Read/Write) Select THEN DO for Rule 11 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic DO Channel B: condition #2 Logic DO Channel C: condition #1 Logic DO Channel D: condition #0 Logic DO Channel(write to Flash RAM)
022B	40556	R/W	(Read/Write) Select THEN DO for Rule 12 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic DO Channel B: condition #2 Logic DO Channel C: condition #1 Logic DO Channel D: condition #0 Logic DO Channel(write to Flash RAM)
022C	40557	R/W	(Read/Write) Select THEN DO for Rule 13 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic DO Channel B: condition #2 Logic DO Channel C: condition #1 Logic DO Channel D: condition #0 Logic DO Channel(write to Flash RAM)
022D	40558	R/W	(Read/Write) Select THEN DO for Rule 14 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic DO Channel B: condition #2 Logic DO Channel C: condition #1 Logic DO Channel D: condition #0 Logic DO Channel(write to Flash RAM)
022E	40559	R/W	(Read/Write) Select THEN DO for Rule 15 R: ABCD(hex) W: ABCD(hex) A: condition #3 Logic DO Channel B: condition #2 Logic DO Channel C: condition #1 Logic DO Channel

			D: condition #0 Logic DO Channel(write to Flash RAM)
022F	40560	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 1: OFF 2: Counter reset 3: DO pulse output start 4: DO pulse output stop 5: SNMP Trap (write to Flash RAM)
0230	40561	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 1: OFF 2: Counter reset 3: DO pulse output start 4: DO pulse output stop 5: SNMP Trap (write to Flash RAM)
0231	40562	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 1: OFF 2: Counter reset 3: DO pulse output start 4: DO pulse output stop 5: SNMP Trap (write to Flash RAM)
0232	40563	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 : DO ON 1: DO OFF 2: DI Counter reset 3: DO pulse output start 4: DO pulse output stop 5: SNMP Trap (write to Flash RAM)
0233	40564	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 1: OFF 2: Counter reset 3: DO pulse output start 4: DO pulse output stop 5: SNMP Trap (write to Flash RAM)
0234	40565	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 1: OFF 2: Counter reset 3: DO pulse output start 4: DO pulse output stop 5: SNMP Trap(write to Flash RAM)
0235	40566	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 1: OFF 2: Counter reset 3: DO pulse output start 4: DO pulse output stop 5: SNMP Trap (write to Flash RAM)
0236	40567	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 1: OFF 2: Counter reset 3: DO pulse output start 4: DO pulse output stop 5: SNMP Trap (write to Flash RAM)

0237	40568	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)
0238	40569	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 13 D: xxxx x= 0:disable 1:enable rule 12 (write to Flash RAM)
0239	40570	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)
023A	40571	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)
023B	40572	R/W	(Read/Write) Which DI channel is reset for Rule 8 R/W: ABCD(hex) A: Reset #3 Logic DI Channel B: Reset #2 Logic DI Channel C: Reset #1 Logic DI Channel D: Reset #0 Logic DI Channel(write to Flash RAM)

023C	40573	R/W	(Read/Write) Which DI channel is reset for Rule 9 R/W: ABCD(hex) A: Reset #3 Logic DI Channel B: Reset #2 Logic DI Channel C: Reset #1 Logic DI Channel D: Reset #0 Logic DI Channel(write to Flash RAM)
023D	40574	R/W	(Read/Write) Which DI channel is reset for Rule 10 R/W: ABCD(hex) A: Reset #3 Logic DI Channel B: Reset #2 Logic DI Channel C: Reset #1 Logic DI Channel D: Reset #0 Logic DI Channel(write to Flash RAM)
023E	40575	R/W	(Read/Write) Which DI channel is reset for Rule 11 R/W: ABCD(hex) A: Reset #3 Logic DI Channel B: Reset #2 Logic DI Channel C: Reset #1 Logic DI Channel D: Reset #0 Logic DI Channel(write to Flash RAM)
023F	40576	R/W	(Read/Write) Which DI channel is reset for Rule 12 R/W: ABCD(hex) A: Reset #3 Logic DI Channel B: Reset #2 Logic DI Channel C: Reset #1 Logic DI Channel D: Reset #0 Logic DI Channel(write to Flash RAM)
0240	40577	R/W	(Read/Write) Which DI channel is reset for Rule 13 R/W: ABCD(hex) A: Reset #3 Logic DI Channel B: Reset #2 Logic DI Channel C: Reset #1 Logic DI Channel D: Reset #0 Logic DI Channel(write to Flash RAM)
0241	40578	R/W	(Read/Write) Which DI channel is reset for Rule 14 R/W: ABCD(hex) A: Reset #3 Logic DI Channel B: Reset #2 Logic DI Channel C: Reset #1 Logic DI Channel D: Reset #0 Logic DI Channel(write to Flash RAM)
0242	40579	R/W	(Read/Write) Which DI channel is reset for Rule 15 R/W: ABCD(hex)

			A: Reset #3 Logic DI Channel B: Reset #2 Logic DI Channel C: Reset #1 Logic DI Channel D: Reset #0 Logic DI Channel(write to Flash RAM)
0243	40580	R/W	(Read/Write) set “if logic” DI counter value for rule 8 R/W: ABCD(hex) low word
0244	40581	R/W	(Read/Write) set “if logic” DI counter value for rule 8 R/W: ABCD(hex) hi word
0245	40582	R/W	(Read/Write) set “if logic” DI counter value for rule 9 R/W: ABCD(hex) low word
0246	40583	R/W	(Read/Write) set “if logic” DI counter value for rule 9 R/W: ABCD(hex) hi word
0247	40584	R/W	(Read/Write) set “if logic” DI counter value for rule 10 R/W: ABCD(hex) low word
0248	40585	R/W	(Read/Write) set “if logic” DI counter value for rule 10 R/W: ABCD(hex) hi word
0249	40586	R/W	(Read/Write) set “if logic” DI counter value for rule 11 R/W: ABCD(hex) low word
024A	40587	R/W	(Read/Write) set “if logic” DI counter value for rule 11 R/W: ABCD(hex) hi word
024B	40588	R/W	(Read/Write) set “if logic” DI counter value for rule 12 R/W: ABCD(hex) low word
024C	40589	R/W	(Read/Write) set “if logic” DI counter value for rule 12 R/W: ABCD(hex) hi word
024D	40590	R/W	(Read/Write) set “if logic” DI counter value for rule 13 R/W: ABCD(hex) low word
024E	40591	R/W	(Read/Write) set “if logic” DI counter value for rule 13 R/W: ABCD(hex) hi word
024F	40592	R/W	(Read/Write) set “if logic” DI counter value for rule 14 R/W: ABCD(hex) low word
0250	40593	R/W	(Read/Write) set “if logic” DI counter value for rule 14 R/W: ABCD(hex) hi word
0251	40594	R/W	(Read/Write) set “if logic” DI counter value for rule 15 R/W: ABCD(hex) low word
0252	40595	R/W	(Read/Write) set “if logic” DI counter value for rule 15 R/W: ABCD(hex) hi word
0253	40596	R/W	(Read/Write) internal flags R: xxxxxxxxxxxxxxxx

			X:0 or 1
0254	40597	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
0255	40598	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
0256	40599	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
0257	40600	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
0258	40601	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
0259	40602	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
025A	40603	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
025B	40604	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
025C	40605	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 6550 DI Counter Register Map (Input Register, Function Code=04)			
Protocol Address (Hex)	**PLC Address (Decimal)**	**Access**	**Description**
0000	30001	R	Channel 0 DI Counter Value Hi-Word
0001	30002	R	Channel 0 DI Counter Value Low-Word
0002	30003	R	Channel 1 DI Counter Value Hi-Word
0003	30004	R	Channel 1 DI Counter Value Low-Word
0004	30005	R	Channel 2 DI Counter Value Hi-Word
0005	30006	R	Channel 2 DI Counter Value Low-Word
0006	30007	R	Channel 3 DI Counter Value Hi-Word
0007	30008	R	Channel 3 DI Counter Value Low-Word
0008	30009	R	Channel 4 DI Counter Value Hi-Word
0009	30010	R	Channel 4 DI Counter Value Low-Word
000A	30011	R	Channel 5 DI Counter Value Hi-Word
000B	30012	R	Channel 5 DI Counter Value Low-Word
000C	30013	R	Channel 6 DI Counter Value Hi-Word
000D	30014	R	Channel 6 DI Counter Value Low-Word
000E	30015	R	Channel 7 DI Counter Value Hi-Word
000F	30016	R	Channel 7 DI Counter Value Low-Word

0010	30017	R	Channel 8 DI Counter Value Hi-Word
0011	30018	R	Channel 8 DI Counter Value Low-Word
0012	30019	R	Channel 9 DI Counter Value Hi-Word
0013	30020	R	Channel 9 DI Counter Value Low-Word
0014	30021	R	Channel 10 DI Counter Value Hi-Word
0015	30022	R	Channel 10 DI Counter Value Low-Word
0016	30023	R	Channel 11 DI Counter Value Hi-Word
0017	30024	R	Channel 11 DI Counter Value Low-Word
0018	30025	R	Channel 12 DI Counter Value Hi-Word
0019	30026	R	Channel 12 DI Counter Value Low-Word
001A	30027	R	Channel 12 DI Counter Value Hi-Word
001B	30028	R	Channel 12 DI Counter Value Low-Word
001C	30029	R	Channel 13 DI Counter Value Hi-Word
001D	30030	R	Channel 13 DI Counter Value Low-Word

JetI/O 6550 Digital Input Register Map (Input Status, FC=02)			
Protocol Address (Hex)	**PLC Address (Decimal)**	**Access**	**Description**
0000	10001	R	Channel 0 Digital input signal
0001	10002	R	Channel 1 Digital input signal
0002	10003	R	Channel 2 Digital input signal
0003	10004	R	Channel 3 Digital input signal
0004	10005	R	Channel 4 Digital input signal
0005	10006	R	Channel 5 Digital input signal
0006	10007	R	Channel 6 Digital input signal
0007	10008	R	Channel 7 Digital input signal
0008	10009	R	Channel 8 Digital input signal
0009	10010	R	Channel 9 Digital input signal
000A	10011	R	Channel 10 Digital input signal
000B	10012	R	Channel 11 Digital input signal
000C	10013	R	Channel 12 Digital input signal
000D	10014	R	Channel 13 Digital input signal

JetI/O 6550 Digital Output Register Map (Coil Status, FC=01)			
Protocol Address (Hex)	**PLC Address (Decimal)**	**Access**	**Description**
0000	00001	R/W	Channel 0 Digital Output signal
0001	00002	R/W	Channel 1 Digital Output signal
0002	00003	R/W	Channel 2 Digital Output signal

0003	00004	R/W	Channel 3 Digital Output signal
0004	00005	R/W	Channel 4 Digital Output signal
0005	00006	R/W	Channel 5 Digital Output signal
0006	00007	R/W	Channel 6 Digital Output signal
0007	00008	R/W	Channel 7 Digital Output signal

Note: When the host failed, writing output coil is invalid and return exception code 04.

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. SYNTAX: *DisplayString* (SIZE (0..31))	Public, Read Only
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 – 6550

Object ID (OID)	Description	Community, R/W Access
diStatus	This object shows you the status of the Digital Input channels. Use Binary display mode to read the value. The binary number 0 represent the DI OFF, 1 represent DI ON mode. The last bit represent the first channel(Ch0). For example: Ch0-3 is DI OFF, Ch4-13 is DI ON, Ch14-15 remains 00, the value you can get is 00111111 11110000(3FF0 in 16xHex)	RO
diMode	This object shows you the mode of the Digital Input channels. Use Binary display mode to read the value. The binary number 0 represent the DI mode, 1 represent Event Counter mode. The last bit represent the first channel(Ch0). For example: Ch0-3 is DI, Ch4-13 is Event Counter mode, Ch14-15 remains 00, the value you can get/set is 00111111 11110000(3FF0 in 16xHex, 16368 in 10xHex). Type 16368 to set the value.	RW
diCounterStatus	This object shows you the status of the Event Counter channel. Use Binary display mode to read the value. The binary number 0 represent the Event Counter OFF/Stop, 1 represent Event Counter ON/Start. The last bit represent the first channel(Ch0). For example: Ch0-3 is OFF, Ch4-5 is Start, Ch6-13 is Stop, Ch14-15 remains 00, the value you can get/set is 00000000 00110000(0030 in 16xHex, 48 in 10xHex) Type 48 to set the value.	RW
diCounterOverflowStatus	This object shows you the status of the Event Counter overflow. The binary number 0 represents not overflow, 1 represents overfolw. The last bit represent the first channel(Ch0). The first	RO

	2 bits(Ch14-15) remain 00	
diCounterTriggerMode	This object shows you the status of the Event Counter Trigger mode. The binary number 0 represent the Low to High mode, 1 represent the High to Low mode. The last bit represent the first channel(Ch0). The first 2 bits(Ch14-15) remain 00	RW
diClrCounterValue	This object shows you the Event Counter Reset mode. Use Binary display mode to read the value. The binary number 0 represents not reset mode, 1 represents reset mode. The last bit represent the first channel(Ch0). The first 2 bits (Ch14-15) remain 00.	RW
doStatus	This object shows you the status of the Digital Output channels. Use Binary display mode to read the value. The binary number 0 represent the DO OFF, 1 represent the DO ON. The last bit represent the first channel(Ch0). The first 2 bits(Ch14-15) remain 00	RW
doPulseOperateStatus	This object shows you the status of the Pulse Output channels. Use Binary display mode to read the value. The binary number 0 represent the Pulse Output OFF, 1 represent the Pulse Output ON. The last bit represent the first channel(Ch0). The first 2 bits(Ch14-15) remain 00	RW
do00PulseLowWidth	Low value of the Pulse mode parameter. The unit of the value is millisecond.	RW
do00PulseHiWidth	High value of the Pulse mode parameter. The unit of the value is millisecond	RW
do01PulseLowWidth	Low value of the Pulse mode parameter. The unit of the value is millisecond.	RW
do01PulseHiWidth	High value of the Pulse mode parameter. The unit of the value is millisecond	RW
do02PulseLowWidth	Low value of the Pulse mode parameter. The unit of the value is millisecond.	RW
do02PulseHiWidth	High value of the Pulse mode parameter. The unit of the value is millisecond	RW
do03PulseLowWidth	Low value of the Pulse mode parameter.	RW

	The unit of the value is millisecond.	
do03PulseHiWidth	High value of the Pulse mode parameter. The unit of the value is millisecond	RW
do04PulseLowWidth	Low value of the Pulse mode parameter. The unit of the value is millisecond.	RW
do04PulseHiWidth	High value of the Pulse mode parameter. The unit of the value is millisecond	RW
do05PulseLowWidth	Low value of the Pulse mode parameter. The unit of the value is millisecond.	RW
do05PulseHiWidth	High value of the Pulse mode parameter. The unit of the value is millisecond	RW
do06PulseLowWidth	Low value of the Pulse mode parameter. The unit of the value is millisecond.	RW
do06PulseHiWidth	High value of the Pulse mode parameter. The unit of the value is millisecond	RW
do07PulseLowWidth	Low value of the Pulse mode parameter. The unit of the value is millisecond.	RW
do07PulseHiWidth	High value of the Pulse mode parameter. The unit of the value is millisecond	RW
doMode	This object shows you the mode of the Digital Output channels. Use Binary display mode to read the value. The binary number 0 represent the DO mode, 1 represent the Pulse Output mode.The last bit represent the first channel(Ch0). For example: Ch0-3 is DO, Ch4-13 are Pulse Output mode, Ch14-15 remain 00, the value you can get/set is 00111111 11110000(3FF0 in 16xHex)	RW
ruleEnable	This object shows you which rule is triggered.	RO

6.2 Revision History

Version	Description	Date
1.6	● Update General page ● Update I/O Configuration page ● Update “Data” page ● Update “Logic Rules” page ● Update “Peer to Peer I/O” page ● Add “SNMP” page ● Add new Modbus/TCP registers in FW208	Sept. 1, 2009
1.5	● Fix error of digital input modbus address	Mar. 18, 2008
1.4	● Add configuration backup/restore ● Simply firmware upgrade procedure	Dec. 24, 2008
1.21	● Change Pin No. table ● Change Device Finder Utility to Device Finder Popup Window due to Device Finder Utility is merged. ● Remove Web configure ● 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 Password Registers, 41000-41003 ● Web display, not web configuration ● Add Peer-to-Peer mapping	Jun. 25, 2008
1.2	Add Emulation mode operation description. Add Terminal mode operation description. 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 Reset command description Add description for Modbus/TCP address mapping. Correct Wordings: Logic rule ->I/O rule, I/O module -> I/O server… Add description of the SNMP and Trap.	Apr. 17, 2008
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 V0.3 to V1.0	Jan. 30, 2008