

# MODBUS TCP/UDP Network Adapter

# GN-9289

## User Manual



Version 1.01

**2019 CREVIS Co.,Ltd**

---

DOCUMENT CHANGE SUMMARY				
REV	PAGE	REMARKS	DATE	EDITOR
1.00	New Document		2018/07/30	MJ Kwon
1.01	48	Modbus special Register Map	2019/05/31	MJ Kwon

## Table of Contents

<b>1. Important Notes</b> .....	7
<b>1.1. Safety Instruction</b> .....	8
<b>1.1.1. Symbols</b> .....	8
<b>1.1.2. Safety Notes</b> .....	8
<b>2. Specification</b> .....	9
<b>2.1. The Interface</b> .....	9
<b>2.1.1. GN-9289(MODBUS TCP)</b> .....	9
<b>2.2. General Specification</b> .....	10
<b>2.2.1. General Specification</b> .....	10
<b>2.2.2. Interface Specification</b> .....	11
<b>2.3. GN-9289 LED Indicator</b> .....	12
<b>2.3.1. Module Status LED (MOD)</b> .....	12
<b>2.3.2. Physical Connection LED (LINK)</b> .....	12
<b>2.3.3. Exchange Data/Traffic Present LED (ACTIVE)</b> .....	13
<b>2.3.4. Expansion I/O Module Status LED (IOS)</b> .....	13
<b>2.3.5. Field Power Status LED</b> .....	13
<b>3. Dimension</b> .....	14
<b>3.1. GN-9289</b> .....	14
<b>4. Mechanical Set Up</b> .....	15
<b>4.1. Total Expansion</b> .....	15
<b>4.2. Plugging and Removal of the Components</b> .....	15
<b>4.3. Module mounting</b> .....	16
<b>4.4. How to supply the power correctly</b> .....	17
<b>5. Convenience Function</b> .....	18
<b>5.1. Web Server</b> .....	18
<b>5.2. IAP Functionality</b> .....	20
<b>6. GN-9289 Communication Interface</b> .....	22
<b>6.1. RJ-45 Socket</b> .....	22
<b>6.2. Dip Switch (TBD)</b> .....	22
<b>6.3. RS232 Port for MODBUS/RTU, Touch Panel or IOGuide</b> .....	23
<b>6.4. MODBUS/TCP IP – Address Setup</b> .....	24

6.4.1 IP-Address Setup using BOOTP/DHCP Sever.....	24
6.4.2 IP-Address Setup using DIP switch(Manual function).....	25
7. I/O Process Image Map.....	29
7.1. MODBUS Interface Register/Bit Map.....	30
7.2. Example of Input Process Image (Input Register) Map.....	31
7.3. Example of Output Process Image (Output Register) Map.....	32
8. MODBUS TCP/ UDP INTERFACE.....	33
8.1. MODBUS TCP/ UDP Protocol.....	33
8.1.1 Comparison of MODBUS TCP/ UDP And MODUB/RTU.....	33
8.1.2 MODBUS TCP/ UDP MBAP Header.....	34
8.2. Supported MODBUS Function Codes.....	35
8.2.1 1 (0x01) Read Coils.....	35
8.2.2 2 (0x02) Read Discrete Inputs.....	36
8.2.3 3 (0x03) Read Holding Resgisters.....	37
8.2.4 4 (0x04) Read Input Resgisters.....	38
8.2.5 5 (0x05) Write Single Coil.....	39
8.2.6 6 (0x06) Write Single Register.....	40
8.2.7 8 (0x08) Diagnostics.....	41
8.2.8 15 (0x0F) Write Multiple Coils.....	43
8.2.9 16 (0x10) Write Multiple Resgisters.....	44
8.2.10 23 (0x17) Read/Write Multiple Resgisters.....	45
8.2.11 Error Response.....	46
8.3. MODBUS Special Register Map.....	47
8.3.1 Adapter Identification Special Resgister (0x1000, 4096).....	47
8.3.2 Adapter Watchdog Time, other Time Special Register (0x1020, 4128).....	48
8.3.3 Adapter TCP/IP Special Register (0x1040, 4160).....	48
8.3.4 Adapter Information Special Register (0x1100, 4352).....	49
8.3.5 Expasion Slot Information Special Resister (0x2000, 8192).....	50
8.4. Supported MODBUS Function Codes.....	53
9. Object models.....	54
9.1. Supported Objects.....	54
9.2. Identity Object.....	55

9.2.1 Common Services .....	55
9.2.2 Class Attributes.....	55
9.2.3 Instance Attributes.....	55
9.3. Message Router Object.....	57
9.3.1 Common Services .....	57
9.3.2 Class Attributes.....	57
9.3.3 Instance Attributes.....	57
9.4. Assembly Object.....	58
9.4.1 Common Services .....	58
9.4.2 Class Attributes.....	58
9.4.3 Class Attributes.....	58
9.5. Connection Manager Object.....	58
9.5.1 Class Attributes, Instance Attribute .....	58
9.6. Port Object.....	59
9.6.1 Common Services .....	59
9.6.2 Class Attributes.....	59
9.6.3 Class Attributes.....	59
9.7. TCP/IP Object.....	60
9.7.1 Common Services .....	60
9.7.2 Class Attributes.....	60
9.7.3 Class Attributes.....	60
9.7.3.1. Status Instance Attributes.....	61
9.7.3.2. Status Instance Attributes.....	61
9.8. Ethernet Link Object.....	62
9.8.1 Common Services .....	62
9.8.2 Class Attributes.....	62
9.8.3 Class Attributes.....	63
9.9. G-Bus Manager Object.....	64
9.9.1 Common Services .....	64
9.9.2 Class Attributes.....	64
9.9.3 Class Attributes.....	64
9.10. Expansion Slot Object.....	66

<b>9.10.1 Common Services</b> .....	66
<b>9.10.2 Class Attributes</b> .....	66
<b>9.10.3 Instance Attributes</b> .....	66
<b>10. Trouble Shooting</b> .....	69
<b>10.1 How to diagnose by LED indicator</b> .....	69

## 1. Important Notes

Solid state equipment has operational characteristics differing from those of electromechanical equipment.

Safety Guidelines for the Application, Installation and Maintenance of Solid State Controls describes some important differences between solid state equipment and hard-wired electromechanical devices.

Because of this difference, and also because of the wide variety of uses for solid state equipment, all persons responsible for applying this equipment must satisfy themselves that each intended application of this equipment is acceptable.

In no event will CREVIS be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular installation, CREVIS cannot assume responsibility or liability for actual use based on the examples and diagrams.

### Warning!

✓ **If you don't follow the directions, it could cause a personal injury, damage to the equipment or explosion**

- Do not assemble the products and wire with power applied to the system. Else it may cause an electric arc, which can result into unexpected and potentially dangerous action by field devices. Arching is explosion risk in hazardous locations. Be sure that the area is non-hazardous or remove system power appropriately before assembling or wiring the modules.
- Do not touch any terminal blocks or IO modules when system is running. Else it may cause the unit to an electric shock or malfunction.
- Keep away from the strange metallic materials not related to the unit and wiring works should be controlled by the electric expert engineer. Else it may cause the unit to a fire, electric shock or malfunction

### Caution!



✓ **If you disobey the instructions, there may be possibility of personal injury, damage to equipment or explosion. Please follow below Instructions.**

- " Check the rated voltage and terminal array before wiring. Avoid the circumstances over 50°C of temperature. Avoid placing it directly in the sunlight.
- " Avoid the place under circumstances over 85% of humidity.
- " Do not place Modules near by the inflammable material. Else it may cause a fire.
- " Do not permit any vibration approaching it directly.
- " Go through module specification carefully, ensure inputs, output connections are made with the specifications. Use standard cables for wiring.


" Use Product under pollution degree 2 environment.

## 1.1. Safety Instruction

### 1.1.1. Symbols

<p><b>DANGER</b></p> 	<p>Identifies information about practices or circumstances that can cause an explosion in a hazardous environment, which may lead to personal injury or death property damage, or economic loss</p>
<p><b>IMPORTANT</b></p>	<p>Identifies information that is critical for successful application and understanding of the product</p>
<p><b>ATTENTION</b></p> 	<p>Identifies information about practices or circumstances that can lead to personal injury, property damage, or economic loss. Attentions help you to identity a hazard, avoid a hazard, and recognize the consequences</p>

### 1.1.2. Safety Notes

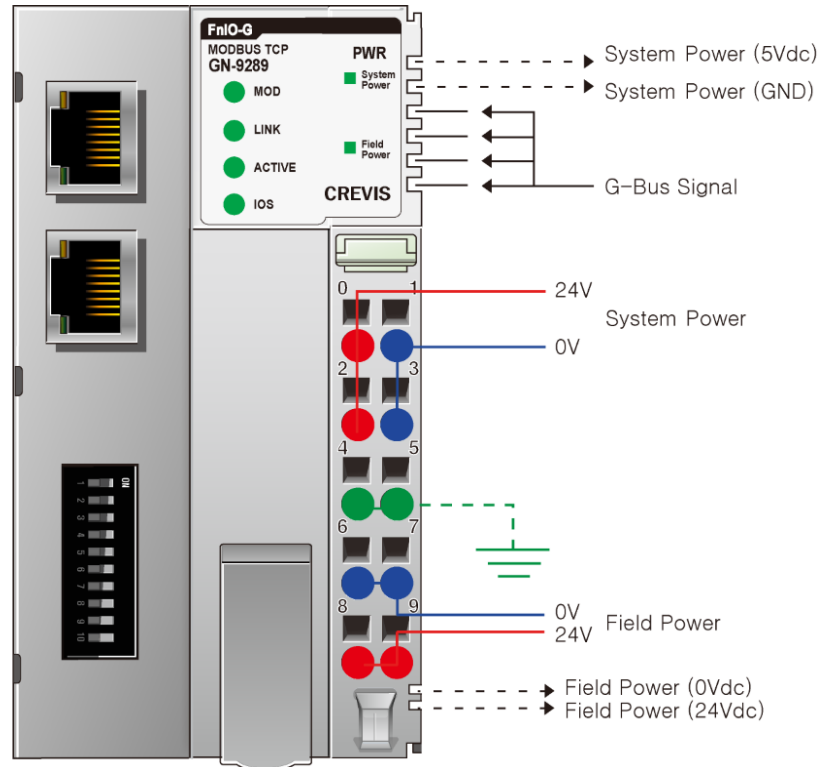
<p><b>DANGER</b></p> 	<p>The modules are equipped with electronic components that may be destroyed by electrostatic discharge. When handling the modules, ensure that the environment (persons, workplace and packing) is well grounded. Avoid touching conductive components, G-BUS Pin.</p>
--	---



## 2. Specification

### 2.1. The Interface

#### 2.1.1 GN-9289(MODBUS TCP)



System power and Field power must be supplied separately.

Pin No.	Signal Description	Signal Description	Pin No.
0	System Power, 24V	System Power, Ground	1
2	System Power, 24V	System Power, Ground	3
4	F.G	F.G	5
6	Field Power, Ground	Field Power, Ground	7
8	Field Power, 24V	Field Power, 24V	9

## 2.2. General Specification

### 2.2.1. General Specification

General specification	
System Power	Supply voltage : 24Vdc nominal Supply voltage range : 15~32Vdc Protection : Output current limit (Min. 1.5A) Reverse polarity protection
Power Dissipation	70mA typical @ 24Vdc
Current for I/O Module	1.5A @ 5Vdc (When using in '60°C ~ 70°C' temperature environment, the power dissipation is limited to 0.8A.)
Isolation	System power to internal logic : Non-isolation System power I/O driver : Isolation
Field Power	Supply voltage : 24Vdc typical (Max. 32Vdc) * Field Power Range is different depending on IO Module series. Refer to IO Module`s Specification.
Max. Current Field Power Contact	DC 10A Max
Weight	162g
Module Size	54mm x 99mm x 70mm
Environmental specification	
Operating Temperature	60°C ~ 70°C : Power dissipation is limited to 0.8A. -40°C ~ 60°C : 1.5A full load is allowed.
UL Temperature	-20°C~60°C
Storage Temperature	-40°C~85°C
Relative Humidity	5% ~ 90% non-condensing
Mounting	DIN rail

\* When using Ethernet/IP only, tcp timeout connection time value should be '0'. '0' value means tcp connection time out is disabled. Refer to 'Adapter TCP/IP Special Register (0x1040, 4160)' at section '3.3.3'

\*\* When using in '60°C ~ 70°C' temperature environment, the power dissipation is limited to 0.8A.

## 2.2.2 Interface Specification

Items	Specification
<b>Input Specification</b>	
Adapter Type	Slave node (MODBUS/TCP,MODBUS/UDP Server)
Protocol	MODBUS/TCP,MODBUS/UDP,HTTP,DHCP,10 TCP Connections
Sub-Protocol	*Ethernet/IP
Max. Expantsion Module	63 slots
Max. Data Size(Input + Output)	Max 128 bytes each slot
Max Length Bus Line	Up to 100m from Ethernet Hub/Switch with twisted CAT5 UTP/STP
Max. Nodes	Limited by Ethernet Specification.
Baud Rate	10/100Mbps, Auto-negotiation, Full duplex
Interface Connector	RJ-45 socket * 2pcs
IP-Address Setup	Via DHCP/BOOTP or IOGuide(Crevis Software)
IP-Address Range	xxx.xxx.xxx.1 ~ 253 (User area) xxx.xxx.xxx.254 ~ 255 (Reserved for IAP Function)
Serial Port	RS232 for MODBUS/RTU, Touch Pannel or IOGuide
Serial Configuration (RS232)	Node : 1 (Fixed) Baud Rate : 115200 (Fixed) Data bit : 8 (Fixed) Parity bit : No parity (Fixed) Stop bit : 1 (Fixed)
Indicator	6 LEDs 1 Green/Red, Module Status (MOD) 1 Green, Physical Connection (LINK) 1 Green, Exchange Data/Traffic Present (ACTIVE) 1 Green/Red, Expansion I/O Module Status (IOS) 1 Green, System Power Status 1 Green, Field Power Status 2 LEDs (each RJ45 Connector) 1 Yellow, Link/Active 1 Green, Not used
Module Location	Starter module left side of G-Series system
Field Power Detection	About 14Vdc

## 2.3. GN-9289 LED Indicator

LED No.	LED Function / Description	LED Color
MOD	Module Status	Green/Red
LINK	Physical Connection	Green
ACTIVE	Exchange Data/Traffic Present	Green
IOS	Extension Module Status	Green/Red
System Power	System Power Enable	Green
Field Power	Field Power Enable	Green

### 2.3.1. Module Status LED (MOD)

Status	LED	To indicate
Not Powered	OFF	Not power is supplied to the unit.
Device Operational	Green	The unit is operating in normal condition.
Device in Standby	Flashing Green	The device needs commissioning due to configuration missing, incomplete or incorrect.
MODBUS Error	Green/Red Toggle	MODBUS error such as watchdog error, etc.
Minor Fault	Flashing Red	Recoverable Fault. - EEPROM checksum fault.
Unrecoverable Fault	Red	The device has an unrecoverable fault. - Memory error or CPU watchdog error.

### 2.3.2. Physical Connection LED(LINK)

Status	LED	To indicate
Not Powered or Not Linked	OFF	Device may not be powered
Adapter physical connected	Green	Adapter Ethernet Controller physically connected

### 2.3.3 Exchange Data/Traffic Present LED(ACTIVE)

Status	LED	To indicate
Not Powered	OFF	Device is idle or may not be powered.
Adapter exchange data	Flashing Green	Adapter(slave) exchange data/Traffic present. About 10msec flashing.

### 2.3.4 Extension Module Status LED(LOS)

Status	LED	To indicate
Not Powered	OFF	Device may not be powered.
No Expansion Module	Flashing Red	Adapter has no expansion module
Internal Bus Connection, Run Exchanging I/O	Green	Exchanging I/O data.
Expansion Configuration Failed	Red	One or more expansion module occurred in fault state. <ul style="list-style-type: none"> <li>- Detected invalid expansion module ID.</li> <li>- Overflowed Input/Output Size</li> <li>- Too many expansion module</li> <li>- Initialization failure</li> <li>- Communication failure.</li> <li>- Changed expansion module configuration.</li> <li>- Mismatch vendor code between adapter and expansion module.</li> </ul>

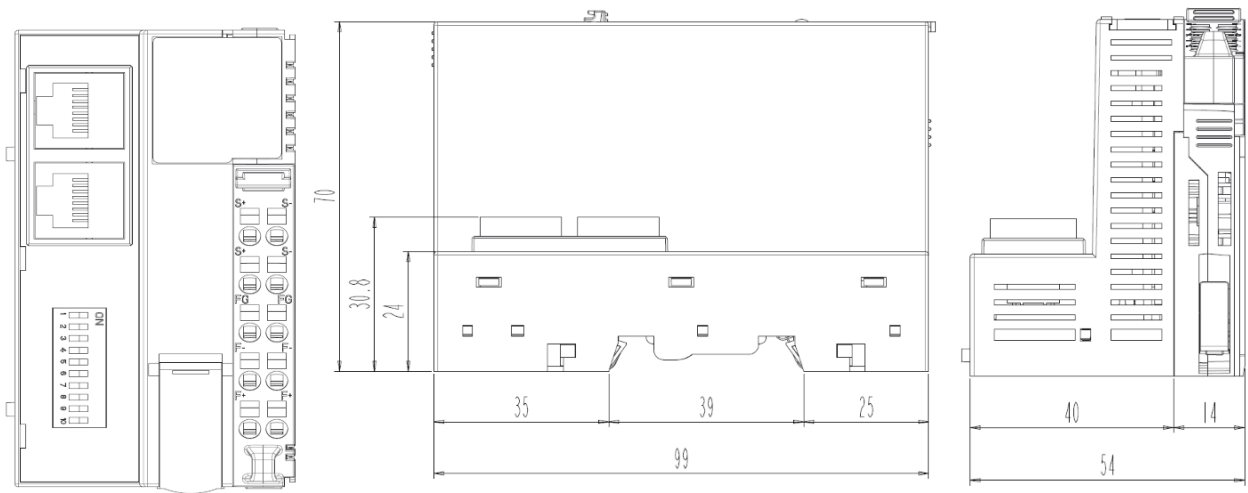
### 2.3.5 Field Power Status LED

Status	LED	To indicate
No field, System power	OFF	Not supplied 24Vdc field power, 5Vdc system power.
Supplied field, System power	Green	Supplied 24Vdc field power, 5Vdc system power.

### 3. Dimension

#### 3.1 GN-9289

(mm)



## 4. Mechanical Set Up

### 4.1 Total Expansion

The number of the module assembly that can be connected is 63. So the maximum length is 426mm  
Exception GN-9289 is excepted to calculate maximum length because that is double width module.

### 4.2. Plugging and Removal of the Components.



As above figure in order to safeguard the FnIO module from jamming, it should be fixed onto the DIN rail with locking level. To do so, fold on the upper of the locking lever.

To pull out the FnIO module, unfold the locking lever as below figure.

**DANGER**

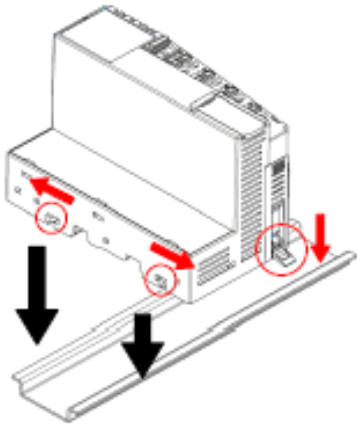


Before work is done on the components, the voltage supply must be turned off.

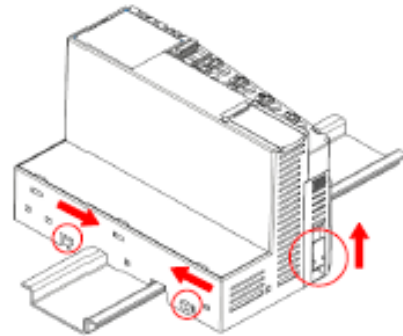
## 4.3 Module mounting

### How to mount on DIN-rail

- ① Press down the module lightly on the DIN-rail  
※ DIN-rail lockers and lever should be unlocked in this step.

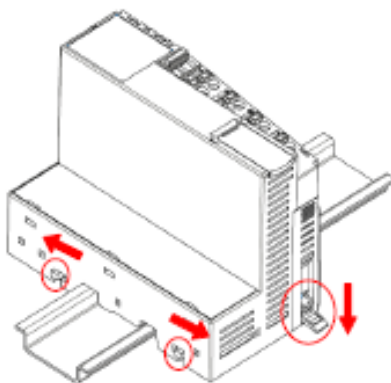


- ② Lock DIN-rail lockers and lever in right direction

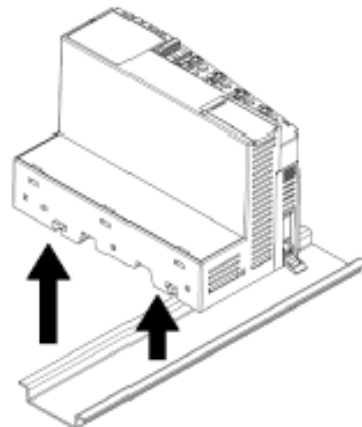


### How to dismount from DIN-rail

- ① Unlock DIN-rail lockers and lever in right direction

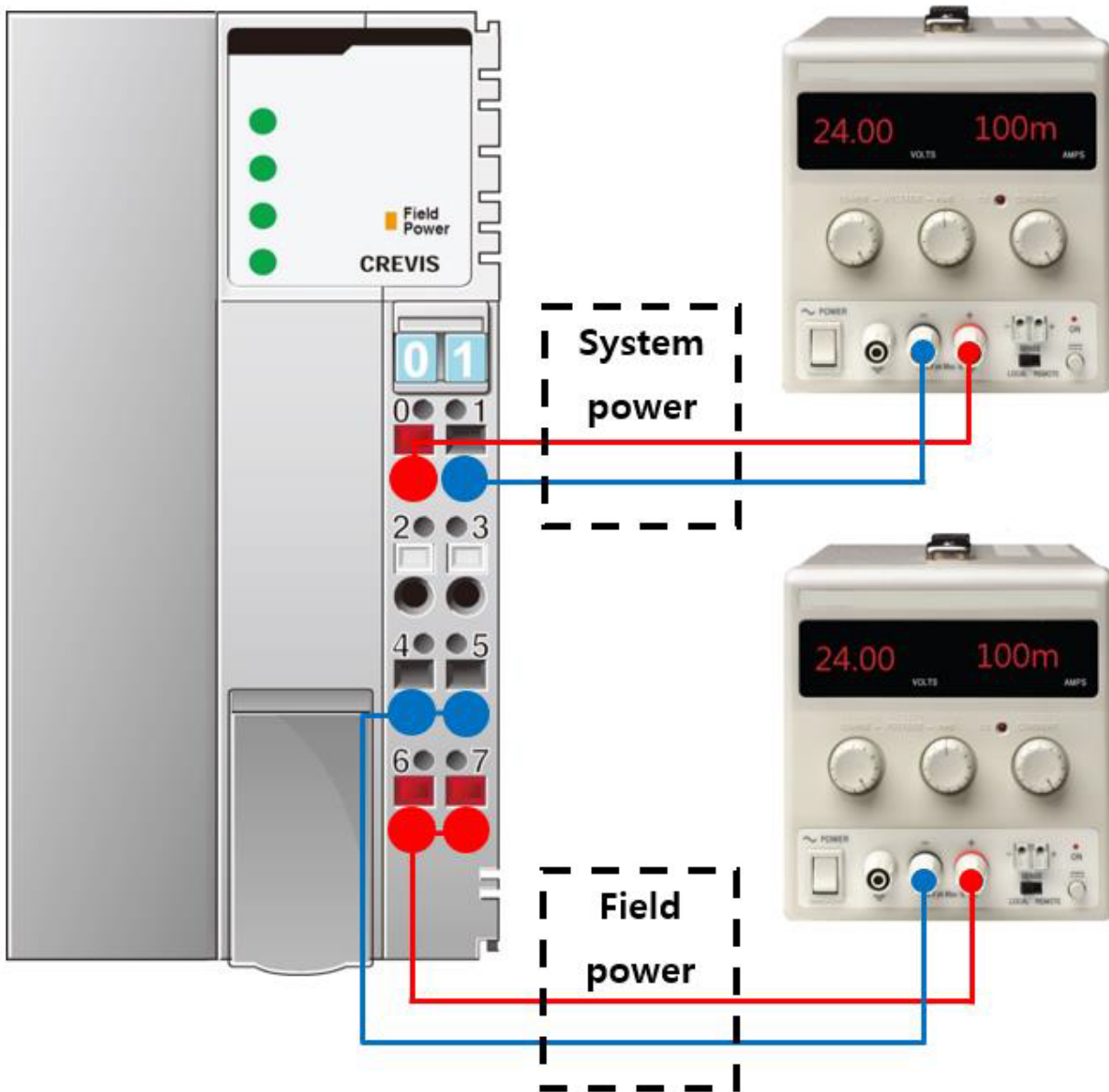


- ② Pull up the module to remove from the DIN-rail





#### 4.4 How to supply the power correctly




- Refer the pin map of power RTB before wiring.
- Current capacity margin should be considered when you supply the power.
- Supply voltage 24Vdc which is recommended voltage level.
- **System and Field power must be supplied separately as the picture above.**

※ If the power is supplied to each System and Field power from same power source, it would be vulnerable to power noise.

## 5. Convenience Function

### 5.1. Web Server

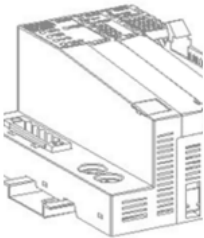
Network Adapter



www.crevis.co.kr

Network Adapter

Expansion Module



Crevis FnIO
The Creative present makes Vision of future

Network Adapter  
**GN-9289 (Modbus/TCP,UDP with Ethernet/IP) Network Adapter**

---

Io Input Data / Io Output Data

---

- IP Address : 192.168.100.100
- Subnet Mask : 255.255.0.0
- Gateway : 192.168.0.1
- MAC Address : 00:14:F7:00:00:02
- DIP DHCP/BOOTP : Disabled
- DIP IP Address : Disabled

---


- TCP/UDP Connections : Available
- Ethernet/IP Connections : Available
- MODBUS/TCP Connections : Available
- MODBUS/UDP Connections : Available
- HTTP(Web Server) Connections : Available
- MODBUS/RTU(RS232) Communication : Available

---

- Firmware Revision : 1.000(09/03/2015)
- Expansion Modules : 1 module(s)
- IO Size(Input) : 0 byte(s)
- IO Size(Output) : 2 byte(s)

---

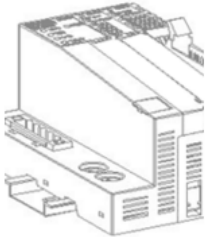
Expansion Module



www.crevis.co.kr

Network Adapter

Expansion Module



**Crevis FnIO**      The Creative present makes Vision of future

---

Network Adapter  
**GN-9289 (Modbus/TCP,UDP with Ethernet/IP) Network Adapter**

---

Io Input Data / Io Output Data

---

Slot#	Descriptions	Input Reg. Mapping	Output Reg. Mapping
<b>Slot#1</b>	GT-222F, 16DO, 24Vdc, Source		

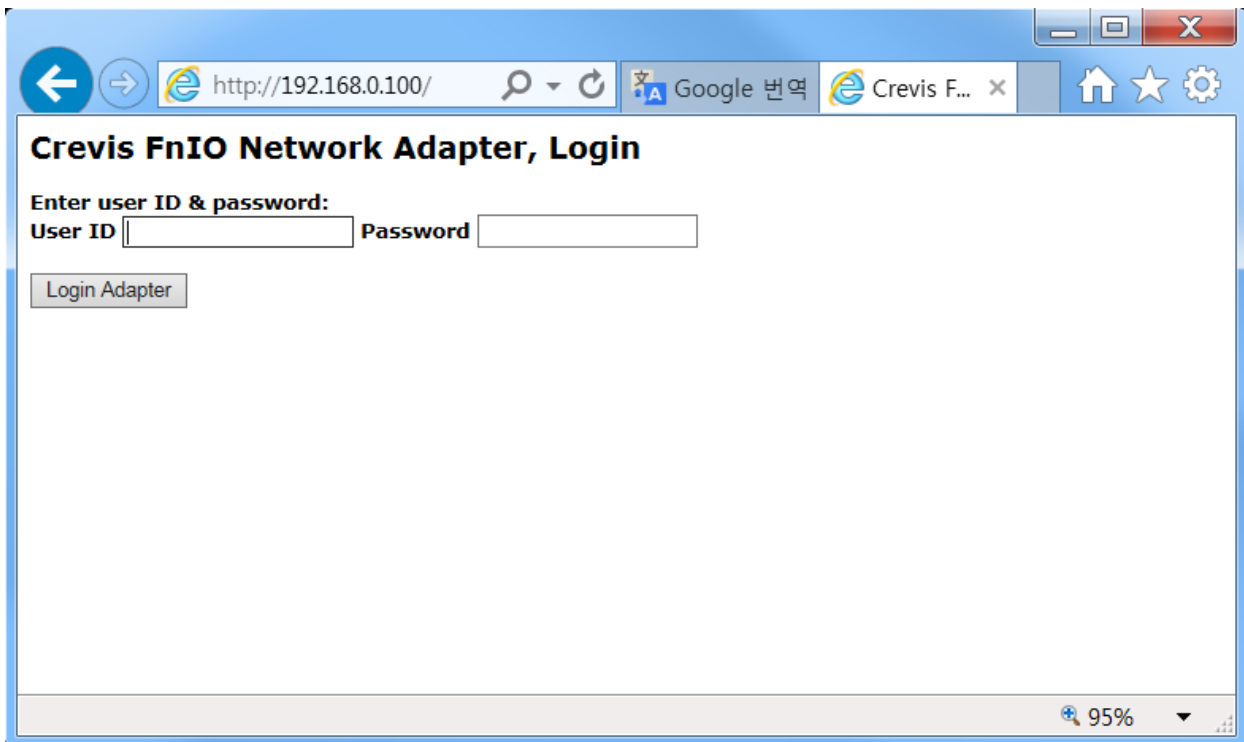
---

## 5.2 IAP Functionality

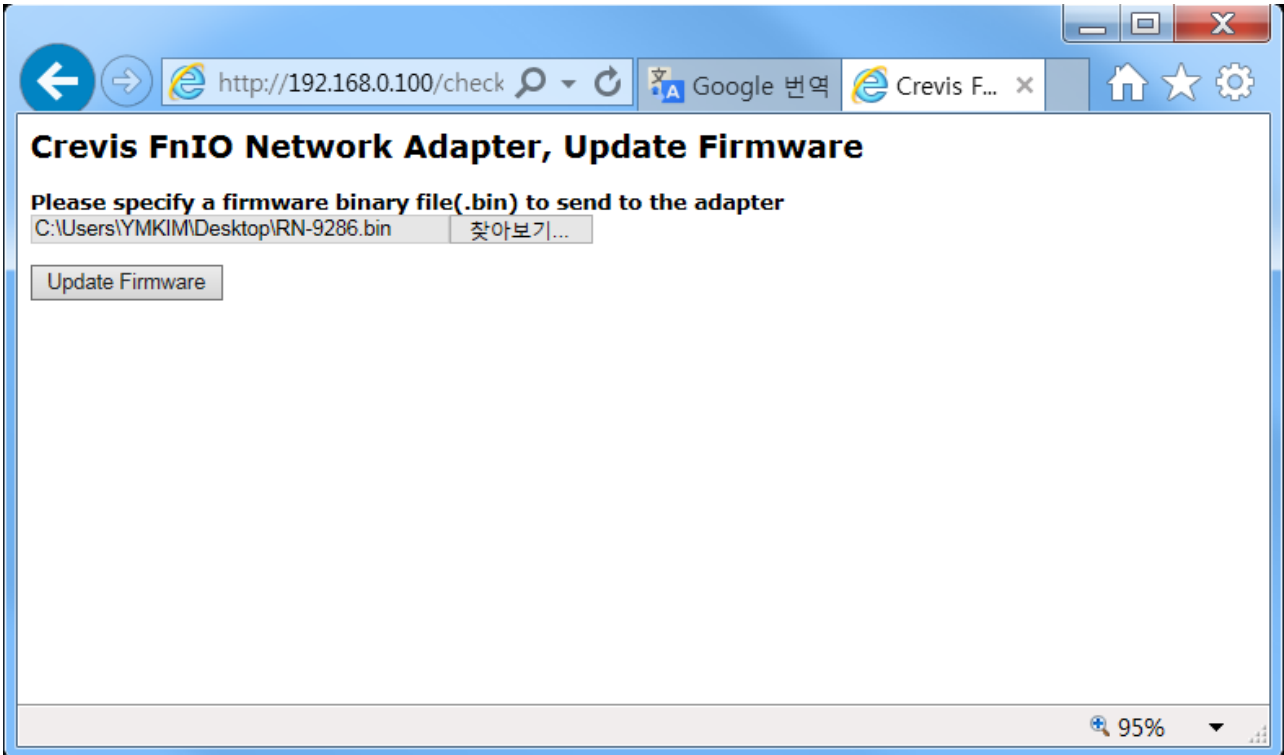
After setting up the dip switch as 254 or 255, Power off and on in order to start the IAP mode.  
(254=Fixed IP Address(192.168.0.100)/255=Last used IP Address)

After Login the Web Browser connecting

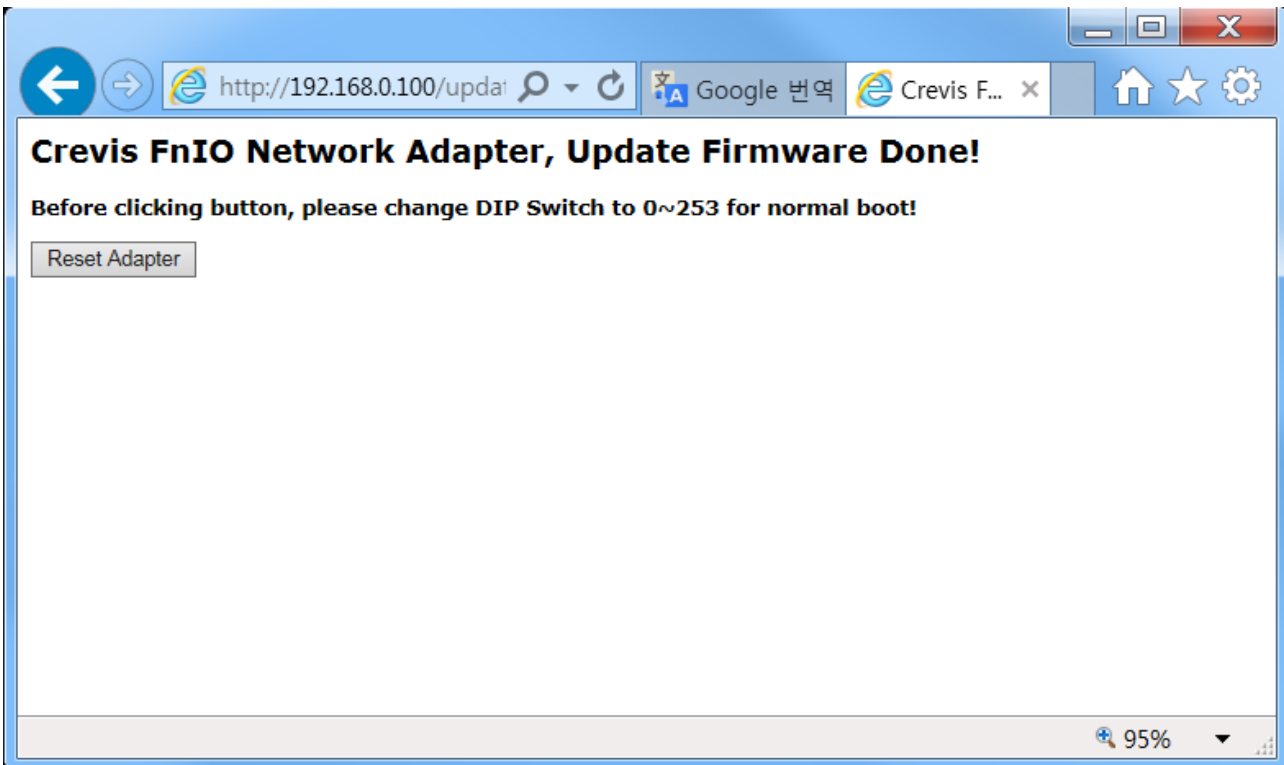
(User ID : Crevis, Password : Crevis)



To update the binary file.

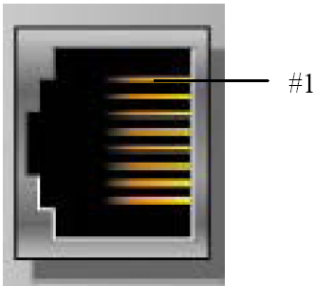


Has been updated. Please reboot the adapter.



## 6. GN-9289 Communication Interface

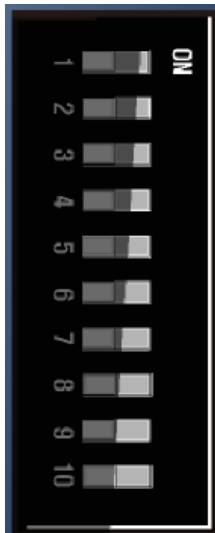
### 6.1 RJ-45 Socket



Shielded RJ-45 Socket

RJ-45	Signal Name	Description
1	TD+	Transmit +
2	TD-	Transmit -
3	RD+	Receive +
4		
5		
6	RD-	Receive -
7		
8		
Case	Shield	

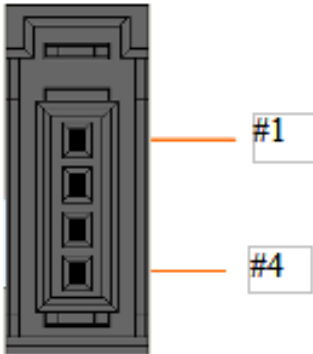
### 6.2 Dip Switch (TBD)



DIP Pole#	Description	
1	IP_DIP bit#0	Lowest IP Address when Pole#10=ON ex) XXX.XXX.XXX.IP_DIP
2	IP_DIP bit#1	
3	IP_DIP bit#2	
4	IP_DIP bit#3	
5	IP_DIP bit#4	
6	IP_DIP bit#5	
7	IP_DIP bit#6	
8	IP_DIP bit#7	
9	= ON : Enable DHCP/BOOTP *	
10	= ON : Use Lowest IP Address with IP_DIP value	

\* DHCP/BOOTP have to be set in special register(default : BOOTP).  
(0x1045, ref 3.3.3).

### 6.3 RS232 Port for MODBUS/RTU, Touch Panel or IOGuide



Pin#	Signal Name	Description
1	Reserved	----
2	TXD	RS232 TXD
3	RXD	RS232 RXD
4	GND	RS232 GND

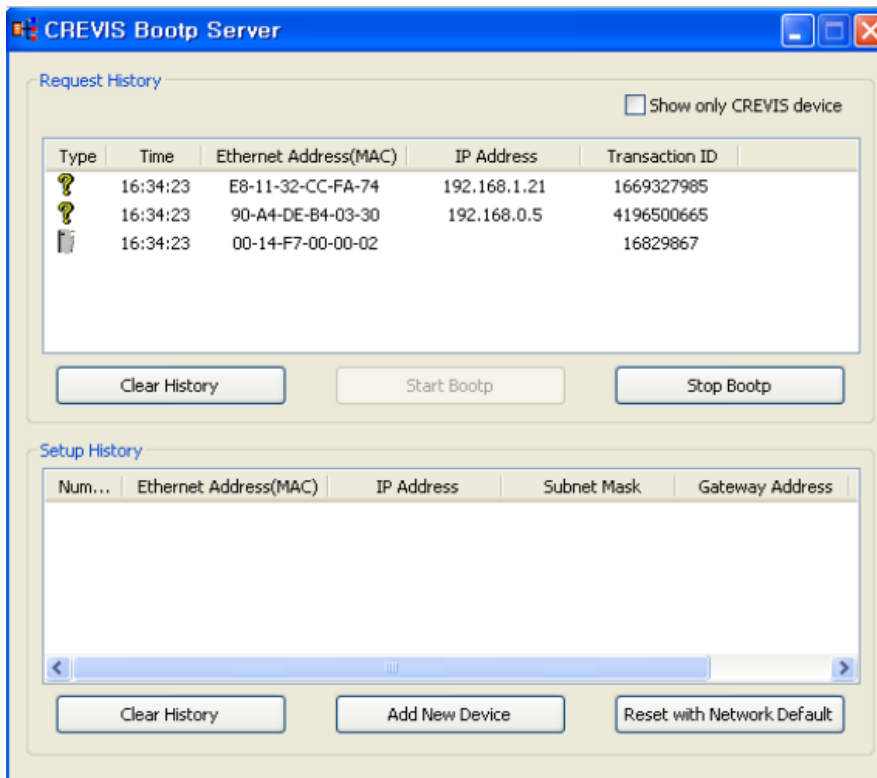
## 6.4. MODBUS/TCP IP – Address Setup

### 6.4.1 IP-Address Setup using BOOTP/DHCP Sever

If the adapter BOOTP/DHCP enabled (DIP Pole#9 ON), the adapter sends BOOTP/DHCP request message of 20 times every 2sec. If BOOTP/DHCP sever does not response, the Adapter applies its IP Address with EEPROM (Latest saved IP Address).

The following is an example of adapter IP-Address setup that can be used with a third party BOOTP/DHCP server.

\* CREVIS IO Guide Pro' s BOOTP server





## 6.4.2 IP-Address Setup using DIP switch(Manual function)

If the adapter DIP Pole#10 is ON, lowest IP address is set by DIP Pole#1~#8 manually. Refer to 2.4.2.

These are examples of adapter IP-Address setup by manual function.

Ex) XXX. XXX. XXX. 1



Ex) XXX. XXX. XXX. 2



Ex) XXX. XXX. XXX. 8



Ex) XXX. XXX. XXX. 253



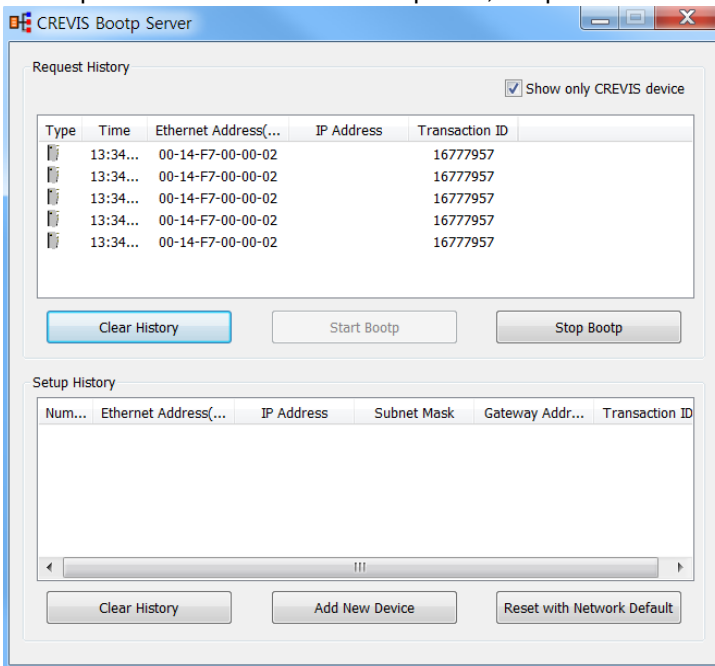
• **IP-Address Setup using BOOTP/DHCP Server**

The following is an example of adapter IP-Address setup using CREVIS IO Guide Pro's BOOTP server.

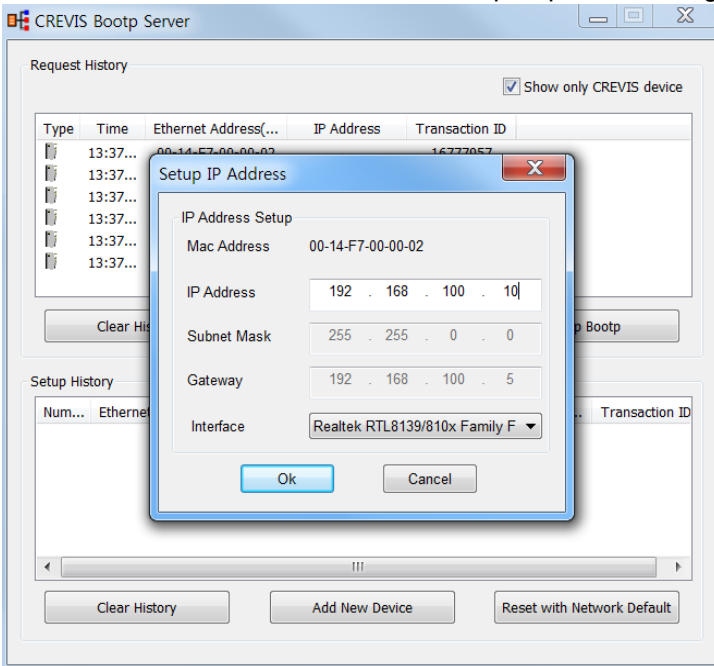
Click 'Start Bootp' to enable Bootp server to detect Bootp request on the network.

If the adapter is Bootp/DHCP enabled (Dip Pole #9 ON), adapter will send Bootp/DHCP request 20times every 5seconds, after power reset.

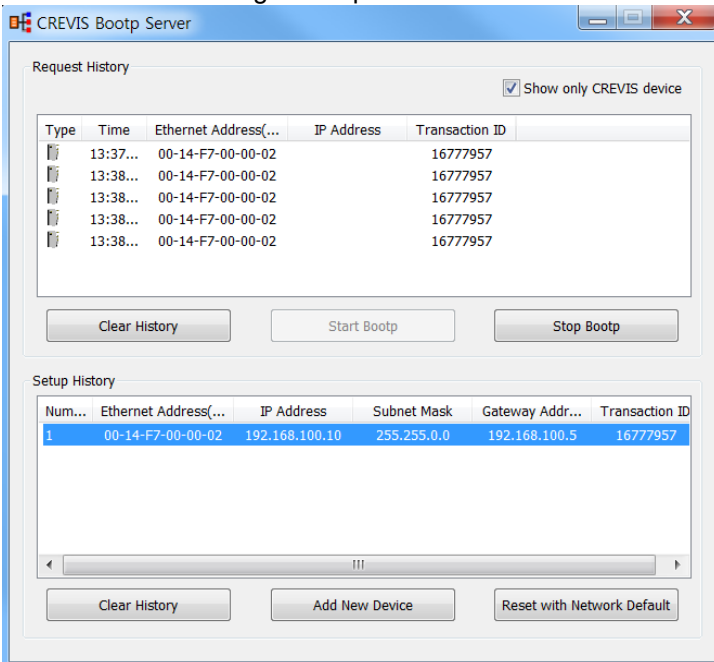
If Bootp/DHCP server does not response, adapter will use the latest saved IO address in EEPROM.



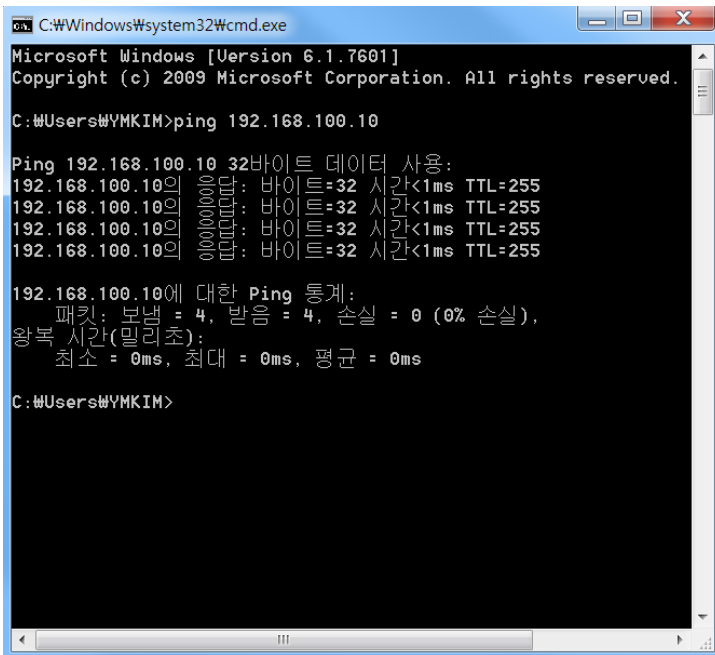
Double click on one of the detected 'Bootp request message' and set the IP address.



The IP address setting is completed.

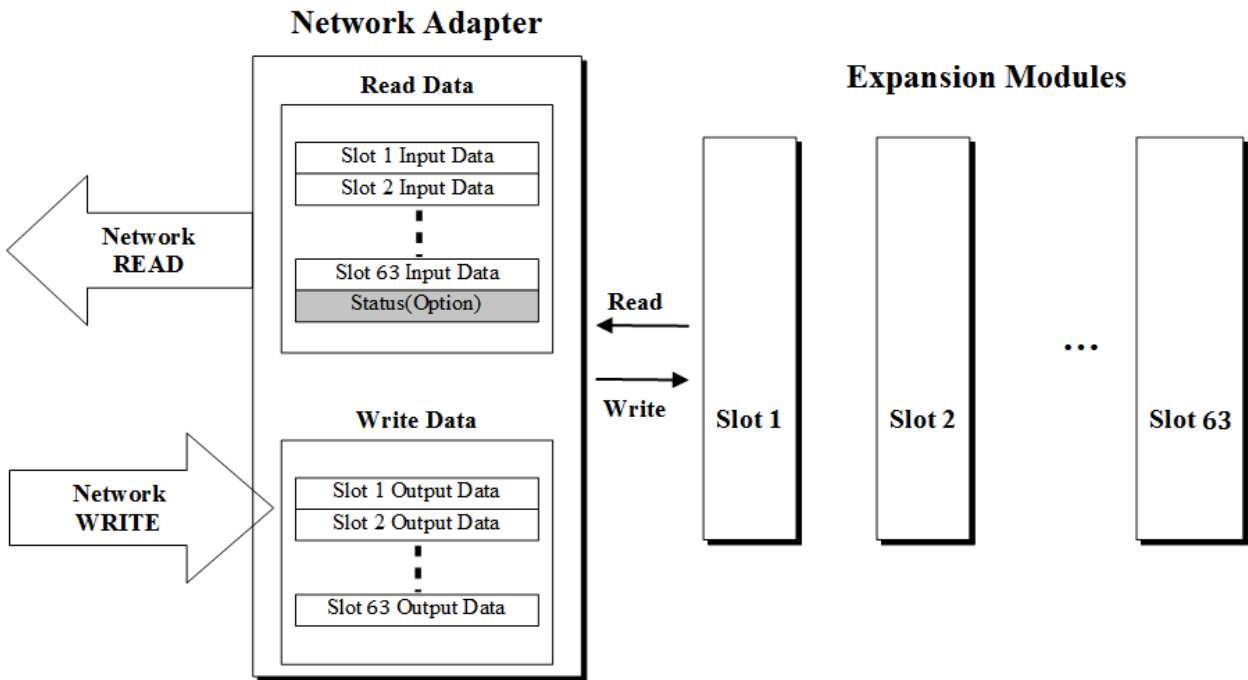


Check IP address of the normal.



### 7. I/O Process Image Map

An expansion module may have 3 types of data as I/O data, configuration parameter and memory register. The data exchange between network adapter and expansion modules is done via an I/O process image data by G-Series protocol. The following figure shows the data flow of process image between network adapter and expansion modules.



## 7.1. MODBUS Interface Register/Bit Map

### Register Map

Start Address	Read/Write	Description	Func. Code
0x0000 ~	Read	Process input image registers (Real Input Register)	3,4,23
0x0800 ~	Read/Write	Process output image registers (Real Output Register)	3,16,23
0x1000 *	Read	Adapter Identification special registers.	3,4,23
0x1020 *	Read/Write	Adapter Watchdog, other time special register.	3,4,6,16,23
0x1100 *	Read/Write	Adapter Information special registers.	3,4,6,16,23
0x2000 *	Read/Write	Expansion Slot Information special registers.	3,4,6,16,23

\* The special register map must be accessed by read/write of every each address (one address).

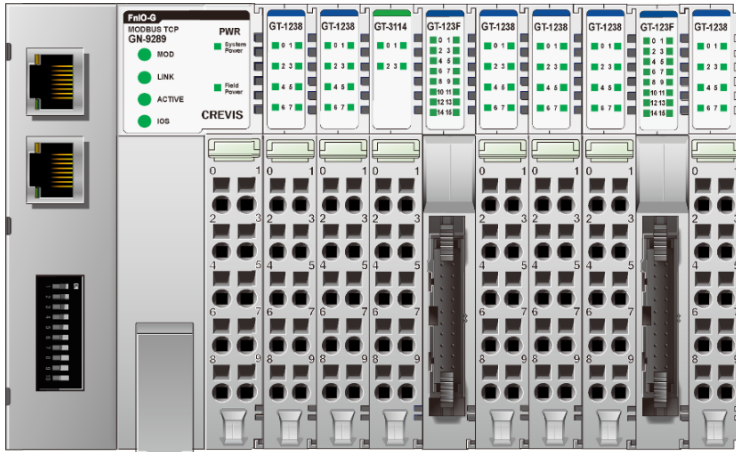
### Register Map

Start Address	Read/Write	Description	Func. Code
0x0000~	Read	Process input image bits All input registers area are addressable by bit address. Size of input image bit is size of input image register * 16.	2
0x1000~	Read/Write	Process output image bits All output registers area are addressable by bit address. Size of output image bit is size of output image register * 16.	1,5,15

## 7.2. Example of Input Process Image (Input Register) Map

Input image data depends on slot position and expansion slot data type. Input process image data is only ordered by expansion slot position

\* For example slot configuration



Slot No.	Module Description
#0	MODBUS/TCP Adapter
#1	8-discrete output
#2	8-discrete output
#3	4-analog output
#4	16-discrete input
#5	8-discrete input
#6	8-discrete input
#7	8-discrete input
#8	16-discrete input
#9	8-discrete input

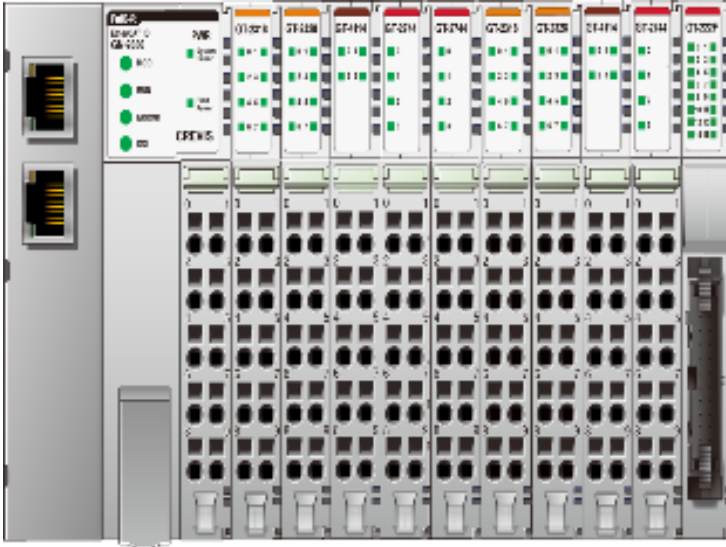
### Input Process Image

Address	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0x0001	Discrete Input 8 pts (Slot#2)								Discrete Input 8 pts (Slot#1)							
0x0002	Analog Input Ch0 high byte (Slot#3)								Analog Input Ch0 low byte (Slot#3)							
0x0003	Analog Input Ch1 high byte (Slot#3)								Analog Input Ch1 low byte (Slot#3)							
0x0004	Analog Input Ch2 high byte (Slot#3)								Analog Input Ch2 low byte (Slot#3)							
0x0005	Analog Input Ch3 high byte (Slot#3)								Analog Input Ch3 low byte (Slot#3)							
0x0006	Discrete Input 8 pts (Slot#4)								Discrete Input 8 pts (Slot#4)							
0x0007	Discrete Input 8 pts (Slot#6)								Discrete Input 8 pts (Slot#5)							
0x0008	Discrete Input 8 pts (Slot#8)								Discrete Input 8 pts (Slot#7)							
0x0009	Discrete Input 8 pts (Slot#9)								Discrete Input 8 pts (Slot#8)							

### 7.3. Example of Output Process Image (Output Register) Map

Output image data depends on slot position and expansion slot data type. Output process image data is only ordered by expansion slot position.

\* For example slot configuration



Slot No.	Module Description
#0	MODBUS/TCP Adapter
#1	8-discrete output
#2	8-discrete output
#3	4-analog output
#4	4- relay output
#5	4-relay output
#6	8-discrete output
#7	8-discrete output
#8	4-analog output
#9	4-relay output
#10	16-discrete output

#### Output Process Image

Address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0800	Discrete Output 8 pts (Slot#2)								Discrete Output 8 pts (Slot#1)							
0x0801	Analog Output Ch0 high byte (Slot#3)								Analog Output Ch0 low byte (Slot#3)							
0x0802	Analog Output Ch1 high byte (Slot#3)								Analog Output Ch1 low byte (Slot#3)							
0x0803	Analog Output Ch2 high byte (Slot#3)								Analog Output Ch2 low byte (Slot#3)							
0x0804	Analog Output Ch3 high byte (Slot#3)								Analog Output Ch3 low byte (Slot#3)							
0x0805	Empty, Don't Care				Discrete Out 4 pts (Slot#5)				Empty, Don't Care				Discrete Out 4 pts (Slot#4)			
0x0806	Discrete Output low 8 pts (Slot#7)								Discrete Output low 8 pts (Slot#6)							
0x0807	Analog Output Ch0 high byte (Slot#8)								Analog Output Ch0 low byte (Slot#8)							
0x0808	Analog Output Ch1 high byte (Slot#8)								Analog Output Ch1 low byte (Slot#8)							
0x0809	Analog Output Ch2 high byte (Slot#8)								Analog Output Ch2 low byte (Slot#8)							
0x080A	Analog Output Ch3 high byte (Slot#8)								Analog Output Ch3 low byte (Slot#8)							
0x080B	Discrete Output low 8 pts (Slot#10)								Empty, Don't Care				Discrete Out 4 pts (Slot#9)			
0x080C	Empty, Don't Care								Discrete Output high 8 pts (Slot#10)							



## 8. MODBUS TCP/ UDP INTERFACE

### 8.1. MODBUS TCP/ UDP Protocol

The MODBUS messaging service provides a Client/Server communication between devices connected on an Ethernet TCP/IP network. All MODBUS/TCP messages are sent via TCP on registered port 502. Refer to Modbus\_Messaging\_Implementation\_Guide\_V1\_0a.pdf.

#### 8.1.1. Comparison of MODBUS TCP/ UDP And MODBUS/RTU

This header provides some differences compared to the MODBUS RTU application data unit used on serial line:

- The MODBUS 'slave address' field usually used on MODBUS Serial Line is replaced by a single byte 'Unit Identifier' within the MBAP Header. The 'Unit Identifier' is used to communicate via devices such as bridges, routers and gateways that use a single IP address to support multiple independent MODBUS end units.
- All MODBUS requests and responses are designed in such a way that the recipient can verify that a message is finished. For function codes where the MODBUS PDU has a fixed length, the function code alone is sufficient. For function codes carrying a variable amount of data in the request or response, the data field includes a byte count.
- When MODBUS is carried over TCP, additional length information is carried in the MBAP header to allow the recipient to recognize message boundaries even if the message has been split into multiple packets for transmission. The existence of explicit and implicit length rules, and use of a CRC-32 error check code (on Ethernet) results in an infinitesimal chance of undetected corruption to a request or response message.

#### MODBUS TCP/ UDP

MBAP Header		Function	Data		
7 chars		1 char	Up to 252 chars		
MODBUS/ RTU		Function	Data	CRC Check	End
Start	Address	Function	Data	CRC Check	End
≥ 3.5 char	1 char	1 char	Up to 252 chars	2 chars	≥ 3.5 char

Function and data field of MODBUS/TCP are identical to function and data field of MODBUS/RTU.

### 8.1.2. MODBUS TCP/ UDP MBAP Header

The MBAP (MODBUS Application Protocol) header contains the following fields.

Fields	Length	Description	Client	Server
Transaction Identifier	2bytes	Identification of a MODBUS Request /Response transaction.	Initialized by the client	Recopied by the server from the received
Protocol Identifier	2bytes	0 = MODBUS protocol	Initialized by the client	Recopied by the server from the received
Length	2bytes	Number of following bytes	Initialized by the client (Request)	Initialized by the server (Response)
Unit Identifier	1byte	Identification of a remote slave connected on a serial line or on other buses.	Initialized by the client	Recopied by the server from the received

Transaction Identifier - It is used for transaction pairing, the MODBUS server copies in the response the transaction identifier of the request.

- Protocol Identifier – It is used for intra-system multiplexing. The MODBUS protocol is identified by the value 0.
- Length - The length field is a byte count of the following fields, including the Unit Identifier and data fields.
- Unit Identifier – This field is used for intra-system routing purpose. Typically MODBUS server must be returned with the same value set by MODBUS client.

## 8.2. Supported MODBUS Function Codes

Function Code	Function	Description
1(0x01)	Read Coils	Read output bit
2(0x02)	Read Discrete Inputs	Read input bit
3(0x03)	Read Holding Registers	Read output word
4(0x04)	Read Input Registers	Read input word
5(0x05)	Write Single Coil	Write one bit output
6(0x06)	Write Single Register	Write one word output
8(0x08)	Diagnostics	Read diagnostic register
15(0x0F)	Write Multiple Coils	Write a number of output bits
16(0x10)	Write Multiple registers	Write a number of output words
23(0x17)	Read/Write Multiple registers	Read a number of input words /Write a number of output words

- Refer to MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1a

### 8.2.1. 1 (0x01) Read Coils

This function code is used to read from 1 to 2000 contiguous status of coils in a remote device. The Request PDU specifies the starting address, i.e. the address of the first coil specified, and the number of coils. In the PDU Coils are addressed starting at zero. Therefore coils numbered 1-16 are addressed as 0-15. The coils in the response message are packed as one coil per bit of the data field. Status is indicated as 1= ON and 0= OFF.

#### Request

Field name	Example
Function Code	0x01
Starting Address Hi	0x10
Starting Address Lo	0x00
Quantity of Outputs Hi	0x00
Quantity of Outputs Lo	0x0A

#### Response

Field name	Example
Function Code	0x01
Byte Count	0x02
Output Status	0x55
Output Status	0x02

In case of address 0x1015~0x1000 output bit value: 10101010\_01010101.

### 8.2.2. 2 (0x02) Read Discrete Inputs

This function code is used to read from 1 to 2000 contiguous status of discrete inputs in a remote device. The Request PDU specifies the starting address, i.e. the address of the first input specified, and the number of inputs. In the PDU Discrete Inputs are addressed starting at zero. Therefore Discrete inputs numbered 1-16 are addressed as 0-15.

The discrete inputs in the response message are packed as one input per bit of the data field.

Status is indicated as 1= ON; 0= OFF.

#### Request

Field name	Example
Function Code	0x02
Starting Address Hi	0x00
Starting Address Lo	0x00
Quantity of Inputs Hi	0x00
Quantity of Inputs Lo	0x0A

#### Response

Field name	Example
Function Code	0x02
Byte Count	0x02
Input Status	0x80
Input Status	0x00

- In case of address 0x0015~0x0000 input bit value: 00000000\_10000000.

### 8.2.3. 3 (0x03) Read Holding Resgisters

This function code is used to read the contents of a contiguous block of holding registers in a remote device. The Request PDU specifies the starting register address and the number of registers.

The register data in the response message are packed as two bytes per register, with the binary contents right justified within each byte. For each register, the first byte contains the high order bits and the second contains the low order bits.

#### Request

Field name	Example
Function Code	0x03
Starting Address Hi	0x08
Starting Address Lo	0x00
Quantity of Register Hi	0x00
Quantity of Register Lo	0x02

#### Response

Field name	Example
Function Code	0x03
Byte Count	0x04
Output Register#0 Hi	0x11
Output Register#0 Lo	0x22
Output Register#1 Hi	0x33
Output Register#1 Lo	0x44

- In case of address 0x0800, 0x0801 output register value: 0x1122, 0x3344.

#### 8.2.4. 4 (0x04) Read Input Resgisters

This function code is used to read from 1 to approx. 125 contiguous input registers in a remote device. The Request PDU specifies the starting register address and the number of registers. The register data in the response message are packed as two bytes per register, with the binary contents right justified within each byte. For each register, the first byte contains the high order bits and the second contains the low order bits.

##### Request

Field name	Example
Function Code	0x04
Starting Address Hi	0x00
Starting Address Lo	0x00
Quantity of Register Hi	0x00
Quantity of Register Lo	0x02

##### Response

Field name	Example
Function Code	0x03
Byte Count	0x04
Input Register#0 Hi	0x00
Input Register#0 Lo	0x80
Input Register#1 Hi	0x00
Input Register#1 Lo	0x00

-In case of address 0x0000, 0x0001 input register value: 0x0080, 0x0000.

### 8.2.5. 5 (0x05) Write Single Coil

This function code is used to write a single output to either ON or OFF in a remote device. The requested ON/OFF state is specified by a constant in the request data field. A value of FF 00 hex requests the output to be ON. A value of 00 00 requests it to be OFF. All other values are illegal and will not affect the output.

#### Request

Field name	Example
Function Code	0x05
Output Address Hi	0x10
Output Address Lo	0x01
Output Value Hi	0xFF
Output Value Lo	0x00

#### Response

Field name	Example
Function Code	0x05
Output Address Hi	0x10
Output Address Lo	0x01
Output Value Hi	0xFF
Output Value Lo	0x00

- Output bit of address 0x1001 turns ON.

### 8.2.6. 6 (0x06) Write Single Register

This function code is used to write a single holding register in a remote device. Therefore register numbered 1 is addressed as 0. The normal response is an echo of the request, returned after the register contents have been written.

#### Request

Field name	Example
Function Code	0x06
Register Address Hi	0x08
Register Address Lo	0x00
Register Value Hi	0x11
Register Value Lo	0x22

#### Response

Field name	Example
Function Code	0x06
Register Address Hi	0x08
Register Address Lo	0x00
Register Value Hi	0x11
Register Value Lo	0x22

- In case of address 0x0800 output register value: 0x0000 changes to 0x1122.



### 8.2.7. 8 (0x08) Diagnostics

MODBUS function code 08 provides a series of tests for checking the communication system between a client ( Master) device and a server ( Slave), or for checking various internal error conditions within a server. The function uses a two–byte sub-function code field in the query to define the type of test to be performed. The server echoes both the function code and sub-function code in a normal response. Some of the diagnostics cause data to be returned from the remote device in the data field of a normal response.

#### Request

Field name	Example
Function Code	0x08
Sub-Function Hi	0x00
Sub-Function Lo	0x00
Data Hi	0x11
Data Lo	0x22

#### Response

Field name	Example
Function Code	0x08
Sub-Function Hi	0x00
Sub-Function Lo	0x00
Data Hi	0x11
Data Lo	0x22

#### Sub-function 0x0000(0) Return Query Data

The data passed in the request data field is to be returned (looped back) in the response. The entire response message should be identical to the request.

Sub-function	Data Field (Request)	Data Field (Response)	Description
0x0000(0)	Any	Echo Request Data	

#### Sub-function 0x0001(1) Restart Communications Option

The remote device could be initialized and restarted, and all of its communications event counters are cleared.

Especially, data field 0x55AA make the remote device to restart with factory default setup of EEPROM.

Sub-function	Data Field (Request)	Data Field (Response)	Description
0x0001(1)	0x0000 or 0xFF00	Echo Request Data	Reset
0x0001(1)	0x55AA+0xAB7B+Sumcheck	Echo Request Data	Reset with Factory default <sup>1)</sup>
0x0001(1)	0x55AA+0xAA55+Sumcheck	Echo Request Data	Reset with Factory default <sup>2)</sup>

1),2) IP Address, Subnet Mask Address, Gateway Address will be the factory defaults value.

2) Mac Address will be the factory default value.

**Sub-function 0x000A(10) Clear Counters and Diagnostic Register**

The goal is to clear all counters and the diagnostic register. Counters are also cleared upon power-up.

Sub-function	Data Field (Request)	Data Field (Response)	Description
0x000A(10)	0x0000	Echo Request Data	

**Sub-function 0x000B(11) Return Bus Message Count**

The response data field returns the quantity of messages that the remote device has detected on the communications system since its last restart, clear counters operation, or power-up.

Sub-function	Data Field (Request)	Data Field (Response)	Description
0x000B(11)	0x0000	Total Message Count	

**Sub-function 0x000C(12) Return Bus Communication Error Count**

The response data field returns the quantity of CRC errors encountered by the remote device since its last restart, clear counters operation, or power-up.

Sub-function	Data Field (Request)	Data Field (Response)	Description
0x000C(12)	0x0000	CRC Error Count	

**Sub-function 0x000D(13) Return Bus Exception Error Count**

The response data field returns the quantity of MODBUS exception responses returned by the remote device since its last restart, clear counters operation, or power-up.

Exception responses are described and listed in section 3.2.11.

Sub-function	Data Field (Request)	Data Field (Response)	Description
0x000D(13)	0x0000	Exception Error Count	

**Sub-function 0x000E(14) Return Slave Message Count**

The response data field returns the quantity of messages addressed to the remote device, or broadcast, that the remote device has processed since its last restart, clear counters operation, or power-up.

Sub-function	Data Field (Request)	Data Field (Response)	Description
0x000E(14)	0x0000	Slave Message Count	

**Sub-function 0x000F(15) Return Slave No Response Count**

The response data field returns the quantity of messages addressed to the remote device for which it has returned no response (neither a normal response nor an exception response), since its last restart, clear counters operation, or power-up.

Sub-function	Data Field (Request)	Data Field (Response)	Description
0x000F(15)	0x0000	Slave No Response Count	

**Sub-function 0x0064(100) Return Slave ModBus, Internal Bus Status**

The response data field returns the status of ModBus and Internal Bus addressed to the remote device. This status values are identical with status 1word of input process image.

Sub-function	Data Field (Request)	Data Field (Response)	Description
0x0064(100)	0x0000	ModBus, Internal Bus Status	Same as status 1word

**Sub-function 0x0065(101) Return Slave Watchdog Error Count**

The response data field returns the quantity of watchdog error addressed to the remote device since its last restart, clear counters operation, or power-up.

Sub-function	Data Field (Request)	Data Field (Response)	Description
0x0065(101)	0x0000	Watchdog Error Count	

**8.2.8. 15 (0x0F) Write Multiple Coils**

This function code is used to force each coil in a sequence of coils to either ON or OFF in a remote device. The Request PDU specifies the coil references to be forced. Coils are addressed starting at zero. A logical '1' in a bit position of the field requests the corresponding output to be ON. A logical '0' requests it to be OFF. The normal response returns the function code, starting address, and quantity of coils forced.

**Request**

Field name	Example
Function Code	0x0F
Starting Address Hi	0x10
Starting Address Lo	0x00
Quantity of Outputs Hi	0x00
Quantity of Outputs Lo	0x0A
Byte Count	0x02
Output Value#0	0x55
Output Value#1	0x01

**Response**

Field name	Example
Function Code	0x0F
Starting Address Hi	0x10
Starting Address Lo	0x00
Quantity of Outputs Hi	0x00
Quantity of Outputs Lo	0x0A

In case of address 0x1015~0x1000 output bit value: 00000000\_00000000 changes to 00000001\_01010101.

### 8.2.9. 16 (0x10) Write Multiple Resgisters

This function code is used to write a block of contiguous registers (1 to approx. 120 registers) in a remote device.

The requested written values are specified in the request data field. Data is packed as two bytes per register. The normal response returns the function code, starting address, and quantity of registers written.

#### Request

Field name	Example
Function Code	0x0F
Starting Address Hi	0x10
Starting Address Lo	0x08
Quantity of Registers Hi	0x00
Quantity of Registers Lo	0x02
Byte Count	0x04
Register Value#0 Hi	0x11
Register Value#0 Lo	0x22
Register Value#1 Hi	0x33
Register Value#1 Lo	0x44

#### Response

Field name	Example
Function Code	0x0F
Starting Address Hi	0x10
Starting Address Lo	0x08
Quantity of Registers Hi	0x00
Quantity of Registers Lo	0x02

.- In case of address 0x0800, 0x0801 output register value: 0x0000, 0x0000 changes to 0x1122, 0x3344.

### 8.2.10 23 (0x17) Read/Write Multiple Resgisters

This function code performs a combination of one read operation and one write operation in a single MODBUS transaction. The write operation is performed before the read. The request specifies the starting address and number of holding registers to be read as well as the starting address, number of holding registers, and the data to be written. The byte count specifies the number of bytes to follow in the write data field.

The normal response contains the data from the group of registers that were read. The byte count field specifies the quantity of bytes to follow in the read data field.

#### Request

Field name	Example
Function Code	0x17
Read Starting Address Hi	0x08
Read Starting Address Lo	0x00
Quantity of Read Hi	0x00
Quantity of Read Lo	0x02
Write Starting Address Hi	0x08
Write Starting Address Lo	0x00
Quantity of Write Hi	0x00
Quantity of Write Lo	0x02
Byte Count	0x04
Write Reg. Value#0 Hi	0x11
Write Reg. Value#0 Lo	0x22
Write Reg. Value#1 Hi	0x33
Write Reg. Value#1 Lo	0x44

#### Response

Field name	Example
Function Code	0x17
Byte Count	0x04
Read Reg. Value#0 Hi	0x11
Read Reg. Value#0 Lo	0x22
Read Reg. Value#1 Hi	0x33
Read Reg. Value#1 Lo	0x44

In case of address 0x0800, 0x0801 output register value: 0x0000, 0x0000 changes to 0x1122, 0x3344.

### 8.2.11. Error Response

In an exception response, the server sets the MSB of the function code to 1. This makes the function code value in an exception response exactly 80 hexadecimal higher than the value would be for a normal response.

#### Exception Response Example

Field name	Example
Function Code	0x81
Exception Code	0x02

#### Exception Codes

Exception Code	Name	Description
01	Illegal Function	The function code received in the query is not an allowable action for the server (or slave).
02	Illegal Data Address	The data address received in the query is not an allowable address for the server (or slave).
03	Illegal Data Value	A value contained in the query data field is not an allowable value for server (or slave).
04	Slave Device Failure	An unrecoverable error occurred while the server (or slave) was attempting to perform the requested action.
05	Acknowledge	The server (or slave) has accepted the request and is processing it, but a long duration of time will be required to do so.
06	Slave Device Busy	Specialized use in conjunction with programming commands. The server (or slave) is engaged in processing a long-duration program command. The client (or master) should retransmit the message later when the server (or slave) is free.
08	Memory Parity Error	The server (or slave) attempted to read record file, but detected a parity error in the memory. The client (or master) can retry the request, but service may be required on the server (or slave) device.
0A	Gateway Path Unavailable	Specialized use in conjunction with gateways, indicates that the gateway was unable to allocate an internal communication path from the input port to the output port for processing the request.

- GN-9289 response exception code 01, 02, 03, 04 and 06.

### 8.3. MODBUS Special Register Map

The special register map can be accessed by function code 3, 4, 6 and 16. Also the special register map must be accessed by read/write of every each address (one address).

#### 8.3.1. Adapter Identification Special Resgister (0x1000, 4096)

Address	Access	Type, Size	Description
0x1000(4096)	Read	1word	Vendor ID = 0x02E5(741), Crevis. Co., Ltd.
0x1001(4097)	Read	1word	Device type = 0x000C, Network Adapter
0x1002(4098)	Read	1word	Product Code = 0x9000
0x1003(4099)	Read	1word	Firmware revision, if 0x0101, revision 1.01
0x1004(4100)	Read	2word	Product unique serial number
0x1005(4101)	Read	String upto 34byte	Product name string (ASCII) "GN-9289,Modbus/TCP Adapter,GBUS"
0x1006(4102)	Read	1word	Sum check of EEPROM
0x1010(4112)	Read	2word	Firmware release date
0x1011(4113)	Read	2word	Product manufacturing inspection date
0x101E(4126)	Read	7word - 1word - 1word - 1word - 1word - 1word - 2word  15word - 2word - 2word - 2word - 3word - 1word - 1word - 1word - 1word - 2word	Composite Id of following address * RTU mode 0x1100(4352), Modbus RS232 Node. (Fixed 0x0001) 0x1000(4096), Vendor ID 0x1001(4097), Device type 0x1002(4098), Product code 0x1003(4099), Firmware revision 0x1004(4100), Product serial number  *TCP mode 0x1050(4176), IP address 0x1051(4177), Subnet mask 0x1052(4178), Gateway 0x1053(4179), Ethernet physical address (MAC ID) 0x1000(4096), Vendor ID 0x1001(4097), Device type 0x1002(4098), Product code 0x1003(4099), Firmware revision 0x1004(4100), Product serial number

- String Type consists of valid string length (first 1word) and array of characters

### 8.3.2. Adapter Watchdog Time, other Time Special Register (0x1020, 4128)

A watchdog timer can be configured for timeout periods up to 65535(1unit=100msec). The Watchdog timer will timeout (timer decreased, reached 0) if ModBus operation to the slave node does not occur over the configured watchdog value, then the slave adapter forces that slot output value is automatically set to user-configured fault actions and values.

Address	Access	Type, Size	Description
0x1020(4128)	Read/Write	1word	Watchdog time value 16bit unsigned. The time value is represented by multiples of 100msec. The 0 (watchdog timeout disabled) is default value. A changing of watchdog time value resets watchdog error and counter.
0x1021(4129)	Read	1word	Watchdog timer remain value This value is decreased every 100msec
0x1022(4130)	Read	1word	Watchdog error counter, it is cleared by writing address 0x1020
0x1023(4131)	Read/Write	1word	Enable/disable auto recovery Watchdog error when receiving new frame. 0:Disable, 1:Enable(default). Its value is stored in EEPROM.
0x1028(4136)	Read	1word	IO update time, main loop time. (1usec unit)

### 8.3.3. Adapter TCP/IP Special Register (0x1040, 4160)

Address	Access	Type, Size	Description
0x1040(4160)	Read	1word	Reserved
0x1041(4161)	Read/Write	1word	MODBUS/TCP connection timeout time. (0.5sec unit) Maximum time of ModBus connection to stay to be opened without receiving a ModBus request. 0~3600 The 120 (60sec) is default value. The value 0 disables connection time out specially.
0x1042(4162)	Read	1word	Number of ModBus/TCP connected
0x1043(4163)	Read	1word	ModBus/TCP port, fixed 502
0x1044(4164)	Read	1word	Ethernet Interface Speed, 10(10Mbps) or 100(100Mbps)
0x1045(4165) *	Read/Write	1word	IP Setting Method. 0: BOOTP, 1:DHCP
0x1046(4166)	---	---	Reserved.
0x1047(4167)	Read	1word	Status of DIP SW#9 DHCP/BOOTP(Enable/Disable). 0 : OFF, 1 : ON
0x1048(4168)	Read	1word	Enable/disable Lowest IP address via DIP Switch, 1:Enabled
0x1050(4176)	Read/Write	2word	IP address. If 192.168.123.1, then 0xA8C0, 0x017B. After update this value, IP address, Subnet mask and Gateway are applied as new one.
0x1051(4177)	Read/Write	2word	Subnet mask. If 255.255.255.0, then 0xFFFF, 0x00FF.
0x1052(4178)	Read/Write	2word	Gateway. If 192.168.123.254, then 0xA8C0, 0xFE7B.



0x1053(4179)	Read	3word	Ethernet physical address (MAC-ID). If 11-22-33-44-55-66, then 0x2211, 0x4433, 0x6655.
--------------	------	-------	---

\* Power off and then power on, this value is applied.

### 8.3.4. Adapter Information Special Register (0x1100, 4352)

Address	Access	Type, Size	Description																	
0x1100(4352)*	Read/Write	1word	Master fault action option. (Disable : 0x0000, Enable : 0x0001) This option can enable Master fault action option. With master fault action, fault action can be activated with master communication failure. Default is disable.																	
0x1102(4354)	Read	1word	Start address of input image word register. =0x0000																	
0x1103(4355)	Read	1word	Start address of output image word register. =0x0800																	
0x1104(4356)	Read	1word	Size of input image word register.																	
0x1105(4357)	Read	1word	Size of output image word register.																	
0x1106(4358)	Read	1word	Start address of input image bit. = 0x0000																	
0x1107(4359)	Read	1word	Start address of output image bit. =0x1000																	
0x1108(4360)	Read	1word	Size of input image bit.																	
0x1109(4361)	Read	1word	Size of output image bit.																	
0x110A(4362)	Read	1word	Update time for cyclic data change (same as 0x1028)																	
0x110D(4365)	Read	1word	Current Dip Switch State and Field Power Status (MSB) ex) DHCP/Booth enable, Dip SW(0x01), Field Power On = 0x8101																	
0x110E(4366)	Read	upto 33word	Expansion slot's GT-number including GN First 1word is adapter's number, if GN-9289, then 0x9289																	
0x1110(4368)	Read	1word	Number of expansion slot																	
0x1113(4371)	Read	upto 33word	Expansion slot Module Id. First 1word is adapter's module id.																	
0x1119(4377)	Read	1word	Hi byte is ModBus status, low byte is internal bus status. Zero value means 'no error'.																	
			<table border="1"> <thead> <tr> <th>ModBus status</th> <th>Internal bus status(G-Bus)</th> </tr> </thead> <tbody> <tr> <td>0x00 : No Error</td> <td>0x00 : OPERATING</td> </tr> <tr> <td>0x01 : ERR_DIP_SWITCH</td> <td>0x02 : CONNECT_FAULT</td> </tr> <tr> <td>0x40 : ERR_CRC_LRC</td> <td>0x03 : CONFIG_FAULT</td> </tr> <tr> <td>0x80 : ERR_WATCHDOG</td> <td>0x04 : NO_EXPANSION</td> </tr> <tr> <td></td> <td>0x05 : INVALID_ATTR_VALUE</td> </tr> <tr> <td></td> <td>0x06 : TOO_MUCH_DATA</td> </tr> <tr> <td></td> <td>0x07 : VENDOR_ERROR</td> </tr> <tr> <td></td> <td>0x08 : NOT_EXPECTED_SLOT</td> </tr> <tr> <td></td> <td>0x09 : CRC_ERROR</td> </tr> </tbody> </table>	ModBus status	Internal bus status(G-Bus)	0x00 : No Error	0x00 : OPERATING	0x01 : ERR_DIP_SWITCH	0x02 : CONNECT_FAULT	0x40 : ERR_CRC_LRC	0x03 : CONFIG_FAULT	0x80 : ERR_WATCHDOG	0x04 : NO_EXPANSION		0x05 : INVALID_ATTR_VALUE		0x06 : TOO_MUCH_DATA		0x07 : VENDOR_ERROR	
ModBus status	Internal bus status(G-Bus)																			
0x00 : No Error	0x00 : OPERATING																			
0x01 : ERR_DIP_SWITCH	0x02 : CONNECT_FAULT																			
0x40 : ERR_CRC_LRC	0x03 : CONFIG_FAULT																			
0x80 : ERR_WATCHDOG	0x04 : NO_EXPANSION																			
	0x05 : INVALID_ATTR_VALUE																			
	0x06 : TOO_MUCH_DATA																			
	0x07 : VENDOR_ERROR																			
	0x08 : NOT_EXPECTED_SLOT																			
	0x09 : CRC_ERROR																			
0x111D(4381)	Read	1word	Adapter G-Series Revision.																	

\*After the system is reset, the new "Set Value" action is applied.

\*\* If the slot location is changed, set default value automatically (all expansion slot are live).

### 8.3.5. Expansion Slot Information Special Resister (0x2000, 8192)

Each expansion slot has 0x20(32) address offset and same information structure.

Slot#1 0x2000(8192)~0x201F(8223)	Slot#2 0x2020(8224)~0x203F(8255)
Slot#3 0x2040(8256)~0x205F(8287)	Slot#4 0x2060(8288)~0x207F(8319)
Slot#5 0x2080(8320)~0x209F(8351)	Slot#6 0x20A0(8352)~0x20BF(8383)
Slot#7 0x20C0(8384)~0x20DF(8415)	Slot#8 0x20E0(8416)~0x20FF(8447)
Slot#9 0x2100(8448)~0x211F(8479)	Slot#10 0x2120(8480)~0x213F(8511)
Slot#11 0x2140(8512)~0x215F(8543)	Slot#12 0x2160(8544)~0x217F(8575)
Slot#13 0x2180(8576)~0x219F(8607)	Slot#14 0x21A0(8608)~0x21BF(8639)
Slot#15 0x21C0(8640)~0x21DF(8671)	Slot#16 0x21E0(8672)~0x21FF(8703)
Slot#17 0x2200(8704)~0x221F(8735)	Slot#18 0x2220(8736)~0x223F(8767)
Slot#19 0x2240(8768)~0x225F(8799)	Slot#20 0x2260(8800)~0x227F(8831)
Slot#21 0x2280(8832)~0x229F(8863)	Slot#22 0x22A0(8864)~0x22BF(8895)
Slot#23 0x22C0(8896)~0x22DF(8927)	Slot#24 0x22E0(8928)~0x22FF(8959)
Slot#25 0x2300(8960)~0x231F(8991)	Slot#26 0x2320(8992)~0x233F(9023)
Slot#27 0x2340(9024)~0x235F(9055)	Slot#28 0x2360(9056)~0x237F(9087)
Slot#29 0x2380(9088)~0x239F(9119)	Slot#30 0x23A0(9120)~0x23BF(9151)
Slot#31 0x23C0(9152)~0x23DF(9183)	Slot#32 0x23E0(9184)~0x23FF(9215)
Slot#33 0x2400(9216)~0x241F(9247)	Slot#34 0x2420(9248)~0x243F(9279)
.....	
Slot#63 0x27C0(10176)~0x27DF(10207)	

Address Offset	Expansion Slot#1	Expansion Slot#2	Expansion Slot#3	Expansion Slot#4	...	Expansion Slot#63
+ 0x00(+0)	0x2000(8192)	0x2020(8224)	0x2040(8256)	0x2060(8288)	...	0x27C0(10176)
+ 0x01(+1)	0x2001(8193)	0x2021(8225)	0x2041(8257)	0x2061(8289)	...	0x27C1(10177)

					....	
+ 0x02(+2)	0x2002(8194)	0x2022(8226)	0x2042(8258)	0x2062(8290)	... ....	0x27C2(10178)
+ 0x03(+3)	0x2003(8195)	0x2023(8227)	0x2043(8259)	0x2063(8291)	... ....	0x27C3(10179)
+ 0x04(+4)	0x2004(8196)	0x2024(8228)	0x2044(8260)	0x2064(8292)	... ....	0x27C4(10180)
+ 0x05(+5)	0x2005(8197)	0x2025(8229)	0x2045(8261)	0x2065(8293)	... ....	0x27C5(10181)
+ 0x06(+6)	0x2006(8198)	0x2026(8230)	0x2046(8262)	0x2066(8294)	... ....	0x27C6(10182)
+ 0x07(+7)	0x2007(8199)	0x2027(8231)	0x2047(8263)	0x2067(8295)	... ....	0x27C7(10183)
+ 0x08(+8)	0x2008(8200)	0x2028(8232)	0x2048(8264)	0x2068(8296)	... ....	0x27C8(10184)
+ 0x09(+9)	0x2009(8201)	0x2029(8233)	0x2049(8265)	0x2069(8297)	... ....	0x27C9(10185)
+ 0x0A(+10)	0x200A(8202)	0x202A(8234)	0x204A(8266)	0x206A(8298)	... ....	0x27CA(10186)
+ 0x0B(+11)	0x200B(8203)	0x202B(8235)	0x204B(8267)	0x206B(8299)	... ....	0x27CB(10187)
+ 0x0C(+12)	0x200C(8204)	0x202C(8236)	0x204C(8268)	0x206C(8300)	... ....	0x27CC(10188)
+ 0x0D(+13)	0x200D(8205)	0x202D(8237)	0x204D(8269)	0x206D(8301)	... ....	0x27CD(10189)
+ 0x0E(+14)	0x200E(8206)	0x202E(8238)	0x204E(8270)	0x206E(8302)	... ....	0x27CE(10190)
+ 0x0F(+15)	0x200F(8207)	0x202F(8239)	0x204F(8271)	0x206F(8303)	... ....	0x27CF(10191)
+ 0x10(+16)	0x2010(8208)	0x2030(8240)	0x2050(8272)	0x2070(8304)	... ....	0x27D0(10192)
+ 0x11(+17)	0x2011(8209)	0x2031(8241)	0x2051(8273)	0x2071(8305)	... ....	0x27D1(10193)
+ 0x12(+18)	0x2012(8210)	0x2032(8242)	0x2052(8274)	0x2072(8306)	... ....	0x27D2(10194)
+ 0x13(+19)	0x2013(8211)	0x2033(8243)	0x2053(8275)	0x2073(8307)	... ....	0x27D3(10195)
+ 0x14(+20)	0x2014(8212)	0x2034(8244)	0x2054(8276)	0x2074(8308)	... ....	0x27D4(10196)
+ 0x15(+21)	0x2015(8213)	0x2035(8245)	0x2055(8277)	0x2075(8309)	... ....	0x27D5(10197)

0x15(+21)					...	
+ 0x16(+22)	0x2016(8214)	0x2036(8246)	0x2056(8278)	0x2076(8310)	...	0x27D6(10198)
+ 0x17(+23)	0x2017(8215)	0x2037(8247)	0x2057(8279)	0x2077(8311)	...	0x27D7(10199)
+ 0x18(+24)	0x2018(8216)	0x2038(8248)	0x2058(8280)	0x2078(8312)	...	0x27D8(10200)
+ 0x19(+25)	0x2018(8217)	0x2038(8249)	0x2058(8281)	0x2078(8313)	...	0x27D9(10201)
+ 0x1A(+26)	0x201A(8218)	0x203A(8250)	0x205A(8282)	0x207A(8314)	...	0x27DA(10202)
+ 0x1B(+27)	0x201B(8219)	0x203B(8251)	0x205B(8283)	0x207B(8315)	...	0x27DB(10203)
+ 0x1C(+28)	0x201C(8220)	0x203C(8252)	0x205C(8284)	0x207C(8316)	...	0x27DC(10204)
+ 0x1D(+29)	0x201D(8221)	0x203D(8253)	0x205D(8285)	0x207D(8317)	...	0x27DD(10205)
+ 0x1E(+30)	0x201E(8222)	0x203E(8254)	0x205E(8286)	0x207E(8318)	...	0x27DE(10206)
+ 0x1F(+31)	0x201F(8223)	0x203F(8255)	0x205F(8287)	0x207F(8319)	...	0x27DF(10207)

Address Offset	Access	Type, Size	Description
+ 0x02(+2) **	Read	1word	Input start register address of input image word this slot.
+ 0x03(+3) **	Read	1word	Input word's bit offset of input image word this slot.
+ 0x04(+4) **	Read	1word	Output start register address of output image word this slot.
+ 0x05(+5) **	Read	1word	Output word's bit offset of output image word this slot.
+ 0x06(+6) **	Read	1word	Input bit start address of input image bit this slot.
+ 0x07(+7) **	Read	1word	Output bit start address of output image bit this slot.
+ 0x08(+8) **	Read	1word	Size of input bit this slot
+ 0x09(+9) **	Read	1word	Size of output bit this slot
+ 0x0A(+10)* *	Read	n word	Read input data this slot
+	Read/W	n word	Read/write output data this slot

0x0B(+11)**	rite		
+ 0x0E(+14)	Read	1word	GT-number, if GT-1238, returns 0x1238
+ 0x0F(+15)	Read	String upto 72byte	First 1word is length of valid character string. If GT-1238, returns "00 1E 52 54 2D 31 32 33 38 2C 20 38 44 49 2C 20 32 34 56 64 63 2C 20 55 6E 69 76 65 72 73 61 6C 00 00" Valid character size = 0x001E =30 characters, "GT-1238, 8DI, 24Vdc, Universal"
+ 0x10(+16)	Read	1word	Size of configuration parameter byte
+ 0x11(+17)**	Read/W rite	n word	Read/write Configuration parameter data, Refer to each IO parameter Specification.
+ 0x17(+23)	Read	2word	Firmware Revision ex) 0x00010010 (Major revision 1 /Minor revision 1, Rev 1.001)
+ 0x19(+25)	Read	2word	Firmware release date.

\* After the system is reset, the new "Set Value" action is applied.

\*\* Nothing of output, input, memory or configuration parameter corresponding slot returns Exception 02.

## 8.4. Supported MODBUS Function Codes

MODBUS Reference Documents

<http://www.modbus.org>

MODBUS Tools

<http://www.modbustools.com>, modbus poll

<http://www.win-tech.com>, modscan32

## 9. Object models

EtherNet/IP was developed from a very widely implemented standard used for transferring data between two devices in DeviceNet and ControlNet, called the Common Industrial Protocol (CIP). Every CIP node is modeled as a collection of objects. An object provides an abstract representation of a particular component within a device. Anything not described in object form is not visible through the CIP protocol. CIP objects are structured into classes, instances, and attributes.

A class of objects represents the same kind of system component. An object instance is the actual representation of a particular object within a class. Each instance of a class has the same attributes, but it has its own particular set of attribute values.

The objects and their components are addressed by uniform addressing scheme consisting of:

- \* Media Access Control Identifier (MAC ID), an integer identification value assigned to each node on a CIP network
- \* Class Identifier (Class ID), an integer identification value assigned to each Object Class accessible from the network
- \* Instance Identifier (Instance ID), an integer identification value assigned to an Object Instance that identifies it among all instances of the same class.
- \* Attribute Identifier (Attribute ID), an integer identification value assigned to a class and/or instance attribute.
- \* Service code, an integer identification value which denotes a particular object instance and/or object class function.

### 9.1. Supported Objects

**Supported Object**

Name of Object	Type	Number of Instances	Class Code
Identity	Required	1	01 <sub>HEX</sub>
Message Router	Required	1	02 <sub>HEX</sub>
Assembly	Required	2	04 <sub>HEX</sub>
Connection Manager	Required	1	06 <sub>HEX</sub>
Port	Required	1	F4 <sub>HEX</sub>
TCP/IP Interface	Required	1	F5 <sub>HEX</sub>
Ethernet Link	Required	1	F6 <sub>HEX</sub>
GBus Manager	Vendor-specific	1	70 <sub>HEX</sub>
Expansion Slot	Vendor-specific	1~63	71 <sub>HEX</sub>

## 9.2. Identity Object

Class Code: 01<sub>HEX</sub>

### 9.2.1. Common Services

Service Code	Implemented for		Service Name	Value
	Class	Instance		
0x01	Yes	Yes	Get_Attribute_All	
0x05	No	Yes	Reset	0: Reset Only 1: Reset and Factory Default
0x0E	No	Yes	Get_Attribute_Single	

### 9.2.2. Class Attributes

Instance ID	Attribute ID	Access Rule	Name	Data Type	Value
0	1	Get	Revision	UINT	0001 <sub>HEX</sub>
	2	Get	Max Instance	UINT	0001 <sub>HEX</sub>
	6	Get	Maximum ID Number Class Attributes	UINT	0000 <sub>HEX</sub>
	7	Get	Maximum ID Number Instance Attributes	UINT	0000 <sub>HEX</sub>

### 9.2.3. Instance Attributes

Instance ID	Attribute ID	Access Rule	Name	Data Type	Value
1	1	Get	Vendor ID	UINT	741 <sub>DEC</sub> (Crevis Co., Ltd)
	2	Get	Device Type	UINT	0C <sub>HEX</sub> (Communications Adapter)
	3	Get	Product Code	UINT	9000 <sub>HEX</sub> (GN-9289)
	4	Get	Revision - Major - Minor	Structure of: USINT USINT	1 ~ 9 1 ~ 255
	5	Get	Status	WORD	Device status. Defined in standard.
	6	Get	Serial Number	UDINT	Unique Number
	7	Get	Product Name -String Length -ASCII String	Short_String USINT STRING	34 <sub>DEC</sub> "9289,Modbus/TCP Adapter,GBUS"
	<i>Vendor-specific</i>				

	100	Get	Device Fault Code	USINT	00 <sub>HEX</sub> : Normal Operation Bit 0: No expansion slot Bit 1: Too many expansion slot Bit 2: Overflow I/O size Bit 3: I/O Configuration failure Bit 4: EEPROM Checksum fault Bit 6: Invalid Module ID Bit 7: Firmware fault
	104	Get	Firmware Release Date	UDINT	YYYYMMDD <sub>HEX</sub>



## 9.3. Message Router Object

Class Code: 02<sub>HEX</sub>

### 9.3.1. Common Services

Service Code	Implemented for		Service Name
	Class	Instance	
0x01	Yes	No	Get_Attribute_All
0x0E	No	Yes	Get_Attribute_Single

### 9.3.2. Class Attributes

Instance ID	Attribute ID	Access Rule	Name	Data Type	Value
0	1	Get	Revision	UINT	0001 <sub>HEX</sub>
	4	Get	Number of Attribute	UINT	0000 <sub>HEX</sub>
	5		Number of Service	UINT	0000 <sub>HEX</sub>
	6	Get	Maximum ID Number Class Attributes	UINT	0000 <sub>HEX</sub>
	7	Get	Maximum ID Number Instance Attributes	UINT	0000 <sub>HEX</sub>

### 9.3.3. Instance Attributes

Instance ID	Attribute ID	Access Rule	Name	Data Type	Value
1	1	Get	Object Class List	STRUCT of UINT Array of UINT	10 <sub>DEC</sub> 09 00 01 00 02 00 04 00 06 00 F4 00 F5 00 F6 00 70 00 71 00
	2	Get	Number Available	UINT	16 <sub>DEC</sub> Maximum number of connections supported

## 9.4. Assembly Object

Class Code: 04<sub>HEX</sub>

### 9.4.1. Common Services

Service Code	Implemented for		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	No	Yes	Set_Attribute_Single

### 9.4.2. Class Attributes

Instance ID	Attribute ID	Access Rule	Name	Data Type	Value
0	1	Get	Revision	UINT	0002 <sub>HEX</sub>

### 9.4.3. Class Attributes

Instance ID	Attribute ID	Access Rule	Name	Data Type	Value
1	3	Get	Input (Produced) Process Image Data	Array BYTE n	Input process image data
2	3	Set/Get	Output (Consumed) Process Image Data	Array BYTE n	Output process image data

## 9.5. Connection Manager Object

Class Code: 06<sub>HEX</sub>

### 9.5.1. Class Attributes, Instance Attribute

None

## 9.6. Port Object

Class Code: F4<sub>HEX</sub>

### 9.6.1. Common Services

Service Code	Implemented for		Service Name
	Class	Instance	
0x01	Yes	Yes	Get_Attribute_All
0x0E	Yes	Yes	Get_Attribute_Single

### 9.6.2. Class Attributes

Instance ID	Attribute ID	Access Rule	Name	Data Type	Value
0	1	Get	Revision	UINT	0001 <sub>HEX</sub>
	2	Get	Max Instance	UINT	0001 <sub>HEX</sub>
	3	Get	Num Instances	UINT	0001 <sub>HEX</sub>
	8	Get	Entry Port	UINT	0001 <sub>HEX</sub>
	9	Get	All Ports	ARRAY of STRUCT UINT UINT	0000 <sub>HEX</sub> 0000 <sub>HEX</sub> 0004 <sub>HEX</sub> 0002 <sub>HEX</sub>

### 9.6.3. Instance Attributes

Instance ID	Attribute ID	Access Rule	Name	Data Type	Value
1	1	Get	Port Type	UINT	0004 <sub>HEX</sub> , TCP/IP Port
	2	Get	Port Number	UINT	0002 <sub>HEX</sub> , CIP port number associate with port
	3	Get	Port Object	UINT Padded EPATH	
	4	Get	Port Name	Short_String	=0
	7	Get	Node Address	Padded EPATH	

## 9.7. TCP/IP Object

Class Code: F5<sub>HEX</sub>

### 9.7.1. Common Services

Service Code	Implemented for		Service Name
	Class	Instance	
0x01	Yes	Yes	Get_Attribute_All
0x0E	Yes	Yes	Get_Attribute_Single
0x02	No	Yes	Set_Attribute_All
0x10	No	Yes	Set_Attribute_Single

### 9.7.2. Class Attributes

Instance ID	Attribute ID	Access Rule	Name	Data Type	Value
0	1	Get	Revision	UINT	0001 <sub>HEX</sub>
	2	Get	Max Instance	UINT	0001 <sub>HEX</sub>
	3	Get	Num Instances	UINT	

### 9.7.3. Instance Attributes

Instance ID	Attribute ID	Access Rule	Name	Data Type	Value
1	1	Get	Status	DWORD	See section 4.7.3.1.
	2	Get	Configuration Capability	DWORD	00000006 <sub>HEX</sub>
	3	Get/Set	Configuration Control	DWORD	See section 4.7.3.2.
	4	Get	Physical Link Path Size of Path Path	STRUCT of: UINT Padded- PATH	0002 <sub>HEX</sub> 00 00 20 F6 24 01
	5	Get/Set	Interface Configuration	STRUCT of: UDINT UDINT UDINT UDINT UDINT STRING	IP address Network Mask Gateway Address Name Server Name Server 2 Domain Name

### 9.7.3.1. Status Instance Attributes

This attribute indicates the status of the TCP/IP network interface.

**Table 4.7.1. Status Attribute**

Bit	Description
0-3	0 – The Interface Configuration attribute has not been configured. 1 – The Interface Configuration attribute contains valid configuration from BOOTP, DHCP, or non-volatile storage. 2 – The Interface Configuration attribute contains valid configuration, obtained from DIP switch. 3-15 – Reserved.
4	Indicates pending configuration change in TTL and/or Mcast config.
5-31	Reserved

### 9.7.3.2. Configuration Control Instance Attributes

This attribute is a bitmap to control network configuration.

**Table 4.7.1. Configuration Control Attribute**

Bit	Description
0-3	Determine how the device shall obtain its initial configuration at startup. 0 – The device shall use the interface configuration values previously stored in EEPROM. 1 – The device shall use the interface configuration values via BOOTP. 2 – The device shall use the interface configuration values via DHCP upon start-up. 3-15 – Reserved.
4	If TRUE, the device shall resolve host names by querying a DNS server.
5-31	Reserved

## 9.8. Ethernet Link Object

Class Code: F6<sub>HEX</sub>

### 9.8.1. Common Services

Service Code	Implemented for		Service Name
	Class	Instance	
0x01	Yes	Yes	Get_Attribute_All
0x0E	Yes	Yes	Get_Attribute_Single

### 9.8.2. Class Attributes

Instance ID	Attribute ID	Access Rule	Name	Data Type	Value
0	1	Get	Revision	UINT	0002 <sub>HEX</sub>
	2	Get	Max Instance	UINT	0001 <sub>HEX</sub>
	3	Get	Num Instances	UINT	0001 <sub>HEX</sub>

### 9.8.3. Instance Attributes

Instance ID	Attribute ID	Access Rule	Name	Data Type	Value
1	1	Get	Interface Speed	UDINT	10 <sub>DEC</sub> , 100 <sub>DEC</sub>
	2	Get	Interface Flags	DWORD	Bit 0: Link Active Bit 1 : Full Duplex Bit 2~4 : Auto negotiation Bit 5 : Manual Setting required Reset Bit 6 : Local Hardware Fault Others : 0
	3	Get	Physical Address	ARRAY of 6 USINTs	Same as MAC address

## 9.9. G-Bus Manager Object

Class Code: 70<sub>HEX</sub>

### 9.9.1. Common Services

Service Code	Implemented for		Service Name
	Class	Instance	
0x0E	No	Yes	Get_Attribute_Single
0x10	No	Yes	Set_Attribute_Single

### 9.9.2. Class Attributes

None

### 9.9.3. Instance Attributes

Instance ID	Attribute ID	Access Rule	Name	Data Type	Value
1	1	Get	Number of I/O Slot	USINT	(include deactivated slot)
	2	Get	Num of Activated Slot	USINT	
	3	Get	Num of Deactivated Slot	USINT	
	4	Get	External IDs	Array of 64 WORD	See Table 4.9.6. See Appendix A.1.
	5	Get/Set*	Selection of Input (Produced) Process Image Mode	USINT	(default 2) Fixed
	6	Get/Set*	Selection of Output (Consumed) Process Image Mode	USINT	(default 0) Fixed
	10	Get	G-Bus Status	USINT	0: Normal Operation 1: GBus Standby 2: GBus Connection Fault 3: Expansion Configuration Fault 4: No expansion module
	11	Get	Input(Prodedced) Byte Size	UINT	IO input byte size
	12	Get	Output (Consumed) Byte Size	UINT	IO output byte size
	13	Get/Set	Enable Input Run/Idle Header	BOOL	0: Disabled Input Run/Idle Header(default) Fixed
	14	Get/Set	Enable Output Run/Idle Header	BOOL	1: Enabled Output Run/Idle Header(default) Fixed
	15	Get/Set	Output Reset at stop	BOOL	0: Disabled(default) 1: Enable



\*After the system is reset, the new "Set Value" action is applied. If slot location is changed, default value is set automatically.

**Table 9.9.6. External IDs (=Expansion Module ID)**

Word	Description
0	Network Adapter Module External ID = 0x00
1	External ID for slot position #1
2	External ID for slot position #2
.	.
.	.
.	.
62	External ID for slot position #62
63	External ID for slot position #63

## 9.10 Expansion Slot Object

Class Code: 71<sub>HEX</sub>

### 9.10.1 Common Services

Service Code	Implemented for		Service Name
	Class	Instance	
0x0E	No	Yes	Get_Attribute_Single
0x10	No	Yes	Set_Attribute_Single

### 9.10.2. Class Attributes

None

### 9.10.3. Instance Attributes

Instance ID	Attribute ID	Access Rule	Name	Data Type	Value
1~63  (Slot Address)	1	Get	Module External ID	USINT	Crevis Module ID
	2	Get	I/O Data Code - Input Data Code - Output Data Code	Structure of USINT USINT	See Table 9.10.1
	3	Get	Input Offset Table -Byte Offset -Bit Offset	Structure of USINT USINT	Byte offset in the Input Assembly Corresponding bit offset in the byte (If Input data length is zero, then return Empty.)
	4	Get	Output Offset Table - Byte Offset -Bit Offset	Structure of USINT USINT	Byte offset in the Output Assembly Corresponding bit offset in the byte (If Output data length is zero, then return Empty.)
	5	Get	Input Data	Array of BYTE	Read Input data size defined by attribute 2. If Input data length is zero, then return Empty.
	6	Get/Set	Output Data	Array of BYTE	Read/Write Output data size defined by attribute 2. If Output data length is zero, then return Empty.
	7	Get/Set	Active Flag	BOOL	0: This slot is activated 1: This slot is deactivated

	8	Get	Configuration Parameter Data Length	USINT	GBUS I/O Parameter
	9	Get/Set	R/W Configuration Data	n Byte	Data array size defined by attribute 8.
	100	Get	Product Code	4 Byte	See Table 9.10.2
	101	Get	Catalog Number	4 Byte	
	102	Get	Firmware Revision	Structure of USINT USINT	Expansion Module Firmware Revision

\*After the system is reset, the new "Set Value" action is applied. If slot location is changed, default value is set automatically.

**Table 9.10.1. I/O Data Code Format**

Byte#	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
+0	Input Data Type		Input Data Length					
+1	Output Data Type		Output Data Length					

<b>Input/output Type:</b> 0 0: No I/O Data 0 1: Byte Data 1 0: Word Data 1 1: Bit Data	<b>Input/output Data Length:</b> 0 0 0 0 0: 0 Bit/Byte/Word 0 0 0 0 1: 1 Bit/Byte/Word 0 0 0 1 0: 2 Bit/Byte/Word 0 0 0 1 1: 3 Bit/Byte/Word ... 1 1 1 1 1: 63 Bit/Byte/Word
--	--

**Table 9.10.2 Product Code Format**

Byte#	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
+0	Connection Type							
+1	Assembly Type							
+2	Output Information							
+3	Input Information							

**Connection Type**

Byte#	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
+0	Reserved						Mem	IO

<b>IO (Input/output Connection)</b> IO = 0: does not support Input/output Connection IO = 1: support Input/output Connection	<b>MEM (Memory Register Service)</b> MEM = 0: does not support Memory Register Service Connection MEM = 1: support Memory Register Service Connection
--	---

**\* Assembly Type**

Byte#	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
+1	Unit_Type		Priority		S	Reserved		

<b>Unit_Type</b> 0 0: Not Used 0 1: Input Module 1 0: Output Module 1 1: I/O Both Modules	<b>Priority (Input/output Data Priority for assembly)</b> 0 0: Priority 0 (low) - usually it is used by Byte/Bit Type Discrete module. 0 1: Priority 1 1 0: Priority 2 - usually it is used by Analog I/O module. 1 1: Priority 3 (high)
<b>S (Status for Profibus Slot Diagnostic)</b> 0: No Status 1: Support Word Input Diagnostic(0x8000 = -32678)	

For example: ST-3234(current analog input 4~20 mA, 14bit)

Status	Input Data
Normal	0x0000(4mA) ~ 0x3FFF (20mA)
Open Wire or Underrange (0~3mA)	0x8000(~32678)

\*

#### Input/Output Information

Byte#	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
+2	Data_Type		Data_Length						Output Information
+3	Data_Type		Data_Length						Input Information

<b>Input/output Type:</b> 0 0: Byte Data 0 1: Word Data 1 0: Bit Data 1 1: have no Input or Output Data	<b>Input/output Data Length:</b> 0 0 0 0 0: 0 Bit/Byte/Word 0 0 0 0 1: 1 Bit/Byte/Word 0 0 0 1 0: 2 Bit/Byte/Word 0 0 0 1 1: 3 Bit/Byte/Word ... 1 1 1 1 1: 63 Bit/Byte/Word
---	--

#### MODBUS Reference

##### MODBUS Reference Documents

<http://www.modbus.org>

##### MODBUS Tools

<http://www.modbustools.com> , MODBUS poll

<http://www.win-tech.com> , MODSCAN32

## 10. Trouble Shooting

### 10.1. How to diagnose by LED indicator

LED Status	Cause	Action
All LED turns off	- No power	- Check main power Cable
MOD LED Green/Red Toggle	- MODBUS error such as watchdog error, etc.	- Check Modbus watchdog time vlaue(see index 0x1020
MOD LED is red	- Occurrence critical error in firmware	- Contact Sales team and send module for repair.
LINK LED turns off	- Device may not be powered	- Check main power Cable
ACTIVE LED turns off	- Device is idle or may not be powered.	- Check main power Cable
IOS LED turns off	- Device may not be powered.	- Check main power Cable
IOS LED flashes red	- Adapter has no expansion module	- Add one or more expansion modules.
IOS LED is red	<p>One or more expansion module occurred in fault state.</p> <ul style="list-style-type: none"> <li>- Detected invalid expansion module ID.</li> <li>- Overflowed Input/Output Size</li> <li>- Too many expansion module</li> <li>- Initialization failure</li> <li>- Communication failure.</li> <li>- Changed expansion module configuration.</li> <li>- Mismatch vendor code between adapter and expansion module.</li> </ul>	<ul style="list-style-type: none"> <li>- Use expansion slot up to 63.</li> <li>- Compose that IO total size is not excess.</li> <li>- Check status of expansion IO connection.</li> <li>- Check the vendor code of module.</li> </ul>
Field Power LED turns off	- Field power is not supplied.	<ul style="list-style-type: none"> <li>- Check main power Cable</li> <li>- Contact Sales team and send module for repair.</li> </ul>
System Power LED turns off	- System power is not supplied.	<ul style="list-style-type: none"> <li>- Check main power Cable</li> <li>- Contact Sales team and send module for repair.</li> </ul>

## APPENDIX A

### A.1 Product List

No.	GT-Number	Description	ID(hex)
<b>Digital Input Module</b>			
1	GT-1238	8 Points, Universal, 24Vdc, 10RTB	1238
2	GT-123F	16 Points, Universal, 24Vdc, 20P connector	123F
3	GT-12DF	16 Points, Universal, 24Vdc, 18RTB	12DF
4	GT-12FA	32 Points, Universal, 24Vdc, 40P connector	12FA
5	GT-1804	4 Points, 120Vac, 10RTB	1804
6	GT-1904	4 Points, 240Vac, 10RTB	1904
<b>Digital Output Module</b>			
7	GT-2318	8 Points, Sink, 24Vdc/0.5A, 10RTB	2318
8	GT-2328	8 Points, Source, 24Vdc/0.5A, 10RTB	2328
9	GT-221F	16 Points, Sink, 24Vdc/0.3A, 20P connector	221F
10	GT-222F	16 Points, Source, 24Vdc/0.3A, 20P connector	222F
11	GT-225F	16 Points, Sink, 24Vdc/0.3A, 18RTB	225F
12	GT-226F	16 Points, Source, 24Vdc/0.3A, 18RTB	226F
13	GT-22BA	32 Points, Sink, 24Vdc/0.3A, 40P connector	22BA
14	GT-22CA	32 Points, Source, 24Vdc/0.3A, 40P connector	22CA
15	GT-2618	8 Points, Sink, 24Vdc/2A, 10RTB	2618
16	GT-2628	8 Points, Source, 24Vdc/2A, 10RTB	2628
17	GT-2734	4 Points, MOS Relay, 240Vdc/ac, 0.5A, 10RTB	2734
18	GT-2744	4 Points, Relay, 24Vdc/2A, 240Vac/2A, 10RTB	2744
19	GT-2764	4 Points, MOS Relay, 24Vdc/ac, 2A, 10RTB	2764
20	GT-2784	4 Points, MOS Relay, 110Vdc/ac, 1A, 10RTB	2784
<b>Analog Input Module</b>			
21	GT-3001	LoadCell (TBD)	3001
22	GT-3114	4 Channels, 0~20, 4~20mA, 12bits, 10RTB	3114
23	GT-3154	4 Channels, 0~20, 4~20mA, 16bits, 10RTB	3154
24	GT-3118	8 Channels, 0~20, 4~20mA, 12bits, 10RTB	3118
25	GT-3158	8 Channels, 0~20, 4~20mA, 16bits, 10RTB	3158
26	GT-311F	16 Channels, 0~20, 4~20mA, 12bits, 20P connector	311F
27	GT-315F	16 Channels, 0~20, 4~20mA, 16bits, 20P connector	315F
28	GT-317F	16 Channels, 0~20, 4~20mA, 12bits, 18RTB	317F
29	GT-319F	16 Channels, 0~20, 4~20mA, 16bits, 18RTB	319F
30	GT-3424	4 Channels, 0~10, 0~5, 1~5Vdc, 12bits, 10RTB	3424
31	GT-3464	4 Channels, 0~10, 0~5, 1~5Vdc, 16bits, 10RTB	3464
32	GT-3428	8 Channels, 0~10, 0~5, 1~5Vdc, 12bits, 10RTB	3428
33	GT-3468	8 Channels, 0~10, 0~5, 1~5Vdc, 16bits, 10RTB	3468
34	GT-342F	16 Channels, 0~10, 0~5, 1~5Vdc, 12bits, 20P connector	342F

35	GT-346F	16 Channels, 0~10, 0~5, 1~5Vdc, 16bits, 20P connector	346F
36	GT-347F	16 Channels, 0~10, 0~5, 1~5Vdc, 12bits, 18RTB	347F
37	GT-349F	16 Channels, 0~10, 0~5, 1~5Vdc, 16bits, 18RTB	349F
38	GT-3704	4 Channels, RTD, 10RTB	3704
39	GT-3708	8 Channels, RTD, 20P connector	3708
40	GT-3804	4 Channels, Thermocouple, 10RTB	3804
41	GT-3808	8 Channels, Thermocouple, 20P connector	3808
42	GT-3901	AC Measurement	3901
43	GT-3914	4 Channels, Differential, 0~20, 4~20, +/-20mA, 12Bits, 10RTB	3914
44	GT-3934	4 Channels, Differential, 0~20, 4~20, +/-20mA, 16Bits, 10RTB	3934
45	GT-3918	8 Channels, Differential, 0~20, 4~20, +/-20mA, 12Bits, 18RTB	3918
46	GT-3938	8 Channels, Differential, 0~20, 4~20, +/-20mA, 16Bits, 18RTB	3938
47	GT-3924	4 Channels, Differential, 0~5, 0~10, +/-5, +/-10Vdc, 12Bits, 10RTB	3924
48	GT-3944	4 Channels, Differential, 0~5, 0~10, +/-5, +/-10Vdc, 16Bits, 10RTB	3944
49	GT-3928	8 Channels, Differential, 0~5, 0~10, +/-5, +/-10Vdc, 12Bits, 18RTB	3928
50	GT-3948	8 Channels, Differential, 0~5, 0~10, +/-5, +/-10Vdc, 16Bits, 18RTB	3948
<b>Analog Output Module</b>			
51	GT-4114	4CH, 0~20mA, 12Bits, 10RTB	4114
52	GT-4154	4CH, 0~20mA, 16Bits, 10RTB	4154
53	GT-4118	8CH, 0~20mA, 12Bits, 10RTB	4118
54	GT-4158	8CH, 0~20mA, 16Bits, 10RTB	4158
55	GT-4424	4CH, 0~10Vdc, 12Bits, 10RTB	4424
56	GT-4464	4CH, 0~10Vdc, 16Bits, 10RTB	4464
57	GT-4428	8CH, 0~10Vdc, 12Bits, 10RTB	4428
58	GT-4468	8CH, 0~10Vdc, 16Bits, 10RTB	4468
59	GT-417F	16CH, 0~20mA, 12Bits, 18RTB	417F
60	GT-419F	16CH, 0~20mA, 16Bits, 18RTB	419F
61	GT-442F	16CH, 0~10Vdc, 12Bits, 20P Connector	442F
62	GT-446F	6CH, 0~10Vdc, 16Bits, 20P Connector	446F
63	GT-447F	16CH, 0~10Vdc, 12Bits, 18RTB	447F
64	GT-449F	16CH, 0~10Vdc, 16Bits, 18RTB	449F
<b>Special Module</b>			
65	GT-5102	2CH, Encoder, Input, 5Vdc, 10RTB	5102
66	GT-5211	1CH, RS 232, RTS/CTS, Full Duplex Type, 10RTB	5211
67	GT-5212	2CH, RS 232, Full Duplex Type, 10RTB	5212
68	GT-5221	1CH, RS 485, Full Duplex Type, 10RTB	5221
69	GT-5231	1CH, RS 485, Half Full Duplex Type, 10RTB	5231
70	GT-5232	2CH, RS 485, Half Full Duplex Type, 10RTB	5232
71	GT-5352	2CH, Synchronous Serial Interface Input, 10RTB	5352
72	GT-5521	1CH, Stepper Module (TBD)	5521
<b>Power Module</b>			
73	GT-7408	Shield Module	7408
74	GT-7508	Common for 0Vdc	7508
75	GT-7511	Power Expansion, In 24Vdc, Out 1A/5Vdc	7511
76	GT-7518	Common for 24Vdc	7518

---

77	GT-7588	Common for 0Vdc, 24Vdc	7588
78	GT-7641	Field Power, 5/24/48 Vdc, 110/220 Vac	7641

## A.2. Glossary

- System Power : The power for starting up CPU.
- Field Power : The power for input and output line.
- Terminator Resistor : Resistor for prevention reflected wave.
- EDS : Electronic Data Sheet.
- Sink : The method of in/output power supply if a device has no power source.
- Source : The method of in/output power supply if a device has the power source.