

---

# FnIO G - Series

## GL-9089

---

**GL-9089(MODBUS TCP/UDP Network Adapter)**

Date : 2020.11.04

## Table of Contents

History.....	4
1. ENVIRONMENT SPECIFICATION.....	4
2. GL-9089 (MODBUS TCP/UDP NETWORK ADAPTER).....	6
2.1 GL-9089 Specification.....	6
2.2 GL-9089 Wiring Diagram.....	7
2.3 GL-9089 LED Indicator.....	8
2.3.1 LED Indicator.....	8
2.3.2 MOD (Module Status LED).....	8
2.3.3 LINK (Physical Connection LED).....	8
2.3.4 ACTIVE (Exchange Data/Traffic Present LED).....	8
2.3.5 IOS LED (Extension Module Status LED).....	9
2.4 GL-9089 Electrical Interface.....	10
2.4.1 RJ-45 Socket.....	10
2.4.2 Dip Switch.....	10
2.5 MODBUS/TCP IP – Address Setup.....	11
2.5.1 IP-Address Setup using BOOTP/DHCP Sever.....	11
2.5.2. IP-Address Setup using Dip Switch(Manual Function).....	12
2.6. MODBUS/TCP IP – Web Server.....	13
2.7. I/O Process Image Map.....	14
2.7.1 MODBUS Interface Register/Bit Map.....	14
2.7.2. Example of Input Process Image (Input Register) Map.....	15
2.7.3. Example of Output Process Image (Output Register) Map.....	16
3. MODBUS TCP/ UDP INTERFACE.....	17
3.1. MODBUS TCP/ UDP Protocol.....	17
3.1.1. Comparison of MODBUS TCP/ UDP And MODUB/RTU.....	17
3.1.2. MODBUS TCP/ UDP MBAP Header.....	17
3.2. Supported MODBUS Function Codes.....	18
3.2.1. 1 (0x01) Read Coils.....	18
3.2.2. 2 (0x02) Read Discrete Inputs.....	19
3.2.3. 3 (0x03) Read Holding Resgisters.....	19
3.2.4. 4 (0x04) Read Input Resgisters.....	20
3.2.5. 5 (0x05) Write Single Coil.....	20
3.2.6. 6 (0x06) Write Single Register.....	21
3.2.7. 8 (0x08) Diagnostics.....	21
3.2.8. 15 (0x0F) Write Multiple Coils.....	24
3.2.9. 16 (0x10) Write Multiple Resgisters.....	25
3.2.10. 23 (0x17) Read/Write Multiple Resgisters.....	26
3.2.11. Error Response.....	27
3.3. MODBUS Special Register Map.....	28
3.3.1 Adapter Identification Special Resgister (0x1000, 4096).....	28
3.3.2 Adapter Watchdog Time, other Time Special Register (0x1020, 4128).....	28
3.3.3 Adapter TCP/IP Special Register (0x1040, 4160).....	28
3.3.4. Adapter Information Special Register (0x1100, 4352).....	29
3.3.5 Expasion Slot Information Special Resister (0x2000, 8192).....	30
3.4. MODBUS Reference.....	31
4. OBJECT MODELS.....	32
4.1 Supported Objects.....	32
4.2 Identity Object.....	32
4.2.1. Common Services.....	32
4.2.2 Class Attributes.....	33
4.2.3 Instance Attributes.....	33
4.3. Message Router Object.....	34
4.3.1 Common Services.....	34
4.3.2 Class Attributes.....	34
4.3.3 Instance Attributes.....	34
4.4 Assembly Object.....	34
4.4.1 Common Services.....	34
4.4.2 Class Attributes.....	34
4.5. Connection Manager Object.....	35

---

4.5.1 Class Attributes, Instance Attribute.....	35
4.6. Port Object.....	35
4.6.1 Common Services.....	35
4.6.2 Class Attributes.....	35
4.6.3. Instance Attributes.....	35
4.7.TCP/IP Object.....	36
4.7.1. Common Services.....	36
4.7.2. Class Attributes.....	36
4.7.3. Instance Attributes.....	36
4.7.3.1. Status Instance Attributes.....	36
4.7.3.2. Configuration Control Instance Attributes.....	37
4.8.Ethernet Link Object.....	37
4.8.1. Common Services.....	37
4.8.2. Class Attributes.....	37
4.8.3. Instance Attributes.....	37
4.9. Fn-Bus Manager Object.....	38
4.9.1 Common Services.....	38
4.9.2. Class Attributes.....	38
4.9.3. Instance Attributes.....	38
4.10. Expansion Slot Object.....	39
4.10.1 Common Services.....	39
4.10.2 Class Attributes.....	39
4.10.3 Instance Attributes .....	39
4.11. Ethernet/IP Reference.....	41

## History

REV.	PAGES	REMARKS	DATE	Editor
1.00			Nov. 4, 2019	Joonho Park
	6,7	UL Update	Mar. 03, 2020	Joonho Park
1.01	29	Modbus special register map Update(0x1119)	July 07, 2020	Joonho Park
1.01			September 01, 2020	Joonho Park
1.02	6	TCP Timeout Remark delete	November 04, 2020	Joonho Park

## 1. ENVIRONMENT SPECIFICATION

Test Equipment	
Operating Temperature	-40°C ~ 60°C : 1A 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

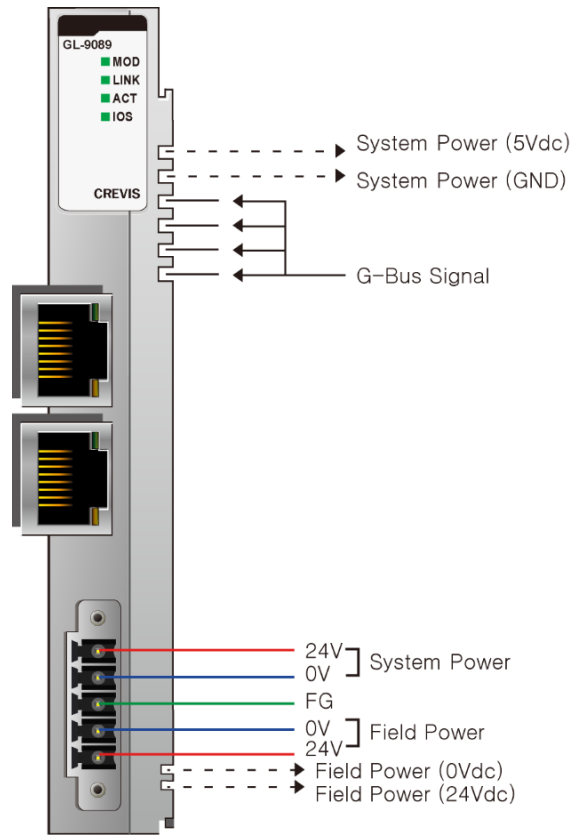
General specification	
Shock Operating	IEC 60068-2-27
Vibration Resistance	Based on IEC 60068-2-6 DNVGL-CG-0039 : Vibration Class B, 4g
Industrial Emissions	EN 61000-6-4/A11 : 2011
Industrial Immunity	EN 61000-6-2 : 2005
Installation Position	Vertical and horizontal installation is available.
Product Certifications	CE, UL, FCC

## 2. GL-9089 (MODBUS TCP/UDP NETWORK ADAPTER)

### 2.1 GL-9089 Specification

Items	Specification
<b>Communication Interface 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	16 slots
Max. Input / Output Data Size	Max. Input 256 bytes / Output 256 bytes
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)
IAP Mode	When DIP Switch 1 to 8 setting is 254 or 255 (Using only Internet Explorer / recommended version 11)
Indicator	4 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) 2 LEDs (each RJ45 Connector) 1 Yellow, Link/Active 1 Green, Not used
Module Location	Starter module left side of G-Series system
<b>General specification</b>	
UL System Power	Supply voltage : 24Vdc nominal, Class2
System Power	Supply voltage : 24Vdc nominal Supply voltage range : 15~28.8Vdc Reverse polarity protection
Power Dissipation	75mA typical @ 24Vdc
Current for I/O Module	1.0A @ 5Vdc
Isolation	System power to internal logic : Non-isolation System power I/O driver : Isolation
UL Field Power	Supply voltage : 24Vdc nominal, Class2
Field Power	Supply voltage : 24Vdc typical (Max. 28.8Vdc) * Field Power Range is different depending on IO Module series. Refer to IO Module`s Specification.
Max. Current Field Power Contact	DC 8A Max
Wiring	I/O Cable Max. 2.0mm <sup>2</sup> (AWG 14)
Torque	0.8Nm(7 lb-in)
Weight	76g
Module size	22mm x 109mm x 70mm
Environment Condition	Refer to '1. Environment Specification'

## 2.2 GL-9089 Wiring Diagram



Pin No.	Signal Description
1	System Power, 24V
2	System Power, Ground
3	F.G
4	Field Power, Ground
5	Field Power, 24V

## 2.3 GL-9089 LED Indicator

### 2.3.1 LED Indicator



LED	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

### 2.3.2 MOD (Module Status LED)

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.

### 2.3.3 LINK (Physical Connection LED)

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

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

Protocol	Status	LED	To indicate
Modbus TCP	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.
Ethernet/IP	Not Powered No IP Address	OFF	Device does not have IP address or may not be powered
	CIP Connections	Green	Device has an IP address and at least one established connection.
	No Connections	OFF	Device has obtained an IP address, but has



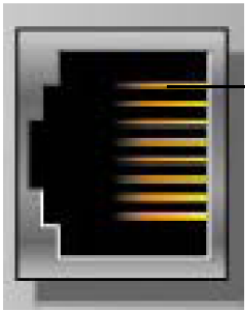
			no established connections.
	Connection Time-out	Flashing Red	Connection time out in one or more of the connections the device has.

**2.3.5 IOS LED (Extension Module Status LED)**

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>

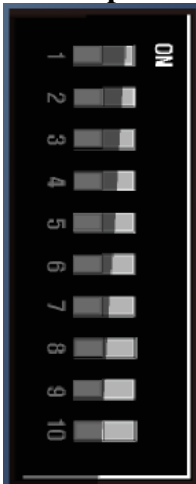
## 2.4 GL-9089 Electrical Interface

### 2.4.1 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	Shield RJ-45 Socket

### 2.4.2 Dip Switch



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).

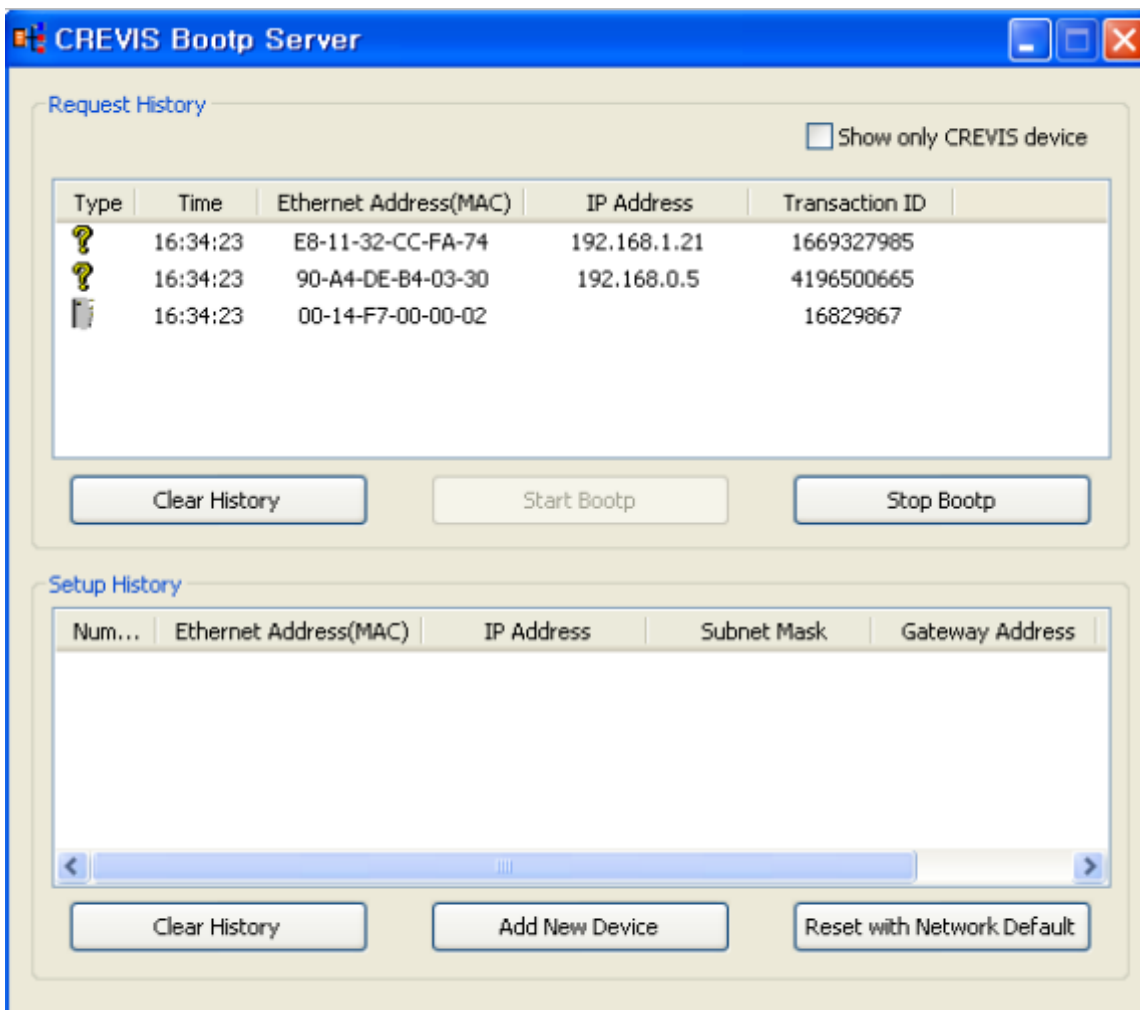
## 2.5 MODBUS/TCP IP – Address Setup

### 2.5.1 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.

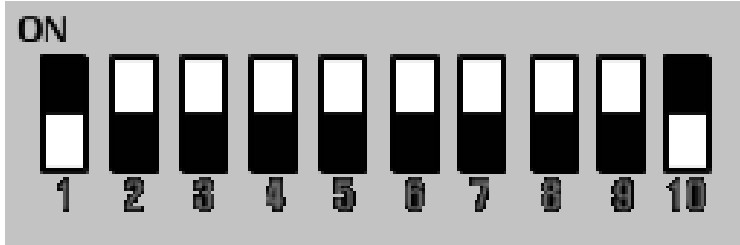
- CREVIS IO Guide Pro' s BOOTP server



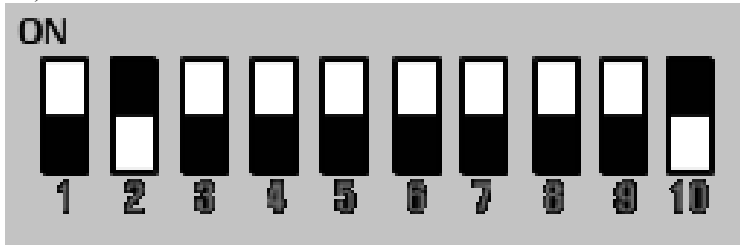
## 2.5.2. IP-Address Setup using Dip Switch(Manual Function)

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.

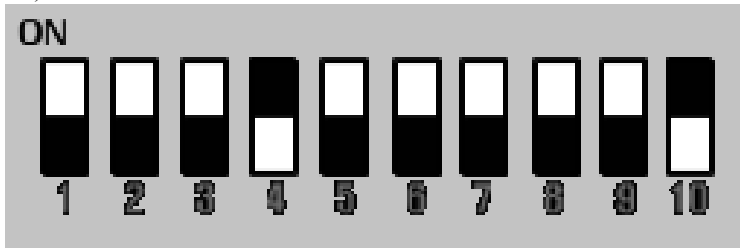
Ex) xxx . xxx . xxx . 1



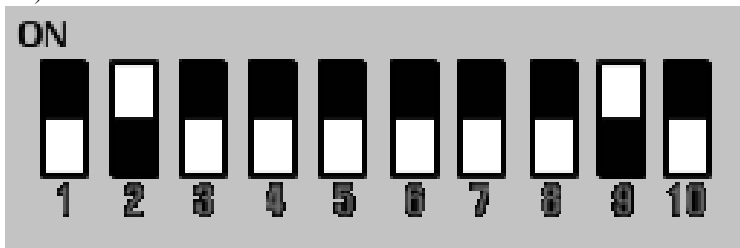
Ex) xxx . xxx . xxx . 2



Ex) xxx . xxx . xxx . 8



Ex) xxx . xxx . xxx . 253



## 2.6. MODBUS/TCP IP – Web Server

### • Network Adapter

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

**Network Adapter**  
GL-9089 (Modbus/TCP, UDP with Ethernet/IP) Network Adapter

[Io Input Data](#) / [Io Output Data](#)

- IP Address : 192.168.100.100
- Subnet Mask : 255.255.255.0
- Gateway : 192.168.0.1
- MAC Address : 00:14:F7:00:00:04
- 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(10/15/2019)
- Expansion Modules : 2 module(s)
- IO Size(Input) : 2 byte(s)
- IO Size(Output) : 2 byte(s)

### • Expansion Module

**Expansion Module**

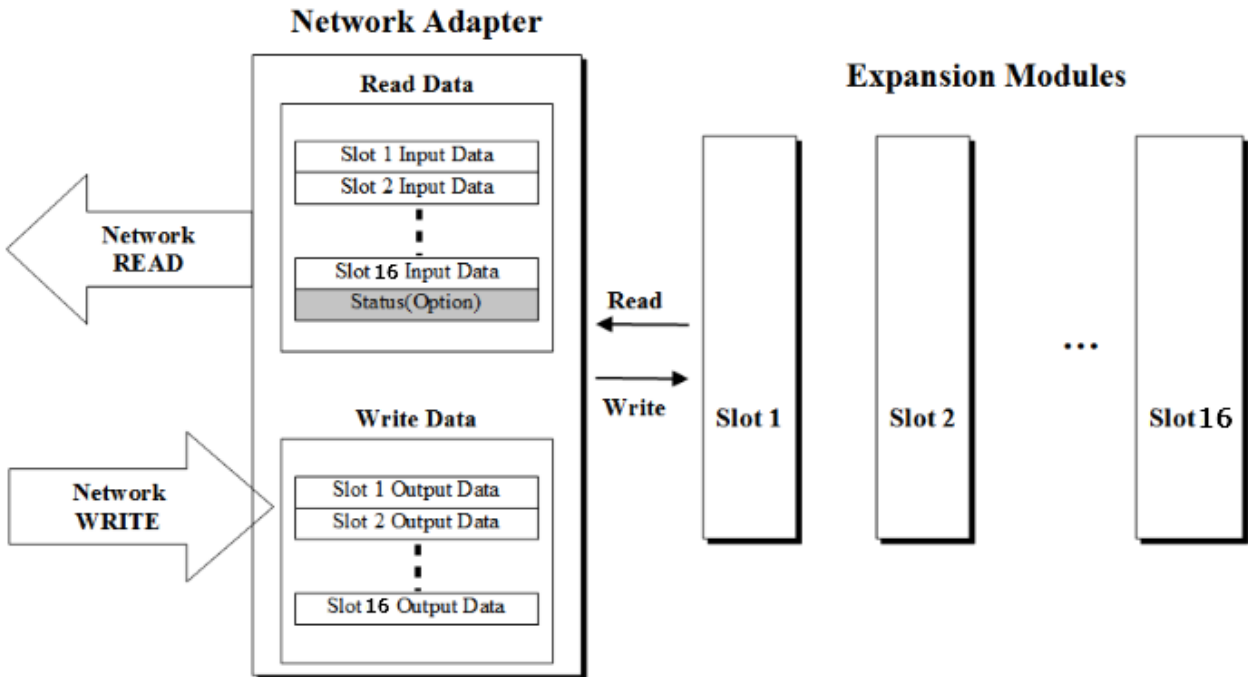
**Network Adapter**  
GL-9089 (Modbus/TCP, UDP with Ethernet/IP) Network Adapter

[Io Input Data](#) / [Io Output Data](#)

Slot#	Descriptions	Input Reg. Mapping	Output Reg. Mapping
<a href="#">Slot#1</a>	GT-221F, 16DO, 24Vdc, Sink	_____	0x0800/0
<a href="#">Slot#2</a>	GT-123F, 16DI, 24Vdc, Universal	0x0000/0	_____

## 2.7. 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



### 2.7.1 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).

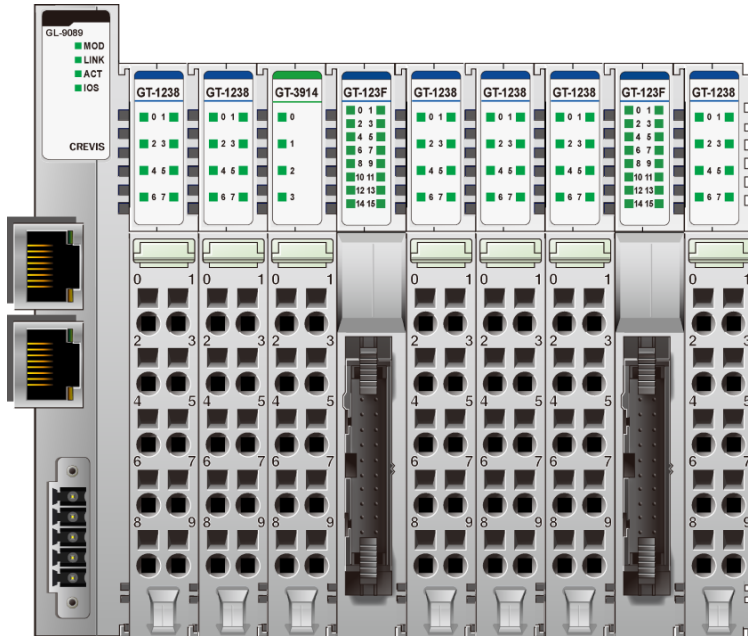
#### • Register Map

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

### 2.7.2. 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 input
#2	8-discrete input
#3	4-analog input
#4	16-discrete input
#5	8-discrete input
#6	8-discrete input
#7	8-discrete input
#8	16-discrete input
#9	8-discrete input

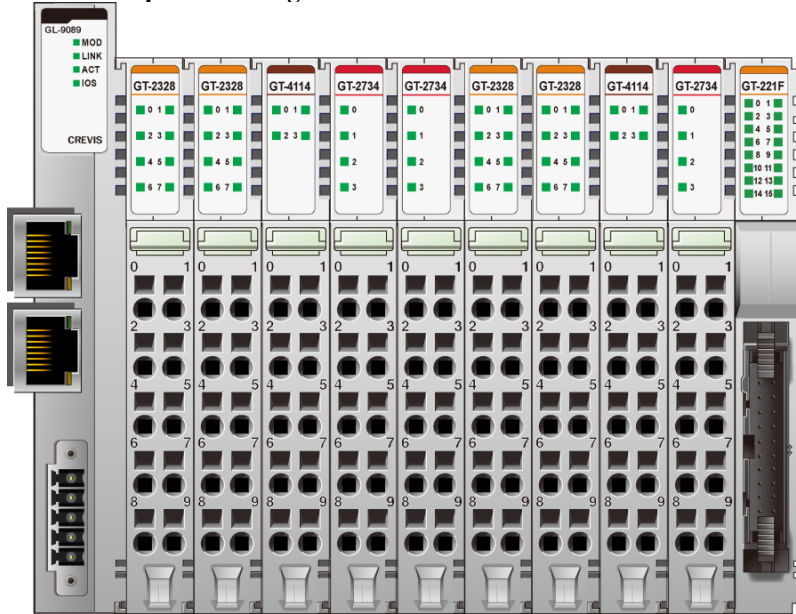
- Input Process Image

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)							

### 2.7.3. 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	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
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)							



### 3. MODBUS TCP/ UDP INTERFACE

#### 3.1. 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.

##### 3.1.1. 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

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.

##### 3.1.2. 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.

### 3.2. 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

#### 3.2.1. 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.

### 3.2.2. 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.

### 3.2.3. 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	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 0x0800, 0x0801 output register value: 0x1122, 0x3344.

### 3.2.4. 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.

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.

### 3.2.5. 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.

### 3.2.6. 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.

### 3.2.7. 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) Watchdog time value, auto recovery will be the factory defaults value.

2) Mac Address, IP Address, Subnet Mask Address, Gateway 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 1 word of input process image.

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

### 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	

### 3.2.8. 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.



### 3.2.9. 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.

### 3.2.10. 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.

### 3.2.11. 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.

- GL-9089 response exception code 01, 02, 03, 04 and 06.

### 3.3. 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).

#### 3.3.1 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 = 0x9170
0x1003(4099)	Read	1word	Firmware revision, if 0x0101, revision 1.01
0x1004(4100)	Read	2words	Product unique serial number
0x1005(4101)	Read	String upto 18bytes	Product name string (ASCII) “GL-9089,Modbus/TCP Adapter,GBUS”
0x1006(4102)	Read	1word	Sum check of EEPROM
0x1010(4112)	Read	2words	Firmware release date
0x1011(4113)	Read	2words	Product manufacturing inspection date
0x101E(4126)	Read	15words - 2words - 2words - 2words - 3words - 1word - 1word - 1word - 1word - 1word - 2words	Composite Id of following address 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

#### 3.3.2 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)

#### 3.3.3 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	1 word	Number of ModBus/TCP connected
0x1043(4163)	Read	1 word	ModBus/TCP port, fixed 502
0x1044(4164)	Read	1 word	Ethernet Interface Speed, 10(10Mbps) or 100(100Mbps)
0x1045(4165)*	Read/Write	1 word	IP Setting Method. 0: BOOTP, 1:DHCP
0x1046(4166)	---	---	Reserved.
0x1047(4167)	Read	1 word	Status of DIP SW#9 DHCP/BOOTP(Enable/Disable). 0 : OFF, 1 : ON
0x1048(4168)	Read	1 word	Enable/disable Lowest IP address via DIP Switch, 1:Enabled
0x1050(4176)	Read/Write	2words	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	2words	Subnet mask. If 255.255.255.0, then 0xFFFF, 0x00FF.
0x1052(4178)	Read/Write	2words	Gateway. If 192.168.123.254, then 0xA8C0, 0xFE7B.
0x1053(4179)	Read	3words	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.

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

Address	Access	Type, Size	Description																						
0x1100(4352)*	Read/Write	1 word	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	1 word	Start address of input image word register. =0x0000																						
0x1103(4355)	Read	1 word	Start address of output image word register. =0x0800																						
0x1104(4356)	Read	1 word	Size of input image word register.																						
0x1105(4357)	Read	1 word	Size of output image word register.																						
0x1106(4358)	Read	1 word	Start address of input image bit. = 0x0000																						
0x1107(4359)	Read	1 word	Start address of output image bit. =0x1000																						
0x1108(4360)	Read	1 word	Size of input image bit.																						
0x1109(4361)	Read	1 word	Size of output image bit.																						
0x110A(4362)	Read	1 word	Update time for cyclic data change (same as 0x1028)																						
0x110D(4365)	Read	1 word	Current Dip Switch State ex) DHCP/Booth enable, Dip SW(0x01) = 0x0101																						
0x110E(4366)	Read	upto 33words	Expansion slot's GT-number including GL First 1word is adapter's number, if GL-9089, then 0x9089																						
0x1110(4368)	Read	1 word	Number of expansion slot																						
0x1113(4371)	Read	upto 33words	Expansion slot Module Id. First 1word is adapter's module id.																						
0x1119(4377)	Read	1 word	Hi byte is ModBus status, low byte is internal bus status. Zero value means 'no error'. <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">ModBus status</th> <th style="width: 50%;">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>0x01 : COMMUNICATION_FAULT</td> </tr> <tr> <td>0x40 : ERR_CRC_LRC</td> <td>0x02 : CONNECT_FAULT</td> </tr> <tr> <td>0x80 : ERR_WATCHDOG</td> <td>0x03 : CONFIG_FAULT</td> </tr> <tr> <td></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	0x01 : COMMUNICATION_FAULT	0x40 : ERR_CRC_LRC	0x02 : CONNECT_FAULT	0x80 : ERR_WATCHDOG	0x03 : CONFIG_FAULT		0x04 : NO_EXPANSION		0x05 : INVALID_ATTR_VALUE		0x06 : TOO_MUCH_DATA		0x07 : VENDOR_ERROR		0x08 : NOT_EXPECTED_SLOT		0x09 : CRC_ERROR
ModBus status	Internal bus status(G-Bus)																								
0x00 : No Error	0x00 : OPERATING																								
0x01 : ERR_DIP_SWITCH	0x01 : COMMUNICATION_FAULT																								
0x40 : ERR_CRC_LRC	0x02 : CONNECT_FAULT																								
0x80 : ERR_WATCHDOG	0x03 : CONFIG_FAULT																								
	0x04 : NO_EXPANSION																								
	0x05 : INVALID_ATTR_VALUE																								
	0x06 : TOO_MUCH_DATA																								
	0x07 : VENDOR_ERROR																								
	0x08 : NOT_EXPECTED_SLOT																								
	0x09 : CRC_ERROR																								
0x111D(4381)	Read	1 word	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).

### 3.3.5 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)

Address Offset	Expansion Slot#1	Expansion Slot#2	Expansion Slot#3	Expansion Slot#4	.....	Expansion Slot#16
+ 0x00(+0)	0x2000(8192)	0x2020(8224)	0x2040(8256)	0x2060(8288)	.....	0x21E0(8672)
+ 0x01(+1)	0x2001(8193)	0x2021(8225)	0x2041(8257)	0x2061(8289)	.....	0x21E1(8673)
+ 0x02(+2)	0x2002(8194)	0x2022(8226)	0x2042(8258)	0x2062(8290)	.....	0x21E2(8674)
+ 0x03(+3)	0x2003(8195)	0x2023(8227)	0x2043(8259)	0x2063(8291)	.....	0x21E3(8675)
+ 0x04(+4)	0x2004(8196)	0x2024(8228)	0x2044(8260)	0x2064(8292)	.....	0x21E4(8676)
+ 0x05(+5)	0x2005(8197)	0x2025(8229)	0x2045(8261)	0x2065(8293)	.....	0x21E5(8677)
+ 0x06(+6)	0x2006(8198)	0x2026(8230)	0x2046(8262)	0x2066(8294)	.....	0x21E6(8678)
+ 0x07(+7)	0x2007(8199)	0x2027(8231)	0x2047(8263)	0x2067(8295)	.....	0x21E7(8679)
+ 0x08(+8)	0x2008(8200)	0x2028(8232)	0x2048(8264)	0x2068(8296)	.....	0x21E8(8680)
+ 0x09(+9)	0x2009(8201)	0x2029(8233)	0x2049(8265)	0x2069(8297)	.....	0x21E9(8681)
+ 0x0A(+10)	0x200A(8202)	0x202A(8234)	0x204A(8266)	0x206A(8298)	.....	0x21EA(8682)
+ 0x0B(+11)	0x200B(8203)	0x202B(8235)	0x204B(8267)	0x206B(8299)	.....	0x21EB(8683)
+ 0x0C(+12)	0x200C(8204)	0x202C(8236)	0x204C(8268)	0x206C(8300)	.....	0x21EC(8684)
+ 0x0D(+13)	0x200D(8205)	0x202D(8237)	0x204D(8269)	0x206D(8301)	.....	0x21ED(8685)
+ 0x0E(+14)	0x200E(8206)	0x202E(8238)	0x204E(8270)	0x206E(8302)	.....	0x21EE(8686)
+ 0x0F(+15)	0x200F(8207)	0x202F(8239)	0x204F(8271)	0x206F(8303)	.....	0x21EF(8687)
+ 0x10(+16)	0x2010(8208)	0x2030(8240)	0x2050(8272)	0x2070(8304)	.....	0x21E0(8688)
+ 0x11(+17)	0x2011(8209)	0x2031(8241)	0x2051(8273)	0x2071(8305)	.....	0x21E1(8689)
+ 0x12(+18)	0x2012(8210)	0x2032(8242)	0x2052(8274)	0x2072(8306)	.....	0x21E2(8690)
+ 0x13(+19)	0x2013(8211)	0x2033(8243)	0x2053(8275)	0x2073(8307)	.....	0x21E3(8691)
+ 0x14(+20)	0x2014(8212)	0x2034(8244)	0x2054(8276)	0x2074(8308)	.....	0x21E4(8692)
+ 0x15(+21)	0x2015(8213)	0x2035(8245)	0x2055(8277)	0x2075(8309)	.....	0x21E5(8693)
+ 0x16(+22)	0x2016(8214)	0x2036(8246)	0x2056(8278)	0x2076(8310)	.....	0x21E6(8694)
+ 0x17(+23)	0x2017(8215)	0x2037(8247)	0x2057(8279)	0x2077(8311)	.....	0x21E7(8695)
+ 0x18(+24)	0x2018(8216)	0x2038(8248)	0x2058(8280)	0x2078(8312)	.....	0x21E8(8696)
+ 0x19(+25)	0x2019(8217)	0x2039(8249)	0x2059(8281)	0x2079(8313)	.....	0x21E9(8697)
+ 0x1A(+26)	0x201A(8218)	0x203A(8250)	0x205A(8282)	0x207A(8314)	.....	0x21EA(8698)
+ 0x1B(+27)	0x201B(8219)	0x203B(8251)	0x205B(8283)	0x207B(8315)	.....	0x21EB(8699)
+ 0x1C(+28)	0x201C(8220)	0x203C(8252)	0x205C(8284)	0x207C(8316)	.....	0x21EC(8700)
+ 0x1D(+29)	0x201D(8221)	0x203D(8253)	0x205D(8285)	0x207D(8317)	.....	0x21ED(8701)
+ 0x1E(+30)	0x201E(8222)	0x203E(8254)	0x205E(8286)	0x207E(8318)	.....	0x21EE(8702)
+ 0x1F(+31)	0x201F(8223)	0x203F(8255)	0x205F(8287)	0x207F(8319)	.....	0x21EF(8703)

Address Offset	Access	Type, Size	Description
+ 0x02(+2) **	Read	1 word	Input start register address of input image word this slot.
+ 0x03(+3) **	Read	1 word	Input word's bit offset of input image word this slot.

+ 0x04(+4) **	Read	1 word	Output start register address of output image word this slot.
+ 0x05(+5) **	Read	1 word	Output word's bit offset of output image word this slot.
+ 0x06(+6) **	Read	1 word	Input bit start address of input image bit this slot.
+ 0x07(+7) **	Read	1 word	Output bit start address of output image bit this slot.
+ 0x08(+8) **	Read	1 word	Size of input bit this slot
+ 0x09(+9) **	Read	1 word	Size of output bit this slot
+ 0x0A(+10)**	Read	n words	Read input data this slot
+ 0x0B(+11)**	Read/Write	n words	Read/write output data this slot
+ 0x0E(+14)	Read	1 word	GT-number, if GT-1238, returns 0x1238
+ 0x0F(+15)	Read	String upto 72bytes	First 1 word 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	1 word	Size of configuration parameter byte
+ 0x11(+17)**	Read/Write	n words	Read/write Configuration parameter data, Refer to each IO parameter Specification.
+ 0x17(+23)	Read	2 words	Firmware Revision ex) 0x00010010 (Major revision 1 /Minor revision 16, Rev 1.016)
+ 0x19(+25)	Read	2 words	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.

### 3.4. MODBUS Reference

MODBUS Reference Documents

<http://www.modbus.org>

MODBUS Tools

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

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



## 4. 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.

### 4.1 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>
FnBus Manager	Vendor-specific	1	70 <sub>HEX</sub>
Expansion Slot	Vendor-specific	1~16	71 <sub>HEX</sub>

### 4.2 Identity Object

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

#### 4.2.1. 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	



4.2.2 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>

4.2.3 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	9170 <sub>HEX</sub> (GL-9089)	
	4	Get	Revision - Major - Minor	Structure of: USINT USINT	1 ~ 9 1 ~ 255	
	5	Get	Status	WORD	Device status. Defined in standard. (Fixed now, use later)	
	6	Get	Serial Number	UDINT	Unique Number	
	7	Get	Product Name -String Length -ASCII String	Short String USINT STRING	34 <sub>DEC</sub> “GL-9089,Modbus/TCP Adapter,GBUS”	
	Vendor-specific					
	100	Get	Device Fault Code	USINT	0x00 : OPERATING 0x02 : CONNECT_FAULT 0x03 : CONFIG_FAULT 0x04 : NO_EXPANSION 0x05 : INVALID_ATTR_VALUE 0x06 : TOO_MUCH_DATA 0x07 : VENDOR_ERROR 0x08 : NOT_EXPECTED_SLOT 0x09 : CRC_ERROR	
	104	Get	Firmware Release Date	UDINT	YYYYMMDD <sub>HEX</sub>	

### 4.3. Message Router Object

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

#### 4.3.1 Common Services

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

#### 4.3.2 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>

#### 4.3.3 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

### 4.4 Assembly Object

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

#### 4.4.1 Common Services

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

#### 4.4.2 Class Attributes

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

#### 4.4.3 Class Attributes

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

## 4.5. Connection Manager Object

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

### 4.5.1 Class Attributes, Instance Attribute

None

## 4.6. Port Object

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

### 4.6.1 Common Services

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

### 4.6.2 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>

### 4.6.3. 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	02 00 20 F5 24 01
	4	Get	Port Name	Short_String	=0
	7	Get	Node Address	Padded EP ATH	Not used

## 4.7.TCP/IP Object

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

### 4.7.1. 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

### 4.7.2. 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	0001 <sub>HEX</sub>

### 4.7.3. 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> (Offset) 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

#### 4.7.3.1. 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 – Reserved. 3 – Reserved.
4	Indicates pending configuration change in TTL and/or Mcast config.
5-31	Reserved

### 4.7.3.2. 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

## 4.8. Ethernet Link Object

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

### 4.8.1. Common Services

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

### 4.8.2. 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>

### 4.8.3. Instance Attributes

Instance ID	Attribute ID	Access Rule	Name	Data Type	Value
1	1	Get	Interface Speed	UDINT	FFFF <sub>HEX</sub> Auto Detect
	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

## 4.9. Fn-Bus Manager Object

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

### 4.9.1 Common Services

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

### 4.9.2. Class Attributes

None

### 4.9.3. 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.
	10	Get	Fn-Bus Status	USINT	1: INIT STATE 2: IDLE STATE 3: RUN STATE 4: STOP STATE 5: FAULT STATE 6: RESET STATE 7: CRCERR STATE 8: PAUSE STATE
	11	Get	Input (Produced) 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:Disable(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 4.9.6. 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
.	.
.	.
15	External ID for slot position #15
16	External ID for slot position #16

## 4.10. Expansion Slot Object

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

### 4.10.1 Common Services

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

### 4.10.2 Class Attributes

None

### 4.10.3 Instance Attributes

Instance ID	Attribute ID	Access Rule	Name	Data Type	Value
1~16 (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 4.10.1
	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.
	8	Get	Configuration Parameter Data length	USINT	FnBUS 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 4.10.2
	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 4.10.1. 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 4.10.2. 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						<b>Mem</b>	<b>IO</b>

<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	<b>Unit Type</b>		<b>Priority</b>		<b>S</b>	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 4~20mA, 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	<b>Data Type</b>		<b>Data Length</b>				Output Information	
+3	<b>Data Type</b>		<b>Data Length</b>				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
---	--



## 4.11. Ethernet/IP Reference

Ethernet/IP Reference Documents

<http://www.odva.org>

<http://www.ethernet-ip.org>

Ethernet/IP Tools

<http://www.pyramid-solutions.com>