

FnIO G-Series:

GN-948X

EtherCAT Slave & Modbus Programmable I/O



Table of Contents

History.....	4
1.Environment Specification.....	5
2.G-Series Light Version Programmable I/O.....	6
2.1.Specification.....	6
2.2.Ethernet connection specification	8
2.3.Serial connection specification	8
2.4.Wiring Diagram.....	9
2.4.1.Power Connector.....	9
2.5.LED Indicator.....	10
2.5.1.LED Indicator.....	10
2.5.2.MOD (Module Status LED).....	10
2.5.3.NET (Network Status LED).....	10
2.5.4.RUN (PLC Run/Stop Status LED).....	10
2.5.5.IOS (Expansion Module Status LED).....	10
2.5.6.Exception Indication	11
2.5.7.Field Power LED	11
2.6.Electrical Interface.....	12
2.6.1.RJ-45 Socket.....	12
2.6.2.D-Sub 9Pin.....	12
2.6.3.Toggle Switch.....	12
2.6.4.Push Switch.....	12
2.6.5.Dip Switch.....	12
2.7.Web-Server	13
2.7.1.Web-Server Address.....	13
2.7.2.Expansion Module.....	13
2.7.3.Codesys PLC.....	14
2.7.4.Network Setting	14
3.EtherCAT Interface	15
3.1.EtherCAT Basics.....	15
3.2.EtherCAT State Machine.....	15
3.3.CoE Interface.....	16
3.3.1.Parameter management in the EtherCAT system.....	16
4.Modbus Interface.....	18
4.1.Supported Modbus Function code	18
4.1.1.1 (0x01) Read Coils.....	18
4.1.2.2 (0x02) Read Discrete Inputs.....	19
4.1.3.3 (0x03) Read Holding Registers.....	19
4.1.4.4 (0x04) Read Input Registers.....	20
4.1.5.5 (0x05) Write Single Coil.....	21

4.1.6.6 (0x06) Write Single Register.....	21
4.1.7.8 (0x08) Diagnostics.....	22
4.1.8.15 (0x0F) Write Multiple Coils.....	24
4.1.9.16 (0x10) Write Multiple Registers.....	25
4.1.10.23 (0x17) Read/Write Multiple Registers.....	26
4.1.11.Error Response.....	27
4.2.MODBUS Special Register Map.....	28
4.2.1.Adapter Register Mapping.....	28
4.2.2.Adapter Identification Special Register (0x1000, 4096).....	28
4.2.3.Adapter Information Special Register (0x1100, 4352).....	29
4.2.4.Adapter Setting Special Register (0x1600, 5632).....	30
4.2.5.Expansion Slot Information Special Resister (0x2000, 8192).....	31
4.3.MODBUS Reference	32

History

Rev	Pages	Remarks	Date	Editor
1.00	-	Draft	May 03 , 2019	-
1.20	6	[M] Ethernet connection specification	Dec 30 , 2020	Minkyung, Park
1.30	6,7,8	[M] Specification, Ethernet connection, Number of max sockets	Mar 11 , 2021	Minkyung, Park
1.40	6	[A] Specifications (MQTT)	Jun 22 , 2021	Minkyung, Park
1.41	5	[M] Environment specification	Nov 17 , 2021	Minkyung, Park
2.00	-	[U] Codesys Version-up (3.5.11.3 → 3.5.17.3)	Apr 08 , 2022	Minkyung, Park
2.01	6, 7	[M] Non-Volatile Memory and Battery description	Jan 09 , 2023	Minkyung, Park
2.02	5, 6	[M] UL Temperature specification, User Management (supporting) Non-Volatile Memory Size	Jan 27 , 2023	Minkyung, Park
2.03	6,11,25	[M] Supplement SNMP contents, Exception indication, 110D content	Jun 27 , 2023	Minkyung, Park
2.04	22	[A] Restart Communications Option (program reset)	Aug 16 , 2023	Minkyung, Park
2.05	30	[A] Modbus 7bit Setting	Nov 02 , 2023	Minkyung, Park
2.06	6	[A] Additional Explanation for Online Changes	Mar 07 , 2024	Minkyung, Park
2.07	30	[A] - Modbus registers (0x1634, 0x1635, 0x1636)	Aug 20 , 2025	Minkyung, Park

* [M]: Modify, [A]: Add, [U]: Update

1. Environment Specification

Environmental Specification	
Operation Temperature	-25°C~60°C
UL Temperature	-25°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, 4g
Industrial Emissions	EN61000-6-4/All : 2011
Industrial Immunity	EN 61000-6-2 : 2019
Installation Position	Vertical and horizontal installation is available
Product Certifications	CE, UL

2. G-Series Light Version Programmable I/O

2.1. Specification

Communication Interface Specification			
Items	Specification		
	GN-9481	GN-9482	GN-9483
Programming	CODESYS V3.5.17.3		
Program Memory	512 Kbytes	16 Mbytes	
Data Memory	96 Kbytes	16 Mbytes	
Non-Volatile Memory	4 Kbytes	12 Kbytes	
	Retain: 2 Kbytes	Retain:6 Kbytes	
	Persistent Retain: 2 Kbytes	Persistent Retain: 6 Kbytes	
Run-Time System	Multiple PLC Tasks		
Program Languages	IEC 61131-3 (LD, IL, ST, FBD, SFC)		
MQTT 1)	O		O
MQTT Sparkplug B	X		O
SSL/TLS	X		X
User management 2)	X		O
IIOT Library	X		X
SNMP (Agent Only) 3)	O		O
SNTP	O		O
OPC DA Server	X		O
OPC UA Server & Client	X		O
Online Change 4*)	X		O
Source Upload/Download	X		O
File system	X		O
File transmit	X		O
TFTP	X		O
SQL4CODESYS	X		O
Breakpoint	X		O
Weather Forecast	X		O
Webvisualization 5)	X	X	O
RTC 6)	Retain Time : < 15 day / Accuracy : < 2min/month (Status : fully recharged battery at room temperature)		
Max. Task	10		
Max. Cycle Task	10		
Max. Status Task	10		
Max. Data Size (In+Out)	Max 128Byte each slot		
Max. Expansion Module	63 Slots		
Process Time	0.0306us	0.1667usec	

1) MQTT does not support TLS.

2) Provide functionality in a limited form.

3) Only the standard format "RFC1213-MIB" is provided.

4) Online Chagne: re-downloads only the changed parts of an application that is already running on the controller without initializing variables.

*Precautions for Online Changes

Due to product characteristics, performing a re-download may impact PLC logic execution, causing delays.

Therefore, please proceed with "online change" only when the equipment is stable and in a safe state.

5) Webvisualization cannot be supported in Internet Explorer.

6) RTC (room temperature)

Battery charging time	Retain time	*** RTC Warning
4 hours	> 2 day	There are 2 operating problems when the battery is discharged. - Retain data is not save.(GN-9481) - RTC data is not stored and is the initial value.
12 hours	> 12 day	
16 hours	> 15 day	

- Recommend charging for at least 16 hours when the battery is discharge.
- Retain time may vary depending on temperature and environment.

Interface Specification			
Adapter Type		Master & Slave node (Modbus TCP , Modbus RTU), EtherCAT Slave node	
EtherCAT			
EtherCAT Protocol		EtherCAT Slave	
Interface Connector		RJ-45 Socket * 2pcs	
Baud rate		100Mbps	
Max. PDO		RxPDO	64ea
		TxPDO	64ea
Max. Size		Moduer	Adjusted based on the size of the IO
		NonModuler	256 Bytes
Max. Network Nodes		65,535	
Ethernet			
Interface Connector		RJ-45 Socket * 1pcs	
Baud rate		10/100Mbps, Auto-negotiation, Full Duplex	
Ethernet Protocol	GN-9481	Modbus/TCP, Modbus/UDP, SNTP, MQTT, HTTP (Web-Server), DHCP/BOOTP	
	GN-9482/83	Modbus/TCP, Modbus/UDP, SNTP, SNMP, MQTT, DHCP/BOOTP, HTTP (Webvisualization*, Web-Server), OPC-server	
Max. Socket	GN-9481	UDP: 16, TCP: 16	
	GN-9482/83	UDP: 16, TCP: 64	
Serial			
Interface Connector		D-Sub 9Pin Connector * 1pcs	
Serial Protocol		Modbus/RTU	
Serial Interface		RS232/RS485 (supporting Touch Panel)	
Baud rate		2400~115200 bps (Default: 115200 bps)	
General specification			
Indicator (6 LEDs)		MOD	1 Green/Red, Module Status
		NET	1 Green/Red, EtherCAT Status
		RUN	1 Green/Red, PLC Run/Stop Status
		IOS	1 Green/Red, Expansion I/O Module Status
		System	1 Green, System Power Status
		Field	1 Green, Field Power Status
Power Dissipation		75mA typical @ 24Vdc	
UL System Power		Supply voltage: 24Vdc nominal, Class 2	
System Power		Supply voltage: 24Vdc nominal	
		Supply voltage range: 15 ~ 30Vdc	
		Reverse polarity protection	
UL Field Power		Supply voltage: 24Vdc nominal, Class 2	
Field Power		Supply voltage: 24Vdc typical (Max. 30Vdc)	
Current Field Power Contact		Max. DC 10A	
Wiring		I/O Cable Max. 2.0mm2(AWG 14)	
Torque		0.8 Nm(7 lb-in)	
Current for Expansion Module		1.5A @ 5Vdc	
Isolation		System power to internal logic : Non-isolation	
		System power I/O driver : Isolation	
Weight		<167g	
Module Size		54mm x 99mm x 70mm	
Environment Condition		Refer to ‘1. Environment Specificaiton’	

2.2. Ethernet connection specification

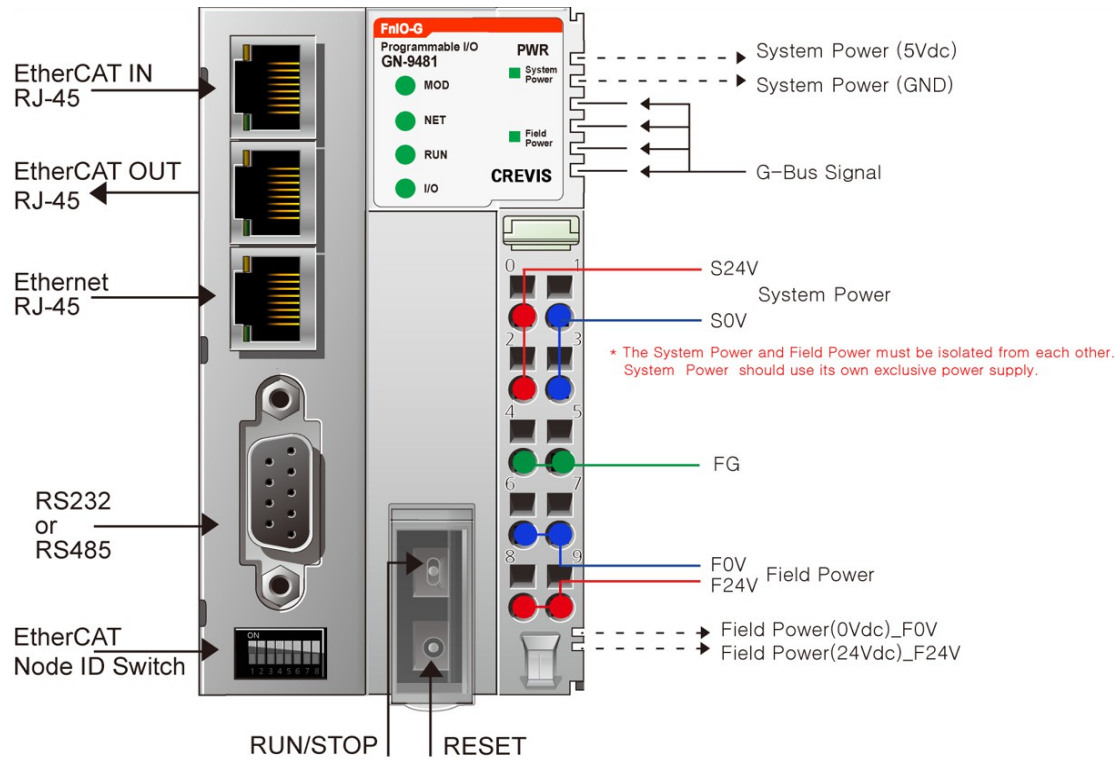
Function*	Model	Max. number of concurrent communications
Webvisualization	GN-9483	One for each functions are available at the same time
ARTI (OPC-server)	GN-9482/83	
CODESYS link	GN-9481/82/83	
Network-variable	GN-9481/82/83	
Modbus/TCP Master	GN-9481	16 Modbus/TCP Slaves can be connected
	GN-9482/83	64 Modbus/TCP Slaves can be connected
Modbus/TCP Slave	GN-9481	16 Modbus/TCP Masters can be connected
	GN-9482/83	64 Modbus/TCP Masters can be connected
EtherCAT Slave	GN-9481/82/83	
Web-server	GN-9481	16 clients can be opened
	GN-9482/83	64 clients can be opened

* While using these features, can use up to a maximum number of sockets (GN-9481: 16, GN-9482/83: 64) at the same time.

2.3. Serial connection specification

Function	Model	Max. number of concurrent communications
Modbus RTU Master	GN-9481/82/83	RS232: 1 Slaves can be connected
		RS485: 31 Slaves can be connected

2.4. Wiring Diagram



2.4.1. Power Connector

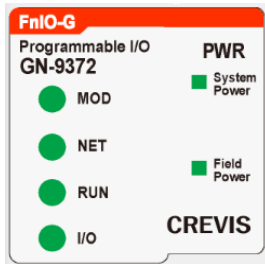
Pin No.	Signal Description	Pin No.	Signal Description
0	System Power, 24V	1	System Power, Ground
2	System Power, 24V	3	System Power, Ground
4	F.G	5	F.G
6	Field Power, Ground	7	Field Power 0V, Ground
8	Field Power, 24V	9	Field Power, 24V

* Warning

- The System Power and Field Power must be isolated from each other.
- System Power should use its own exclusive power supply.

2.5. LED Indicator

2.5.1. LED Indicator



LED No.	LED Description	LED Color
MOD	Module Status	Green/Red
NET	Network Status	Green/Red
RUN	PLC Status	Green/Red
IOS	Expansion IO Status	Green/Red
System Power	System Power Enable	Green
Field Power	Field Power Enable	Green

2.5.2. MOD (Module Status LED)

Status	LED is	To indicate
Not Powered	Off	Device has no power supplied.
Normal operation	Green	The device is operating normally.
Diagnostic	Red	Stack memory over flow or Assertion error.
	Red Blinking	Eeprom or file system error.

2.5.3. NET (Network Status LED)

Status	LED is	To indicate
Initialize	Off	Non-Operating or Initialize
Operate	Green	Operating
	Green Blinking	Pre-Operating LED ON: 200ms / LED OFF: 200ms
		Safe-Operating LED ON: 200ms / LED OFF: 1s
Error	Red	Network Error

2.5.4. RUN (PLC Run/Stop Status LED)

Status	LED is	To indicate
No PLC Program	Off	Device has no program.
PLC Run	Green	The PLC program is in the running state.
PLC Stop	Green Blinking	The PLC program is in a stopped state.
Codesys Task Watchdog	Red	Codesys task watchdog has occurred.
Diagnostic	Red Blinking	PLC program and expansion I/O modules do not match.

2.5.5. IOS (Expansion Module Status LED)

Status	LED is	To indicate
No Expansion I/O	OFF	Device has no expansion modules.
Have Expansion I/O	Green	Device has expansion modules.
Configuration Fault	Red	Replace expansion modules or fail to initialize. - Detect invalid expansion module ID. - Initial protocol failure. - Mismatch vendor code between adapter and expansion module. - Changed expansion module configuration.
Connection Fault	Red Blinking	One or more expansion module occurred in fault state. - Exceeded number of expansion modules. - Communication failure. - I/O size overflow.

2.5.6. Exception Indication

To indicate	LED			
	MOD	NET	RUN	IOS
Booting 1)	Green Blinking	Green Blinking	-	-
IAP Mode 2)	Green/Red Toggle	-	Off	Off
Program reset	-	-	Green/Red Toggle (every 0.25s)	-
Factory reset	Green/Red Toggle (every 0.25s)			
IO Watchdog error	Red	-	Red	-
CODESYS License error	-	-	Green/Red Toggle (every 2s)	-
Heap memory over flow	Red	Red	-	-
Hard Fault	Red			

'-' : Current LED status

1) BOOTP/DHCP requests a new IP address. (You can change the IP setting mode. Refer to Modbus Register 0x160B.)

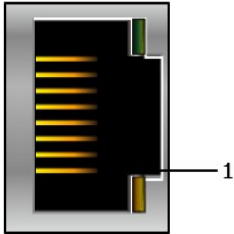
2) The IP Address to access IAP web-server during IAP Mode: 192.168.100.10 (Recommended to use FireFox)

2.5.7. Field Power LED

Status	LED is	To indicate
Not supplied field power	OFF	Not supplied 24Vdc field power.
Supplied field power	Green	Supplied 24Vdc field power.

2.6. Electrical Interface

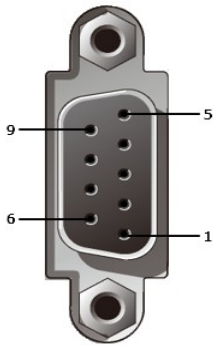
2.6.1. RJ-45 Socket



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

2.6.2. D-Sub 9Pin

* RS-232/RS-485 Port for MODBUS/RTU, Touch Panel or IO Guide



Pin Number	Signal Name	Description
1	-	
2	TXD	RS232 TXD
3	RXD	RS232 RXD
4	-	
5	GND	RS232 GND
6	D+	RS485 D+
7	-	
8	D-	DS485 D-
9	/ISP	Low Active, Internal Pull-up

2.6.3. Toggle Switch

Toggle Switch	Signal Name	Description
UP	RUN	PLC Run
DOWN	STOP	PLC Stop

2.6.4. Push Switch

Push Switch	Module is	Description
Push < 5sec	Stop State	Transition to PLC stop state.
Push > 5sec	PLC Reset	The PLC program and retain memory will be erased.
Push > 20sec	Factory Reset	The PLC program and parameters will be erased.
Push hold and Power Reset	IAP mode	Firmware download. (Recommended to use FireFox.)

2.6.5. Dip Switch



Dip Switch	Description
1 ~ 8	EtherCAT Node Setting

2.7. Web-Server

2.7.1. Web-Server Address

: <http://Network Adapter IP/setup.htm>



www.crevis.co.kr

[Network Adapter](#)

[Expansion Module](#)

[CodeSys PLC](#)

[Network Setting](#)

Crevis FnIO The Creative present makes Vision of future

[Network Adapter](#)
GN-9481(Programmable IO)

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

- IP Address : 192.168.100.40
- Subnet Mask : 255.255.255.0
- Gateway : 192.168.100.254
- MAC Address : 00:14:F7:00:5A:28

- MODBUS/TCP Connections : Available
- MODBUS/UDP Connections : Available
- CODESYS/UDP Connections : Available
- HTTP(Web Server) Connections : Available
- MODBUS/RTU(RS232) Communication : Available
- MODBUS/RTU(RS485) Communication : Available

- Firmware Revision : 1.000(01/30/2019)
- Expansion Modules : 2 module(s)
- IO Size(Input) : 4 byte(s)
- IO Size(Output) : 4 byte(s)

- CODESYS(IEC61131-3) V3.5 SP11 PLC : Available

2.7.2. Expansion Module

: Provide the expansion modules that attached to Network Adapter



www.crevis.co.kr

[Network Adapter](#)

[Expansion Module](#)

[CodeSys PLC](#)

[Network Setting](#)

Crevis FnIO The Creative present makes Vision of future

[Network Adapter](#)
GN-9481(Programmable IO)

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

Slot#	Descriptions	Input Reg. Mapping	Output Reg. Mapping
Slot#01	GT-22CA, 32DO, 24Vdc, Source		0x0800/0 (4byte)
Slot#02	GT-12FA, 32DI, 24Vdc, Universal	0x0000/0 (4byte)	

2.7.3. Codesys PLC

: Provide Codesys PLC information and current RTC time. RTC time can be changeable in this page

 www.crevis.co.kr	Crevis FnIO The Creative present makes Vision of future
Network Adapter	Network Adapter
Expansion Module	GN-9481(Programmable IO)
CodeSys PLC	Io Input Data / Io Output Data
Network Setting	

- Vendor Name : "Crevis Co., Ltd"

- Vendor ID : 0x10AD

- Device ID : 0x1003

- Device Type : 0x1000

PLC Logic : ""

- Project Name : ""

- Author : ""

- Version : ""

- Description : ""

- Profile : ""

- Last Updated Time :

- Switch(Run/Stop) : Run

- PLC Status : Stop

- Current RTC Date: 2018-08-13 Time: 14:38:10

Enter RTC: (Please follow the date and time format)

- Date: Time:

Click Button if you want to get Current time from PC

2.7.4. Network Setting

: Provide current IP configuration. IP parameter can change in this page.

 www.crevis.co.kr	Crevis FnIO The Creative present makes Vision of future
Network Adapter	Network Adapter
Expansion Module	GN-9481(Programmable IO)
CodeSys PLC	Io Input Data / Io Output Data
Network Setting	

Current IP Configuration

- IP Address : 192.168.100.40

- Subnet Mask : 255.255.255.0

- Gateway : 192.168.100.254

- MAC Address : 00:14:F7:00:5A:28

Change IP Parameter

- IP address:

- Subnet mask:

- Gateway:

* DO NOT FORGET the new IP configuration before power reset! *

* Please write down the addresses before you forget it! *

Click Button if you want to reset power & use new IP parameters

3. EtherCAT Interface

3.1. EtherCAT Basics

The EtherCAT protocol uses an officially assigned EtherType inside the Ethernet Frame. The use of this EtherType allows transport of control data directly within the Ethernet frame without redefining the standard Ethernet frame. The frame may consist of several sub-telegrams, each serving a particular memory area of the logical process images that can be up to 4 gigabytes in size. Addressing of the Ethernet terminals can be in any order because the data sequence is independent of the physical order. Broadcast, Multi-cast and communication between slaves are possible.

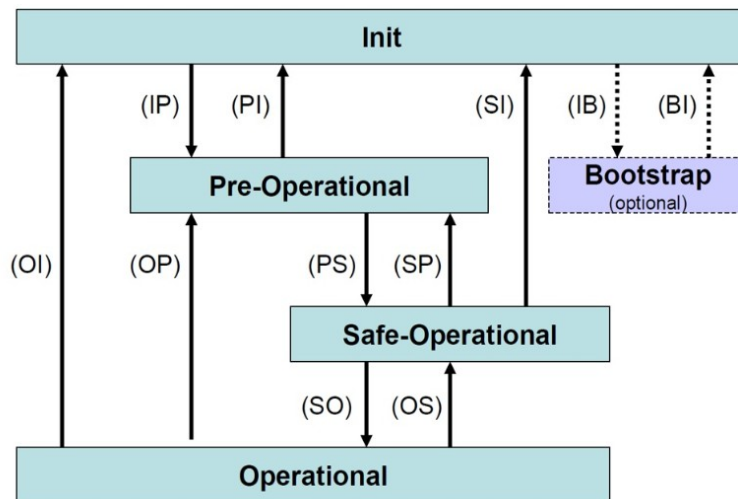
3.2. EtherCAT State Machine

The state of the EtherCAT slave is controlled via the EtherCAT State Machine (ESM). Depending upon the state, different functions are accessible or executable in the EtherCAT slave. Specific commands must be sent by the EtherCAT master to the device in each state, particularly during the boot up of the slave.

A distinction is made between the following states:

- Init
- Pre-Operational
- Safe-Operational
- Operational
- Bootstrap

The regular state of each EtherCAT slave after bootup is the OP state.



■ Init

After switch-on the EtherCAT slave in the Init state. No mailbox or process data communication is possible. The EtherCAT master initializes sync manager channels 0 and 1 for mailbox communication.

■ Pre-Operational (Pre-Op)

During the transition between Init and Pre-Op the EtherCAT slave checks whether the mailbox was initialized correctly. In Pre-Op state mailbox communication is possible, but not process data communication. The EtherCAT master initializes the sync manager channels for process data (from sync manager channel 2), the FMMU channels and, if the slave supports configurable mapping, PDO mapping or the sync manager PDO assignment. In this state the settings for the process data transfer and perhaps terminal-specific parameters that may differ from the default settings are also transferred.

■ Safe-Operational (Safe-OP)

During transition between Pre-Op and Safe-Op the EtherCAT slave checks whether the sync manager channels for process data communication and, if required, the distributed clocks settings are correct. Before it acknowledges the change of state, the EtherCAT slave copies current input data into the associated DP-RAM areas of the EtherCAT slave controller (ECSC).

In Safe-Op state mailbox and process data communication is possible, although the slave keeps its outputs in a safe state, while the input data are updated cyclically.

■ Operational (Op)

Before the EtherCAT master switches the EtherCAT slave from Safe-Op to Op it must transfer valid output data.

In the Op state the slave copies the output data of the masters to its outputs. Process data and mailbox communication is possible.

■ Bootstrap

In the Boot state the slave firmware can be updated. The Boot state can only be reached via the Init state.

In the Boot state mailbox communication via the file access over EtherCAT (FoE) protocol is possible, but no other mailbox communication and no process data communication.

3.3. CoE Interface

3.3.1. Parameter management in the EtherCAT system

The CiA organization (CAN in Automation) pursues among other things the goal of creating order and exchange ability between devices of the same type by the standardization of device descriptions. For this purpose so-called profiles are defined, which conclusively describe the changeable and unchangeable parameters of a device. Such a parameter encompasses at least the following characteristics:

- ✓ Index number – for the unambiguous identification of all parameters. The index number is divided into a main index and a subindex in order to mark and arrange associated parameters.
 - Main index
 - Subindex, offset by a colon ':'
- ✓ Official name – in the form of an understandable, self-descriptive text
- ✓ Specification of changeability, e.g. whether it can only be read or can also be written
- ✓ A value – depending upon the parameter the value can be a text, a number or another parameter index.

■ Index Range

The relevant ranges for EtherCAT fieldbus users are:

x1000 : This is where fixed identity information for the device is stored, including name, manufacturer, serial number etc., plus information about the current and available process data configurations.

x8000 : This is where the operational and functional parameters for all channels are stored, such as filter settings or output frequency.

Other important ranges are:

x4000 : In some EtherCAT devices the channel parameters are stored here (as an alternative to the x8000 range).

x6000 : Input PDOs ("input" from the perspective of the EtherCAT master)

x7000 : Output PDOs ("output" from the perspective of the EtherCAT master)

3.3.2. Communication Objects

*This value can be changed depending on the configuration of expansion modules

** GBUS Status

00	Normal Operation	02	Communicaiton Fault	03	Configuration Failed	04	No Expansion Module
07	Vendor Error	08	Not Expected Slot	09	CRC Error		

Index	Sub-index	Name	Flags	Default value
1000		Device type	RO	0x00001389
1001		Gbus Status	RO	Normal Operation : 0x00**
1002		Master Fault Aaction	RW	0x00
1008		Device name	RO	GN-948x(Crevis)
1009		Hardware version	RO	GN-948x.v1
100A		Software version	RO	1.000
1018		Identity	RO	0x05
	01	Vendor ID (Crevis: 029D)	RO	0x0000029D
	00	Moduler Type Product code	RO	0x474E948x
	00	Non-Moduler Type Product code	RO	0x474F948x
	03	Revision	RO	0x0001000
	04*	Serial number	RO	0xFFFFFFFF
	05	Release date	RO	0x20160823
10F1		Error Settings	RO	0x02
	01	Local Error Reaction	RO	0x00000000
	02	Sync Error Counter Limit	RO	0x00000004
1601*		Slot#x, GT--xxxx,RXPDO	RO	0xnn
	01	SubIndex 001	RO	0x7010:01, 8
	...	...	...	...
	nn	SubIndex nnn	RO	0x7010:nn, 8
1A01*		Slot#x, GT-xxxx, TXPDO	RO	0xnn
	01	SubIndex 001	RO	0x6010:01, 8
	...	...	...	...
	nn	SubIndex nnn	RO	0x6010:nn, 8
1C00		Sync manager type	RO	0x04
	01	SubIndex 001	RO	0x01
	02	SubIndex 002	RO	0x02
	03	SubIndex 003	RO	0x03
	04	SubIndex 004	RO	0x04
1C12		RxPDO assign	RO	0x01
	01	SubIndex 001	RO	0x1601
1C13		TxPDO assign	RO	0x02
	01	SubIndex 001	RO	0x1A01
	02	SubIndex 002	RO	0x1A02
6010*		GT-xxxx(Input)	RO	0xnn
	01	Byte#0	RO	0x00
	...	...	...	...
	nn	Byte#nnn	RO	0x00
7010*		GT-xxxx(Output)	RO	0xnn
	01	Byte#0	RW P	0x00
	...	...	...	...
	nn	Byte#nnn	RW P	0x00
8000		GN-948x(Parameter)	RO	
	01	Byte#0	RW	
	02	Byte#1	RW	
	03	Byte#2	RW	
	04	Byte#3	RW	
8nn0*		GT-xxxx(Parameter)	RO	
	01	Byte#0	RW	
	...	...	...	
	nn	Byte#nnn	RW	
F000		Module device profile	RO	
	01	Module index distance	RO	
	02	Maximum numver of modules	RO	
F010*		Module List	RO	
	01	Subindex 001 (GN-948x)	RO	0x0000948x
	...	...	...	...
	63	Subindex 063	RO	0x0000xxxx
F050		Detected Module Ident List	RO	
	01...	SubIndex 001	RO	

4. Modbus Interface

4.1. Supported Modbus Function code

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

4.1.1. 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.

4.1.2. 2 (0x02) Read Discrete Inputs

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 0x0015~0x0000 input bit value: 00000000_10000000.

4.1.3. 3 (0x03) Read Holding Registers

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.

4.1.4. 4 (0x04) Read Input Registers

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	0x04
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.

4.1.5. 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.

4.1.6. 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.

4.1.7. 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+Sumcheck	Echo Request Data	Reset with Default Setting 1)
0x0001(1)	0x55AA+0xAB7B+Sumcheck 3)	Echo Request Data	Reset with Factory default 1) 2)

1) Delete PLC program.

2) IP, Subnet Mask, Gateway, RS232/485 setting, and Bootp/DHCP mode will be the factory defaults value.

3) Refer to 3.2.2 for Sumcheck (0x1006)

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

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
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 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, Expansion Status

The response data field returns the status of ModBus and expansion 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, G-Bus Status	Same as status 1 word

4.1.8. 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.

4.1.9. 16 (0x10) Write Multiple Registers

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.

4.1.10. 23 (0x17) Read/Write Multiple Registers

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.

4.1.11. 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.
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.

4.2. 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).

4.2.1. Adapter Register Mapping

Address	IEC Address	Contents
0x0000~0x07FF	%IW0~%IW2047	2048 words Input and Internal memory (Area is write-protected)
0x0800~0x0FFF	%QW0~%QW2047	2048 words Output and Internal memory (Area is write-enabled)
0x1000~0x1FFF	-	Special Function Register (PIO Information)
0x2000~0x2FFF	-	Special Function Register (Slot Information)
0x4000~0x5FFF	%MW0~%MW8191	8192 words Internal memory (Area is write-enabled)

4.2.2. Adapter Identification Special Register (0x1000, 4096)

Address	Access	Type, Size	Description
0x1000(4096)	Read	1word	Vendor ID = 0x029D(669), Crevis. Co., Ltd.
0x1001(4097)	Read	1word	Device type = 0x000C, Network Adapter
0x1002(4098)	Read	1word	Product Code = 0x9130(GN-9481)/0x9140(GN-9482)/0x9150(GN-9483)
0x1003(4099)	Read	1word	Firmware revision, if 0x0101, revision 1.001
0x1005(4101)	Read	String upto 34bytes	First 1 word of the product name string represents the length of the string. Example) response as following "00 1C 47 4E 2D 39 34 38 31 28 50 49 4F 29 00" Valid character size = 0x0017 =29 characters "GN-9481(PIO)"
0x1006(4102)	Read	1word	Sum check of EEPROM
0x1010(4112)	Read	2words	Firmware release date
0x1013(4115)	Read	1word	Module ID = 0x9481(GN-9481)
0x101E(4126)	Read	15words	Composite Id of following address 0xA8C0(Lo_IP Addr),0x3264(Hi_IP Addr),0xFFFF(Lo_NetMask), 0x00FF(Hi_NetMask),0xA8C0(GateWay),0xFE64(GateWay), 0x1400(MacAddr),0x00F7(MacAddr),0xBA83(MacAddr), 0x02E5(VendorCode),0x000C(DeviceType),0x91F0(ProductCode), 0x0203(FW Rev),0x0510(FW ReleasData),0x2021(FW ReleasYear)

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

4.2.3. Adapter Information Special Register (0x1100, 4352)

Address	Access	Type, Size	Description
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.
0x110D(4365)*	Read	1word	Field Power State & Switch State (Dip Switch, Run, Stop, Reset)
0x110E(4366)	Read	upto 63words	Expansion slot's GT-number If the PIO is connected with GT-222F and GT-123F, then 0x222F 0x123F
0x1110(4368)	Read	1word	Number of expansion slot.
0x1113(4371)	Read	upto 63words	Expansion slot module id. First 1word is product code. Product Code = 0x9130(GN-9481)/0x9140(GN-9482)/0x9150(GN-9483)
0x111E(4382)	Read	1word	Reserved. Adapter IO identification vendor code.

*Address: 110D

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Power	-	-	-	-	-	DIP8	DIP7	DIP6	DIP5	DIP4	DIP3	DIP2	DIP1	RESET	RUN/STOP

4.2.4. Adapter Setting Special Register (0x1600, 5632)

Address	Access	Type, Size	Description
0x1600(5632)	Read	2words	IP Address (ex: A8C0 6464 = 192.168.100.100)
0x1602(5634)	Read	2words	Subnet Mask (ex: FFFF 0000 = 255.255.0.0)
0x1604(5636)	Read	2words	Gate way (ex: A8C0 6401 = 192.168.100.1)
0x1606(5638) 1)	Read /Write	1word	RS-232C Baud rate (2400bps~115200bps) - 0: 115200 (default)
0x1607(5639) 2)	Read /Write	1word	RS-232C Setting 1 nibble Data bit(0 : 8bit(default), 1 : 9bit, 2 : 7bit) 2 nibble Stop bit(0 : 1bit(default), 1 : 2bit) 3 nibble Parity bit(0 : none(default), 1: odd, 2 : even) 4 nibble Reserve
0x1608(5640) 1)	Read /Write	1word	RS-485 Baud rate. (2400bps~115200bps) - 0: 115200 (default)
0x1609(5641) 2)	Read /Write	1word	RS-485 Setting 1 nibble Data bit(0 : 8bit(default), 1 : 9bit, 2 : 7bit) 2 nibble Stop bit(0 : 1bit(default), 1 : 2bit) 3 nibble Parity bit(0 : none(default), 1: odd, 2 : even) 4 nibble Reserve
0x160A(5642) 3)	Read /Write	1word	Modbus Station High 1byte Station No. of RS-232C (default : 1) Low 1byte Station No. of RS-485 (default : 1)
0x160B(5643)	Read /Write	1word	IP Setting Method. BootP/DHCP 0x0000 disable BootP 0x8000 (default) DHCP 0x8001
0x1610(5648)	Read	3words	Mac Address (ex: 1400 00F7 0101 = 00.14.F7.00.01.01)
0x1614(5652)	Read	1word	Serial connection method 0x0000 Crevis Modbus/RTU (default) 0x8000 RS232 Enable for CODESYS Function block / RTU Master CODESYS Serial Port Configuration Setting: COM Port 1 0x8001 RS485 Enable for CODESYS Function block / RTU Master CODESYS Serial Port Configuration Setting: COM Port 2
0x1616(5654)	Read /Write	1word	EtherCAT Type Setting 0: Moduler Type(default), 1: Non-Moduler Type
0x1617(5655)	Read	1word	Non-Moduler Input Size
0x1618(5656)	Read	1word	Non-Moduler Output Size
0x1619(5657)	Read	1word	EtherCAT ID (0 ~ 255) / EtherCAT Node Setting Switch Data
0x1620(5664)	Read /Write	4words	RTC (ex : 0010 0F28 0317 07E0 = 2016 - 03.23 - 15:40 - 16) 1 word 00ss (ss : sec) 2 word hhmm (hh : hour, mm : min) 3 word mmdd (mm : month, dd : day) 4 word yyyy (yyyy : year)
0x1634(5684)	Read /Write	1word	WebServer Setting - 0: Disable, 1: Enable (default) - The settings will be applied after restarting the system
0x1635(5685) 4)	Read /Write	1word	Ethernet Port Setting (0: Disable, 1: Enable (default)) Read High 1byte: User-defined Values Low 1byte: Current Port Status Write 0: Disable, 1: Enable

1) RS-232C/485 Baud rate Setting

Value	Baud rate	Value	Baud rate	Value	Baud rate	Value	Baud rate
0	115200 (default)	2	4800	4	19200	6	57600
1	2400	3	9600	5	38400	7	115200

2) RS-232C/485 Setting : Explanation of the registers with addresses 0x1607 and 0x1609.

MSB											LSB				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data Bit				Stop Bit				Parity Bit				Reserved			

3) Modbus Station : Explanation of the registers with addresses 0x160A.

MSB																LSB
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RS-232C Modbus station								RS-485 Modbus station								

4) Ethernet Port Control

: Since the device is equipped with a single Ethernet port, the port status must be changed from Disabled to Enabled using one of the following methods

- Access the designated address via a serial connection and enable the port.
- Perform a factory reset.

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

Each expansion slot has the same information structure as the address offset.

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)
+ 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)	0x2019(8217)	0x2039(8249)	0x2059(8281)	0x2079(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	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 word	Read input data this slot
+ 0x0B(+11)	Read/ Write	n word	Read/write output data this slot
+ 0x0E(+14)	Read	1 word	GT-number, if GT-22CA, returns 0x22CA
+ 0x0F(+15)	Read	String upto 72bytes	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	1 word	Size of configuration parameter byte
+ 0x11(+17)	Read/ Write	n word	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 2, Rev 1.02)
+ 0x19(+25)	Read	2 words	Firmware release data.

4.3. MODBUS Reference

MODBUS Reference Documents	http://www.modubs.org
MODBUS Tools	http://www.modbustools.com , modbus poll http://www.win-tech.com , modscan32