FnIO G-Series

GN-9212

GN-9212 (DeviceNet Network Adapter)

Table of Contents

History

REV.	PAGES	REMARKS	DATE	Editor
1.00		Preliminary	July 28, 2016	DHLEE

1. Environment Specification

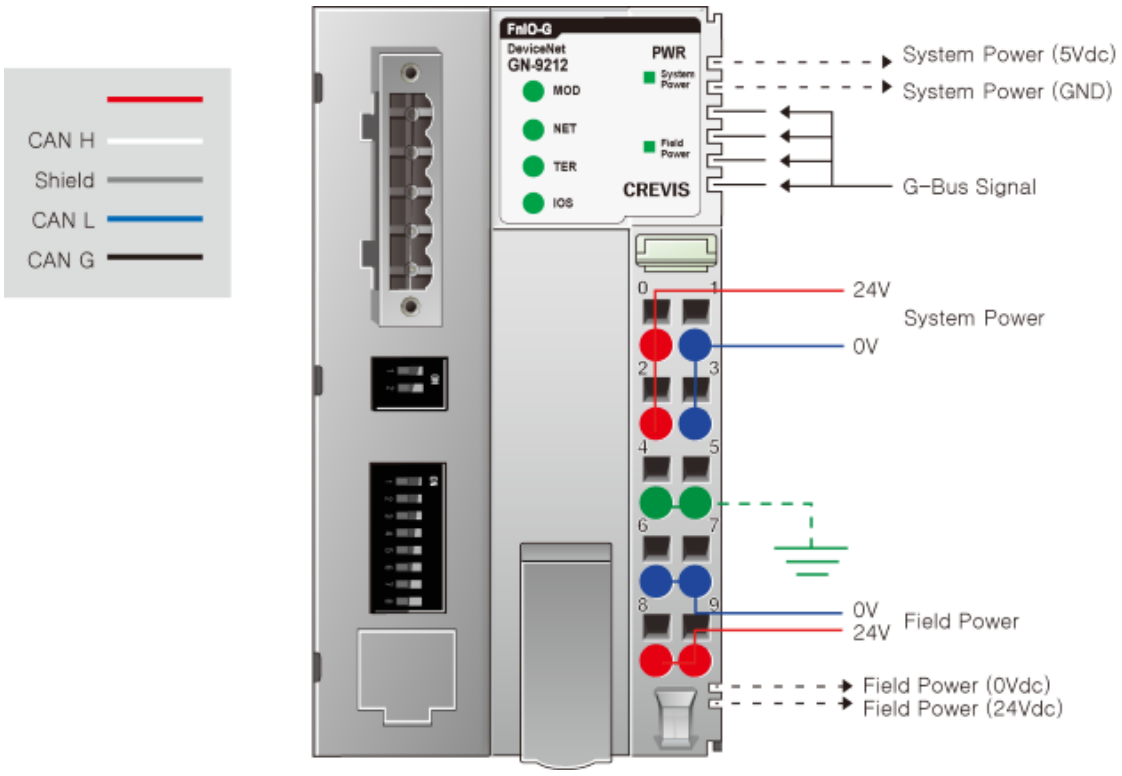
EnvironmentalSpecification	
Operating Temperature	60℃ ~ 70℃ : Power dissipation is limited to 0.8A. -40℃ ~ 60℃ : 1.5A full load is allowed.
UL Temperature	-20℃~60℃
Storage Temperature	-40℃~85℃
Relative Humidity	5% ~ 90% non-condensing
Mounting	DIN rail
General Specification	
Shock Operating	IEC 60068-2-27
Vibration Resistance	Sine Vibration (Based on IEC 60068-2-6) - 5 ~ 25Hz : ±1.6mm - 25 ~ 300Hz : 4g - Sweep Rate : 1 Oct/min, 20 Sweeps Random Vibration (Based on IEC 60068-2-64) - 10 ~ 40 Hz : 0.0125 g^2/Hz - 40 ~ 100 Hz : 0.0125 → 0.002 g^2/Hz - 100 ~ 500 Hz : 0.002 g^2/Hz - 500 ~ 2000 Hz : 0.002 → 1.3 x $10^{-4}g^2$/Hz - Test time : 1hrs for each test
Industrial Emissions	EN 61000-6-4/A11 : 2011
Industrial Immunity	EN 61000-6-2 : 2005
Installation Pos. / Protect. Class	Variable/IP20
Product Certifications	CE, UL(TBD)

2. GN-9212 (DeviceNet Network Adapter)

2.1. GN-9212 Specification

Items	Specification
Communication Interface Specification	
Adapter Type	Group 2 Only Slave
Max. Expansion Slot	63 Slots
I/O Data Size	Max 128 bytes each slot
Max. Length Bus Line	Max.100m@500Kbps, Max. 250m@250Kbps, Max. 500m@125Kbps
Max. Network Node	64 Nodes
Baud Rate	125Kbps(Max. 500m) 250Kbps(Max. 250m) 500Kbps(Max. 100m)
Protocol	Poll, Bit-Strobe, Cyclic, COS
Node MAC ID Setup	DIP Switch
Terminating Resistance Setup	DIP Switch
Bus Connection	5 Pin Open-Style Connector
Other Serial Port	RS232 for MODBUS/RTU, Touch Panel or IOGuide
Serial Configuration (RS232)	Node : 1 (Fixed) Baud Rate : 115200 (Fixed) Data bit : 8 (Fixed) Parity bit : No parity (Fixed) Stop bit : 1 (Fixed)
Indicator	6 Status LEDs 1 Green/Red, Module Status (MOD) 1 Green/Red, Network Status (NET) 1 Green, Terminating Resistance Status (TER) 1 Green/Red, Expansion I/O Module Statsus (IOS) 1 Green, System Power Status 1 Green, Field Power Status
Module Location	Starter module left side of G-Series system
Field Power Detection	About 14Vdc
General Specification	
System Power	Supply voltage : 24Vdc nominal Supply voltage range : 16~32Vdc Protection : Output current limit (Min. 1.5A) Reverse polarity protection
Power Dissipation	70mA @ 24Vdc
Current for I/O Module	1.5A @ 5Vdc
Isolation	System power to internal logic : Non-Isolation System power I/O driver : Isolation
Field Power	Supply voltage : 24Vdc typical (Max. 32Vdc) * Field Power Range is defferent depending on IO Module series. Refer to IO Module's Specification.
Max. Current Field Power Contact	DC 10A Max
Weight	154g
Module Size	54mm x 99mm x 70mm
Environment Condition	**Refer to '1. Environment Specification'**

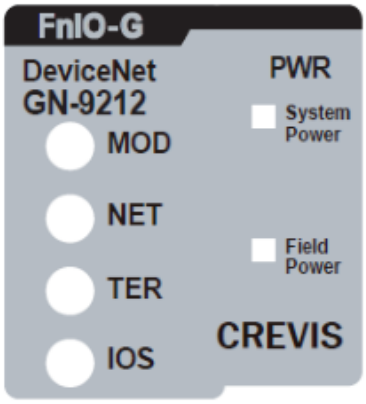
2.2. GN-9212 Wiring Diagram



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

2.3. GN-9212 LED Indicator

2.3.1. LED Indicator



LED No.	LED Function / Description	LED Color
MOD	Module Status	Green/Red
NET	Network Status	Green/Red
TER	Termination Resistance Status	Green
IOS	Extension module Status	Green/Red
System Power	System Power Enable	Green
Field Power	Field Power Enable	Green

2.3.2. MOD (Module Status LED)

Status	LED	To indicate
Not Powered	OFF	Power is not supplied to the unit.
Device Operational	Green	The unit is operating in normal condition.
Device in Standby	Flashing Green	EEPROM parameter is not initialized yet. Serial Number is zero value (0x00000000)
Minor Fault	Flashing Red	Device has an recoverable Fault.
Unrecoverable Fault	Red	Device has an unrecoverable fault.

2.3.3. NET (Network Status LED)

Status	LED	To indicate
Not Powered Not On-line	OFF	Device is not on-line or may not be powered. - Not completed Dup-MAC_ID test yet
On-line, Not connected	Flashing Green	Device is on-line but has no connections in the established state. - Passed Dup-MAC_ID test - Not allocated to a master
On-line, Connected	Green	Device is on-line and allocated to a master.
Connection Time-out	Flashing Red	One or more I/O connections are in the time-out state.
Critical Communication Failure	Red	Failed communication - Duplicate MAC ID - Bus-off

2.3.4. TER (Terminating Resistance LED)

Status	LED	To indicate
Not applied	OFF	Terminating resistance is not applied.
Applied	On	Terminating resistance is applied.

2.3.5. IOS LED (R-Series Internal Bus Status LED)

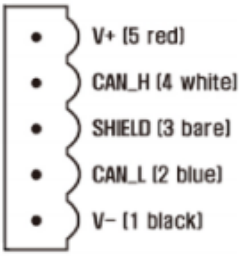
Status	LED	To indicate
Not Powered No Expansion Module	OFF	Device has no expansion module or may not be powered.
R-Series Internal Bus On-line, Do not Exchanging I/O	Flashing Green	R-Series Internal bus is normal but does not exchanging I/O data. (Passed the expansion module configuration)
R-Series Internal Bus Connection, Run Exchanging I/O	Green	Exchanging I/O data.
R-Series Internal Bus Connection Fault during Exchanging I/O	Red	One or more expansion module occurred in fault state. - Changed expansion module configuration. - Internal Bus communication failure.
Expansion Configuration Failed	RedFlashing Red	Failed to initalize expansion module. - Detect invalid expansion module ID. - Overflow Input/Output size. - Too many expansion module. - Initial protocol failure. - Mismatch vendor code between adapter and expension module.

2.3.6. Field Power, System Power LED (Field Power, System Power Status LED)

Status	LED	To indicate
Not supplied field, system power	OFF	Not supplied 24Vdc field power, 5Vdc system power.
Supplied field, system power	Green	Supplied 24Vdc field power, 5Vdc system power.

2.4. GN-9212 Electrical Interface

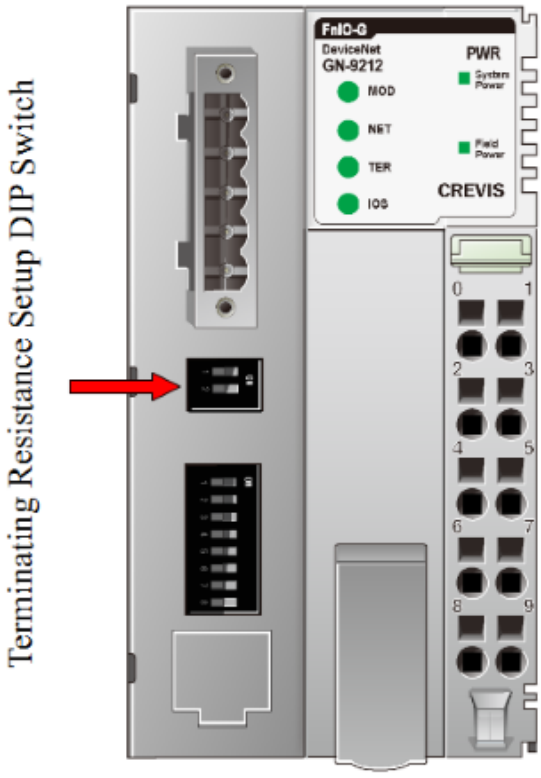
2.4.1. DeviceNet Open Connector



Pin		lor	Description
5*		d	System Power 24Vdc (16~32Vdc)
4		ıite	Transceiver High
3			Shield
2		ıe	Transceiver Low
1*	V-	Black	System Power 0Vdc

*Instead of RTB wiring for 24Vdc supply of the system power,
DeviceNet connector wiring is also available for the 24Vdc supply of the system power.
And when DeviceNet connector wiring is used for the system power,
RTB wiring for the system power should not be used together and vice versa.

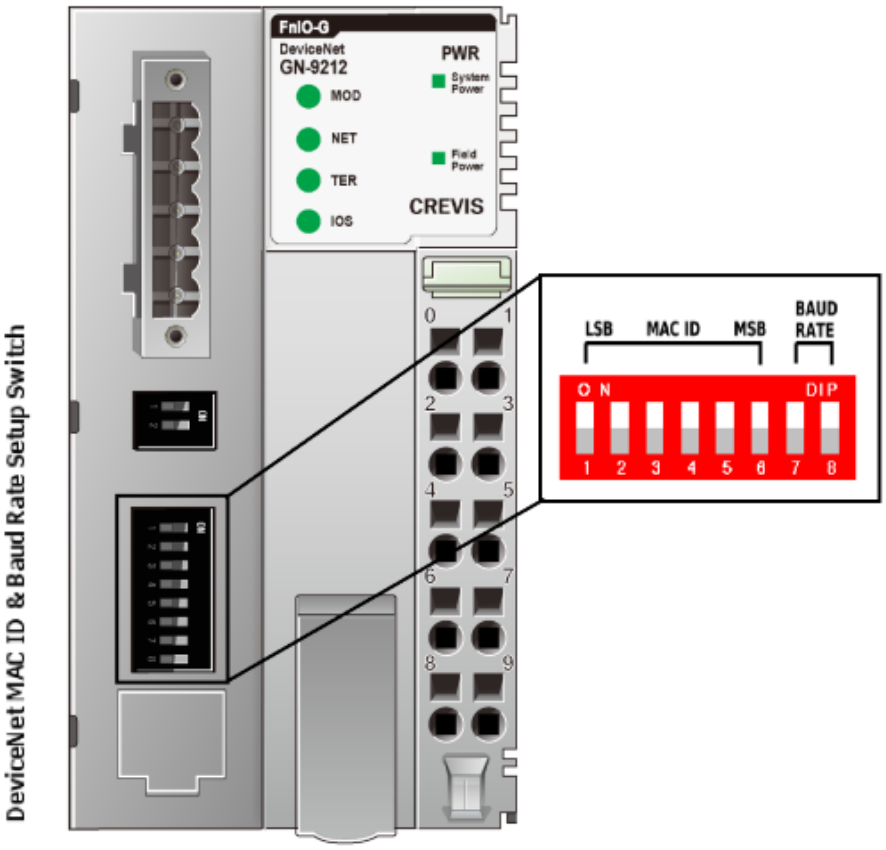
2.4.2. Terminating Resistance(112ohm) Setup



Terminating Resistance Switch	#1	#2
Applied	On	On
Not applied	Off	Off

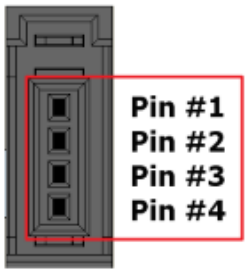
2.4.3. DeviceNet MAC ID & Buad Rate Setup

Each DeviceNet Adapter must have an unique MAC ID (from 0 to 63) so that it can be addressed independently from other nodes.



MAC ID	1	2	3	4	5	6	BAUD RATE	7	8
0	Off	Off	Off	Off	Off	Off	**125kbps**	Off	Off
1	On	Off	Off	Off	Off	Off	**250kbps**	On	Off
~							**500kbps**	Off	On
63	On	On	On	On	On	On	**AUTO**	On	On

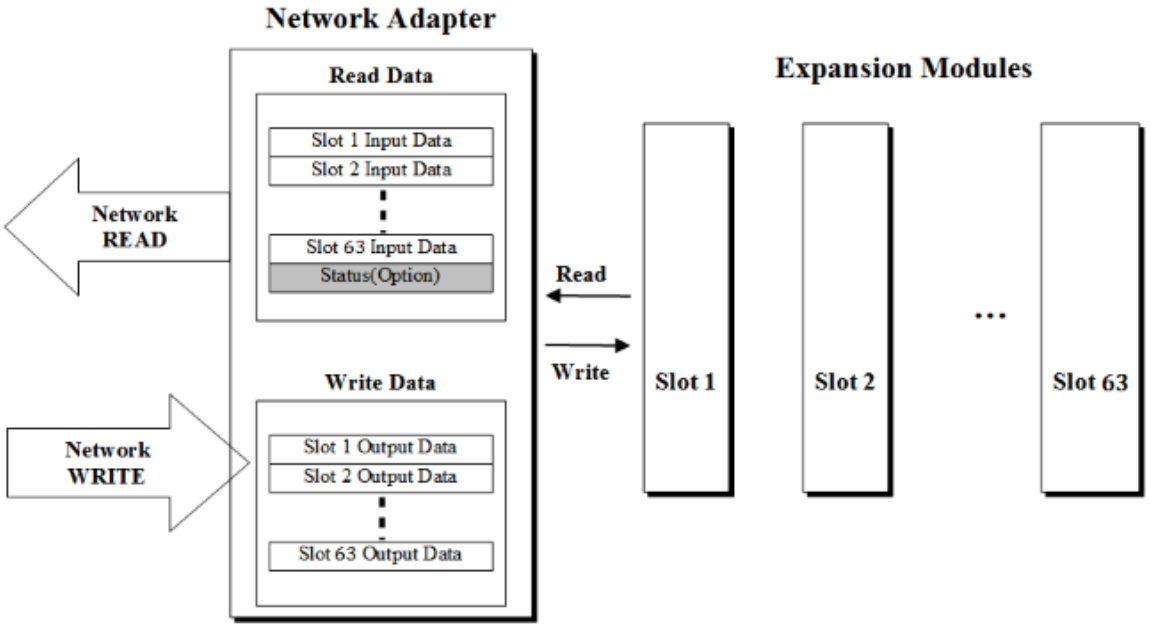
2.4.4. RS232 Port for MODBUS/RTU, Touch panel or IOGuide



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

2.5. 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 R-Series Internal Bus protocol. The following figure shows the data flow of process image between network adapter and expansion modules.



3. Object Models

A DeviceNet node is modeled as a collection of Objects. An Object provides an abstract representation of a particular component within a product. The realization of this abstract object model within a product is implementation dependent. In other words, a product internally maps this object model in a fashion specific to its implementation.

The objects and their components are addressed by a uniform addressing scheme consisting of:
Media Access Control Identifier (MAC ID), an integer identification value assigned to each node on a DeviceNet 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.
Supported Objects

- Device Type Number: $0C_{HEX}$ (Communications Adapter)

Name of Object	Type	Number of Instances	Class Code
Identity	Required	1	01_{HEX}
Message Router	Required	1	02_{HEX}
DeviceNet	Required	1	03_{HEX}
Assembly	Required	2	04_{HEX}
Connection	Required	4	05_{HEX}
Acknowledge Handler	Required	1	$2B_{HEX}$
R-Series Internal Bus Manager	Vendor-specific	1	70_{HEX}
Expansion Slot	Vendor-specific	1~63	71_{HEX}

Objects Behavior, Interface

Object	Behavior	Interface
Identity	Device identification, reset service	Message Router
DeviceNet	Configures port attributes	Message Router
Assembly	Defines I/O data format and concatenates configuration data	I/O Connection or Message Router
Connection	Contains the number of logical ports into or out-of the device	Message Router
Acknowledge Handler	Manage the reception of message acknowledgments	Message Router
R-Series Internal Bus Manager	Management functions for the R-Series Internal Bus	Message Router
Expansion Slot	Management functions for the expansion slot	Message Router

3.1. Object Setting

3.1.1. Identity Object

Class Code: 01_{HEX}

Common Services

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

Class Attributes
None

Instance Attributes

Instance ID	Attribute ID	Access Rule	Name	Data Size	Value
1	1	Get	Vendor ID	1 word	0x02E5 (741), Crevis. Co., Ltd.
	2	Get	Device Type	1 word	0x000C (Network Adapter)
	3	Get	Product Code	1 word	0x9020 (GN-9212)
	4	Get	Revision - Major - Minor	Structure of: 1 byte 1 byte	 1 ~ 9 01 ~ 255
	5	Get	Status	1 word	Defined in Spec (0x0005) *
	6	Get	Serial Number	2 word	Unique Number
	7	Get	Product Name - String Length - ASCII String	Structure of: 1 byte STRING	1A (26) "GN-9212_DeviceNet,G-Series"
	9	Get	Check Sum	1 word	EEPROM Checksum Code
	100(64h)	Get	I/O Main State	1 word	0x01: Init State 0x02: Idle State 0x03: Run State 0x04: Stop State 0x05: Fault State 0x06: Reset State 0x07: CRC error State
	Vendor-specific				
	102(66h)	Get	Firmware Code	1 byte	0x71
	103(67h)	Get	ODVA Conformance Test Revision	UINT	0x0A17 → "2002. 10. 22."
	104(68h)	Get	Firmware Release Date	UDINT	0xYYYYMMDD ex) 0x20160817 → 2016/08/17
	107(6Bh)	Get	Inspection Date	UDINT	0xYYYYMMDD

* Spec. = The CIP Networks library, ODVA

3.1.2. Message Router Object

Class Code: 02_{HEX}

Common Services
None

Class Attributes
None

Instance Attributes
None

3.1.3. DeviceNet Object

Class Code: 03HEX

Common Services

Service Code	Implemented for		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	No	Yes	Set_Attribute_Single
0x4B	No	Yes	Allocate_Master/Slave_Connection_Set
0x4C	No	Yes	Release_Master/Slave_Connection_Set

Class Attributes

Instance ID	Attribute ID	Access Rule	Name	Data Size	Value
0	1	Get	Revision	1 word	02, 00

Instance Attributes

Instance ID	Attribute ID	Access Rule	Name	Data Size	Value
1	1	Get/Set*	MAC ID	1 byte	0 ~ 63
	2	Get	Baud Rate**	1 byte	0=125K, 1=250K, 2=500K, 3=Auto
	3	Get/Set	Bus off Interrupt	BOOL	faulted node recovery,(0x01 : Enable)
	4	Get	Bus-Off Counter	USINT	0 ~ 255
	5	Get	Allocation Information - Allocation Choice - Master's MAC ID	Structure of: BYTE USINT	- Allocation Choice Defined in Spec. - Master's MAC ID 255: unallocated 0~63: Master MAC ID
	8	Get	MACID Switch Value	USINT	0 ~ 99 Actual value of Switch
	9	Get	Baud Rate Switch Value	USINT	0 : 125kbps 1 : 250kbps 2 : 500kbps 3 : Auto
	Vendor-specific				
	100(64h)	Get	Auto-Baud Action**	BOOL	0: Enabled (default) (Not allowed to set the Baud Rate from Network) 1: Disabled (Allowed to set the Baud Rate from Network)
	101(65h) (Reserved)	Get/Set	Quick Start	BOOL	0 : Normal Start-up 1 : Quick Start-up

*The MAC ID Switch value = 0~63: Not allowed to set the MAC ID from Network.
Behavior: Changed new MAC ID → Device will be restarted.
**Refer to 2.4.3 (DeviceNet MAC ID & Buad Rate Setup)

3.1.4. Assembly Object

Class Code: 04_{HEX}

Common Services

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

Class Attributes
None

Input Instance Attributes

Input/output Instance ID

Instance ID	Attribute ID	Access Rule	Name	Data Type	Value
100(64h)	3	Get	Input (Produced) Process Image Data	Array n BYTE	Input process current image data
150(96h)	3	Set/Get	Output (Consumed) Process Image Data	Array n BYTE	Output process current image data

3.1.5. Connection Object

Class Code: 05_{HEX}

Common Services

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

Class Attributes
None

Instance Attributes for Explicit Messaging Connection

Instance ID	Attribute ID	Access Rule	Name	Data Type	Value
1	1	Get	state	USINT	Defined in Spec * 0x03 : The connection has been validly/fully configured and the configuration has been successfully applied.
	2	Get	instance_type	USINT	0: Explicit Message
	3	Get	transportClass_trigger	BYTE	83_{HEX}
	4	Get	produced_connection_id	UINT	*0x040B : MAC ID=01, Message group 2, Message ID 3
	5	Get	consumed_connection_id	UINT	*0x040C : MAC ID=01, Message ID 4
	6	Get	initial_comm_characteristics	BYTE	21_{HEX}
	7	Get	produced_connection_size	UINT	0x0206 (=518)
	8	Get	consumed_connection_size	UINT	0x0206 (=518)
	9	Get/Set	expacted_packet_rate	UINT	2504 (default) Timer Resolution of 8msec
	12	Get/Set	watchdog_timeout_action	USINT	3 : Deferred Delete (default)
	13	Get	produced_connection_path_length	UINT	00, 00
	14	Get	produced_connection_path	Array of USINT	Empty
	15	Get	consumed_connection_path_length	UINT	00, 00
	16	Get	consumed_connection_path	Array of USINT	Empty

attribute 3 transport Class trigger = 0x83 → Direction=Server,
Production Trigger=IGNORED,
Transport Class = 3.
This is the value assigned to this attribute within the server end-point of an Explicit Messaging Connection

Instance Attributes for Poll I/O Connection

Instance ID	Attribute ID	Access Rule	Name	Data Type	Value
2	1	Get	State	USINT	Defined in Spec
	2	Get	instance_type	USINT	1: I/O Message
	3	Get	transportClass_trigger	BYTE	82_{HEX}
	4	Get	produced_connection_id	UINT	* 0x03C1 : MAC ID=01, Message ID=6, Unconnected Explicit Request Message
	5	Get	consumed_connection_id	UINT	* 0x040D : MAC ID=01, Message ID=5, Group 2 message Identifier
	6	Get	initial_comm_characteristics	BYTE	01_{HEX}
	7	Get	produced_connection_size	UINT	Followed by IO process image
	8	Get	consumed_connection_size	UINT	Followed by IO process image
	9	Get/Set	expacted_packet_rate	UINT	Timer Resolution of 8msec * 200(decimal)
	12	Get	watchdog_timeout_action	USINT	0: Time Out (default)
	13	Get	produced_connection_path_length	UINT	0 or 6
	14	Get	produced_connection_path	Array of USINT	
	15	Get	consumed_connection_path_length	UINT	0 or 6
	16	Get	consumed_connection_path	Array of USINT	

Instance Attributes for Bit-Strobe I/O Connection

Instance ID	Attribute ID	Access Rule	Name	Data Type	Value
3	1	Get	state	USINT	Defined in Spec
	2	Get	instance_type	USINT	1: I/O Message
	3	Get	transportClass_trigger	BYTE	82_{HEX}
	4	Get	produced_connection_id	UINT	*0x0381 : MAC ID=01, Message ID=14, Message group 1
	5	Get	consumed_connection_id	UINT	*0X0400 : MAC ID = 00, Message ID = 0, Message group 2
	6	Get	initial_comm_characteristics	BYTE	02_{HEX}
	7	Get	produced_connection_size	UINT	0x08
	8	Get	consumed_connection_size	UINT	0x08
	9	Get/Set	expacted_packet_rate	UINT	Timer Resolution of 8msec * 200
	12	Get	watchdog_timeout_action	USINT	0: Time Out (default)
	13	Get	produced_connection_path_length	UINT	0 or 6
	14	Get	produced_connection_path	Array of USINT	
	15	Get	consumed_connection_path_length	UINT	0 or 6
	16	Get	consumed_connection_path	Array of USINT	

Instance Attributes for COS I/O Connection (Acknowledged)

Instance ID	Attribute ID	Access Rule	Name	Data Type	Value
4	1	Get	State	USINT	Defined in Spec
	2	Get	instance_type	USINT	1: I/O Message
	3	Get	transportClass_trigger	BYTE	12_{HEX}
	4	Get	produced_connection_id	UINT	
	5	Get	consumed_connection_id	UINT	
	6	Get	initial_comm_characteristics	BYTE	1
	7	Get	produced_connection_size	UINT	Followed by IO Process image
	8	Get	consumed_connection_size	UINT	Followed by IO Process image
	9	Get/Set	expected_packet_rate	UINT	Timer Resolution of 8msec
	12	Get/Set	watchdog_timeout_action	USINT	0: Time Out (default)
	13	Get	produced_connection_path_length	UINT	0 or 6
	14	Get	produced_connection_path	Array of USINT	
	15	Get	consumed_connection_path_length	UINT	4
	16	Get	consumed_connection_path	Array of USINT	20 2B 24 01
	17	Get/Set	production_inhibit_time	UINT	00, 00

Instance Attributes for COS I/O Connection (Unacknowledged)

Instance ID	Attribute ID	Access Rule	Name	Data Type	Value
4	1	Get	State	USINT	Defined in Spec * 0x01 : Configuring
	2	Get	instance_type	USINT	1: I/O Message
	3	Get	transportClass_trigger	BYTE	10_{HEX}
	4	Get	produced_connection_id	UINT	* 0x0341 MAC ID : 01, Message ID=13, Message Group 1
	5	Get	consumed_connection_id	UINT	$0FFFF_{HEX}$
	6	Get	initial_comm_characteristics	BYTE	$0F_{HEX}$
	7	Get	produced_connection_size	UINT	Followed by IO Process image
	8	Get	consumed_connection_size	UINT	Followed by IO Process image
	9	Get/Set	expacted_packet_rate	UINT	Timer Resolution of 8msec
	12	Get/Set	watchdog_timeout_action	USINT	0: Time Out (default)
	13	Get	produced_connection_path_length	UINT	0 or 6
	14	Get	produced_connection_path	Array of USINT	
	15	Get	consumed_connection_path_length	UINT	0
	16	Get	consumed_connection_path	Array of USINT	Empty
	17	Get/Set	production_inhibit_time	UINT	00, 00

3.1.6. Acknowledge Handler Object

Class Code: 2B$_{HEX}$

Common Services

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

Class Attributes
None

Instance Attributes

Instance ID	Attribute ID	Access Rule	Name	Data Type	Value
1	1	Set	Acknowledge Timer	UNIT	Default: 10
	2	Get	Retry Limit	USINT	1
	3	Get	COS Producing Connection Instance	UINT	4

3.1.7. G-Series Internal Bus Manager Object

Class Code: 70HEX (112D)

Common Services

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

Class Attributes
None

Instance Attributes

Instance ID	Attribute ID	Access Rule	Name	Data Type	Value
1	1	Get	Number of Slot	USINT	(include deactivated slot)
	4	Get	External IDs	Array of 128 BYTE	See Table 5.1.
	10	Get	G-Series Internal Bus Status	USINT	0x03 : Run state 0x04 : Stop state 0x05 : Fault state 0x07 : CRC state
	11	Get	Input (Produced) Byte Size	UINT	IO input byte size
	12	Get	Output (Consumed) Byte Size	UINT	IO output byte size
	113	Get	Run-time fault code	DWORD	#0 : Gbus error count #1 : Gbus error code (Table 5.2.) #2 : Error slot number #3 : NA status
	150	Get	Firmware Revision	USINT	#0 : Major revision #1 : Minor revision

Table5.1. External IDs (=Expansion Module ID)

Byte	Description
0	Network Adapter Module External ID = 0x9212
1	External ID for slot position #1
2	External ID for slot position #2
...	...
62	External ID for slot position #62
63	External ID for slot position #63

Table5.2. Gbus error code

Byte	Description
0x00	Normal Operation
0x02	Connection Fault
0x03	Configuration Fault
0x04	No Expansion module
0x05	Invalid attribute value
0x06	Too much data
0x07	Vendor Error
0x08	Not expected slot
0x09	CRC error

3.1.8. Expansion Slot Object

Class Code: 71HEX (113D)

Common Services

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

Class Attributes
None

Instance Attributes

Instance ID	Attribute ID	Access Rule	Name	Data Type	Value
1~63 (Slot Address)	1	Get	Module External ID	USINT	IO Name = External ID (2Byte) ex) IO Name : GT-1238 = External ID : 0x1238
	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 attributes 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 attributes 2. If Output data length is zero, then return Empty.
	8	Get	Configuration Parameter Data length	USINT	Refer to Configuration Parameter document
	9	Get/Set	R/W Configuration Data	n Byte	Data array size defined by attributes 8.
	100	Get	Product Code	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 changed slot location, set default value automatically.

4. MODBUS Interface

4.1. 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

4.2. Supported MODBUS Function Codes

Function Code	Function	Description
1(0x01)	Read Coils (Read output bit)	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.
2(0x02)	Read Discrete Inputs (Read input bit)	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.
3(0x03)	Read Holding Registers (Read output word)	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.

4(0x04)	Read Input Registers (Read input word)	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.
5(0x05)	Write Single Coil (Write one bit output)	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.
6(0x06)	Write Single Register (Write one word output)	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.
8(0x08)	Diagnostics (Read diagnostic register) *Refer to the 4.2.1	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.
15(0x0F)	Write Multiple Coils (Write a number of output bits)	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.
16(0x10)	Write Multiple registers (Write a number of output words)	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.
23(0x17)	Read/Write Multiple registers (Read a number of input words /Write a number of output words)	Read a number of input words /Write a number of output words 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.

- Refer to MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1a

4.2.1. 8 (0x08) Diagnostics

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, 0xFF00	Echo Request Data	Reset Only

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

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

4.2.2. 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 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.

4.3. 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).

4.3.1. Adapter Identification Special Register (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 = 0x9020(GN-9212)
0x1003(4099)	Read	1word	Firmware revision, if 0x0101, revision 1.01
0x1004(4100)	Read	2word	Product unique serial number
0x1005(4101)	Read	String upto 34byte	Product name string (ASCII) "GN-9212 DeviceNet,G-Series"
0x1006(4102)	Read	1word	Sum check of EEPROM
0x1010(4112)	Read	2word	Firmware release date
0x1013(4115)	Read	1word	Firmware Code = 0x9212
0x101E(4126)	Read	7word - 1word - 1word - 1word - 1word - 1word - 2word	Composite Id of following address 0x1100(4352), Modbus RS232 Node. (Fixed 0x0001) 0x1000(4096), Vendor ID 0x1001(4097), Device type 0x1002(4098), Product code 0x1003(4099), Firmware revision 0x1004(4100), Product serial number

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

4.3.2. 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.
0x110A(4362)	Read	1word	Update time for cyclic data change (same as 0x1028)
0x110C(4364)	Read	1word	Field power status
0x110D(4365)	Read	1word	Current Dip Switch State and Field Power Status (MSB) ex) Dip SW(0x01), Field Power On = 0x8001
0x110E(4366)	Read	upto 33word	Expansion slot's GT-number including GN First 1word is adapter's number, if GN-9289, then 0x9289
0x1110(4368)	Read	1word	Number of expansion slot
0x1113(4371)	Read	upto 33word	Expansion slot Module Id. Refer to Appendix A.1 Product List. First 1word is adapter's module id.
0x1119(4377)	Read	1word	Hi byte is ModBus status, low byte is internal status. Zero value means 'no error'.
0x111D(4381)	Read	1word	Adapter G-Series Revision. If 0x013C, G-Series Revision is 1.60

*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).

4.3.3. Expansion Slot Information Special Register (0x2000, 8192)

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

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

Address Offset	Expansion Slot#1	Expansion Slot#2	Expansion Slot#3	Expansion Slot#4	……..	Expansion Slot#63
+ 0x00(+0)	0x2000(8192)	0x2020(8224)	0x2040(8256)	0x2060(8288)	…….	0x27C0(10176)
+ 0x01(+1)	0x2001(8193)	0x2021(8225)	0x2041(8257)	0x2061(8289)	…….	0x27C1(10177)
+ 0x02(+2)	0x2002(8194)	0x2022(8226)	0x2042(8258)	0x2062(8290)	…….	0x27C2(10178)
+ 0x03(+3)	0x2003(8195)	0x2023(8227)	0x2043(8259)	0x2063(8291)	…….	0x27C3(10179)
+ 0x04(+4)	0x2004(8196)	0x2024(8228)	0x2044(8260)	0x2064(8292)	…….	0x27C4(10180)
+ 0x05(+5)	0x2005(8197)	0x2025(8229)	0x2045(8261)	0x2065(8293)	…….	0x27C5(10181)
+ 0x06(+6)	0x2006(8198)	0x2026(8230)	0x2046(8262)	0x2066(8294)	…….	0x27C6(10182)
+ 0x07(+7)	0x2007(8199)	0x2027(8231)	0x2047(8263)	0x2067(8295)	…….	0x27C7(10183)
+ 0x08(+8)	0x2008(8200)	0x2028(8232)	0x2048(8264)	0x2068(8296)	…….	0x27C8(10184)
+ 0x09(+9)	0x2009(8201)	0x2029(8233)	0x2049(8265)	0x2069(8297)	…….	0x27C9(10185)
+ 0x0A(+10)	0x200A(8202)	0x202A(8234)	0x204A(8266)	0x206A(8298)	…….	0x27CA(10186)
+ 0x0B(+11)	0x200B(8203)	0x202B(8235)	0x204B(8267)	0x206B(8299)	…….	0x27CB(10187)
+ 0x0C(+12)	0x200C(8204)	0x202C(8236)	0x204C(8268)	0x206C(8300)	…….	0x27CC(10188)
+ 0x0D(+13)	0x200D(8205)	0x202D(8237)	0x204D(8269)	0x206D(8301)	…….	0x27CD(10189)
+ 0x0E(+14)	0x200E(8206)	0x202E(8238)	0x204E(8270)	0x206E(8302)	…….	0x27CE(10190)
+ 0x0F(+15)	0x200F(8207)	0x202F(8239)	0x204F(8271)	0x206F(8303)	…….	0x27CF(10191)
+ 0x10(+16)	0x2010(8208)	0x2030(8240)	0x2050(8272)	0x2070(8304)	…….	0x27D0(10192)
+ 0x11(+17)	0x2011(8209)	0x2031(8241)	0x2051(8273)	0x2071(8305)	…….	0x27D1(10193)
+ 0x12(+18)	0x2012(8210)	0x2032(8242)	0x2052(8274)	0x2072(8306)	…….	0x27D2(10194)
+ 0x13(+19)	0x2013(8211)	0x2033(8243)	0x2053(8275)	0x2073(8307)	…….	0x27D3(10195)
+ 0x14(+20)	0x2014(8212)	0x2034(8244)	0x2054(8276)	0x2074(8308)	…….	0x27D4(10196)
+ 0x15(+21)	0x2015(8213)	0x2035(8245)	0x2055(8277)	0x2075(8309)	…….	0x27D5(10197)
+ 0x16(+22)	0x2016(8214)	0x2036(8246)	0x2056(8278)	0x2076(8310)	…….	0x27D6(10198)
+ 0x17(+23)	0x2017(8215)	0x2037(8247)	0x2057(8279)	0x2077(8311)	…….	0x27D7(10199)
+ 0x18(+24)	0x2018(8216)	0x2038(8248)	0x2058(8280)	0x2078(8312)	…….	0x27D8(10200)
+ 0x19(+25)	0x2018(8217)	0x2038(8249)	0x2058(8281)	0x2078(8313)	…….	0x27D9(10201)
+ 0x1A(+26)	0x201A(8218)	0x203A(8250)	0x205A(8282)	0x207A(8314)	…….	0x27DA(10202)
+ 0x1B(+27)	0x201B(8219)	0x203B(8251)	0x205B(8283)	0x207B(8315)	…….	0x27DB(10203)
+ 0x1C(+28)	0x201C(8220)	0x203C(8252)	0x205C(8284)	0x207C(8316)	…….	0x27DC(10204)
+ 0x1D(+29)	0x201D(8221)	0x203D(8253)	0x205D(8285)	0x207D(8317)	…….	0x27DD(10205)
+ 0x1E(+30)	0x201E(8222)	0x203E(8254)	0x205E(8286)	0x207E(8318)	…….	0x27DE(10206)
+ 0x1F(+31)	0x201F(8223)	0x203F(8255)	0x205F(8287)	0x207F(8319)	…….	0x27DF(10207)

Address Offset	Access	Type, Size	Description
+ 0x02(+2) **	Read	1word	Input start register address of input image word this slot.
+ 0x03(+3) **	Read	1word	Input word's bit offset of input image word this slot.
+ 0x04(+4) **	Read	1word	Output start register address of output image word this slot.
+ 0x05(+5) **	Read	1word	Output word's bit offset of output image word this slot.
+ 0x06(+6) **	Read	1word	Input bit start address of input image bit this slot.
+ 0x07(+7) **	Read	1word	Output bit start address of output image bit this slot.
+ 0x08(+8) **	Read	1word	Size of input bit this slot
+ 0x09(+9) **	Read	1word	Size of output bit this slot
+ 0x0A(+10)**	Read	n word	Read input data this slot
+ 0x0B(+11)**	Read/Write	n word	Read/write output data this slot
+ 0x0E(+14)	Read	1word	GT-number, if GT-1238, returns 0x1238
+ 0x0F(+15)	Read	String upto 72byte	First 1word is length of valid character string. If GT-1238, returns "00 1E 52 54 2D 31 32 33 38 2C 20 38 44 49 2C 20 32 34 56 64 63 2C 20 55 6E 69 76 65 72 73 61 6C 00 00" Valid character size = 0x001E =30 characters, "GT-1238, 8DI, 24Vdc, Universal"
+ 0x10(+16)	Read	1word	Size of configuration parameter byte
+ 0x11(+17)**	Read/Write	n word	Read/write Configuration parameter data, up to 8byte. Refer to A.2 ***
+ 0x17(+23)	Read	2word	Firmware Revision ex) 0x00010010 (Major revision 1 /Minor revision 1, Rev 1.001)
+ 0x19(+25)	Read	2word	Firmware release date.

* After the system is reset, the new "Set Value" action is applied.
** Nothing of output, input, memory or configuration parameter corresponding slot returns Exception 02.