FnIO M - Series:

M9386

M9386 EtherCAT ID Type Network Adapter (Single Type)

Table of Contents

Table of Contents....2
History....4
1.ENVIRONMENT SPECIFICATION....5
2.M9386 (EtherCAT ID Type Network Adapter)....6
2.1.M9386 Specification....6
2.2.M9386 Wiring Diagram....7
2.3.M9386 LED Indicator....8
2.3.1.LED Indicator....8
2.3.2.MOD (Module Status LED)....8
2.3.3.RUN (Current Runnig Status LED)....8
2.3.4.ERR (Error State LED)....8
2.3.5.I/O (Extension Module Status LED)....8
2.4.M7001 LED Indicator....9
2.4.1.LED Indicator....9
2.4.2. RUN(RUN Status LED)....9
2.4.3. PRI(Primary Status LED)....9
2.4.4.ACT(Active Status LED)....9
2.4.5.Field Power LED (Field Power Status LED)....9
2.5.M9386 Electrical Interface....10
2.5.1.5 Pin open connector....10
2.5.2.Dip Switch....10
2.5.3.RS232 Port for MODBUS/RTU, Touch Panel or IO-Guide....10
2.6.EtherCAT ID Type Setup....11
2.6.1.Hot Connection On TwinCAT....11
3.EtherCAT Basics....13
3.1.EtherCAT State Machine....13
3.2.CoE Interface....14
3.2.1.parameter management in the EtherCAT system....14
3.2.2.Communication Objects
4.MODBUS INTERFACE....17
4.1.Process Image Map....17
4.1.1.MODBUS Interface Register/Bit Map....17
4.2.MODBUS Interface Register/Bit Map....18
4.3.Supported MODBUS Function Codes....18
4.4.MODBUS Transmission Mode....19

4.4.1.RTU Transmission Mode....19
4.4.2.ASCII Transmission Mode....19
4.4.3.1 (0x01) Read Coils....19
4.4.4.2 (0x02) Read Discrete Inputs....20
4.4.5.3 (0x03) Read Holding Registers....20
4.4.6.4 (0x04) Read Input Registers....21
4.4.7.5 (0x05) Write Single Coil....21
4.4.8.6 (0x06) Write Single Register....22
4.4.9.8 (0x08) Diagnostics....22
4.4.10.15 (0x0F) Write Multiple Coils....25
4.4.11.16 (0x10) Write Multiple Registers....26
4.4.12.23 (0x17) Read/Write Multiple Registers....27
4.4.13.Error Response....28
4.5.MODBUS Special Register Map....30
4.5.1.Adapter Identification Special Register (0x1000, 4096)....30
4.5.2.Adapter Hotswap Register (0x1060, 4192)....30
4.5.3.Adapter Information Special Register (0x1100, 4352)....31
4.5.4.Expansion Slot Information Special Resister (0x2000, 8192)....32
4.6.Supported MODBUS Function Codes....34

History

REV.	PAGES	REMARKS	DATE	Editor
-		Preliminary	2018/6/18	BS HA
1.00			2019/04/03	YM KIM
1.01	5,6, 39,40	Image, UL Spec, Torque, Hotswap Function Vibration, Product, ATEX certification	2020/05/25	CW SEO
1.02	37	Modify Firmware Revision	2020/10/29	CW SEO
1.03		Remove Description pages of Hot Swap Function, Use in Hazardous Envirnments and Caution(Before using the unit)	2020/12/07	SJ LIM

1. ENVIRONMENT SPECIFICATION

Environmental specification	
Operating Temperature	-25℃~60℃
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	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, ABS, DNV, ATEX

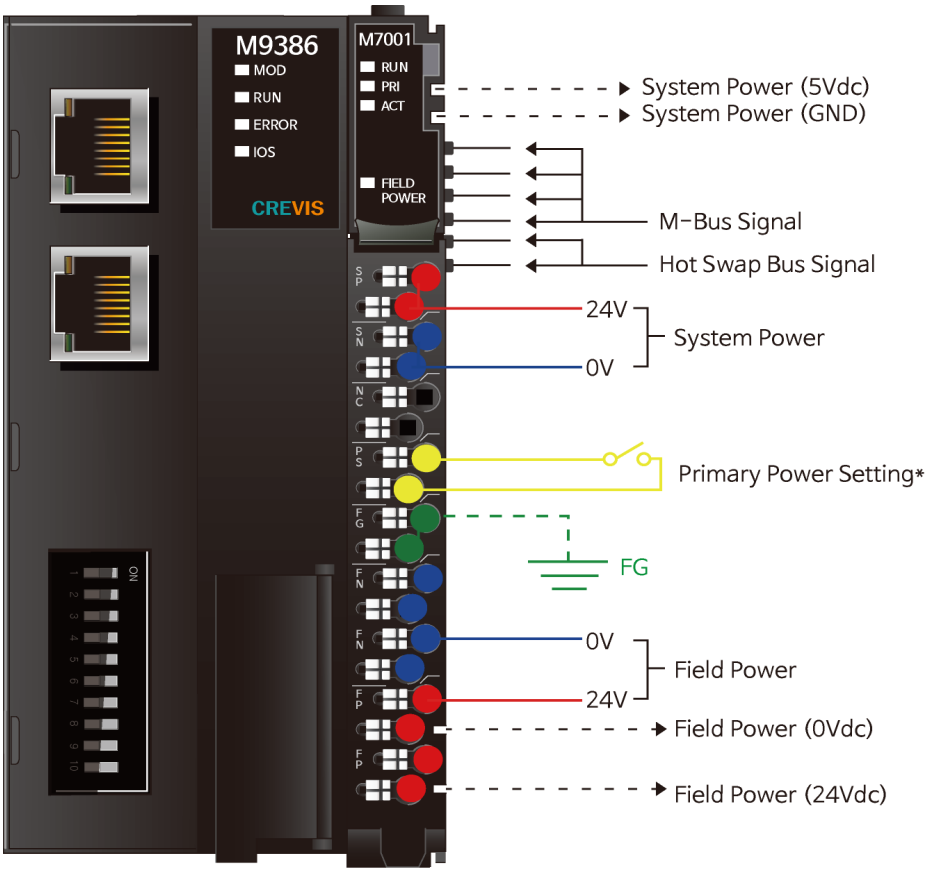
2. M9386 (EtherCAT ID Type Network Adapter)

2.1. M9386 Specification

Items	Specification
Communication Interface Specification	
Adapter Type	Slave Node (EtherCAT)
Protocol	EtherCAT
Max. Expansion Slot	63 slots
I/O Data Size	Max 128 bytes each slot
Max. Network Node	65,535
Baud Rate	100Mbps
Bus Connection	2 x RJ-45
Mac Address / IP Address	Not needed
Other Serial Port	RS232 for MODBUS/RTU, Touch Panel or IOGuide(Crevis Software)
Serial Configuration (RS232)	Node : 1 (Fixed) Baud Rate : 115200 (Fixed) Data bit : 8 (Fixed) Parity bit : No parity (Fixed) Stop bit : 1 (Fixed)
Indicator	4 LEDs 1 Green/Red, Module Status (MOD) 1 Green, Network Status (RUN) 1 Red, Error Status (ERROR) 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 M-Series system
General specification (Supplied by M7001)	
UL System Power	Supply voltage : 24Vdc nominal, Class 2
System Power	Supply voltage : 24Vdc nominal Supply voltage range : 15~28.8Vdc Protection : Output current limit, Reverse polarity protection
Power Dissipation	70mA typical @ 24Vdc
Current for I/O Module	2.0A @ 5Vdc (If except for NA, current for I/O module is about 1.7A)
Isolation	System power to internal logic : Non-isolation System power I/O driver : Isolation
UL Field Power	Supply voltage : 24Vdc nominal, Class 2
Field Power	Supply voltage : Class 2, 24Vdc typical (Max. 28.8Vdc) * Field Power Range is different depending on IO Module series. Refer to IO Module`s Specification.
Single Wire	0.205㎟ - 1.3㎟ (24-16 AWG)
Torque	0.8Nm(7 lb-in)
Max. Current Field Power Contact	DC 10A Max
Weight	179g
Module Size	54mm x 110mm x 75mm
Environment Condition	**Refer to '1. Environment Specification'**

* Class 2, adjacent to voltage rating (30Vmax)

2.2. M9386 Wiring Diagram



*** Primary Power Setting (P.S pin)**
- Short the P.S pin to set one of the two M7001 as the primary power.

Pin No.		Signal Description
0	SP	System Power, 24V
1	SP	System Power, 24V
2	SN	System Power, 0V(GND)
3	SN	System Power, 0V(GND)
4	NC	----------
5	NC	----------
6	PS	Primary Power Setting
7	PS	Primary Power Setting
8	FG	F.G
9	FG	F.G
10	FN	Field Power 0V (GND)
11	FN	Field Power 0V (GND)
12	FN	Field Power 0V (GND)
13	FN	Field Power 0V (GND)
14	FP	Field Power 24V
15	FP	Field Power 24V
16	FP	Field Power 24V
17	FP	Field Power 24V

Series No	Through Air	Over Surface	CTI
RTB18C	1.5mm	1.5mm	175≤CTI≤400

Spacings : The follwing minimum spcing in inches(millimeters) shall be maintained between uninsulated live parts of opposite polarity; and beween an uninsulated live part and a grounded Part including any mounting surface or exposed metal part.

2.3. M9386 LED Indicator

2.3.1. LED Indicator

LED No.	LED Function / Description	LED Color
MOD	Module Status	Green/Red
RUN	Current Running Status	Green
ERR	Error Status (EtherCAT)	Red
I/O	Extension Module Status	Green/Red

2.3.2. MOD (Module Status LED)

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

2.3.3. RUN (Current Running Status LED)

Status	LED	To indicate
Init	OFF	State of the EtherCAT State Machine: INIT = Initialization.
Pre-Operation	Blinking	State of the EtherCAT State Machine: PREOP = Pre-Operation.
Safe-Operation	Single Flash	State of the EtherCAT State Machine: SAFEOP = Safe-Operation.
Initialization or Bootstrap	Flashes	State of the EtherCAT State Machine: BOOT = Bootstrap (Update of the coupler firmware)
Operational	ON	State of the EtherCAT State Machine: Operational.

2.3.4. ERR (Error State LED)

Status	LED	To indicate
No Error	OFF	No Error.
Invalid Configuration	Blinking	Invalid Configuration.

2.3.5. IOS LED (Extension Module Status LED)

Status	LED	To indicate
Not Powered	OFF	Device has no expansion module or may not be powered.
Incorrect IO Module	Flashing Red	If Hotswap function is enable, configured module is incorrect.
Internal Bus Connection, Run Exchanging I/O	Green	Exchanging I/O data.
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. - Mismatch vendor code between adapter and expansion module.
Expansion Configuration	Flashing	Failed to initialize expansion module.

Failed	Red	- Detect invalid expansion module ID. - Overflow Input/Output size. - No expansion module. - Too many expansion module. - Initial protocol failure.

2.4. M7001 LED Indicator

2.4.1. LED Indicator

LED No.	LED Function / Description	LED Color
RUN	M-Bus Status	Green
PRI	Primary Status	Green
ACT	Active	Green
Field Power	Field Power Enable	Green

2.4.2. RUN(RUN Status LED)

Status	LED	To indicate
Main Power Module	Green	Supplied 5Vdc system power.
Substitution Power Module	Off	Not Supplied 5Vdc system power.

2.4.3. PRI(Primary Status LED)

Status	LED	To indicate
Main Power Module	Green	Primary power module.
Substitution Power Module	Off	Secondary power module or not use redundancy function.

2.4.4. ACT(Active Status LED)

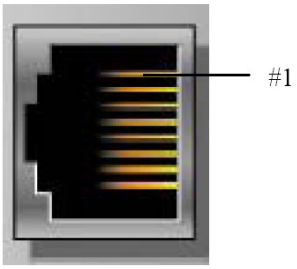
Status	LED	To indicate
Main Power Module	Green	When the Power Module is operating in main operation.
Substitution Power Module	Off	Standby with Substitution Power Module.

2.4.5. Field Power LED (Field Power Status LED)

Status	LED	To indicate
No field power	Off	Not supplied 24Vdc field power.
Supplied field power	Green	Supplied 24Vdc field power.

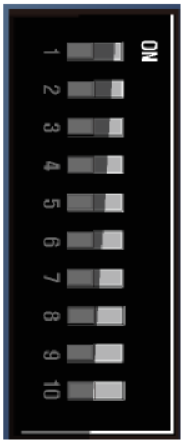
2.5. M9386 Electrical Interface

2.5.1. 5 Pin open connector



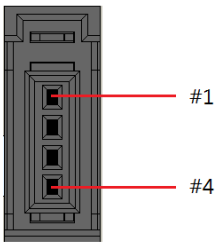
RJ-45	Signal Name	Description
1	TD+	Transmit +
2	TD-	Transmit -
3	RD+	Receive +
4	-	
5	-	
6	RD-	Receive -
7	-	
8	-	
Case	Shield	

2.5.2. Dip Switch



DIP Pole#	Description
1	IdentificationValue DIP bit#0
2	IdentificationValue DIP bit#1
3	IdentificationValue DIP bit#2
4	IdentificationValue DIP bit#3
5	IdentificationValue DIP bit#4
6	IdentificationValue DIP bit#5
7	IdentificationValue DIP bit#6
8	IdentificationValue DIP bit#7
9	Not Used
10	Not Used

2.5.3. RS232 Port for MODBUS/RTU, Touch Panel or IO-Guide



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

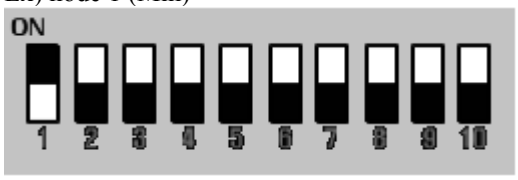
2.6. EtherCAT ID Type Setup

2.6.1. Hot Connection On TwinCAT

Hot connection function can be used to remove a node from a preconfigured Configuration or change the location of nodes and flexible. This feature is available only Ethercat ID Type in TwinCAT.
The user can use the external Dip Switch settings of the Adapter Identification Value.

For an example of using an external Dip Switch (Refer to 2.4.2.)

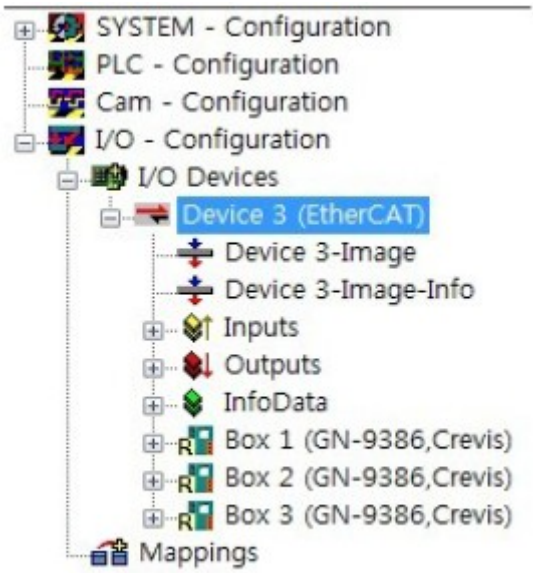
Ex) node 1 (Min)



Ex) node 255 (Max)

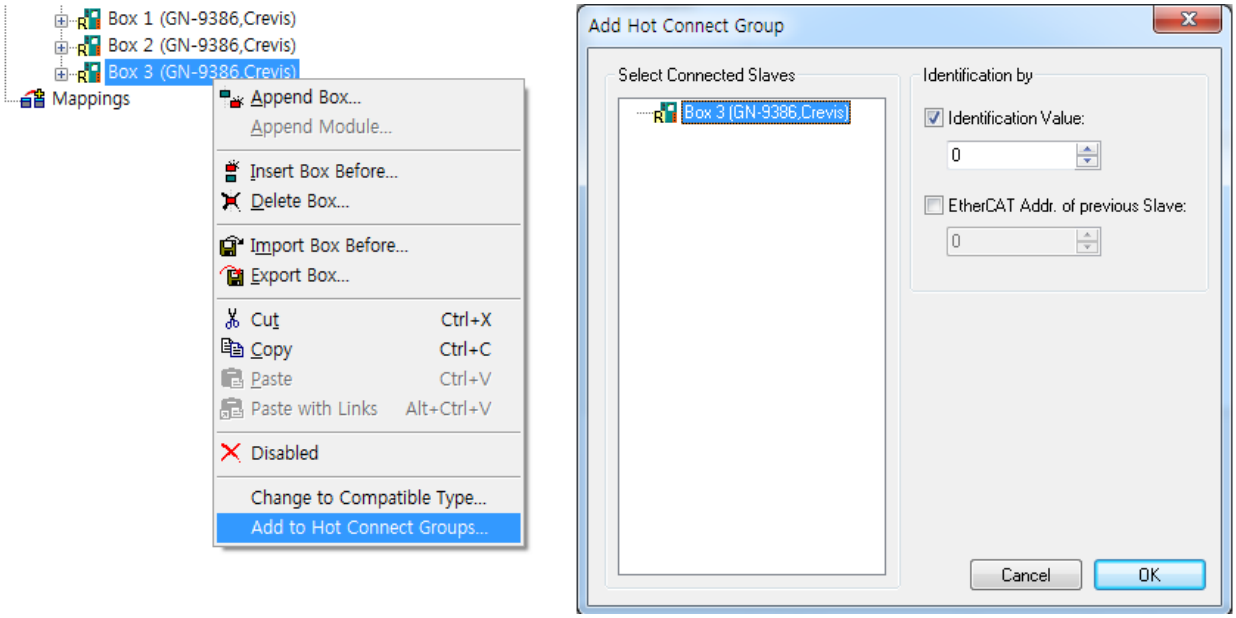
Hot Connection setting procedure.

1. Add the Ethercat ID Type in TwinCAT.

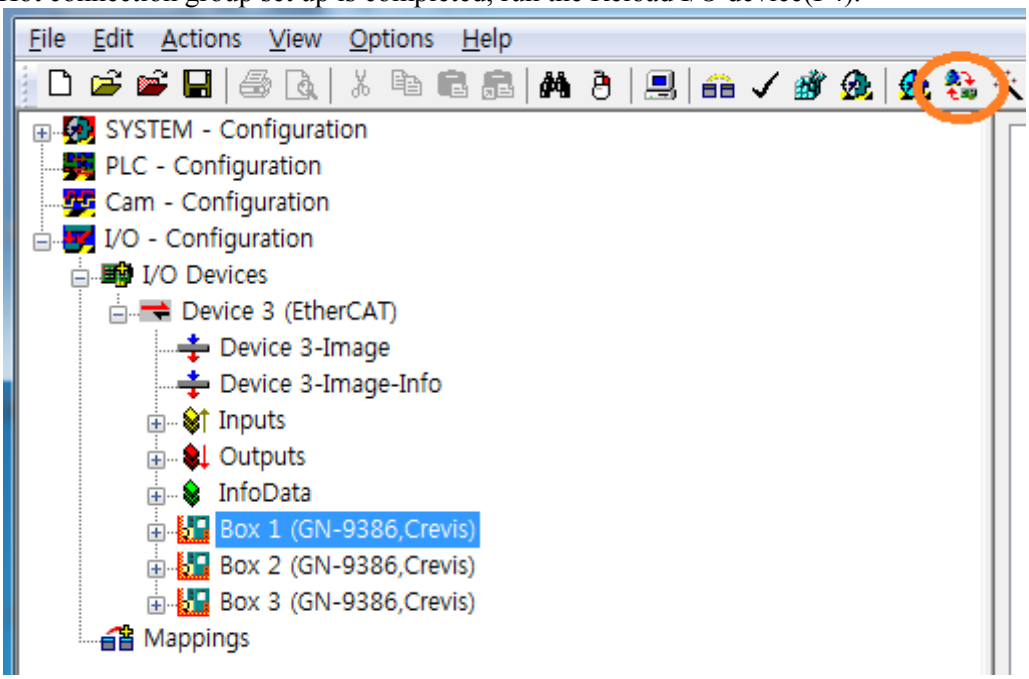


2. The Hot Connect Group settings.

Set the identification value same as dip-switch.



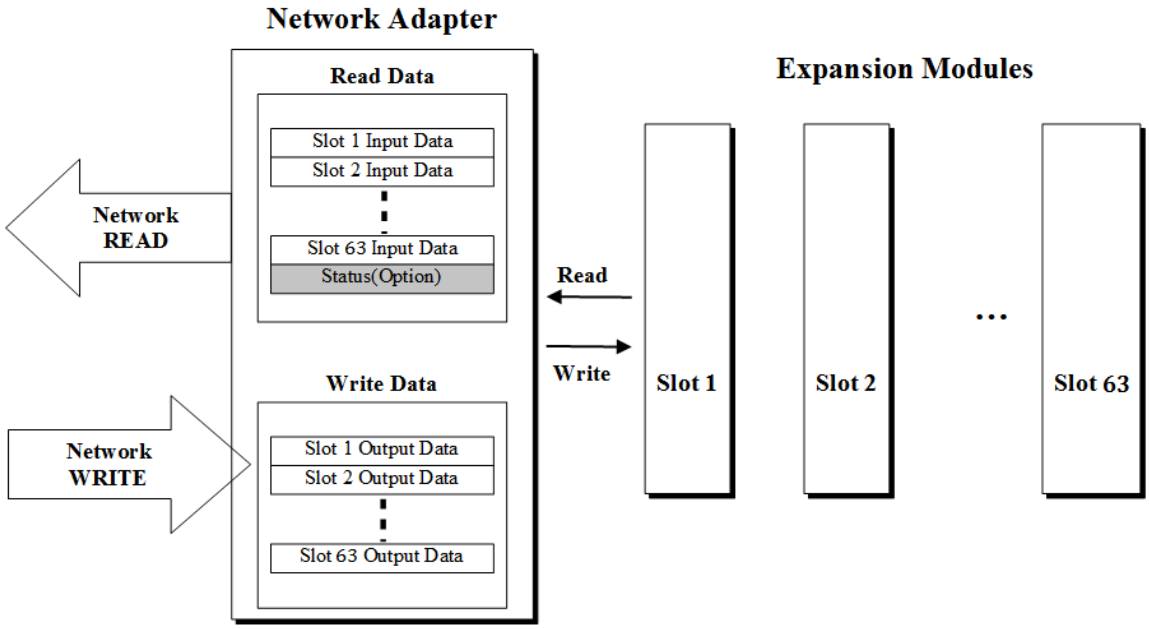
3. Hot connection group set up is completed, run the Reload I/O device(F4).



4. Now you can use the Hot connection feature.
Node is not overlapped between products. If there are same nodes, It should be changed.

2.7. I/O Process Image Map

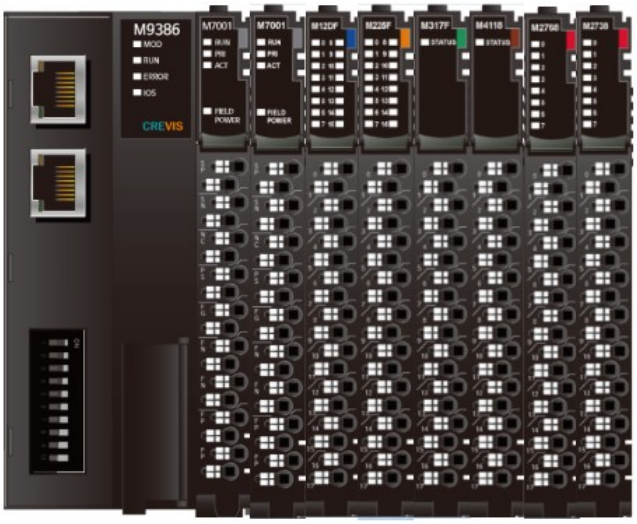
An expansion module may have 3 types of data as I/O data, configuration parameter and memory register.
The data exchange between network adapter and expansion modules is done via an I/O process image data by G-Series protocol. The following figure shows the data flow of process image between network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



2.7.1. 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 Address	Module Description
#0	EtherCAT Adapter
#1	Power module
#2	Power module
#3	16-discrete input
#4	16-discrete output
#5	16-analog input
#6	8-discrete input
#7	8-discrete output
#8	8-discrete output

- **Input Process Image**

TXPDO	Entries	Byte	Bit 7	Bit 6	Bti 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x1A01	0x6010	0	Power module (Slot#1)							
0x1A02	0x6020	1	Power module (Slot#2)							
0x1A03	0x6030	2	Analog Input Ch0 low byte (Slot#5)							
		3	Analog Input Ch0 high byte (Slot#5)							
		4	Analog Input Ch1 low byte (Slot#5)							
		5	Analog Input Ch1 high byte (Slot#5)							
		...	...							
		...	...							
		30	Analog Input Ch14 low byte (Slot#5)							
		31	Analog Input Ch14 high byte (Slot#5)							
		32	Analog Input Ch15 low byte (Slot#5)							
		33	Analog Input Ch15 high byte (Slot#5)							

- **Output Process Image**

RXPDO	Entries	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x1601	0x7010	0	Discrete Output 8 pts (Slot#1)							
0x1602	0x7020	1	Discrete Output 8 pts (Slot#2)							
0x1603	0x7030	2	Analog Output Ch0 low byte (Slot#6)							
		3	Analog Output Ch0 high byte (Slot#6)							
		4	Analog Output Ch1 low byte (Slot#6)							
		5	Analog Output Ch2 high byte (Slot#6)							
		...	...							
		...	...							
		14	Analog Output Ch6 low byte (Slot#6)							
		15	Analog Output Ch6 high byte (Slot#6)							
		16	Analog Output Ch7 low byte (Slot#6)							
		17	Analog Output Ch7 high byte (Slot#6)							
0x1604	0x7040	10	Discrete Output low 8 pts (Slot#7)							
0x1605	0x7050	12	Discrete Output low 8 pts (Slot#8)							

3. 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

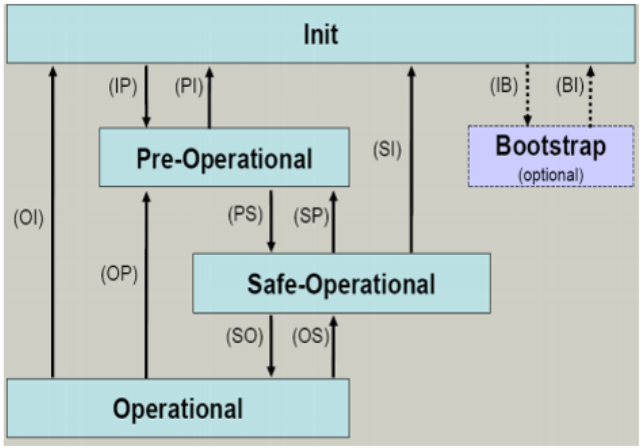
3.1. EtherCAT State Machine

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 and
- 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.

3.2. CoE Interface

3.2.1. 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)

3.2.2. Communication Objects

Index	Sub-index	Name	Flags	Default value
1000	Device type		RO	0x00001389
1001	Gbus Status		RO	Normal Operation : 0x03 **
1002	Master Fault Action		RW	0xA010
1003	Hot Swap Disable		RW	Enable : 0x00, Disable : 0x01
1008	Device name		RO	M9386(Crevis)
1009	Hardware version		RO	M9386.v1
100A	Software version		RO	1.000
1018	Identity		RO	0x05
	01	Vendor ID (Crevis: 029D)	RO	0x0000029D
	02	Product code	RO	0x004D9386
	03	Revision	RO	0x00010000
	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,Mxxxx,RXPDO		RO	0xnn
	01	SubIndex 001	RO	0x7010:01, 8
	...	...	...	...
	nn	SubIndex nnn	RO	0x7010:nn, 8
1A01*	Slot#x, Mxxxx,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
7010*	Mxxxx		RO	0xnn
	01	Byte#0	RW P	0x00
	...	...	...	...
	nn	Byte#nnn	RW P	0x00
8000	M9386(Parameter)		RO	
	01	Byte#0	RW	
	02	Byte#1	RW	
	03	Byte#2	RW	

	04	Byte#3	RW	
8nn0*	Mxxxx(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 (M9386)	RO	0x00009386
	...	...	...	...
	63	Subindex 063	RO	0x0000xxxx
F050	Detected Module Ident List		RO	
	01...	SubIndex 001	RO	

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

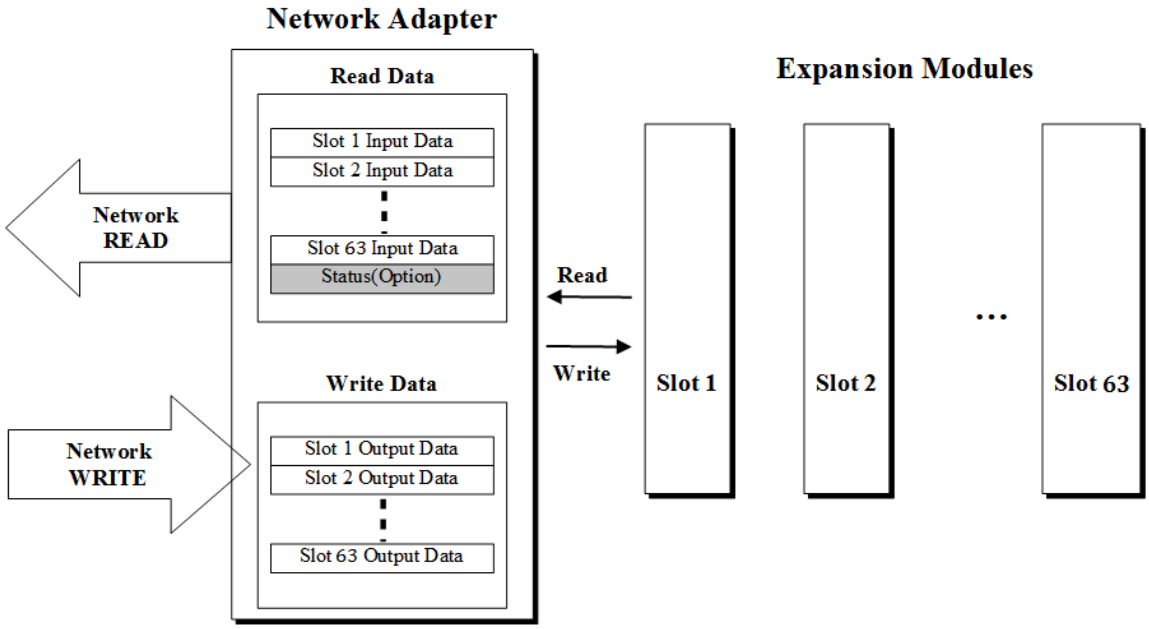
** Mbus Status

0x01 : INIT_STATE
0x02 : IDLE_STATE
0x03 : RUN_STATE
0x04 : STOP_STATE
0x05 : FAULT_STATE
0x06 : RESET_STATE
0x07 : CRCERR_STATE
0x08 : PAUSE_STATE
0x80* : At Hot swap mode
expansion module error

4. MODBUS INTERFACE

4.1. 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 M-Series protocol. The following figure shows the data flow of process image between network adapter and expansion modules.



4.1.1. 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. 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

4.3. 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

4.4. MODBUS Transmission Mode

Two different serial transmission modes are defined : The RTU mode and the ASCII mode. It defines the bit contents of message fields transmitted serially on the line. It determines how information is packed into the message fields and decoded.

4.4.1. RTU Transmission Mode

When devices communicate on a MODBUS serial line using the RTU (Remote Terminal Unit) mode, each 8–bit byte in a message contains two 4–bit hexadecimal characters. The main advantage of this mode is that its greater character density allows better data throughput than ASCII mode for the same baud rate. Each message must be transmitted in a continuous stream of characters.

Start	Address	Function	Data	CRC Check	End
≥ 3.5 char	1 char	1 char	Up to 252 chars(s)	2 chars	≥ 3.5 char

4.4.2. ASCII Transmission Mode

When devices are setup to communicate on a MODBUS serial line using ASCII (American Standard Code for Information Interchange) mode, each 8–bit byte in a message is sent as two ASCII characters. This mode is used when the physical communication link or the capabilities of the device does not allow the conformance with RTU mode requirements regarding timers management.

Start	Address	Function	Data	LRC Check	End
1 char ‘:’	2 chars	2 chars	Up to 2x252 char(s)	2 chars	2 chars CR,LF

4.4.3. 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	RTU	ASCII	ASCII (bus line)
Start of Frame	-	t1-t2-t3	“:”	0x3A
Slave Address	0x63	0x63	“63”	0x36, 0x33
Function Code	0x01	0x01	“01”	0x30, 0x31
Starting Address Hi	0x10	0x10	“10”	0x31, 0x30
Starting Address Lo	0x00	0x00	“00”	0x30, 0x30
Quantity of Outputs Hi	0x00	0x00	“00”	0x30, 0x30
Quantity of Outputs Lo	0x10	0x10	“10”	0x31, 0x30
Error Check (CRC/LRC)	-	0x31, 0x44	“7C”	0x37, 0x43
End of Frame	-	t1-t2-t3	CR, LF	0x0D, 0x0A

- **Response**

Field name	Example	RTU	ASCII	ASCII (bus line)
Start of Frame	---	t1-t2-t3	“:”	0x3A
Slave Address	0x63	0x63	“63”	0x36, 0x33
Function Code	0x01	0x01	“01”	0x30, 0x31
Byte Count	0x02	0x02	“02”	0x30, 0x32
Output Status	0x00	0x00	“00”	0x30, 0x30
Output Status	0x00	0x00	“00”	0x30, 0x30
Error Check (CRC/LRC)	---	0x40, 0x34	“9A”	0x39, 0x41
End of Frame	---	t1-t2-t3	CR, LF	0x0D, 0x0A

4.4.4. 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	RTU	ASCII	ASCII (bus line)
Start of Frame	---	t1-t2-t3	":"	0x3A
Slave Address	0x63	0x63	"07"	0x36, 0x33
Function Code	0x02	0x02	"02"	0x30, 0x32
Starting Address Hi	0x00	0x00	"00"	0x30, 0x30
Starting Address Lo	0x00	0x00	"00"	0x30, 0x30
Quantity of Inputs Hi	0x00	0x00	"00"	0x30, 0x30
Quantity of Inputs Lo	0x10	0x10	"0A"	0x31, 0x30
Error Check (CRC/LRC)	---	0x71, 0x84	"ED"	0x38, 0x42
End of Frame	---	t1-t2-t3	CR, LF	0x0D, 0xA

- **Response**

Field name	Example	RTU	ASCII	ASCII (bus line)
Start of Frame	---	t1-t2-t3	":"	0x3A
Slave Address	0x63	0x63	"63"	0x36, 0x33
Function Code	0x02	0x02	"02"	0x30, 0x32
Byte Count	0x02	0x02	"02"	0x30, 0x32
Input Status	0x00	0x00	"00"	0x30, 0x30
Input Status	0x00	0x00	"00"	0x30, 0x30
Error Check (CRC/LRC)	---	0x40, 0x70	"99"	0x39, 0x39
End of Frame	---	t1-t2-t3	CR, LF	0x0D, 0xA

4.4.5. 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	RTU	ASCII	ASCII (bus line)
Start of Frame	---	t1-t2-t3	":"	0x3A
Slave Address	0x63	0x63	"63"	0x36, 0x33
Function Code	0x03	0x03	"03"	0x30, 0x33
Starting Address Hi	0x10	0x10	"10"	0x31, 0x30
Starting Address Lo	0x00	0x00	"00"	0x30, 0x30
Quantity of Register Hi	0x00	0x00	"00"	0x30, 0x30
Quantity of Register Lo	0x01	0x01	"01"	0x30, 0x31
Error Check (CRC/LRC)	---	0x88, 0x88	"89"	0x38, 0x39
End of Frame	---	t1-t2-t3	CR, LF	0x0D, 0x0A

- **Response**

Field name	Example	RTU	ASCII	ASCII (bus line)
Start of Frame	---	t1-t2-t3	":"	0x3A
Slave Address	0x63	0x63	"63"	0x36, 0x33
Function Code	0x03	0x03	"03"	0x30, 0x33
Byte Count	0x02	0x02	"02"	0x30, 0x32
Output Register#0 Hi	0x02	0x02	"02"	0x30, 0x32
Output Register#0 Lo	0xE5	0xE5	"E5"	0x45, 0x35
Error Check (CRC/LRC)	---	0x81, 0x67	"B1"	0x42, 0x31
End of Frame	---	t1-t2-t3	CR, LF	0x0D, 0x0A

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

4.4.6. 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.

- **Request**

Field name	Example	RTU	ASCII	ASCII (bus line)
Start of Frame	---	t1-t2-t3	":"	0x3A
Slave Address	0x63	0x63	"63"	0x36, 0x33
Function Code	0x04	0x04	"04"	0x30, 0x34
Starting Address Hi	0x10	0x10	"10"	0x31, 0x30
Starting Address Lo	0x00	0x00	"00"	0x30, 0x30
Quantity of Register Hi	0x00	0x00	"00"	0x30, 0x30
Quantity of Register Lo	0x01	0x01	"01"	0x30, 0x31
Error Check (CRC/LRC)	---	0x3D, 0x48	"88"	0x38, 0x38
End of Frame	---	t1-t2-t3	CR, LF	0x0D, 0x0A

- **Response**

Field name	Example	RTU	ASCII	ASCII (bus line)
Start of Frame	---	t1-t2-t3	":"	0x3A
Slave Address	0x63	0x63	"63"	0x36, 0x33
Function Code	0x04	0x04	"04"	0x30, 0x34
Byte Count	0x02	0x02	"02"	0x30, 0x32
Input Register#0 Hi	0x02	0x02	"02"	0x30, 0x32
Input Register#0 Lo	0xE5	0xE5	"E5"	0x45, 0x35
Error Check (CRC/LRC)	---	0x80, 0x13	"B0"	0x42, 0x30
End of Frame	---	t1-t2-t3	CR, LF	0x0D, 0xA

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

4.4.7. 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	RTU	ASCII	ASCII (bus line)
Start of Frame	---	t1-t2-t3	":"	0x3A
Slave Address	0x63	0x63	"36"	0x36, 0x33
Function Code	0x05	0x05	"05"	0x30, 0x35
Output Address Hi	0x10	0x10	"10"	0x31, 0x30
Output Address Lo	0x00	0x00	"00"	0x30, 0x30
Output Value Hi	0xFF	0xFF	"FF"	0x46, 0x46
Output Value Lo	0x00	0x00	"00"	0x30, 0x30
Error Check (CRC/LRC)	---	0x80, 0xB8	"8Y"	0x38, 0x59
End of Frame	---	t1-t2-t3	CR, LF	0x0D, 0xA

- **Response**

Field name	Example	RTU	ASCII	ASCII (bus line)
Start of Frame	---	t1-t2-t3	":"	0x3A
Slave Address	0x63	0x63	"36"	0x36, 0x33
Function Code	0x05	0x05	"05"	0x30, 0x35
Output Address Hi	0x10	0x10	"10"	0x31, 0x30
Output Address Lo	0x00	0x00	"00"	0x30, 0x30
Output Value Hi	0xFF	0xFF	"FF"	0x46, 0x46
Output Value Lo	0x00	0x00	"00"	0x30, 0x30
Error Check (CRC/LRC)	---	0x80, 0xB8	"8Y"	0x38, 0x59
End of Frame	---	t1-t2-t3	CR, LF	0x0D, 0xA

- Output bit of address 0x1001 turns ON.

4.4.8. 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	RTU	ASCII	ASCII (bus line)
Start of Frame	---	t1-t2-t3	":"	0x3A
Slave Address	0x63	0x63	"63"	0x36, 0x33
Function Code	0x06	0x06	"06"	0x30, 0x36
Register Address Hi	0x08	0x08	"08"	0x30, 0x38
Register Address Lo	0x00	0x00	"00"	0x30, 0x30
Register Value Hi	0x00	0x00	"00"	0x30, 0x30
Register Value Lo	0xFF	0xFF	"FF"	0x46, 0x46
Error Check (CRC/LRC)	---	0xC3, 0xA8	"90"	0x39, 0x30
End of Frame	---	t1-t2-t3	CR, LF	0x0D, 0xA

- **Response**

Field name	Example	RTU	ASCII	ASCII (bus line)
Start of Frame	---	t1-t2-t3	":"	0x3A
Slave Address	0x63	0x63	"63"	0x36, 0x33
Function Code	0x06	0x06	"06"	0x30, 0x36
Register Address Hi	0x08	0x08	"08"	0x30, 0x38
Register Address Lo	0x00	0x00	"00"	0x30, 0x30
Register Value Hi	0x00	0x00	"00"	0x30, 0x30
Register Value Lo	0xFF	0xFF	"FF"	0x46, 0x46
Error Check (CRC/LRC)	---	0xC3, 0xA8	"90"	0x39, 0x30
End of Frame	---	t1-t2-t3	CR, LF	0x0D, 0xA

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

4.4.9. 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	RTU	ASCII	ASCII (bus line)
Start of Frame	---	t1-t2-t3	":"	0x3A
Slave Address	0x07	0x07	"07"	0x30, 0x37
Function Code	0x08	0x08	"08"	0x30, 0x38
Sub-Function Hi	0x00	0x00	"00"	0x30, 0x30
Sub-Function Lo	0x00	0x00	"00"	0x30, 0x30
Data Hi	0x11	0x11	"11"	0x31, 0x31
Data Lo	0x22	0x22	"22"	0x32, 0x32
Error Check (CRC/LRC)	---	0x6C, 0x24	"BE"	0x42, 0x45
End of Frame	---	t1-t2-t3	CR, LF	0x0D, 0xA

- **Response**

Field name	Example	RTU	ASCII	ASCII (bus line)
Start of Frame	---	t1-t2-t3	":"	0x3A
Slave Address	0x07	0x07	"07"	0x30, 0x37
Function Code	0x08	0x08	"08"	0x30, 0x38
Sub-Function Hi	0x00	0x00	"00"	0x30, 0x30
Sub-Function Lo	0x00	0x00	"00"	0x30, 0x30
Data Hi	0x11	0x11	"11"	0x31, 0x31
Data Lo	0x22	0x22	"22"	0x32, 0x32
Error Check (CRC/LRC)	---	0x6C, 0x24	"BE"	0x42, 0x45
End of Frame	---	t1-t2-t3	CR, LF	0x0D, 0xA

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+0xAA55+Sumcheck	Echo Request Data	Reset with Factory

Sub-function 0x000A(10) Clear Counters and Diagnostic Register
The goal is to clear all counters and the diagnostic register. Counters are also cleared upon power–up.

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

Sub-function 0x000B(11) Return Bus Message Count
The response data field returns the quantity of messages that the remote device has detected on the communications system since its last restart, clear counters operation, or power–up.

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

Sub-function 0x000C(12) Return Bus Communication Error Count
The response data field returns the quantity of CRC errors encountered by the remote device since its last restart, clear counters operation, or power–up.

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

Sub-function 0x000D(13) Return Bus Exception Error Count
The response data field returns the quantity of MODBUS exception responses returned by the remote device since its last restart, clear counters operation, or power–up.
Exception responses are described and listed in section 3.2.11.

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

Sub-function 0x000E(14) Return Slave Message Count
The response data field returns the quantity of messages addressed to the remote device, or broadcast, that the remote device has processed since its last restart, clear counters operation, or power–up.

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

Sub-function 0x000F(15) Return Slave No Response Count
The response data field returns the quantity of messages addressed to the remote device for which it has returned no response (neither a normal response nor an exception response), since its last restart, clear counters operation, or power–up.

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

Sub-function 0x0064(100) Return Slave ModBus, Internal Bus Status
The response data field returns the status of ModBus and Internal Bus addressed to the remote device.
This status values are identical with status 1word of input process image.

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

Sub-function 0x0065(101) Return Slave Watchdog Error Count
The response data field returns the quantity of watchdog error addressed to the remote device since its last restart, clear counters operation, or power–up.

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

4.4.10. 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	RTU	ASCII	ASCII (bus line)
Start of Frame	---	t1-t2-t3	":"	0x3A
Slave Address	0x63	0x63	"63"	0x36, 0x33
Function Code	0x0F	0x0F	"0F"	0x30, 0x46
Starting Address Hi	0x10	0x10	"10"	0x31, 0x30
Starting Address Lo	0x00	0x00	"00"	0x30, 0x30
Quantity of Outputs Hi	0x00	0x00	"00"	0x30, 0x30
Quantity of Outputs Lo	0x10	0x10	"10"	0x31, 0x30
Byte Count	0x02	0x02	"02"	0x30, 0x32
Output Value#0	0x0F	0x0F	"0F"	0x30, 0x46
Output Value#1	0x00	0x00	"00"	0x30, 0x30
Error Check (CRC/LRC)	---	0x47, 0x73	"5D"	0x35, 0x44
End of Frame	---	t1-t2-t3	CR, LF	0x0D, 0xA

- **Response**

Field name	Example	RTU	ASCII	ASCII (bus line)
Start of Frame	---	t1-t2-t3	":"	0x3A
Slave Address	0x63	0x63	"63"	0x36, 0x33
Function Code	0x0F	0x0F	"0F"	0x30, 0x46
Starting Address Hi	0x10	0x10	"10"	0x31, 0x30
Starting Address Lo	0x00	0x00	"00"	0x30, 0x30
Quantity of Outputs Hi	0x00	0x00	"00"	0x30, 0x30
Quantity of Outputs Lo	0x10	0x10	"10"	0x31, 0x30
Error Check (CRC/LRC)	---	0x58, 0x85	"6E"	0x36, 0x45
End of Frame	---	t1-t2-t3	CR, LF	0x0D, 0xA

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

4.4.11. 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	RTU	ASCII	ASCII (bus line)
Start of Frame	---	t1-t2-t3	":"	0x3A
Slave Address	0x63	0x63	"63"	0x36, 0x33
Function Code	0x10	0x10	"10"	0x31, 0x30
Starting Address Hi	0x08	0x08	"08"	0x30, 0x38
Starting Address Lo	0x00	0x00	"00"	0x30, 0x30
Quantity of Registers Hi	0x00	0x00	"00"	0x30, 0x30
Quantity of Registers Lo	0x01	0x01	"01"	0x30, 0x31
Byte Count	0x02	0x02	"02"	0x30, 0x32
Register Value#0 Hi	0x00	0x00	"00"	0x30, 0x30
Register Value#0 Lo	0xFF	0xFF	"FF"	0x46, 0x46
Error Check (CRC/LRC)	---	0xDE, 0xB2	"83"	0x38, 0x33
End of Frame	---	t1-t2-t3	CR, LF	0x0D, 0xA

- **Response**

Field name	Example	RTU	ASCII	ASCII (bus line)
Start of Frame	---	t1-t2-t3	":"	0x3A
Slave Address	0x63	0x63	"63"	0x36, 0x33
Function Code	0x10	0x10	"10"	0x31, 0x30
Starting Address Hi	0x08	0x08	"08"	0x30, 0x38
Starting Address Lo	0x00	0x00	"00"	0x30, 0x30
Quantity of Registers Hi	0x00	0x00	"00"	0x30, 0x30
Quantity of Registers Lo	0x01	0x01	"01"	0x30, 0x31
Error Check (CRC/LRC)	---	0x0B, 0xEB	"84"	0x38, 0x34
End of Frame	---	t1-t2-t3	CR, LF	0x0D, 0xA

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

4.4.12. 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	RTU	ASCII	ASCII (bus line)
Start of Frame	---	t1-t2-t3	":"	0x3A
Slave Address	0x63	0x63	"63"	0x36, 0x33
Function Code	0x17	0x17	"17"	0x31, 0x37
Read Starting Address Hi	0x00	0x00	"00"	0x30, 0x30
Read Starting Address Lo	0x00	0x00	"00"	0x30, 0x30
Quantity of Read Hi	0x00	0x00	"00"	0x30, 0x30
Quantity of Read Lo	0x01	0x01	"01"	0x30, 0x31
Write Starting Address Hi	0x08	0x08	"08"	0x30, 0x38
Write Starting Address Lo	0x00	0x00	"00"	0x30, 0x30
Quantity of Write Hi	0x00	0x00	"00"	0x30, 0x30
Quantity of Write Lo	0x01	0x01	"01"	0x30, 0x31
Byte Count	0x02	0x02	"02"	0x30, 0x32
Write Reg. Value#0 Hi	0x00	0x00	"00"	0x30, 0x30
Write Reg. Value#0 Lo	0xFF	0xFF	"FF"	0x46, 0x46
Error Check (CRC/LRC)	---	0x1B, 0xCC	"7B"	0x37, 0x42
End of Frame	---	t1-t2-t3	CR, LF	0x0D, 0x0A

- **Response**

Field name	Example	RTU	ASCII	ASCII (bus line)
Start of Frame	---	t1-t2-t3	":"	0x3A
Slave Address	0x63	0x63	"63"	0x36, 0x33
Function Code	0x17	0x17	"17"	0x31, 0x37
Byte Count	0x02	0x02	"02"	0x30, 0x32
Read Reg. Value#0 Hi	0x00	0x00	"00"	0x30, 0x30
Read Reg. Value#0 Lo	0xFF	0xFF	"FF"	0x46, 0x46
Error Check (CRC/LRC)	---	0x04, 0x3C	"85"	0x38, 0x35
End of Frame	---	t1-t2-t3	CR, LF	0x0D, 0xA

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

4.4.13. 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	RTU	ASCII	ASCII (bus line)
Start of Frame	---	t1-t2-t3	":"	0x3A
Slave Address	0x07	0x07	"07"	0x30, 0x37
Function Code	0x81	0x81	"81"	0x38, 0x31
Exception Code	0x02	0x02	"02"	0x30, 0x32
Error Check (CRC/LRC)	---	0x22, 0xC0	"76"	0x37, 0x36
End of Frame	---	t1-t2-t3	CR, LF	0x0D, 0xA

- **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.

- M9386 response exception code 01, 02, 03, 04 and 06.

4.5. 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).

4.5.1. 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 = 0xA010
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) “M9386,EtherCAT ID Type,M-Series”
0x1006(4102)	Read	1word	Sum check of EEPROM
0x1010(4112)	Read	2word	Firmware release date
0x1011(4113)	Read	2word	Product manufacturing inspection date
0x101E(4126)	Read	7word - 1word - 1word - 1word - 1word - 1word - 2word	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 consists of valid string length (first 1word) and array of characters

4.5.2. Adapter Hotswap Register (0x1060, 4192)

Address	Access	Type, Size	Description
0x1060(4192)	Read/ Write	1word	Hot swap status 0 : Enable 1 : Disable
0x1062(4194)*	Read	1word	Error slot detection 0 : No error slot 1 : Error slot detection
0x1063(4195)*	Read	4word	Error slot location, 8x8 bit

* 0x1062 and 0x1063 functions are only available if hot swap(0x1060) is enabled.

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

<table>
<tr><th>Address</th><th>Access</th><th>Type, Size</th><th>Description</th></tr>
<tr><td>0x1100(4352)*</td><td>Read/Write</td><td>1word</td><td>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. Also, can activate hold last state as IO parameter.</td></tr>
<tr><td>0x1102(4354)</td><td>Read</td><td>1word</td><td>Start address of input image word register. =0x0000</td></tr>
<tr><td>0x1103(4355)</td><td>Read</td><td>1word</td><td>Start address of output image word register. =0x0800</td></tr>
<tr><td>0x1104(4356)</td><td>Read</td><td>1word</td><td>Size of input image word register.</td></tr>
<tr><td>0x1105(4357)</td><td>Read</td><td>1word</td><td>Size of output image word register.</td></tr>
<tr><td>0x1106(4358)</td><td>Read</td><td>1word</td><td>Start address of input image bit. = 0x0000</td></tr>
<tr><td>0x1107(4359)</td><td>Read</td><td>1word</td><td>Start address of output image bit. =0x1000</td></tr>
<tr><td>0x1108(4360)</td><td>Read</td><td>1word</td><td>Size of input image bit.</td></tr>
<tr><td>0x1109(4361)</td><td>Read</td><td>1word</td><td>Size of output image bit.</td></tr>
<tr><td>0x110A(4362)</td><td>Read</td><td>1word</td><td>Update time for cyclic data change (same as 0x1028)
(Unit : us)</td></tr>
<tr><td>0x110E(4366)</td><td>Read</td><td>upto 33word</td><td>Expansion slot’s M-number including
First 1word is adapter’s number, if M9386, then 0x9386</td></tr>
<tr><td>0x1110(4368)</td><td>Read</td><td>1word</td><td>Number of expansion slot</td></tr>
<tr><td>0x1113(4371)</td><td>Read</td><td>upto 33word</td><td>Expansion slot Module Id.
First 1word is adapter’s product code.</td></tr>
<tr><td>0x1119(4377)</td><td>Read</td><td>1word</td><td>High byte is ModBus status, low byte is internal status.
Zero value means ‘no error’.
<table>
<tr><th>ModBus Status</th><th>Internal bus status(M-Bus)</th></tr>
<tr><td>0x00 : No Error
0x01 : Error Dip Switch
0x40 : Error CRC LRC
0x80 : Error Watchdog</td><td>0x01 : Init State
0x02 : Idle State
0x03 : Run State
0x04 : Stop State
0x05 : Fault State
0x06 : Reset State
0x07 : CRC Error State
0x08 : PauseState
0x09 : Master Fault State
0x80* : At Hot swap mode
expansion module error</td></tr>
</table></td></tr>
<tr><td>0x111D(4381)</td><td>Read</td><td>1word</td><td>Adapter M-Series Revision.</td></tr>
</table>

* After the system is reset, the new “Set Value” action is applied.

**If The Hot Swap Status is enabled, the flag bit is raised when the module is removed.

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

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

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

Address Offset	Expansion Slot#1	Expansion Slot#2	Expansion Slot#3	Expansion Slot#4	……..	Expansion Slot#63
+ 0x00(+0)	0x2000(8192)	0x2020(8224)	0x2040(8256)	0x2060(8288)	…….	0x27C0(10176)
+ 0x01(+1)	0x2001(8193)	0x2021(8225)	0x2041(8257)	0x2061(8289)	…….	0x27C1(10177)
+ 0x02(+2)	0x2002(8194)	0x2022(8226)	0x2042(8258)	0x2062(8290)	…….	0x27C2(10178)
+ 0x03(+3)	0x2003(8195)	0x2023(8227)	0x2043(8259)	0x2063(8291)	…….	0x27C3(10179)
+ 0x04(+4)	0x2004(8196)	0x2024(8228)	0x2044(8260)	0x2064(8292)	…….	0x27C4(10180)
+ 0x05(+5)	0x2005(8197)	0x2025(8229)	0x2045(8261)	0x2065(8293)	…….	0x27C5(10181)
+ 0x06(+6)	0x2006(8198)	0x2026(8230)	0x2046(8262)	0x2066(8294)	…….	0x27C6(10182)
+ 0x07(+7)	0x2007(8199)	0x2027(8231)	0x2047(8263)	0x2067(8295)	…….	0x27C7(10183)
+ 0x08(+8)	0x2008(8200)	0x2028(8232)	0x2048(8264)	0x2068(8296)	…….	0x27C8(10184)
+ 0x09(+9)	0x2009(8201)	0x2029(8233)	0x2049(8265)	0x2069(8297)	…….	0x27C9(10185)
+ 0x0A(+10)	0x200A(8202)	0x202A(8234)	0x204A(8266)	0x206A(8298)	…….	0x27CA(10186)
+ 0x0B(+11)	0x200B(8203)	0x202B(8235)	0x204B(8267)	0x206B(8299)	…….	0x27CB(10187)
+ 0x0C(+12)	0x200C(8204)	0x202C(8236)	0x204C(8268)	0x206C(8300)	…….	0x27CC(10188)
+ 0x0D(+13)	0x200D(8205)	0x202D(8237)	0x204D(8269)	0x206D(8301)	…….	0x27CD(10189)
+ 0x0E(+14)	0x200E(8206)	0x202E(8238)	0x204E(8270)	0x206E(8302)	…….	0x27CE(10190)
+ 0x0F(+15)	0x200F(8207)	0x202F(8239)	0x204F(8271)	0x206F(8303)	…….	0x27CF(10191)
+ 0x10(+16)	0x2010(8208)	0x2030(8240)	0x2050(8272)	0x2070(8304)	…….	0x27D0(10192)
+ 0x11(+17)	0x2011(8209)	0x2031(8241)	0x2051(8273)	0x2071(8305)	…….	0x27D1(10193)
+ 0x12(+18)	0x2012(8210)	0x2032(8242)	0x2052(8274)	0x2072(8306)	…….	0x27D2(10194)
+ 0x13(+19)	0x2013(8211)	0x2033(8243)	0x2053(8275)	0x2073(8307)	…….	0x27D3(10195)
+ 0x14(+20)	0x2014(8212)	0x2034(8244)	0x2054(8276)	0x2074(8308)	…….	0x27D4(10196)
+ 0x15(+21)	0x2015(8213)	0x2035(8245)	0x2055(8277)	0x2075(8309)	…….	0x27D5(10197)
+ 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	M-number, if M-1238, returns 0x1238
+ 0x0F(+15)	Read	String upto 72byte	First 1word is length of valid character string. If M12DF, returns "00 23 4D 31 32 44 46 2C 20 31 36 44 49 2C 20 32 34 56 64 63 2C 20 55 6E 69 76 65 72 73 61 6C 20 31 38 " Valid character size = 0x001E =30 characters, "M12DF, 16DI, 24Vdc, Universal 18RTB"
+ 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 16, Rev 1.016)
+ 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.

4.6. Supported MODBUS Function Codes

MODBUS Reference Documents

http://www.modbus.org

MODBUS Tools

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

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