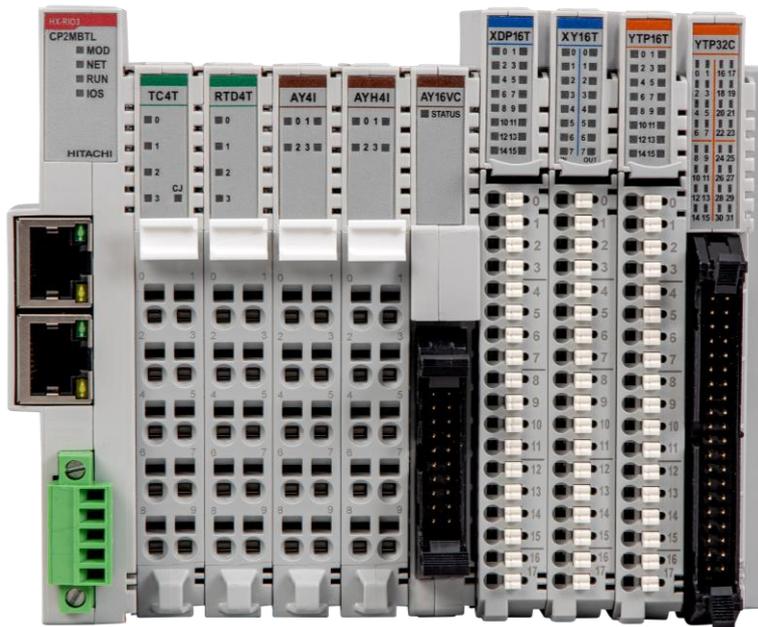


Modbus Programmable Light I/O RIO3-CP2MBTL

User Manual



Version 2.101

REVISION HISTORY				
REV	PAGE	REMARKS	DATE	EDITOR
2.00	All	New Document	Sep. 2022	Faber
2.001	8 14	WebVisu information removed, COM Port information added	Sep. 2022	Faber
2.1	22 20	External Features Library RTC function	Aug. 2024	Lankala
2.101	10 16, 43 11 36	System supply voltage (15V → 18V) Exception Indication, 110D content, 7bit setting Additional Explanation for Online Changes Restart Communications Option (program reset)	Feb.2025	Lankala

Table of Contents

1	Important Notes	6
1.1	Safety Instructions	7
1.1.1	Symbols	7
1.1.2	Safety Notes	7
1.1.3	Certification.....	7
2	RIO3-Series System	8
2.1	Electrical Interface.....	8
2.2	I/O Process Image Map	9
3	RIO3-CP2MBTL specification.....	10
3.1	General specification	10
3.2	Environment specification.....	10
3.3	Programmable specification.....	11
3.4	Interface specification	12
3.5	Ethernet connection specification	12
3.6	Serial connection specification	13
4	Module Description.....	14
4.2	Wiring Diagram.....	14
4.3	LED Indicator.....	15
4.3.1	MOD (Module Status LED).....	15
4.3.2	NET (Network Status LED).....	15
4.3.3	RUN (PLC Run/Stop Status LED).....	15
4.3.4	IOS (Expansion Module Status LED).....	16
4.3.5	Field Power Status LED	16
4.3.6	Exception indication	16
4.4	RJ-45 Socket.....	17
4.5	Toggle Switch and Push Button.....	17
4.6	DIP Switch	18

5	HX-RIO3 Series PIO (Programmable I/O) Functions.....	19
5.1	Webserver.....	19
5.2	Default Network Setting.....	21
5.3	RTC (Real Time Clock) Function.....	21
6	External Feature Library	23
6.1	Battery State of charge	23
6.2	Read / Write Special Register (PIO Parameter Access).....	24
6.2.1	Read Special Register	24
6.2.2	Write Special Register.....	24
6.2.3	Example to change Special Register.....	25
6.3	Read / Write Slot Parameter.....	26
6.3.1	Read Slot Parameter.....	26
6.3.2	Write Slot Parameter.....	26
6.3.3	Example to access the extended I/O module parameter	27
7	Upgrade Firmware	28
7.1	Using IAP over Ethernet.....	28
8	Modbus Interface	30
8.1	MODBUS Interface Register / Bit Map.....	30
8.1.1	Register Map.....	30
8.1.2	Bit Map	30
8.2	MODBUS Transmission Mode.....	30
8.2.1	RTU Transmission Mode.....	30
8.2.2	ASCII Transmission Mode	31
8.3	Supported Modbus Function Codes.....	31
8.3.1	1 (0x01) Read Coils	31
8.3.2	2 (0x02) Read Discrete Inputs.....	32
8.3.3	3 (0x03) Read Holding Registers	33
8.3.4	4 (0x04) Read Input Registers.....	33

8.3.5	5 (0x05) Write Single Coil	34
8.3.6	6 (0x06) Write Single Register	35
8.3.7	8 (0x08) Diagnostics	35
8.3.8	15 (0x0F) Write Multiple Coils	38
8.3.9	16 (0x10) Write Multiple Registers	39
8.3.10	23 (0x17) Read/Write Multiple Registers.....	40
8.3.11	Error Response	41
8.4	Modbus Special Register Map	42
8.4.1	Adapter Register Mapping	42
8.4.2	Adapter Identification Special Register (0x1000, 4096).....	42
8.4.3	Adapter Information Special Register (0x1100, 4352).....	43
8.4.4	Adapter Setting Special Register (0x1600, 5632).....	43
8.4.5	Expansion Slot Information Special Resister (0x2000, 8192).....	45
A	APPENDIX - Modbus Reference	46
B	APPENDIX - Product List.....	46

1 Important Notes

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

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

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

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

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

Warning!

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

Caution!

- ✓ **If you disobey the instructions, there may be possibility of personal injury, damage to equipment or explosion. Please follow below Instructions.**
- ✓ Check the rated voltage and terminal array before wiring. Avoid the circumstances over 50°C of temperature. Avoid placing it directly in the sunlight.
- ✓ Avoid the place under circumstances over 85% of humidity.
- ✓ Do not place Modules near by the inflammable material. Else it may cause a fire.
- ✓ Do not permit any vibration approaching it directly.
- ✓ Go through module specification carefully, ensure inputs, Output connections are made with the specifications. Use standard cables for wiring.
- ✓ Use Product under pollution degree 2 environment.

1.1 Safety Instructions

1.1.1 Symbols

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

1.1.2 Safety Notes

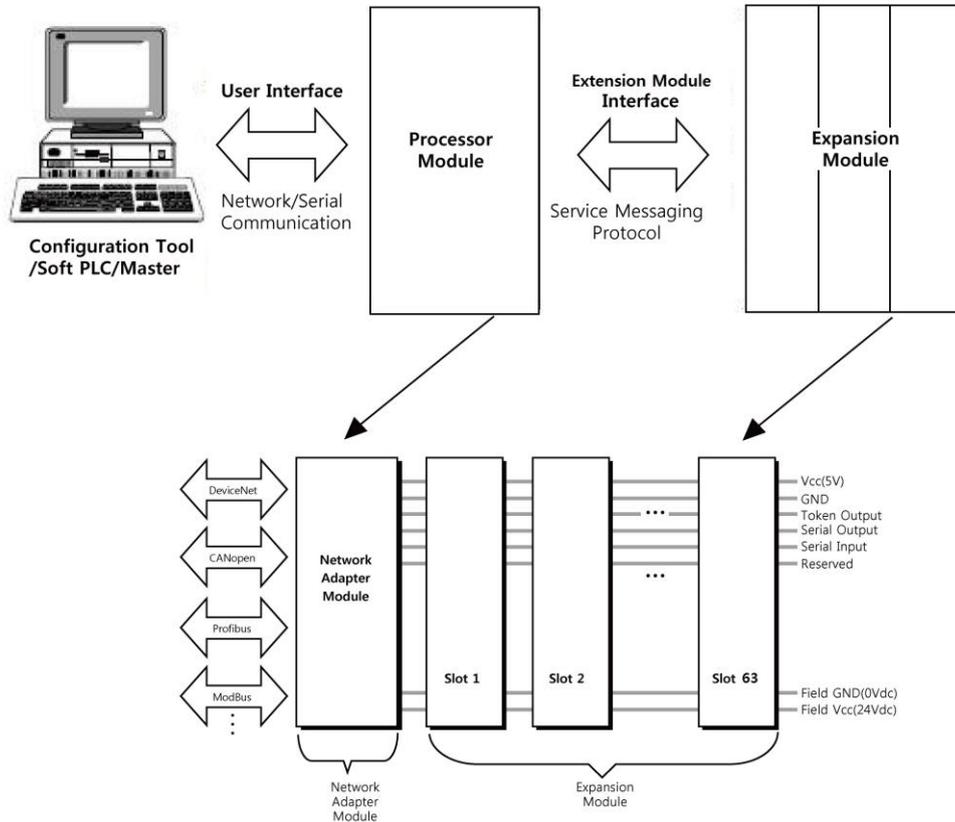
<p>DANGER</p> 	<p>The modules are equipped with electronic components that may be destroyed by electrostatic discharge. When handling the modules, ensure that the environment (persons, workplace and packing) is well grounded. Avoid touching conductive components, GBUS Pin.</p>
---	--

1.1.3 Certification



2 RIO3-Series System

2.1 Electrical Interface



Network Adapter Module

The Network Adapter Module forms the link between the field bus and the field devices with the Expansion Modules. The connection to different field bus systems can be established by each of the corresponding Network Adapter Module, e.g. for MODBUS TCP, Ethernet IP, EtherCAT, PROFINET, PROFIBUS, CANopen, MODBUS/Serial etc.

Expansion Module

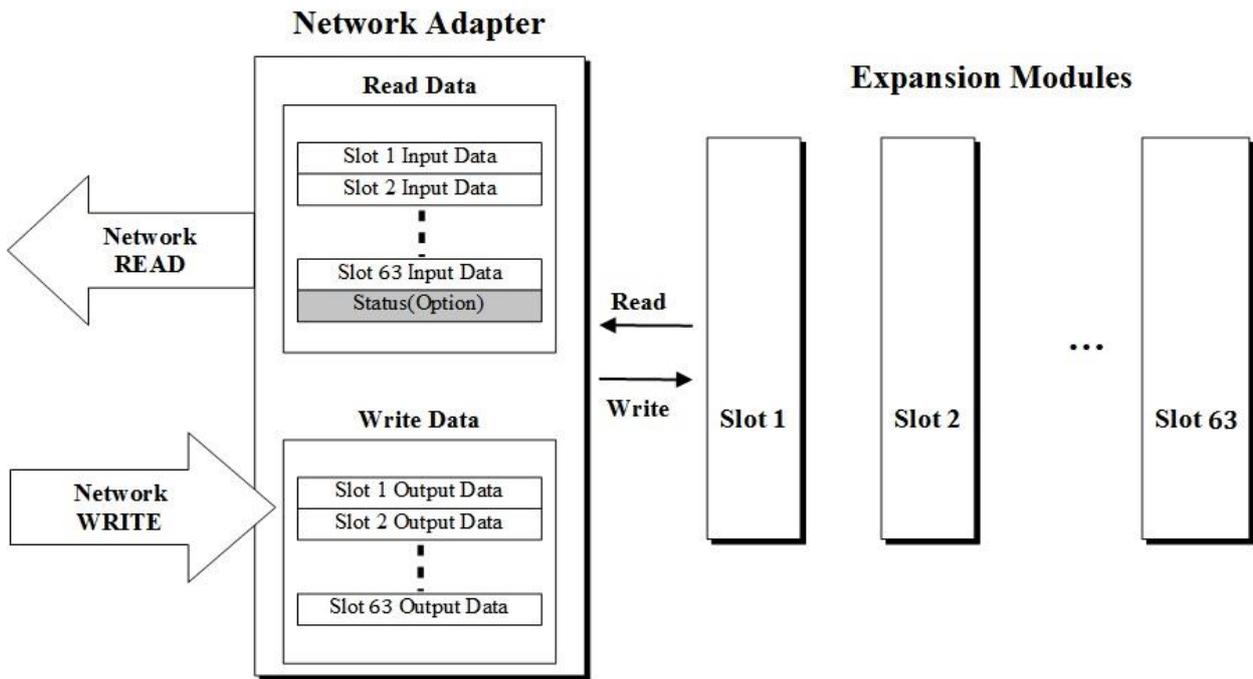
The Expansion Modules are supported a variety of input and output field devices. There are digital and analog input/output modules and special function modules.

Two types of Messages

Service Messaging / I/O Messaging

2.2 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 internal protocol. The following figure shows the data flow of process image between network adapter and expansion modules.



3 RIO3-CP2MBTL specification

3.1 General specification

General specification	
Items	Specification
Power Dissipation	50mA typical @ 24Vdc
UL System Power	Supply voltage: 24Vdc nominal, Class 2
System Power	Supply voltage: 24Vdc nominal Supply voltage range: 18 ~ 28.8Vdc Reverse polarity protection
UL Field Power	Supply voltage: 24Vdc nominal, Class 2
Field Power*	Supply voltage: 24Vdc typical (Max. 30Vdc)
Max. Current Field Power Contact	Max. DC 8A
Wiring	I/O Cable Max. 2.0mm ² (AWG 14)
Torque	0.8 Nm (7 lb-in)
Current for Expansion Module	1.0A @ 5Vdc
Isolation	System power to internal logic: non-isolation System power I/O driver: Isolation
Weight	84g
Module Size	22mm x 109mm x 70mm

3.2 Environment specification

Environmental specification	
Operating Temperature	-25°C~60°C
UL Temperature	-25°C~60°C
Storage Temperature	-40°C~85°C
Relative Humidity	5% ~ 90% non-condensing
Mounting	DIN rail
Vibration Resistance	
Shock Operating	IEC 60068-2-27
Vibration Resistance	IEC 60068-2-6, 4g
Industrial Emissions	EN 61000-6-4/A11: 2011
Industrial Immunity	EN 61000-6-2: 2019
Installation Position	Vertical and horizontal installation
Protection class	IP20
Product Certification	
Certifications	CE, UL

3.3 Programmable specification

Programmable Specification	
Items	Specification
Programming (CODESYS)	V3.5.17.3
Program Memory	16 MByte
Data Memory	16 MByte
Non-Volatile Memory	32 KByte Retain: 16 KByte Persistent: 16 KByte
Run-Time System	Multiple PLC Tasks
Program Languages	IEC 61131-3 (LD, IL, ST, FBD, SFC)
MQTT ¹⁾	✓
MQTT Sparkplug B	✓
SSL/TLS	X
User Management*	X
SNMP (Agent only) ²⁾	✓
OPC DA Server	✓
OPC UA Server & Client	✓
Modbus TCP/RTU	✓
Online Change ³⁾	✓
Source Upload/Download	✓
File system	✓
File transmit	✓
TFTP	✓
SQL4CODESYS	✓
Breakpoint	✓
RTC ⁴⁾	Retain Time: < 15 day / Accuracy: < 2min/month (Status: fully recharged battery at room temperature)
Max. Task	10
Max. Cycle Task	10
Max. Status Task	10
Max. Data Size (Input + Output)	Max. 128Byte each slot
Max. Expansion Module	63 Slots
Process Time	0.1440μsec

* Features Deleted from Latest Version.

In the latest version, security features are enhanced, and encryption must be added to use them. User management features and SSL/TLS are not supported because encryption features are not currently added.

1) MQTT does not support TLS.

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

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

*Precautions for Online Changes

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

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

4) RTC (at 25°C)

Battery charging time	Retain time at 25°C	*** RTC Warning There will be operating problems when the battery is discharged. - RTC data will not be stored and will be the initial value.
4 hours	Up to 2 days	
12 hours	Up to 12 days	
16 hours	Up to 15 days	

- Recommend charging for at least 16 hours when the battery is discharged.
- Retain time may vary depending on temperature and environment.
- The battery is one of the components on PCB in the plastic housing. It means that it can't be checked visually but can be charged by supplying the voltage to the system power.

3.4 Interface specification

Interface Specification	
RJ45 Ethernet port	
Adapter Type	Master & Slave node (Modbus TCP, Modbus RTU)
Baud rate	10/100Mbps, Auto-negotiation, Full Duplex
Ethernet Interface	RJ-45 socket * 1pcs
Ethernet Protocol	Modbus TCP, Modbus UDP, SNTP, SNMP, MQTT, DHCP/BOOTP, HTTP (Webserver), OPC-server
Max. Socket	UDP: 16, TCP: 64
RJ45 Serial port	
Serial Interface	RJ-45 socket * 1pcs
Serial Protocol	Modbus RTU Baud Rate: 2400~115200 bps, (Default: 115200 bps)
Indicator	
LED Indicator	4 LEDs 1 Green/Red, Module Status (MOD) 1 Green/Red, Network Status (NET) 1 Green/Red, PLC Run/Stop Status (RUN) 1 Green/Red, Expansion I/O Module Status (IOS)

3.5 Ethernet connection specification

Function*	Model	Max. number of concurrent communications
ARTI (OPC-server)	RIO3-CP2MBTL	One for each function are available at the same time
CODESYS link		
Network-variable		
Modbus/TCP Master		64 Modbus/TCP Slaves can be connected
Modbus/TCP Slave		64 Modbus/TCP Masters can be connected
Webserver		64 clients can be opened

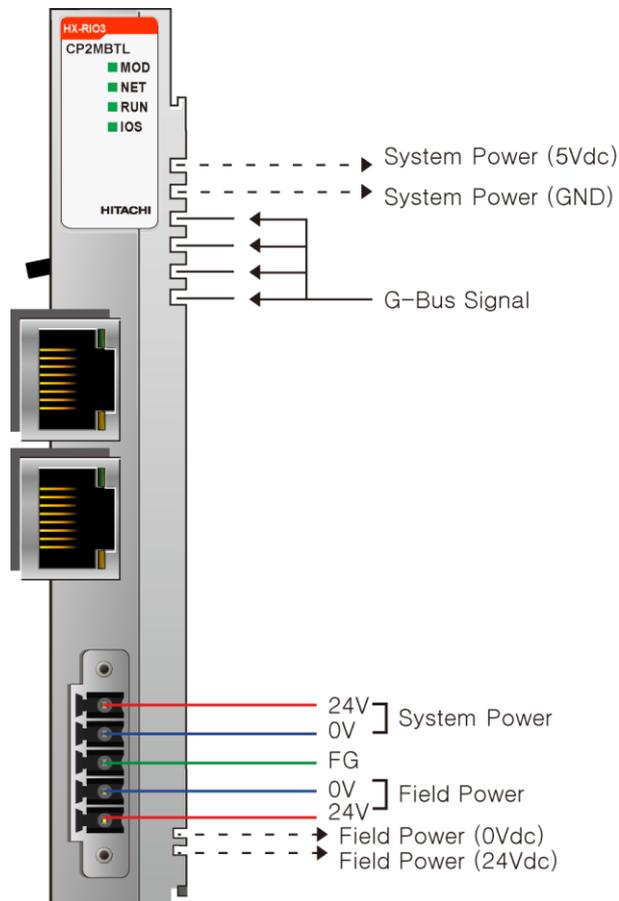
* While using these features, can use up to a maximum number of sockets (64) at the same time.

3.6 Serial connection specification

Function	Model	Max. number of concurrent communications
Modbus RTU Master	RIO3-CP2MBTL	RS-232: 1 Slave can be connected RS-485: 247 Slaves can be connected

4 Module Description

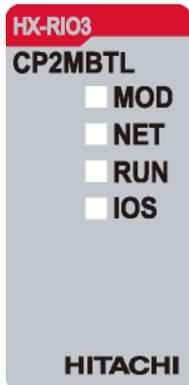
4.2 Wiring Diagram



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

The system power must not be connected with field power. Use separate voltage supplies.

4.3 LED Indicator



LED	LED Function / Description	LED Color
MOD	Module Status	Green/Red
NET	Current Communication Status	Green/Red
RUN	PLC Run / Stop	Green/Red
IOS	Expansion module	Green/Red

4.3.1 MOD (Module Status LED)

Status	LED is	To indicate
Not Powered	OFF	Power is not supplied to the unit.
Normal, Operational	Green	The unit is operating in normal condition.
IAP Mode*	Toggling Green & Red	Available for firmware download using Firefox.
Firmware Fault	Red	The unit has occurred unrecoverable fault in self-testing.
Recoverable Fault	Blinking Red	The unit has occurred recoverable fault in self-testing. - EEPROM checksum fault.

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

4.3.2 NET (Network Status LED)

Status	LED is	To indicate
Not Powered	OFF	Network off-line.
Link	Green	The LAN cable is physically connected to the ethernet port.

* Blinking Green MOD & NET LED: BOOTP/DHCP is requesting for new IP address.

4.3.3 RUN (PLC Run/Stop Status LED)

Status	LED is	To indicate
None PLC	OFF	Device has no program.
PLC Run	Green	PLC program is running.
PLC Stop	Blinking Green	PLC program stops.
Firmware Fault	Red	The unit has occurred unrecoverable fault in self-testing.
Diagnostic	Blinking Red	PLC program and expansion I/O modules do not match.

4.3.4 IOS (Expansion Module Status LED)

Status	LED is	To indicate
Expansion I/O	OFF	Device has no expansion modules or not powered.
Expansion I/O	Green	Device has expansion modules.
Configuration Fault	Red	Replace expansion modules or fail to initialize. <ul style="list-style-type: none"> - Detect invalid expansion module ID. - Initial protocol failure. - Mismatch vendor code between adapter and expansion module. - Changed expansion module configuration.
Connection Fault	Blinking Red	One or more expansion module occurred in fault state. <ul style="list-style-type: none"> - Too many expansion modules. - Communication failure. - Overflowed I/O size.

4.3.5 Field Power Status LED

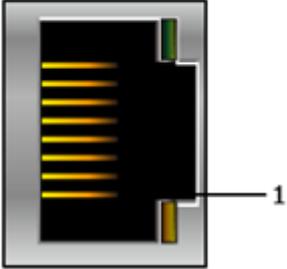
State	LED state	To indicate
Not Supplied Field Power	Off	Not supplied 24V dc field power.
Supplied Field Power	Green	Supplied 24V dc field power.

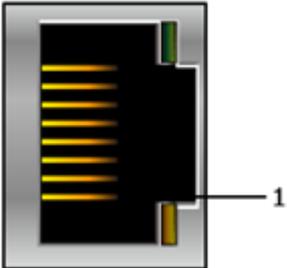
4.3.6 Exception indication

- 1) BOOTP/DHCP requests a new IP address. (You can change the IP setting mode. Refer to Modbus Register 0x160B.)
- 2) The IP Address to access IAP web-server during IAP Mode: 192.168.100.10 (Recommended to use FireFox)

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

4.4 RJ-45 Socket

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

Serial Socket 1pcs 	RJ-45*	Signal Name	Description
	1	D+	RS485 D+
	2	D-	RS485 D-
	3	GND	
	4	/ISP	/ISP
	5	-	
	6	GND	
	7	TXD	RS232 TXD
	8	RXD	RS232 RXD
	Case	Shield	

* COM Port 1: RS-232

COM Port 2: RS-485

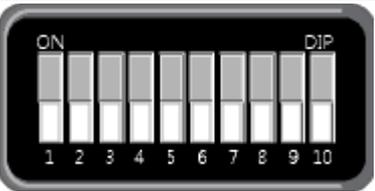
4.5 Toggle Switch and Push Button

	Toggle Switch Status	Module is	Description
	Up	RUN	PLC Run
	Down	STOP	PLC Stop

	Push Button	Module is	Description
	Push and detach	Reset	Reset CODESYS PLC program and make the program be in the stop status.
	Push for 5sec	PLC Reset	Erase CODESYS PLC program and retain memory.
	Push for 20sec	Factory Reset	Erase CODESYS PLC program and Network parameter reset.
	Push hold and Power Reset	IAP mode	Firmware download via Firefox.

4.6 DIP Switch

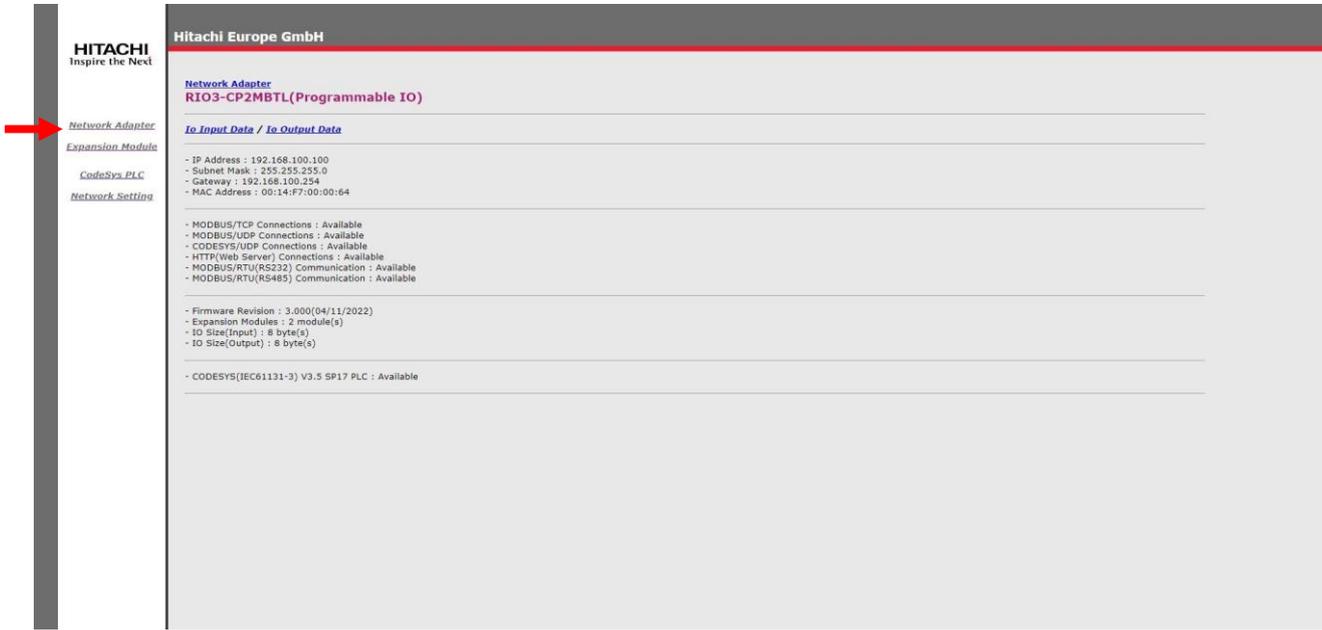
	DIP Switch Status		Description
	1	2	
	ON	ON	RS485 terminating resistor enabled (1 k Ω)
OFF	OFF	RS485 terminating resistor disabled	

	Description
	No function (Reserved)

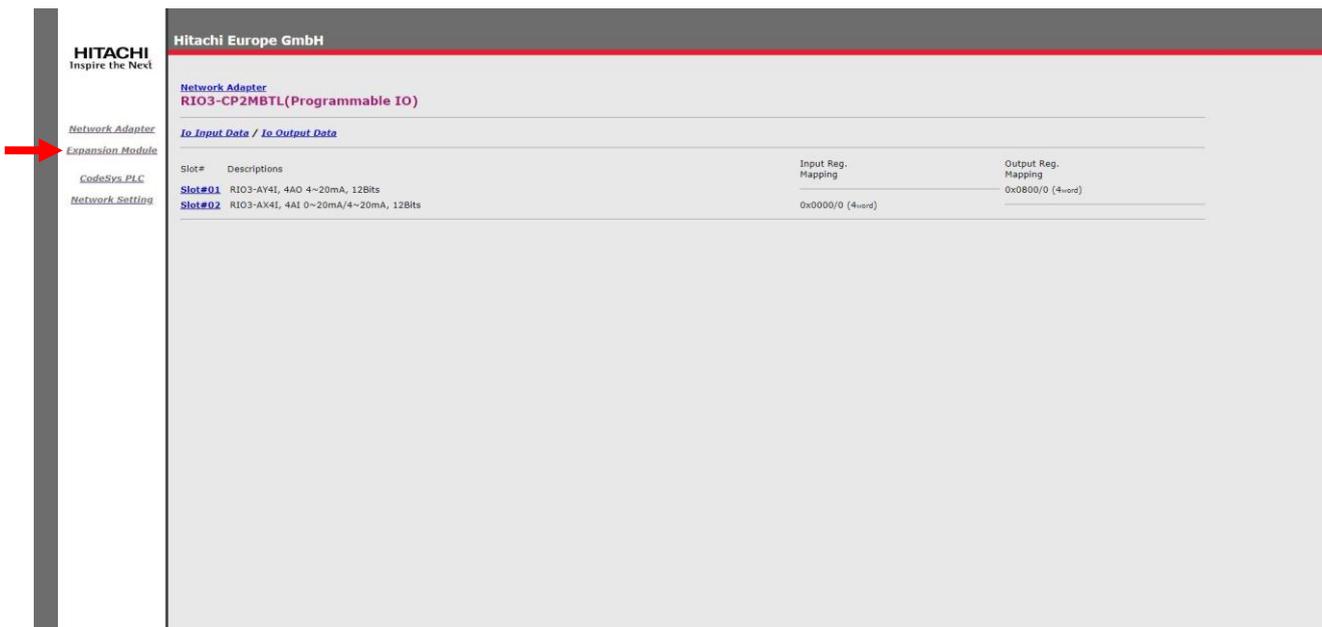
5 HX-RIO3 Series PIO (Programmable I/O) Functions

5.1 Webserver

- (1) Main page is showing various information for PIO status.
- (2) To access the webserver, <http://IPAddress/setup.htm> (e.g. "http://192.168.100.100/setup.htm")



- (3) Click on the expansion menu, user can check the extension module status.



- (4) User can easily change and set the RTC time. If you click the 'Get Time' button, the clock time from PC will be adapted to PIO.

HITACHI
Inspire the Next

Hitachi Europe GmbH

Network Adapter
RIO3-CP2MBTL(Programmable IO)

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

- Vendor Name : "Hitachi Europe GmbH"
- Vendor ID : 0x10AD
- Device ID : 0x3071
- Device Type : 0x1000

PLC Logic : "Application"
- Project Name : "RIO3-CP2MBTL"
- Author : ""
- Version : ""
- Description : ""
- Profile : "CODESYS V3.5 SP18"
- Last Updated Time : Aug 11, 2022 05:43:02(GMT+00)

- Switch(Run/Stop) : Run
- PLC Status : Run
- Retain Save Count : 0

- Current RTC Date: 2014-04-09 Time: 08:31:40

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

Click Button if you want to get Current time from PC

- (5) Using the Network setting, user can set the IP/Subnet mask/Gateway.

HITACHI
Inspire the Next

Hitachi Europe GmbH

Network Adapter
RIO3-CP2MBTL(Programmable IO)

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

Current IP Configuration
- IP Address : 192.168.100.100
- Subnet Mask : 255.255.255.0
- Gateway : 192.168.100.254
- MAC Address : 00:14:F7:00:00:64

Change IP Parameter
- IP address:
- Subnet mask:
- Gateway:

* DO NOT FORGET the new IP configuration before power reset! *
* Please write down the addresses before you forget it! *

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

5.2 Default Network Setting

When user forget the IP address, after power on the PIO and push the Reset switch in front of PIO.

The switch should be pushed for 20 seconds at least. PIO will be fall into the factory default mode.

All of PIO LED will blink in Green/Red.

Default Network Setting	
IP address	192.168.100.100
Subnet Mask	255.255.255.0
Gateway	192.168.100.254

5.3 RTC (Real Time Clock) Function

RTC can be checked & set using the following methods;

(1) CODESYS

Verify & change the current time.

- 2021Y 01M 01D 00h 00m 00s

- RTV value every 100ms

Expression	Type	Value
* FB_SetRTC	RTCLK.SetDateAndT...	
* FB_GetRTC	RTCLK.GetDateAnd...	
currentDateTime	DATE_AND_TIME	DT#2021-1-1-0:0:52
* TIMER	TON	
xInit	BOOL	FALSE

```

1 // RTC POU
2
3 IF xInit[FALSE] THEN
4   xInit[FALSE] := FALSE;
5   FB_SetRTC.dtDateAndTime[DT#2021-1-1-0:0:0] := DATE_AND_TIME#2021-01-01-00:00:00;
6   FB_SetRTC(xExecute[TRUE] := TRUE);
7 END_IF
8
9 FB_GetRTC(xExecute[FALSE] := TIMER.C[FALSE]);
10
11 IF FB_GetRTC.xDone[TRUE] THEN
12   currentDateTime[DT#2021-1-1-0:0:52] := FB_GetRTC.dtDateAndTime[DT#2021-1-1-0:0:52];
13 END_IF
14
15 TIMER(IN[TRUE] := NOT(TIMER.C[FALSE]), PT[T#100ms] := T#100MS);RETURN
  
```

(2) Web-Server

Verify & change the current time.

The screenshot displays the Hitachi web server interface for a Hitachi Europe GmbH device. The left sidebar contains navigation links: HITACHI Inspire the Next, Network Adapter, Expansion Module, CodeSys PLC, and Network Setting. The main content area shows the following information:

- Hitachi Europe GmbH**
- Network Adapter**
RIO3-CP2MBTL(Programmable IO)
- Io Input Data / Io Output Data**
- Vendor Name : "Hitachi Europe GmbH"
Vendor ID : 0x10AD
Device ID : 0x2071
Device Type : 0x1000
- PLC Logic : "Application"
Project Name : "RIO3-CP2MBTL"
Author : ""
Version : ""
Description : ""
Profile : "CODESYS V3.5 SP18"
Last Updated Time : Aug 11, 2022 05:43:02(GMT+00)
- Switch(Run/Stop) : Run
PLC Status : Run
Retain Save Count : 0
- Current RTC Date : 2014-04-09 Time : 06:31:40
- Enter RTC; (Please follow the date and time format)**
Date: Time:
- Change
- Click Button if you want to get Current time from PC
Get time

A red rectangular box highlights the 'Current RTC Date' section, the 'Enter RTC' instruction, and the input fields for date and time.

6 External Feature Library

6.1 Battery State of charge

- ❖ RTC data or RETAIN data will not be saved if the battery is not charged enough.



Scope	Name	Type	Comment
Output	xError	BOOL	TRUE: Battery charging required FALSE: Battery charged
Output	eErrorID	Error	(1) No_Err (2) Battery_Reset_RTC (3) Battery_Reset_Retain (4) Battery_Reset_RTC_And_Retain

(1) Code about Battery State

```

PROGRAM Battery
VAR
    Battery_FLAG : BOOL;
    Battery : ExtFeatures.LastBatteryStatus;
    Error : BOOL;
    ErrorCode : ExtFeatures.ERROR;
END_VAR

IF Battery_FLAG = TRUE THEN
    Battery();
    Error := Battery.xError;
    ErrorCode := Battery.eErrorID;
    Battery_FLAG := FALSE;
END_IF
    
```

(2) Result

- ❖ IF xError : It means "True", so the battery should be charged through supplying the system power.

Battery	ExtFeatures.LastBat...				
xError	BOOL	TRUE			
eErrorID	ERROR	BATTERY_RESET_RTC_AND_RETAIN			
Error	BOOL	TRUE			
ErrorCode	ERROR	BATTERY_RESET_RTC_AND_RETAIN			

6.2 Read / Write Special Register (PIO Parameter Access)

- ❖ The MODBUS Special Register Map listed in the specification can do “read / write” by programming logic through the library as the followings ;

6.2.1 Read Special Register



Scope	Name	Type	Comment
Input	wAddress	WORD	Special register address
	wSize	WORD	Word size to read
Output	wBuffer	ARRAY [0..127] OF WORD	read Buffer
	wRegSize	WORD	
	eErrorID	ERROR	(1) No_Err : No Error (2) Invalid_Address : Incorrect address – unsupported address (3) Invalid_Size : Incorrect access size

6.2.2 Write Special Register



Scope	Name	Type	Comment
Input	wAddress	WORD	Special register address
	wSize	WORD	Word size to write
	wBuffer	ARRAY [0..127] OF WORD	Write buffer
Output	wRegSize	WORD	
	eErrorID	ERROR	(1) No_Err : No Error (2) Invalid_Address : Incorrect address – unsupported address (3) Invalid_Size : Incorrect access size

6.2.3 Example to change Special Register

- (1) Test to change the RS-232 Baud rate – 7 (115200)

Address	Access	Type, Size	Description
0x1600(5632)	Read	2words	IP Address. (ex : A8C0 6464 = 192.168.100.100)
0x1602(5634)	Read	2words	Subnet Mask. (ex : FFFF 0000 = 255.255.0.0)
0x1604(5636)	Read	2words	Gate way. (ex : A8C0 0100 = 192.168.0.1)
0x1606(5638)*	Read /Write	1word	RS-232C Baud rate. (2400bps~115200bps) - 0 : 115200 (default) - 1 : 2400 - 2 : 4800 - 3 : 9600 - 4 : 19200 - 5 : 38400 - 6 : 57600 - 7 : 115200

- (2) Write Special Register sample logic
❖ After setting the value, the login & download is required.

- wAddress : Special Register Address = 16#1606
- wSize : Special Register size = 1
- wBuffer : buffer that perform write = 16#0007(115200)

```
(*===== Write Special Register =====*)
IF Write Flag THEN
  Write.wAddress := 16#1606; // special register address
  Write.wSize := 1; // word to write
  Write.wBuffer[0] := 16#0007;
  Write();
  Write_Flag := FALSE;
END_IF
-
```

- (3) Read Slot Parameter sample logic
❖ Log-in is required in order to verify that the Write is successful

- wAddress : Special Register Address = 16#1606
- wSize : Special Register size = 1
- wBuffer : buffer that perform Read

```
IF Read Flag THEN
  Read.wAddress := 16#1606; // special register address
  Read.wSize := 1; // word to read
  Read();
  Read_Flag := FALSE;
END_IF
```

wBuffer[0]	WORD	16#0007
wBuffer[1]	WORD	16#0000
wBuffer[2]	WORD	16#0000

6.3 Read / Write Slot Parameter

- ❖ **Caution** : Continuous writing of the I/O parameter can cause problems in the I/O slot. So it needs to be used only when required.

6.3.1 Read Slot Parameter



Scope	Name	Type	Comment
Input	bSlotAddress	BYTE	Slot address
	bSize	BYTE	Byte size to read
Output	bParamSize	BYTE	parameter size
	bBuffer	ARRAY [0..254] OF BYTE	read Buffer
	eErrorID	ERROR	(1) No_Err : No Error (2) Invalid_Address : Incorrect slot address – 0 or exceeded number of expansion slots (3) Invalid_Size : Incorrect parameter size – 0 or exceeded parameter size

6.3.2 Write Slot Parameter



Scope	Name	Type	Comment
Input	bSlotAddress	BYTE	Slot address
	bOffset	BYTE	Offset
	bSize	BYTE	Byte size to write
	bBuffer	ARRAY [0..254] OF BYTE	write buffer
Output	bParamSize	BYTE	parameter size
	eErrorID	ERROR	(1) No_Err : No Error (2) Invalid_Address : Incorrect slot address – 0 or exceeded number of expansion slots (3) Invalid_Size : Incorrect parameter size – (bSize or bSize+bOffset) > parameter size of slot

6.3.3 Example to access the extended I/O module parameter

RIO3-AXH4V

Valid Parameter length: 6 Bytes

Parameter Data

Bit No	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Byte 0	Voltage Range for Channel 0 (H00: 0~10Vdc, H01: 0~5Vdc, H02: 1~5Vdc)							
Byte 1	Voltage Range for Channel 1 (H00: 0~10Vdc, H01: 0~5Vdc, H02: 1~5Vdc)							
Byte 2	Voltage Range for Channel 2 (H00: 0~10Vdc, H01: 0~5Vdc, H02: 1~5Vdc)							
Byte 3	Voltage Range for Channel 3 (H00: 0~10Vdc, H01: 0~5Vdc, H02: 1~5Vdc)							
Byte 4	Filter Time (H00: Default Filter (20) / H01: Fastest ~ / H3E : Slowest)							
Byte 5	Not used (=00)							

(1) Create a WriteslotParam sample logic

- ❖ After setting the value, the login & download is required.

- bSlotaddress : Slot Address : 1 (if first slot : 1 / if 10 th slot : 10)
- bSize : Data size of the parameters of the Module : 6
- boffset : offset : 0
- bBuffer[0] : Channel 0 Parameter Byte : H0(0~10V)
- bBuffer[1] : Channel 1 Parameter Byte : H1(0~5V)
- bBuffer[2] : Channel 2 Parameter Byte : H2(1~5V)

```
// ##### Write slot parameter #####
IF Write_Flag THEN
    Writeslotparam.bslotAddress := 1; // Slot Address (First slot)
    Writeslotparam.bSize       := 6; // number of bytes to write
    Writeslotparam.boffset     := 0; // offset
    // HEX 00 : 0~10V / H01 : 0~5V / H02 : 1~5V
    Writeslotparam.bBuffer[0] := 16#00; //parameter byte 0
    Writeslotparam.bBuffer[1] := 16#01; //parameter byte 1
    Writeslotparam.bBuffer[2] := 16#02; //parameter byte 2
    Writeslotparam.bBuffer[3] := 16#00; //parameter byte 3
    Writeslotparam.bBuffer[4] := 16#00; //parameter byte 4
    Writeslotparam();

```

(2) Create a ReadslotParam sample logic

- ❖ Log-in is required in order to verify that the Write is successful

- bsize : Set the parameter data size to read
- bSlotaddress : Slot Address(if first slot : 1 / if 10 th slot : 10)

```
IF Read_Flag THEN
    MemUtils.memset(ADR(Read_Data[0]), 0, sizeof(Read_Data))
    ReadslotParam.bSlotAddress := 1; // Slot Address (First slot)
    ReadslotParam.bSize := 6; // number of bytes to read
    ReadslotParam();

    MemUtils.SysMemCpy(ADR(Read_Data[0]),ADR(ReadslotParam.bBuffer));
    Read_Flag := FALSE;
END_IF
```

Read_Data	TYPE	HEX VALUE
Read_Data[0]	BYTE	16#00
Read_Data[1]	BYTE	16#01
Read_Data[2]	BYTE	16#02
Read_Data[3]	BYTE	16#03
Read_Data[4]	BYTE	16#00
Read_Data[5]	BYTE	16#00

7 Upgrade Firmware

7.1 Using IAP over Ethernet

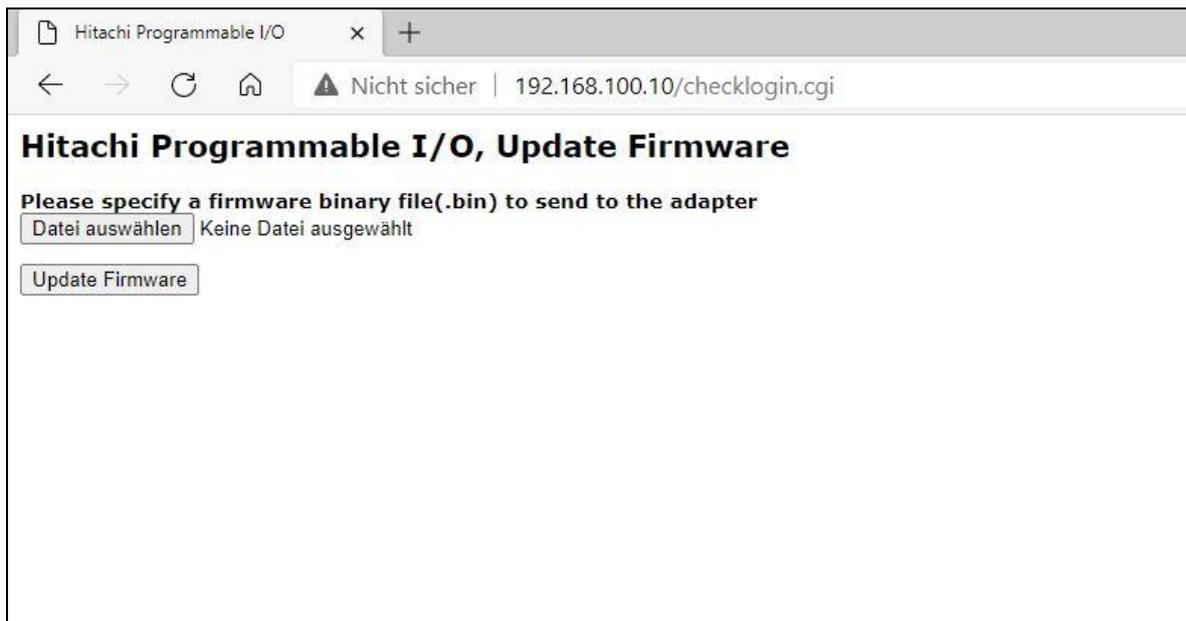
- (1) Apply a power with pushing a reset button(Mod LED will blink Green/Red).
- (2) Execute Firefox.(It is recommended to use Firefox)
- (3) Connect to **192.168.100.10** and login (User ID :**Hitachi** / Password : **Hitachi**)



The screenshot shows a web browser window with the title "Hitachi Programmable I/O, Firmv". The address bar displays "Nicht sicher | 192.168.100.10". The main content area is titled "Hitachi Programmable I/O, Login" and contains the following elements:

- Text: "Enter user ID & password:"
- Form fields: "User ID" and "Password", both with empty input boxes.
- Button: "Login Adapter"

- (4) Search the file to download using 'search button'.



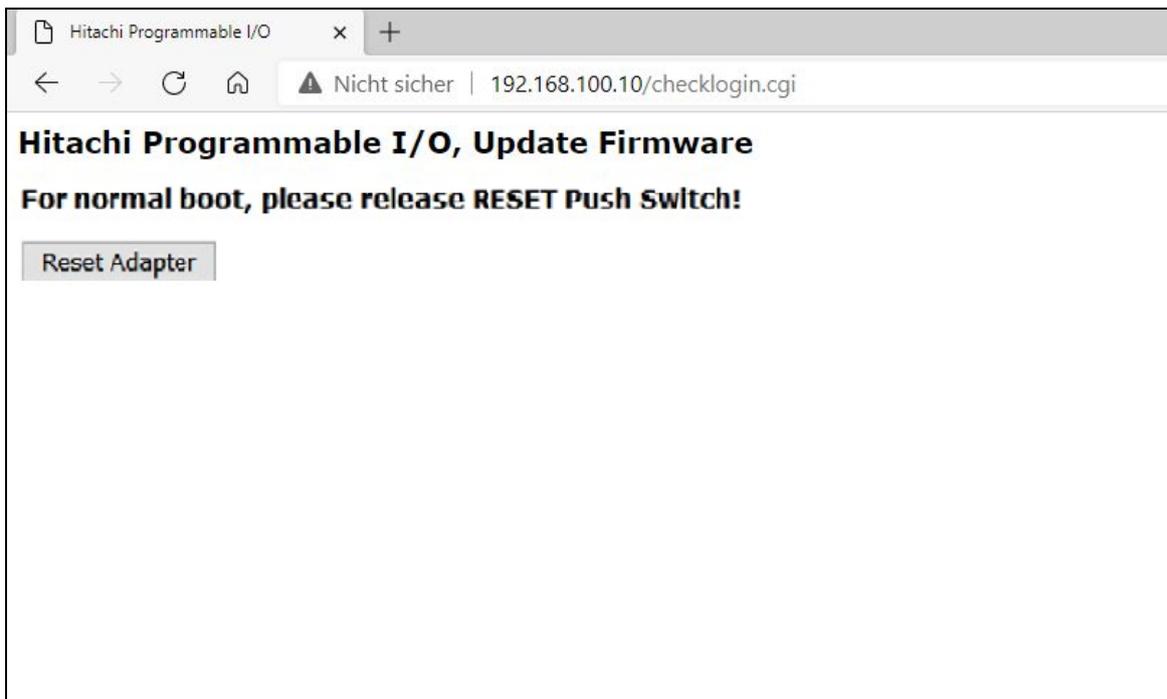
The screenshot shows a web browser window with the title "Hitachi Programmable I/O". The address bar displays "Nicht sicher | 192.168.100.10/checklogin.cgi". The main content area is titled "Hitachi Programmable I/O, Update Firmware" and contains the following elements:

- Text: "Please specify a firmware binary file(.bin) to send to the adapter"
- Form field: "Datei auswählen" with the text "Keine Datei ausgewählt" next to it.
- Button: "Update Firmware"

(5) Click 'Upload Button'.



(6) If it finish, you can see a below message (File Upload Done!). Click 'Reset Adapter' button.



8 Modbus Interface

8.1 MODBUS Interface Register / Bit Map

8.1.1 Register Map

Start Address	Read/Write	Description	Func. Code
0x0000 ~	Read	Process input image registers (Real Input Register)	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).

8.1.2 Bit Map

Start Address	Read/Write	Description	Func. Code
0x0000 ~	Read	Process input image bits All input registers area is addressable by bit address. Size of input image bit is size of input image register * 16.	2
0x0800 ~	Read/Write	Process output image bits All output registers area is addressable by bit address. Size of output image bit is size of output image register * 16.	1, 5, 15

8.2 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.

8.2.1 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 baudrate. Each message must be transmitted in a continuous stream of characters.

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

8.2.2 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 requirement regarding timers management.

Start	Address	Function	Data	CRC Check	End
1 char	2 chars	2 chars	Up to 252 chars	2 chars	2 chars CR,LF

8.3 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

8.3.1 1 (0x01) Read Coils

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

- Request

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

- **Response**

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

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

8.3.2 2 (0x02) Read Discrete Inputs

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

- **Request**

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

- **Response**

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

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

8.3.3 3 (0x03) Read Holding Registers

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

- **Request**

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

- **Response**

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

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

8.3.4 4 (0x04) Read Input Registers

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

- Request

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

- Response

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

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

8.3.5 5 (0x05) Write Single Coil

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

- Request

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

- Response

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

Output bit of address 0x1001 turns ON

8.3.6 6 (0x06) Write Single Register

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

- **Request**

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

- **Response**

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

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

8.3.7 8 (0x08) Diagnostics

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

- **Request**

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

- **Response**

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

Sub-function 0x0000 (0) Return Query Data

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

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

Sub-function 0x0001 (1) Restart Communications Option

The remote device could be initialized and restarted, and all of its communications event counters are cleared. Especially, data field 0x55AA make the remote device to restart with factory default setup of EEPROM.

Sub-function	Data Field (Request)	Data Field (Response)	Description
0x0001 (1)	0x0000 or 0xFF00	Echo Request Data	Reset
0x0001 (1)	0x55AA	Echo Request Data	Reset with Default Setting ¹⁾
0x0001 (1)	0x55AA+0xAB7B+Sumcheck ³⁾	Echo Request Data	Reset with Factory default ^{1) 2)}

1) Delete PLC program.

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

3) Refer to 3.2.2 for Sum check (0x1006)

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 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.

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

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

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

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

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

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

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

Sub-function 0x0064 (100) Return Slave Modbus, Expansion Status

The response data field returns the status of Modbus and expansion addressed to the remote device.

This status values are identical with status 1word of input process image.

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

8.3.8 15 (0x0F) Write Multiple Coils

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

- Request

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

- Response

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

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

8.3.9 16 (0x10) Write Multiple Registers

This function code is used to write a block of contiguous registers (1 to approx. 120 registers) in a remote device. The requested written values are specified in the request data field. Data is packed as two bytes per register. The normal response returns the function code, starting address, and quantity of registers written.

- **Request**

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

- **Response**

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

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

8.3.10 23 (0x17) Read/Write Multiple Registers

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

- **Request**

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

- **Response**

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

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

8.3.11 Error Response

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

Exception Response Example

Field name	Example
Function Code	0x81
Exception Code	0x02

Exception Codes

Field name	Example	Description
01	Illegal Function	The function code received in the query is not an allowable action for the server (or slave).
02	Illegal Data Address	The data address received in the query is not an allowable address for the server (or slave).
03	Illegal Data Value	A value contained in the query data field is not an allowable value for server (or slave).
04	Slave Device Failure	An unrecoverable error occurred while the server (or slave) was attempting to perform the requested action.
06	Slave Device Busy	Specialized use in conjunction with programming commands. The server (or slave) is engaged in processing a long-duration program command. The client (or master) should retransmit the message later when the server (or slave) is free.
01	Illegal Function	The function code received in the query is not an allowable action for the server (or slave).

8.4 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 each address (one address).

8.4.1 Adapter Register Mapping

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

8.4.2 Adapter Identification Special Register (0x1000, 4096)

Address	Access	Type, Size	Description
0x1000 (4096)	Read	1word	Vendor ID = 0x10AD, Hitachi Europe GmbH
0x1001 (4097)	Read	1word	Device type = 0x1000, Network Adapter
0x1002 (4098)	Read	1word	Product Code
0x1003 (4099)	Read	1word	Firmware revision, if 0x0101, revision 1.001
0x1005 (4101)	Read	String up to 34bytes	Product name string First 1word is length of valid character string Example) response as following "00 0c 52 49 4f 33 2d 43 50 32 4d 42 54 4c 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00" Valid character size = 0x000C = 12 characters " RIO3-CP2MBTL"
0x1006 (4102)	Read	1word	Sum check of EEPROM
0x1010 (4112)	Read	2words	Firmware release date
0x1013 (4115)	Read	1word	Module ID
0x101E (4126)	Read	15words	Composite Id of following address 0xA8C0(Lo_IP Addr), 0x3264(Hi_IP Addr), 0xFFFF(Lo_NetMask), 0x00FF(Hi_NetMask), 0xA8C0(GateWay), 0xFE64(GateWay), 0x1400(MacAddr), 0x00F7(MacAddr), 0xBA83(MacAddr), 0x02E5(VendorCode), 0x000C(DeviceType), 0x91F0(ProductCode), 0x0203(FW_Rev), 0x0510(FW_ReleasData), 0x2021(FW_ReleasYear)

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

8.4.3 Adapter Information Special Register (0x1100, 4352)

Address	Access	Type, Size	Description
0x1102(4354)	Read	1word	Start address of input image word register.=0x0000
0x1103 (4355)	Read	1word	Start address of output image word register. =0x0800
0x1104 (4356)	Read	1word	Size of input image word register.
0x1105 (4357)	Read	1word	Size of output image word register.
0x1106(4358)	Read	1word	Start address of input image bit. = 0x0000
0x1107 (4359)	Read	1word	Start address of output image bit. =0x1000
0x1108 (4360)	Read	1word	Size of input image bit.
0x1109(4361)	Read	1word	Size of output image bit.
0x110D (4365)*	Read	1word	Switch State (Dip Switch, Run, Stop, Reset)
0x110E (4366)	Read	Up to 63words	Expansion slot's RIO3-number
0x1110 (4368)	Read	1word	Number of expansion slot.
0x1113 (4371)	Read	Up to 63words	Expansion slot module id. First 1word is product code
0x111E (4382)	Read	1word	Reserved. Adapter IO identification vendor code.

*Address: 110D

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	DIP10	DIP9	DIP8	DIP7	DIP6	DIP5	DIP4	DIP3	DIP2	DIP1	RESET	RUN/ STOP

8.4.4 Adapter Setting Special Register (0x1600, 5632)

Address	Access	Type, Size	Description
0x1600 (5632)	Read	2words	IP Address (COA8 6464 = 192.168.100.100)
0x1602 (5634)	Read	2words	Subnet Mask (FFFF 0000 = 255.255.0.0)
0x1604(5636)	Read	2words	Gateway (COA8 0001 = 192.168.0.1)
0x1606 (5638)*	Read/Write	1word	RS-232C Baud rate. (2400bps~115200bps) 0 : 115200 (default) 1 : 2400 2 : 4800 3 : 9600 4 : 19200 5 : 38400 6 : 57600 7 : 115200
0x1607 (5639)*	Read/Write	1word	*RS-232C Setting. 1 nibble : Data bit(0 : 8bit(default), 1 : 9bit, 2: 7bit) 2 nibble : Stop bit(0 : 1bit(default), 1 : 2bit) 3 nibble : Parity bit(0 : none(default), 1: odd, 2 : even) 4 nibble : Reserve
0x1608 (5640)*	Read/Write	1word	RS-485 Baud rate. (2400bps~115200bps)

			0 : 115200 (default) 1 : 2400 2 : 4800 3 : 9600 4 : 19200 5 : 38400 6 : 57600 7 : 115200
0x1609 (5641)*	Read/Write	1word	RS-485 Setting. 1 nibble : Data bit(0: 8bit(default), 1: 9bit, 2: 7bit) 2 nibble : Stop bit(0: 1bit(default), 1: 2bit) 3 nibble : Parity bit(0: none(default), 1: odd, 2: even) 4 nibble : Reserve
0x160A (5642)*	Read/Write	1word	**Modbus Station. High 1byte : Station No. of RS-232C (default: 1) Low 1byte : Station No. of RS-485 (default: 1)
0x160B (5643)	Read/Write	1word	IP Setting Method. BootP/DHCP disable 0x0000 BootP: 0x8000 (default) DHCP: 0x8001
0x1610 (5648)	Read	3words	Mac Address (1400 00F7 0101 = 00.14.F7.00.01.01)
0x1614 (5652)*	Read	1word	Serial connection Method 0x0000 : Crevis Modbus/RTU (default) 0x8000 : RS232 Enable for CODESYS Function block / RTU Master. CODESYS Serial Port Configuration Setting: COM Port 1 0x8001 : RS485 Enable for CODESYS Function block / RTU Master. CODESYS Serial Port Configuration Setting: COM Port 2
0x1620 (5664)	Read/Write	4words	RTC 1 word : 00ss (ss : sec) 2 word : hhmm (hh : hour, mm : min) 3 word : mmdd (mm : month, dd : day) 4 word : yyyy (yyyy : year) (0010 0F28 0317 07E0 = 2016 - 03.23 - 15:40 - 16)

* **RS-232C/485 Setting:** This description for 0x1607 / 0x1609 register with bit.

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

** **Modbus Station:** This description for 0x160A register with bit.

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

8.4.5 Expansion Slot Information Special Resistor (0x2000, 8192)

Adr Offset	Exp. Slot#1	Exp. Slot#2	Exp. Slot#3	Exp. Slot#4	Exp. 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)

Adr Offset	Access	Type, Size	Description
0x02 (+2) **	Read	1 word	Input start register address of input image word this slot.
0x03 (+3) **	Read	1 word	Input word's bit offset of input image word this slot.
0x04 (+4) **	Read	1 word	Output start register address of output image word this slot.
0x05 (+5) **	Read	1 word	Output word's bit offset of output image word this slot.
0x06 (+6) **	Read	1 word	Input bit start address of input image bit this slot.
0x07 (+7) **	Read	1 word	Output bit start address of output image bit this slot.
0x08 (+8) **	Read	1 word	Size of input bit this slot
0x09 (+9) **	Read	1 word	Size of output bit this slot
0x0A (+10) **	Read	n word	Read input data this slot
0x0B (+11) **	Read/Write	n word	Read/write output data this slot
0x0E (+14)	Read	1 word	RIO3-number
0x0F (+15)	Read	String up to 72bytes	First 1word is length of valid character string. If RIO3-YTP8, returns "00 15 52 49 4F 33 2D 59 54 50 38 2C 20 38 44 4F 2C 20 32 34 56 64 63 00 00 00 00 00 00 00 00 00 00" Valid character size = 0x001E =21 characters, "RIO3-YTP8, 8DO, 24Vdc"
0x10 (+16)	Read	1 word	Size of configuration parameter byte
0x11 (+17) **	Read/Write	n word	Read/write Configuration parameter data, Refer to each IO parameter Specification.
0x17 (+23)	Read	2 words	Firmware Revision 0x00010010 (Major revision 1 / Minor revision 2, Rev 1.02)
0x19 (+25)	Read	2 words	Firmware release data.

* 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.

A APPENDIX - Modbus Reference

Modbus Reference Documents

<http://www.modbus.org>

Modbus Tools

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

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

B APPENDIX - Product List

Please refer the separate HX-RIO3 product list document