CONTEC

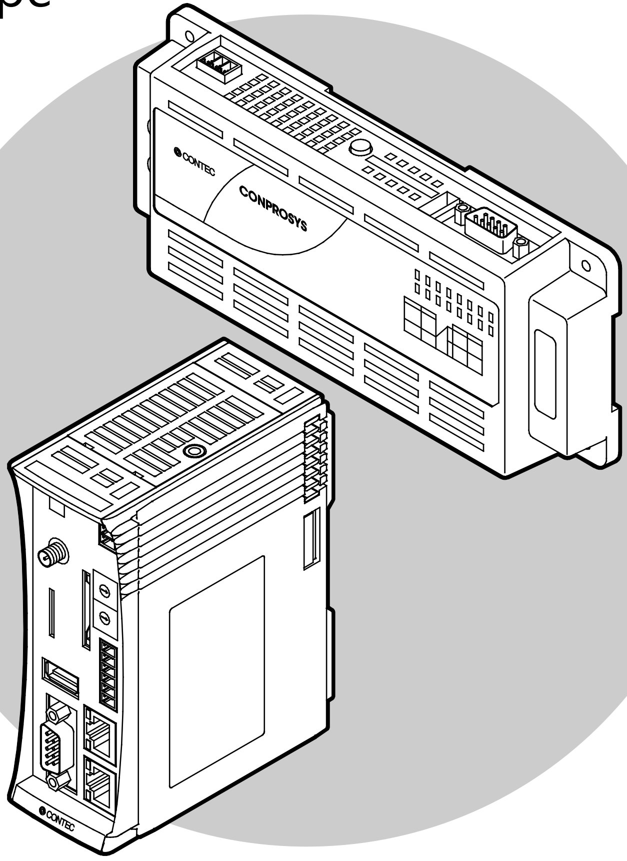
CONPROSYS

System Setup Guide

PAC Series

Integrated Type / Configurable Type

CONTENTS



CONTEC CO., LTD.

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Introduction

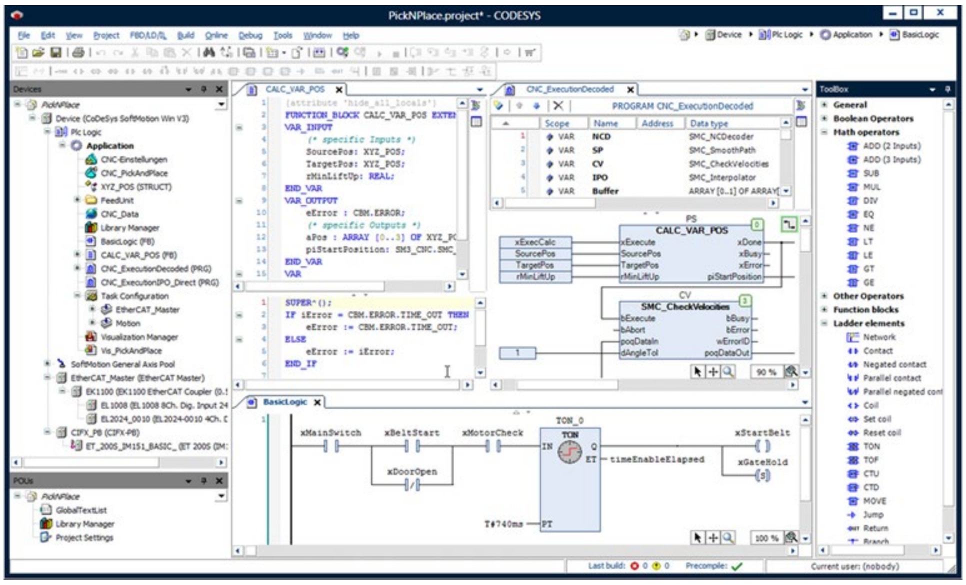
This reference manual guides how to configure the software in the product. The following chapter provides necessary information of the product such as the product configuration and manuals before actual use.

1.About the Product

The PAC series model has pre-installed software PLC “CODESYS”.

1. PAC series functionality

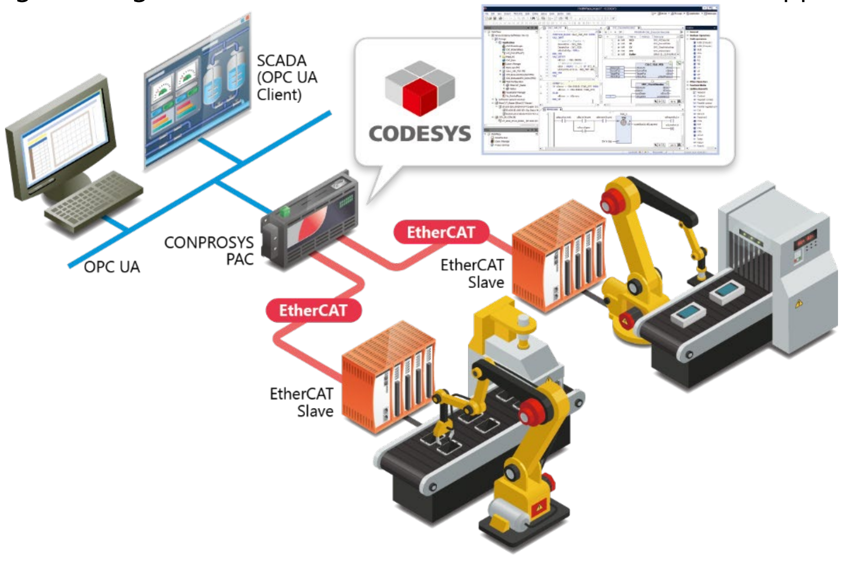
◆ Support programming by “CODESYS” that complies with international standard IEC61131-3



```
{attribute 'hide_all_locals'}
FUNCTION_BLOCK CALC_VAR_POS EXTE
VAR_INPUT
    (* specific Inputs *)
    SourcePos: XYZ_POS;
    TargetPos: XYZ_POS;
    rMinLiftUp: REAL;
END_VAR
VAR_OUTPUT
    eError : CBM.ERROR;
    (* specific Outputs *)
    aPos : ARRAY [0..3] OF XYZ_P
    piStartPosition: SM3_CNC.SMC_
END_VAR
VAR
```

```
SUPER^();
IF iError = CBM.ERROR.TIME_OUT THEN
    eError := CBM.ERROR.TIME_OUT;
ELSE
    eError := iError;
END_IF
```

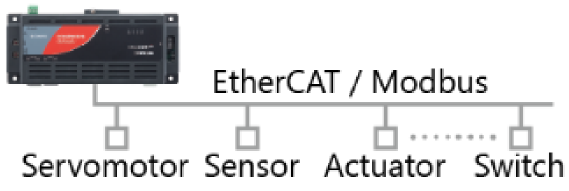
- An integrated development environment “CODESYS” for PLC programming, field bus setting can be downloaded free of charge.
- As “CODESYS” is complied with international standard IEC61131-3, it opens communication and makes it possible to select such as I/O or drive peripherals without depending on the specific communication methods.
- Six programming languages including ST, LD, FBD, SFC, IL, and CFC are supported. Also, object oriented programming defined as the third edition of IEC61131-3 are supported.



◆ Fieldbus I/O can be directly assigned to variables

EtherCAT/Modbus Fieldbus support

- Open field network EtherCAT/Modbus master functions within.
  In the CODESYS integrated development environment, fieldbus I/O can be directly assigned to variables in the same manner as the slave I/O.



◆ Device information can be easily monitored through a web browser

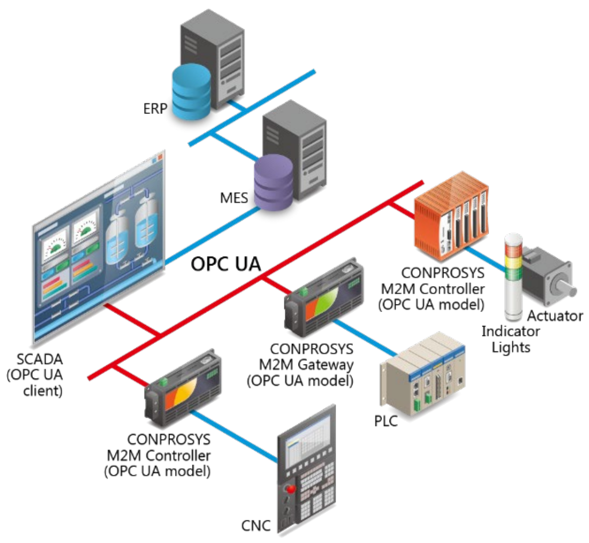
Web monitoring function

- Monitoring screens can be developed smoothly.
- Devices can be viewed through a web browser without the use of a server.

◆ Steady data exchange with SCADA/MES/ERP system

OPC UA server function (SCADA/MES/ERP linking support)

- The built-in OPC UA server provides the ability to exchange data between the controller and SCADA software and MES/ERP system securely and steadily.



2.Manual composition

This manual is composed as follows:

Chapters	Descriptions
Introduction	Manuals on the product are introduced. Read them as required.
Safety Precautions	Safety precautions are listed.
Set the Computer Network	**PC setup** This chapter describes the network settings with PC before using the product.
CONPROSYS WEB Setting	**CONPROSYS WEB Setting** This chapter describes the function settings in the CONPROSYS WEB Setting.
Easy Data Process and Control	
CODESYS Installation	**Detailed usage with "CODESYS"** This chapter describes of "CODESYS" settings and the functions.
Basic Programming	
Communication Settings	
Monitoring Edit	
Set Up Troubleshooting	This chapter describes troubleshooting when the product does not function properly.
System Reference	This chapter describes hardware specifications and CONPROSYS HMI specifications.
Appendix	The specification of the data format and others are listed.
Customer Support and Inquiry	Services and inquiry.
Index	Index of the manual.

3.Procedure until ready to use

The followings show the standard procedure until the product is ready to use.

Connect with a PC	Refer to **page 19**
↓	
Set up the network of PC	Refer to **page 20**
↓	
Set up with CONPROSYS WEB Setting	Refer to **page 25**
↓	
Set Username and Password	Refer to **page 30**
↓	
Save the settings	Refer to **page 36**
↓	
Install the CODESYS into a host PC	Refer to **page 46**
↓	
Install CODESYS Package	Refer to **page 48**
↓	
Programming with CODESYS	Refer to **page 53**
↓	
Set up the product	Refer to Reference Manual (Hardware)

*This procedure can be different depending on the user's environment or system types.

4. Related manuals

The manuals related to the product are listed below.

Read them as necessary along with this document.

◆ Must read the followings.

Name	When to read	Contents	How to get
Product Guide	Must read this after opening the package.	This lists the product configuration and describes the precautions.	Included in the package (Printed matter)
Setup Manual	Read this when setting up the product.	This describes the required items for setup and configuration procedure.	Download from the Contec website (PDF)
Reference Manual (Hardware)	Read this when operating the product.	This describes the hardware aspects such as functions and settings.	Download from the Contec website (PDF)
Reference Manual (Software) or System Setup Guide	Read this when setting up the "CONPROSYS WEB Setting".	This describes how to set each function of "CONPROSYS WEB Setting".	Download from the Contec website (PDF)

◆ Download manuals

Download the manuals from the following URL.

Download https://www.contec.com/download/

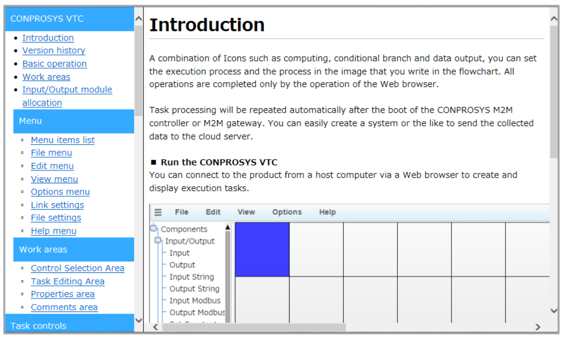
5.Online Help

We offer the detailed information on "CONPROSYS VTC" for assembling processing tasks such as calculation and control as well as on "CONPROSYS HMI" for operating and editing the monitoring screen through the Online Help.

Consult the Online Help as necessary.

◆ CONPROSYS VTC (Visual Task Control).

Online Help http://data.conprosys.com/help/task/V1/en/



CONPROSYS VTC

- Introduction
- Version history
- Basic operation
- Work areas
- Input/Output module allocation

Menu

- Menu items list
- File menu
- Edit menu
- View menu
- Options menu
- Link settings
- File settings
- Help menu

Work areas

- Control Selection Area
- Task Editing Area
- Properties area
- Comments area

Task controls

Introduction

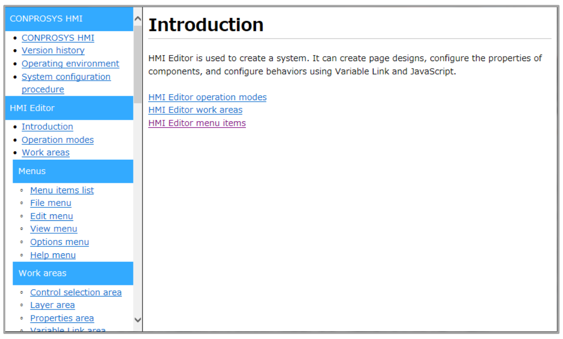
A combination of Icons such as computing, conditional branch and data output, you can set the execution process and the process in the image that you write in the flowchart. All operations are completed only by the operation of the Web browser.

Task processing will be repeated automatically after the boot of the CONPROSYS M2M controller or M2M gateway. You can easily create a system or the like to send the collected data to the cloud server.

■ **Run the CONPROSYS VTC**
You can connect to the product from a host computer via a Web browser to create and display execution tasks.

◆ CONPROSYS HMI (Human Machine Interface)

Online Help http://data.conprosys.com/help/hmi/V1/en/



CONPROSYS HMI

- CONPROSYS HMI
- Version history
- Operating environment
- System configuration procedure

HMI Editor

- Introduction
- Operation modes
- Work areas

Menus

- Menu items list
- File menu
- Edit menu
- View menu
- Options menu
- Help menu

Work areas

- Control selection area
- Layer area
- Properties area
- Variable Link area

Introduction

HMI Editor is used to create a system. It can create page designs, configure the properties of components, and configure behaviors using Variable Link and JavaScript.

HMI Editor operation modes
HMI Editor work areas
HMI Editor menu items

6.Check the firmware version

Before running the product, visit our website to check the firmware version and update to the latest one if necessary.
Updating firmware to the latest version will resolve troubles and stabilize the operation.

Download https://www.contec.com/download/

* Refer to the “Firmware version up **(page 42)**” for further details.

Safety Precautions

Understand the following definitions and precautions to use the product safely. Never fail to read them before using the product.

1. Safety Information

This document provides safety information using the following symbols to prevent accidents resulting in injury or death and the destruction of equipment and resources.

Understand the meanings of these labels to operate the equipment safely.

⚠**DANGER**	DANGER indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.
⚠**WARNING**	WARNING indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.
⚠**CAUTION**	CAUTION indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury or in property damage.

2. Handling Precautions

⚠ CAUTION

- The specifications of the product are subject to change without notice for enhancement and quality improvement. Even when using the product continuously, be sure to read the manual in the CONTEC's website and understand the contents.
- Do not modify the software.
  CONTEC will bear no responsibility for any problems, etc., resulting from modifying the software.
- Regardless of the foregoing statement, CONTEC assumes no responsibility for any errors that may appear in this document or for results obtained by the user as a result of using the software.

3. Security Warning

When connecting to the network, be aware of security-related problems. See the examples of Security measures below and set up the product properly along with the network devices.

1. Information security risks

- Unauthorized access from the outside through a network could cause the system halt, data damage, or exposure to malware. *1
- Invaded and used as a stepping stone, a device might attack the others through networks. (a victim becomes an assailant)
- Information might leak without realizing due to the connection to the network.
- Secondary damages such as harmful rumors, liability in damages, social credibility fall, and opportunity loss are expected led by the troubles described above.

*1: Malware (Malicious Software) is software that brings harm to a computer system and performs unintended operations.

2. Security measures – e.g.

- Do not keep using the default password. (Refer to the product manual for the password setting).
- Set a strong password.

> Combined with upper and lowercase letters, and numbers so that it cannot be easily analogized by others.

- Change the password periodically.
- Disable unnecessary network services and functions.
- Restrict access to the network with network devices. *2
- Restrict ports to be released on the network with network devices. *2
- Create a closed network connection using such as dedicated network or VPN*3

*2: Inquire for setting procedure to manufacturers.

*3: VPN (Virtual Private Network) a secured network that wards off unauthorized access by protecting the communication path with authentication and encryption.

> Unfortunately, there are no perfect ways to avert unauthorized access or close a security hole that are endlessly found day and night.
> Please understand that risks are always involved with the Internet connection, and we strongly recommend a user should constantly update information security measures.

Set the Computer Network

This chapter describes how to connect the product with a PC, set the network, and check the communication.

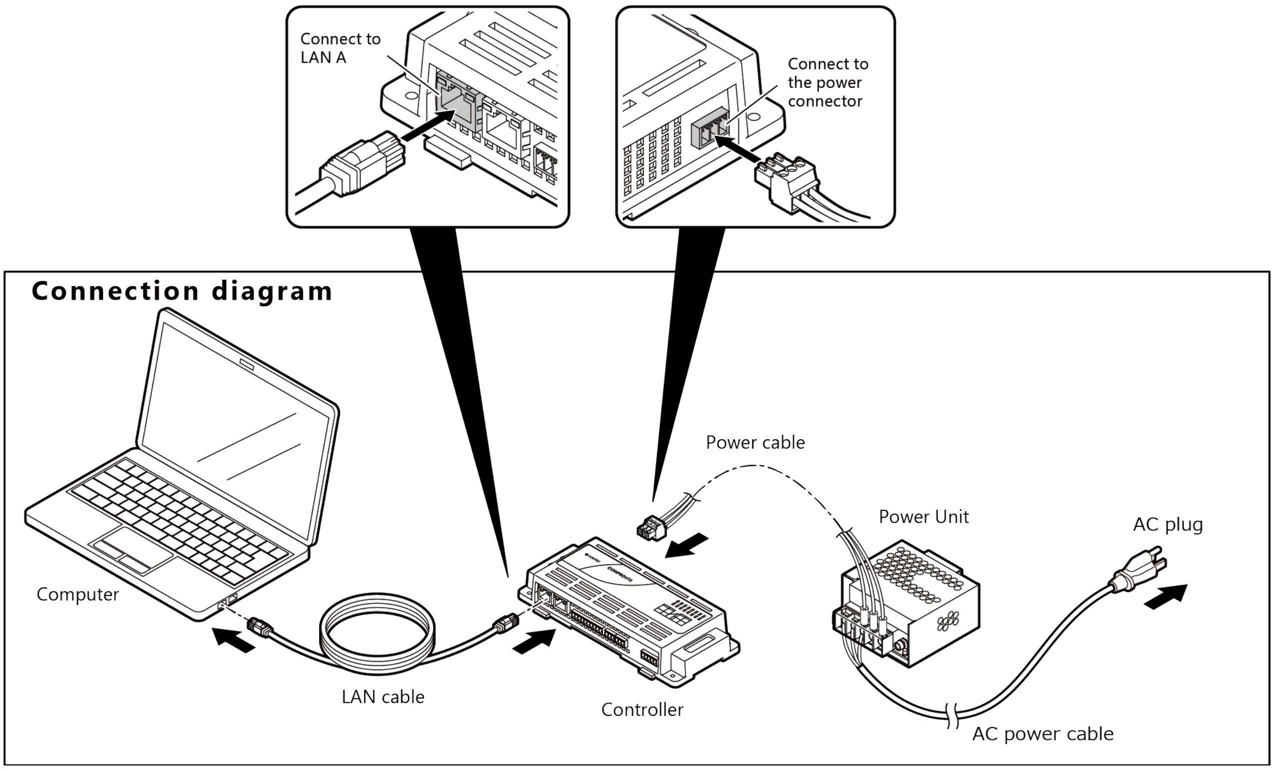
1.Connect with a PC

To set the product, you need to first set up the network between the PC and the product in order to establish communication.

First, connect the product with the PC.

1 Follow the instructions below to connect the computer, the controller, and the power unit.

* Refer to "**Reference Manual (Hardware)**" for how to create a power cable.



2 Connect the power unit with AC plug, then turn on the computer.

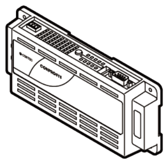
* After connecting the power unit with AC plug, it takes a few minutes for the controller to complete the start-up. (approx. 1-2 min)

2.Set the Computer Network

Follow the “Computer Network Setting Procedure” described below and set the network to make the IP addresses as shown.

Computer

IP address	10 . 1 . 1 . **200**
Subnet mask	255 . 0 . 0 . 0


Controller

IP address	10 . 1 . 1 . **101**
Subnet mask	255 . 0 . 0 . 0

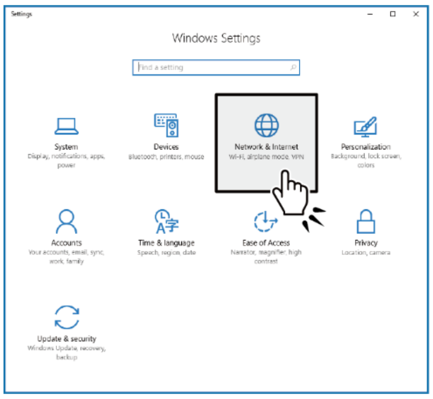
* The factory default setting

The product must set a unique IP address for the bold part (**200** or **101**) that is not used by other devices on your network.

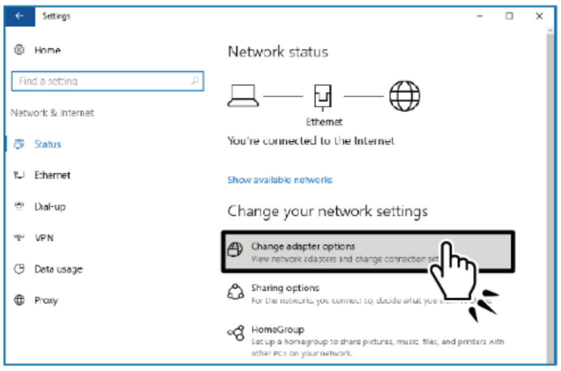
* If proxy is set to your PC, do not use the proxy.

1. Computer Network Setting Procedure (for Windows 10)

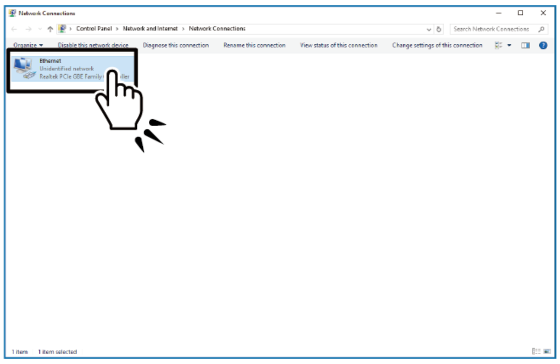
1 Click the [Network & Internet] on [Windows Settings] screen.


Settings
Windows Settings
Find a setting
System
Display, notifications, apps, power
Devices
Bluetooth, printers, mouse
Network & Internet
Wi-Fi, airplane mode, VPN
Personalization
Background, lock screen, colors
Accounts
Your accounts, email, sync, work, family
Time & language
Speech, region, date
Ease of Access
Narrator, magnifier, high contrast
Privacy
Location, camera
Update & security
Windows Update, recovery, backup

2 Click the [Change adapter options] in the [Network status].

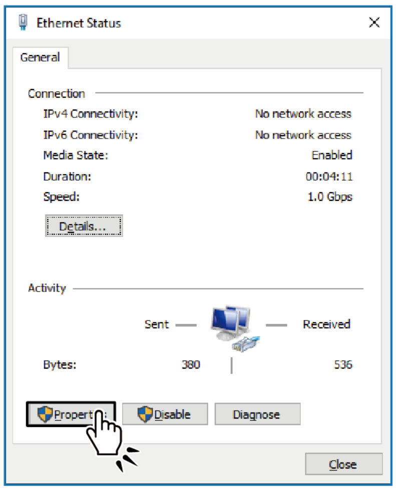

Settings
Home
Find a setting
Network & Internet
Status
Ethernet
Dial-up
VPN
Data usage
Proxy
Network status
Ethernet
You're connected to the Internet
Show available networks
Change your network settings
Change adapter options
View network adapters and change connection set
Sharing options
For the networks you connect to, decide what you
HomeGroup
Set up a homegroup to share pictures, music, files, and printers with other PCs on your network.

3 Double-click the appeared [Ethernet].


Network Connections
Control Panel › Network and Internet › Network Connections
Search Network Connections
Organize
Disable this network device
Diagnose this connection
Rename this connection
View status of this connection
Change settings of this connection
1 item
1 item selected

Ethernet
Unidentified network
Realtek PCIe GBE Family Controller

4 Click the [Property] in the [Ethernet Status] dialog box...



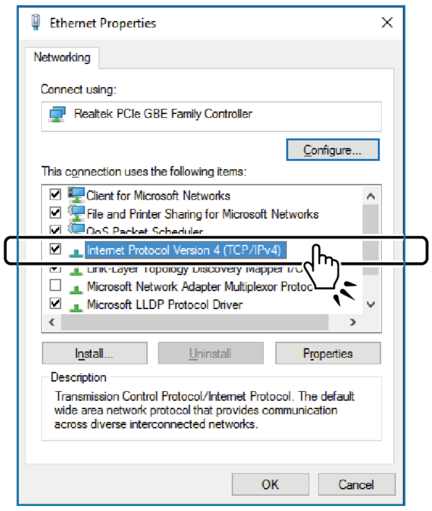
Connection

IPv4 Connectivity:	No network access
IPv6 Connectivity:	No network access
Media State:	Enabled
Duration:	00:04:11
Speed:	1.0 Gbps

Activity

	Sent	Received
Bytes:	380	536

5 Double-click the [Internet protocol version 4(TCP/IPv4)] in the [Ethernet Properties] dialog box.



Connect using:

Realtek PCIe GBE Family Controller

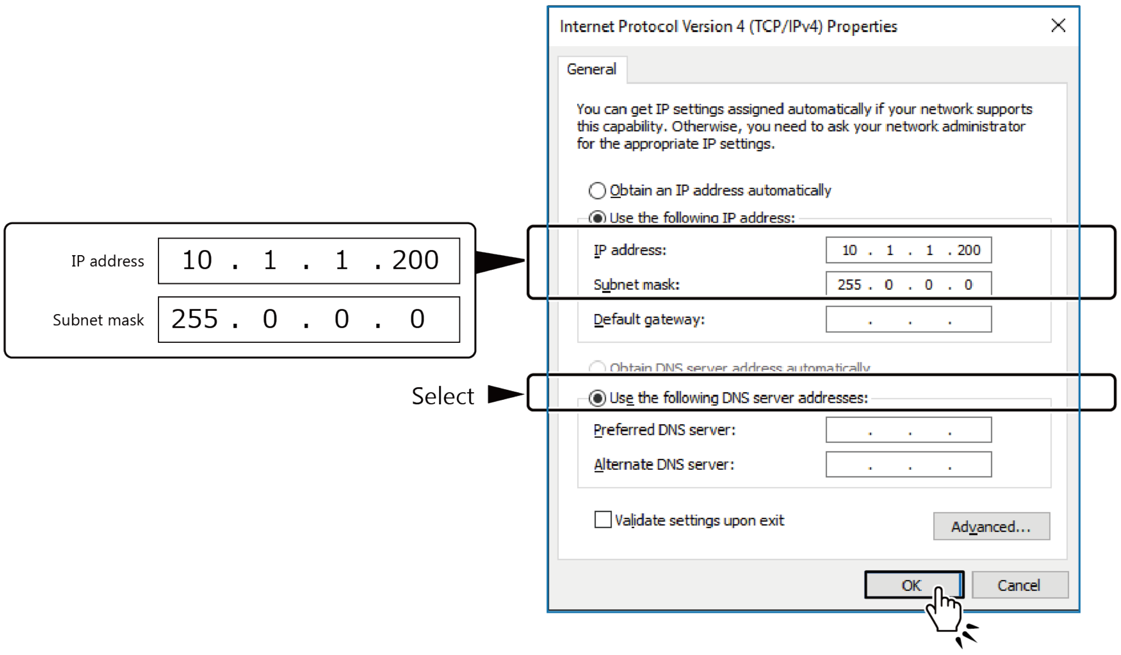
This connection uses the following items:

- [x] Client for Microsoft Networks
- [x] File and Printer Sharing for Microsoft Networks
- [x] QoS Packet Scheduler
- [x] Internet Protocol Version 4 (TCP/IPv4)
- [x] Link-Layer Topology Discovery Mapper I/O
- [ ] Microsoft Network Adapter Multiplexor Protoc
- [x] Microsoft LLDP Protocol Driver

Description

Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

6 In the [Internet protocol version 4 (TCP/IPv4) property], set IP address and Subnet mask as shown below.



IP address	10 . 1 . 1 . 200
Subnet mask	255 . 0 . 0 . 0

Select

Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically
● Use the following IP address:

IP address:	10 . 1 . 1 . 200
Subnet mask:	255 . 0 . 0 . 0
Default gateway:	. . .

○ Obtain DNS server address automatically
● Use the following DNS server addresses:

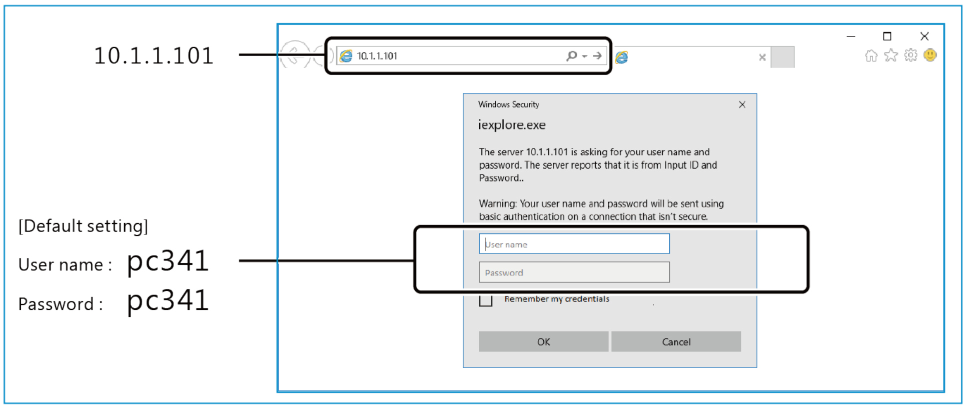
Preferred DNS server:	. . .
Alternate DNS server:	. . .

☐ Validate settings upon exit

7 Click the [OK] →the [OK] → the [Close] to close the dialog box and complete the network setting.

3.Check Communication

1 Start the Internet Explorer 11 on your computer. Enter IP address (10.1.1.101) of the controller in the address bar, then press [Enter] key.
The dialog box asking for the User name and Password appears, enter them and click the [OK].



[Default setting]

User name : pc341

Password : pc341

Windows Security

iexplore.exe

The server 10.1.1.101 is asking for your user name and password. The server reports that it is from Input ID and Password..

Warning: Your user name and password will be sent using basic authentication on a connection that isn't secure.

User name

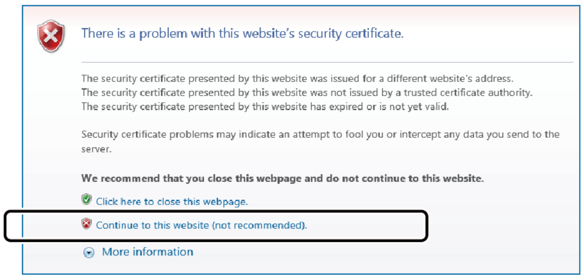
Password

Remember my credentials

* Refer to “**Compatible Web Browser (page 25)**” for details of compatible web browser.

* For an actual operation, change User name and Password in the [User name and Password] from the Maintenance menu.

* After entering IP address and pressing [Enter] key, the “Security certificate” might appear on the screen. Choose “Continue to this website”.



There is a problem with this website’s security certificate.

The security certificate presented by this website was issued for a different website's address.
The security certificate presented by this website was not issued by a trusted certificate authority.
The security certificate presented by this website has expired or is not yet valid.

Security certificate problems may indicate an attempt to fool you or intercept any data you send to the server.

We recommend that you close this webpage and do not continue to this website.

Click here to close this webpage.

Continue to this website (not recommended).

More information

2 If [Status menu] of Web browser menu appears, it indicates the success of the communication between the computer and the controller.

CONPROSYS WEB Setting

This chapter describes the product system and functions.

1.CONPROSYS WEB Setting Outline

The functions of the product can be set easily with the "CONPROSYS WEB Setting" through a browser.

1. Compatible Web Browser

CONPROSYS WEB Setting is compatible with the following browsers.

Compatible Web browser	Supported Version
Microsoft Internet Explorer	Ver. 11 or later
Google Chrome	Ver. 52 or late
Mozilla Firefox	Ver. 55 or late

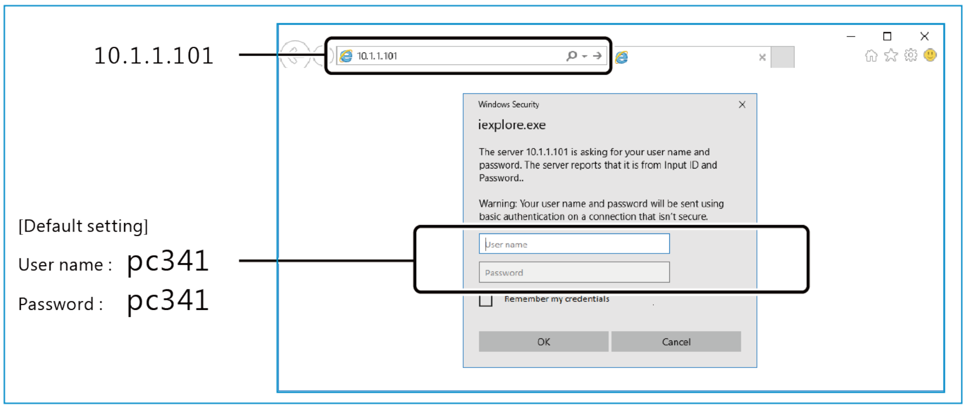
* Problems may arise due to the use of incompatible browser. Be sure to use the web browser that is compatible.

2. Start Up CONPROSYS WEB Setting

Start the Web browser on your computer that is connected with the controller. Enter IP address (10.1.1.101) of the controller in the address bar, then press the [Enter] key.

The dialog box asking for the User name and Password appears, enter them and click the [OK].

* Refer to "**Setup Manual**" for how to connect the controller with your computer.



The server 10.1.1.101 is asking for your user name and password. The server reports that it is from Input ID and Password..

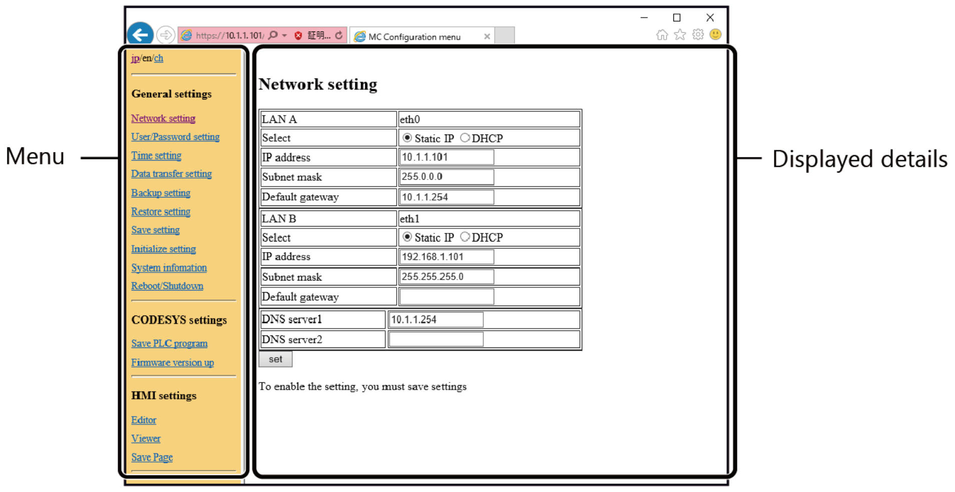
Warning: Your user name and password will be sent using basic authentication on a connection that isn't secure.

* After entering IP address and pressing [Enter] key, the "Security certificate" might appear on the screen. Choose "Continue to this website "then.

3. CONPROSYS WEB Setting Basic Operation

◆ CONPROSYS WEB Setting Page Structure

Click the menu item on the left side of the screen. This opens a page to set the details of the menu on the right side of the screen.



Network setting

LAN A	eth0
Select	[✓ Static IP] DHCP
IP address	10.1.1.101
Subnet mask	255.0.0.0
Default gateway	10.1.1.254
LAN B	eth1
Select	[✓ Static IP] DHCP
IP address	192.168.1.101
Subnet mask	255.255.255.0
Default gateway	
DNS server1	10.1.1.254
DNS server2	

To enable the setting, you must save settings

2.Menu Function List

Menu Functions are listed below.

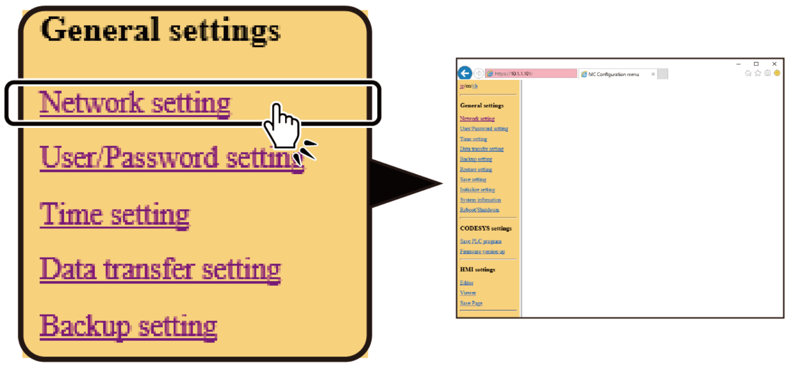
Menu item name	Function	Description
General setting		
Network setting	Set up the network such as "IP address".	Page 28
User/Password setting	Set up the wireless LAN setting such as "IP address".	Page 30
Time setting	Set up the name of NTP server that obtains the time and date.	Page 31
Data transfer setting	Set up the destination of the measured data to be transferred.	Page 32
Serial setting	Set up the communication parameter of serial port.	Page 33
Backup setting	Back up the monitoring screen, task program, and general settings.	Page 34
Restore setting	Restore a monitoring screen, task program, and general settings from a backup file.	Page 35
Save setting	Save general settings into ROM.	Page 36
Initialize setting	Restore all the settings to their factory defaults.	Page 37
System information	Display the details of the controller.	Page 37
Reboot/Shutdown	Reboot the controller or shut down the power of the controller.	Page 41
CODESYS setting		
Save PLC program	Save the PLC program into ROM area.	Page 41
Firmware version up	Update the firmware.	Page 42
HMI setting		
Editor	Display the monitoring edit page (CONPROSYS HMI).	Page 43
Viewer	Display the monitoring screen (CONPROSYS HMI)	Page 44
Save Page	Save the contents of the page created by Editor (CONPROSYS.HMI) into ROM.	Page 44

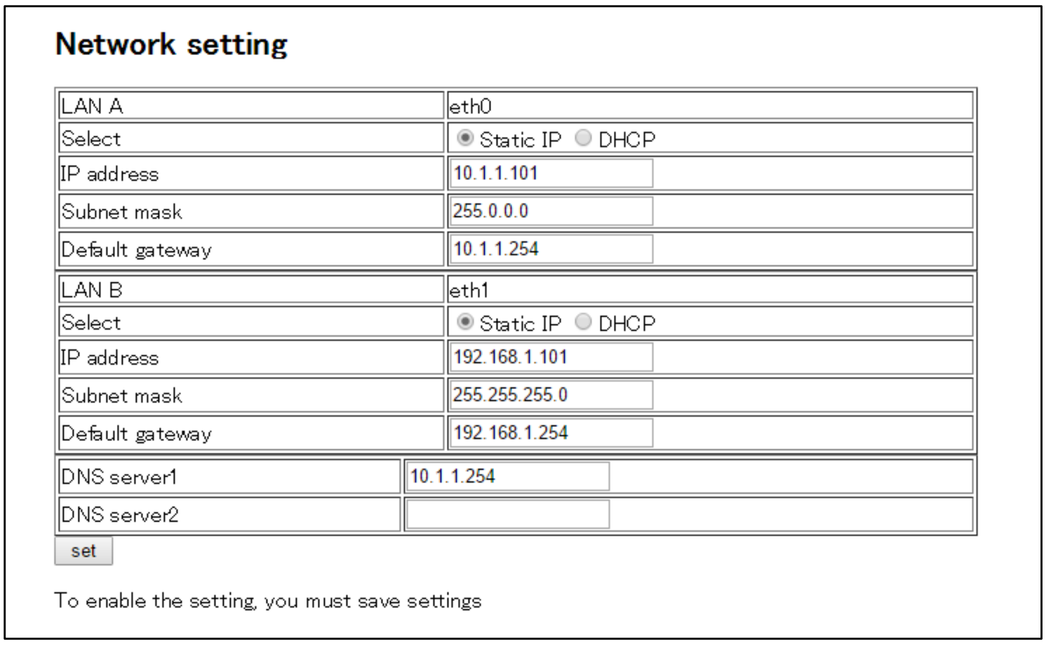
3. Function Details

1. Network setting

Set up IP address of the product and check network communications.

* Networks can be set for LAN A and LAN B respectively.





Network setting

LAN A	eth0
Select	[✓ Static IP] DHCP
IP address	10.1.1.101
Subnet mask	255.0.0.0
Default gateway	10.1.1.254
LAN B	eth1
Select	[✓ Static IP] DHCP
IP address	192.168.1.101
Subnet mask	255.255.255.0
Default gateway	192.168.1.254
DNS server1	10.1.1.254
DNS server2	

set

To enable the setting, you must save settings

◆ Select

Select how to set IP address.

[Static]: Set up IP address and Subnet mask and so on.

[DHCP]: Get IP address automatically with DHCP client.

[Default]: Static

◆ IP Address

Set up IP address.
It is enabled when [Static IP address] is selected.

[Default]: 10.1.1.101

◆ Subnet Mask

Set up Subnet mask.
It is enabled when [Static IP address] is selected.

[Default]: 255.0.0.0

◆ Default Gateway

Set up IP address of default gateway.
It is enabled when [Static IP address] is selected.
Leave it blank when this is not set.

[Default]: 10.1.1.254

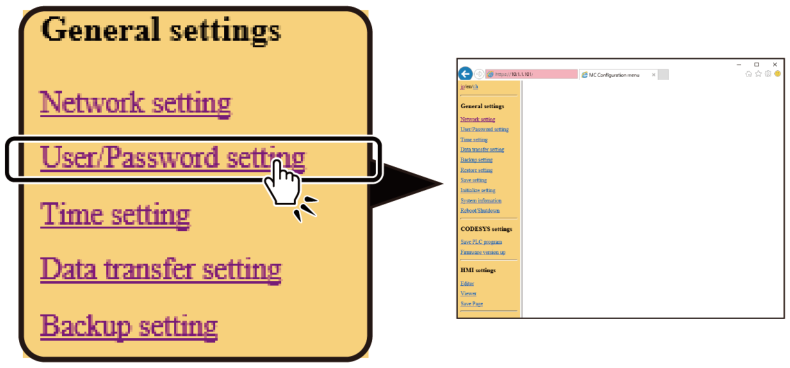
◆ DNS server1, DNS server2

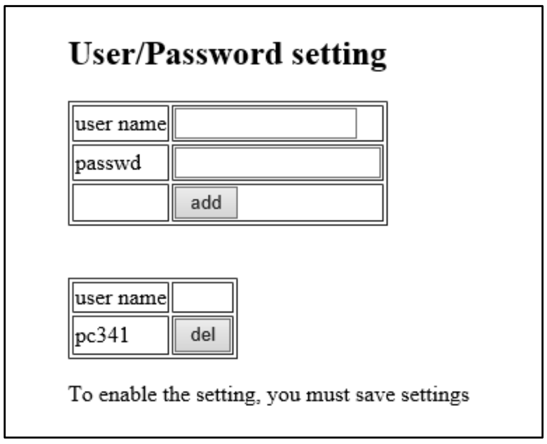
Set up IP address of DNS server.
It is enabled when [Static IP address] is selected.
Leave it blank when this is not set.

[Default]: DNS server1: 10.1.1.254
DNS server2: (Leave as it blank)

2. User/Password setting

Add or delete a user to log in CONPROSYS WEB Setting through a Web browser.
Change a password from the factory default setting and make your own password.





User/Password setting

user name	
passwd	
	add

user name	
pc341	del

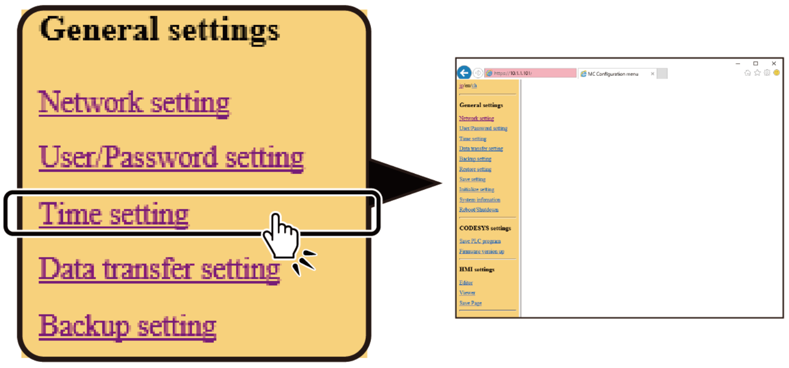
To enable the setting, you must save settings

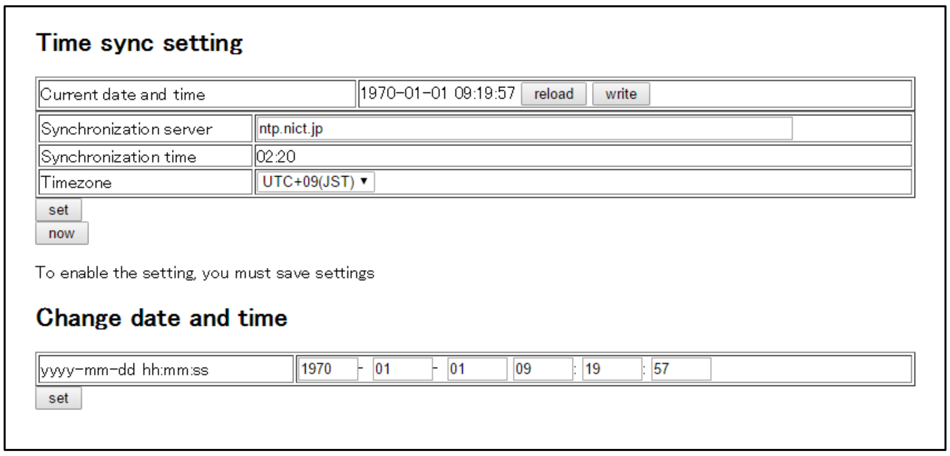
◆ User Name /Password

Add or delete a user to log in CONPROSYS WEB Setting through a Web browser.
Enter user name and password using 1 to 31 letters of one-byte alphanumeric character.
Changed settings become available after rebooting the product.
For stronger security, a different user from the default is added upon actual operation, and the default user is deleted.

3. Time setting

Set up the name of NTP server that obtains the time and date.





Time sync setting

Current date and time	1970-01-01 09:19:57 reload write
Synchronization server	ntp.nict.jp
Synchronization time	02:20
Timezone	UTC+09(JST)

set
now

To enable the setting, you must save settings

Change date and time

yyyy-mm-dd hh:mm:ss	1970 - 01 - 01 09 : 19 : 57

set

◆ Time sync setting

Current date and time

Display the current data and time.

By clicking [reload], time is updated from the synchronization server.

By clicking [write], the current time is written to ROM.

Synchronization server

Set the address of NTP server when using NTP.

[Setting]: FQDN or IP address

[Default]: ntp.nict.jp

Synchronization time

Synchronization time indicates that the product updates time from Synchronization server upon booting and at 02:20 each day. (It is not a setting item)

Time zone

Set a time difference between UTC (Universal Time, Coordinated) and the time of the local region where the product is used.

[Default]: UTC+09(JST-9)

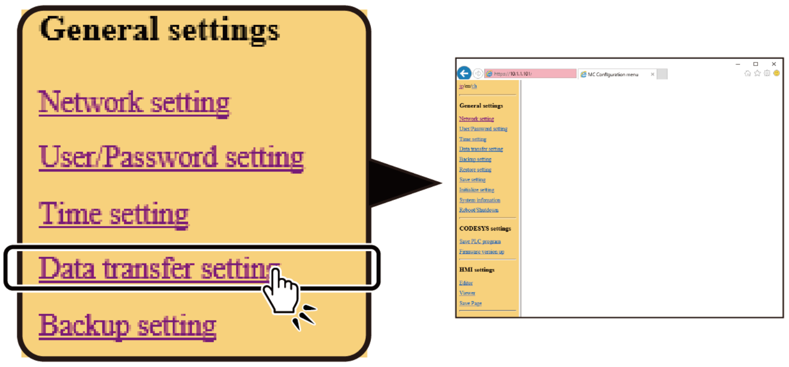
◆ Change data and time

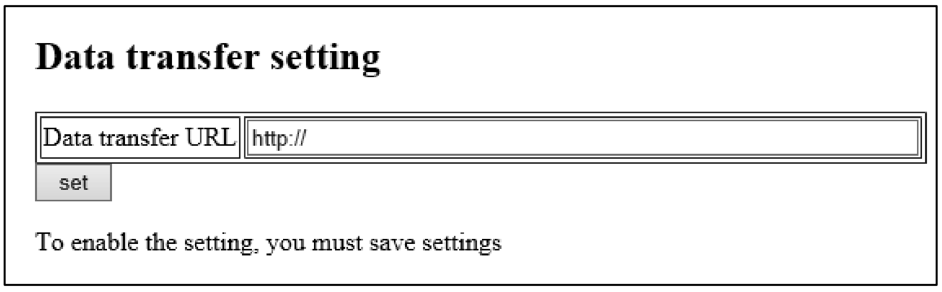
When setting date and time manually, enter date and time in the field, then update.

[Setting]: 1970-01-01 00:00:00 - 2038-1-19-03:14:07
[Default]: 1970-01-01 00:00:00

4. Data transfer setting

Set up the destination of the measured data to be transferred.
Refer to "**Data Transfer Format (P181)**" for the details.





Data transfer setting

Data transfer URL	http://

set

To enable the setting, you must save settings

◆ Data transfer URL

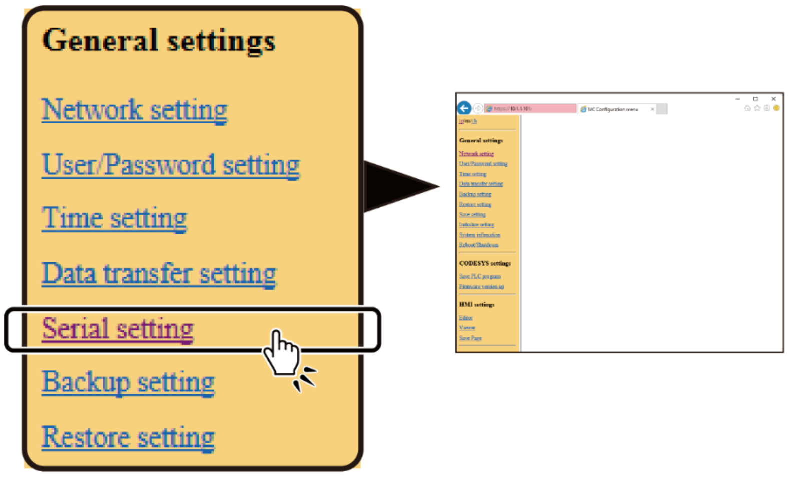
Enter URL of the server to send measured data.

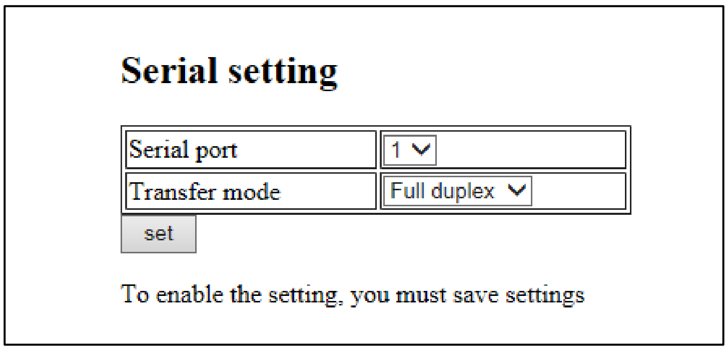
[Default]: No settings

5. Serial setting

Set up communication parameter of serial port.

First, select the port number you wish to set for the serial port, then set communication parameter.





Serial setting

Serial port	1
Transfer mode	Full duplex

set

To enable the setting, you must save settings

◆ Serial port

The list of serial port numbers set for communication parameter is displayed.

Here, select the serial communication number you wish to set.

Serial number differs depending on the product model.　Refer to "**Serial Port (page 69)**" for details.

◆ Transfer mode

Set the communication mode of data transfer.

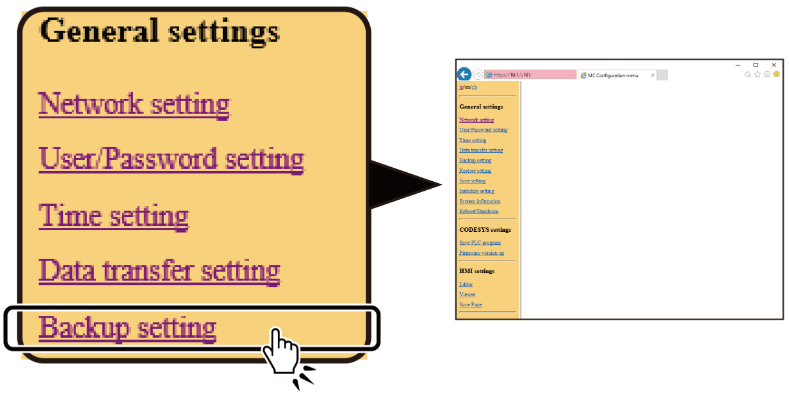
The mode should be the same as that of the hardware settings.

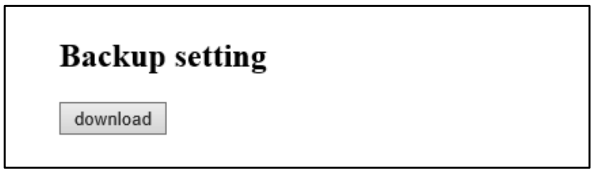
[Setting]: Full duplex, Half duplex

[Default]: Full duplex

6. Backup setting

Create a backup file of a monitoring screen, a task program, and general settings.



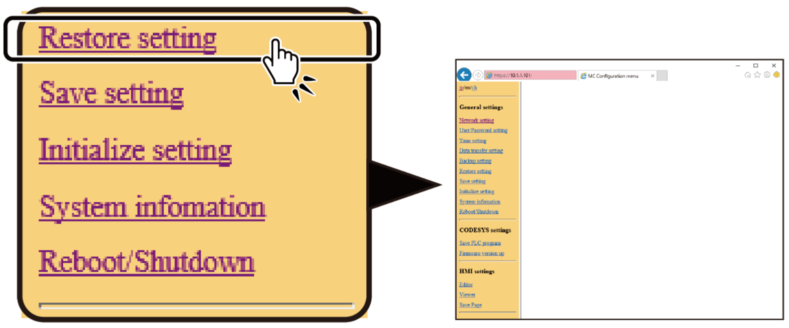


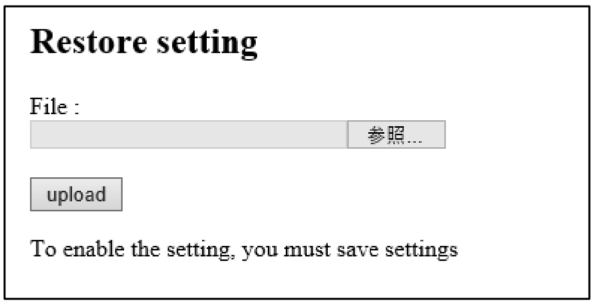
Click the [download] to save backup of general settings.

Default file name is “config dat”.　Save the file with a new name.

7. Restore setting

Restore a monitoring screen, a task program, and general settings from a backup file.





To enable the setting, you must save settings

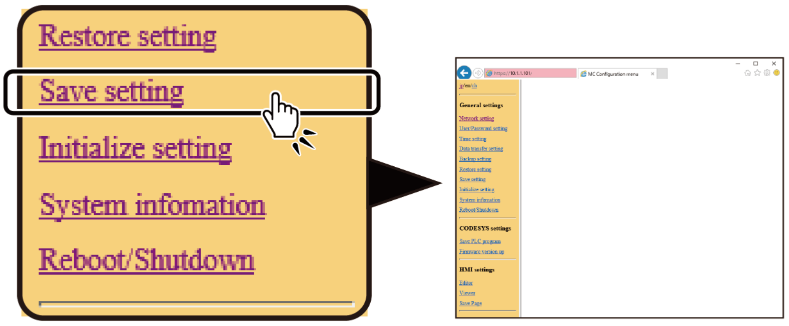
Choose the backup file you created in the backup setting through the [Browse], then click the [upload].

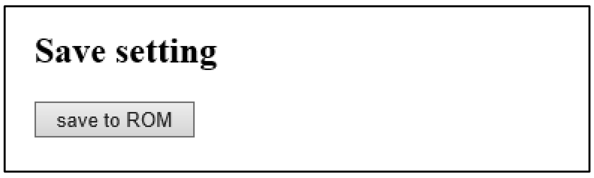
Although you can check the restoring settings on the setting screen respectively, it is required to perform the [Save setting] and the [Reboot] to enable the settings.

8. Save setting

Save the contents set in CONPROSYS WEB Setting to ROM.

Click the [save to ROM]. PWR LED starts flashing and saving begins.





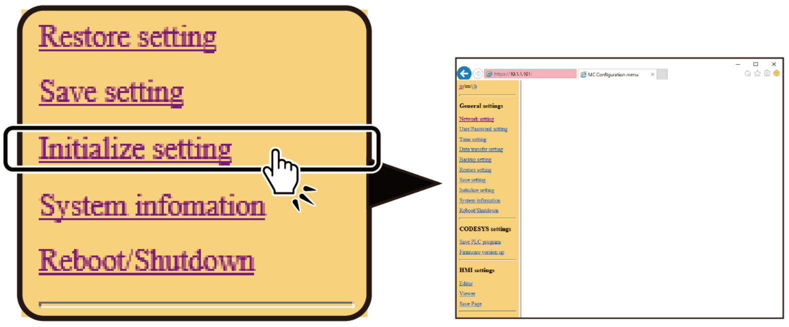
⚠ CAUTION

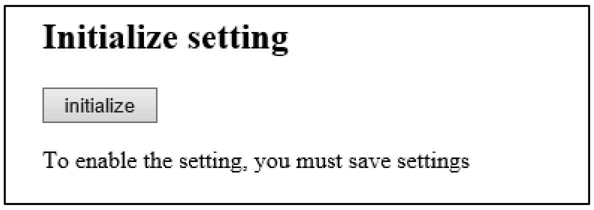
Do not turn off the power until PWR LED flashing has stopped.　(approx.: five seconds)

Without saving, the contents return to those before setting at rebooting or shutting down.

9. Initialize setting

Initialize all of the settings to their factory defaults.





Initialize setting

initialize

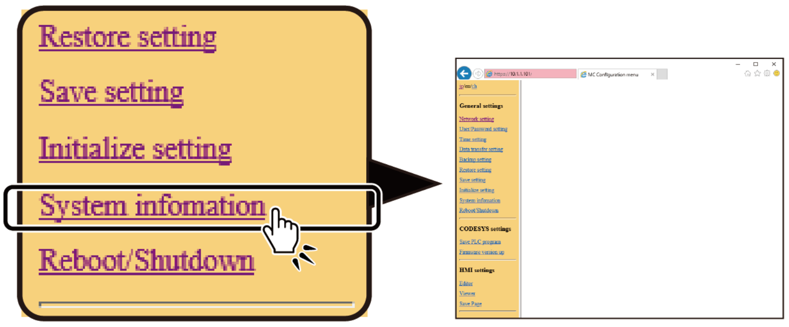
To enable the setting, you must save settings

Click the [initialize] to initialize the settings.

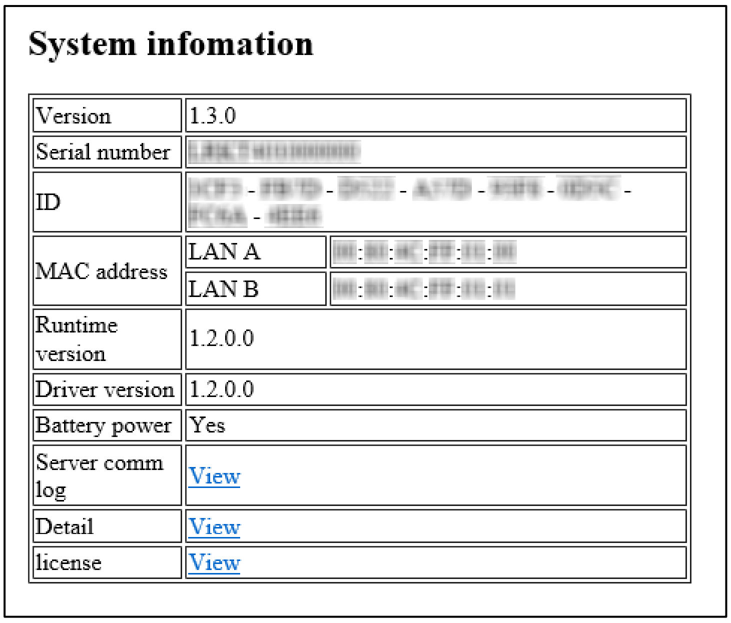
Perform the [Save setting] and the [Reboot] to enable the initialized settings.

10. System information

Display the details of the system information of the product.



The screen below is shown.



System infomation

Version	1.3.0	
Serial number	[redacted]	
ID	[redacted]	
MAC address	LAN A	[redacted]
	LAN B	[redacted]
Runtime version	1.2.0.0	
Driver version	1.2.0.0	
Battery power	Yes	
Server comm log	View	
Detail	View	
license	View	

◆ Version

Display the firmware version.

◆ Serial number

Display the serial number.

◆ ID

Display the required ID to register in CDS or CDS2 of cloud server.

◆ MAC address

Display MAC address of wired LAN.

◆ Run time version

Display the run-time version.

◆ Driver version

Display the driver version.

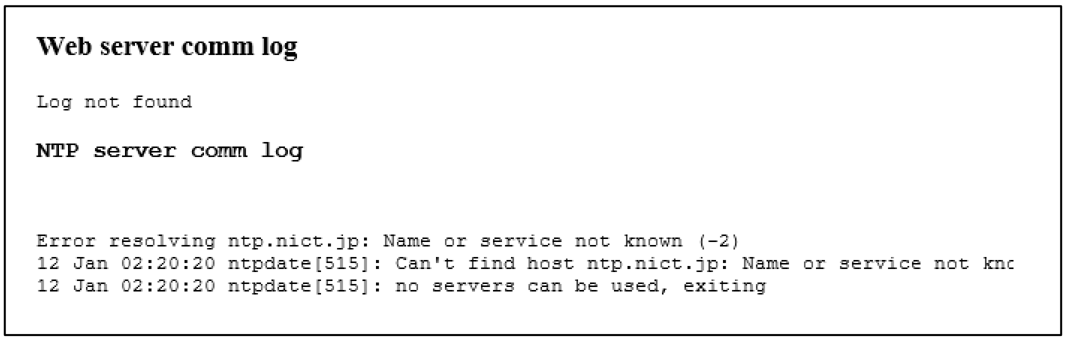
◆ Battery Residual Capacity

Display whether the battery runs out.

◆ Web server comm log

Server communication log displays the communication log from the server.

Item	Description
Web server comm log	Show the latest communication log from the server specified as the data transfer URL.
NTP server communication log	Show the latest communication log from the specified SMTP server.



Web server comm log

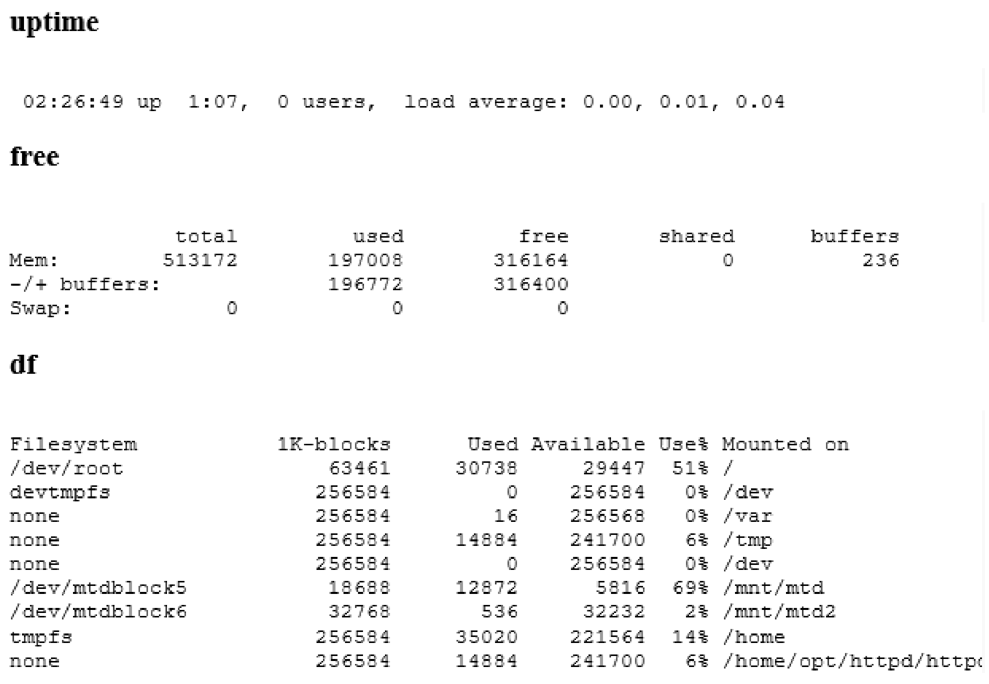
```
Log not found
```

NTP server comm log

```
Error resolving ntp.nict.jp: Name or service not known (-2)
12 Jan 02:20:20 ntpdate[515]: Can't find host ntp.nict.jp: Name or service not knc
12 Jan 02:20:20 ntpdate[515]: no servers can be used, exiting
```

◆ Detail

The followings are the details of system information.



uptime

```
 02:26:49 up  1:07,  0 users,  load average: 0.00, 0.01, 0.04
```

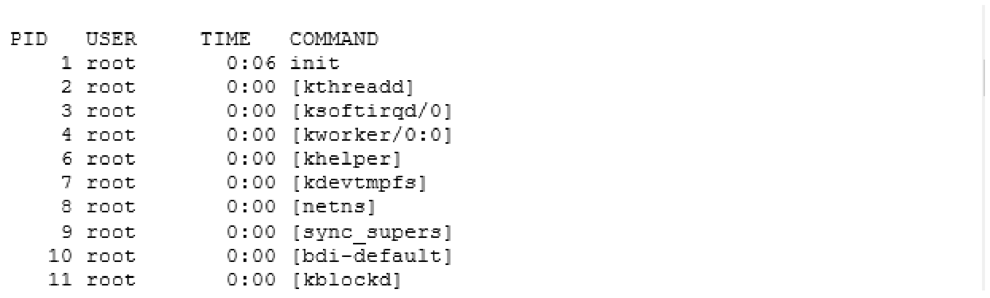
free

```
             total         used         free       shared      buffers
Mem:        513172       197008       316164            0          236
-/+ buffers:             196772       316400
Swap:            0            0            0
```

df

```
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/root                63461     30738     29447  51% /
devtmpfs                256584         0    256584   0% /dev
none                    256584        16    256568   0% /var
none                    256584     14884    241700   6% /tmp
none                    256584         0    256584   0% /dev
/dev/mtdblock5           18688     12872      5816  69% /mnt/mtd
/dev/mtdblock6           32768       536     32232   2% /mnt/mtd2
tmpfs                   256584     35020    221564  14% /home
none                    256584     14884    241700   6% /home/opt/httpd/httpc
```

ps aux



```
PID   USER      TIME   COMMAND
    1 root       0:06 init
    2 root       0:00 [kthreadd]
    3 root       0:00 [ksoftirqd/0]
    4 root       0:00 [kworker/0:0]
    6 root       0:00 [khelper]
    7 root       0:00 [kdevtmpfs]
    8 root       0:00 [netns]
    9 root       0:00 [sync_supers]
   10 root       0:00 [bdi-default]
   11 root       0:00 [kblockd]
```

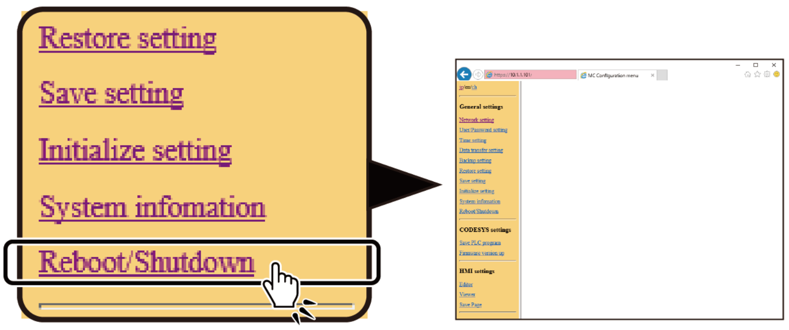
◆ License

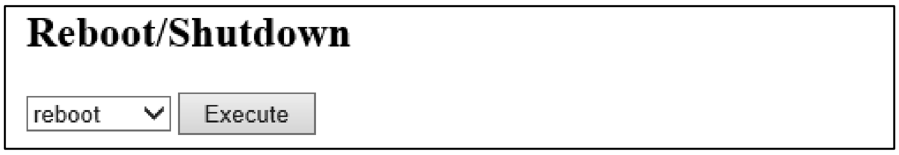
Click the [View] to display the license information of the software.

The approval is required to use the software.

11. Reboot/Shutdown

This reboots or shut downs the product.





Select “reboot” or “shutdown” and then click the [start].

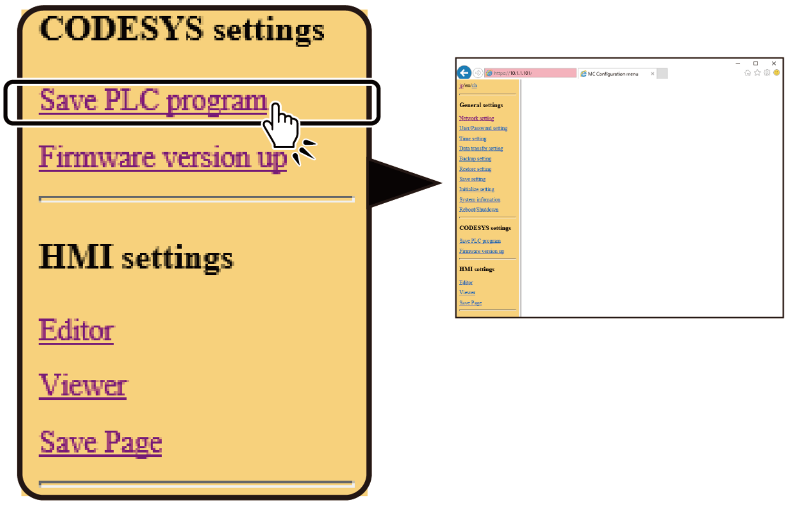
LED flashing indicates rebooting or shutting down in process.

12. Save PLC program

Save PLC program to ROM.

Click the [Write] and LEDs of ST1 and ST2 start flashing.

Flashing stops when the saving is completed.



⚠CAUTION

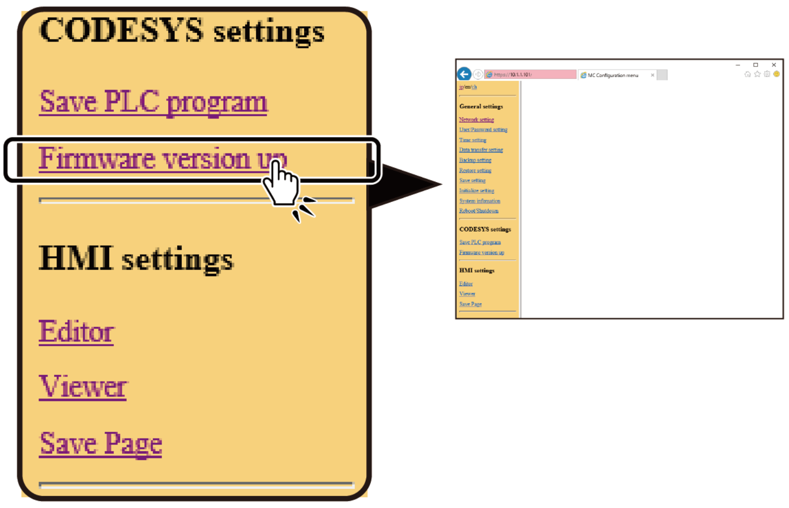
ST1 and ST2 LEDs of the product continue to flash while saving the settings.

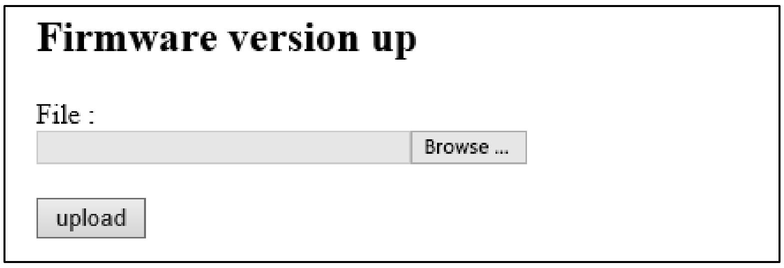
Do not turn off the power until flashing has stopped. (approx.: five seconds)

Without saving, the contents return to those before setting at rebooting or shutting down.

13. Firmware version up

Update the firmware with “version up” file.





Firmware version- up file can be downloaded from the CONTEC website.

Download https://www.contec.com/download/

From the [Browse] button, specify the firmware and click the [Upload] button.

* The file is compressed by ZIP format. Decompress it and use the bin file extension.

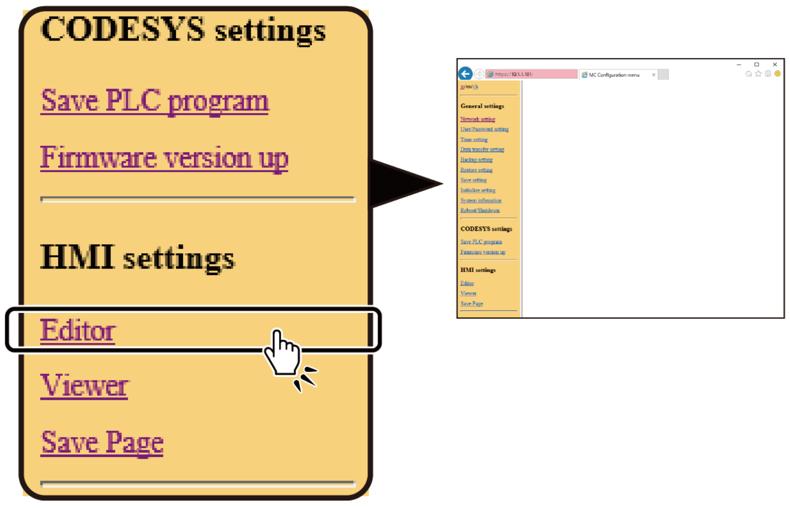
⚠ CAUTION

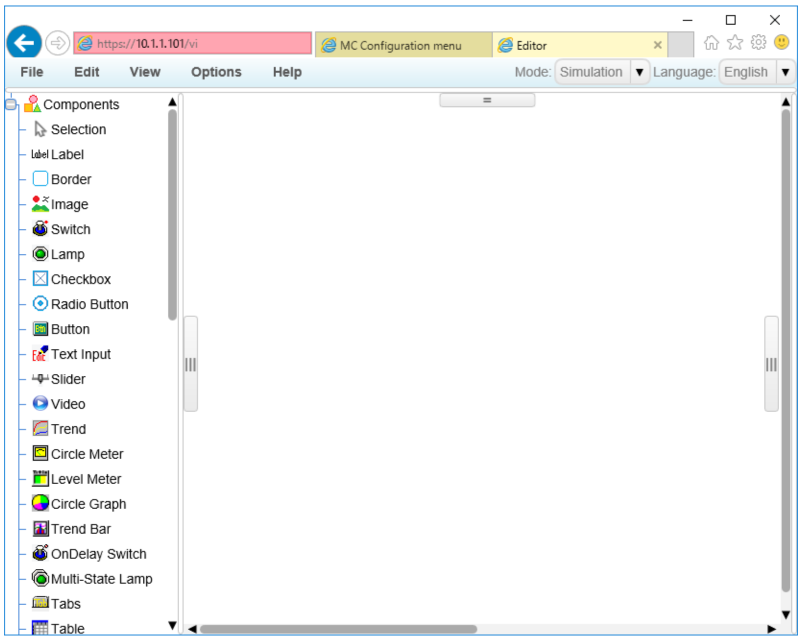
ST1 and ST2 LEDs of the product keep flashing while upgrading.
Do not turn off the power while LEDs are flashing. Otherwise, data get damaged and it disables the product to startup.

14. Editor

Display the page to create or change the monitoring screen.

Create your own monitoring screen by placing control items on the screen.



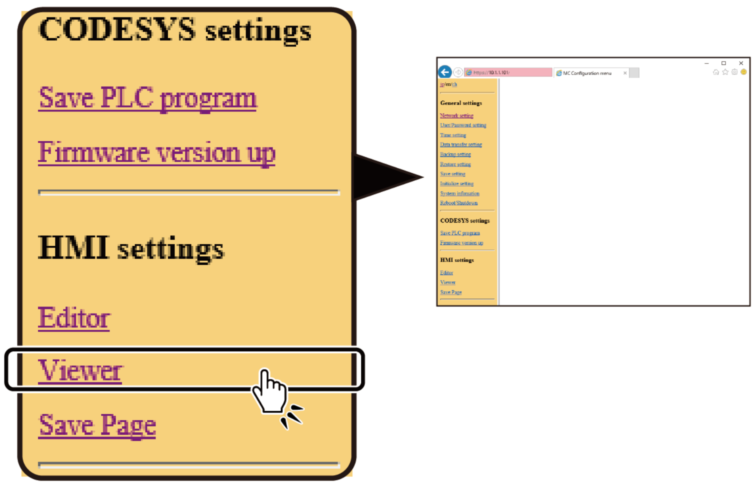


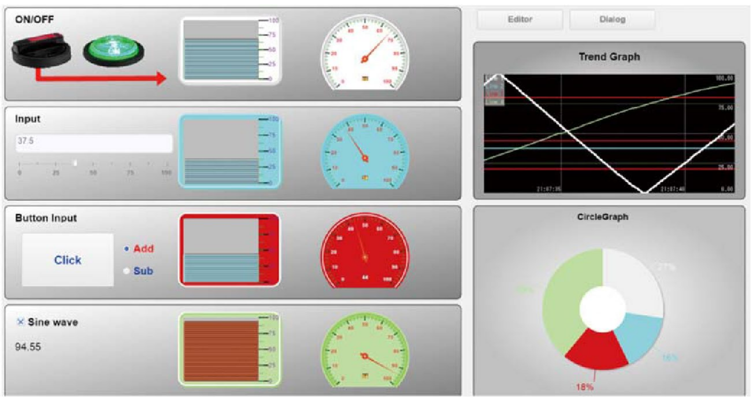
* Refer to "**Monitoring Edit (page 156)**" for details.

15. Viewer

View the monitoring screen.

Input signal status can be viewed on the monitoring screen.

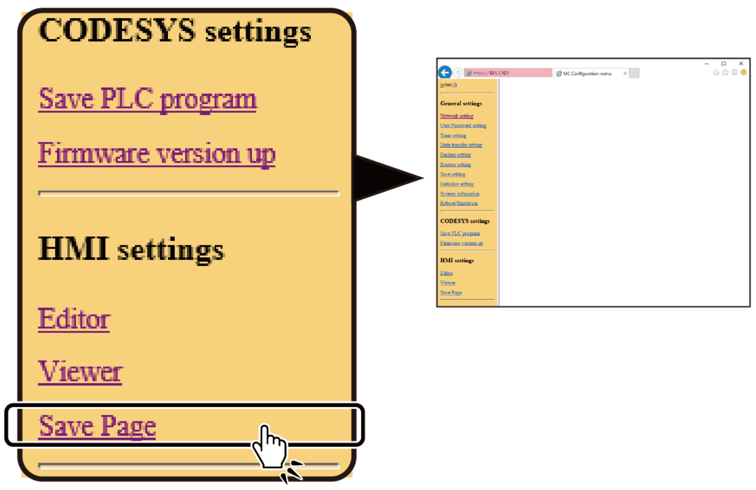




* Refer to “**Monitoring Edit (page 156)** ” for details.

16. Save Page

Save the page created in Editor to ROM.



CODESYS Installation

This chapter describes the CODESYS installation procedure.

1.Installation of the CODESYS

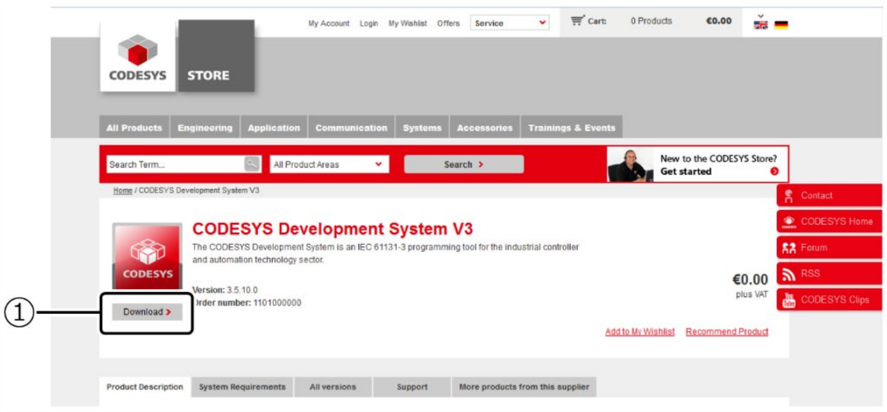
With PAC series, you can utilize the “CODESYS”, a device-independent system that is compliant with the IEC 61131-3 standard, for PLC or HMI development.

To use the CODESYS, installation of the CODESYS development environment and “CODESYS Package” for CONPROSYS are required.

1 How to obtain the CODESYS development environment
CODESYS development environment can be downloaded from the CODESYS website.

2 Through Web browser, access to the CODESYS Store.
Click (1) the [Download] button.
The CODESYS is consistently upgraded. Make certain you choose the latest version.

Download http://store.codesys.com/codesys.html



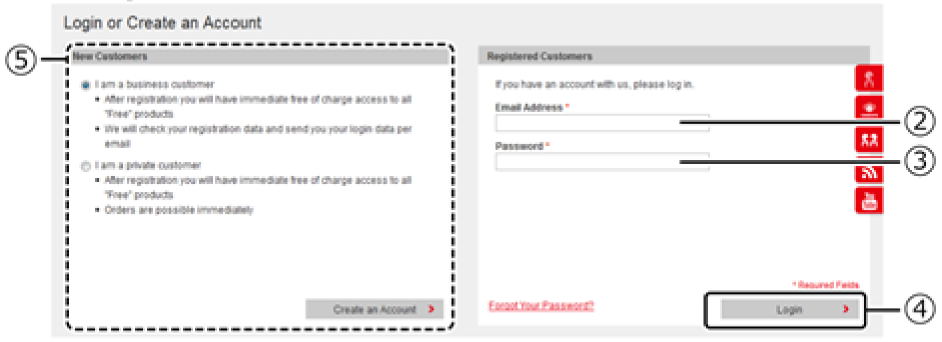
CODESYS Development System V3

The CODESYS Development System is an IEC 61131-3 programming tool for the industrial controller and automation technology sector.

Version: 3.5.10.0
Order number: 1101000000

3 “Login or Create an Account” page appears after clicking the [Download] button.
If you are a registered customer, enter your email address in (2) “Email Address”, and password in (3) “Password”, then click (4) the [Login] button.
If you are a new user, register through (5) “New Customers” to log in.

▼ "Login or Create an Account" screen



- I am a business customer
  - After registration you will have immediate free of charge access to all "Free" products
  - We will check your registration data and send you your login data per email
- I am a private customer
  - After registration you will have immediate free of charge access to all "Free" products
  - Orders are possible immediately

If you have an account with us, please log in.

4 In the “Registered Customers”, enter “Email Address” and “Password” to log in.
After log-in, click the “Download” to start downloading.

5 Decompress the downloaded file, and execute the setup file (extention.exe file) to install CODESYS.

2.Installation of the Package

To use this product with the CODESYS, installation of CODESYS Package for CONPROSYS is necessary.

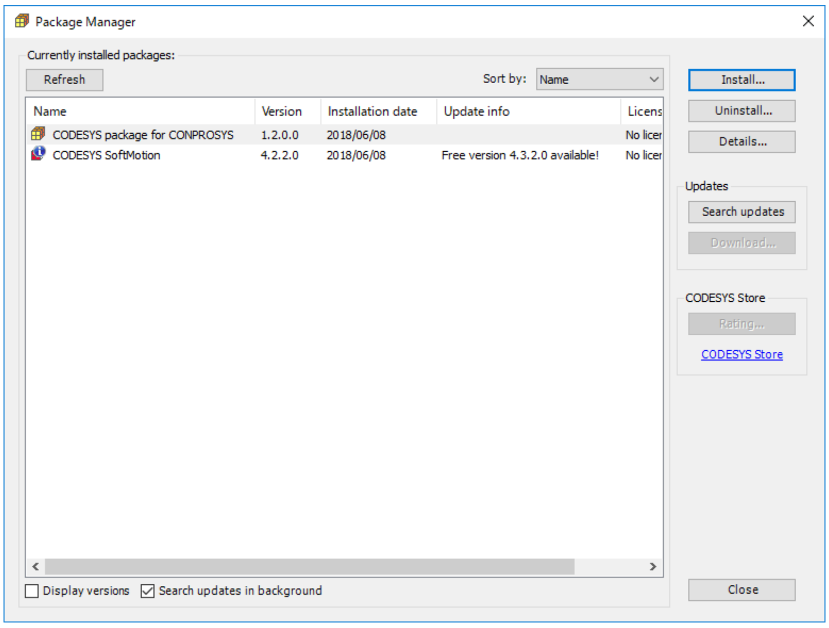
1 Go to the CONTEC website. From the menu Downloads, select [Drivers & Software].
Enter the product name or specify the category (PAC system) and click the [Search].
Select and click the product driver.

Download https://www.contec.com/

2 Download the CONPROSYS Package.

3 Run the CODESYS and select [Tools] - [Package Manager...] from CODESYS menu.

4 Click the [Install...], decompress the downloaded file, and specify the package file.
File name: CONTEC CONPROSYS PAC Series package

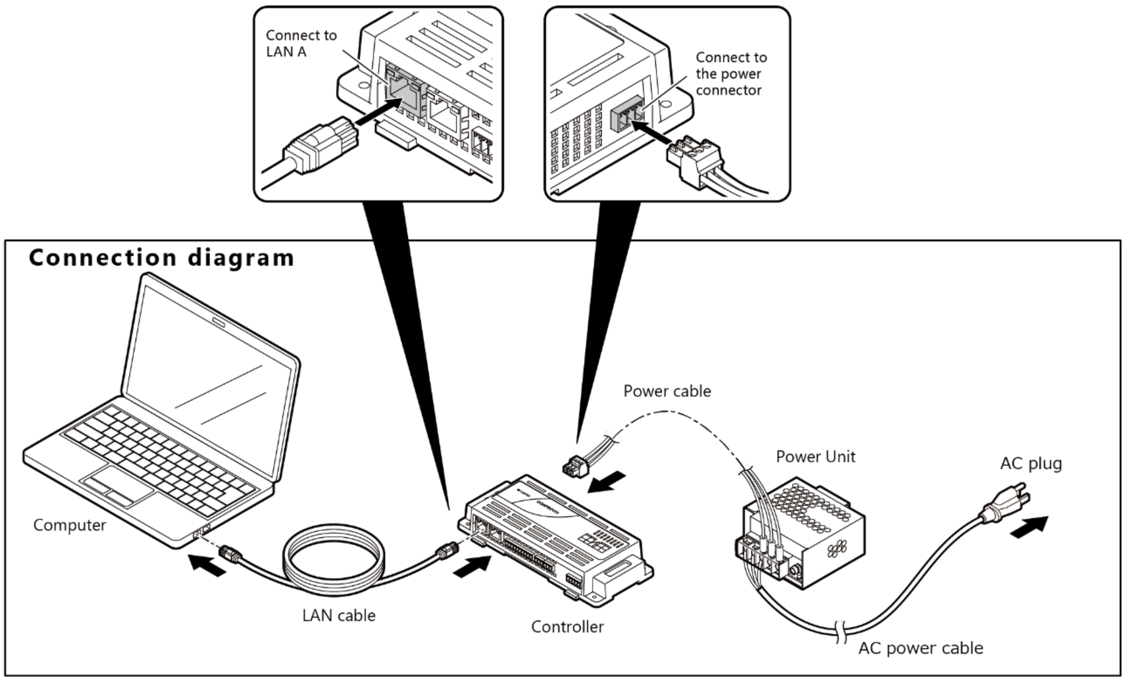


Name	Version	Installation date	Update info	Licens
CODESYS package for CONPROSYS	1.2.0.0	2018/06/08		No licer
CODESYS SoftMotion	4.2.2.0	2018/06/08	Free version 4.3.2.0 available!	No licer

3.PC and CONPROSYS cable connection

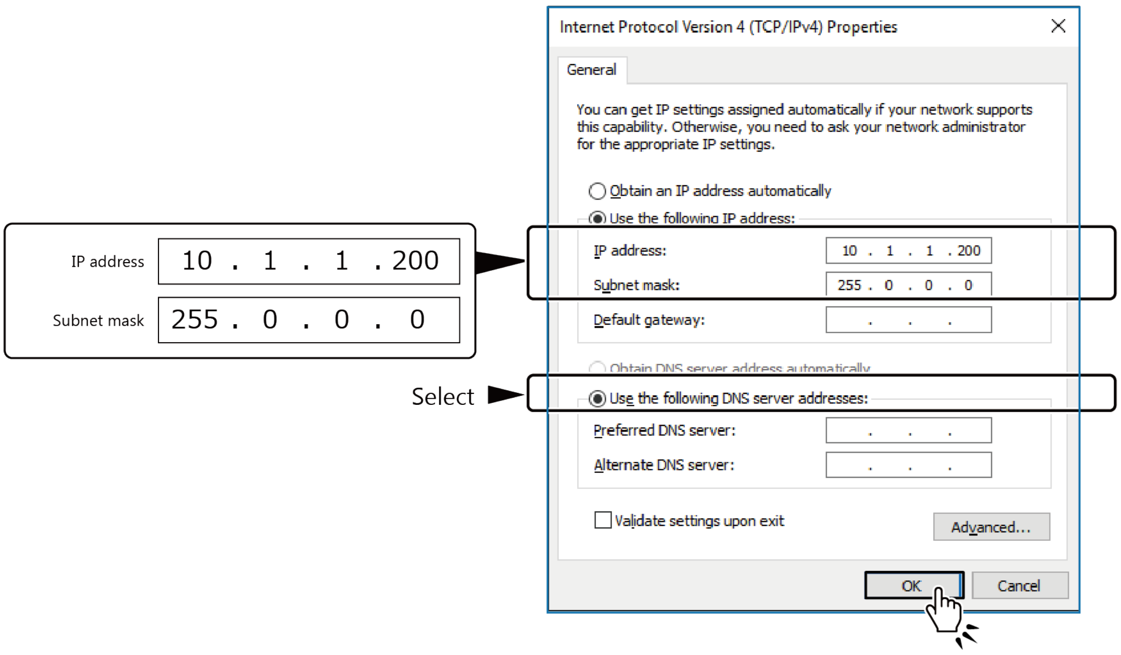
1 Connect the LAN port of CODESYS development PC and the LAN A port of this controller with Ethernet cable.

* Use LAN A for communication between the controller and PC, and LAN B for EtherCAT communication and Modbus TCP Slave connection.



2 Power on the controller.

3 Change network setting for PC.
LAN A port of this product is set to "10.1.1.101" as a factory default, and Subnet mask is "255.0.0.0" as a factory default.
When IP address of LAN A is [10.1.1.101], therefore, set the IP address for your PC to "10.1.1.102", for example.　Set the subnet mask for your PC to "255.0.0.0", for example.


Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

- Obtain an IP address automatically
- [✓ Use the following IP address:]

IP address:	10 . 1 . 1 . 200
Subnet mask:	255 . 0 . 0 . 0
Default gateway:	. . .

- Obtain DNS server address automatically
- [✓ Use the following DNS server addresses:]

Preferred DNS server:	. . .
Alternate DNS server:	. . .

Validate settings upon exit

* The PLC controller must set a unique IP address that is not used by other devices on your network. IP address of PLC controller can be set through web browser.

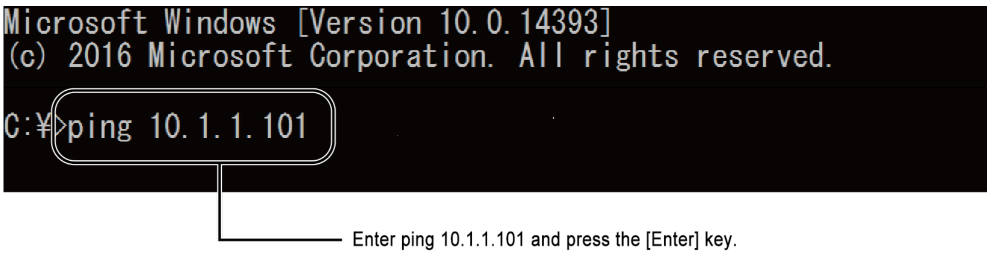
* Refer to "**Set the Computer Network (page 18)**" for detailed settings of ID address.

4 Use the ping command and confirm that PLC controller is connected with the PC.
Open Command prompt and execute ping command for IP address 10.1.1.101.

Format

```
ping IP address of the product
```

The server unit should respond if it is operating.



```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:¥>ping 10.1.1.101
```

Enter ping 10.1.1.101, and press the [Enter] key on your keyboard.

4. Firmware version up

Updates the firmware related to CODESYS.

1 Go to the CONTEC website. From the menu Downloads, select [Drivers & Software].
Enter the product name or specify the category (PAC system) and click the [Search].
Select and click the product driver.

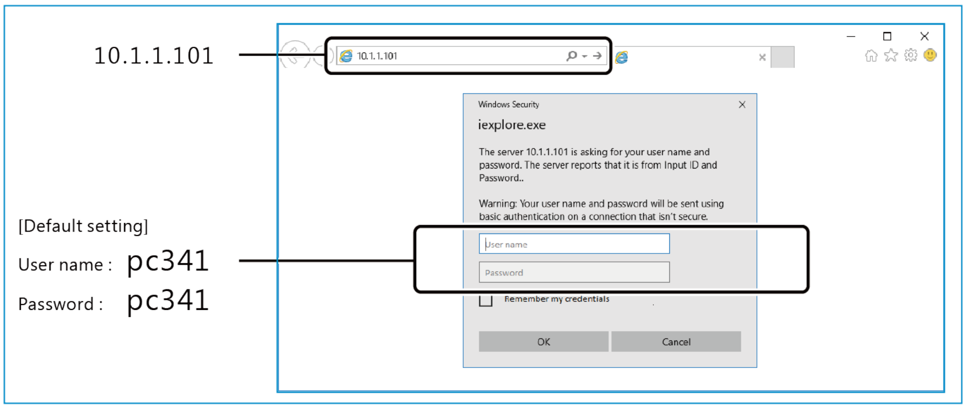
Download https://www.contec.com/

2 Download firmware.
(The downloaded file is compressed by ZIP format. Decompress it and use the bin file extension.)

3 Connect a host PC and this product as described in "**PC and CONPROSYS cable connection (page 49)**".

4 Start web browser and enter "http://10.1.1.101/" (PLC controller IP address) in the address field.
*It is also accessible through "https://10.1.1.101/". If the [Certificate Error] message appears, choose the "Continue to this website".

5 Enter user name "pc341" and password "pc341" to log in.
Firmware version can be checked in "System information" of "General setting".



Windows Security

iexplore.exe

The server 10.1.1.101 is asking for your user name and password. The server reports that it is from Input ID and Password..

Warning: Your user name and password will be sent using basic authentication on a connection that isn't secure.

☐ Remember my credentials

6 From [CODESYS setting], select the [Firmware version up] and click the [Browse] to open the decompressed the firmware.

7 Click the [upload] button and firmware starts upgrading. While upgrading the firmware, ST1 and ST2 LEDs continue to flash.

⚠ CAUTION

ST1 and ST2 LEDs of the product keep flashing while upgrading.
Do not turn off the power while LEDs are flashing. Otherwise, data get damaged and it disables the product to startup.

8 Rebooting starts automatically upon the completion of upgrading. Check whether the version has been promoted in the [system information] menu.

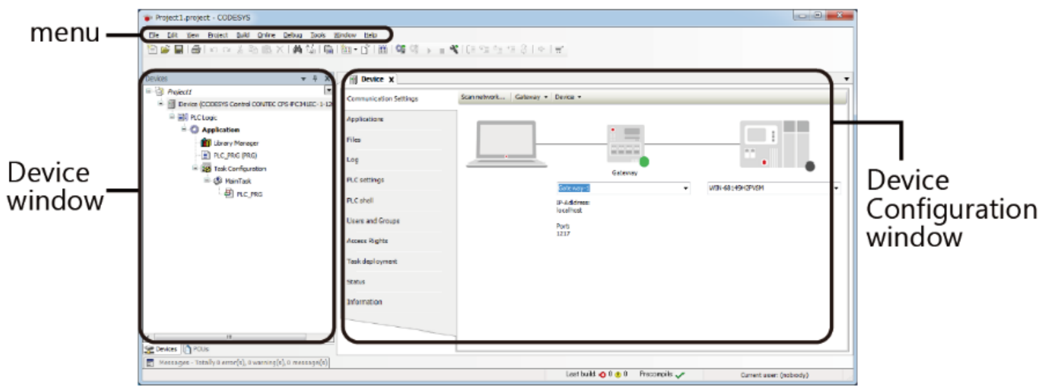
Basic Programming

This chapter describes basic procedure to operate this product with CODESYS.

1.Nomenclature of CODESYS Components

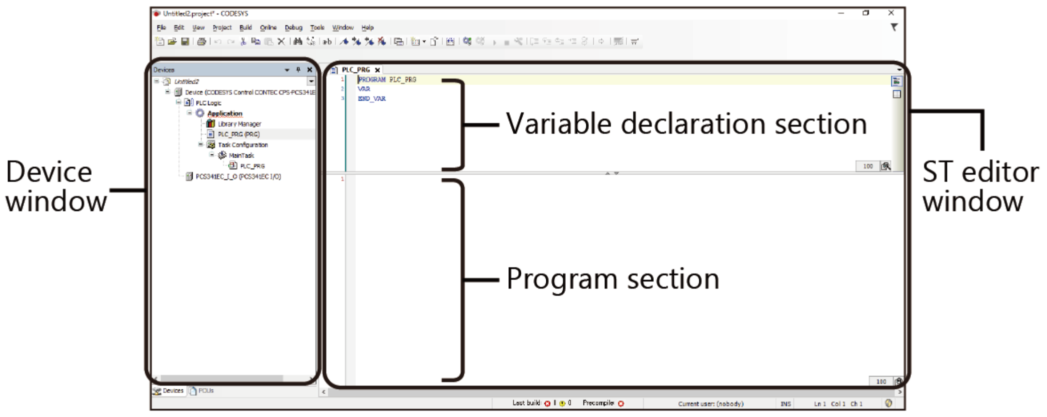
The basic screen display of CODESYS is viewed as below.

1. Device window, Device Configuration window



* Double-click the “Device (CODESYS Control CONTEC CPS-PCXXXXXXXXXX)” icon on Device window to open Device Configuration window.

2. ST Editor window



* Double-click the “PLC_PRG (PRG)” icon on Device window to open ST Editor window.

2.Format and programming of ST language

ST language is used for an example programming in this manual.

Formats of basic ST language such as assignment expression, conditional expression, and commenting out are listed.

◆ Comment

Add a comment in the program.

```
/*     Comment */
(*   Comment *)
//     Comment
```

- To comment out several lines, separate them with [Conditional expression] or [(*   *)].
- To comment out one line, add [//] at the beginning.

◆ Assignment expression

Assign a value to variable.

```
DO0 := 1;
```

- [Variable]:= [Value]; indicates assigning a value to variable.
- Add [;] at the end of line.

◆ Data type declaration

Declare data type of variable.

```
StartFlag : BOOL := TRUE;
Error : BOOL;
TestString : STRING := 'Test String!';
```

- First line: An assigning value to the variable [StartFlag] is Boolean (True/False), then assign true.
- Second line: An assigning value to the variable [Error] is Boolean.
- Third line: An assigning value to the variable [TestString] is a string, then assign [TestString!].

◆ Conditional expression (IF - THEN)

Determine a condition, then execute or skip routine.

```
IF StartFlag THEN
(Routine)
END_IF
```

If a value of variable [StartFlag] is [True], then execute the routine placed between END_IF.

◆ Conditional expression (CASE - OF)

Determine a condition, then execute or skip routine.

```
CASE State OF
0:
(Routine 1)
1:
(Routine 2)
END_CASE
```

- If the value of variable [State] is [0], execute routine 1.
- If the value of variable [State] is [1], execute routine 2.

* For details of ST language, please refer to the following site “PLCopen”.

PLCopen https://www.plcopen.org

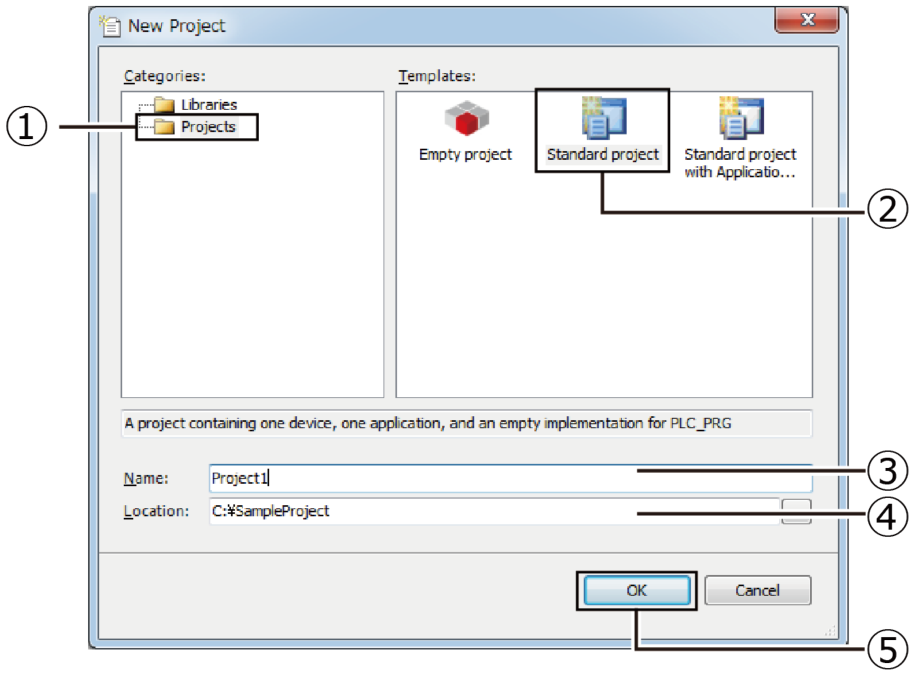
3.Basic programming procedure

1. Create a New Project

1 Select [File] - [New project] on the CODESYS menu.

2 In the New Project dialog, select (1) [Projects] from the Categories and (2) [Standard project] from the Templates.

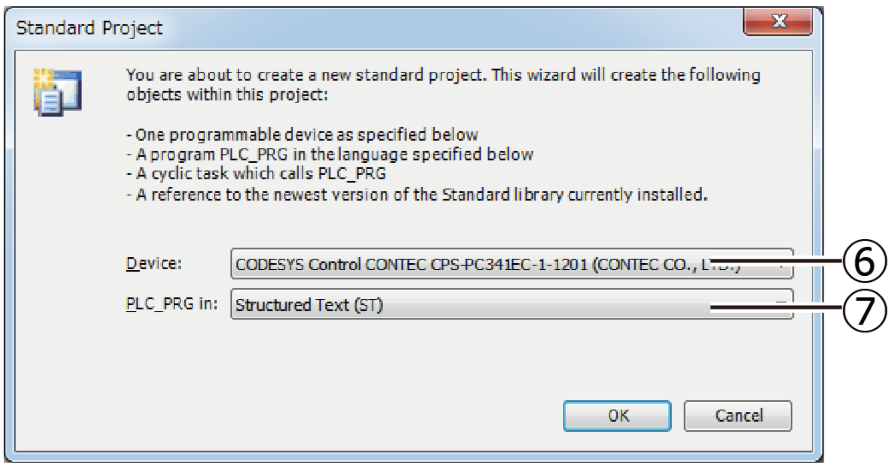
Specify the (3) Name and the (4) Location of the project, then click the [OK].



A project containing one device, one application, and an empty implementation for PLC_PRG

Name: Project1
Location: C:¥SampleProject

3 In the Standard Project dialog, select the controller to be used in the (6) [Device]. Device can be shown as [CODESYS Control CONTEC CPS-PCXXXXXXXXXX]. From the (7) “PLC_PRG”, choose Structured Text (ST).



You are about to create a new standard project. This wizard will create the following objects within this project:

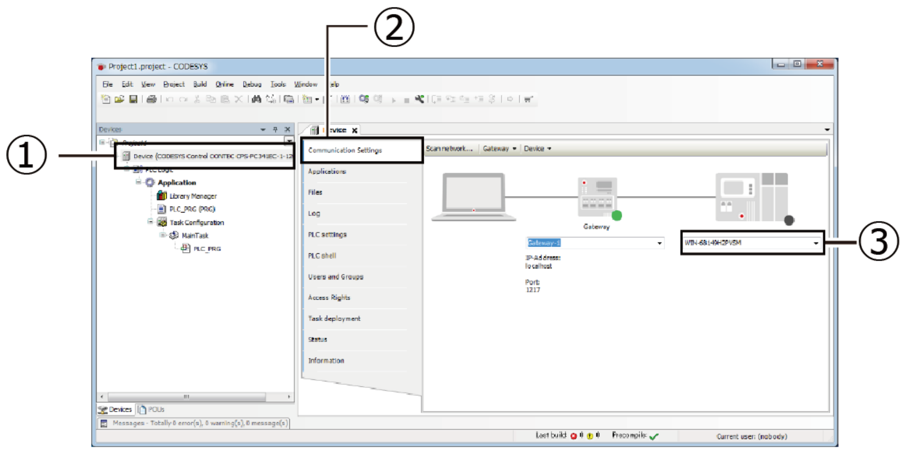
- One programmable device as specified below
- A program PLC_PRG in the language specified below
- A cyclic task which calls PLC_PRG
- A reference to the newest version of the Standard library currently installed.

Device: CODESYS Control CONTEC CPS-PC341EC-1-1201 (CONTEC CO., LTD.)
PLC_PRG in: Structured Text (ST)

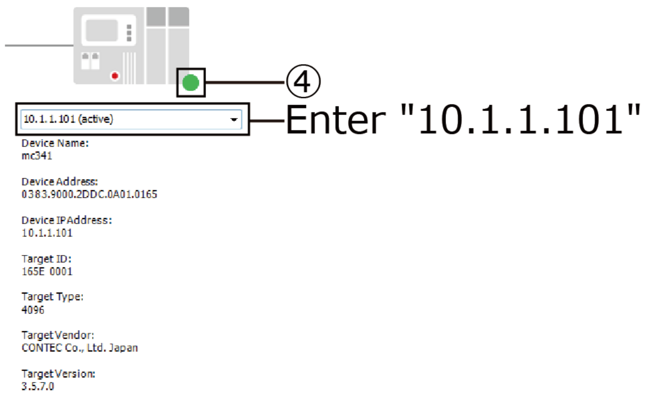
2. Connect controller from CODESYS

1 On the Devices window, double click the (1) “Device (CODESYS Control CONTEC CPS-PCXXXXXXXXXX)” icon to open “Device Configuration window”. Select the (2) [Communication Settings] in the Device tab.

2 Enter “10.1.1.101" into the (3) “Combo-box”, then press [Enter] key on your keyboard.



3 If the color of the circle changes to green (4), it indicates the connection to the controller has been established.



10.1.1.101 (active)

Device Name:
mc341

Device Address:
0383.9000.2DDC.0A01.0165

Device IPAddress:
10.1.1.101

Target ID:
165E 0001

Target Type:
4096

Target Vendor:
CONTEC Co., Ltd. Japan

Target Version:
3.5.7.0

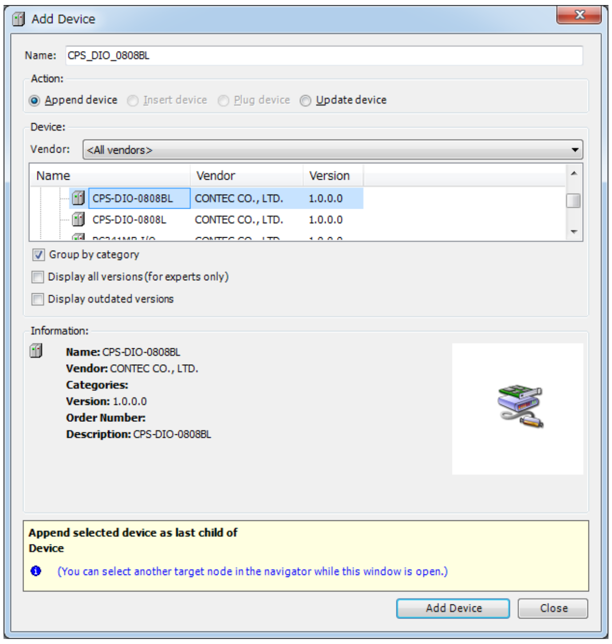
3. Add I/O modules

You can combine configurable type controllers to satisfy your desire.

* For CODESYS device window, I/O functions within the controller are pre-registered under the tree hierarchy (listed as in PCS341XX I/O). Therefore, the following procedure is unnecessary.

1 Right-click on the controller of CODESYS device window and select the [Add device].

2 In the dialog box, select the I/O module you wish to add and click the [Add device] button. I/O module is located under the tree hierarchy of [Others].


Add Device

Name: CPS_DIO_0808BL

Action:
[✓ Append device] Insert device Plug device Update device

Device:
Vendor: <All vendors>

Name	Vendor	Version
CPS-DIO-0808BL	CONTEC CO., LTD.	1.0.0.0
CPS-DIO-0808L	CONTEC CO., LTD.	1.0.0.0

[✓] Group by category
[ ] Display all versions (for experts only)
[ ] Display outdated versions

Information:
Name: CPS-DIO-0808BL
Vendor: CONTEC CO., LTD.
Categories:
Version: 1.0.0.0
Order Number:
Description: CPS-DIO-0808BL

Append selected device as last child of
Device
(You can select another target node in the navigator while this window is open.)

4. I/O Variable Definition

In order to input or output with CODESYS, input variable to I/O Mapping or Parameters, and define variables of I/O channel.

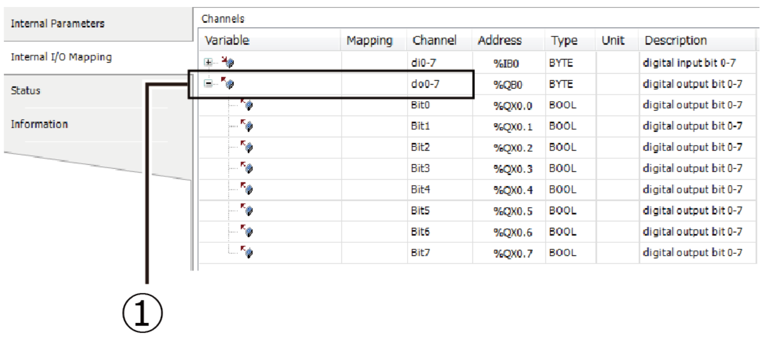
In this example, assigning variable [DO00] in bit 0 of CPS-DIO-0808BL DO port is defined as I/O variables.
This example also applies to most of other I/O modules or I/O in the different controllers.

1 Add [CPS-DIO-0808BL] as demonstrated in the previous page "Add I/O modules", and double click the icon on the device window.

* I/O functions within the controller are pre-registered under the tree hierarchy (listed as in PCS341XX I/O) of CODESYS device window.

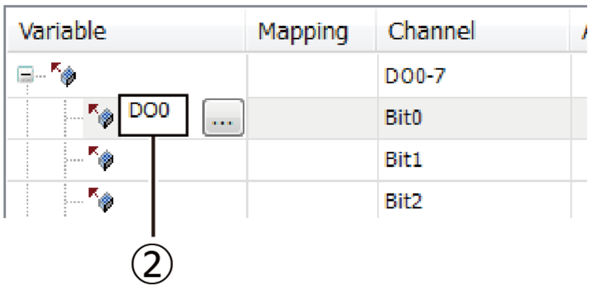
2 When the configuration window appears, select [Internal I/O Mapping].

3 (1) Open the tree that has "do0-7" strings in the [Channel] row.



Variable	Mapping	Channel	Address	Type	Unit	Description
		di0-7	%IB0	BYTE		digital input bit 0-7
		do0-7	%QB0	BYTE		digital output bit 0-7
		Bit0	%QX0.0	BOOL		digital output bit 0-7
		Bit1	%QX0.1	BOOL		digital output bit 0-7
		Bit2	%QX0.2	BOOL		digital output bit 0-7
		Bit3	%QX0.3	BOOL		digital output bit 0-7
		Bit4	%QX0.4	BOOL		digital output bit 0-7
		Bit5	%QX0.5	BOOL		digital output bit 0-7
		Bit6	%QX0.6	BOOL		digital output bit 0-7
		Bit7	%QX0.7	BOOL		digital output bit 0-7

4 Double click on the blank part of the [Variable] row and [Bit0] line in the (2) channel, then define "DO0".



Variable	Mapping	Channel
		DO0-7
DO0		Bit0
		Bit1
		Bit2

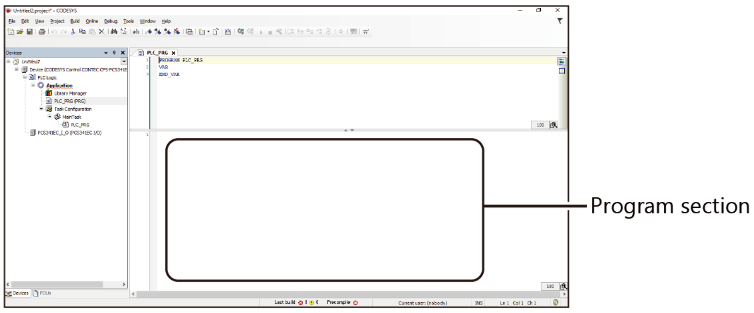
5. Create a program and build

Create PLC program with defined I/O variable“DO0” that is demonstrated in the previous page “I/O Variable Definition”, and set ON output from digital output [Bit0].

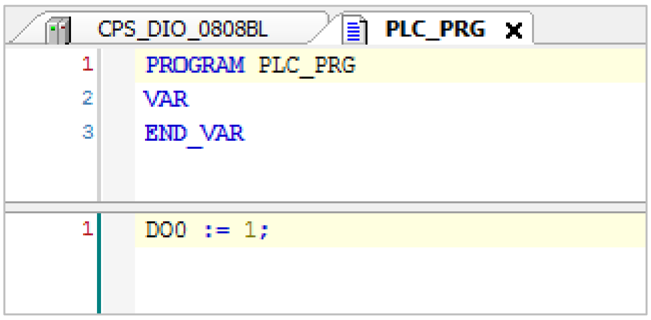
This example demonstrates programming with ST language.

1 Double click the [PLC_PRG (PRG)] icon on the device window to open the [ST Editor window].

2 In the “Program” of ST Editor window, write the following source code.



```
DO0 := 1;
```



```
PROGRAM PLC_PRG
VAR
END_VAR
```

```
DO0 := 1;
```

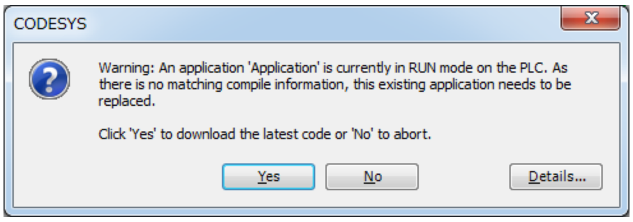
As variable DO0 is defined in output bit 0 in the CPS-DIO-0808BL, ON signal is output from digital output [Bit0] in the CPS-DIO-0808BL.

3 Now, see whether you can execute the [Rebuild] in CODESYS [Build] menu to confirm the process succeeds.
If the process fails, chances are that I/O definition is set inappropriately, firmware version needs to be updated, or there might be a wrong library in the Library manager.

6. Download and run program

1 When building a program demonstrated in the previous page [Create a program and build] is completed, log in from [Online] – [Login] on CODESYS menu.

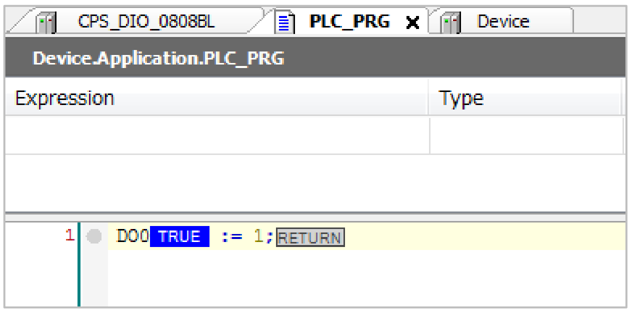
2 Click the [Yes] in the download confirmation dialog.


CODESYS

Warning: An application 'Application' is currently in RUN mode on the PLC. As there is no matching compile information, this existing application needs to be replaced.

Click 'Yes' to download the latest code or 'No' to abort.

3 Select the [Start] in the [Debug] menu to operate the program.
Check that DO Bit0 LED placed on the front panel of CPS-DIO-0808BL is on.
Also, on ST Editor window, the current value of “TRUE” can be seen next to “DO0” variables.



```
DO0 TRUE := 1;RETURN
```

7. Save the PLC program into ROM

Since the PLC program is downloaded to RAM, the program will be disposed upon shutting down. If you want to avoid this, save the programs into ROM area.

1 Start the Web browser on the PC connected with the controller and enter "http://10.1.1.101/" (the IP address of the PLC controller) in the address field.

* Also, accessible through "https://10.1.1.101/".
If the [Certificate Error] message appears, choose the "Continue to this website".

2 Enter user name "pc341" and password "pc341" to log in.

3 From [CODESYS setting], select the [Save PLC program] and click the [Save to ROM].
ST1 and ST2 LEDs continue to flash until writing has accomplished.

⚠CAUTION

ST1 and ST2 LEDs of the product keep flashing while saving the settings.

Do not turn off the power until PWR LED flashing has stopped.　(approx.: five seconds)

Without saving the contents, they return to those before setting at rebooting or shutting down.

8. Delete the PLC program saved in ROM

To delete the PLC program saved in ROM area, follow the steps below.

1 Log in to the controller under the CODESYS development environment.
* See the "**Connect controller from CODESYS (page 58)**" for log-in.

2 Execute [Online] – [Reset origin] in the CODESYS menu.

3 Follow the same procedure described in the "**Save the PLC program into ROM (page 63)**", and save the PLC program into ROM.
Reboot the controller to delete the program.

4.Interface definition per model

How to define I/O interface of Parameter, I/O Mapping, and Serial port with CODESYS are described below.

1. Parameter

Parameter is set only once upon starting a controller. It is listed in the [Internal Parameters] on CODESYS.

Model	Parameter Name	Notation on CODESYS	Meaning
CPS-PC341EC-1-9201	None	---	---
CPS-PC341MB-ADSC1-9201	Digital filter	di filter	Select the value of digital I/O filter.(*1) Default setting : not used
CPS-PCS341EC-DS1-1201 CPS-PCS341MB-DS1-1201	General Purpose Input/Output direction	GPIO0 - GPIO3	Select Input/Output direction from "Input" or "Output". Default setting : Input
CPS-DIO-0808L CPS-DIO-0808BL CPS-DIO-0808RL CPS-DI-16L CPS-DI-16RL	Digital filter	di filter	Select the value of digital input filter. Default setting : not used
	Built-in battery	internal power	Select ON/OFF of built-in battery. (CPS-DIO-0808BL exclusively) Default setting : ON
CPS-SSI-4P	Wire type	wire0-3	Select "3-wire" or "4-wire" for a wire type. Default setting : 4-wire

*1 Digital input filter value

Input Filter Value	
not used	1.024 msec
0.25 μsec	2.048 msec
0.5 μsec	4.096 msec
1.0 μsec	8.192 msec
2.0 μsec	16.384 msec
4.0 μsec	32.768 msec
8.0 μsec	65.536 msec
16.0 μsec	131.072 msec
32.0 μsec	
64.0 μsec	
128.0 μsec	
256.0 μsec	
512.0 μsec	

2. I/O Mapping

I/O Mapping conducted repeatedly per cycle time. It is listed in the [Internal I/O Mapping] on CODESYS.

Model	I/O Mapping Name	Notation on CODESYS	Meaning
CPS-PC341EC-1-9201	Battery status	battery	Battery residual status 0=No 1=Yes
CPS-PC341MB-ADSC1-9201	Digital output	do0-1	Digital output bit 0 to 1
	Digital input	di0-3	Digital input bit 0 to 3
	Counter input 0	cnt0	Counter input channel 0 Range: 0-16,777,215
	Counter input 1	cnt1	Counter input channel 1 Range: 0-16,777,215
	Analog input 0	ai0	Analog input channel 0 Range: 0-4,095
	Analog input 1	ai1	Analog input channel 1 Range: 0-4,095

	Counter clear	cnt clear	Counter value clear command
			Bit0 = ON Clear counter 0
			Bit1 = ON Clear counter 1

	Battery status	battery	Battery residual status 0=No 1=Yes
CPS-PCS341EC-DS1-1201 CPS-PCS341MB-DS1-1201	Battery status	battery	Battery residual status 0=No 1=Yes
	Digital output *	digital outputs (GPIO0..GPIO3)	Digital output bit 0 to 3
	Digital input *	digital inputs (GPIO0..GPIO3)	Digital input bit 0 to 3
CPS-DIO-0808L CPS-DIO-0808BL CPS-DIO-0808RL	Digital output	do0-7	Digital output bit 0 to 7 Range: 0-65,535
	Digital input	di0-7	Digital input bit 0 to 7 Range:0-65,535
CPS-AI-1608LI CPS-AI-1608ALI	Analog input	ai0-7	Analog input channel 0-7 Range:0-65,535
CPS-AO-1604LI CPS-AO-1604VLI	Analog output	ao0-3	Analog output channel 0-3 Range:0 -65,535
CPS-RRY-4PCC	Digital output	do0-3	Digital output bit 0 to 3
CPS-DI-16L CPS-DI-16RL	Digital input	di0 0-7 di1 0-7	Digital input channel 0 bit 0 to 7 Digital input channel 1 bit 0 to 7
CPS-DO-16L CPS-DO-16RL	Digital output	do0 0-7 do1 0-7	Digital output channel 0 bit 0 to 7 Digital output channel 1 bit 0 to 7
CPS-SSI-4P	Temperature input	temp0-3	Temperature input channel 0 to 3 Range: -200 to 800, when a sensor is disconnected -999
	SSI data input(*1)	data0-3	SSI data input channel 0 to 3

Model	I/O Mapping Name	Notation on CODESYS	Meaning
CPS-CNT-3202I	Counter input	cnt val0-1	Counter input channel 0 to 1
	Status input	status0-1	Status input channel 0 to 1
	Control flag output	out ctrl0-1	Control flag output channel 0 to 1 Bit0=0 means Stop, Bit0=1 means Start Changing from Bit1=0 to Bit1=1 sets count value.
	Control flag input	in ctrl0-1	Control flag input channel 0 to 1
	Event input	event0-1	Event input channel 0 to 1 When counter input value coincides with counter value output, it indicates Bit0=1.
	Event reset output	event reset0-1	Event reset output channel 0 to 1 Changing each bit from 0 to 1 resets an event.
	Z mode output	zmode0-1	Z mode output channel0 to 1 0=Not used, 1=Next one time, 2=Every time
	Z logic output	zlogic0-1	Z logic output channel 0 to 1 0= Positive, 1= Negative
	Direction output	dir0-1	Direction output channel 0 to 1 0=Down, 1=Up
	Phase output(*2)	phase0-1	Phase output channel 0 to 1 0=Single-phase, 1=2-phase, 2=Gate control
	Multiply output(*2)	multi0-1	Multiply output channel 0 to 1 0=Multiply by 1, 1= Multiply by 2, 2= Multiply by 4
	Clear output(*2)	clr0-1	Clear output channel 0 to 1 0=Asynchronous clear, 1=Synchronous clear
	Digital filter output(*3)	filter0-1	Digital filter output channel 0 to 1
	One-shot pulse width output(*4)	pulse0-1	One-shot pulse width output channel 0 to 1
	Counter value output	cnt set0-1	Counter value output channel 0 to 1
	Compare value output	cnt match0-1	Compare value output channel 0 to 1

* 4-bit of each I/O mapping is defined for general purpose Input/Output (GPIO). However, I/O that can be used actually will be defined based on the General Purpose Input/Output direction Parameter.

*1 SSI data format

	Start address								Start address + 1								Start address + 2								Start address + 3							
Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	Status								S	MSB																						LSB
	A	Reserved						V		4096°C ↓												1°C ↓										1/1024°C ↓
1°C	0	*	*	*	*	*	*	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
-1 °C	0	*	*	*	*	*	*	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
-1/102°C	0	*	*	*	*	*	*	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
-999°C (Disconnection)	1	*	*	*	*	*	*	*	1	1	1	1	0	0	0	0	0	1	1	0	0	1	*	*	*	*	*	*	*	*	*	*

S: Sign, A: Sensor abnormality, V: Valid data

*: Undefined

*2 Phase/Multiply/Clear

Types of pulse signals (operation modes)
2-phase Input, Synchronous Clear, Multiply by 1
2-phase Input, Synchronous Clear, Multiply by 2
2-phase Input, Synchronous Clear, Multiply by 4
2-phase Input, Asynchronous Clear, Multiply by 1
2-phase Input, Asynchronous Clear, Multiply by 2
2-phase Input, Asynchronous Clear, Multiply by 4
Single-phase Input, Asynchronous Clear, Multiply by 1
Single-phase Input with Gate Control, Asynchronous Clear, Multiply by 1
Single-phase Input with Gate Control, Asynchronous Clear, Multiply by 2

*3 Digital filter

Input Filter Value	
0h	0.1μsec
1h	6.52μsec
2h	25.7μsec
3h	32.1μsec
4h	204.9μsec
5h	211.3μsec
6h	230.5μsec
7h	236.9μsec
8h	819.3μsec
9h	825.7μsec
Ah	844.9μsec
Bh	851.3μsec
Ch	1024.1μsec
Dh	1030.5μsec
Eh	1049.7μsec
Fh	1056.1μsec

*4 One-shot pulse width

Pulse width setting data × 409.6 = Pulse width (μsec)

3. Serial Port

Model	Product notation	Port number on CODESYS
CPS-PC341MB-ADSC1-9201	COM A	Port 1
	COM B	Port 2
CPS-PCS341EC-DS1-1201 CPS-PCS341MB-DS1-1201	COM A	Port 1
CPS-COM-1PC CPS-COM-1PD CPS-COM-2PC CPS-COM-2PD	The product with 1 channel COM A The product with 2 channels COMA, COM B	The port numbers of serial communication module are allocated in order closer to the controller starting from Port 2, Port 3-. (Port 1 is for the controller unit)

4. OPC-UA Symbol configuration

By registering variables used in IEC program as symbols, OPC-UA communication and monitoring with CONPROSYS HMI software can be performed.

Symbols can be set freely, however, the products have been set with symbols of the factory default as listed below.

OPC-UA server will automatically start running after turning on the power.

Model	I/O name	OPC-UA symbol name	Access, Data type, Range
CPS-PC341EC-1-9201	None	---	---
CPS-PC341MB-ADSC1-9201	Digital output bit0 Digital output bit1	DO0 DO1	Read/Write, BIT, 0 or 1
	Digital input bit0 Digital input bit1 Digital input bit2 Digital input bit3	DI0 DI1 DI2 DI3	Read, BIT, 0 or 1
	Analog input 0 Analog input 1	AI0 AI1	Read, DWORD, 0 - 4095
	Counter input 0 Counter input 1	CNT0 CNT1	Read, DWORD, 0 - 16777215
	Counter clear 0 Counter clear 1	CNTCLR0 CNTCLR1	Read/Write, BIT, 0 or 1
CPS-PCS341EC-DS1-1201 CPS-PCS341MB-DS1-1201	Digital input bit0 Digital input bit1 Digital input bit2 Digital input bit3	DI0 DI1 DI2 DI3	Read, BIT, 0 or 1

Communication Settings

This chapter describes the communication style and the settings of PAC series.

The table below lists communication styles and compatible models

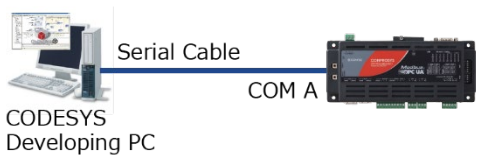
Communication Setting	Model	Slave model
Serial Communication between the controller and the PC	CPS-PC341MB-ADSC1-9201 CPS-PCS341EC-DS1-1201 CPS-PCS341MB-DS1-1201	---
Use the controller as EtherCAT master	CPS-PC341EC-1-9201 CPS-PCS341EC-DS1-1201	CPS-ECS341-1-011 Slave support devices of other makers.
Use the controller as Modbus TCP master	CPS-PC341MB-ADSC1-9201 CPS-PCS341MB-DS1-1201	Slave support devices of CONTEC's including CPS-MC341-ADSC1-111 Slave support devices of other makers
Use the controller as Modbus TCP slave	CPS-PC341MB-ADSC1-9201 CPS-PCS341MB-DS1-1201	---
Use the controller as Modbus RTU master	CPS-PC341MB-ADSC1-9201 CPS-PCS341MB-DS1-1201	Slave support devices of CONTEC's including CPSN-MCB271-S1-041 Slave support devices of other makers.
Use the controller as Modbus RTU slave	CPS-PC341MB-ADSC1-9201 CPS-PCS341MB-DS1-1201	---

1.Serial Communication between the controller and the PC

Model	Slave model
CPS-PC341MB-ADSC1-9201 CPS-PCS341EC-DS1-1201 CPS-PCS341MB-DS1-1201	---

1. Serial Communication Preparation

Connect a controller and a PC via serial cable to send data from the controller to the PC and receive data from the PC to the controller.　Use COM port A within the controller for COM port.



2. Serial Communication Library setting

1 Create a new CODESYS project.
Choose ST for IEC program language.
*See “**Create a New Project (page 57)**” for how to create the project.

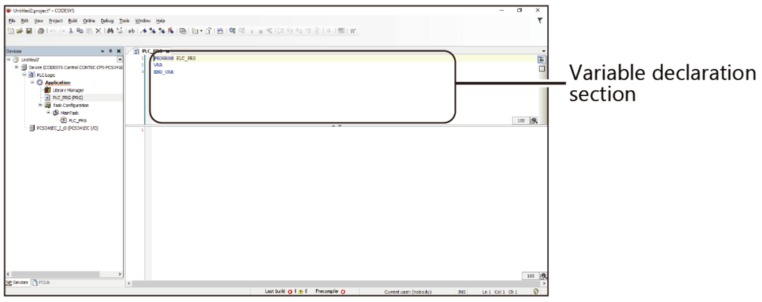
2 Double click the [Library Manager] on the device window.

3 Click the “Add Library”, select the “Serial Communication” under “Use Cases”, then click the “OK”.

3. Create and execute a transfer program

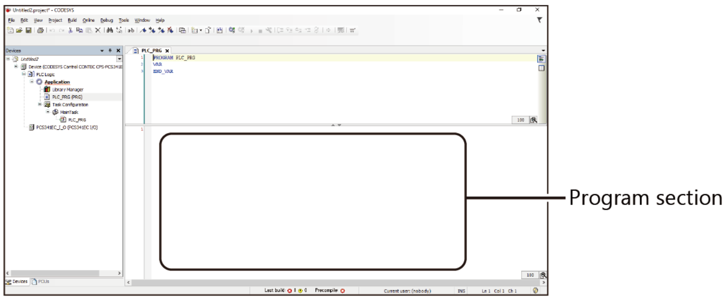
1 Double click the [PLC_PRG (PRG)] icon on the device window to open the “ST editor window”.

2 When “ST editor window” appears, write the following source code between VAR and END_VAR in the “Variable declaration”.



```
StartFlag : BOOL := TRUE;
State : INT := 0;
Com1Params : ARRAY [1..7] OF COM.PARAMETER;
Com1Open : COM.Open;
Com1Close : COM.Close;
Com1Write : COM.Write;
Error : BOOL;
TestString : STRING := 'Test String!';
```

3 Write the following source code in “Program” under “ST editor window”.



```
IF StartFlag THEN
    CASE State OF
    0:
        Com1Params[1].udiParameterId := COM.CAA_Parameter_Constants.udiPort;
        Com1Params[1].udiValue := 3;
        Com1Params[2].udiParameterId := COM.CAA_Parameter_Constants.udiBaudrate;
        Com1Params[2].udiValue := 9600;
        Com1Params[3].udiParameterId := COM.CAA_Parameter_Constants.udiParity;
        Com1Params[3].udiValue := INT_TO_UDINT(COM.PARITY.NONE);
        Com1Params[4].udiParameterId := COM.CAA_Parameter_Constants.udiStopBits;
        Com1Params[4].udiValue := INT_TO_UDINT(COM.STOPBIT.ONESTOPBIT);
        Com1Params[5].udiParameterId := COM.CAA_Parameter_Constants.udiTimeout;
        Com1Params[5].udiValue := 0;
        Com1Params[6].udiParameterId := COM.CAA_Parameter_Constants.udiByteSize;
        Com1Params[6].udiValue := 8;
        Com1Params[7].udiParameterId := COM.CAA_Parameter_Constants.udiBinary;
        Com1Params[7].udiValue := 0;
        Com1Open(xExecute:= TRUE, usiListLength:= SIZEOF(Com1Params) /
                              SIZEOF(COM.PARAMETER), pParameterList:= ADR(Com1Params));
        IF Com1Open.xError THEN
          Error := TRUE;
          State := 1000;
        END_IF
        IF Com1Open.xDone THEN
          State := 1;
        END_IF
      1:
        Com1Write(xExecute := TRUE,hCom:= Com1Open.hCom,pBuffer:=
                              ADR(TestString),szSize:= SIZEOF(TestString));
        IF Com1Write.xError THEN
          Error := TRUE;
          State := 1000;
        END_IF
        IF Com1Write.xDone THEN
          State := 2;
        END_IF
      2:
        Com1Close(xExecute := TRUE, hCom:= Com1Open.hCom);
        IF Com1Close.xError THEN
          Error := TRUE;
        END_IF
        IF Com1Close.xDone OR Com1Close.xError THEN
          State := 1000;
        END_IF
      1000:
        StartFlag := FALSE;
      END_CASE
    END_IF
```

4 Now, see whether you can execute [Rebuild"] in CODESYS [Build] menu to confirm the process succeeds.

◆ Character string transfer

1 Select [Online] – [Login] on the CODESYS menu.

2 Confirmation dialog box for download appears. Click the "OK".

3 With terminal connected to serial port from PC, wait in a reception state.

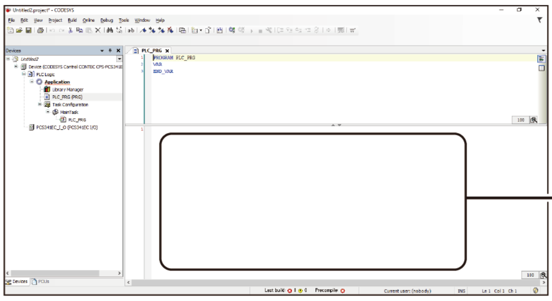
4 Select [Debug] – [Start] and the program operation will start.
The character string "Test String!" can be viewed on the PC terminal monitor.

4. Create and execute a reception program

1 Create a new project for a reception program.
Choose ST for IEC program language.
*See "**Create a New Project (page 57)**" for how to create the project.

2 Double click the [PLC_PRG (PRG)] icon on the device window to open the "ST editor window".

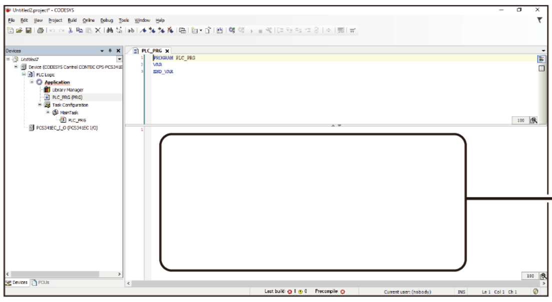
3 When "ST editor window" appears, write the following source code between VAR and END_VAR in the "Variable declaration".


Untitled2.project* - CODESYS
File Edit View Project Build Online Debug Tools Window Help
Devices
Untitled2
Device (CODESYS Control CONTEC CPS-PCS341E
PLC Logic
Application
Library Manager
PLC_PRG (PRG)
Task Configuration
MainTask
PLC_PRG
PCS341EC_I_O (PCS341EC I/O)
PLC_PRG
Program section

```
PROGRAM PLC_PRG
VAR
END_VAR
```

```
StartFlag : BOOL := TRUE;
State : INT := 0;
Com1Params : ARRAY [1..7] OF COM.PARAMETER;
Com1Open : COM.Open;
Com1Close : COM.Close;
Com1Write : COM.Write;
Error : BOOL;
TestString : STRING := 'Test String!';
```

4 Write the following source code in “Program” under “ST editor window”.


Untitled2.project* - CODESYS
File Edit View Project Build Online Debug Tools Window Help
Devices
Untitled2
Device (CODESYS Control CONTEC CPS-PCS341E
PLC Logic
Application
Library Manager
PLC_PRG (PRG)
Task Configuration
MainTask
PLC_PRG
PCS341EC_I_O (PCS341EC I/O)
PLC_PRG
Program section

```
PROGRAM PLC_PRG
VAR
END_VAR
```

```
IF StartFlag THEN
CASE State OF
  0:
    Com1Params[1].udiParameterId := COM.CAA_Parameter_Constants.udiPort;
    Com1Params[1].udiValue := 1;
    Com1Params[2].udiParameterId := COM.CAA_Parameter_Constants.udiBaudrate;
    Com1Params[2].udiValue := 9600;
    Com1Params[3].udiParameterId := COM.CAA_Parameter_Constants.udiParity;
    Com1Params[3].udiValue := INT_TO_UDINT(COM.PARITY.NONE);
    Com1Params[4].udiParameterId := COM.CAA_Parameter_Constants.udiStopBits;
    Com1Params[4].udiValue := INT_TO_UDINT(COM.STOPBIT.ONESTOPBIT);
    Com1Params[5].udiParameterId := COM.CAA_Parameter_Constants.udiTimeout;
    Com1Params[5].udiValue := 0;
    Com1Params[6].udiParameterId := COM.CAA_Parameter_Constants.udiByteSize;
    Com1Params[6].udiValue := 8;
    Com1Params[7].udiParameterId := COM.CAA_Parameter_Constants.udiBinary;
    Com1Params[7].udiValue:= 0;
    Com1Open(xExecute:= TRUE, usiListLength:= SIZEOF(Com1Params) /
                    SIZEOF(COM.PARAMETER), pParameterList:= ADR(Com1Params));
    IF Com1Open.xError THEN
      Error := TRUE;
      State := 1000;
    END_IF
    IF Com1Open.xDone THEN
      State := 1;
    END_IF
  1:
    Com1Read(xExecute := TRUE, hCom:= Com1Open.hCom, pBuffer:=
                    ADR(TestString), szBuffer:= SIZEOF(TestString));
    IF Com1Read.xError THEN
      Error := TRUE;
      State := 1000;
    END_IF
    IF Com1Read.xDone THEN
      State := 2;
```

```
        END_IF
      2:
        Com1Close(xExecute := TRUE, hCom:= Com1Open.hCom);
        IF Com1Close.xError THEN
          Error := TRUE;
        END_IF
        IF Com1Close.xDone OR Com1Close.xError THEN
          State := 1000;
        END_IF
      1000:
        StartFlag := FALSE;
      END_CASE
    END_IF
```

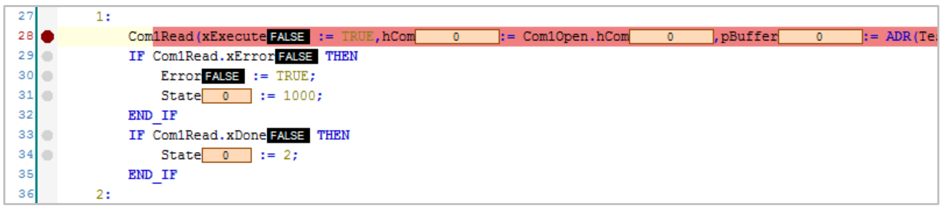
◆ Character string transfer

1 Select [Online] – [Login] on the CODESYS menu.

2 Confirmation dialog box for download appears. Click the “Yes”

3 With terminal connected to serial port from PC, wait in a transfer state.

4 Enter a break point in a reception-function of program source code. In the line calling Com1Read-function, select “Debug” – “New break point”.

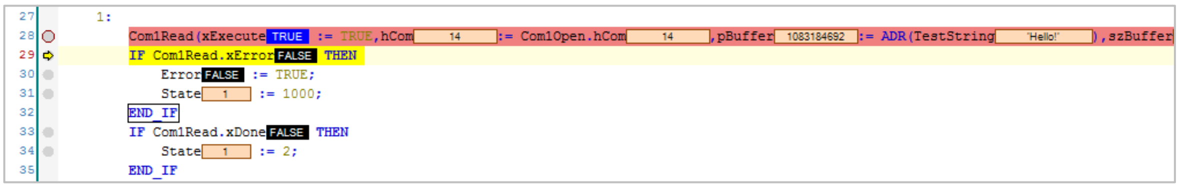


```
    1:
        Com1Read(xExecute FALSE := TRUE,hCom 0 := Com1Open.hCom 0 ,pBuffer 0 := ADR(Te
        IF Com1Read.xError FALSE THEN
            Error FALSE := TRUE;
            State 0 := 1000;
        END_IF
        IF Com1Read.xDone FALSE THEN
            State 0 := 2;
        END_IF
    2:
```

5 Select [Debug] – [Start] and the program operation will start.
The program will pause at the position of the break point.

6 Enter the string to transfer through PC terminal.

7 Press F10 on CODESYS side to execute the program 1 step.
You can check the string sent through PC terminal is saved in the Test String variable.



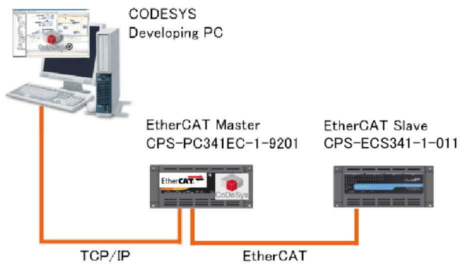
```
    1:
        Com1Read(xExecute TRUE := TRUE,hCom 14 := Com1Open.hCom 14 ,pBuffer 1083184692 := ADR(TestString 'Hello!' ),szBuffer
        IF Com1Read.xError FALSE THEN
            Error FALSE := TRUE;
            State 1 := 1000;
        END_IF
        IF Com1Read.xDone FALSE THEN
            State 1 := 2;
        END_IF
```

2.Use the controller as EtherCAT Master

Model	Slave model
CPS-PC341EC-1-9201 CPS-PCS341EC-DS1-1201	CPS-ECS341-1-011 Slave support devices of other maker's

1. Device preparation

This section describes an example of CODESYS programming with the CPS-ECS341-1-011 as EtherCAT slave.



1 Connect LAN A of CPS-PC341EC-1-9201 and LAN port of PC via Ethernet cable.

2 Connect LAN B of CPS-PC341EC-1-9201 and IN port of CPS-ECS341-1-011 via Ethernet cable.

2. Add Slave Device

To use the EtherCAT slave controller, registration of the ESI (EtherCAT Slave Information) file is required.

With the CONTEC slave controller, the ESI file is already pre-installed in the CODESYS package. Therefore, registering 1-3 below is unnecessary.

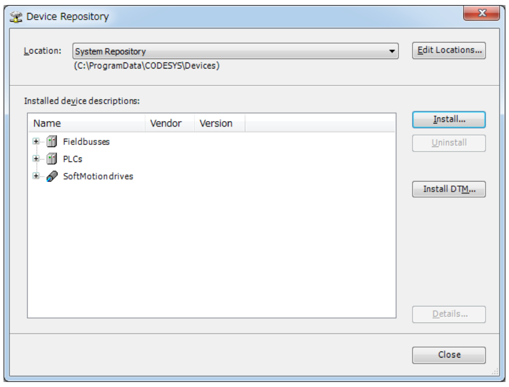
When using the slave device of other maker's, follow 1 to 3 to install the ESI (EtherCAT Slave Information) file.

1 Create a New CODESYS project.
Choose the name of connected EtherCAT model for device, and ST for IEC program language.
*See **"Create a New Project (page 57)**" for how to create the project.

2 Download the ESI file from the maker you intend to use the device.

3 Select the [Tools] - [Device Repository] on the CODESYS menu.

4 Click the [Install] and specify ESI file in "CONTEC_ CPS-ECS341-1-011_v100". (or the downloaded ESI file of other maker's)


Device Repository

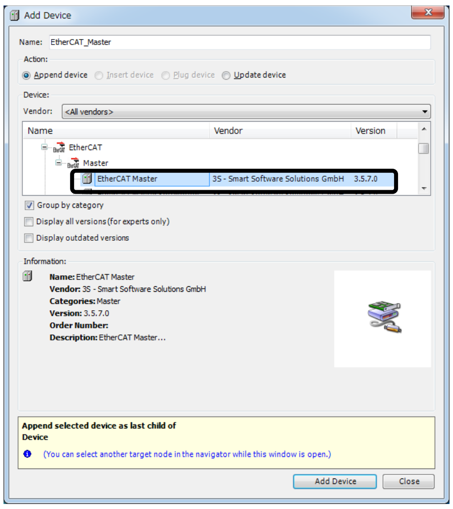
Location: System Repository
(C:\ProgramData\CODESYS\Devices)

Installed device descriptions:

Name	Vendor	Version
Fieldbusses		
PLCs		
SoftMotion drives		

5 Right-click on the [Device (CODESYS Control CONTEC CPS-PC341EC-1-9201)] icon on the device window and select the [Add Device].

6 On the Add device window, select the [EtherCAT Master] and click the [Add Device].
The [EtherCAT Master] is located under [Fieldbusses - EtherCAT – Master] tree hierarchy.


Add Device

Name: EtherCAT_Master

Action:
[✓ Append device] Insert device Plug device Update device

Device:
Vendor: <All vendors>

Name	Vendor	Version
EtherCAT		
Master		
EtherCAT Master	3S - Smart Software Solutions GmbH	3.5.7.0

[✓] Group by category
[ ] Display all versions (for experts only)
[ ] Display outdated versions

Information:
Name: EtherCAT Master
Vendor: 3S - Smart Software Solutions GmbH
Categories: Master
Version: 3.5.7.0
Order Number:
Description: EtherCAT Master...

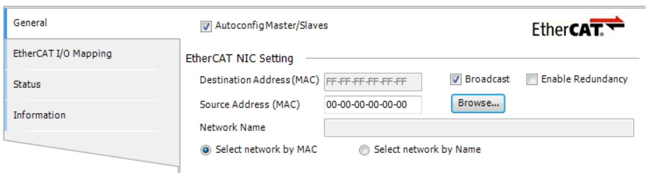
Append selected device as last child of
Device
(You can select another target node in the navigator while this window is open.)

7 Right-click on the added [EtherCAT_Master] icon on the device window and click the [Add Device].

8 Select [CPS-ECS341-1-011] or the device name you intend to use, and click the [Add Device] in the [Add Device] window.
"CPS-ECS341-1-011" is located under "Fieldbuses – EtherCAT –Slave –CONTEC Co., Ltd. – CONPROSYS EtherCAT" tree hierarchy.

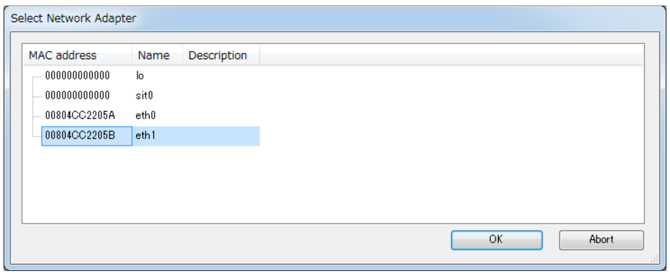
9 Double click the [EtherCAT_Master (EtherCAT Master)] icon on the device window to open the configuration window.

10 In [General] tab, click the [Browse] button and select [eth1].



[✓] Autoconfig Master/Slaves

EtherCAT NIC Setting
Destination Address (MAC) FF-FF-FF-FF-FF-FF [✓] Broadcast [ ] Enable Redundancy
Source Address (MAC) 00-00-00-00-00-00 Browse...
Network Name
[✓ Select network by MAC] Select network by Name


Select Network Adapter

MAC address	Name	Description
000000000000	lo	
000000000000	sit0	
00804CC2205A	eth0	
00804CC2205B	eth1	

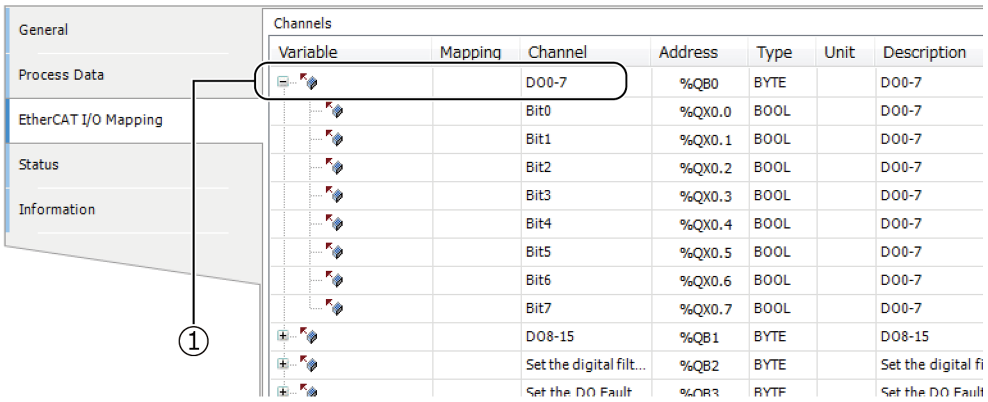
3. Set Slave Device

This section describes how to set EtherCAT slave device to output ON/OFF data continuously from bit0 of output 2-port of CPS-DIO-0808L.

1 Double click the [CPS-DIO-0808L] icon on the device window.

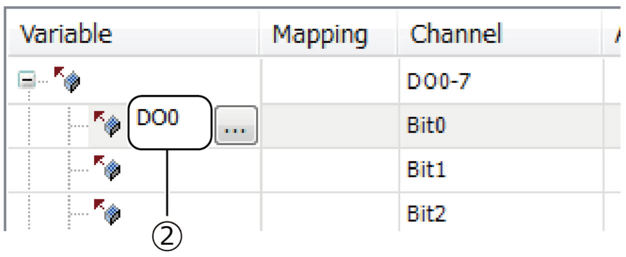
2 When the CPS-DIO-0808L configuration window appears, select [EtherCAT I/O Mapping].

3 Open the tree that has “DO0-7” string in (1) “Channel” row.



Variable	Mapping	Channel	Address	Type	Unit	Description
		DO0-7	%QB0	BYTE		DO0-7
		Bit0	%QX0.0	BOOL		DO0-7
		Bit1	%QX0.1	BOOL		DO0-7
		Bit2	%QX0.2	BOOL		DO0-7
		Bit3	%QX0.3	BOOL		DO0-7
		Bit4	%QX0.4	BOOL		DO0-7
		Bit5	%QX0.5	BOOL		DO0-7
		Bit6	%QX0.6	BOOL		DO0-7
		Bit7	%QX0.7	BOOL		DO0-7
		DO8-15	%QB1	BYTE		DO8-15
		Set the digital filt...	%QB2	BYTE		Set the digital fi
		Set the DO Fault	%QB3	BYTE		Set the DO Fault

4 Double click on the blank part of the [Variable] row and [Bit0] line in the (2) channel, then define “DO0”.



Variable	Mapping	Channel
		DO0-7
DO0		Bit0
		Bit1
		Bit2

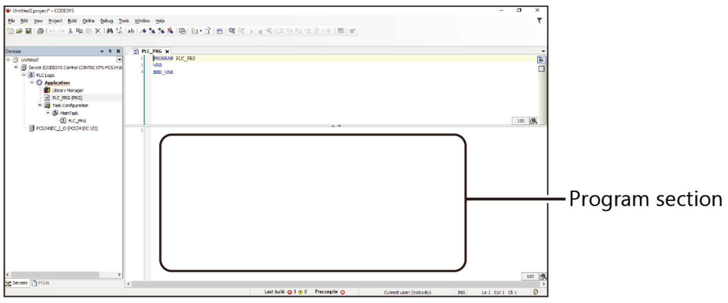
4. Create a program and build

Step 1 and 2 are the instructions of creating a program to output ON/OFF data continuously from bit0 of output 2-port of CPS-DIO-0808L.

Step 3 and 4 are the cycle time setting to check LED flashing.

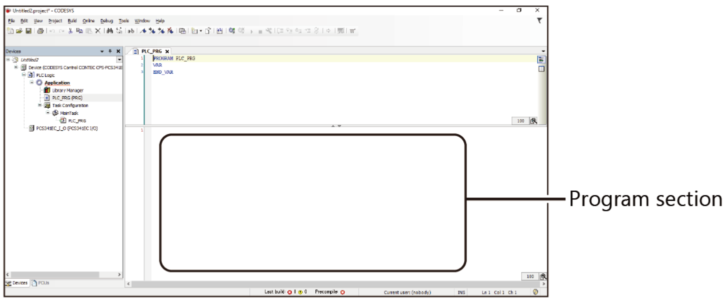
1 Double click the [PLC_PRG (PRG)] icon on the device window to open the "ST editor window". When ST editor window appears, write the following source code.

2 When ST editor window appears, write the following source code in the Program.



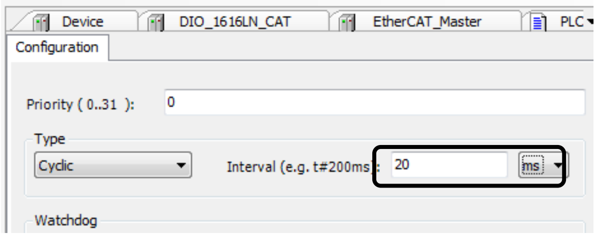
```
DO0 := NOT DO0;
```

The program above set "1" and "0" to "DO0" variable repeatedly for each cycle.



3 Double click the [MainTask] icon on the device window.

4 Change [Interval] from "4000μs" to "20ms".



Priority (0..31): 0

Type: Cyclic

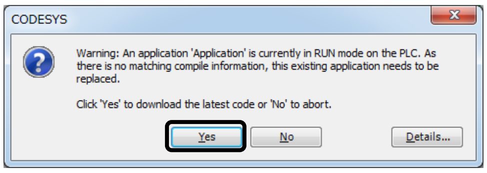
Interval (e.g. t#200ms): 20 ms

5 Execute [Build] – [Rebuild] on the CODESYS menu and check whether the [Build] process succeeds.

5. Download and run program

1 Select [Online] – [Login] on the CODESYS menu.

2 Click the [Yes] in the download confirmation dialog.



CODESYS

Warning: An application 'Application' is currently in RUN mode on the PLC. As there is no matching compile information, this existing application needs to be replaced.

Click 'Yes' to download the latest code or 'No' to abort.

3 Select [Debug] – [Start] and EtherCAT operation will start. Flashing Bit0 LED of CPS-DIO-0808Lcan be viewed.

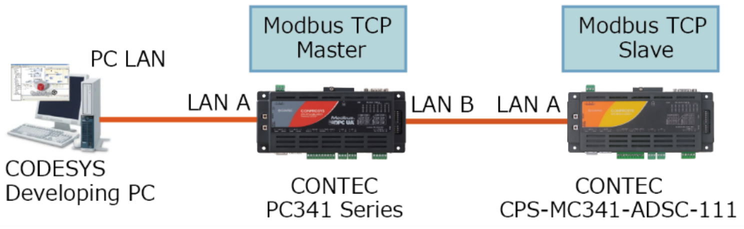
3.Use the controller as Modbus TCP Master

Model	Slave model
CPS-PC341MB-ADSC1-9201 CPS-PCS341MB-DS1-1201	Slave support devices of CONTEC's including CPS-MC341-ADSC1-111 Slave support devices of other maker's

1. Device preparation

This section describes an example of using a controller as Modbus TCP Master to input/output data with Modbus TCP Slave.

Here, CONTEC CPS- PC341MB-ADSC1-9201 is used as Modbus TCP Master, and CONTEC CPS-MC341-ADSC1-111 is used as Modbus TCP Slave.



1 Connect LAN A of the controller to LAN port of CODESYS developing PC.

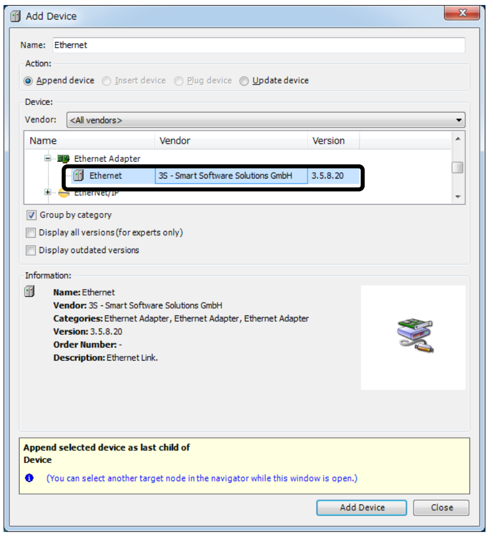
2 Connect LAN B of the controller to CONTEC CPS-MC341-ADSC1-111.

2. Add slave device

1 Create a New CODESYS project.
Choose the name of connected Modbus model for device, and ST for IEC program language.
*See "**Create a New Project (page 57)**" for how to create the project.

2 Right-click on the controller [Device (CODESYS Control CONTEC CPS-PCXXXXXXXXXX)] on the device window and select the [Add device].

3 In the dialog box, select [Fieldbuses - Ethernet Adapter – Ethernet] and click the [Add device] button.


Add Device

Name: Ethernet

Action:
[✓ Append device] Insert device Plug device Update device

Device:
Vendor: <All vendors>

Name	Vendor	Version
Ethernet Adapter		
Ethernet	3S - Smart Software Solutions GmbH	3.5.8.20
EtherNet/IP		

[✓] Group by category
[ ] Display all versions (for experts only)
[ ] Display outdated versions

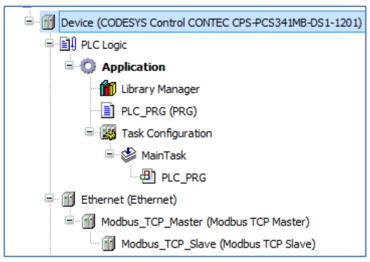
Information:
Name: Ethernet
Vendor: 3S - Smart Software Solutions GmbH
Categories: Ethernet Adapter, Ethernet Adapter, Ethernet Adapter
Version: 3.5.8.20
Order Number: -
Description: Ethernet Link.

Append selected device as last child of
Device
(You can select another target node in the navigator while this window is open.)

4 Right-click on the [Ethernet] on the device window and click the [Add device].

5 In the [Add device] dialog box, add [Fieldbuses - Modbus - Modbus TCP Master].

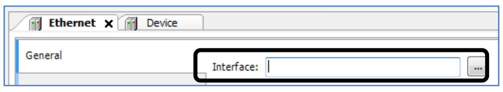
6 Again, right-click on the [Modbus TCP Master] and select the [Add device]. Then add [Fieldbuses - Modbus - Modbus TCP Slave].


```
Device (CODESYS Control CONTEC CPS-PCS341MB-DS1-1201)
  PLC Logic
    Application
      Library Manager
      PLC_PRG (PRG)
      Task Configuration
        MainTask
          PLC_PRG
  Ethernet (Ethernet)
    Modbus_TCP_Master (Modbus TCP Master)
      Modbus_TCP_Slave (Modbus TCP Slave)
```

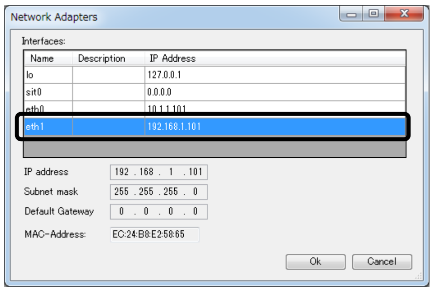
7 On the Device window, double click the [Ethernet] icon to open the Device Configuration window.

8 Open the [General] tab.

9 Click the button next to [Interface] textbox, and select [eth1] in the list. [eth1] will be set as an internal name of LAN B.



Interface:



Interfaces:

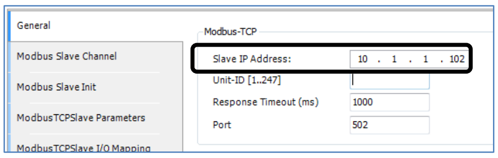
Name	Description	IP Address
lo		127.0.0.1
sit0		0.0.0.0
eth0		10.1.1.101
eth1		192.168.1.101

IP address	192 . 168 . 1 . 101
Subnet mask	255 . 255 . 255 . 0
Default Gateway	0 . 0 . 0 . 0
MAC-Address:	EC:24:B8:E2:58:65

10 On the Device window, double click the [Modbus_TCP_Slave] icon to open Device Configuration window.

11 Open the [General] tab.

12 Enter IP address that has been set in the slave device into the [Slave IP address].



Slave IP Address:	10 . 1 . 1 . 102
Unit-ID [1..247]	
Response Timeout (ms)	1000
Port	502

3. Modbus TCP Master setting

This section describes how to create samples of 2-bit digital output and 4-bit digital input.

The table below is the function code of the slave device CPS-MC341-ADSC1-111.

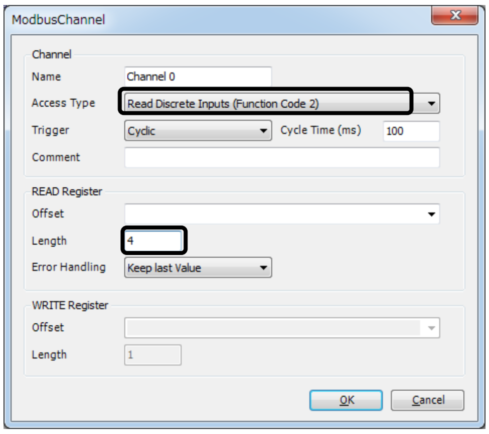
Name/Function code	Address	Data
Read Coils (Code 1)	0	Digital output bit 0
	1	Digital output bit 1
Read Discrete Inputs (Code 2)	0	Digital input bit 0
	1	Digital input bit 1
	2	Digital input bit 2
	3	Digital input bit 3
Read Input Registers (Code 4)	0	Analog input channel 0 (16-bit)
	1	Analog input channel 1 (16-bit)
	2	Counter input channel 0 upper 16-bit
	3	Counter input channel 0 lower 16-bit
	4	Counter input channel 1 upper 16-bit
	5	Counter input channel 1 lower 16-bit
Write Single Coil (Code 5)	0	Digital output bit 0
	1	Digital output bit 1

* The samples are created with Write Single Coil (code 5) for 2-bit digital output, and Read Discrete Inputs (code 2) for 4-bit digital input.

1 On the Devices window, double click the [Modbus_TCP_Slave] icon to open Device Configuration window.

2 Open the [Modbus Slave channel] tab.

3 Click the [Add channel] button. Select [Read Discrete Inputs (function code 2)] from Access type, and enter “4” into Length filed in the Read register, then click the [OK].



ModbusChannel

Channel

Name	Channel 0
Access Type	Read Discrete Inputs (Function Code 2)
Trigger	Cyclic
Cycle Time (ms)	100
Comment	

READ Register

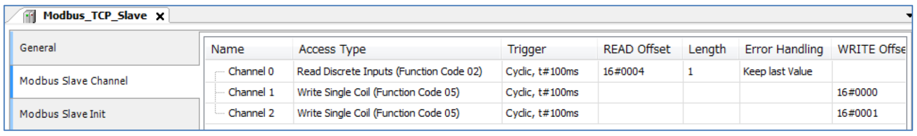
Offset	
Length	4
Error Handling	Keep last Value

WRITE Register

Offset	
Length	1

4 Click the [Add channel] button. Select [Write Single Coil (function code 5)] from Access type, and enter “0” into Write register offset, then click the [OK].

5 Click the [Add channel] button. Select [Write Single Coil (function code 5)] from Access type again, and enter “1” into Write register offset, then click the [OK].

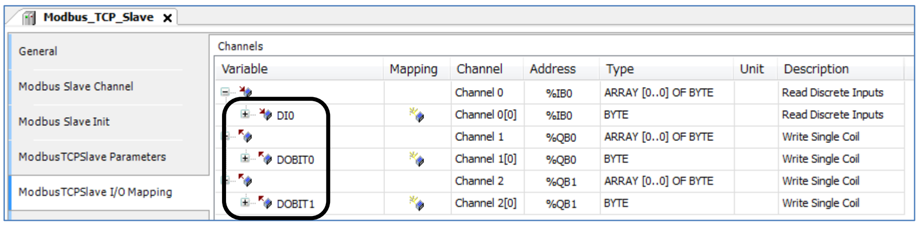

Modbus_TCP_Slave

General
Modbus Slave Channel
Modbus Slave Init

Name	Access Type	Trigger	READ Offset	Length	Error Handling	WRITE Offse
Channel 0	Read Discrete Inputs (Function Code 02)	Cyclic, t#100ms	16#0004	1	Keep last Value	
Channel 1	Write Single Coil (Function Code 05)	Cyclic, t#100ms				16#0000
Channel 2	Write Single Coil (Function Code 05)	Cyclic, t#100ms				16#0001

6 Click the [Modbus TCP Slave I/O Mapping].

7 Allocate "DI0","DOBIT0","DOBIT1" to Channel0 [0], Channel1 [0], Channel2 [0] respectively.


Modbus_TCP_Slave

General
Modbus Slave Channel
Modbus Slave Init
ModbusTCPSlave Parameters
ModbusTCPSlave I/O Mapping

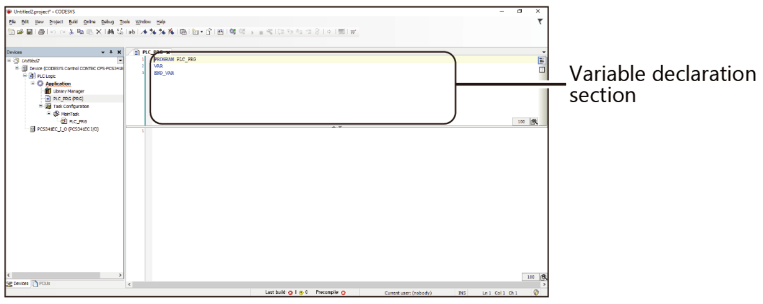
Channels

Variable	Mapping	Channel	Address	Type	Unit	Description
		Channel 0	%IB0	ARRAY [0..0] OF BYTE		Read Discrete Inputs
DI0		Channel 0[0]	%IB0	BYTE		Read Discrete Inputs
		Channel 1	%QB0	ARRAY [0..0] OF BYTE		Write Single Coil
DOBIT0		Channel 1[0]	%QB0	BYTE		Write Single Coil
		Channel 2	%QB1	ARRAY [0..0] OF BYTE		Write Single Coil
DOBIT1		Channel 2[0]	%QB1	BYTE		Write Single Coil

4. Create and build a program

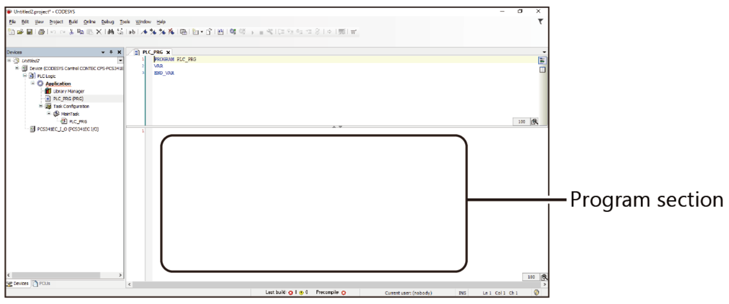
1 Double click the [PLC_PRG (PRG)] icon on the device window to open the “ST editor window”.

2 When ST editor window appears, write the following source code between VAR and END_VAR in the “Variable declaration”.



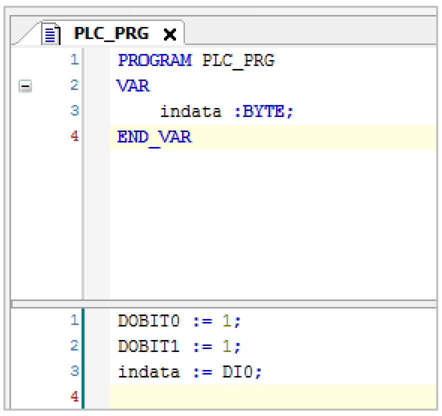
```
indata :BYTE;
```

3 Write the following source code in “Program” under “ST editor window”.



```
DOBIT0 := 1;
DOBIT1 := 1;
indata := DI0;
```

This program outputs ON signal to digital output bit 0 and 1, and inputs signal from digital input port.



```
PROGRAM PLC_PRG
VAR
    indata :BYTE;
END_VAR
```

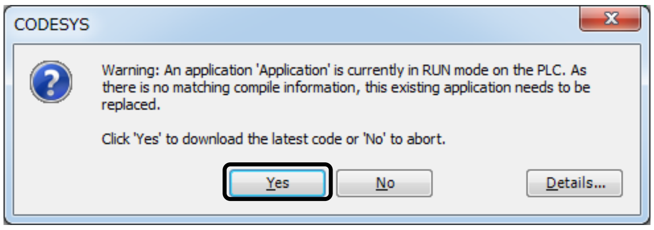
```
DOBIT0 := 1;
DOBIT1 := 1;
indata := DI0;
```

4 Now, see whether you can execute [Rebuild] in CODESYS [Build] menu to confirm the process succeeds properly.

5. Download and run program

1 Select the [Online] – [Login] on the CODESYS menu.

2 Click the [Yes] in the download confirmation dialog.



CODESYS

Warning: An application 'Application' is currently in RUN mode on the PLC. As there is no matching compile information, this existing application needs to be replaced.

Click 'Yes' to download the latest code or 'No' to abort.

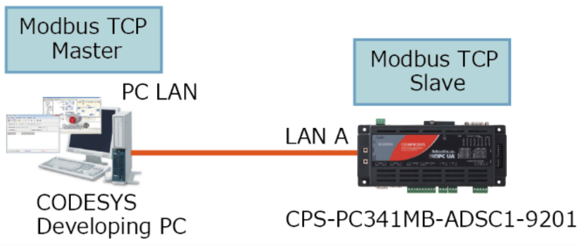
3 Select the [Debug] - [Start] and the program operation will start.
You can see that LED of DO Bit0, 1 on the front panel of CPS-PC341MB-ADSC1-9201 is on.

4. Use the controller as Modbus TCP Slave

Model	Slave model
CPS-PC341MB-ADSC1-9201 CPS-PCS341MB-DS1-1201	---

1. Device preparation

This section describes an example of using a controller as Modbus TCP Slave to input/output data with Modbus TCP Master.
Here, the PC with CODESYS developing environment is used as Modbus TCP Slave, and CONTEC CPS-PC341MB-ADSC1-9201 is used as Modbus TCP Slave for CODESYS programming.
QModMaster is used as Modbus TCP Master software.



1 Install Modbus TCP Master software QModMaster into the PC.
QModMaster can be downloaded free through the URL below.

Download http://sourceforge.net/projects/qmodmaster/

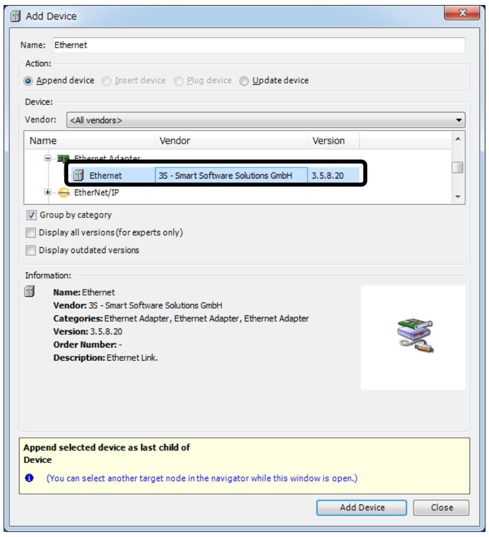
2 Connect LAN A of CPS-PC341MB-ADSC1-9201 and LAN port of PC with CODESYS developing environment via Ethernet cable.

2. Slave setting

1 Create a new CODESYS project.
Choose ST for IEC program language.
*See "**Create a New Project (page 57)**" for how to create the project.

2 Right-click on the controller [Device (CODESYS Control CONTEC CPS-PCXXXXXXXXXX)] on the device window, and select the [Add device].

3 In the [Add device] dialog box, select [Fieldbuses - Ethernet Adapter - Ethernet] and click the [Add device] button.
* This adds EthernetAdapter.


Add Device

Name: Ethernet

Action:
[✓ Append device] Insert device Plug device Update device

Device:
Vendor: <All vendors>

Name	Vendor	Version
Ethernet Adapter		
Ethernet	3S - Smart Software Solutions GmbH	3.5.8.20
EtherNet/IP		

[✓] Group by category
[ ] Display all versions (for experts only)
[ ] Display outdated versions

Information:
Name: Ethernet
Vendor: 3S - Smart Software Solutions GmbH
Categories: Ethernet Adapter, Ethernet Adapter, Ethernet Adapter
Version: 3.5.8.20
Order Number: -
Description: Ethernet Link.

Append selected device as last child of
Device
(You can select another target node in the navigator while this window is open.)

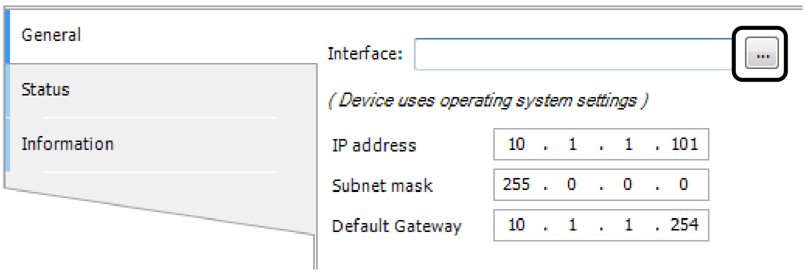
4 Right-click on the [Ethernet] and select the [Add device].

5 In the [Add device] dialog box, select the [Fieldbusses – Modbus - ModbusTCP Slave Device - ModbusTCP Slave Device].
* This adds Modbus TCP Slave device under Ethernet Adapter device.

6 On the Devices window, double click the [Ethernet] icon to open Device Configuration window.

7 Open the [General] tab.

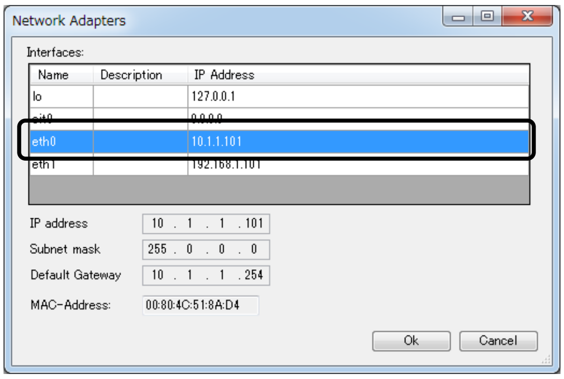
8 Click the button next to the [Interface] textbox, and select the [eth0] in the list.
* IP address is set for Ethernet Adapter
* "eth0" is set as an internal name of LAN A.



Interface:

(Device uses operating system settings)

IP address	10 . 1 . 1 . 101
Subnet mask	255 . 0 . 0 . 0
Default Gateway	10 . 1 . 1 . 254



Interfaces:

Name	Description	IP Address
lo		127.0.0.1
sit0		0.0.0.0
eth0		10.1.1.101
eth1		192.168.1.101

IP address	10 . 1 . 1 . 101
Subnet mask	255 . 0 . 0 . 0
Default Gateway	10 . 1 . 1 . 254
MAC-Address:	00:80:4C:51:8A:D4

9 On the Devices window, double click the [PC341MB_I_O] icon to open Device Configuration window.

10 Click the [Internal I/O Mapping].

11 Define variables as follows:

Variable (Input value)	Channel	Address
	do0-1	
DO0	Bit0	%QX20.0
DO1	Bit1	%QX20.1
	di0-3	
DI0	Bit0	%IX20.0
DI1	Bit1	%IX20.1
DI2	Bit2	%IX20.2
DI3	Bit3	%IX20.3
CNT0	cnt0	%ID6
CNT1	cnt1	%ID7
AI0	ai0	%ID8
AI1	ai1	%ID9

12 On the Devices window, double click the [ModbusTCP_Slave_Device] icon to open Device Configuration window.

13 Click the [Modbus TCP Slave Device I/O Mapping].

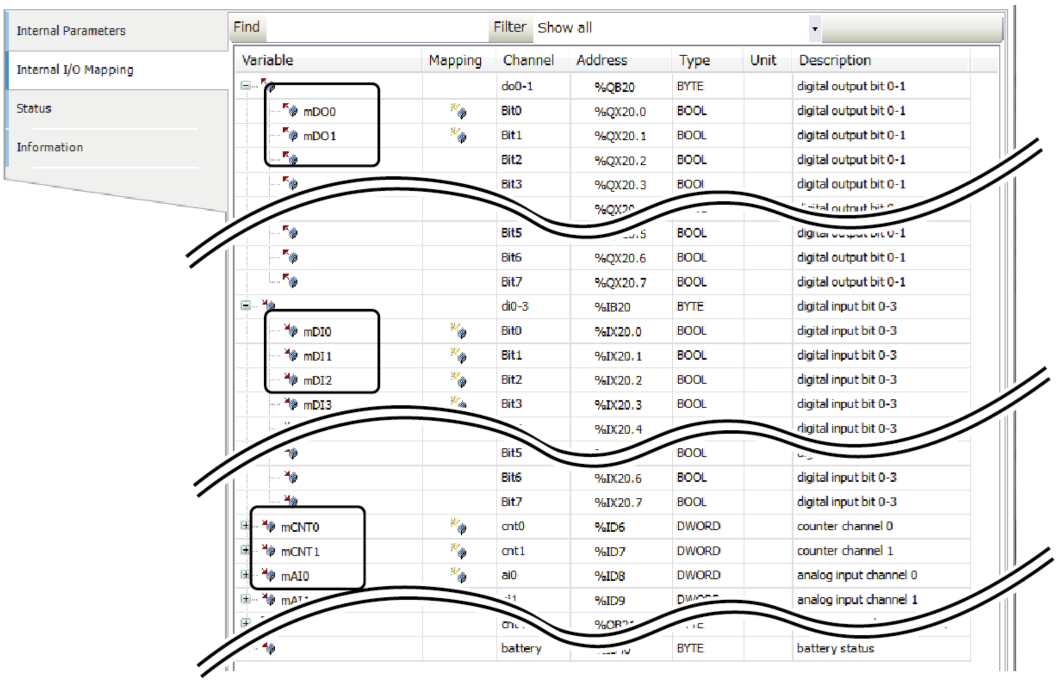
14 Define variables as follows:

Variable (Input value)	Channel	Address
	Input	%IW1
	Input[0]	%IW1
mDO0	Bit0	%IX2.0
mDO1	Bit1	%IX2.1
	:	
	Bit15	%IX3.7
	Input[1]	%IW2
	:	
	Input[9]	%IW10
	Output	%QW1
	Output[0]	%QW1
mDI0	Bit0	%QX0.0
mDI1	Bit1	%QX0.1
mDI2	Bit2	%QX0.2
mDI3	Bit3	%QX0.3
	:	
	Bit15	%QX1.7
mAI0	Output[1]	%QW2
MAI1	Output[2]	%QW3
mCNT0_L	Output[3]	%QW4
mCNT0_H	Output[4]	%QW5
mCNT1_L	Output[5]	%QW6
mCNT1_H	Output[6]	%QW7
	:	
	Output[9]	%QW10

* Now, I/O Mapping is set, and I/O Mapping for Modbus Slave is defined.
Data communication with Modbus Master can be carried out by defining variables into I/O Mapping of Modbus Slave.

* You can see that variables for output in Inputs Mapping area, and variables for input in Outputs Mapping area are defined. This means Master inputs for Slave in Inputs Mapping area, and Slave outputs for Master in Outputs Mapping area.

* Also note that counter data is 24-bit, which cannot be stored in Modbus WORD area, lower 16-bit is stored into mCNT0_L, and the upper 8-bit is stored into mCNT0_H.

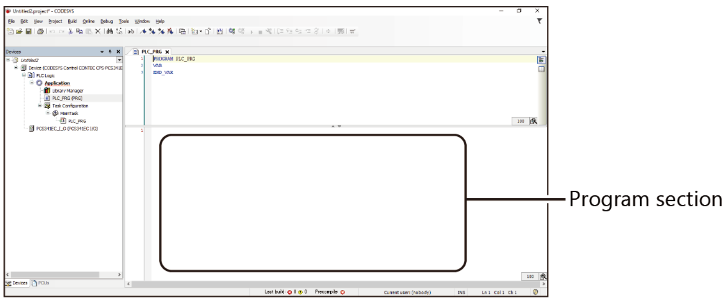


Variable	Mapping	Channel	Address	Type	Unit	Description
		do0-1	%QB20	BYTE		digital output bit 0-1
mDO0	✓	Bit0	%QX20.0	BOOL		digital output bit 0-1
mDO1	✓	Bit1	%QX20.1	BOOL		digital output bit 0-1
		Bit2	%QX20.2	BOOL		digital output bit 0-1
		Bit3	%QX20.3	BOOL		digital output bit 0-1
		[illegible]	[illegible]	[illegible]		[illegible]
		Bit5	[illegible]	BOOL		[illegible]
		Bit6	%QX20.6	BOOL		digital output bit 0-1
		Bit7	%QX20.7	BOOL		digital output bit 0-1
		di0-3	%IB20	BYTE		digital input bit 0-3
mDI0	✓	Bit0	%IX20.0	BOOL		digital input bit 0-3
mDI1	✓	Bit1	%IX20.1	BOOL		digital input bit 0-3
mDI2	✓	Bit2	%IX20.2	BOOL		digital input bit 0-3
mDI3	✓	Bit3	%IX20.3	BOOL		digital input bit 0-3
		[illegible]	%IX20.4	[illegible]		digital input bit 0-3
		Bit5	[illegible]	BOOL		[illegible]
		Bit6	%IX20.6	BOOL		digital input bit 0-3
		Bit7	%IX20.7	BOOL		digital input bit 0-3
mCNT0	✓	cnt0	%ID6	DWORD		counter channel 0
mCNT1	✓	cnt1	%ID7	DWORD		counter channel 1
mAI0	✓	ai0	%ID8	DWORD		analog input channel 0
[illegible]		[illegible]	%ID9	[illegible]		analog input channel 1
		[illegible]	[illegible]	[illegible]		[illegible]
		battery	[illegible]	BYTE		battery status

3. Create a program and build

1 Double click the [PLC_PRG (PRG)] icon on the device window to open the “ST editor window”.

2 Write the following source code in “Program” under “ST editor window”.



```
mDI0 := DI0;
mDI1 := DI1;
mDI2 := DI2;
mDI3 := DI3;
mAI0 := DWORD_TO_WORD(AI0);
mAI1 := DWORD_TO_WORD(AI1);
mCNT0_L := DWORD_TO_WORD(CNT0 AND 16#0000FFFF);
mCNT0_H := DWORD_TO_WORD(SHR(CNT0, 16) AND 16#0000FFFF);

mCNT1_L := DWORD_TO_WORD(CNT1 AND 16#0000FFFF);
mCNT1_H := DWORD_TO_WORD(SHR(CNT1, 16) AND 16#0000FFFF);
DO0 := mDO0;
DO1 := mDO1;
```

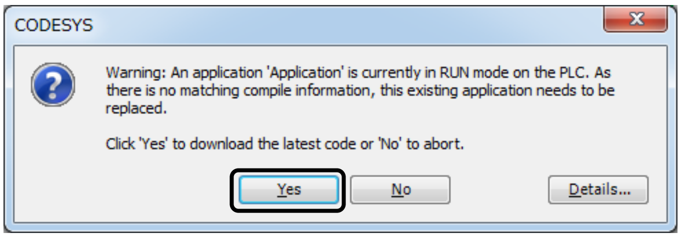
* On IEC program, copy the data of digital input, analog input, and counter input from I/O variables to Modbus variables.　For digital output, copy the data from Modbus variables to I/O variables.

4. Download and run program

Following steps 1-10 describe an example of outputting 1 into digital output bit 0.
Step 11 and 12 describe an example of inputting counter input 0 data.

1 Select the [Online] – [Login] on the CODESYS menu.

2 Click the [Yes] in the download confirmation dialog.


CODESYS

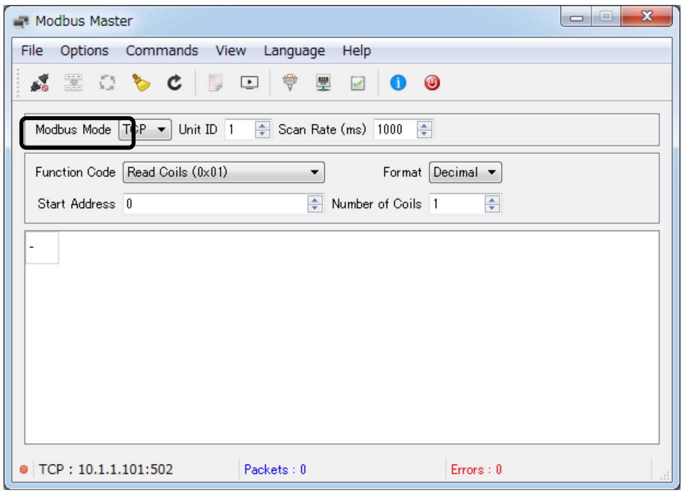
Warning: An application 'Application' is currently in RUN mode on the PLC. As there is no matching compile information, this existing application needs to be replaced.

Click 'Yes' to download the latest code or 'No' to abort.

3 Select the [Debug] – [Start] and the program operation will start

4 As described in the step 3 status, start QModMaster.

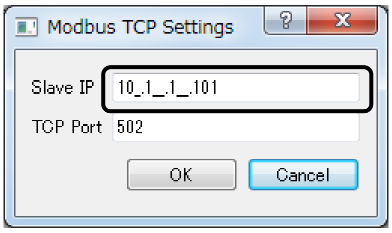
5 Choose the [TCP] in the [Modbus Mode].


Modbus Master

Modbus Mode: TCP
Unit ID: 1
Scan Rate (ms): 1000
Function Code: Read Coils (0x01)
Format: Decimal
Start Address: 0
Number of Coils: 1

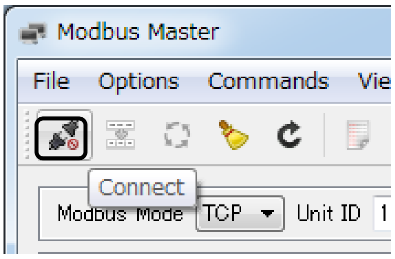
TCP : 10.1.1.101:502
Packets : 0
Errors : 0

6 From the [Options], select the “Modbus TCP” and set IP address “10.1.1.101” of slave device.

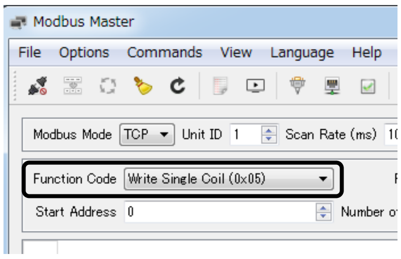

Modbus TCP Settings

Slave IP: 10_.1_.1_.101
TCP Port: 502

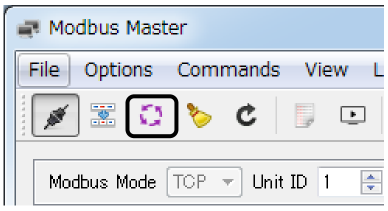
7 Click the [Connect] icon.



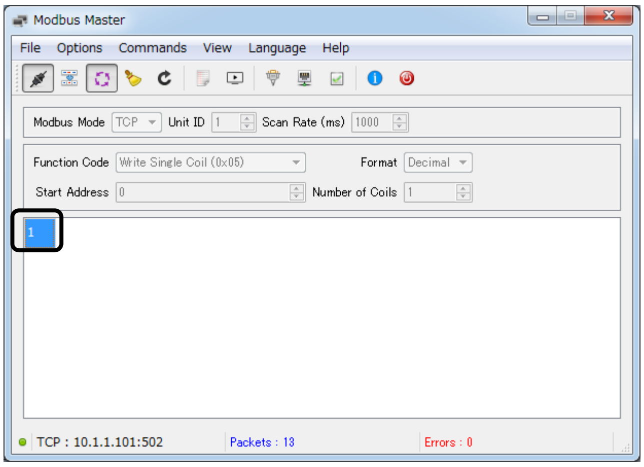
8 Choose the [Write Single Coil(0x05)] in “Function Code”.



9 Set "0" in the “Start Address" and click the [Scan] icon.



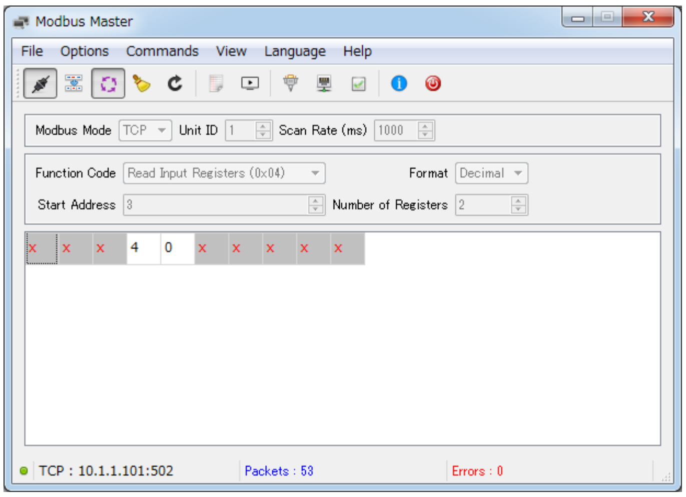
10 Double click on the part where "0" is shown and change it to "1".
You can see that DO0 LED of CONTEC CPS-PC341MB-ADSC1-9201 is on.



Modbus Mode TCP Unit ID 1 Scan Rate (ms) 1000
Function Code Write Single Coil (0x05) Format Decimal
Start Address 0 Number of Coils 1
1
TCP : 10.1.1.101:502 Packets : 13 Errors : 0

11 Choose the [Read Input Registers (0x04)] in “Function Code”.

12 Enter “3” into “Start Address”, and “2” into “Number of Registers”, then click the [Scan] icon. Lower 16-bit of the counter 0 is displayed on the left, and Upper 8-bit of the counter is displayed on the right.



Modbus Mode TCP Unit ID 1 Scan Rate (ms) 1000

Function Code Read Input Registers (0x04) Format Decimal

Start Address 3 Number of Registers 2

x	x	x	4	0	x	x	x	x	x

TCP : 10.1.1.101:502 Packets : 53 Errors : 0

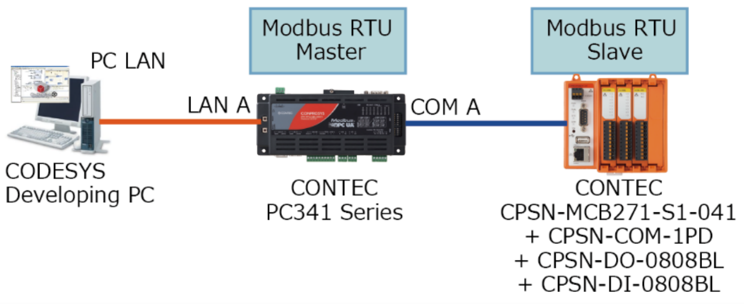
5.Use the controller as Modbus RTU Master

Model	Slave model
CPS-PC341MB-ADSC1-9201 CPS-PCS341MB-DS1-1201	Slave support devices of CONTEC's including CPSN-MCB271-S1-041 Slave support devices of other maker's

1. Device preparation

This section describes an example of using a controller as Modbus RTU Master to input/output data with Modbus RTU Slave.

Here, CONTEC CPS-PC341MB-ADSC1-9201 is used as Modbus RTU Master, and CONTEC CPSN-MCB271-S1-041 is used as Modbus RTU Slave.



1 Connect LAN A of the controller to LAN port of CODESYS developing PC.

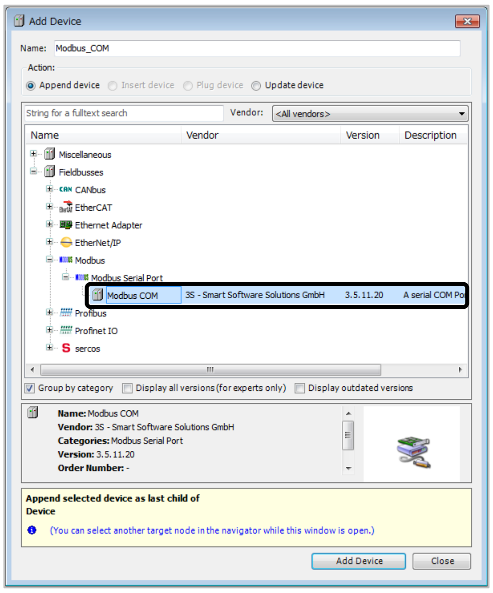
2 Connect COM A of the controller to COM port of CPSN-COM-1PD set in the CPU unit.
*Serial communication parameters of the CPSN-MCB271-S1-041 and the controller should be the same.

2. Add slave device

1 Create a New CODESYS project.
Choose the name of connected Modbus model for device, and ST for IEC program language.
*See “**Create a New Project (page 57)**” for how to create the project.

2 Right-click on the controller [Device (CODESYS Control CONTEC CPS-PCXXXXXXXXXX)] on the device window and select the [Add device].

3 In the dialog box, select [Fieldbuses – Modbus – Modbus Serial Port – Modbus COM] and click the [Add device] button.


Add Device

Name: Modbus_COM

Action: [✓ Append device] Insert device Plug device Update device

String for a fulltext search Vendor: <All vendors>

Name	Vendor	Version	Description
Miscellaneous			
Fieldbusses			
CANbus			
EtherCAT			
Ethernet Adapter			
EtherNet/IP			
Modbus			
Modbus Serial Port			
Modbus COM	3S - Smart Software Solutions GmbH	3.5.11.20	A serial COM Po
Profibus			
Profinet IO			
sercos			

☑ Group by category ☐ Display all versions (for experts only) ☐ Display outdated versions

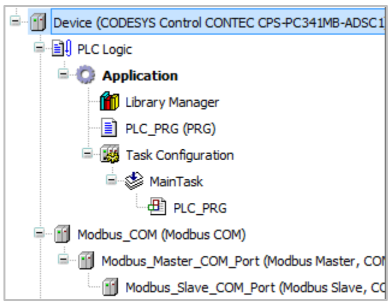
Name: Modbus COM
Vendor: 3S - Smart Software Solutions GmbH
Categories: Modbus Serial Port
Version: 3.5.11.20
Order Number: -

Append selected device as last child of
Device
(You can select another target node in the navigator while this window is open.)

4 Right-click on the [Modbus_COM] on the device window and click the [Add device].

5 In the [Add device] dialog box, add [Fieldbuses - Modbus - Modbus Serial Master – Modbus Master, COM Port].

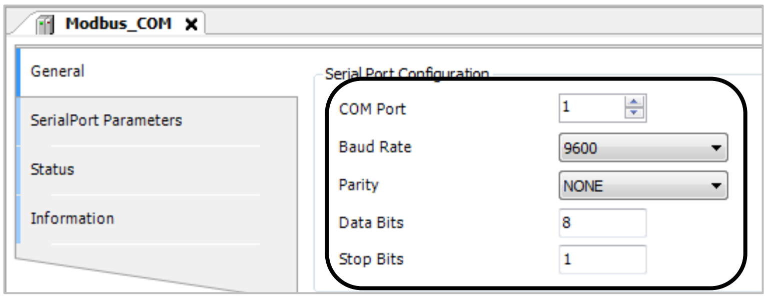
6 Again, right-click on the [Modbus_Master_COM_Port] and select the [Add device]. Then add [Fieldbuses - Modbus - Modbus Serial Slave - Modbus Slave, COM Port].


```
Device (CODESYS Control CONTEC CPS-PC341MB-ADSC1
  PLC Logic
    Application
      Library Manager
      PLC_PRG (PRG)
      Task Configuration
        MainTask
          PLC_PRG
  Modbus_COM (Modbus COM)
    Modbus_Master_COM_Port (Modbus Master, CO
      Modbus_Slave_COM_Port (Modbus Slave, C
```

7 On the Device window, double click the [Modbus_COM] icon to open the Device Configuration window.

8 Open the [General] tab.

9 Set the parameter in order to communication by Modbus RTU.

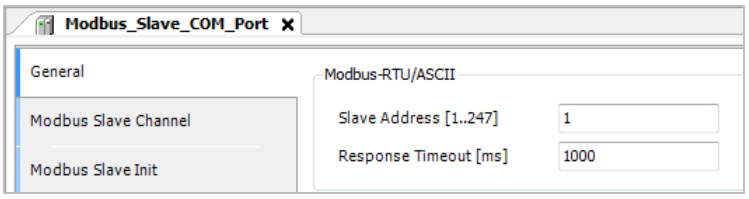


COM Port	1
Baud Rate	9600
Parity	NONE
Data Bits	8
Stop Bits	1

10 On the Device window, double click the [Modbus_Slave_COM_Port] icon to open Device Configuration window.

11 Open the [General] tab.

12 Enter the slave device address into the [Slave Address].



Slave Address [1..247]	1
Response Timeout [ms]	1000

3. Modbus RTU Master setting

This section describes how to create samples of 2-bit digital output and 8-bit digital input.
Here, Modbus register of slave device is registered as below.

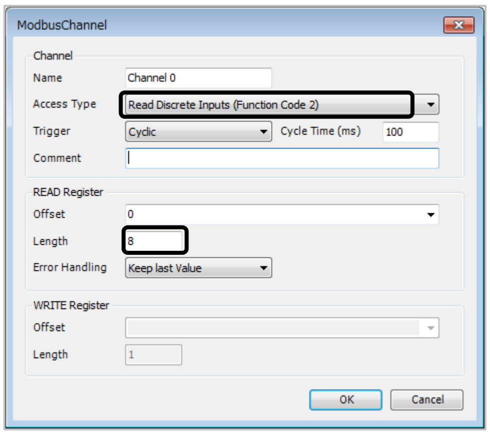
Name/Function code	Address	Data
Read Discrete Inputs (Code 2)	0 to 7	Digital input bit 0 to 7 of CPSN-DI-0808BL
Write Single Coil (Code 5)	0	Digital output bit 0 of CPSN-DO-0808BL
	1	Digital output bit 1 of CPSN-DO-0808BL

* The samples are created with Write Single Coil (code 5) for 2-bit digital output, and Read Discrete Inputs (code 2) for 8-bit digital input.

1 On the Devices window, double click the [Modbus_Slave_COM_Port] icon to open Device Configuration window.

2 Open the [Modbus Slave Channel] tab.

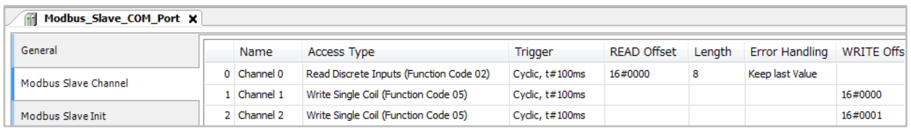
3 Click the [Add channel] button. Select [Read Discrete Inputs (function code 2)] from Access type, and enter “8” into Length filed in the Read register, then click the [OK].



Name: Channel 0
Access Type: Read Discrete Inputs (Function Code 2)
Trigger: Cyclic
Cycle Time (ms): 100
Comment:
Offset: 0
Length: 8
Error Handling: Keep last Value
Offset:
Length: 1

4 Click the [Add channel] button. Select [Write Single Coil (function code 5)] from Access type, and enter “0” into Write register offset, then click the [OK].

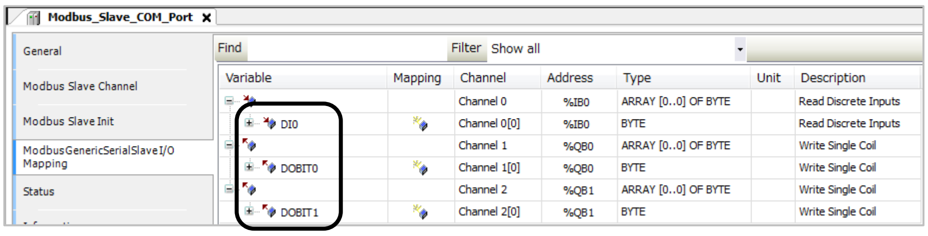
5 Click the [Add channel] button. Select [Write Single Coil (function code 5)] from Access type again, and enter “1” into Write register offset, then click the [OK].



	Name	Access Type	Trigger	READ Offset	Length	Error Handling	WRITE Offs
0	Channel 0	Read Discrete Inputs (Function Code 02)	Cyclic, t#100ms	16#0000	8	Keep last Value	
1	Channel 1	Write Single Coil (Function Code 05)	Cyclic, t#100ms				16#0000
2	Channel 2	Write Single Coil (Function Code 05)	Cyclic, t#100ms				16#0001

6 Click the [ModbusGenericSerialSlave I/O Mapping].

7 Allocate "DI0","DOBIT0","DOBIT1" to Channel0 [0], Channel1 [0], Channel2 [0] respectively.

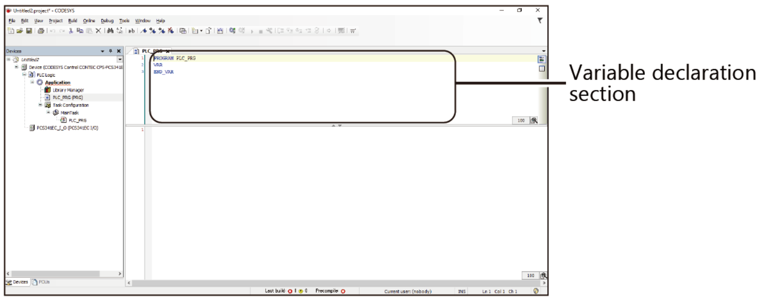


Variable	Mapping	Channel	Address	Type	Unit	Description
		Channel 0	%IB0	ARRAY [0..0] OF BYTE		Read Discrete Inputs
DI0		Channel 0[0]	%IB0	BYTE		Read Discrete Inputs
		Channel 1	%QB0	ARRAY [0..0] OF BYTE		Write Single Coil
DOBIT0		Channel 1[0]	%QB0	BYTE		Write Single Coil
		Channel 2	%QB1	ARRAY [0..0] OF BYTE		Write Single Coil
DOBIT1		Channel 2[0]	%QB1	BYTE		Write Single Coil

4. Create and build a program

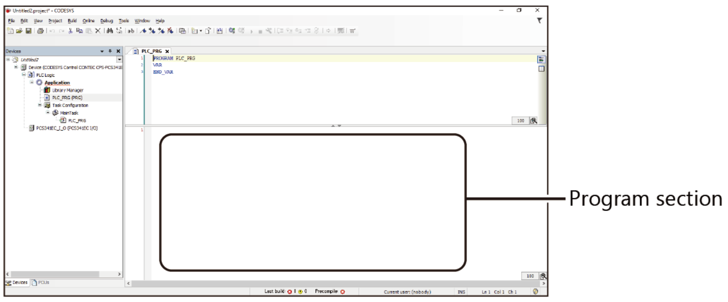
1 Double click the [PLC_PRG (PRG)] icon on the device window to open the “ST editor window”.

2 When ST editor window appears, write the following source code between VAR and END_VAR in the “Variable declaration”.



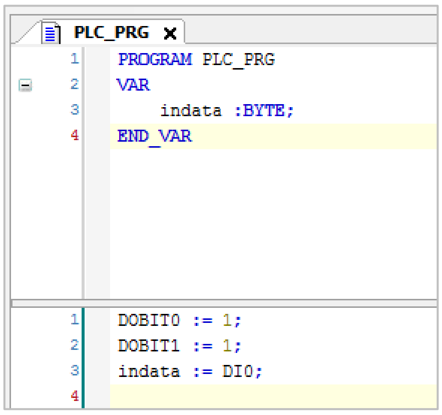
```
indata :BYTE;
```

3 Write the following source code in “Program” under “ST editor window”.



```
DOBIT0 := 1;
DOBIT1 := 1;
indata := DI0;
```

This program outputs ON signal to digital output bit 0 and 1, and inputs signal from digital input bit 0.



```
PROGRAM PLC_PRG
VAR
    indata :BYTE;
END_VAR
```

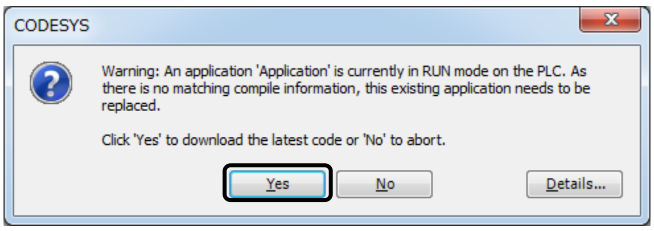
```
DOBIT0 := 1;
DOBIT1 := 1;
indata := DI0;
```

4 Now, see whether you can execute [Rebuild] in CODESYS [Build] menu to confirm the process succeeds properly.

5. Download and run program

1 Select the [Online] – [Login] on the CODESYS menu.

2 Click the [Yes] in the download confirmation dialog.



Warning: An application 'Application' is currently in RUN mode on the PLC. As there is no matching compile information, this existing application needs to be replaced.

Click 'Yes' to download the latest code or 'No' to abort.

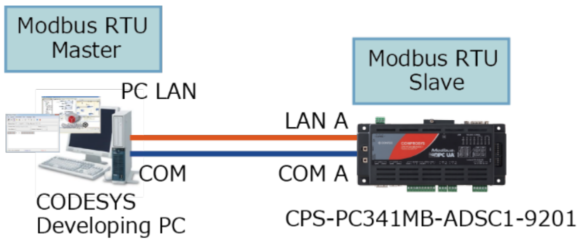
3 Select the [Debug] - [Start] and the program operation will start.
You can see that LED of DO Bit0, 1 on the front panel of CPSN-DO-0808BL is on.

6.Use the controller as Modbus RTU Slave

Model	Slave model
CPS-PC341MB-ADSC1-9201 CPS-PCS341MB-DS1-1201	---

1. Device preparation

This section describes an example of using a controller as Modbus RTU Slave to input/output data with Modbus RTU Master.
Here, the PC with CODESYS developing environment is used as Modbus RTU Slave, and CONTEC CPS-PC341MB-ADSC1-9201 is used as Modbus RTU Slave for CODESYS programming.
QModMaster is used as Modbus RTU Master software.



1 Install Modbus RTU Master software QModMaster into the PC.
QModMaster can be downloaded free through the URL below.

Download http://sourceforge.net/projects/qmodmaster/

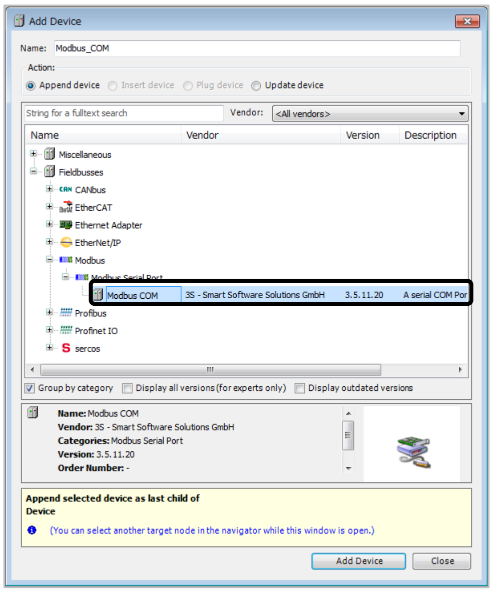
2 Connect COM A of CPS-PC341MB-ADSC1-9201 and COM port of PC with CODESYS developing environment via Serial cable.

2. Slave setting

1 Create a new CODESYS project.
Choose ST for IEC program language.
*See "**Create a New Project (page 57)**" for how to create the project.

2 Right-click on the controller [Device (CODESYS Control CONTEC CPS-PCXXXXXXXXXX)] on the device window, and select the [Add device].

3 In the [Add device] dialog box, select [Fieldbuses – Modbus – Modbus Serial Port – Modbus COM] and click the [Add device] button.
* This adds Modbus_COM Adapter.


Add Device

Name: Modbus_COM

Action: [✓ Append device] Insert device Plug device Update device

String for a fulltext search Vendor: <All vendors>

Name	Vendor	Version	Description
Miscellaneous			
Fieldbusses			
CANbus			
EtherCAT			
Ethernet Adapter			
EtherNet/IP			
Modbus			
Modbus Serial Port			
Modbus COM	3S - Smart Software Solutions GmbH	3.5.11.20	A serial COM Por
Profibus			
Profinet IO			
sercos			

[✓ Group by category] Display all versions (for experts only) Display outdated versions

Name: Modbus COM
Vendor: 3S - Smart Software Solutions GmbH
Categories: Modbus Serial Port
Version: 3.5.11.20
Order Number: -

Append selected device as last child of
Device
(You can select another target node in the navigator while this window is open.)

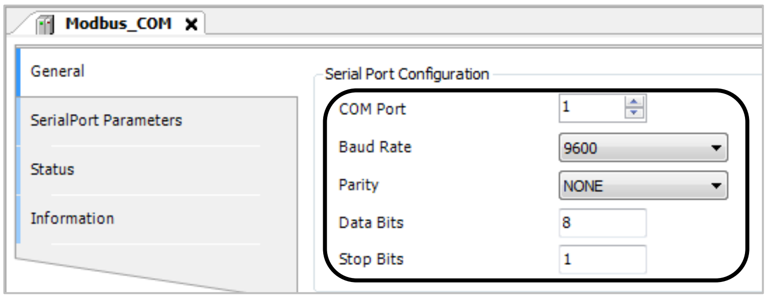
4 Right-click on the [Modbus_COM] and select the [Add device].

5 In the [Add device] dialog box, select the [Fieldbusses - Modbus - Modbus Serial Device – Modbus Serial Device].
* This adds Modbus Serial Slave device under Modbus_COM Adapter device.

6 On the Devices window, double click the [Modbus_COM] icon to open Device Configuration window.

7 Open the [General] tab.

8 Set the parameter in order to communication by Modbus RTU.

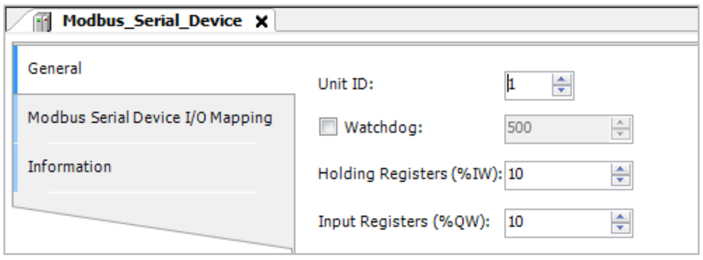


COM Port: 1
Baud Rate: 9600
Parity: NONE
Data Bits: 8
Stop Bits: 1

9 On the Devices window, double click the [Modbus_Serial_Device] icon to open Device Configuration window.

10 Open the [General] tab.

11 Enter the slave device address into the [Unit ID].



Unit ID: 1
Watchdog: 500
Holding Registers (%IW): 10
Input Registers (%QW): 10

12 On the Devices window, double click the [PC341MB_I_O] icon to open Device Configuration window.

13 Click the [Internal I/O Mapping].

14 Define variables as follows:

Variable (Input value)		Channel	Address
		do0-1	
	DO0	Bit0	%QX20.0
	DO1	Bit1	%QX20.1
		di0-3	
	DI0	Bit0	%IX20.0
	DI1	Bit1	%IX20.1
	DI2	Bit2	%IX20.2
	DI3	Bit3	%IX20.3
CNT0		cnt0	%ID6
CNT1		cnt1	%ID7
AI0		ai0	%ID8
AI1		ai1	%ID9

15 On the Devices window, double click the [Modbus_Serial_Device] icon to open Device Configuration window.

16 Click the [Modbus Serial Device I/O Mapping].

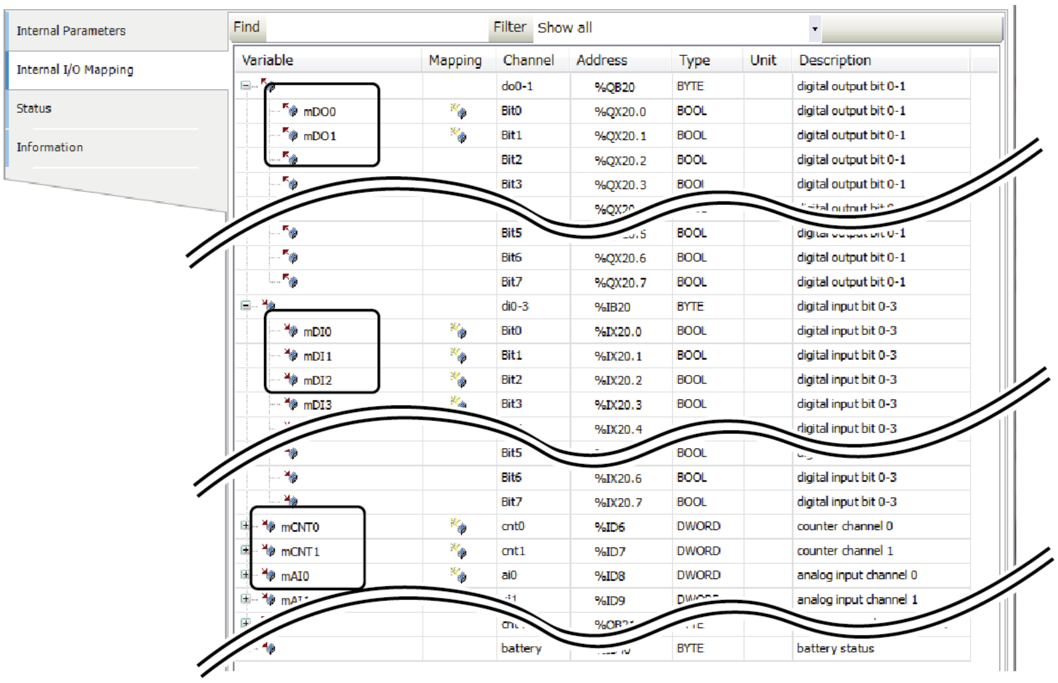
17 Define variables as follows:

Variable (Input value)	Channel	Address
	Input	%IW1
	Input[0]	%IW1
mDO0	Bit0	%IX2.0
mDO1	Bit1	%IX2.1
	:	
	Bit15	%IX3.7
	Input[1]	%IW2
	:	
	Input[9]	%IW10
	Output	%QW1
	Output[0]	%QW1
mDI0	Bit0	%QX0.0
mDI1	Bit1	%QX0.1
mDI2	Bit2	%QX0.2
mDI3	Bit3	%QX0.3
	:	
	Bit15	%QX1.7
mAI0	Output[1]	%QW2
MAI1	Output[2]	%QW3
mCNT0_L	Output[3]	%QW4
mCNT0_H	Output[4]	%QW5
mCNT1_L	Output[5]	%QW6
mCNT1_H	Output[6]	%QW7
	:	
	Output[9]	%QW10

* Now, I/O Mapping is set, and I/O Mapping for Modbus Slave is defined.
Data communication with Modbus Master can be carried out by defining variables into I/O Mapping of Modbus Slave.

* You can see that variables for output in Inputs Mapping area, and variables for input in Outputs Mapping area are defined. This means Master inputs for Slave in Inputs Mapping area, and Slave outputs for Master in Outputs Mapping area.

* Also note that counter data is 24-bit, which cannot be stored in Modbus WORD area, lower 16-bit is stored into mCNT0_L, and the upper 8-bit is stored into mCNT0_H.

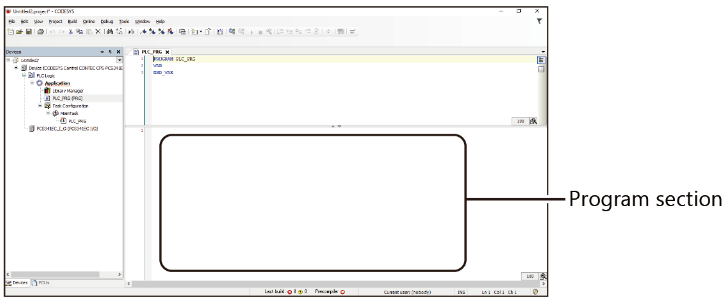


Variable	Mapping	Channel	Address	Type	Unit	Description
		do0-1	%QB20	BYTE		digital output bit 0-1
mDO0	✓	Bit0	%QX20.0	BOOL		digital output bit 0-1
mDO1	✓	Bit1	%QX20.1	BOOL		digital output bit 0-1
		Bit2	%QX20.2	BOOL		digital output bit 0-1
		Bit3	%QX20.3	BOOl		digital output bit 0-1
		[illegible]	[illegible]	[illegible]		[illegible]
		Bit5	[illegible]	BOOL		[illegible]
		Bit6	%QX20.6	BOOL		digital output bit 0-1
		Bit7	%QX20.7	BOOL		digital output bit 0-1
		di0-3	%IB20	BYTE		digital input bit 0-3
mDI0	✓	Bit0	%IX20.0	BOOL		digital input bit 0-3
mDI1	✓	Bit1	%IX20.1	BOOL		digital input bit 0-3
mDI2	✓	Bit2	%IX20.2	BOOL		digital input bit 0-3
mDI3	✓	Bit3	%IX20.3	BOOL		digital input bit 0-3
		[illegible]	%IX20.4	[illegible]		digital input bit 0-3
		Bit5	[illegible]	BOOL		[illegible]
		Bit6	%IX20.6	BOOL		digital input bit 0-3
		Bit7	%IX20.7	BOOL		digital input bit 0-3
mCNT0	✓	cnt0	%ID6	DWORD		counter channel 0
mCNT1	✓	cnt1	%ID7	DWORD		counter channel 1
mAI0	✓	ai0	%ID8	DWORD		analog input channel 0
[illegible]	[illegible]	[illegible]	%ID9	[illegible]		analog input channel 1
		[illegible]	[illegible]	[illegible]		[illegible]
		battery	[illegible]	BYTE		battery status

3. Create a program and build

1 Double click the [PLC_PRG (PRG)] icon on the device window to open the “ST editor window”.

2 Write the following source code in “Program” under “ST editor window”.



```
mDI0 := DI0;
mDI1 := DI1;
mDI2 := DI2;
mDI3 := DI3;
mAI0 := DWORD_TO_WORD(AI0);
mAI1 := DWORD_TO_WORD(AI1);
mCNT0_L := DWORD_TO_WORD(CNT0 AND 16#0000FFFF);
mCNT0_H := DWORD_TO_WORD(SHR(CNT0, 16) AND 16#0000FFFF);

mCNT1_L := DWORD_TO_WORD(CNT1 AND 16#0000FFFF);
mCNT1_H := DWORD_TO_WORD(SHR(CNT1, 16) AND 16#0000FFFF);
DO0 := mDO0;
DO1 := mDO1;
```

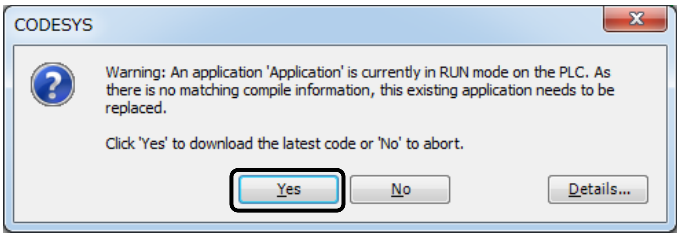
* On IEC program, copy the data of digital input, analog input, and counter input from I/O variables to Modbus variables. For digital output, copy the data from Modbus variables to I/O variables.

4. Download and run program

Following steps 1-10 describe an example of outputting 1 into digital output bit 0.
Step 11 and 12 describe an example of inputting counter input 0 data.

1 Select the [Online] – [Login] on the CODESYS menu.

2 Click the [Yes] in the download confirmation dialog.


CODESYS

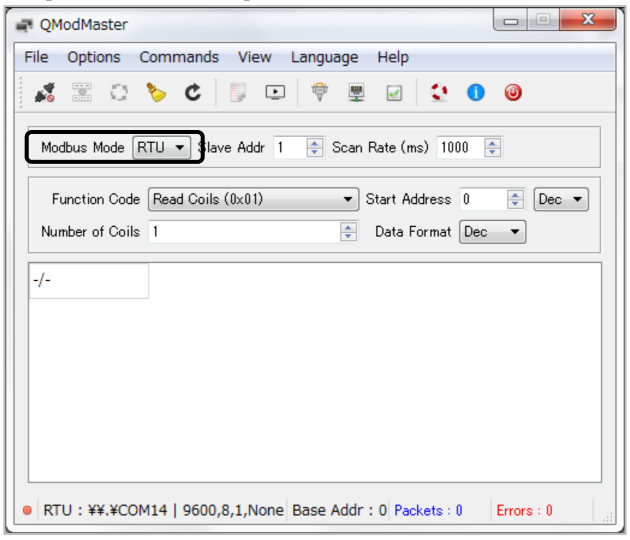
Warning: An application 'Application' is currently in RUN mode on the PLC. As there is no matching compile information, this existing application needs to be replaced.

Click 'Yes' to download the latest code or 'No' to abort.

3 Select the [Debug] – [Start] and the program operation will start

4 As described in the step 3 status, start QModMaster.

5 Choose the [RTU] in the [Modbus Mode].



Modbus Mode RTU | Slave Addr 1 | Scan Rate (ms) 1000

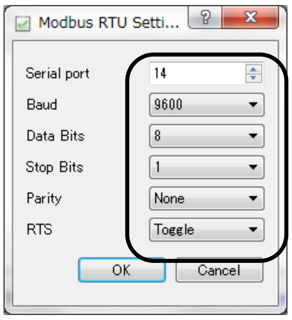
Function Code Read Coils (0x01) | Start Address 0 Dec

Number of Coils 1 | Data Format Dec

-/-

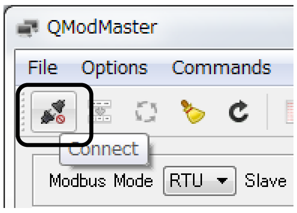
RTU : ¥¥.¥COM14 | 9600,8,1,None Base Addr : 0 Packets : 0 Errors : 0

6 From the [Options], select the "Modbus RTU" and set the parameter in order to communication by Modbus RTU.


Modbus RTU Setti...

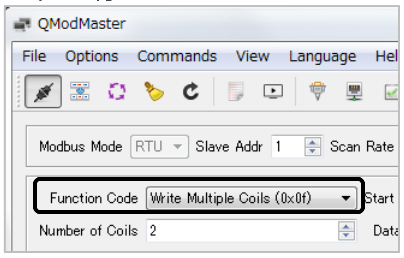
Serial port	14
Baud	9600
Data Bits	8
Stop Bits	1
Parity	None
RTS	Toggle

7 Click the [Connect] icon.



Modbus Mode RTU

8 Choose the [Write Multiple Coils(0x0f)] in “Function Code”.

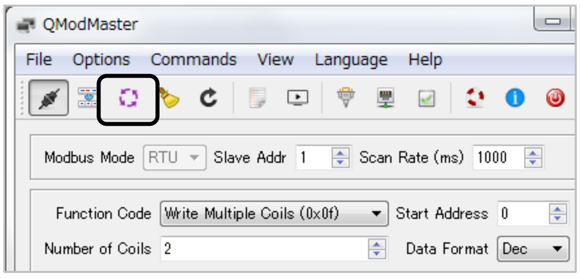


Modbus Mode RTU Slave Addr 1

Function Code Write Multiple Coils (0x0f)

Number of Coils 2

9 Set "0" into the "Start Address", and "2" into "Number of Coils", then click the [Scan] icon.

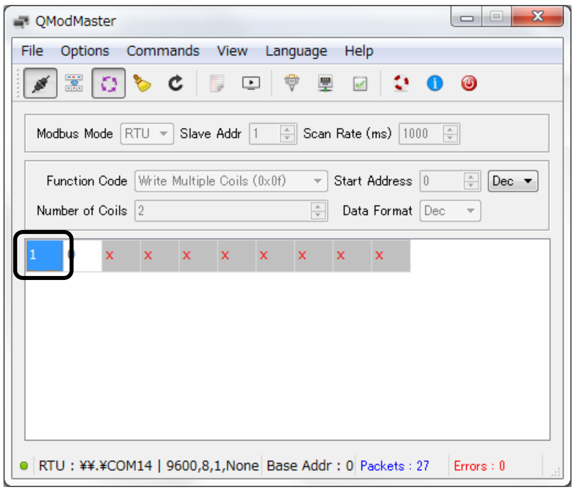


Modbus Mode RTU Slave Addr 1 Scan Rate (ms) 1000

Function Code Write Multiple Coils (0x0f) Start Address 0

Number of Coils 2 Data Format Dec

10 Double click on the part where "0" is shown and change it to "1".
You can see that DO0 LED of CONTEC CPS-PC341MB-ADSC1-9201 is on.



Modbus Mode RTU Slave Addr 1 Scan Rate (ms) 1000

Function Code Write Multiple Coils (0x0f) Start Address 0 Dec

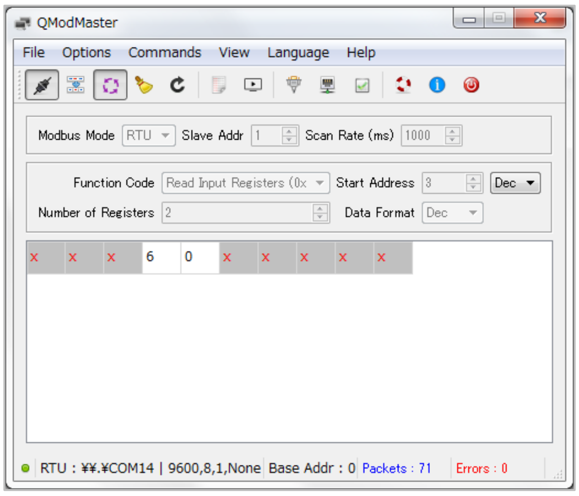
Number of Coils 2 Data Format Dec

1

RTU : ¥¥.¥COM14 | 9600,8,1,None Base Addr : 0 Packets : 27 Errors : 0

11 Choose the [Read Input Registers (0x04)] in “Function Code”.

12 Enter “3” into “Start Address”, and “2” into “Number of Registers”, then click the [Scan] icon. Lower 16-bit of the counter 0 is displayed on the left, and Upper 8-bit of the counter is displayed on the right.



Modbus Mode RTU Slave Addr 1 Scan Rate (ms) 1000

Function Code Read Input Registers (0x Start Address 3 Dec

Number of Registers 2 Data Format Dec

x x x 6 0 x x x x x

RTU : ¥¥.¥COM14 | 9600,8,1,None Base Addr : 0 Packets : 71 Errors : 0

7.OPC UA Server setting

Add the OPC-UA Server functions to CODESYS.

1. Device preparation

1 Open the project created in the "**Basic programming procedure (page 57)**" on CODESYS.

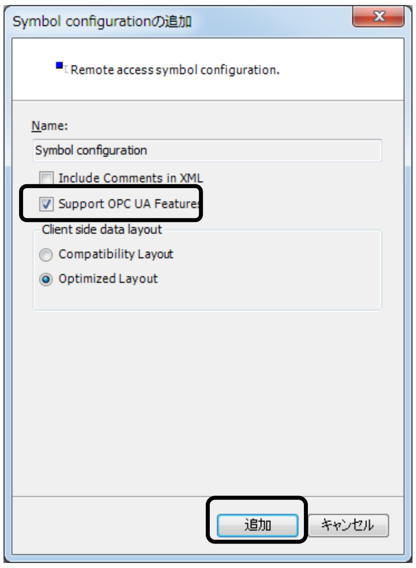
2 In this section, use UaExpert by Unified Automation as OPC-UA Client software.
UaExpert can be downloaded free through URL below. (User registration is necessary)

Download https://www.unified-automation.com/downloads/opc-ua-clients.html

2. CODESYS setting

1 On the Devices window, right-click on the [Application], and select [Add object] - [Symbol configuration].

2 In the "Add Symbol configuration" dialog box, check off in a box next to [Support OPC UA Features], and click the [Add] button.


Symbol configurationの追加

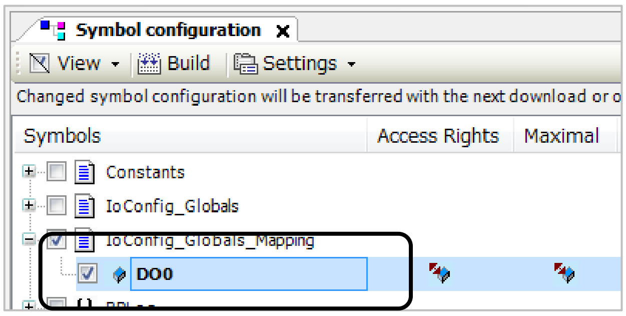
Remote access symbol configuration.

Name: Symbol configuration

[ ] Include Comments in XML
[✓] Support OPC UA Features

Client side data layout
( ) Compatibility Layout
(●) Optimized Layout

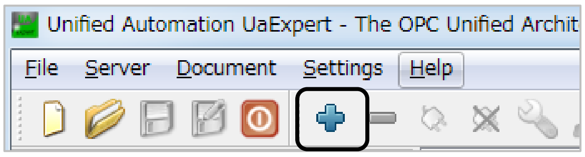
3 When “Symbol configuration” window appears, select variables to be used for OPC-UA Server from the list.
In this example, check off the box of [DO0] under the [IoConfig_Global_Mapping].



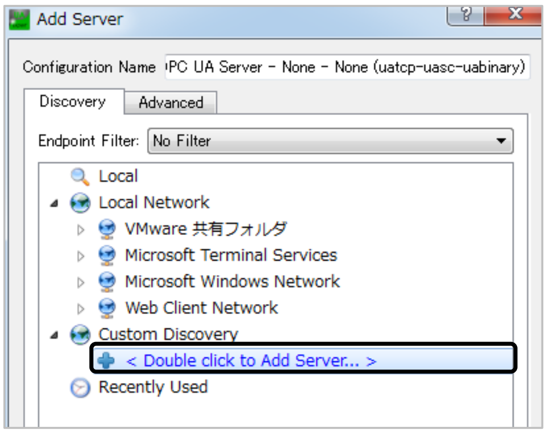
4 Build the program to see whether errors don’t occur, then log in to the controller to start the operation.

3. OPC-UA Client setting

1 Start UaExpert of OPC-UA Client software and click the [+] icon to add server.



2 Double click the [Double click to Add Server...].


Add Server

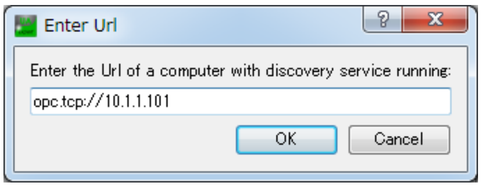
Configuration Name: PC UA Server - None - None (uatcp-uasc-uabinary)

Discovery | Advanced

Endpoint Filter: No Filter

- Local
- Local Network
  - VMware 共有フォルダ
  - Microsoft Terminal Services
  - Microsoft Windows Network
  - Web Client Network
- Custom Discovery
  - < Double click to Add Server... >
- Recently Used

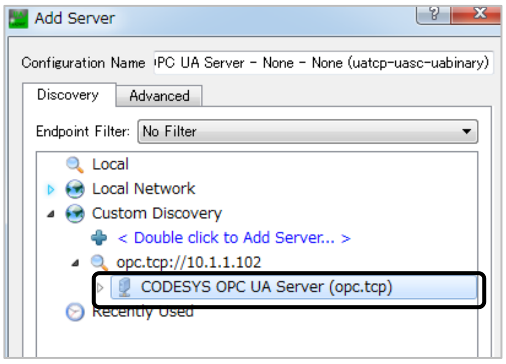
3 Enter IP address of the controller "opc.tcp://10.1.1.101" into URL.


Enter Url

Enter the Url of a computer with discovery service running:

opc.tcp://10.1.1.101

4 When the tree created, select the [CODESYS OPC UA Server] and click the [OK].


Add Server

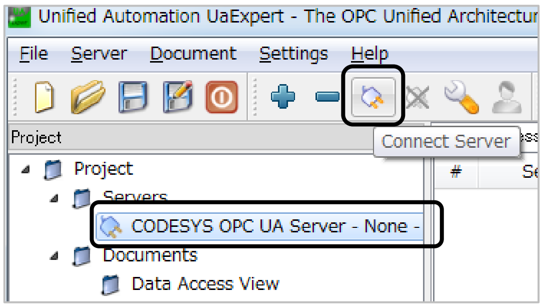
Configuration Name: PC UA Server - None - None (uatcp-uasc-uabinary)

Discovery | Advanced

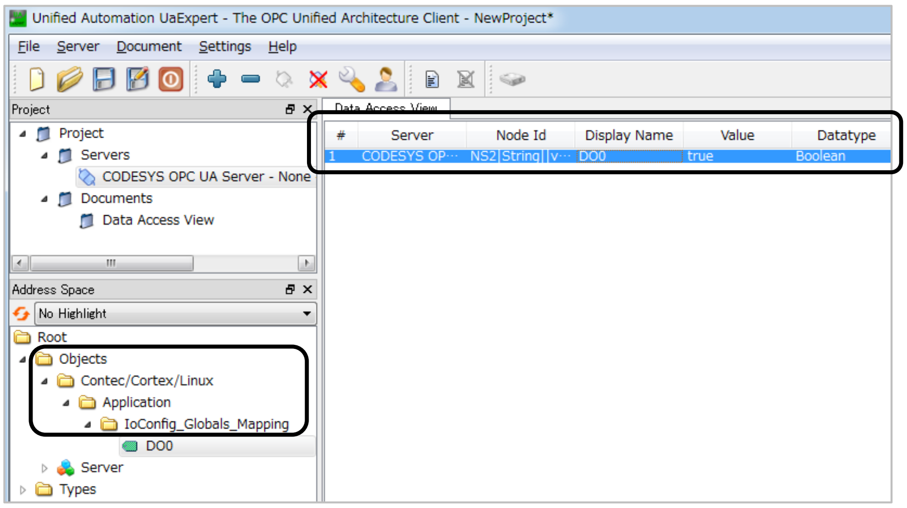
Endpoint Filter: No Filter

- Local
- Local Network
- Custom Discovery
  - < Double click to Add Server... >
  - opc.tcp://10.1.1.102
    - CODESYS OPC UA Server (opc.tcp)
- Recently Used

5 Choose the server and click the Connect Server icon.


Unified Automation UaExpert - The OPC Unified Architectur
File Server Document Settings Help
Connect Server
Project
Project
Servers
CODESYS OPC UA Server - None -
Documents
Data Access View
#
Se

6 Variables registered in CODESYS OPC Configurator is listed under [Root¥Objects¥Contec/Cortex/Linux¥Application¥PLC_PRG¥] in the [Address Space] tree hierarchy.
Drag variables you wish to monitor (DO0 -in this example) into the [Data Access View] window to check the status of the variables.

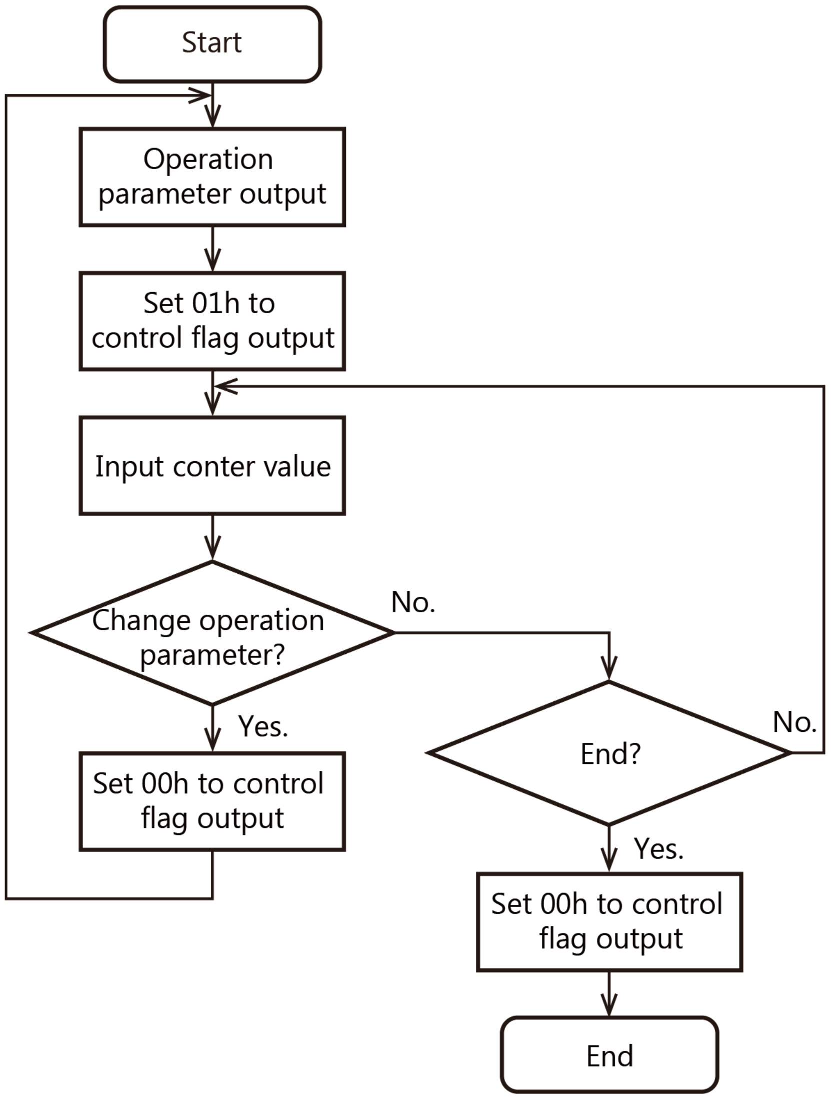


#	Server	Node Id	Display Name	Value	Datatype
1	CODESYS OP···	NS2|String||v···	DO0	true	Boolean

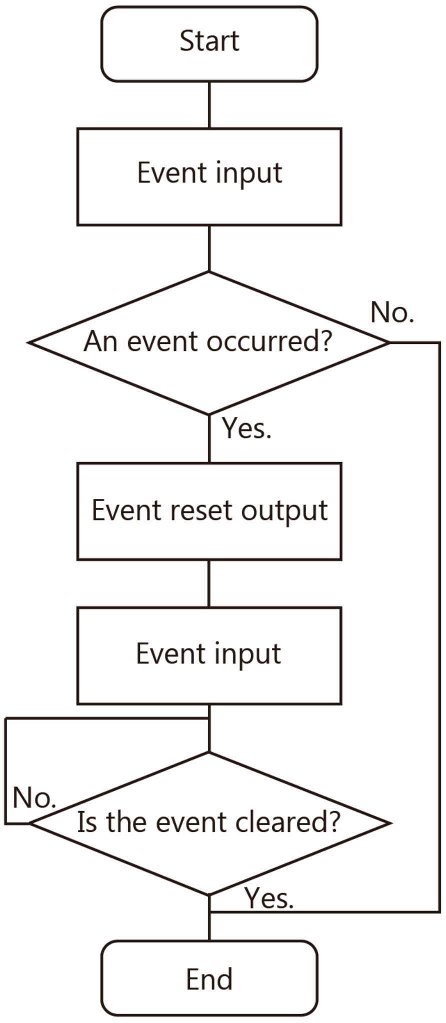
8.Counter Input

To perform the counter input, connect the I/O module CPS-CNT-3202I to CPU configurable controller. After completing the programming preparation described in the "**Basic programming procedure (page 57)**", Follow steps in the diagrams below to perform programming.

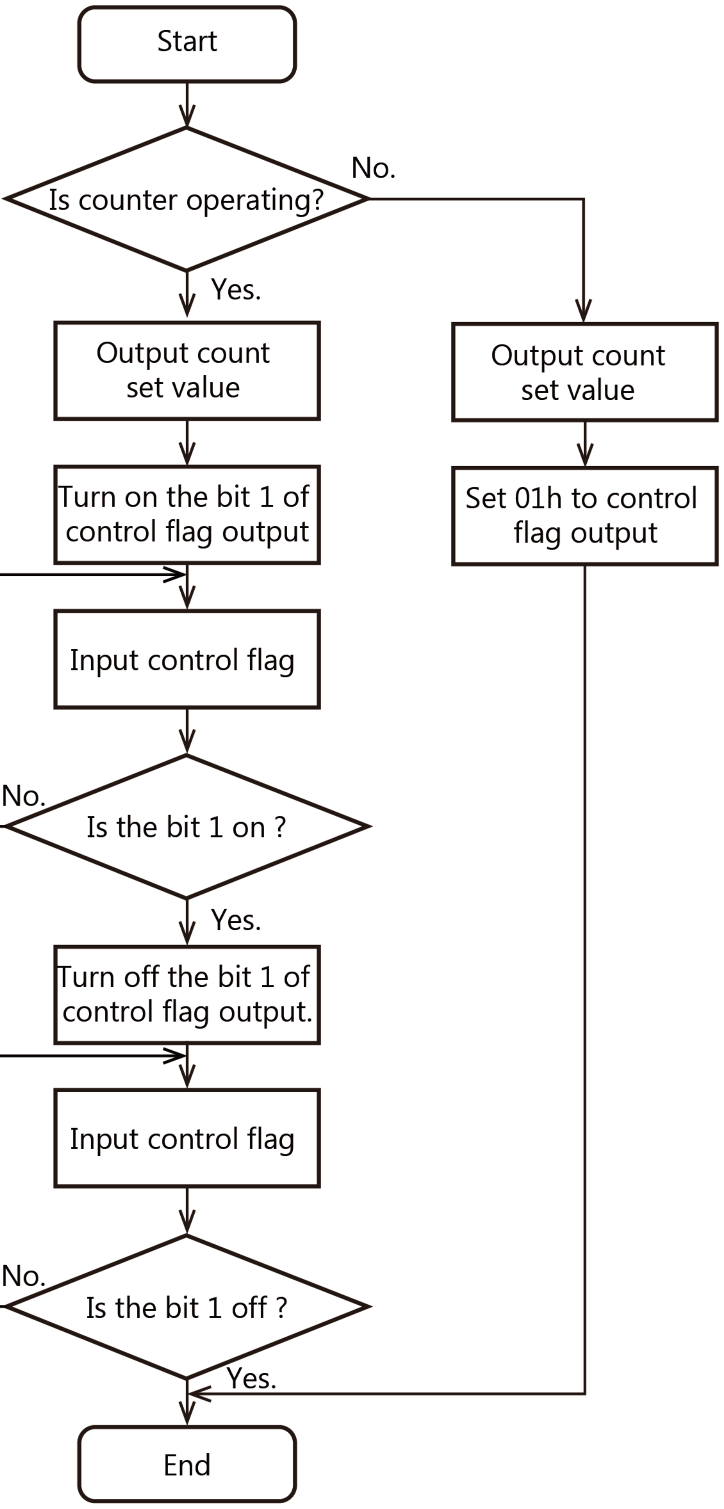
1. Basic procedure flow


Start
Operation parameter output
Set 01h to control flag output
Input conter value
Change operation parameter?
No.
Yes.
Set 00h to control flag output
End?
No.
Yes.
Set 00h to control flag output
End

2. Event procedure flow



3. Count value setting operation flow



4. Sample program

Sample programs listed in the previous pages are offered.

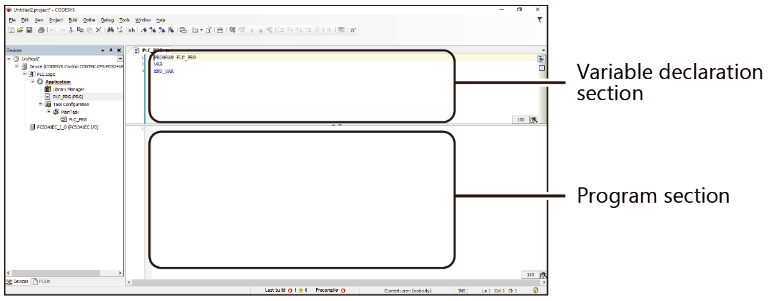
Program title	Function
Sample 1	To set CPS-CNT-3202I and perform the counter input.
Sample 2	When an event in which the value of counter input coincides with Compare value output occurs.
Sample 3	To change the count value during an operation.

Here are sample programming steps.

1 Double click the [PLC_PRG (PRG)] icon on the device window to open the "ST editor window".

2 When "ST editor window" appears, write the following source code between VAR and END_VAR in the "Variable declaration".

3 Write the following source code in "Program" under "ST editor window".


```
PROGRAM PLC_PRG
VAR
END_VAR
```

◆ Sample 1

To set CPS-CNT-3202I and perform the counter input.

Variable declaration

```
uiState: UINT := 0;
dwCounter: DWORD;
```

Program

```
CASE uiState OF
        0:
                CNT0_OutCtrl := 0;
                CNT0_EventReset := 0;
                uiState := 1;
        1:
                CNT0_ZMode := 0;
                CNT0_ZLogic := 0;
                CNT0_Dir := 0;
                CNT0_Phase := 1;
                CNT0_Multi := 0;
                CNT0_Clear := 1;

                CNT0_DiFilter := 0;
                CNT0_Pulse := 0;
                CNT0_CntSet := 0;
                CNT0_CntMatch := 0;
                uiState := 2;
        2:
                CNT0_OutCtrl := 16#01;
                uiState := 3;
        3:
                dwCounter := CNT0_CntVal;
                IF DWORD_TO_DINT(dwCounter) >= 100 THEN
                        uiState := 4;
                END_IF
        4:
                CNT0_OutCtrl := 0;
                uiState := 5;
        5:
END_CASE
```

◆ Sample 2

When an event in which the value of counter input coincides with Compare value output occurs.

Variable declaration

```
uiState: UINT := 0;
dwCounter: DWORD;
byEvent: BYTE;
```

Program

```
CASE uiState OF
        0:
                CNT0_OutCtrl := 0;
                CNT0_EventReset := 0;
                uiState := 1;
        1:
                CNT0_ZMode := 0;
                CNT0_ZLogic := 0;
                CNT0_Dir := 0;
                CNT0_Phase := 1;
                CNT0_Multi := 0;
                CNT0_Clear := 1;
                CNT0_DiFilter := 0;
                CNT0_Pulse := 0;
                CNT0_CntSet := 0;
                CNT0_CntMatch := 100;
                uiState := 2;
        2:
                CNT0_OutCtrl := 16#01;
                uiState := 3;
        3:
                dwCounter := CNT0_CntVal;
                byEvent := CNT0_Event;

                IF (byEvent AND 16#01) = 16#01 THEN
                        uiState := 4;
                END_IF
        4:
                CNT0_EventReset := 16#01;
                uiState := 5;
        5:
                byEvent := CNT0_Event;
                IF (byEvent AND 16#01) = 16#00 THEN
                        uiState := 6;
                END_IF
        6:
                CNT0_OutCtrl := 0;
                uiState := 7;
        7:
END_CASE
```

◆ Sample 3

To change the count value during an operation.

Variable declaration

```
uiState: UINT := 0;
dwCounter: DWORD;
byEvent: BYTE;
```

Program

```
CASE uiState OF
        0:
                CNT0_OutCtrl := 0;
                CNT0_EventReset := 0;
                uiState := 1;
        1:
                CNT0_ZMode := 0;
                CNT0_ZLogic := 0;
                CNT0_Dir := 0;
                CNT0_Phase := 1;
                CNT0_Multi := 0;
                CNT0_Clear := 1;
                CNT0_DiFilter := 0;
                CNT0_Pulse := 0;
                CNT0_CntSet := 100;
                CNT0_CntMatch := 0;
                uiState := 2;
        2:
                CNT0_OutCtrl := 16#01;
                uiState := 3;
        3:
                dwCounter := CNT0_CntVal;

                IF DWORD_TO_DINT(dwCounter) >= 200 THEN
                        uiState := 4;
                END_IF
        4:
                CNT0_CntSet := 100;
                CNT0_OutCtrl := CNT0_OutCtrl OR 16#02;
                uiState := 5;
        5:
                byInCtrl := CNT0_InCtrl;
                dwCounter := CNT0_CntVal;

                IF (byInCtrl AND 16#02) = 16#02 THEN
                        uiState := 6;
                END_IF
        6:
                CNT0_OutCtrl := CNT0_OutCtrl AND 16#FD;
                uiState := 7;


        7:
                byInCtrl := CNT0_InCtrl;
                dwCounter := CNT0_CntVal;


                IF (byInCtrl AND 16#02) = 16#00 THEN
                        uiState := 8;
                END_IF
        8:
                CNT0_OutCtrl := 0;
                uiState := 9;
        9:
END_CASE
```

9.File Access

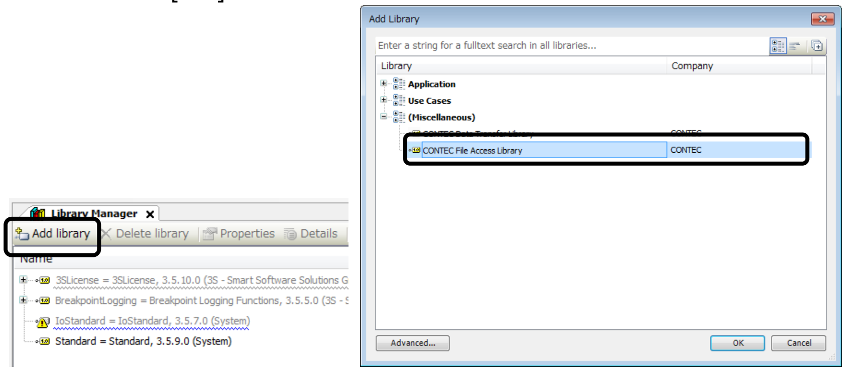
Assign access permission to the file of the memory device connected to this product

1. File Access Library setting

1 Create a New CODESYS project.
Choose ST for IEC program language.
*See “**Create a New Project (page 57)**” for how to create the project.

2 Double-click the [Library Manager] on the device window.

3 Click “Add Library” and choose “CONTEC File Access Library” under “(Miscellaneous)” tree hierarchy, then click the [OK].


Library Manager
Add library
Delete library
Properties
Details
Name
3SLicense = 3SLicense, 3.5.10.0 (3S - Smart Software Solutions G
BreakpointLogging = Breakpoint Logging Functions, 3.5.5.0 (3S - S
IoStandard = IoStandard, 3.5.7.0 (System)
Standard = Standard, 3.5.9.0 (System)
Add Library
Enter a string for a fulltext search in all libraries...
Library
Company
Application
Use Cases
(Miscellaneous)
CONTEC File Access Library
CONTEC
Advanced...
OK
Cancel

2. Function list

Name	Function
CFA_FileAccess	Control access to the specified area
CFA_FileOpen	Open a file
CFA_FileClose	Close a file
CFA_FileCloseAll	Close all files
CFA_FileRead	Read a file
CFA_FileWrite	Write a file
CFA_FileSeek	Move a file pointer
CFA_FileGetLine	Read one line from strings
CFA_FilePutLine	Write a string in a file
CFA_FileDelete	Delete a file
CFA_StringSeparate	Separate a string

3. Data type

◆ ERROR

Name	Value	Meaning
NO_ERROR	0	This indicates successful completion.
INVALID_PARAM	1	This indicates an invalid parameter.
INTERNAL_ERROR	2	This indicates resource insufficiency.
INVALID_HANDLE	3	This indicates an invalid handle number.
NOT_EXIST	4	This indicates no file exists.
EXIST	5	This indicates the file already exists.
READ_ONLY_FS	6	File writing was executed on the mounted reading only device.
NOT_MOUNT	7	This indicates nothing is mounted.
GENERAL_ERROR	8	This indicates a general error.

◆ AREATYPE

Name	Value	Meaning
ROM	0	This specifies a ROM area.
RAM	1	This specifies a RAM area.
SD	2	This specifies a SD card.
USB	3	This specifies a USB device.

◆ SEEKWHENCE

Name	Value	Meaning
CUR	0	This indicates the current position of the file pointer.
END	1	This indicates the end of the file.
SET	2	This indicates the beginning of the file.

4. Function details

◆ CFA_FileAccess

Operating functions

This controls access to the specified area.

Input value

Name	Data type	Meaning
eArea	AREATYPE	This specifies area types to access. As RAM area is always Read/Write-able, no control by this function is necessary.
bMount	BOOL	This specifies mounting to the file system. This is necessary if the access area is USB device. TRUE: This indicates mounting to the file system. FALSE: This indicates unmounting the file system.
bWriteEnable	BOOL	This specifies writing and reading permission. TRUE: This permits writing. FALSE: This prohibits writing.

Output value

Name	Data type	Meaning
eError	ERROR	This indicates an error code

◆ CFA_FileOpen

Operating functions

This opens a file in the specified area.

Input value

Name	Data type	Meaning
eArea	AREATYPE	This specifies an area type to access.
sFileName	STRING(256)	This specifies a file name.
sFileMode	STRING(8)	This specifies a file open mode. "r" : This opens for reading. "w" : This opens for writing. "a" : This opens to add writing. "r+" : This opens for reading and writing. "w+" : This opens for reading and writing. "a+" : This opens for reading and to add writing.

Output value

Name	Data type	Meaning
eError	ERROR	This indicates an error code.
hFile	DWORD	This specifies a handle number of the file.

◆ CFA_FileClose

Operating functions

This closes a file.

Input value

Name	Data type	Meaning
hFile	DWORD	This specifies a handle number of the file.

Output value

Name	Data type	Meaning
eError	ERROR	This indicates an error code.

◆ CFA_FileCloseAll

Operating functions

This closes all files that are open.

Input value

None

Output value

Name	Data type	Meaning
eError	ERROR	This indicates an error code.

◆ CFA_FileRead

Operating functions

This reads data from a file.

Input value

Name	Data type	Meaning
hFile	DWORD	This specifies a handle number of the file
pBuffer	POINTOR TO BYTE	This specifies an address of the area to store data that are read.
szBuffer	WORD	This specifies a size of the area to store data.

Output value

Name	Data type	Meaning
eError	ERROR	This indicates an error code.
szSize	WORD	This indicates the number of data that are read.

◆ CFA_FileWrite

Operating functions

This writes data in the file.

Input value

Name	Data type	Meaning
hFile	DWORD	This specifies a handle number of the file
pBuffer	POINTOR TO BYTE	This specifies an address of the data area to write.
szSize	WORD	This specifies the number of data to write.

Output value

Name	Data type	Meaning
eError	ERROR	This indicates an error code.

◆ CFA_FileSeek

Operating functions

This moves a file pointer to the specified position.

Input value

Name	Data type	Meaning
hFile	DWORD	This specifies a handle number of the file
diOffset	DINT	This specifies the number of the bytes to move from the reference position.
eWhence	SEEKWHENCE	This specifies the reference position.

Output value

Name	Data type	Meaning
eError	ERROR	This indicates an error code.

◆ CFA_FileGetLine

Operating functions

This reads one line in strings from a file.

The read carriage return will be deleted.

Input value

Name	Data type	Meaning
hFile	DWORD	This specifies a handle number of the file.
pszString	POINTOR TO STRING	This specifies an address of the area to store data that are read.
szSring	WORD	This specifies a size of the area to store strings. szString – 1 is the number of maximum characters. Specify a sufficient size to include carriage return.

Output value

Name	Data type	Meaning
eError	ERROR	This indicates an error code.
szSize	WORD	This indicates the string size that are read.

◆ CFA_FilePutLine

Operating functions

This writes a string to a file.

A carriage return will be automatically added at the end of the string.

Input value

Name	Data type	Meaning
hFile	DWORD	This specifies a handle number of the file
pszString	POINTOR TO STRING	This specifies an address to write a string.

Output value

Name	Data type	Meaning
eError	ERROR	This indicates an error code.

◆ CFA_FileDelete

Operating functions

This deletes a file in the specified area.

Input value

Name	Data type	Meaning
eArea	AREATYPE	This specifies an area type to access.
sFileName	STRING(256)	This specifies a file name.

Output value

Name	Data type	Meaning
eError	ERROR	This indicates an error code.

◆ CFA_StringSeparate

Operating functions

This separates a string with a delimiter.

Input value

Name	Data type	Meaning
pszString	POINTER TO STRING	This specifies an address to separate a string.
pszSeparator	POINTER TO STRING	This specifies an address of the group of the delimiters.
pszLeft	POINTER TO STRING	This specifies an address to store strings that are separated with a delimiter and positioned on its left side.
pszRight	POINTER TO STRING	This specifies an address to store strings that are separated with a delimiter and positioned on its right side.

Output value

Name	Data type	Meaning
eError	ERROR	This indicates an error code.

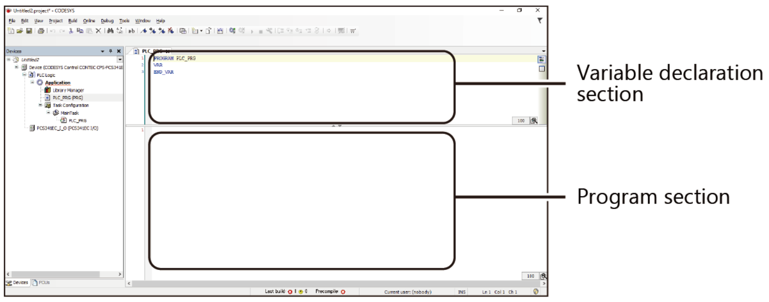
5. Sample program

Here are sample programming steps.

1 Double click the [PLC_PRG (PRG)] icon on the device window to open the “ST editor window”.

2 When “ST editor window” appears, write the following source code between VAR and END_VAR in the “Variable declaration”.

3 Write the following source code in “Program” under “ST editor window”.



```
PROGRAM PLC_PRG
VAR
END_VAR
```

Untitled2.project* - CODESYS
File Edit View Project Build Online Debug Tools Window Help
Devices
Untitled2
Device (CODESYS Control CONTEC CPS-PCS341E
PLC Logic
Application
Library Manager
PLC_PRG (PRG)
Task Configuration
MainTask
PLC_PRG
PCS341EC_I_O (PCS341EC I/O)
PLC_PRG
Variable declaration section
Program section

◆ Sample1

To write a file on a USB device.

Variable declaration

```
uiState: UINT := 0;
eAreaType: INT := CONTEC_File_Access_Library.AREATYPE.USB;
sFileName: CONTEC_File_Access_Library.FILENAME := 'sample1.dat';
hFile: CONTEC_File_Access_Library.HANDLE := 16#FFFFFFFF;
eError: CONTEC_File_Access_Library.ERROR;
byBuffer: ARRAY[0..255] OF BYTE;
iIndex: INT;
CFA_FileAccess: CONTEC_File_Access_Library.CFA_FileAccess;
CFA_FileOpen: CONTEC_File_Access_Library.CFA_FileOpen;
CFA_FileClose: CONTEC_File_Access_Library.CFA_FileClose;
CFA_FileWrite: CONTEC_File_Access_Library.CFA_FileWrite;
```

Program

```
CASE uiState OF
    0:
        CFA_FileAccess.eArea := eAreaType;
        CFA_FileAccess.bMount := TRUE;
        CFA_FileAccess.bWriteEnable := TRUE;
        CFA_FileAccess(eError => eError);
        IF eError = 0 THEN
            uiState := 1;
        ELSE
            uiState := 5;
        END_IF
    1:
        CFA_FileOpen.eArea := eAreaType;
        CFA_FileOpen.sFileName := sFileName;
        CFA_FileOpen.sFileMode := 'w';
        CFA_FileOpen(hFile => hFile, eError => eError);
        IF eError = 0 THEN
            uiState := 2;
        ELSE
            uiState := 4;
        END_IF
    2:
        FOR iIndex := 0 TO 255 DO
            byBuffer[iIndex] := INT_TO_BYTE(iIndex);
        END_FOR
        CFA_FileWrite.pBuffer := ADR(byBuffer);
        CFA_FileWrite.szSize := 256;
        CFA_FileWrite(hFile := hFile, eError => eError);
        uiState := 3;
    3:
        CFA_FileClose(hFile := hFile, eError => eError);
        uiState := 4;
    4:
        CFA_FileAccess.eArea := eAreaType;
        CFA_FileAccess.bMount := FALSE;

        CFA_FileAccess.bWriteEnable := FALSE;
        CFA_FileAccess(eError => eError);
        uiState := 5;
    5:
END_CASE
```

◆ Sample2

To read a file on a USB device.

Variable declaration

```
uiState: UINT := 0;
eAreaType: INT := CONTEC_File_Access_Library.AREATYPE.USB;
sFileName: CONTEC_File_Access_Library.FILENAME := 'sample1.dat';
hFile: CONTEC_File_Access_Library.HANDLE := 16#FFFFFFFF;
eError: CONTEC_File_Access_Library.ERROR;
byBuffer: ARRAY[0..127] OF BYTE;
iIndex: INT;
szSize: WORD;
CFA_FileAccess: CONTEC_File_Access_Library.CFA_FileAccess;
CFA_FileOpen: CONTEC_File_Access_Library.CFA_FileOpen;
CFA_FileClose: CONTEC_File_Access_Library.CFA_FileClose;
CFA_FileRead: CONTEC_File_Access_Library.CFA_FileRead;
```

Program

```
CASE uiState OF
        0:
                CFA_FileAccess.eArea := eAreaType;
                CFA_FileAccess.bMount := TRUE;
                CFA_FileAccess.bWriteEnable := FALSE;
                CFA_FileAccess(eError => eError);
                IF eError = 0 THEN
                        uiState := 1;
                ELSE
                        uiState := 5;
                END_IF

        1:
                CFA_FileOpen.eArea := eAreaType;
                CFA_FileOpen.sFileName := sFileName;
                CFA_FileOpen.sFileMode := 'r';
                CFA_FileOpen(hFile => hFile, eError => eError);

                IF eError = 0 THEN
                        uiState := 2;
                ELSE
                        uiState := 4;
END_IF

        2:
                FOR iIndex := 0 TO 127 DO
                        byBuffer[iIndex] := 0;
                END_FOR
                CFA_FileRead.pBuffer := ADR(byBuffer);
                CFA_FileRead.szBuffer := 128;
                CFA_FileRead(hFile := hFile, szSize => szSize, eError => eError);


                IF eError = 0 THEN
                        IF szSize = 0 THEN
                                uiState := 3;
                        END_IF
                ELSE
                        uiState := 3;
                END_IF

        3:
                CFA_FileClose(hFile := hFile, eError => eError);
                uiState := 4;

        4:
                CFA_FileAccess.eArea := eAreaType;
                CFA_FileAccess.bMount := FALSE;
                CFA_FileAccess.bWriteEnable := FALSE;
                CFA_FileAccess(eError => eError);
                uiState := 5;

        5:
END_CASE
```

◆ Sample3

To get a file size by using a file pointer move function.

Variable declaration

```
uiState: UINT := 0;
eAreaType: INT := CONTEC_File_Access_Library.AREATYPE.USB;
sFileName: CONTEC_File_Access_Library.FILENAME := 'sample1.dat';
hFile: CONTEC_File_Access_Library.HANDLE := 16#FFFFFFFF;
eError: CONTEC_File_Access_Library.ERROR;
diFileSize: DINT := 0;
CFA_FileAccess: CONTEC_File_Access_Library.CFA_FileAccess;
CFA_FileOpen: CONTEC_File_Access_Library.CFA_FileOpen;
CFA_FileClose: CONTEC_File_Access_Library.CFA_FileClose;
CFA_FileSeek: CONTEC_File_Access_Library.CFA_FileSeek;
```

Program

```
CASE uiState OF
        0:
                CFA_FileAccess.eArea := eAreaType;
                CFA_FileAccess.bMount := TRUE;
                CFA_FileAccess.bWriteEnable := FALSE;
                CFA_FileAccess(eError => eError);
                IF eError = 0 THEN
                        uiState := 1;
                ELSE
                        uiState := 5;
                END_IF

        1:
                CFA_FileOpen.eArea := eAreaType;
                CFA_FileOpen.sFileName := sFileName;
                CFA_FileOpen.sFileMode := 'r';
                CFA_FileOpen(hFile => hFile, eError => eError);
                IF eError = 0 THEN
                        uiState := 2;
                ELSE
                        uiState := 4;
                END_IF

        2:
                CFA_FileSeek.diOffset := 0;
                CFA_FileSeek.eWhence := CONTEC_File_Access_Library.SEEKWHENCE.END;
                CFA_FileSeek(hFile := hFIle, diPos => diFileSize, eError => eError);
                uiState := 3;

        3:
                CFA_FileClose(hFile := hFile, eError => eError);
                uiState := 4;

        4:
                CFA_FileAccess.eArea := eAreaType;
                CFA_FileAccess.bMount := FALSE;
                CFA_FileAccess.bWriteEnable := FALSE;
                CFA_FileAccess(eError => eError);
                uiState := 5;

        5:
END_CASE
```

◆ Sample 4

To delete a file on a USB device.

Variable declaration

```
uiState: UINT := 0;
eAreaType: INT := CONTEC_File_Access_Library.AREATYPE.USB;
sFileName: CONTEC_File_Access_Library.FILENAME := 'sample1.dat';
eError: CONTEC_File_Access_Library.ERROR;
CFA_FileAccess: CONTEC_File_Access_Library.CFA_FileAccess;
CFA_FileDelete: CONTEC_File_Access_Library.CFA_FileDelete;
```

Program

```
CASE uiState OF
        0:
                CFA_FileAccess.eArea := eAreaType;
                CFA_FileAccess.bMount := TRUE;
                CFA_FileAccess.bWriteEnable := TRUE;
                CFA_FileAccess(eError => eError);
                IF eError = 0 THEN
                        uiState := 1;
                ELSE
                        uiState := 3;
                END_IF

        1:
                CFA_FileDelete.eArea := eAreaType;
                CFA_FileDelete.sFileName := sFileName;
                CFA_FileDelete(eError => eError);
                uiState := 2;

        2:
                CFA_FileAccess.eArea := eAreaType;
                CFA_FileAccess.bMount := FALSE;
                CFA_FileAccess.bWriteEnable := FALSE;
                CFA_FileAccess(eError => eError);
                uiState := 3;
        3:
END_CASE
```

◆ Sample5

To write or read strings by single line for the file on RAM area.

Variable declaration

```
uiState: UINT := 0;
eAreaType: INT := CONTEC_File_Access_Library.AREATYPE.RAM;
sFileName: CONTEC_File_Access_Library.FILENAME := 'sample2.txt';
hFile: CONTEC_File_Access_Library.HANDLE := 16#FFFFFFFF;
eError: CONTEC_File_Access_Library.ERROR;
sStringData1: STRING(256) := 'Entry1=1234567890';
sStringData2: STRING(256) := 'Entry2=abcdefghijklmnopqrstuvwxyz';
sStringData3: STRING(256);
sSeparator: STRING(8) := '=';
sStringLeft: STRING(256);
sStringRight: STRING(256);
iIndex: INT;
szSize: WORD;
CFA_FileOpen: CONTEC_File_Access_Library.CFA_FileOpen;
CFA_FileClose: CONTEC_File_Access_Library.CFA_FileClose;
CFA_FilePutLine: CONTEC_File_Access_Library.CFA_FilePutLine;
CFA_FileGetLine: CONTEC_File_Access_Library.CFA_FileGetLine;
CFA_StringSeparate: CONTEC_File_Access_Library.CFA_StringSeparate
```

Program

```
CASE uiState OF
        0:
                CFA_FileOpen.eArea := eAreaType;
                CFA_FileOpen.sFileName := sFileName;
                CFA_FileOpen.sFileMode := 'w';
                CFA_FileOpen(hFile => hFile, eError => eError);
                IF eError = 0 THEN
                        uiState := 1;
                ELSE
                        uiState := 7;
                END_IF

        1:
        CFA_FilePutLine.pszString := ADR(sStringData1);
        CFA_FilePutLine(hFile := hFile, eError => eError);
        IF eError = 0 THEN
                uiState := 2;
        ELSE
                uiState := 3;
        END_IF

    2:
```

```
CFA_FilePutLine.pszString := ADR(sStringData2);
          CFA_FilePutLine(hFile := hFile, eError => eError);
          uiState := 3;


3:
          CFA_FileClose(hFile := hFile, eError => eError);
          uiState := 4;

4:
          CFA_FileOpen.eArea := eAreaType;
          CFA_FileOpen.sFileName := sFileName;
          CFA_FileOpen.sFileMode := 'r';
          CFA_FileOpen(hFile => hFile, eError => eError);
          IF eError = 0 THEN
                    uiState := 5;
          ELSE
                    uiState := 7;
          END_IF

5:
          FOR iIndex := 0 TO 256 DO
                    sStringData3[iIndex] := 0;
                    sStringLeft[iIndex] := 0;
                    sStringRight[iIndex] := 0;
          END_FOR
          CFA_FileGetLine.pszString := ADR(sStringData3);
          CFA_FileGetLine.szString := 256;
          CFA_FileGetLine(hFile := hFile, szSize => szSize, eError => eError);

          IF eError = 0 THEN
                    IF szSize <> 0 THEN
                              CFA_StringSeparate.pszString := ADR(sStringData3);
                              CFA_StringSeparate.pszSeparator := ADR(sSeparator);
                              CFA_StringSeparate.pszLeft := ADR(sStringLeft);
                              CFA_StringSeparate.pszRight := ADR(sStringRight);
                              CFA_StringSeparate(eError => eError);
                    ELSE
                              uiState := 6;
                    END_IF
          ELSE
                    uiState := 6;
          END_IF

6:
          CFA_FileClose(hFile := hFile, eError => eError);
          uiState := 7;
7:
          END_CASE
```

10. Transfer data to Cloud server

This section describes how to send data that are collected by the controller to the Cloud service.
Data are separated with a comma in a CSV format to be saved.

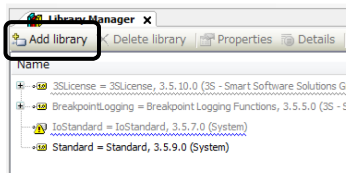
1. File Access Library setting

1 Create a New CODESYS project.
Choose ST for IEC program language.

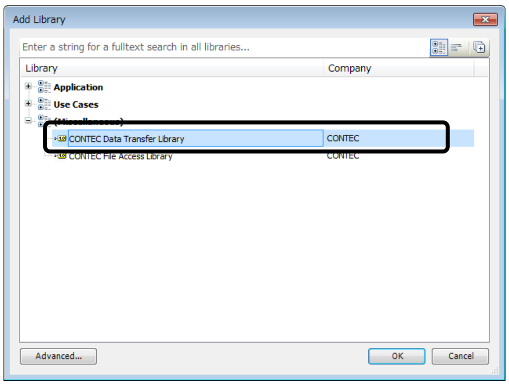
* *See "**Create a New Project (page 57)**" for how to create the project.

2 Double-click the [Library Manager] on the device window.

3 Click the [Add Library] and choose [CONTEC File Access Library] under "(Miscellaneous)" tree hierarchy, then Click the [OK].



3SLicense = 3SLicense, 3.5.10.0 (3S - Smart Software Solutions G
BreakpointLogging = Breakpoint Logging Functions, 3.5.5.0 (3S - S
IoStandard = IoStandard, 3.5.7.0 (System)
Standard = Standard, 3.5.9.0 (System)



Library	Company
CONTEC Data Transfer Library	CONTEC
CONTEC File Access Library	CONTEC

2. Function list

Name	Function
CFA_FileAccess	Control access to the specified area
CFA_FileOpen	Open a file
CFA_FileClose	Close a file
CFA_FileCloseAll	Close all files
CFA_FileRead	Read a file
CFA_FileWrite	Write a file
CFA_FileSeek	Move a file pointer
CFA_FileGetLine	Read one line from strings
CFA_FilePutLine	Write a string in a file
CFA_FileDelete	Delete a file
CFA_StringSeparate	Separate a string

3. Function list

◆ ERROR

Name	Value	Meaning
NO_ERROR	0	This indicates successful completion.
INVALID_PARAM	1	This indicates an invalid parameter.
INTERNAL_ERROR	2	This indicates resource insufficiency.
TX_NODATA	3	This indicates no file exists or no data in the data file.
TX_PENDING	4	Data sending was not performed due to the calling the Cloud sending function in a short period.
TX_ERROR	5	A data file was deleted as it was returned as an error.
TX_RESEND	6	As the transmission failed, the data file is set in the resending.

◆ DATATYPE

Name	Value	Meaning
DATETIME	0	This indicates time data of SYSTIME type
CRLF	1	This is a carriage return indicating the end of the line.
VAL_STRING	2	This indicates STRING type data.
VAL_BOOL	3	This indicates BOOL type data. The size of the memory area is 8 bits.
VAL_BYTE	4	This indicates unsigned 8-bit data.
VAL_WORD	5	This indicates unsigned 16-bit data.
VAL_DWORD	6	This indicates unsigned 32-bit data.
VAL_LWORD	7	This indicates unsigned 64-bit data.
VAL_SINT	8	This indicates signed 8-bit data.
VAL_INT	9	This indicates signed 16-bit data.
VAL_DINT	10	This indicates signed 32-bit data.
VAL_LINT	11	This indicates signed 64-bit data.
VAL_REAL	12	This indicates single precision floating point number data.
VAL_LREAL	13	This indicates double precision floating point number data.

◆ FILEPARAM

Name	Value	Meaning
MAX_SIZE	0	This indicates a data file maximum size in 1K byte.
TIME_FORMAT	1	This indicates the format type of the time data. 0: YYYYMMDD,hhmmss, millisecond 1: YYYYMMDDhhmmss 2: YYYYMMDDhhmm 3: YYYYMMDD 4: hhmmss 5: millisecond

◆ CLOUDPARAM

Name	Value	Meaning
RETRY_TIME	0	This is the time period (second) to resend data when it was set in the resending.
RETRY_NUM	1	This is the number of the resend files on hold.

◆ CLOUDINFO

Name	Value	Meaning
RETRY_NUM	0	This indicates the number of the resend files.

◆ CLOUDCTRL

Name	Value	Meaning
RETRY_CLR	0	This clears the resend files.

4. Function details

◆ CDT_FileSetValue

Operating functions

Create a data file to send to the server.

Data are separated with a comma in a CSV format to be saved.

Input value

Name	Data type	Meaning
iFileNo	INT	This specifies file number (0 to 9).
eType	DATATYPE	This specifies the data type to set to a data file.
pValue	POINTER TO BYTE	This indicates the address to set data. For time data, set the value that was obtained by SysTimeRtcHighResGet(). When 0 is assigned, get the time within the function to set. Specify 0 for a carriage return.

Output value

Name	Data type	Meaning
eError	ERROR	This indicates an error code.

◆ CDT_FileDelete

Operating functions

Delete the data file that is currently being created.

Input value

Name	Data type	Meaning
iFileNo	INT	This specifies file number (0 to 9).

Output value

Name	Data type	Meaning
eError	ERROR	This indicates an error code.

◆ CDT_FileSetParameter

Operating functions

Set the details of parameter regarding a data file.

Input value

Name	Data type	Meaning
iFileNo	INT	This specifies file number (0 to 9).
eParamNo	FILEPARAM	This specifies the parameter number. (MAX_SIZE) This sets the maximum size of the data file in 1K bytes. The oldest data will be deleted when the data exceeds the specified maximum. When 0 is assigned, the size has no limitation. Data : WORD type Default setting : 0 (TIME_FORMAT) This sets the format of the time data. Data : INT type Default setting : 1
pValue	POINTER TO BYTE	This indicates the address to set data.

Output value

Name	Data type	Meaning
eError	ERROR	This indicates an error code.

◆ CDT_CloudSend

Operating functions

Send the created data file to the Cloud service.

The data file that has been sent will be deleted.

Input value

Name	Data type	Meaning
iFileNo	INT	This specifies file number (0 to 9).
xExecute	BOOL	This specifies the execution status. Transmission processing starts when FALSE changes to TRUE.

Output value

Name	Data type	Meaning
xDone	BOOL	This indicates successful completion.
xBusy	BOOL	This indicates function is currently being processed.
xError	BOOL	This indicates completion with an error.
eError	ERROR	This indicates an error code.

◆ CDT_CloudSetParameter

Operating functions

Set the details of parameter for sending to the Cloud.

Input value

Name	Data type	Meaning
eParamNo	CLOUDPARAM	This specifies the parameter number. (RETRY_TIME) This sets the time period (second) to resend data when it was set in the resending. When 0 is assigned, the set file in the resend will be transmitted upon the Cloud sending function is executed. Data : WORD type Default setting : 0 (RETRY_NUM) This sets the number of the resend files on hold. When 0 is assigned, the resend file will not be held. Data : DWORD type Default setting : FFFFFFFFh
pValue	POINTER TO BYTE	This indicates the address to set value.

Output value

Name	Data type	Meaning
eError	ERROR	This indicates an error code.

◆ CDT_CloudGetInformation

Operating functions

Get information regarding sending to the Cloud.

Input value

Name	Data type	Meaning
eInfoNo	CLOUDINFO	This specifies the information number. (RETRY_NUM) This gets the number of resend files Data : DWORD type
pValue	POINTER TO BYTE	This specifies an address of the area to store information.

Output value

Name	Data type	Meaning
eError	ERROR	This indicates an error code.

◆ CDT_CloudControl

Operating functions

Specify the control for sending to the Cloud.

Input value

Name	Data type	Meaning
eCtrlNo	CLOUDCTRL	This specifies the controlling number. (RETRY_CLR) This clears the resend files. Data : none
pValue	POINTER TO BYTE	This specifies an address of the area to store controlling information.

Output value

Name	Data type	Meaning
eError	ERROR	This indicates an error code.

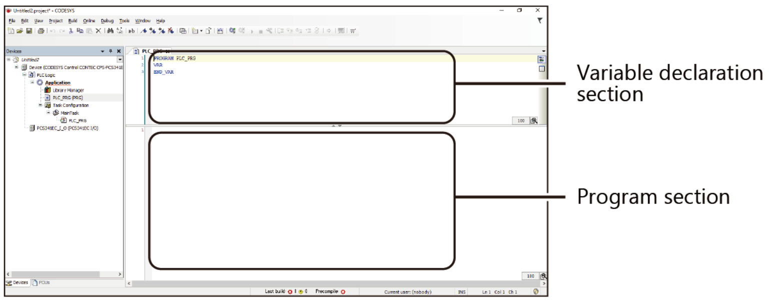
5. Sample

Here are sample programming steps.

1 Double click the [PLC_PRG (PRG)] icon on the device window to open the “ST editor window".

2 When “ST editor window” appears, write the following source code between VAR and END_VAR in the “Variable declaration”.

3 Write the following source code in “Program” under “ST editor window”.



```
PROGRAM PLC_PRG
VAR
END_VAR
```

◆ Sample1

To send the digital input data of CPS-DIO-0808RL and the temperature data of CPS-SSI-4P to the Cloud service.

* In this sample program, the CPS-DIO-0808RL and the CPS-SSI-4P are connected to the configurable type controller.

Variable declaration

```
DiData: ARRAY[0..7] OF BOOL;
TempData: ARRAY[0..3] OF LREAL;
dwIntervalTime: DWORD := 100;
dwTimerCount1: DWORD;
dwTimerCount2: DWORD;
uiState: UINT := 0;
uiCloudTxState: UINT;
iFileNo: INT := 0;
usFileSize: WORD := 8;
iTimeFormat: INT := 1;
dwCounter: DWORD;
iIndex: INT;
eError: CONTEC_Data_Transfer_Library.ERROR;
eCloudTxError: CONTEC_Data_Transfer_Library.ERROR;
CDT_FileSetValue: CONTEC_Data_Transfer_Library.CDT_FileSetValue;
CDT_FileSetParameter: CONTEC_Data_Transfer_Library.CDT_FileSetParameter;
CDT_CloudSend: CONTEC_Data_Transfer_Library.CDT_CloudSend;
CDT_CloudSetParameter: CONTEC_Data_Transfer_Library.CDT_CloudSetParameter;
CDT_CloudGetInformation: CONTEC_Data_Transfer_Library.CDT_CloudGetInformation;
```

Program

```
CASE uiState OF
        0:
                CDT_FileSetParameter.eParamNo :=
CONTEC_Data_Transfer_Library.FILEPARAM.MAX_SIZE;
                CDT_FileSetParameter.pValue := ADR(usFileSize);
                CDT_FileSetParameter(iFileNo := iFileNo, eError => eError);

                CDT_FileSetParameter.eParamNo :=
                        CONTEC_Data_Transfer_Library.FILEPARAM.TIME_FORMAT;
                CDT_FileSetParameter.pValue := ADR(iTimeFormat);
                CDT_FileSetParameter(iFileNo := iFileNo, eError => eError);
                dwTimerCount1 := 0;
                dwTimerCount2 := 0;
                wCounter := 0;
                uiCloudTxState := 0;
                uiState := 1;

        1:
                DO_BYTE := DO_BYTE + 1;
                DiData[0] := DI_BIT0;
                DiData[1] := DI_BIT1;
                DiData[2] := DI_BIT2;
                DiData[3] := DI_BIT3;
                DiData[4] := DI_BIT4;
                DiData[5] := DI_BIT5;
                DiData[6] := DI_BIT6;
                DiData[7] := DI_BIT7;
                TempData[0] := TEMP0;
                TempData[1] := TEMP1;
                TempData[2] := TEMP2;
                TempData[3] := TEMP3;

                dwTimerCount1 := dwTimerCount1 + dwIntervalTime;
                IF dwTimerCount1 >= 60000 THEN
                        dwTimerCount1 := 0;

                        CDT_FileSetValue.iFileNo := iFileNo;
                        CDT_FileSetValue.eType :=
CONTEC_Data_Transfer_Library.DATATYPE.DATETIME;
                        CDT_FileSetValue.pValue := 0;
                        CDT_FileSetValue(eError => eError);

                        dwCounter := dwCounter + 1;
                        CDT_FileSetValue.eType :=
CONTEC_Data_Transfer_Library.DATATYPE.VAL_DWORD;
                        CDT_FileSetValue.pValue := ADR(dwCounter);
                        CDT_FileSetValue(eError => eError);

                        CDT_FileSetValue.eType :=
CONTEC_Data_Transfer_Library.DATATYPE.VAL_BOOL;
                        FOR iIndex := 0 TO 7 DO
```

```
                    CDT_FileSetValue.pValue := ADR(DiData[iIndex]);
                    CDT_FileSetValue(eError => eError);
                END_FOR

                CDT_FileSetValue.eType :=
CONTEC_Data_Transfer_Library.DATATYPE.VAL_LREAL;

                FOR iIndex := 0 TO 3 DO
                    CDT_FileSetValue.pValue := ADR(TempData[iIndex]);
                    CDT_FileSetValue(eError => eError);
                END_FOR

                CDT_FileSetValue.eType :=
CONTEC_Data_Transfer_Library.DATATYPE.CRLF;
                CDT_FileSetValue.pValue := 0;
                CDT_FileSetValue(eError => eError);
            END_IF

            dwTimerCount2 := dwTimerCount2 + dwIntervalTime;

            CASE uiCloudTxState OF
        0:
                IF dwTimerCount2 >= 600000 THEN
                    dwTimerCount2 := 0;
                    CDT_CloudSend.iFileNo := iFileNo;
                    CDT_CloudSend(xExecute := FALSE);
                    CDT_CloudSend(xExecute := TRUE);
                    uiCloudTxState := 1;
                END_IF
        1:
                CDT_CloudSend();
                IF CDT_CloudSend.xBusy = FALSE THEN
                    IF CDT_CloudSend.xDone OR CDT_CloudSend.xError THEN
                        eCloudTxError := CDT_CloudSend.eError;
                        uiCloudTxState := 0;
                    END_IF
                END_IF
            END_CASE
END_CASE
```

◆ Sample2

To set the time data obtained in application.

It is necessary to add “SysTime” in “System”-“SysLibs” of the [Library Manager] for SysTimeRtcHighResGet().

Variable declaration

```
uiState: UINT := 0;
iFileNo: INT := 0;
TimeStamp: SYSTIME;
eError: CONTEC_Data_Transfer_Library.ERROR;
CDT_FileSetValue: CONTEC_Data_Transfer_Library.CDT_FileSetValue;
```

Program

```
CASE uiState OF
        0:
                SysTimeRtcHighResGet(TimeStamp);

                CDT_FileSetValue.iFileNo := iFileNo;
                CDT_FileSetValue.eType := CONTEC_Data_Transfer_Library.DATATYPE.DATETIME;
                CDT_FileSetValue.pValue := ADR(TimeStamp);
                CDT_FileSetValue(eError => eError);

                CDT_FileSetValue.eType := CONTEC_Data_Transfer_Library.DATATYPE.CRLF;
                CDT_FileSetValue.pValue := 0;
                CDT_FileSetValue(eError => eError);
                uiState := 1;

        1:
END_CASE
```

Monitoring Edit

This chapter describes the CONPROSYS HMI (Human Machine Interface), with which you can create a monitoring screen through a web browser and check the operations, errors, or standstill in the facilities.

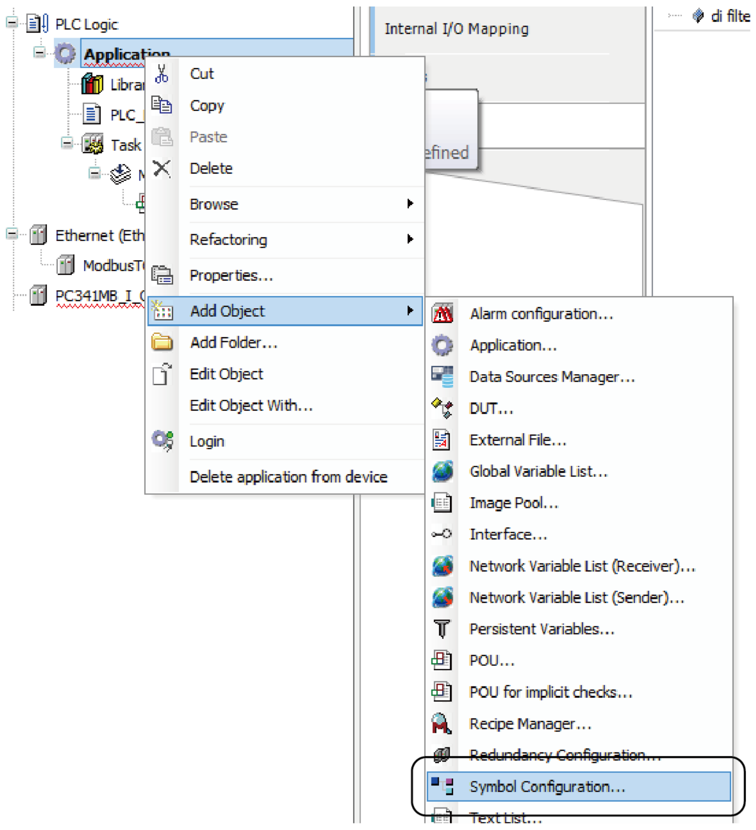
1.Use CONPROSYS HMI

To perform monitoring with this product, create a monitoring screen by the CONPROSYS HMI software that is included in the controller.

1. CODESYS project setting

1 Create a CODESYS project by following the procedure described in the “Create a New Project (page 57)” and “Connect controller from CODESYS (page 58)” of the “Basic programming procedure (page 57)”.

2 On the Devices window, right-click on the [Application], and select the [Add object] - [Symbol configuration].



3 In the “Add Symbol configuration” dialog box, click the [Add] button.

2. Create and build a program

This example demonstrates the following CODESYS program.
I/O functions from hardware are not used in this sample, only software variables are listed.

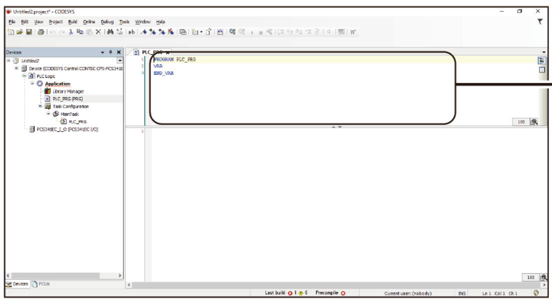
Incrementer is a variable that performs count-up per cycle time.

Switch variable controls starting and ending operations.

Status variable keeps current status as character strings.

1 Double click the [PLC_PRG (PRG)] icon on the device window to open the “ST editor window".

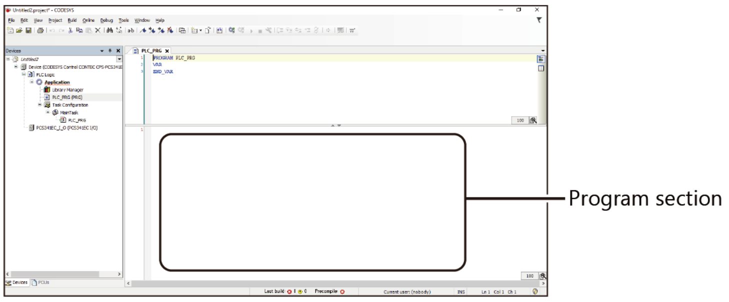
2 When “ST editor window” appears, write the following source code between VAR and END_VAR in the “Variable declaration”.



```
PROGRAM PLC_PRG
VAR
END_VAR
```

```
Switch: BOOL;
Incrementer: INT;
Status: STRING[10];
```

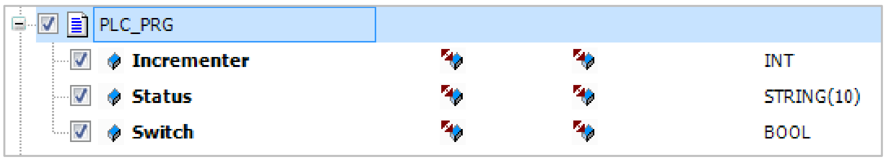
3 Write the following source code in "Program" under "ST editor window".



```
IF Switch = TRUE THEN
            Incrementer := Incrementer + 1;
            IF Incrementer = 100 THEN
                        Incrementer := 0;
            END_IF
            Status := 'BUSY';
ELSE
            Status := 'STOP';
END_IF
```

4 Now, see whether you can execute [Rebuild] in CODESYS [Build] menu to confirm the process succeeds.

5 Double –click the [Symbol configuration] on CODESYS window, and check off the boxes of "Incrementer", "Status", and "Switch" that are located under PLC_PRG tree hierarchy.



PLC_PRG	
Incrementer	INT
Status	STRING(10)
Switch	BOOL

*Variables registered under the "IoConfig_Globals_Mapping" or "PLC_PRG" tree of Symbol configuration can be used in CONPROSYS HMI. Other variables registered in the tree would be ignored.

6 Select the [Online] – [Login] on CODESYS menu.
* Click the [Yes] in the download confirmation dialog.

3. Create a monitoring screen with HMI Editor

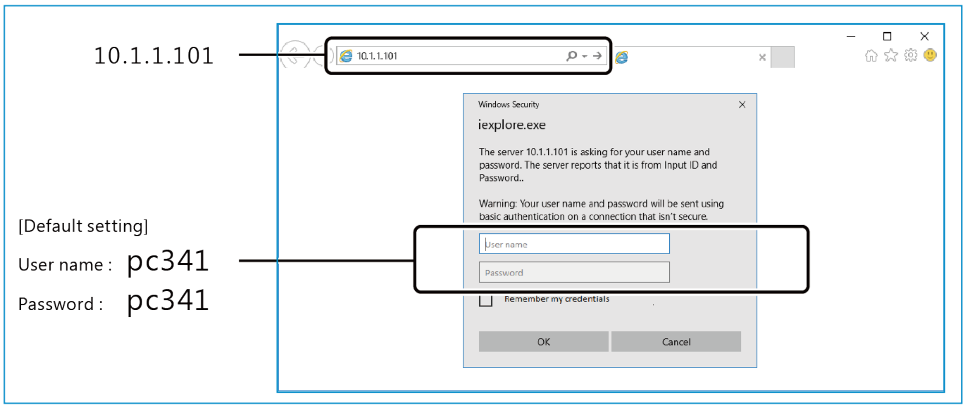
1 Start the Web browser on the PC connected with the controller and enter “http://10.1.1.101/” (the IP address of the PLC controller) in the address field.

* See “**Set the Computer Network (page 18)**” for connecting controller

* Also, accessible through “https://10.1.1.101/”. If the [Certificate Error] message appears, choose the "Continue to this website".

2 Enter user name “pc341” and password “pc341” to log in.

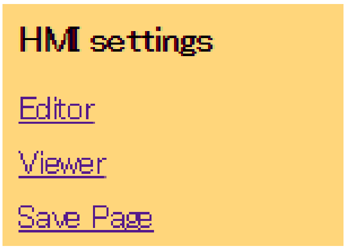
.



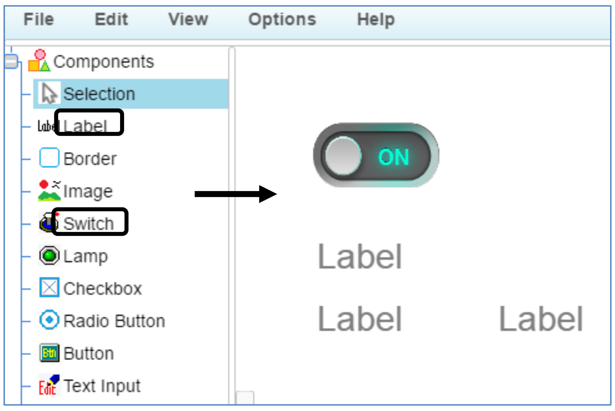
The server 10.1.1.101 is asking for your user name and password. The server reports that it is from Input ID and Password..

Warning: Your user name and password will be sent using basic authentication on a connection that isn't secure.

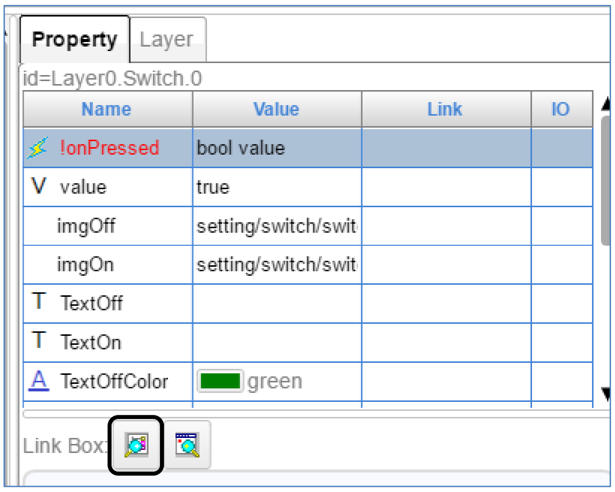
3 Select “Editor” in HMI settings to start CONPROSYS HMI.



4 Drag one “Switch” and three “Labels” from “Components” located left-hand on the screen and drop them to the Layer area.



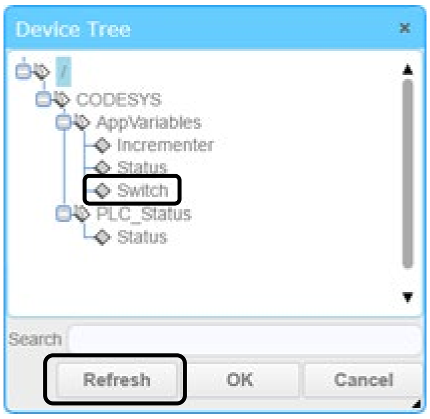
5 Activate the ON switch and click the [Link Box] icon in Property tab located right-hand on the editor screen.



id=Layer0.Switch.0

Name	Value	Link	IO
!onPressed	bool value		
V value	true		
imgOff	setting/switch/swit		
imgOn	setting/switch/swit		
T TextOff			
T TextOn			
A TextOffColor	🟩 green		

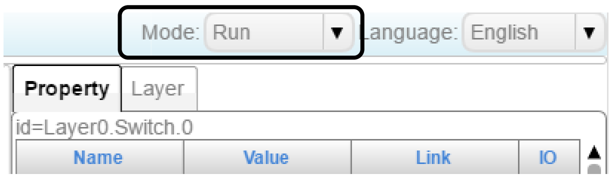
6 Click "Refresh" in Device Tree. This updates the lists of the variables used in CODESYS. Choose "Switch" and click "OK." This allocates variable Switch to Switch control.



7 Follow the same procedure of step 5 and 6 above, assign Incrementer variable and Status variable with the two Labels.
For the one Label, enter 'Status' in Text of Property to display the letter "Status" on the label.

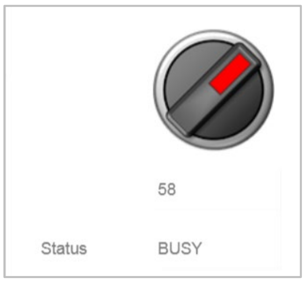
4. HMI run by viewer

1 Choose “Run” in the Mode located upper right on the screen.



2 From the [Debug] menu in CODESYS, Select the [Start].

3 Turning on the ON switch placed on the canvas updates the counts of Incrementer range from 0 to 100. Turning it off stops the updating.
1 is seen in Status, which indicates PLC is in operation. The count changes to 0 when the operation in CODESYS stops.



4 Operate by Viewer on web browser, not with Editor.
First, save the HMI page created in the [File] – [Run by viewer].
Save it under user folder with the name Page1.
Next, select the [File] – [Run by viewer]. This opens a new tab and Viewer page appears.

* The saved page will be disposed upon rebooting the controller. There are two ways to save pages into ROM area as follows:

- Click “Save the setting to ROM” in “File”.
- Click “Save Page” in HMI settings on WEB setting monitor.

5. Available components with IEC data type

See below for available IEC data types of CONPROSYS HMI.

- BOOL
- BYTE
- USINT
- WORD
- UINT
- DWORD
- UDINT
- LWORD
- ULINT
- SINT
- INT
- DINT
- LINT
- REAL
- LREAL
- STRING

Available IEC data type of each component

Component name	BOOL type	STRING type	Other type
Label	○	○	○
Border	×	×	×
Image	×	×	×
Switch	○	×	×
Lamp	○	×	×
Checkbox	○	×	×
Radio Button	×	×	×
Button	×	×	×
Text Input	○	○	○
Slider	○	×	○
Video	×	×	×
Trend	○	×	○
Circle Meter	○	×	○
Level Meter	○	×	○
Circle Graph	○	×	○
Trend Bar	○	×	○
OnDelay Switch	○	×	×
Multi-State Lamp	○	×	○

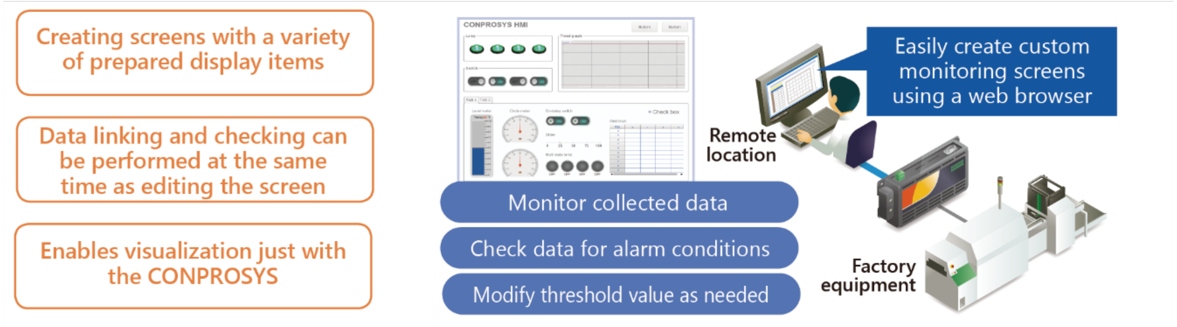
2.CONPROSYS HMI Outline

1. What You Can Do With CONPROSYS HMI

Place the supplied controls on the page and create a monitoring screen. Input signal status can be monitored on the screen. The screen can be created through a Web browser.

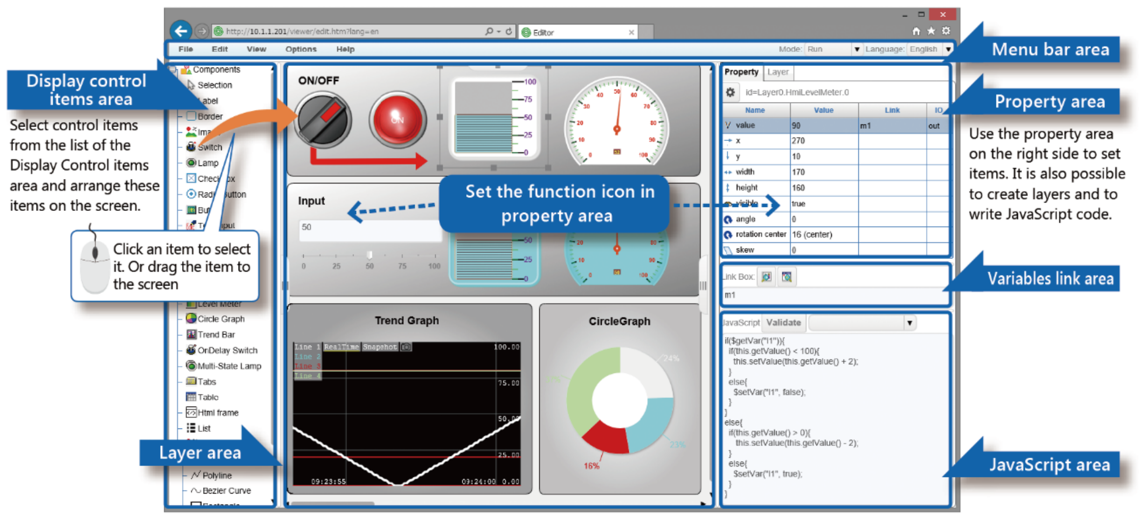
Neither the knowledge of language programming nor the special development environment is necessary. Just drag and drop a number of controls on the page to create the screen.

Control settings or linking data with a sensor can be done on the property screen.


Creating screens with a variety of prepared display items
Data linking and checking can be performed at the same time as editing the screen
Enables visualization just with the CONPROSYS
CONPROSYS HMI
Easily create custom monitoring screens using a web browser
Remote location
Monitor collected data
Check data for alarm conditions
Modify threshold value as needed
Factory equipment

Device setting, data saving, and script debugging are completed with a Web browser

Support multiple platforms (Windows, Android)



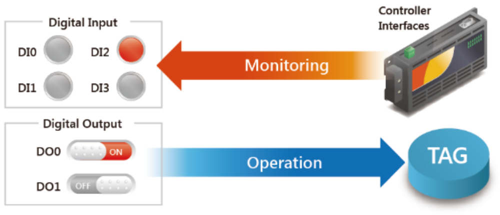
Name	Value	Link	IO
value	90	m1	out
x	270		
y	10		
width	170		
height	160		
visible	true		
angle	0		
rotation center	16 (center)		
skew	0		

```
if($getVar("I1")){
  if(this.getValue() < 100){
    this.setValue(this.getValue() + 2);
  }
  else{
    $setVar("I1", false);
  }
}
else{
  if(this.getValue() > 0){
    this.setValue(this.getValue() - 2);
  }
  else{
    $setVar("I1", true);
  }
}
```

The screen above is the image when creating the monitoring screen.
The factory default setting is a white background.

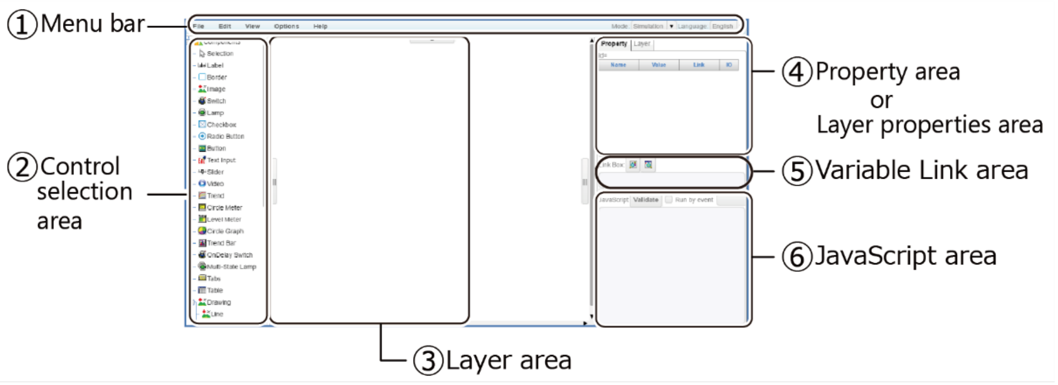
Data linking with CONPROSYS VTC

It is possible to use internal variables (TAGs) to link with CONPROSYS VTC. This makes it possible to display the results of processing with VTC and to apply the HMI operations to VTC flow control.



2. HMI Editor Work Areas

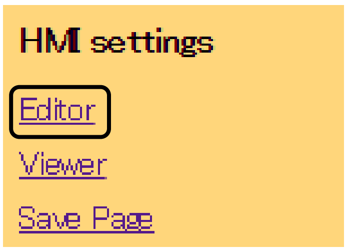
HMI Editor has the following work areas.



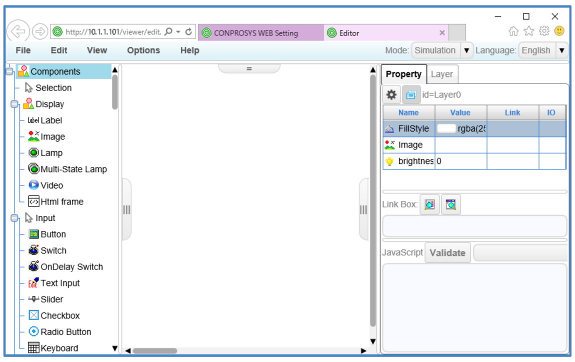
No.	Name	Function
1	Menu bar	The menu bar is used to execute commands with menu buttons.
2	Control selection area	Select the controls to place on the layer area.
3	Layer area	This area is where the page is designed.
4	Properties area	Modify the property variables of controls on this area.
	Layer properties area	Add or delete layers, and modify layer properties on this area.
5	Variable Link area	This area is for linking control properties and variables.
6	JavaScript area	This area is used to configure the behavior of JavaScript for controls.

3. Create a Monitoring Screen

From the CONPROSYS WEB Setting, click the “Editor” in “HMI settings”.



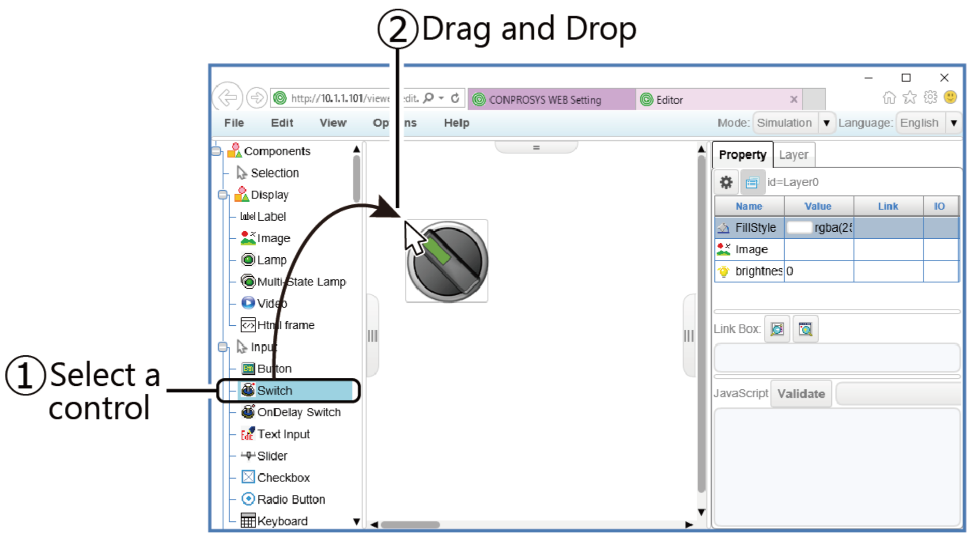
CONPROSYS HMI starts up and you can create the monitoring screen.



4. Basic Procedure for Creating a Monitoring Screen

◆ Place a Control

(1) Select a control from the Components tree displayed on the left. (2) Drag and drop it on the Layer area.

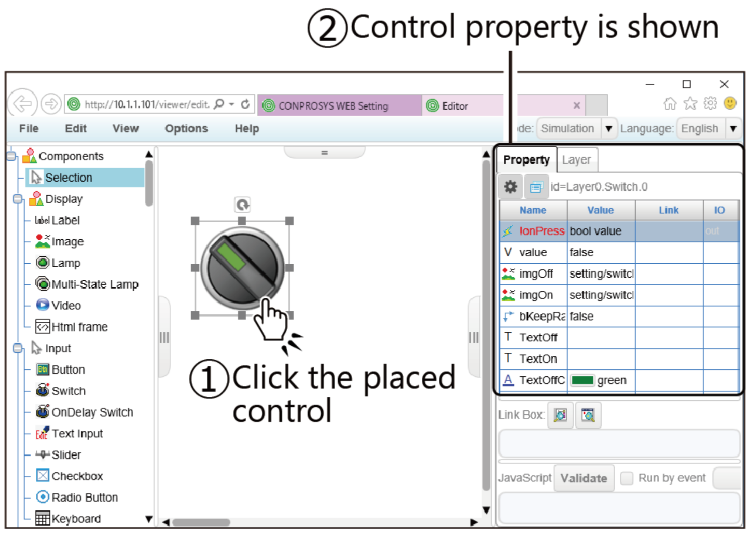


Name	Value	Link	IO
FillStyle	rgba(2		
Image			
brightnes	0		

◆ Configure the Properties of Controls

(1)Click the placed control. (2)The property of the control is shown in the “Property” on the right side of the screen.

The properties area allows you to change the values, set the data to link with I/O devices or other controls.

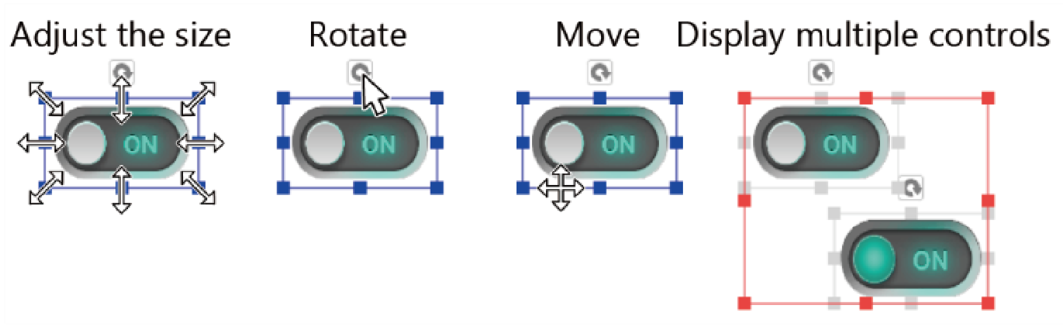


Name	Value	Link	IO
IonPress	bool value		
value	false		
imgOff	setting/switc		
imgOn	setting/switc		
bKeepRa	false		
TextOff			
TextOn			
TextOffC	🟩 green		

◆ Align the Position or Adjust the Size of Controls

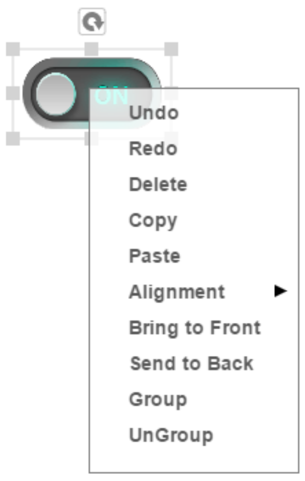
Click the placed control to activate. Drag the border to change the position, adjust the size or the angle.

Controls can be selected together and changed or adjusted simultaneously.



◆ Copy and Delete Controls

Right-click the activated control to show the editing menu. In this menu, such as coping or deleting controls can be performed.

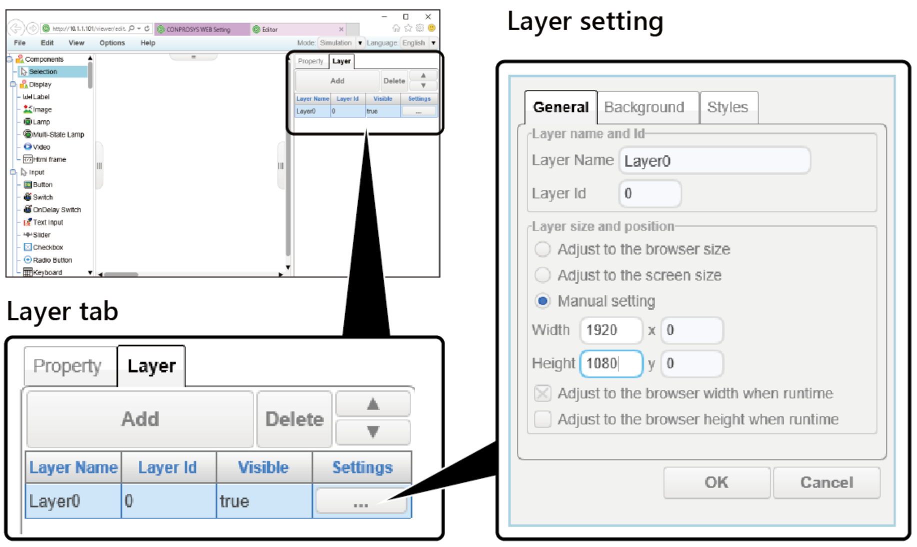


◆ Configure the Layer

Select "Layer" tab at the upper right on the screen to open the layer properties.

In the "Settings" of the layer properties area, click the [...] button to open the dialog box.

You can set a size of monitoring screen or the background here.



Layer Name	Layer Id	Visible	Settings
Layer0	0	true	...

Layer name and Id
Layer Name: Layer0
Layer Id: 0

Layer size and position
- Adjust to the browser size
- Adjust to the screen size
- [✓ Manual setting]

Width 1920 x 0
Height 1080 y 0

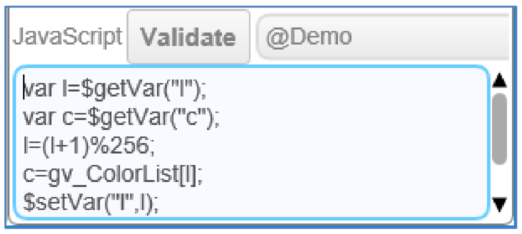
- Adjust to the browser width when runtime
- Adjust to the browser height when runtime

◆ JavaScript

You can enter code using JavaScript as necessary.

If a particular control logic is needed to run the system, code the behaviors of the system using JavaScript in "JavaScript" area.

Refer to "Online Help" for usable JavaScript functions for each control.



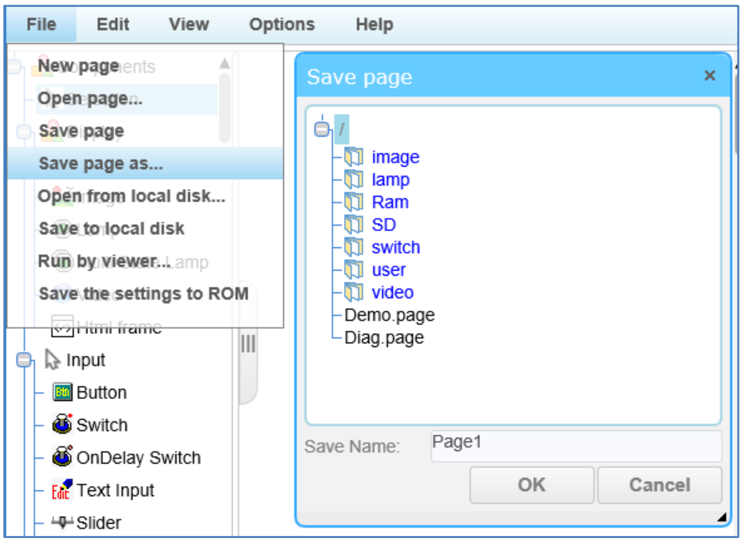
```
var l=$getVar("l");
var c=$getVar("c");
l=(l+1)%256;
c=gv_ColorList[l];
$setVar("l",l);
```

◆ Save the Settings to ROM.

After creating the monitoring screen, save the file with a new name.

After saving, perform "Save to ROM" in the [File] before shutting down the power.

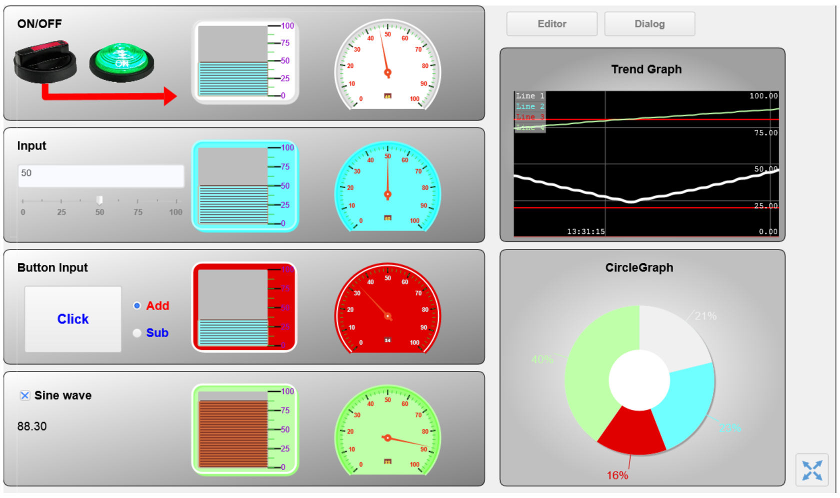
* If you do not save the settings to ROM, the contents return to those before setting upon rebooting or shutting down.



◆ Display the Created Screens

From WEB menu, click the "Monitoring view" in "Status menu" and the monitoring screen appears.

The page that is saved in "user/Page1.page" on the monitor can be viewed.



*When viewing the screen with a specific name, specify the URL listed below through a browser.

http://<IP address>**:**<port number>**/viewer/view.htm?pagepath=**<page file path>
&lang=<language>

<Page file path>: Specify a name of the page. An example: /user/Page1.page

<Language>: Specify the language to view. An example: jp indicates Japanese. Specifying the language can be omitted.

3.Summary of Available Controls

CONPROSYS HMI provides the following controls.

Control	Name	Description
Label	Label	This control displays a string.
▢	Border	This control is a border with a title.
	Image	This control displays an image.
	Switch	This control is a switch that can output an ON/OFF status.
●	Lamp	This control is a lamp that can display an ON/OFF status.
☒	Checkbox	This control is a checkbox that can output an ON/OFF status and display a string.
◉	Radio Button	This control is a radio button to select a single condition from multiple conditions.
Btn	Button	This control is a clickable button that displays a text string.
Edit	Text Input	This control is used to input and display text.
	Slider	This control is used to output data with a slider.
▶	Video	This control is used to play videos.
	Trend	This control is used to display chronological data as a graph.
	Circle Meter	This control is used to display data as a circle meter.
	Level Meter	This control is used to display data as a level meter.
	Circle Graph	This control is used to display data as a circle graph.
	Trend Bar	This control is used to display data as trend lines or bars.
	OnDelay Switch	This control is a switch that can output an ON/OFF status after being pressed in specified seconds.
◉	Multi-State Lamp	This control is a lamp that can display multiple differing status values.
	Timer	This control is used to keep counting between the maximum value and the minimum value periodically.
1	Calendar	This control is used to display and set the date.
	Clock	This control is used to display the current time.
☰	Drop-down List	This control is used to display the value in drop-down list format.
▦	Keyboard	This control is used to display and inputs the value in software keyboard format.
1	Number to Color	This control is used to convert a number to a color string.
bit	Number to Bits	This control is used to convert numbers and binary values.
Tab	Tabs	This control is used to create multiple tabs that can be displayed by switching.
▦	Table	This control is used to display data in table format.
<>	Html frame	This control is used to display another Html document in the frame.
☰	List	This control is used to display data by a list.
/	Line	This control is used to draw a line on the page.

Control	Name	Description
⩘	Polyline	This control is used to draw a polyline on the page.
∿	Bezier Curve	This control is used to draw a Bezier curve on the page.
□	Rectangle	This control is used to draw a rectangle on the page.
▢	Round Rectangle	This control is used to draw a rounded rectangle on the page.
△	Polygon	This control is used to draw a polygon on the page.
○	Ellipse	This control is used to draw a circle or ellipse on the page.
⮧	Pipe	This control is used to draw a pipe-style continuous line on the page.

Refer to "**Online Help**" for the details of CONPROSYS HMI operation and functions.

Online Help http://data.conprosys.com/help/hmi/V1/en/

Set Up Troubleshooting

This section describes how to check and solve the troubles when the product does not function properly.

1. If you encounter a problem?

Perform the following checks if you encounter a problem in the use of this product.

1. General

◆ Check the LEDs on the front panel

- Check that PWR LED is on.
- Check that ST1 LED is flashing.

◆ Check the network port LEDs

Check the LEDs on the UTP connector at the front of the unit.

The Link/Act LED lights up if the network cable is correctly connected to a hub.

If not, refer to the "Hardware Setup Guide" and check the connection.

The Link/Act LED will be flashing when communication is in progress via the network port.

◆ Use the ping command from a host computer and confirm that the server unit responds.

Ping the IP address of the server unit.

The server unit should respond if it is operating.

Example: The following response should be received when the server unit is set to IP address 10.1.1.101:

```
ping 10.1.1.101<Enter>:
Reply from 10.1.1.101: bytes=32 time<10ms TTL=255
Reply from 10.1.1.101: bytes=32 time<10ms TTL=255
Reply from 10.1.1.101: bytes=32 time<10ms TTL=255
Response is displayed
```

If you are unsure of the IP address of the server unit, you can restore the default factory settings (IP address 10.1.1.101) by turning on the power to the unit with SW1-2 switch on (left).

⚠ CAUTION

If you turn off (right) SW1-2 switch, the product starts up with the previous settings that are saved to ROM.

◆ If your user name and password are not recognized when you connect from a browser on a host computer:

Both the user name and password are case sensitive (upper and lower case letters are treated as different). Make sure that the Caps Lock key is off and try again.

If you have forgotten your user name or/and password, you can restore the default factory settings by turning on the power to the product with SW1-2 (left).

(IP address also starts with the default factory settings)

⚠CAUTION

This also initializes all other settings.

◆ - If the ping command receives a response but a "page not found" message appears when you try to connect from a browser.

Setup your browser as follows:

Proxy server setting

Set "do not use proxy server".

Dialup setting

Set "do not dial".

◆ Product does not function properly

Contact CONTEC to have the product examined.

System Reference

This section lists specifications of hardware and CONPROSYS HMI.

1.Hardware Specification

1. CPS-PC341EC-1-9201 specification

Item		CPS-PC341EC-1-9201
CODESYS supporting functions	Version	V3.5 SP7 Patch2 or later version
	Language	LD, SFC, FBD, ST, IL, CFC (IEC61131-3-complied)
	Field bus	EtherCAT Master, Modbus TCP Slave
	Communication protocol	OPC-UA Server
Program size	ROM size	1MB
	Maximum steps	250K steps
CPU basic performance	Basic instruction execution speed (LD)	1.6ns
	Application instruction execution speed (ST)	5.8ns
	Scan time	74µs (20000 steps)
EtherCAT performance	Input processing time (LD)	144ns
	Output processing time (ST)	138ns
	Scan time	166µs (64 inputs and 64 outputs)

2. CPS-PC341MB-ADSC1-9201 specification

Item		CPS-PC341MB-ADSC1-9201
CODESYS supporting functions	Version	V3.5 SP7 Patch2 or later version
	Language	LD, SFC, FBD, ST, IL, CFC (IEC61131-3-complied)
	Field bus	Modbus TCP Master / Slave, Modbus RTU Master / Slave
	Communication protocol	OPC-UA Server
Program size	ROM size	1MB
	Maximum steps	250K steps
CPU basic performance	Basic instruction execution speed (LD)	1.6ns
	Application instruction execution speed (ST)	5.8ns
	Scan time	74µs (at 20000 steps)

3. CPS-PCS341EC-DS1-1201 specification

Item		CPS-PCS341EC-DS1-1201
CODESYS supporting functions	Version	V3.5 SP7 Patch2 or later version
	Language	LD, SFC, FBD, ST, IL, CFC (IEC61131-3-complied)
	Field bus	EtherCAT Master, Modbus TCP Slave
	Communication protocol	OPC-UA Server
Program size	ROM size	1MB
	Maximum steps	250K steps
CPU basic performance	Basic instruction execution speed (LD)	1.6ns
	Application instruction execution speed (ST)	5.8ns
	Scan time	74μs (20000 steps)
EtherCAT performance	Input processing time (LD)	144ns
	Output processing time (ST)	138ns
	Scan time	166μs (64 inputs and 64 outputs)

4. CPS-PCS341MB-DS1-1201 specification

Item		CPS-PCS341MB-DS1-1201
CODESYS supporting functions	Version	V3.5 SP7 Patch2 or later version
	Language	LD, SFC, FBD, ST, IL, CFC (IEC61131-3-complied)
	Field bus	Modbus TCP Master / Slave, Modbus RTU Master / Slave
	Communication protocol	OPC-UA Server
Program size	ROM size	1MB
	Maximum steps	250K steps
CPU basic performance	Basic instruction execution speed (LD)	1.6ns
	Application instruction execution speed (ST)	5.8ns
	Scan time	74μs (20000 steps)

2.CONPROSYS HMI Specification

Item		Specification
Supporting IEC data type		BOOL BYTE USINT WORD UINT DWORD UDINT LWORD ULINT SINT INT DINT LINT REAL LREAL STRING
CONPROSYS HMI reserved variables		PLC status Item name: Status/PLCStatus Access: R Data type: BYTE Data range: 0 (PLC stopped) or 1 (PLC busy)
Maximum number of usable variables		128
Factory setting	CPS-PC341EC-1-9201	None
	CPS-PC341MB-ADSC1-9201	Digital output bit 0 - 1 Item name : IoVariables/DO0 - 1 Access : R/W Data type : BIT Data range : 0 or 1 Digital input bit 0 - 3 Item name : IoVariables/DI0 - 3 Access : R Data type : BIT Data range : 0 or 1 Analog input channel 0 - 1 Item name : IoVariables/AI0 - 1 Access : R Data type : DWORD Data range : 0 - 4095 Counter input channel 0 - 1 Item name : IoVariables/CNT0 - 1 Access : R Data type : DWORD Data range : 0 - 16777215 Counter input channel 0 - 1 clear Item name : IoVariables/CNTCLR0 - 1 Access : R/W Data type : BIT Data range : 0 or 1 (1 is clear)

Item		Specification
		PLC status Item name : Status/PLCStatus Access : R Data type : BYTE Data range : 0 (PLC stopped) or 1 (PLC busy)
	CPS-PCS341EC-DS1-1201 CPS-PCS341MB-DS1-1201	Digital input bit 0 - 3 I tem name : IoVariables/DI0 - 3 Access : R Data type : BIT Data range : 0 or 1 PLC status Item name : Status/PLCStatus Access : R Data type : BYTE Data range : 0 (PLC stopped) or 1 (PLC busy)

Appendix

This section describes additional information of specification and the product.

1.Data Transfer Format

Data is transferred to the server via “http” or “https”.
Data is posted to the specified URL by the following parameters.

◆ Transfer Parameter List

Transfer Contents	Parameter 1	Parameter 2
Measured data file	file=data	filename=YYYYMMDDhhmm.csv

◆ Response from a Web Server

Response	Description	Operation
Code: 200 X-AggregateInfo-Result: OK	Normal	Delete the files that are already sent.
Code: 400	Invalid ID, Authentication code error, Format error	Delete the files that failed to be sent.
Others	Other errors	Keep the failed files to resend.

◆ Telegram, e.g.

http Request

```
POST /XXXXX HTTP/1.1<CR_LF>
User-Agent: XXXXX<CR_LF>
Host: 192.168.1.110<CR_LF>
Accept: */*<CR_LF>
Content-Length: 40602<CR_LF>
Expect: 100-continue<CR_LF>
Content-Type: multipart/form-data; boundary=------------------------43ac9283b67c39f1<CR_LF>
Content-Disposition: form-data; name=“data”; filename=“201401011000.csv”<CR_LF>
Content-Type: text/plain;charset=UTF-8<CR_LF>
<CR_LF>
[Measured data]
--------------------------43ac9283b67c39f1<CR_LF>
Content-Disposition: form-data; name=“err”; filename=“201401011000_e.csv “<CR_LF>
Content-Type: text/plain;charset=UTF-8<CR_LF>
<CR_LF>
```

http Response (Normal)

```
HTTP/1.1 200 OK<CR_LF>
Server: Apache-Coyote/1.1<CR_LF>
Content-Type: text/plain;charset=UTF-8<CR_LF>
Content-Length: XXXX<CR_LF>
Date: Wed, 01 Jan 2014 10:00:01 GMT<CR_LF>
X-AggregateInfo-Result: OK<CR_LF>
```

Connection timeout........................20 seconds

Web Sever response timeout.........60 seconds

◆ Data Transfer Web Sever

Use the URL you specified in "Data transfer setting" -"Data transfer URL" in CONPROSYS WEB Setting.

◆ Data File

Group	Item	Description	Notes
Header	Terminal ID	X(7)9(6)	Serial number is listed
	Reservation		Not in use
Data 1- Data n			Data items are listed with "," (comma)
Footer	Reservation		Not in use
	Transfer type	9(1)	0: Normal transfer 1: Resent

◆ Data

Data format can be specified as you wish after the column of the Date and Time.

Column	Indication	Parameter 2
1	Date and Time	Date and time of the data measured (YYYYMMDDhhmmss)
2	---	

Customer Support and Inquiry

CONTEC provides the following support services for you to use CONTEC products more efficiently and comfortably.

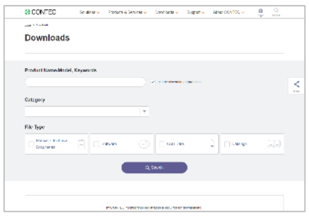
1.Services

CONTEC offers the useful information including product manuals that can be downloaded through the Contec website.

Download

https://www.contec.com/download/

You can download updated driver software, firmware, and differential manuals in several languages.　Membership registration (myCONTEC) is required to use the services.



Index

Revision History

MONTH YEAR	Summary of Changes
September 2015	The First Edition
June 2018	Changed the layout of the manual.
December 2018	Added the sections of serial setting and Modbus RTU details.

CONTEC CO., LTD. 3-9-31, Himesato, Nishiyodogawa-ku, Osaka 555-0025, Japan

https://www.contec.com/

PAC Series, Integrated Type/Configurable Type, System Setup Guide

NA04589 (LYTJ744) 12112018_rev4 [02222016] December 2018 Edition