MITSUBISHI
ELECTRIC
Changes for the Better

Programmable Controllers

MELSEC Q series

MELSEC PROCESS CONTROL
Technical Guide

CONTENTS

1 OVERVIEW

This technical guide explains the programming techniques with the creation procedure of the loop control programs and program examples using process control FBD software package (PX Developer).

1.1 Features of MELSEC Process Control

(1) Materialization of advanced process control with the MELSEC-Q series

- **Process CPU/Redundant CPU that materializes high-performance process control**
  The PLCs that materialize the substantial loop control instructions and high-speed loop operation processing, which are equivalent to dedicated controller such as DCS, enable advanced process control.

- **High function analog module that materializes analog input/output necessary for process control**
  High function analog module that is equipped such as channel isolation, high accuracy, high resolution, and alarm/disconnection detection function materializes analog input/output functions necessary for process control.

- **PX Developer that materializes simple engineering of loop control**
  Process control FBD software package PX Developer enables simple creation of loop control programs which are complex and cumbersome to create with conventional ladder language.

- **Monitoring screen with GOT screen generator function**
  Using together with the GOT1000 or SoftGOT, process control monitoring screens can be readily created.

(2) Reduction of system configuration and modification cost

- **Integrating loop control and sequence control into a single CPU**
  Both loop control and sequence control can be done with a single process CPU/redundant CPU, therefore, hardware cost reduction is possible.

- **Cost reduction of program modification with PX Developer**
  The loop controls can be modified such as adding loops with PX Developer even after operating the systems (pasting loop control FB to the programs only). Also, test run adjustment can be executed immediately with a tuning monitor screen, therefore, program modification cost reduction is possible.

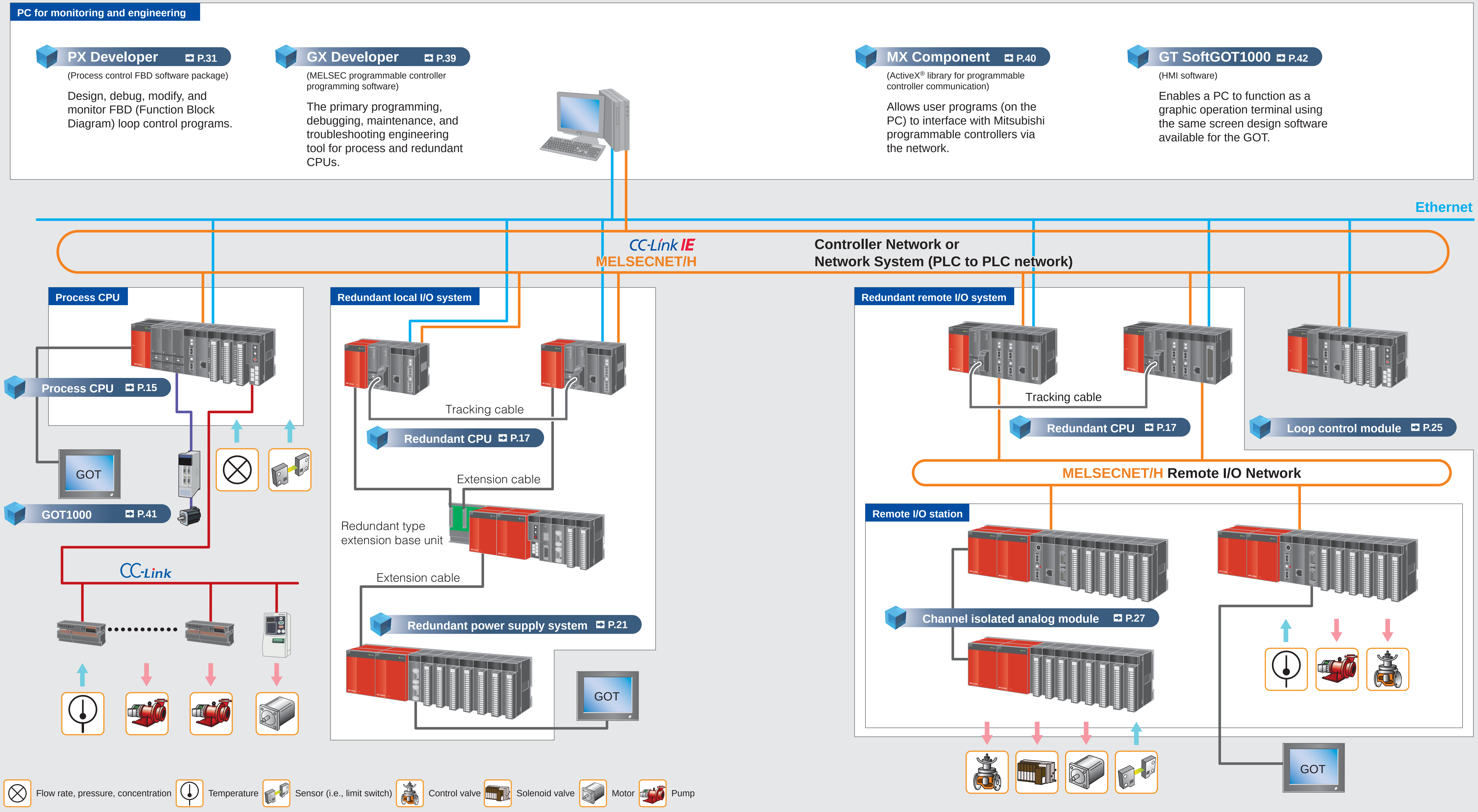
(3) Improved maintainability and reliability

- **Replacement of I/O module in online mode is possible**
  When an analog or I/O module fails, it can be replaced online without stopping or turning off the process CPU/redundant CPU. (Operation in GX Developer is required.)

- **Improving reliability with redundant system configuration**
  With the redundancy of the basic systems including the CPU module, power supply module, base unit, and network module, the standby system takes over the control to continue the system operation when the control system fails. Therefore, system reliability can be improved.

- **Alternative control in sensor failure**
  When sensors such as flow meter, manometer, and open/close limit switch of ON/OFF valve and so on fail, alternative control (to assume a normal operation) can be executed by simulated inputting the correct input signal forcibly on a faceplate of PX Developer.

Reduce costs by combining FA (Factory Automation) and PA (Process Automation) in the same platform.



PC for monitoring and engineering

PX Developer P.31
(Process control FBD software package)
Design, debug, modify, and monitor FBD (Function Block Diagram) loop control programs.

GX Developer P.39
(MELSEC programmable controller programming software)
The primary programming, debugging, maintenance, and troubleshooting engineering tool for process and redundant CPUs.

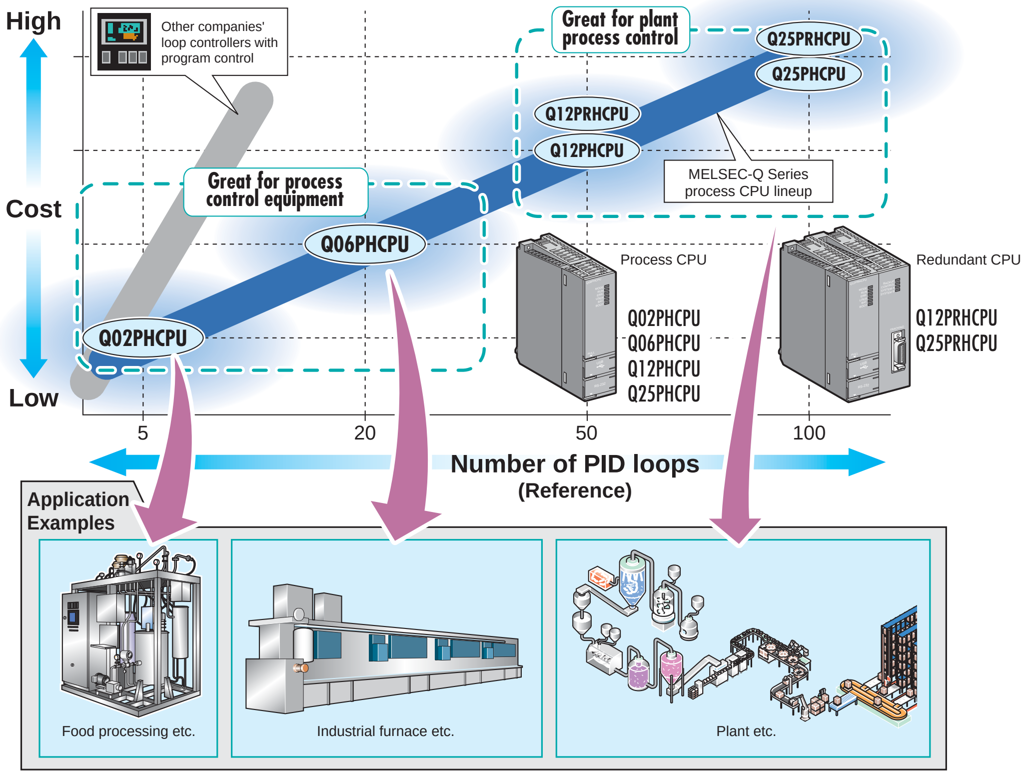
MX Component P.40
(ActiveX® library for programmable controller communication)
Allows user programs (on the PC) to interface with Mitsubishi programmable controllers via the network.

GT SoftGOT1000 P.42
(HMI software)
Enables a PC to function as a graphic operation terminal using the same screen design software available for the GOT.

⊗	Flow rate, pressure, concentration
	Temperature
	Sensor (i.e., limit switch)
	Control valve
	Solenoid valve
	Motor
	Pump

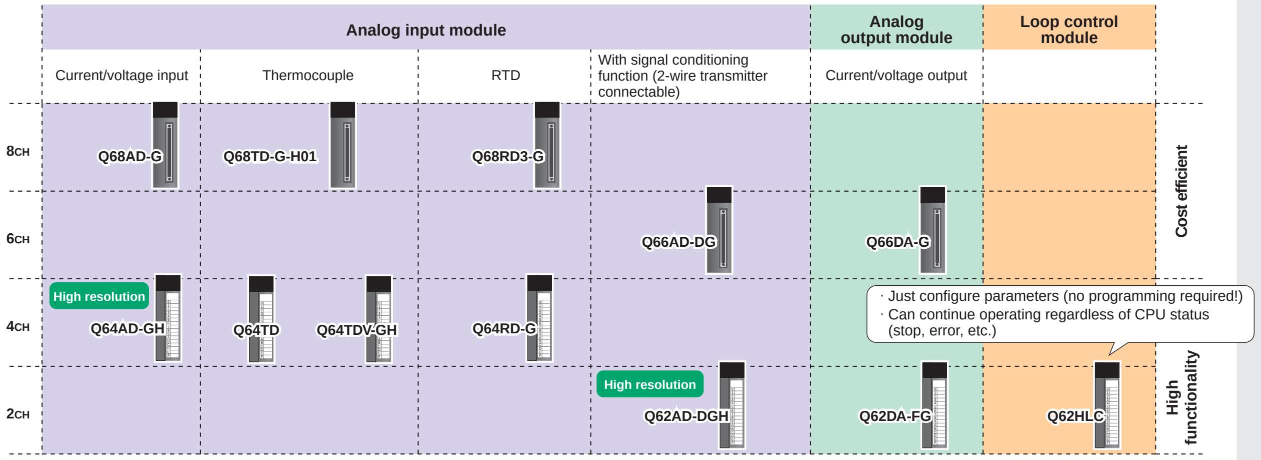
CPU (process & redundant)

Choose the optimum CPU based on the number of PID loops required and the scale of the application.



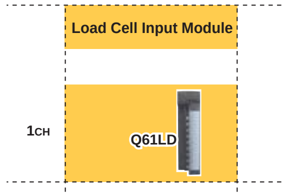
Modules best suited for loop control

Choose from a wide selection of channel isolated analog modules for loop control.



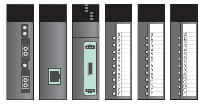
	Analog input module				Analog output module	Loop control module	
	Current/voltage input	Thermocouple	RTD	With signal conditioning function (2-wire transmitter connectable)	Current/voltage output		
8CH	Q68AD-G	Q68TD-G-H01	Q68RD3-G				Cost efficient
6CH				Q66AD-DG	Q66DA-G		
4CH	High resolution Q64AD-GH	Q64TD Q64TDV-GH	Q64RD-G				
2CH				High resolution Q62AD-DGH	Q62DA-FG	Q62HLC	High functionality

- Just configure parameters (no programming required!)
- Can continue operating regardless of CPU status (stop, error, etc.)



	Load Cell Input Module
1CH	Q61LD

MELSEC-Q Series modules



Q Series I/O modules already being used in other systems can be used with process and redundant CPUs, thereby reducing the number of required spare parts and lowering maintenance costs.

*For detailed information about each module, please refer to the appropriate user's manual.

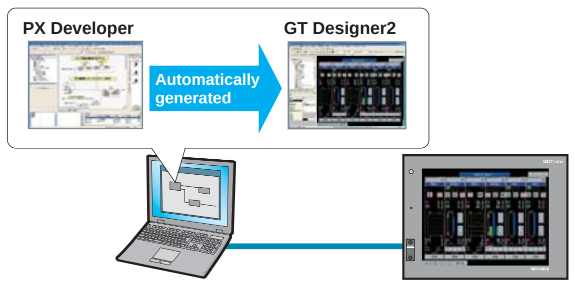
Process monitoring solutions

Choose a control and monitoring solution that is right for the situation.

Category	Application	Solution	Feature
HMI based	General monitoring and control	GOT1000 Series **GOT screen generation function** 	· Excellent environmental resistance · Compatible with high resolution (15" XGA) · Create process control and monitoring screens for the GOT1000 series automatically
PC based		Combination of PX Developer Monitor tool and SoftGOT **SoftGOT interface** 	· Use GT Designer2 or GT Designer3 to create GOT screens for the PC · Graphically represent operations on the shop floor to aid the quick understanding of system status
	Complex monitoring and control	Commercial SCADA **SCADA system interface** 	· Better flexibility and range of functionality · Links to enterprise system

GOT screen generation function

GOT screens for monitoring, tuning, etc. can be automatically generated from programs created using PX Developer. This feature eliminates the time consuming tasks of assigning devices and programming GOT screens to substantially reduce development time. See page 37 for details.



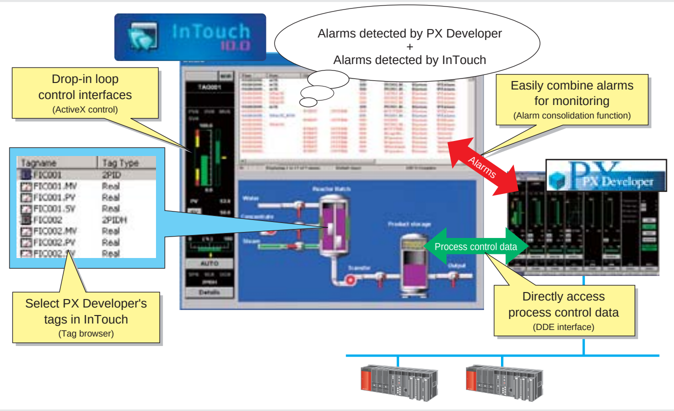
SoftGOT interface

PX Developer monitoring functions can be 'called' or opened directly from GT SoftGOT1000 screens. Consequently, the development time for creating GOT screens can be reduced.

SCADA system interface

SCADA systems such as Wonderware InTouch by Invensys Systems, Inc. can be used to create advanced graphical displays of the system status. The PX Developer monitoring tool is designed to interface with SCADA software and its monitoring functions may be called and opened directly by the SCADA software.

The company and product names above are trademarked by their respective companies.



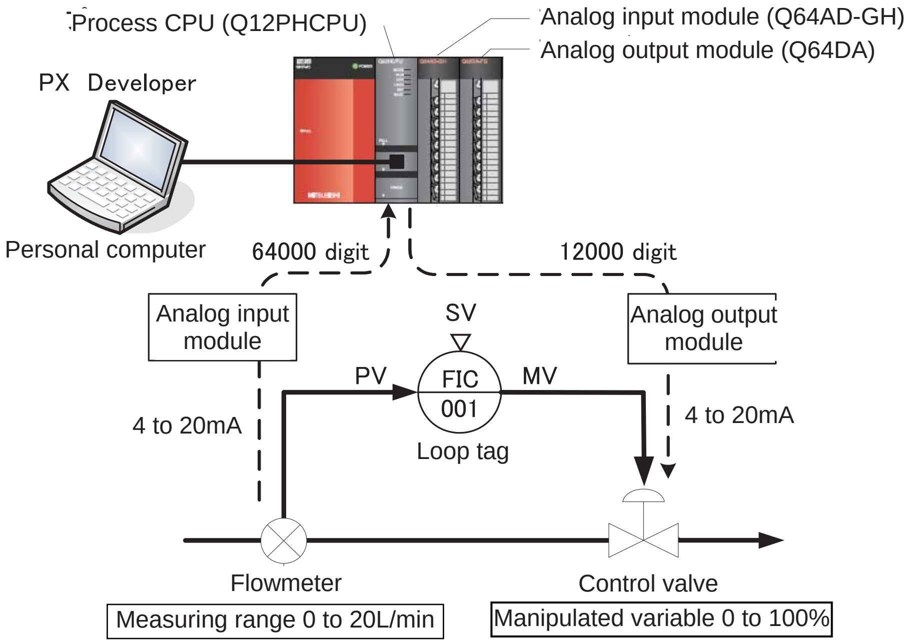
Tagname	Tag Type
FIC001	2PID
FIC001.MV	Real
FIC001.PV	Real
FIC001.SV	Real
FIC002	2PIDH
FIC002.MV	Real
FIC002.PV	Real
FIC002.SV	Real

2. CREATING LOOP CONTROL PROGRAMS AND OPERATION CHECK

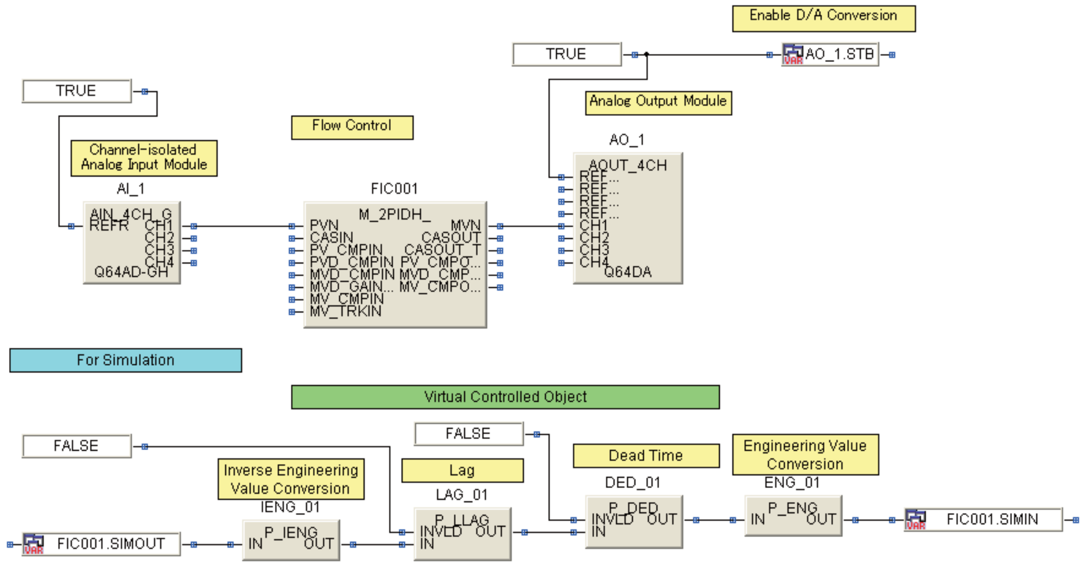
This chapter explains a procedure of creating loop control programs and operation check with process CPU + PX Developer, showing simple examples.

2.1 Creating Loop in PX Developer

Create a loop control with system configuration shown below as an example.



The following figure is a program which is corresponding to system configuration shown above. This program imports PV (process variable) to tag FB (FIC001) from module FB (AI_1) which corresponds to analog input module (Q64AD-GH), executes PID control, and outputs the result to module FB (AO_1) which corresponds to analog output module (Q64DA).



In this example, use 2-degree-of-freedom advanced PID control tag FB (M_2PIDH_T_) which optimizes both the control of disturbance response and target tracking as a loop tag.

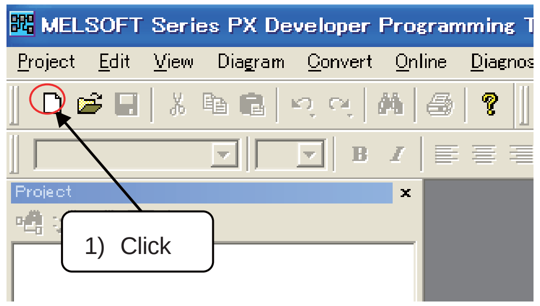
2.1.1 Flow chart of creating advanced PID control loop, operation check, and operation monitoring

This section explains a procedure of creating a project including PID loop adjustment.

1) Creating a project	Create a project and start programming in PX Developer.
2) Registering module FB corresponding to analog module	Register analog modules to be used on the module FB declaration window.
3) Registering tag FBs for loop control	Register tag FBs for loop control on the tag FB declaration window.
4) Creating a loop control program	Paste module FB, tag FB registered in 2), 3) on a FBD sheet and connect together.
5) Setting loop control parameters	Set parameters of the tag FB on the FB property window.
6) Creating simulation program	Add input/output data loopback program required to simulate the PID loop control.
7) Compiling program and download to process CPU	Compile the created program and download the PLC parameter, program, data (default) to process CPU
8) Monitoring PID loop with faceplate	Monitor PID loop with faceplate (controller panel) of PX Developer.
9) PID loop adjustment with tuning screen (Monitor Tool)	Start Monitor Tool and adjust PID parameters with tuning screen.

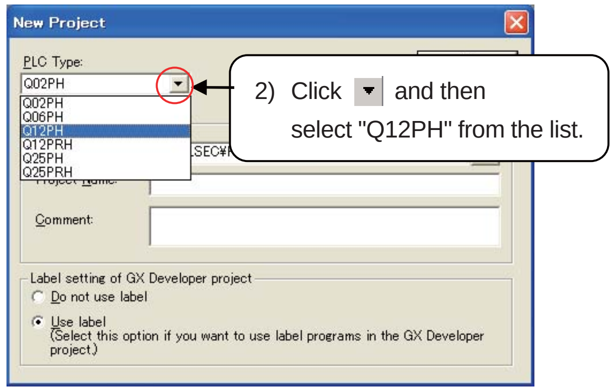
2.1.2 Creating new projects

A project should be created before programming with the programming tool.
This section explains how to create a new project.



1) Click the new project button 🗋 on the toolbar. (Can also be performed by selecting Menu [Project] → [New Project]).

↓

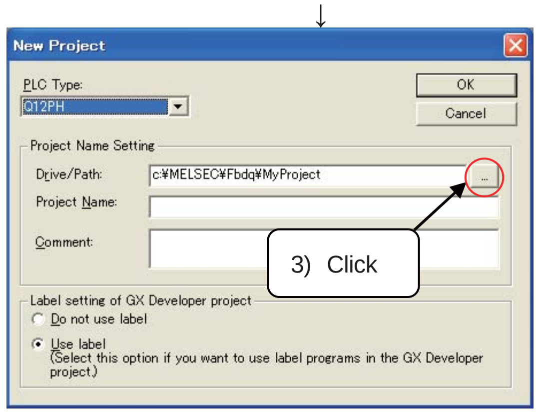


PLC Type: Q02PH
Q02PH
Q06PH
Q12PH
Q12PRH
Q25PH
Q25PRH

2) Click ▼ and then select "Q12PH" from the list.

2) Click ▼ to the right of PLC type and select "Q12PH" on the "New Project" dialog box. (Select PLC type in accordance with the PLC CPU type to be used.)

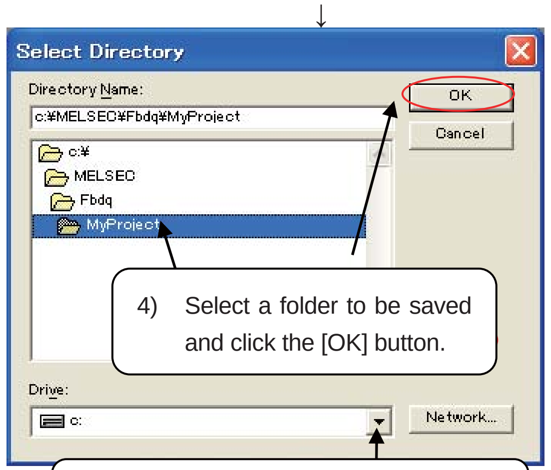
↓



PLC Type: Q12PH

Drive/Path: c:¥MELSEC¥Fbdq¥MyProject

3) Click ... to the right of the Driver/Path text box.



Directory Name: c:¥MELSEC¥Fbdq¥MyProject

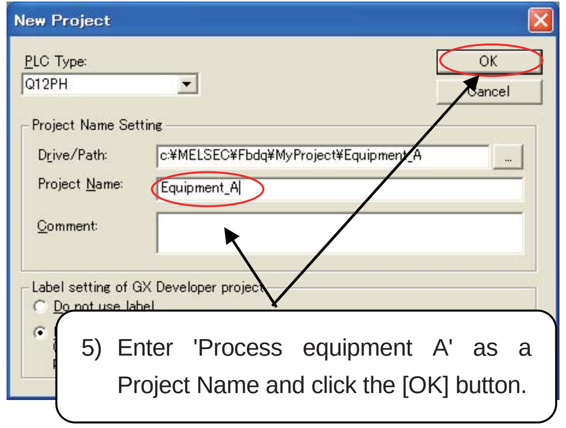
c:¥
MELSEC
Fbdq
MyProject

Drive: c:

4) Select a folder to be saved and click the [OK] button.

Click ▼ and then select the drive when changing the drive.

4) The "Select Directory" dialog box is displayed. Click ▼ to the right of Drive and select the drive when changing the drive.
To change a save destination folder, operate a folder tree in the middle of the dialog box, select a folder for saving the project, and then click the [OK] button.



PLC Type: Q12PH

Project Name Setting

Drive/Path: c:¥MELSEC¥Fbdq¥MyProject¥Equipment_A

Project Name: Equipment_A

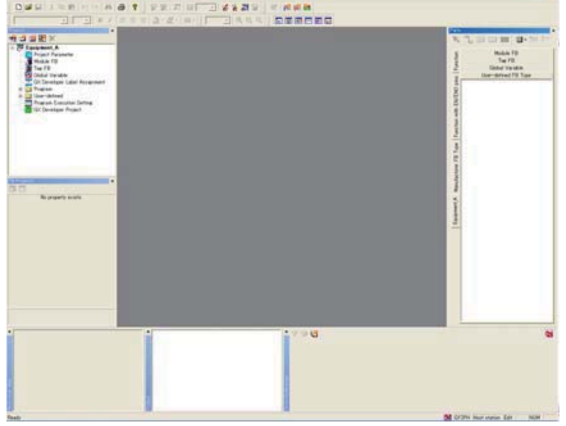
Comment:

Label setting of GX Developer project

○ Do not use label

5) Enter 'Process equipment A' as a Project Name and click the [OK] button.

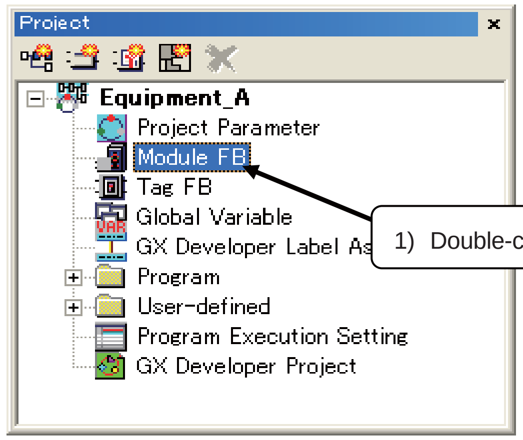
5) Enter 'Equipment A' in Project Name text box, and click the [OK] button.



6) "Equipment A" project is created.

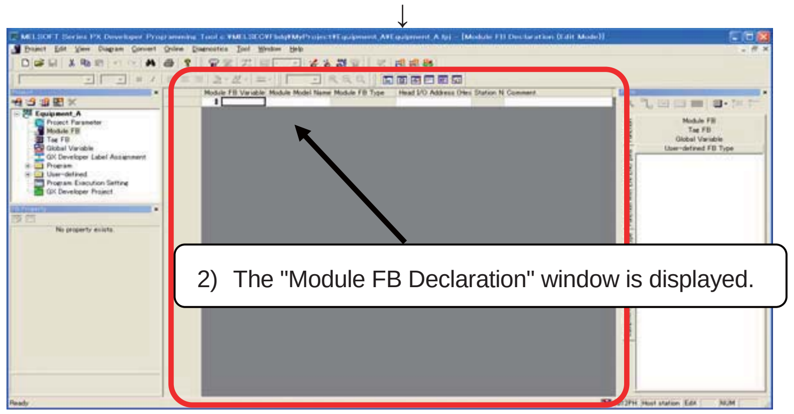
2.1.3 Registering Module FB corresponding to analog module

Register analog modules to be used on the "Module FB Declaration" window.



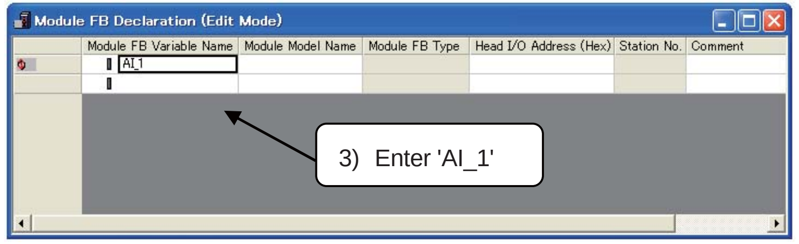
1) Double-click

1) Double-click Module FB on the "Project" window.



2) The "Module FB Declaration" window is displayed.

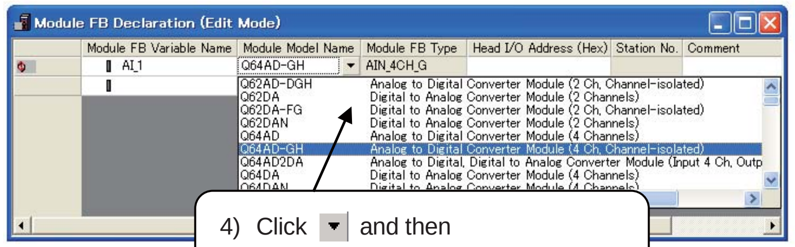
2) The "Module FB Declaration" window is displayed.



Module FB Variable Name	Module Model Name	Module FB Type	Head I/O Address (Hex)	Station No.	Comment
AI_1					

3) Enter 'AI_1'

3) Enter Module FB Variable Name corresponding to analog input module.
Enter 'AI_1' as Module FB Variable Name as an example.

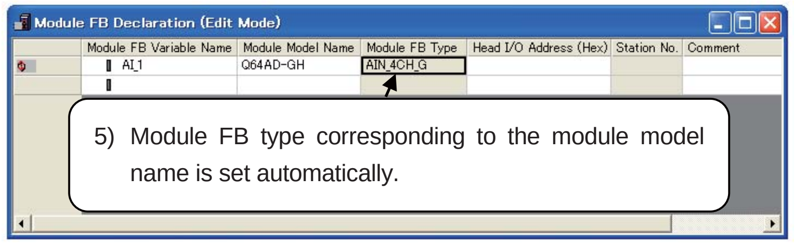


Module FB Variable Name	Module Model Name	Module FB Type	Head I/O Address (Hex)	Station No.	Comment
AI_1	Q64AD-GH	AIN_4CH_G			

Q62AD-DGH	Analog to Digital Converter Module (2 Ch, Channel-isolated)
Q62DA	Digital to Analog Converter Module (2 Channels)
Q62DA-FG	Digital to Analog Converter Module (2 Ch, Channel-isolated)
Q62DAN	Digital to Analog Converter Module (2 Channels)
Q64AD	Analog to Digital Converter Module (4 Channels)
Q64AD-GH	Analog to Digital Converter Module (4 Ch, Channel-isolated)
Q64AD2DA	Analog to Digital, Digital to Analog Converter Module (Input 4 Ch, Outp
Q64DA	Digital to Analog Converter Module (4 Channels)

4) Click ▼ and then select "Q64AD-GH" from the list.

4) Select Module Model Name. Selecting a "Module Model Name" cell displays ▼ to the right of the cell. Click ▼ to display Module Model Name select list and select "Q64AD-GH".



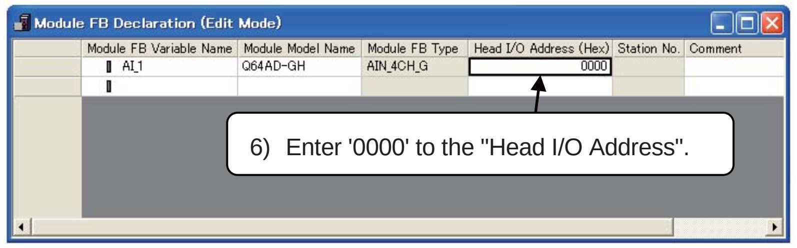
Module FB Declaration (Edit Mode)

Module FB Variable Name	Module Model Name	Module FB Type	Head I/O Address (Hex)	Station No.	Comment
AI_1	Q64AD-GH	AIN_4CH_G			

5) Module FB type corresponding to the module model name is set automatically.

5) "Module FB Type" is automatically set when module model name is selected.
(Module FB Type cannot be edited.)

↓



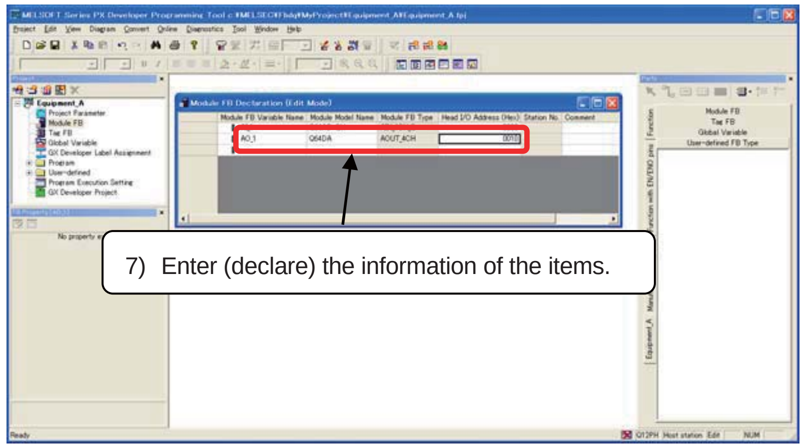
Module FB Declaration (Edit Mode)

Module FB Variable Name	Module Model Name	Module FB Type	Head I/O Address (Hex)	Station No.	Comment
AI_1	Q64AD-GH	AIN_4CH_G	0000		

6) Enter '0000' to the "Head I/O Address".

6) Input the head I/O address of an input target module with 4-bit hexadecimal number in the "Head I/O Address (Hex)
Input "0000' as an example.

↓

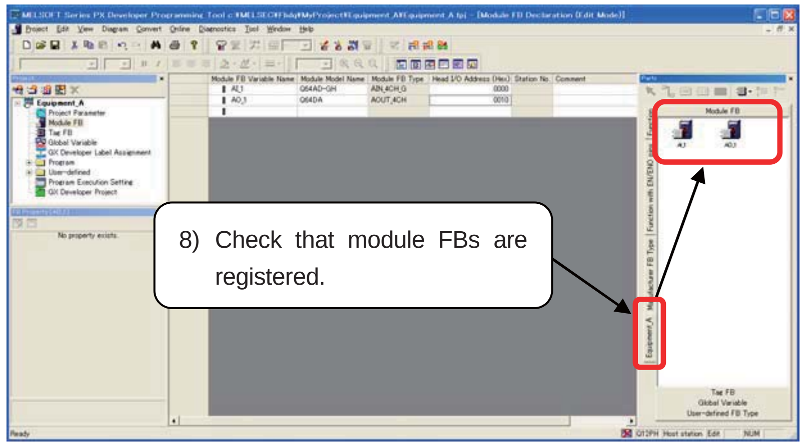


7) Enter (declare) the information of the items.

7) Declare a module FB corresponding to analog output module as follows with same operation as shown in 3) to 6).

Module FB Variable Name:"AO_1"
Module Model Name :"Q64DA"
Head I/O Address :"0010"

↓



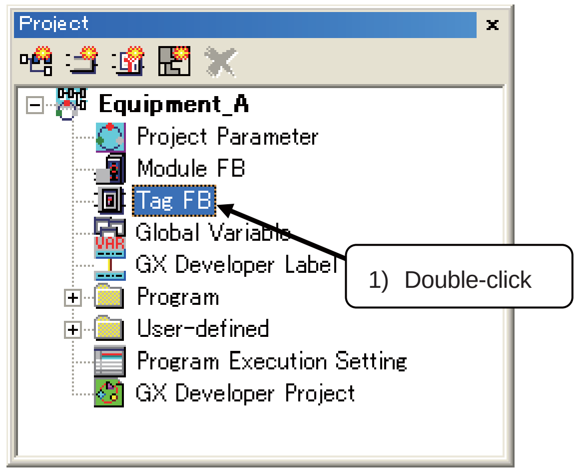
8) Check that module FBs are registered.

8) Select the same tab name as project name "Equipment A" on the "Parts" window and select <<Module FB>> tab.
Check that the module FBs are registered.

2.1.4 Registering tag FBs for loop control

Register tag FBs for loop control on the Tag FB Declaration window

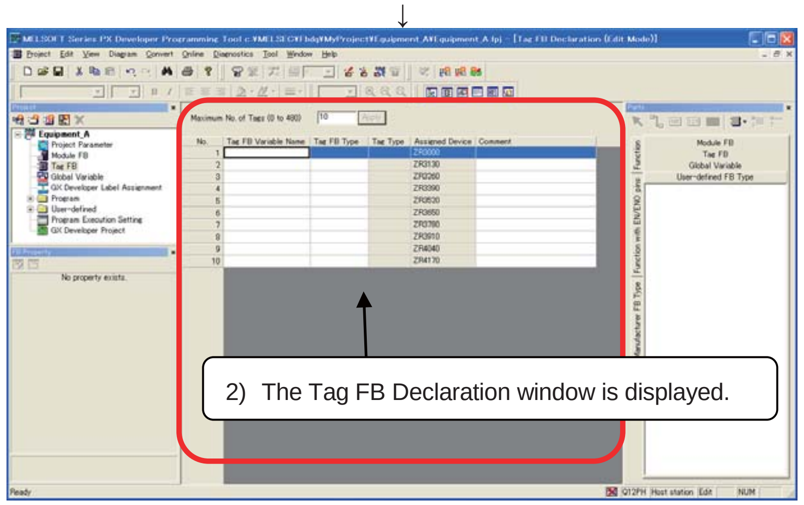
Tag FB is a FB which has a process control function such as a controller and indicator. Tag FB has data for execution of process control (such as PV, SV, MV, P, I, D, PV high/low limit value), is composed 130 word data per 1 tag. (The start device for each tag is displayed on an Assigned Device cell of Tag FB Declaration window.)



1) Double-click

1) Double-click Tag FB on the "Project" window.

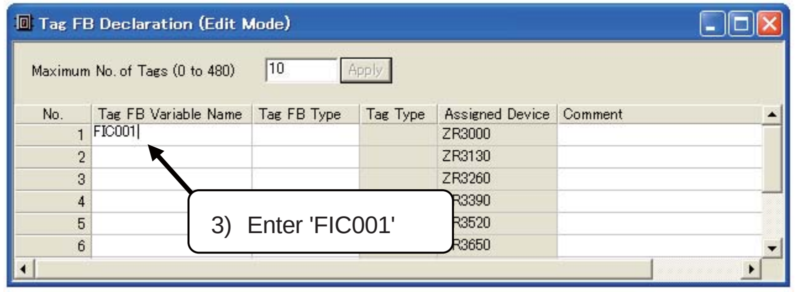
↓



2) The Tag FB Declaration window is displayed.

2) The Tag FB Declaration window is displayed.

↓



Tag FB Declaration (Edit Mode)

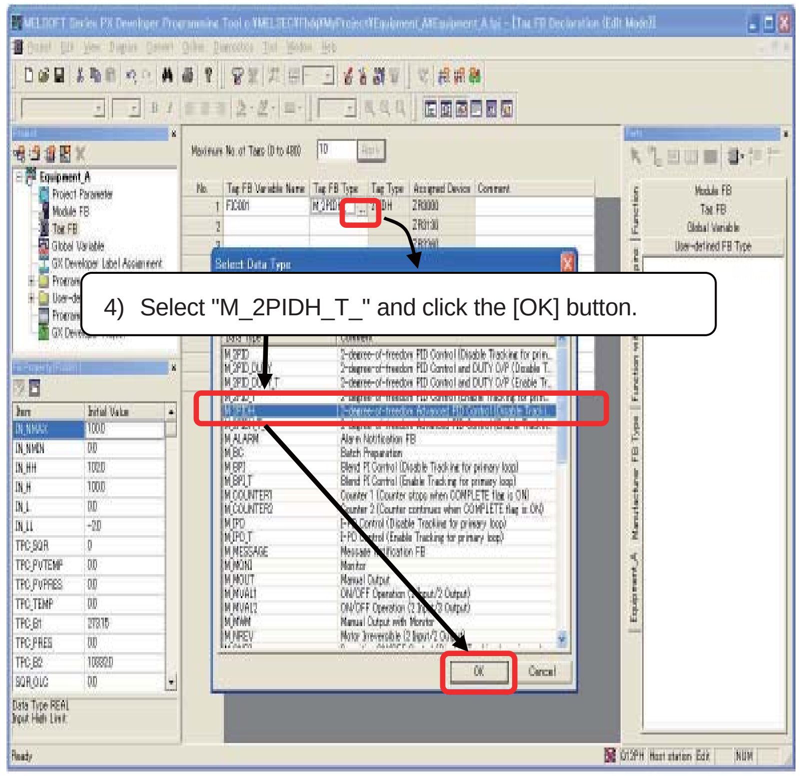
Maximum No. of Tags (0 to 480) 10 Apply

No.	Tag FB Variable Name	Tag FB Type	Tag Type	Assigned Device	Comment
1	FIC001			ZR3000	
2				ZR3130	
3				ZR3260	
4				R3390	
5				R3520	
6				R3650	

3) Enter 'FIC001'

3) Enter tag name 'FIC001' in a Tag FB Variable Name cell.

↓

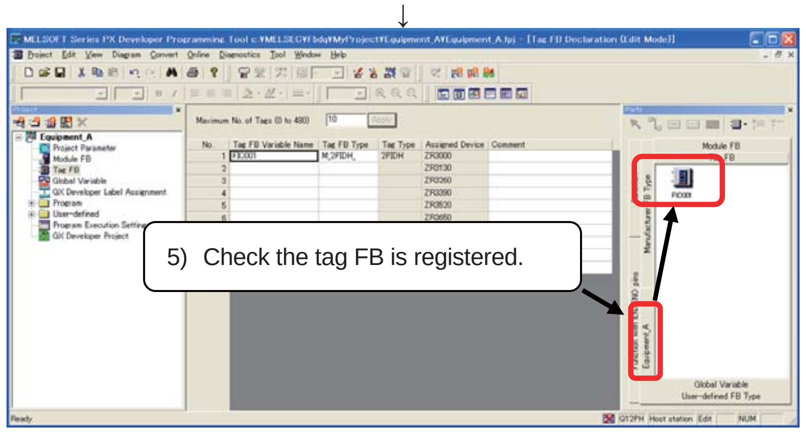


4) Select "M_2PIDH_T_" and click the [OK] button.

4) Selecting a "Tag FB Type" cell displays ... to the right of the cell. Clicking ... displays the "Select Data Type" dialog. Select a data type to be used.

Select "M_2PIDH_T_" (2-degree-of-freedom advanced PID control) as an example and click the [OK] button.

↓

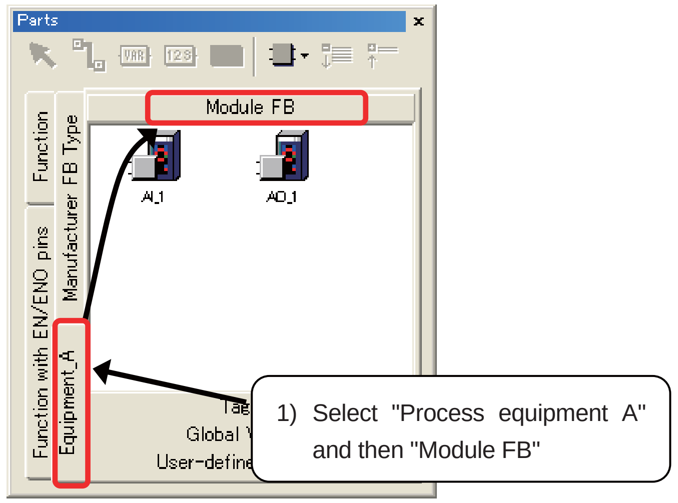


5) Check the tag FB is registered.

5) Select the same tab name as the project name "Equipment A" on the "Parts" window, and then select <<Tag FB>> tab. Check the declared tag FB is registered.

2.1.5 Creating loop control programs

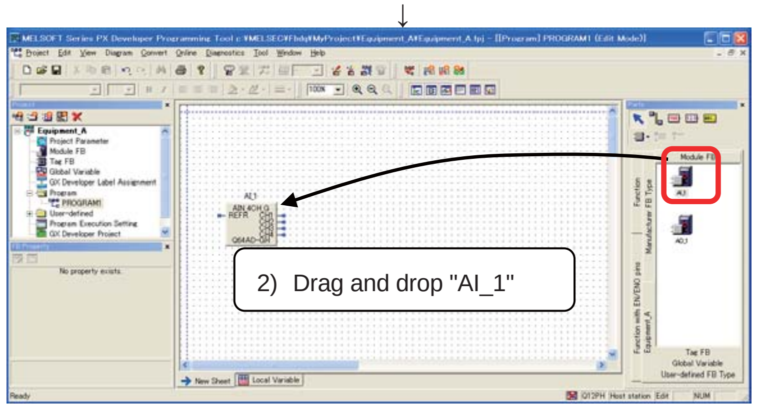
Paste the declared module FBs, tag FB on the FBD sheet and connect them.



1) Select "Process equipment A" and then "Module FB"

1) Select the same tab name as the project name "Equipment A" on the "Parts" window, and then select <<Module FB>> tab.

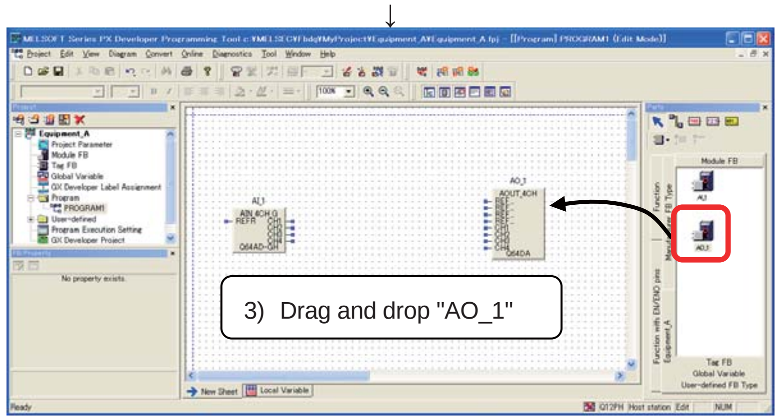
↓



2) Drag and drop "AI_1"

2) Drag and drop the icon "AI_1" of declared module FB from the <<Module FB>> tab selected in 1) to the program definition window to paste.

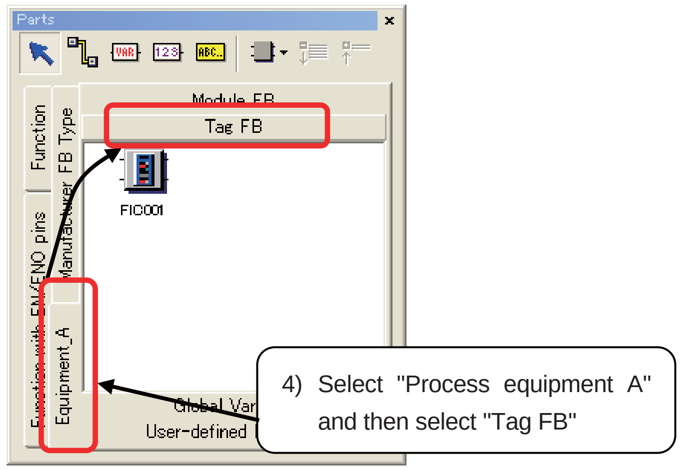
↓



3) Drag and drop "AO_1"

3) Drag and drop the icon "AO_1" of declared module FB to the program definition window to paste as shown 2).

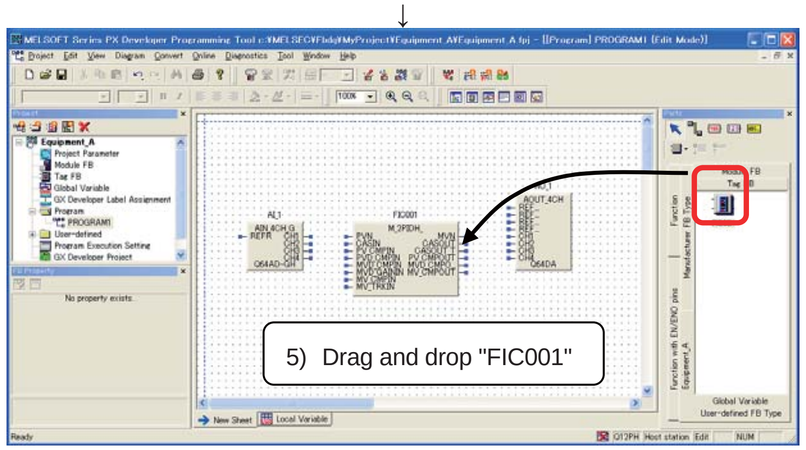
↓



4) Select "Process equipment A" and then select "Tag FB"

4) Select the same tab name as the project name "Equipment A" on the "Parts" window, and then select <<Tag FB>> tab.

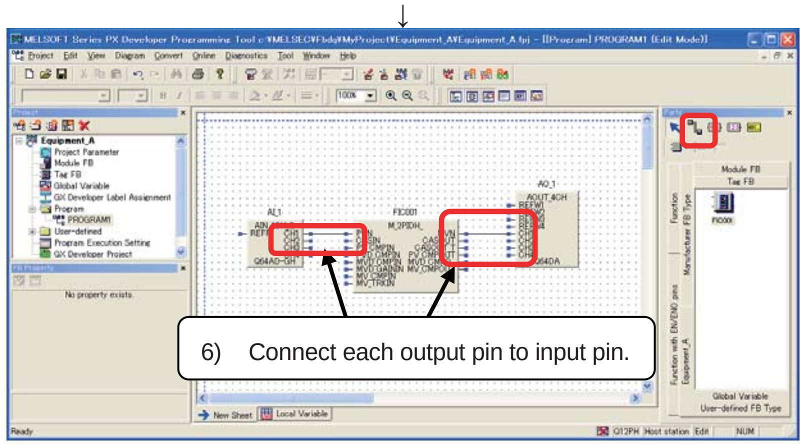
↓



5) Drag and drop "FIC001"

5) Drag and drop the icon " FIC001" on the <<Tag FB>> tab selected in 4) to the program definition window to paste.

↓

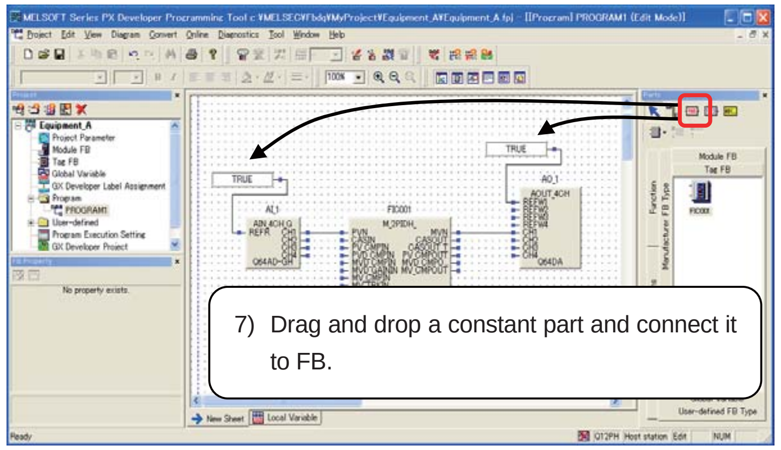


6) Connect each output pin to input pin.

6) Click the Connector button on the top of the "Parts" window, and then connect output pins of each FB to the following input pin.

In this example, connect "CH1" pin of module FB "AI_1" to "PVN" pin of tag FB "FIC001", "MVN" pin of tag FB "FIC001" to "CH1" pin of module FB "AO_1".

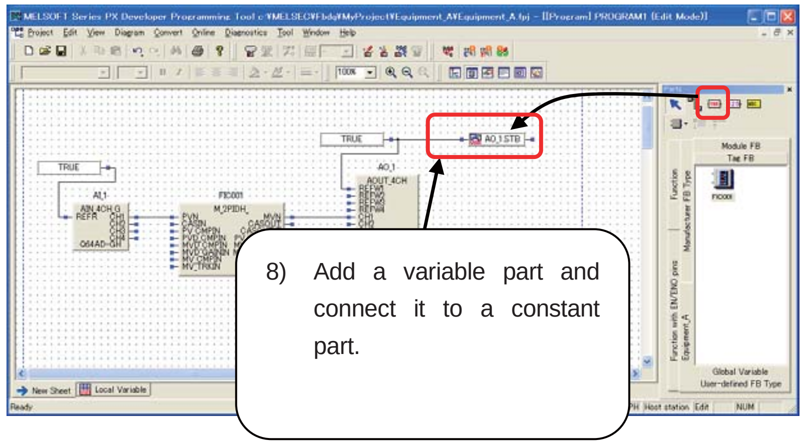
↓



7) Drag and drop a constant part and connect it to FB.

7) Drag and drop a constant part on the "Parts" window to the program definition window twice and enter "TRUE" ("True" value).
Then, connect each of 2 output pins of constant part to "REFR" pin of module FB "AI_1", "REFW1" pin of module FB "AO_1".

* Setting REFW1 pin TRUE enters digital value from CH1 of module FB "AO_1". Each channel has REFW pin.

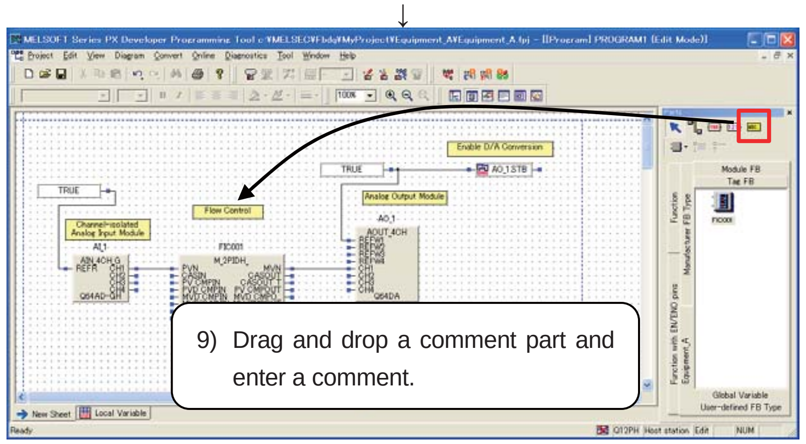


8) Add a variable part and connect it to a constant part.

8) Add a procedure which enables D/A conversion of module FB "AO_1". Paste a variable part on the "Parts" window and connect it as shown in the left figure.

* This variable is a public variable of module FB "AO_1". Therefore, pasting variable part, double-clicking it, and then entering 'AO_1.STB' registers a variable name.

↓



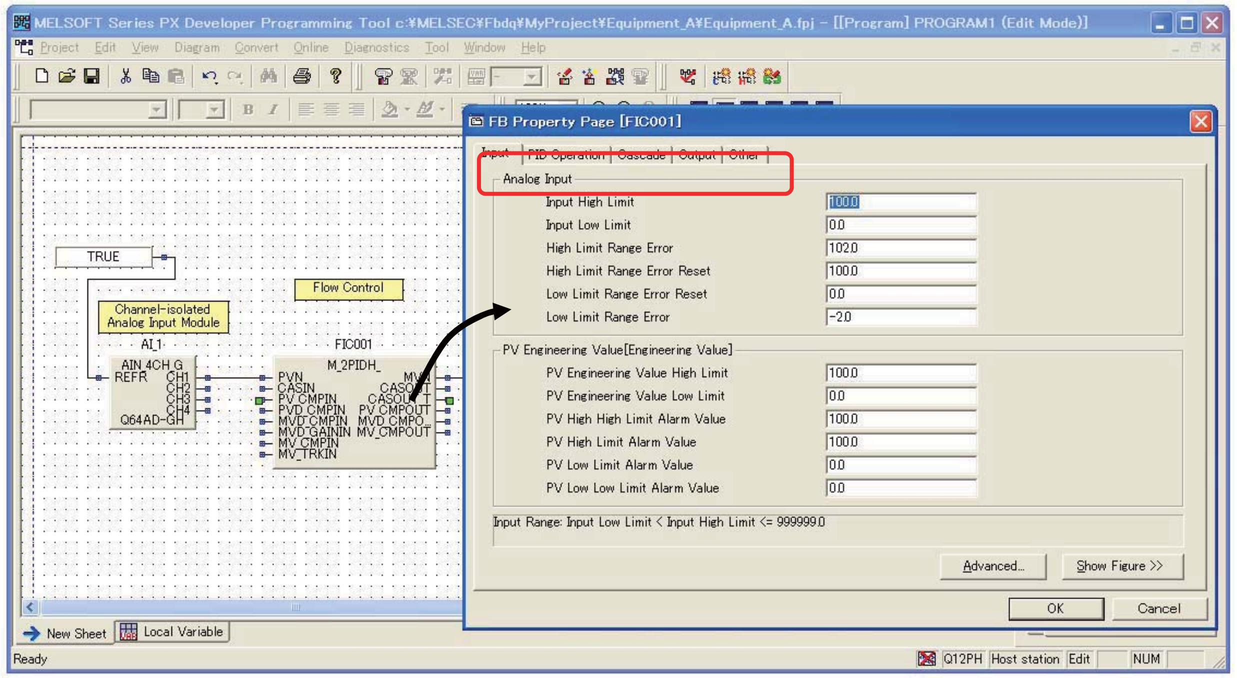
9) Drag and drop a comment part and enter a comment.

9) Drag and drop comment parts and enter comments if necessary.

2.1.6 Setting loop control parameters

Set the parameters of a tag FB on the FB property window.

Basic parameter settings of tag FB are set on the FB property page.
The FB property page is displayed by selecting FB property page on the pop-up menu, which is displayed by right-clicking a tag FB on a program.
The default parameters have been set; therefore, change them in accordance with the systems. In this example, set the minimum requirements for loop control.



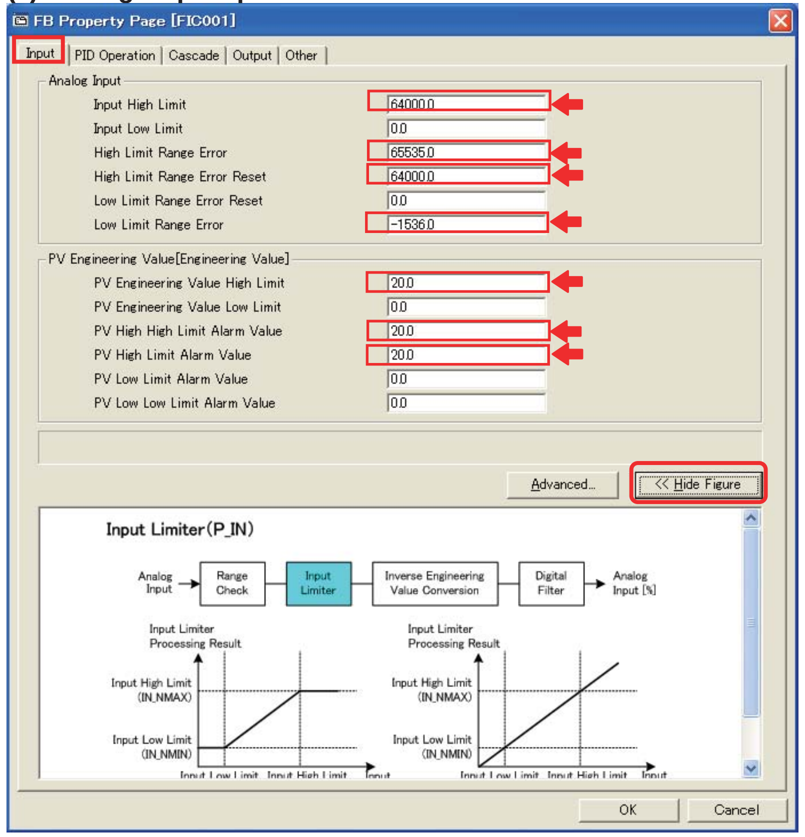
Analog Input
Input High Limit 100.0
Input Low Limit 0.0
High Limit Range Error 102.0
High Limit Range Error Reset 100.0
Low Limit Range Error Reset 0.0
Low Limit Range Error -2.0

PV Engineering Value[Engineering Value]
PV Engineering Value High Limit 100.0
PV Engineering Value Low Limit 0.0
PV High High Limit Alarm Value 100.0
PV High Limit Alarm Value 100.0
PV Low Limit Alarm Value 0.0
PV Low Low Limit Alarm Value 0.0

Input Range: Input Low Limit < Input High Limit <= 999999.0

The following explains the parameter setting data of "Input" "PID Operation" "Output".
The example shown above uses the default settings of parameter except "Input" "PID Operation" "Output". For details of each parameter, refer to Appendix 1.

(1) Setting "Input" parameter



Analog Input
Input High Limit 64000.0
Input Low Limit 0.0
High Limit Range Error 65535.0
High Limit Range Error Reset 64000.0
Low Limit Range Error Reset 0.0
Low Limit Range Error -1536.0

PV Engineering Value[Engineering Value]
PV Engineering Value High Limit 20.0
PV Engineering Value Low Limit 0.0
PV High High Limit Alarm Value 20.0
PV High Limit Alarm Value 20.0
PV Low Limit Alarm Value 0.0
PV Low Low Limit Alarm Value 0.0

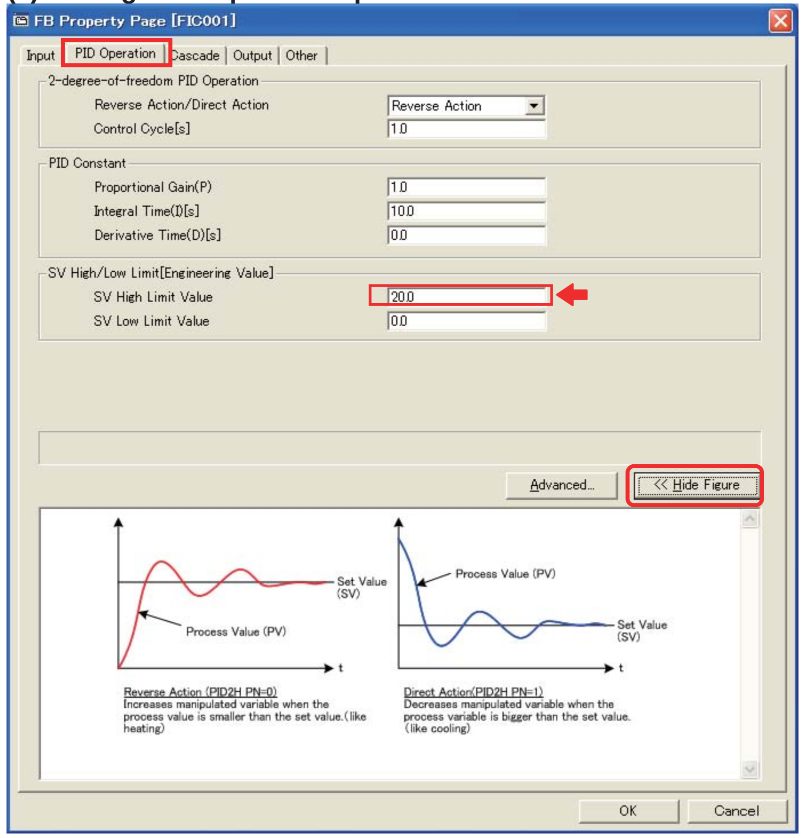
1) Set the input parameter in accordance with each analog module. In this example, change items indicated ← in the figure to the left as the table below.
Items other than them remain default settings.

Clicking [Show Figure >>] button displays the detail description of the parameters.
(Figure to the left is the status that the detail description is displayed.
Clicking [<< Hide Figure] button hides the detail description.)

Group	Item (Variable name)	Contents	Setting value indicates ← the value changed
Analog Input	Input High Limit (IN_NMAX)	Set high limit value for the range of A/D conversion values (such as 0 to 4000, 0 to 8000) input from an analog input module.	64000.0
	Input Low Limit (IN_NMIN)	Set low limit value for the range of A/D conversion values (such as 0 to 4000, 0 to 8000) input from an analog input module.	0.0
	High Limit Range Error (IN_HH)	Set reference value of high limit exceeding error (range high limit error) for A/D conversion values input from an analog input module.	65535.0
	High Limit Range Error Reset (IN_H)	Set reference value of error reset performed after high limit range error occurrence.	64000.0
	Low Limit Range Error Reset (IN_L)	Set reference value of error reset performed after low limit range error occurrence.	0.0
	Low Limit Range Error (IN_LL)	Set reference value of low limit exceeding error (range low limit error) for A/D conversion values input from an analog input module.	-1536.0
PV Engineering variable [Engineering variable]	PV Engineering variable High Limit (RH)	Set high limit value for using A/D conversion value inputs from an analog input module as PV engineering variables.	20.0
	PV Engineering variable Low Limit (RL)	Set low limit value for using A/D conversion value inputs from an analog input module as PV engineering variables.	0.0
	PV High High Limit Alarm Value (HH)	Set reference value of high high limit exceeding alarm for PV engineering variable.	20.0
	PV High Limit Alarm Value (PH)	Set reference value of high limit exceeding alarm for PV engineering variable.	20.0

	PV Low Limit Alarm Value (PL)	Set reference value of low limit exceeding alarm for PV engineering value.	0.0
	PV Low Low Limit Alarm Value(LL)	Set reference value of low low limit exceeding alarm for PV engineering value.	0.0

(2) Setting "PID Operation" parameter


FB Property Page [FIC001]

2-degree-of-freedom PID Operation
Reverse Action/Direct Action: Reverse Action
Control Cycle[s]: 1.0

PID Constant
Proportional Gain(P): 1.0
Integral Time(I)[s]: 10.0
Derivative Time(D)[s]: 0.0

SV High/Low Limit[Engineering Value]
SV High Limit Value: 20.0
SV Low Limit Value: 0.0

Reverse Action (PID2H PN=0)
Increases manipulated variable when the process value is smaller than the set value.(like heating)

Direct Action(PID2H PN=1)
Decreases manipulated variable when the process variable is bigger than the set value.(like cooling)

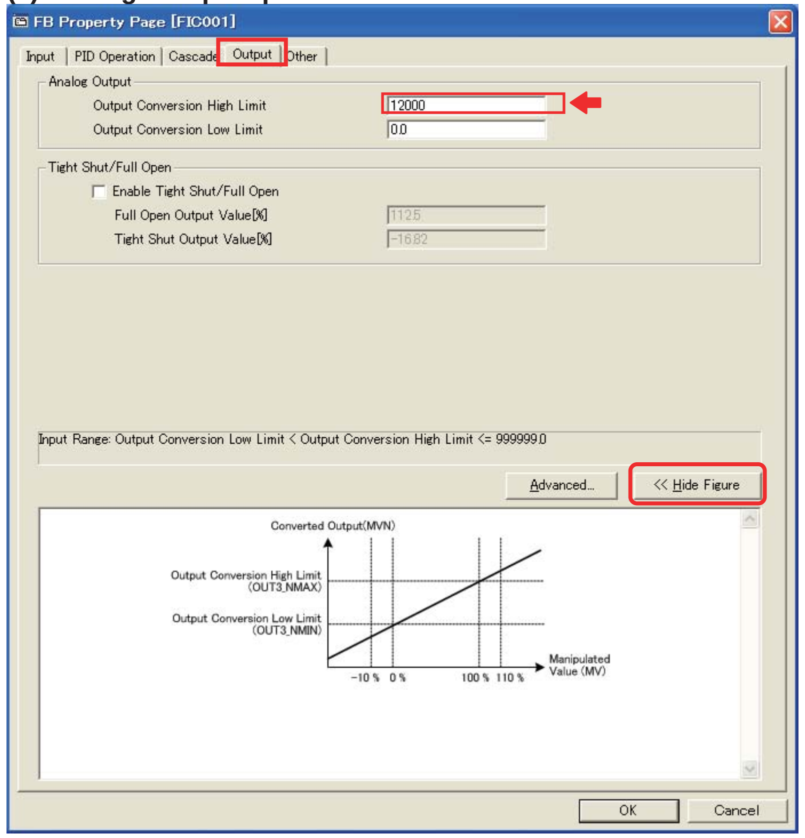
2) Set PID operation parameters in accordance with systems to be used. In this example, change the items indicated ← in the figure to the left as the table below. Items other than them remain default settings.

Clicking [Show Figure >>] button displays the detail description of the parameter.
(Figure to the left is the status that the detail description is displayed.
Clicking [<< Hide Figure] button hides the detail description.)

Group	Item (Variable name)	Contents	Setting value ← indicates the value changed
2-degree-of-freedom PID operation	Reverse action/ direct action (PID2H_PN)	Reverse action which increases the manipulated variable (MV) when the process variable (PV) decreases more than the setting value (SV). Direct action which increases the manipulated variable (MV) when the process variable (PV) increases more than the setting value (SV).	Reverse action
	Control cycle [second] (CT)	Indicate PID operation cycle and set the time (second) that is the integral number multiple of execution cycle T (the default is 200ms in the execution cycle of FBD program).	1.0
PID constant	Proportional gain (P)	Set the proportional gain in P operation. Proportional gain equals 100/proportional band (%).	1.0
	Integral time (I) [second]	Set the integral time in I operation.	10.0
	Derivative time (D) [second]	Set the derivative time in D operation.	0.0
SV high/low limit	SV high limit value (SH)	Set the high limit value of high/low limiter processing to SV (target).	20.0
	SV low limit value (SL)	Set the low limit value of high/low limiter processing to SV (target).	0.0

---

(3) Setting "Output" parameter


FB Property Page [FIC001]

Analog Output
Output Conversion High Limit: 12000
Output Conversion Low Limit: 0.0

Tight Shut/Full Open
☐ Enable Tight Shut/Full Open
Full Open Output Value[%]: 112.5
Tight Shut Output Value[%]: -16.82

Input Range: Output Conversion Low Limit < Output Conversion High Limit <= 999999.0

3) Set the output parameter in accordance with each analog output. In this example, change items indicated ← in the figure to the left as the table below.
Items other than them remain default settings.

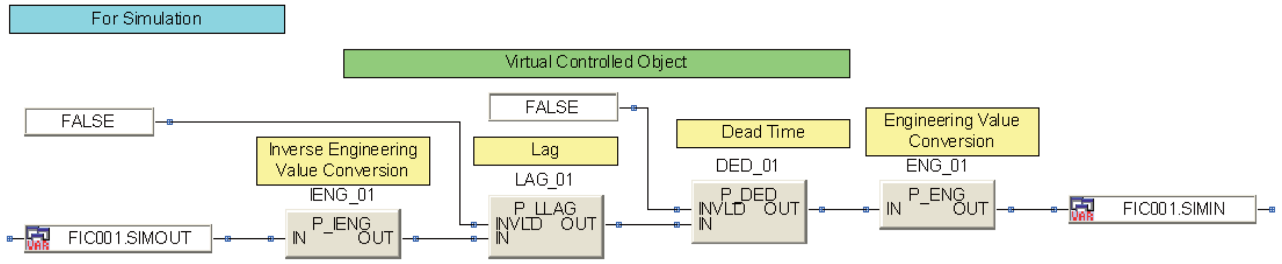
Clicking [Show Figure >>] button displays the detail description of the parameter.
(Figure to the left is the status that the detail description is displayed.
Clicking [<< Hide Figure] button hides the detail description.)

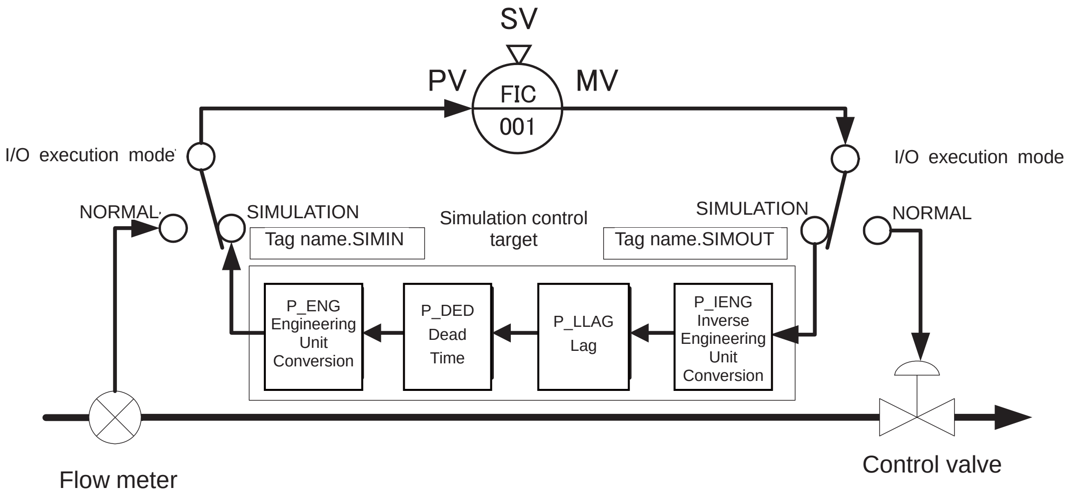
Group	Item (Variable name)	Contents	Setting value ← indicates the value changed
Analog output	Output conversion high limit (OUT3_NMAX)	Set the high limit value for the range of D/A conversion values (such as 0 to 4000, 0 to 8000) for writing to an analog output module.	12000.0
	Output conversion low limit (OUT3_NMIN)	Set the low limit value for the range of D/A conversion values (such as 0 to 4000, 0 to 8000) for writing to an analog output module.	0.0

2.1.7 Creating simulation programs

Add input/output data loopback programs required to simulate the PID loop control.

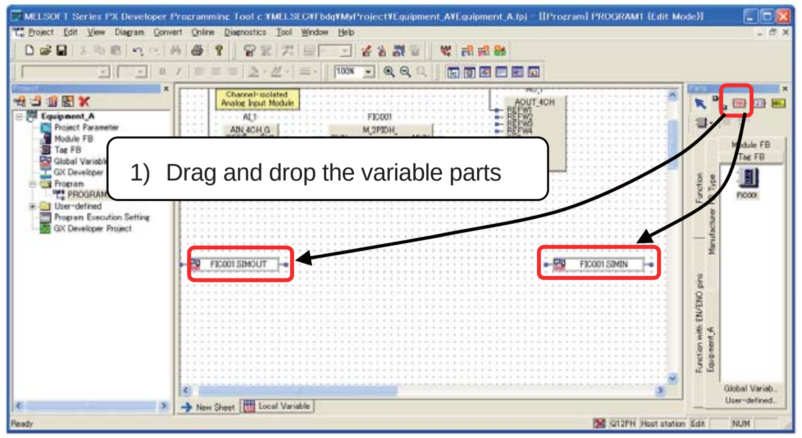
The simulation program is used for PID loop control operation check without connecting to the control devices such as sensors and control valves.
An example of simulation program is shown below. This processing is for returning MV output of tag FB "FIC001" to PV input of tag FB "FIC001" in the program.
First of all, convert simulation output "FIC001.SIMOUT" of tag FB "FIC001" to percentage with P_IENG". Then, simulate lag time caused by response of control target with "P_LLAG" and dead time caused by response of control target with "P_DED". Then convert input data (%) to input data of tag with "P_ENG", and enter it to simulation input "FIC001.SIMIN" of tag FB "FIC001".





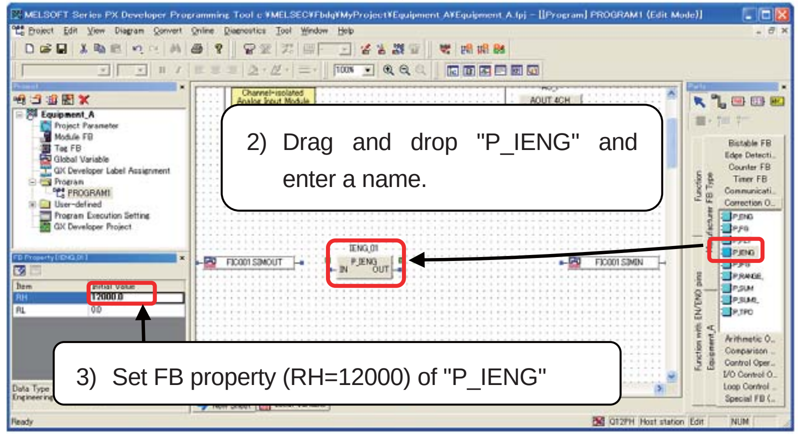
Loop

Switching I/O execution mode of tag FB from NORMAL mode to SIMULATION mode enables simulation operation. Switching I/O execution mode can be executed with faceplate.


1) Drag and drop the variable parts

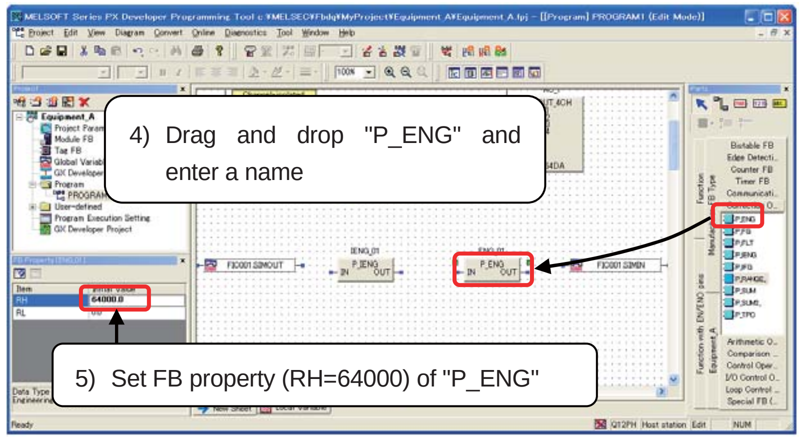
1) Drag and drop a variable part twice and enter variable names "FIC001.SIMOUT" (Tag name.SIMOUT), "FIC001.SIMIN" (Tag name.SIMIN).

* This variable is a public variable of tag FB "FIC001". Therefore, pasting variable part, double-clicking it, and then entering "FIC001.SIMOUT", and "FIC001.SIMIN" registers variable names.


2) Drag and drop "P_IENG" and enter a name.

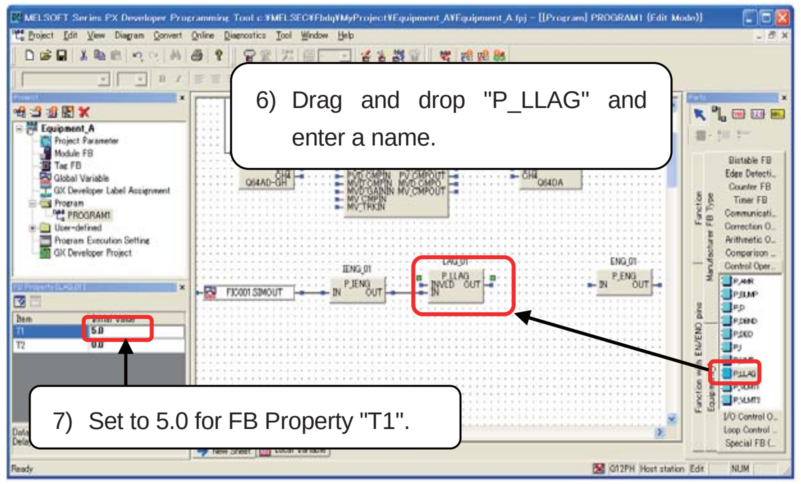
3) Set FB property (RH=12000) of "P_IENG"

2) Select the <<Correction Operation FB>> tab in the <<Manufacturer FB parts>> tab on the "Parts" window, drag and drop "P_IENG" (inverse engineering value conversion), and enter 'IENG_01' as a FB name.
3) Click the pasted FB "IENG_01", and enter (0 to 12000) for the range of analog output module to the "FB Property" of "RH", "RL".


4) Drag and drop "P_ENG" and enter a name

5) Set FB property (RH=64000) of "P_ENG"

4) Select the <<Correction Operation FB>> tab in the <<Manufacturer FB parts>> tab on the "Parts" window, drag and drop "P_ENG" (engineering value conversion), and enter 'ENG_01' as a FB name.
5) Click the pasted FB "ENG_01", and enter (0 to 64000) for the range of analog input module to "FB Property" of "RH", "RL".



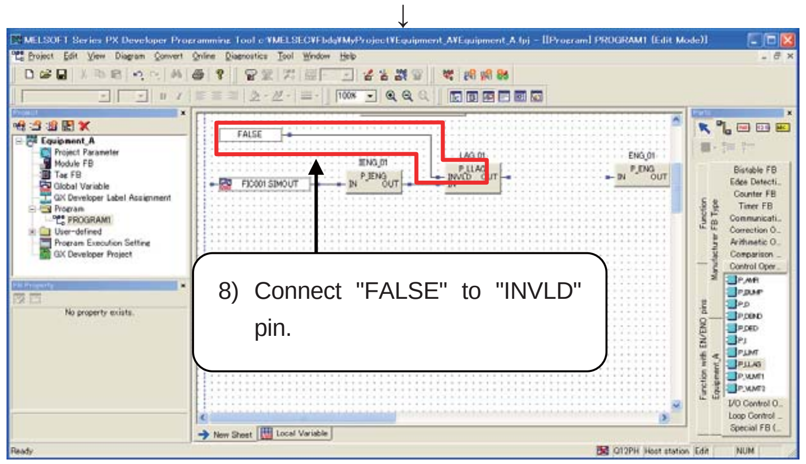
6) Drag and drop "P_LLAG" and enter a name.

7) Set to 5.0 for FB Property "T1".

6) Select the << Control Operation FB >> tab in the <<Manufacturer FB parts>> tab on the "Parts" window, drag and drop "P_LLAG" (Lead-lag), and enter "LAG_01" as a FB name.

7) Click the pasted FB "LAG_01", and set to 5.0 for "FB Property" "T1" (Lag time). Set 0 for "T2" (Lead time).

↓



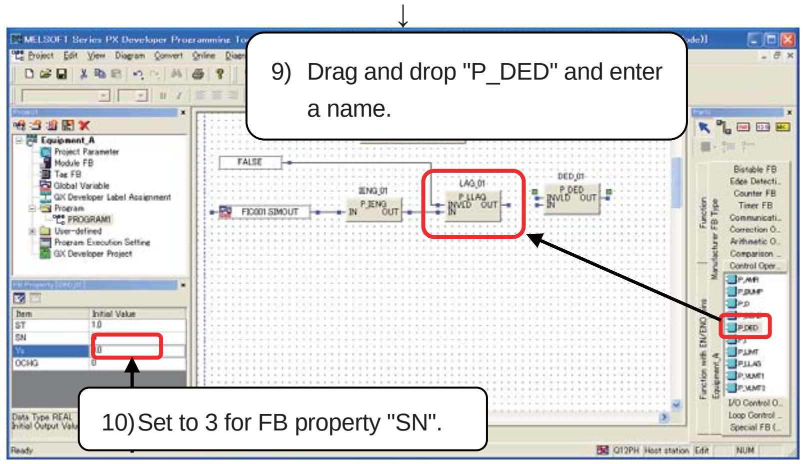
8) Connect "FALSE" to "INVLD" pin.

8) Drag and drop a constant part on the "Parts" window, enter "FALSE" and then connect it to "INVLD" pin of "P_LLAG".

P_LLAG executes lead-lag compensation for input value and outputs it when operation signal (INVLD) is FALSE.

In this setting, lag time is 5.0 seconds.

↓



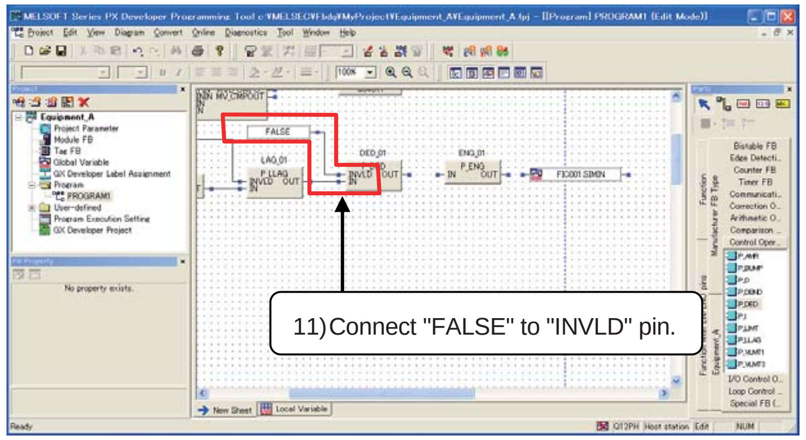
9) Drag and drop "P_DED" and enter a name.

10) Set to 3 for FB property "SN".

9) Select the <<Control Operation FB >> tab in the <<Manufacturer FB parts>> tab on the "Parts" window, drag and drop "P_DED" (Dead time), and enter "DED_01" as a FB name.

10) Click the pasted FB "DED_01", and set to 3 for "SN" (sampling number) of "FB Property"

↓



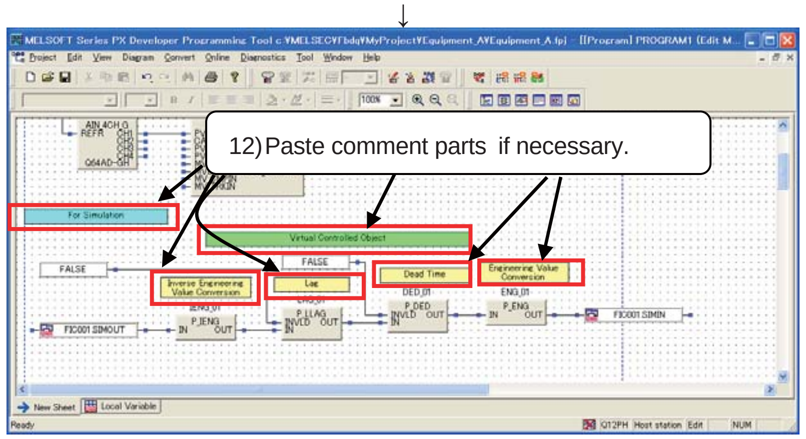
11) Connect "FALSE" to "INVLD" pin.

11) Drag and drop a constant part on the "Parts" window, enter "FALSE" and then connect it to "INVLD" pin of "P_DED".

P_DED gives dead time for input value and outputs it when operation signal (INVLD) is FALSE.

In this setting, dead time is 3.0 seconds.

↓

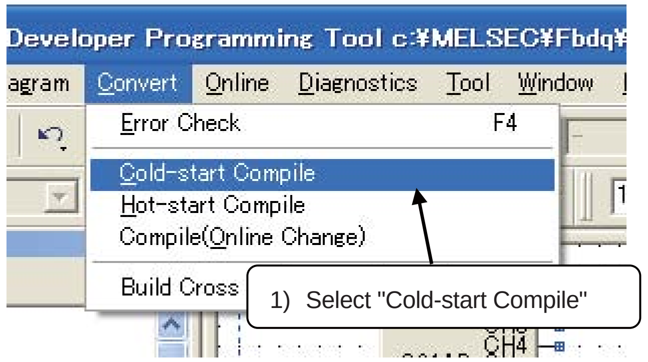


12) Paste comment parts if necessary.

12) Connect lines as shown in the figure to the left, and drag and drop comment parts and enter comments if necessary.

2.1.8 Compiling programs and writing to process CPU

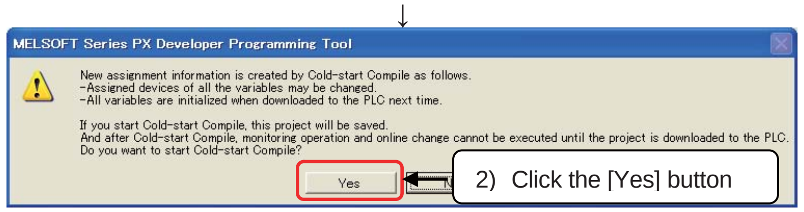
Compile the created program and download the PLC parameter, program, data (default) to process CPU.



1) Select "Cold-start Compile"

1) Select [Convert] → [Cold-start Compile] on the menu to execute cold-start compile.

↓



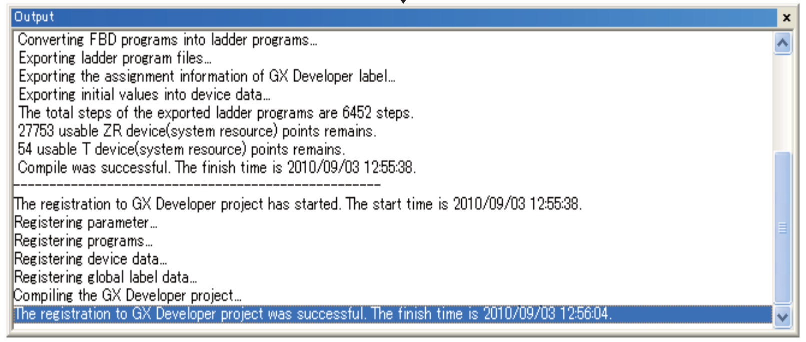
New assignment information is created by Cold-start Compile as follows.
-Assigned devices of all the variables may be changed.
-All variables are initialized when downloaded to the PLC next time.

If you start Cold-start Compile, this project will be saved.
And after Cold-start Compile, monitoring operation and online change cannot be executed until the project is downloaded to the PLC.
Do you want to start Cold-start Compile?

2) Click the [Yes] button

2) Click the [Yes] button.

↓

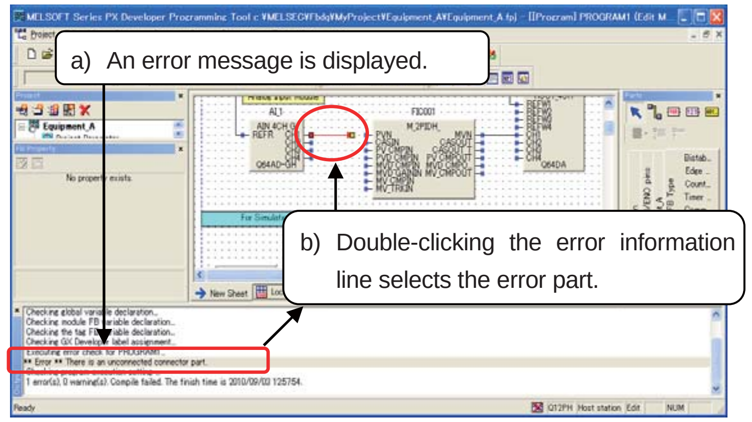


```
Converting FBD programs into ladder programs...
Exporting ladder program files...
Exporting the assignment information of GX Developer label...
Exporting initial values into device data...
The total steps of the exported ladder programs are 6452 steps.
27753 usable ZR device(system resource) points remains.
54 usable T device(system resource) points remains.
Compile was successful. The finish time is 2010/09/03 12:55:38.
------------------------------------------------------------
The registration to GX Developer project has started. The start time is 2010/09/03 12:55:38.
Registering parameter...
Registering programs...
Registering device data...
Registering global label data...
Compiling the GX Developer project...
The registration to GX Developer project was successful. The finish time is 2010/09/03 12:56:04.
```

3) The message of processing contents is displayed on the "Output" window during compiling.
When the compile is normally completed, the message of "The registration to GX Developer project was successful. The finish time is …"is displayed on the "Output" window.

↓

<When a compile error occurs>



a) An error message is displayed.

b) Double-clicking the error information line selects the error part.

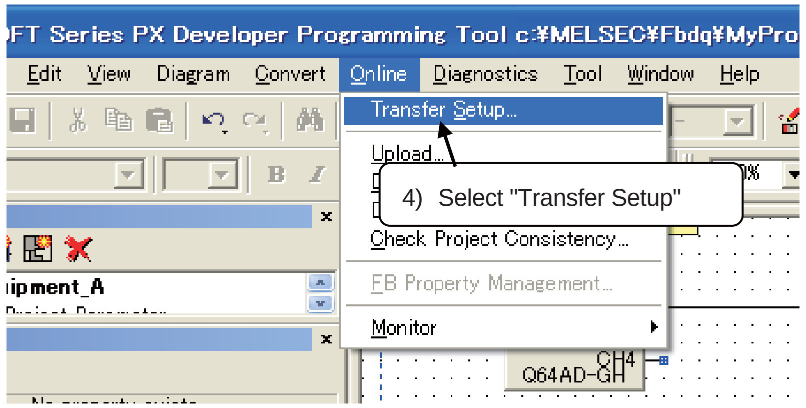
```
Checking global variable declaration...
Checking module FB variable declaration...
Checking the tag FB variable declaration...
Checking GX Developer label assignment...
Executing error check for PROGRAM1...
** Error ** There is an unconnected connector part.
1 error(s), 0 warning(s). Compile failed. The finish time is 2010/09/03 12:57:54.
```

a) If an error occurs, the error message is displayed on the "Output" window, and the compile is stopped.

b) Double-clicking the error information line on the output window selects the error part (such as connector) on the FBD sheet when errors occurred by such as disconnection of the lines, connecting to FBs/variables which have different data types.

Execute compile again after modifying the error parts.

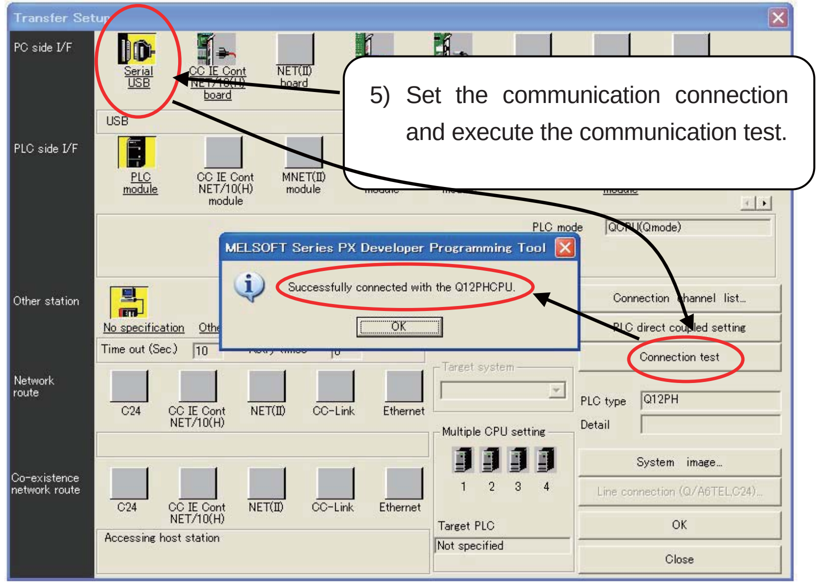
After the compile is normally completed, write the PLC parameter, program, data (default) to process CPU. In this example, switch the Process CPU (the switch is in front of the process CPU) to the STOP status, and then write to the process CPU. Check that the writing is completed normally, and then switch the process CPU to the RUN status.



4) Select "Transfer Setup"

4) Select [Online] → [Transfer Setup] on the menu to display the Transfer Setup dialog.

↓

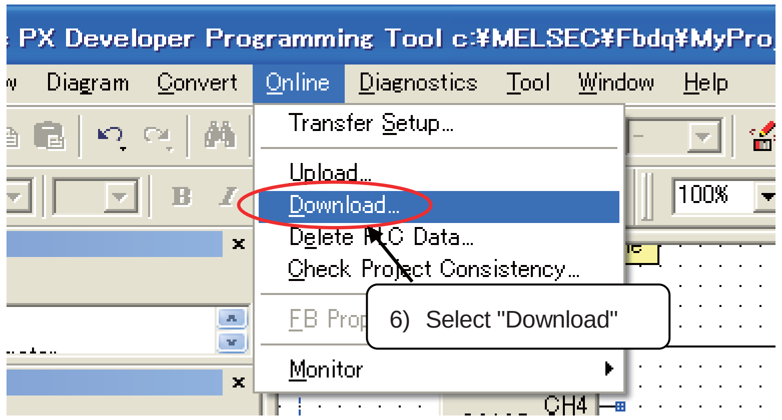


Successfully connected with the Q12PHCPU.

5) Set the communication connection and execute the communication test.

5) Set and check the communication connection to the process CPU on the "Transfer setup" window.

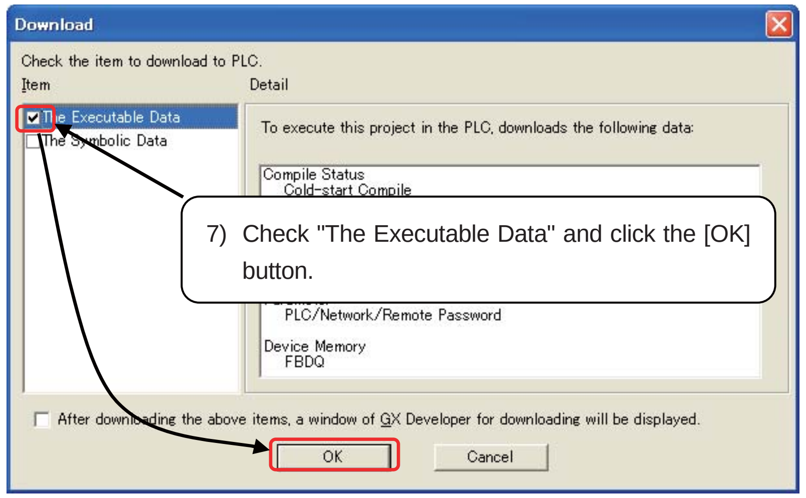
↓



6) Select "Download"

6) Select [Online] → [Download] and then click the [OK] button on the "Download" window.

↓



Check the item to download to PLC.

☑ The Executable Data

☐ The Symbolic Data

To execute this project in the PLC, downloads the following data:

Compile Status
Cold-start Compile

PLC/Network/Remote Password

Device Memory
FBDQ

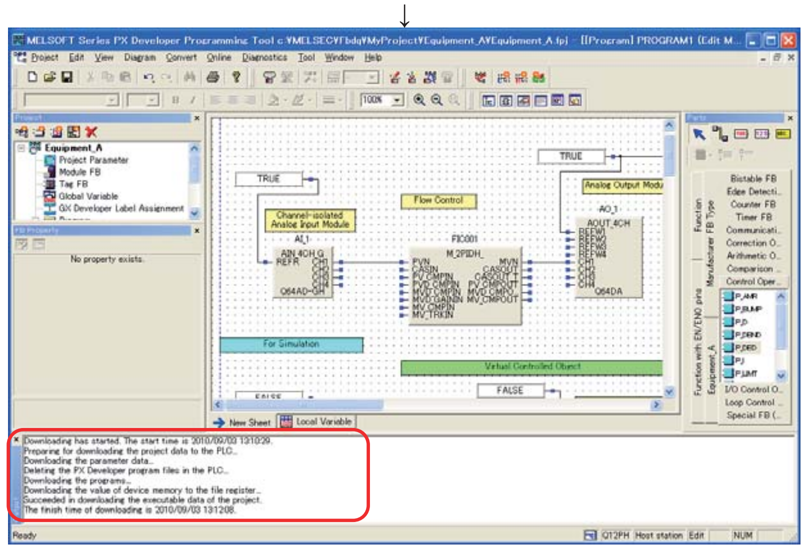
☐ After downloading the above items, a window of GX Developer for downloading will be displayed.

7) Check "The Executable Data" and click the [OK] button.

<Switch the Process CPU to the STOP status.>

7) Check "The Executable Data" and click the [OK] button.

↓



```
Downloading has started. The start time is 2010/09/03 13:10:29.
Preparing for downloading the project data to the PLC...
Downloading the parameter data...
Deleting the PX Developer program files in the PLC...
Downloading the programs...
Downloading the value of device memory to the file register...
Succeeded in downloading the executable data of the project.
The finish time of downloading is 2010/09/03 13:12:08.
```

8) When completing to write to the CPU, the end message is displayed on the "Output" window.

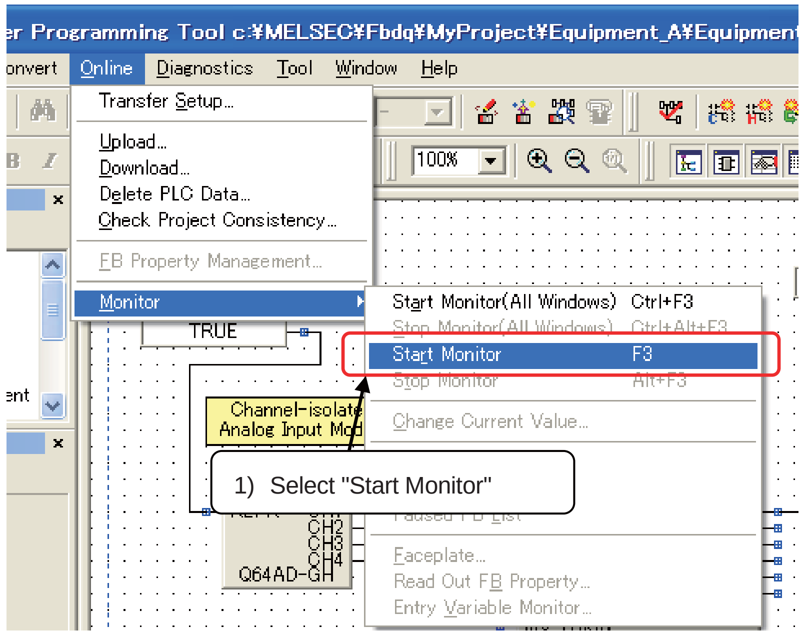
Check the downloading is normally completed, and then
<Switch the Process CPU to the RUN status>.

2.1.9 Monitoring PID loop with Faceplate

Monitor PID loop with faceplate (controller panel) of PX Developer.

Monitor the PID loop and check the PID loop operation on the programming tool.

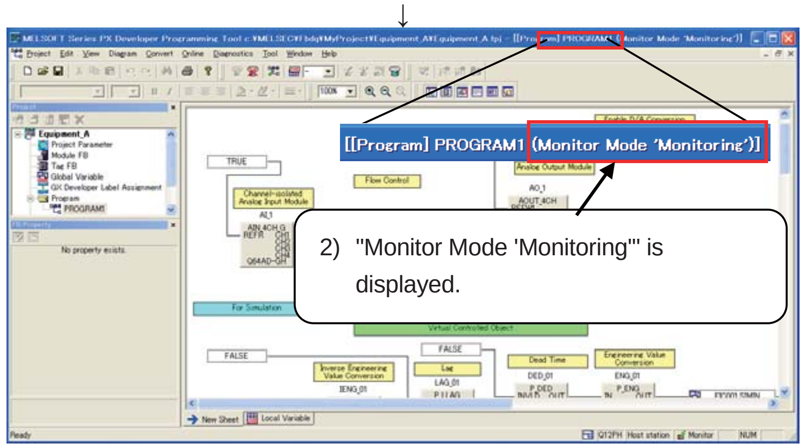
(1) Starting online monitor



1) Select "Start Monitor"

1) Select [Online] → [Monitor] → [Start Monitor] in the menu and start online monitor on the FBD.
(Pressing the F3 key also starts monitoring.)

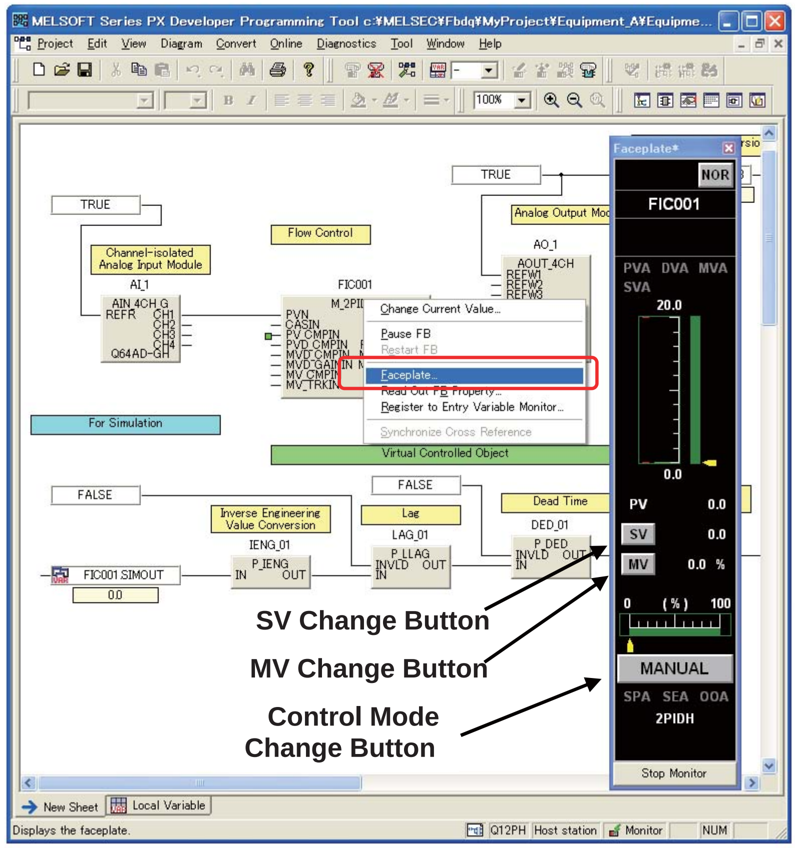
↓



2) "Monitor Mode 'Monitoring'" is displayed.

2) Online monitoring is started and "Monitor Mode 'Monitoring'" is displayed.

↓



SV Change Button

MV Change Button

Control Mode Change Button

3) Right-clicking the tag FB (FIC001) and selecting "Faceplate..." displays the face plate in accordance with the type of PID. The status of PID loop (such as PV, SV, MV, control mode, alarm) can be monitored.

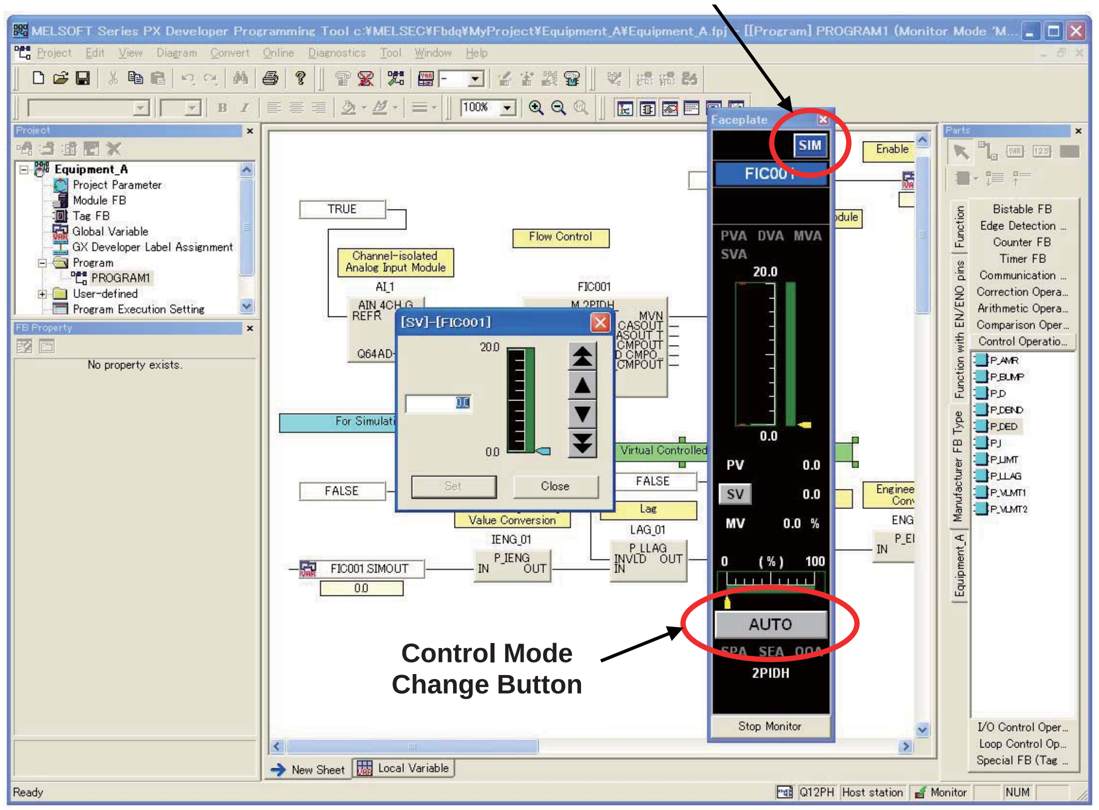
---

(2) Changing the I/O mode

1) Switch the control mode to MANUAL with the control mode change button on the Faceplate, and then click the I/O mode change button shown in the upper-right corner. Selecting SIMULATION on the "Change I/O mode" dialog enables the simulation operation of the PID loop.

2) Switching control mode to AUTO with the control mode change button, and then clicking the SV button to set/change the SV enables to check the PV change by simulation operation (tracking to SV).

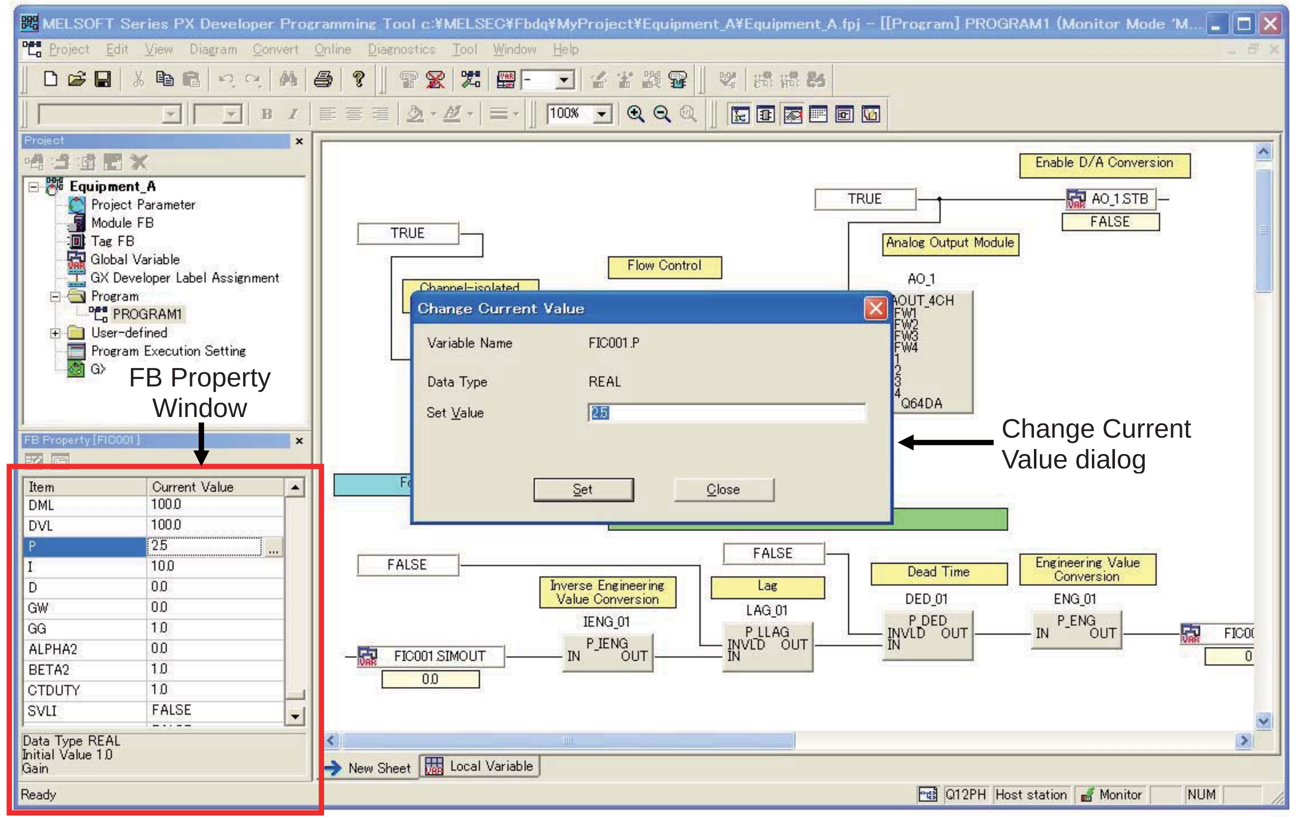
I/O mode change button



Control Mode Change Button

(3) Changing variable with the FB property

The online monitor/change for each parameter of the PID loops can be executed on the FBD. Clicking tag FB in the status of the FBD online monitoring displays the FB property window and enables to monitor the current value for each parameter of tag FBs.
When changing current value, clicking a "Current Value" cell and then ... to the right of the cell displays the "Change Current Value" dialog. Input a value to be changed and then click the [Set] button.



MELSOFT Series PX Developer Programming Tool c:¥MELSEC¥Fbdq¥MyProject¥Equipment_A¥Equipment_A.fpj - [[Program] PROGRAM1 (Monitor Mode 'M...

Change Current Value

Variable Name FIC001.P
Data Type REAL
Set Value 2.5

FB Property [FIC001]

Item	Current Value
DML	100.0
DVL	100.0
P	2.5
I	10.0
D	0.0
GW	0.0
GG	1.0
ALPHA2	0.0
BETA2	1.0
CTDUTY	1.0
SVLI	FALSE

Data Type REAL
Initial Value 1.0
Gain

(4) Reflecting to default of FB property

With the following procedure, read current values of the property from the "FB Property Management" window and reflect them to default in a project.

1) Click [Online] → [FB Property Management...] in the online monitor status.
2) Click the [Read All] button. (The position that differs from the default is displayed in red.)
3) Check Substitute checkbox of the items to be reflected to the default.（Select all for default）
4) Click the [Substitute All] button.

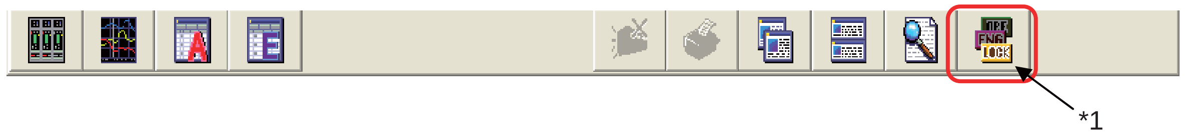
The processing shown above reflects current values as the default in writing to a PLC CPU after completing cold-start compile.

2.1.10 PID loop adjustment with tuning screen (Monitor tool)

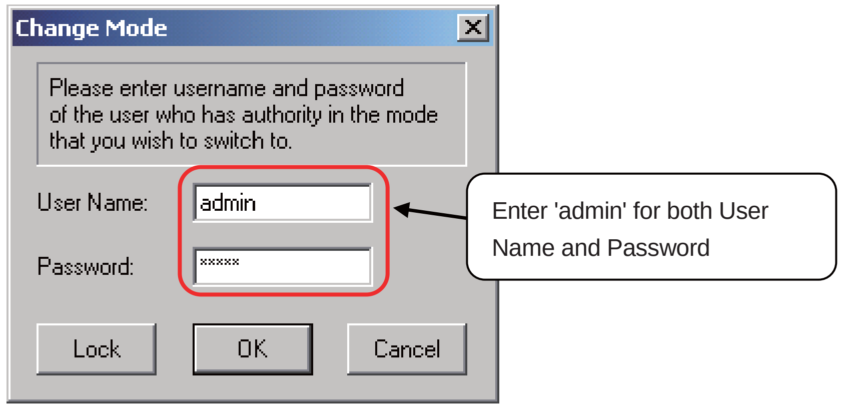
PID loop adjustment can be executed with the tuning screen of Monitor tool (a screen which has functions of faceplate + tuning trend + tag monitor).

(1) Changing mode

Starting the Monitor tool displays the monitor toolbar on the top of the screen as shown below.



Click the [Change Mode] button (*1), enter User Name: 'admin', Password: 'admin' on the dialog box as shown below, and then click the [OK] button. When the switch to the engineer mode conformation message is displayed, click the [OK] button to switch the operation mode to the engineer mode.



Change Mode

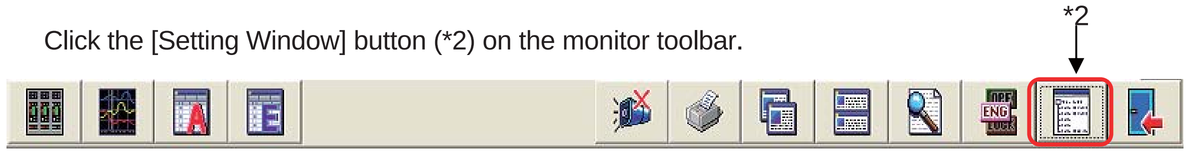
Please enter username and password of the user who has authority in the mode that you wish to switch to.

User Name: admin
Password: *****

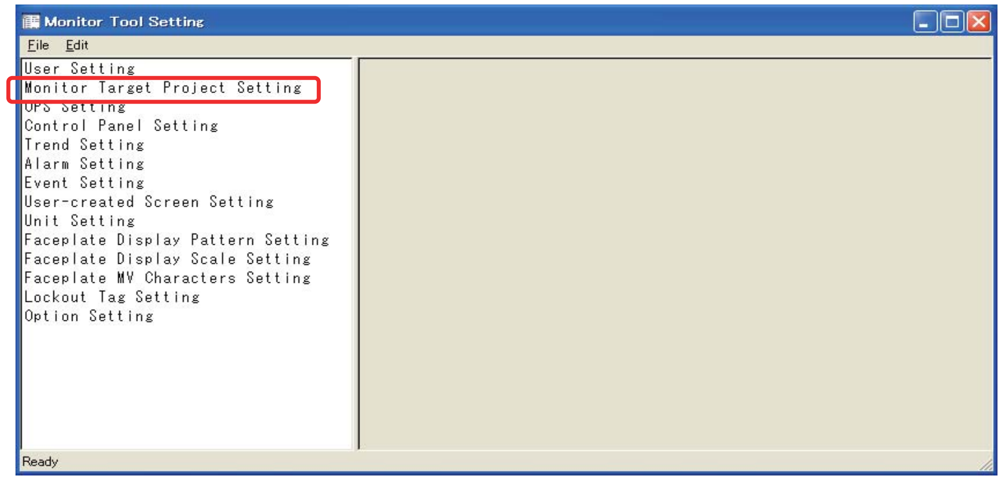
The tool bar is displayed as shown below on the engineer mode. (Change Mode button is displayed as ”ENG” and two buttons are added on the right side of the tool bar.

(2) Project setting

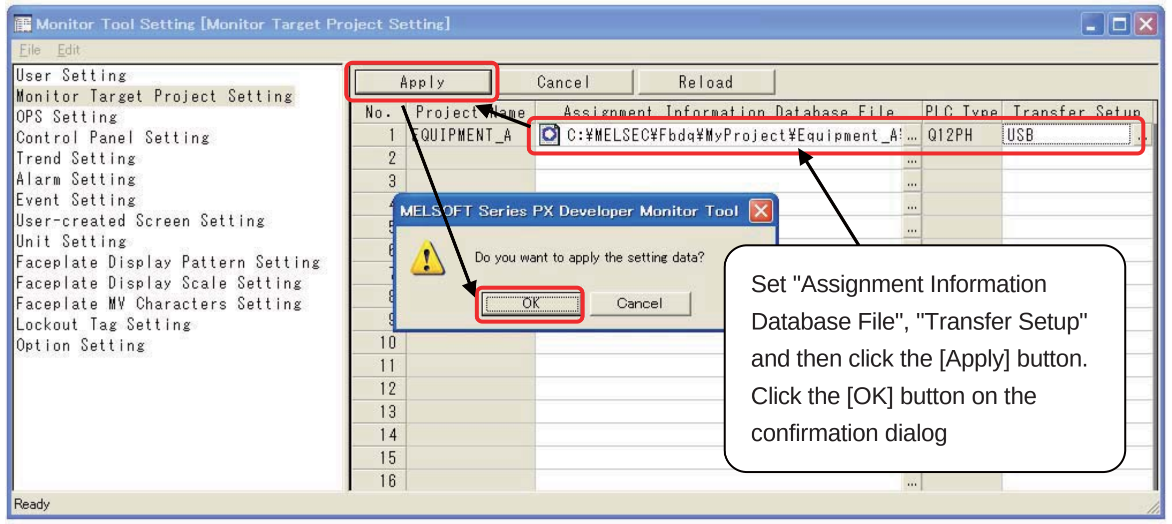
Click the [Setting Window] button (*2) on the monitor toolbar.



Click “Monitor Target Project Setting” on the "Monitor Tool Setting" window as shown below.



Set the assignment information database file (path) and transfer setup on the line No.1. For assignment information database file, set .MDB file which exists in the same directory as the project (.FPJ file) created in the programming tool. For transfer setup, click the ... button to set a communication route whose monitor tool connects to a Process CPU.



No.	Project Name	Assignment Information Database File	PLC Type	Transfer Setup
1	EQUIPMENT_A	C:¥MELSEC¥Fbdq¥MyProject¥Equipment_A	Q12PH	USB

Do you want to apply the setting data?

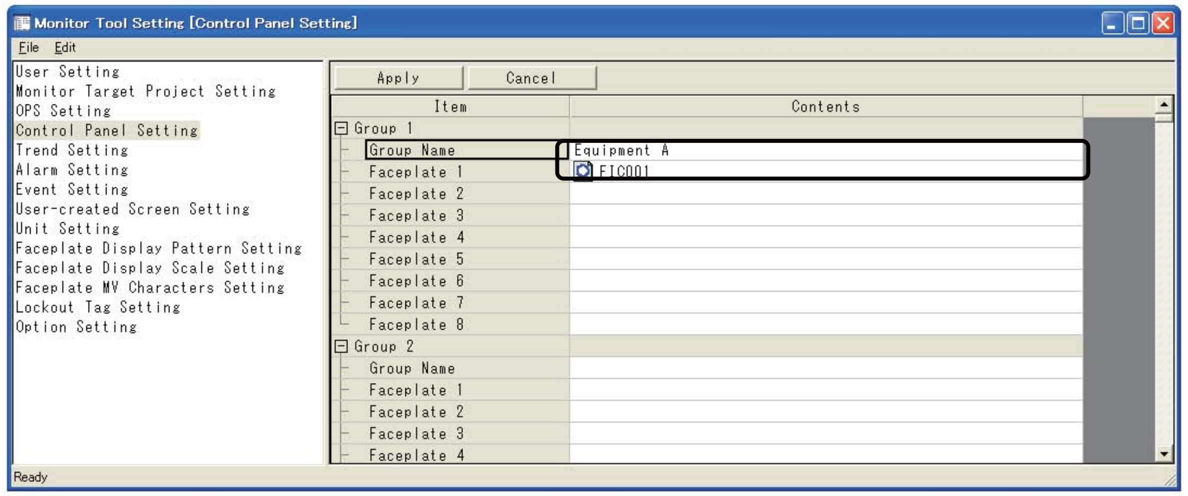
Set "Assignment Information Database File", "Transfer Setup" and then click the [Apply] button. Click the [OK] button on the confirmation dialog

Click the [Apply] button and then click the [OK] button.
Writes tag information registered in a programming tool to the Monitor tool.

---

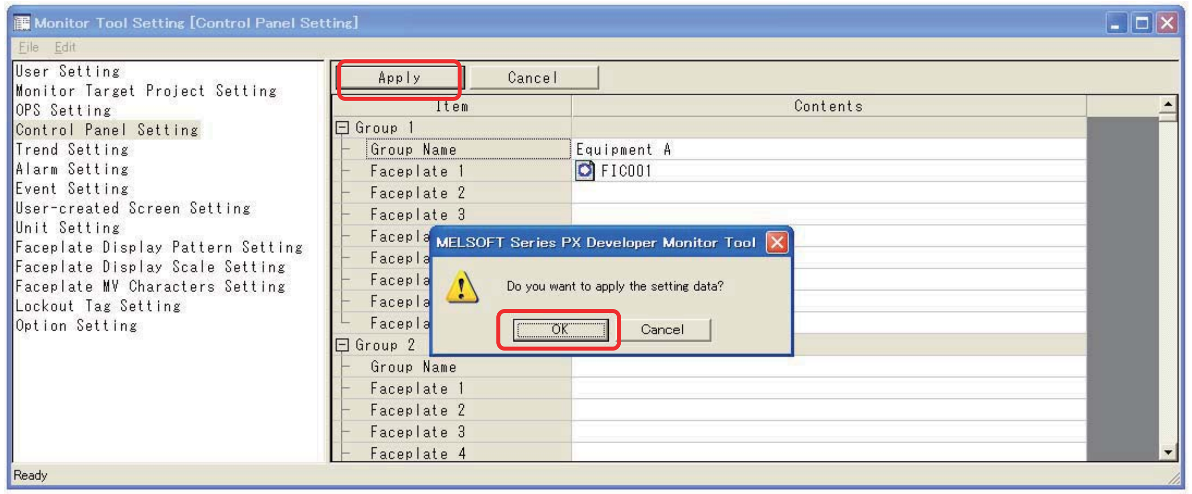
(3) Control panel setting

Click "Control Panel Setting" on the "Monitor Tool Setting" window, enter 'Equipment A' in the "Group Name" cell of "Group 1, and enter the tag name 'FIC001' which is registered in the programming tool in the "Faceplate 1" cell.



Item	Contents
Group 1	
Group Name	Equipment A
Faceplate 1	FIC001
Faceplate 2	
Faceplate 3	
Faceplate 4	
Faceplate 5	
Faceplate 6	
Faceplate 7	
Faceplate 8	
Group 2	
Group Name	
Faceplate 1	
Faceplate 2	
Faceplate 3	
Faceplate 4	

Click the [Apply] button and then click the [OK] button.
Faceplate corresponding to tag "FIC001" which is registered in the programming tool is assigned to group 'Equipment A'.



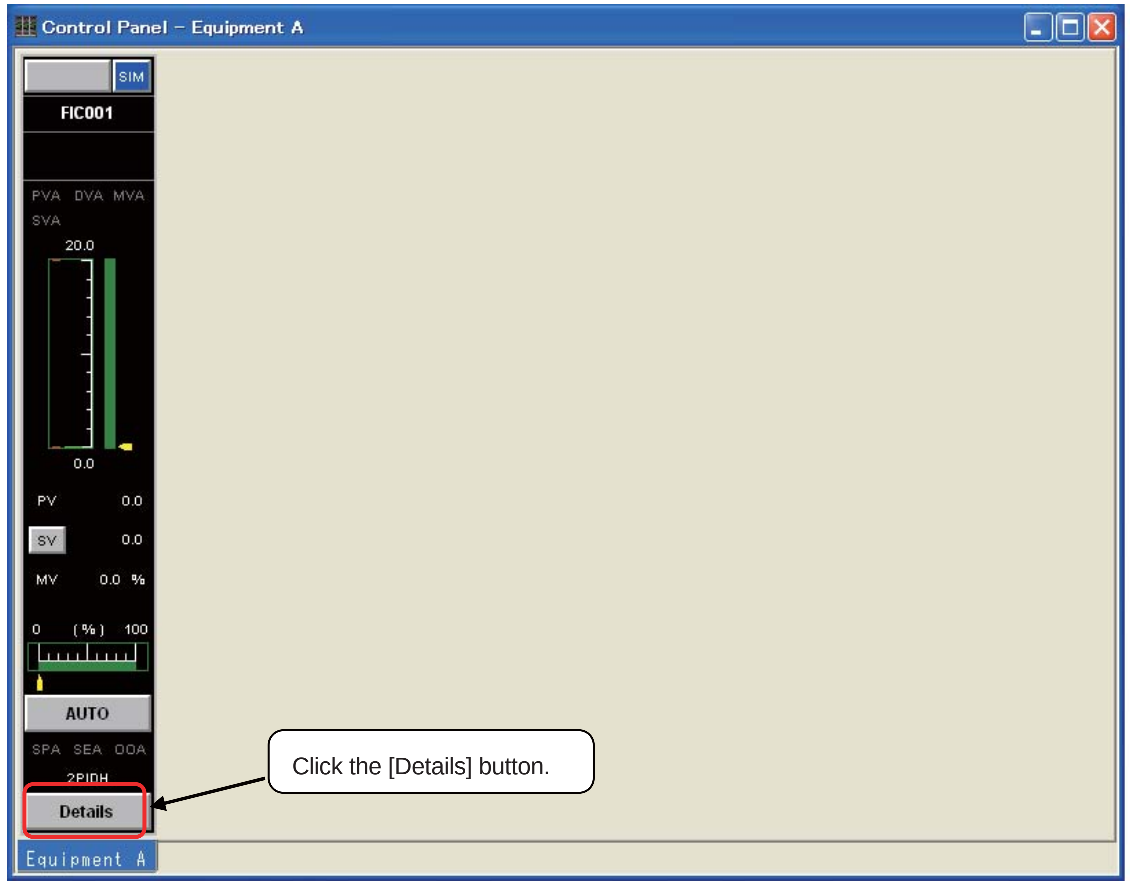
Item	Contents
Group 1	
Group Name	Equipment A
Faceplate 1	FIC001

Do you want to apply the setting data?

(4) Tuning screen

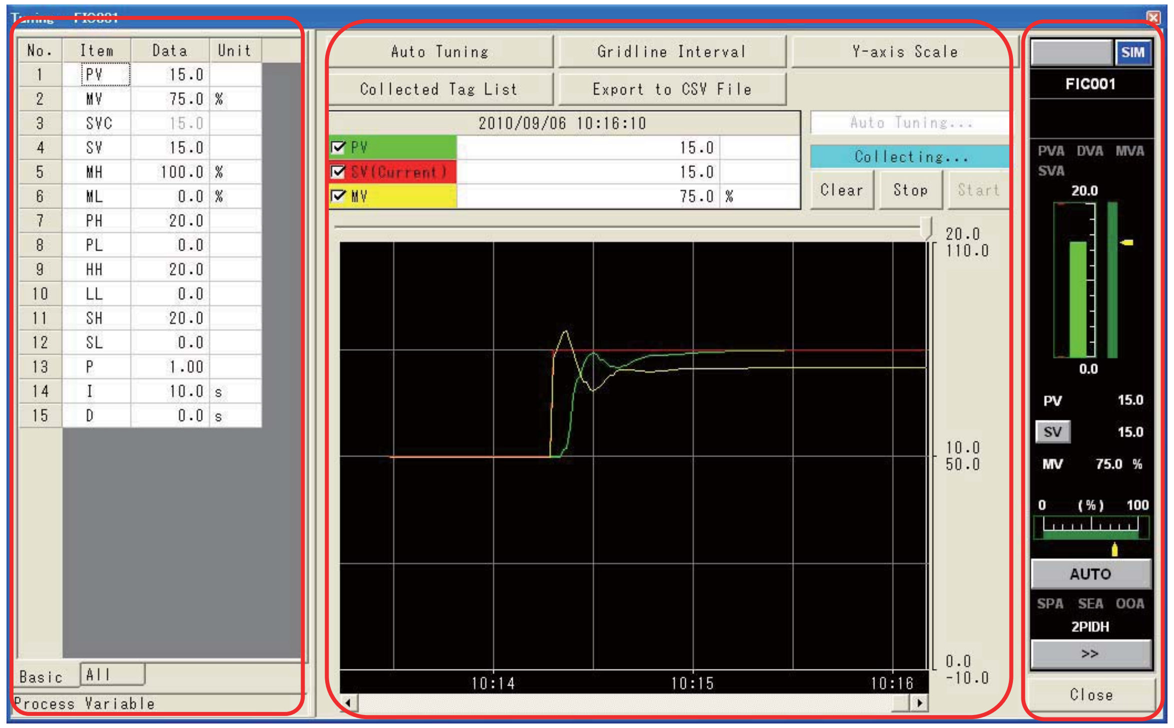
Click the [Control Panel] button (*3) on the monitor toolbar.

A group name and faceplate which are registered in " (3) Control panel setting" are displayed.
Click the faceplate [Details] button on the "FIC001".



Click the [Details] button.

Click the [Details] button on the faceplate "FIC001" to display the tuning screen as shown below.
The tuning screen is composed of the tag monitor, tuning trend graph and faceplate.



No.	Item	Data	Unit
1	PV	15.0	
2	MV	75.0	%
3	SVC	15.0	
4	SV	15.0	
5	MH	100.0	%
6	ML	0.0	%
7	PH	20.0	
8	PL	0.0	
9	HH	20.0	
10	LL	0.0	
11	SH	20.0	
12	SL	0.0	
13	P	1.00	
14	I	10.0	s
15	D	0.0	s

2010/09/06 10:16:10

☑ PV	15.0	
☑ SV(Current)	15.0	
☑ MV	75.0	%

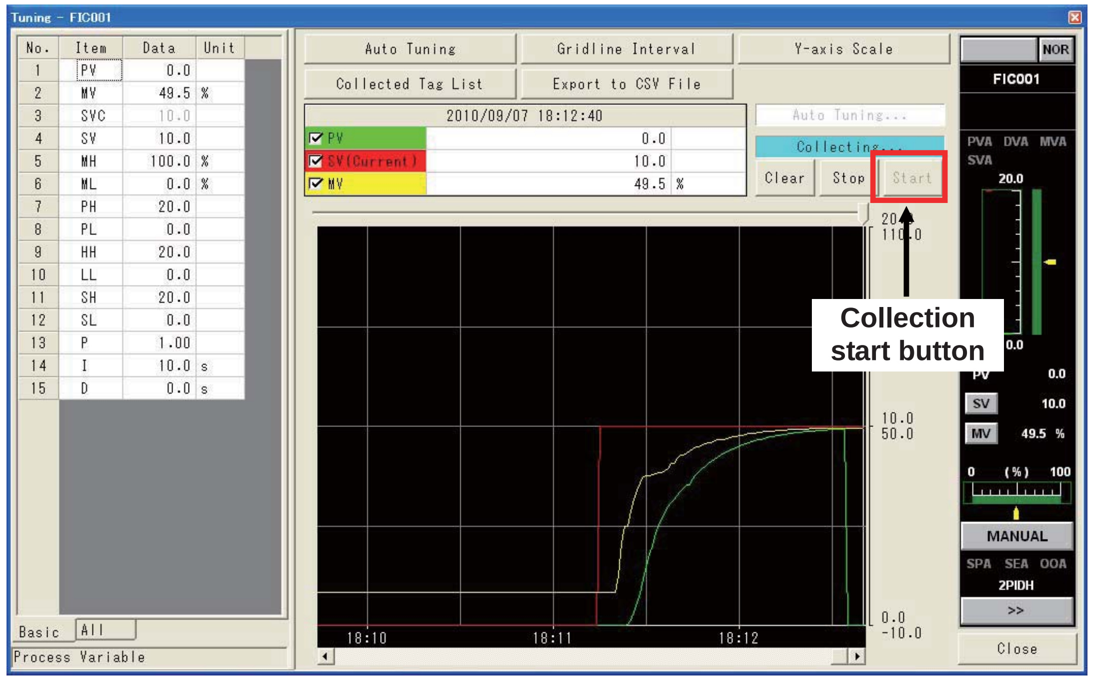
Tag monitor

Tuning trend graph

Faceplate

(5) Starting online monitor

Click the collection start button ([Start] button) to start the trend data collection/display the monitor (the display above the collection start button is switched to "Collected" from "Stopped").



Tuning - FIC001

No.	Item	Data	Unit
1	PV	0.0	
2	MV	49.5	%
3	SVC	10.0	
4	SV	10.0	
5	MH	100.0	%
6	ML	0.0	%
7	PH	20.0	
8	PL	0.0	
9	HH	20.0	
10	LL	0.0	
11	SH	20.0	
12	SL	0.0	
13	P	1.00	
14	I	10.0	s
15	D	0.0	s

2010/09/07 18:12:40

☑ PV	0.0	
☑ SV(Current)	10.0	
☑ MV	49.5	%

(6) Mode change

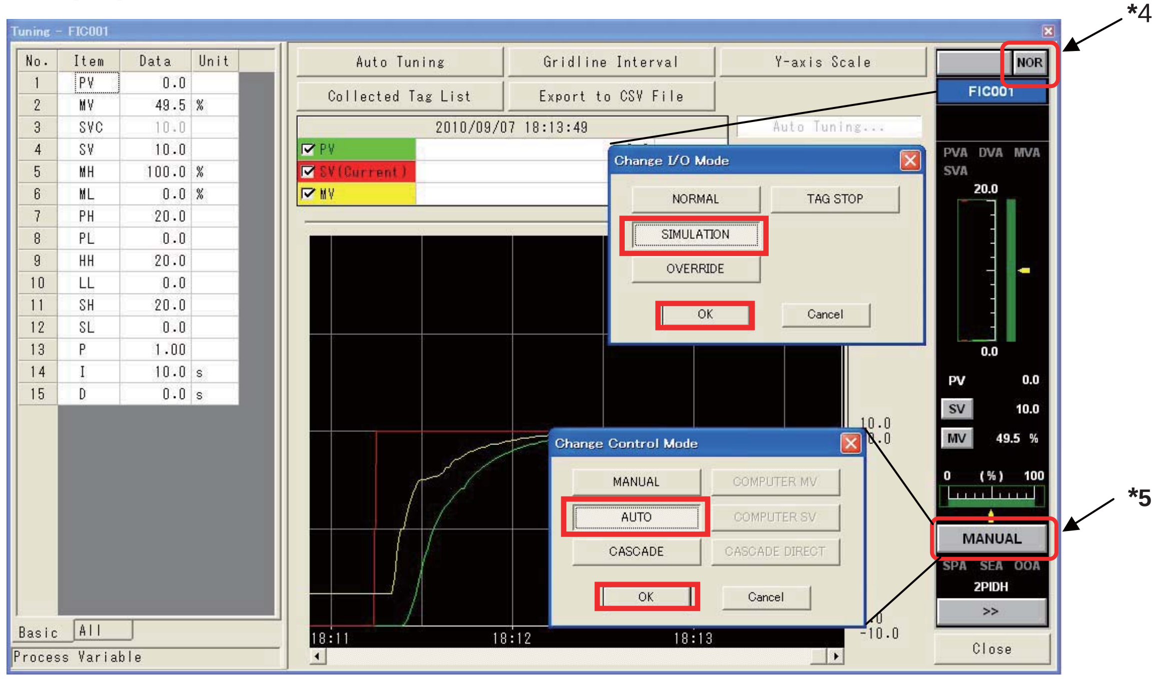
- **Change the I/O mode to the SIMULATION**

  Click the I/O mode change button (*4) on the faceplate.

  The "Change I/O Mode" dialog is displayed. Click the [SIMULATION] button and then click the [OK] button.
- **Change the control mode to the AUTO**

  Click the control mode change button (*5) on the faceplate.

  The "Change Control Mode" dialog is displayed. Click the [AUTO] button and then click the [OK] button.



Tuning - FIC001

No.	Item	Data	Unit
1	PV	0.0	
2	MV	49.5	%
3	SVC	10.0	
4	SV	10.0	
5	MH	100.0	%
6	ML	0.0	%
7	PH	20.0	
8	PL	0.0	
9	HH	20.0	
10	LL	0.0	
11	SH	20.0	
12	SL	0.0	
13	P	1.00	
14	I	10.0	s
15	D	0.0	s

2010/09/07 18:13:49

---

(7) Adjusting PID loop

Change SV on the faceplate, and monitor the PV/SV/MV condition on the tuning trend.
Also, adjust the target tracking performance of PV corresponding to SV with changing P/I/D constant on the tag monitor.

- **Change SV**

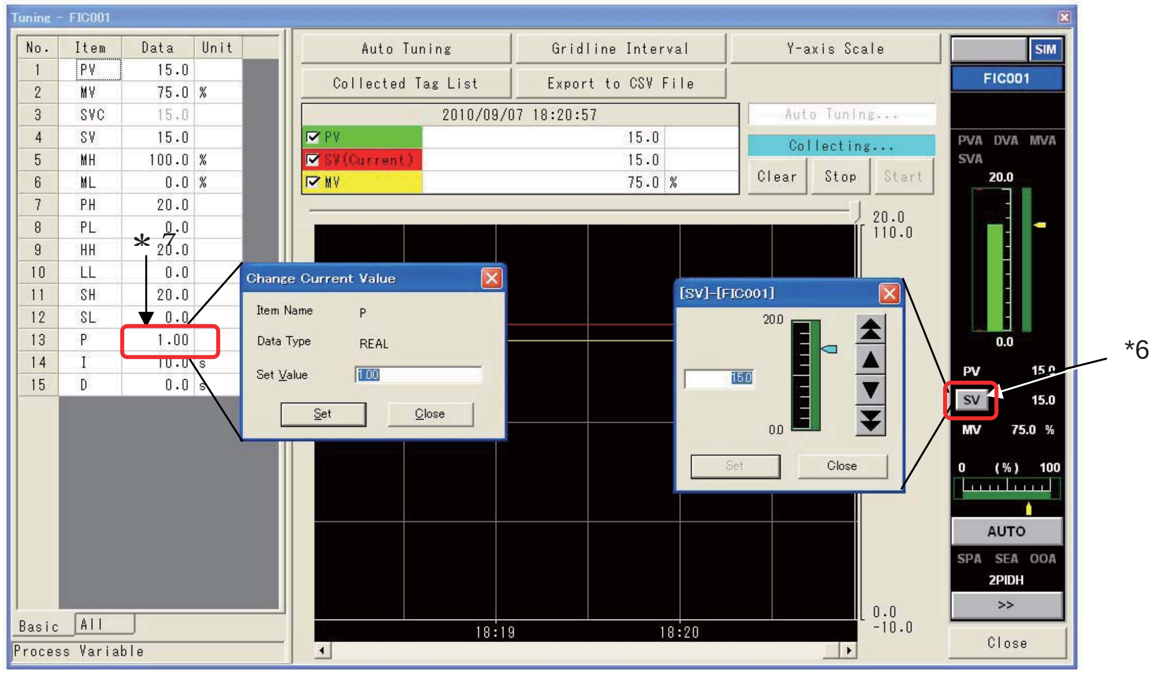
  Click the [SV] button (*6) on the faceplate.

  The "Change Current Value" dialog is displayed. Enter the value to be changed and click the [Set] button.

- **Change PID parameters**

  Click the [Data] cell of the tag data items to be changed on the tag monitor.

  Clicking ... to the right of the cell displays the "Change Current Value". Enter the values to be changed and click the [Set] button.



Tuning - FIC001

No.	Item	Data	Unit
1	PV	15.0	
2	MV	75.0	%
3	SVC	15.0	
4	SV	15.0	
5	MH	100.0	%
6	ML	0.0	%
7	PH	20.0	
8	PL	0.0	
9	HH	20.0	
10	LL	0.0	
11	SH	20.0	
12	SL	0.0	
13	P	1.00	
14	I	10.0	s
15	D	0.0	s

2010/09/07 18:20:57

☑ PV	15.0	
☑ SV(Current)	15.0	
☑ MV	75.0	%

Change Current Value

Item Name P
Data Type REAL
Set Value 1.00

[SV]-[FIC001]

15.0

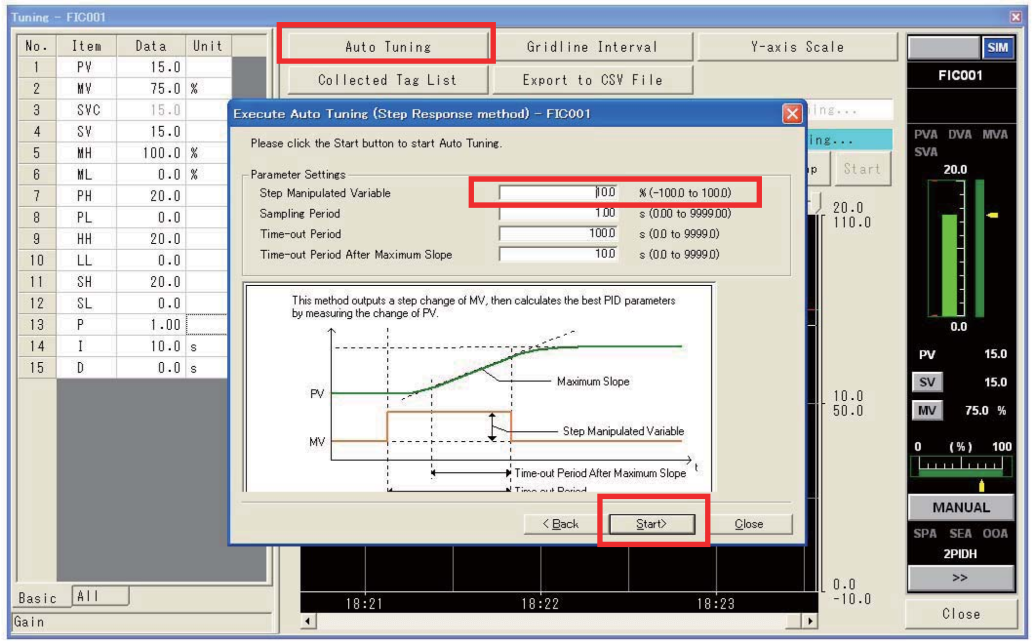
(8) Auto tuning (adjust automatically PID parameters to optimal values with step response method)

- **Change the control mode to MANUAL**

  Click the control mode change button on the faceplate.
  The "Change I/O Mode" dialog is displayed. Click the [MANUAL] button and then click the [OK] button.

- **Start tuning**

  Click the [Auto Tuning] button on the "Tuning" window.
  The "Auto Tuning" dialog is displayed. Enter '10.0' to "Step Manipulated Variable" and then click the [Start] button.



Tuning – FIC001

No.	Item	Data	Unit
1	PV	15.0	
2	MV	75.0	%
3	SVC	15.0	
4	SV	15.0	
5	MH	100.0	%
6	ML	0.0	%
7	PH	20.0	
8	PL	0.0	
9	HH	20.0	
10	LL	0.0	
11	SH	20.0	
12	SL	0.0	
13	P	1.00	
14	I	10.0	s
15	D	0.0	s

Execute Auto Tuning (Step Response method) – FIC001

Please click the Start button to start Auto Tuning.

Parameter Settings

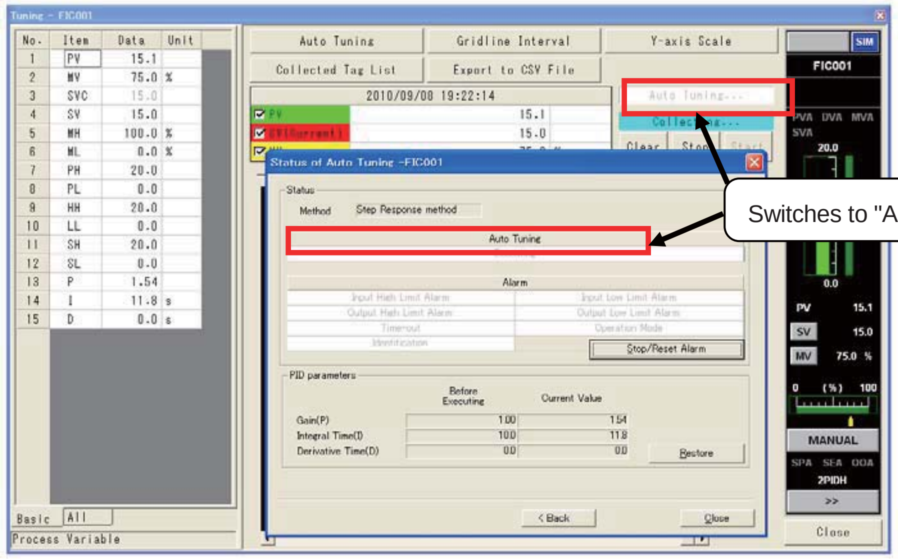
Step Manipulated Variable	10.0	% (-100.0 to 100.0)
Sampling Period	1.00	s (0.00 to 9999.00)
Time-out Period	100.0	s (0.0 to 9999.0)
Time-out Period After Maximum Slope	10.0	s (0.0 to 9999.0)

This method outputs a step change of MV, then calculates the best PID parameters by measuring the change of PV.

The "Confirm Start" dialog is displayed. Click the [Yes] button.

Confirm Start

Do you want to start?



Tuning – FIC001

No.	Item	Data	Unit
1	PV	15.1	
2	MV	75.0	%
3	SVC	15.0	
4	SV	15.0	
5	MH	100.0	%
6	ML	0.0	%
7	PH	20.0	
8	PL	0.0	
9	HH	20.0	
10	LL	0.0	
11	SH	20.0	
12	SL	0.0	
13	P	1.54	
14	I	11.8	s
15	D	0.0	s

Status of Auto Tuning –FIC001

Status

Method: Step Response method

Auto Tuning

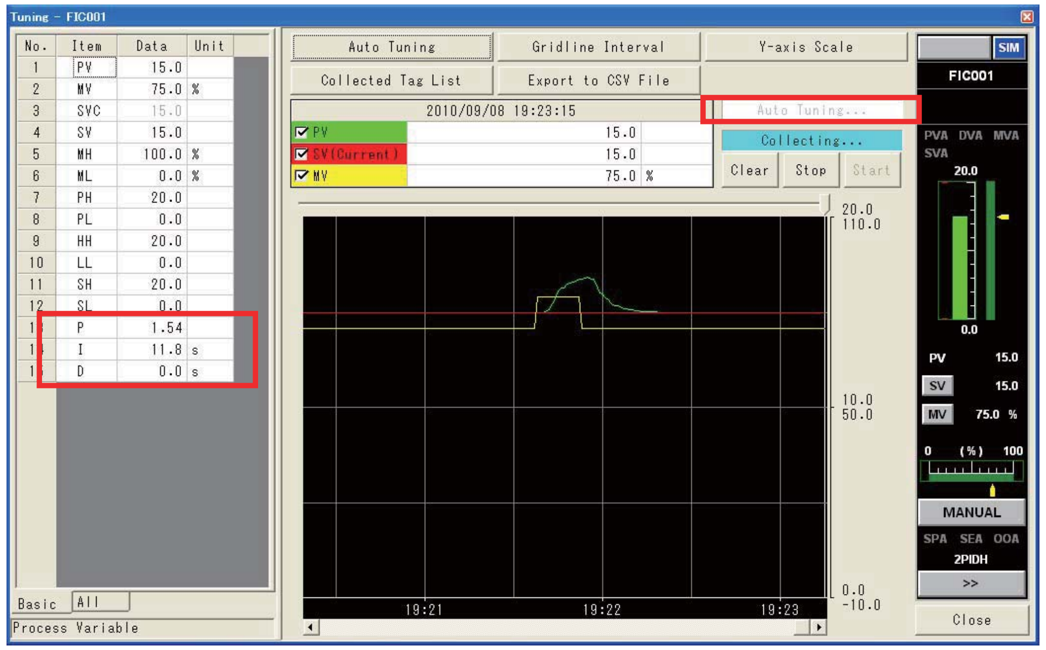
PID parameters

	Before Executing	Current Value
Gain(P)	1.00	1.54
Integral Time(I)	10.0	11.8
Derivative Time(D)	0.0	0.0

Switches to "Auto Tuning..."

After a while, the display of "Auto Tuning…" is turned off (grayed out). Click the [Close] button on the "Auto Tuning" dialog and close the dialog.

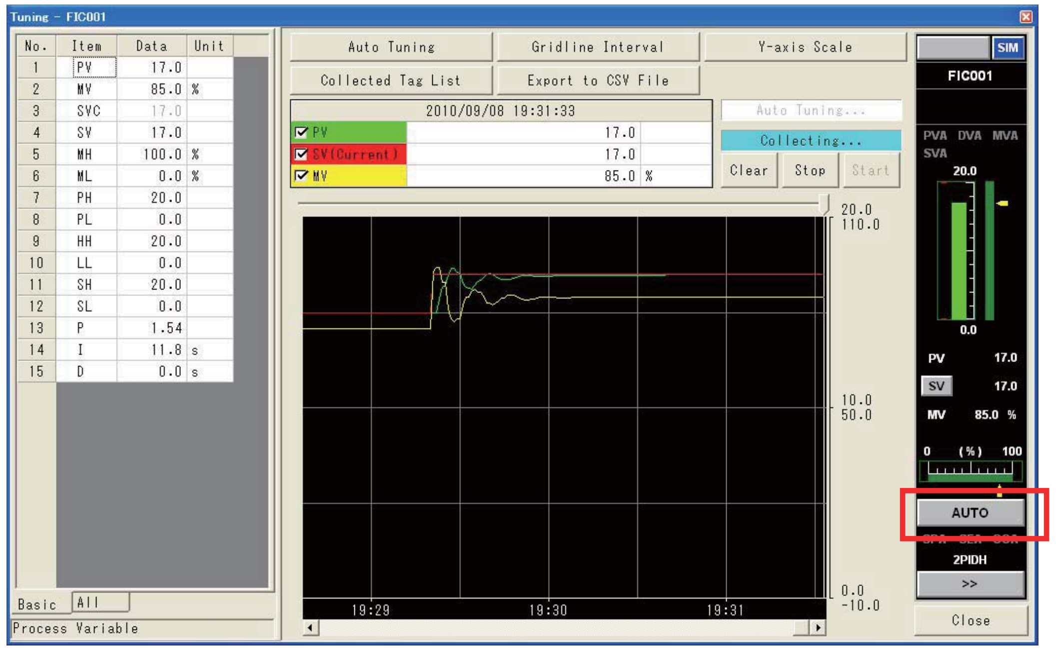
After completing the auto tuning, the display of "Auto Tuning…"is turned off, refreshes PID constants automatically.



Tuning – FIC001

No.	Item	Data	Unit
1	PV	15.0	
2	MV	75.0	%
3	SVC	15.0	
4	SV	15.0	
5	MH	100.0	%
6	ML	0.0	%
7	PH	20.0	
8	PL	0.0	
9	HH	20.0	
10	LL	0.0	
11	SH	20.0	
12	SL	0.0	
13	P	1.54	
14	I	11.8	s
15	D	0.0	s

Change control mode to [AUTO] and monitor the PV/SV/MV condition on the tuning trend.
Check the target tracking performance of PV corresponding to SV after auto tuning.



Tuning – FIC001

No.	Item	Data	Unit
1	PV	17.0	
2	MV	85.0	%
3	SVC	17.0	
4	SV	17.0	
5	MH	100.0	%
6	ML	0.0	%
7	PH	20.0	
8	PL	0.0	
9	HH	20.0	
10	LL	0.0	
11	SH	20.0	
12	SL	0.0	
13	P	1.54	
14	I	11.8	s
15	D	0.0	s

2.2 Programming Procedure of User-defined FB

The user-created FB type can be defined in the programming tool.
The programming man-hour of FBD can be shortened by defining the processing used frequently in the program as use-defined FBs in advance.

2.2.1 Structure of user-defined FB

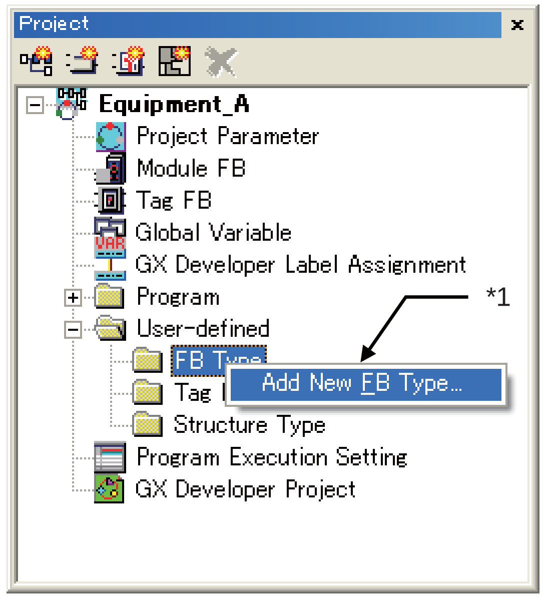
User-defined FB is composed of such as previously-prepared function parts, FB parts (tag access FB excluded).
(Note) The user-defined FB cannot be pasted to the definition window of the FB (it should not be pasted on itself).

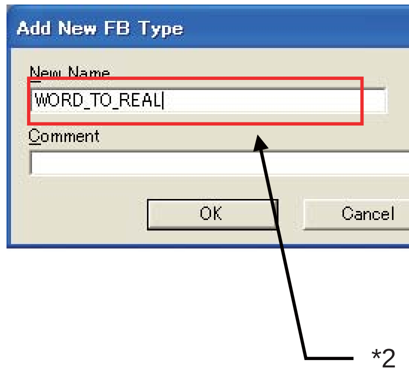
2.2.2 Programming procedure of user-defined FB

The following shows a programming procedure of user-defined FB which converts WORD type to REAL type as an example.

1) Right-click [FB Type] under [User-defined] on the Project window and click [Add New FB Type...]. (Following figure *1)

2) Enter 'Convert WORD to REAL' in the New Name on the Add New FB Type window (Following figure *2). Comment is optional.

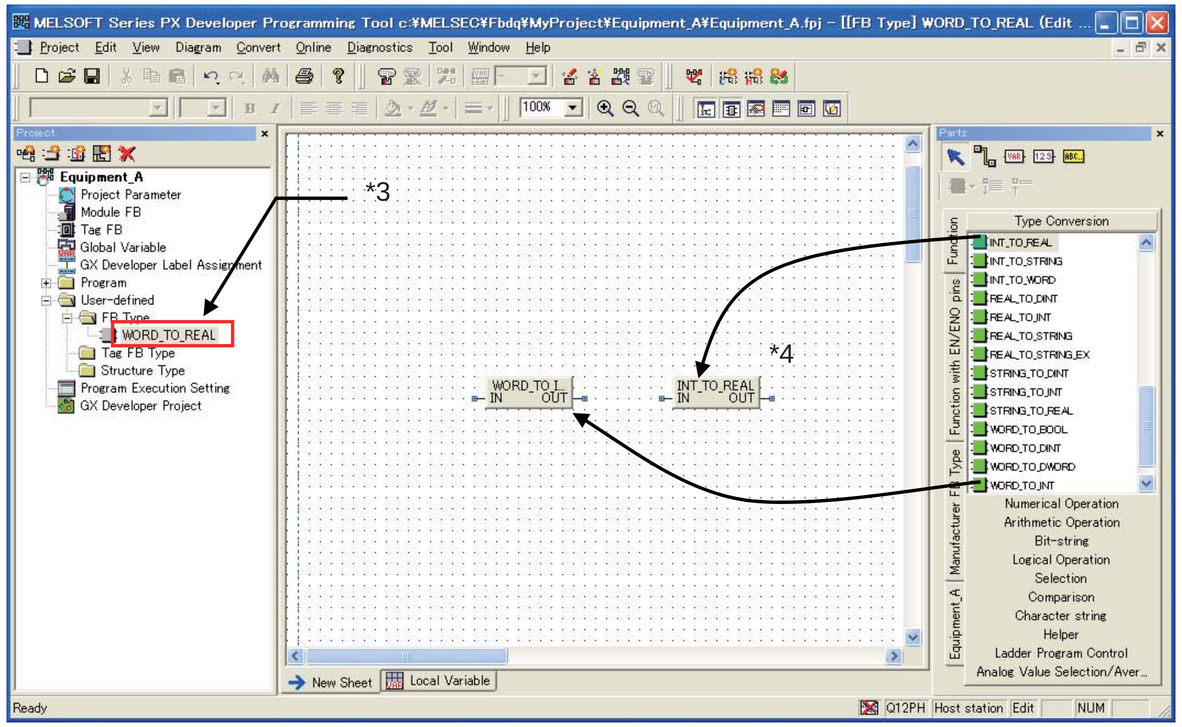




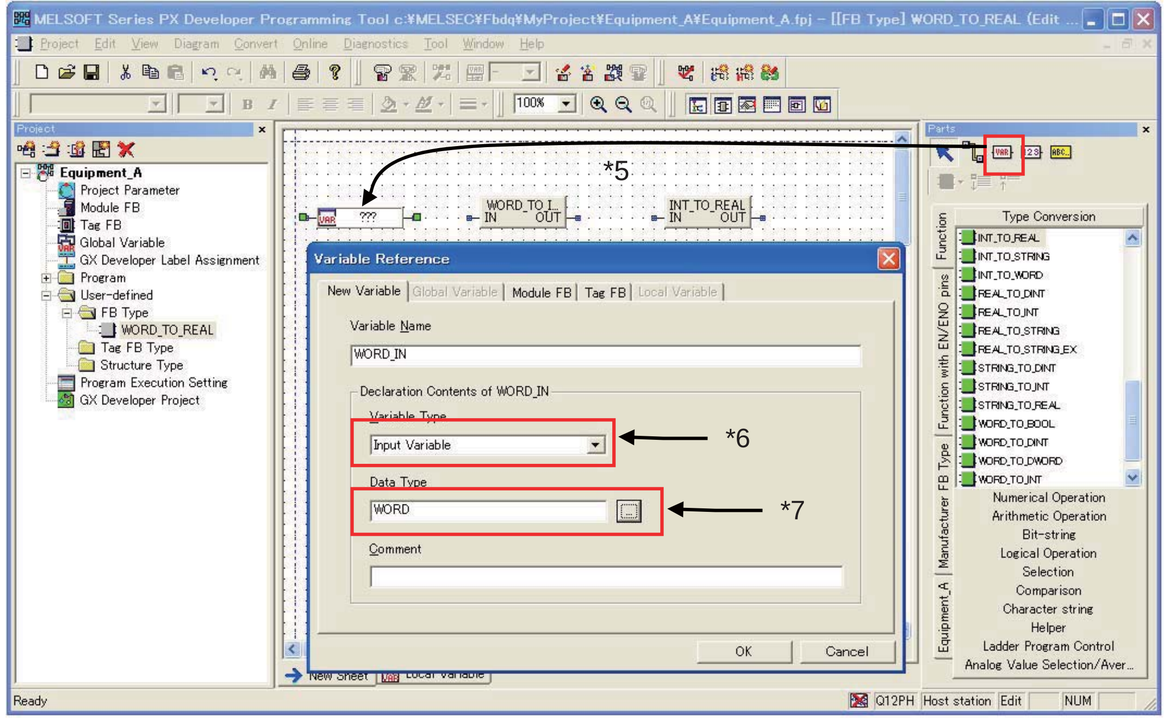
New Name: WORD_TO_REAL

3) Double click [FB Type] under [User-defined] on the Project window, double click "Convert WORD to REAL" (Figure*3 in the next page)

4) User-defined FB "Convert WORD to REAL" sheet is displayed. Drag and drop WORD_TO_INT (WORD → INT Conversion), INT_TO_REAL (INT → REAL Conversion) under [Function] - [Type Conversion] on the Parts window (Figure*4 in the next page)



5) Drag and drop variable parts from the Parts window (following figure *5), enter 'WORD_IN' (any name can be entered) in the variable name "???", and then press the [Enter] key. The Variable Reference dialog is displayed. Click ▾ to the right of the Variable Type and select "Input Variable". (Following figure *6)
Then click ... to the right of Data Type and select "WORD" (following figure *7)

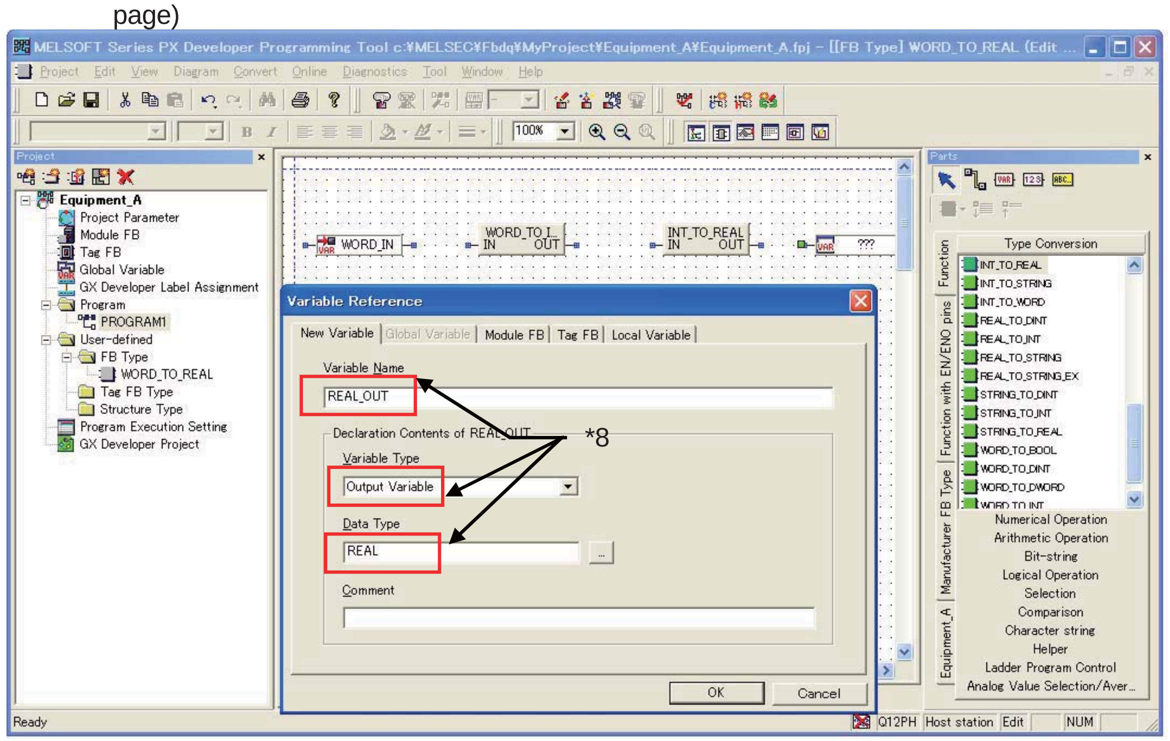


Variable Name: WORD_IN
Variable Type: Input Variable
Data Type: WORD

･･･ Selecting "input variable" as a variable part type displays input pin on the user-defined FB. In this example, input pin whose name is "WORD_IN" is created.

6) Create variable with data type "REAL", variable name "REAL_OUT" (any name can be entered), variable type "Output Variable" with the same procedure as shown above. (Figure*8 in the next

page)



Variable Name: REAL_OUT

Declaration Contents of REAL_OUT

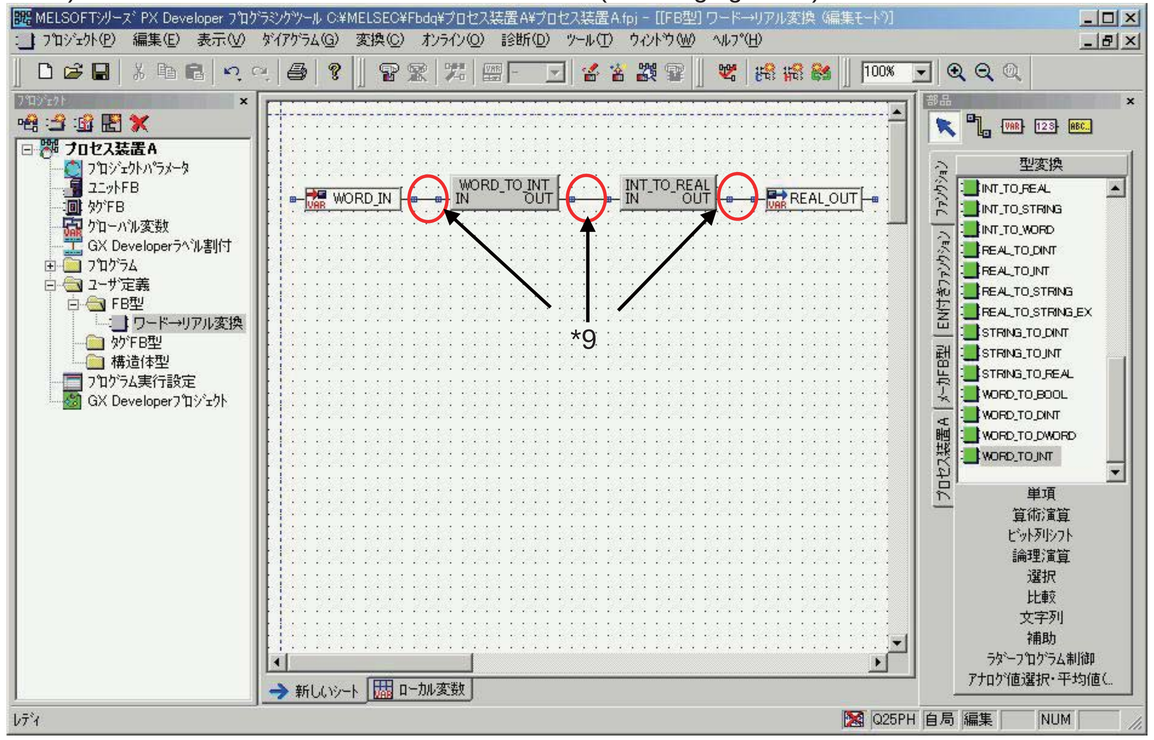
Variable Type: Output Variable

Data Type: REAL

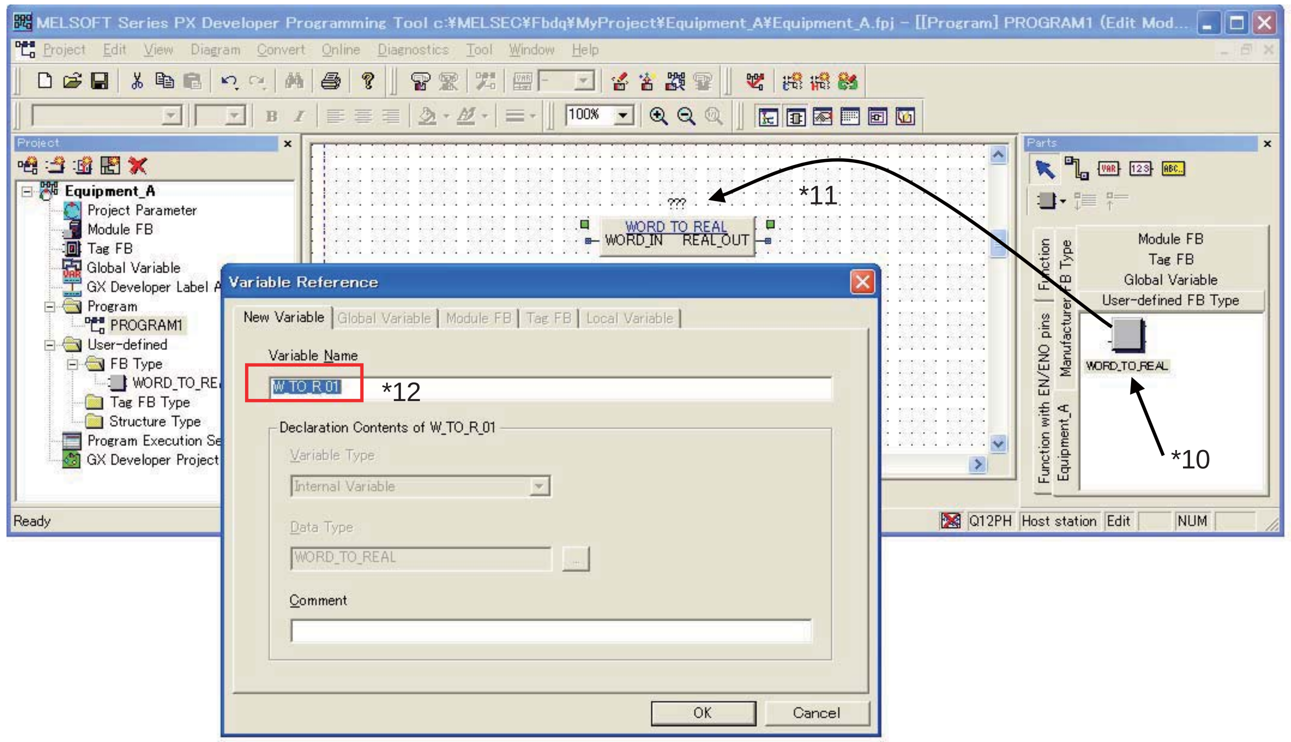
Comment

・・・ Selecting output variable as a variable part type displays output pin on the user-defined FB. In this example, output pin whose name is "REAL_OUT" is created.

7) Connect variables and FBs with connector (following figure *9).



8) "Convert WORD to REAL" is displayed on the Parts window by clicking [Equipment A] → [User-defined FB Type] (Figure*10 in the next page). To use created user-defined FB, drag and drop them on a sheet. (Figure*11 in the next page)
Set a variable name. (Figure*12 in the next page)



Variable Name: W_TO_R_01

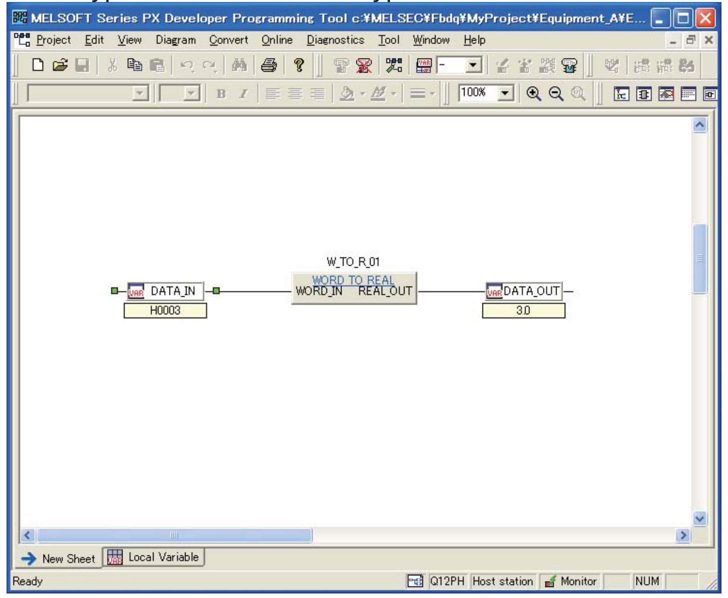
Declaration Contents of W_TO_R_01

Variable Type: Internal Variable

Data Type: WORD_TO_REAL

Comment

9) Connect WORD type variable and REAL type variable.



2.3 Programming Procedure of User-defined Tag FB

In Section 2.1, a procedure of creating loop control programs with 2-degree-of-freedom advanced PID control is explained.
Creating user original PID control FB enables to execute different control from PID control FBs which are equipped as standard. User-created PID control FB is expressed user-defined tag FB.
This section explains a procedure of user-defined tag FB creation.

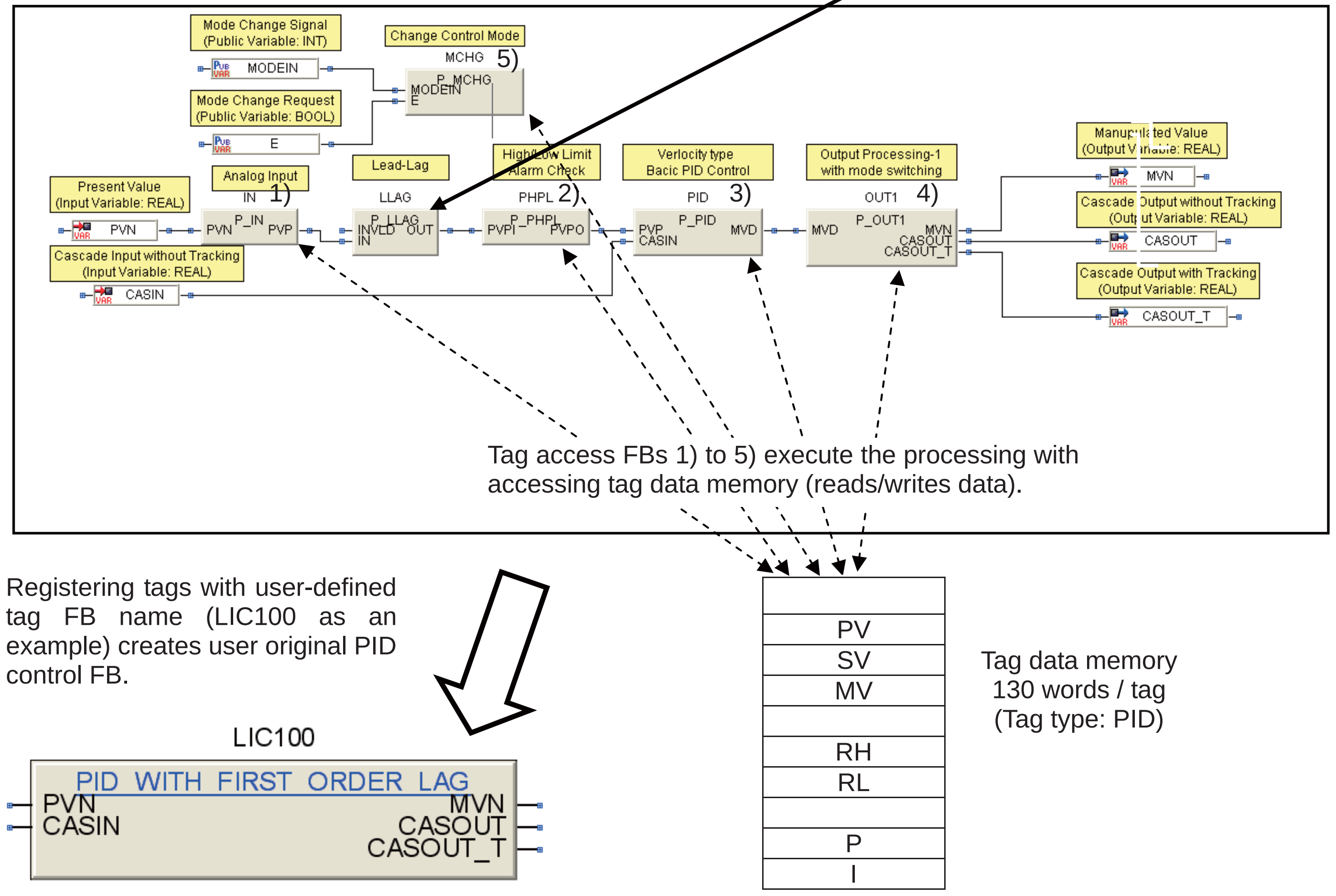
2.3.1 Structure of user-defined tag FB

(1) Basic creation of user-defined tag FB is to add 1 data structure (tag data memory structure corresponding to tag types such as velocity type PID, 2-degree-of-freedom PID) which user-defined tag FB to be created accesses and paste the following 5 tag access FBs to a user-defined tag FB sheet. Adding user created processing (such as square root, lead-lag, and function generator) completes user created PID control FB.

1) Analog input processing (P_IN)
2) High/low limit alarm check (P_PHPL)
3) Such as Velocity type PID control (P_PID)
4) Such as Output processing-1 with mode switching (P_OUT1)
5) Control mode change (P_MCHG)

User-defined tag FB sheet (FB name: PID with first order lag)

User created operation processing (lead-lag operation as an example) is added.



Tag access FBs 1) to 5) execute the processing with accessing tag data memory (reads/writes data).

Registering tags with user-defined tag FB name (LIC100 as an example) creates user original PID control FB.

Tag data memory
130 words / tag
(Tag type: PID)

(2) Tag access FB is a FB which executes operation processing with accessing (reads/writes data) tag data memory assigned to user-defined tag FB and can be pasted to a user-defined tag FB sheet only. Tag access FB cannot be pasted to a program sheet or user-defined FB sheet (different from user-defined tag FB sheet, user-defined FB does not have tag data memory).
The followings are the type of tag access FBs.

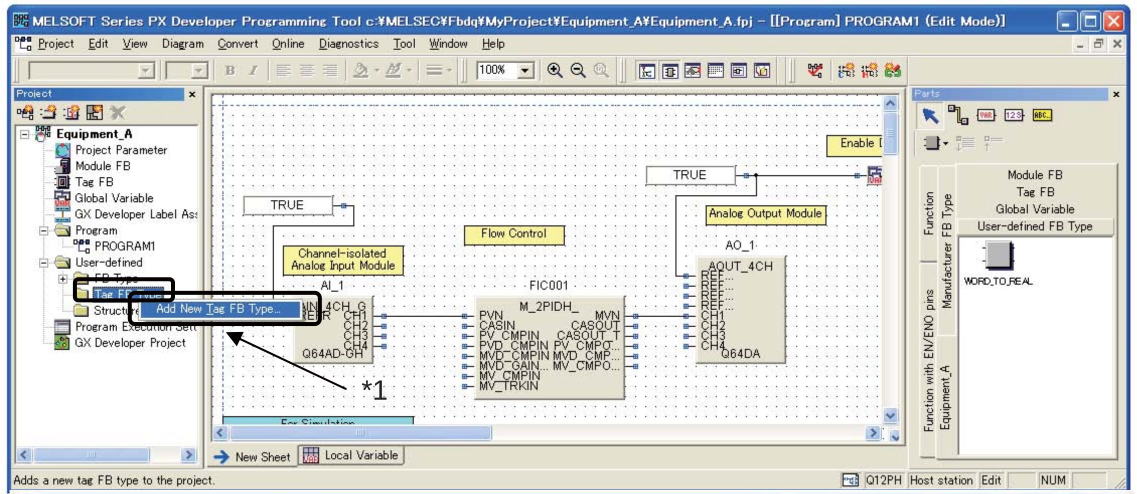
- Loop control operation FB
  Operation FB for loop control such as Velocity type PID control (P_PID), 2-degree-of-freedom PID control (P_2PID), Ratio control (P_R), High/low limit alarm check (P_PHPL).
- I/O control FB
  FB for I/O processing such as Analog input processing (P_IN), Pulse integration (P_PSUM), Output processing-1 with mode switching (P_OUT1), Manual output (P_MOUT).
- Special FB
  Control mode change (P_MCHG)

(3) All FBs, which include tag access FB, function part, variable parts, constant parts and comment parts, can be pasted to a user-defined tag FB sheet.

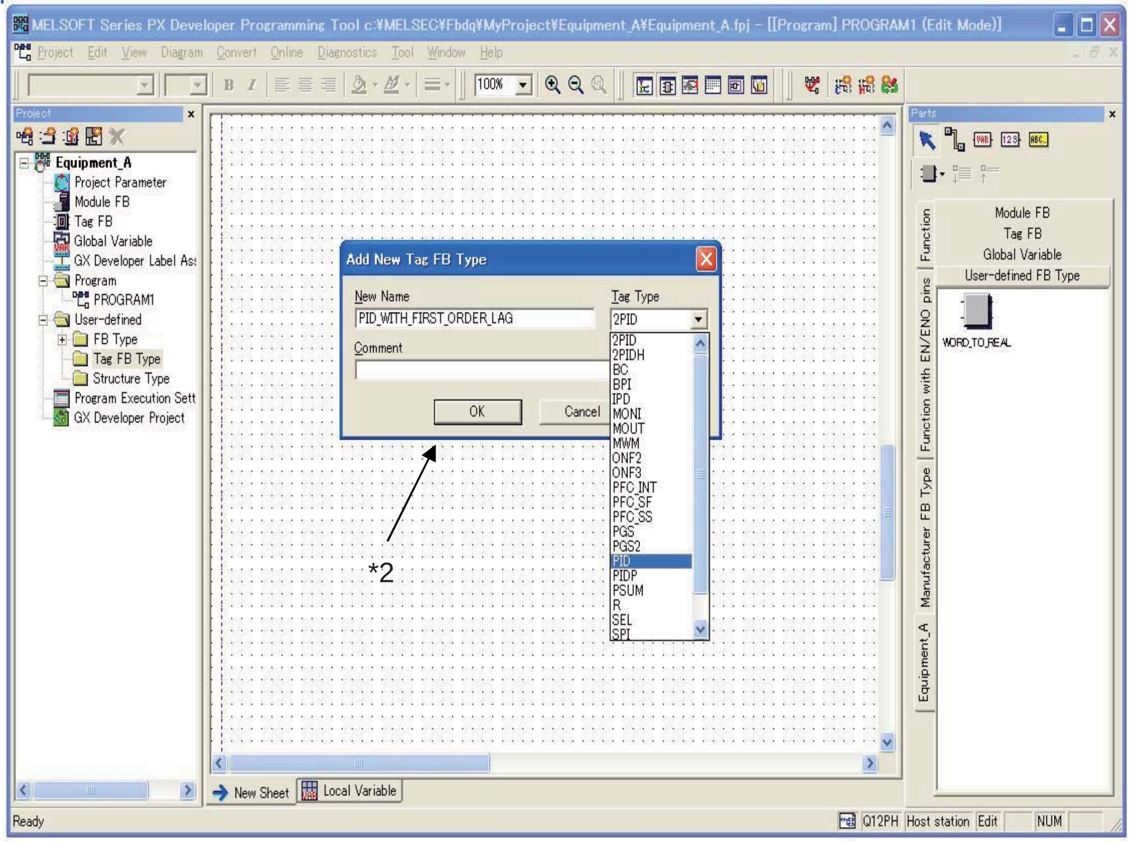
2.3.2 Programming procedure of user-defined tag FB

The following explains a programming procedure of PID_WITH_FIRST_ORDER_LAG as user-defined tag FB.

1) Right-click [User-defined] → [Tag FB Type] on the Project window, and click [Add New Tag FB Type...] (Following figure *1)
2) Enter 'PID_WITH_FIRST_ORDER_LAG' in the New Name on the Add New Tag FB Type window (following figure *2). Click ▾ to the right of Tag Type and select "PID". Comment is optional.
   - ••• As an example, user-defined tag FB name is "PID_WITH_FIRST_ORDER_LAG", and tag data memory type for user-defined tag FB is "PID"(PID type).



Adds a new tag FB type to the project.



New Name: PID_WITH_FIRST_ORDER_LAG

Tag Type: 2PID

Comment

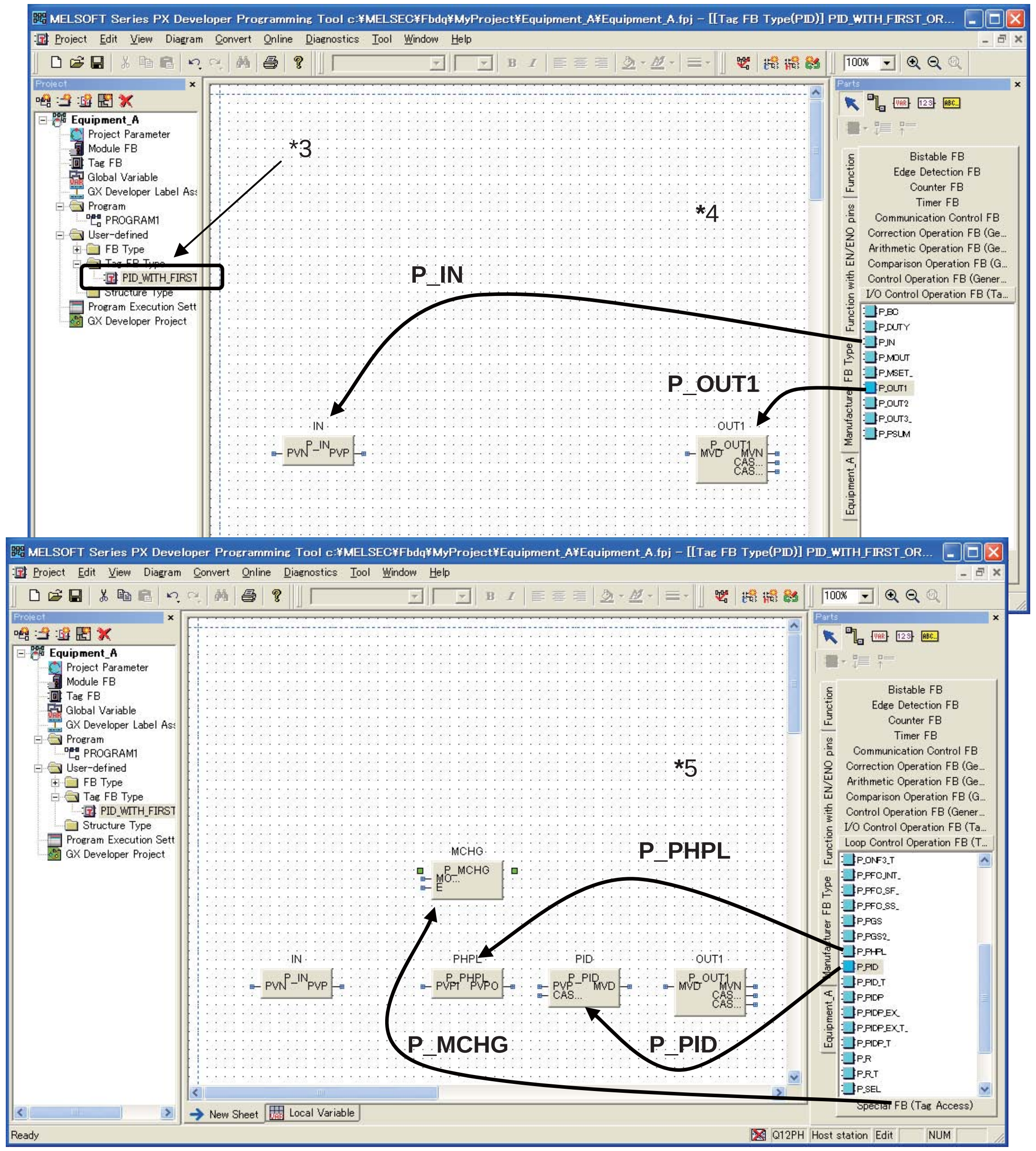
3) Double-click [User-defined] → [Tag FB Type] on the Project window and double-click

"PID_WITH_FIRST_ORDER_LAG" (Figure*3 in the next page)

4) User-defined tag FB "PID_WITH_FIRST_ORDER_LAG" sheet is displayed. Drag and drop P_IN (Analog input processing), P_OUT1 (Output processing-1 with mode switching) under [Manufacturer FB Type] → [I/O Control Operation FB (Tag access)] on the Parts window.

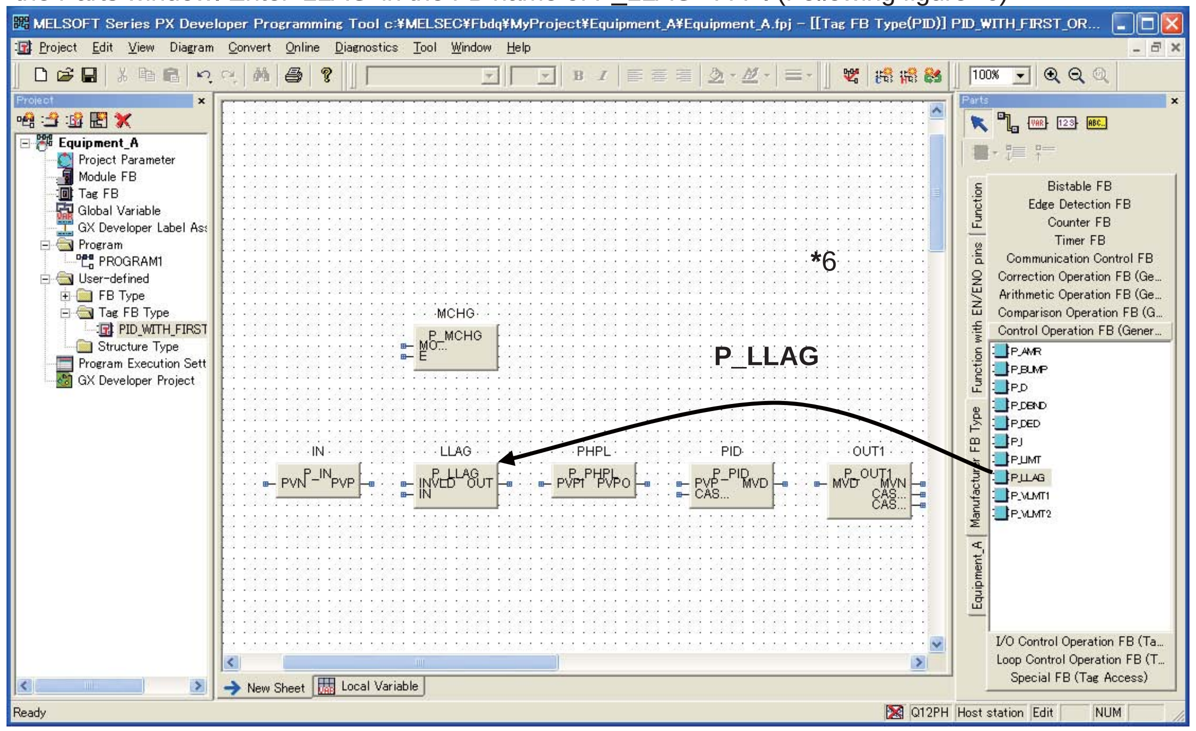
   Enter 'IN' (any word can be entered) in the FB name of P_IN "???", press the [Enter]. The Variable Reference dialog is displayed. Click the [OK] button.
   Enter 'OUT1' (any word can be entered) in the FB name of P_OUT1 "???" with same procedure as shown above. (Following figure *4)

5) Drag and drop P_PHPL (High/low limit alarm check), P_PID (Velocity type PID control) under [Manufacturer FB Type] → [Loop control operation FB (Tag access)], and drag and drop P_MCHG (Control mode change) under [Manufacturer FB Type] → [Special FB (Tag access)] on the Parts window. Enter 'PHPL' in the FB name of P_PHPL "???", 'PID' in FB name P_PID "???", 'MCHG' in FB name of P_MCHG "???". (Following figure *5)



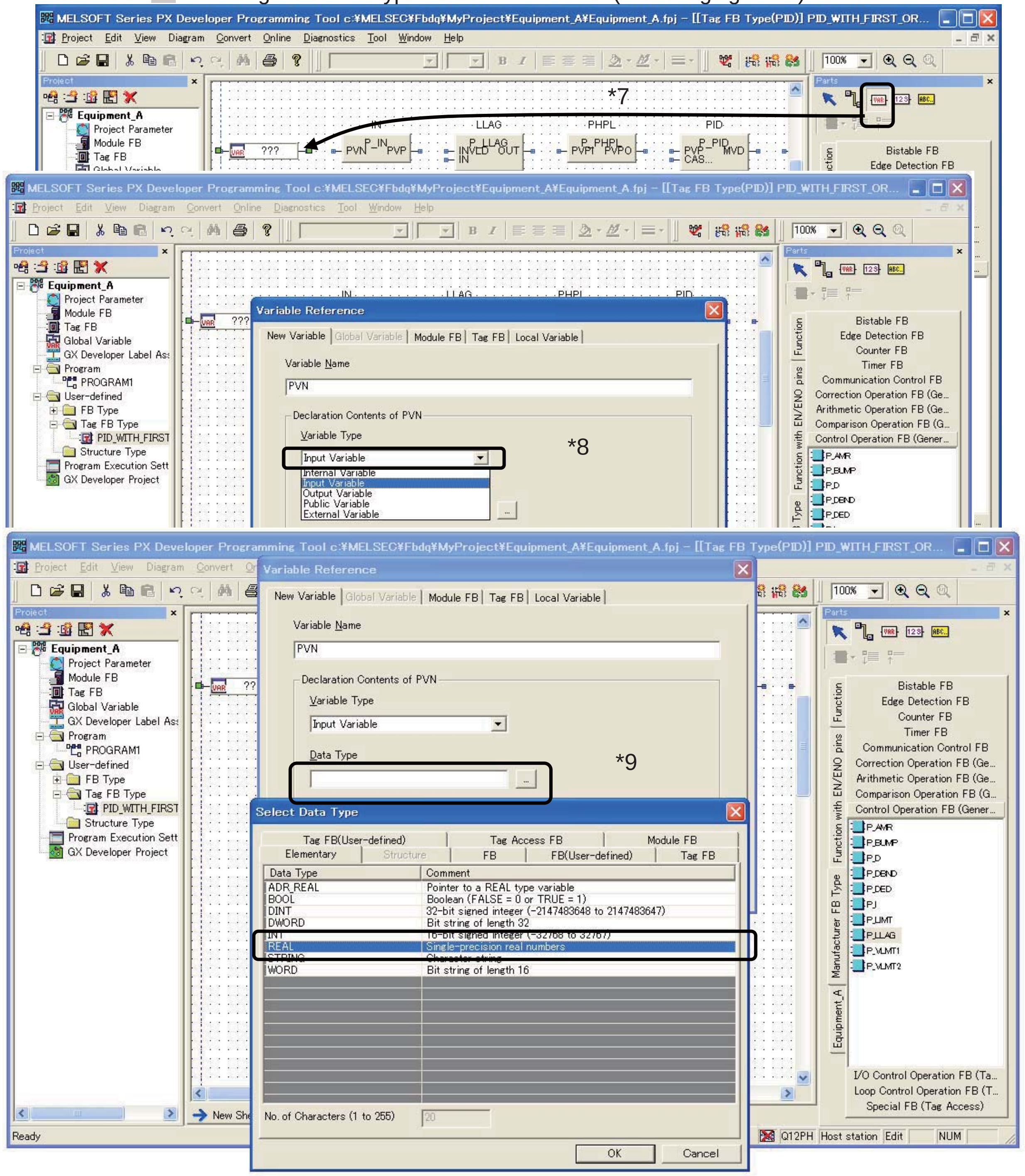
••• Basic 5 FBs (tag access FB) which compose a user created PID control are pasted as shown above. The following explains a user created operation processing (first order lag as an example).

6) Drag and drop P_LLAG (Lead-Lag) under [Manufacturer FB Type] → [Control Operation FB] on the Parts window. Enter 'LLAG' in the FB name of P_LLAG "???". (Following figure *6)



---

7) Drag and drop a variable part from Parts window (following figure *7), enter 'PVN' in the variable name "???" (any name can be entered), and then press the [Enter] key. The Variable Reference dialog is displayed. Click ▼ to the right of Variable Type and select "Input Variable". (Following figure *8)
Then click ... to the right of Data Type and select "REAL". (Following figure *9)



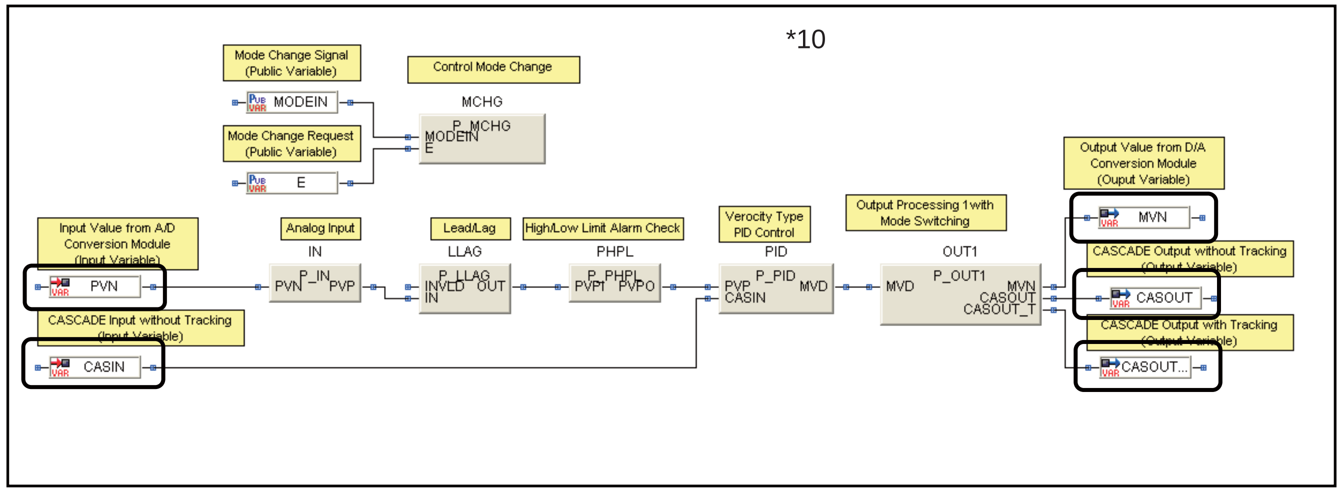
Data Type	Comment
ADR_REAL	Pointer to a REAL type variable
BOOL	Boolean (FALSE = 0 or TRUE = 1)
DINT	32-bit signed integer (-2147483648 to 2147483647)
DWORD	Bit string of length 32
INT	16-bit signed integer (-32768 to 32767)
REAL	Single-precision real numbers
STRING	Character string
WORD	Bit string of length 16

··· An input variable of the user-defined tag FB is created in 7). Selecting input variable as a variable part type displays an input pin on the user-defined tag FB. In this example, the input pin whose name is "PVN" and data type is REAL (single precision floating decimal) is created.

8) Paste the following variables with the same procedure as shown above. (Following figure *10)

- Variable name "CASIN"
  Variable Type: Input Variable, Data Type: REAL, Definition: cascade input from primary loop in the cascade connection of PID.
- Variable name "MVN"
  Variable Type: Output Variable, Data Type: REAL, Definition: output value of PID operation (manipulated variable MV) and indicates DA conversion value (such as 0 to 4000, 0 to 8000) to analog output module.
- Variable name "CASOUT"
  Variable Type: Output Variable, Data Type: REAL, Definition: output value (% value of manipulated variable MV) to secondary loop in the cascade connection of PID (without tracking).
- Variable name "CASOUT_T"
  Variable Type: Output Variable, Data Type: ADR_REAL (indirect address to REAL type variable), Definition: output value (memory address in which this value is stored, not % value of manipulated variable MV) to secondary loop in the cascade connection of PID (with tracking)

*10



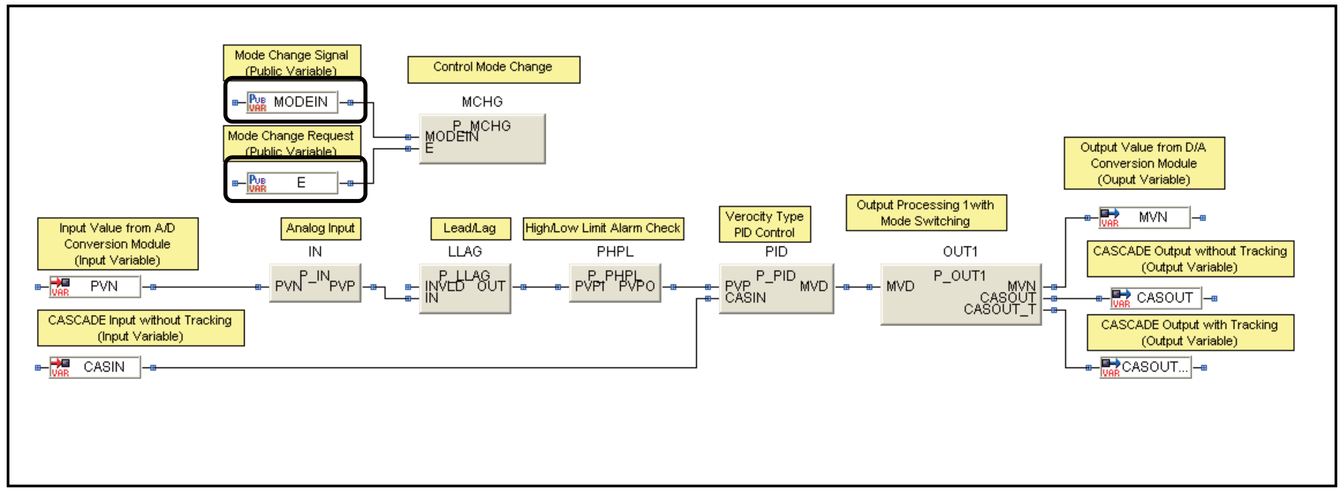
··· Selecting output variable as a variable part type displays the output pins on the user-defined tag FB. In this example, output pins whose names are "MVN", "CASOUT", "CASOUT_T" are created.

9) Paste the following variables. (Following figure *11)

- Variable name "MODEIN"
  Variable Type: Public Variable, Data Type: INT, Definition: Mode change signal to P_MCHG (Change control mode FB). Control mode - 1: MANUAL, 2: AUTO, 3: CASCADE, 4: COMPUTER MV, 5: COMPUTER SV, 6: CASCADE DIRECT
- Variable name "E"
  Variable Type: Public Variable, Data Type: BOOL Definition: Control mode change request (TRUE: Execute, FALSE: Stop) to P_MCHG (Change control mode FB)

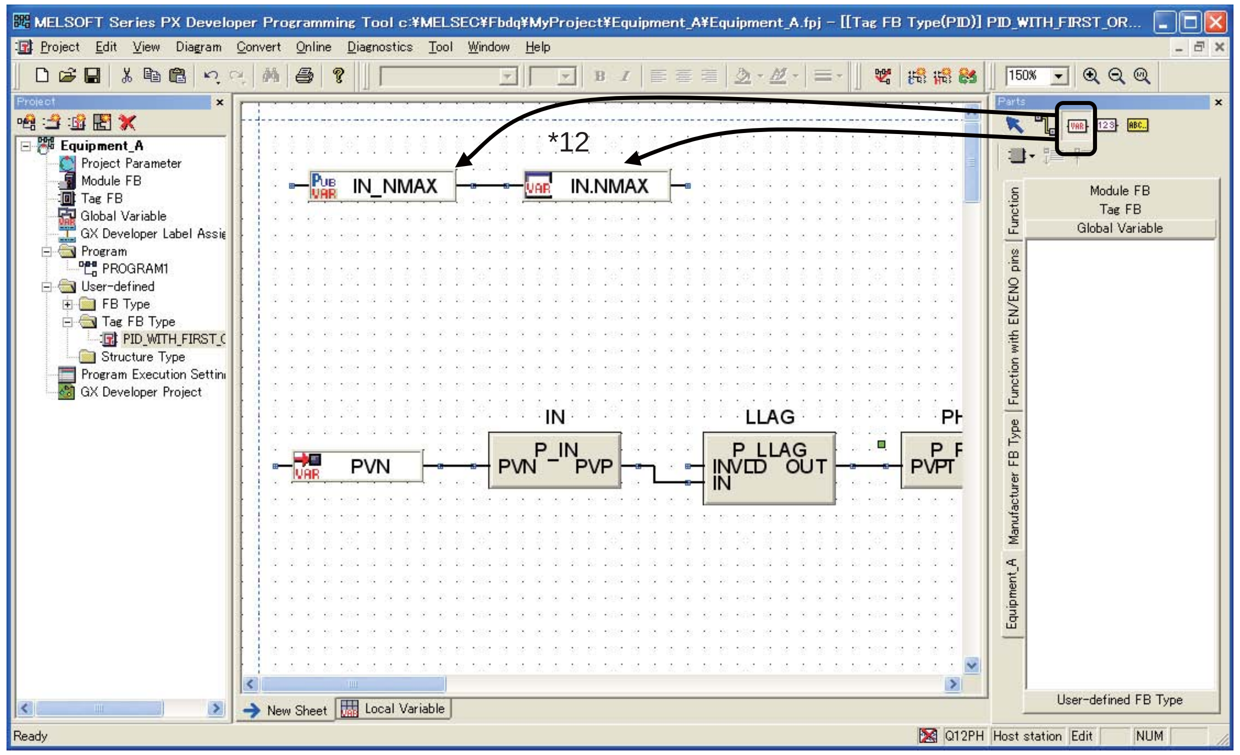
  ··· Public variable is a variable which is inside of a FB and can also be accessed (read/written data) by any FBs other than the FB. (Internal variable cannot be accessed by any FBs other than the FB).
  Furthermore, selecting public variable as variable type displays public variable on the property window of user-defined tag FB (or user-defined FB) and enables to input an initial value.
  Do not define the public variable as input pin (input variable) or output pin (output variable), use as a parameter which can be accessed by outside of the user-defined tag FB.

*11



10) Connect each variable and FB with connector (following figure *11).

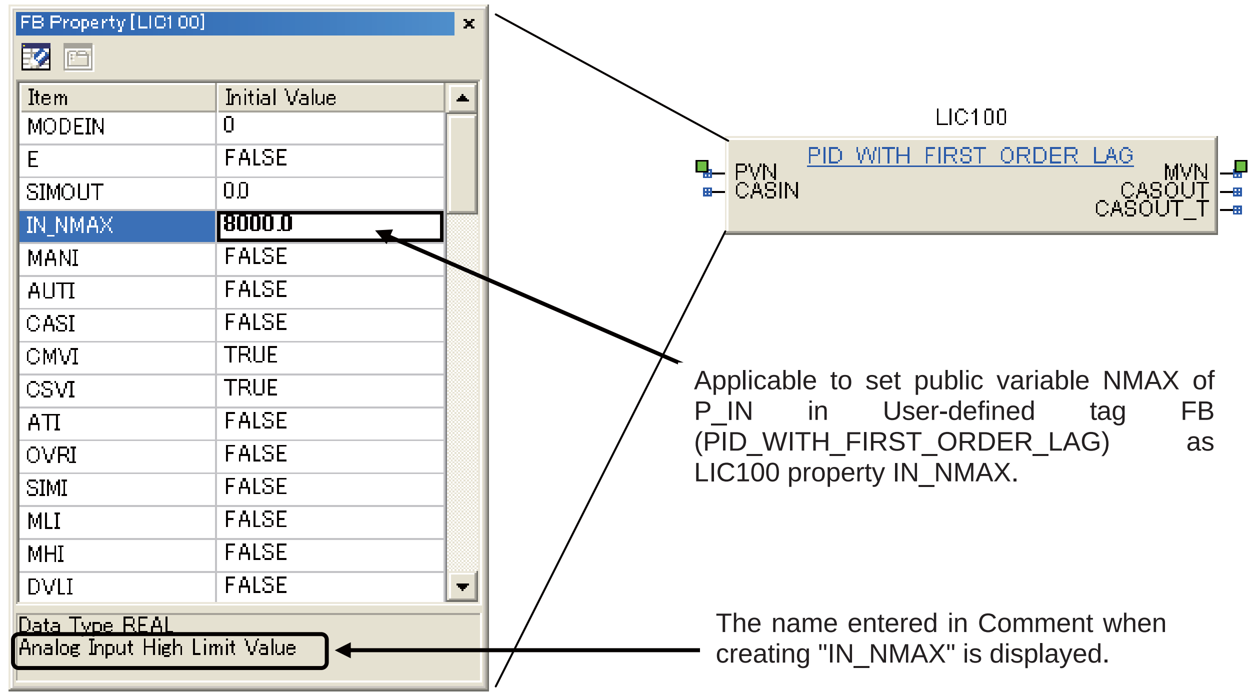
11) Drag and drop 2 variable parts from the Parts window, enter 'IN_NMAX' in the variable name "???" on the left side, and press the [Enter] key. The Variable Reference dialog is displayed. Click ▼ to the right of Variable Type and select "Public Variable". Then click ... to the right of Data Type and select "REAL". Enter 'Analog input high limit' in the Comment.
Enter 'IN.NMAX' in the variable name "???" on the right side, and press the [Enter] key.
Connect variable "IN_NMAX" and "IN.NMAX" with connector (following figure *12).



••• This procedure explains the processing of substituting public variable "IN_NMAX" of user-defined tag FB to NMAX value (Analog input high limit) which is public variable(operation constant) of P_IN (FB name is IN in analog PV processing FB) (Refer to the next page)

FB property window of "LIC100"

Substance of User-defined tag FB



FB Property [LIC100]

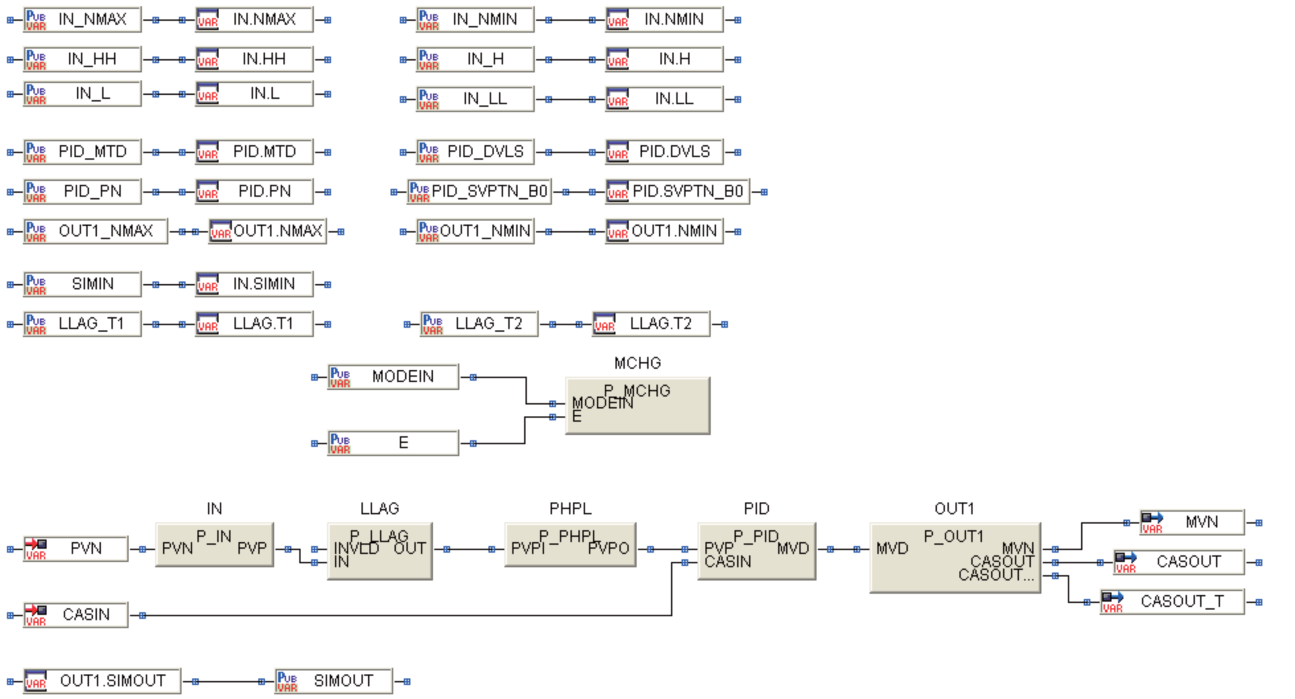
Item	Initial Value
MODEIN	0
E	FALSE
SIMOUT	0.0
IN_NMAX	8000.0
MANI	FALSE
AUTI	FALSE
CASI	FALSE
CMVI	TRUE
CSVI	TRUE
ATI	FALSE
OVRI	FALSE
SIMI	FALSE
MLI	FALSE
MHI	FALSE
DVLI	FALSE

Data Type REAL
Analog Input High Limit Value

Applicable to set public variable NMAX of P_IN in User-defined tag FB (PID_WITH_FIRST_ORDER_LAG) as LIC100 property IN_NMAX.

The name entered in Comment when creating "IN_NMAX" is displayed.

12) Create a substitution program to display public variables on the property window of user-defined tag FB substance as shown below (the same procedure as 11)).

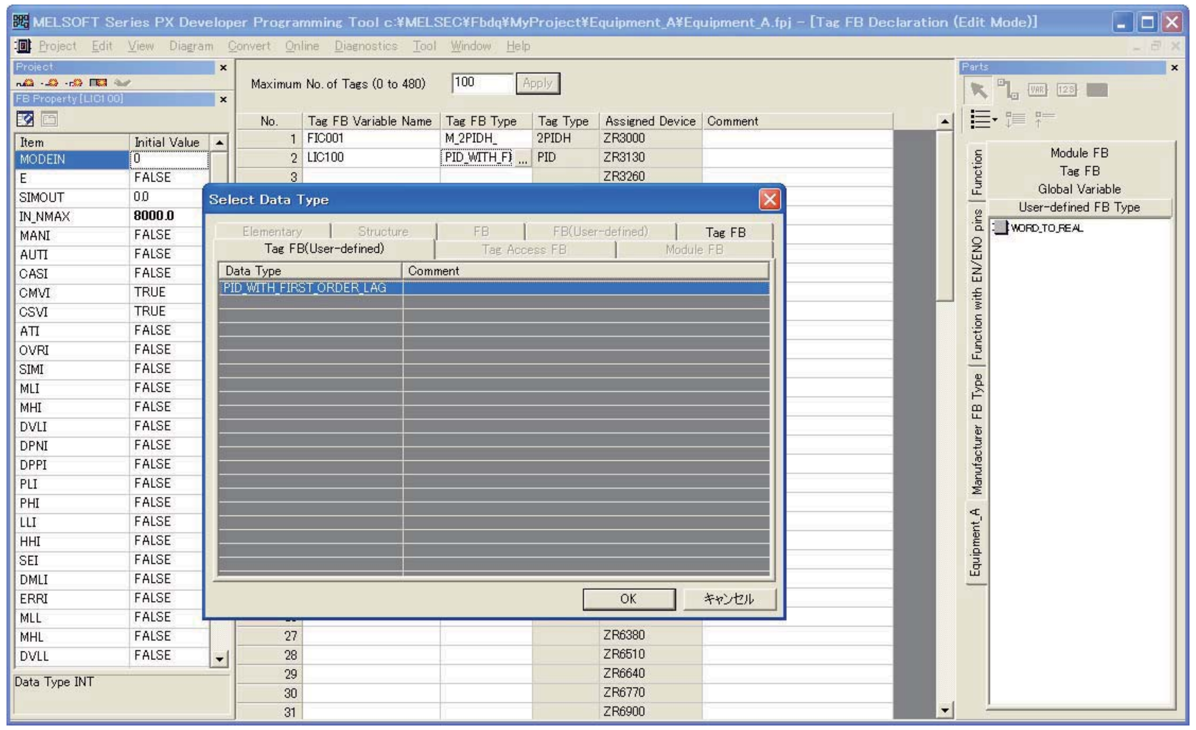


Variable Name	Variable Type	Data Type	Comment
IN_NMAX	Public Variable	REAL	Analog input high limit
IN.NMAX	(Internal Variable)	(REAL)	
IN_NMIN	Public Variable	REAL	Analog input low limit
IN.NMIN	(Internal Variable)	(REAL)	
IN_HH	Public Variable	REAL	Analog input high limit range error
IN.HH	(Internal Variable)	(REAL)	
IN_H	Public Variable	REAL	Analog input high limit range error reset
IN.H	(Internal Variable)	(REAL)	
IN_LL	Public Variable	REAL	Analog input low limit range error
IN.LL	(Internal Variable)	(REAL)	
IN_L	Public Variable	REAL	Analog input low limit range error reset
IN.L	(Internal Variable)	(REAL)	
PID_MTD	Public Variable	REAL	Derivative gain
PID.MTD	(Internal Variable)	(REAL)	
PID_DVLS	Public Variable	REAL	Large deviation alarm hysteresis
PID.DVLS	(Internal Variable)	(REAL)	
PID_PN	Public Variable	INT	Reverse action/direct action
PID.PN	(Internal Variable)	(INT)	
PID_SVPTN_B0	Public Variable	BOOL	Setting value (SV) used
PID.SVPTN_B0	(Internal Variable)	(BOOL)	
OUT1_NMAX	Public Variable	REAL	Analog output conversion high limit
OUT1.NMAX	(Internal Variable)	(REAL)	
OUT1_NMIN	Public Variable	REAL	Analog output conversion low limit
OUT1.NMIN	(Internal Variable)	(REAL)	
LLAG_T1	Public Variable	REAL	Lag time
LLAG.T1	(Internal Variable)	(REAL)	
LLAG_T2	Public Variable	REAL	Lead time
LLAG.T2	(Internal Variable)	(REAL)	
SIMIN	Public Variable	REAL	Simulation input
IN.SIMIN	(Internal Variable)	(REAL)	
OUT1.SIMOUT	(Internal Variable)	(REAL)	
SIMOUT	Public Variable	REAL	Simulation output

To create "FB name.XXX" such as "IN.NMAX" and "PID.MTD", enter such as 'IN.NMAX' and 'PID.MTD' in the variable name, and press the [Enter] key. (The Variable Reference dialog is not displayed.)

13) Double-click [Tag FB] on the Project window to display the "Tag FB Declaration" window (following figure *13). Enter a name of the user-defined tag FB substance in a Tag FB Variable Name cell (enter "LIC100" as an example).
Click ... to the right of the "Tag FB Type" cell, select the <<Tag FB (User-defined)>> tab on the Select Data Type dialog, select "PID_WITH_FIRST_ORDER_LAG" as data type, and then click the [OK] button.
Now, the User-defined tag FB can be used as a part.

*13



MELSOFT Series PX Developer Programming Tool c:¥MELSEC¥Fbdq¥MyProject¥Equipment_A¥Equipment_A.fpj - [Tag FB Declaration (Edit Mode)]

Maximum No. of Tags (0 to 480) 100

No.	Tag FB Variable Name	Tag FB Type	Tag Type	Assigned Device	Comment
1	FIC001	M_2PIDH_	2PIDH	ZR3000	
2	LIC100	PID_WITH_FI	PID	ZR3130	
3				ZR3260	
27				ZR6380	
28				ZR6510	
29				ZR6640	
30				ZR6770	
31				ZR6900	

FB Property [LIC100]

Item	Initial Value
MODEIN	0
E	FALSE
SIMOUT	0.0
IN_NMAX	8000.0
MANI	FALSE
AUTI	FALSE
CASI	FALSE
CMVI	TRUE
CSVI	TRUE
ATI	FALSE
OVRI	FALSE
SIMI	FALSE
MLI	FALSE
MHI	FALSE
DVLI	FALSE
DPNI	FALSE
DPPI	FALSE
PLI	FALSE
PHI	FALSE
LLI	FALSE
HHI	FALSE
SEI	FALSE
DMLI	FALSE
ERRI	FALSE
MLL	FALSE
MHL	FALSE
DVLL	FALSE

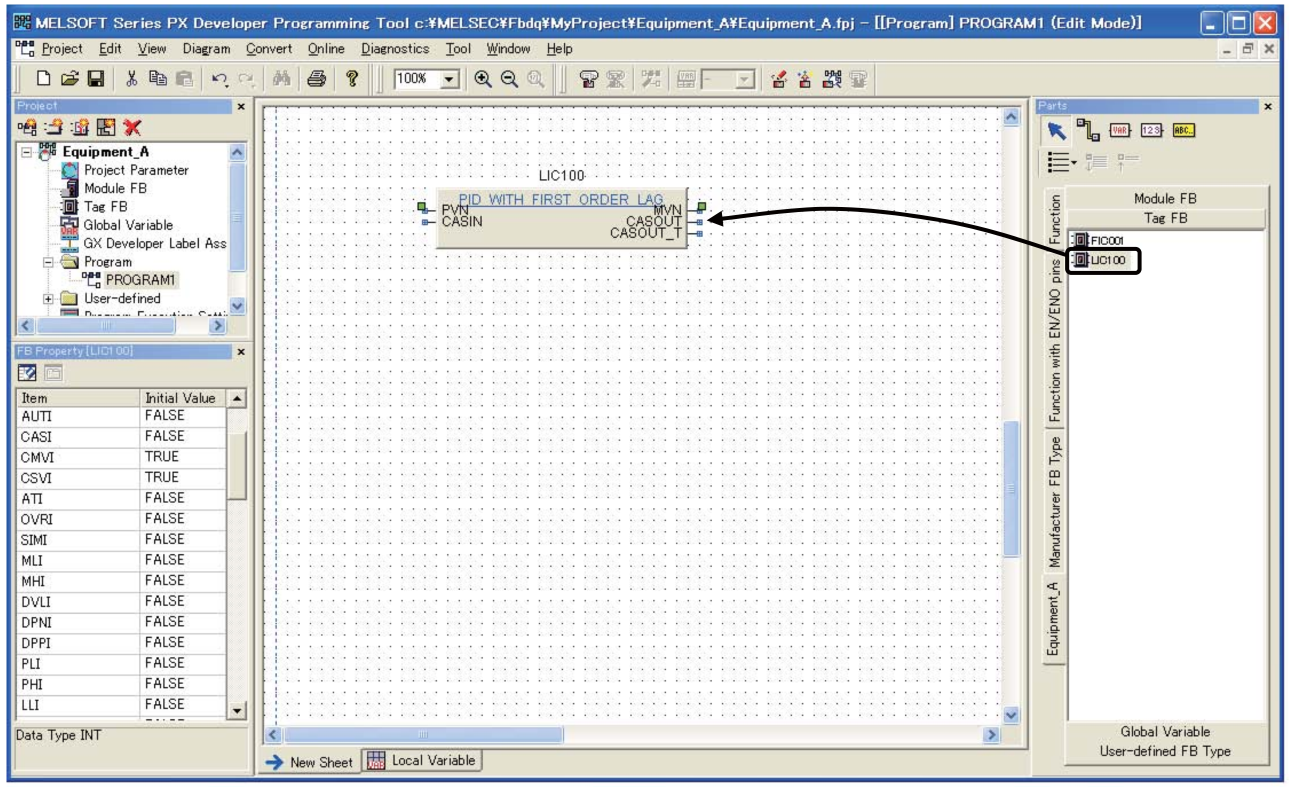
Data Type INT

Select Data Type

Data Type	Comment
PID_WITH_FIRST_ORDER_LAG	

14) When using the User-defined tag FBs in a program, drag and drop the substance of user-defined tag FB "LIC100" which registered in 13) on a sheet from the Tag FB area of the Parts window. (Following figure *14).

*14



Item	Initial Value
AUTI	FALSE
CASI	FALSE
CMVI	TRUE
CSVI	TRUE
ATI	FALSE
OVRI	FALSE
SIMI	FALSE
MLI	FALSE
MHI	FALSE
DVLI	FALSE
DPNI	FALSE
DPPI	FALSE
PLI	FALSE
PHI	FALSE
LLI	FALSE

Data Type INT

---

3 PROGRAM/FB EXAMPLES IN PX Developer

This section describes the practical program examples (include user-defined FB/user-defined Tag FB）with FBD of PX Developer.

3.1 List of Programs

【 3.2 Program Examples (common processing) 】

No.	Item	Description
3.2.1	PID simulation loopback	The examples of PID simulation user-defined FB.
3.2.2	Control mode change (MAN_AUTO)	The examples of control mode change (MAN_AUTO) user-defined FB.
3.2.3	Control mode change (MAN_AUTO_CAS)	The examples of control mode change (MAN_AUTO_CAS) user-defined FB.
3.2.4	Control mode change (MAN_AUTO_CAS_CMV_CSV)	The examples of control mode change (MAN_AUTO_CAS_CMV_CSV) user-defined FB.
3.2.5	Disabling control mode change	The examples of disabling control mode change (MAN, AUTO, CAS, CMV, CSV) to specified mode.
3.2.6	Sensor error loop stop	The examples of executing the loop stop and switching the control mode to MANUAL automatically when a sensor error occurs.
3.2.7	Count value to analog instantaneous value	The examples of user-defined FB which converts a count current value to an analog instantaneous value.
3.2.8	Count difference value (QD60P8-G) to analog instantaneous value	The examples of user-defined FB which converts an every-second count difference value to be input from a pulse input module QD60P8-G to an analog instantaneous value.
3.2.9	Sensor burnout preset	The examples of considering a process variable as an regular value when a sensor burnout occurs
3.2.10	Writing MV, SV from upper computer	The examples of writing MV, SV to a tag in CMV mode or CSV mode.
3.2.11	REAL type ×N times to INT type conversion	The examples of user-defined FB which multiplies real type data by N (10, 100, …) and then converts it to integer.
3.2.12	WORD type to REAL type conversion	The examples of user-defined FB which converts word output data such as CC-Link module FB to real type which is a data type of loop control input.

【 **3.3 Program Examples (loop control related)** 】

No.	Item	Description
3.3.1	Cascade control	The examples of Cascade control.
3.3.2	Selection control (input high selector)	The examples of Selection control (input high selector).
3.3.3	Ratio control	The examples of ratio control with Pulse Factor Controller.
3.3.4	Output override (low selector)	The examples of output overrides (low selector) control with Loop selector.
3.3.5	Process variable tracking (when upper is not loop tag)	The examples of process variable tracking when upper is not loop tag.
3.3.6	Process variable tracking (In MAN mode switching)	The examples of process variable tracking in MAN mode switching.
3.3.7	Heating-cooling program control	The examples of split control and program control for heating/ cooling.
3.3.8	Cross limit control	The examples of a control which improves combustion efficiency with executing appropriate air fuel ratio control at such as a combustion furnace.
3.3.9	Temperature correction (with square root)	The examples of temperature correction with square root process.
3.3.10	Pressure correction (with square root)	The examples of pressure correction with square root process.
3.3.11	Temperature/pressure correction (with square root)	The examples of temperature/pressure correction with square root process.
3.3.12	First order lag dead time	The examples of first order lag + dead time processing.
3.3.13	Dead time compensation	The examples of user-defined FB which executes (1－e-LS) / (1 + TS) processing.
3.3.14	2 OUT OF 3	The examples of user-defined FB in case of one sensor out of three failures, imports normal value with the other two sensors.
3.3.15	Deviation variable gain PID	The examples of user-defined tag FB which has deviation input and broken line correction.

【 **3.4 Program Examples (digital/sequence control related)** 】

No.	Item	Description
3.4.1	Single solenoid	The examples of opening and closing of single solenoid valve
3.4.2	Double solenoid	The examples of opening and closing of double solenoid valve

---

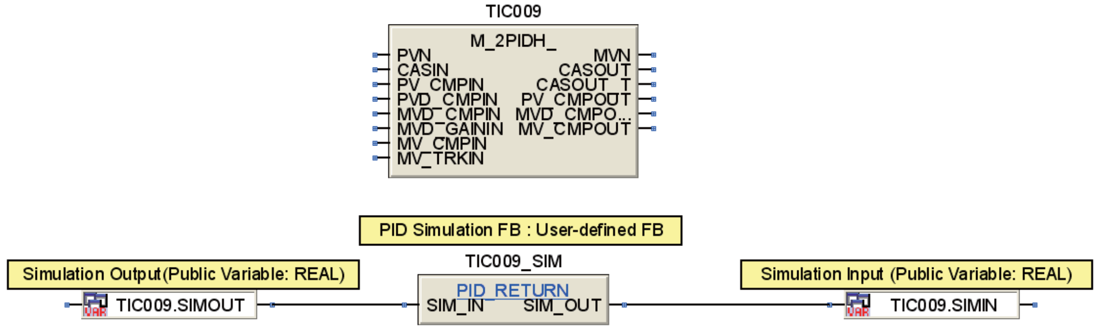
3.2 Program examples (common processing)

3.2.1 PID simulation loopback

Function	• Return output of a Tag FB to input and executes loop simulation. • For details of user-defined FB (PID_RETURN) used in this example, refer to (2) in this section.

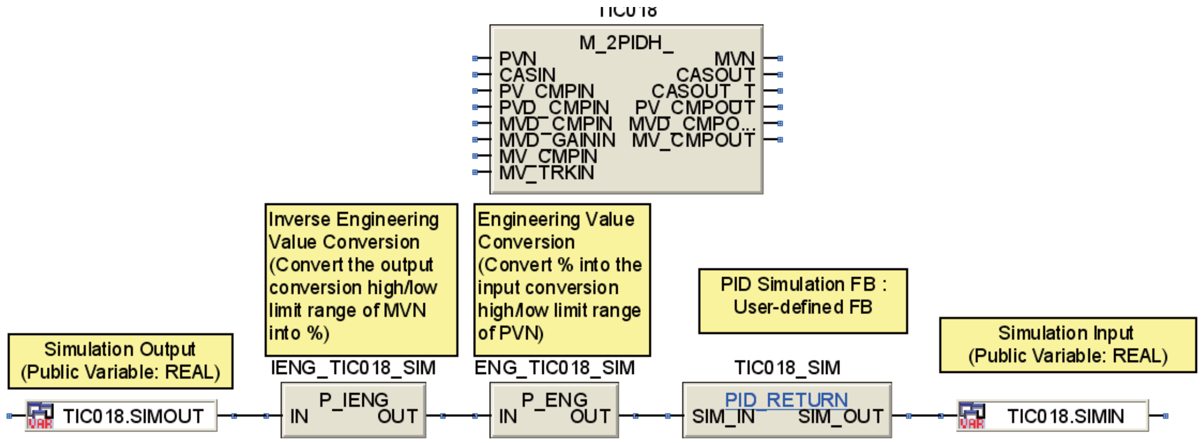
(1) Program example

1) When input conversion high/low limit and output conversion high/low limit of tag FB are same



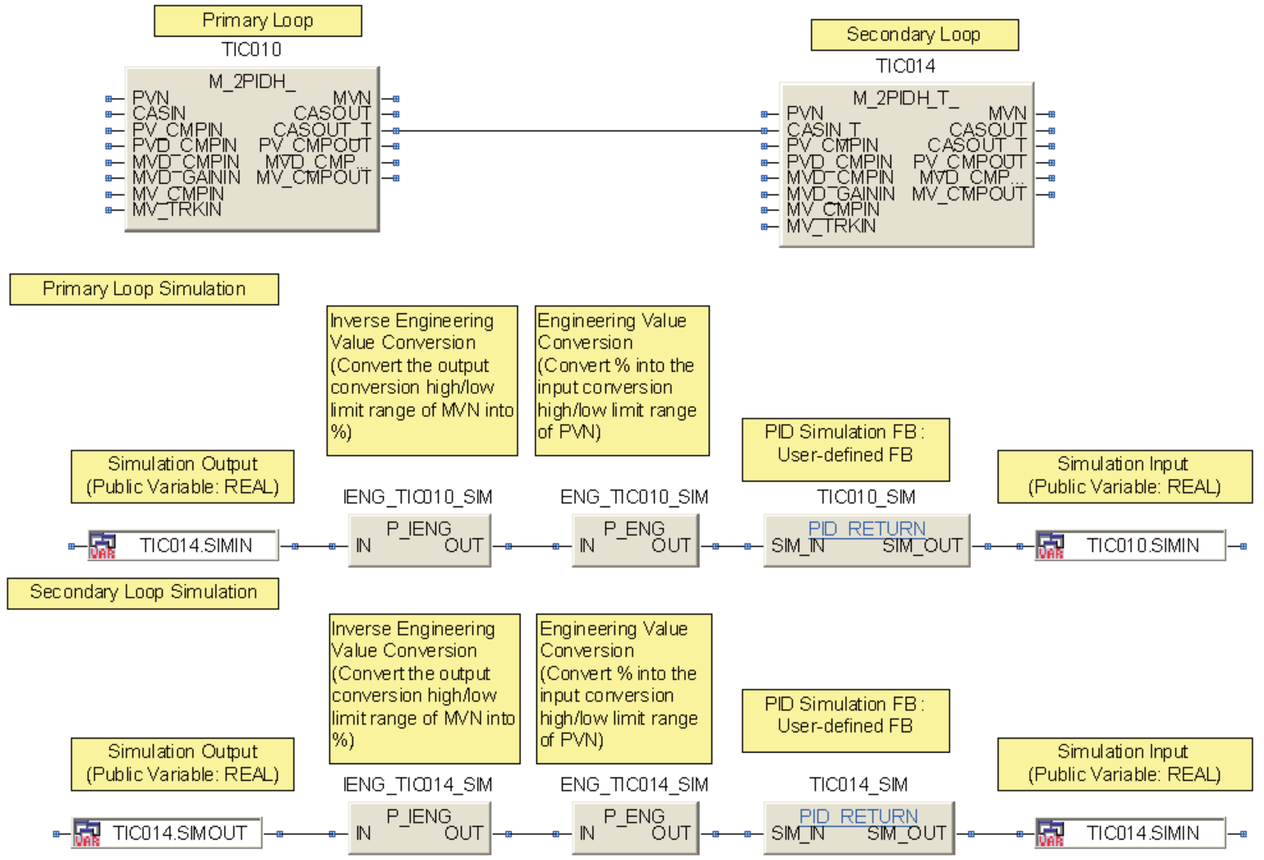
2) When input conversion high/low limit and output conversion high/low limit of tag FB are different

Point	• When input conversion high/low limit of input variable PVN and output conversion high/low limit of output variable MVN of Tag FB (TIC018) are different. Convert the range (output conversion high/low limit range of MVN) whose tag name is .SIMOUT to the range (input conversion high/low limit range of input variable PVN) whose tag name is .SIMIN with the inverse engineering unit conversion (P_IENG) and engineering unit conversion (P_ENG).



3) When input conversion high/low limit and output conversion high/low limit of primary, secondary tag FBs are different in the cascade control

Point	• When input conversion high/low limit of input variable PVN and output conversion high/low limit of output variable MVN of primary, secondary loop tags are different in the cascade control. Convert the range (output conversion high/low limit range of MVN) whose tag name is .SIMOUT to the range (input conversion high/low limit range of input variable PVN) whose tag name is .SIMIN with the inverse engineering unit conversion (P_IENG) and engineering unit conversion (P_ENG).

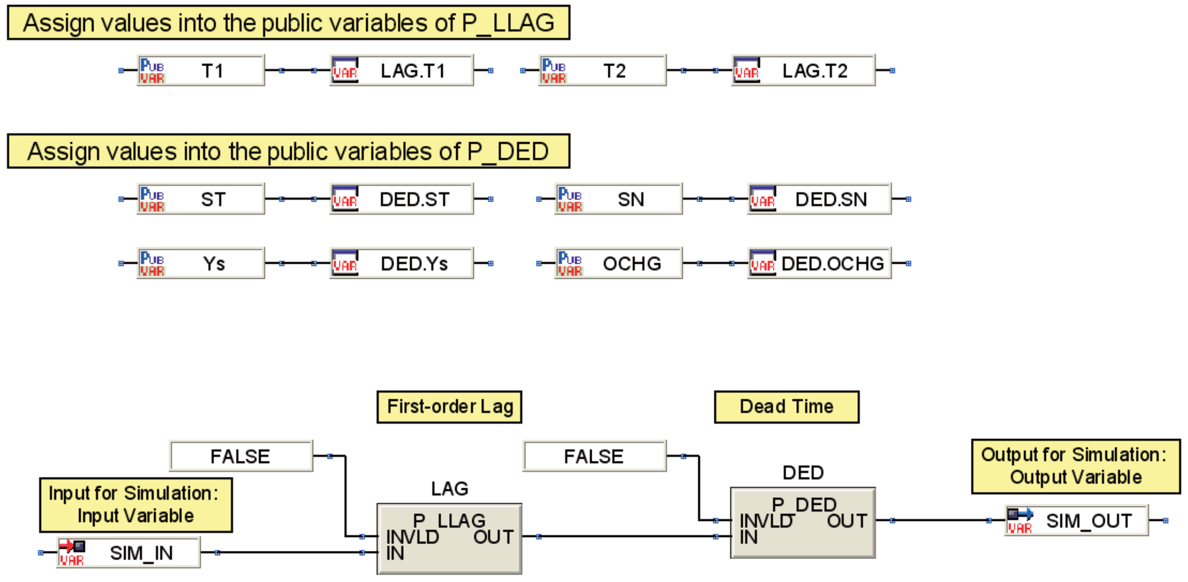


---

(2) Example of user-defined FB

1) PID simulation loopback User-defined FB (PID_RETURN)

Point	• Execute first order lag and dead time processing and output the return data for simulation.



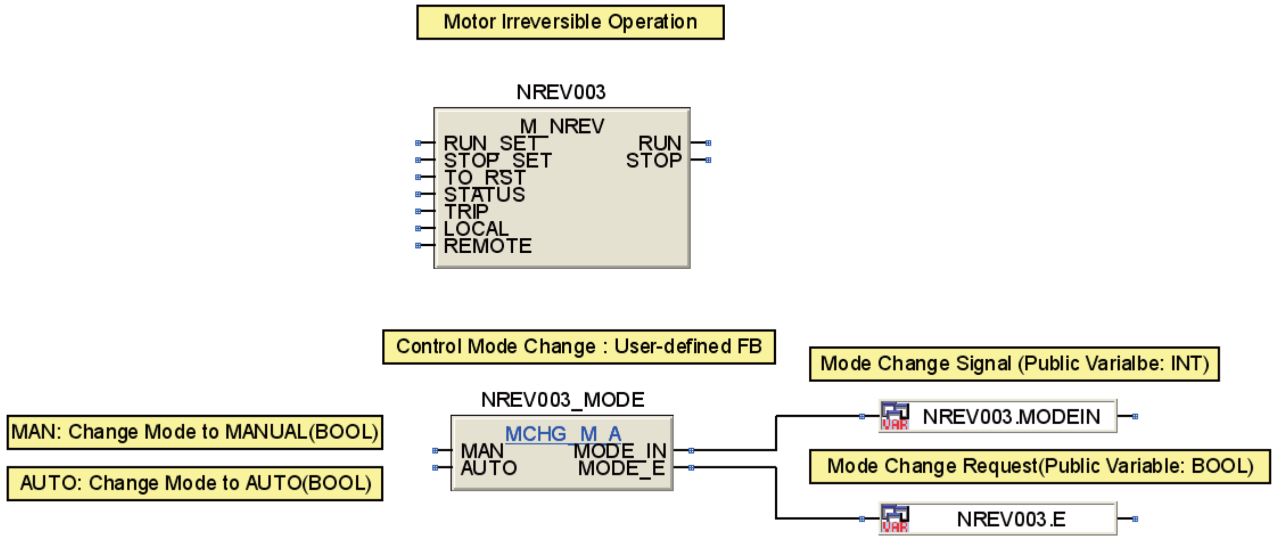
Pin	Variable name	Variable type	Data type	Contents
Input	SIM_IN	Input variable	REAL	Input first order lag, dead time for simulation
Output	SIM_OUT	Output variable	REAL	Output first order lag, dead time for simulation

3.2.2 Control mode change (MAN_AUTO)

Function	• Execute mode change (MAN, AUTO) for motor irreversible operation (M_NREV). • For details of user-defined FB (MCHG_M_A) used in this example, refer to (2) in this section.

(1) Program example

1) Control mode change (MAN_AUTO)

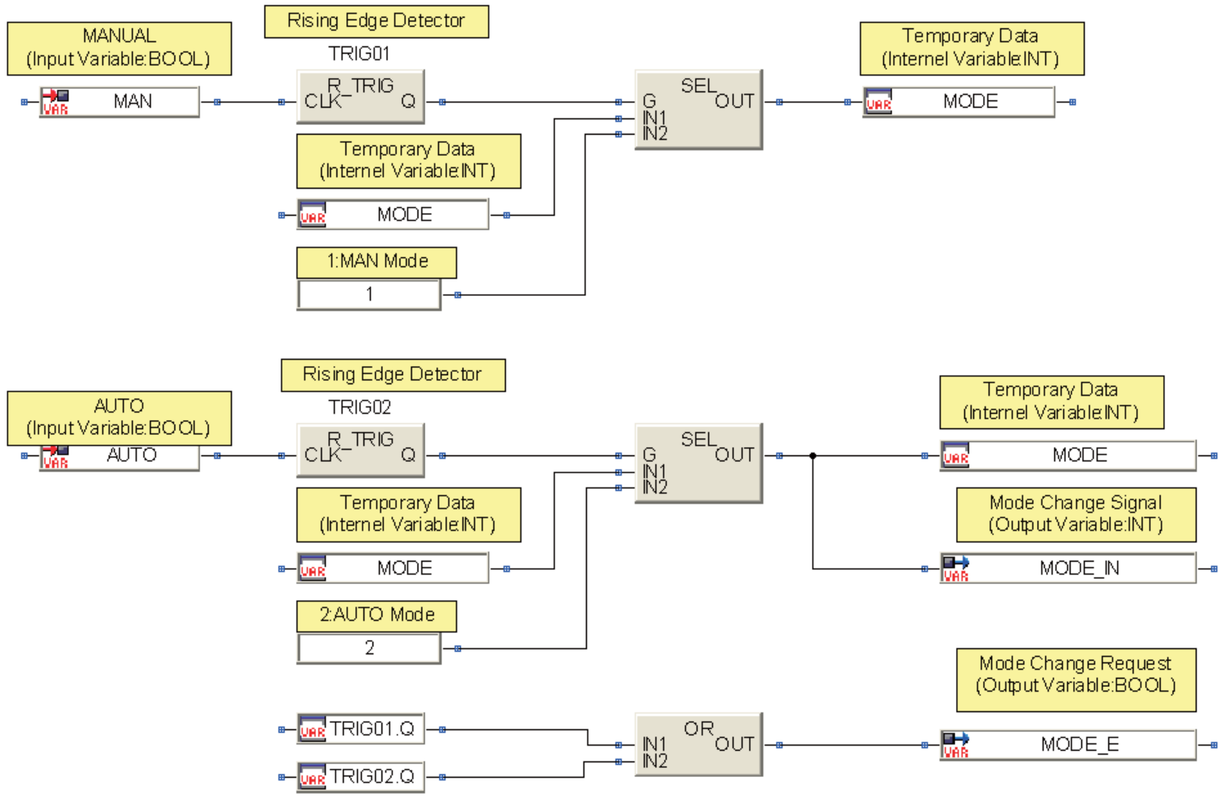


---

(2) Example of user-defined FB

1) Control mode change (MAN_AUTO) User-defined FB (MCHG_M_A)

Point	• Switching the signal for MAN from FALSE to TRUE changes the control mode to MAN mode. • Switching the signal for AUTO from FALSE to TRUE changes the control mode to AUTO mode.



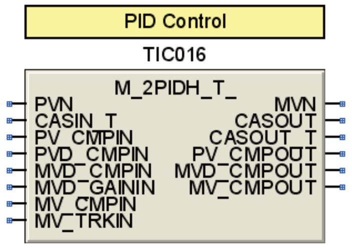
Pin	Variable name	Variable type	Data type	Contents
Input	MAN	Input variable	BOOL	TRUE: MAN mode
Input	AUTO	Input variable	BOOL	TRUE: AUTO mode
Output	MODE_IN	Output variable	INT	Control mode (1: MAN, 2: AUT)
Output	MODE_E	Output variable	BOOL	Change request (TRUE: Execute)

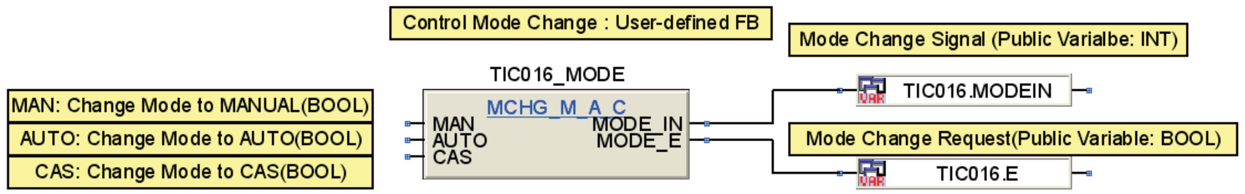
3.2.3 Control mode change (MAN_AUTO_CAS)

Function	• Execute mode change (MAN, AUTO, CAS) of 2-degree-of-freedom advanced PID control (M_2PIDH_T_). • For details of user-defined FB (MCHG_M_A_C) used in this example, refer to (2) in this section.

(1) Program example

1) Control mode change (MAN_AUTO_CAS)

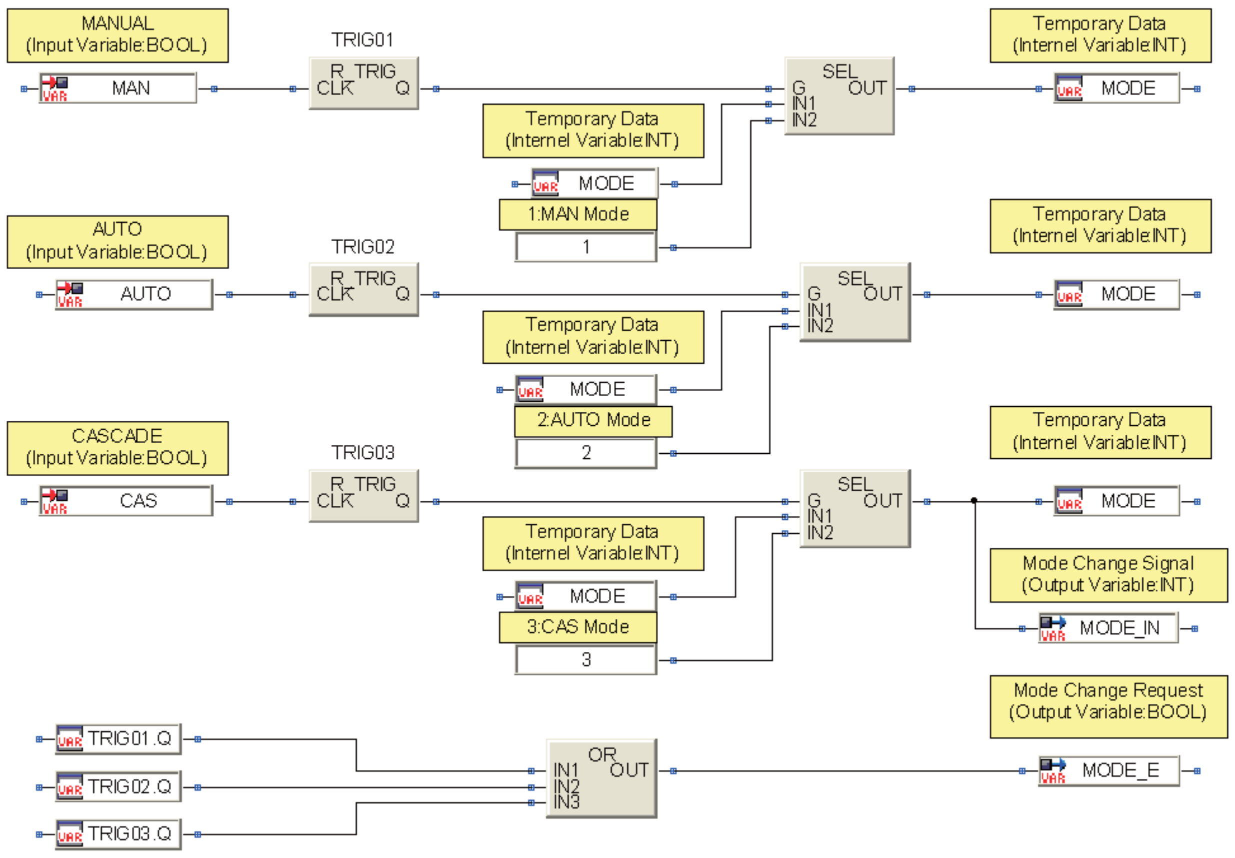




(2) Example of user-defined FB

1) Control mode change (MAN_AUTO_CAS) User-defined FB (MCHG_M_A_C)

Point	• Switching the signal for MAN from FALSE to TRUE changes the control mode to MAN mode. • Switching the signal for AUTO from FALSE to TRUE changes the control mode to AUTO mode. • Switching the signal for CAS from FALSE to TRUE changes the control mode to CAS mode.



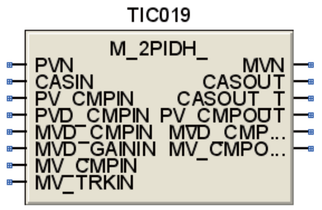
Pin	Variable name	Variable type	Data type	Contents
Input	MAN	Input variable	BOOL	TRUE: MAN mode
Input	AUTO	Input variable	BOOL	TRUE: AUTO mode
Input	CAS	Input variable	BOOL	TRUE: CAS mode
Output	MODE_IN	Output variable	INT	Control mode (1:MAN, 2:AUT, 3:CAS)
Output	MODE_E	Output variable	BOOL	Change request (TRUE: Execute)

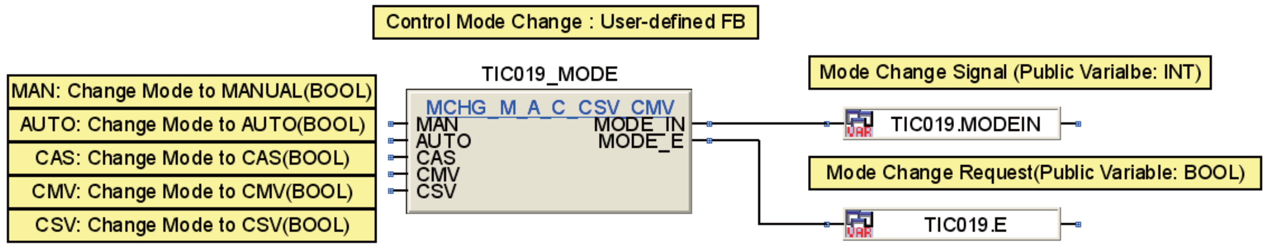
3.2.4 Control mode change (MAN_AUTO_CAS_CMV_CSV)

(1) Program example

Function	• Execute mode change (MAN, AUTO, CAS, CMV, CSV) of 2-degree-of-freedom advanced PID control (M_2PIDH_T_). • For details of user-defined FB (MCHG_M_A_C_CSV_CMV) used in this example, refer to (2) in this section.

1) Control mode change (MAN_AUTO_CAS_CMV_CSV)



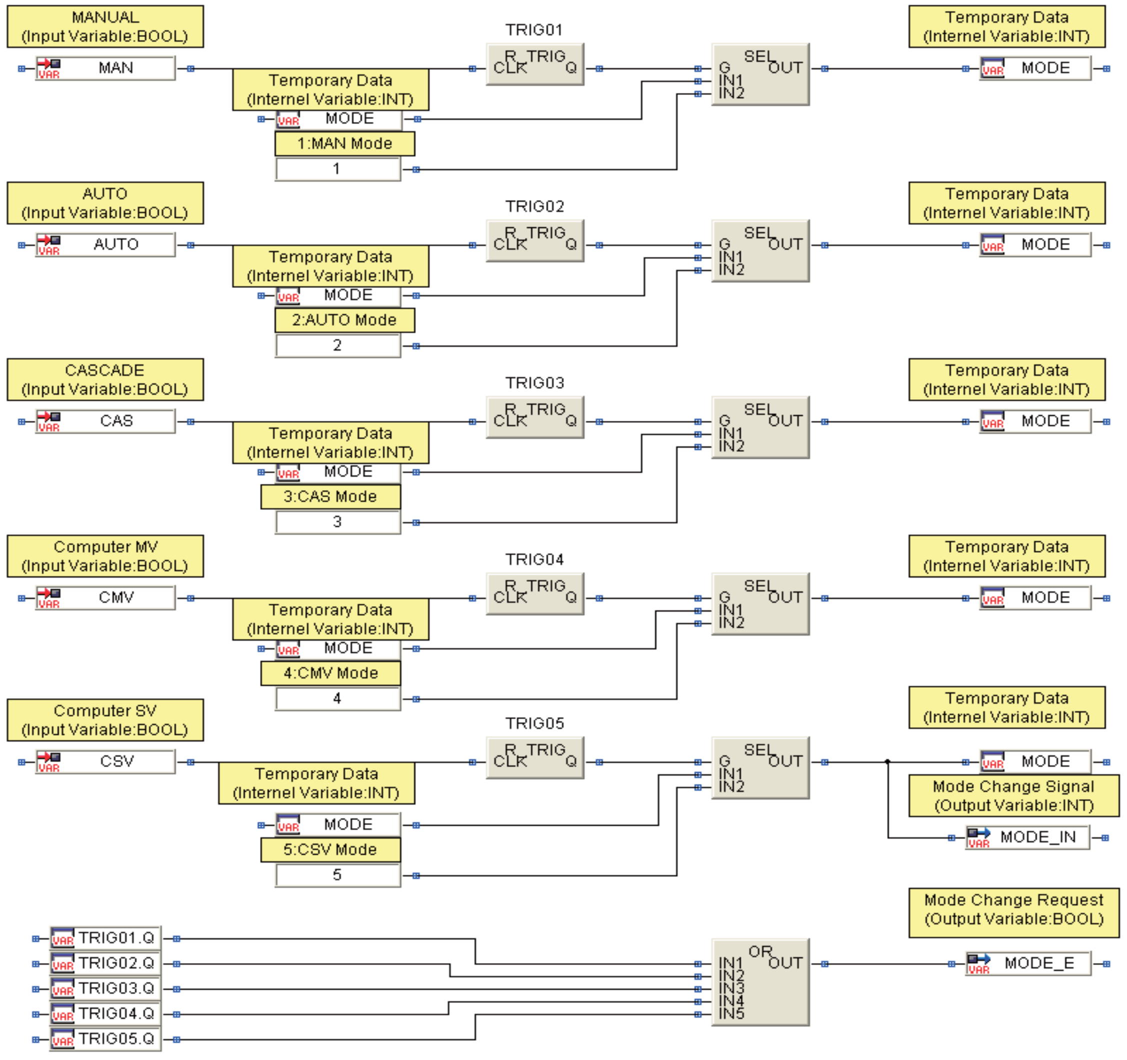


---

(2) Example of user-defined FB

1) Control mode change (MAN_AUTO_CAS_CMV_CSV) User-defined FB (MCHG_M_A_C_CSV_CMV)

Point	• Switching signal for MAN from FALSE to TRUE changes the control mode to MAN mode. • Switching signal for AUTO from FALSE to TRUE changes the control mode to AUTO mode. • Switching signal for CAS from FALSE to TRUE changes the control mode to CAS mode. • Switching signal for CMV from FALSE to TRUE changes the control mode to CMV mode. • Switching signal for CSV from FALSE to TRUE changes the control mode to CSV mode.



Pin	Variable name	Variable type	Data type	Contents
Input	MAN	Input variable	BOOL	TRUE: MAN mode
Input	AUTO	Input variable	BOOL	TRUE: AUTO mode
Input	CAS	Input variable	BOOL	TRUE: CAS mode
Input	CMV	Input variable	BOOL	TRUE: CMV mode
Input	CSV	Input variable	BOOL	TRUE: CSV mode
Output	MODE_IN	Output variable	INT	Control mode (1:MAN, 2:AUT, 3:CAS, 4: CMV, 5: CSV)
Output	MODE_E	Output variable	BOOL	Change request (TRUE: Execute)

3.2.5 Disabling control mode change

(1) Program example

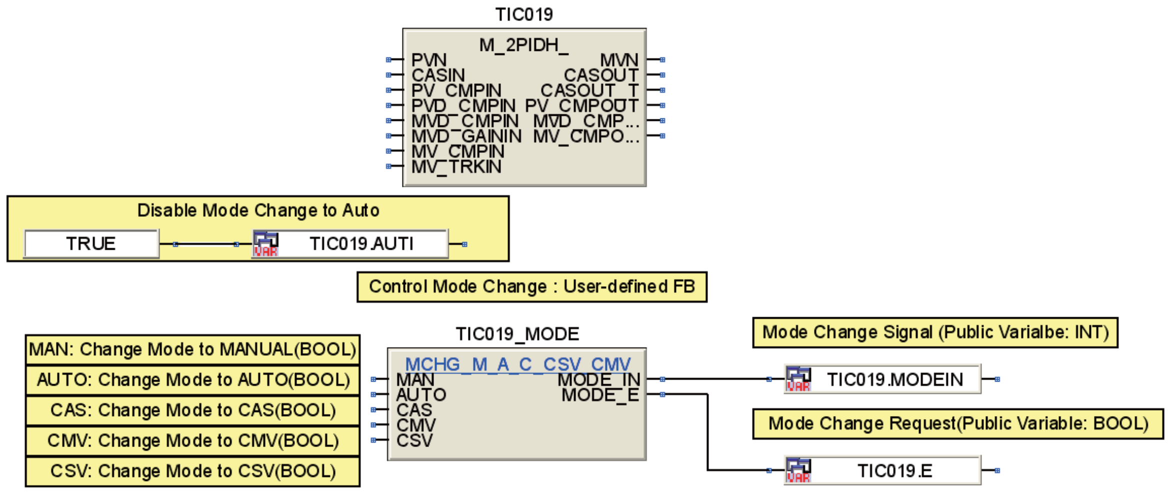
Function	• Disable control mode change (MAN, AUTO, CAS, CMV, CSV) of 2-degree-of-freedom advanced PID control (M_2PIDH_T_) to specified mode.

1) Disabling control mode change parameter

Set the disable mode change item of corresponding loop tag memory to TRUE to disable control mode change.

MDIH disable mode change item	Contents
MANI	Set to disable the change to MANUAL mode.
AUTI	Set to disable the change to AUTO mode
CASI	Set to disable the change to CASCADE mode
CMVI	Set to disable the change to COMPUTER MV mode (The mode of manual operation with upper computer)
CSVI	Set to disable the change to COMPUTER SV mode (The mode for automatic operation with upper computer)

2) Program example to disable the change to AUTO mode

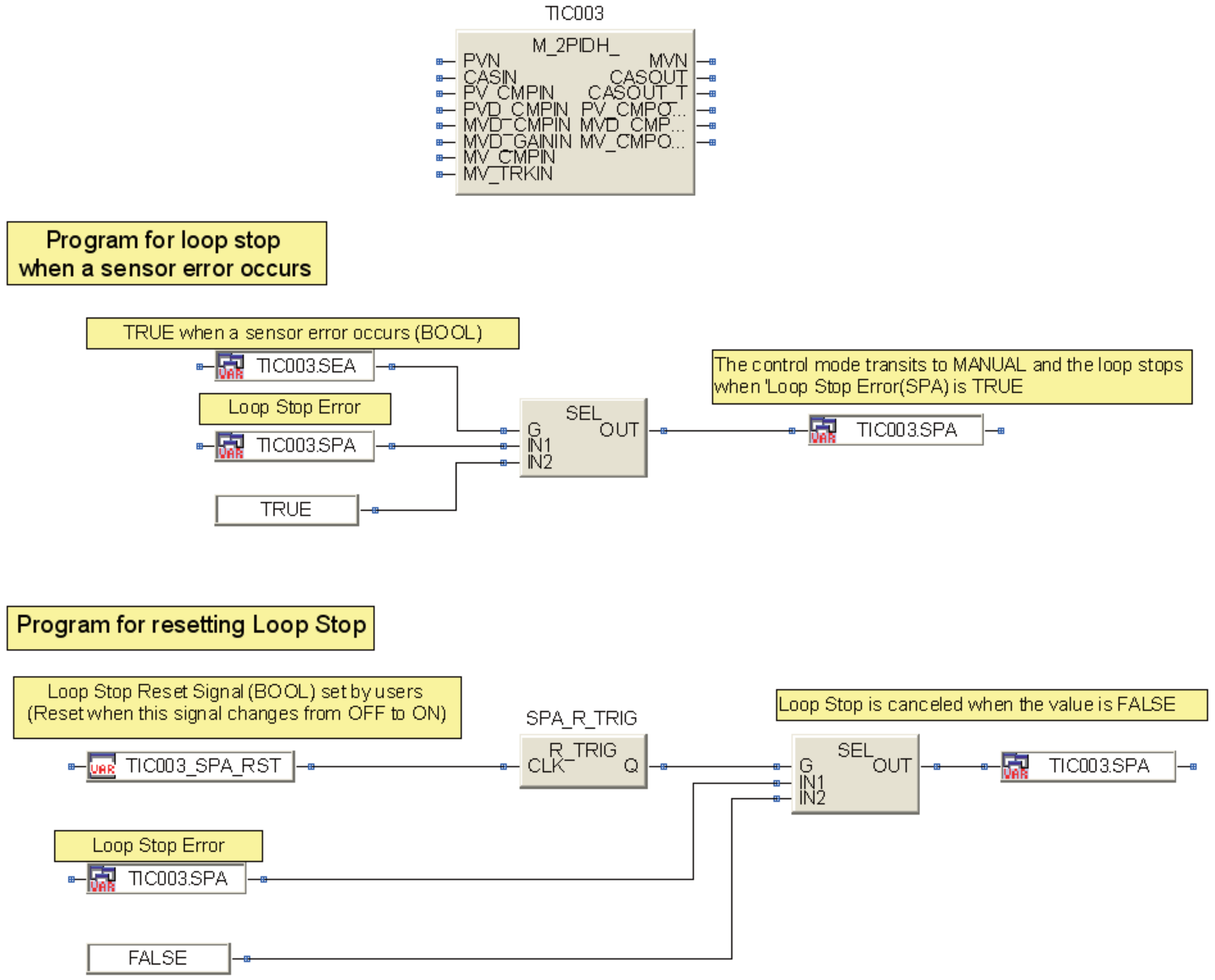


3.2.6 Sensor error loop stop

Function	• Execute loop stop by sensor error and loop stop cancel.

(1) Program example

Point	• When sensor error (tag name.SEA) is occurred by such as sensor malfunction, executing stop alarm by programming (tag name.SPA is TRUE) stops the loop processing automatically. • Executing loop stop alarm (tag name.SPA is TRUE) switches the control mode to MANUAL mode automatically. The alarm of sensor error (SEA) is also cancelled automatically. • Cancelling loop stop alarm (SPA) is as follows. (Clearing sensor error does not cancel loop stop alarm automatically.) (a) Cancel with user program: Cancels the stop alarm by switching tag name.SPA to FALSE with such as a program example of sensor loop stop cancel processing as shown below. (b) Cancel with faceplate of PX Developer monitor tool: Cancels the stop alarm by pressing the loop stop alarm (SPA) display button (displayed by executing loop stop alarm (SPA)) on faceplate.

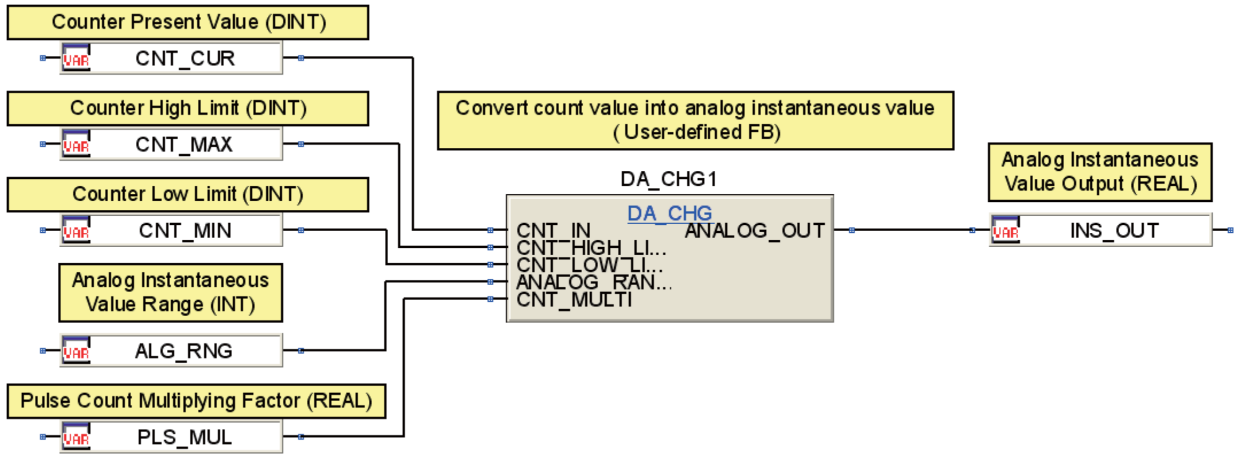


3.2.7 Count value to analog instantaneous value

Function	• Convert count current value to analog instantaneous value. • For details of user-defined FB (DA_CHG) used in this example, refer to (2) in this section.

(1) Program example

Point	• Execution interval which uses this user-defined FB should be set to 1000ms in the program execution setting. (Set the timer execution for the execution type, normal-speed or low-speed for the type of execution interval, 1000ms for interval in the program execution setting. In addition, set to 1000ms for the interval of normal-speed or low-speed in the program execution setting of project parameter.)

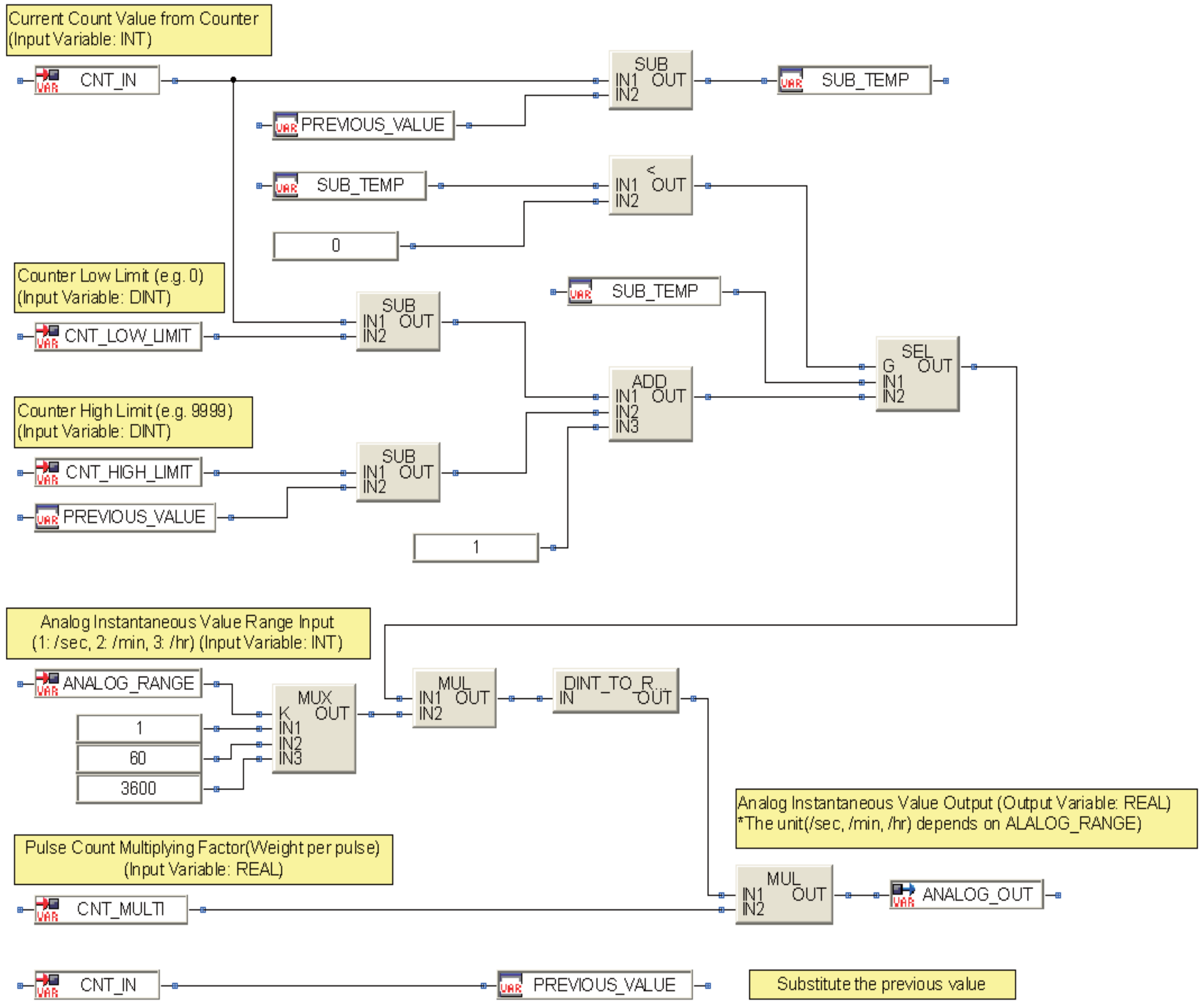


---

(2) Example of user-defined FB

1) Count value to analog instantaneous value User-defined FB (DA_CHG)

Point	• Inputting count current value outputs the analog instantaneous value. The range of instantaneous value can be selected from /sec, /min, /hr, and inputs corresponding value to the instantaneous value range input pin.



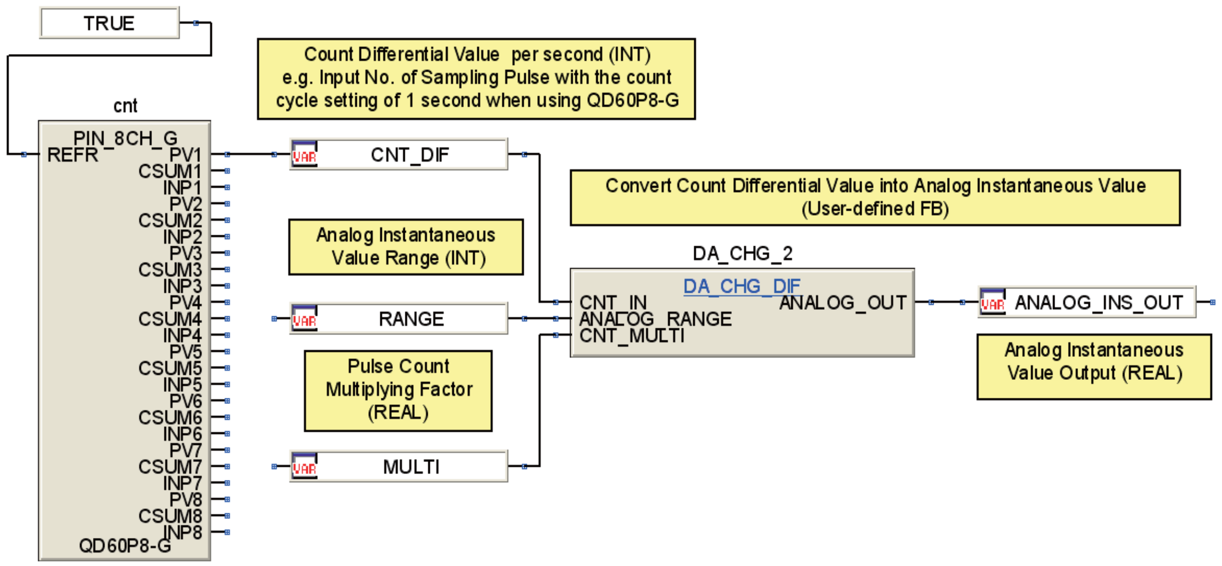
Pin	Variable name	Variable type	Data type	Contents
Input	CNT_IN	Input variable	DINT	Current value of counter
Input	CNT_HIGH_LIMIT	Input variable	DINT	Counter high limit (high limit of a revolution of counter)
Input	CNT_LOW_LIMIT	Input variable	DINT	Counter low limit (low limit after a revolution of counter)
Input	ANALOG_RANGE	Input variable	INT	Analog instantaneous value range (1: /sec, 2: /min, 3: /hr)
Input	CNT_MULTI	Input variable	REAL	Pulse count multiplying factor (weight per pulse count)
Output	ANALOG_OUT	Output variable	REAL	Output the analog instantaneous value

3.2.8 Count difference value (QD60P8-G) to analog instantaneous value

Function	• Output analog instantaneous value from count difference value per second (the number of sampling pulse when count cycle setting value is 1 second) which is input from channel-isolated pulse input module (QD60P8-G) • For details of user-defined FB (DA_CHG_DIF) used in this example, refer to (2) in this section.

(1) Program example

Point	• For count difference value, enter difference value per second. Example: Enter the number of sampling pulse when setting the count cycle setting value to 1 second to the count difference value with channel-isolated pulse input module (QD60P8-G).

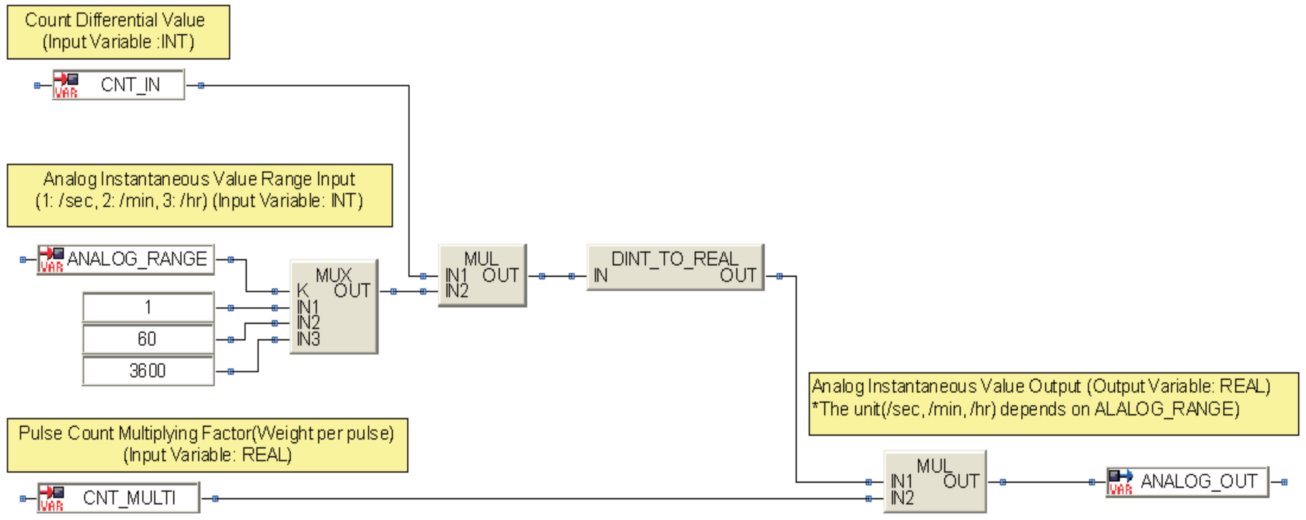


---

(2) Example of user-defined FB

1) Count difference value (Per second) → Analog instantaneous value User-defined FB (DA_CHG_DIF)

Point	• Output analog instantaneous value from count difference value per second (for example: the number of sampling pulse when count cycle setting value of channel-isolated pulse input module QD60P8-G is 1 second) from count difference value per second. The range of instantaneous value can be selected from /sec, /min, /hr, and inputs corresponding value to instantaneous value range input pin.

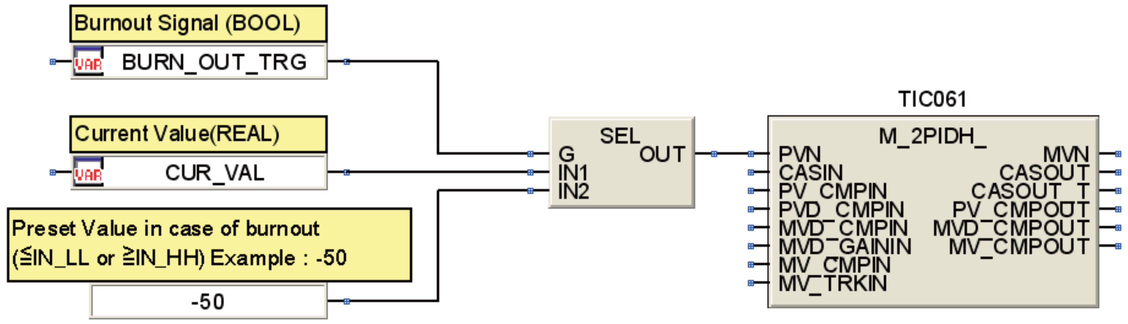


Pin	Variable name	Variable type	Data type	Contents
Input	CNT_DIF_IN	Input variable	INT	Count difference value (per second)
Input	ANALOG_RANGE	Input variable	INT	Analog instantaneous value range (1：/sec、2：/min、3：/hr)
Input	CNT_MULTI	Input variable	REAL	Pulse count multiplying factor (weight per pulse count)
Output	ANALOG_OUT	Output variable	REAL	Output the analog instantaneous value

3.2.9 Sensor burnout preset

Function	• Switch process variable to preset value and then output when a sensor burnout occurs

(1) Program example

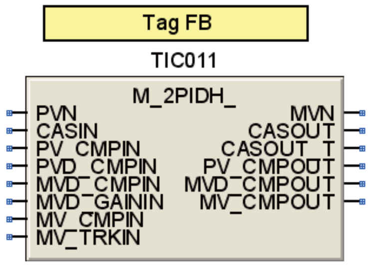


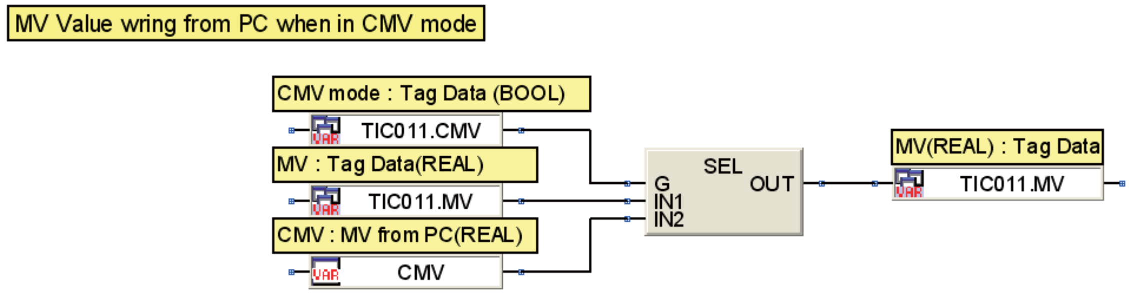
3.2.10 Writing MV, SV FB from upper computer

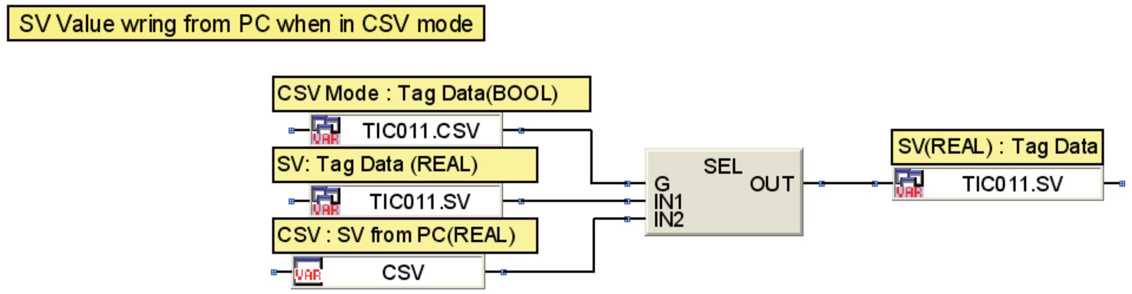
Function	• Write MV or SV to a tag FB in the control mode of CMV (COMPUTER MV) or CSV (COMPUTER SV).

(1) Program example

Point	• The setting range of MV is -10 to 110. • The setting range of SV is RL to RH.



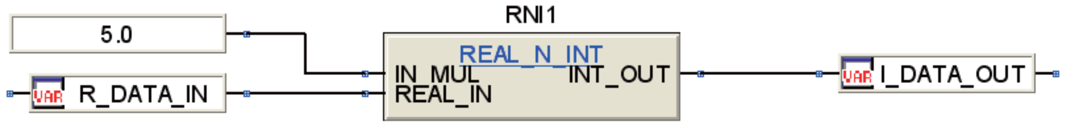




3.2.11 REAL type ×N times to INT type conversion

Function	• Multiplie real type data by N (10, 100, ..) and then convert it to integer type. • For details of user-defined FB (REAL_N_INT) used in this example, refer to (2) in this section.

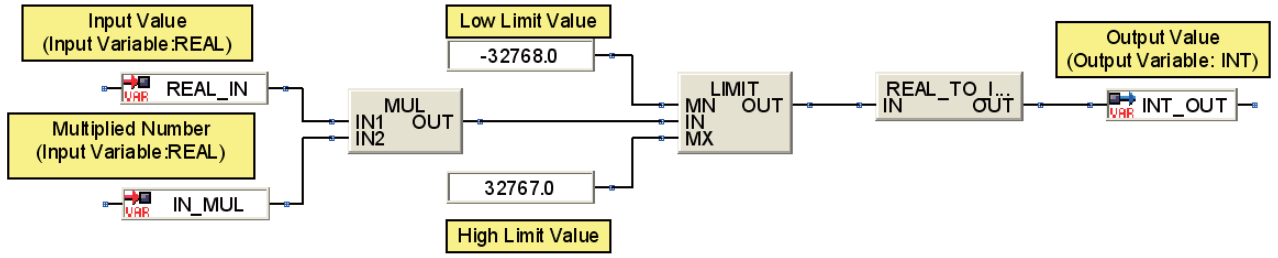
(1) Program example



(2) Example of user-defined FB

1) REAL type × N times to INT type conversion User-defined FB (REAL_N_INT)

Point	• Output the result of multiplying REAL data input to input variable REAL_IN by REAL data input to IN_MUL to output variable INT_OUT with converting to integer data.



Pin	Variable name	Variable type	Data type	Contents
Input	REAL_IN	Input variable	REAL	Input value
Input	IN_MUL	Input variable	REAL	N times
Output	INT_OUT	Output variable	INT	Output value

3.2.12 WORD type to REAL type conversion

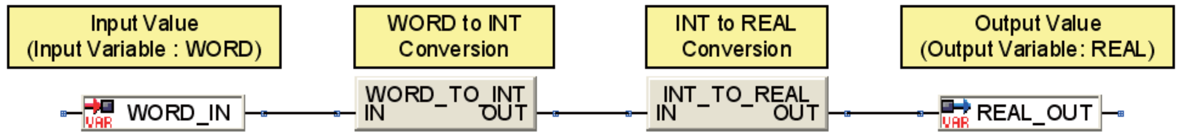
Function	• Convert WORD type data to REAL type. (Used when inputting WORD output data of CC-Link module FB to loop tag FB) • For details of user-defined FB (WORD_REAL) used in this example, refer to (2) in this section.

(1) Program example

(2) Example of user-defined FB

① WORD type to REAL type conversion User-defined FB (WORD_REAL)

Point	• Convert WORD type data input to input variable WORD_IN to REAL type data, and output to output variable REAL_OUT.



Pin	Variable name	Variable type	Data type	Contents
Input	WORD_IN	Input variable	WORD	Input value
Output	REAL_OUT	Output variable	REAL	Output value

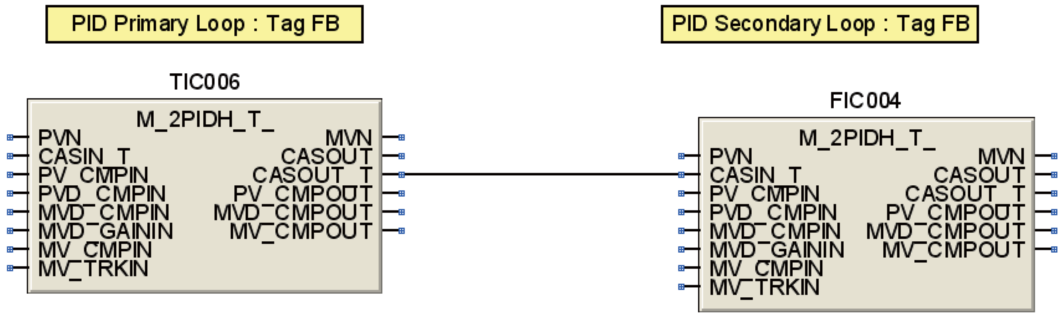
3.3 Program Examples (loop control related)

3.3.1 Cascade control

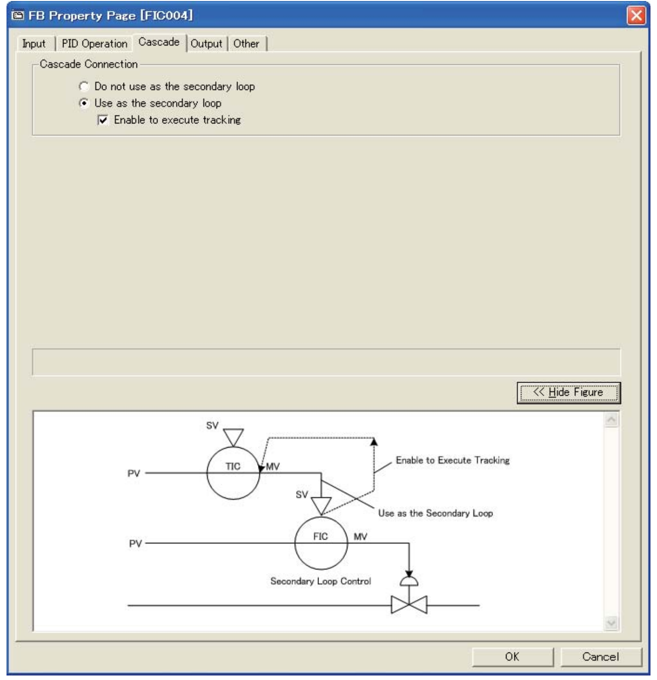
Function	• Execute the cascade control with loop tag FB

(1) Program example

Point	• Connecting cascade output (CASOUT_T) of primary loop and cascade input of secondary loop (CASIN_T) enables the cascade control which has the tracking function. • When the control mode of secondary loop is AUTO, MANUAL, tracking with transferring SV of secondary loop to MV of primary loop switches to bumpless.



Setting example of secondary side loop tracking in cascade connection (With FB property page of 2-degree-of-freedom advanced PID control)


FB Property Page [FIC004]

Cascade Connection

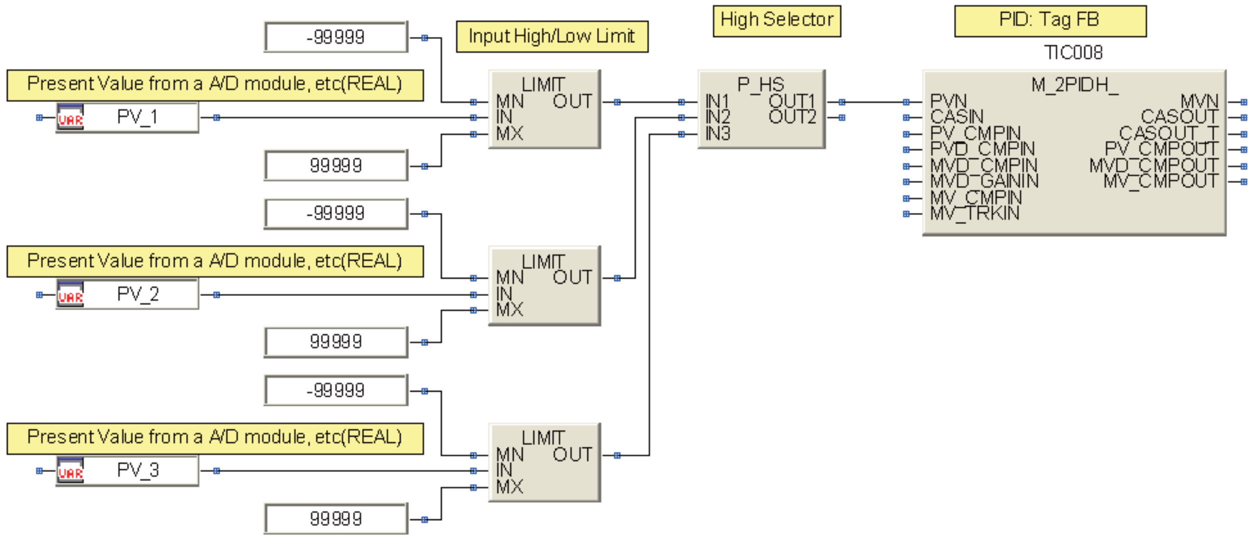
○ Do not use as the secondary loop
● Use as the secondary loop
☑ Enable to execute tracking

3.3.2 Selection control (input high selector)

Function	• Select the maximum value of multiple input values and output them as PV of the loop control.

(1) Program example

Point	• Find the maximum input value with high selector (P_HS) of process function.

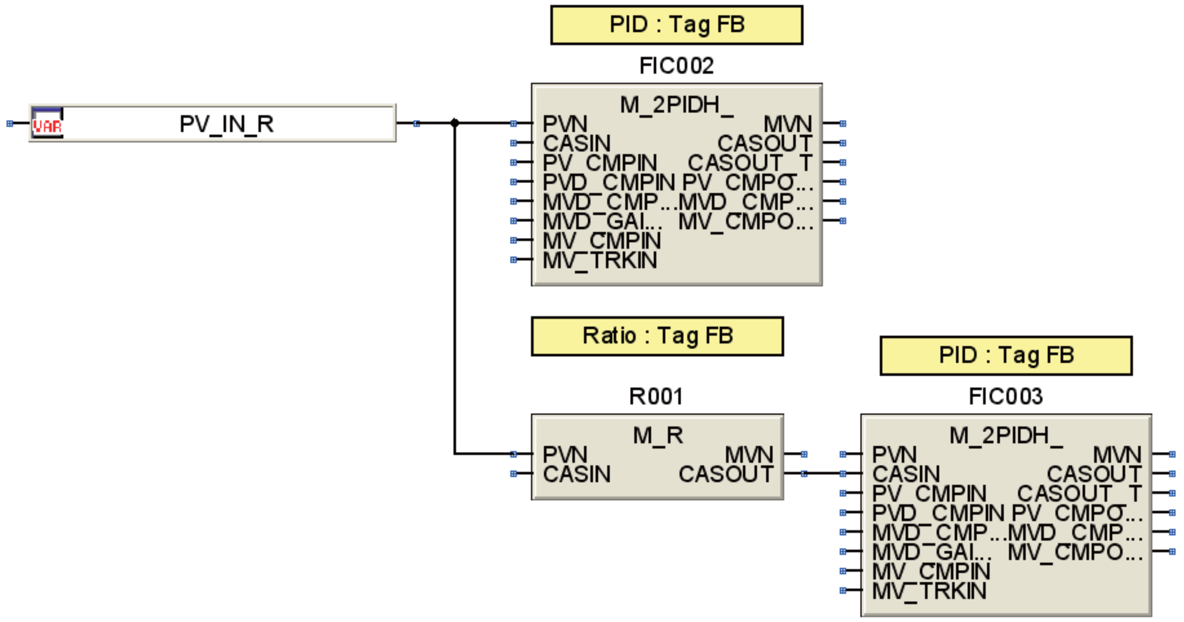


3.3.3 Ratio control

Function	• Execute the ratio control with loop tag FB.

(1) Program example

Point	• Control PV of tag (FIC002) to be the ratio of tag (FIC003) PV is preset value by inputting the value which multiplies PV of tag (FIC002) by the ratio of Pulse Factor Controller (R001) as SV of tag (FIC003)

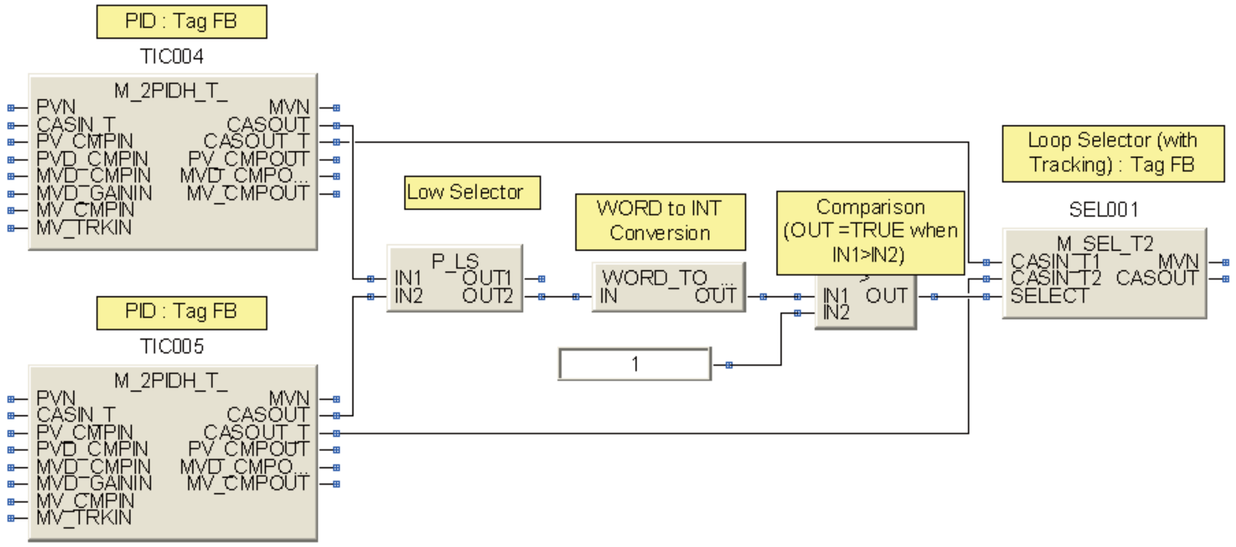


3.3.4 Output override (low selector)

Function	• Compare MVs of 2 PID loops, select the smaller MV with a Selector, and then execute output override.

(1) Program example

Point	• Compare MVs of 2 PID loops, and output the smaller MV with Selector (M_SEL_T2). • When the control mode of Selector (M_SEL_T2) is MAN, execute MV tracking to the PID loop which is connected to CASIN_T1, CASIN_T2 pins.

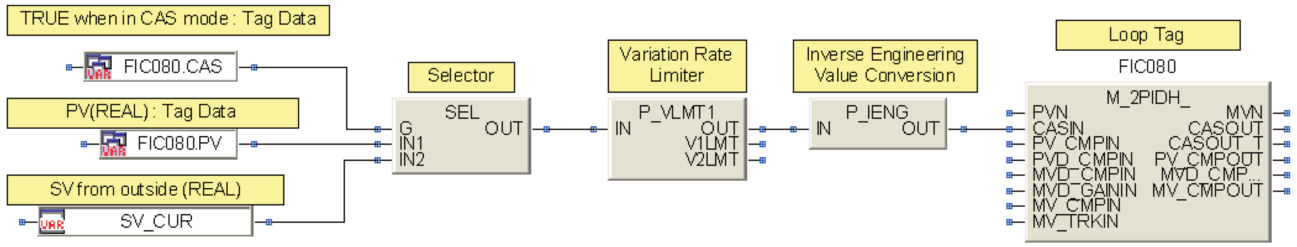


3.3.5 Process variable tracking (when upper is not loop tag)

Function	• Process variable tracking when upper is not loop tag. • When the control mode is other than CAS, prevent sudden SV change in switching to CAS by PV tracking.

(1) Program example

Point	• Execute PV tracking with SEL when the control mode of loop tag is other than CAS. • Execute rate of change limiter with P_VLMT1. • Execute inverse engineering unit conversion into % value with P_IENG and then connect to cascade input of loop tag.

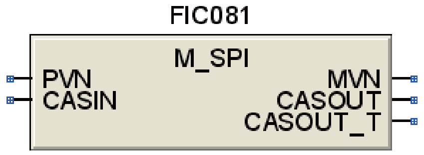


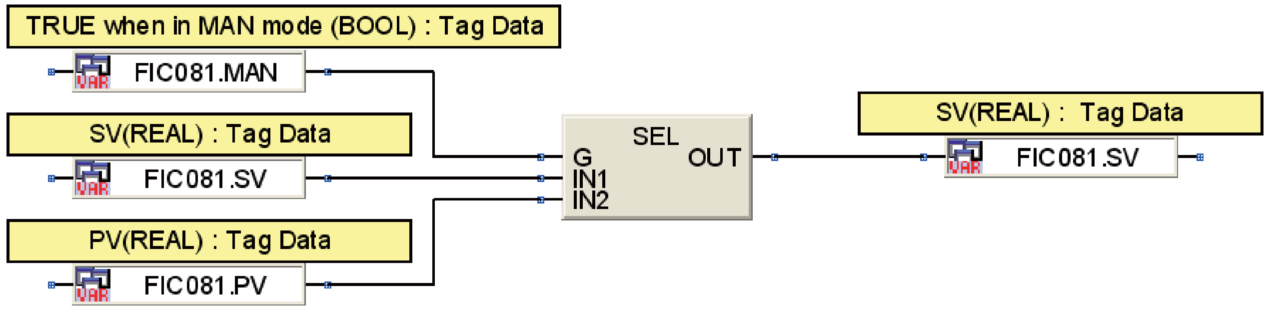
3.3.6 Process variable tracking (bumpless function in mode change)

Function	• When the control mode is MAN, tracking with transferring PV to SV fulfils bumpless when changing to AUTO.

(1) Program example

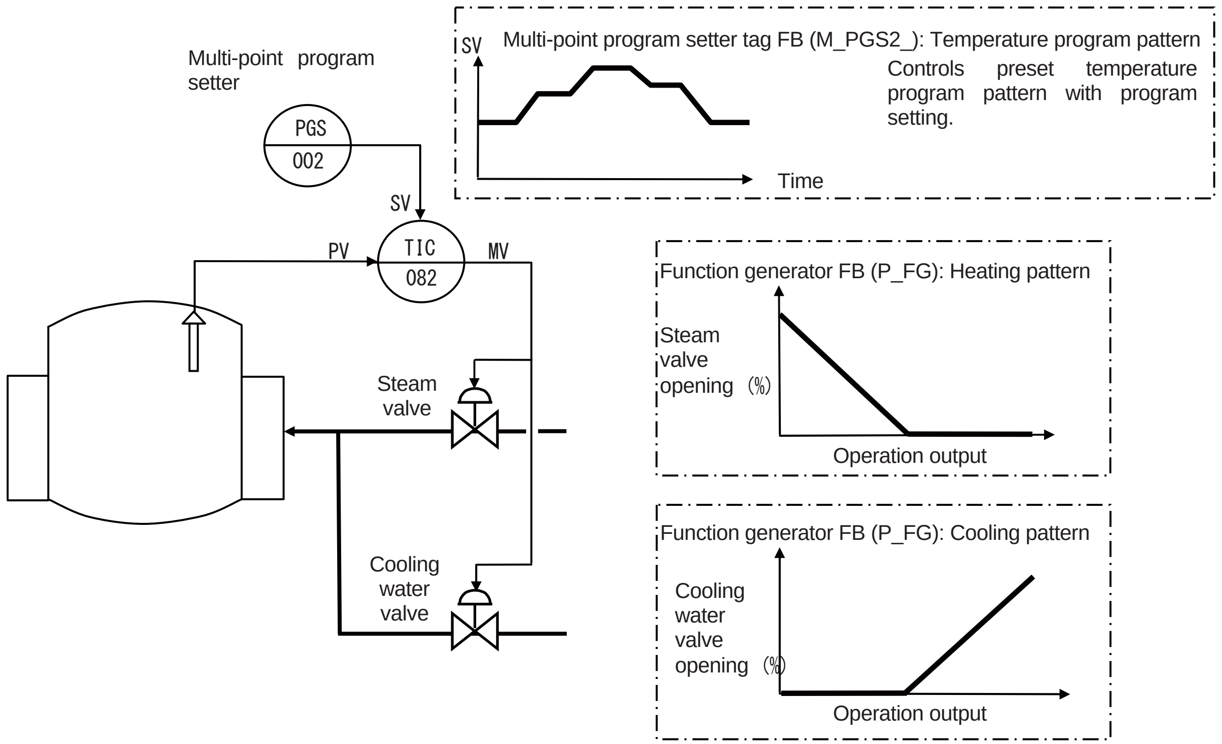
Point	• As this function is already loaded, programming the following program is not needed for 2-degree-of-freedom advanced PID control (M_2PIDH_, M_2PIDH_T_). • Use this program when executing process variable tracking with loop tag FB other than 2-degree-of-freedom advanced PID control (M_2PIDH_, M_2PIDH_T_).





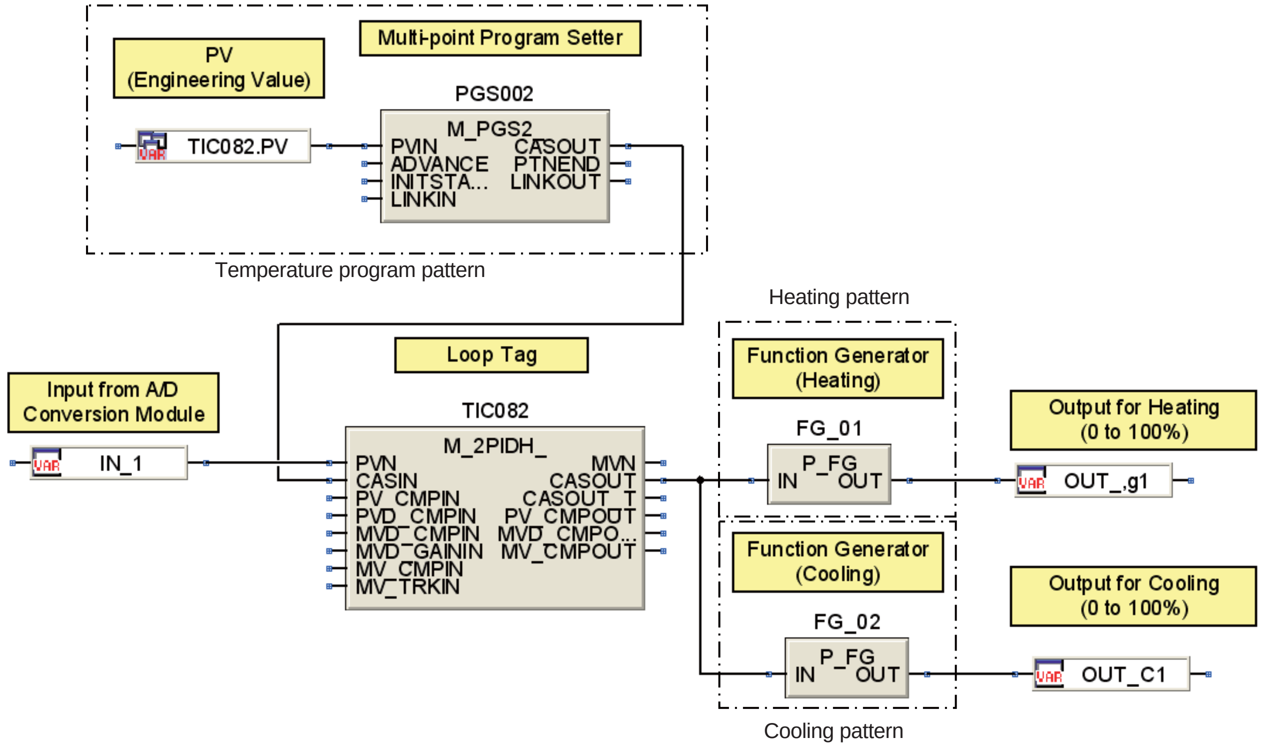
3.3.7 Heating-cooling program control

Function	• Execute the heating-cooling control with one controller. • Execute the split and program controls for heating-cooling controls.



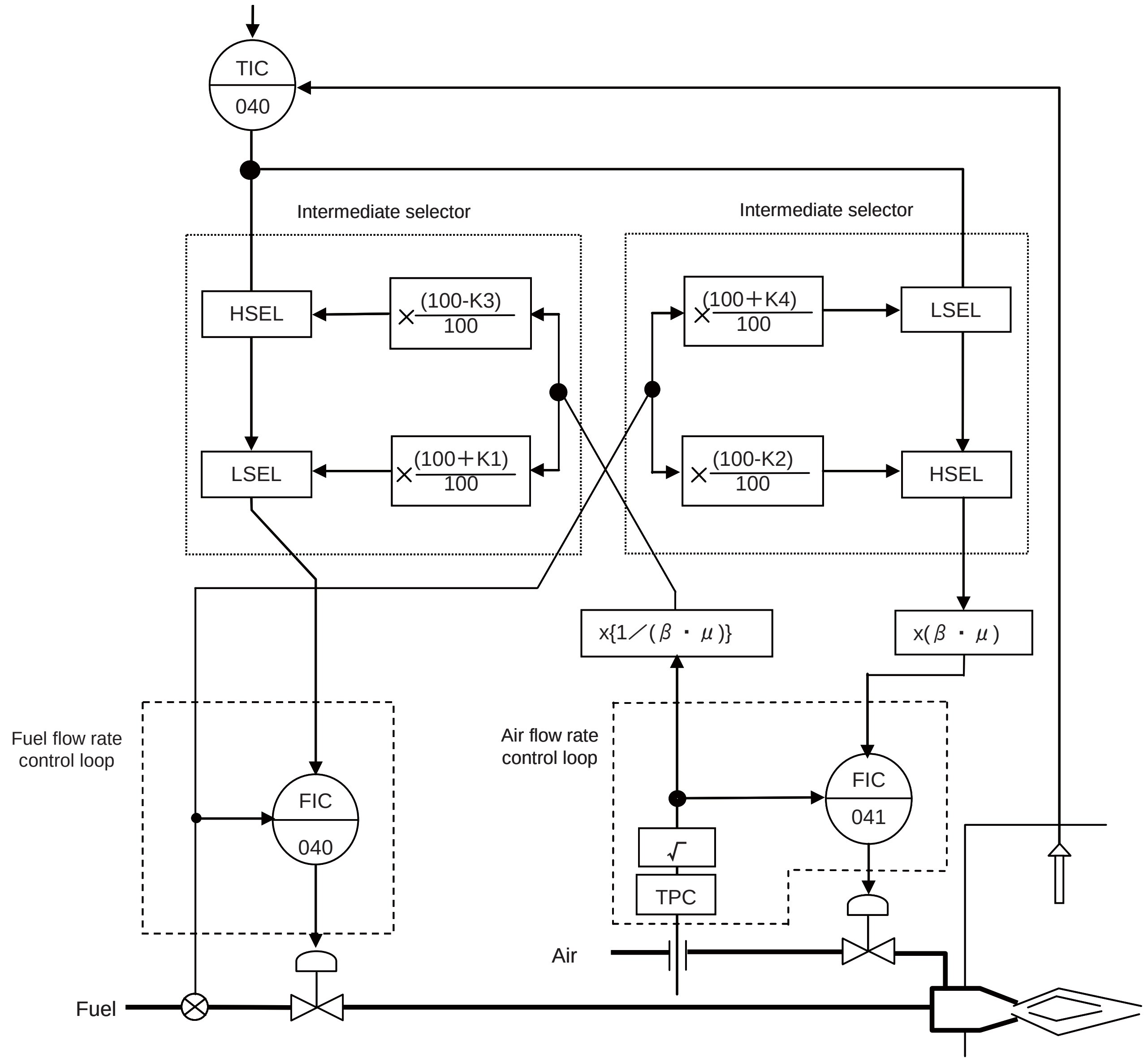
(1) Program example

Point	• Execute the split control by multiple connecting of a Function generator FB (P_FG) to loop tag output. Also, execute the program control with connecting cascade to a program setter.



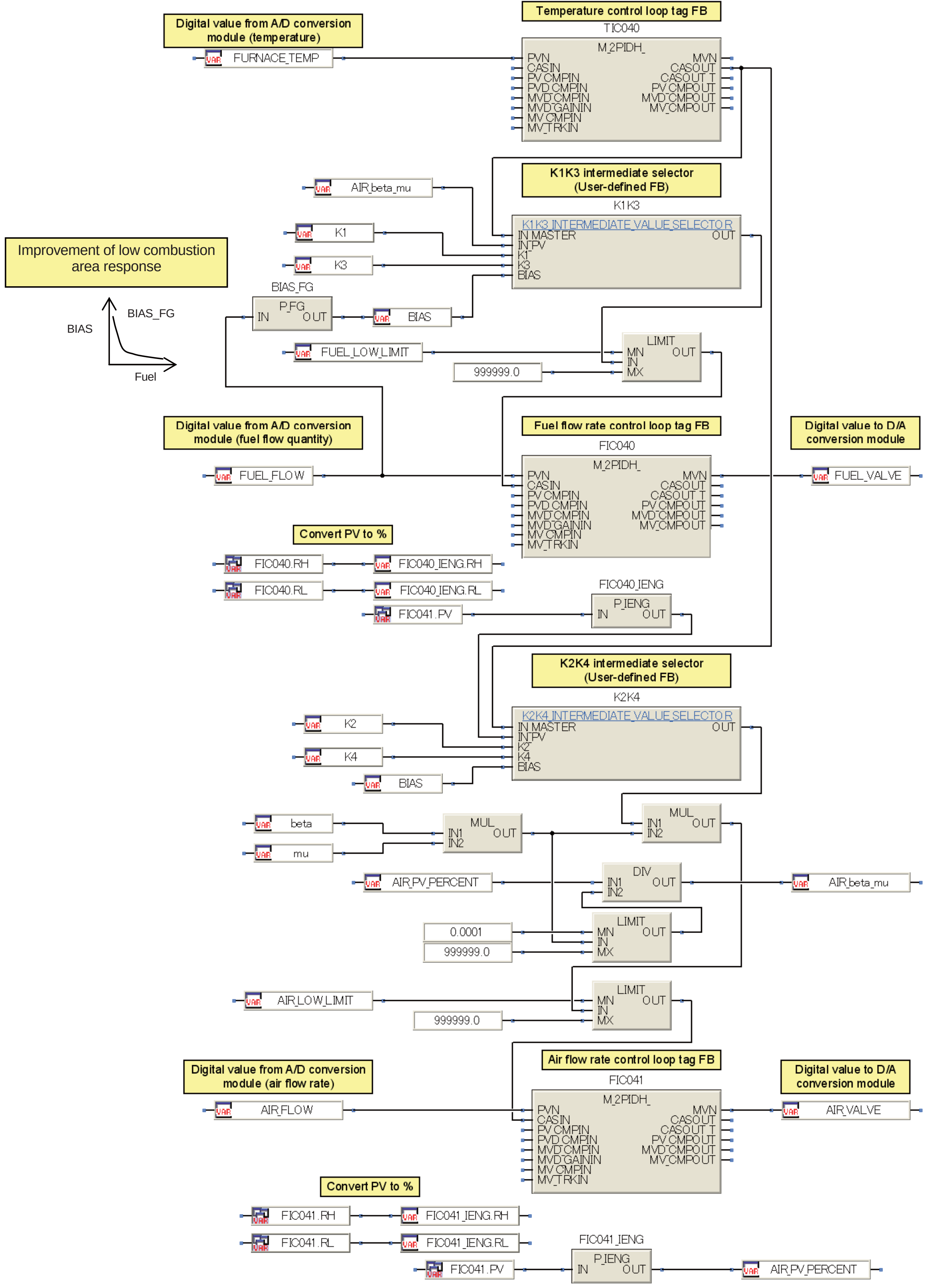
3.3.8 Cross limit control

Function	• A control which improves combustion efficiency with executing appropriate air fuel ratio control at such as a combustion furnace and both fuel and air control loops give high/low limit to its own loop in accordance with counterpart MV.



Remark
HSEL: HIGH selector
LSEL: LOW selector
TPC: Temperature/pressure correction
μ: Air excess ratio =actual air/ theoretical air
β: Conversion coefficient = (fuel flow measuring range maximum value × fuel theoretical air) / air flow measuring range maximum value
K1: Smoke limit in rising load (%)
K2: Smoke limit in reducing load (%)
K3: Air excess limit in reducing load (%)
K4: Air excess limit in rising load (%)

(1) Program example



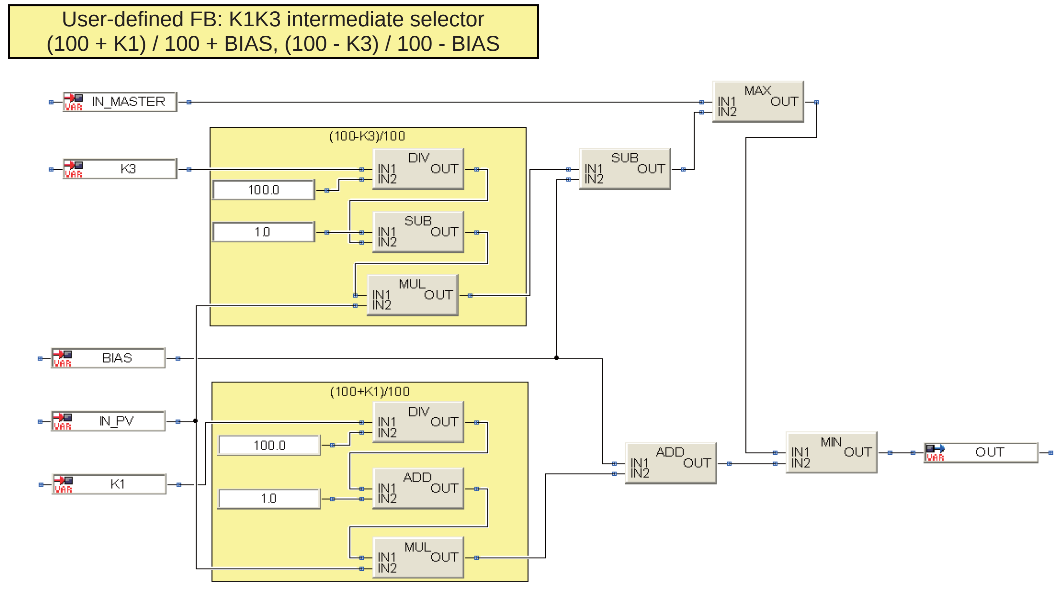
---

(2) Example of user-defined FB

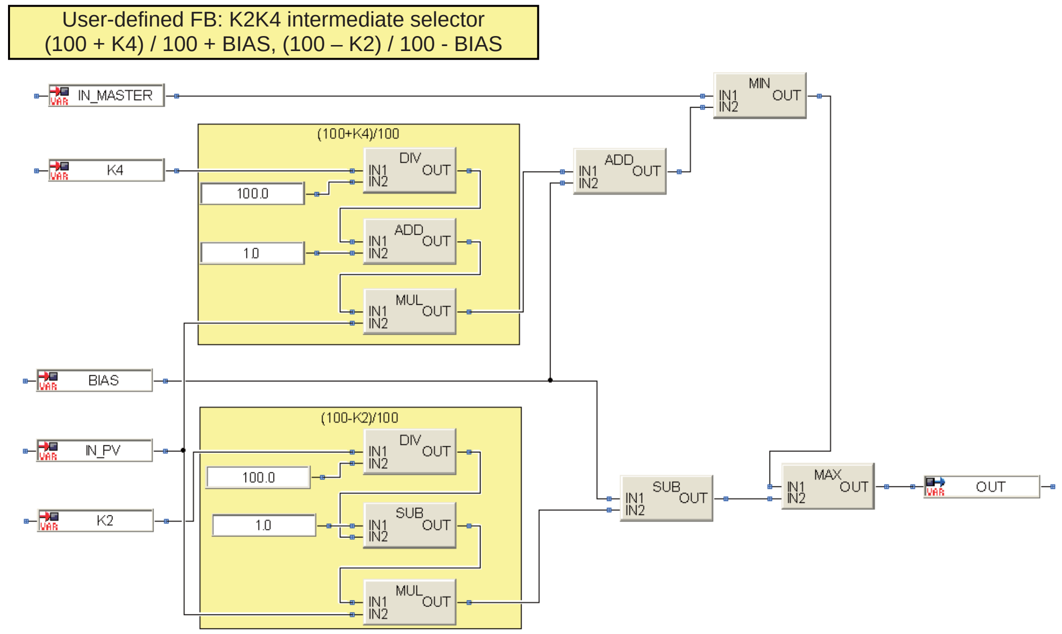
1) Cross limit user-defined FB

Point	• This user-defined FB is configured with user-defined FB shown in 2) to 5) in this section.

(a) Intermediate selector User-defined FB (K1K3 intermediate selector)



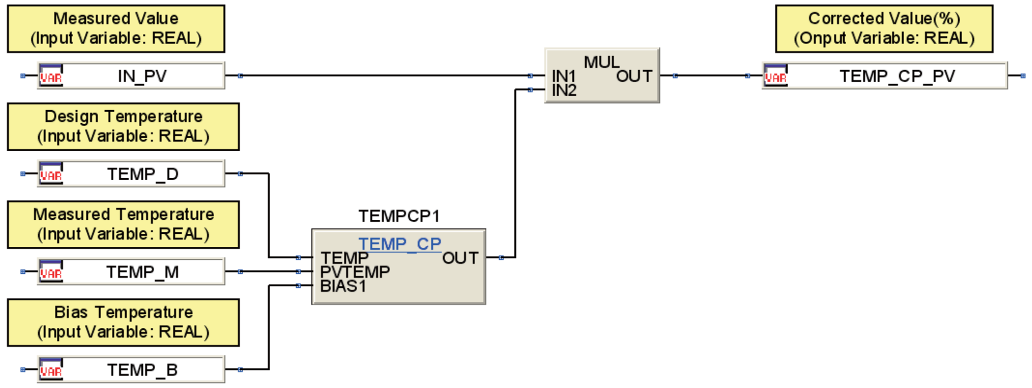
(b) Intermediate selector User-defined FB (K2K4 intermediate selector)



3.3.9 Temperature correction (with square root)

Function	• Execute the temperature correction to linear flow characteristic output from flowmeter such as area flowmeter. • For details of user-defined FB (TEMP_CP), refer to (2) in this section.

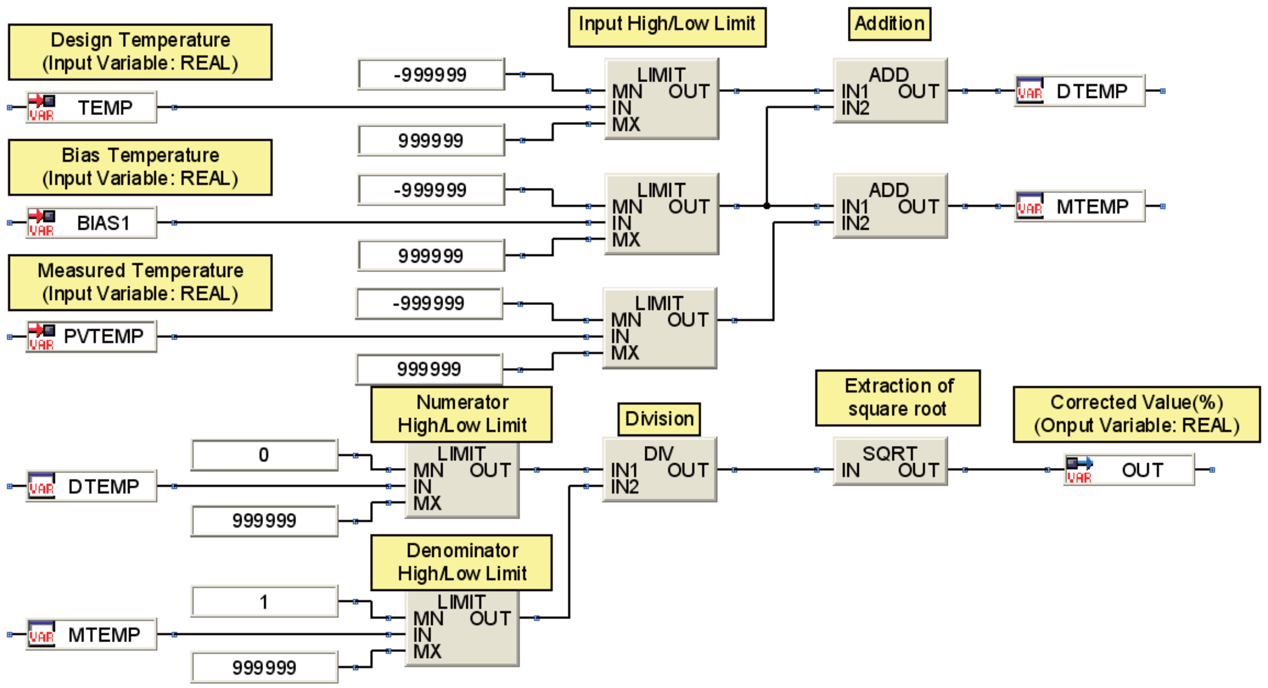
(1) Program example



(2) Example of user-defined FB

1) Temperature correction (with square root) User-defined FB (TEMP_CP)

Point	• Execute the operation of temperature correction (with square root) = $\sqrt{\ }${(design temperature + bias temperature) / (measured temperature + bias temperature)}.

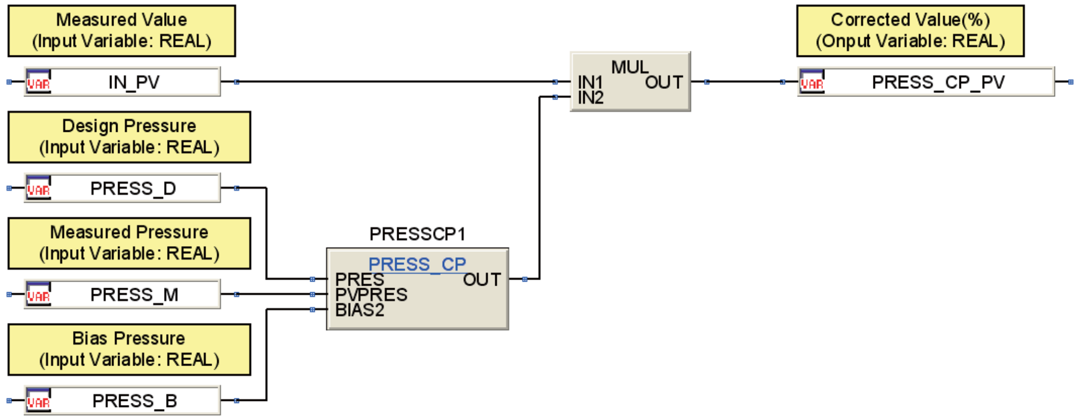


Pin	Variable name	Variable type	Data type	Contents
Input	TEMP	Input variable	REAL	Design temperature
Input	BIAS1	Input variable	REAL	Bias temperature
Input	PVTEMP	Input variable	REAL	Measured temperature
Output	OUT	Output variable	REAL	Correction value

3.3.10 Pressure correction (with square root)

Function	• Execute pressure correction to linear flow characteristic output from flowmeter such as area flowmeter. • For details of user-defined FB (PRESS_CP), refer to (2) in this section.

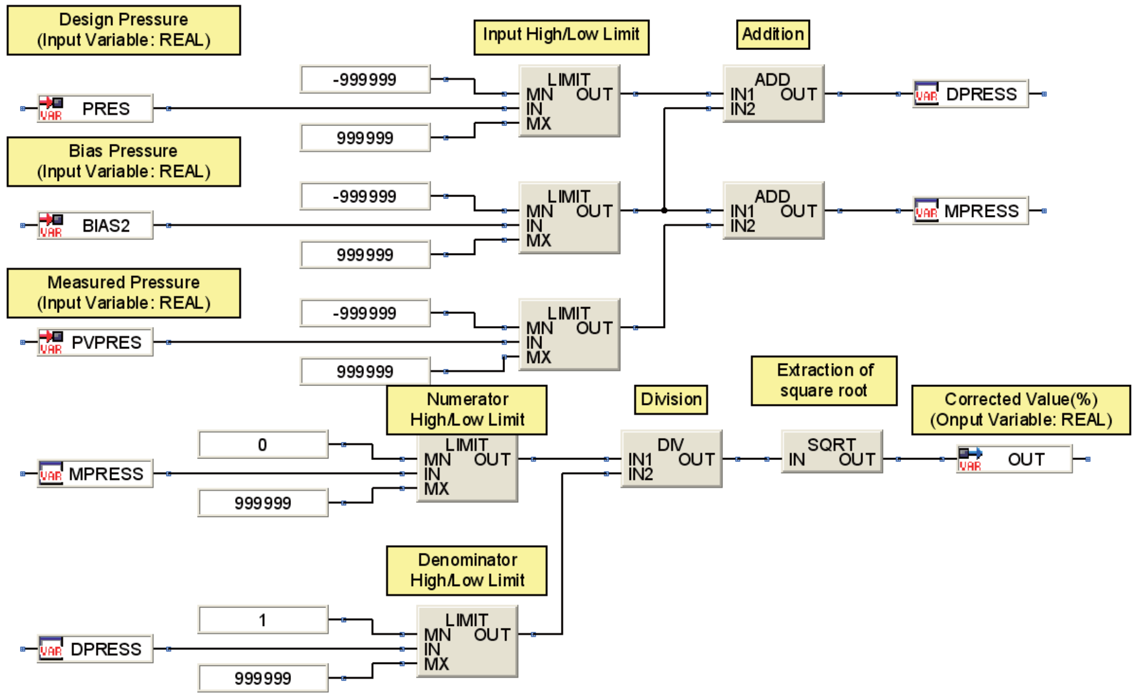
(1) Program example



(2) Example of user-defined FB

1) Pressure correction (with square root) User-defined FB (PRESS_CP)

Point	• Execute operation of pressure correction (with square root) = $\sqrt{\ }${(measured pressure + bias pressure) / (design pressure + bias pressure)}

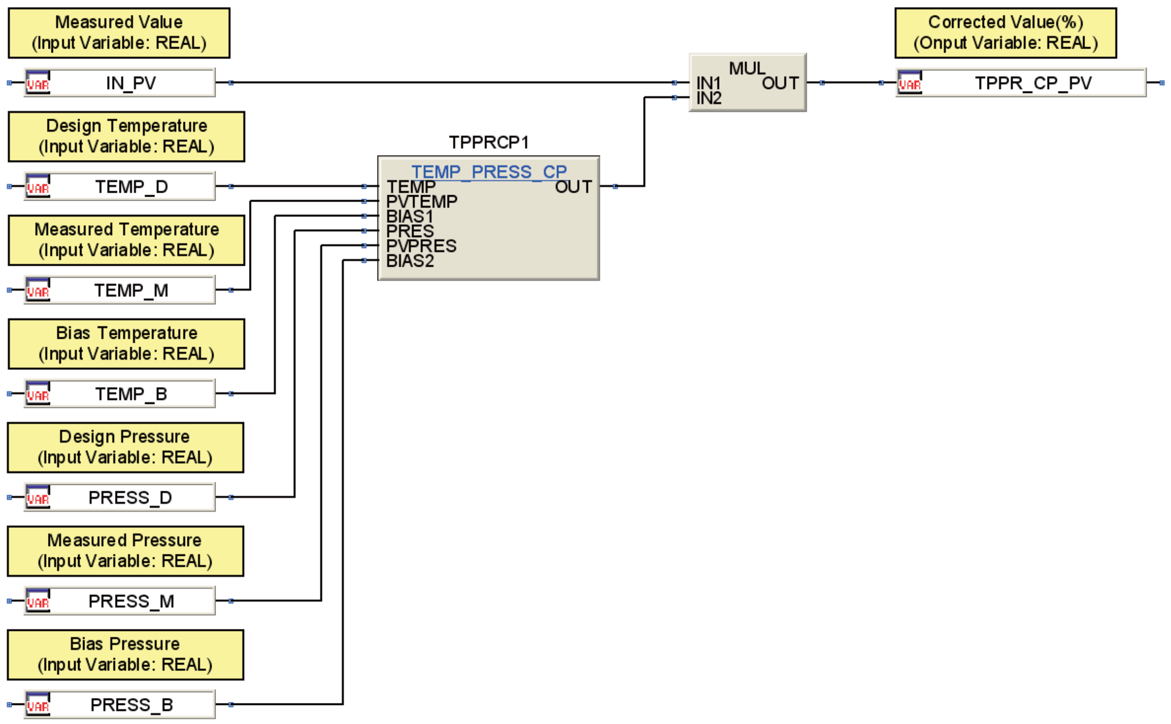


Pin	Variable name	Variable type	Data type	Contents
Input	PRES	Input variable	REAL	Design pressure
Input	BIAS2	Input variable	REAL	Bias pressure
Input	PVPRES	Input variable	REAL	Measured pressure
Output	OUT	Output variable	REAL	Correction value

3.3.11 Temperature/pressure correction (with square root)

Function	• Execute temperature/pressure correction to linear flow characteristic output from flowmeter such as area flowmeter. • For details of user-defined FB (TEMP_PRESS_CP), refer to (2) in this section.

(1) Program example

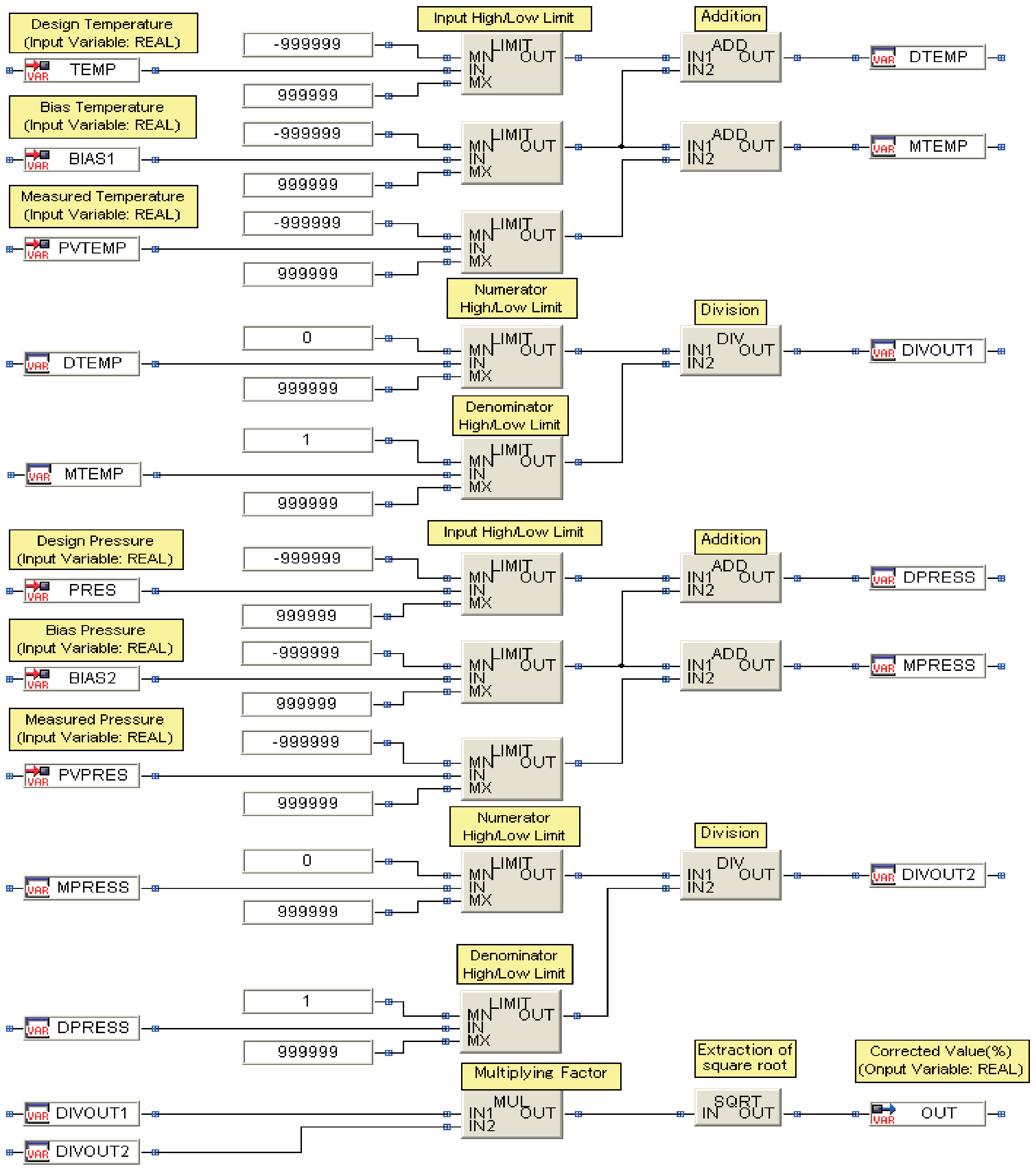


---

(2) Example of user-defined FB

1) Temperature/pressure correction (with square root) User-defined FB (TEMP_PRESS_CP)

Point	• Execute operation of $\sqrt{\ }${(measured pressure + bias pressure) / (design pressure + bias pressure)} × $\sqrt{\ }${(design temperature + bias temperature) ／ (measured temperature + bias temperature))

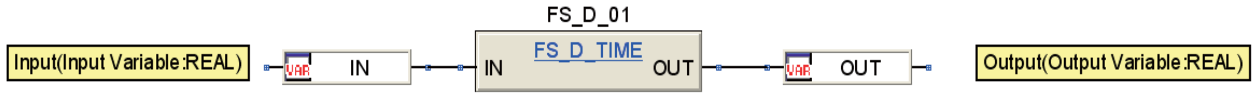


Pin	Variable name	Variable type	Data type	Contents
Input	TEMP	Input variable	REAL	Design temperature
Input	BIAS1	Input variable	REAL	Bias temperature
Input	PVTEMP	Input variable	REAL	Measured temperature
Input	PRES	Input variable	REAL	Design pressure
Input	BIAS2	Input variable	REAL	Bias pressure
Input	PVPRES	Input variable	REAL	Measured pressure
Output	OUT	Output variable	REAL	Correction value

3.3.12 First order lag/dead time

Function	• Add first order lag/dead time process to input. • For details of user-defined FB (FS_D_TIME), refer to (2) in this section.

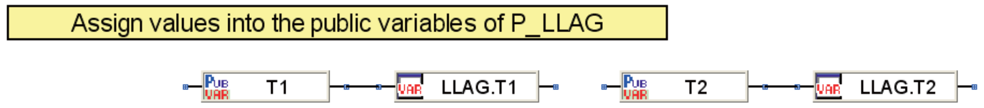
(1) Program example

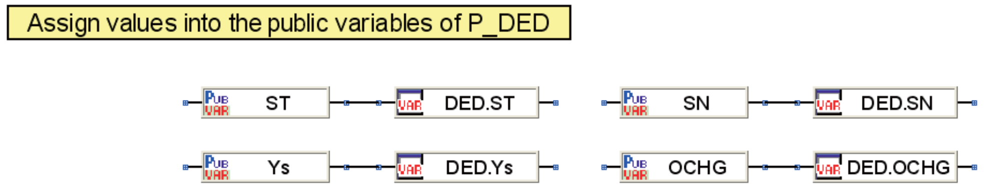


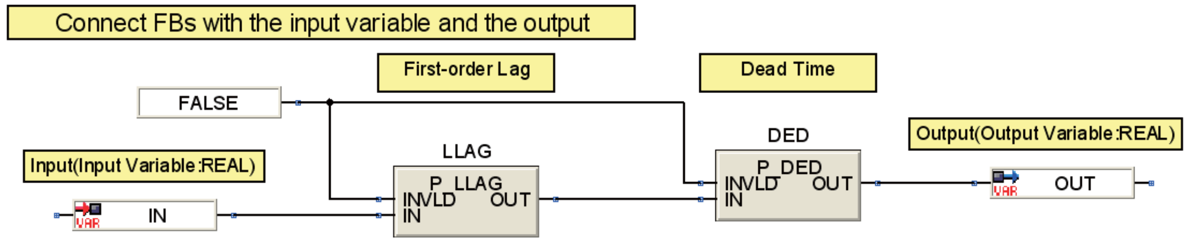
(2) Example of user-defined FB

1) First order lag/dead time User-defined FB (FS_D_TIME)

Point	• Execute processing of first order lag, dead time to input data for input variable IN and output from output variable OUT.





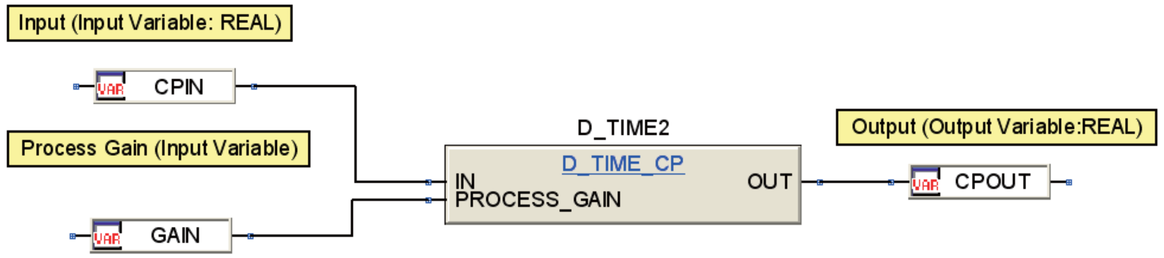


Pin	Variable name	Variable type	Data type	Contents
Input	IN	Input variable	REAL	PV input
Output	OUT	Output variable	REAL	PV output

3.3.13 Dead time compensation

Function	• Execute dead time compensation to input and then output. • For details of user-defined FB (D_TIME_CP), refer to (2) in this section.

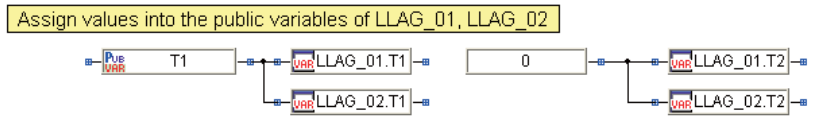
(1) Program example

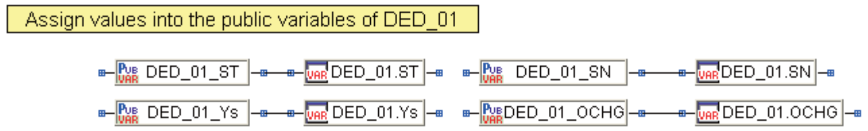


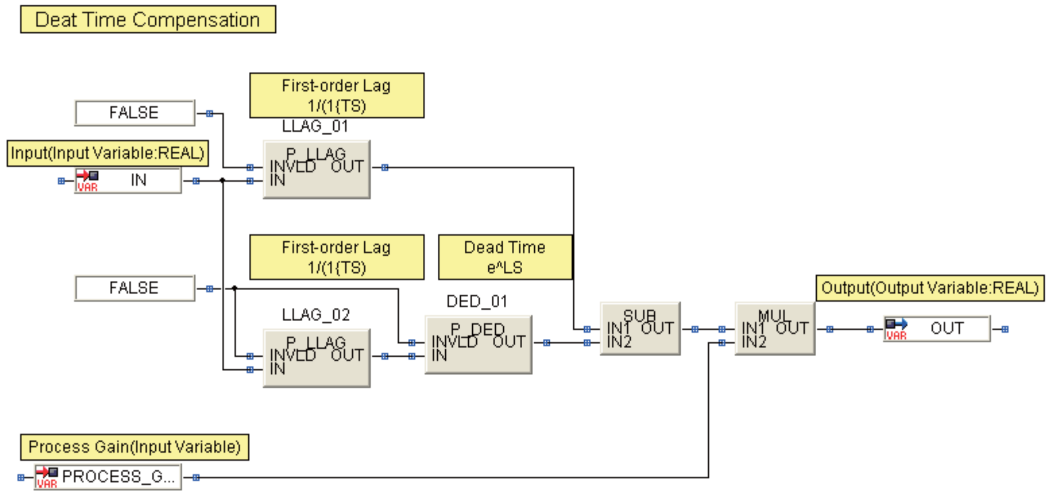
(2) Example of user-defined FB

1) Dead time compensation User-defined FB (D_TIME_CP)

Point	• Execute the conversion of K (1 - e-LS)/(1 + TS) to input data for input variable IN and output to output variable OUT.





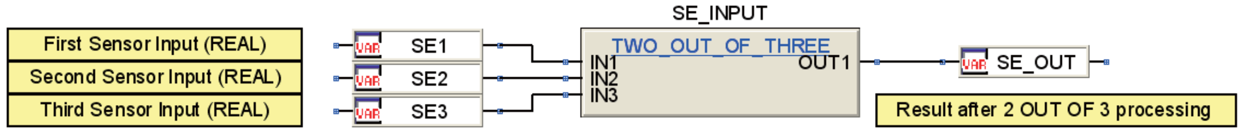


Pin	Variable name	Variable type	Data type	Contents
Input	IN	Input variable	REAL	PV input
Input	PROCESS_GAIN	Input variable	REAL	Process gain
Output	OUT	Output variable	REAL	PV output

3.3.14 2 OUT OF 3

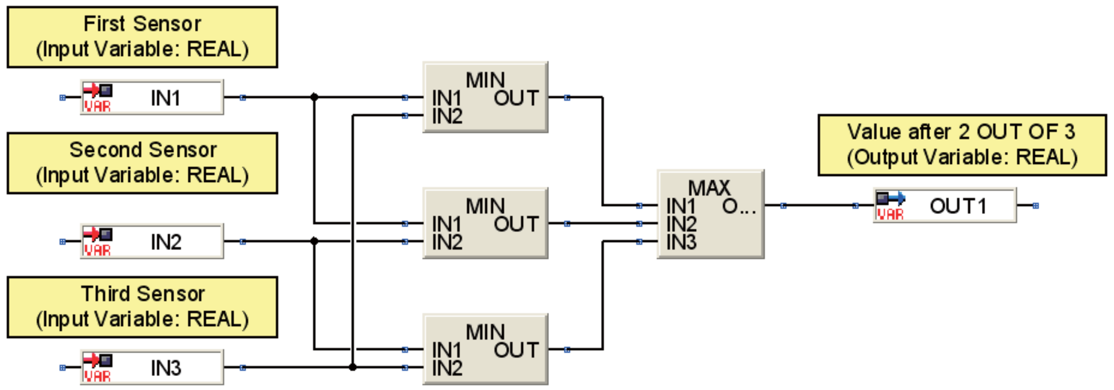
Function	• To avoid the affection by sensor error, in case of one out of three sensors failure, imports normal value with the other two sensors. • For details of user-defined FB (2_OUT_OF_3), refer to (2) in this section.

(1) Program example



(2) Example of user-defined FB

1) 2 OUT OF 3 User-defined FB (2 _OUT_OF_ 3)



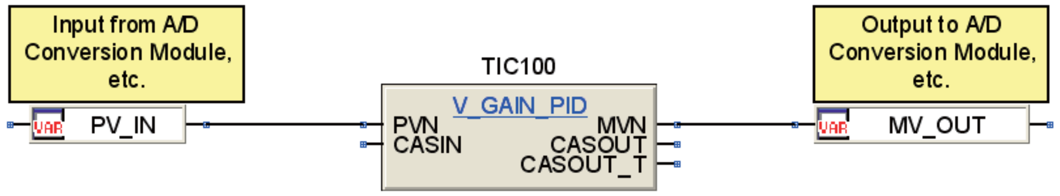
Pin	Variable name	Variable type	Data type	Contents
Input	IN1	Input variable	REAL	Sensor input 1
Input	IN2	Input variable	REAL	Sensor input 2
Input	IN3	Input variable	REAL	Sensor input 3
Output	OUT	Output variable	REAL	Output

3.3.15 Deviation variable gain PID

Function	• Deviation variable gain PID which has deviation input and function generator (P_FG). • For details of user-defined tag FB (V_GAIN_PID), refer to (2) in this section.

(1) Program example

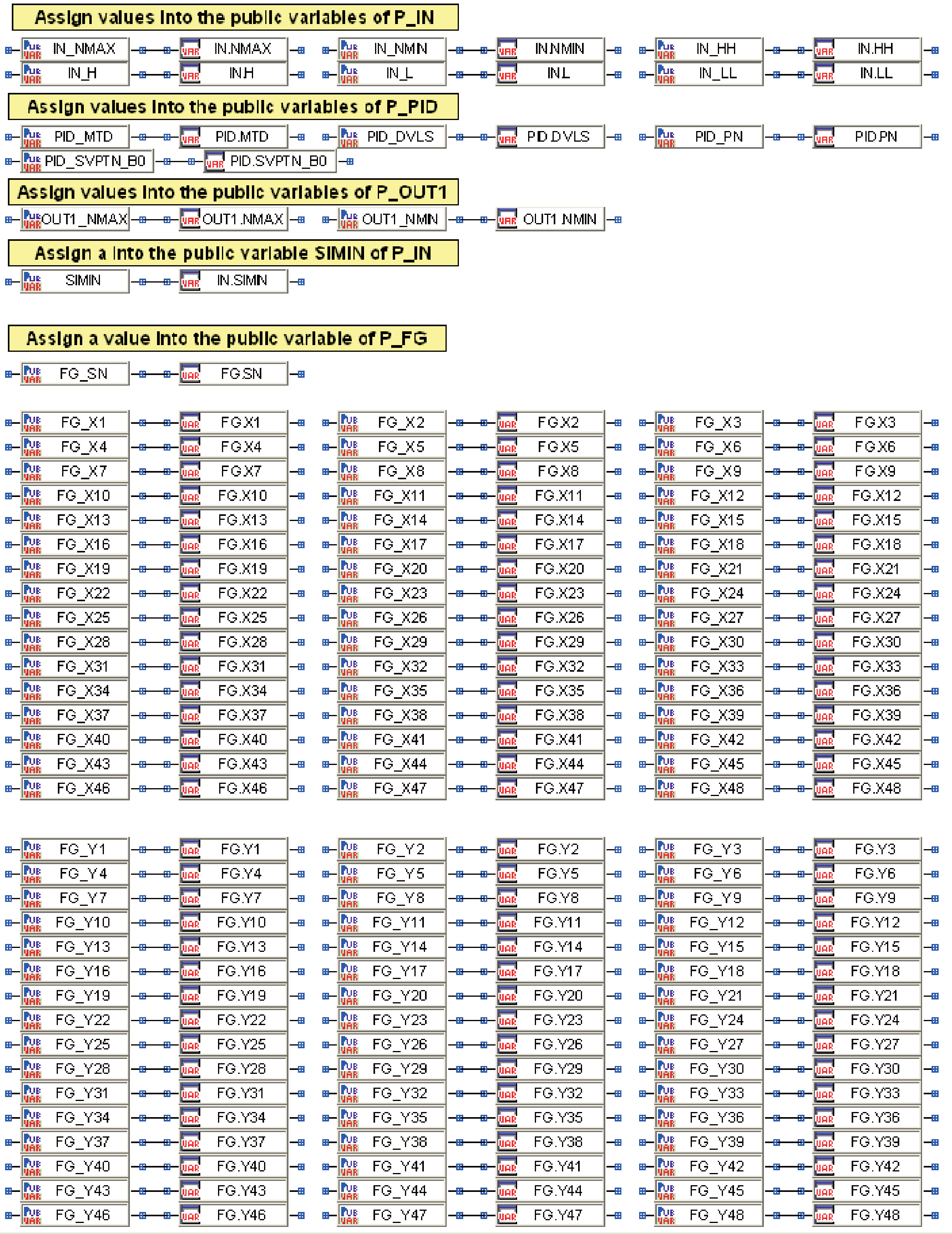
Point	• Create the deviation variable gain PID control (with deviation input and broken line correction) with user-defined tag FB and paste it to a program.

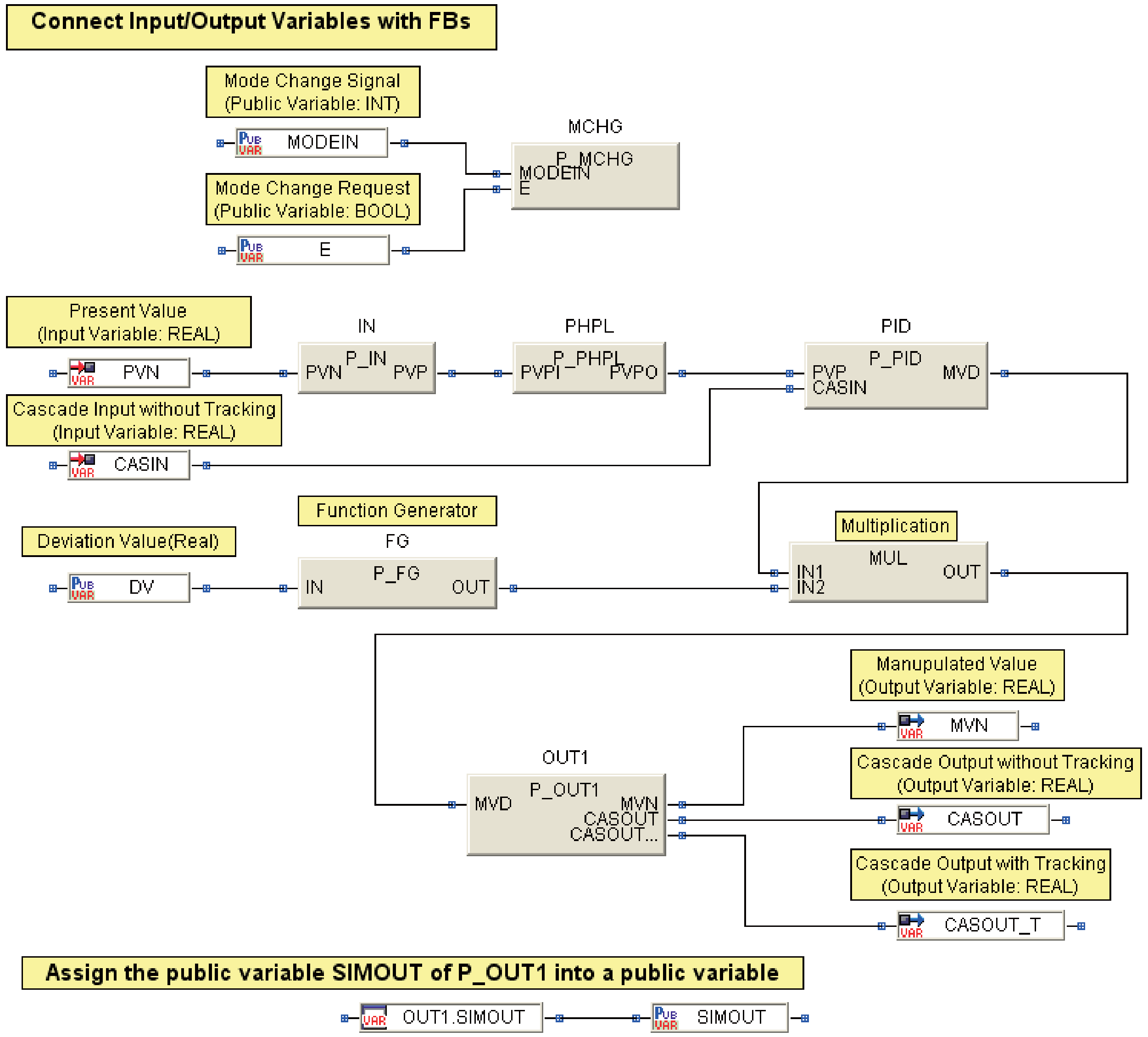


(2) Example of user-defined FB

1) Deviation variable gain PID User-defined FB (V_GAIN_PID)

Point	• Execute substitution to/from public variable to display FB public variable which composes user-defined tag FB as property of tag FB.





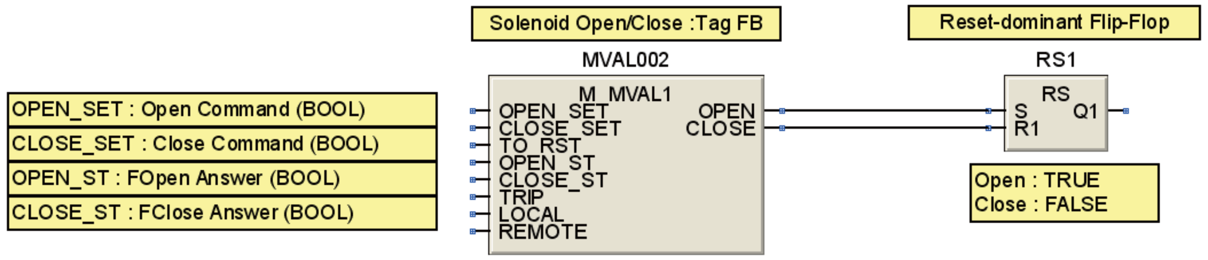
Pin	Variable name	Variable type	Data type	Contents
Input	PVIN	Input variable	REAL	PV input
Input	CASIN	Input variable	REAL	Cascade input without tracking
Output	MVN	Output variable	REAL	MV Output
Output	CASOUT	Output variable	REAL	Cascade output without tracking
Output	CASOUT_T	Output variable	ADR_REAL	Cascade output without tracking

3.4 Program Examples (digital/sequence control related)

3.4.1 Single solenoid

Function	• Output TRUE to open instruction and FALSE to close instruction from user program continuously when the control mode is AUTO.

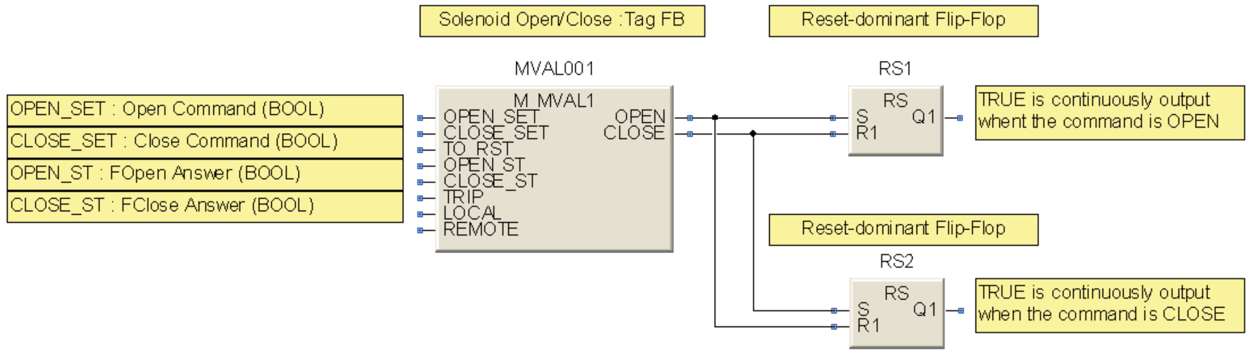
(1) Program example



3.4.2 Double solenoid

Function	• Output TRUE to both open instruction and close instruction from the external continuously when the control mode is AUTO.

(1) Program example

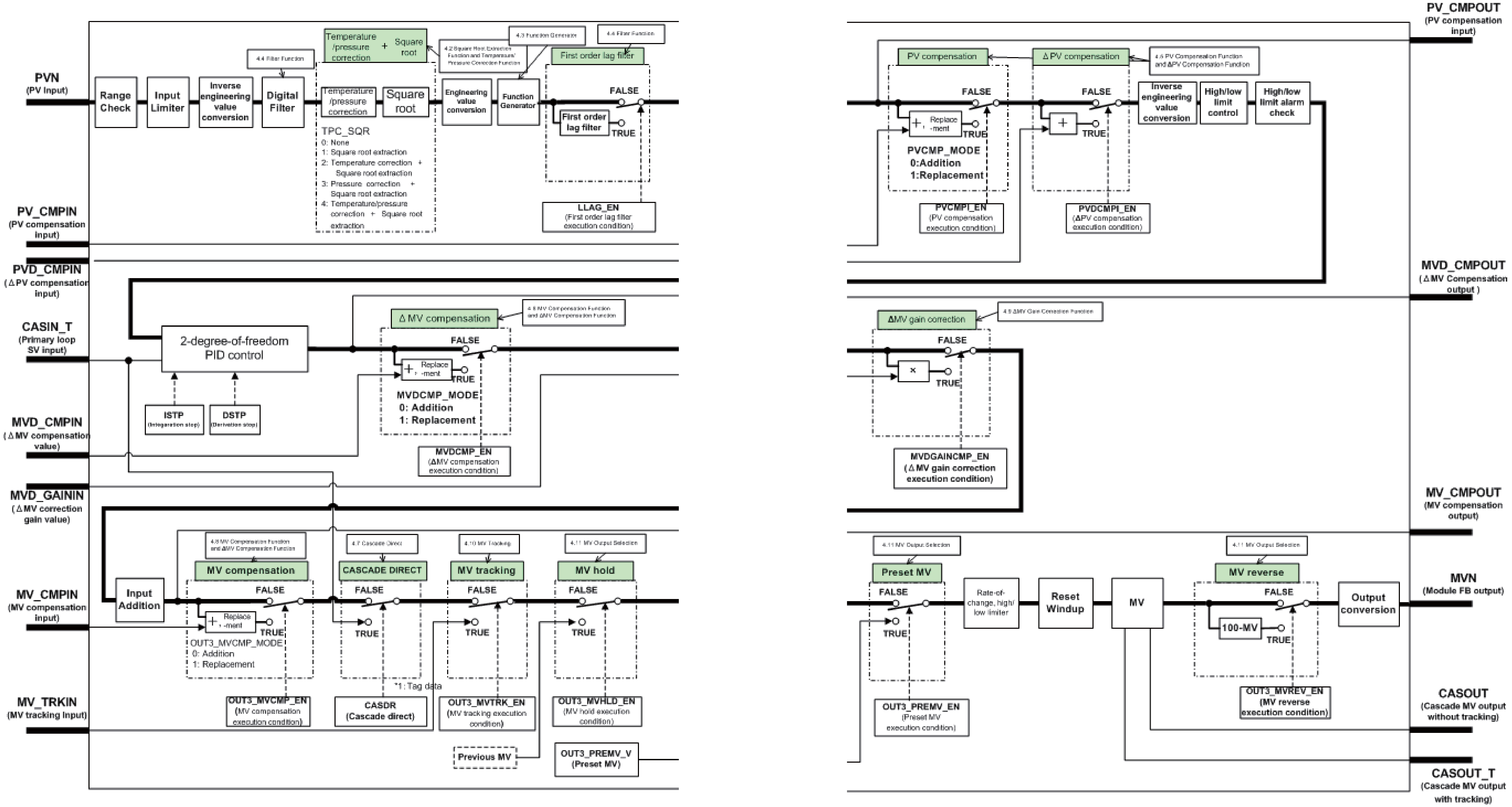


---

4 FUNCTION DETAILS OF 2-DEGREE-OF-FREEDOM ADVANCED PID CONTROL TAG

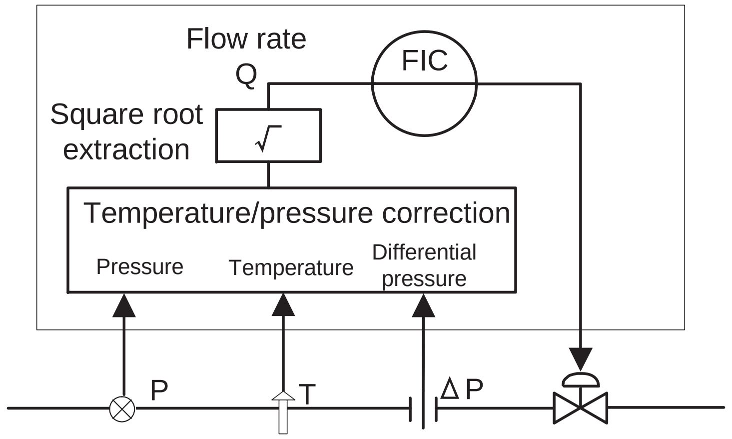
This chapter explains the function details of 2-degree-of-freedom advanced PID control tag which can be applied to simple to advanced controls.
Section 4.1 describes 2-degree-of-freedom advanced PID control tag (M_2PIDH_T_) block diagram.

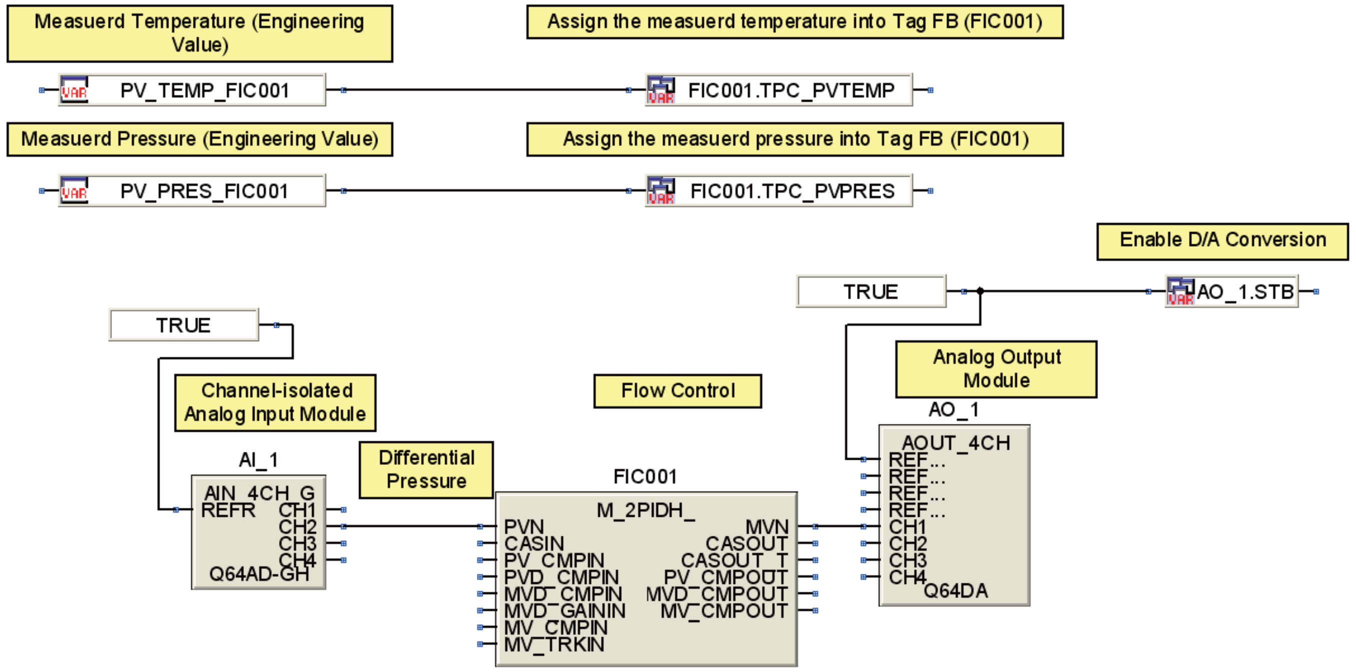
4.1 2-degree-of-freedom Advanced PID Control Tag (M_2PIDH_T_) Block Diagram



4.2 Square Root Extraction Function and Temperature/Pressure Correction Function

Execute the square root extraction to convert differential pressure signals of square characteristics into linear flow characteristic when measuring flow rate through differential pressure of orifice or venturi tube. Add temperature/pressure correction if necessary when measuring gas.
The 2-degree-of-freedom advanced PID control enables those functions with setting parameters.





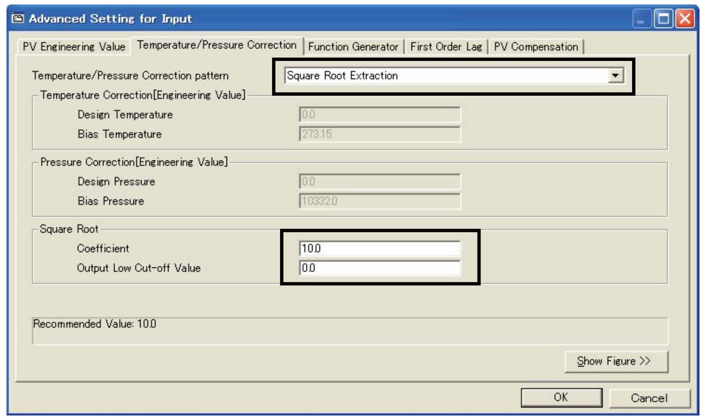
The following explains the temperature/pressure correction settings in flow rate control program shown above.

<Square root extraction>

The following shows the parameter settings when execute square root extraction.

Item	Contents
Temperature/ Pressure Correction pattern	Square Root Extraction
Coefficient	10 (Note 1)
Output Low Cut-off Value	10% (Note 2)

Set the items as shown above.



Temperature/Pressure Correction pattern: Square Root Extraction

Temperature Correction[Engineering Value]

Design Temperature: 0.0

Bias Temperature: 273.15

Pressure Correction[Engineering Value]

Design Pressure: 0.0

Bias Pressure: 10332.0

Square Root

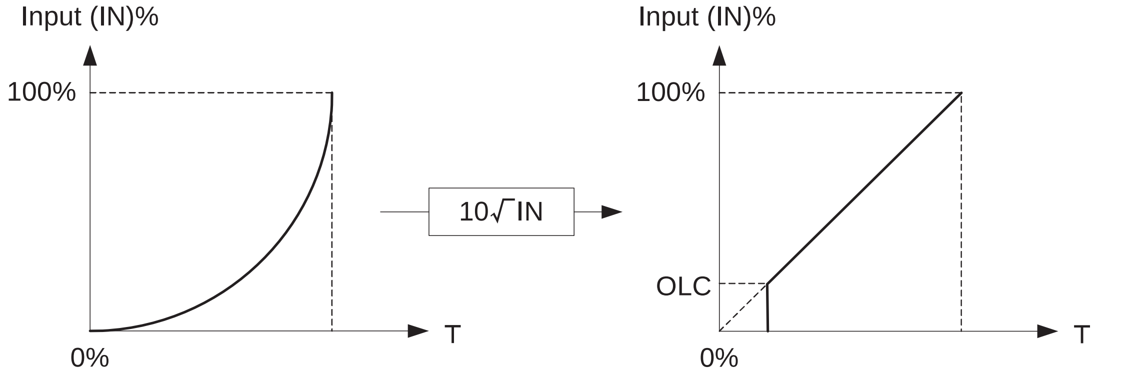
Coefficient: 10.0

Output Low Cut-off Value: 0.0

Recommended Value: 10.0

Property page

(Note 1) When input (IN) is in percentage (%), set the coefficient (K) to K=10.
Extract square root (10√IN) with K=10, make 0 to 100% of input (IN) corresponding to 0 to 100% of output (OUT). (When input = 100%, output = 10√100 = 100%).



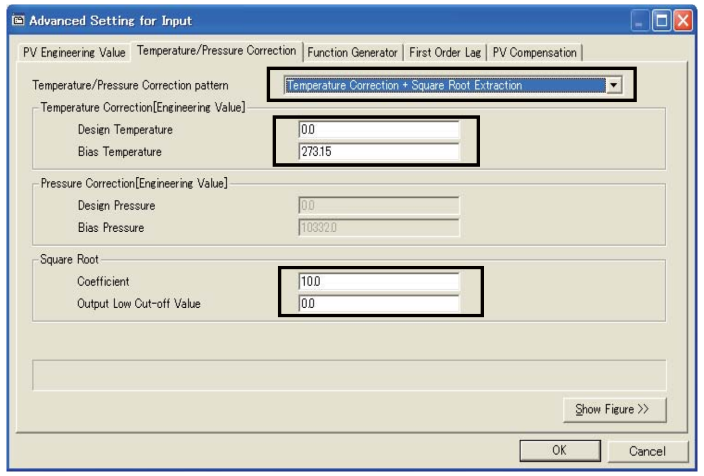
(Note 2) Generally, output low cut-off value (OLC) is applied when input (IN) is in percentage (%). (Output low cut-off value (OLC) shall be set in 10% level)

<Temperature correction + Square root extraction>

The following is the parameter settings for executing temperature correction and square root extraction.

Item	Contents
Temperature/ Pressure Correction pattern	Temperature Correction + Square Root Extraction
Design Temperature	0.0°C (Note 3)
Bias Temperature	273.15°C (Note 4)
Coefficient	10
Output Low Cut-off Value	10%

Set the items as shown above.



Temperature/Pressure Correction pattern: Temperature Correction + Square Root Extraction

Temperature Correction[Engineering Value]

Design Temperature: 0.0

Bias Temperature: 273.15

Pressure Correction[Engineering Value]

Design Pressure: 0.0

Bias Pressure: 10332.0

Square Root

Coefficient: 10.0

Output Low Cut-off Value: 0.0

Property page

(Note 3) Set the temperature specified in the design specification.
Use the same unit as measured temperature.

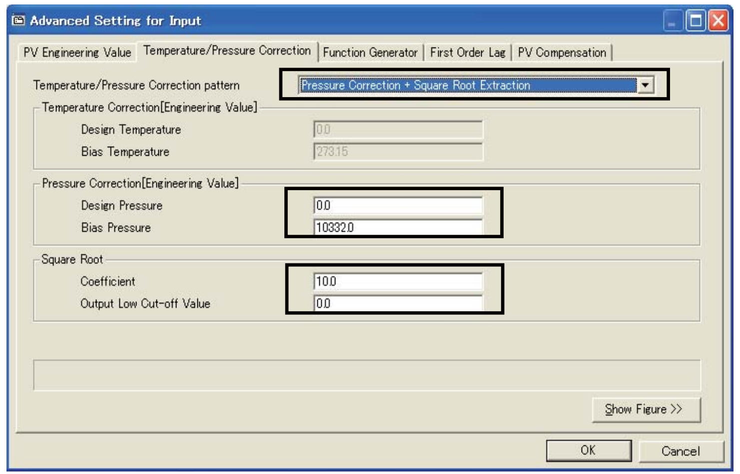
(Note 4) Set the bias temperature to perform the correction calculation with absolute temperature.
Set to 273.15 when Celsius is used for the design temperature and measured temperature.

<Pressure correction + Square root extraction>

The following shows the parameter settings when execute pressure correction + square root extraction.

Item	Contents
Temperature/ Pressure Correction pattern	Pressure Correction + Square Root Extraction
Design Pressure	0.0mmH_2O (Note 5)
Bias Pressure	10332.0mmH_2O (Note 6)
Coefficient	10
Output Low Cut-off Value	10%

Set the items as shown above.



Temperature/Pressure Correction pattern: Pressure Correction + Square Root Extraction

Temperature Correction[Engineering Value]

Design Temperature: 0.0

Bias Temperature: 273.15

Pressure Correction[Engineering Value]

Design Pressure: 0.0

Bias Pressure: 10332.0

Square Root

Coefficient: 10.0

Output Low Cut-off Value: 0.0

Property page

(Note 5) Set the pressure specified in the design specification.
Use the same unit as measured pressure.

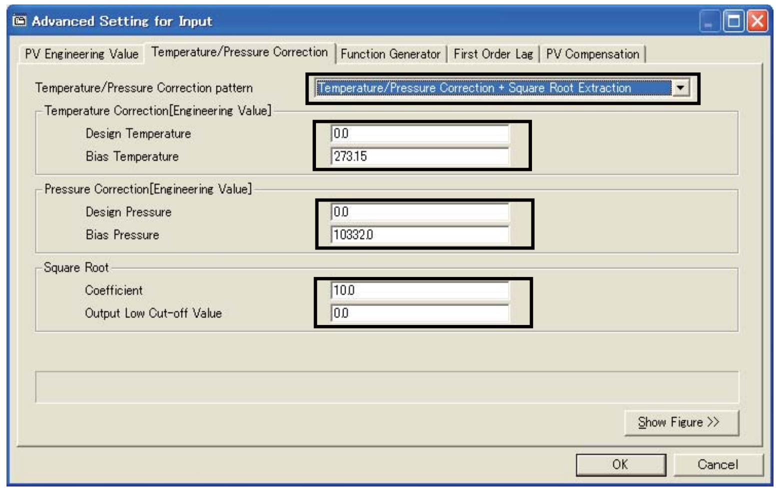
(Note 6) Set the bias pressure to perform the correction calculation with absolute pressure.
Set to 101.3 when kilo Pascal (kPa) is used for the design pressure and measured pressure.

＜Temperature/pressure correction + Square root extraction＞

The following shows the parameter settings when execute temperature/pressure correction + square root extraction.

Item	Contents
Temperature/ Pressure Correction pattern	Temperature Correction + Square Root Extraction
Design Temperature	0.0°C
Bias Temperature	273.15°C (Note 7)
Design Pressure	0.0mmH_2O (Note 8)
Bias Pressure	10332.0mmH_2O (Note 9)
Coefficient	10
Output Low Cut-off Value	10%

Set the items as shown above.



Temperature/Pressure Correction pattern: Temperature/Pressure Correction + Square Root Extraction

Temperature Correction[Engineering Value]

Design Temperature: 0.0

Bias Temperature: 273.15

Pressure Correction[Engineering Value]

Design Pressure: 0.0

Bias Pressure: 10332.0

Square Root

Coefficient: 10.0

Output Low Cut-off Value: 0.0

Property page

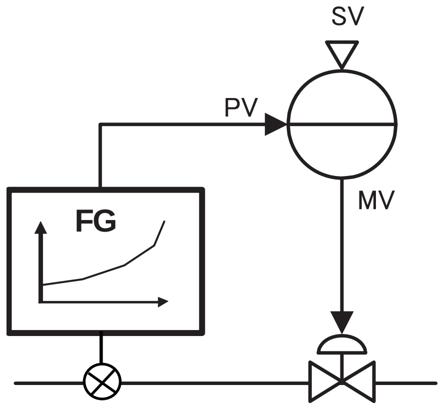
(Note 7) Set the bias temperature to perform the correction calculation with absolute temperature.
Set to 273.15 when Celsius is used for the design temperature and measured temperature.

(Note 8) Set the pressure specified in the design specification.
Use the same unit as measured pressure.

(Note 9) Set the bias pressure to perform the correction calculation with absolute pressure.
Set to 101.3 when kilo Pascal (kPa) is used for the design pressure and measured pressure.

4.3 Function Generator

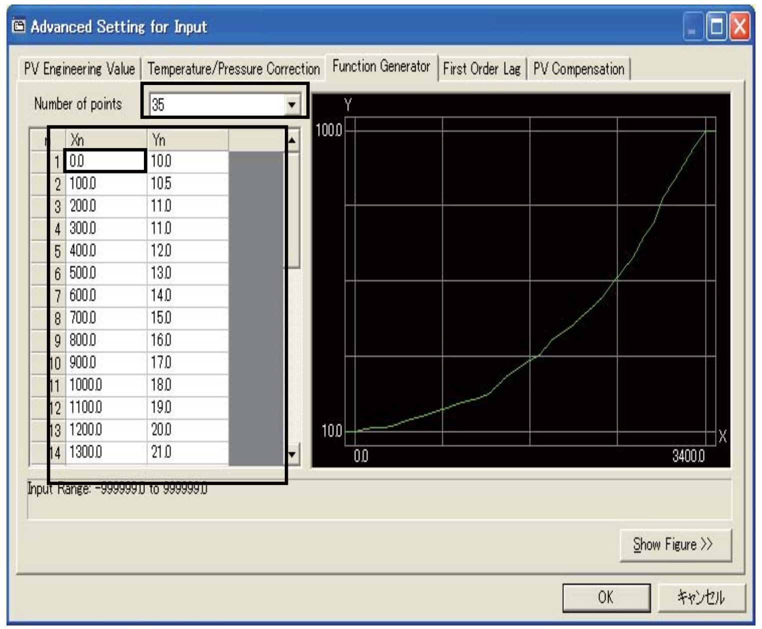
Approximate and correct by broken line correction processing when the value from the measure target is not in direct proportion to process variable from the sensor.



The following shows the parameter settings when execute broken line correction processing.

Item	Contents
Number of points	Set the number of points used in the broken line correction processing.
Input coordinates (X-coordinates)	Set the input coordinates (X-coordinates) of the broken line correction processing.
Output coordinates (Y-coordinates)	Set the output coordinates (Y-coordinates) of the broken line correction processing

Set the items as shown above.



Number of points: 35

n	Xn	Yn
1	0.0	10.0
2	100.0	10.5
3	200.0	11.0
4	300.0	11.0
5	400.0	12.0
6	500.0	13.0
7	600.0	14.0
8	700.0	15.0
9	800.0	16.0
10	900.0	17.0
11	1000.0	18.0
12	1100.0	19.0
13	1200.0	20.0
14	1300.0	21.0

Input Range: -999999.0 to 999999.0

Input broken line coordinates and display broken line graph.

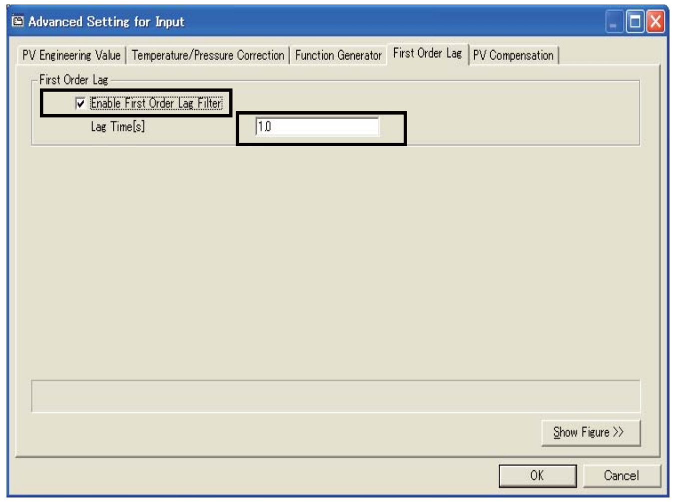
4.4 Filtering Function

(1) First order lag filter function

Use the first order lag filter to suppress the sudden change and the noise of input value so that the PV engineering variable is stable.
Set PV filter coefficient of (2) Digital filter function in this section to 0 and use the first lag filter.
The following shows the parameter settings when using the first lag filter.

Item	Contents
Enable First Order lag Filter	Check for enablement
Lag time [s] (time constant)	Set the time constant [s]. 0 to 999999

Set the items as shown above.



First Order Lag

☑ Enable First Order Lag Filter

Lag Time[s] 1.0

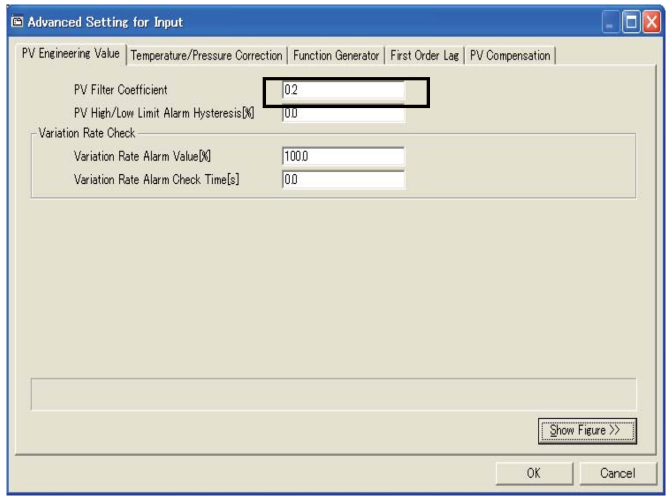
Property page

(2) Digital filter function

Execute digital filter (index filter) processing to PV which completed the execution of PV input high/low limiter processing.
The following shows the setting of filter coefficient (PV filter coefficient).

Item	Contents
PV Filter Coefficient	0.0 to 1.0

Set the items as shown above.



PV Filter Coefficient 0.2

PV High/Low Limit Alarm Hysteresis[%] 0.0

Variation Rate Check

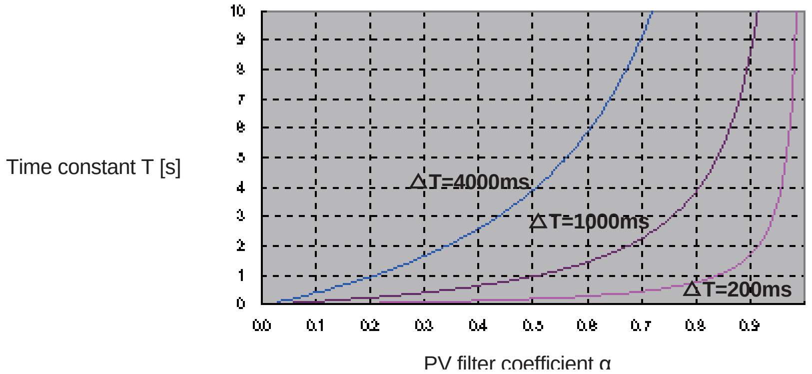
Variation Rate Alarm Value[%] 100.0

Variation Rate Alarm Check Time[s] 0.0

Property page

0.2 is set as initial value.
The effect of the filter differs depending on execution cycles.
The figure shown below is the relations between PV filter coefficient α and execution cycle ΔT as reference.
（Refer to Appendix 9 Process Control Related Terminology "Filter", "Execution cycle"）

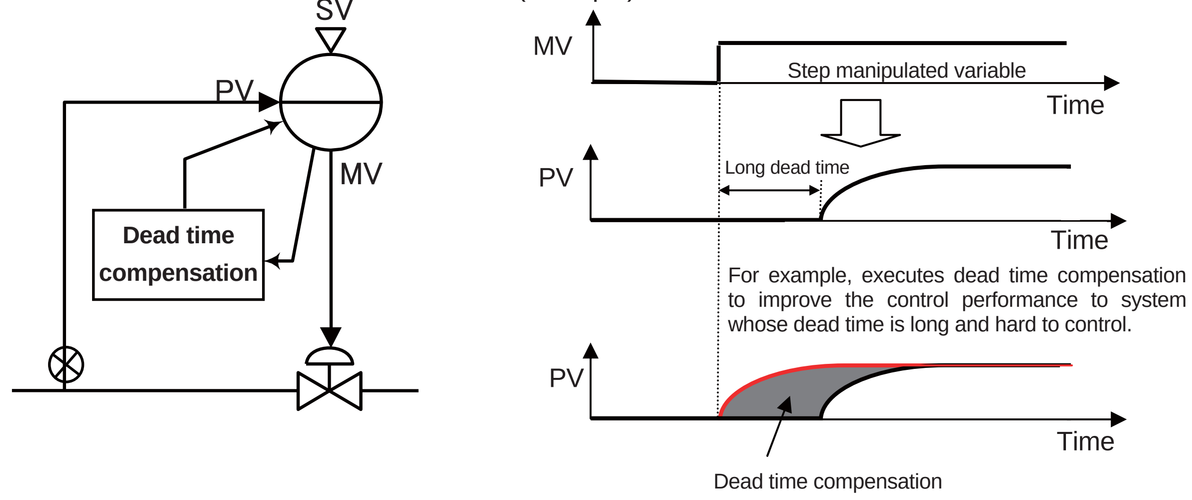
Relations between execution cycle ΔT of PV filter coefficient α and time constant T



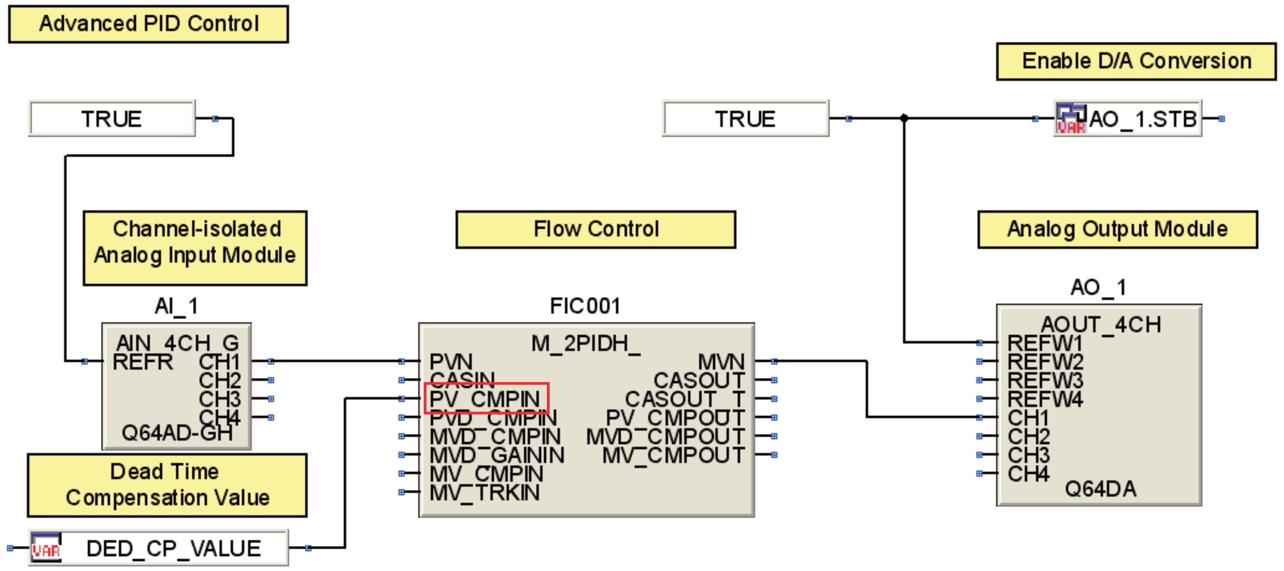
4.5 PV Compensation Function and ΔPV Compensation Function

To control processes with a long dead time, execute addition or replacement of the compensation value from the external (e.g. dead time compensation method) to PV input.

(Example)



For example, executes dead time compensation to improve the control performance to system whose dead time is long and hard to control.



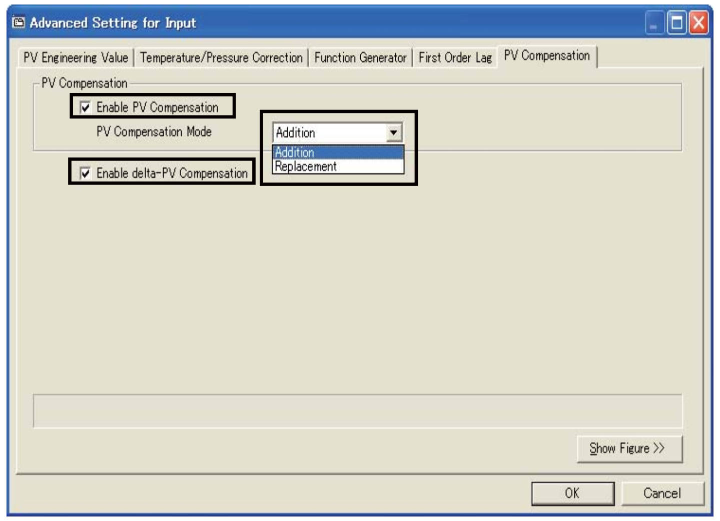
Input PV compensation value to input pin (PV_CMPIN).

PV Compensation program example

The following shows the parameter settings when execute PV compensation and ΔPV compensation

Item		Contents
PV Compensation	Enable PV Compensation	Check for enablement
	PV Compensation mode	Select Addition or Replacement.
delta-PV Compensation	Enable delta-PV Compensation	(Note 10)

Set the items as shown above.


Advanced Setting for Input

Tabs: PV Engineering Value | Temperature/Pressure Correction | Function Generator | First Order Lag | PV Compensation

PV Compensation

☑ Enable PV Compensation

PV Compensation Mode: Addition (Addition / Replacement)

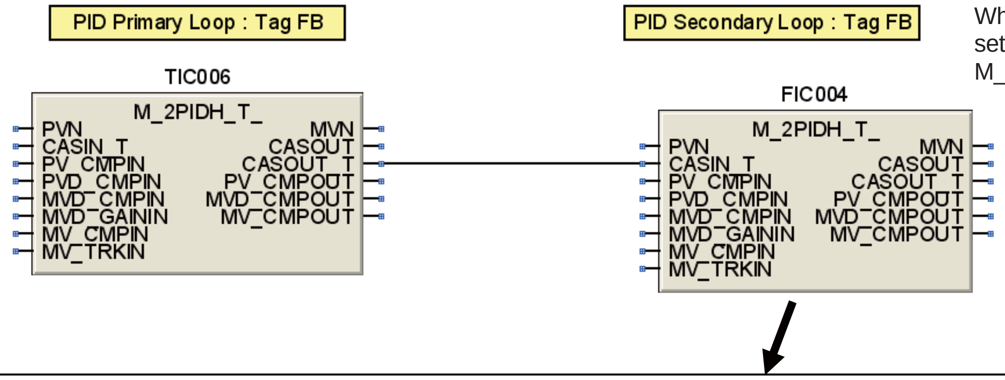
☑ Enable delta-PV Compensation

Property page

(Note 10) When ΔPV compensation is enabled, internally estimates the compensation value input in velocity type (ΔPV).
Then, adds the integration value to PV engineering variable. Therefore, even if compensation value becomes 0 due to such as the effect of break, the sudden change of PV engineering variable can be avoided.
When ΔPV compensation is set to Disable, the internal integration value of the compensation value is reset (set to 0).

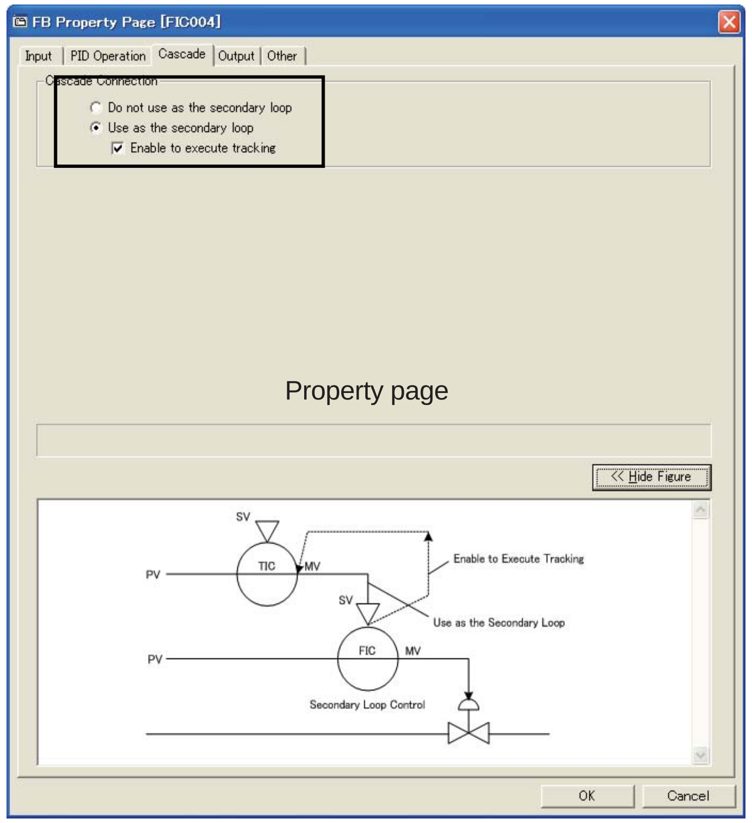
4.6 Cascade Connection

When execute cascade connection, set the tracking of cascade secondary and primary loops to secondary loop tag.
When execute the tracking to primary loop, use M_2PIDH_T_ which has the tracking function for tag FB type of secondary loop.
For primary loop tag FB type, both without tracking function (M_2PIDH_) and with tracking function (M_2PIDH_T_) can be used.



When execute tracking, set tag FB type to M_2PIDH_T_.

Item	Contents
Cascade Connection	Select whether or not to use as the secondary loop.
	When used as secondary loop, set whether or not to execute tracking


FB Property Page [FIC004]

Tabs: Input | PID Operation | Cascade | Output | Other

Cascade Connection

○ Do not use as the secondary loop

● Use as the secondary loop

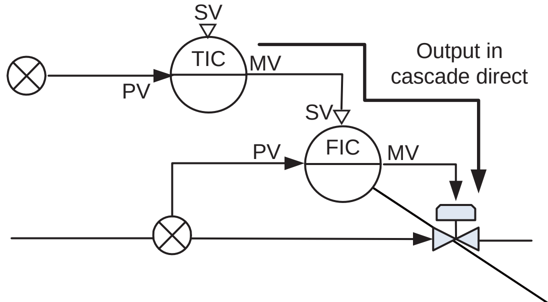
☑ Enable to execute tracking

Property page

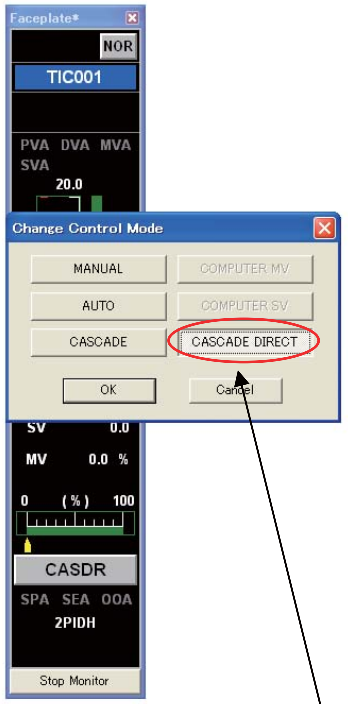
Setting example of secondary loop tracking in cascade connection

4.7 Cascade Direct

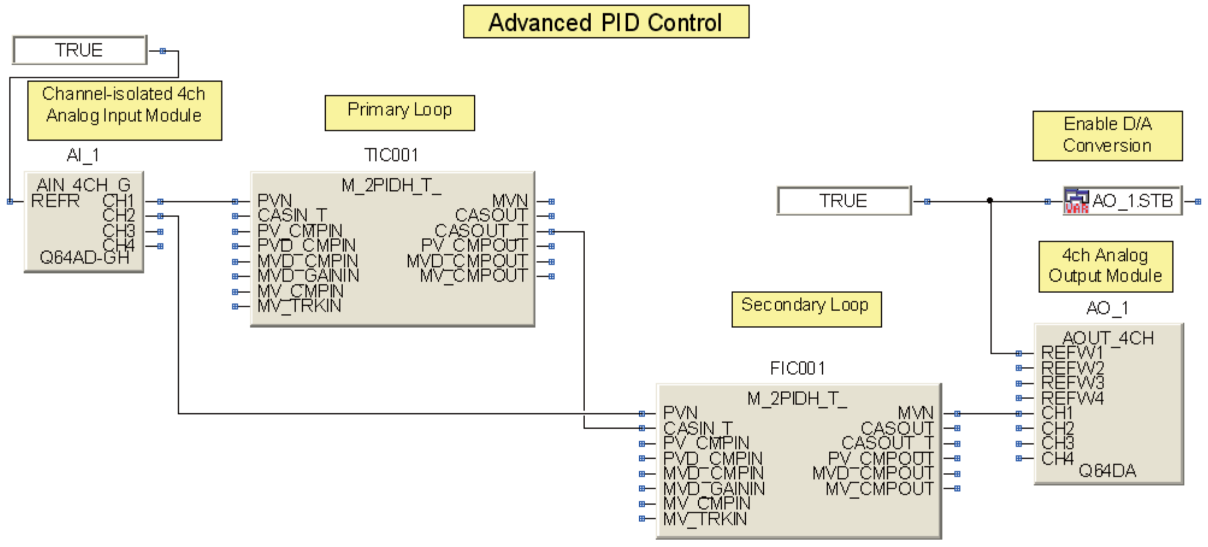
This is the mode for directly outputting the output value of primary loop as the output value of secondary loop in the cascade connection.



Loop with cascade direct



Cascade direct is enabled when switching the control mode of secondary loop (FIC001) from faceplate of the tuning panel to CASCADE DIRECT.



Program example of cascade connection

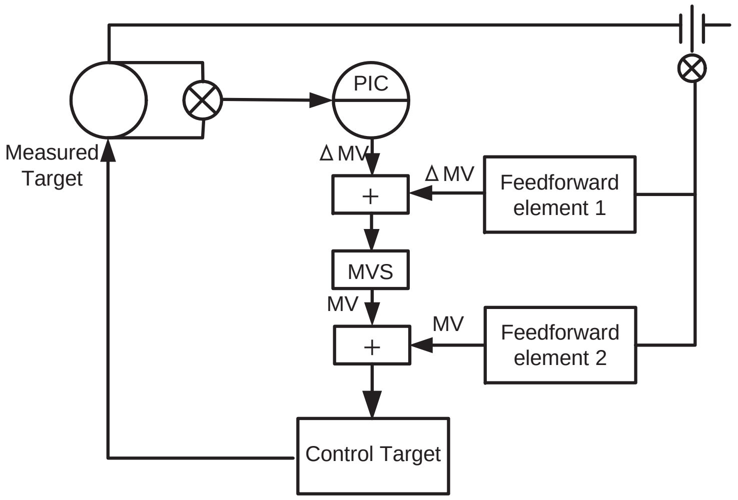
4.8 MV Compensation Function and ΔMV Compensation Function

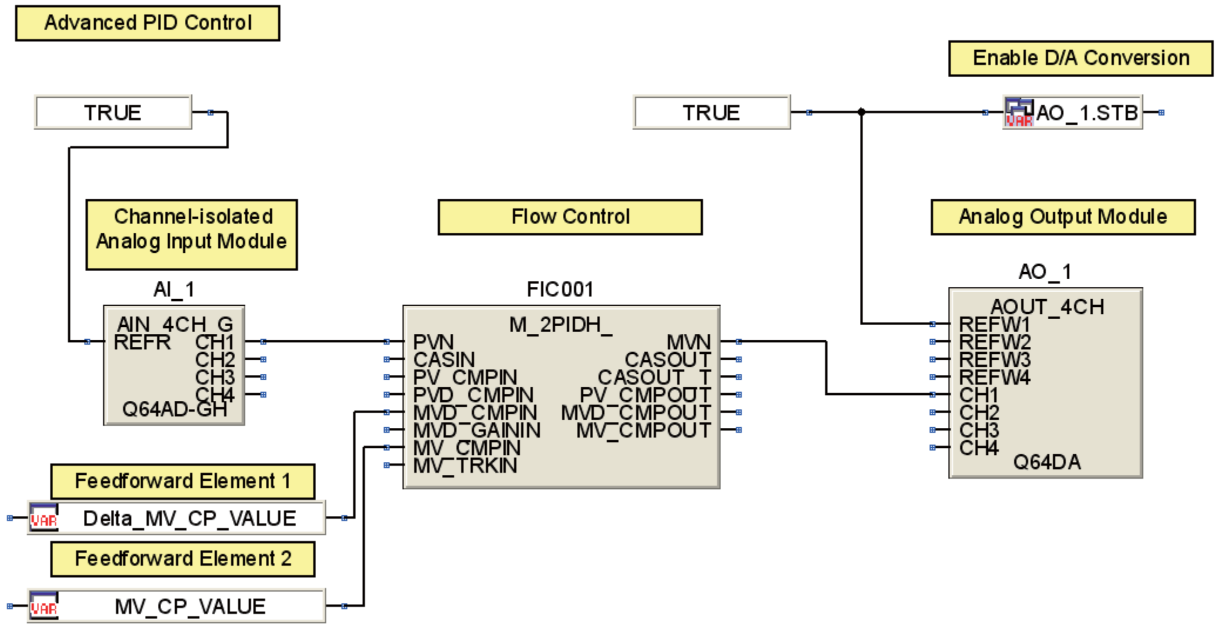
(1) Example of application to feedforward control

The feedforward control is applied to 3 component control and combustion control of boiler.

The feedback control is generally combined with the feedforward control when variation of the operation is clear since time lag occurs when responding to disturbance with feedback control only.

Set output quantity of feedforward control to the compensation value of ΔMV compensation or MV compensation.



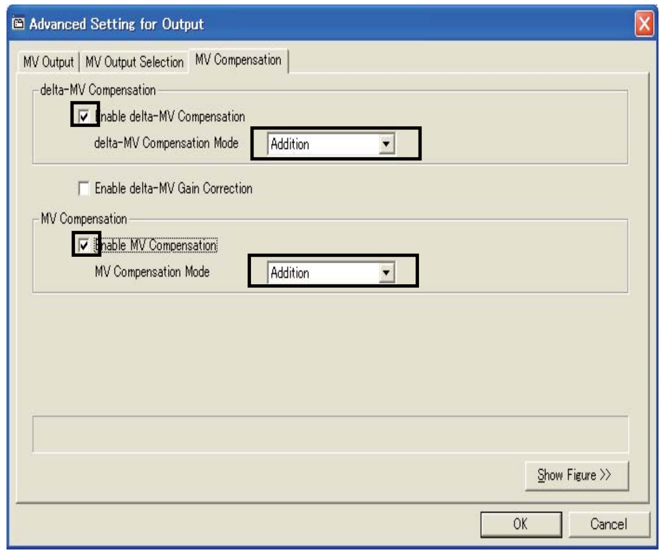


Program example of MV compensation and ΔMV compensation

The following shows the parameter settings when execute MV compensation and ΔMV compensation.

Item		Contents
MV Compensation	Enable MV Compensation	Check for enablement
	MV Compensation Mode	Select Addition or Replacement.
Delta-MV Compensation	Enable delta-MV Compensation	Check for enablement
	Delta-MV Compensation Mode	Select Addition or Replacement.

Set the items as shown above.


Advanced Setting for Output

MV Output | MV Output Selection | MV Compensation

delta-MV Compensation

☑ Enable delta-MV Compensation

delta-MV Compensation Mode: Addition

☐ Enable delta-MV Gain Correction

MV Compensation

☑ Enable MV Compensation

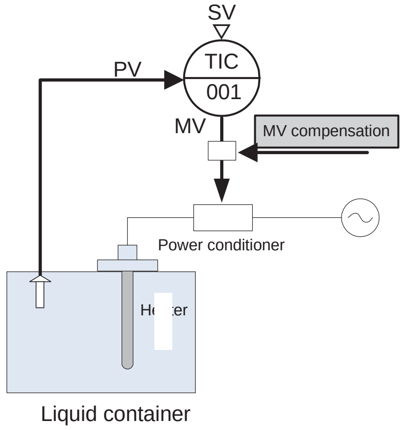
MV Compensation Mode: Addition

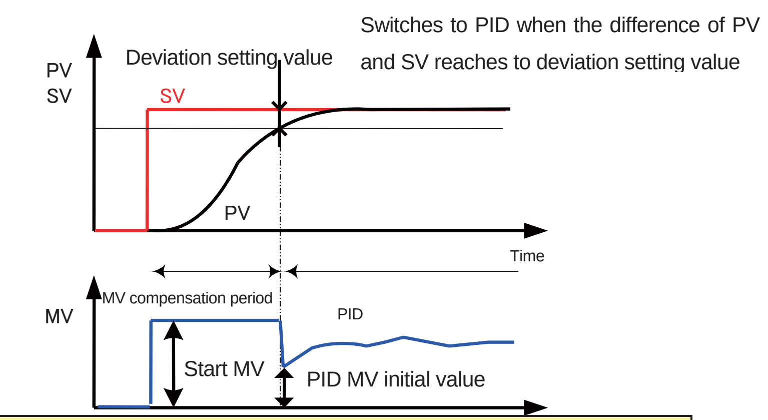
Property page

---

(2) Example of application to batch PID

This is an example of application to batch PID which outputs constant variable of MV in the start and switches to PID when the difference of PV and SV reaches to the deviation setting value. Constant variable of MV output in the start and PID MV initial value in switching to PID can be applied with MV compensation input (MV_CMPIN of TICOO1 in this example). Batch PID prevents overshoot of MV and PV which is occurred in temperature rising of late response process to which general PID is applied. Deviation setting value, constant variable in the start (start MV), and PID MV initial value are decided by actual operation of control target.



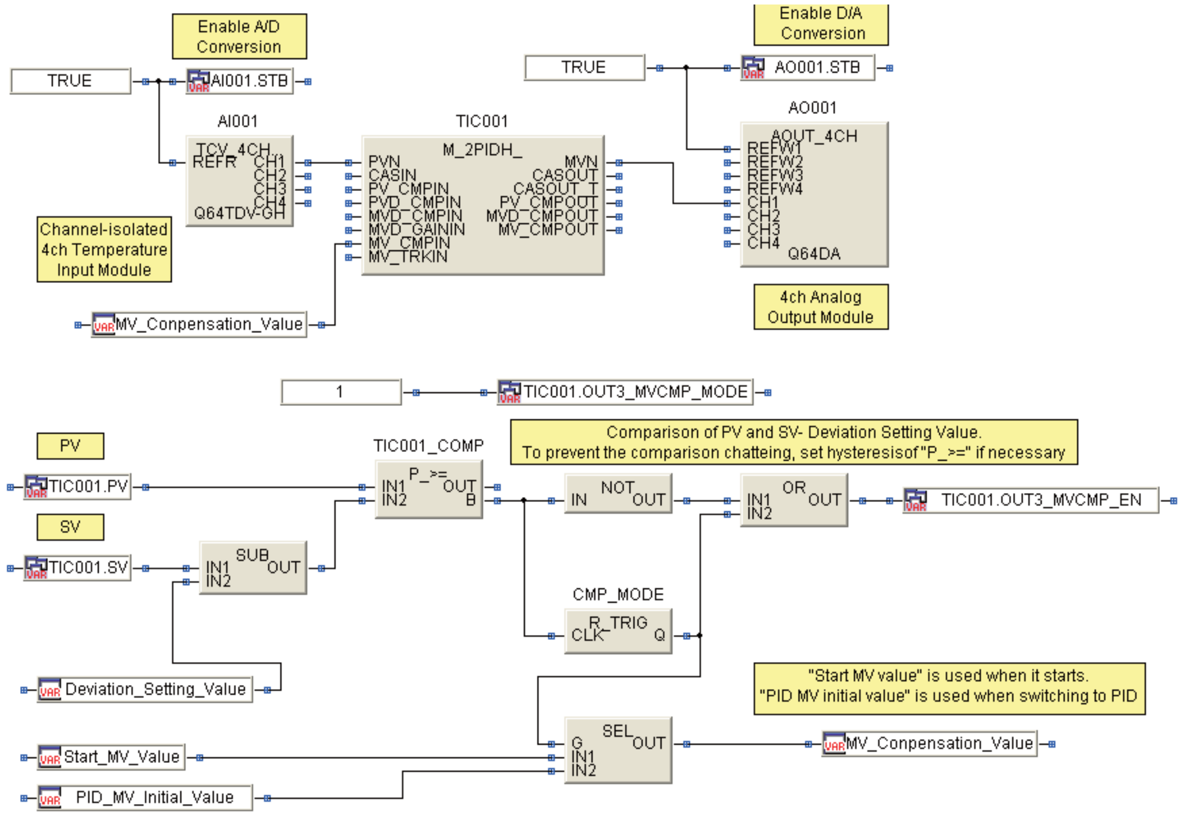


Processing abstract

1) Input MV compensation to M 2PIDH loop tag
  - Connect MV compensation value to input pin MV_CMPIN.

2) MV compensation mode setting
  - Set the MV compensation function to the replacement mode. (Replace MV to MV compensation value). (Executes when MV compensation execution condition is TRUE)

3) MV compensation period
  - Substitute start MV to MV compensation value.
  - While PV < (SV - Deviation setting value), MV compensation execution condition is TRUE. (While MV compensation execution condition is TRUE, enables the MV compensation function)
  - Substitute PID MV initial value to MV compensation value in switching to PID.

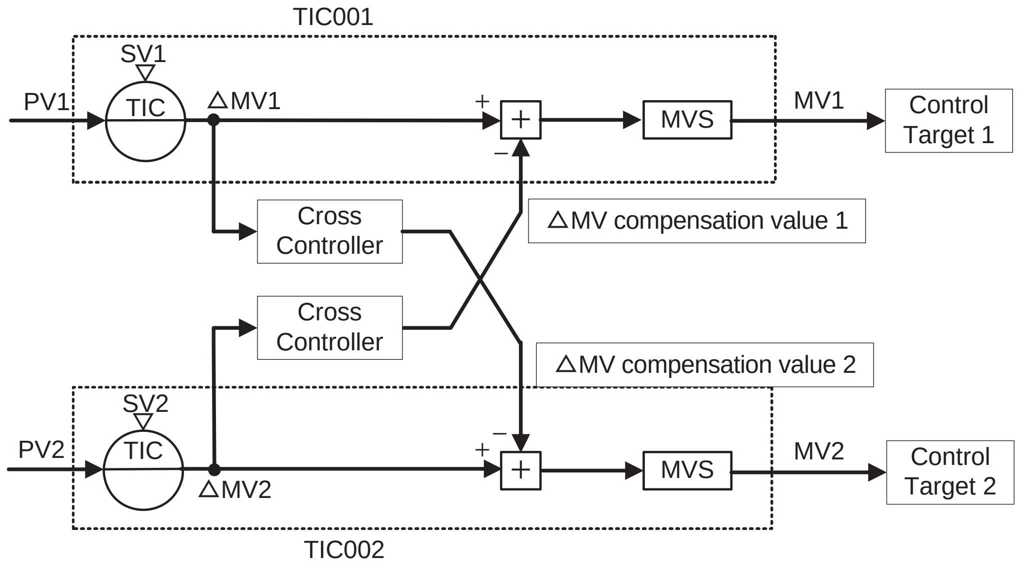


Program example of batch PID

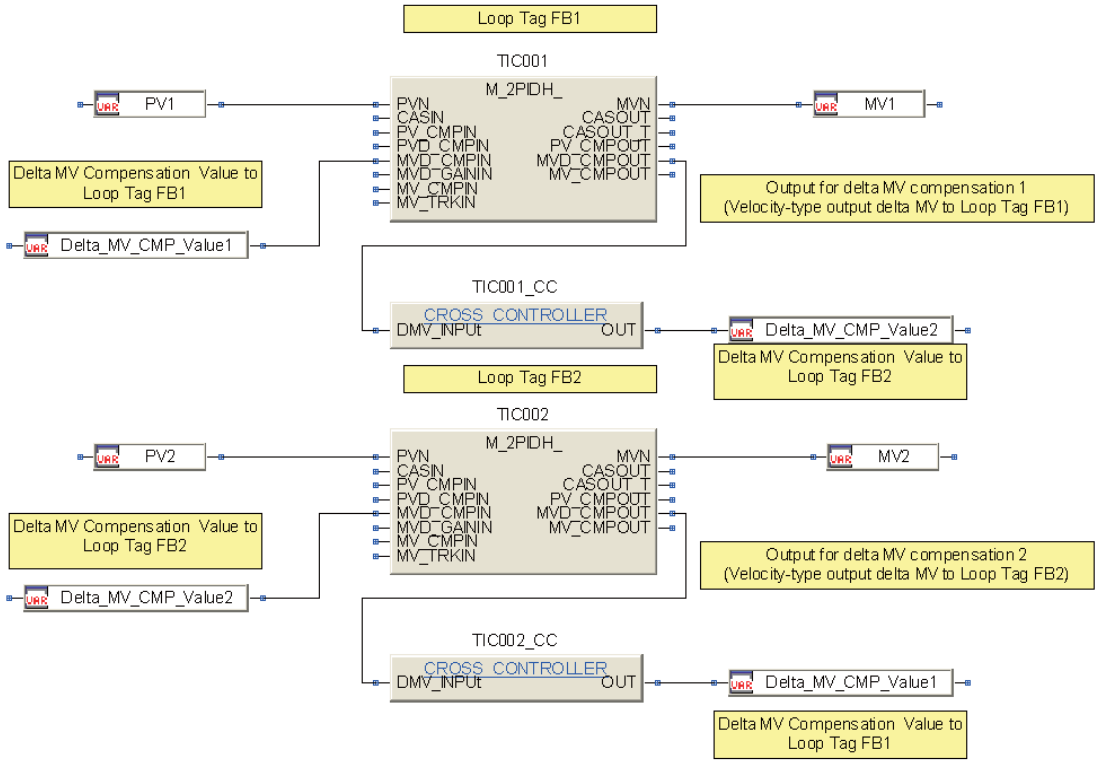
(3) Example of application to decoupling control

This is an example of application to decoupling control which prevents interaction between control loops and enables control systems to be operated as independent control systems when the target controls interact each other.

To eliminate the interaction between control loops, combine cross controller (decoupling compensator) as feedforward component.



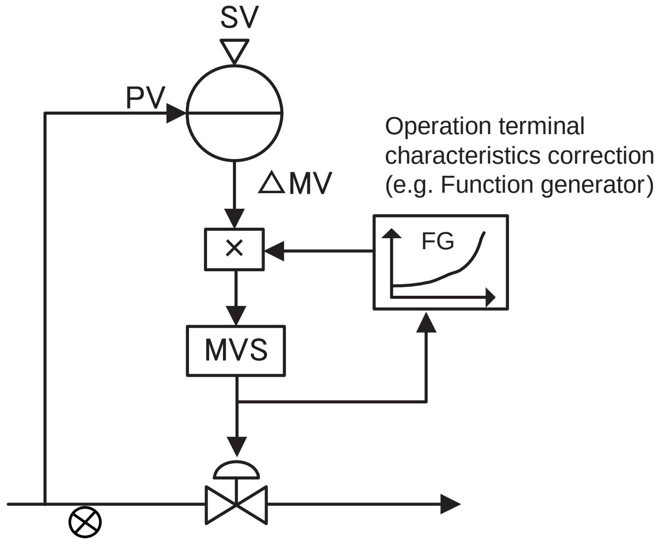
Substitute MVD_CMPOUT (output for ΔMV compensation) as velocity output ΔMV of loop tag FB to cross controller and input to MVD_CMPIN (ΔMV compensation value) of another system loop tag FB as feedforward element. ΔMV compensation mode is addition (=0: default).

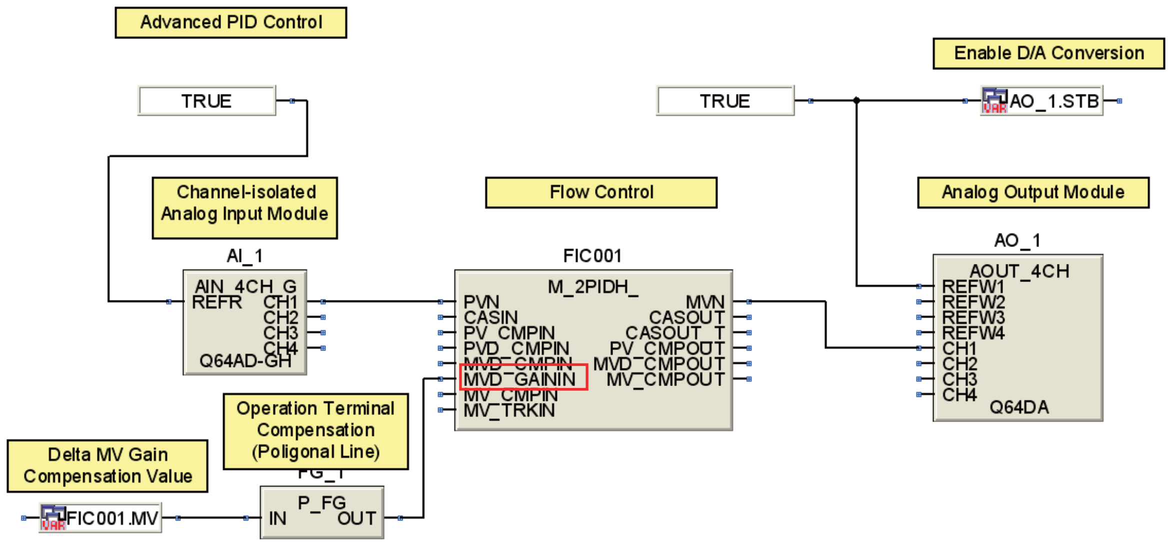


Program example of decoupling control

4.9 ΔMV Gain Correction Function

It executes operation terminal characteristics correction by executing ΔMV gain correction and linearization of manipulated variable with such as function generator to stabilize the controls to loads and changes of the target value.





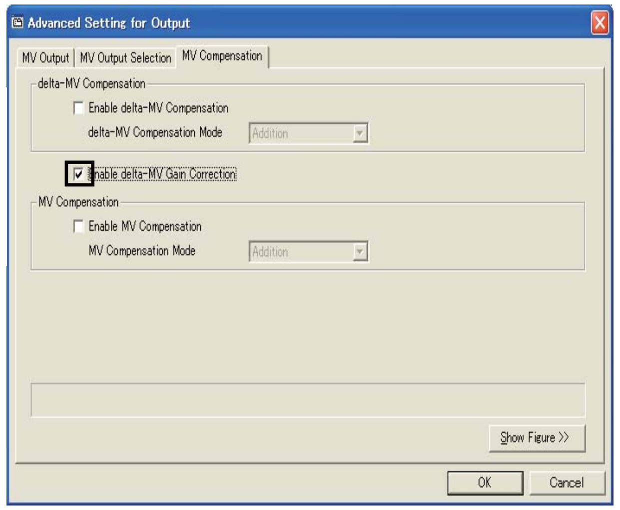
Input ΔMV correction gain value to input pin (MVD_GAININ)

Program example of ΔMV gain correction

The following shows the parameter settings when execute ΔMV gain correction.

Item	Contents
Enable delta-MV Gain Correction	Check for enablement

Set the items as shown above.


Advanced Setting for Output

MV Output | MV Output Selection | MV Compensation

delta-MV Compensation
☐ Enable delta-MV Compensation
delta-MV Compensation Mode: Addition

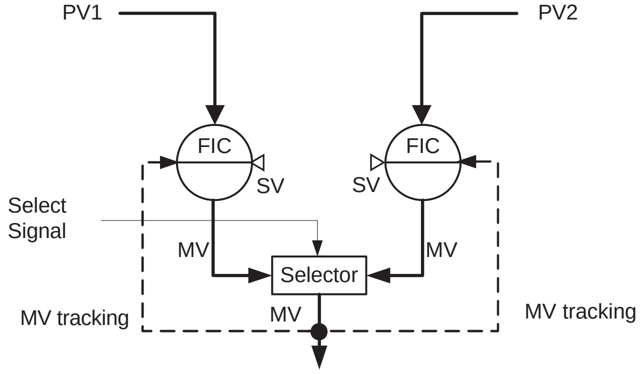
☑ Enable delta-MV Gain Correction

MV Compensation
☐ Enable MV Compensation
MV Compensation Mode: Addition

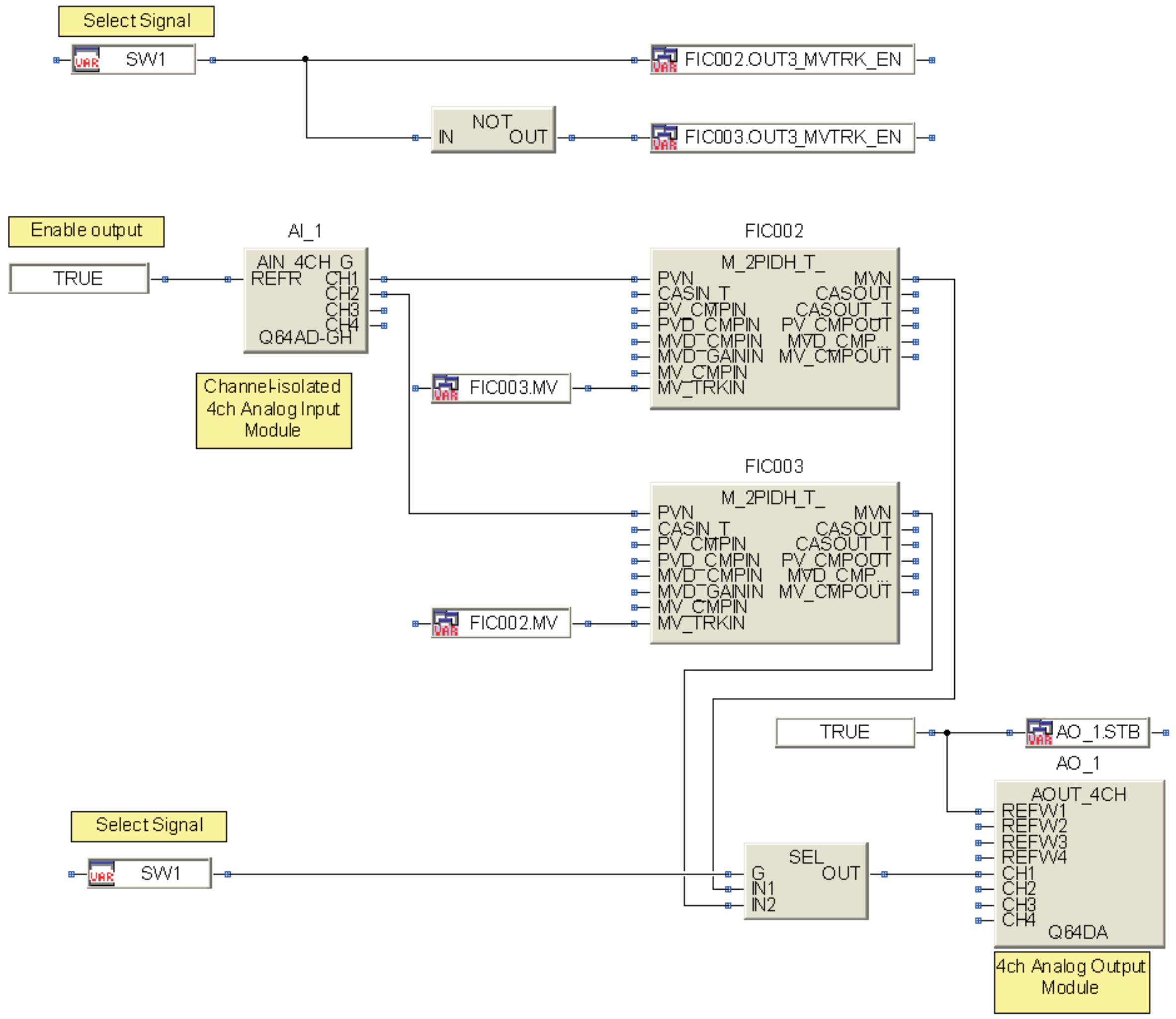
Property page

4.10 MV Tracking

Switch MV to tracking input. (Except in manual mode)



In 2 PID control loops, tracking with transferring MV whose loop is selected by selector to MV whose loop is not selected by selector executes the bumpless when switching to loop.
In the figure shown below, executes tracking with transferring MV of loop 1 to MV of loop 2 when select signal SW1 is FALSE. Executes tracking with transferring MV of loop2 to MV of loop 1 when select signal SW1 is TRUE.



Program example of MV tracking

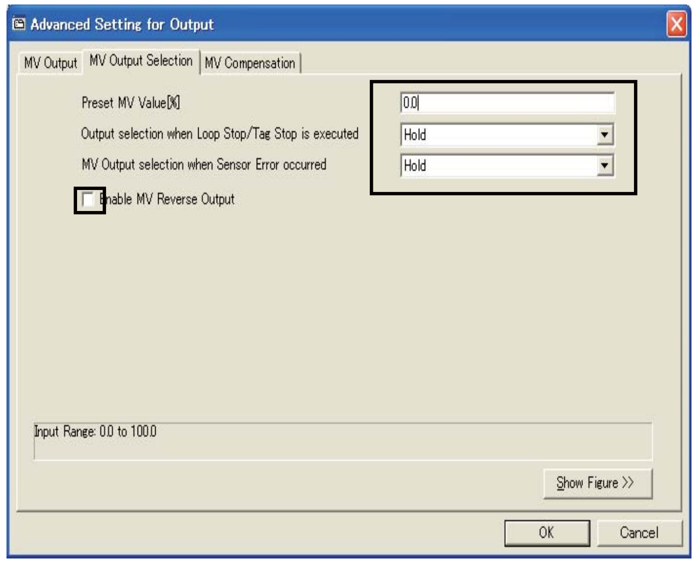
4.11 MV Output Selection

Select the output (MV hold, preset MV) for the event of an error or tag stop.

The following shows FB property settings regarding preset MV, output selection in abnormal occasions and MV reverse output.

Item	Contents
Preset MV [%]	Regard preset MV as output value.
Output selection when loop Stop/Tag Stop is executed	Set the method for MV output in loop stop or tag stop. (Note 11)
MV Output selection when Sensor Error occurred	Set the method for MV output in the occurrence of sensor alarm. (Note12)
Enable MV Reverse Output	Set whether or not to output MV reverse. (Note13)

Set the items as shown above.



Preset MV Value[%] 0.0

Output selection when Loop Stop/Tag Stop is executed Hold

MV Output selection when Sensor Error occurred Hold

☐ Enable MV Reverse Output

Input Range: 0.0 to 100.0

Property page

(Note 11) Select either "Hold" or "Preset value". When "Preset" is selected, MV cannot be changed on a faceplate.

(Note 12) Select from "Hold", "Preset MV output", and "Do not hold and output preset MV". When selecting "Do not hold and output preset MV", the result of PID operation + Output addition processing is output.

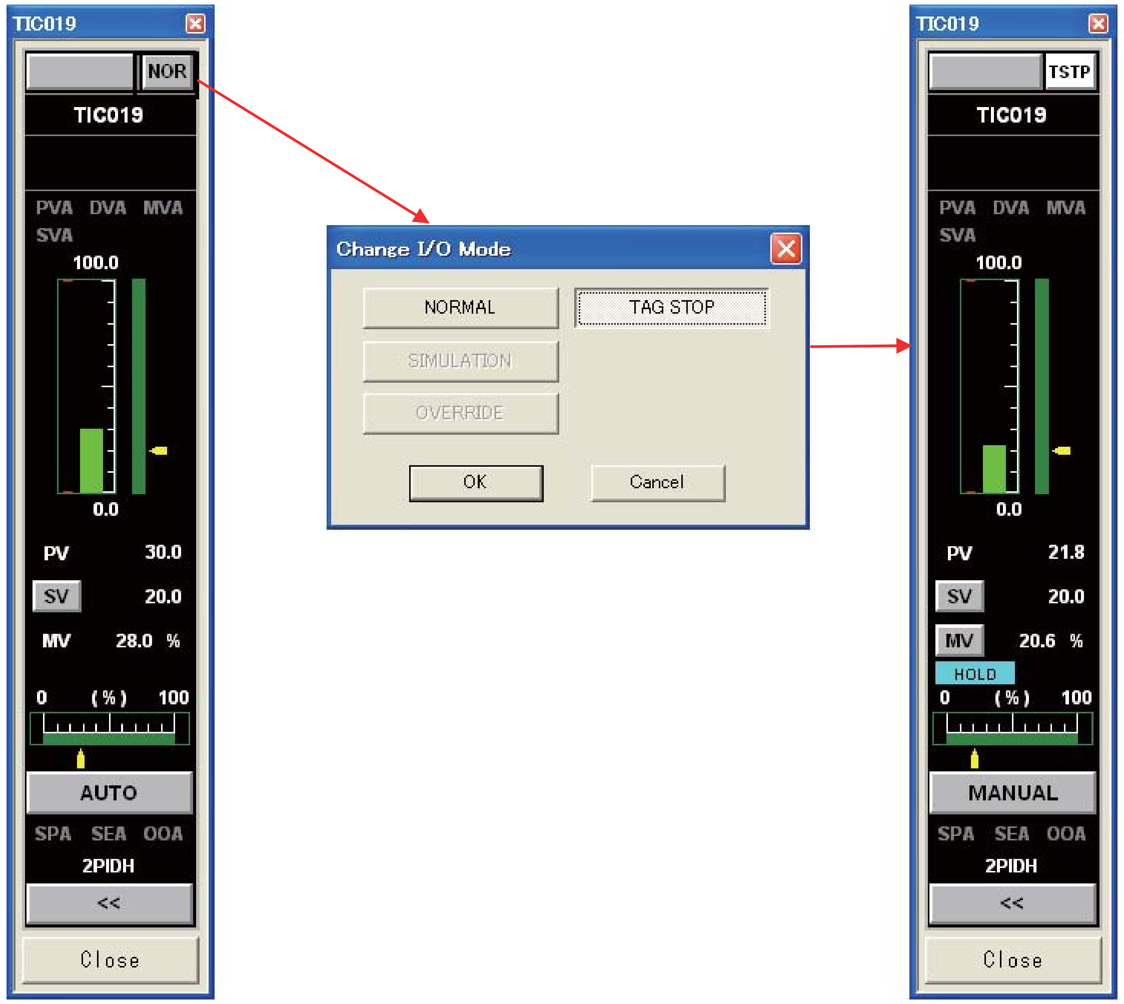
(Note 13) When selecting "MV reverse output", outputs variable which is after execution of MV inverted processing (100 - MV).

[REMARK]

Loop stop (SPA)	: A function which stops loop process operation in the event of loop error. Loop stops by changing tag name.SPA as TRUE
Tag stop (TSTP)	: A function which stops loop process operation with faceplate (TAG STOP)
Output open alarm (OOA)	: A function which activates the alarm when a disconnection is detected on the module FB of the output side. Output open alarm occurs by changing tag name.OOA as TRUE.
Sensor error (SEA)	: Sensor alarm occurs in the event of analog input high limit range error or low limit range error.

4.12 Tag Stop Function

The tag stop function stops the input processing and loop process operation.
Setting tag stop changes the control mode to MANUAL automatically.
"Preset" or "Hold" can be selected for MV in changing to MANUAL mode. (Refer to Section 4.11)
The tag stop function can be set with the Change I/O mode dialog box of faceplate. For details of I/O mode change, refer to the "PX Developer Operating Manual (Monitor Tool)".



Appendix 1 Specifications of Parameters for Loop Control

The following explains the details of the parameters for 2-degree-of-freedom advanced PID control FB (M_2PIDH_T_) as representative examples

Appendix 1.1 Minimum parameters required for 2-degree-of-freedom advanced PID control

Item (variable name)	Data type	Contents	Setting range	Initial value
IN_NMAX	REAL	**Analog input high limit value** High limit value for the range of A/D conversion values (such as 0 to 4000, 0 to 8000) input from an analog input module. (Example) When using the range of 0 to 64000 for Q64AD-GH → Set to "64000". After checking range errors (IN_HH,IN_H,IN_LL, IN_L) which are mentioned later in this section, executes input high limit processing which sets AD conversion value to IN_NMAX when AD conversion value ≧ IN_NMAX.	-999999 to 999999	100.0
IN_NMIN	REAL	**Analog input low limit value** Low limit value for the range of A/D conversion values (such as 0 to 4000, 0 to 8000) input from an analog input module. (Example) When using the range of 0 to 64000 for Q64AD-GH → Set to "0". After checking range errors (IN_HH,IN_H,IN_LL,IN_L) which are mentioned later in this section, executes input low limit processing which sets AD conversion value to IN_NMIN when AD conversion value ≦ IN_NMIN.	-999999 to 999999	0.0
IN_HH	REAL	**Analog input high limit range error value** Reference value of high limit exceeding error (range high limit error) for A/D conversion values input from an analog input module. When AD conversion value is greater than this value, high limit range error (sensor alarm SEA) occurs. (Example) When using the range of 0 to 64000 for Q64AD-GH → Set to "65535".	-999999 to 999999	110.0
IN_H	REAL	**Analog input high limit range error reset value** Reference value of error reset performed after high limit range error occurrence. When A/D conversion value is smaller than this value, the high limit range error (sensor alarm SEA) is reset. (Example) When using the range of 0 to 64000 for Q64AD-GH → Set to "64000".	-999999 to 999999	100.0
IN_LL	REAL	**Analog input low limit range error value** Reference value of low limit exceeding error (range low limit error) for A/D conversion values input from an analog input module. When A/D conversion value is smaller than this value, low limit range error (Sensor alarm SEA) occurs. (Example) When using the range of 0 to 64000 for Q64AD-GH → Set to "-1536".	-999999 to 999999	-10.0

Item (variable name)	Data type	Contents	Setting range	Initial value
IN_L	REAL	**Analog input low limit range error reset value** Set reference value of error reset performed after low limit range error occurrence. When A/D conversion value is greater than this value, the low limit range error (Sensor alarm SEA) is reset. (Example) When using the range of 0 to 64000 for Q64AD-GH → Set "0".	-999999 to 999999	0.0
PID2H_PN	INT	**Reverse action / direct action** Set PID operation pattern. Reverse action increases the manipulated variable (MV) when the process variable (PV) decreases more than the setting value (SV). Direct action increases the manipulated variable (MV) when the process variable (PV) increases more than the setting value (SV).	0:Reverse action 1:Direct action	0
OUT3_NMAX	REAL	**Analog output conversion high limit value** High limit value for the range of D/A conversion values (such as 0 to 4000, 0 to 8000) for writing to an analog output module. (Example) When using the range of 0 to 12000 for Q64DA → Set to "12000".	-999999 to 999999	100.0
OUT3_NMIN	REAL	**Analog output conversion low limit value** Low limit value for the range of D/A conversion values (such as 0 to 4000, 0 to 8000) for writing to an analog output module. (Example) When using the range of 0 to 12000 for Q64DA → Set to "0".	-999999 to 999999	0.0

Appendix 1.2 Parameters set for 2-degree-of-freedom advanced PID control if necessary

(1) Parameter setting when engineering variable is used

Engineerring variables (variables which are described with Industrial unit such as l/min, kg, °C) is used for process variable (PV) and setting value (SV). For manipulated variable (MV), % is fixed and cannot be described with engineering variable.

Item (variable name)	Data type	Contents	Setting range	Initial value
UNIT	INT	**Engineering unit** Units for parameters which have engineering variables (PV, SV, RH, RL, PH, PL, HH, LL, SH, SL). Set a specified unit name No. which is registered on the Unit setting screen of the monitor tool to UNIT (unit setting is needed in the monitor tool).	0 to 127	0
N	INT	**Number of digits after decimal point** Number of digits after decimal point for the parameters shown below. Enables to be valid in displaying on the faceplate or monitor tool. Parameters which enable N to be valid： ••• PV, SV, RH, RL, PH, PL, HH, LL, SH, SL	0 to 4	1
RH	REAL	**PV engineering variable high limit** High limit value for using A/D conversion value input from an analog input module as PV engineering variable. (Example) When using PV engineering variable of 0 to 200 °C → Set RH to "200". RH corresponds to IN_NMAX (high limit of analog input).	-999999 to 999999	100.0
RL	REAL	**PV engineering variable low limit** Low limit value for using A/D conversion value input from an analog input module as PV engineering variable. (Example) When using PV engineering variable of 0 to 200 °C → Set RL to "0". RL corresponds to IN_NMIN (low limit of analog input)	-999999 to 999999	0.0
PH	REAL	**PV high limit alarm value** Reference value of high limit exceeding alarm for PV engineering variable which is converted with RL, RH. When PV engineering variable is greater than this value, the PV high limit alarm (PHA) occurs.	RL to RH	100.0
PL	REAL	**PV low limit alarm value** Reference value of low limit exceeding alarm for PV engineering variable which is converted with RL, RH. When PV engineering variable is smaller than this value, PV low limit alarm (PLA) occurs.	RL to RH	0.0
HH	REAL	**PV high high limit alarm value** Reference value of high high limit exceeding alarm for PV engineering variable which is converted with RL, RH. When PV engineering variable is greater than this value, PV high high limit alarm (HHA) occurs	RL to RH	100.0
LL	REAL	**PV low low limit alarm value** Reference value of low low limit exceeding alarm for PV engineering variable which is converted with RL, RH. When PV engineering variable is smaller than this value, the input low low limit alarm (LLA) occurs.	RL to RH	0.0
HS	REAL	**PV high/low limit alarm hysteresis** Hysteresis range for alarm recovery after the occurrence of PV high limit, high high limit, low limit or low low limit exceeding alarm. Set with percentage (0 to 100%) against (RH - RL) as hysteresis range. After the occurrence of PV high limit alarm, PV ≦ (PH - HS) recovers PV high limit alarm, after the occurrence of PV high high limit alarm, PV ≦ (HH - HS) recovers PV high high limit alarm, after the occurrence of PV low limit alarm, PV ≧ (PL + HS) recovers PV low limit alarm, after the occurrence of PV low low limit alarm, PV ≧ (LL + HS) recovers PV low low limit alarm.	0 to 100 (Unit: %)	0.0
PID2H_SVLMT_EN	BOOL	**SV high/low limiter** Set Enable/Disable of SV high/low limiter processing. TRUE: Execute, FALSE: Not execute	TRUE, FALSE	FALSE
SH	REAL	**SV high limit value** • SV high/low limiter processing execution (PID2H_SVLMT_EN: TRUE) Executes limiter processing to setting value (current) SV1C not to exceed SH. • SV high/low limiter processing stop (PID2H_SVLMT_EN: FALSE) Limiter processing explained above is not executed.	RL to RH	100.0
SL	REAL	**SV low limit value** • SV high/low limiter processing execution (PID2H_SVLMT_EN: TRUE) Executes limiter processing to setting value (current) SVC not to drop below SL. • SV high/low limiter processing stop (PID2H_SVLMT_EN: FALSE) Limiter processing explained above is not executed.	RL to RH	0.0

(2) P, I, D constant, rate-of-change alarm, deviation alarm

Item (variable name)	Data type	Contents	Setting range	Initial value
P	REAL	**Proportional gain** When it is 0, proportioning, integral and derivative controls are not executed.	0 to 99	1.00
I	REAL	**Integral time** Integral control is not executed if the integral time is 0.	0 to 9999 (Unit: Sec.)	10.0
D	REAL	**Derivative time** Derivative control is not executed if the derivative time is 0.	0 to 9999 (Unit: Sec.)	0.0
PID2H_MTD	REAL	**Derivative gain** Derivative gain is a constant to determine the characteristics of imperfect derivative. The number is normally needless to change (change only when imperfect derivative characteristics should be adjusted strictly).	0 to 9999	8.0
DVL	REAL	**Deviation limit value** Allowable variation range. Set with percentage (0 to 100%) against (RH - RL) as the variation range of the deviation. Although large deviation alarm (DVLA) occurs when |Deviation| > Deviation limit value is established, deviation value limit will not be executed.	0 to 100 (Unit: %)	100.0
PID2H_DVLS	REAL	**Large deviation alarm hysteresis** Hysteresis range for recovering alarm after large deviation alarm (DVLA) occurred. Set with percentage (0 to 100%) against (RH - RL) as hysteresis range Recovered when |Deviation| ≦ (DVL - DVLS) is established after the occurrence of large deviation alarm.	0 to 100 (Unit: %)	2.0
CTIM	REAL	**PV rate-of-change alarm check time** The time to check a rate-of-change alarm of PV. The rate-of-change alarm is checked within this period. Set a period with value which is the execution cycle ΔT (execution cycle in unit of FBD program) multiplied by an integral number.	0 to 9999 (Unit: Sec.)	0.00
DPL	REAL	**PV rate-of-change alarm value** PV variation range for checking the rate-of-change alarm value of PV. Set it with percentage (0 to 100%) against (RH - RL). Positive rate-of-change alarm (DPPA) occurs with (PV current value – Previous PV before execution cycle ΔT) ≧ DPL, negative rate-of-change alarm (DPNA) occurs with (PV current value - Previous PV before execution cycle ΔT) ≦ -DPL. After the occurrence of the alarm (PV current value - Previous PV before execution cycle ΔT) < DPL recovers positive rate-of-change alarm (DPPA), (PV current value - Previous PV before execution cycle ΔT) > -DPL recovers negative rate-of-change alarm (DPNA). The PV will not be restricted even when the rate-of-change alarm occurs.	0 to 100 (Unit: %)	100.0

Item (variable name)	Data type	Contents	Setting range	Initial value
DSVL	REAL	**SV rate-of-change high limit value** Set the limit value of SV rate-of-change to prevent the rapid variation of SV.	0 to 100 (Unit: %)	100.0
ALPHA	REAL	**PV filter coefficient** Set filter coefficient of digital filter (index filter) processing to A/D conversion value (value after processing of input high/low limiter). When using first order lag filter, set PV filter coefficient to 0 and set in the parameters LLAG_EN、LLAG_T1 which are explained later.	0 to 1	0.20
CT	REAL	**Control cycle** Indicate PID operation cycle and set the time (second) that is the integral number multiple of execution cycle ΔT (The default is 200ms in the execution cycle of FBD program). 2-degree-of-freedom advanced PID control FB (M_2PIDH_) is consisted of 4 processing (analog input P_IN, high/low limit alarm check P_PHPL, 2-degree-of-freedom advanced PID control P_2PIDH_, output processing-3 with mode switching P_OUT3), executes P_IN, P_PHPL, P_OUT3 per execution cycle ΔT, P_2PIDH_ per control cycle CT.	0 to 9999 (Unit: Sec.)	1.00
DML	REAL	**Output rate-of-change high limit value** MV allowable variation range as output rate-of-change high limit value. Set it with percentage (0 to 100%) that is to MV (%). MV variation range is checked in every execution cycle ΔT. When |MV variation range| > DML, or |MV variation range| > - DML, output rate-of-change limit alarm (DMLA) occurred and MV variation range is limited by DML. (After ΔT, previous MV + DML = Current MV is established). This enables to convert MV into ramp status when SV is rapidly changed and not to output rapidly changed manipulated variable. Output rate-of-change limit alarm (DMLA) recovers when |MV variation range| ≦ DML.	0 to 100 (Unit: %)	100.0
MV	REAL	**Initial MV** Set the initial MV.	0 to 100 (Unit: %)	100.0
MH	REAL	**MV high limit value** High limit value for MV high/low limiter processing. When MV after output rate-of-change limit > MH, output high limit alarm (MHA) occurs and the MV is limited by the MH (output high limiter). Outputs high limit alarm (MHA) recovers when MV ≦ MH. When the control mode is MAN, CMV, the processing shown above is not executed.	-10 to 110 (Unit: %)	100.0

Item (variable name)	Data type	Contents	Setting range	Initial value
ML	REAL	**MV low limit value** Low limit value for MV high/low limiter processing. When MV after output rate-of-change limit < ML, output low limit alarm (MLA) occurs and the MV is limited by the MV low limit value (output low limiter). Output low limit alarm (MLA) recovers when MV ≧ ML. When the control mode is MAN, CMV, the processing shown above is not executed.	-10 to 110 (Unit: %)	0.0
GW	REAL	**Gap range** Gap range (0 to 100%) when executing PID control with gap (a control for reducing deviation used in PID operation by increasing the gap range to actual deviation). Set the actual deviation (0 to 100%) for executing PID control with gap PID control with gap is executed if |Actual deviation| ≦ Gap range.	0 to 100 (Unit: %)	0.0
GG	REAL	**Gap gain** Gap gain when executing PID control with gap. Set the gain in relation to the actual deviation (0 to 100%) for executing PID control with gap. Actual deviation x GG is the deviation used in PID operation.	0 to 99	1.0
ALPHA2	REAL	**2-degree-of-freedom parameter Alpha** Set the value of feed forward proportional. If α is tuned up, the proportion effect on setting value changing will become smaller.	0 to 1	0.0
BETA2	REAL	**2-degree-of-freedom parameter Beta** Set the value of feed forward derivative. If β is tuned down, the derivative effect on setting value changing will become bigger.	0 to 1	1.0
SVHI	BOOL	**SV high limit alarm** TRUE : Disable SV high limit alarm detection FALSE: Enable SV high limit alarm detection	TRUE, FALSE	FALSE
SVLI	BOOL	**SV low limit alarm** TRUE : Disable SV low limit alarm detection FALSE: Enable SV low limit alarm detection	TRUE, FALSE	FALSE
DSVLI	BOOL	**SV rate-of-change limit alarm** TRUE : Disable SV rate-of-change limit alarm detection FALSE: Enable SV rate-of-change limit alarm detection	TRUE, FALSE	FALSE
SVLL	BOOL	**SV low limit alarm level** TRUE : SV low limit alarm is major alarm. FALSE: SV low limit alarm is minor alarm.	TRUE, FALSE	FALSE
SVHL	BOOL	**SV high limit alarm level** TRUE : SV high limit alarm is major alarm. FALSE: SV high limit alarm is minor alarm.	TRUE, FALSE	FALSE
DSVLL	BOOL	**SV rate-of-change limit alarm level** TRUE : SV rate-of-change limit alarm is major alarm. FALSE: SV rate-of-change limit alarm is minor alarm.	TRUE, FALSE	FALSE

(3) Cascade connection, tracking

Item (variable name)	Data type	Contents	Setting range	Initial value
PID2H_TRK	INT	**Tracking flag** When executing cascade connection, set whether or not to execute tracking from the secondary loop to the primary loop • When executing tracking, set a parameter PID_TRK of secondary loop to 1. • When not executing tracking, set a parameter PID_TRK of primary loop to 0.	0, 1	0
PID2H_SVPTN_B0	BOOL	**Cascade connection** Set the cascade connection status. • When executing the cascade connection, set a parameter PID_SVPTN_B0 of secondary loop to FALSE. • When not executing the cascade connection, set a parameter PID_SVPTN_B0 of secondary loop to TRUE.	TRUE, FALSE	TRUE
PID2H_SVPTN_B1	BOOL	**The primary loop cascade output use status of secondary loop SV** Set to connect to cascade output of primary loop tag FB as SV of secondary loop. • When a primary loop executes the cascade connection with loop tag FB (such as M_2PIDH_), set a secondary loop parameter PID_SVPTN_B1 to FALSE. • When primary loop is not loop tag FB or tag FB which cannot execute the cascade connection, set TRUE.	TRUE, FALSE	TRUE

(4) Change-forbidden mode/operation, alarm detection status, alarm importance (level) settings

Item (variable name)	Data type	Contents	Setting range	Initial value
MANI	BOOL	**Change to MANUAL mode** TRUE : Change to MANUAL mode is forbidden. FALSE: Change to MANUAL mode is permitted.	TRUE, FALSE	FALSE
AUTI	BOOL	**Change to AUTO mode** TRUE : Change to AUTO mode is forbidden. FALSE: Change to AUTO mode is permitted.	TRUE, FALSE	FALSE
CASI	BOOL	**Change to CASCADE mode** TRUE : Change to CASCADE mode is forbidden. FALSE: Change to CASCADE mode is permitted.	TRUE, FALSE	FALSE
CMVI	BOOL	**Change to COMPUTER MV mode (mode of manual operation with upper computer)** TRUE : Change to COMPUTER MV mode is forbidden. FALSE: Change to COMPUTER MV mode is permitted.	TRUE, FALSE	FALSE
CSVI	BOOL	**Change to COMPUTER SV mode (mode of automatic operation with upper computer)** TRUE : Change to COMPUTER SV mode is forbidden. FALSE: Change to COMPUTER SV mode is permitted.	TRUE, FALSE	FALSE
CASDRI	BOOL	**Change to CASCADE DIRECT mode** TRUE : Move to CASCADE DIRECT mode is forbidden. FALSE: Move to CASCADE DIRECT mode is permitted.	TRUE, FALSE	FALSE
TSTPI	BOOL	**Change to TAG STOP mode** TRUE : Move to TAG STOP mode is forbidden. FALSE: Move to TAG STOP mode is permitted.	TRUE, FALSE	FALSE
ATI	BOOL	**Change to AUTO TUNING mode** TRUE : Execution of auto tuning is forbidden. FALSE: Execution of auto tuning is permitted.	TRUE, FALSE	FALSE
OVRI	BOOL	**Change to OVERRIDE mode** TRUE : Change to OVERRIDE mode is forbidden. FALSE: Change to OVERRIDE mode is permitted.	TRUE, FALSE	FALSE
SIMI	BOOL	**Change to SIMULATION mode** TRUE : Change to SIMULATION mode is forbidden. FALSE: Change to SIMULATION mode is permitted.	TRUE, FALSE	FALSE
MLI	BOOL	**MV output low limit alarm** TRUE : Disable MV output low limit alarm detection. FALSE: Enable MV output low limit alarm detection.	TRUE, FALSE	FALSE
MHI	BOOL	**MV output high limit alarm** TRUE : Disable MV output high limit alarm detection. FALSE: Enable MV output high limit alarm detection.	TRUE, FALSE	FALSE
DVLI	BOOL	**Large deviation alarm** TRUE : Disable large deviation alarm detection. FALSE: Enable large deviation alarm detection.	TRUE, FALSE	FALSE

Item (variable name)	Data type	Contents	Setting range	Initial value
DPNI	BOOL	**PV negative rate-of-change alarm (DPNA)** TRUE : Disable PV negative rate-of-change alarm detection FALSE: Enable PV negative rate-of-change alarm detection	TRUE, FALSE	FALSE
DPPI	BOOL	**PV positive rate-of-change alarm (DPPA)** TRUE : Disable PV positive rate-of-change alarm detection FALSE: Enable PV positive rate-of-change alarm detection	TRUE, FALSE	FALSE
PLI	BOOL	**PV input low limit alarm** TRUE : Disable PV input low limit alarm detection FALSE: Enable PV input low limit alarm detection	TRUE, FALSE	FALSE
PHI	BOOL	**PV input high limit alarm** TRUE : Disable PV input high limit alarm detection FALSE: Enable PV input high limit alarm detection	TRUE, FALSE	FALSE
LLI	BOOL	**PV input low low limit alarm** TRUE : Disable PV low low limit alarm detection FALSE: Enable PV low low limit alarm detection	TRUE, FALSE	FALSE
HHI	BOOL	**PV input high high limit alarm** TRUE : Disable PV input high high limit alarm detection FALSE: Enable PV input high high limit alarm detection	TRUE, FALSE	FALSE
SEI	BOOL	**Sensor error alarm (SEA)** TRUE : Disable sensor error alarm detection FALSE: Enable sensor error alarm detection	TRUE, FALSE	FALSE
DMLI	BOOL	**MV output rate-of-change limit alarm (DMLA)** TRUE : Disable MV output rate-of-change limit alarm detection FALSE: Enable MV output rate-of-change limit alarm detection	TRUE, FALSE	FALSE
ERRI	BOOL	**All alarms** TRUE : Disable all alarms detection FALSE: Enable all alarms detection	TRUE, FALSE	FALSE
MLL	BOOL	**MV output low limit alarm (MLA) level** TRUE : MV output low limit alarm is major alarm FALSE: MV output low limit alarm is minor alarm	TRUE, FALSE	FALSE
MHL	BOOL	**MV output high limit alarm (MHA) level** TRUE : MV output high limit alarm is major alarm FALSE: MV output high limit alarm is minor alarm	TRUE, FALSE	FALSE
DVLL	BOOL	**Large deviation alarm (DVLA) level** TRUE : large deviation alarm is major alarm FALSE: large deviation alarm is minor alarm	TRUE, FALSE	FALSE
DPNL	BOOL	**PV negative rate-of-change alarm (DPNA) level** TRUE : PV negative rate-of-change alarm is major alarm FALSE: PV negative rate-of-change alarm is minor alarm	TRUE, FALSE	FALSE
DPPL	BOOL	**PV positive rate-of-change alarm (DPPA) level** TRUE : PV positive rate-of-change alarm is major alarm FALSE: PV positive rate-of-change alarm is minor alarm.	TRUE, FALSE	FALSE

Item (variable name)	Data type	Contents	Setting range	Initial value
PLL	BOOL	**PV input low limit alarm (PLA) level** TRUE : PV input low limit alarm is major alarm. FALSE: PV input low limit alarm is minor alarm.	TRUE, FALSE	FALSE
PHL	BOOL	**PV input high limit alarm (PHA) level** TRUE : PV input high limit alarm is major alarm. FALSE: PV input high limit alarm is minor alarm.	TRUE, FALSE	FALSE
LLL	BOOL	**PV input low low limit alarm (LLA) level** TRUE : PV input low low limit alarm is major alarm. FALSE: PV input low low limit alarm is minor alarm.	TRUE, FALSE	FALSE
HHL	BOOL	**PV input high high limit alarm (HHA) level** TRUE : PV input high high limit alarm is major alarm. FALSE: PV input high high limit alarm is minor alarm.	TRUE, FALSE	FALSE
SENL	BOOL	**Sensor error alarm (SEA) level** TRUE : sensor error alarm is major alarm. FALSE: sensor error alarm is minor alarm.	TRUE, FALSE	FALSE
DMLL	BOOL	**MV output rate-of-change limit alarm (DMLA) level** TRUE : MV output rate-of-change limit alarm is major alarm. FALSE: MV output rate-of-change limit alarm is minor alarm.	TRUE, FALSE	FALSE
SPL	BOOL	**Stop alarm (SPA) level** TRUE : Stop alarm is major alarm. FALSE: Stop alarm is minor alarm.	TRUE, FALSE	FALSE

---

(5) Parameter settings when using temperature/pressure correction

Item (variable name)	Data type	Contents	Setting range	Initial value
TPC_PVTEMP	REAL	**Measured temperature** Input measured temperature (engineering variable) for temperature/pressure correction.	-999999 to 999999	0.0
TPC_PVPRES	REAL	**Measured pressure** Input measured pressure (engineering variable) for temperature/pressure correction.	-999999 to 999999	0.0
TPC_SQR	INT	**Temperature/pressure correction pattern** 0: None, 1: Square root extraction, 2: Temperature correction + square root extraction, 3: Pressure correction + square root extraction, 4: Temperature/pressure correction + square root extraction	0 to 4	0
TPC_TEMP	REAL	**Design temperature** Set the temperature specified in the design specification. Use the same unit as measured temperature.	-999999 to 999999	0.0
TPC_B1	REAL	**Bias temperature** Set the bias temperature to perform the correction calculation with absolute temperature. Set to "273.15" when Celsius is used for the design temperature and measured temperature.	-999999 to 999999	273.15
TPC_PRES	REAL	**Design pressure** Set the pressure specified in the design specification. Use the same unit as measured pressure.	-999999 to 999999	0.0
TPC_B2	INT	**Bias pressure** Set the bias pressure to perform the correction calculation with absolute pressure. Set to "101.3" when kilo Pascal (kPa) is used for the design pressure and measured pressure.	-999999 to 999999	10332.0
SQR_K	REAL	**Square root extraction: coefficient** Set a coefficient of Square root extraction (Set to 10.0 when executing operation with percentage (%).)	0 to 999999	10.0
SQR_OLC	REAL	**Square root extraction: output low cut-off value** Output is cut off when the value becomes unstable due to small input value. Normally set to around 10.0.	0 to 999999	0.0
SQR_DENSITY	REAL	**Square root extraction: density correction value** Input a value of (design density / measured density).	0 to 999999	1.0

(6) Parameter settings when using broken line and other compensation

Item (variable name)	Data type	Contents	Setting range	Initial value
FG_SN	INT	**Input variable correction: number of points** Set the total number of points used in the broken line correction processing.	0 to 48	0.0
FG_X1 to FG_X48	REAL	**Input variable correction: input coordinates** Set the input coordinates (X-coordinates) of the broken line correction processing.	-999999 to 999999	0.0
FG_Y1 to FG_Y48	REAL	**output variable correction: output coordinates** Set the output coordinates (Y-coordinates) of the broken line correction processing.	-999999 to 999999	0.0
LLAG_EN	BOOL	**Enable first order lag filter** Set the Enable/Disable setting of first order lag filter to an input variable. TRUE: Enable, FALSE: Disable	TRUE, FALSE	FALSE
LLAG_T1	REAL	**First order lag filter time** Set the lag time (second) of the first order lag filter to an input variable.	0 to 999999	1.0
PVCMP_EN	BOOL	**PV compensation execution condition** Set the enable/disable setting of PV compensation processing. Substitute compensation value to input variable PV_CMPIN. TRUE: Execute, FALSE: Stop	TRUE, FALSE	FALSE
PVCMP_MODE	INT	**PV compensation mode** Select either addition or replacement of compensation value which is substituted to input variable PV_CMPIN. 0: Addition, 1: Replacement	0 to 1	0
PVDCMP_EN	BOOL	**ΔPV compensation execution condition** Set execute/stop of compensation to PV by the external compensation value difference (ΔPV). Substitute compensation value to input variable PVD_CMPIN. TRUE: Execute, FALSE: Stop	TRUE, FALSE	FALSE
PID2H_PVTRK_EN	BOOL	**PV tracking execution condition** Set execute/stop of PV tracking function that matches SV and PV to prevent sharp changes of MV in control mode switching MANUAL → AUTO TRUE: Execute, FALSE: Stop	TRUE, FALSE	FALSE
PID2H_ISTP	BOOL	**Integration stop signal** Stop the operation of integral element. TRUE: Integration stop, FALSE: No processing	TRUE, FALSE	FALSE
PID2H_DSTP	BOOL	**Derivation stop signal** Stop the operation of derivative element. TRUE: Derivation stop, FALSE: No processing	TRUE, FALSE	FALSE
PID2H_LMT_ISTP	BOOL	**When MV rate-of-change limiter alarm occurred, select stop integration.** Stop the operation of integral element when MV rate-of-change limiter alarm occurs and the integral component is the same with the limit direction. TRUE: Stop, FALSE: Not stop	TRUE, FALSE	FALSE
MVDCMP_EN	BOOL	**ΔMV compensation execution condition** Set enable/disable compensation to velocity type output ΔMV of velocity type PID operation. Substitute velocity type compensation value to input variable MVD_CMPIN. TRUE: Execute, FALSE: Stop	TRUE, FALSE	FALSE
MVDCMP_MODE	INT	**ΔMV compensation mode** Select either addition or replacement of ΔMV compensation value. Substitute ΔMV compensation value to input variable MVD_CMPIN. 0: Addition, 1: Replacement	TRUE, FALSE	FALSE
MVDGAINCMP_EN	BOOL	**ΔMV gain correction execution condition** Set execute/stop of adjustment to velocity type output ΔMV of velocity type PID operation. Substitute gain correction value to input variable MVD_GAININ. TRUE: Execute, FALSE: Stop	TRUE, FALSE	FALSE
OUT3_MVCMP_EN	BOOL	**MV compensation execution condition** Set execute/stop of adjustment to position type output MV of velocity type PID operation. Substitute compensation value to input variable MV_CMPIN. TRUE: Execute, FALSE: Stop	TRUE, FALSE	FALSE
OUT3_MVCMP_MOD	INT	**MV compensation mode** Select either addition or replacement of compensation value which is substituted to input variable MV_CMPIN. Substitute compensation value to input variable MV_CMPIN. 0: Addition, 1: Replacement	0 to 1	0
OUT3_PREMV_EN	BOOL	**Preset MV execution condition** Set execute/stop of MV output preset MV switch. Switch to value which is substituted to preset MV output OUT3_PREMV_V. TRUE: Execute, FALSE: Stop	TRUE, FALSE	FALSE
OUT3_PREMV_V	REAL	**Preset MV** Substitute preset MV. (Unit: %)	0 to 100	0.0

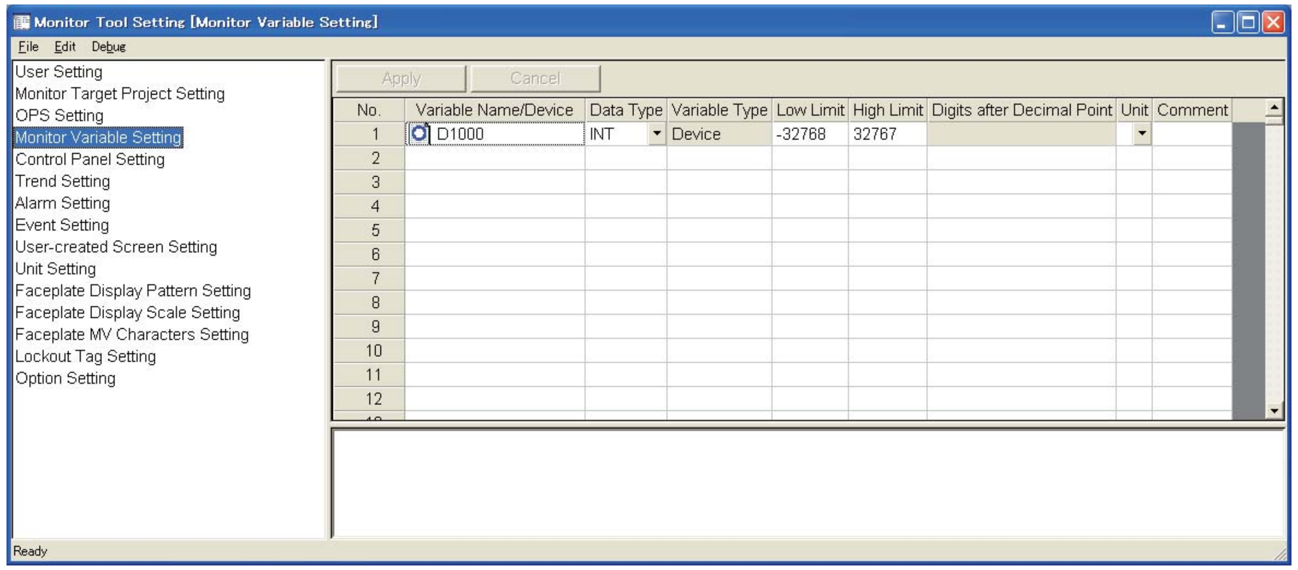
Item (variable name)	Data type	Contents	Setting range	Initial value
OUT3_MVHLD_EN	BOOL	**MV hold execution condition** Set execute/stop to hold MV When TRUE, hold MV in switching FALSE → TRUE. TRUE: Execute, FALSE: Stop	TRUE, FALSE	FALSE
OUT3_MVTRK_EN	BOOL	**MV tracking execution condition** Set execute/stop of MV output tracking input switch. Switch to a value which is substituted to MV tracking input OUT3_MV_TRKIN. TRUE: Execute, FALSE: Stop	TRUE, FALSE	FALSE
OUT3_STP_OTYPE	INT	**Output selection when loop stop/tag stop is executed** Select MV output in tag stop by stop alarm (SPA) or loop stop. 0: Hold, 1: Preset value	0 to 1	0
OUT3_SEA_OTYPE	INT	**MV output selection when sensor error occurs** Set a method for MV output in the occurrence of sensor alarm (SEA). 0: Hold, 1: Preset MV output, 2: Do not hold nor output preset MV	0 to 2	0
OUT3_ARW_EX_EN	BOOL	**Pull MV internal operation value back, when it exceeds MV internal operation high/low limit value** Set execute/stop of pull back to proportion other than integration, derivation by countermeasures against reset windupreset windup. TRUE: Execute, FALSE: Not execute	TRUE, FALSE	FALSE
OUT3_MVPH	REAL	**MV internal operation high limit value** Pull MV internal operation high limit value back, when it exceeds MV internal operation high/low limit value OUT3_ARW_EX_EN is TRUE, MV internal operation value exceeds MV internal operation high/low limit value. (Unit: %)	MH to 999999	100.0
OUT3_MVPL	REAL	**MV internal operation low limit value** Pull MV internal operation low limit value back, when it exceeds MV internal operation high/low limit value OUT3_ARW_EX_EN is TRUE, MV internal operation value exceeds MV internal operation high/low limit value. (Unit :%)	-999999 to ML	0.0
OUT3_MVREV_EN	BOOL	**MV reverse execution condition** Set Execute/Stop of MV inverse processing (100-MV). TRUE: Execute, FALSE: Stop	TRUE, FALSE	FALSE

Appendix 2 Method to Show Data Other Than Tag Data in PX Developer Monitor

Here is an explanation about how to show data other than Tag Data in the Trend Graph.

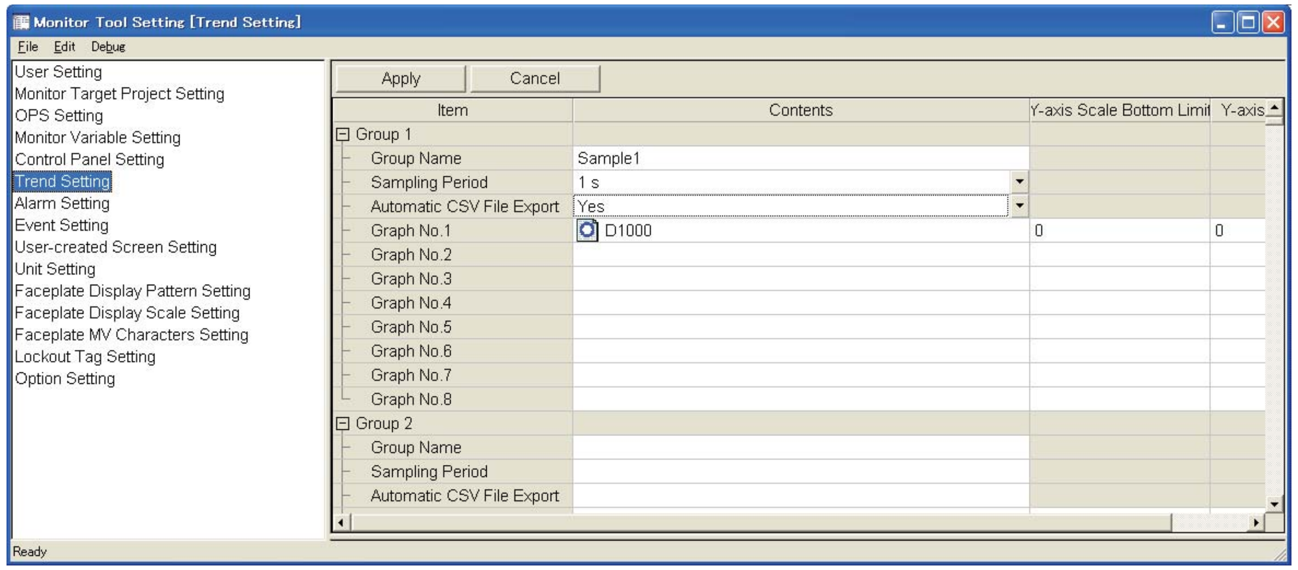
Example. **Display the value of D1000 as Integer type in PX Developer Monitor Tool**

Step1: Register D1000 as a Monitor Variable



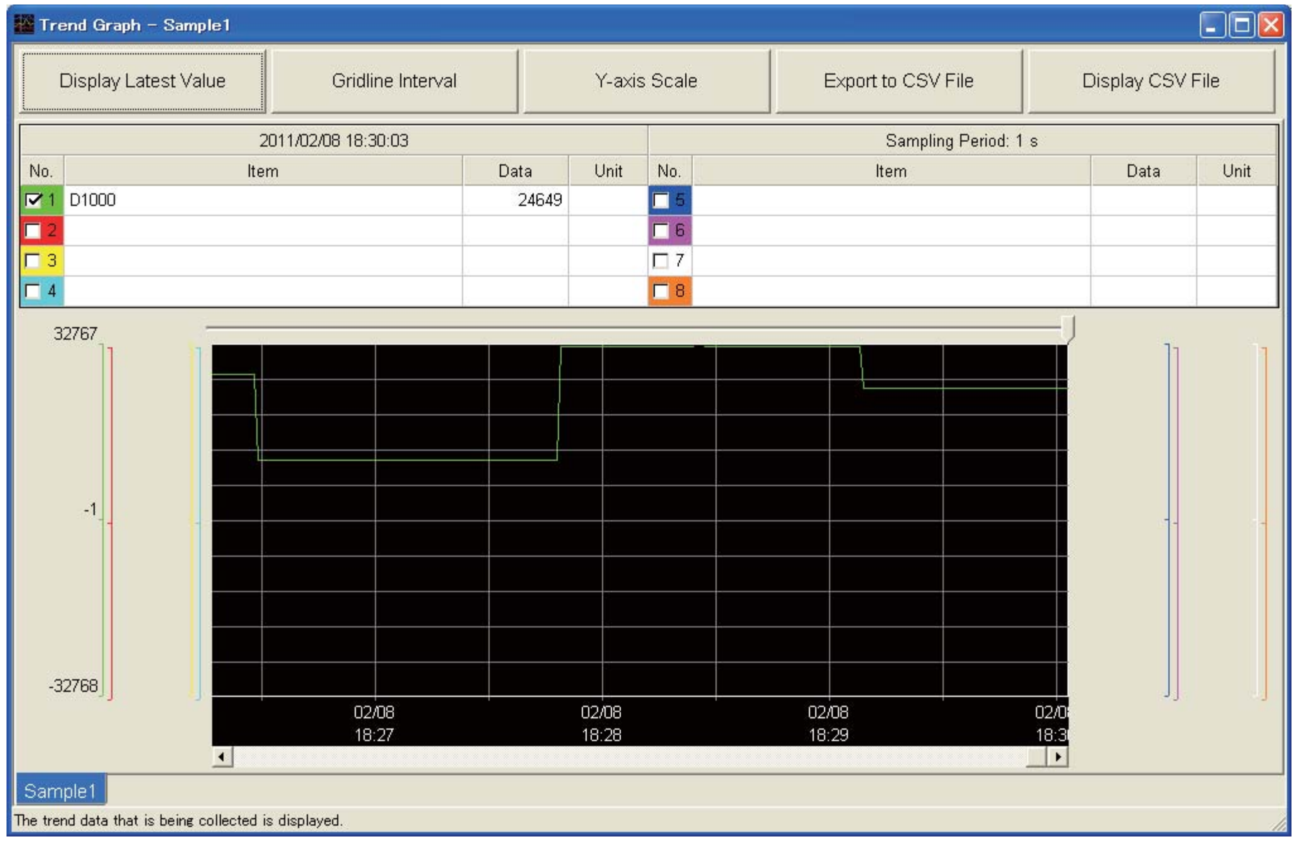
No.	Variable Name/Device	Data Type	Variable Type	Low Limit	High Limit	Digits after Decimal Point	Unit	Comment
1	D1000	INT	Device	-32768	32767			
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								

Step 2: Set the Monitor Variable in the Trend Setting.



Item	Contents	Y-axis Scale Bottom Limit	Y-axis
⊟ Group 1			
Group Name	Sample1		
Sampling Period	1 s		
Automatic CSV File Export	Yes		
Graph No.1	D1000	0	0
Graph No.2			
Graph No.3			
Graph No.4			
Graph No.5			
Graph No.6			
Graph No.7			
Graph No.8			
⊟ Group 2			
Group Name			
Sampling Period			
Automatic CSV File Export			

Step 3: Confirm if the the value set in the step 2 is displayed in the Trend Graph Window.

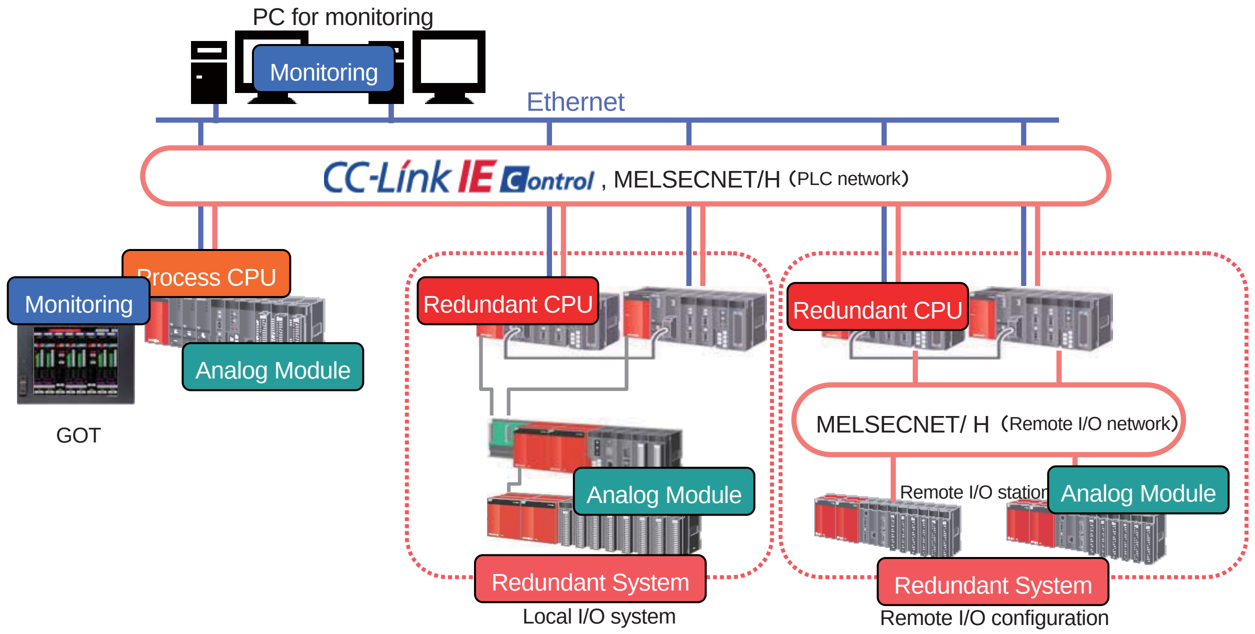


2011/02/08 18:30:03				Sampling Period: 1 s			
No.	Item	Data	Unit	No.	Item	Data	Unit
☑ 1	D1000	24649		☐ 5			
☐ 2				☐ 6			
☐ 3				☐ 7			
☐ 4				☐ 8			

The trend data that is being collected is displayed.

Appendix 3 MELSEC Process Control Product Selection Guide

This section explains the selection methods of process CPU/redundant CPU, redundant system, analog module, monitoring when programming the loop control with PX Developer.
A figure below is an example of process control system with MELSEC process control.



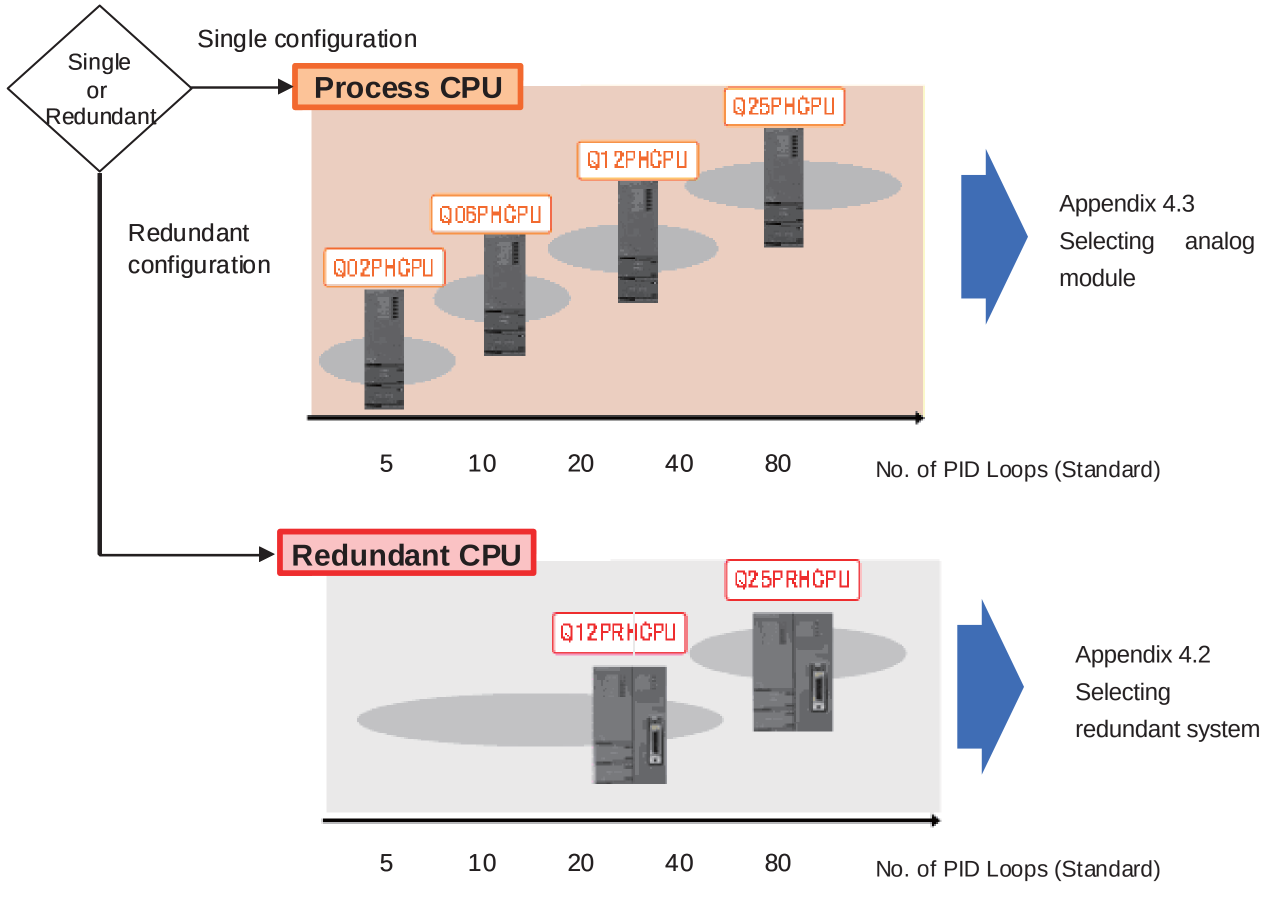
The following explains the selection methods.

Selection target	Description of selection method
Process CPU Redundant CPU	Appendix 3.1 Selecting Process CPU, Redundant CPU
Redundant System	Appendix 3.2 Selecting Redundant System
Analog Module	Appendix 3.3 Selecting Analog Module
Monitoring	Appendix 3.4 Selecting Monitor Operation

Appendix 3.1 Selecting Process CPU, Redundant CPU

- Select Process CPU or Redundant CPU in accordance with system configuration.
- Then, select[*1] a CPU type name with regarding the number of loops as standard.

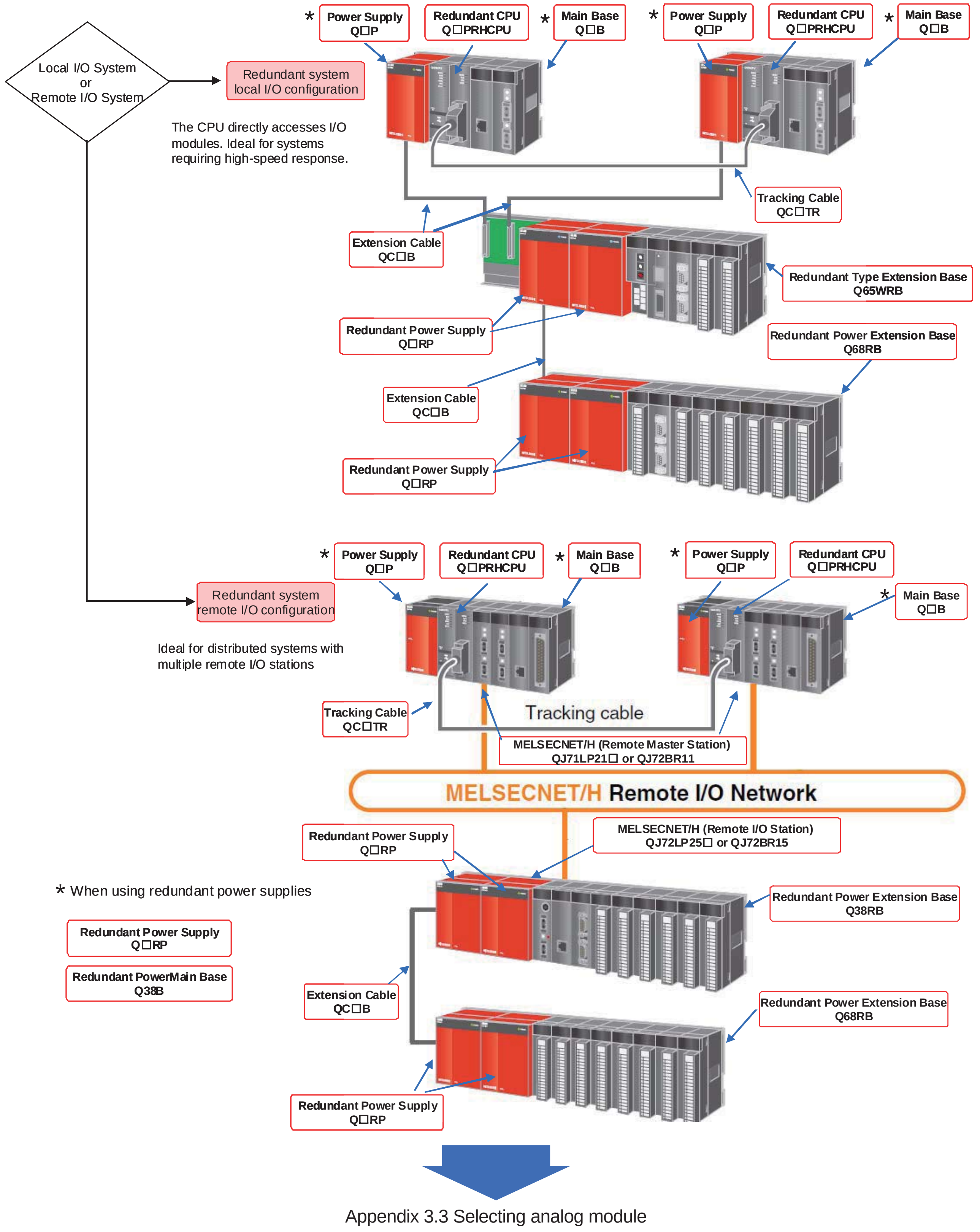
▨ Shown below describes the standard of corresponding loop number range.



*1 For selecting CPU type name, a method of calculating FBD program steps to be created can be used.
For details, refer to Appendix 3.5.

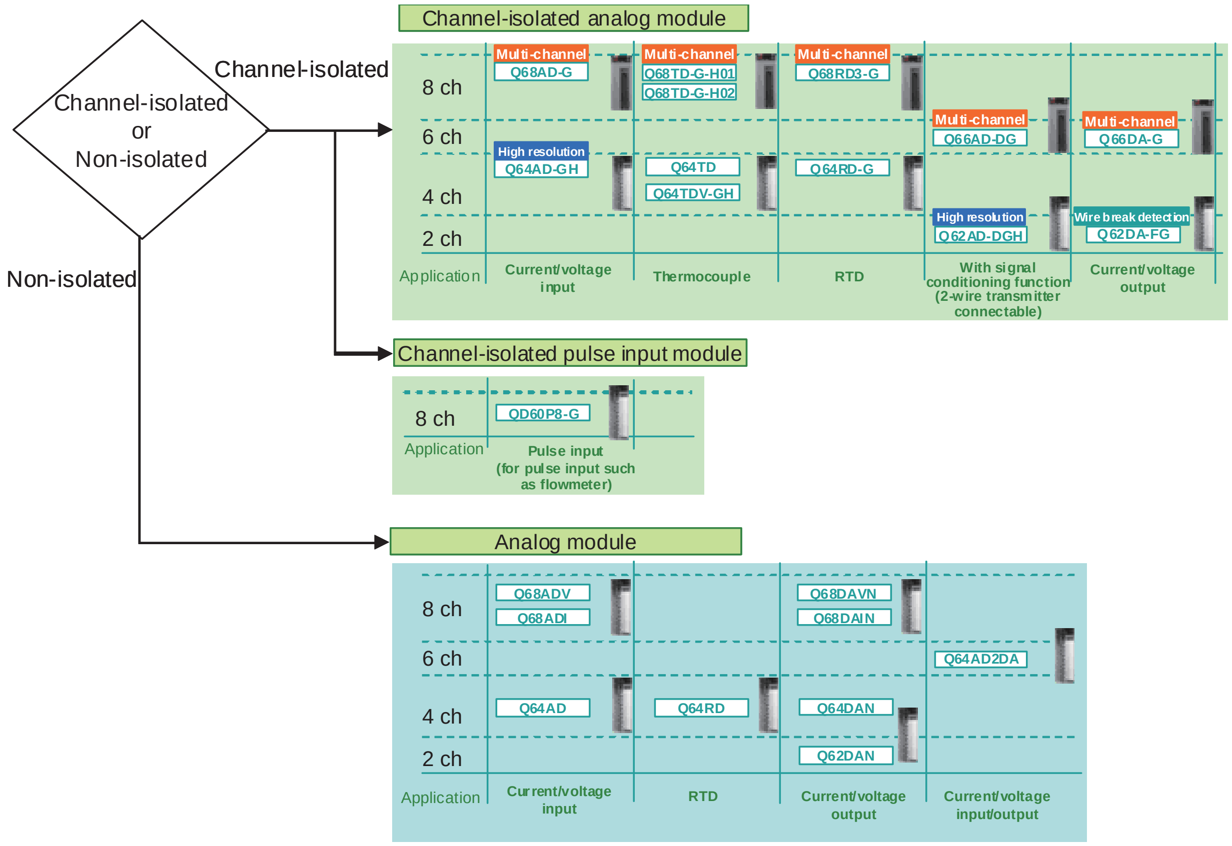
Appendix 3.2 Selecting redundant system

- Select from Local I/O system and remote I/O system.
- Select such as main base, power supply, cables which are in accordance with selected system.



Appendix 3.3 Selecting analog module

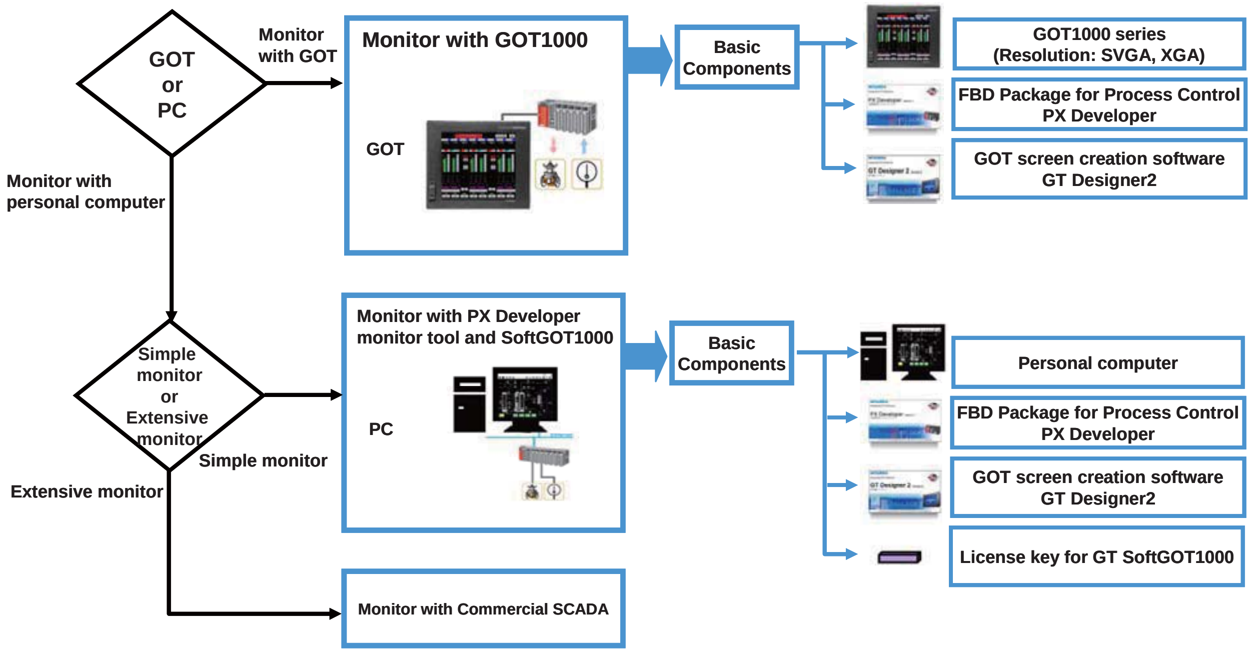
- Select units in accordance with input signal type or points.



Appendix 3.4 Selecting Monitor Operation

Appendix 3.4 Selecting Monitor Operation

- Select monitor operation in accordance with system scale or use.



Appendix 3.5 (Supplementation) Selecting with the number of FBD program steps of Process CPU, Redundant CPU

The following describes the rough selection of Process CPU, Redundant CPU by the number of FBD steps when programming with PX Developer.

The procedure of rough selection is following.
Procedure 1: Calculate rough steps of FBD program
Procedure 2: Rough selection of type of CPU with the rough number of steps

Procedure 1: Calculate rough steps of FBD program

Calculate rough steps with a formula as shown below.

$$\underbrace{\textbf{Rough number of FBD program steps}}_{②} = \underbrace{\textbf{Rough number of tag FB and module FB steps}}_{①} \times (2 \text{ to } 8)$$

① Rough step number calculation of tag FB and module FB

The rough number of steps of tag FB and module FB =
(Unit: Step)

$$\Sigma\left(\text{Rough number of steps per tag FB type} + (30 \times \text{number of same tag FBs})\right)$$

$$+\ \Sigma\left(\text{Rough number of steps per module FB type} + (30 \times \text{number of module FB})\right) + 1{,}000$$

The number of fixed steps for system management

<Rough number of steps per tag FB type>

[Loop tag FB]

No.	FB name	Rough number of steps
1	M_PID、M_PID_T、M_PID_DUTY、M_PID_DUTY_T	1,000
2	M_PIDP、M_PIDP_T、M_PIDP_EX_、M_PIDP_EX_T_	800
3	M_SPI、M_SPI_T	900
4	M_IPD、M_IPD_T	900
5	M_BPI、M_BPI_T	900
6	M_2PID、M_2PID_T、M_2PID_DUTY、M_2PID_DUTY_T	1,000
7	M_R、M_R_T	800
8	M_ONF2、M_ONF2_T	700
9	M_ONF3、M_ONF3_T	700
10	M_MONI	400
11	M_MWM	600
12	M_BC	500
13	M_PSUM	300
14	M_SEL、M_SEL_T1、M_SEL_T2	400
15	M_MOUT	300
16	M_PGS	400
17	M_PGS2_	2,000
18	M_2PIDH_、M_2PIDH_T_	5,000*
19	M_SWM_	1,500
20	M_PFC_SF_	2,800
21	M_PFC_SS_	3,300
22	M_PFC_INT_	2,700

（* PX Developer ver.1.21X or later）

[Alarm tag FB, Message tag FB]

No.	FB name	Rough number of steps
1	M_ALARM	200
2	M_MESSAGE	200

[Status tag FB]

No.	FB name	Rough number of steps
1	M_NREV	800
2	M_REV	900
3	M_MVAL1	800
4	M_MVAL2	900
5	M_TIMER1、M_TIMER2	500
6	M_COUNTER1、M_COUNTER2	500
7	M_PB_	600

< Rough number of steps per tag module FB type>

[Analog module FB]

No.	FB name	Corresponding module model name	Rough number of steps
1	AIN_4CH	Q64AD	400
2	AIN_8CH	Q68ADV/Q68ADI	400
3	AIN_2CH_DG	Q62AD-DGH	400
4	AIN_4CH_G	Q64AD-GH	500
5	AIN_6CH_DG	Q66AD-DG	700
6	AIN_8CH_G	Q68AD-G	800
7	AOUT_2CH	Q62DA/Q62DAN	400
8	AOUT_4CH	Q64DA/Q64DAN	400
9	AOUT_8CH	Q68DAV/Q68DAVN/Q68DAI/Q68DAIN	500
10	AOUT_2CH_G	Q62DA-FG	500
11	AOUT_6CH_G	Q66DA-G	600
12	AIN_4CH_AOUT_2CH	Q64AD2DA	1,700

[Temperature input module FB]

No.	FB name	Corresponding module model name	Rough number of steps
1	TC_4CH	Q64TD	400
2	TCV_4CH	Q64TDV-GH	500
3	TC_8CH_G	Q68TD-G/Q68TD-G-H01/ Q68TD-G-H02	500
4	RTD_4CH	Q64RD/Q64RD-G	400
5	RTD_8CH_G	Q68RD3-G	500

[Counter module FB]

No.	FB name	Corresponding module model name	Rough number of steps
1	HIC_2CH	QD62/QD62E/QD62D	500
2	PIN_8CH_G	QD60P8-G	1,900

[I/O module FB]

No.	FB name	Corresponding module model name	Rough number of steps
1	DIN_8PT	QX28	200
2	DIN_16PT	QX10/QX40/QX40-S1/QX70/QX80	200
3	DIN_32PT	QX41/QX41-S1/QX71/QX81	200
4	DIN_64PT	QX42/QX42-S1/QX72/QX82/QX82-S1	200
5	DOUT_8PT	QY18A/QY68A	200
6	DOUT_16PT	QY10/QY22/QY40P/QY50/QY70/QY80	200
7	DOUT_32PT	QY41P/QY71/QY81P	200
8	DOUT_64PT	QY42P	300
9	DINOUT_64PT	QH42P	200
10	DINOUT_15PT	QX48Y57	200

[CC-Link module FB]

No.	FB name	Rough number of steps
1	CCLINK_1	400
2	CCLINK_2	500
3	CCLINK_3	700
4	CCLINK_4	900

- **Calculation example**

Tag FB type	Number of tag FBs
2-degree-of-freedom advanced PID control (M_2PIDH_)	30 (Loop)
Monitor (Indicator) (M_MONI)	90 (Loop)
Motor irreversible, solenoid valve operation (M_NREV)	50 (Loop)

- Analog input 120 points
- Analog output 30 points

- Digital input 100 points + spare
- Digital output 100 points + spare

Module FB type	Number of module FBs
Isolated analog input Q64AD-GH (AIN_4CH_G)	30 (120 points)
Isolated analog output Q62DA-FG (AOUT_2CH_G)	15 (30 points)
Digital input QX41 (DIN_32PT)	5 (160 points)
Digital output QY41P (DOUT_32PT)	4 (128 points)

$$\textbf{Rough number of steps of tag FB and module FB (Unit: Step)} = \Sigma\left(\begin{matrix}\text{Rough number of steps}\\\text{per tag FB type}\end{matrix} + (30 \times \text{number of the same tag FB type})\right)$$

$$+\left(\begin{matrix}\text{Rough number of steps}\\\text{per module FB type}\end{matrix} + (30 \times \text{number of the same module FB type})\right) + 1,000$$

= (5,000 + (30 × 30 tag)) ← M_2PIDH_minutes
\+ (400 + (30 × 90 tag)) ← M_MONI minutes
\+ (800 + (30 × 50 tag)) ← M_NREV minutes
\+ (500 + (30 × 30 module)) ← AIN_4CH_G minutes
\+ (500 + (30 × 15 module)) ← AOUT_2CH_G minutes
\+ (200 + (30 × 5 module)) ← DIN_32PT minutes
\+ (200 + (30 × 4 module)) ← DOUT_32PT minutes
\+ 1,000

= 15,320 steps

② Calculation of FBD program rough steps

$$\textbf{Rough number of steps of FBD program} = \begin{matrix}\text{Rough number of steps of}\\\text{tag FB and module FB}\end{matrix} \times (2 \text{ to } 8)$$

= 15,320 steps ×(2 to 8)
= **30,640 to 122,560 steps**

2 to 8 times are applied by consideration of other programs with correction operation FBs, functions (such as logical operation, Arithmetic operation, Comparison, Selection) in addition to the programs of tag FB and module FB.

Procedure 2: Rough selection of type of CPU with rough number of steps

(1) Rough selection standard

- **Q02PHCPU**

[Rough number of steps of tag FB and module × 5] + Number of user-created ladder steps < Program capacity of Q02PHCPU (28K step) × 70% = **20K step**

5 times is applied from the average of 2 to 8 times mentioned in the previous page.

The other 30% is regarded as margin.
(It is for increases of future customization. Can be changed by users)

- **Q06PHCPU**

[Rough number of steps of tag FB and module × 5] + Number of user-created ladder steps < Program capacity of Q06PHCPU (60K step) × 70% = **42K step**

5 times is applied from the average of 2 to 8 times mentioned in the previous page.

The other 30% is regarded as margin.
(It is for increases of future customization. Can be changed by users)

- **Q12PHCPU, Q12PRHCPU**

[Rough number of steps of tag FB and module × 5] + Number of user-created ladder steps < Program capacity of Q12PHCPU (124K step) × 70% = **87K step**

5 times is applied from the average of 2 to 8 times mentioned in the previous page.

The other 30% is regarded as margin.
(It is for increases of future customization. Can be changed by users)

- **Q25PHCPU, Q25PRHCPU**

[Rough number of steps of tag FB and module × 5] + Number of user-created ladder steps < Program capacity of Q25PHCPU (252K steps) × 70% **＝176K steps**

5 times is applied from the average of 2 to 8 times mentioned in the previous page.

The other 30% is regarded as margin.
(It is for increases of future customization. Can be changed by users)

(2) Selection example

1) When 10K is set as the number of user-created ladder steps in the example of procedure 1

[13.32K × 5] + 10K

Number of User-created ladder steps

= 76.6K steps < **87K steps** (Rough selection standard for Q12PHCPU)

Therefore, select **Q12PHCPU as rough selection standard.**

2) When 50K is set as the number of user-created ladder steps in the example of procedure 1

[13.32K × 5] + 50K

Number of User-created ladder steps

= 116.6K steps < **176K steps** (Rough selection standard for Q25PHCPU)

Therefore, select **Q25PHCPU as rough selection standard.**

Appendix 4 Function Outline of Programming Tool/Monitor Tool

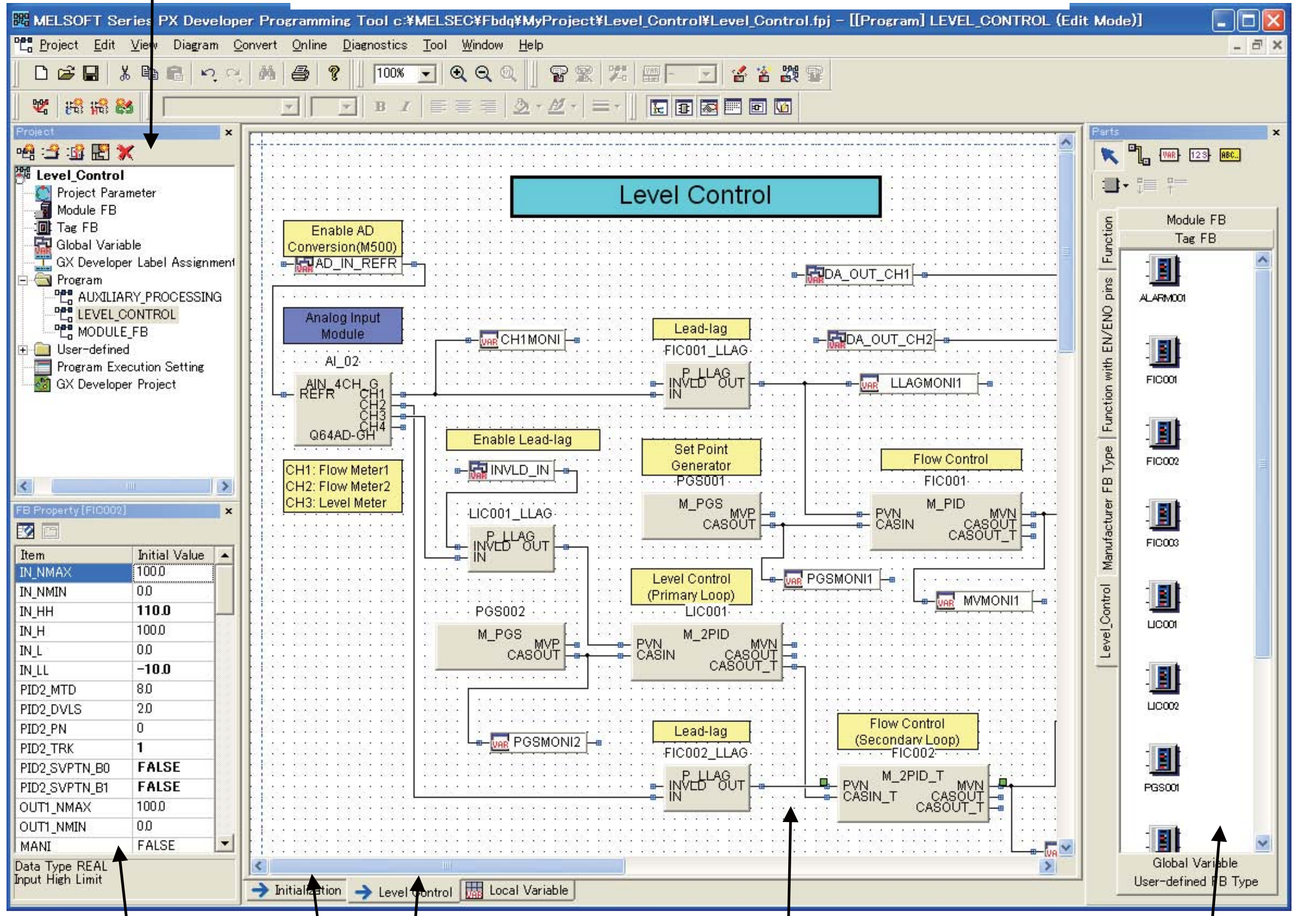
Appendix 4.1 List of programming tool functions

No.	Functions	Contents
1	Project creation	• Creates a new project • Saves a project with a new name • Opens a project • Closes a project • To set project parameters • To set permission for program and user-defined FB
2	Tag registration	• Registers names (tag name) of loop control tag, digital control tag (such as motor Irreversible/Motor Reversible tag, ON/OFF solenoid control tag) to be used in a program (Q12/25PHCPU, QnPRHCPU: maximum 480 tags, Q02/06PHCPU: maximum 120 tags)
3	I/O module registration	• Registers such as names of analog module/I/O module, module model name, head start address used in programming tool of PX Developer. (Only modules on main base or extension base can be registered)
4	Global variable registration	• Registers such as global variable name, data type used in a FBD program. (maximum 32,000)
5	GX Developer label assignment	• Registers to assign label names (global label) in GX Developer to a FBD program (global variable) in PX Developer to exchange data of a FBD program in PX Developer and that of a ladder program in GX Developer with labels. (maximum 5,000)
6	FBD program creation/edit	• Creates/edits FBD program, user-defined FB, user-defined tag FB by dragging/dropping FB/function from the parts window and then connecting them. • As the edit function, copies/ deletes a program, user-defined FB/user-defined tag FB and so on. • Creates/edits (such as copy/delete) a structure. • To set the initial values of FB parameters (such as P, I, D constants and high/low limit value) as FB properties. • Number of programs: maximum 200 • Number of sheets: maximum 32 sheets / program. Maximum paper size per sheet is A3 x 2. • The number of FBs, functions, variables/constant parts, comment parts, connectors to be attached per sheet is unlimited. • FB execution order can be displayed.
7	Program execution setting	• To set execution method/timing (such as Timer execution type, Interruption execution type, execution cycle, event execution condition) of each program.
8	Open a GX Developer project	• Opens a GX Developer project and then creates/edits a ladder program, to set the network parameters.
9	Compile	• Compiles a created FBD program or a ladder program with labels created in GX Developer, and then creates executable codes (ladder codes) for CPU.
10	Download	• Downloads compiled executable codes and symbolic data of FBD program to CPU.
11	Upload	• Reads complied executable codes and symbolic data of FBD program from CPU and restores the project.
12	Online monitor	• Monitoring variables (includes tag data) in real-time is possible. • Changing the contents of variables online is possible. • Faceplate display is possible for tag FB. • Reading current value of FB property in a batch /substitution for initial value of FB property is possible with FB property management.
13	Debug/diagnostics	• Possible to stop/restart each FB • When an error occurs in FBD program that is running, displaying error code/FB name in where the error occurred is possible
14	Cross reference	• Displays the list of the places where variables are declared and used. • Sort and filter display (display only the data that satisfies the specified condition) is possible.
15	Print	• Prints the registered/set data, FBD program, user-defined FB/user-defined tag FB.

Appendix 4.2 Screen configuration of programming tool

Project window

- To set such as project parameter, modules being used, tags, program names, program execution interval.



FB Property window

- To set initial values of data in FB.
- Displaying and changing the current values are possible in monitoring.

Sheets

- Maximum 32 sheets / program

Program/FB definition window

- The programs and user-defined FBs are created in this window.

↓

A program is a unit in which process is described. Up to 200 programs can be created. One program contains up to 32 sheets. A sheet is a form used to paste FB/functions or connector lines and to describe the process. The program execution methods, such as execution cycle, can be set in this window.

Parts window

- The FB/function parts which can be pasted to the programs and user-defined FBs are displayed in this window.

Appendix 4.3 List of monitor tool functions

<table>
<tr><th>No.</th><th>Functions</th><th>Contents</th></tr>
<tr><td>1</td><td>Number of monitoring CPU modules</td><td>• Number of CPUs (QnPH/QnPRH) under monitoring: maximum 16 CPUs*
(Maximum 8 CPUs can be monitored per personal computer.)
*For redundant CPU, a control system and a standby system are counted in set for 1 CPU.</td></tr>
<tr><td>2</td><td>Number of monitor tags</td><td>• Maximum 7,680 tags (Maximum 480 tags / CPU × 16 CPUs) can be monitored.</td></tr>
<tr><td>3</td><td>Control panel</td><td>• Displays a faceplate simulating a process control device, and monitors/adjusts (such as operation mode change, SV/MV change) loop control tags, digital control tags. The pop-up tuning screen can be opened from this screen.
• 8 faceplates/screen (one group) × maximum 500 screens = maximum 4,000 faceplates
• Other than general monitoring (AUTO/MANUAL/CASCADE), monitoring/operation can be executed in OVERRIDE mode/SIMULATION mode. When sensor failure/disconnection, substitution operation such as inputting specified PV manually to continue the operation can be executed in OVERRIDE mode. Simple check of loop motion and so on can be executed in SIMULATION mode by returning the MV to PV forcedly.
• Lockout tag is possible per faceplate unit.</td></tr>
<tr><td>4</td><td>Trend graph</td><td>• Displays time sequence change chart for tag data/global variable/device item values in historical/real time trend graph format.
• 8 items / screen (one group) × maximum 125 screens = maximum 1,000 items
• Collection cycle: 1 sec. /10 sec. /1 min. /5 min. /10 min.
• Recordable time:
<table>
<tr><td>Sampling period</td><td>1 sec.</td><td>10 sec.</td><td>1 min.</td><td>5 min.</td><td>10 min.</td></tr>
<tr><td>Recordable time</td><td>2.77 hours</td><td>27.7 hours</td><td>6.9 days</td><td>34.7 days</td><td>69.4 days</td></tr>
</table>
• The collected data can be output into a CSV file manually.
• Outputs the collected data in CSV format text file automatically.</td></tr>
<tr><td>5</td><td>Alarm list display</td><td>• Displays the list of maximum 2,000 alarm records (loop control tag alarm and alarm messages for alarm tag).
• Tag faceplate on which an alarm is displayed can be displayed.
• The list can be output into a CSV file manually.
• Output the alarm contents in CSV format text file automatically when automatic alarm CSV file export is set on Alarm/Event of the option settings.</td></tr>
<tr><td>6</td><td>Event list display</td><td>• Displays the list of maximum 2,000 event records (user operation history, event messages for each digital tag and message tag).
• The list can be output into a CSV file manually.
• Output the event contents in CSV format text file automatically when automatic event CSV file export is set on Alarm/Event of the option settings.</td></tr>
<tr><td>7</td><td>Start User application</td><td>• Starts an application program (such as graphic screen) which is created by users.
(Maximum 4 programs can be started)</td></tr>
<tr><td>8</td><td>Pop-up faceplate/ pop-up tuning screen</td><td>• Displays a faceplate in the form of a pop-up window.
• Clicking the [Details] button displays the pop-up tuning screen.
• The pop-up tuning screen is composed of faceplate, tuning trend graph, and tag monitor function.
• On the tuning trend graph screen, PV/MV/SV of tag (loop) are displayed one after another in the passing time sequence or with historical trend graph. The sampling period is 1 second and, up to 20,000 points (5.5 hours) can be recorded. It can be output in CSV format text file manually.
• The maximum number of tuning data collected with pop-up faceplate, tuning screen simultaneously is 16. Desired 2 tuning screens out of them can be displayed simultaneously.
• The contents of tag item are monitored online with its tag monitor. The contents of tag item can be changed online.
• Auto tuning operation can be executed from the tuning screen. (For loop tags which contains the auto tuning function only.)</td></tr>
<tr><td>9</td><td>Buzzer stop</td><td>• Stops alarm buzzer.</td></tr>
<tr><td>10</td><td>Print screen</td><td>• Hardcopies whole screen in the print display.</td></tr>
<tr><td>11</td><td>Screen align</td><td>• Overlap the diagrams in display or redisplay after vertical arrangement. (Up to 4 screens)</td></tr>
<tr><td>12</td><td>Find</td><td>• Finds tags, control panels, and trend graphs.</td></tr>
<tr><td>13</td><td>Manage Mode</td><td>• Contains 3 modes of Lock/Operator/Engineer and limits the functions by switching the modes.</td></tr>
<tr><td>14</td><td>Tag data external I/F</td><td>• ActiveX control for displaying the faceplate and the ActiveX control button to display the faceplate are prepared. Pasting them on the ActiveX control support tool of such as Visual Basic® and then setting the property enables faceplate display/monitoring/operation with the external applications. (The monitor tool of PX Developer needs to be in execution status.)
• Reading tag data by tag name (such as "FIC001.PV") with Visual Basic® program is possible.
• Writing tag data by tag name (such as " FIC001.SV") with Visual Basic® program is possible.</td></tr>
<tr><td>15</td><td>GOT screen generator</td><td>• Generates a GOT screen (faceplate, control panel, trend graph screen, alarm list screen, tuning screen, tag setting screen, program setting screen) project for monitoring and tuning tags.
• GOT screen project can be generated automatically from PX Developer project by using the monitor tool to execute simple setting such as placing screens and then following the wizard to enter required items.</td></tr>
<tr><td>16</td><td>Server/client monitoring</td><td>• Large-scale process control systems can be configured with server/client monitoring.
• PC for monitoring: 2 servers (primary, secondary), 7 clients
• The number of connected process CPUs/redundant CPUs: up to 16 CPUs *
* For redundant CPU, a control system and a standby system are counted in set for 1 CPU.</td></tr>
</table>

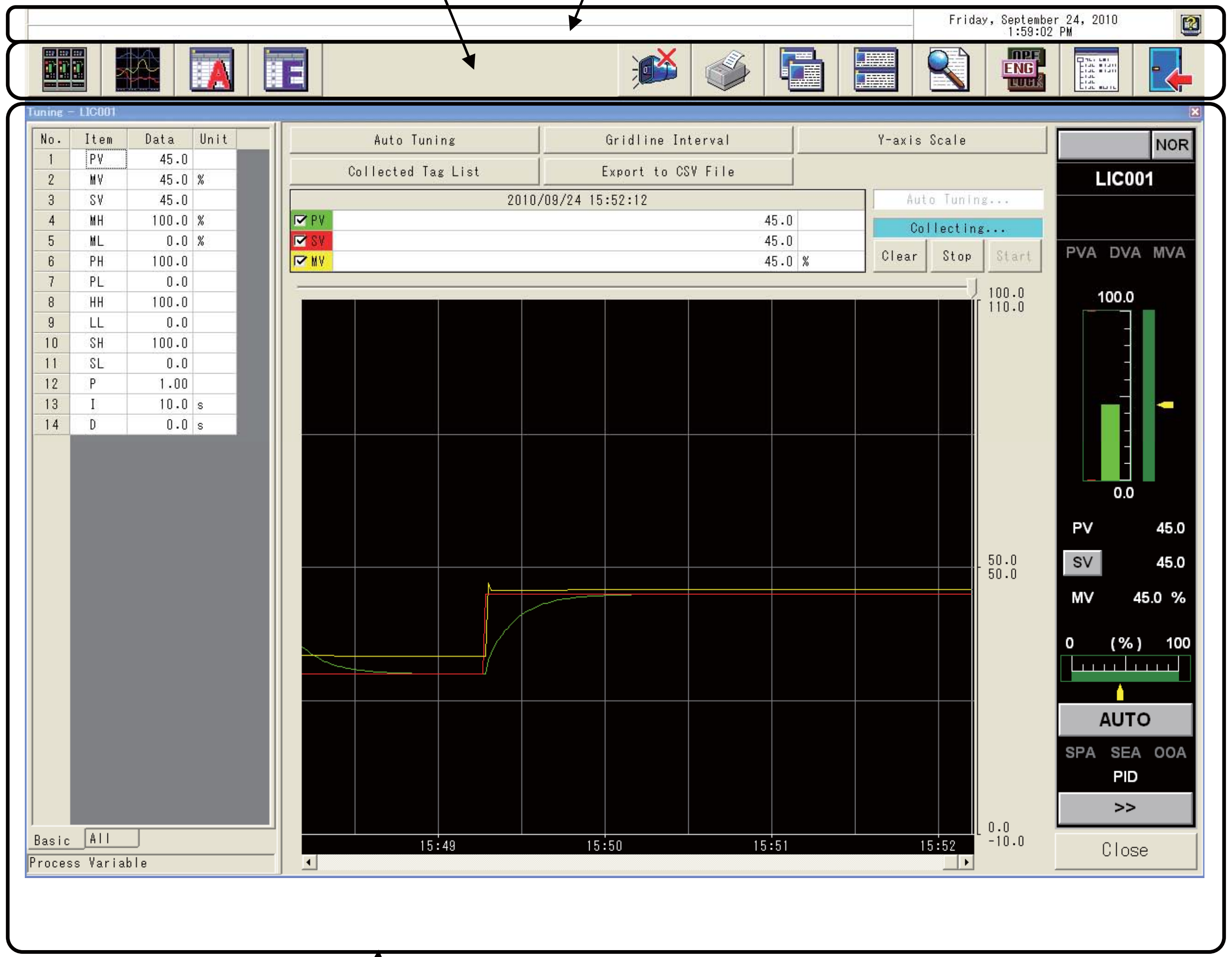
Appendix 4.4 Screen configuration of monitor tool

Monitor toolbar

- Icons that call out each monitor function are displayed.

Alarm and event display

- The latest two alarms or event messages are displayed.



No.	Item	Data	Unit
1	PV	45.0	
2	MV	45.0	%
3	SV	45.0	
4	MH	100.0	%
5	ML	0.0	%
6	PH	100.0	
7	PL	0.0	
8	HH	100.0	
9	LL	0.0	
10	SH	100.0	
11	SL	0.0	
12	P	1.00	
13	I	10.0	s
14	D	0.0	s

☑ PV 45.0
☑ SV 45.0
☑ MV 45.0 %

PV 45.0
SV 45.0
MV 45.0 %

Monitor function display area

- Each monitor function (control panel/trend graph/faceplate/tuning panel/alarm list/event list screen) is displayed

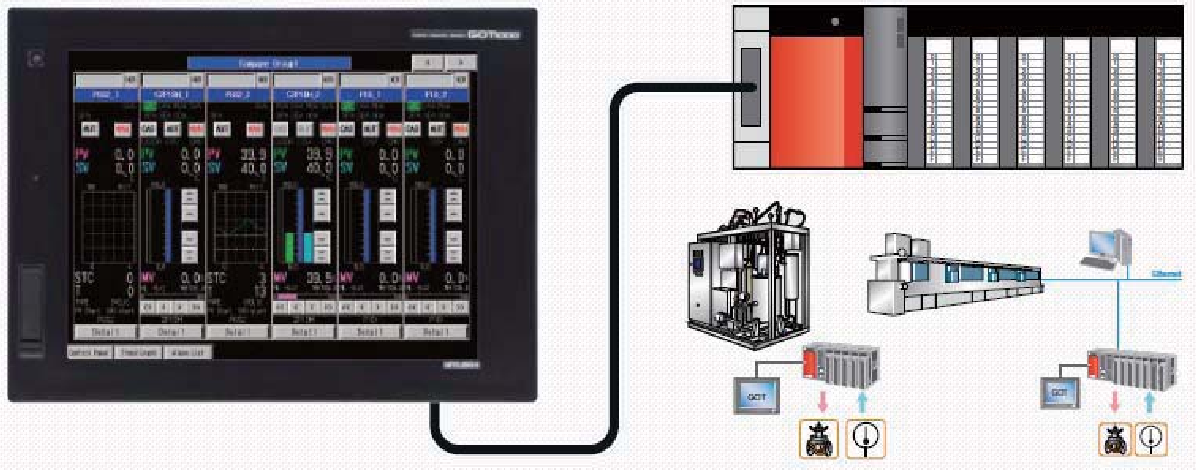
---

Appendix 5 Function Outline of GOT Screen Generator

The GOT screen generator function allows GOT1000 to easily monitor/control projects created with PX Developer. Applying environmental degradation resistance GOT1000 enables to monitor the equipment and worksites. A screen project of GOT1000 screen design software “GT Designer2” can be created automatically with the GOT screen generator function of PX Developer monitor tool.
The following describes the feature of the GOT screen generator function.

Equipment/shop floor monitoring by GOT1000

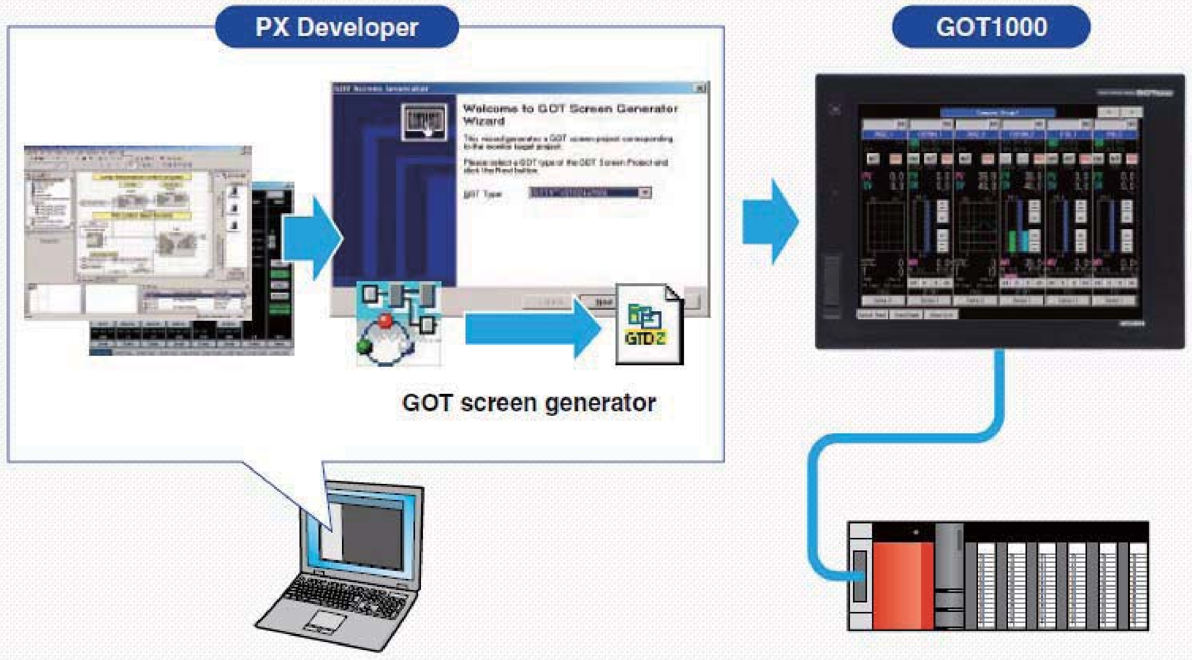
GOT1000 can be used for monitoring equipment and shop floor.



Easy to create GOT1000 process control monitoring screens

- Faceplates and tuning screens for GOT1000 can be automatically generated from PX Developer projects.
- Tag's assigned device settings or programs are not needed for the auto-generated screens.

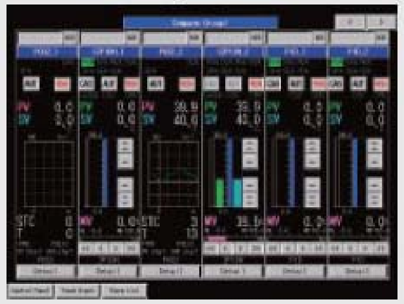
*Only CPU which is connected to the host station can be monitored.
Multiple CPUs which are connected to other stations cannot be monitored.



The following screens can be generated with the GOT screen generator function.

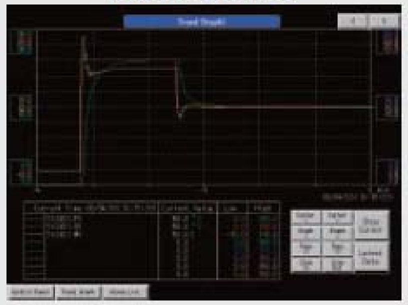
Faceplate

Control panel screen



Alarm list screen

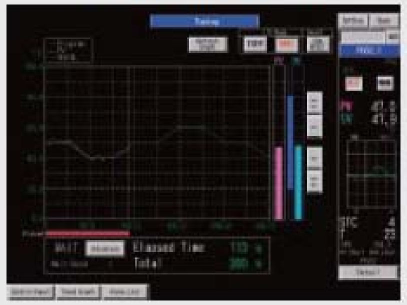
Trend graph screen



Tuning screen

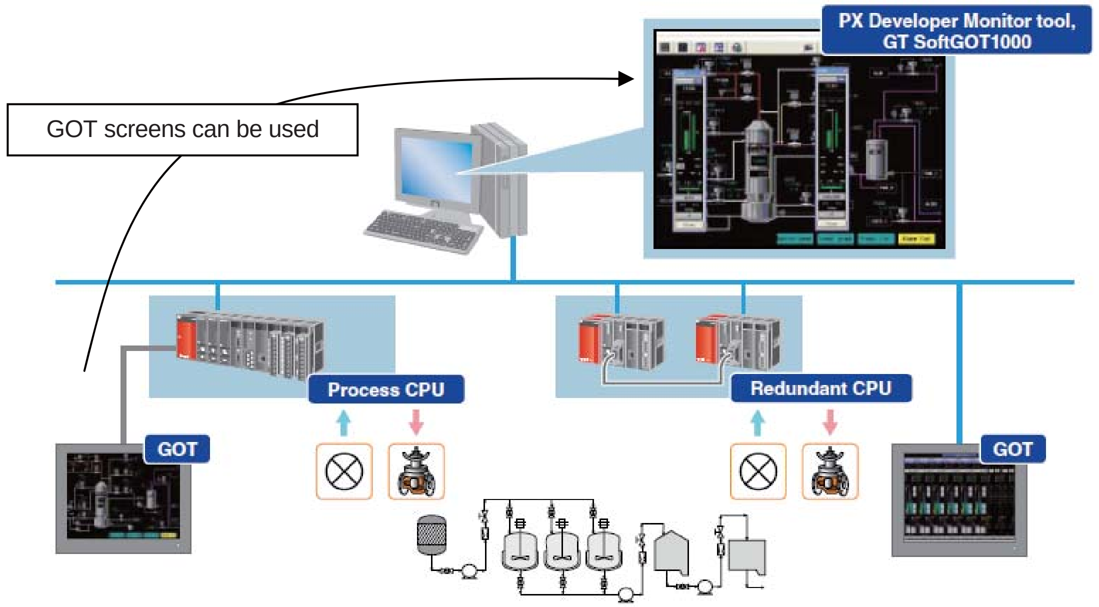
Tag setting screen

Program setting screen



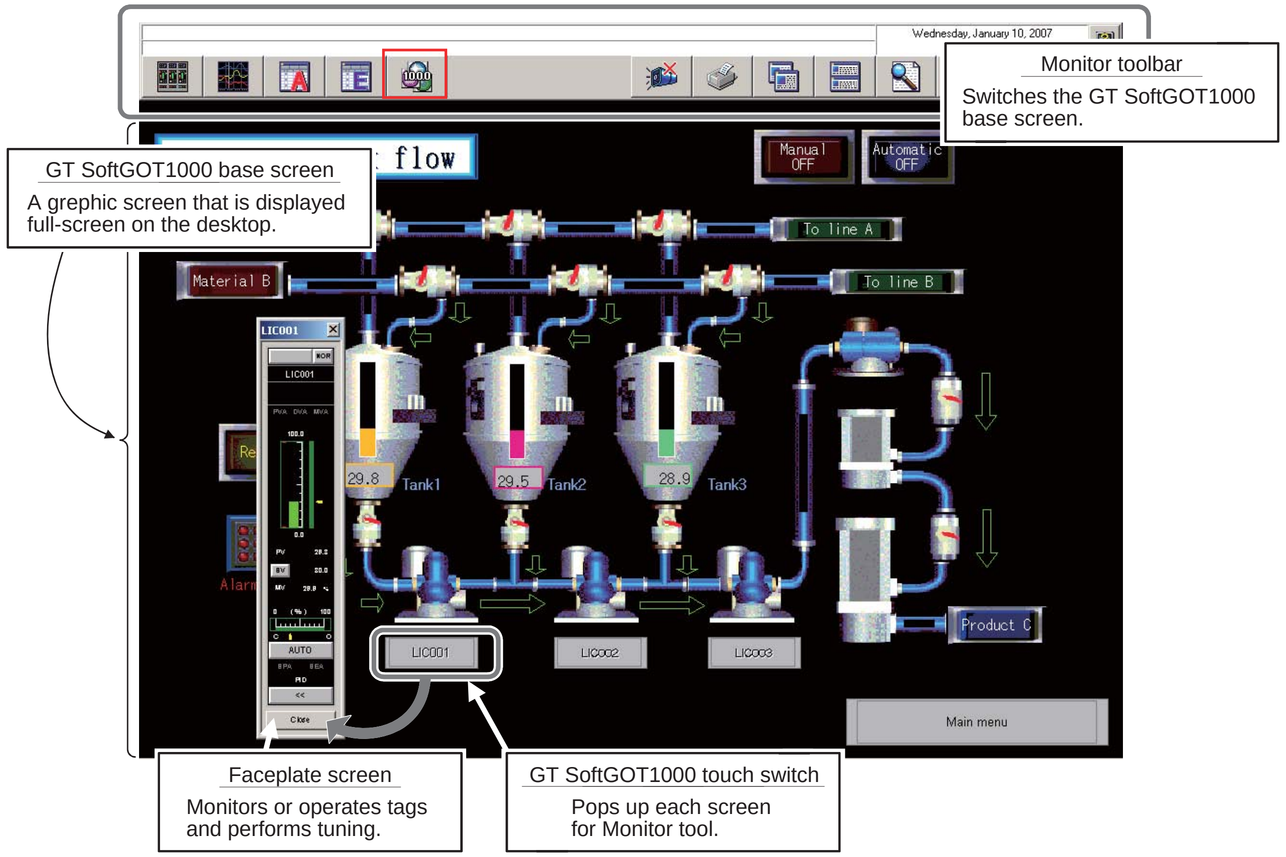
Appendix 6 Interaction between Monitor Tool and GOT SoftGOT1000

The interaction function between the monitor tool and GT SoftGOT1000 calls the monitor tool function of PX Developer from GT SoftGOT1000 and starts GT SoftGOT1000 from the monitor tool of PX Developer with specifying the base screen. Monitoring with a personal computer is possible.



The graphic screen for GT SoftGOT1000 can be created with GT Designer2. A created screen which is used for such as work space monitoring can be used without modification.

(Example)



Monitor toolbar

Switches the GT SoftGOT1000 base screen.

GT SoftGOT1000 base screen

A grephic screen that is displayed full-screen on the desktop.

Faceplate screen

Monitors or operates tags and performs tuning.

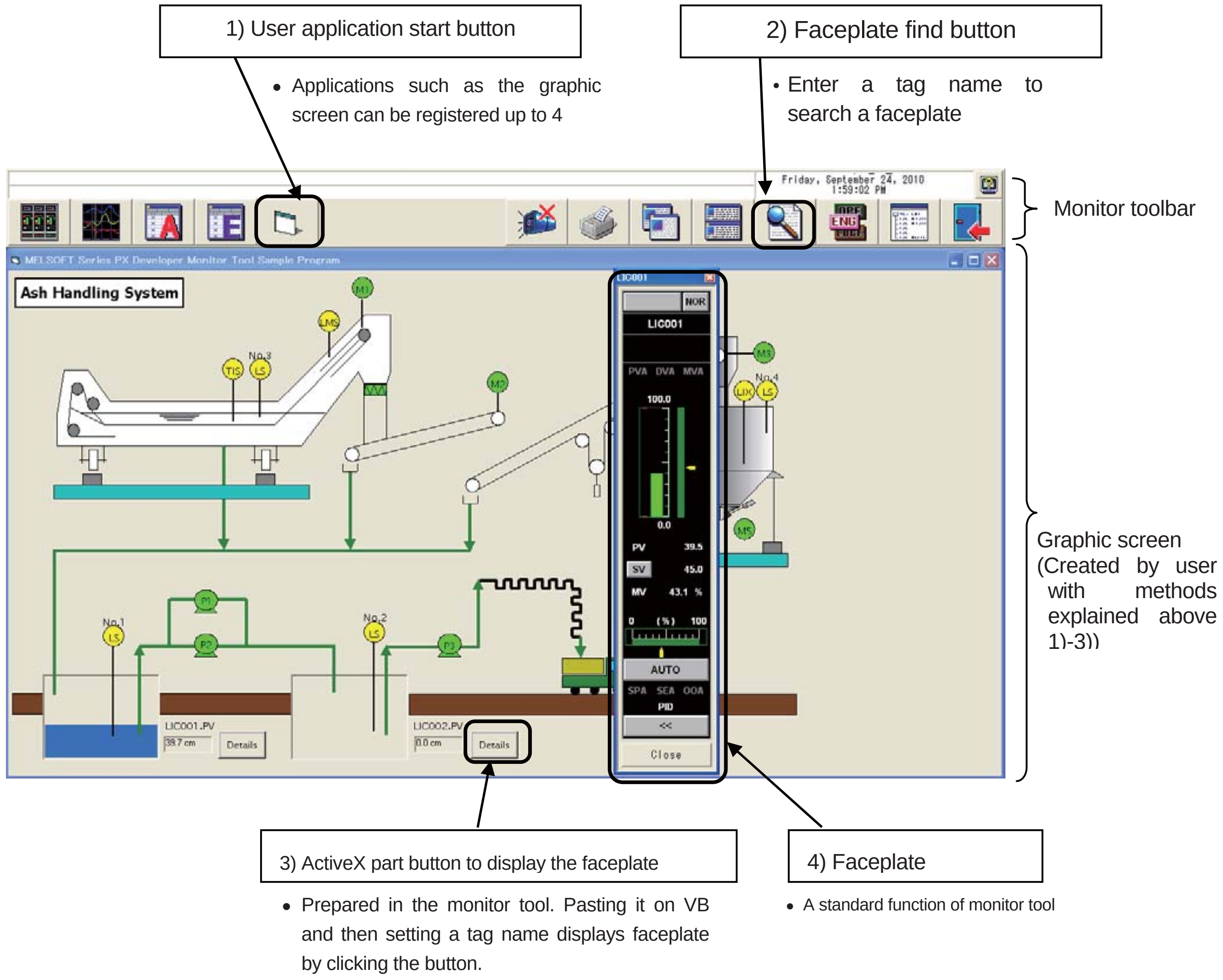
GT SoftGOT1000 touch switch

Pops up each screen for Monitor tool.

Appendix 7 Interaction between Monitor Tool and Graphic Screen Created in Visual Basic®

The combination of PX Developer monitor tool and a graphic screen created in MX Component (ActiveX library for MELSEC communication) + Visual Basic® language (hereafter written VB) configures the monitor function.

Register a graphic screen created in MX Component + VB in "1) User application start button" in the monitor tool and then display the graphic screen by clicking the button. To display "4) Faceplate", which corresponds to the tag on the graphic screen, paste "3) ActiveX parts button to display the faceplate, and then display the faceplate by clicking the button. MX Component is ActiveX library for MELSEC communication of our product, can be accessed from VB to a device memory of MELSEC.



1) User application start button

- Applications such as the graphic screen can be registered up to 4

2) Faceplate find button

- Enter a tag name to search a faceplate

Monitor toolbar

Graphic screen (Created by user with methods explained above 1)-3))

3) ActiveX part button to display the faceplate

- Prepared in the monitor tool. Pasting it on VB and then setting a tag name displays faceplate by clicking the button.

4) Faceplate

- A standard function of monitor tool

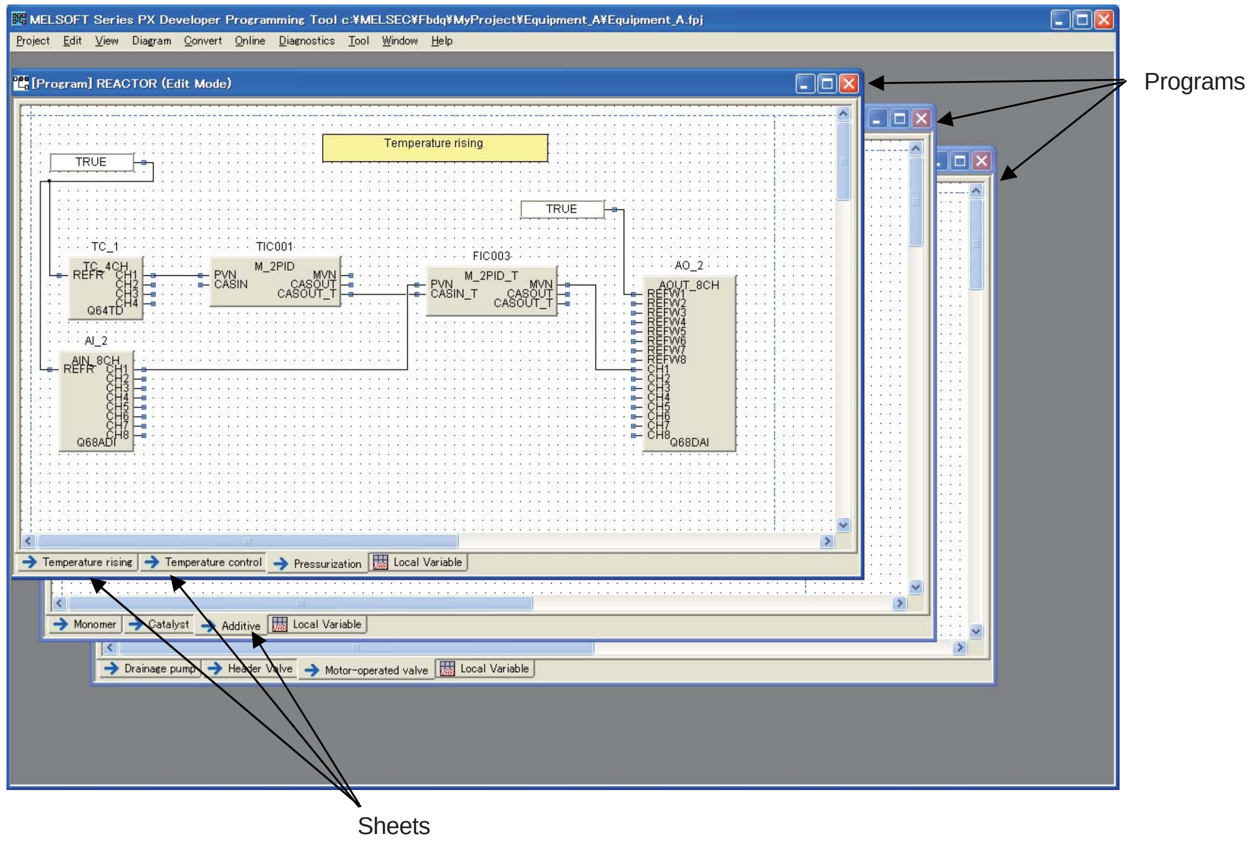
Tag data can be read by entering a tag name such as "FIC001.PV" to method function (read tag data function) for Visual Basic® language. Also, Tag data can be written by entering a tag name such as "FIC001.SV" to method function (read tag data function) for Visual Basic® language.

Appendix 8 Program Execution Functions

Appendix 8.1 Program structure

The following diagram shows the program structure to be created by users with PX Developer.

- Project
  - Program 1
    - Sheet 1 — FB/function
    - Sheet 2 — FB/function
    - Sheet 3 — FB/function
    - Sheet 32 — FB/function
    - Local variable list
  - Program 2
    - Sheet 1 — FB/function
    - Sheet 2 — FB/function
    - Sheet 3 — FB/function
    - Sheet 32 — FB/function
    - Local variable list
  - Program 200
    - Sheet 1 — FB/function
    - Sheet 2 — FB/function
    - Sheet 3 — FB/function
    - Sheet 32 — FB/function
    - Local variable list



(1) Create one group of programs to be executed in Process/Redundant CPU as a project, register programs under the project. The number of programs which can be registered under one project is maximum 200; however, the capacity and the number of files to be written are limited depending on CPUs.

(2) One program contains up to 32 program sheets and 1 local variable list sheet, and users program by pasting FB/function to program sheets. Local variables used in the program sheets are displayed on the local variable list sheet.

(3) The differences between FB and function are as follows.

FB	Function
• Software part which contains internal memory. • When same values are input to FBs, output values are changed depending on internal memory value. • Creating user original FB and categorizing parts into hierarchical structure are possible. (Creates a FB by using FBs/Functions)	• Software part which does not contain internal memory. • When same values are input to FBs, output values do not changed. • Creating user original FB and categorizing parts into hierarchical structure are not possible.

(4) Classification of FB and function

- FB
  - Manufacturer FB
    - General FB
    - Process FB
    - Module FB
  - User-defined FB
    - General FB
    - Tag FB

Manufacturer FB		FBs that are contained in PX Developer (Library)
	General FB	IEC61131-3 standard (FBD standard) FBs such as Bistable, Timer, Counter and FBs for message communication among multiple process CPUs.
	Process FB	FB for loop control such as PID operation, correction operation
	Module FB	FB for I/O processing such as analog module, I/O module
User-defined FB		FBs created by users in PX Developer
	General FB	FBs with user generated processing other than loop control
	Tag FB	FBs with user generated loop controls such as PID operation

- Function
  - General function
  - Process function

General function	IEC61131-3 standard (FBD standard) functions such as Logical operation, Arithmetic operation, Selection/Comparison, and functions for ladder program execution control.
Process function	Helper function for loop control (such as high selector/low selector/middle value selection)

<General function>

1) Type conversion function (_E indicates the function with Enable.)

No.	Function name	Description
1	INT_TO_REAL(_E)	Convert data type INT to REAL.
2	DINT_TO_REAL(_E)	Convert data type DINT to REAL.
3	INT_TO_DINT(_E)	Convert data type INT to DINT.
4	DINT_TO_INT(_E)	Convert data type DINT to INT.
5	INT_TO_BCD(_E)	Convert data type INT to BCD.
6	DINT_TO_BCD(_E)	Convert data type DINT to BCD.
7	INT_TO_WORD(_E)	Convert data type INT to WORD.
8	DINT_TO_WORD(_E)	Convert data type DINT to WORD.
9	INT_TO_DWORD(_E)	Convert data type INT to DWORD.
10	DINT_TO_DWORD(_E)	Convert data type DINT to DWORD.
11	INT_TO_BOOL(_E)	Convert data type INT to BOOL.
12	DINT_TO_BOOL(_E)	Convert data type DINT to BOOL.
13	REAL_TO_INT(_E)	Convert data type REAL to INT.
14	REAL_TO_DINT(_E)	Convert data type REAL to DINT.
15	BCD_TO_INT(_E)	Convert data type BCD to INT.
16	BCD_TO_DINT(_E)	Convert data type BCD to DINT.
17	WORD_TO_INT(_E)	Convert data type WORD to INT.
18	WORD_TO_DINT(_E)	Convert data type WORD to DINT.
19	WORD_TO_BOOL(_E)	Convert data type WORD to BOOL.
20	DWORD_TO_BOOL(_E)	Convert data type DWORD to BOOL.
21	DWORD_TO_INT(_E)	Convert data type DWORD to INT.
22	DWORD_TO_DINT(_E)	Convert data type DWORD to DINT.
23	WORD_TO_DWORD(_E)	Convert data type WORD to DWORD.
24	DWORD_TO_WORD(_E)	Convert data type DWORD to WORD.
25	INT_TO_STRING(_E)	Convert data type INT to STRING.
26	DINT_TO_STRING(_E)	Convert data type DINT to STRING.
27	REAL_TO_STRING(_E)	Convert data type REAL to STRING (exponent form).
28	REAL_TO_STRING_EX(_E)	Convert data type REAL to STRING (decimal point form).
29	STRING_TO_INT(_E)	Convert data type STRING to INT.
30	STRING_TO_DINT(_E)	Convert data type STRING to DINT.
31	STRING_TO_REAL(_E)	Convert data type STRING to REAL.
32	BOOL_TO_INT(_E)	Convert data type BOOL to INT.
33	BOOL_TO_DINT(_E)	Convert data type BOOL to DINT.
34	BOOL_TO_WORD(_E)	Convert data type BOOL to WORD.
35	BOOL_TO_DWORD(_E)	Convert data type BOOL to DWORD.

2) Numerical operation function (_E indicates the function with Enable.)

No.	Function name	Description
1	ABS(_E)	Absolute value
2	SQRT(_E)	Square root
3	LN(_E)	Natural logarithm (e)
4	LOG(_E)	Common logarithm (10)
5	EXP(_E)	Natural exponential
6	SIN(_E)	SIN (sine) operation
7	COS(_E)	COS (cosine) operation
8	TAN(_E)	TAN (tangent) operation
9	ASIN(_E)	SIN^{-1} (principal arc sine) operation
10	ACOS(_E)	COS^{-1} (principal arc cosine) operation
11	ATAN(_E)	TAN^{-1} (principal arc tangent) operation

3) Arithmetic operation function (_E indicates the function with Enable.)

No.	Function name	Description
1	ADD(_E)	Addition
2	MUL(_E)	Multiplication
3	SUB(_E)	Subtraction
4	DIV(_E)	Division
5	MOD(_E)	Modulus Operation

4) Bit-string function (_E indicates the function with Enable.)

No.	Function name	Description
1	SHL(_E)	Shift left
2	SHR(_E)	Shift right
3	ROL(_E)	Rotate left
4	ROR(_E)	Rotate right

5) Logical operation function (_E indicates the function with Enable.)

No.	Function name	Description
1	AND(_E)	And
2	OR(_E)	OR
3	XOR(_E)	XOR
4	NOT(_E)	NOT

6) Selection function (_E indicates the function with Enable.)

No.	Function name	Description
1	SEL(_E)	Input value selection
2	MAX(_E)	Maximum value selection
3	MIN(_E)	Minimum value selection
4	LIMIT(_E)	High/low limit control
5	MUX(_E)	Multiplexer

7) Comparison function (_E indicates the function with Enable.)

No.	Function name	Description
1	>(_E)	Comparison >
2	>=(_E)	Comparison ≧
3	=(_E)	Comparison =
4	<=(_E)	Comparison ≦
5	<(_E)	Comparison <
6	<>(_E)	Comparison ≠

8) Character string function (_E indicates the function with Enable.)

No.	Function name	Description
1	LEN(_E)	Detect and output the input string length.
2	LEFT(_E)	Output specified number of characters from leftmosted.
3	RIGHT(_E)	Output specified number of characters from rightmosted.
4	MID(_E)	Output specified number of characters from specified position.
5	CONCAT(_E)	Concatenate two characters.
6	INSERT(_E)	Insert characters.
7	DELETE(_E)	Delete substring.
8	REPLACE(_E)	Replace characters.
9	FIND(_E)	Find characters.

9) Helper function (_E indicates the function with Enable.)

No.	Function name	Description
1	UNBIND(_E)	Unbind WORD type data into 16 BOOL type data.
2	BIND(_E)	Arrange the 16 BOOL type data into WORD/DWORD type data (low-order word).
3	MAKE_DWORD(_E)	Merge 2 WORD type data into 1 DWORD type data.
4	HI_WORD(_E)	Output the high-order word of DWORD type data.
5	LO_WORD(_E)	Output the low-order word of DWORD type data.
6	IS_CONNECTED(_E)_	Output the connection statuses of input pin.

10) Ladder program control function (_E indicates the function with Enable.)

No.	Function name	Description
1	CALL_DINT(_E)	Subroutine program call (DINT type argument).
2	CALL_REAL(_E)	Subroutine program call (REAL type argument).
3	PSCAN(_E)	Set a sequence program into the scan execution mode.
4	PSTOP(_E)	Set a sequence program into standby mode.
5	POFF(_E)	Set a sequence program into standby mode including reset of the outputs.
6	PLOW(_E)	Set a sequence program into the low-speed execution mode.

<General FB>

1) Bistable FB

No.	FB name	Description
1	SR	SR flip-flop
2	RS	RS flip-flop
3	LATCH_BOOL	Latch (BOOL type)
4	LATCH_ REAL	Latch (REAL type)
5	LATCH_WORD	Latch (WORD type)
6	LATCH_DWORD	Latch (DWORD type)

2) Edge detection FB

No.	FB name	Description
1	R_TRIG	Rising edge detector
2	F_TRIG	Falling edge detector
3	EDGE_CHECK	Edge detection input

3) Counter FB

No.	FB name	Description
1	CTU	Up-counter
2	CTD	Down-counter
3	CTUD	Up-down-counter

4) Timer FB

No.	FB name	Description
1	TP_HIGH	Pulse timer (high-speed timer)
2	TP_LOW	Pulse timer (low-speed timer)
3	TON_HIGH	ON delay timer (high-speed timer)
4	TON_LOW	ON delay timer (low-speed timer)
5	TOF_HIGH	OFF delay timer (high-speed timer)
6	TOF_LOW	OFF delay timer (low-speed timer)

5) Communication control FB

No.	FB name	Description
1	SEND	Sending data to PLC CPUs of other stations (maximum 16 words) (on MELSECNET/10, H, Ethernet)
2	RECV	Receiving data from PLC CPUs of other stations (maximum 16 words) (on MELSECNET/10, H, Ethernet)

<Process function>

1) Analog value selection and average value function

No.	Function name	Description
1	P_HS P_HS_E (with Enable)	High selector
2	P_LS P_LS_E (with Enable)	Low selector
3	P_MID P_MID_E (with Enable)	Middle value selection
4	P_AVE P_AVE_E (with Enable)	Average value
5	P_ABS P_ABS_E (with Enable)	Absolute value

<Process FB>

[General process FB]

1) Correction operation FB

No.	FB name	Description
1	P_FG	Function generator
2	P_IFG	Inverse function generator
3	P_FLT	Standard filter (moving average)
4	P_ENG	Engineering unit conversion
5	P_IENG	Inverse engineering unit conversion
6	P_TPC	Temperature/pressure correction
7	P_SUM	Summation (analog integration)
8	P_SUM2_	Summation (analog integration) Internal integer integration
9	P_RANGE_	Range conversion

2) Arithmetic operation FB

No.	FB name	Description
1	P_ADD	Addition for process (with coefficient)
2	P_SUB	Subtraction for process (with coefficient)
3	P_MUL	Multiplication for process (with coefficient)
4	P_DIV	Division for process (with coefficient)
5	P_SQR	Square root for process (with coefficient)

3) Comparison operation FB

No.	FB name	Description
1	P_>	Compare greater than process (with setting value)
2	P_<	Compare less than process (with setting value)
3	P_=	Compare equal to process (with setting value)
4	P_>=	Compare greater than or equal to process (with setting value)
5	P_<=	Compare less than or equal to process (with setting value)

4) Control operation FB

No.	FB name	Description
1	P_LLAG	Lead-lag compensation
2	P_I	Integral
3	P_D	Derivative
4	P_DED	Dead time operation
5	P_LIMT	High/low limiter
6	P_VLMT1	Rate-of-change limiter1
7	P_VLMT2	Rate-of-change limiter2 (with output limitation)
8	P_DBND	Dead band
9	P_BUMP	Bumpless transfer (Bumpless switch control)
10	P_AMR	Analog memory (Increase or decrease output value by certain ratio.)

[Tag access FB]

1) I/O control operation FB

No.	FB name	Description
1	P_IN	Analog input processing
2	P_OUT1	Output processing-1 with mode switching (with integration/Anti Reset Windup)
3	P_OUT2	Output processing-2 with mode switching (without integration/Anti Reset Windup)
4	P_OUT3_	Output processing-3 with mode switching (with MV compensation/ integration/Anti Reset Windup/Tight shut/Full open)
5	P_MOUT	Manual output
6	P_DUTY	Time proportioning output
7	P_PSUM	Pulse integration
8	P_BC	Batch counter
9	P_MSET_	Manual setter

2) Loop control operation FB

No.	FB name	Description
1	P_R (without tracking to primary loop) P_R_T (with tracking to primary loop)	Ratio control
2	P_PID (without tracking to primary loop) P_PID_T (with tracking to primary loop)	Velocity type PID control
3	P_2PID (without tracking to primary loop) P_2PID_T (with tracking to primary loop)	2-degree-of-freedom PID control
4	P_2PIDH_ (without tracking to primary loop) P_2PIDH_T_ (with tracking to primary loop)	2-degree-of-freedom advanced PID control
5	P_PIDP (without Tracking to primary loop, without tracking from secondary loop) P_PIDP_T (with tracking to primary loop, without tracking from secondary loop)	Position Type PID control
6	P_PIDP_EX_ (without tracking to primary loop, with tracking from secondary loop) P_PIDP_EX_T_ (with tracking to primary loop, with tracking from secondary loop)	Position Type PID control
7	P_SPI (without tracking to primary loop) P_SPI_T (with tracking to primary loop)	Sample PI control
8	P_IPD (without tracking to primary loop) P_IPD_T (with tracking to primary loop)	I-PD control
9	P_BPI (without tracking to primary loop) P_BPI_T (with tracking to primary loop)	Blend PI control
10	P_PHPL	High/low limit alarm check
11	P_ONF2 (without tracking to primary loop) P_ONF2_T (With tracking to primary loop)	2 position ON/OFF
12	P_ONF3 (without tracking to primary loop) P_ONF3_T (with tracking to primary loop)	3 position ON/OFF
13	P_PGS	Program setter
14	P_PGS2_	Multi-point program setter
15	P_SEL (without tracking to primary loop) P_SEL_T1 (with tracking to primary loop) P_SEL_T2 (with tracking to primary loop)	Loop Selector 1 tracking for P_SEL_T1 Up to 2 trackings for P_SEL_T2
16	P_PFC_SF_	Predictive functional control (simple first order lag)
17	P_PFC_SS_	Predictive functional control (simple second order lag)
18	P_PFC_INT_	Predictive functional control (Integral process)

3) Special FB

No.	FB name	Description
1	P_MCHG	Control mode change

[Tag FB]

1) Loop tag FB

No.	FB name	Description
1	M_PID (without tracking to primary loop) M_PID_T (with tracking to primary loop)	Velocity type PID control (with auto tuning) (Taking function of P_IN + P_PHPL + P_PID + P_OUT1 as a single FB.)
2	M_PID_DUTY (without tracking to primary loop) M_PID_DUTY_T (with tracking to primary loop)	Velocity type PID control (Time proportioning output type, with auto tuning) (P_IN + P_PHPL + P_PID + P_DUTY as a single FB)
3	M_2PID (without tracking to primary loop) M_2PID_T (with tracking to primary loop)	2-degree-of-freedom PID control (with auto tuning) (Taking function of P_IN + P_PHPL + P_2PID + P_OUT1 as a single FB.)
4	M_2PID_DUTY (without tracking to primary loop) M_2PID_DUTY_T (with tracking to primary loop)	2-degree-of-freedom PID control (Time proportioning output type, with auto tuning) (Taking function of P_IN + P_PHPL + P_2PID + P_DUTY as a single FB.)
5	M_2PIDH_ (without tracking to primary loop) M_2PIDH_T_ (with tracking to primary loop)	2-degree-of-freedom advanced PID control (with auto tuning) (Taking function of P_IN + P_PHPL + P_2PIDH + P_OUT3 as a single FB and then adding PV/MV correction to it.)
6	M_PIDP (without tracking to primary loop) M_PIDP_T (with tracking to primary loop)	Position type PID control (Taking function of P_IN + P_PHPL + P_PIDP as a single FB.)
7	M_PIDP_EX_ (without tracking to primary loop) M_PIDP_EX_T_ (with tracking to primary loop)	Position type PID control (Taking function of P_IN + P_PHPL + P_PIDP_EX_ as a single FB.)
8	M_SPI (without tracking to primary loop) M_SPI_T (with tracking to primary loop)	Sample PI control (Taking function of P_IN + P_PHPL + P_SPI + P_OUT1 as a single FB.)
9	M_IPD (without tracking to primary loop) M_IPD_T (with tracking to primary loop)	I-PD control (Taking function of P_IN + P_PHPL + P_IPD + P_OUT1 as a single FB.)
10	M_BPI (without tracking to primary loop) M_BPI_T (with tracking to primary loop)	Blend PI control (Taking function of P_IN + P_PHPL + P_BPI + P_OUT1 as a single FB.)
11	M_R (without tracking to primary loop) M_R_T (with tracking to primary loop)	Ratio control (Taking function of P_IN + P_PHPL + P_R + P_OUT2 as a single FB.)
12	M_ONF2 (without tracking to primary loop) M_ONF2_T (with tracking to primary loop)	2 position ON/OFF control (Taking function of P_IN + P_PHPL + P_ONF2 as a single FB.)
13	M_ONF3 (without tracking to primary loop) M_ONF3_T (with tracking to primary loop)	3 position ON/OFF control (Taking function of P_IN＋P_PHPL＋P_ONF3 as a single FB.)
14	M_MONI	Monitor (Indicator) (Taking function of P_IN + P_PHPL as a single FB.)
15	M_MWM	Manual output with monitor (Taking function of P_IN + P_PHPL + P_MOUT as a single FB.)
16	M_BC	Batch preparation (Taking function of P_PSUM + P_BC as a single FB.)
17	M_PSUM	Pulse integrator
18	M_SEL (without tracking to primary loop) M_SEL_T1 (with tracking to primary loop) M_SEL_T2 (with tracking to primary loop)	Loop selector 1 tracking for M_SEL_T1 Up to 2 trackings for M_SEL_T2
19	M_MOUT	Manual output
20	M_PGS	Program setter
21	M_PGS2_	Multi-point program setter
22	M_SWM_	Manual setter with monitor (P_IN + P_PHPL + P_MSET_ as a single FB.)
23	M_PFC_SF_	Predictive functional control (simple first order lag) (P_IN + P_PHPL + P_PFC_SF_ as a single FB.)
24	M_PFC_SS_	Predictive functional control (simple second order lag) (P_IN + P_PHPL + P_PFC_SS_ as a single FB.)
25	M_PFC_INT_	Predictive functional control (Integral process) (P_IN + P_PHPL + P_PFC_INT_ as a single FB.)

2) Status tag FB

No.	FB name	Description
1	M_NREV	Motor irreversible (2 Input, 2 Output)
2	M_REV	Motor reversible (2 Input, 3 Output)
3	M_MVAL1	ON/OFF operation (2 Input, 2 Output)
4	M_MVAL2	ON/OFF operation (2 Input, 3 Output)
5	M_TIMER1	Timer 1 (Timer stops when COMPLETE flag is ON)
6	M_TIMER2	Timer 2 (Timer continues when COMPLETE flag is ON)
7	M_COUNTER1	Counter 1 (Counter stops when COMPLETE flag is ON)
8	M_COUNTER2	Counter 2 (Counter continues when COMPLETE flag is ON)
9	M_PB_	Push button operation (5 Input, 5 Output)

3) Alarm tag FB

No.	FB name	Description
1	M_ALARM	8 pins alarm tag

4) Message tag FB

No.	FB name	Description
1	M_MESSAGE	8 pins message tag

<Module FB>

1) Analog Module FB

No.	FB name	Description
1	AIN_4CH	Read analog data from 4-channel analog input module. (Corresponding module: Q64AD)
2	AIN_8CH	Read analog data from 8-channel analog input module. (Corresponding module: Q68ADV/Q68ADI)
3	AIN_4CH_G	Read analog data from channel-isolated-4-channel analog input module. (Corresponding module: Q64AD-GH)
4	AIN_8CH_G	Read analog data from channel-isolated-8-channel analog input module. (Corresponding module: Q68AD-G)
5	AIN_2CH_DG	Read analog data from channel-isolated-2-channel analog input module with signal conditioning function. (Corresponding module: Q62AD-DGH)
6	AIN_6CH_DG	Read analog data from channel-isolated-6-channel analog input module with signal conditioning function. (Corresponding module: Q66AD-DG)
7	AOUT_2CH	Write analog data to 2-channel analog output module. (Corresponding module: Q62DA/Q62DAN)
8	AOUT_4CH	Write analog data to 4-channel analog output module. (Corresponding module: Q64DA/Q64DAN)
9	AOUT_8CH	Write analog data to 8-channel analog output module. (Corresponding module: Q68DAV/Q68DAVN/Q68DAI/Q68DAIN)
10	AOUT_2CH_G	Write analog data to channel-isolated-2-channel analog output module. (Corresponding module: Q62DA-FG)
11	AOUT_6CH_G	Write analog data to channel-isolated-6-channel analog output module. (Corresponding module: Q66DA-G)
12	AIN_4CH_AOUT_2CH	Read/write analog data from/to analog input/output module (Input 4 channels, Output 2 channels). (Corresponding module: Q64AD2DA)

2) Temperature input module FB

No.	FB name	Description
1	TC_4CH	Read temperature data from 4-channel thermocouple input module. (Corresponding module: Q64TD)
2	TC_8CH_G	Read temperature data from channel-isolated-8-channel thermocouple input module. (Corresponding module: Q68TD-G-H01/Q68TD-G-H02)
3	TCV_4CH	Read temperature data from channel-isolated-4-channel temperature/micro-voltage input module (mV can be directly input) (Corresponding module: Q64TDV-GH)
4	RTD_4CH	Read temperature data from 4-channel temperature input module. (Corresponding module: Q64RD/Q64RD-G)
5	RTD_8CH_G	Read temperature data from channel-isolated-8-channel temperature-measuring resistor input module. (Corresponding module: Q68RD3-G)

3) Counter module FB

No.	FB name	Description
1	HIC_2CH	Read pulse count data from 2-channel high-speed counter module. (Corresponding module: QD62/QD62E/QD62D)
2	PIN_8CH_G	Read pulse count data from channel-isolated-8-channel pulse input module. (Corresponding module: QD60P8-G)

4) Digital I/O Module FB

No.	FB name	Description
1	DIN_8PT	Read from 8 points digital input module. (Corresponding module: QX28)
2	DIN_16PT	Read from 16 points digital input module. (Corresponding module: QX10/QX40/QX40-S1/QX70/QX80)
3	DIN_32PT	Read from 32 points digital input module. (Corresponding module: QX41/QX71/QX81)
4	DIN_64PT	Read from 64 points digital input module. (Corresponding module: QX42/QX72)
5	DOUT_8PT	Write to 8 points digital output module. (Corresponding module: QY18A/QY68A)
6	DOUT_16PT	Write to 16 points digital output module. (Corresponding module: QY10/QY22/QY40P/QY50/QY70/QY80)
7	DOUT_32PT	Write to 32 points digital output module. (Corresponding module: QY41P/QY71/QY81P)
8	DOUT_64PT	Write to 64 points digital output module. (Corresponding module: QY42P)
9	DINOUT_64PT	Read from/ write to 32 points input/32 points output I/O mixed module. (Corresponding module: QH42P)
10	DINOUT_15PT	Read from/ write to 8 points input/7 points output I/O mixed module. (Corresponding module: QX48Y57)

5) CC-Link module FB

No.	FB name	Description
1	CCLINK_1	Data communication with remote station occupying 1 station. (Read from/ write to buffer memory of CC-Link master module.)
2	CCLINK_2	Data communication with remote station occupying 2 stations. (Read from/ write to buffer memory of CC-Link master module.)
3	CCLINK_3	Data communication with remote station occupying 3 stations. (Read from/ write to buffer memory of CC-Link master module.)
4	CCLINK_4	Data communication with remote station occupying 4 stations. (Read from/ write to buffer memory of CC-Link master module.)

(Note) Module FBs other than I/O module FB are applicable to a main base where Process CPU is mounted on and a module which mounted on an extension base only, however, not applicable to modules mounted on a main/extension base of MELSECNET/H remote station. For I/O module FB, applicable to modules mounted on main/extension base of both Process CPU and MELSECNET/H remote station.

Data communication between analog/temperature input/counter/CC-Link module mounted on main/extension base of MELSECNET/H remote station and PX Developer can be simply executed by setting automatic refresh between buffer memory and remote I/O link register (W) with GX Configurator module, and assigning link register (W) to global variable of PX Developer.

Appendix 8.2 Program execution method

The programs created in PX Developer are executed in Process CPU/Redundant CPU in accordance with settings of PX Developer as following methods.

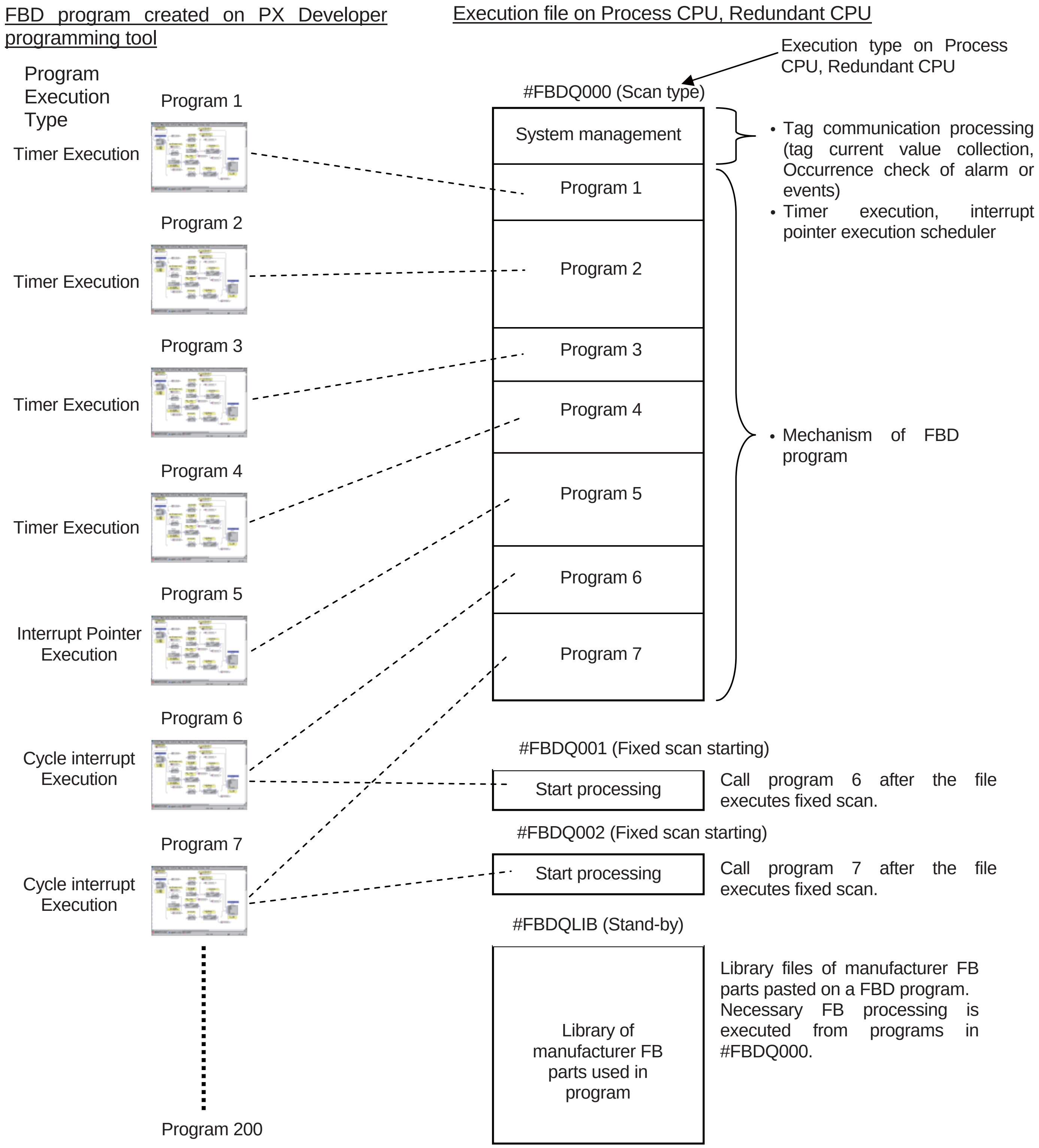
- Timer execution type
  - High speed (200 ms cycle)
  - Normal speed (200ms × m cycle, m = 2/3/4/5)
  - Low speed (200ms × n cycle, n = 5/10/20/25/50)
  - Scan (Each scan interval. Same execution method as ladder program)
- Interrupt execution type
  - Cycle interrupt (setting cycle 1 ms to 999 ms)
  - Random interrupt (interrupt with interrupt pointer I0 to I255)

The following table shows the differences between timer execution type and interrupt execution type, and the application (proper use).

Item	Timer execution type	Interrupt execution type
Features	1) Timer execution is a method to execute programs regularly based on execution cycles or phase by adding up the CPU module scan times. Timer execution type can be selected from the following four types. (High-speed/Normal-speed /Low-speed/Scan) Timer execution at any timing other than those based on scan types will cause an error of up to + 1 scan time. Normal-speed execution type program is executed once in an execution of high-speed execution type program (normal execution type cycle ÷ high-speed execution type cycle (200ms)). Correspondingly, low-speed execution type program is executed once in an execution of high-speed execution type program (low-speed execution type cycle ÷ high-speed execution cycle (200ms)). For example, when high-speed is 200ms interval and normal-speed is 1s interval, normal program is executed once in 5 times. Therefore, when scan time exceeds 200ms, each program cannot be executed with setting interval. Also, when normal-speed execution interval is 1000ms, scan time is 300ms, high-speed program is executed every 300ms. In this instance, normal-speed program is executed once in 5 executions of high-speed program, and the execution interval of normal-speed program is 1.5s. (Error 500ms). Therefore, the scan time should be 200ms or less. 2) Scan type program is started at each scan interval as ladder program.) 3) Setting execution condition of high-speed/normal/low-speed/scan type program beforehand executes programs when execution conditions meet. This enables event processing such as starting error processing program only when error occurs. When no execution condition is set, programs start immediately by execution factors of high-speed/normal/low-speed/scan type. The execution conditions are set up to 8 conditions per one program, can be connected with AND/OR to create compound conditions. 4) When the same interval is set in multiple timer execution type programs, (for example 5 normal-speed programs) the priority of the program executions can be set. Also, when multiple normal-speed/low-speed programs are existed, shifting (load balancing) the execution timings by setting the phases is possible.	1) CPU interrupt execution function executes the programs. 2) Cycle interrupt executes programs by interrupt execution with setting cycle between 1ms to 999 ms. In this instance, no regular margin of error occurs (does not affected by scan times), however, overhead time takes 165μs from reaching the setting cycle to programs being executed. 3) Random interrupt executes programs when interrupt factors occur in each interrupt pointer I0 to I255. Interrupt factors of I0 to I255 are such as interruption from interruption input module (QI60), intelligent function module/network module, and occurrence of an error. (For details, refer to User's manuals of each CPU or MELSEC-Q DATA Book) 4) When an execution factor of interrupt execution type program occurs during execution of a timer execution type program or a user-created ladder program, stop the program, execute an interrupt execution type program and then resume the program. When an execution factor of random interrupt program occurs during execution of cycle interrupt program, stop the program, execute a random interrupt program, and then resume the cycle interrupt program.
Application (proper use)	1) High-speed/normal/low-speed execution is appropriate for processes with controls whish require regular intervals such as general PID control 2) Scan type execution is appropriate for the sequence control required by the process control (controls of such as solenoid valve, ON/OFF solenoid valve, motor, and pump), processing which require regular executions (regular interval execution is not needed) such as error detection and processing the detected errors. 3) For High-speed/normal/low-speed execution, if processing does no complete within the setting interval, it causes delays in the interval.	1) Cycle interrupt is appropriate for processes require interval which is higher than 200ms or the strict fixed scan cycle. 2) Random interrupt is appropriate for processes processing event interrupt (interruption occurs irregularly). 3) For cycle interrupt, if processing does no complete within the setting interval, it causes delays in the interval.

Appendix 8.3 Relation between FBD program and execution file on Process CPU, Redundant CPU.

The following shows the mechanism of compiling FBD program created in the PX Developer programming tool to write to CPU and executing the file and relation between FBD programs and files executed in Process CPU.



(1) Compiling FBD program created in the PX Developer programming tool creates ladder files #FBDQ000, #FBDQLIB, #FBDQ001 and later.
#FBDQ001 and later are created per file when the fixed scan execution is set as FBD program execution type. (For example, when the fixed scan execution is set for 5 FBD programs, 5 files of #FBDQ001, #FBDQ002, #FBDQ003, #FBDQ004, #FBDQ005 will be created.)

When the fixed scan execution is set for FBD program, actual programs are created in #FBDQ000 and start processing is created in #FBDQ001 and later.
The maximum number of programs which can be set the fixed scan execution is 100.
In this instance, files to be created are 100 which are from #FBDQ001 to #FBDQ100.

(2) The number of executable files in CPU is 28 for Q02PHCPU, 60 for Q06PHCPU, 124 for each Q12PHCPU, Q12PRHCPU, Q25PHCPU (*), Q25PRHCPU (*), therefore, design to be the total number of files is the executable number of files or less for each CPU.

* The number of files can be saved in CPU is 252 for each Q25PHCPU, Q25PRHCPU.

Appendix 9 Process Control Related Terminology

The terms listed in this section is for process control related terms which are useful for process control engineering with PX Developer.

- Generic terms and abbreviations used in the terminology

PX Developer	————	Abbreviation for process control FBD software package
QnPHCPU	————	Generic term for Process CPU (Q02PHCPU/Q06PHCPU/Q12PHCPU/Q25PHCPU)
QnPRHCPU	————	Generic term for Redundant CPU (Q12PRHCPU/Q25PRHCPU)

- The terms are listed in alphabetical order, numerical order
- Related terms are indicated with →, antonyms are indicated with⇔.

<A>

Absolute pressure

The amount of pressure measured by full (absolute) vacuum as standard. When indicating as absolute pressure, add abs after engineering units.
Example: 5kg/cm²abs
⇔ Gauge pressure

Alarm status

Indicates the alarm occurrence status of tag alarm such as high high limit alarm (HH), high limit alarm (H), low limit alarm (L), low low limit alarm (LL).

Alarm level

The levels of alarm item importance of tag alarm. The levels are major alarm, minor alarm.

Analog input/output module

- Input module
  An input module imports analog standardized signal of 4 to 20mA, 0 to 5V DC from a sensor to PLC CPU.
  The types for Channels of input module are the channel Isolated type which channels share a common line, the non-insulated type which channels has different common lines, the distributor with 2-wire transmitter type, and direct connectable type with thermocouple and temperature-measuring resistor.
- Output module
  An output module exports analog standardized signal of 4 to 20mA, 0 to 5V DC from PLC CPU to a final control element.
  The types for Channels of output module are the channel Isolated type which channels share a common line and the non-insulated type which channels has different common lines.

AUTO

→ AUTO mode

Auto tuning

Method that detects dynamic characteristics by moving the plant and automatically obtains proportional gain (Kp), integral time (Ti), and derivative time (Td) of PID.
Auto tuning can be performed with step response method for QnPHCPU, QnPRHCPU.
→ Optimal value adjustment

AUTO mode

The mode controlled by setting value (SV) set on the HMI screen.

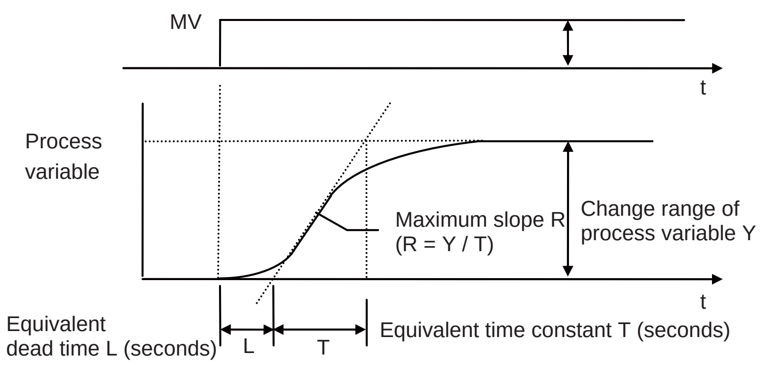
Auto tuning function

Auto tuning function has two methods: the Step Response method and the Limit Cycle method. The limit cycle method is less affected by process variable noise compared to the step response method, and stabilized tuning results can be obtained.

(1) Step Response method

- For actual plant, outputs step MV, and determines the PID constants by maximum slope and equivalent dead time.

(a) Setting constants by the step response method



Common constants

Control	Proportional gain	Integral time (seconds)	Derivative time (seconds)
P	$\frac{1}{R \times L} \times \frac{MV (\%)}{100}$		
PI	$\frac{0.9}{R \times L} \times \frac{MV (\%)}{100}$	3.33L	
PID	$\frac{1.2}{R \times L} \times \frac{MV (\%)}{100}$	2L	0.5L

Obtain the optimum value from the above formula, and perform the fine tuning.

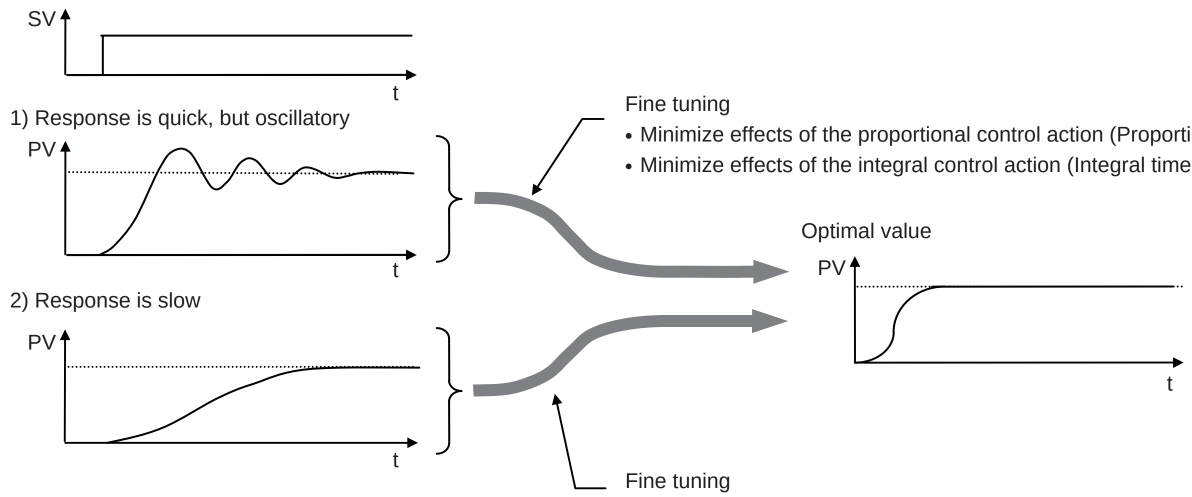
(Example) Equivalent dead time L: 8 seconds, Equivalent time constant T: 16 seconds, Change range of process variable Y: 0.25,
Step manipulated variable: 20%, Maximum slope R = 0.25/16 ＝ 0.016

For PI control, based on the above chart: Proportional gain = $\frac{0.9}{R \times L} \times \frac{MV (\%)}{100} = \frac{0.9}{0.016 \times 8} \times \frac{20}{100} = 1.4$

Integral time = 3.33L = 3.33 x 8 = 26.6 seconds
Derivative time = 0 second

(b) Fine tuning
Obtain a value close to the optimum value, and perform the fine tuning.

For the change in the target value:



Fine tuning
- Minimize effects of the proportional control action (Proportional gain: Smaller)
- Minimize effects of the integral control action (Integral time: Bigger)

Fine tuning
- Maximize effects of the proportional control action（Proportional gain: Bigger）
- Maximize effects of the integral control action（Integral time: Smaller）

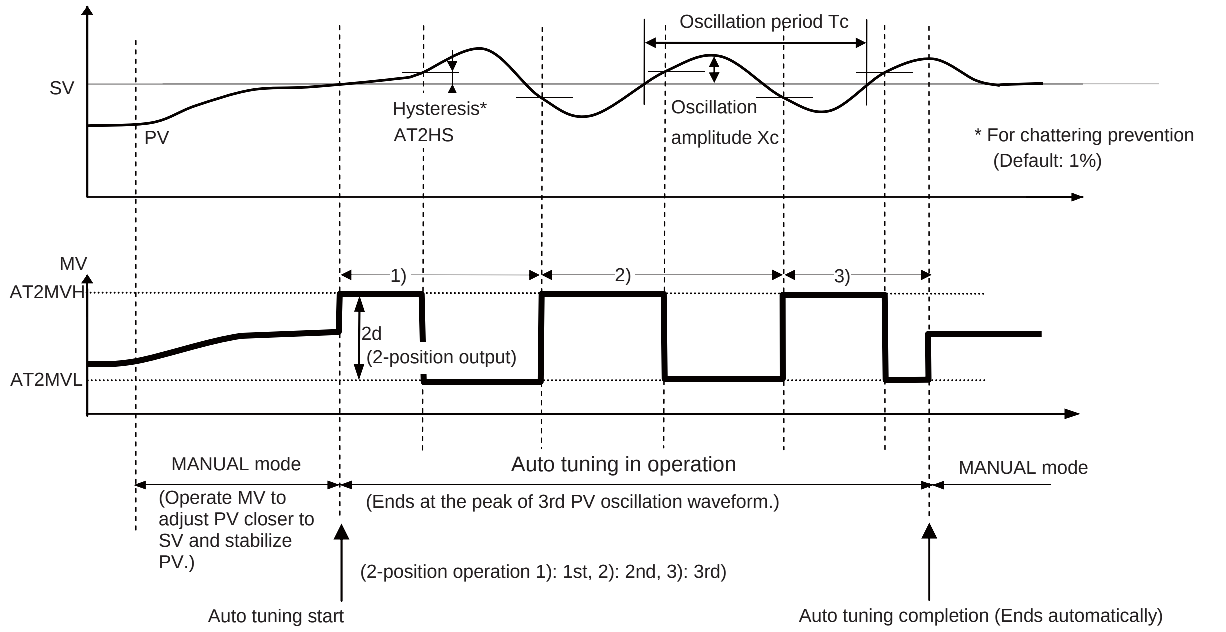
(2) Limit Cycle method

- In the limit cycle method, PV is temporarily oscillated by performing the 2-position operation (ON/OFF operation) output three times and constants are set by the PV amplitude and oscillation period.

(a) Generation and measurement of limit cycle waveform

Oscillates PV by the 2-position operation output, and measures amplitude Xc and oscillation period from the 2nd and 3rd waveform data whose oscillation waveform is stabilized.

The following chart shows a waveform example of auto tuning. (When PV ≦ SV, and reverse action at the start of auto tuning)



* For chattering prevention (Default: 1%)

(b) Calculation of threshold sensitivity (Ku) and threshold period (Tu)

Calculate optimum PID constant from threshold sensitivity (Ku) and threshold period (Tu).

Threshold sensitivity $Ku = 4d / (\pi\sqrt{}\ (Xc^2 - AT2HS^2))$

Xc : Oscillation amplitude

d : Amplitude of 2-position operation output ((AT2MVH - AT2MVL) / 2）

Threshold period Tu = Tc

Tc : Oscillation period

(c) Calculation of optimum PID constant

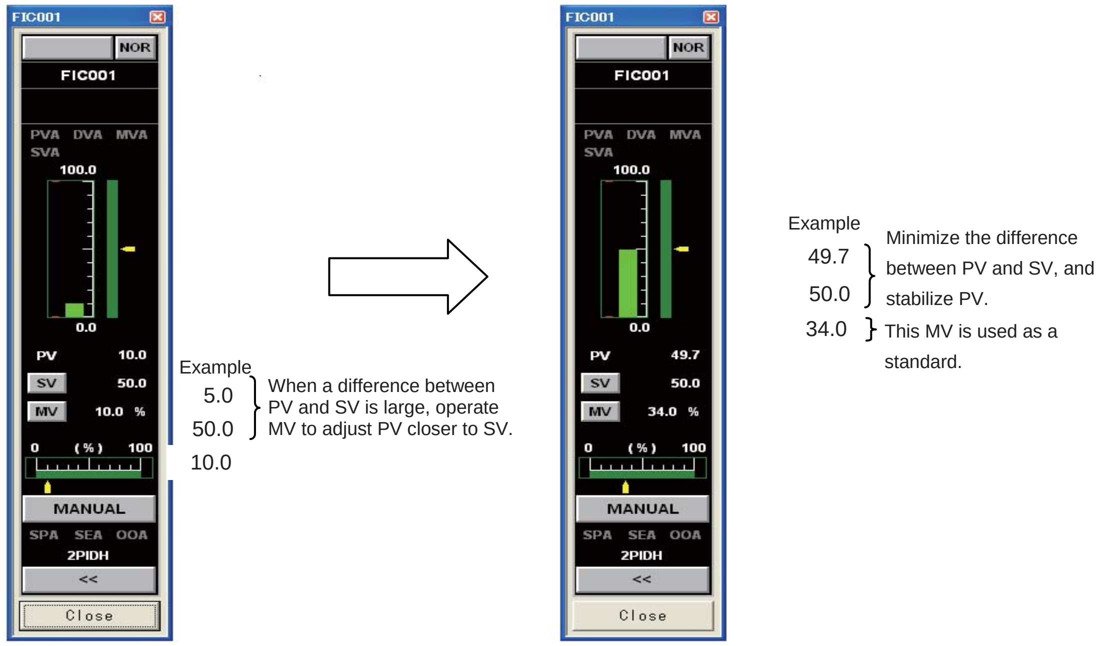
Calculate optimum PID constant from threshold sensitivity (Ku) and threshold period (Tu).

Control Type	Control operation	Proportional gain (Kp)	Integral time (Ti)	Derivative time (Td)	Empirical rule	Remarks
Constant-value control	PI	0.45Ku	0.83Tu	0	Ziegler Nichols's method	Improves the disturbance response
	PID	0.6Ku	0.5Tu	0.125 Tu		
Follow-up control	PI	0.3Ku	1.0Tu	0	CHR method	Suppresses the overshoot when the setting value is changed
	PID	0.45Ku	0.6Tu	0.1 Tu		

(d) Setting high limit (AT2MVH)/low limit (AT2MVL) of 2-position operation output.

The following is an example of setting 2-position operation output which reduces impact on processing as much as possible when performing the auto tuning.

Set the control mode to MANUAL mode, and set SV to be used in the operation. Then, adjust PV closer to SV by operating MV, and stabilize PV.



Example
5.0
50.0 } When a difference between PV and SV is large, operate MV to adjust PV closer to SV.
10.0

Example
49.7
50.0 } Minimize the difference between PV and SV, and stabilize PV.
34.0 } This MV is used as a standard.

Set the 2-position operation output according to MV when PV is stabilized as a standard. Specify a value for the amplitude d of the 2-position operation output not to impact the processing when PV is oscillated up and down as SV in a center.

(Example)

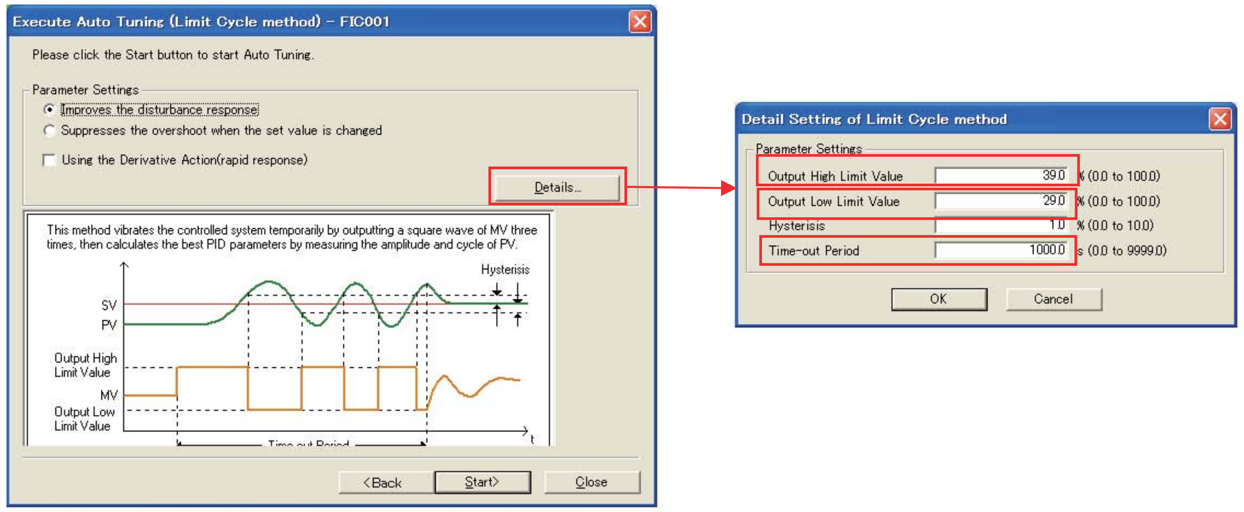
When MV is 34% at PV stabilization, and amplitude of 2-position operation output is 5%, values for output high limit (AT2MVH) and output low limit (AT2MVL)) become as follows.

AT2MVH = MV + d = 34% + 5% = 39%

AT2MVL = MV - d = 34% - 5% = 29%

Set the above output high limit (AT2MVH) and output low limit (AT2MVL)) in PX Developer monitor tool.

For the operation of auto tuning and the screens below, refer to ”PX Developer Version 1 Operating Manual (Monitor tool)”.



Please click the Start button to start Auto Tuning.

Parameter Settings

● Improves the disturbance response

○ Suppresses the overshoot when the set value is changed

☐ Using the Derivative Action(rapid response)

This method vibrates the controlled system temporarily by outputting a square wave of MV three times, then calculates the best PID parameters by measuring the amplitude and cycle of PV.

Parameter Settings

Output High Limit Value 39.0 % (0.0 to 100.0)

Output Low Limit Value 29.0 % (0.0 to 100.0)

Hysterisis 1.0 % (0.0 to 10.0)

Time-out Period 1000.0 s (0.0 to 9999.0)

Consider processing characteristics, set a time-out period three times more than an oscillation period which is measured by the 2-position operation output.

(e) Fine tuning

Fine tuning of PID constants is as same as that for Step Response method.

<B>

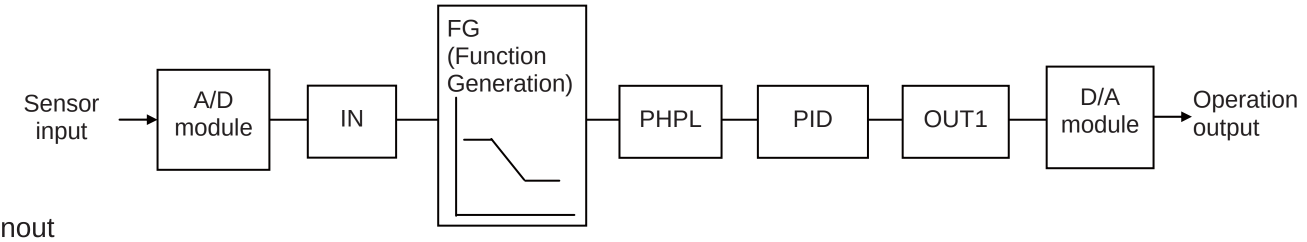
Batch process control

A type of control which produces various products with the same equipment or devices. It has processes of Polymerization, mixture.
Complicating controls such as switching recipes for each kind of products, selecting processes, CIP are required. Recently, the batch process control type is increased.
In addition, production operation in the batch production process (batch recipe registration, batch reservation, execution recipe expansion, batch progress management, batch sequence execution management, device monitoring, and performance collection) are called batch management. One of standards in batch management is ISA SP88 model.
A type of control which produces the same products with the same equipment or devices is called continuous process control.

Broken line correction

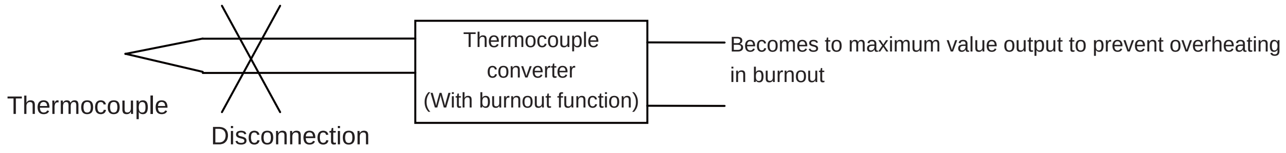
It is used when the value from the process target is not in proportion to process variable from the sensor. Input value is approximated and corrected by broken line. Process FB P_FG is applied to the broken line correction.

(Example)



Burnout

When converter input is in the non-input mode caused by such as sensor disconnection, follow through upper or lower limit of converter output signal.
Example: For thermocouple, becomes to maximum value of thermocouple converter output to prevent overheating in burnout,.



Bumpless

At the time of AUTO ⇔ MANUAL mode switching, this function prevents step changes caused by sharp change of manipulated variable (MV) output, and ensures MV to be converted smoothly.

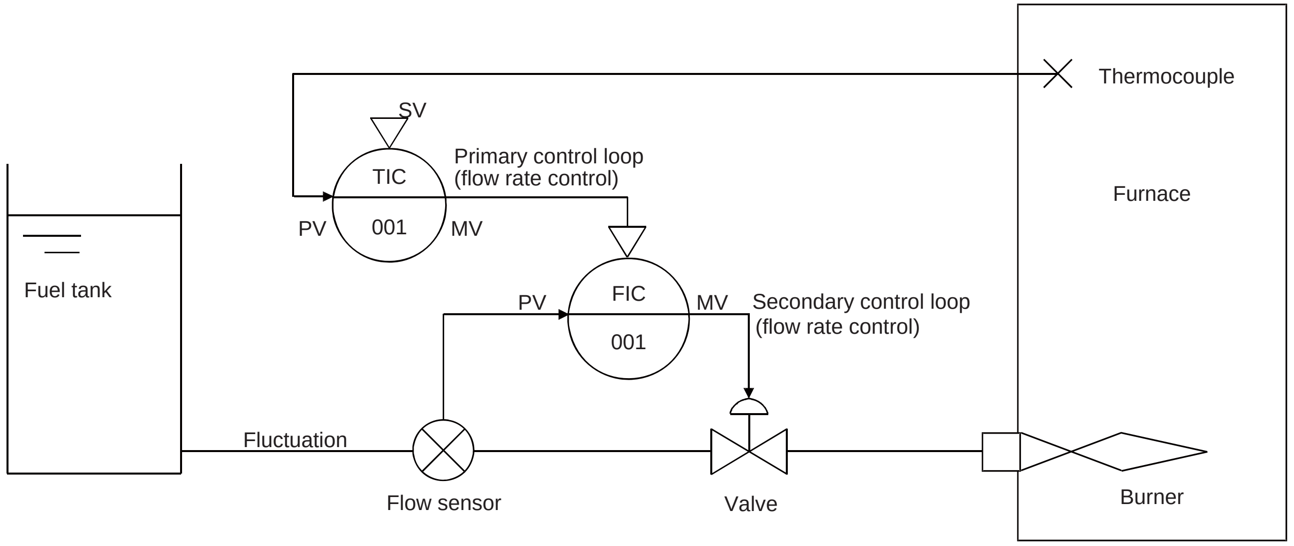
<C>

CAS/CASCADE

→ CASCADE mode

Cascade control

Cascade control is composed of double loop of primary loop and secondary loop. It is the control that removes the effect on the process and improves the whole control performance by checking out disturbance entering secondary loop in an early stage as well as absorbing them into secondary loop.
Generally, the response of secondary loop is desirable to be over 3 times faster than primary loop.
→ Tracking (CASCADE with tracking)
The following diagram is an example of controlling the furnace temperature in a certain value. It absorbs fuel supply variation by flow rate control of secondary control loop and improves response characteristics of temperature control as a whole.



→ Tracking (Cascade with tracking)

CASCADE mode

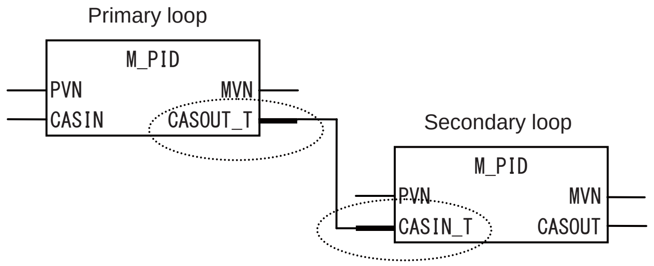
This is the mode for cascade control which controls primary loop output variable (MV) as secondary loop setting value (SV). This mode is also used when regarding setting value (SV) as primary indicated value such as interlock operation with other loops and the case of combination with Program setter.

CASIN pin/CASOUT pin

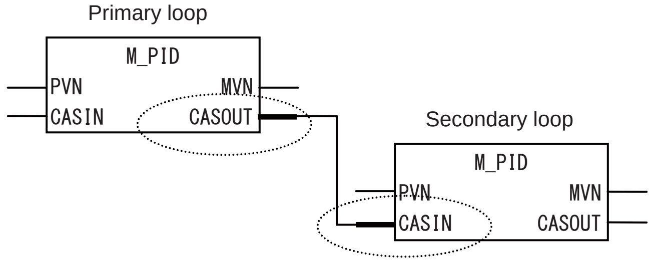
CASIN pin: Input pins on tag FB for setting value input of cascade control secondary loop and for which from PCL program.
”_T” is added to the names of pins which have the tracking function.

CASOUT pin: Output pins from cascade control primary loop to secondary loop.
”_T” is added to the names of pins which have the tracking function.

- With tracking

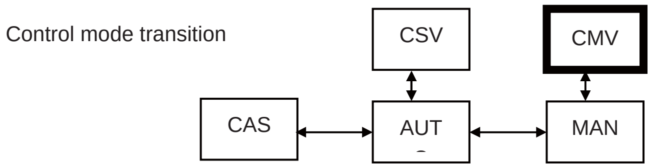


- Without tracking



CMV

Abbreviation for COMPUTER MV. One of control modes and changes MV from upper computer.



Cold junction compensation

A compensation function for thermocouple input module to reduce a measurement error caused by changes in ambient temperature of the criterial terminal. For temperature measurement using a thermocouple module, the ambient temperature of the criterial terminal needs to be maintained at 0°C. However, it is difficult to maintain at 0°C in reality. This function reduces measurement errors by adding a thermal EMF equivalent to the ambient temperature to the internal amplifier

Cold start

A system which outputs from the reset values not the previous values when restart after a power failure of control system.
On the other hand, a system which outputs from the previous values is called hot start.
⇔ Hot start

Control cycle

A cycle of control activity. With continuous control function block, activity such as input processing starts every execution cycle, however, PID control operation starts every control cycle. (Control cycle should be set to be the integral number multiple of execution cycle.) Instructions which can be set a control cycle are PID, BPI, IPD, 0NF2, ONF3, R, 2PID.
→ Execution cycle

(Reference) Selection example of control cycle (CT)
In PID control, when Integral time is relatively big (long), bigger (longer) the control cycle (CT) improves the control performance.

Selection example of control cycle

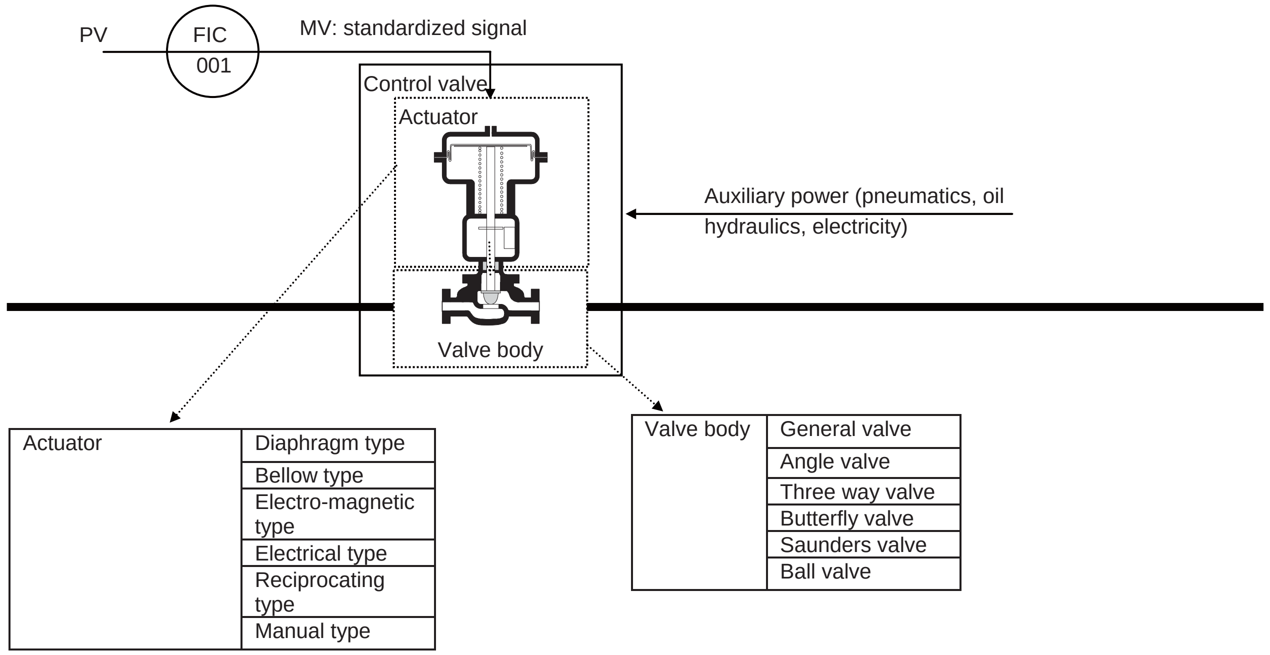
Integral time (Ti)	Control cycle (CT) (Standard)
1 second to 40 seconds	1 second
41 seconds to 80 seconds	2 seconds
81 seconds to 160 seconds	4 seconds

Control mode

A switch which changes the control mode such as MANUAL (MANUAL, MAN, M), AUTO (AUTO, AUT, A), CASCADE (CASCADE, CAS, C). Normally, a switch from CAS to MAN, and MAN to CAS are via AUTO. In stop alarm, it switches from CAS to MAN automatically. There is the operation mode as well.

Control valve

By operation signal from a controller of automatic control, operates valve body with auxiliary power such as pneumatics, oil hydraulics, electricity, and controls the variables to the specified ones. Composed of actuator and valve body.

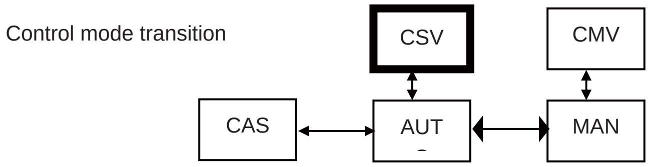


Actuator	Diaphragm type
	Bellow type
	Electro-magnetic type
	Electrical type
	Reciprocating type
	Manual type

Valve body	General valve
	Angle valve
	Three way valve
	Butterfly valve
	Saunders valve
	Ball valve

CSV

Abbreviation for COMPUTER SV. One of control modes and changes SV from upper computer.



<D>

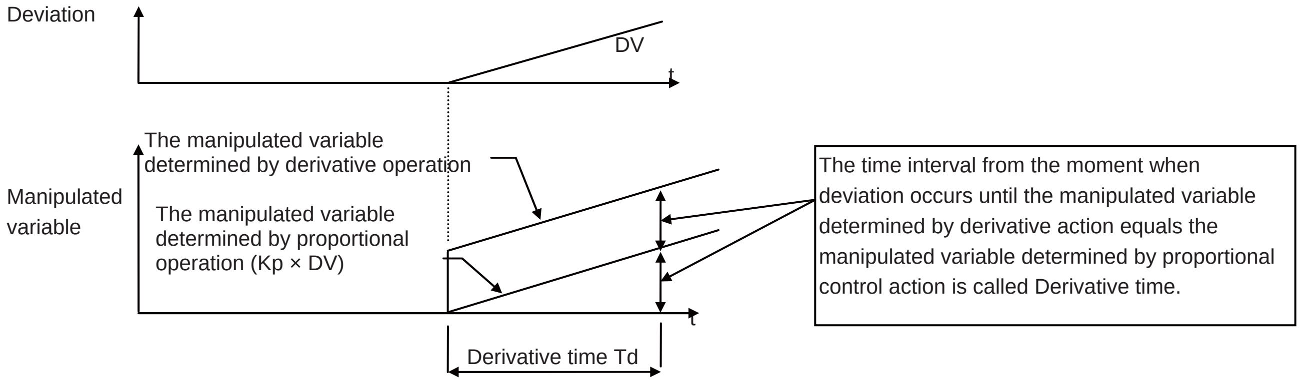
D operation

Derivative action

This is the operation that imposed on the manipulated variable that is in proportion to the rate of change (difference between the current value and the last value) of deviation DV (the difference between process variable and setting value).

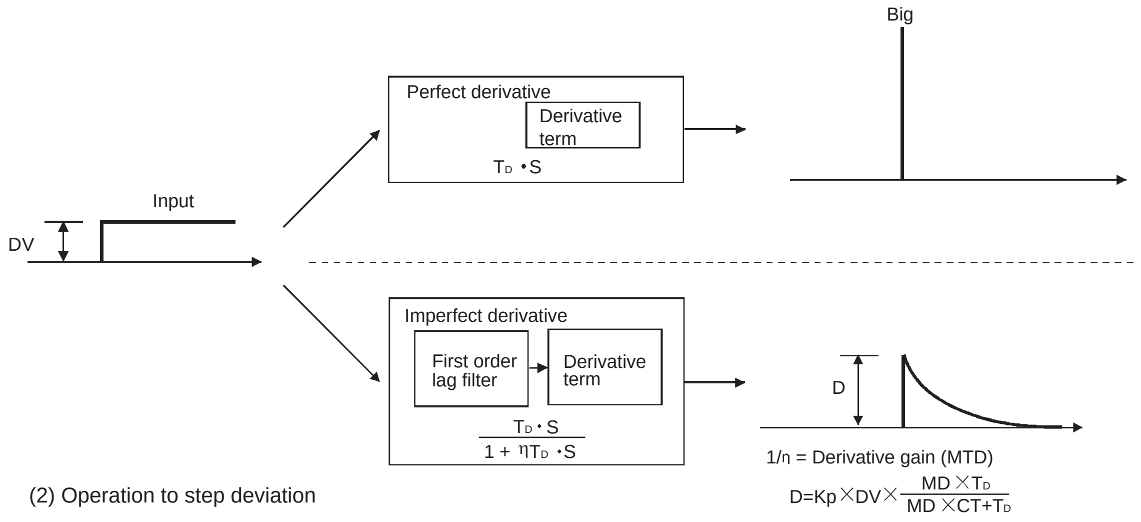
The time interval from the moment when deviation occurs until the manipulated variable determined by derivative action equals the manipulated variable determined by proportional control action is called Derivative time "Td".

(1) When deviation is changing at a constant rate

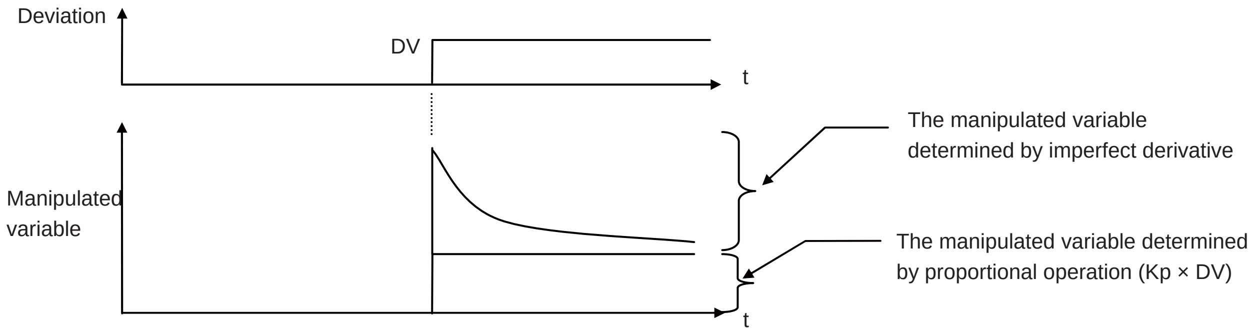


- Imperfect derivative

If derivative is applied to deviation as it is, it may be affected by increase of high-frequency noise, and since the time range of MV is narrow (e.g. in case of step-shaped change, it will be output only at the moment like pulse shape.).
There may be the bad influence that the energy which outputs final control element fully is not given.
Therefore, normally the derivative term input with imperfect differentiation for which filter shall be applied once.
The derivative action of QnPHCPU, QnPRHCPU is imperfect derivative.



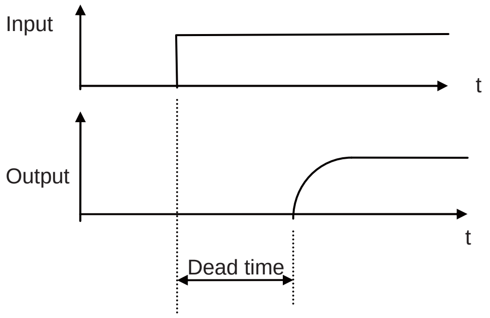
(2) Operation to step deviation



When derivation time is relatively smaller	Derivation effect becomes lighter.
When derivation time is relatively bigger	Derivation effect becomes stronger. Cause short-period hunting and the system may become unstable.

Dead time

Time interval of output variable change to input variable change. P_DED of process FB is applied.



DCS

Distributed digital control system with microcomputer.

DDC

A control with digital display controller.

Derivative action/Derivative time

→ D operation

Design temperature

In temperature/pressure correction of flow rate, when measuring flow rate by using different temperature from design specification temperature, the correction to convert to flow rate in design specification temperature is needed. Design temperature in this case is design specification temperature.

Design pressure

In temperature/pressure correction of flow rate, when measuring flow rate by using different pressure from design specification pressure, the correction to convert to flow rate in design specification pressure is needed. Design pressure in this case is design specification pressure.

Deviation

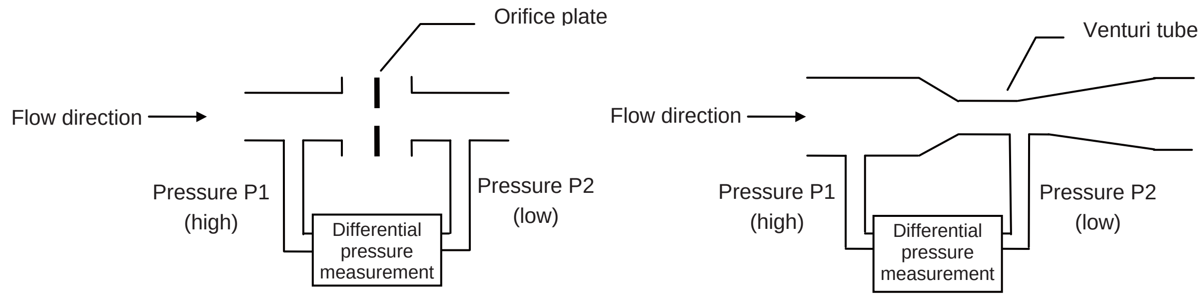
The difference between setting value SV and process variable PV

Deviation derivative type PID

→PV-derivative type PID/PV-proportional and -derivative type (I-PD control)

Differential pressure

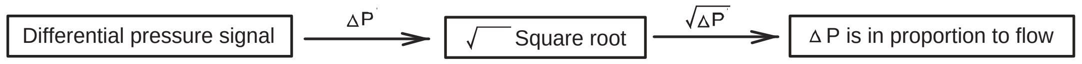
Pressure measured based on pressures other than atmosphere pressure and full vacuum. To differentiate from the others, add diff. after units.
Example: 1kg/cm²diff.
Applied to such as flow rate measurement by differential pressure.



$$Q = K \bullet \sqrt{\Delta P/\gamma}$$

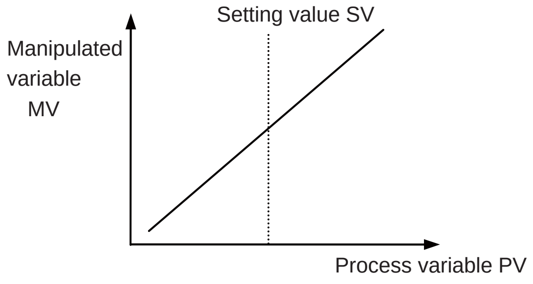
Q: Flow rate, Differential pressure ΔP: (P1 - P2), K: Proportional constant, γ: Density

When measuring flow rate by using differential pressure, the proportional characteristics will be obtained through square root extraction of differential pressure data.
Temperature/pressure correction (P_TPC) is used if necessary.



Direct action

In PID control, an activity to increase manipulated variable MV against increase of process variable PV.
(Example: cooler)
⬄ Reverse action

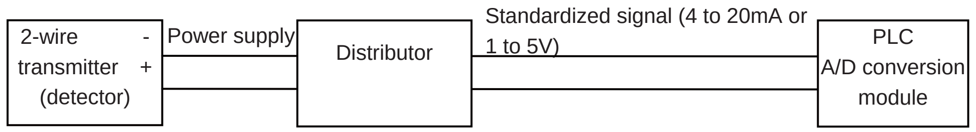


Disable alarm

Disable alarm detection to alarm items of tag alarm by setting as disable.

Distributor

A signal distributor which supplies power to a 2-wire transmitter (detector), and retrieves standardized signals (4 to 20mA or 1 to 5V) from a PLC.



DV

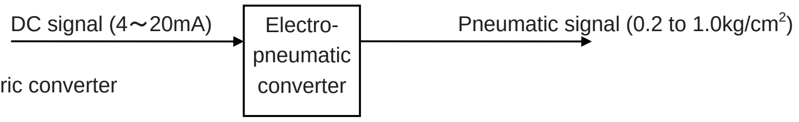
Deviation
The difference between setting value (SV) and process variable (PV)

<E>

Electro-pneumatic converter

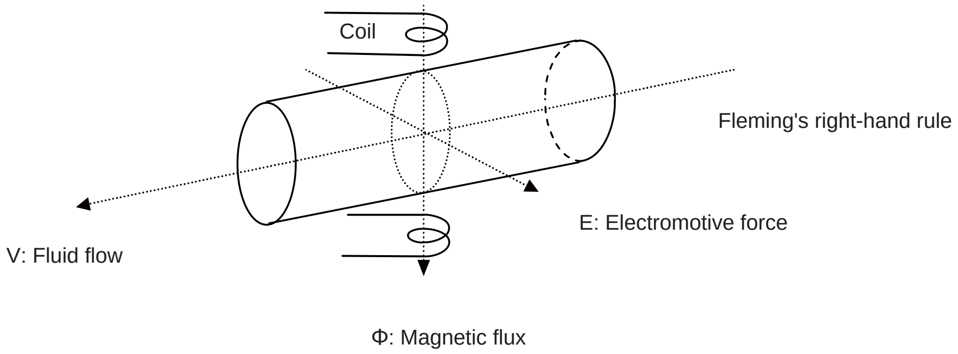
A converter which converts a standardized signal (electrical signal) to a standardized signal (pneumatic signal). Electro-pneumatic transducer.

⬄ Pneumatic-electric converter



Electromagnetic flowmeter

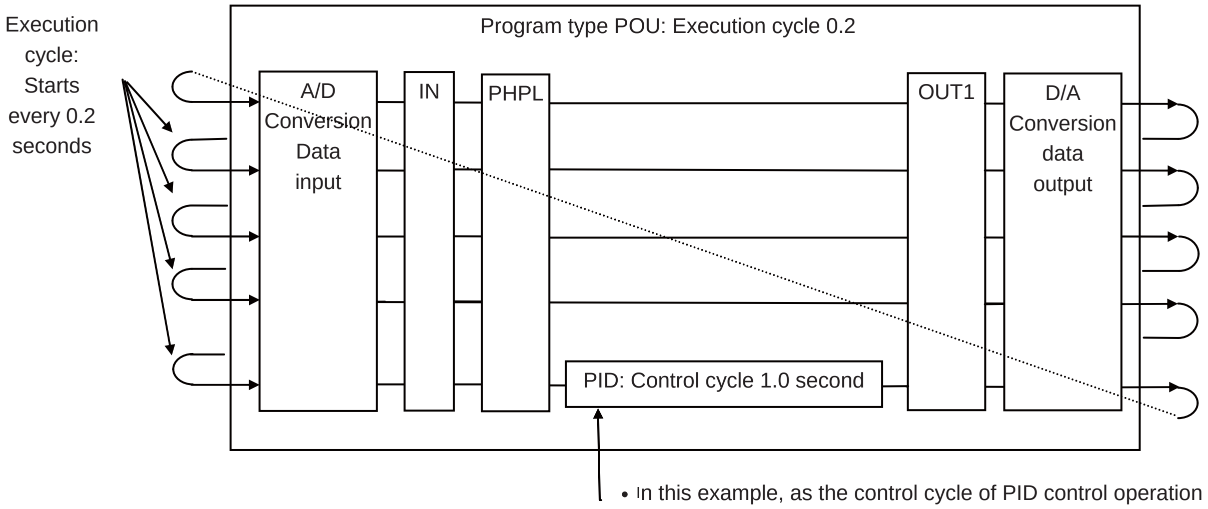
When conductive fluid flows across magnetic field, induces electromotive force in proportion to flow velocity. A flowmeter which detects a flow rate by this theory is called electromagnetic flowmeter.
→ Flowmeter



Execution cycle/Control cycle

Program type POU which consists of such as IN, PHPL, OUT1 starts at regular cycles. This cycle is called execution cycle. In PX Developer, the execution cycle of high-speed (100ms), normal speed (200 to 500ms), low-speed (500 to 5000ms) can be set. For control operation cycle such as PID, BPI, set as a control cycle (CT) differently from an execution cycle. Control cycle should be set to be the integral number multiple of execution cycle.
→ Control cycle

The relation between execution cycle and control cycle
Example: execution cycle of PID control is 0.2 seconds, and control cycle of PID instruction is 1.0 second.



- In this example, as the control cycle of PID control operation instruction is 1.0 second, PID processing is executed in every 1.0 second (set the control cycle as integral multiple of the execution cycle.)
- Control operation instructions which can be set control cycles are BPI, IPD, ONF2, ONF3, R, 2PID besides PID.
- If the control cycle is not the integral multiple of execution cycle, round off the number after the decimal point of control cycle (CT)/execution cycle (T) and multiply the execution cycle to calculate the control cycle.
  Example: When the execution cycle (CT) is 2.5 seconds, and the control cycle (ΔT) is 1 second.
  2.5/1.0=2.5 Round 2.5 off, and the control cycle becomes the execution cycle times 3.

<F>

Filter

(1) First order lag filter

This is used as filter for eliminating noise etc. of process variable (PV). Execute the first order lag operation by the following expression.

$$PVf = \frac{T1 \times PVf_{n-1}}{T1 + \triangle T} + \frac{\triangle T \times PV}{T1 + \triangle T}$$

T1: Time constant(s), △T: Execution cycle, PV: Present input value, PVf_{n-1}: Previous filter value

Process FB (P_LLAG) of lead-lag compensation is applied.

(2) Digital filter (Index filter)

This is used as filter for eliminating noise etc. of process variable (PV).
Operate the sum of weight (PV filter coefficient) of Current process variable and previous filter value.

$PVf = PV + \alpha (PVfn-1 – PV)$

α: PV filter coefficient, PV: Present input value, PVfn-1: Previous filter value
(For the effect of filter by PV filter coefficient α, refer to 4.3 Filtering Function (2) Digital filter function)

The digital filter function of analog input process FB (P_IN) is applied.

(3) Moving average filter

Output the average value of 'SN' pieces of input data that are sampled at data collection interval.
Process FB (P_FLT) of standard filter is applied.

Flowmeter

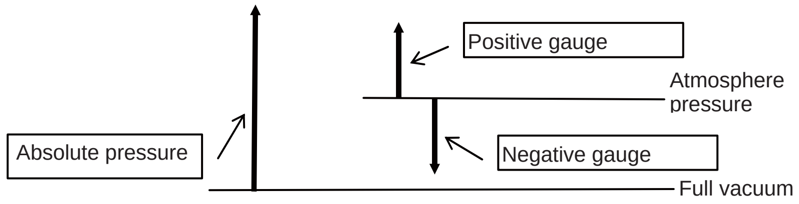
The following shows the representative measurement methods of flowmeter. Hydrometry is often applied in processes along with manometry, thermometry.

Volume hydrometry	Differential pressure type (orifice, venturi tube): detected by differential pressure type.	Liquid○, Gas○, Steam○
	Variable area type: detected by float position.	Liquid○, Gas○, Steam○
	Electro-magnetic type: detected by electromotive force.	Liquid○, Gas×, Steam×
	Supersonic type: detected by propagation time differences or Doppler effect.	Liquid○, Gas○, Steam△
Integrated volume hydrometry	Positive displacement (oval gear, Roots type): detected by the number of revolutions.	Liquid○, Gas○, Steam×
	Vortex (Karman vortex) type: detected by frequencies of Karman vortex	Liquid○, Gas○, Steam○
	Turbine type: detected by the number of revolutions	Liquid○, Gas○, Steam△
Mass hydrometry	Coriolis type: detected by Coriolis force	Liquid○, Gas△, Steam×
	Thermal type: detected by temperature rise of fluid when heating.	Liquid△, Gas○, Steam×

<G>

Gauge pressure

Pressure volume described based on atmosphere pressure (＝0), and widely used. Pressure higher than atmosphere pressure is positive pressure, lower than atmosphere pressure is negative pressure. When differentiation from absolute pressure is needed, add G after the unit.
Example: 3kg/cm²G
→ Absolute pressure, Differential pressure



1atm (Pressure) = 101325 N/m² = 1.03 kgf/cm²
= 101325Pa = 101.325kPa = 0.101325Mpa

1Pa = 1N/m²
1kgf/cm² = 98kPa

Absolute pressure = Atmosphere pressure + gauge pressure

<H>

HH

→ High high alarm

High selector

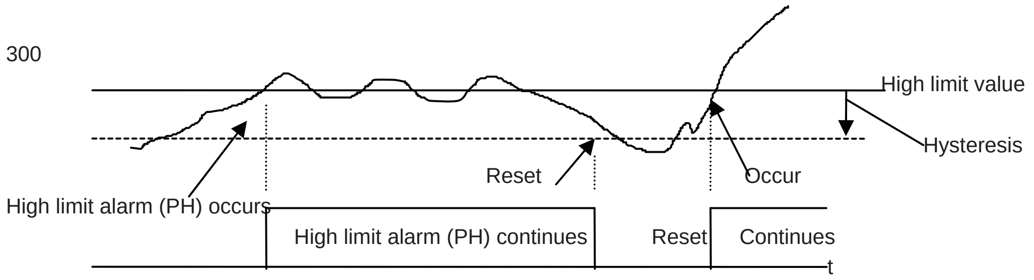
→ Selection control

High alarm/High high alarm

High limit alarm (PH)/high high limit alarm (HH)

Hysteresis

A characteristic which outputs variables depending on directivity past record of input variables.



Hot start

A system which outputs from the previous values when restart after a power failure of control system
→ Cold start

<I>

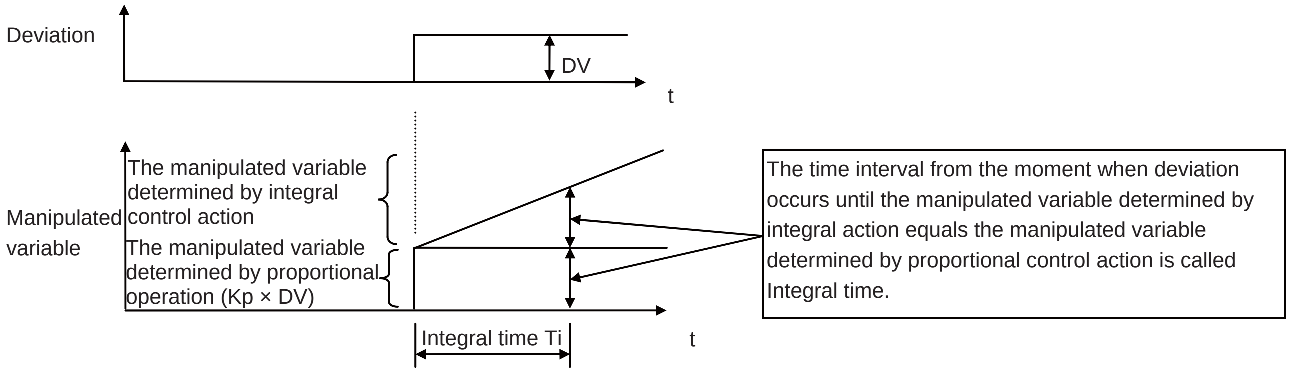
Instrumentation flow chart

A flow chart which shows entire control system and describes such as piping, detector, final control element, controller in symbols.

I operation

Integral operation
The action that continuously changes the manipulated variables, in order to eliminate deviation DV (difference between process variable and setting value).
It can eliminate the offset caused by proportional action. The time interval from the moment when deviation occurs until the manipulated variable determined by integral action equals the manipulated variable determined by proportional control action is called Integral time "Ti"

- Operation to step deviation



The time interval from the moment when deviation occurs until the manipulated variable determined by integral action equals the manipulated variable determined by proportional control action is called Integral time.

When integral time Ti is relatively smaller	The integral effect becomes stronger, and the time for eliminating offset becomes shorter. However, hunting may easily occur.
When integral time Ti is relatively bigger	The integral effect becomes lighter, and the time for eliminating offset becomes longer.

I/O mode

Describes an operation mode connected to I/O (input/output).
The mode types are NOR (NORMAL), SIM (SIMULATER), and OVER(OVERRIDE).

Mode	Contents
NOR	A normal status which is connected to input-output card.
SIM	Executes simulation activity with separating input-output card.
OVER	In such as sensor failure, executes operation with separating only signal from input card and communicating signal from output card. Input data can be set from a screen.

I-PD control

→PV-derivative type PID

Identification

Find process parameter (PID constant) by Step response method.

Industrial unit data

Measured data expressed in actual industrial unit rather than expressed in percentage.

Integral operation/Integral time

→ I operation

Input override

A function which enables process variable (PV) simulated input when input signal fails.

- Loop tag
  A function when cannot attain the proper PV input signal due to such as detecting sensor errors, the input status can be set on a screen.
  However, external output is executed.
  (It is used when batch sequence transition is to be executed.)
- Status tag
  A function when cannot attain the correct input status due to such as imperfect contact of the switch, the input status can be set on a screen. However, external output is executed.
  (It is used when batch sequence transition is to be executed.)

→ I/O mode

<L>

Level meter

The following shows the representative level meter types.

Contact type	differential pressure (liquid-operated), float-type (buoyancy), purge, electrode, capacitance
Non-contact type	ultrasonic、microwave type

LL

→ Low low alarm

Lockout tag

An eye-graph tag displayed on a faceplate that indicates precautions for operations and restrictions according to operation authority.

Loop

Control loop which constitutes feedback loop such as PID control.

Loop tag

A tag which has the loop control functions such as PID control, and a faceplate.
→ Status tag

Low alarm/Low low alarm

Low limit alarm (PL)/low low limit alarm (LL)

Low selector

→ Selection control

<M>

MAN/MANUAL

→ MANUAL mode

Major alarm

An alarm for occurrence of serious error which cannot continue operations of a process, and of devices/equipment failure.

MANUAL mode

In automatic control such as PID control, a mode which can be set and changed the manipulated variables (MV) by an operator manually.

Manometer

A device which measures pressure, the following shows the representative types. Manometer is used in a process along with thermometry and hydrometry.

Electric type	Resistance type, piezoelectric type
Elastic type	Bourdon-tube, diaphragm, bellows type
Liquid column type	U-tube, single - pipe system

→ Gauge pressure, Absolute pressure

Mass flowmeter

One of flowmeters which measures fluid mass. When the fluid temperature or pressure changes drastically, the density of fluid also changes, temperature/pressure correction needs to be executed against volume flow, therefore, complicated systems and accidental error factors are caused. In this case, Mass hydrometry is preferred and recently, often used. Types of mass flowmeter are such as Coriolis type which utilizes that twisting power (Coriolis force) occurred in vibrating U-tube is proportional to mass flow rate which passes through the tube and thermal type which measures temperature rise of fluid in heating.
→ Flowmeter

Minor alarm

Alarm which does not be severe obstacle to the operation

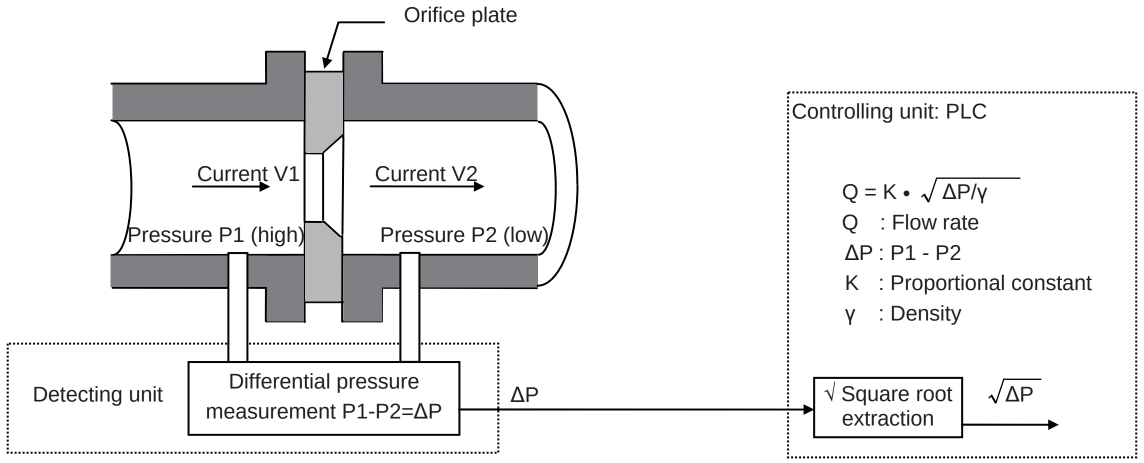
MV

Manipulated variable

<O>

Orifice

Drosselgerate (orifice plate) which is equipped in a conduit line for measuring differential pressure which occurs before and after throttling depending on the volume of flow rate.

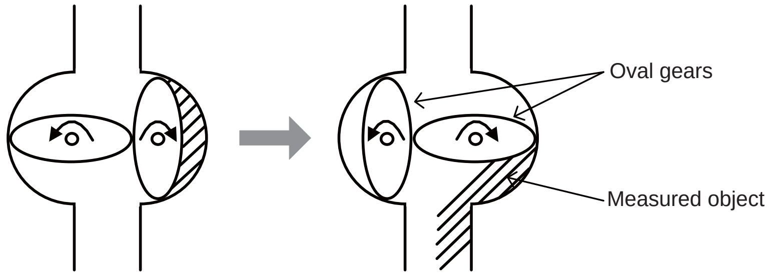


Output signal processing

Functions of process control instruction such as: an output rate-of-change limiter, output limiter, output clamp, output value tracking, and output signal conversion.

Oval gear flowmeter

A positive displacement flowmeter which measures flow rate by turning oval gears.



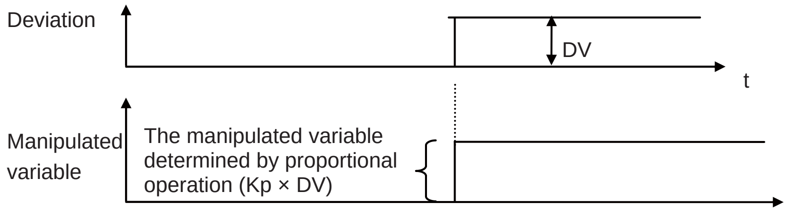
<P>

P operation

Proportional operation
The operation that obtains the manipulated variable in proportion to deviation DV (difference between process variable and setting value)

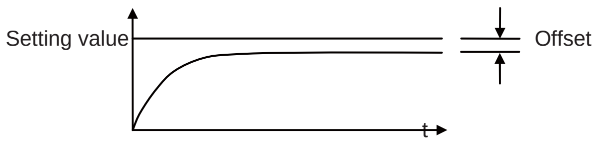
Manipulated variable = Proportional gain Kp × Deviation DV

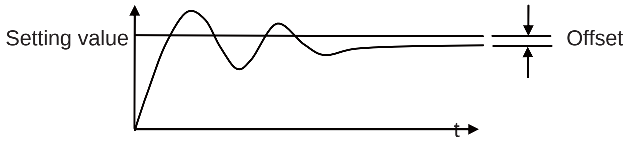
- Operation for step deviation



When proportional gain Kp is relatively smaller	Control operation becomes slower
When proportional gain Kp is relatively bigger	Control operation becomes faster and easy to cause hunting

The error to setting value is called offset. Offset will occur in proportional control action.





P&I flow chart

Piping and instrumentation flow chart which shows entire control system and describes such as piping, detector, final control element, controller in symbols.

PID operation

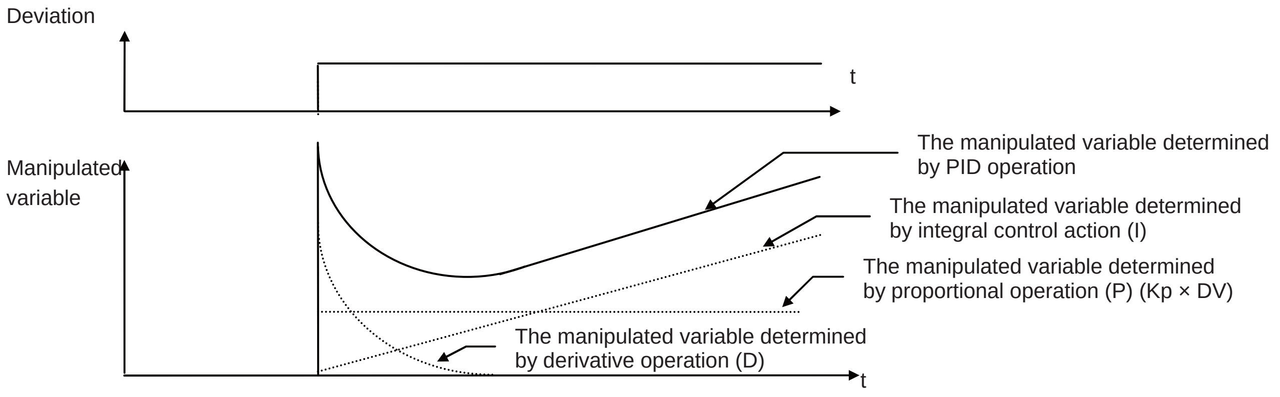
This is the control operation which operates and outputs the manipulated variable (MV) to have the process variable (PV) approach the setting value (SV) rapidly and correctly by combining P control action, I control action and D control action.
Besides, if P, I, D operation are not all included in the control, it is called P control or PI control according to the control action included.
PI operation is mainly for flow rate control, pressure control, temperature control. PID operation is mainly for temperature control.

QnPH/QnPRH of PLC velocity type process variable derivation PID expression

	Direct action	Reverse action
Deviation DVn	DVn = PVn − SVn	DVn = SVn − PVn
Output variation ΔMV	$\Delta MV = \underbrace{\underline{Kp}}_{\text{Gain}} \times \{ \underbrace{(DVn - DVn\text{-}1)}_{\text{Proportional}} + \underbrace{\frac{CT}{Ti} \times DVn}_{\text{Integral}} + \underbrace{\underline{Bn}}_{\text{Derivative}} \}$ Proportional, integral and derivative term of ΔMV are as follows. (Adding proportional, integral and derivative term becomes PID operational expression as shown above.) • Proportional term : $\Delta MV = Kp \times (DVn — DVn\text{-}1)$ • Integral term : $\Delta MV = Kp \times \frac{CT}{Ti} \times DVn$ • Derivative term : $\Delta MV = Kp \times Bn$ (Bn as follows)	
Bn	$Bn = Bn\text{-}1 + \frac{Md \times Td}{Md \times CT + Td} \times \{ (PVn - 2PVn\text{-}1 + PVn\text{-}2) - \frac{CT \times Bn\text{-}1}{Td} \}$	$Bn = Bn\text{-}1 + \frac{Md \times Td}{Md \times CT + Td} \times \{ - (PVn - 2PVn\text{-}1 + PVn\text{-}2) - \frac{CT \times Bn\text{-}1}{Td} \}$

Kp: Gain Ti: Integral time Td: Derivative time Md: Derivative gain CT: Control cycle
DVn: Deviation DVn-1: Previous deviation value PVn: Process variable PVn-1: Previous process value
PVn-2: Process value before last

- Operation for step deviation



PH

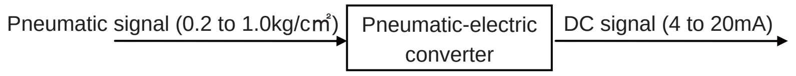
→ High alarm

PL

→ Low alarm

Pneumatic-electric converter

A converter which converts a standardized signal (pneumatic signal) to a standardized signal (electrical signal). Pneumatic-electric transducer



⇔ Electro-pneumatic converter

Piping and instrumentation flow chart

→P& flow chart

Position type PID control

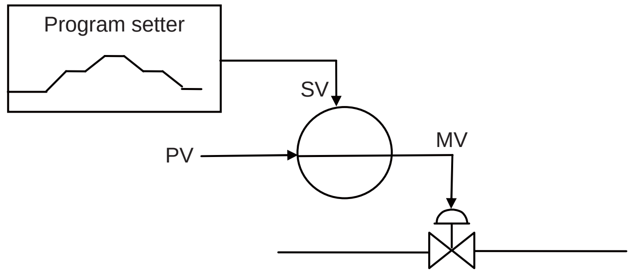
Position type PID control is an operational method to find manipulated variable (MV) from the difference (deviation) between PID operational method setting value (SV) and process variable (PV). On the other hand, velocity type PID Control is an operational method to find a change volume of manipulated variable (ΔMV) from deviation.
→ Velocity type PID Control

Pressure bias

Temperature/pressure correction operation is executed with absolute unit (absolute temperature, absolute pressure).
Pressure bias is the correction values for converting design pressure/measured pressure to absolute pressure.

Program Control

It is a control method to change the setting value by the pre-set program.
It is used for such as temperature control.
It needs to combine the program setter and PID control for using.



Process control

To adjust or control the variables which influence the operation status of industrial processes to meet the specified setting value.

Proportional operation/Proportional gain

→ P operation

Proportional band

In proportional activity, input variation range (%) against the change of output effective variation range from 0% to 100%. For PLC, proportional gain Kp is applied, not proportional band.
100/Proportional gain Kp = Proportional band
→Proportional gain

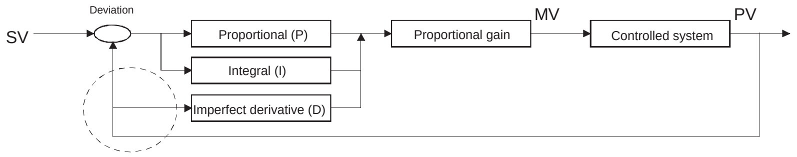
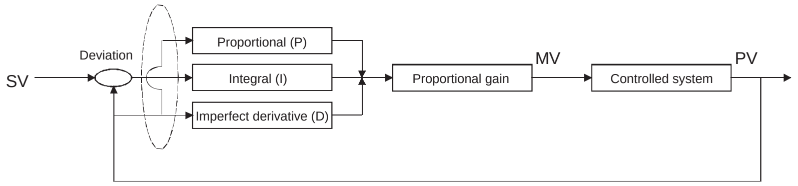
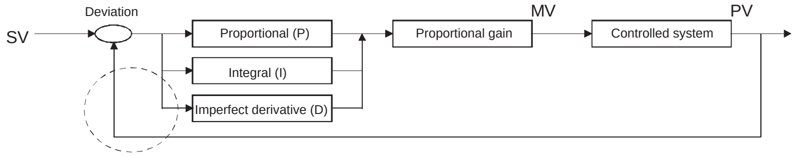
Pulse input module

Input module which counts metric pulse signal from flowmeter.

PV

Process variable

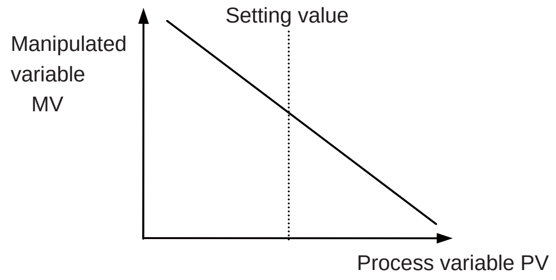
PV-derivative type PID/PV-proportional and -derivative type (I-PD control)/Deviation derivative type PID

Type	Block diagram	Contents	Tag access FB Tag FB
PV-derivative type PID (PI-D control)		In the deviation derivative type shown in this table, there is a problem which MV is rapidly changed when SV is rapidly changed because the influence of its derivative action is too large. Therefore using PV on the deviation enables to avoid the influence of the sudden change of setting value. PV-derivative type is a control method which uses PV to derivative term without using deviation to reduce the drastic change of MV by derivative action in deviation change caused by setting value change. It is called PV-derivative type PID or PI-D control.	P_PID P_2PID (α=0,β=1) M_PID M_2PID (α=0,β=1)
PV-proportional and -derivative type (I-PD control)		In comparison with PV-derivative type, I-PD type uses PV on proportional item in addition to derivative item. This control is also applicable to the situation when the setting value is changed, rapid change to final control element and system are expected to be avoided, and also slow response is preferred. However, a response to the SV change becomes late.	P_IPD P_2PID (α=1,β=1) M_IPD M_2PID (α=1,β=1)
Deviation derivative type		Since target tracking performance of SV change is high, applied for program control and secondary loop of cascade control.	P_2PID (α=0,β=0) M_2PID (α=0,β=0)

<R>

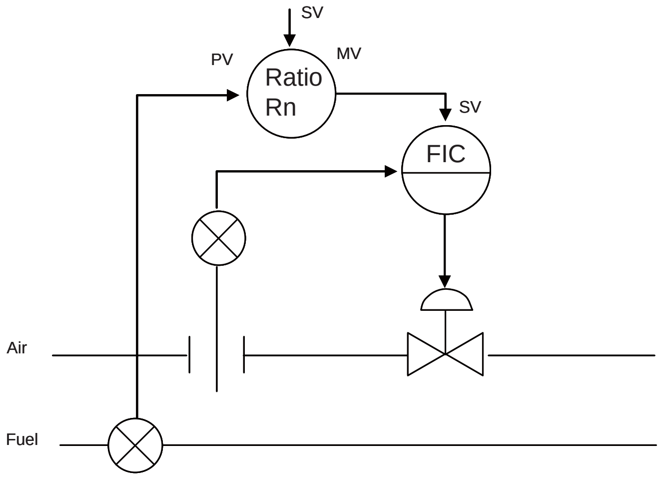
Reverse action

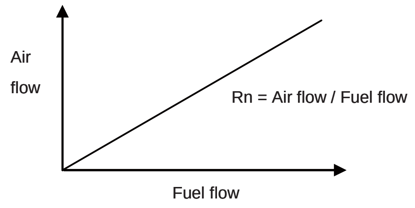
In PID control, an activity to increase manipulated variable (MV) against decrease of process variable (PV) compared to setting value (SV).
(Example: heater)
⇔ Direct action

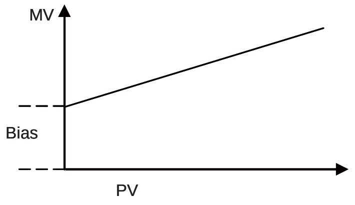


Ratio Control

This control holds the proportional relation between more than 2 variables, such as a control that SV changes in a constant ratio to other variables.
(Example) Air - fuel ratio control

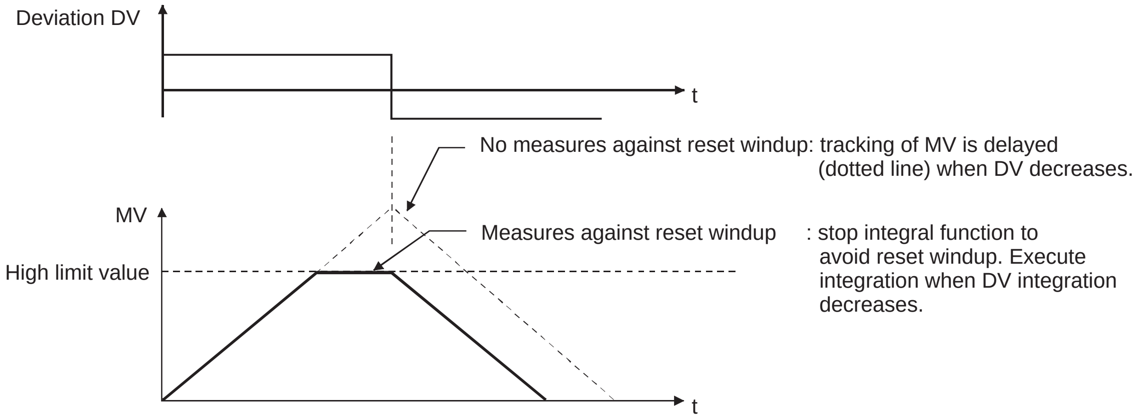






Reset Windup

Reset windup is a problem that deviation is accumulated continuously when an integral element exceeds saturation limit in the case of excessive deviation. Also called Integral windup.
In order to reset the value to the high/low limit value when MV exceeds high/low limit, and to response immediately when the deviation is inverted, a measure against reset windup needs to be implemented to stop the integral action toward the exceeded direction when the value is exceeded the specific limit. (Bold line in the diagram below)
A measure against reset windup is implemented for QnPHCPU, and QnPRHCPU.



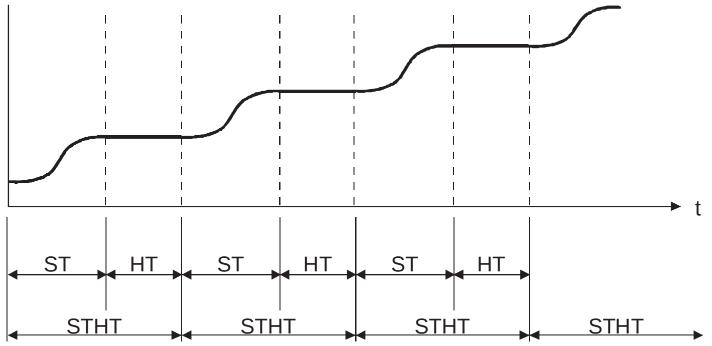
No measures against reset windup: tracking of MV is delayed (dotted line) when DV decreases.

Measures against reset windup : stop integral function to avoid reset windup. Execute integration when DV integration decreases.

<S>

Sample PI Control

When PID control is applied on a system whose dead time is long, MV will be continuously updated before MV effect is confirmed.
Sample PI control executes only for a control cycle in every control cycle, and then holds the output after that.



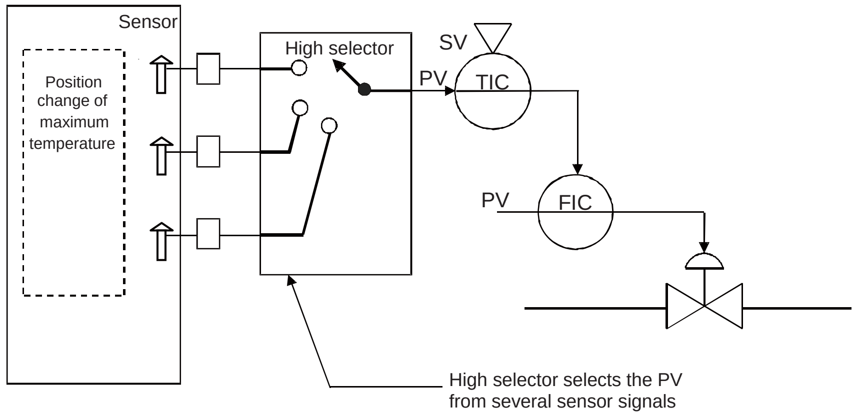
ST: Operating time, HT: Hold time=STHT-ST, STHT: Sampling interval

Secondary loop

Secondary loop of cascade control
→ CASCADE control

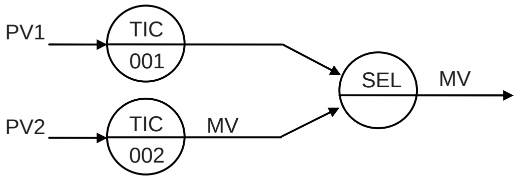
Selection Control

(1) This is a control method that selects the necessary signals (high selector, low selector, intermediate value selector, etc.) among multiple sensor signals or operation signals to control the system.
(Example) When the highest temperature position changes, the control is performed by selecting the highest temperature among two or more measurement points.

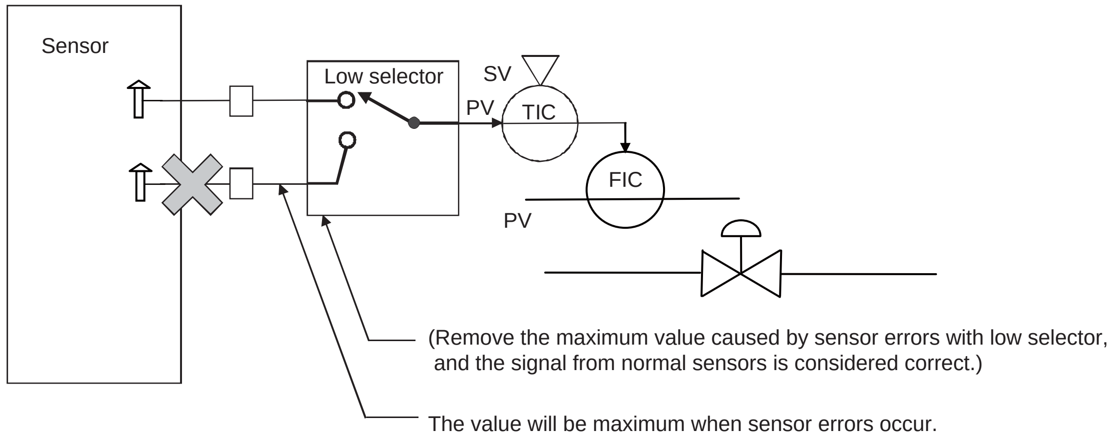


High selector selects the PV from several sensor signals

(2) The process in which the value is selected at the output side is called override control.



(3) The redundancy of the system is realized by installing two or more sensors to avoid sensor wire break and trouble and selecting the normal one.
(Connect multiple sensors, combine the low, high and intermediate selectors according to the status (when burnout occurs, the signal of sensor is the maximum or the minimum.) when burnout occurs and obtain normal sensor signals.)
(Example) In case that the input signal from the sensor becomes maximal when sensor errors such as wire break occur.



(Remove the maximum value caused by sensor errors with low selector, and the signal from normal sensors is considered correct.)

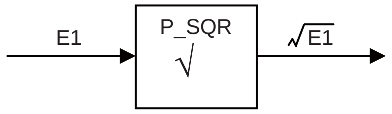
The value will be maximum when sensor errors occur.

SIMULATION mode

A mode to perform a simulation test using simulation I/O data instead of actual I/O data.
→ I/O mode

Square Root Extraction

√ (root) calculation function. When measuring flow rate through differential pressure of orifice or venturi tube, the signal which is obtained from sensor has square characteristics. This control linearizes the signals. Process FB "P_SQR" is applied to this function.
→ Orifice, Differential pressure



Solenoid valve

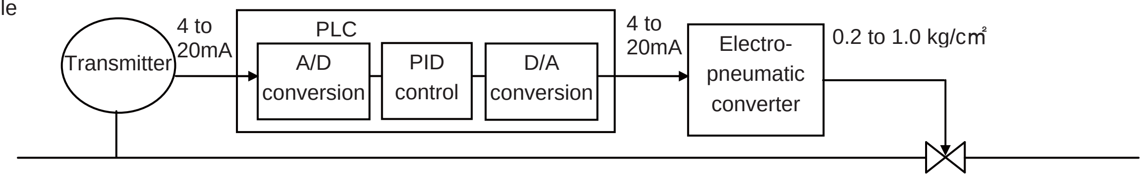
A solenoid valve opens/closes a valve with electromagnetic force.

Standardized signal

An input/output process control signal (such as process variable signal or operation signal) whose range is standardized.
Even in a minimum process variable limit, a failure or a disconnection of transmitter or converter can be detected by applying 4mA current.

Signal type	Signal range
Current signal	4 to 20mA DC
Voltage signal	1～5V DC
Pneumatic signal	0.2 to 1.0kg/㎠

Example



Status tag

A tag which contains a faceplate with the ON/OFF control function such as start/stop of electric motor or open/close of solenoid valve.
→ Loop tag

SV

Setting value

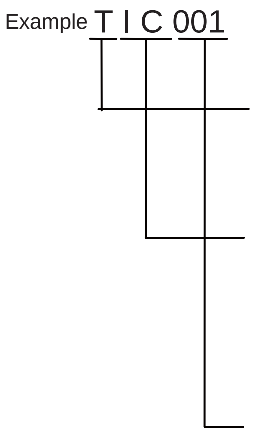
<T>

Tag

Tags for identification attached to process control equipment.

Tag number

Tag numbers are unique management numbers used for identifying process control equipment. A tag number consists of the variable symbols, function symbols, and individual numbers. Prescribed by JIS Z8204.



Variable symbol
- Indicates process variable.
  T indicates temperature for this example.

Function symbol
- Indicates functions such as instruction, correction, and alarm.
  For two or more functions, indicate in the order of: I, R, C, T, Q, S, Z.
  IC indicates instruction and correction for this example.

Individual number
- A number to identify measurement and control loop.
  001 indicates No. 001 for this example.

	Variable symbol	Function symbol
A		Alarm
C		Correction
D	Density or specific gravity	
F	Instantaneous flow rate	
G	Position or length	
H	Manual operation	
I		Instruction
K	Time	
L	Level such as liquid level	
M	Humidity or moisture	
P	Pressure or vacuum	
Q	Quality (Ex. composition, concentration, conductivity)	Integration
R	Radiation	Record
S	Speed, rotating speed, frequency	Switch
T	Temperature	Transmission
V	Viscosity	
W	Weight or force	
Z		Safety or emergency

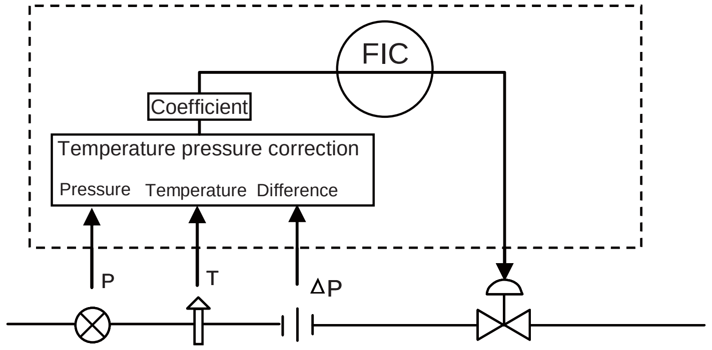
This example indicates the loop number 001 which instructs and corrects temperature.

Temperature bias

Temperature/pressure correction operation uses absolute unit (absolute temperature, absolute pressure).
Temperature bias is a corrected value to convert design temperature/measured temperature to absolute temperature.

Temperature/Pressure Correction

When the fluid conditions (temperature, pressure), of which the differential pressure measured by equipment which has diagram such as orifice, are not the same as the design conditions, it shall be corrected.
Correction shall be performed by process variable to multiply the temperature/pressure correction coefficient.
In addition, when equipment with diaphragm such as orifice is used, the obtained value is square of the flow rate. So that extraction of square root shall be applied.



$$\text{Flow rate Q} = \text{Temperature correction} \times \text{Pressure correction} \times \text{Coefficient}\sqrt{\text{Differential pressure}}$$

$$= \sqrt{\frac{\text{Measured pressure}}{\text{Design pressure}}} \times \sqrt{\frac{\text{Design temperature}}{\text{Measured temperature}}} \times \text{Coefficient}\sqrt{\text{Differential pressure}}$$

$$= \sqrt{\frac{P}{P'}} \times \sqrt{\frac{T'}{T}} \times \text{Coefficient}\sqrt{\Delta P}$$

T,T': Absolute temperature
P,P': Absolute pressure

(Reference) Example of gas temperature/pressure correction
Q: Measured flow rate (process variable of flowmeter), T: Design temperature (°C), T1: Measured temperature (°C),
P: Design pressure (kPa), P1: Measured pressure (kPa)
When: temperature/pressure correction is {(T + 273.15) / (T + 273.15)} × {(P1 + 101.3) / (P + 101.3)},
formulas to calculate Q1= actual flow rate (after temperature/pressure correction), are as shown below.

Type	Output characteristic of flowmeter	Correction formula
differential pressure type flowmeter (orifice, venturi tube)	Square-root characteristic	$Q1 = \sqrt{Q} \times \sqrt{\text{Temperature/pressure correction}}$
	Linear characteristic (when square-root characteristic is converted to linear characteristic)	$Q1 = Q \times \sqrt{\text{Temperature/pressure correction}}$
Area flowmeter	Linear characteristic	$Q1 = Q \times \sqrt{\text{Temperature/pressure correction}}$
Vortex flowmeter (Karman vortex)	Linear characteristic	Q1 = Q × Temperature/pressure correction

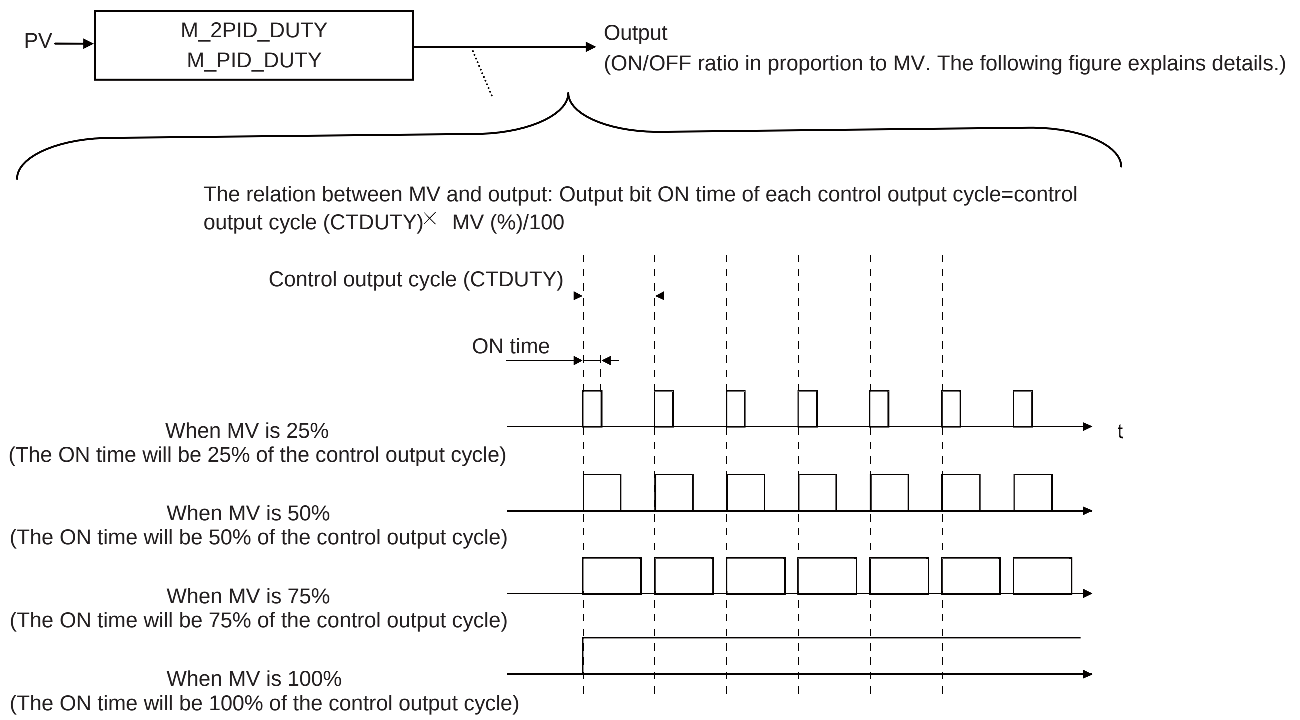
Thermometer

A device which measures temperature. Representative types are described in the following table. Many thermometers are used in process control.

Contact type	Thermocouple (B, S, R, K, E, J)	-180°C to 1550°C (Temperature range as reference)
	Resistance bulb (pt, 3-wire type, 4-wire type)	-180°C to 500°C
	Thermistor	-50°C to 200°C
Non-contact type	Optical pyrometer	700°C to 3000°C
	Radiation thermometer	-50°C to 4000°C

Time Proportioning Control

Time proportioning control changes the ON/OFF ratio in proportion to the PID operation result, controls such as heater.



The relation between MV and output: Output bit ON time of each control output cycle=control output cycle (CTDUTY)× MV (%)/100

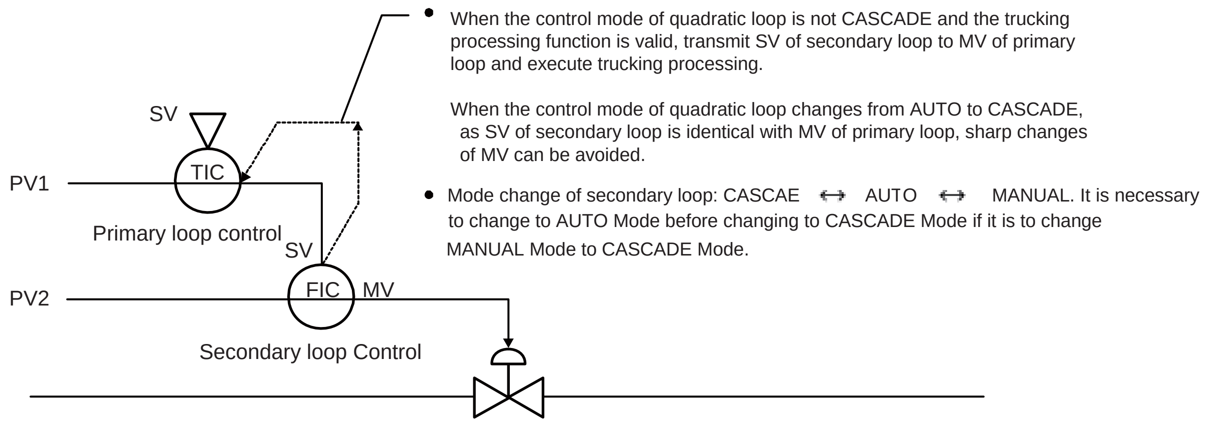
Tracking

Tracking is a function to follow-up a specific signal to accord with another signal.

The following explains an example of using the tracking function.

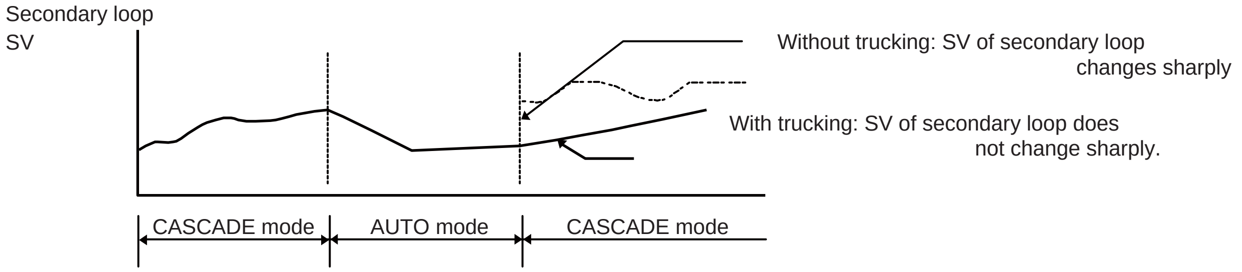
(1) Tracking example of cascade loop

For the control loop which composes cascade loop, if the control mode switching of secondary loop is executed, the SV of secondary loop shall be transmit to MV of primary loop, in order to prevent sharp changes of SV.



- When the control mode of quadratic loop is not CASCADE and the trucking processing function is valid, transmit SV of secondary loop to MV of primary loop and execute trucking processing.

  When the control mode of quadratic loop changes from AUTO to CASCADE, as SV of secondary loop is identical with MV of primary loop, sharp changes of MV can be avoided.
- Mode change of secondary loop: CASCAE ↔ AUTO ↔ MANUAL. It is necessary to change to AUTO Mode before changing to CASCADE Mode if it is to change MANUAL Mode to CASCADE Mode.



Tuning trend

A trend screen which displays a tuning status of loop in real time. It displays PV, SV, and MV.

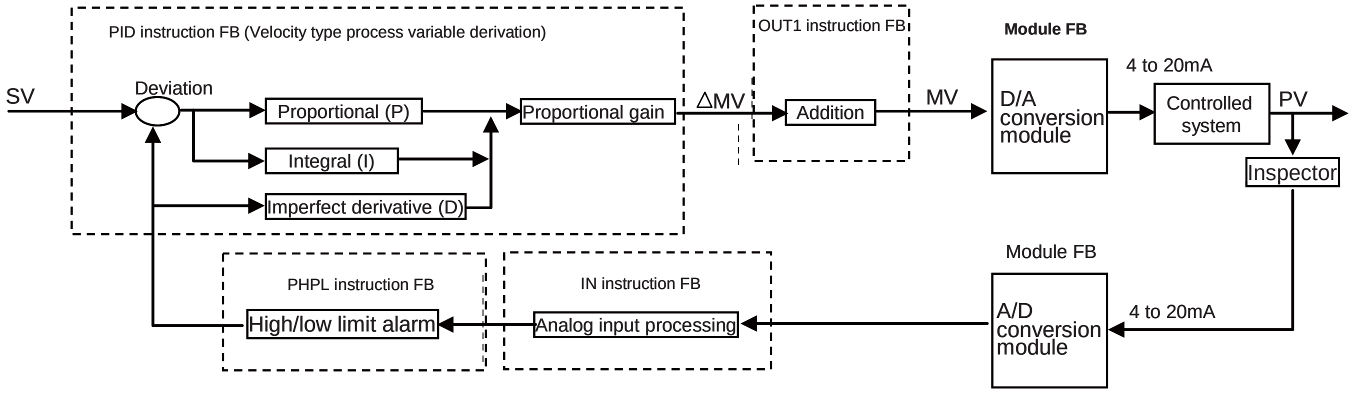
<V>

Velocity type PID Control

Velocity type PID control is an operation method to obtain difference (ΔMV) between PV and SV. ΔMV from PID operation is added in the OUT processing and the manipulated variable MV is output because the velocity type is a computing method to obtain operation difference ΔMV by comparing to values of previous operation. Compared to PID position type, velocity type is more convenient in operation of bumpless manual-auto switching, prevention of reset windup, complicated control and slow change when gain is changed. Hereby the velocity type has become the mainstream choice.

Current MV = the previous MV + current change ΔMV

Example of velocity type process variable derivation PID using PLC FB instruction.



<W>

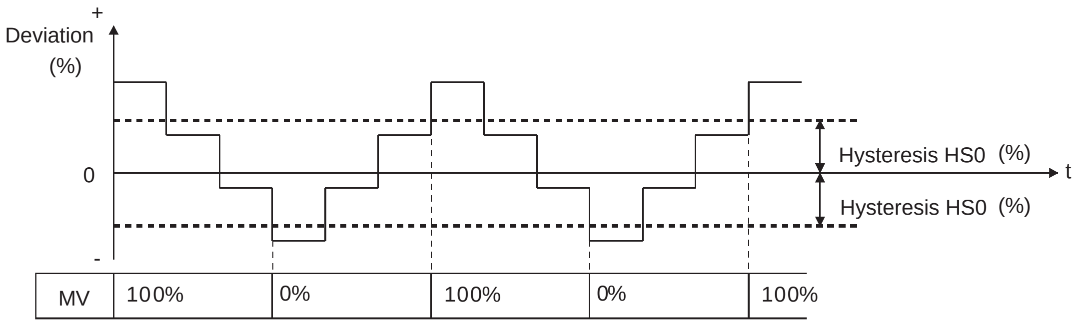
Watchdog timer error alarm

An alarm which occurs when a status answer back time takes longer than the specified time after the control command such as open/close is output. A disconnection of control line, control power OFF, and contactor failure are possible causes.

<2>

2 position ON/OFF Control

This is a method that outputs 2 steps of MV signals for deviation to control the system.



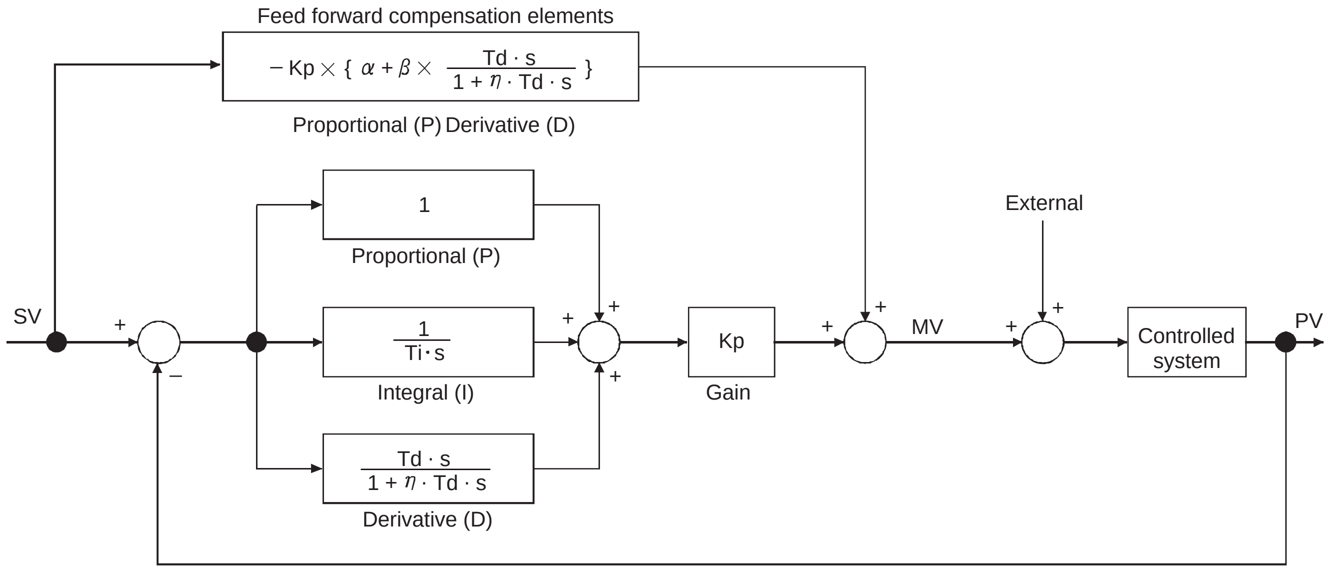
Direct action DV (%) = PV (%) – SV (%)
Reverse action DV (%) = SV (%) – PV (%)

SV (%) = {(SV - low limit of engineering variable) / (high limit of engineering variable - low limit of engineering variable)} × 100
PV (%) = {(PV - low limit of engineering variable) / (high limit of engineering variable - low limit of engineering variable)} × 100
Hysteresis (%) is the percentage to (High limit of engineering variable - Low limit of engineering variable).

2-degree-of-freedom PID Control

2-degree-of-freedom PID control is a control method which can optimize PID constants for both disturbance response and target tracking as compare to the former PID control. 2-degree-of freedom parameters, α and β, are used for this control. (When both α and β are 0, the control is the same as the former PID control.)

* In conventional PID control, the optimum PID constants that correspond to SV change for target tracking and disturbance response differ. This causes an antinomy such as when the optimum value is set for one side, the value of the other side is not optimum.



QnPH/QnPRH PLC 2-degree-of-freedom PID operation expression

	Direct action	Reverse action
Deviation DVn	DVn = PVn - SVn	DVn = SVn - PVn
Output variation ΔMV	$\Delta MV = Kp \times \{ (1-\alpha) \times (DVn - DVn\text{-}1) + \frac{CT}{Ti} \times DVn + (1 - \beta) \times Bn + \alpha \times Cn + \beta \times Dn \}$	
Bn	$Bn = Bn\text{-}1 + \frac{Md \times Td}{Md \times CT + Td} \times \{ (DVn - 2DVn\text{-}1 + DVn\text{-}2) - \frac{CT \times Bn\text{-}1}{Td} \}$	
Cn	PVn - PVn-1	—(PVn — PVn-1)
Dn	$Dn = Dn\text{-}1 + \frac{Md \times Td}{Md \times CT + Td} \times \{ (PVn - 2PVn\text{-}1 + PVn\text{-}2) - \frac{CT \times Dn\text{-}1}{Td} \}$	$Dn = Dn\text{-}1 + \frac{Md \times Td}{Md \times CT + Td} \times \{ - (PVn - 2PVn\text{-}1 + PVn\text{-}2) - \frac{CT \times Dn\text{-}1}{Td} \}$

Kp: Gain Ti: Integral time Td: Derivative time Md: Derivative gain CT: Control cycle
DVn: Deviation DVn-1: Previous deviation value DVn-2: Deviation value before last PVn: Process variable
PVn-1: Previous process variable PVn-2: Process variable before last
α: 2-degree-of-freedom PID parameter (feedforward proportional)
β: 2-degree-of-freedom PID parameter (feedforward derivative)

When 2-degree-of-freedom PID control is applied, characteristics can be changed by adjusting α and β after the constants of P, I and D are determined.

When α = 0, β = 0: Deviation PID

Derivative action is effective to deviation (difference between setting value and process variable), so that the target tracking performance corresponding to the change of setting value will be better.

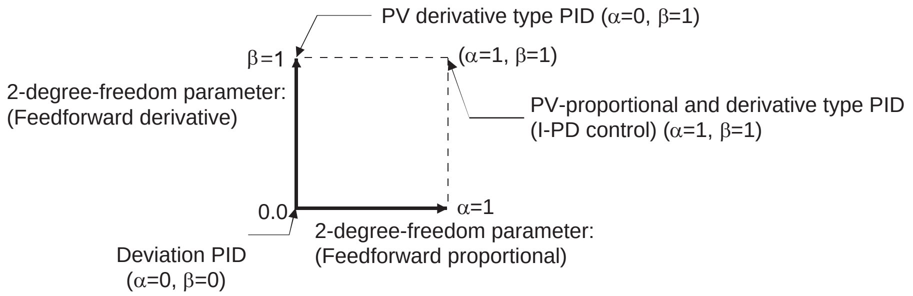
When α = 0, β = 1: PV-derivative type (Basic PID control in PX Developer)

As derivation operation is effective to process variable, in comparison with derivative PID, disturbance response will be accelerated. On the other hand, the target tracking performance corresponding to the change of setting value will decrease.

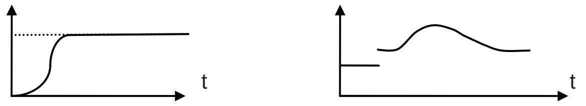
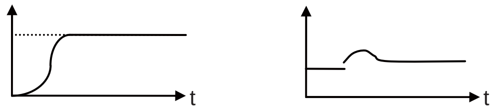
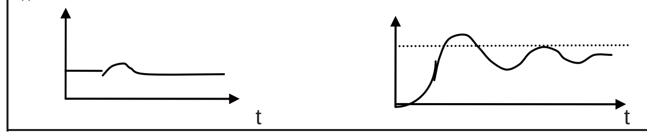
When α = 1, β = 1: PV-proportional and derivative type (I-PD control in PX Developer)

As both proportional and derivative action is effective to process variable and integral control action is effective to deviation (difference between setting value and process variable), the target tracking performance, in comparison with PV-derivative PID, will decrease corresponding to the change of setting value.

This control is also applicable when over shoot is not allowed and when the setting value is changed, and rapid change to final control element and system are expected to be avoided, and also slow response is preferred because the manipulated variable does not change suddenly in change of setting value



- The response of traditional PID control (1-degree-of-freedom) and 2-degree-of-freedom PID control

Traditional PID control (1-degree-of-freedom)	2-degree-of-freedom PID control
Optimum value for setting value change ○ ---→ Not optimum for disturbance ×  	Optimum value for setting value change ○ ---→ Optimum value for disturbance ○ 
Optimum value for disturbance ○ ---→ Not optimum for setting value change ×  	

- Adjustment method of 2-degree-of-freedom PID control
  (1) Calculate PID constant by using auto tuning.
  (2) Fine tune PID constants (basic parameters of PID: Kp, Ti, Td) to optimize the response performance for disturbance if necessary.
    - Proportional gain K_P
      If Kp KP is tuned down, the manipulated variable will become smaller, and it will take a longer time to be stable.
      If Kp is tuned up, the manipulated variable will become bigger, there may be oscillation in response due to the enhancement of compensation operation.
    - Integral time Ti
      If Ti is tuned down, integral control action will be enhanced, and the response will sometimes be oscillation. (Oscillation period becomes longer.)
      If Ti is tuned up, integral effect will be smaller, and it will take a long time to be stable.
    - Derivative time Td
      If Td is tuned down, derivative effect will be smaller, and derivative will only effect for a short period of time.
      If Td is tuned up, derivative effect will be bigger, short-period oscillation will occur, and sometimes the system will be quite unstable.
  (3) Hold the optimum disturbance response, while adjusting 2-degree-offreedom parameter (α, β) to optimize the target tracking response.
    - If α is tuned up, the manipulated variable in relation to setting value change will become smaller, and it will take a longer time to be stable.

If α is tuned down, the manipulated variable in relation to setting value change will be bigger, and response will sometimes be oscillatory due to the enhancement of compensation operation.

- If β is tuned up, the derivative effect corresponding to setting value change will be smaller, and derivation will only effect for a short period of time.
  If β is tuned down, the derivative effect in relation to setting value change will become bigger, and short-time period oscillation will occur, sometimes the system will be unstable.

The response performance corresponding to setting value change when α is changed is as follows.

Quick: α = 0, Medium: α = 0.65, Slow: α = 1

(Here β =1. the derivative action corresponding to setting value change makes manipulated variable change sharply (kick), and shocks on the final control element and system. Therefore when β =1, usually treat the derivative action in relation to setting value change as invalid.)

2-degree-of-freedom advanced PID control

→2-degree-of-freedom PID control

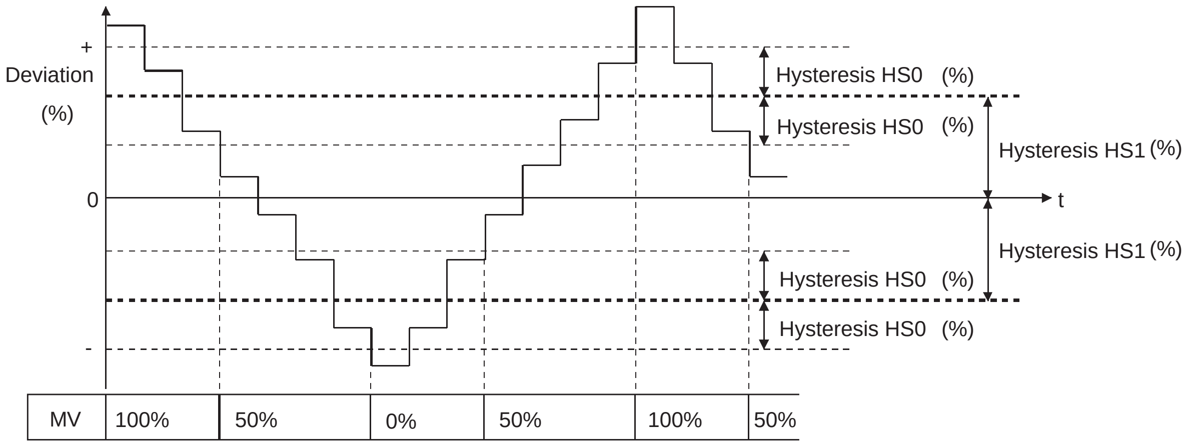
2-degree-of-freedom advanced PID control tag FB (M_2PIDH_) is an advanced tag FB by adding functions such as: MV compensation, PV compensation, temperature/pressure correction, tag stop, PV tracking, preset MV, MV rate-of-change limiter, and cascade direct to 2-degree-of-freedom PID control tag FB (M_2PID). From simple controls to advanced controls such as variable gain PID control, compensation operations, correction operation, and feedforward control, this tag FB can be used in a wide range of controls.
The following table shows the main internal functions. (For detailed block diagram, refer to section 4.1.)

Processing input area	PID operation area	Operation output area
Range check (Sensor error detection)	SV rate-of-change limiter	ΔMV compensation
Input limiter	SV high/low limiter	ΔMV variable gain compensation
Digital filter	PV tracking (PV → SV)	ΔMV integration
Temperature/pressure correction	Deviation check	MV compensation
Square root extraction	2-degree-of-freedom PID operation	Preset MV output
Function generator	Disable alarm detection	MV hold
First order lag filter	Auto tuning	MV tracking
PV compensation	Integration stop switch	MV rate-of-change high/low limiter
ΔPV compensation	Deviation stop switch	Anti-reset windup
Processing at sensor error	Loop stop/execution	MV reverse
High high limit/high limit /low limit/low low limit check		Output conversion
PV rate-of-change check		Disable alarm detection
Disable alarm detection		Processing at sensor error
		Cascade direct

<3>

3 position ON/OFF Control

This is a control method that outputs 3 steps of MV signals for deviation to control the system.



Direct action : DV (%) = PV (%) – SV (%)
Reverse action : DV (%) = SV (%) – PV (%)

SV (%) = {(SV - low limit of engineering variable) / (high limit of engineering variable - low limit of engineering variable)} × 100
PV (%) = {(PV - low limit of engineering variable) / (high limit of engineering variable - low limit of engineering variable)} × 100
Hysteresis (%) is expressed by percentage corresponding to (High limit value of engineering variable – Low limit value of engineering variable).

Empowering Industries CC-Link CC-Link IE MELSOFT

Mitsubishi Electric Programmable Controllers

Precautions before use

This publication explains the typical features and functions of the products herein and does not provide restrictions or other information related to usage and module combinations. Before using the products, always read the product user manuals. Mitsubishi Electric will not be held liable for damage caused by factors found not to be the cause of Mitsubishi Electric; opportunity loss or lost profits caused by faults in Mitsubishi Electric products; damage, secondary damage, or accident compensation, whether foreseeable or not, caused by special factors; damage to products other than Mitsubishi Electric products; or any other duties.

⚠ For safe use

- To use the products given in this publication properly, always read the relevant manuals before beginning operation.
- The products have been manufactured as general-purpose parts for general industries, and are not designed or manufactured to be incorporated in a device or system used in purposes related to human life.
- Before using the products for special purposes such as nuclear power, electric power, aerospace, medicine or passenger-carrying vehicles, consult with Mitsubishi Electric.
- The products have been manufactured under strict quality control. However, when installing the products where major accidents or losses could occur if the products fail, install appropriate backup or fail-safe functions in the system.

Country/Region	Sales office	Tel/Fax
USA	MITSUBISHI ELECTRIC AUTOMATION, INC. 500 Corporate Woods Parkway, Vernon Hills, IL 60061, U.S.A.	Tel : +1-847-478-2100 Fax : +1-847-478-2253
Mexico	MITSUBISHI ELECTRIC AUTOMATION, INC. Mexico Branch Boulevard Miguel de Cervantes Saavedra 301, Torre Norte Piso 5, Ampliacion Granada, Miguel Hidalgo, Ciudad de Mexico, Mexico, C.P.115200	Tel : +52-55-3067-7512
Brazil	MITSUBISHI ELECTRIC DO BRASIL COMERCIO E SERVICOS LTDA. Avenida Adelino Cardana, 293, 21 andar, Bethaville, Barueri SP, Brasil	Tel : +55-11-4689-3000 Fax : +55-11-4689-3016
Germany	MITSUBISHI ELECTRIC EUROPE B.V. German Branch Mitsubishi-Electric-Platz 1, 40882 Ratingen, Germany	Tel : +49-2102-486-0 Fax : +49-2102-486-7780
UK	MITSUBISHI ELECTRIC EUROPE B.V. UK Branch Travellers Lane, UK-Hatfield, Hertfordshire, AL10 8XB, U.K.	Tel : +44-1707-28-8780 Fax : +44-1707-27-8695
Ireland	MITSUBISHI ELECTRIC EUROPE B.V. Irish Branch Westgate Business Park, Ballymount, Dublin 24, Ireland	Tel : +353-1-4198800 Fax : +353-1-4198890
Italy	MITSUBISHI ELECTRIC EUROPE B.V. Italian Branch Centro Direzionale Colleoni - Palazzo Sirio, Viale Colleoni 7, 20864 Agrate Brianza (MB), Italy	Tel : +39-039-60531 Fax : +39-039-6053-312
Spain	MITSUBISHI ELECTRIC EUROPE, B.V. Spanish Branch Carretera de Rubi, 76-80-Apdo. 420, E-08190 Sant Cugat del Valles (Barcelona), Spain	Tel : +34-935-65-3131 Fax : +34-935-89-1579
France	MITSUBISHI ELECTRIC EUROPE B.V. French Branch 25, Boulevard des Bouvets, 92741 Nanterre Cedex, France	Tel : +33-1-55-68-55-68 Fax : +33-1-55-68-57-57
Czech Republic	MITSUBISHI ELECTRIC EUROPE B.V. Czech Branch, Prague Office Pekarska 621/7, 155 00 Praha 5, Czech Republic	Tel : +420-255-719-200
Poland	MITSUBISHI ELECTRIC EUROPE B.V. Polish Branch ul. Krakowska 48, 32-083 Balice, Poland	Tel : +48-12-347-65-00
Sweden	MITSUBISHI ELECTRIC EUROPE B.V. (Scandinavia) Hedvig Mollersgata 6, 223 55 Lund, Sweden	Tel : +46-8-625-10-00 Fax : +46-46-39-70-18
Russia	MITSUBISHI ELECTRIC (RUSSIA) LLC St. Petersburg Branch Piskarevsky pr. 2, bld 2, lit "Sch", BC "Benua", office 720; 195027 St. Petersburg, Russia	Tel : +7-812-633-3497 Fax : +7-812-633-3499
Turkey	MITSUBISHI ELECTRIC TURKEY A.S. Umraniye Branch Serifali Mah. Kale Sok. No:41 34775 Umraniye - Istanbul, Turkey	Tel : +90-216-969-2500 Fax : +90-216-661-4447
UAE	MITSUBISHI ELECTRIC EUROPE B.V. Dubai Branch Dubai Silicon Oasis, P.O.BOX 341241, Dubai, U.A.E.	Tel : +971-4-3724716 Fax : +971-4-3724721
South Africa	ADROIT TECHNOLOGIES 20 Waterford Office Park, 189 Witkoppen Road, Fourways, South Africa	Tel : +27-11-658-8100 Fax : +27-11-658-8101
China	MITSUBISHI ELECTRIC AUTOMATION (CHINA) LTD. Mitsubishi Electric Automation Center, No.1386 Hongqiao Road, Shanghai, China	Tel : +86-21-2322-3030 Fax : +86-21-2322-3000
Taiwan	SETSUYO ENTERPRISE CO., LTD. 6F, No.105, Wugong 3rd Road, Wugu District, New Taipei City 24889, Taiwan	Tel : +886-2-2299-2499 Fax : +886-2-2299-2509
Korea	MITSUBISHI ELECTRIC AUTOMATION KOREA CO., LTD. 7F to 9F, Gangseo Hangang Xi-tower A, 401, Yangcheon-ro, Gangseo-Gu, Seoul 07528, Korea	Tel : +82-2-3660-9569 Fax : +82-2-3664-8372
Singapore	MITSUBISHI ELECTRIC ASIA PTE. LTD. 307 Alexandra Road, Mitsubishi Electric Building, Singapore 159943	Tel : +65-6473-2308 Fax : +65-6476-7439
Thailand	MITSUBISHI ELECTRIC FACTORY AUTOMATION (THAILAND) CO., LTD. 12th Floor, SV.City Building, Office Tower 1, No. 896/19 and 20 Rama 3 Road, Kwaeng Bangpongpang, Khet Yannawa, Bangkok 10120, Thailand	Tel : +66-2682-6522-31 Fax : +66-2682-6020
Vietnam	MITSUBISHI ELECTRIC VIETNAM COMPANY LIMITED Unit 01-04, 10th Floor, Vincom Center, 72 Le Thanh Ton Street, District 1, Ho Chi Minh City, Vietnam	Tel : +84-28-3910-5945 Fax : +84-28-3910-5947
Indonesia	PT. MITSUBISHI ELECTRIC INDONESIA Gedung Jaya 8th Floor, JL. MH. Thamrin No.12, Jakarta Pusat 10340, Indonesia	Tel : +62-21-31926461 Fax : +62-21-31923942
India	MITSUBISHI ELECTRIC INDIA PVT. LTD. Pune Branch Emerald House, EL-3, J Block, M.I.D.C., Bhosari, Pune-411026, Maharashtra, India	Tel : +91-20-2710-2000 Fax : +91-20-2710-2100
Australia	MITSUBISHI ELECTRIC AUSTRALIA PTY. LTD. 348 Victoria Road, P.O. Box 11, Rydalmere, N.S.W 2116, Australia	Tel : +61-2-9684-7777 Fax : +61-2-9684-7245

MITSUBISHI ELECTRIC CORPORATION
HEAD OFFICE: TOKYO BLDG., 2-7-3, MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
www.MitsubishiElectric.com

L(NA)08200E-C 1103(IP)

New publication, effective March. 2011.
Specifications are subject to change without notice.