Function Blocks Overview

IM 33K03E21-50E

IM 33K03E21-50E
5th Edition

Introduction

The integrated production control system CENTUM VP is the products of distributed control system (DCS) for large and medium-small scale process control.

This manual is written for use as a part of reference manual describing the functional details of CENTUM VP. This reference manual focuses on the function block details. You can read the required chapters when you need the details of function blocks.

In this manual, FFCS, FFCS-L, and FFCS-V are all referred to as FFCS series. If the individual types of FCS need to be particularly mentioned, the FCSs may be described respectively with their particular types.

This manual consists of the following chapters:

- 1. Structure of a Function Block
  This chapter describes an overview of structural components and the basic structure of function block.
- 2. I/O Connection
  This chapter describes the input and output terminals of function blocks, the connectable destinations and the connection methods.
- 3. Input Processing
  This chapter describes the input processing methods common to all the function blocks.
- 4. Output Processing
  This chapter describes the output processing common to all the regulatory control blocks and calculation blocks.
- 5. Alarm Processing – FCS
  This chapter describes the FCS alarm processing features on detecting alarms and notifying the alarms.
- 6. Block Mode and Status
  This chapter describes the block mode and the block status of the function blocks.
- 7. Process Timing
  This chapter describes the process timings of the regulatory control blocks, the calculation blocks, the sequence control blocks and the SEBOL blocks.
- 8. CENTUM-XL Compatibility
  This chapter describes the CENTUM-XL compatibility which allows the function blocks in CENTUM VP to run the same as those in CENTUM-XL.

Media No. IM 33K03E21-50E (DVD) 5th Edition : Aug. 2014 (YK)

CENTUM VP Document Map

Installation

- CENTUM VP Installation — IM 33K01C10-50E
- License Management — IM 33K01C20-50E
- CENTUM VP Security Guide — IM 33K01C30-50E

Engineering

- Field Control Stations Reference — IM 33K03E10-50E
- Function Blocks Overview — IM 33K03E21-50E
- Function Blocks Reference Vol.1 — IM 33K03E22-50E
- Function Blocks Reference Vol.2 — IM 33K03E23-50E
- Function Blocks Reference Vol. 3 — IM 33K03E24-50E
- Human Interface Stations Reference Vol. 1 — IM 33K03F21-50E
- Human Interface Stations Reference Vol. 2 — IM 33K03F22-50E
- Engineering Reference Vol. 1 — IM 33K03G21-50E
- Engineering Reference Vol. 2 — IM 33K03G22-50E
- Engineering Reference Vol. 3 — IM 33K03G23-50E
- Consolidated Alarm Management Software Reference — IM 33K03H20-50E
- Batch Management System Reference — IM 33K03J10-50E
- SEBOL Reference — IM 33K03K10-50E
- Communication with Subsystems Using RIO — IM 33K03L10-50E
- Communication with Subsystems Using FIO — IM 33K03L20-50E
- Communication Devices Reference — IM 33K03M10-50E
- Optional Functions Reference — IM 33K03N10-50E
- Communication with PROFIBUS Systems — IM 33K03P10-50E
- Compliance with FDA: 21CFR Part 11 — IM 33K03Q10-50E
- Generic Subsystem Gateway Reference — IM 33K03R10-50E
- System Integration OPC Station Reference — IM 33K03R20-50E
- Unified Gateway Station Reference — IM 33K03R30-50E
- Vnet/IP — IM 33K10A10-50E
- Engineering Test Guide — IM 33K10H20-50E
- Engineering Tutorial — IM 33K02E10-50E
- APCS — IM 33K15U10-50E
- CENTUM Data Access Library — IM 33K15P10-50E
- FOUNDATION fieldbus Reference — IM 33K20T10-50E
- FOUNDATION fieldbus Engineering Guide — IM 33K20T20-50E

Operation and Monitoring

- HIS Operation — IM 33K02D10-50E
- Operating Messages — IM 33K02D20-50E

Hardware

- Peripherals — IM 33K50C10-50E
- Field Control Stations — IM 33K50E40-50E
- Input & Output Modules — IM 33K50G10-50E
- Turbomachinery I/O Modules — IM 33K10U10-50E
- Communication Devices — IM 33K50D10-50E
- Migrated-FCS(FIO) — IM 33K55W10-50E

Safety Precautions for Use

■ Safety, Protection, and Modification of the Product

- To protect the system controlled by the Product and the Product itself and to ensure safe operation, please observe the safety precautions described in this Manual. Yokogawa Electric Corporation ("YOKOGAWA") assumes no liability for safety if users fail to observe the safety precautions and instructions when operating the Product.
- If the Product is used in a manner not specified in the User's Manuals, the protection provided by the Product may be impaired.
- If any protection or safety circuit is required for the system controlled by the Product or for the Product itself, please install it externally.
- Use only spare parts that are approved by YOKOGAWA when replacing parts or consumables of the Product.
- Do not use the Product and its accessories such as power cords on devices that are not approved by YOKOGAWA. Do not use the Product and its accessories for any purpose other than as intended by YOKOGAWA.
- Modification of the Product is strictly prohibited.
- The following symbols are used in the Product and User's Manuals to indicate the accompanying safety precautions:

  ⚠ Indicates that caution is required for operation. This symbol is labeled on the Product to refer the user to the User's Manuals for necessary actions or behaviors in order to protect the operator and the equipment against dangers such as electrical shocks. In the User's Manuals, you will find the precautions necessary to prevent physical injury or death, which may be caused by accidents, such as electrical shocks resulting from operational mistakes.

  ⏚ Identifies a protective conductor terminal. Before using the Product, you must ground the protective conductor terminal to avoid electric shock.

  ⏚ Identifies a functional grounding terminal. A terminal marked "FG" also has the same function. This terminal is used for grounding other than protective grounding. Before using the Product, you must ground this terminal.

  ∼ Indicates an AC supply.

  ⎓ Indicates a DC supply.

  | Indicates the ON position of a power on/off switch.

  ○ Indicates the OFF position of a power on/off switch.

■ Notes on Handling User's Manuals

- Hand over the User's Manuals to your end users so that they can keep the User's Manuals on hand for convenient reference.
- Thoroughly read and understand the information in the User's Manuals before using the Product.
- For the avoidance of doubt, the purpose of the User's Manuals is not to warrant that the Product is suitable for any particular purpose but to describe the functional details of the Product.
- Contents of the User's Manuals are subject to change without notice.
- Every effort has been made in the preparation of the User's Manuals to ensure the accuracy of their contents. However, should you have any questions or find any errors, contact

us or your local distributor. The User's Manuals with incorrectly ordered pages or missing pages will be replaced.

■ Warning and Disclaimer

- Except as specified in the warranty terms, YOKOGAWA shall not provide any warranty for the Product.
- YOKOGAWA shall not be liable for any indirect or consequential loss incurred by either using or not being able to use the Product.

■ Notes on Software

- YOKOGAWA makes no warranties, either expressed or implied, with respect to the Software Product's merchantability or suitability for any particular purpose, except as specified in the warranty terms.
- Please purchase the appropriate number of licenses of the Software Product according to the number of computers to be used.
- No copy of the Software Product may be made for any purpose other than backup; otherwise, it is deemed as an infringement of YOKOGAWA's Intellectual Property rights.
- Keep the software medium of the Software Product in a safe place.
- No reverse engineering, reverse compiling, reverse assembling, or converting the Software Product to human-readable format may be performed for the Software Product.
- No part of the Software Product may be transferred, converted, or sublet for use by any third-party, without prior written consent from YOKOGAWA.

Documentation Conventions

■ Symbol Marks

The following symbols are used throughout the User's Manuals.

CAUTION
Identifies instructions that must be observed to avoid physical injury, electric shock, or death.

WARNING
Identifies instructions that must be observed to prevent damages to the software or hardware, or system failures of the Product.

IMPORTANT
Identifies important information required to understand operations or functions.

TIP
Identifies additional information.

SEE ALSO
Identifies referenced content.

In the online manuals, clicking on the reference link shown in green displays the referenced content. This action does not apply to the reference link shown in black.

■ Typographical Conventions

The following typographical conventions are used throughout the User's Manuals.

● Commonly Used Conventions throughout the User's Manuals

- Character string to be entered
  The characters that must be entered are shown in monospace font as follows:

  Example:

  `FIC100.SV=50.0`

- ▼ Mark
  This symbol indicates the description for an item for which you should make a setting in the product's engineering window.

  While operating an engineering window, the help information for the selected item can be accessed from "Builder Definition Items" in the Help menu. Listing more than one definition item after this symbol implies that the paragraph on the page describes more than one definition items.

  Example:

  `▼ Tag Name, Station Name`

- Δ Mark
  Indicates that a space must be entered between character strings.
  Example:

  `.ALΔPIC010Δ-SC`

- Character string enclosed by braces { }
  Indicates character strings that may be omitted.

  Example:

  `.PRΔTAG{Δ.sheet name}`

Conventions Used to Show Key or Button Operations

- Characters enclosed by brackets [ ]
  When characters are enclosed by brackets in the description of a key or button operation, it indicates a key on the keyboard, a key on the operation keyboard, a button name in a window, or an item in a list box displayed in a window.

  Example:

  To alter the function, press the [ESC] key.

Conventions Used in Command Syntax or Program Statements

The following conventions are used within a command syntax or program statement format:

- Characters enclosed by angle brackets < >
  Indicate character strings that user can specify freely according to certain guidelines.

  Example:

  #define <Identifier> <Character string>

- "..." Mark
  Indicates previous command or argument that may be repeated.

  Example:

  Imax (arg1, arg2, ...)

- Characters enclosed by brackets [ ]
  Indicate character strings that may be omitted.

  Example:

  sysalarm <format character string> [, <output value>…]

- Characters enclosed by separators | |
  Indicates character strings that can be selected from more than one option.

  Example:

  opeguide | <format character string> [, <output value>...] |
  | OG, <element number> |

Drawing Conventions

Drawings used in the User's Manuals may be partially emphasized or simplified for convenience of description, so that the unnecessary parts are omitted from the drawings.

Drawings of the window may be slightly different from the actual screen shots with different settings or fonts; the difference is not extended to the range that may hamper the understanding of basic functionalities and operation and monitoring tasks.

Copyright and Trademark Notices

■ All Rights Reserved

■ Trademark Acknowledgements

- CENTUM, ProSafe, Vnet/IP, PRM, Exaopc, and STARDOM are registered trademarks of YOKOGAWA.
- Microsoft, Windows, Windows Vista, Windows Server, Visual Basic, Visual C++, and Visual Studio are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.
- Adobe, Acrobat, and Adobe Reader are registered trademarks of Adobe Systems Incorporated.
- Ethernet is a registered trademark of Xerox Corporation.
- EtherNet/IP is a trademark of the Open DeviceNet Vendor Association (ODVA).
- FOUNDATION in FOUNDATION fieldbus is a registered trademark of Fieldbus Foundation.
- HART is a registered trademark of the HART Communication Foundation.
- Java is a registered trademark of Oracle Corporation.
- MELSEC is a registered trademark of Mitsubishi Electric Corporation.
- Memocon-SC is a registered trademark of Yaskawa Electric Corporation.
- Modicon and Modbus are registered trademarks of Schneider Electric SA.
- PROFIBUS is a registered trademark of the PROFIBUS User Organization.
- SIEMENS and SIMATIC are registered trademarks of Siemens Industrial Automation Ltd.
- SmartPlant is a registered trademark of Intergraph Corporation.
- SYSMAC is a registered trademark of OMRON Corporation.
- All other company and product names mentioned in the User's Manuals are trademarks or registered trademarks of their respective companies.
- We do not use TM or ® mark to indicate those trademarks or registered trademarks in the User's Manuals.
- We do not use logos in the User's Manuals.

Blank Page

Function Blocks Overview

IM 33K03E21-50E 5th Edition

CONTENTS

Blank Page

1. Structure of a Function Block

A function block consists of the following components:

- Input and output terminals that exchange data with devices outside of the external function block
- Four processing functions of input processing, calculation processing, output processing, and alarm processing
- Constants and variable data used to execute processing functions. Especially, an abbreviated name called “data item” is assigned to data that is referenced or set during the operation.

The function block performs input processing, calculation processing, and output processing in sequence for an input signal read from the input terminal, and writes an output signal from the output terminal.

This chapter describes an overview of each structural component of the function block as well as a basic structure of the function block.

■ Basic Structure of the Function Block

The figure below shows a basic structure of the function block.

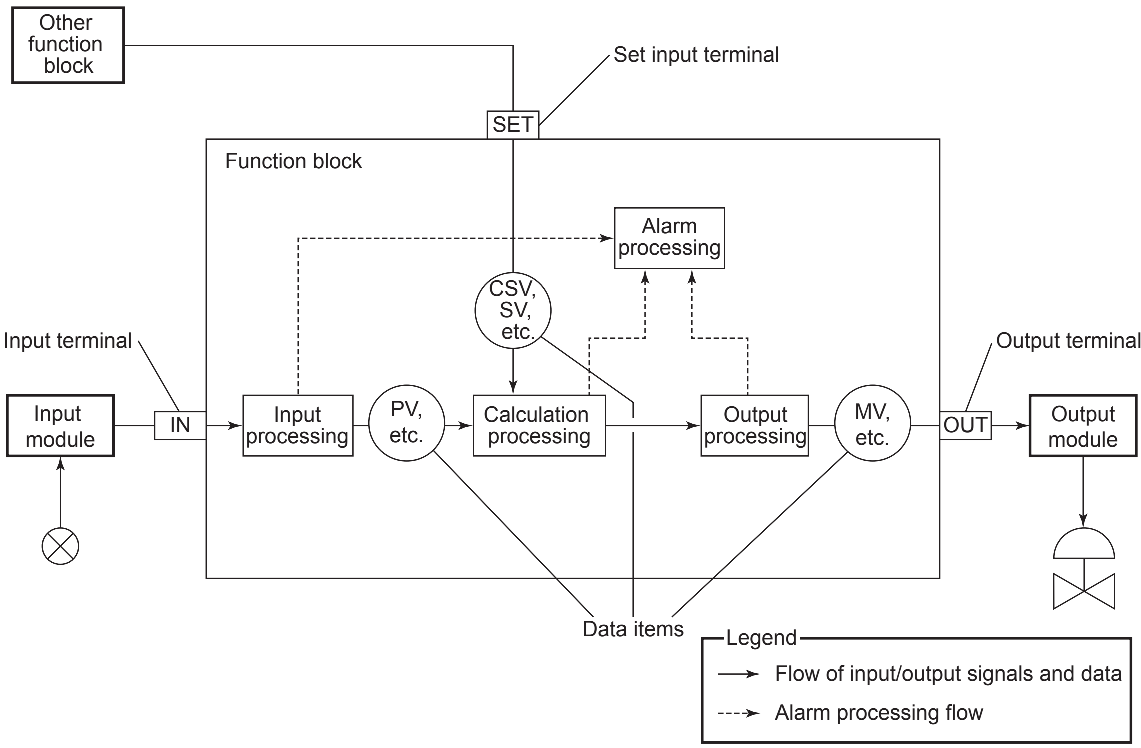


Figure 1-1 Basic Structure of the Function Block

● Input/Output Terminals

A function block performs data input/output with the process control input/output and other function blocks via input/output terminals.

An input terminal (IN), set input terminal (SET) and output terminal (OUT) are basic input/output terminals. The function block has some other input/output terminals according to the type of the function block used.

SEE ALSO For more information about the connection destinations of the input/output terminals, refer to:

2., “I/O Connection” on page 2-1

For more information about specific input/output terminals of each function block, refer to:

- 1., "Regulatory Control" in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)
- 1., "Arithmetic Calculation, Logic Operation" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)
- 2., "Sequence Control" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)
- 3., "Faceplate Blocks" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)
- 1., "Sequential Function Chart" in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)
- 2., "Unit Supervision" in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)
- 3., "Valve Pattern Monitors" in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)
- 4., "Offsite Block" in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)
- 5., "System Function Blocks" in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)

● Input Processing

Input processing changes an input signal read from the connection destination of the input terminal of the function block into data that is suitable for calculation processing (control calculation, numeric calculation, etc). Various types of input processing are performed according to the type of the function block and the input signal format.

SEE ALSO For more information about the basic input processing in the regulatory control block and calculation block, refer to:

3., “Input Processing” on page 3-1

For more information about input processing specific to each function block, refer to:

- 1., "Regulatory Control" in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)
- 1., "Arithmetic Calculation, Logic Operation" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)
- 2., "Sequence Control" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)
- 3., "Faceplate Blocks" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)
- 1., "Sequential Function Chart" in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)
- 2., "Unit Supervision" in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)
- 3., "Valve Pattern Monitors" in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)
- 4., "Offsite Block" in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)
- 5., "System Function Blocks" in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)

● Calculation Processing

Calculation processing reads data obtained by input processing, performs calculation processing according to the type of the function block, and outputs the processing result.

For example, a regulatory control block reads a process variable (PV), performs computation for regulatory control, and outputs the computation result as a manipulated value (MV).

Because the calculation processing determines the function of each function block, the processing contents vary depending on the type of the function block.

SEE ALSO For more information about the calculation processing of each function block, refer to:

- 1., "Regulatory Control" in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)
- 1., "Arithmetic Calculation, Logic Operation" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)
- 2., "Sequence Control" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)
- 3., "Faceplate Blocks" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)
- 1., "Sequential Function Chart" in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)
- 2., "Unit Supervision" in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)
- 3., "Valve Pattern Monitors" in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)
- 4., "Offsite Block" in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)
- 5., "System Function Blocks" in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)

● Output Processing

Output processing outputs data obtained by calculation processing to the connection destination of the output terminal as an output signal.

Various types of output processing are performed according to the type of the function block and the output signal format.

SEE ALSO For more information about the basic output processing in the regulatory control block and calculation block, refer to:

4., “Output Processing” on page 4-1

For more information about output processing specific to each function block, refer to:

- 1., “Regulatory Control” in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)
- 1., “Arithmetic Calculation, Logic Operation” in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)
- 2., “Sequence Control” in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)
- 3., "Faceplate Blocks" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)
- 1., "Sequential Function Chart" in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)
- 2., "Unit Supervision" in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)
- 3., "Valve Pattern Monitors" in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)
- 4., "Offsite Block" in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)
- 5., "System Function Blocks" in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)

● Alarm Processing

Alarm processing performs various types of alarm check during input processing, calculation processing and output processing in order to detect a process error. When an error is detected, the alarm processing reflects the detection of an alarm in the “alarm status” that is one of the data items of the function block, and also notifies a message indicating the detection result to the operation and monitoring.

SEE ALSO For more information about the basic alarm processing in the function block, refer to:

5., “Alarm Processing – FCS” on page 5-1

For more information about the alarm processing specific to each function block, refer to:

- 1., "Regulatory Control" in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)
- 1., "Arithmetic Calculation, Logic Operation" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)
- 2., "Sequence Control" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)
- 3., "Faceplate Blocks" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)
- 1., "Sequential Function Chart" in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)
- 2., "Unit Supervision" in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)
- 3., "Valve Pattern Monitors" in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)
- 4., "Offsite Block" in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)
- 5., "System Function Blocks" in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)

● Data Items

A function block retains various data according to the type of the function block in a database, which includes setup parameters and variable data that may be referenced or set during the operation. Abbreviated names that are assigned to these set parameters and variable data are generically called “data items.”

For instance, the function block can perform calculation processing based on a specific data item value and can reflect that processing result in another data item.

The controls of the function block, such as “MAN” (manual) and “AUT” (auto), and the “block mode” that indicates the output status are some of the data items.

Main data items are as follows:

- Block mode (MODE)
- Block status (BSTS)
- Alarm status (ALRM)
- Process variable (PV)
- Setpoint value (SV)
- Manipulated output value (MV)

SEE ALSO For more information about the block mode, block status and alarm status, refer to:

6., “Block Mode and Status” on page 6-1

For more information about data items that are retained by each function block, refer to:

- 1., "Regulatory Control" in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)
- 1., "Arithmetic Calculation, Logic Operation" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)
- 2., "Sequence Control" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)
- 3., "Faceplate Blocks" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)
- 1., "Sequential Function Chart" in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)
- 2., "Unit Supervision" in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)
- 3., "Valve Pattern Monitors" in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)
- 4., "Offsite Block" in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)
- 5., "System Function Blocks" in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)

Blank Page

2. I/O Connection

By performing the I/O connection, data can be exchanged between a function block and the connection destination according to the connection method.

■ Connection Destination of I/O Connection

With I/O connection, the destination and method of connection for each I/O terminal of a function block is specified. When the I/O connection is performed, process I/O, software I/O, communication I/O and expanded communication I/O (*1), fieldbus I/O and other function blocks can be specified as the connection destination of the function block's I/O terminal.

*1: Expanded Communication I/O can be used for only the FFCS-V.

● Process I/O

- Analog I/O
- Contact I/O

● Software I/O

- Internal switch (common switch)
- Message output

● Communication I/O and Expanded Communication I/O

- Word data
- Bit data

● Fieldbus I/O

- Parameter of fieldbus block

● Other Function Blocks

- Data items of other function blocks
- I/O terminals of other function blocks

■ I/O Connection Methods

The I/O connection methods include data connection, terminal connection and sequence connection.

● Data Connection

This is the I/O connection method used for reading and setting data with respect to the process I/O, software I/O, communication I/O and expanded communication I/O, fieldbus I/O or other function blocks.

● Terminal Connection

This is the I/O connection method used when connecting between cascade control function blocks or connecting function blocks via a selector switch block (SW-33, SW-91). Data is exchanged between the terminals of two function blocks.

● Sequence Connection

This is the I/O connection method used for testing whether or not the connection destination data used by the sequence control satisfies the conditional expression, or for changing block mode, alarm status, data, etc. of the connection destination.

2.1 Data Connection

Data connection is used when exchanging data values and data status between a function block and the data item of the element specified as the connection destination.

■ Data Connection

Data connection is a method in which the element symbol name and data item name of various elements containing data are specified as the I/O connection information to indicate the connection destination of the function block's I/O terminal. Process I/O, software I/O, communication I/O and expanded communication I/O, fieldbus I/O or other function blocks can be specified as an element which contains data.

In data connection, data values and data status are directly exchanged with the data item of the element specified as the connection destination.

■ Data Reference and Data Setting

In data connection, reading data from the connection destination is called "data reference," and writing data into the connection destination from the output terminal of the function block is called "data setting."

● Data Reference

Data reference is a type of data connection in which data is read from the connection destination of the function block's input terminal. The data value of the connection destination is read as an input value of the function block in data reference. Also, the data status of the input data is modified depending upon the data status of the data from the connection destination.

With data reference, data at the same connection destination can be referenced from I/O terminals of multiple function blocks. In this case, the same input data is read to each function block.

● Data Setting

Data setting is a type of data connection in which data is written into the connection destination from the function block's output terminal. The value of the function block's output data is sent to the connection destination. Also, the data status of the connection destination's data is modified depending upon the data status of the output data from the function block.

IMPORTANT

When setting data for the process output, make sure that one output terminal corresponds to one process output.
If data is set for the same process output from output terminals of multiple function blocks, conflict will result at the process output due to different data values set.

■ Destinations of Data Connection

In data connection, process I/O, software I/O, communication I/O and expanded communication I/O, fieldbus I/O or data items of other function blocks can be specified as the connection destination of the function block's I/O terminal.

Data reference and data setting can be performed with each of the connection destinations.

● Data Connection with Process I/O

Data connection with process I/O is an I/O connection that connects the function block's I/O terminal to the process I/Os such as analog I/O and contact I/O .

Since process I/Os do not have I/O terminals, terminal connection cannot be performed.

An example of data connection with process I/O is shown below:

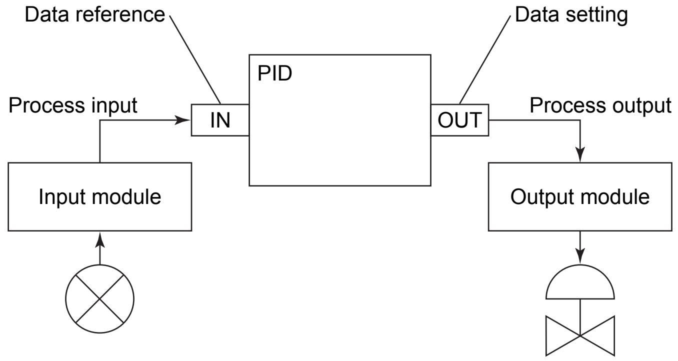


Figure 2.1-1 Data Connection with Process I/O

Data Connection with Software I/O

Data connection with software I/O is an I/O connection that connects an internal switch and the message outputs such as annunciator messages, messages for sequence control, etc. to the function block's I/O terminal.

An example of data connection with software I/O is shown below:

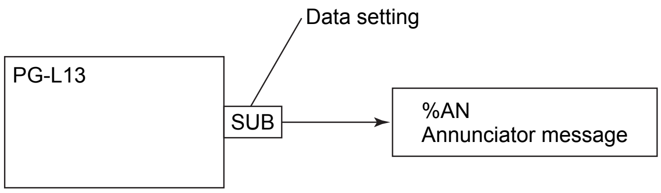


Figure 2.1-2 Data Connection with Software I/O

Data Connection with Communication I/O

Data connection with communication I/O means that communication I/O word/bit data or expanded communication I/O word/bit data is connected to the I/O terminal of a function block.

Its example is shown below.

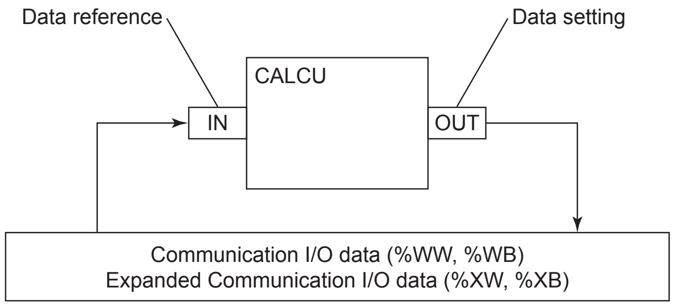


Figure 2.1-3 Data Connection with Communication I/O

SEE ALSO For more information about communication I/O, refer to:

- 1., "General Information Regarding to Subsystem Communication" in the Communication with Subsystems Using RIO (IM 33K03L10-50E)
- 1., "General Information Regarding to Subsystem Communication" in the Communication with Subsystems Using FIO (IM 33K03L20-50E)

Data Connection with Fieldbus I/O

Data connection with fieldbus I/O means that fieldbus block parameters are connected to the I/O terminal of a function block.

Its example is shown below.

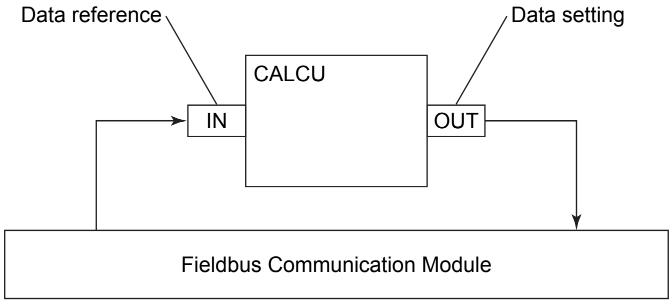


Figure 2.1-4 Data Connection with Fieldbus I/O

SEE ALSO

For more information about data connection with Fieldbus I/O in regarding to FFCS Series, KFCS2 and KFCS, refer to:

A2.2, "Control Loop and Data Flow" in the FOUNDATION fieldbus Reference (IM 33K20T10-50E)

For more information about data connection with Fieldbus I/O in regarding to LFCS2, LFCS, SFCS and PFCS, refer to:

A3.3, "Fieldbus Block Connection" in the FOUNDATION fieldbus Tools (IM 33S05P10-01E)

Data Connection with Other Function Blocks

Data connection with other function blocks is an I/O connection that connects data items such as process variables (PV) and manipulated output values (MV) held in the other function blocks, to the function block's I/O terminals.

An example of data connection with other function blocks' data items is shown below:

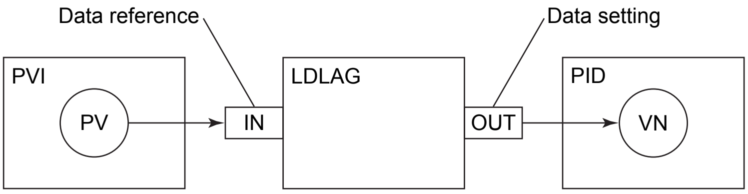


Figure 2.1-5 Data Connection with Other Function Blocks' Data Items

In data connection with other function blocks, data is directly exchanged with the data items of the connection destination. Therefore, there is no need to specify I/O connection information in the function blocks of the connection destination as long as the I/O connection information is specified in the function block of the connection source.

When using calculated input values (RV, RVn) as constants in a calculation block, data can be set for the calculated input values (RV, RVn) of that calculation block. In such a case, however, if data reference or terminal connection (cascade input) is specified for the input terminal corresponding to these calculated input values (RV, RVn), the input action that uses the input terminal has precedence over the other.

An example of data setting for the calculated input value (RV) is shown below:

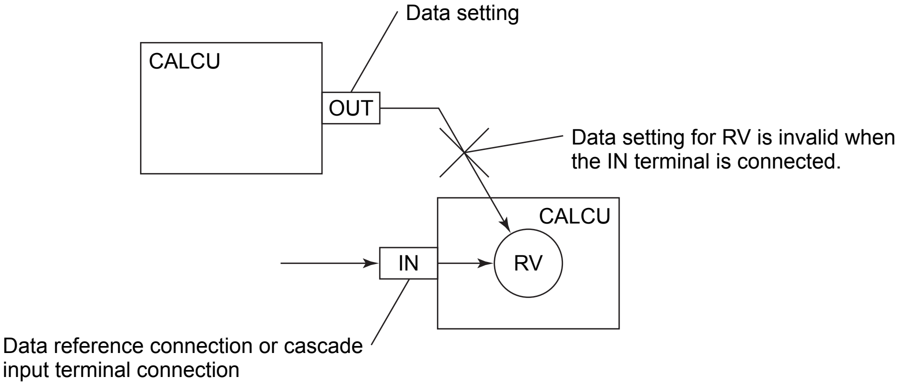


Figure 2.1-6 Data Setting for Calculated Input Value (RV)

■ I/O Connection Information for Data Connection

▼ Input Connection Information, Output Connection, Set Value Input Connection Information

Specify the I/O connection information to the I/O terminal of the function block as follows in order to perform data connection.

Element symbol name.data item name

- Element symbol name
  A tag name, label name, element number or terminal number that identifies the connection destination.
- Data item name
  PV, RV, MV, etc.

In data connection with a process I/O, a tag name, label name or terminal number is specified for the element symbol name, and PV is specified for the data item name.

The terminal number is represented by the following symbols:

%Znnusmm
Terminal (01 to 32)
Slot (1 to 4)
Unit (1 to 5)
Input module (fixed at 01) (*1)
Node number (01 to 08) (*2)

*1: Can only be used with SFCS or PFCS.
*2: Can only be used with LFCS2 or LFCS.

Figure 2.1-7 I/O Information Symbols : LFCS2/LFCS/SFCS/PFCS

%Znnusmm
Terminal (01 to 64) (*1)
Segment (1 to 4) (*2)
Slot (1 to 8)
Node number (01 to 10) (*3)(*4)

*1: For fieldbus communication, terminal “mm” ranges between 01 to 48.
*2: For fieldbus communication, segment “s” ranges between 1 to 4. For process output “s” is fixed as 1. For Analog I/O (HART Compatible) modules, when “s” is set to 2, the terminal is used as a HART variable channel; when “s” is set to 1, the terminal is used as an analog input/output channel.
*3: If the database in KFCS2 is ER bus node expanded type, the range of node number becomes 01 to 15.
*4: The node number of the FCU in FFCS series is fixed at 1 and cannot be changed. The extended node should be numbered from 2.

Figure 2.1-8 I/O Information Symbols : FFCS Series/KFCS2/KFCS

In data connection with software I/O, a tag name or element symbol number is specified for the element symbol name, and PV is specified for the data item name.

In data connection with other function blocks, a tag name is specified for the element symbol name and a data item name that is the target of connection is specified for the data item name.

TIP

The I/O terminal which performs input and output of character string data cannot be connected to a process I/O. The I/O terminals that perform input and output of the character string data are shown in the following:

Table 2.1-1 I/O Terminals for Character Strings

Function block	Terminal
CALCU-C	Q04 to 07, J02, J03
DSW-16C	OUT
BDSET-1C/2C	J01 to J16
BDA-C	J01 to J16

SEE ALSO

For more information about the element numbers, refer to:

"■ Terminal Numbers, Element Numbers" on page 2-23

For more information about the data item names of each function block, refer to:

- 1, "Regulatory Control" in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)
- 1, "Arithmetic Calculation, Logic Operation" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)
- 2, "Sequence Control" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)
- 3, "Faceplate Blocks" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)

■ Data Reference with Respect to Dual-Redundant Input

As indicated below, there are three methods of data reference with respect to dual-redundant input modules, depending on the type of input module.

● Dual-Redundant Analog Input : LFCS2/LFCS/SFCS/PFCS

When reading data from dual-redundant analog input modules, a Dual-Redundant Signal Selector Block (SS-DUAL) is used. Specify an input module for each of the connection destinations of the two input terminals (IN1, IN2) of the SS-DUAL block, respectively.

An example of a dual-redundant input connection is illustrated below.

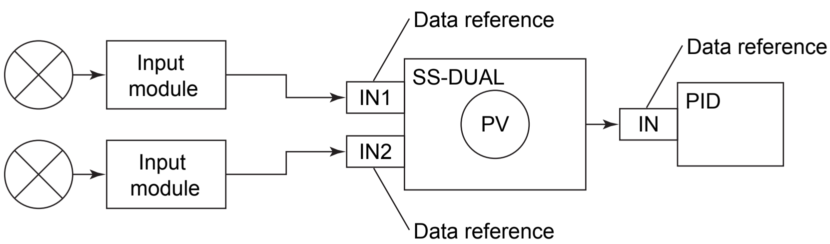


Figure 2.1-9 Dual-Redundant Input Connection : LFCS2/LFCS/SFCS/PFCS

● Dual-Redundant Multi-Point Analog Input : LFCS2/LFCS/SFCS/PFCS

To access the multi-point control analog input modules in dual-redundant configuration, the following settings are required.

- On the IOM module property sheet, check the mark "Duplicate Next Card." The setting is the same for either input modules or output modules.

- For the function block input terminal, specify the terminal number of the module with slot number 1 of the two duplicate modules. The data reference method is the same as that for a non-dual-redundant module.

Normally, the module with slot number 1 is the control side and the module with slot number 2 is the standby side. If the module on control side fails, the module that was on the standby side will take over the control. Function blocks will read data from the new control side module.

SEE ALSO For more information about multi-point control analog I/O module dual-redundant configuration, refer to:

"■ Dual-Multipoint Control Analog I/O : PFCS/LFCS2/LFCS/SFCS" in A3.3.2, "Parameters for Multipoint Control Analog Input/Output" in the Field Control Stations Reference (IM 33K03E10-50E)

● Dual-Redundant Multi-Point Analog Input : FFCS Series/KFCS2/KFCS

To access the multi-point analog input modules in dual-redundant configuration, the following settings are required.

- On the IOM module property sheet, check the mark "Duplicate Next Card."
- For the function block input terminal, specify the terminal number of the module with the smaller slot number of the two duplicate modules. The data reference method is the same as that for a non-dual-redundant module.

Normally, the module with the smaller slot number is the control side and the module with the larger slot number is the standby side. If the module on control side fails, the module that was on the standby side will take over the control. Function blocks will read data from the new control side module.

SEE ALSO For more information about multi-point analog I/O module dual-redundant configuration, refer to:

"■ Dual-FIO Analog Input/Output : FFCS series/KFCS2/KFCS" in A3.4.1, "Parameters for FIO Analog Inputs/Outputs" in the Field Control Stations Reference (IM 33K03E10-50E)

● Dual-Redundant Contact Input

When reading data from dual-redundant status input modules, it is necessary to perform the following operations.

- On the IOM module property sheet, check the mark "Duplicate Next Card."
- For the function block input terminal, specify the terminal number of the module with the smaller slot number of the two duplicate modules. The method of setting data reference is the same as that for a non-dual-redundant module.

Normally, the module with the smaller slot number is to be the control side and the module with the larger slot number is to be the standby side. If the module on control side faults, the module that was on the standby side will take over the control. Function blocks read data from the control side.

SEE ALSO For more information about contact I/O module dual-redundant configuration in regarding to LFCS2, LFCS, SFCS and PFCS, refer to:

"■ Dual-Relay, Contact Terminal, Contact Connector : PFCS/LFCS2/LFCS/SFCS" in A3.3.4, "Parameters for Relay, Contact Terminal or Contact Connector" in the Field Control Stations Reference (IM 33K03E10-50E)

For more information about contact I/O module dual-redundant configuration in regarding to FFCS Series, KFCS2 and KFCS , refer to:

"■ Dual-FIO Contact Input/Output : FFCS series/KFCS2/KFCS" in A3.4.2, "Parameters for FIO Contact Inputs/Outputs" in the Field Control Stations Reference (IM 33K03E10-50E)

■ Data Setting with Respect to Dual-Redundant Output

As indicated below, there are three methods of data setting with respect to dual-redundant output modules, depending on the type of output module.

● Dual-Redundant Analog Output : LFCS2/LFCS/SFCS/PFCS

To write the same output value to dual-redundant analog output modules, the following operation is required:

- Specify “Dual” for each terminal on the IOM definition builder. For redundancy, specify two successive output points (1-2, 3-4, ..., 15-16) to the output modules.
- For output terminal of the function block, specify the output point with the younger number of the two output points. The method of setting data is the same as that for a non-dual-redundant module.

● Dual-Redundant Multi-Point Analog Output : LFCS2/LFCS/SFCS/PFCS

To write data to the multi-point control analog output modules in dual-redundant configuration, the following settings are required.

- On the IOM module property sheet, check the mark “Duplicate Next Card.” The setting is the same for either input modules or output modules.
- For the function block input terminal, specify the terminal number of the module with slot number 1 of the two duplicate modules. The data reference method is the same as that for a non-dual-redundant module.

Normally, the module with slot number 1 is the control side and the module with slot number 2 is the standby side. If the module on control side fails, the module that was on the standby side will take over the control. Function blocks will read data from the new control side module.

SEE ALSO For more information about multi-point control analog I/O module dual-redundant configuration, refer to:

“■ Dual-Multipoint Control Analog I/O : PFCS/LFCS2/LFCS/SFCS” in A3.3.2, “Parameters for Multipoint Control Analog Input/Output” in the Field Control Stations Reference (IM 33K03E10-50E)

● Dual-Redundant Multi-Point Analog Output : FFCS Series/KFCS2/KFCS

To write data to the multi-point analog output modules in dual-redundant configuration, the following settings are required.

- On the IOM module property sheet, check the mark “Duplicate Next Card.”
- For the function block input terminal, specify the terminal number of the module with the smaller slot number of the two duplicate modules. The data reference method is the same as that for a non-dual-redundant module.

Normally, the module with the smaller slot number is the control side and the module with the larger slot number is the standby side. If the module on control side fails, the module that was on the standby side will take over the control. Function blocks will read data from the new control side module.

SEE ALSO For more information about multi-point analog I/O module dual-redundant configuration, refer to:

“■ Dual-FIO Analog Input/Output : FFCS series/KFCS2/KFCS” in A3.4.1, “Parameters for FIO Analog Inputs/Outputs” in the Field Control Stations Reference (IM 33K03E10-50E)

● Dual-Redundant Contact Output

When writing data to dual-redundant contact output modules, it is necessary to perform the following operations in order to write the same output value to the two output modules.

- On the IOM module property sheet, check the mark “Duplicate Next Card.”
- For output terminal of the function block, specify the terminal number of the module with the smaller slot number of the two duplicate modules. The method of setting data is the same as that for a non-dual-redundant module.

Normally, the module with the smaller slot number is to be the control side and the module with the larger slot number to be on the standby side. If the module on control side faults, the module that was on the standby side will take over the control. Function blocks write data to the modules on both sides.

SEE ALSO

For more information about contact I/O module dual-redundant configuration in regarding to PFCS, LFCS2, LFCS and SFCS, refer to:

“■ Dual-Relay, Contact Terminal, Contact Connector : PFCS/LFCS2/LFCS/SFCS” in A3.3.4, “Parameters for Relay, Contact Terminal or Contact Connector” in the Field Control Stations Reference (IM33K03E10-50E)

For more information about contact I/O module dual-redundant configuration in regarding to KFCS2, KFCS and FFCS Series, refer to:

“■ Dual-FIO Contact Input/Output : FFCS series/KFCS2/KFCS” in A3.4.2, “Parameters for FIO Contact Inputs/Outputs” in the Field Control Stations Reference (IM33K03E10-50E)

2.2 Terminal Connection

Terminal connection is used when performing cascade control by connecting I/O terminal of a function block to that of another function block.

■ Terminal Connection

The terminal connection specifies the I/O terminal of a function block as the connection destination of the other function block's I/O terminal. Data is exchanged between the I/O terminals of two function blocks in terminal connection. The connections between I/O terminals of function blocks are well applied to the cascade loops where the upper stream block's output depends on the lower stream block's status.

The terminal connection is mainly used in the following instances:

● Connection Between Function Blocks

The output terminal (OUT) of the upstream function block and the setting input terminal (SET) or input terminal (IN, INn) of the downstream function block are connected under the cascade control.

● Connection by Way of a Switch Block (SW-33, SW-91)

Terminal connection must always be used as the I/O connection method at one or the other of the I/O terminals (input side or output side) of the SW-33 or SW-91 block. The other terminal uses the I/O connection method such as data reference, data setting or terminal connection that applies the case that SW-33 or SW-91 does not intervene.

■ I/O Connection Information for Terminal Connection

When the terminal connection with the I/O terminal of another function block is established, specify the I/O connection information to the I/O terminal of the function block as follows:

Element symbol name.I/O terminal name

- Element symbol name
  A tag name identifies the connection destination.
- I/O terminal name
  IN, OUT, SET, etc.

In terminal connection, I/O terminal of each other must be specified in the both of function blocks: connection source and connection destination. This is because data is exchanged with the I/O terminal of the function block of the connection destination.

2.2.1 Connection Between Function Blocks

This section explains the connection between the output terminal (OUT) of the upstream function block and the setting input terminal (SET) or input terminal (IN, INn) of the downstream function block under the cascade control.

■ Terminal Connection between the Output Terminal (OUT) and Setting Input Terminal (SET)

The following example shows connection between output terminal (OUT) of the upstream function block and the setting input terminal (SET) of the downstream function block under the cascade control. In this example, two I/O terminals are connected by the terminal connection.

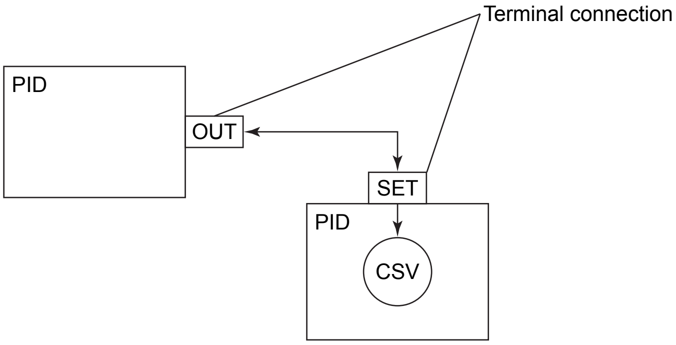


Figure 2.2.1-1 Terminal Connection between the Output Terminal and SET Terminal

In this example, data is sent from the output terminal (OUT) of the upstream function block by way of the setting input terminal (SET) of the downstream function block, then set as a cascade setting value (CSV) of the downstream function block at the end.

■ Terminal Connection between the Output and Input Terminals

The following example shows a connection between output terminal (OUT) of the upstream function block and the input terminal (IN, INn) of the downstream function block. In this example, two I/O terminals are connected by the terminal connection.

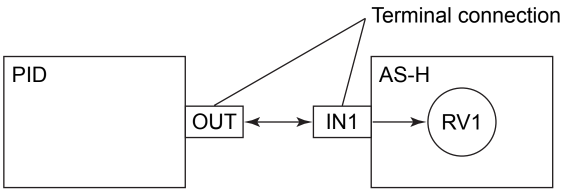


Figure 2.2.1-2 Terminal Connection between the Output and Input Terminals

In this example, data is sent from the output terminal (OUT) of the upstream function block by way of the input terminal (IN1) of the downstream function block, then set as a calculated input value (RV1) of the downstream function block at the end.

■ Function Blocks and Their Target Terminals that Allow Terminal Connection

The following table lists the function blocks that can be connected to the OUT terminal using a terminal connection and the I/O terminals for which terminal connections can be used.

Table 2.2.1-1 List of Function Blocks which can be Connected by the Terminal Connection and Their Target Terminals

<table>
<tr><th>Block type</th><th>Block model name</th><th>Target terminal name</th><th>Corresponding input data</th></tr>
<tr><td rowspan="9">Regulatory control</td><td>PID
PI-HLD
PID-BSW
ONOFF
ONOFF-E
ONOFF-G
ONOFF-GE
PID-TP
PD-MR
PI-BLEND
PID-STC
MILD-SW
VELLIM
FOUT
SPLIT</td><td>SET</td><td>CSV</td></tr>
<tr><td rowspan="2">RATIO
FFSUM</td><td>IN</td><td>PV</td></tr>
<tr><td>SET</td><td>CSV</td></tr>
<tr><td rowspan="3">SS-H/M/L
AS-H/M/L</td><td>IN1</td><td>RV1</td></tr>
<tr><td>IN2</td><td>RV2</td></tr>
<tr><td>IN3</td><td>RV3</td></tr>
<tr><td rowspan="2">SS-DUAL</td><td>IN1</td><td>RV1</td></tr>
<tr><td>IN2</td><td>RV2</td></tr>
<tr><td>XCPL</td><td>IN</td><td>PV</td></tr>
<tr><td>Calculation</td><td>SQRT
EXP
LAG
INTEG
LD
LDLAG
DLAY
DLAY-C
FUNC-VAR</td><td>IN</td><td>RV</td></tr>
</table>

2.2.2 Connection by a Switch Block (SW-33, SW-91)

This section explains the connections between I/O terminals of the function blocks by a switch block (SW-33, SW-91) as well as the connection to the process I/O or software I/O by the switch block.

A terminal connection to a switch block (SW-33, SW-91) of another control station or a sequence connection via the SW-33 or SW-91 block cannot be done.

■ Connection Between Function Blocks by a Switch Block (SW-33, SW-91)

A switch block (SW-33, SW-91) can be placed in the middle of the cascade control loop. In this case, the switch block and its upstream/downstream function blocks are connected by the terminal connection, respectively.

● Connection to a Setting Input Terminal (SET) by a Switch Block (SW-33, SW-91)

The following example shows a connection between an output terminal (OUT) of the upstream function block and a setting input terminal (SET) of the downstream function block by a switch block (SW-33).

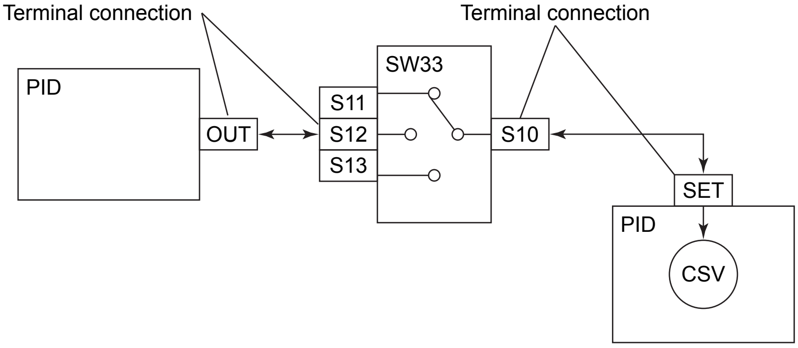


Figure 2.2.2-1 Connection to a Setting Input Terminal (SET) by Way of a Switch Block (SW-33)

● Connection to an Input Terminal by a Switch Block (SW-33, SW-91)

The following example shows a connection between an output terminal (OUT) of the upstream function block and an input terminal (IN) of the downstream function block by a switch block (SW-33).

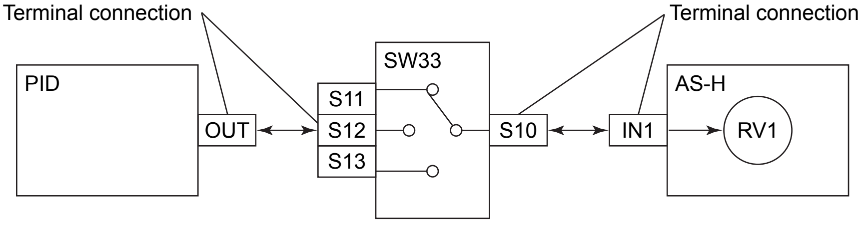


Figure 2.2.2-2 Connection to an Input Terminal by Way of a Switch Block (SW-33)

■ Connection to a Process I/O or Software I/O by a Switch Block (SW-33, SW-91)

An I/O terminal of the function block and a process I/O or software I/O are connected by a switch block (SW-33, SW-91).

In the SW-33, SW-91 block, however, there is no data item to be used for data connection from another function block. Therefore, the I/O terminal on the function block side is connected by the terminal connection and that on the process I/O or software I/O side is connected by data connection.

● Data Reference by a Switch Block (SW-33, SW-91)

In order to input data from a process I/O by a switch block (SW-33, SW-91), one I/O terminal of the SW-33 or SW-91 block is connected by the terminal connection while the other is connected by data reference.

The following example shows data reference by a SW-33 block.

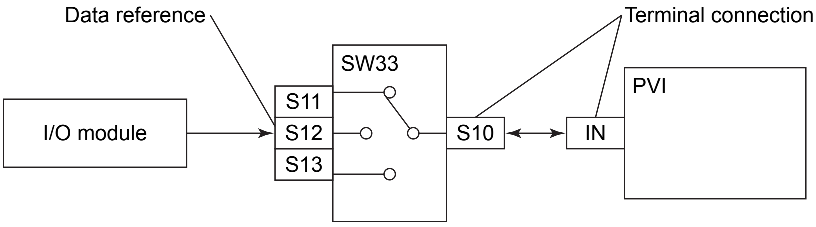


Figure 2.2.2-3 Data Reference by a Switch Block (SW-33)

● Data Setting by a Switch Block (SW-33, SW-91)

In order to output data to a process I/O by a switch block (SW-33, SW-91), one I/O terminal of the SW-33 or SW-91 block is connected by the terminal connection while the other is connected by data setting.

The following example shows data setting by a SW-33 block.

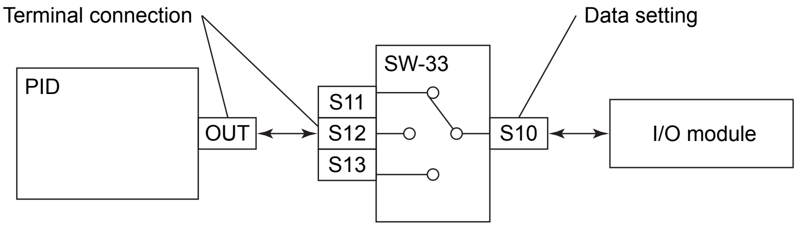


Figure 2.2.2-4 Data Setting by a Switch Block (SW-33)

■ Mixture of Terminal Connection and Data Connection

In the SW-33 or SW-91 block, it is possible to mix two methods; reading data by the terminal connection and by data connection. These two methods can be switched depending on the situation.

The following example shows a mixture of terminal connection and data connection by a SW-33 block.

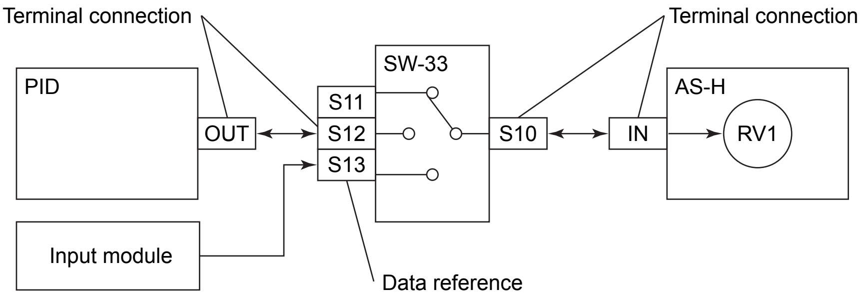


Figure 2.2.2-5 Mixture of Terminal Connection and Data Connection by a Switch Block (SW-33)

2.3 Sequence Connection

Sequence connection is used when testing the conditions of input signals in the function block or manipulating the status of the elements at the output destination.

■ Sequence Connection

In this method, various elements that contain data are specified as the connection destination of the function block's I/O terminal. It is necessary to specify the conditional expression to the input terminal in order to judge the data status, as well as data for manipulating the element status to the output terminal. The sequence connection is the I/O connection method used by sequence controls. In addition to the sequence control block, sequence connection can also be used in the Pulse Count Input Block (PTC) of regulatory control blocks, the Logic Operation Blocks (*1) or the General-Purpose Calculation Blocks (CALCU, CALCU-C) for arithmetic and logic operation functions.

*1: Logic Operation Block can be used in FCSs except PFCS.

■ Condition Testing and Status Manipulation

In sequence connection, a process performed to read data from the connection destination is called "condition testing," and a process performed to write data to the connection destination is called "status manipulation."

In sequence connection, data contained in the element is exchanged to test the condition, and data for status manipulation of the element is exchanged to manipulate the status, respectively, with the element (process I/O, software I/O, or other function blocks) specified as a connection destination.

● Condition Testing

Condition testing is a sequence connection for reading data from the connection destination of the function block's I/O terminal. In condition testing, the data at the connection destination is tested by the condition expression specified to the input terminal, and a logical value (true or false) which indicates established/unestablished of the condition expression is obtained. That is, the condition testing replaces the data read by the function block with a logical value that indicates the status of the connection destination.

● Status Manipulation

Status manipulation is a sequence connection to output to the connection destination from the function block's I/O terminal. In status manipulation, status manipulation of the connection destination specified to the output terminal is performed according to the result of logical operation (true or false) of the function block, then the connection destination status is modified.

■ I/O Connection Information for Sequence Connection

In sequence connection, the I/O connection information is specified to the I/O terminal of the function block as follows. In the Sequence Table Block (ST16, ST16E), specify this information in the condition signal setting area and operation signal setting area.

Element symbol name.data item name.condition specification

Element symbol name.data item name.manipulation specification

- Element symbol name
  Tag name, label name, element number, or terminal number that identifies the connection destination
- Data item name
  Differs according to the type of connection destination

SEE ALSO For more information about condition specification and manipulation specification, refer to:

- 2.2.10, "Condition Signal Description : Referencing Other Function Blocks and I/O Data" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)
- 2.2.11, "Control Signal Description: Referencing Sequence Table" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)
- 2.2.12, "Syntax for Condition Signal Description: Logic Chart Reference in Sequence Table" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)
- 2.2.13, "Description of Action Signal : Status Manipulation for Other Function Blocks and I/O Data" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)
- 2.2.14, "Action Signal Description : Status Manipulation for Sequence Table" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)
- 2.2.15, "Action Signal Description : Status Manipulation for a Logic Chart from a Sequence Table" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)

■ Function Blocks that Allow Sequence Connections and Their Target Terminals

The following table lists function blocks that allow sequence connections as well as their I/O terminals.

Table 2.3-1 A List of Function Blocks and Their Target Terminals that Allow Sequence Connections

Block type	Block model name	Target terminal name
Regulatory control	PTC	OUT
Sequence control	ST16 ST16E	Q01 to Q56, J01 to J56 (*1)
	LC64 LC64-E	Q01 to Q32, J01 to J32 (*1)
	TM CTS CTP	OUT
	VLVM	J01 to J17
Logical operation (*2)	AND OR	OUT, Q01, Q02
	NOT	IN, OUT
	SRS1-S SRS1-R	Q01, Q02, J01
	SRS2-S SRS2-R	Q01, Q02, J01, J02
	WOUT	OUT, Q01, Q02
	OND OFFD TON TOFF	IN, OUT
	GT GE EQ	OUT
General-purpose calculation	CALCU	IN, OUT, Q01 to Q07, J01 to J03
	CALCU-C	IN, OUT, Q01 to Q03, J01

*1: Input/output connection setting areas of sequence tables and logic chart block are equivalent to terminals.
*2: Logic Operation Block can be used in FCSs except PFCS.

Even if the function block has a terminal that allows sequence connections, it cannot be connected by the sequence connection via a switch block (SW-33, SW-91).

A sequence connection (condition testing and status manipulation) cannot be set to the I/O terminals that perform input and output of character string data.

I/O terminals which perform input and output of character string data are as follows:

Table 2.3-2 I/O Terminal for Character String

Function block	Terminal
CALCU-C	Q04 to Q07, J02, J03
DSW-16C	OUT
BDSET-1C/2C	J01 to J16
BDA-C	J01 to J16

2.4 Connection between Control Stations

A data item or I/O terminal of the function block in another control station can be connected to the I/O terminal of the function block in the present control station.

■ Connection between Control Stations

The connection between control stations is an I/O connection method for establishing data connection or terminal connection between the function block of the present control station and that of another control station.

The maximum I/O terminal connection points for each type of field control station (FCS) are as follows.

Field Control Station :	Maximum 160 points (*1)
Field Control Station (Compact Type) :	Maximum 160 points (*2)
Enhanced Field Control Unit (RIO) :	Maximum 512 points (*3)
Field Control Unit (RIO) :	Maximum 512 points (*4)(*5)
Enhanced Field Control Unit (FIO) :	Maximum 512 points (*6)
Standard Field Control Station (FIO) :	Maximum 512 points (*5)(*7)

*1: The maximum number of points connectable to PFCS. If Batch Control database type is applied, this number becomes 64.
*2: The maximum number of points connectable to SFCS
*3: The maximum number of points connectable to LFCS2.
*4: The maximum number of points connectable to LFCS.
*5: The maximum number of points connectable to KFCS or LFCS varies with the following database types.
- The maximum number of points for Unit control (without recipes) type is 128.
- The maximum number of points for Unit control (with recipes) type is 64.
- The maximum number of points for Unit control (with recipes and valve monitors) type is 64.

*6: The maximum number of points connectable to FFCS series or KFCS2.
*7: The maximum number of points connectable to KFCS.

Even between the function blocks that belong to different control stations, the I/O connection can be achieved by a similar procedure to that for the connection between function blocks belong to the same control station.

The following diagram shows an example of cascade control using the connection between control stations.

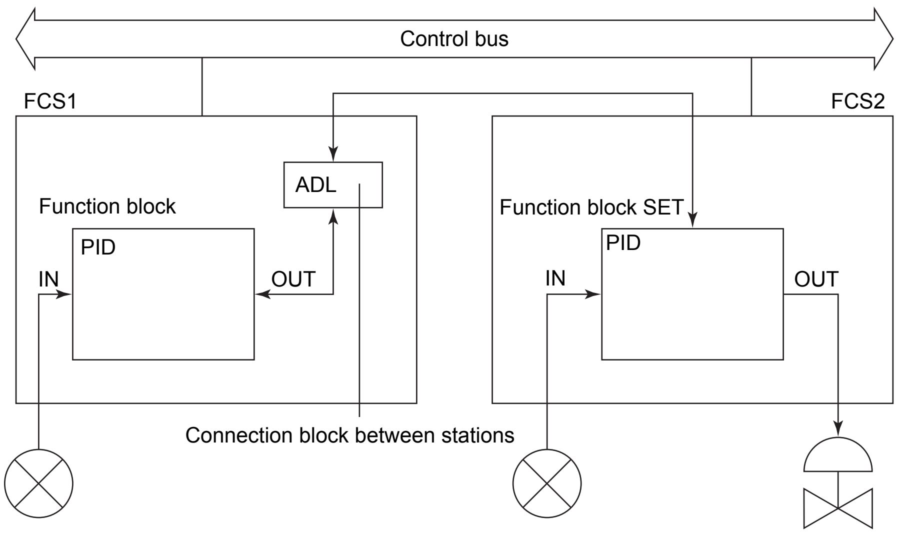


Figure 2.4-1 Connection between Control Stations (Example of the Cascade Control)

■ Cases when Connection between Control Stations is not Allowed

The connection between control stations is not allowed under the following circumstances:

- Sequence connection
- Connection to a process I/O (except for the contact I/O), word data of communication I/O and word data of expanded communication I/O.
- Terminal connection to a switch block (SW-33, SW-91)
- Data connection or terminal connection from a faceplate block
- Connection to an alarm input terminal of a Representative Alarm Block (ALM-R)
- Setting to character string data (The string data can be checked.)
- FOUNDATION fieldbus Faceplate Block OUT terminal

■ Data Connection with Other Control Stations

The inter-station connection block (ADL) is automatically generated if the I/O connection information with respect to a function block of another control station is specified for the I/O terminal of the function block at the connection source by using the Function Block Detail Builder of Control Drawing Builder. Exchanging data with the function block of another control station is done via the ADL block.

The setting items for the I/O connection information are the same as those within the same control station.

The I/O operation and the function block processing remain synchronized because the function block within the same control station performs the processing continuously according to the defined execution order. On the other hand, I/O operations are performed asynchronous to the function block processing in the I/O connection between different control stations. Therefore, communications between control stations should be avoided in applications which require strict timings.

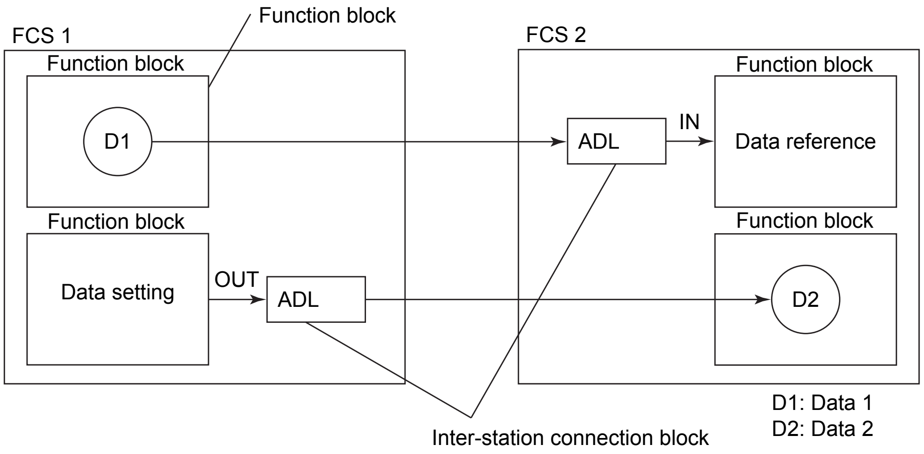


Figure 2.4-2 Connection between Control Stations (Data Connection)

■ Terminal Connection with Other Control Stations

It is possible to establish a terminal connection with a function block belongs to another control station for the cascade control. The connection is possible even if the function block in the downstream of the cascade belongs to another control station.

However, a select switch cannot be placed in the middle of the cascade connection.

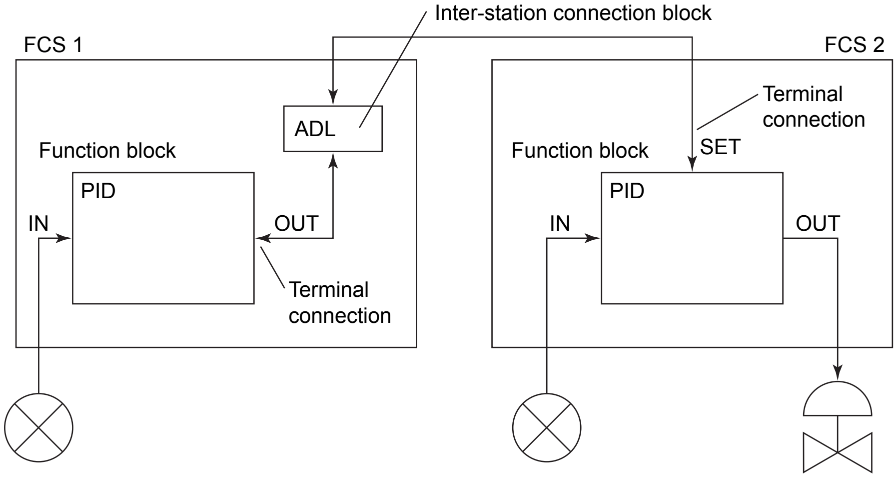


Figure 2.4-3 Connection Between Control Stations (Terminal Connection)

2.5 I/O Connection Information

The I/O connection information is specified in order to identify the connection destination of the function block's I/O terminal.

■ I/O Connection Information

This information is comprised of an element symbol name and data item name, indicating the connection destination of the I/O terminal such as a tag name, label name, element number, etc. The I/O connection information is added to the I/O terminal of the function block.

In addition, in the case of sequence connection, condition testing or status manipulation is also added to the I/O connection information.

The relationship between the connection methods and I/O connection information is as follows:

Table 2.5-1 I/O Connection Information

Connection method		I/O signal		I/O connection information (*1)
Data connection	Data reference	Process I/O		tag name/user defined label name/terminal number.data item name
		Communication I/O (*2) Expanded Communication I/O(*2)(*3)		tag name/element number.data item name
		Fieldbus I/O		tag name/user defined label name/terminal number.data item name
		Software I/O		tag name/element number.data item name
		Function block	Same control drawing	tag name.data item name
			Different control drawing	tag name.data item name
			Different control station	tag name.data item name
	Data setting	Process I/O		tag name/user defined label name/terminal number.data item name
		Communication I/O (*2) Expanded Communication I/O(*2)(*3)		tag name/element number.data item name
		Fieldbus I/O		tag name/user defined label name/terminal number.data item name
		Software I/O		tag name/element number.data item name
		Function block	Same control drawing	tag name.data item name
			Different control drawing	tag name.data item name
			Different control station	tag name.data item name
Terminal connection		Function block	Same control drawing	tag name.I/O connection terminal name
			Different control drawing	tag name.I/O connection terminal name
			Different control station	tag name.I/O connection terminal name
Sequence connection	Condition testing	Process I/O Communication I/O (*2) Expanded Communication I/O(*2)(*3) Software I/O Function block		tag name/user defined label name/terminal number/element number data item name.condition specification
	Status manipulation			tag name/user defined label name/terminal number/element number data item name.operation specification

*1: The description like A/B/... means the I/O information specification have multiple methods. However, some elements have exceptions that certain methods may not be applied.
*2: Access to the data acquired via communication with an external device using a communication module.
*3: Expanded Communication I/O can be used for only the FFCS-V.

On the function block detail builder, for the logical name of the I/O connection information with respect to the function blocks of different control stations, a (>) is added before tag name. However, when AREAOUT block is used on control drawing builder, (>) is not needed.

■ Terminal Numbers, Element Numbers

The following table shows terminal numbers and element numbers included in the I/O connection information.

Table 2.5-2 List of Terminal Numbers and Element Numbers (1/2)

Name		Symbol	Symbol syntax	
Process I/O		%Znnusmm (*1)	nn: Node number	(01 to 10) (*2) (*3)
			u: slot	(1 to 8)
			s	1 is fixed in Process I/O In case of HART compatible modules, analog input/output: s=1; HART variable: s=2.
			mm: terminal	(01 to 64)
		%Znnusmm (*4)	nn: Node number	(fixed at 01) (*5) (01 to 08) (*6)
			u: unit	(1 to 5)
			s: slot	(1 to 4)
			mm: terminal	(01 to 32)
Communication I/O (*7)	Word data	%WWnnnn	nnnn: Serial no.	nnnn: Serial no. (0001 to 4000) (*8)
	Bit data	%WBnnnnbb	nnnn: Serial no.	(0001 to 4000) (*8)
			bb: bit number	(01 to 16)
		%WBnnnnbbSddss(*9)	nnnn: Register number	(0001 to 4000)
			bb: bit number	(01 to 16)
			S	Always capital S
			dd: domain number	
			ss: station number	
Expanded communication I/O(*10)	Word data	%XWnnnn	nnnn: Register number	(0001 to 4000)
	Bit data	%XBnnnnbb	nnnn: Register number	(0001 to 4000)
			bb: bit number	(01 to 16)
		%XBnnnnbbSddss(*9)	nnnn: Register number	(0001 to 4000)
			bb: bit number	(01 to 16)
			S	Always capital S
			dd: domain number	
			ss: station number	

Name	Symbol	Symbol syntax	
Fieldbus I/O	%Znnusmm (*1)	nn: Node number	(01 to 10) (*2) (*3)
		u: slot	(1 to 8)
		s: segment	(1 to 4)
		mm: terminal	(01 to 48)
	%Znnusmm (*4)	nn: Node number	(fixed at 01) (*5) (01 to 08) (*6)
		u: unit	(1 to 5)
		s: slot	(1 to 2)
		mm: terminal	(01 to 32)

*1: A symbol for FFCS series, KFCS2 and KFCS
*2: If the database in KFCS2 is remote node expanded type, the range of node number becomes 01 to 15.
*3: The node for the I/O modules inserted in the slots of FCU is defined as a local node and the node number is 1. This is fixed and cannot be redefined. The extended node should be numbered from 2.
*4: A symbol for LFCS2, LFCS, SFCS and PFCS
*5: Can only be used for SFCS and PFCS.
*6: Can only be used for LFCS2 and LFCS.
*7: With communication I/O, the same I/O points can be accessed as word data (%WW) or bit data (%WB).
*8: For PFCS standard type, it is 0001 to 1000.
*9: Used when accessed from other stations.
*10: In FFCS-V, the communication I/Os are expanded from 4000 words to 8000 words. The expanded 4000 words are referred to as Expanded Communication I/O Data. The expanded communication I/O elements are %XW and %XB for assesses.

Table 2.5-3 List of Terminal Numbers and Element Numbers (2/2)

Name		Symbol	Symbol syntax	
Soft-ware I/O	Common switch	%SWnnnn	nnnn: Serial no.	(0001 to 4000) (except FFCS-V and PFCS) (0001 to 9000) (for FFCS-V) (0001 to 1000) (for PFCS)
	Global switch (*1)	%GSnnnmm	nnn: Serial no.	(001 to 256) (*2)
			mm: Station number	(01 to 64)
	Annunciator message	%ANnnnn	nnnn: Serial no.	(0001 to 0200) (for FFCS, FFCS-L, KFCS2, and LFCS2) (0001 to 2000) (for FFCS-V) (0001 to 0500) (for KFCS, LFCS or SFCS) (0001 to 0200) (for PFCS)
	Printout message (with data)	%PRnnnn	nnnn: Serial no.	(0001 to 1000) (for FFCS, FFCS-L, KFCS2, and LFCS2) (0001 to 2000) (for FFCS-V) (0001 to 0400) (for KFCS or LFCS) (0001 to 0200) (for SFCS) (0001 to 0100) (for PFCS)
	Operation guide message	%OGnnnn	nnnn: Serial no.	(0001 to 0500) (for FFCS, FFCS-L, KFCS2, and LFCS2) (0001 to 1000) (for FFCS-V) (0001 to 0200) (for KFCS, LFCS, and SFCS) (0001 to 0100) (for PFCS)
	Multimedia start message	%VMnnnn	nnnn: Serial no.	(0001 to 0100)
	Sequence message request	%RQnnnn	nnnn: Serial no.	(0001 to 0200) (except PFCS) (0001 to 0100) (for PFCS)
	Supervisory computer event message	%CPnnnn	nnnn: Serial no.	(0001 to 9999)
	Supervisory computer message output for PICOT	%M3nnnn	nnnn: Serial no.	(0001 to 9999)
	Signal event message	%EVnnnn	nnnn: Serial no.	(0001 to 0500) (for FFCS, FFCS-L, KFCS2, and LFCS2) (0001 to 1000) (for FFCS-V) (0001 to 0200) (for KFCS, LFCS or SFCS) (0001 to 0100) (for PFCS)

*1: The global switches are applicable in all FCSs except standard type PFCS.
*2: For ProSafe-RS SCS, the range of serial number “nnn” becomes 001 to 128.

TIP

Same as for KFCS2/KFCS, the data of function blocks can be interactively accessed between FFCS series and APCS (Advanced Process Control Station), as well as FFCS series and GSGW (General-Purpose Subsystem Communication Gateway). However, for CENTUM VP entry class, APCS is not available.

Blank Page

3. Input Processing

The function blocks are provided with various types of input processing methods to convert the input signals for the control calculation and arithmetic calculation.

In this chapter the input processing methods common to all function blocks are explained.

■ Input Processing

Input processing is a general term used for processing for the input signal read from the connection destination of an input terminal, executed by the function block before the calculation processing. There are various forms of input processing corresponding to the function block type and the input signal format.

The Regulatory Control Blocks and Calculation Blocks have the common types of input processing, and some function blocks have the particular types of input processing.

The input processing for the Sequence Tables is unique and differs from that of the Regulatory Control Blocks or Calculation Blocks.

SEE ALSO

For more information about the Sequence Tables input processing, refer to:

2.2.4, “Input Processing of Sequence Table” in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)

For more information about input processing of the function blocks with sequence connection, refer to:

2.3.4, “Input Processing of Logic Chart” in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)

■ Input Processing Common to All Regulatory Control Blocks

The Regulatory Control Blocks have the input signals processed as shown in the figure below. After the processing, the signal becomes process variable (PV).

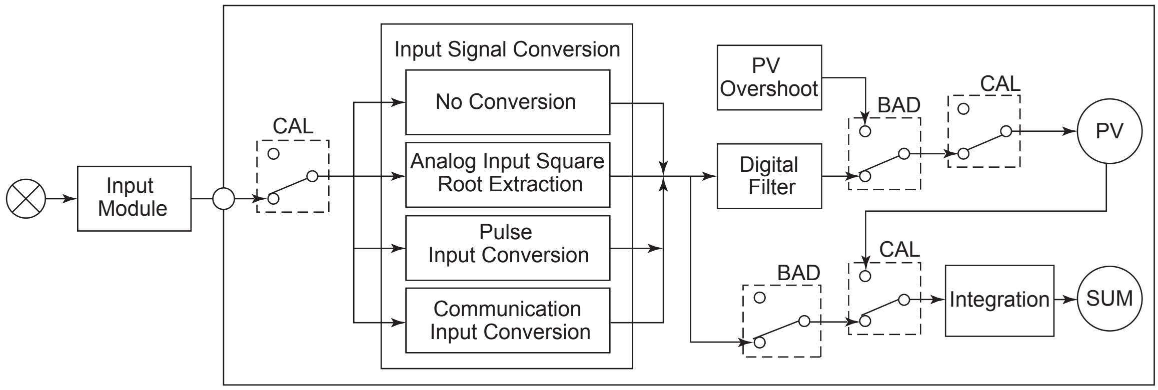


Figure 3-1 Block Chart of Input Processing Common to All Regulatory Control Blocks

■ Input Processing Common to Calculation Blocks

The Calculation Blocks have the input signals processed as shown in the figure below. The calculated input value (RV), calculated output value (CPV) or integrator value (SUM) are obtained after the input processing.

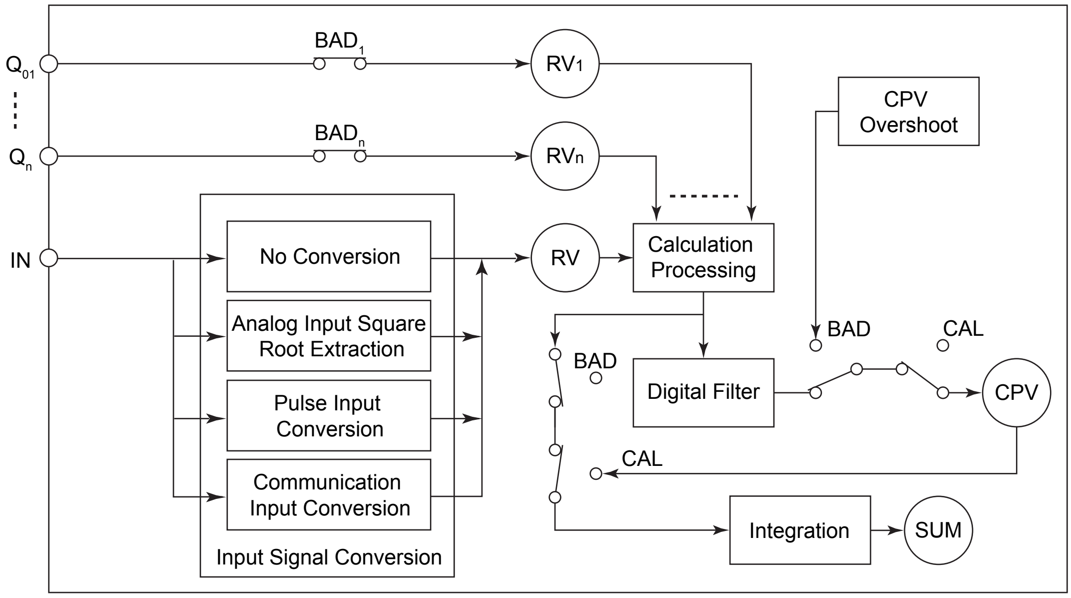


Figure 3-2 Block Chart of Input Processing Common to Calculation Blocks

■ Input Processing Common to Logic Operation Blocks

The Logic Operation Blocks (*1) have the input signals processed as shown in the figure below. The calculated input value (RV) and calculated output value (CPV) are obtained after the input processing.

*1: Logic Operation Block can be used in FCSs except PFCS.

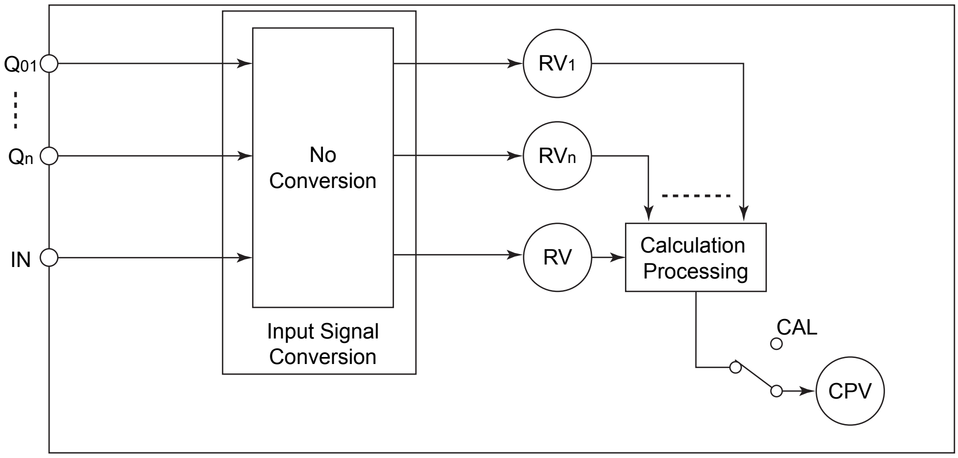


Figure 3-3 Block Chart of Input Processing Common to Logic Operation Blocks

■ Outline of Input Processing Common to Regulatory Control Block and Calculation Block

The outline of each type of input processing common to Regulatory Control Blocks and to Calculation Blocks is explained below.

● Input Signal Conversion

The input signal read from the input module or other function blocks is converted to process variable (PV) or calculated input value (RV) according to the signal type.

● Digital Filter

This digital filter executes the first-order lag processing. Input signal noise can be reduced through digital filtering process in which input signal is filtered for the Regulatory Control Blocks while value after calculation processing is filtered for the Calculation Blocks.

● Integration

The data item (SUM) is set to the integrator value. Input signal is used for the Regulatory Control Blocks while value after calculation processing is used for the Calculation Blocks are used.

● PV/FV/CPV Scale out

If the data status of input signal is invalid (BAD), the process variable (PV), feedback input value (FV) or calculated output value (CPV) is coincided with the scale high limit (SH) or scale low limit (SL) depending on the cause of invalidity (BAD).

● Calibration

For maintenance or test purposes, the process variable (PV) or calculated output value (CPV) can be set manually by using the operation and monitoring function.

■ Input Processing During Abnormal Status

The input processing during abnormal status is different from when it is normal. It is also different between Regulatory Control Blocks and Calculation Blocks.

■ Input Processing for Sequence Connection

For Logic Operation Blocks (*1) and General-Purpose Calculation Blocks (CALCU, CALCU-C), the terminal connection may be used to link the sequence. When the terminal connection is a sequence connection, the input is processed with “condition test.”

*1: Logic Operation Block can be used in FCSs except PFCS.

SEE ALSO For more information about input processing for sequence connection, refer to:

3.7, “Input Processing for Sequence Connection” on page 3-35

■ Input Processing in Different Function Blocks

The input processing supported in function blocks vary with types of function blocks.

SEE ALSO For more information about the input processing in regulatory control blocks, refer to:

“■ Input Processing Possible for Each Regulatory Control Block” in 1.1.3, “Input Processing, Output Processing, and Alarm Processing Possible for Each Regulatory Control Block” in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)

For more information about the input processing in calculation blocks, refer to:

“■ Input Processing Possible in Each Calculation Block” in 1.3.1, “Input Processing, Output Processing, and Alarm Processing Possible for Each Calculation Block” in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)

3.1 Input Signal Conversion

The input signal conversion is the function that converts the input signal read from the input module or other function blocks into process variable (PV) or calculated input value (RV) according to the signal type.

■ Type of Input Signal Conversion

▼ Input Signal Conversion

There are five kinds of common input signal conversion for the Regulatory Control Blocks and Calculation Blocks.

In addition, there are input signal conversion methods specific to particular function blocks.

The input signal conversion type can be set on the Function Block Detail Builder.

- Input Signal Conversion Type
  Select from “No Conversion,” “Square Root,” “Pulse-train,” “Control Priority Type Pulse Train Input,” “Exact Totalization Pulse Train Input” and “Communications.”

  The default setting is “No Conversion.”

● Input Signal Conversion Common to Regulatory Control Blocks and Calculation Blocks

- No Conversion
- Square Root
- Pulse-train/ Control Priority Type Pulse Train Input/ Exact Totalization Pulse Train Input
- Communications

Input signal conversion is performed only when the signal input through the input terminal is the data connection type, one of the I/O connection types. And only the signal transmitted via IN terminal (main input signal) may be converted. Furthermore, the conversion behaves differently according to the signals connected to the IN terminal.

● Input Signal Conversion of Logic Operation Blocks

- Bitwise Logic Operation Blocks, Logic Operation Blocks other than Relational Operation Blocks (*1)
- Bitwise Logic Operation Blocks (*1)
- Relational Operation Blocks (*1)

*1: Logic Operation Block can be used in FCSs except PFCS.

● Input Signal Conversion of Motor Control Blocks (MC-2, MC-2E, MC-3, and MC-3E)

- Feedback Input Signal Conversion
- Answerback Input Signal Conversion
- Feedback Input to Answerback Input Conversion

SEE ALSO For more information about Input Signal Conversion of Motor Control Blocks (MC-2, MC-2E, MC-3, and MC-3E), refer to:

1.17.1, “Input Processing of Motor Control Blocks (MC-2, MC-2E, MC-3, and MC-3E)” in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)

● Input Signal Conversion of Weight-Totalizing Batch Set Block (BSETU-3)

- Weight Measurement Conversion
- SUM Conversion
- ΔSUM Conversion

SEE ALSO For more information about Input Signal Conversion of Weight-Totalizing Batch Set Block (BSETU-3), refer to:

1.22.1, "Input Signal Conversion of Weight-Totalizing Batch Set Block (BSETU-3)" in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)

● Input Signal Conversion of Pulse Count Input Block (PTC)

- Input Signal Conversion for PTC Block

SEE ALSO For more information about Input Signal Conversion of Pulse Count Input Block (PTC), refer to:

"■ Input Signal Conversion of Pulse Count Input Block (PTC)" in 1.32, "Pulse Count Input Block (PTC)" in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)

3.1.1 Input Signal Conversions Common to Regulatory Control Blocks and Calculation Blocks

Input signal conversions common to the Regulatory Control Blocks and Calculation Blocks include “No Conversion,” “Square Root,” “Pulse-train,” “Control Priority Type Pulse Train Input,” “Exact Totalization Pulse Train Input,” “Communications,” and “PV limit.”

The following section describes the conversion methods common to Regulatory Control Blocks and Calculation Blocks.

■ No Conversion

“No Conversion” is selected if the input connection destination is neither the pulse-train input module nor communication module and the square root extraction of input signal is not needed. Also, specify “No Conversion” when the input signal is data referenced from another function block.

When “No Conversion” is selected, the input signal conversion is not performed. However, the raw data (0 to 100 % data) read from analog input modules (except those from thermocouple or RTD modules) to the IN terminal are converted into the form of specified engineering unit and scale high/low limits (SH, SL) for the process variable (PV). The raw data read from the thermocouples and resistance temperature detectors to the IN terminal are not converted. The data read from analog input modules to the input terminals other than the IN terminal are not converted either.

The table below lists the input range between each input module and the raw data.

Table 3.1.1-1 Input Range of Input Module and Raw Data : LFCS2/LFCS/SFCS/PFCS

Input/Output Module Model	Input Type	Input Range	Raw Data
AAM10	Electric Current Input	4 to 20 mA	0 to 100 %
	Voltage Input	1 to 5 V	0 to 100 %
AAM11	Electric Current Input	4 to 20 mA	0 to 100 %
	Voltage Input	1 to 5 V	0 to 100 %
AAM21	mV Input	Definable between -50 and 150 mV	0 to 100 %
	Thermocouple Input Measuring Range of Corresponding	Measuring Range of the Thermocouple	Measured Temperature
	Resistance Temperature Detector Input	Measuring Range of the RTD	Measured Temperature
	Potentiometer Input	Definable between 0 and 30000 ohm	0 to 100 %
AMC80	Voltage Input	1 to 5 V	0 to 100 %
AMM12T	Voltage Input	1 to 5 V	0 to 100 %
AMM22M	mV Input	Definable between -100 and 100 mV	0 to 100 %
AMM22T	Thermocouple Input	Measuring Range of the Thermocouple	Measured Temperature
AMM32T	Resistance Temperature Detector Input	Measuring Range of the RTD	Measured Temperature
AMM42T	Electric Current Input	4 to 20 mA	0 to 100 %

Table 3.1.1-2 Input Range and Raw Data of Input Modules : FFCS Series/KFCS2/KFCS

Type	Input/ Output Module Model	Terminal No.	I/O Type	Input Range	Raw Data
16-Channel Current Input; Non-Isolated	AAI141-S	1 to 16	Current Input	4 to 20 mA	0 to 100 %
16-Channel Current Input; Isolated	AAI143-S	1 to 16	Current Input	4 to 20 mA	0 to 100 %
8-Channel Current Input; Isolated	ASI133-S	1 to 8	Current Input	4 to 20 mA	0 to 100 %
8-Channel Current Input; Isolated	AAI135-S	1 to 8	Current Input	4 to 20 mA	0 to 100 %
16-Channel Voltage Input; Non-Isolated	AAV141-S	1 to 16	Voltage Input	1 to 5 V	0 to 100 %
16-Channel Voltage Input; Non-Isolated	AAV142-S	1 to 16	Voltage Input	Definable within -10 to 10 V	0 to 100 %
16-Channel Voltage Input; isolated	AAV144-S	1 to 16	Voltage Input	1 to 5 V	0 to 100 %
16-Channel Voltage Input (-10 to 10 V); isolated	AAV144-S	1 to 16	Voltage Input	Definable within -10 to 10 V	0 to 100 %
16-Channel Ther-mocouple/mV Input; Isolated	AAT141-S	1 to 16	Thermocouple Input	Rated range	Measured Temperature
			mV Input (%)	Definable within -100 to 150 mV	0 to 100 %
			TC input (V)	-20 to 80 mV	Engineering Unit (V)
12-Channel Ther-mocouple Input; Isolated	AAR181-S	12	RTD Input	Rated range	Measured Temperature
			RTD Input (ohm)	0 to 400 ohm	Engineering Unit (ohm)
16-Channel Ther-mocouple/mV Input; Isolated	AAT145-S	1 to 16	Thermocouple Input	Rated range	Measured Temperature
			mV Input (%)	Definable within -100 to 150 mV	0 to 100 %
			TC input (V)	-20 to 80 mV	Engineering Unit (V)
15-Channel Ther-mocouple Input; Isolated (MX Com-patible)	AAT145-S	1 to 15 (*1)	Thermocouple Input	Rated range	Measured Temperature
			TC input (V)	-20 to 80 mV	Engineering Unit (V)
16-Channel RTD/ Potentiometer Input; Isolated	AAR145-S	1 to 16	RTD Input	Rated range	Measured Temperature
			Potentiometer Input	Definable within 0 to 10Kohms	0 to 100 %
			RTD Input (ohm)	0 to 400 ohm	Engineering Unit (ohm)
16-Channel Ther-mocouple/mV Input; Isolated	AST143-S	1 to 16	Thermocouple Input	Rated range	Measured Temperature
			mV Input (%)	Definable within -100 to 150 mV	0 to 100 %
			TC input (V)	-50 to 75 mV	Engineering Unit (V)

Type	Input/ Output Module Model	Terminal No.	I/O Type	Input Range	Raw Data
8-Channel RTD/ Potentiometer Input; Isolated	ASR133-S	1 to 8	RTD Input	Rated range	Measured Temperature
			Potentiometer Input	Definable within 0 to 10 Kohms	0 to 100 %
			RTD Input (ohm)	Choose from 0 to 650, 0 to 1300, 0 to 2600, 0 to 5200	Engineering Unit (V)
8-Channel Pulse Input	AAP135-S	1to 8	Pulse Input	Number of pulse 0 to 65535; Time stamp (1 ms)	Number of pulse (with time stamp)
16-Channel Pulse Input (PM1 Compatible)	AAP149-S	1 to 16	Pulse Input	Number of pulse 0 to 65535; Time stamp (1 ms)	Number of pulse (with time stamp)
8-Channel Current Input and 8-Channel Current Output; Non-isolated	AAI841-S	1 to 8	Current Input	4 to 20 mA	0 to 100 %
		9 to 16	(Current Output)	–	–
8-Channel Voltage Input and 8-Channel Current Output; Non-isolated	AAB841-S	1 to 8	Voltage Input	1 to 5 V	0 to 100 %
		9 to 16	(Current Output)	–	–
8-Channel Voltage Input and 8-Channel Current Output; Non-isolated (MAC2 Terminal Arrangement)	AAB841-S	1,3,5.. 15 Odd numbers	Voltage Input	1 to 5 V	0 to 100 %
		2,4,6.. 16 Even numbers	(Current Output)	–	–
4-Channel Current Input and 4-Channel Current Output; Isolated	AAI835-S	1 to 4	Current Input	4 to 20 mA	0 to 100 %
		5 to 8	(Current Output)	–	–
8-Channel Pulse Input 8-Channel Current Output (PAC Compatible)	AAP849-S	1,3,5.. 15 Odd numbers	Pulse Input	Number of pulse 0 to 65535; Time stamp (1 ms)	Number of pulse (with time stamp)
		2,4,6.. 16 Even numbers	(Current Output)	–	–
16-Channel Current Input; HART (*2)	AAI141-H	1 to 16	Current Input	4 to 20 mA	0 to 100 %
		1 to 32	HART Variable	–	Engineering Unit
8-Channel Current Input; Isolate channels; HART (*2)	AAI135-H	1 to 8	Current Input	4 to 20 mA	0 to 100 %
		1 to 32	HART Variable	–	Engineering Unit
16-Channel Current Input; Isolated; HART (*2)	AAI143-H	1 to 16	Current Input	4 to 20 mA	0 to 100 %
		1 to 32	HART Variable	–	Engineering Unit
8-Channel Current Input; Isolated; HART (*2)	ASI133-H	1 to 8	Current Input	4 to 20 mA	0 to 100 %
		1 to 32	HART Variable	–	Engineering Unit

Type	Input/ Output Module Model	Terminal No.	I/O Type	Input Range	Raw Data
8-Channel Current Input; 8-Channel Current Output; HART(*2)	AAI841-H	1 to 8	Current Input	4 to 20 mA	0 to 100 %
		9 to 16	(Current Output)	–	–
		1 to 32	HART Variable	–	Engineering Unit
4-Channel Current Input; 4-Channel Current Output; HART(*2)	AAI835-H	1 to 4	Current Input	4 to 20 mA	0 to 100 %
		5 to 8	(Current Output)	–	–
		1 to 32	HART Variable	–	Engineering Unit
16-Channel Voltage and Current Input; HART(*2)	AAB141-H	1 to 16	Voltage Input or Current In-put(*3)	1 to 5 V or 4 to 20 mA	0 to 100 %
		1 to 32	HART Variable	–	Engineering Unit
8-Channel Voltage and Current Input;8-Channel Current Output; HART(*2)	AAB842-H	1 to 8	Voltage Input or Current In-put(*3)	1 to 5 V or 4 to 20 mA	0 to 100 %
		9 to 16	Current Output	4 to 20 mA	–
		1 to 32	HART Variable	–	Engineering Unit
8-Channel Voltage and Current Input;8-Channel Current Output; HART (MAC2 Terminal Ar-rangement)(*2)	AAB842-H	1,3,5.. 15Odd numbers	Voltage Input or Current In-put(*3)	1 to 5 V or 4 to 20 mA	0 to 100 %
		2,4,6.. 16Even numbers	Current Output	4 to 20 mA	–
		1 to 32	HART Variable	–	Engineering Unit

*1: The 16th channel of AAT145 is used as cold junction compensation terminal, so that only 15 channels of the temperature signals from the field can be connected.
*2: On IOM Builder for Analog Input and Output (HART Compatible) modules, terminal number is indicated as %Znnusmm. When “s” is 1, the terminal is used as an analog input or output (Current Input/Current Output) channel. When “s” is 2, the terminal is used as a HART variable channel.
*3: Voltage input or current input for each input channel can be selected on IOM builder. The default is voltage input.

If the input terminal connected to the process I/O is not IN terminal, the data is not converted into engineering unit format, and the range of input signal is fixed to the raw data range shown in the above table. The terminals of the function blocks that do not convert input data into engineering unit format are listed in the table below.

Table 3.1.1-3 Terminals of Function Blocks that do not Convert Data into Engineering Unit Format

Terminal	Function Block
BIN/TIN	PID, PI-HLD, PID-BSW, ONOFF, ONOFF-E, ONOFF-G, ONOFF-GE, PID-TP, PD-MR, PI-BLEND, MLD, MLD-PVI, MLD-SW, RATIO, FFSUM, XCPL
Q1 to Q8	ADD, MUL, DIV, AVE, TPCFL (Temperature, Pressure), ASTM1 (Temperature), ASTM2 (Temperature), CALCU

■ Analog Input Square Root Extraction

The square root extraction of analog input signal can be performed in the function block.

For example, if a differential pressure type flow meter is used, the square root extraction is normally executed in order to convert the analog input signal that indicates differential pressure (differential pressure signal) into the signal that indicates flow (flow signal).

Shown below is the image of analog input square root extraction.

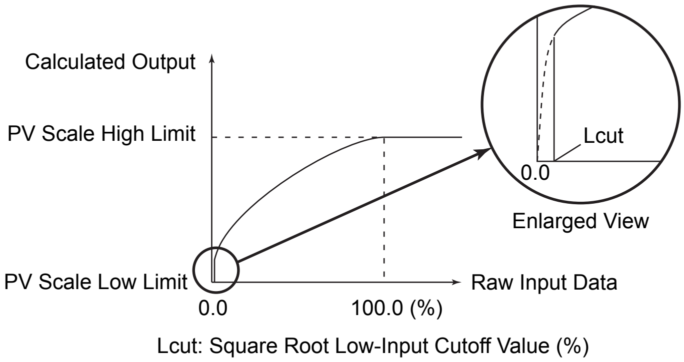


Figure 3.1.1-1 Analog Input Square Root Extraction

Set a square root calculation low-input cut value when performing an analog input square root calculation.

This function changes the value after square root calculation to 0 when the input signal is below the low-input cut value.

The setup for square root calculation low-input cut value can be executed on the Function Block Detail Builder.

Square root calculation low-input cut value:	Set at 0.0 to 100.0 %. The default setting is 0.5 %.

Note that the square root calculation low-input cut value can be set only when “Square Root” is selected as the input signal conversion type.

● Regarding to Square Root Extraction in I/O Module : LFCS2/LFCS/SFCS/PFCS

Square root calculation can be performed in the AAM11 type current/voltage input module. Do not select “Square Root” conversion for the function blocks connected to the AAM11 current/voltage input modules where the square root conversion is already defined on the IOM Builder.

Since AMC80 multi-point control analog I/O module and AAM10 current/voltage input module are not provided with square root extraction function, “Square Root” conversion need to be specified in the function blocks connected to the modules if the square root extraction is required.

● Regarding to Square Root Extraction in I/O Module : FFCS Series/KFCS2/KFCS

The I/O modules for FFCS series, KFCS2 and KFCS do not have Square Root Extraction function. If square root extraction is required, the conversion can be performed in the function block connected to the I/O module by selecting “Square Root” as the input signal conversion on Function Block Builder.

■ Pulse-Train Input Conversion

A process variable (PV) is calculated based on the integrated pulse count value (P) read from the pulse-train input module and its measurement time (t).

The pulse-train input processing calculates PV engineering unit data using the integrated pulse count value (P) stored in sequence in the pulse input buffer and its measurement time (t).

Pulse train input conversion is provided with the following three methods

- Control priority type pulse train input conversion (PULSE)

The accurate measured process variable (PV) and the calculated input value (RV) may be obtained.

- Exact totalization pulse train input conversion (QTPUL)
  The accurate integrator value (SUM) may be obtained.

- Pulse train input conversion (BTHPUL)
  Both conversion methods, i.e., control priority type pulse train input conversion and exact totalization pulse train input conversion are applied. The accurate measured process variables (PV) and the calculated input values (RV) are obtained by control priority type pulse train input conversion while the accurate integrator value (SUM) is obtained by exact totalization pulse train input conversion.

  When applying the pulse train input conversion (BTHPUL) to the following function blocks, it only functions to obtain the calculated input values (RV) same as obtained by control priority type pulse train input conversion.

  ADD, MUL, DIV, SQRT, EXP, LAG, INTEG, LD, RAMP, LDLAG, DLAY, DLAY-C, AVE-M, AVE-C, FUNC-VAR, TPCFL, ASTM1, ASTM2

It is required to specify the conversion method to exact totalization pulse train input conversion (QTPUL) when the converted process variable (PV) or calculated output value (CPV) are used by other function blocks for totalization. Otherwise, the totalized value may result deviation.

However, the pulse rate and size of pulse-train input buffer are the same for all the tree methods of pulse conversion.

Control Priority Type Pulse-Train Input Conversion

Shown below is the block chart of the pulse-train input conversion processing.

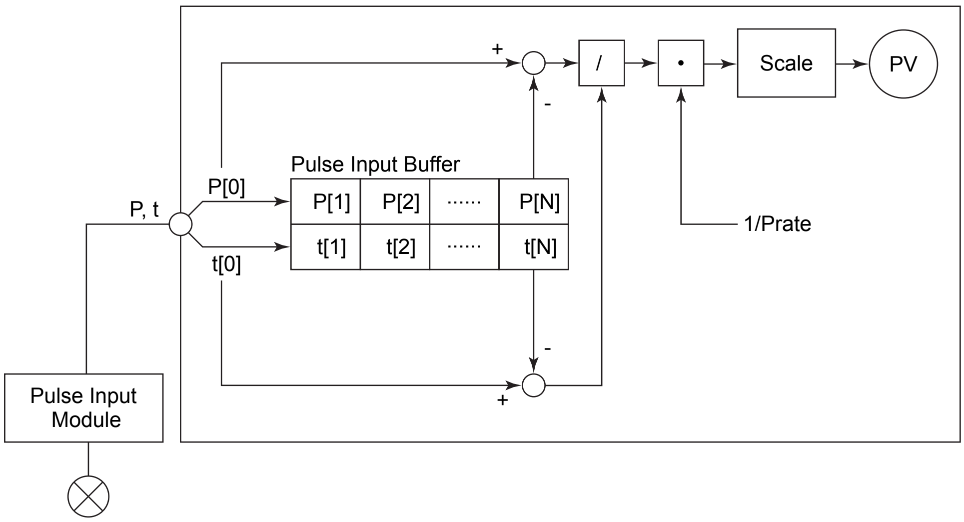


Figure 3.1.1-2 Block Chart of Pulse-Train Input Conversion Processing

The following is the computational expression for the pulse train input conversion:

$$PV=\frac{P[0]-P[N]}{t[0]-t[N]} \cdot \frac{1}{Prate} \cdot (SH-SL)+SL$$

Figure 3.1.1-3 (Example of) the computational expression for the pulse train input conversion

PV	:	process variable (engineering unit)
P[0]	:	current integrated pulse count value
P[N]	:	integrated pulse count value before N scan period
t[0]	:	current integrated pulse count value measurement time

t[N] : integrated pulse count value measurement time before N scan period

Prate : pulse rate (Hz)

SH : PV scale high limit

SL : PV scale low limit (process variable when input pulse frequency is 0 Hz)

N : size of pulse input buffer

● Exact Totalization Pulse Train Input Conversion

Shown below is the block chart of the exact totalization pulse train input conversion processing.

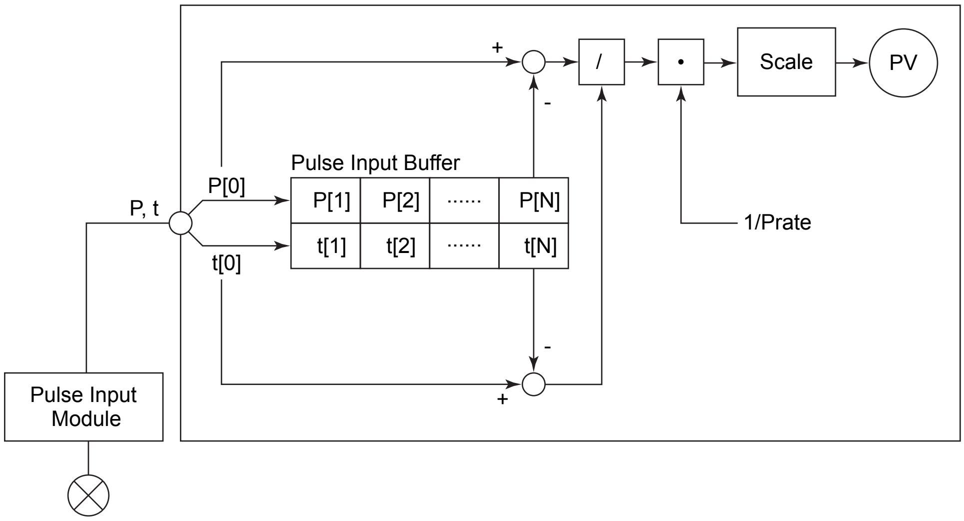


Figure 3.1.1-4 Block Chart of Exact Totalization Pulse Train Input Conversion Processing

The following is the computational expression for the exact totalization pulse train input conversion:

$$PV=\frac{P[0]-P[N]}{N \bullet Ts} \bullet \frac{1}{Prate} \bullet (SH-SL)+SL$$

Figure 3.1.1-5 (Example of) the computational expression for the exact totalization pulse train input conversion

PV : process variable (engineering unit)

P[0] : current integrated pulse count value

P[N] : integrated pulse count value before N scan period

Prate : pulse rate (Hz)

SH : PV scale high limit

SL : PV scale low limit (process variable when input pulse frequency is 0 Hz)

N : size of pulse input buffer

Ts : scan period

With exact totalization pulse train input conversion, the process value (PV) may not stabilize and oscillate during operation, particularly during high-speed scan periods. In this situation, the oscillation of the process value (PV) can be minimized by enlarge the size of input buffer.

● Pulse Rate (Prate)

Pulse rate refers to the input frequency measured when the process variable reaches the scale high limit. It is indicated in the unit of Hz.

The setup for pulse rate can be executed on the Function Block Detail Builder.

Pulse Rate: Set a value within the range between 0.10 and 10000.00 Hz.
The default value is 1 Hz.

The following is the computational expression for pulse rate:

```
Prate = (SH-SL) • (pulse conversion factor)
```

An example of pulse rate calculation is as follows:

If the range between process variable is 0 to 2 kℓ/min and the pulse conversion factor for the flow meter is 2.54 pulse/ℓ, the range between process variables is converted into the time unit (sec.) used for pulse rate as follows.

SL=0

$$SH=2\ (k\ell/min)=\frac{2}{60}\ (k\ell/sec)$$

Figure 3.1.1-6 Example of pulse rate calculation_1

The pulse conversion factor is converted into the flow unit (kℓ) used for process variables.

```
Pulse conversion factor = 2.54 pulse/ℓ = 2.54 • 1000 pulse/kℓ
```

The pulse rate is then calculated by assigning the range between process variable and the pulse conversion factor to the pulse rate computational expression.

$$Prate=\frac{2}{60}\bullet 2.54\bullet 1000=84.67\ (Hz)$$

Figure 3.1.1-7 Example of pulse rate calculation_2

● Pulse Train Input Buffer (N)

▼ Number of Input Buffers

If the pulse rate (input pulse frequency) is low, the instantaneous process variable obtained based on the integrated pulse count values in a short interval will have a large error. In the exact totalization pulse train input conversion, the size of pulse train input buffer (N) is automatically determined so that a suitable value can be obtained for the sample cycle (t[0] - t[N]) according to the pulse rate.

The table below lists the relation between the pulse rate and the size of pulse train input buffer (N) when “Auto” is selected for the pulse train input buffer (N).

Table 3.1.1-4 Pulse Rate and Size of Pulse Train Input Buffer

Pulse Rate (Prate)	Size of Pulse Input Buffer (N)
Prate≤10 Hz	10
10 Hz<Prate<1 KHz	5
Prate≥1 KHz	3

The setup for pulse train input buffer (N) can be executed on the Function Block Detail Builder.

Pulse Train Input Buffer (N): Select from the range between 1 and 10 or “AUTO.”
The default is “AUTO.”

■ Communication Input Conversion

The communication input conversion performs “Data Conversion” and “High/Low-Limit Check” for the input data read from the communication input modules.

● Data Conversion

With communication input, the raw input data read from the input terminal may be in the format specific to the connection destination. Those data need to be converted into process variables (PV) in the engineering unit.

The following is the computational expression for the communication input conversion:

`Y = GAIN • X+BIAS`

`Y:` PV (engineering unit)

`X:` data read from communication input module

`GAIN:` data conversion gain

`BIAS:` data conversion bias

The setup for data conversion gain and data conversion bias can be executed on the Function Block Detail Builder.

- Data Conversion Gain
  Set a 7-digit numeric value using a sign and a decimal point.

  Setting range is between -9999999 and 9999999

  The default setting is 1.000.

- Data Conversion Bias
  Set a 7-digit numeric value using a sign and a decimal point.

  Setting range is between -9999999 and 9999999

  The default setting is 0.000.

● High/Low-Limit Check

The high-low limit check is executed in order to induce the input open alarm status in the function block.

The following conditions are the input high limit set value or the input low limit set value trigger alarms or reset alarms.

- Condition for high limit input open (IOP)
  Y > input high limit detection set value

  Input high limit detection set value = 106.25 % (PV scale span)

- Condition for low-limit input open (IOP-)
  Y < input low limit detection set value

  Input low limit detection set value = -6.25 % (PV scale span)

- Condition of recovering from status IOP
  Y < input high limit detection set value - hysteresis value

- Condition of recovering from status IOP-
  Y > input high limit detection set value + hysteresis value

The input high-limit detection set value and the input low-limit detection set value can also be changed on the Function Block Detail Builder.

- Input high limit detection set value

To be set between -25.0 and 125.0 %

The default settings are 106.25 %

- Input low limit detection set value
  To be set between -25.0 and 125.0 %

  The default settings are -6.25 %

The hysteresis value is the same value used for PV high/low-limit alarm (HI, LO).

■ PV Range Limit: FFCS Series/KFCS2/LFCS2

▼ PV Range Limit

The measure value (PV), feedback value (FV) and calculation value (RV) are limited by the high limit of the scale (SH) and the low limit of the scale (SL). If a value is greater than SH, the SH value will be used; while if a value is smaller than SL, the SL value will be used.

TIP

In CENTUM V and CENTUM-XL, the PV/FV/RV are always be limited in the range of SH and SL. For keeping the system consistency, the PV Range Limit may be applied.

PV Range Limit can be applied to the signal of IN terminal (the main input terminal). For the motor controller blocks (MC-2, MC-2E, MC-3, MC-3E), the PV Range Limit can only be applied to the signal from FB (feedback) terminal.

For dual-redundant signal selector block, the PV Range Limit can be applied to the signals from either RV1 or RV2 terminal.

If PV/FV/RV values are forced from other blocks or HIS, the PV/FV/RV will not be limited. PV Range Limit is executed periodically. When online changing SH/SL or PV Range Limit settings. The changed settings will not become valid right after changing but become valid at the scan timing of the function block.

For the calculation blocks, PV Range Limit can only be applied to the general-purpose calculation blocks (CALCU, CALCU-C) or the data set block with input indicator (DSET-PVI).

IMPORTANT

- For the general-purpose calculation blocks (CALCU, CALCU-C), when PV Range Limit is applied, the calculation input will be limited by the SH and SL of the calculation output (CPV).
- When the Input Signal Conversion is changed to Communication Input, the PV Range Limit will not function.
- When PV Range Limit is applied, the digital filter and totalization calculation will be based on the limited input signals. However, for the pulse inputs of [Exact totalization pulse train input conversion (QTPUL)] and [pulse train input conversion (BTHPUL)], the signals before PV Range Limit will be used.

PV Range Limit can be set on function block detailed builder.

PV Range Limit: Yes/No
The default is “No.”

3.1.2 Input Signal Conversion for Logic Operation Blocks

Input Signal Conversion for Logic Operation Block (*1) includes 3 types, they are “Convert to Integer”, “No Conversion (in Hex.)” and “No Conversion.”

*1: Logic Operation Block can be used in FCSs except PFCS.

■ Input Signal Conversion for Logic Operation Blocks (except for Bitwise Logic Operation Blocks and Relational Operation Blocks)

“Convert to Integer” is fixed for this type of blocks.

The input data from the input connection terminal is converted to calculated input value (RV). If the connection of blocks are reference type, the referred data is converted to the integer and the first digit after decimal point is round off.

■ Input Signal Conversion for Bitwise Logic Operation Blocks

“No Conversion (in Hex.)” is fixed for this type of blocks.

Only a certain types of data are allowed to be connected to the input terminals or to be connected via reference connection. The input processing and the integration functions are not provided. For the data in the connected destination function blocks, only data reference connection type may be applied.

Input signal: Binary string (Integer)

Calculated Input Value (RV) is displayed in hexadecimal in 8 digits.

■ Input Signal Conversion for Relational Operation Blocks

“No Conversion” is fixed for this type of blocks.

Only a certain types of data are allowed to connected to the input terminals or for the reference connection. The input processing and the integration functions are not provided. For the data in the connected destination function blocks, only data reference connection type may be applied.

3.2 Digital Filter

The digital filter is a function to remove the noises from process input signals.

Digital Filter

Input Signal Filtering

The digital filter is a function in which the input signal is processed by the first-order lag filter in order to reduce input signal noise.

Digital Filter for Regulatory Control Block

In the Regulatory Control Blocks, the filtering process is executed for input signal (main input signal) read from the IN terminal only, following input signal conversion.

Digital Filter for Calculation Block

In the Calculation Blocks, the digital filter processing is executed for the General-Purpose Calculation Blocks (CALCU, CALCU-C) and the Data Set Block with Input Indicator Block (DSET-PVI) only. Each block uses a different filtering method.

- In the General-Purpose Calculation Blocks, the digital filter processing is executed following calculation processing.
- In the Data Set Block with Input Indicator Block, the filtering process is executed for input signal (main input signal) read from the IN terminal only, following input signal conversion.

Computational Expression for Digital Filter

The following is the computational expression for the digital filter:

$$Y_n=(1-\alpha) \cdot X + \alpha \cdot Y_{n-1}$$

α: Filter coefficient

X: Input value

Y_n: Current filtering data

Y_{n-1}: Previous filtering data

Shown below is the step response of digital filtering process.

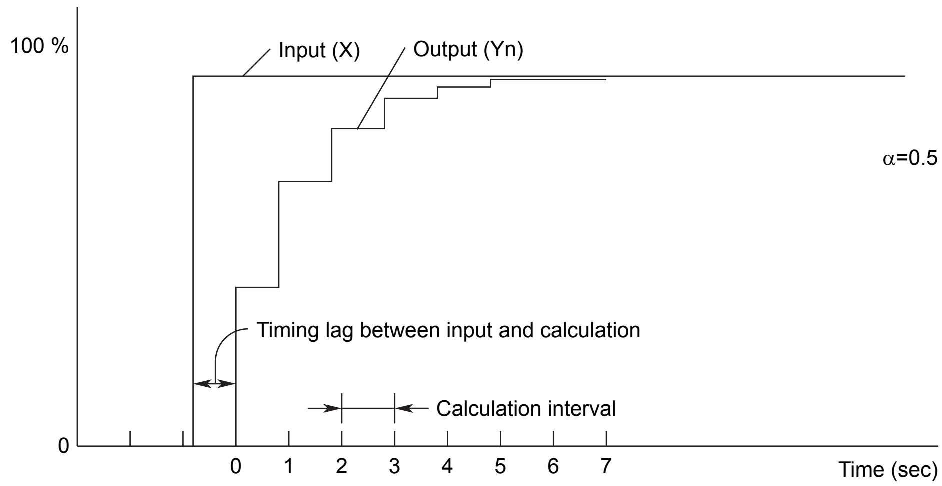


Figure 3.2-1 Step Response of Digital Filtering Process

● Digital Filter Coefficient

▼ Digital Filter Coefficient 1 to 3

There are three kinds of digital filter coefficients. These digital filter coefficients are set by the FCS Constants Builder for each FCS.

- Digital Filter Coefficient 1: 0 to 1.00 (0.01 unit)
- Digital Filter Coefficient 2: 0 to 1.00 (0.01 unit)
- Digital Filter Coefficient 3: 0 to 1.00 (0.001 unit)

The defaults for these digital filter coefficients are set to the values indicated below.

- Digital Filter Coefficient 1: 0.5
  (When the digital filter coefficient is 0.5 and scan period is 1 second, the time constant is 1 second)
- Digital Filter Coefficient 2: 0.75
  (When the digital filter coefficient is 0.75 and scan period is 1 second, the time constant is 3 seconds)
- Digital Filter Coefficient 3: 0.875
  (When the digital filter coefficient is 0.875 and scan period is 1 second, the time constant is 7 seconds)

When high-speed scan is used, the time constant changes in accordance with the scan period. Since the scan period is getting shorter at high-speed scan rate, the time constant is getting smaller accordingly.

For input indicator blocks (PVI), input indicator blocks with deviation alarm (PVI-DV), general-purpose calculation blocks (CALCU), general-purpose calculation blocks with string I/O (CALCU-C), if scan coefficient is specified as 2 or greater on the Function Block Detail Builder, the digital filtering coefficient should be multiplied by the specified scan coefficient.

■ Input Filter Specification

▼ Input Signal Filtering

The digital filter may be defined for each function block in “Input Signal Filtering” on the Function Block Detail Builder.

Input Signal Filtering: Select from "None", "Auto", "1"," 2", or "3"
The default setting is “Auto.”

Given below are the actions performed for each type of the input signal filtering.

- Auto
  If the IN terminal is connected to I/O module including communication module, “Digital Filter Coefficient 1” is used. If the IN terminal is connected to neither communication module nor I/O module, no filtering process is performed.
- None
  No filtering process is performed.
- 1
  Digital Filter Coefficient 1 is used.
- 2
  Digital Filter Coefficient 2 is used.
- 3
  Digital Filter Coefficient 3 is used.

3.3 Integration

Integration refers to the function in which the input signal or the value after calculation processing is integrated.

■ Integration

The integration processing for each of the function blocks is indicated below.

● Integration for Regulatory Control Block

In the Regulatory Control Blocks, the integration process is executed for input signal (main input signal) read from the IN terminal only, following input signal conversion.

The integration process in BSETU-2 and BSETU-3 is different from other regulatory control blocks.

SEE ALSO For more information about integration process in BSETU-2 and BSETU-3, refer to:

1.20.1, "Input Processing of Totalizing Batch Set Blocks (BSETU-2, BSETU-3)" in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)

● Integration for Calculation Block

In the Calculation Blocks, the integration processing is executed for the General-Purpose Calculation Blocks (CALCU, CALCU-C) and the Data Set Block with Input Indicator (DSET-PVI) only. Each block uses a different filtering method.

- In the general-purpose calculation blocks, the calculation output signals are integrated before the processing of the digital filter. If the input signal conversion is specified as [Extract Totalization Pulse Train Input], the calculation input value can be integrated.
- In the Data Set Block with Input Indicator, the integration process is executed for input signal (main input signal) read from the IN terminal only, following input signal conversion.

■ Computational Expression for Integration

▼ Totalizer

The following is the computational expression for the integration:

$$SUM_n = X \cdot \frac{T_s}{T_k} + SUM_{n-1}$$

Figure 3.3-1 (Example of) the computational expression for the integration

X : Integrated input signal

Input value after input signal conversion. However, PV value if the PV data status is CAL.

SUM_n : Current integrator value

SUM_{n-1} : Previous integrator value

T_s : Scan period (sec)

T_k : Time scale conversion coefficient

● Time Scale Conversion Coefficient

▼ Totalizer Time Unit

The time scale conversion coefficient (Tk) is set corresponding to the totalizer time unit.

The table below lists the correlation between the time scale conversion coefficient and the totalizer time unit.

Table 3.3-1 Time Scale Conversion Coefficient and Totalizer Time Unit

Totalizer time unit	Time Scale Conversion Coefficient (Tk)
Second	1
Minute	60
Hour	3600
Day	86400

The time scale conversion coefficient (Tk) is automatically determined when the totalizer time unit is set on the Function Block Detail Builder. The totalizer time unit must be set in the same unit as the process variable (PV). For example, if the unit of PV is "m^3/min," set the totalizer time unit to "minute"

- Number of digits for integrator value
  Up to 8 digits can be used. If the integrator value exceeds 8 digits, the value returns to 0 and the integration processing continues.
- A negative integration input signal value can be integrated as a negative value. However, integration of negative values can be executed only when the low-input cutoff value is negative.
- Unit
  Engineering unit is used.

The totalizer time unit can be defined on the Function Block Detail Builder.

- Totalizer Time Unit
  Select "Second," "Minute," "Hour," "Day" or "None."

  The default setting is "None," however for the BSETU-2 block the default setting is "Hour."

  If "None" is specified as the totalizer time unit, integration will not be executed.

● Low-Input Cut

The integration operation differs by the integration low-input cut value setting as explained below.

- If the low-input cut value is positive (including 0)
  Integration is not executed for the input signal (including negative value) less than the low-input cut value.
- If the low-input cutoff value is negative
  Integration is not executed for the input signal if the absolute value of the input signal is less than that of the low-input cut value.

When the integration of the reverse direction flow measurement (negative value input) is allowed, integration cannot be executed for small flow in either direct or reverse direction if a negative value is set to the low-input cut value.

The low-input cut value can be specified on the Function Block Detail Builder.

- Totalizer Low-Input Cut Value
  Set the data in the same unit of integrator value (PV), or percentage value for the PV scale span.

  If a percentage value is used, add % after the value.

  The default setting is 0 %.

SUM Value Entry

▼ SUM Value Entry

When SUM value entry is allowed, operator can enter a value from the instrument faceplate and tuning view on HIS to the data item SUM when the calculation block is not performing integration. This is irrelevant to whether calculation input signal or calculation output signal is used for integration.

On the Function Block Detail builder, the [SUM Value Entry] can be set to [Allowed] or [Not Allowed] after the setting of [Totalizer Time Unit] is set to [No].

SUM Value Entry: Choose [Allowed] or [Not Allowed].
The default is [Not Allowed].

This setting does not affect the FCS integration.

3.4 PV/FV/CPV Overshoot

The PV/FV/CPV overshoot refers to the function in which the process variable (PV/FV) or the calculated output value (CPV) is coincided with the scale high-limit (SH) or the scale low-limit (SL) when the status of input signal is invalid (BAD).

This section describes PV, FV and CPV overshoot.

■ PV Overshoot

▼ PV Overshoot

When the data status of input signal becomes invalid (BAD), the PV overshoot function overshoots the process variable (PV), or upscales it to scale high-limit or downscales it to scale low-limit. The PV overshoot is supported only for the Regulatory Control Blocks.

Since the PV overshoot is for process input signal, it is executed when the I/O connection type is process I/O.

The following table shows the relationship between the cause for invalidity (BAD) and process variable (PV) when the PV overshoot is used.

Table 3.4-1 Reason for Invalidity (BAD) and Overshoot Value

Cause of invalidity (BAD)	Overshoot
High-limit input open (IOP+)	Upscale to high-limit (SH)
Low-limit input open (IOP-)	Downscale to low-limit (SL)
Process I/O failure or other error	

The PV overshoot can be specified on the Function Block Detail Builder.

PV Overshoot: Select “Overshoot PV” or “Holding PV.”
The default setting is “Holding PV.”

With “Holding PV,” when the data status of process variable (PV) becomes invalid, the last good process variable is held.

Furthermore, when the input signal is not a process input signal, the operation becomes “Holding PV” even though “Overshoot PV” is specified.

■ FV Overshoot

▼ FV Overshoot

When the data status of input signal becomes invalid (BAD), the feedback input value (FV) may be overshot to be the same as the scale high-limit or the scale low-limit. The FV Overshoot is only available for motor control block.

The FV overshoot can be specified on the Function Block Detail Builder.

FV Overshoot: Choose [Holding FV Value] or [Overshoot FV Value].
The default is [Holding FV Value].

■ CPV Overshoot

When the status of input signal is invalid (BAD). The CPV overshoot function overshoots the calculated output value (CPV), or upscales it to the scale high-limit (SH) or downscales it low-limit (SL). CPV overshoot is supported for the General-Purpose Calculation Blocks (CALCU, CALCU-C) and the Data Set Block with Input Indicator Block (DSET-PVI), the Analog Calculation Blocks and the Arithmetic Calculation Blocks except Averaging Block (AVE).

Since the CPV overshoot is for process input signal, it is executed when the I/O connection type is process I/O.

The following table shows the relationship between the cause for invalidity (BAD) and calculated output value (CPV) when the CPV overshoot is used.

Table 3.4-2 Reason for Invalidity (BAD) and Overshoot Value

Cause of invalidity (BAD)	Overshoot
High-limit input open (IOP+)	Upscale to high-limit (SH)
Low-limit input open (IOP-)	Downscale to low-limit (SL)
Process I/O failure or other error	

The CPV overshoot can be specified on the Function Block Detail Builder.

PV Overshoot: Choose [Overshoot PV] or [Holding PV].
The default is [Holding PV].

With “Holding PV,” when the data status of calculated output value (CPV) becomes invalid, the last good calculated output value is held.

Furthermore, when the input signal is not a process input signal, the operation becomes “Holding PV” even though “Overshoot PV” is specified.

3.5 Calibration

The calibration is a function in which the emulated signals for process variables (PV) or calculated output values (CPV) in the function block can be set manually with the operation and monitoring function for maintenance or test purpose. The state in which calibration is being executed is called calibration status.

The calibration mode differs between Regulatory Control Blocks and Calculation Blocks.

IMPORTANT

If the IN terminal of a function block is not connected, the functional block should not be put into the calibration mode. If the function block is put into the calibration mode, the SINT cannot be released from the PV status. Moreover, the data of this function block should not be reference by any calculation.

■ Calibration for Regulatory Control Block

In Regulatory Control Blocks, calibration is executed when the data status of process variable (PV) is set to calibration (CAL) by the operating and monitoring function.

The following are the indications of Regulatory Control Blocks in calibration status:

- The color of the operation and monitoring function PV bar display turns to cyan.
- A process variable (PV) can be set manually.
- The integration is continued with the process variable (PV) entered.
- The alarm check for the process variable (PV) entered is bypassed.
- In the function block with manual mode (MAN), the block mode switches to manual mode. The function block falls to manual (MAN) mode if it has manual mode. Nevertheless, if the option of [Allow to change function block to PRD mode while CAL is on (XL compatible)] is enabled, the function block in primary direct (PRD) mode will retain the PRD mode while the function block switches to calibration status.
- In the case of Motor Control Blocks (MC-2, MC-2E, MC-3, MC-3E), feedback input signal processing and answerback input signal processing are stopped. In this case, the answerback raw signal (RAW) follows the input signal.

The following occurs when the Weight-Totalizing Batch Set Block (BSETU-3) changes to the calibration mode.

- Absolute integrator value (SUM0) and integrator value (SUM) can be set manually.
- Block mode changes to manual (MAN) mode.

■ Calibration for Calculation Block

In Calculation Blocks, calibration is executed when the data status of calculated output value (CPV) is set to calibration (CAL) by operating the operation and monitoring function.

The Calculation Blocks in calibration status behaves as follows:

- The color of the operation and monitoring function CPV bar display turns to cyan.
- The calculated output value (CPV) can be entered manually.
- The integration is continued with the calculated output value (CPV) entered.
- The alarm check for the calculated output value (CPV) entered can be bypassed.
- The calculation stops while the block is still in automatic mode (AUT).

- The output of the secondary calculated output values (CPV1 to CPVn) stops.
- The output of the primary calculated output value (CPV) is processed as usual.

There is no calibration in the Calculation Auxiliary Blocks except the Data set block with input indicator (DSET-PVI).

The block with calculated output value (CPV1) instead of calculated output value (CPV) is in calibration state when the data status of calculated output value (CPV) is set to calibration (CAL). In this state, the calculated output value (CPV1) can be set manually and the output of the values greater than CPV2 stops.

IMPORTANT

When the output destination is cascade open or output fail in the function block in which the output value tracking is set to “Yes,” the tracking function precedes even if the data status of calculated output value (CPV) is in calibration (CAL).

● Calibration for Inter-Terminal Connected Calculation Blocks

Among various Calculation Blocks, the Analog Calculation Blocks listed below may perform calibration only as input processing in a case where data is entered as the calculated input value (RV) from other function blocks using terminal connection to the input terminal (IN).

- Square Root Block (SQRT)
- Exponential Block (EXP)
- First-Order Lag Block (LAG)
- Integration Block (INTEG)
- Derivative Block (LD)
- Lead/Lag Block (LDLAG)
- Dead-Time Block (DLAY)
- Dead-Time Compensation Block (DLAY-C)
- Variable Line-Segment Function Block (FUNC-VAR)

3.6 Input Processing in the Unsteady State

In the unsteady state, the function block may execute a different input processing from that in the normal state. This section explains input processing in the unsteady state. Input processing in the unsteady state differs between the Regulatory Control Blocks and Calculation Blocks.

■ Unsteady State of the Regulatory Control Blocks

The following section describes the unsteady states in the Regulatory Control Blocks and the special input processing.

● Unsteady States

The Regulatory Control Blocks suffer the following unsteady states:

- Input signal error (PV BAD)
  The data status of process variable (PV) is invalid.
- Calibration (PV CAL)
  The data status of process variable (PV) is calibration (CAL).
- Input connection open
  The input connection destination is the selector switch which is in open state.

● Executing Special Input Processing

The special input processing is also executed in the states below, although they are not unsteady states.

- Terminal connection
  The input terminal (IN) is connected via terminal connection with an output terminal of the other function block. This type of connection is used for the cascade loops with blocks such as Ratio Set Block (RATIO).
- Input connection undefined
  The input connection is not defined. The loop is in unconnected state.

SEE ALSO For more information about the input processing of Regulatory Control Blocks in the unsteady state, refer to:

3.6.1, "Input Processing of the Regulatory Control Block in Unsteady State" on page 3-27

■ Unsteady State of the Calculation Blocks

The Calculation Blocks suffer the following unsteady states:

- Input signal error
  The data status of input signal is invalid (BAD).
- Calibration (CAL)
  The data status of calculated output value (CPV) is calibration (CAL).
- The abnormal calculated input value is detected
  The data status of calculated input value (RV) is invalid (BAD).

SEE ALSO For more information about the input processing of calculation block in the unsteady state, refer to:

3.6.2, "Input Processing of the Calculation Block in Unsteady State" on page 3-29

3.6.1 Input Processing of the Regulatory Control Block in Unsteady State

This section explains the input processing of Regulatory Control Blocks in unsteady and special state.

■ Input Processing at Input Signal Error (PV BAD)

The Regulatory Control Blocks executes the following operations when input signal error (PV BAD).

- If the data reference is available, the input signal is read to update the data status without updating the data value.
- Input signal conversion is halted. For pulse input conversion, the contents of pulse input buffer is initialized when the processing is restarted.
- Integration is halted and the integrator value is held. The integration is continued from the held value when the processing is restarted.
- The digital filtering is halted. The previous value is initialized when the processing is restarted.
- The PV/FV overshoot operates when the PV/FV overshoot is specified.

■ Input Processing at Calibration (PV CAL)

The Regulatory Control Blocks executes the following operations at calibration (PV CAL).

- If the data reference is available, the input signal is read to update the data status without updating the data value. However, the data status of the process variable (PV) is calibration (CAL).
- The input signal conversion is halted. For the pulse input conversion, the contents of the pulse input buffer is initialized when the processing is restarted.
- The integration is continued with process variables (PV).
- The digital filtering is halted. The previous value is initialized when the process is restarted from the halt status.

■ Input Processing at Open Input Connection

The Regulatory Control Blocks executes the following operations at open input connection.

- The input signal conversion is halted. For the pulse input conversion, the contents of the pulse input buffer is initialized when the processing is restarted.
- The integration is halted and the integrator value is held. The integration is continued from the hold value when the processing is restarted.
- The digital filter processing is halted. The previous value is initialized when the processing is restarted.

■ Input Processing at Terminal Connection

The Regulatory Control Blocks executes the following operations at terminal connection.

- The input signal conversion is halted.
- Process variables (PV) are integrated.
- The digital filtering is halted.

■ Input Processing at Input Connection Undefined

The Regulatory Control Blocks executes the following operations at input connection undefined.

- The input signal conversion is halted.
- The integration is halted and the integrator value is held.
- The digital filtering is halted.

3.6.2 Input Processing of the Calculation Block in Unsteady State

This section explains the Input processing of the Calculation Blocks in the unsteady state.

■ Input Processing at Input Signal Error

The Calculation Blocks executes the following operations when input signal error.

- When the data status of the primary input from the input terminal (IN) is invalid (BAD), the calculated input value (RV) update, digital filter and integration processings are halted. The previous calculated input value (RV) is held and the data status of calculated input value (RV) becomes invalid (BAD). The digital filter initializes the previous value when the processing is restarted from the halt status. While the integration is halted, the integrator value is held, and the integration is continued from the hold value when the processing is restarted.
- When the data status of the secondary input from the input terminal (Qn) is invalid (BAD), the previous calculated input value (RVn) is held and the data status of the calculated input value (RVn) becomes invalid (BAD).
- When the CPV overshoot is specified, the calculated output value (CPV) is overshoot if the data status of the primary input becomes invalid (BAD).
- Input open data status signals (IOP, IOP-) are not detected for the terminals that are connected in sequence connection. If the input signal could not be obtained, the condition will be tested using the previous input value.

■ Input Processing at Calibration

The Calculation Blocks executes the following operations at calibration (CAL).

- The input signal conversion is executed, and the calculated input value (RV, RVn) and the data status continue to be updated.
- The digital filer is halted. The previous value is initialized when the processing is restarted from the halt status.
- The integration is continued. However, the calculated output value (CPV) is integrated at calibration (CAL).
- The CPV overshoot does not operate at calibration (CAL).

■ Input Processing at Calculated Input Value Error Detection

▼ Calculated Input Value Error Detected

The Calculation Blocks executes the following operations if a calculated input value error were detected.

- When the data status of calculated input value (RV) of the primary input from the input terminal (IN) is invalid (BAD), the calculation is not executed, the data status of the calculated output value (CPV) becomes invalid (BAD) and the previous calculated output value is held.
- When the data status of the calculated input value (RVn) of the secondary input is invalid (BAD), the calculation processing is continued using the previous calculated input value (RVn) held, and the calculated output value (CPV) is updated. However, the data status of the calculated output value (CPV) becomes “questionable” (QST).
- The Arithmetic Calculation Blocks, the General-Purpose Calculation Blocks and the Logic Operation Blocks used for auxiliary inputs perform the input error detection by themselves.

The table below shows the correlation among the data statuses of the calculated input value (RV) of the primary input and the calculated input value (RVn) of the secondary input, and the calculated output value (CPV).

Table 3.6.2-1 Correlation of the Data Statuses RV, RVn and CPV

Calculated input value (RV) of the primary input(*1)	Calculated input value (RVn) of the secondary input(*1)	Calculated output value (CPV) (*1)
BAD	-	BAD
NR	BAD	QST
NR	NR	NR

*1: BAD: Data value BAD
NR: Neither BAD nor QST
QST: Questionable
-: Ignore (don't care)

Input Processing at Calculated Input Value Error Detection in the Arithmetic Calculation

For Arithmetic Calculation Blocks other than the AVE block, the conditions for detecting an error in calculated input values and the data status setting of the calculated output value when an error is detected are defined in the "Calculated input value error detected" of the Function Block Detail Builder.

The method to transfer the data status (IOP, IOP-, OOP, NRDY) of the process I/O relations, which is generated with the calculated input value (RV, RV1) in connection with the above settings, to the calculated output value is specified. The table below describes the specified ranges 0 to 6. The default value is "1."

Table 3.6.2-2 Specification for Calculated Input Value Error Detected in the Arithmetic-Calculation Blocks Except AVE block

Calculated input value error detection specification	Error detection conditions (Data statuses of the calculated input values below are BAD.)(*1)	CPV data status (*1)	Data status transmission origin input value
0	-	-	No transmission
1	RV	BAD	RV
	RV1	QST	
2	RV1	BAD	RV1
	RV	QST	
3	RV and RV1	BAD	RV priority
	RV	QST	No transmission
4	RV and RV1	BAD	RV priority
	RV1	QST	No transmission
5	RV and RV1	BAD	RV priority
	RV or RV1	QST	No transmission
6	RV or RV1	BAD	RV priority

*1: -: Ignore (Don't care)

When the calculated input value error which causes the invalid (BAD) data status of calculated output value (CPV) occurs, the calculation processing is halted, and the previous calculated output value (CPV) is held.

When the calculated input value error which causes the questionable (QST) data status of calculated output value (CPV) occurs, the previous calculated input value is held due to the current calculated input value error. The calculation processing is continued using the previous value (RV) held and the calculated output value (CPV) is updated.

If CPV overshoot is being used, when the data status of the calculated output value is an invalid data value (BAD) because the data status of the calculated input value (RV) of the primary input is an invalid value (BAD), the calculated output value (CPV) overshoot.

● Input Processing at Calculated Input Value Error Detection in the Averaging Block (AVE)

AVE block behaves differently from other arithmetic calculation blocks when input error is detected.

SEE ALSO For more information about the behavior of AVE block when input error is detected, refer to:

"■ Input Processing at Calculated Input Value Error Detection" in 1.7, "Averaging Block (AVE)" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)

● Input Processing at Calculated Input Value Error Detection in the Auxiliary Inputs (RVn)-Used Logic Operation Blocks

For Logic Operation Blocks(*1) using auxiliary inputs (RVn), the conditions for detecting an error in calculated input values and the data status setting of the calculated output value when an error is detected are defined in the "Calculated input value error detected" of the Function Block Detail Builder.

*1: The Logic Operation Blocks can be used in FCSs except PFCS.

The method to transfer the data status (IOP, IOP-, OOP, NRDY) of the process I/O relations, which is generated with the calculated input value (RV1, RV2) in connection with the above settings, to the calculated output value is specified. The table below describes the specified ranges 0 to 6. The default value is "1."

Table 3.6.2-3 Specification for Calculated Input Value Error Detected in the Arithmetic-Calculation

Calculated input value error detection specification	Error detection conditions (Data statuses of the calculated input values below are BAD.) (*1)	CPV data status (*1)	Data status transmission origin input value
0	-	-	No transmission
1	RV1	BAD	RV1
	RV2	QST	
2	RV2	BAD	RV2
	RV1	QST	
3	RV1 and RV2	BAD	RV1 priority
	RV1	QST	No transmission
4	RV1 and RV2	BAD	RV1 priority
	RV2	QST	No transmission
5	RV1 and RV2	BAD	RV1 priority
	RV1 or RV2	QST	No transmission
6	RV1 or RV2	BAD	RV1 priority

*1: -: Ignore (Don't care)

When the calculated input value error which causes the invalid (BAD) data status of calculated output value (CPV) occurs, the calculation processing is halted, and the previous calculated output value (CPV) is held.

When the calculated input value error which causes the questionable (QST) data status of calculated output value (CPV) occurs, the previous calculated input value is held due to the current calculated input value error. The calculation processing is continued using the previous value (RV) held and the calculated output value (CPV) is updated.

● Input Processing at Calculated Input Value Error Detection in the General-Purpose Calculation Blocks (CALCU, CALCU-C)

CALCU and CALCU-C blocks behave differently from other arithmetic calculation blocks when input error is detected.

SEE ALSO For more information about the behavior of CALCU and CALCU-C blocks when input error is detected, refer to:

"■ Input Processing when a Calculation Input Value Error is Detected" in 1.33, "General-Purpose Calculation Blocks (CALCU, CALCU-C)" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)

● Input Processing for Data Status QST: FFCS Series/KFCS2/LFCS2

▼ Pass calculation block's input QST status to CPV

For a calculation block input, if its data status is QST, the QST status can be passed to the calculated output value. For passing the QST status, check the option of [Pass calculation block's input QST status to CPV] on [Constant 3] tab of FCS properties sheet.

When the option is checked or unchecked, the behaviors of CPV data status vary as follows:

- Relations between the status of calculated output value (CPV) and the statuses of the primary input (RV) and the secondary input (RVn)

Table 3.6.2-4 Relationship of Function Block's Data Status of Input (RV, RVn) and Data Status of CPV

Calculated input value (RV) of the primary input(*1) (*2)	Calculated input value (RVn) of the secondary input (*1) (*2)	CPV data status	
		QST is passed	QST is not passed
BAD	-	BAD	BAD
-	BAD	QST	QST
QST	QST	QST	NR
QST	NR	QST	NR
NR	QST	NR	NR
NR	NR	NR	NR

*1: -:Ignore (don't care)
*2: Conditions are tested starting from the top, and if a condition is met, the subsequent conditions are ignored.

- Relations between the status of calculated output value (CPV) and the statuses of calculated input value used in the calculation blocks except AVE block.

Table 3.6.2-5 Relations between the Status of Calculated Output Value (CPV) and the Statuses of Calculated Input Value used in the Calculation Blocks Except AVE block

Calculated input value error detection specification	Error detection conditions (Data statuses of the calculated input values below are BAD.)(*1) (*2)	Error detection conditions (Data statuses of the calculated input values below are QST.)(*1) (*2)	CPV data status (*1)	
			QST is passed	QST is not passed
0	-	-	-	-

Calculated input value error detection specification	Error detection conditions (Data statuses of the calculated input values below are BAD.)(*1) (*2)	Error detection conditions (Data statuses of the calculated input values below are QST.)(*1) (*2)	CPV data status (*1)	
			QST is passed	QST is not passed
1	RV	-	BAD	BAD
	RV1	-	QST	QST
	-	RV	QST	NR
	-	RV1	NR	NR
2	RV1	-	BAD	BAD
	RV	-	QST	QST
	-	RV1	QST	NR
	-	RV	NR	NR
3	RV and RV1	-	BAD	BAD
	RV	-	QST	QST
	RV1	RV	QST	NR
4	RV and RV1	-	BAD	BAD
	RV1	-	QST	QST
	RV	RV1	QST	NR
5	RV and RV1	-	BAD	BAD
	RV or RV1	-	QST	QST
	-	RV and RV1	QST	NR
6	RV or RV1	-	BAD	BAD
	-	RV or RV1	QST	NR

*1: -:Ignore (don't care)
*2: Conditions are tested starting from the top, and if a condition is met, the subsequent conditions are ignored.

- Relations between the status of calculated output value (CPV) and the statuses of the secondary inputs (RV1, RV2) in the calculation block.

Table 3.6.2-6 Relations between the Status of Calculated Output Value (CPV) and the Statuses of the Secondary Inputs (RV1, RV2) in the Calculation block

Calculated input value error detection specification	Error detection conditions (Data statuses of the calculated input values below are BAD.) (*1) (*2)	Error detection conditions (Data statuses of the calculated input values below are QST.) (*1) (*2)	CPV data status(*1) (*3)	
			QST is passed	QST is not passed
0	-	-	-	-
1	RV1	-	BAD	BAD
	RV2	-	QST	QST
	-	RV1	QST	NR
	-	RV2	NR	NR
2	RV2	-	BAD	BAD
	RV1	-	QST	QST
	-	RV2	QST	NR
	-	RV1	NR	NR

Calculated input value error detection specification	Error detection conditions (Data statuses of the calculated input values below are BAD.) (*1) (*2)	Error detection conditions (Data statuses of the calculated input values below are QST.) (*1) (*2)	CPV data status(*1) (*3)	
			QST is passed	QST is not passed
3	RV1 and RV2	-	BAD	BAD
	RV1	-	QST	QST
	RV2	RV1	QST	NR
4	RV1 and RV2	-	BAD	BAD
	RV2	-	QST	QST
	RV1	RV2	QST	NR
5	RV1 and RV2	-	BAD	BAD
	RV1 or RV2	-	QST	QST
	-	RV1 and RV2	QST	NR
6	RV1 or RV2	-	BAD	BAD
	-	RV1 or RV2	QST	NR

*1: -:Ignore (don't care)
*2: Conditions are tested starting from the top, and if a condition is met, the subsequent conditions are ignored.
*3: CPV here denotes the CPV1 and CPV2 for SRS2-R / SRS2-S function block. For other function blocks, it denotes the CPV1.

TIP

Notice that when this option is checked, the closing processing of trend data and the closing Data of report, the quality flag of OPC may be affected.

SEE ALSO

For more information about the action of AVE block when the option of [Pass calculation block's input QST status to CPV] is checked, refer to:

"● Input Processing for Data Status QST: FFCS Series/KFCS2/LFCS2" in "■ Input Processing at Calculated Input Value Error Detection" in 1.7, "Averaging Block (AVE)" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)

For more information about the action of CALCU and CALCU-C Block's when the option of [Pass calculation block's input QST status to CPV] is checked, refer to:

"● Input Processing for Data Status QST: FFCS Series/KFCS2/LFCS2" in "■ Input Processing when a Calculation Input Value Error is Detected" in 1.33, "General-Purpose Calculation Blocks (CALCU, CALCU-C)" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)

For more information about the closing processing of trend data, refer to:

"■ Closing Data Processing Based on Data Status" in 1.7.2, "Structure of the Closing Processed Data" in the Human Interface Stations Reference Vol.2 (IM 33K03F22-50E)

For more information about the closing Data of report, refer to:

"■ Detailed Format Design - Closed-Topic Dialog Box Detail Definition Sheet" in 2.4.2, "Detailed Format Design - Closing Data" in the Optional Functions Reference (IM 33K03N10-50E)

For more information about the quality flag of OPC, refer to:

"● Setting Quality Flag Tables" in "■ HIS Setup Window" in 1.2, "Engineering Related to OPC" in the Optional Functions Reference (IM 33K03N10-50E)

3.7 Input Processing for Sequence Connection

As a special input processing when using the sequence connection, there is “condition testing” for the input signals. When using the sequence connection, the “Calibration” function partially differs.

■ Input Processing for Sequence Connection

For the General-Purpose Calculation Blocks and Logic Operation Blocks (*1), the sequence connection can be used as the I/O connection method.

When using the sequence connection for the Logic Operation Blocks (*1) and for CALCU and CALCU-C blocks, there are two types of special input processing as follows:

- “Condition testing” for input signals
- “Calibration”

*1: Logic Operation Blocks can be used in FCSs except PFCS.

■ Input Processing Block Chart for the Sequence Connection

The input processing block chart for the sequence connection is shown below.

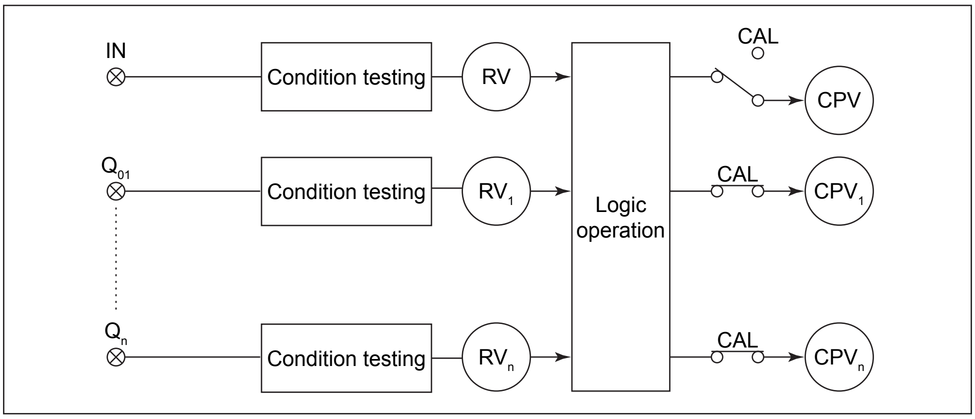


Figure 3.7-1 Block Chart of Input Processing for Sequence Connection

■ Condition Testing

For sequence connection, the input terminals (IN, Qn) of the Logic Operation Blocks(*1) applicable for sequence and CALCU, CALCU-C blocks store the following I/O connection information:

- Information to identify the connection destination, such as tag name, user defined label name, terminal number, and element number
- Information to identify the data item
- Information to indicate the condition specification

*1: Logic Operation Block can be used in FCSs except PFCS.

■ Condition Specification on the I/O Connection Information

Using the I/O connection information and the input signal read from the input terminals, the condition testing may be performed. When the input signals satisfy the conditions, 1 is set to the calculated input value (RV). When the conditions are not satisfied, 0 is set to the calculated input value (RV).

SEE ALSO For more information about the condition specification on the I/O connection information, refer to:

- 2.3.7, "Condition Signal Description : Referencing Other Function Blocks and I/O Data" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)
- 2.3.8, "Syntax for Condition Signal Description : Referencing Logic Chart" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)
- 2.3.9, "Syntax for Condition Signal Description : Referencing Sequence Table in a Logic Chart" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)

■ Input Processing for Calibration

Condition testing is performed to continue updating the calculated input values (RV, RVn).

4. Output Processing

This chapter explains the output processing common to Regulatory Control Blocks and Calculation Blocks.

■ Output Processing

Output processing is a general term, representing that all function blocks, execute certain process to the values obtained from the control computation before output it. There are various forms of output processing corresponding to the function block type and the output signal format. Some forms of output processing are common to Regulatory Control Blocks and Calculation Blocks, while others are specific to certain particular blocks.

The output processing for the sequence control block is unique and differs from that of the Regulatory Control Blocks or Calculation Blocks.

SEE ALSO

For more information about sequence table block output processing details, refer to:

2.2.7, "Output Processing of Sequence Table" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)

For more information about the output processing of the blocks connected in sequence connection, refer to:

2.3.6, "Output Processing of Logic Chart" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)

■ Output Processing Common to Regulatory Control Block

In a Regulatory Control Block, the value obtained from control computation undergoes the output processing, then outputs as the manipulated output variable (MV), as depicted in the figure below.

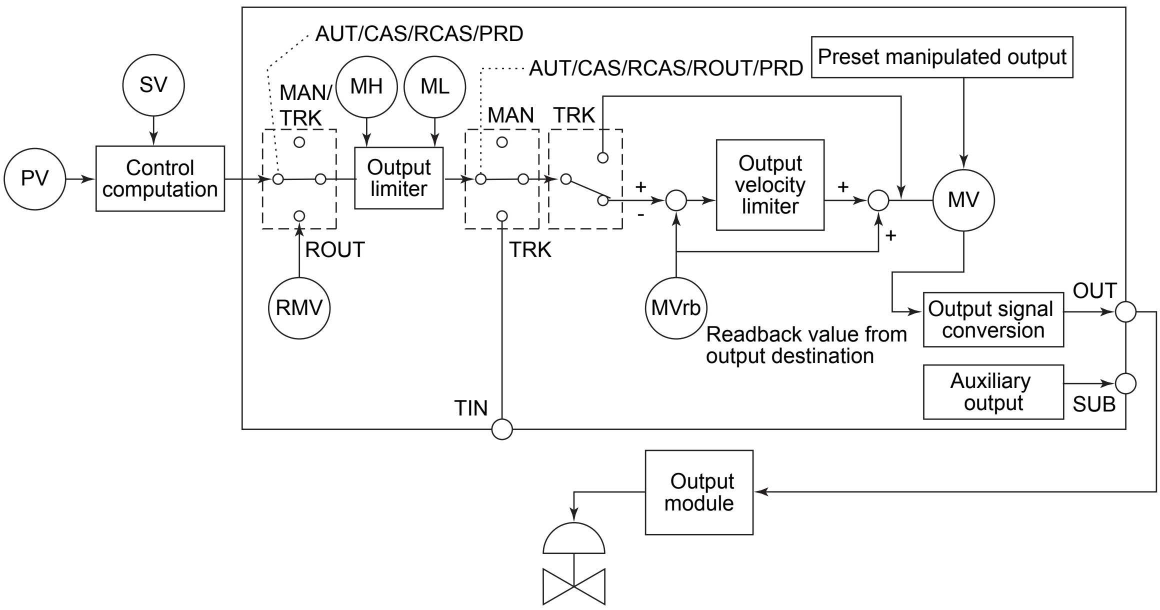


Figure 4-1 Block Chart of Output Processing Common to Regulatory Control Block

TIP

In the Dual-Redundant Signal Selector Block (SS-DUAL) and Signal Selector Blocks (SS-H/M/L), the value obtained by signal selection is output as a selected signal value (PV).

● Output Operation

▼ Control Calculation Output Type

A Regulatory Control Block outputs its manipulated output value (MV) or the vicissitude of that value (ΔMV).

There are two types output action: positional and velocity:

- In positional output action, the output value connects to its destinations unchanged.
- In velocity output action, the amount of change for the current output (ΔMV) is added to the value read back from the connection destination of the output terminal.

The output operation can be specified on the Function Block Detail Builder.

Output action: Selectable from “positional” or “velocity.”
Default is the “positional” action.

SEE ALSO For more information about setting output actions in Regulatory Control Blocks, refer to:

1, “Regulatory Control” in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)

● Output Limiter

It limits the manipulated output value (MV) to be within the high and low limit values.

● Output Velocity Limiter

It limits the amount of change between the current and previous output values to avoid output bumps.

● Output Clamp

It prevents the manipulated output value (MV) from being varied above or below the current output value. This state is called output clamp. In the output clamp state, the data status of the manipulated output value (MV) will be either the clamp high (CLP+) or clamp low (CLP-).

● Preset Manipulated Output

Upon an external command, the block is forced to operate in the manual mode and output it’s manipulated output value (MV) at a predetermined value.

● Output Tracking

It forces the output value to match the value of its output destination or the value of the tracking input signal.

● Output Range Tracking

It forces the scale high/low limits (MSH and MSL) of the manipulated output value (MV) to match the scale high/low limits of its output destination. When a change occurs in the scale high/low limits (MSH and MSL) of the manipulated output value (MV), it recalculates the values of the data related to the manipulated output value (MV).

● Manipulated Output Index

This function displays two indexes in the manipulated output value (MV) scale in the HIS operation monitoring screen of operation and monitoring function. They are called manipulated output indexes.

These indexes are set to indicate the feasible limits of the manipulated output values (MV). They can be used as manipulation reference when operation in the manual mode, and they can be used to verify the normal conditions when operation in the automatic mode. The manipulated output indexes are available only in the Regulatory Control Blocks.

● Output Signal Conversion

The function converts the result of calculation process into a signal that is compatible with the output destination, such as the output module or other function blocks.

Various types of output signal conversion are available for different types of function block and output signal. There are also types of output signal conversion that are common to the Regulatory Control Blocks, as well as those specific to individual function blocks.

● Auxiliary Output

The manipulated output value (MV), change in manipulated output (ΔMV), process variable (PV), or change in process variable (ΔPV) is output to final control elements such as compensation control equipment or external indicator of control stations.

● Output Processing in Unusual Cases

When in unusual cases, the Regulatory Control Blocks process the output different from in usual cases.

● Output Processing Specific to the Motor Control Blocks (MC-2, MC-2E, MC-3, and MC-3E)

The Motor Control Blocks (MC-2, MC-2E, MC-3, and MC-3E) execute a special output processing, which is different from other function blocks.

SEE ALSO For more information about output processing of MC-2, MC-2E, MC-3, and MC-3E, refer to:

1.17.3, “Output Processing of Motor Control Blocks (MC-2, MC-2E, MC-3 and MC-3E)” in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)

● Output Processing for the Blocks with Sequence Connection

The Pulse Count Input Block (PTC) may be connected in sequence connection, one of the connection types. A special output processing for sequence connection with other blocks is supported and referred as “status manipulation.”

● Output Processing during OOP

In general, when output open alarm (OOP) occurs, the output of manipulated variable will stop. However, the following actions can be performed by the output processing according to the settings.

- Function block tracks output module during OOP
- Function block writes to output module during OOP

SEE ALSO For more information about output open alarm, refer to:

5.7, “Output Open Alarm Check” on page 5-20

■ Output Processing Common to All Calculation Blocks

In a Calculation Block, the value obtained from calculation process undergoes the output processing, then outputs as the calculated output value (CPV), as depicted in the figure below.

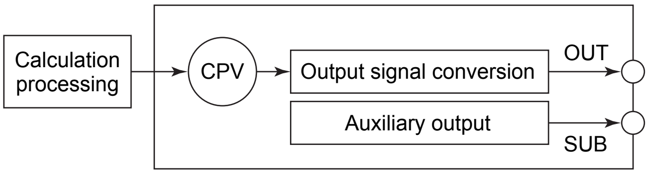


Figure 4-2 Block Diagram of Output Processing Common to the Numerical, Analog and General-Purpose Calculation Blocks

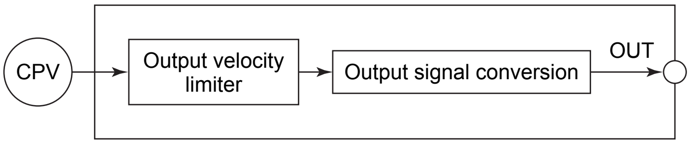


Figure 4-3 Block Diagram of Output Processing Common to Data Set Blocks

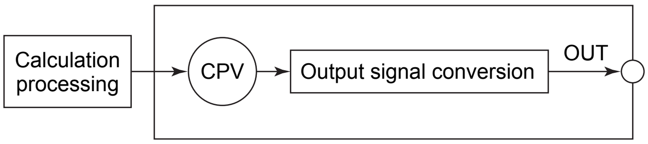


Figure 4-4 Block Diagram of Output Processing Common to the Logic Operation Blocks

TIP

Logic Operation Block can be used in FCSs except PFCS.

● Output Operation

▼ Control Calculation Output Type

A Calculation Block outputs the calculated output value (CPV) or the vicissitude of that value (ΔCPV).

There are two types output action: positional and velocity:

- In positional output action, the output value connects to its destinations unchanged.
- In velocity output action, the amount of change for the current output (ΔCPV) is added to the value read back from the connection destination of the output terminal.

The output operation can be specified on the Function Block Detail Builder.

The Calculation Blocks that can select positional output or velocity output action are Arithmetic Calculation Blocks and Analog Calculation Blocks. Other Calculation Blocks are fixed to the positional output action.

Output action: Selectable from “positional” or “velocity.”
Default is the “positional” action.

● Output Velocity Limiter

It limits the amount of change between the current and previous output values to avoid output bumps.

● Output Clamp

It prevents the manipulated output value (MV) from being varied above or below the current output value. This state is called output clamp. In the output clamp state, the data status of the manipulated output value (MV) will be either the clamp high (CLP+) or clamp low (CLP-).

● Output Tracking

It forces the output value to match the value of its output destination or the value of the tracking input signal.

● Output Signal Conversion

The function converts the result of calculation process into a signal that is compatible with the output destination, such as the output module or other function blocks.

Various types of output signal conversion are available for different types of function block and output signal. There are also types of output signal conversion that are common to the Calculation Blocks, as well as those specific to individual function blocks.

● Auxiliary Output

The calculated output value (CPV), or the change in calculated output (ΔCPV) is output to final control elements such as compensation control equipment and external indicator of control stations.

● Output Processing in Unusual Cases

When in unusual cases, the Calculation Blocks process the output different from in usual cases.

● CPV Pushback

In terminal connection, a function block obtains output value (CPV) by tracking to the downstream function block at the IN terminal, and calculates the calculation input value (RV) backward from the CPV to allow the upstream function block tracking.

● Output Processing for General-Purpose Calculation Blocks (CALCU and CALCU-C) in Sequence Connection

The General-Purpose Calculation Blocks (CALC, CALC-C) and Logic Operation Blocks (*1) may be connected in sequence connection, one of the connection types. A special output processing for sequence connection with other blocks is supported and referred as “status manipulation.”

*1: Logic Operation Block can be used in FCSs except PFCS.

■ Output Processing Applicable to Each Model of Blocks

The different types of output processing may be applied to different models of function blocks.

SEE ALSO

For more information about output processing applicable to each model of regularly blocks, refer to:

“■ Output Processing Possible for Each Regulatory Control Block” in 1.1.3, “Input Processing, Output Processing, and Alarm Processing Possible for Each Regularly Control Block” in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)

For more information about output processing applicable to each model of calculation blocks, refer to:

“■ Output Processing Possible in Each Calculation Block” in 1.3.1, “Input Processing, Output Processing, and Alarm Processing Possible for Each Calculation Block” in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)

4.1 Output Limiter

The output limiter limits the manipulated output value (MV) within the MV High/Low limits (MH and ML) when running in Auto mode. An enhanced output limiter function is provided as the extra function to the output limiter. This function may prevent the output from abrupt action when the manipulated output value (MV) is operated in the manual mode beyond the range of high or low limit, and then the operation mode is switched from manual to auto.

The output limiter is only available for Regulatory Control Blocks.

■ Output Limiter

The output limiter limits the manipulated output value (MV) within the high (MH) and low (ML) limit setpoints.

The output limiter is valid only when the regulatory blocks are running in automatic modes.

TIP

Automatic modes stand for the block modes of CAS, AUT and RCAS.

If all the following conditions are satisfied, automatic modes stand for the block modes of CAS, AUT, RCAS and PRD.

- The FCS is a FFCS series, a KFCS2 or LFCS2 station.
- Enable output limiter in PRD mode

But, it has no effect on 2-position ON/OFF type output, or 3-position ON/OFF type output and Pulse width type output with no feedback.

When the manipulated output value (MV) reaches the limit set by the output limiter, the high or low limit alarm is activated, and the data status of the manipulated output value (MV) will be in the clamp high (CLP+) or clamp low (CLP-) respectively.

The high (MH) and low (ML) limit setpoints are set in the following setting parameters.

- High limit for manipulated output value (MH)
  Data in an engineering unit and within the MV range.

  The default is the ceiling of the MV scale.
- Low limit for manipulated output value (MH)
  Data in an engineering unit and within the MV range.

  The default is the bottom of the MV scale.

TIP

- In the Control Signal Splitter Block (SPLIT), the MVn scale high limit is set in the manipulated output high limit setpoint (MH) and the MVn scale low limit is set in the manipulated output low limit setpoint (ML), for each output.
- In the 13-Zone Program Set Block (PG-L13), the output value always falls within the range between the MV scale high (MSH) and low (MSL) limits.

■ High/Low Limit Bumpfree Capability

The high/low limit bumpfree Capability temporarily expands the manipulated output high and low limit setpoints (MH, ML) in order to avoid such abrupt change in the manipulated output value (MV) caused by the output limiter.

When a Regulatory Control Block is in manual operation mode and a value exceeding the manipulated output high and low limit setpoints is set from the operating and monitoring function, a confirmation request message is displayed to caution the user. If the user performs an acknowledgment operation at this time, a value exceeding the manipulated output high and low limit setpoints can be set. The function block outputs the value set by the user regardless of the manipulated output high and low limit setpoints (MH, ML).

If the manipulated output value (MV) that is set by manual operation exceeds the range as determined by the manipulated output high and low limit setpoints, and if operation is changed to an automatic mode, the output limiter will force the manipulated output value (MV) to change to the manipulated output upper limit setpoint (MH) or the lower limit setpoint (ML) so that the manipulated output value (MV) undergoes an abrupt change.

TIP

Automatic modes stand for the block modes of CAS, AUT and RCAS.

If all the following conditions are satisfied, automatic modes stand for the block modes of CAS, AUT, RCAS and PRD.

- The FCS is a FFCS series, a KFCS2 or LFCS2 station.
- Enable output limiter in PRD mode
- The output is not a pulse-width output

IMPORTANT

In the case that all the following circumstances exist together, the high/low limit bumpfree capability does not function in PRD mode.

- The FCS is a FFCS series, a KFCS2 or LFCS2 station.
- Enable output limiter in PRD mode
- The output is a pulse-width output.

However, even in this case, the output limiter functions in PRD mode. Therefore, the following operations may cause the manipulated variable (MV) to abruptly changes and sends the UP and DOWN outputs. You need to be careful about this phenomenon.

- Changing the block mode from MAN to PRD while the manipulated variable is set to a value beyond the range of the output limits (MH, ML) in MAN mode.
- Setting the high output limit (MH) to a value smaller than the current manipulated variable in PRD mode.
- Setting the low output limit (ML) to a value greater than the current manipulated variable in PRD mode.

SEE ALSO

For more information about the output limiter in PRD mode, refer to:

"● Output Limiter in PRD Mode: FFCS Series/KFCS2/LFCS2" in "■ Primary Direct (PRD) Mode Action" in 1.4 "Control Computation Processing Common to Controller Blocks" in Function Blocks Reference Vol.1 (IM33K03E22-50E)

● When MV is Set in the Manual Mode

The High/Low-limit Expansion function is activated when the manipulated output value (MV) is manually set over the high-limit (MH) or under the low-limit (ML) in manual mode.

- If the MV exceeds the MH setpoint, the value equal to that MV will be the temporarily extended high limit setpoint (MHe) for the manipulated output.
- If the MV falls below the ML setpoint, the value equal to that MV will be the temporarily extended low limit setpoint (MLe) for the manipulated output.

● When the Mode is Switched from Manual to Automatic

When the mode is changed from manual operation to automatic operation, the output limiter operates using the manipulated output high limit setpoint (MHe) or the manipulated output low limit setpoint (MLe) that has been temporarily expanded. Therefore, there will be no abrupt change in the manipulated output value (MV).

- If the calculation process yields a value exceeding the current MHe value, then the current MHe value will be output as the MV. Otherwise, the calculated result will be output, and the MHe value will be replaced by the new MV value.
- If the calculation yields a value that falls below the current MLe value, then the current MLe value will be output as the MV. Otherwise, the calculated result will be output, and the MLe value will be replaced by the new MV value.

The following figure shows the operation of the High/Low-limit Expansion function:

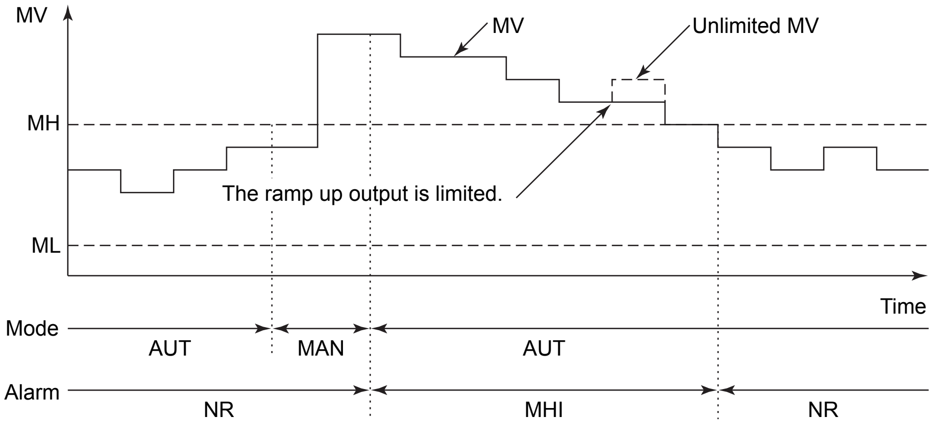


Figure 4.1-1 Operation of High/Low-Limit Expansion Function

Return to Normal Operation

When the manipulated output value (MV) finally returns within the high limit (MH) and low limit (ML) setpoints, the output limiter returns to normal operation.

Disable High/Low Limit Bumpfree Capability

▼ **High/Low Limit bumpfree capability**

High/Low Limit Bumpfree Capability is disabled.

When a block is running in automatic mode and the output high/low limit (MH/ML) is changed from the operation and monitoring function, the manipulated output value (MV) will be forced to the range within the new limit (MH/ML) at the next scan. And when the block is running manual mode (MAN), if the manipulated output (MV) is changed to a value beyond the limit range, a confirmation request message is displayed to caution the user. The manipulated output (MV) can go beyond the limit only after the confirmation is performed. If the block is changed to automatic mode at this moment, the manipulated output value (MV) will be forced to the range within the limit (MH/ML) at the next scan.

TIP

Automatic modes stand for the block modes of CAS, AUT and RCAS.

If all the following conditions are satisfied, automatic modes stand for the block modes of CAS, AUT, RCAS and PRD.

- The FCS is a FFCS series, a KFCS2 or LFCS2 station.
- Enable output limiter in PRD mode

High/Low Limit Bumpfree Capability can be enabled or disabled on the Function Block Detail builder.

High/Low Limit Bumpfree Capability: Choose [Valid] or [Invalid]
The default is [Valid].

The High/Low Limit Bumpfree Capability can be enabled or disabled on the following function blocks:

PID, PI-HLD, PID-BSW, PID-TP, PD-MR, PI-BLEND, PID-STC, MLD-SW, RATIO, VELLIM, AS-H/M/L, FFSUM, XCPL

4.2 Output Velocity Limiter

It is a function to limit the amount of change between the previous and current output values, so as to prevent abrupt changes in the output value.

■ Output Velocity Limiter

▼ Output Change, Output Velocity Limiter, MAN Mode Output Velocity Limiter Bypass

The output velocity limiter limits the amount of change in the output value according to the output velocity limit setting. The output velocity limit is a permissible amount of output change over one scan period.

In the manual operation mode, the operation and monitoring window will display the manipulated output value (MV) as set manually, even if the output velocity limiter has acted to limit the manipulated output value (MV). The output velocity limiter can be disabled by setting the bypass for the MAN-mode output velocity limiter.

Also, the limiter will not function when the output signal is 2-position ON/OFF or 3-position ON/OFF output, or when the block is in the tracking (TRK) mode.

In the PID Controller Block with Batch Switch (PID-BSW), the output velocity limiter does not operate during the time the manipulated output value (MV) is at the manipulated output high limit value or low limit value because the control deviation value has exceeded the deviation alarm setpoint and lockup setpoint.

The output velocity limiter does not function in Time-Proportioning ON/OFF Controller Block regardless the setting for MAN Mode Output Velocity Limiter Bypass.

The output velocity limiter and the bypass for MAN-mode output velocity limiter may be defined on the Function Block Detail Builder.

- Output Velocity Limiter
  Engineering unit data or percentage within the range from 0 to the MV scale span setting in positive values only (six significant figures).

  The default setting is 100.0 %.

- MAN Mode Output Velocity Limiter Bypass
  Selectable between “Yes” and “No.”

  Default is “No.”

4.3 Output Clamp

Output clamp is a function applied to cascade-connected output terminals. It prevents the manipulated output value (MV) from exceeding, or falling below the current value, when the data status of manipulated output value (MV) is in the high-limit (CLP+), or low-limit (CLP-) clamped state.

■ Output Clamp

The output clamp function indicates the status that the manipulated output value (MV) is restrained at the limits of a specified range.

The output clamp function operates only when the output terminal is connected in cascade. The data status CLP+ or CLP- is initiated by one of the following conditions:

- When the output value is limited by the output limiter.
- When the data status of cascade-connected destination is CLP+ or CLP-.

Each of these conditions is explained further in the following paragraph.

● When the Output Value is Limited by the Output Limiter

If the output is limited within the range of high-limit (MH) and low-limit (ML) setpoints of the manipulated output value, then the data status of the function block will be CLP+ or CLP-.

The CLP+ and CLP- have a 2 % (initial value) hysteresis. For example, if the low-limit setpoint is 0 %, and the manipulated output falls to 0 % and activates the output limiter. Then the data status of the manipulated output value (MV) become CLP-. Later, when the manipulated output increases from 0 % and the output limiter is no longer activated, the data status CLP- will continue until the manipulated output value (MV) exceeds 2 %.

● When the Data Status of Cascade-connected Destination Become CLP+ or CLP-

If the output terminal is cascade-connected to a function block whose cascade set value (CSV) is in the status of CLP+ or CLP-, or if the data status of data item connected to the IN terminal of the connection destination at the terminal connection is CLP+ or CLP-, then the data status of manipulated output value at the connection source will also be CLP+ or CLP-.

■ Operation of Output Clamp

▼ Limit Output in Direction when Clamped

When the data status of the cascade connection destination is CLP+ or CLP-, the output direction of the manipulated output value (MV) is restricted, i.e., the value cannot be changed to exceed or falls below the present output value, so that only the manipulated output value (MV) in the direction that cancels CLP+ or CLP- is output.

The restriction of the direction of changes in clamped output can be set with the Function Block Detail Builder.

Limit output in direction when clamped: Selectable from “Yes” and “No.”
Default is “Yes.”

■ Release the Output Clamp

The output clamp will be released under the following conditions:

- When the block mode is switched to the manual (MAN), remote output (ROUT), tracking (TRK) or primary direct mode (PRD).
- When the output connection destination is changed through a switch, etc.

■ Output Clamp and MAN Mode of Primary Loop

In a cascade control loop, if the setting of [Limit Output in Direction when Clamped] of the primary loop is specified to [Yes], the output (MV) of the primary loop in manual (MAN) mode and the set point value of the secondary loop will behave as follows if the output of the secondary loop is clamped.

- The manipulated output value (MV) of the primary loop can be freely manipulated irrelevant to clamping or declamping direction.
- The set point value (SV) of the secondary loop will follow the motion of the primary loop's output (MV) if the motion is in the declamping direction, and will not follow the MV of the primary loop but maintain the current value if the motion is in the clamping direction.

■ Tracking of the Output Clamp Status

The data status CLP+ or CLP- of the manipulated output value (MV) will be copied to the data status of the setpoint (SV, CSV and RSV) and of the remote manipulated output value (RMV). This action is called the tracking of the output clamp status. The tracking function transmits the output clamp status of a downstream function block to an upstream function block.

If, however, the "limit output in direction when clamped" is disabled in the upstream block, the downstream output status will not be transmitted to the upstream.

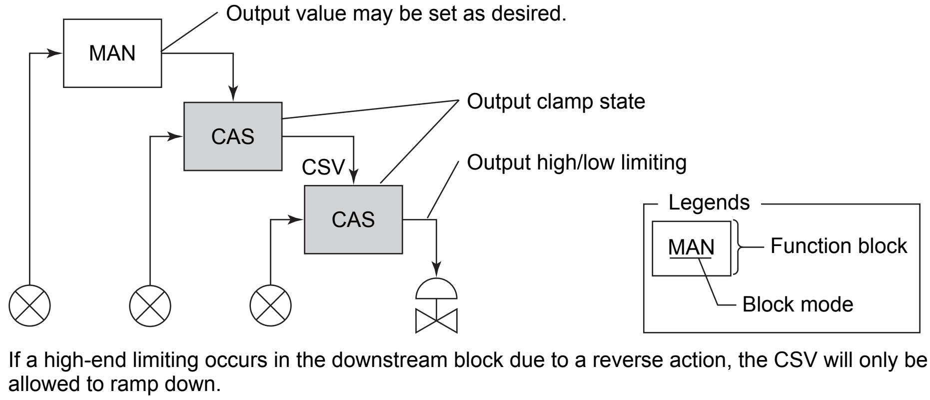


Figure 4.3-1 An Example of Typical Clamping

■ Output Clamp when the High/Low Limit Bumpfree Capability is in Effect

Note When the output High/Low-limit Bumpfree Capability is in effect, the manipulated output value (MV) is not limited by the high-limit (MH) or the low-limit (ML) setpoints.

Whenever MV≥MH or MV≤ML, the manipulated output value (MV) data status is set as CLP+ or CLP- even if the manipulated output value (MV) is not limited by the high/low limit bumpfree capability .

■ An Example of Output Clamp

The following figures illustrate examples of output clamp:

● An Example of Function Blocks Connected in Parallel in Downstream

Even in a case in which multiple function blocks are connected downstream via the Cascade Signal Distributor Block (FOUT), the output clamp status of the downstream function block output is transmitted to the upstream function block via the FOUT block.

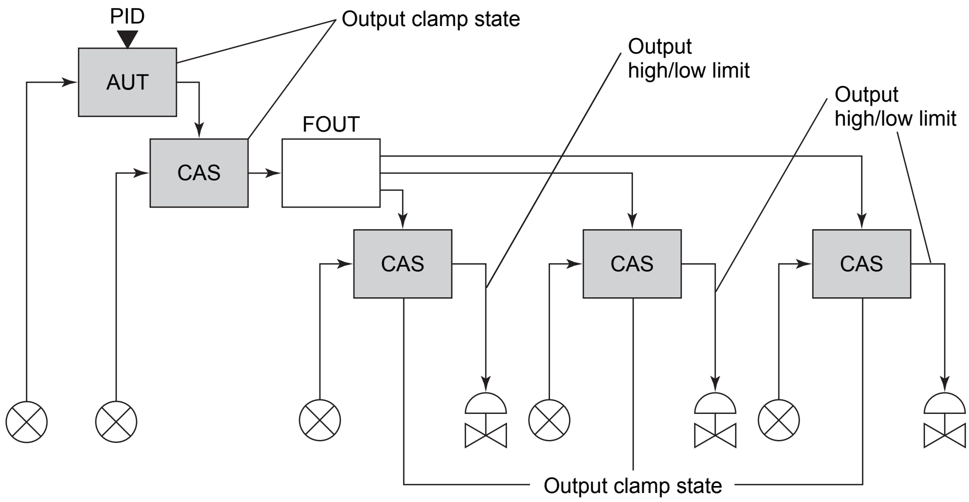


Figure 4.3-2 When Downstream Function Blocks are Connected in Parallel

● When Connecting to IN Terminal of a Ratio Set Block (RATIO)

When RATIO block is connected downstream in a loop, the output clamp status of the RATIO block is transmitted to the upstream function block connected to its SET terminal. The clamp status is not transmitted via IN terminal even though the connection to the IN terminal is the terminal connection type.

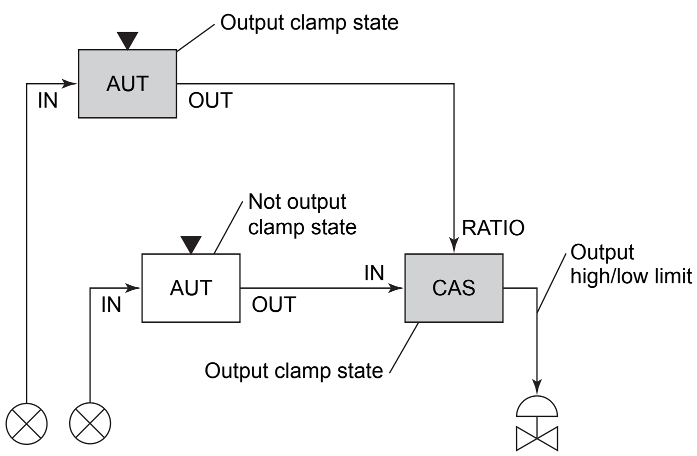


Figure 4.3-3 An Example of Connecting to the Input Terminal of a Ratio Set Block (RATIO)

● When Connecting to an Auto Selector Block (AS-H/M/L)

When an AS-H/M/L block is connected downstream and the AS-H/M/L block signal selector switch position is 4 (auto selection), the output clamp status of the AS-H/M/L block is transmitted to all upstream function blocks. When the signal selector switch position is 1, 2, or 3, the output clamp status is transmitted only to the upstream function block that is selected by the signal selector switch.

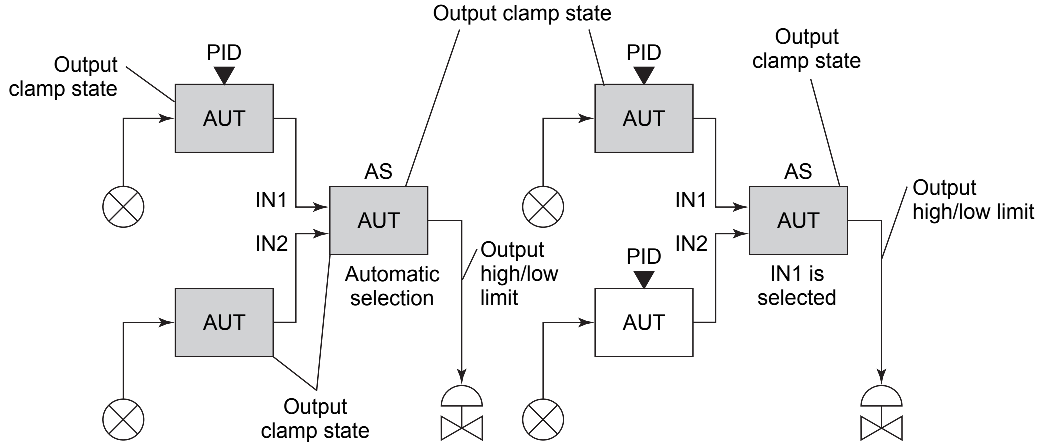


Figure 4.3-4 An Example of Connecting with Auto Selector Block (AS)

● When Connecting to a Switch Block (SW-33, SW-91)

When connected to a downstream function block via the SW-33 or SW-91 block, the output clamp status of the downstream function block is not transmitted upstream when the switch is OFF. It is transmitted when the switch is ON.

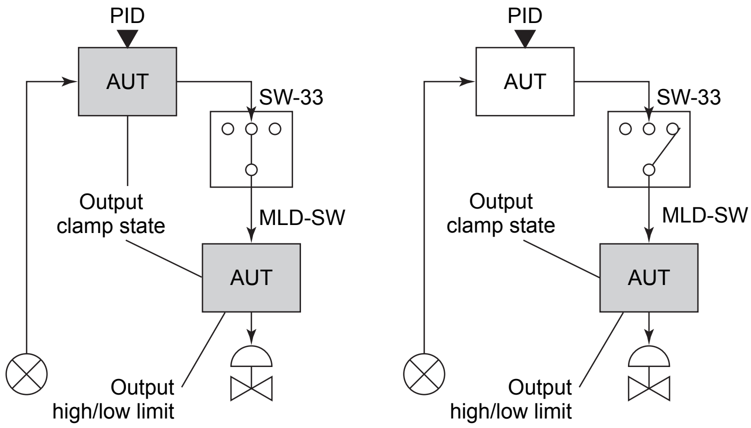


Figure 4.3-5 An Example of Connecting with a Switch Block (SW-33)

■ Precautions When Connecting a Manual Loader Block with Auto/Man SW (MLD-SW) in the Downstream of Cascade Loop

If a Manual Loader Block with Auto/Man SW (MLD-SW) is placed downstream in a cascade-connected control loop, the control block connected upstream must have the “limit output in direction when clamped” function turned off.

The following paragraph explains how to connect a MLD-SW block in the downstream of a cascade loop.

● An Example of Connecting a Manual Loader Block with Auto/Man SW (MLD-SW) in the Downstream of a Cascade loop

Assume the MLD-SW gain (GAIN) is 1, bias (BIAS) is 0, the low-limit alarm setpoint is 0 %, and hysteresis for manipulated output alarm is 2 %, in the loop shown below.

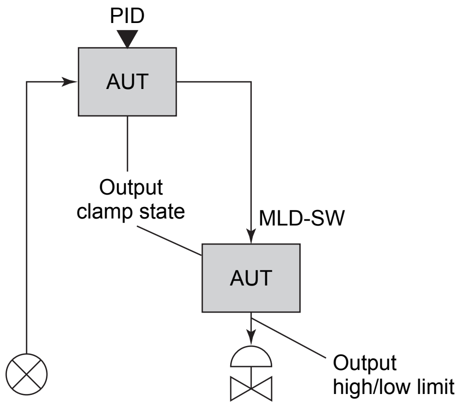


Figure 4.3-6 An Example of Turning Off the Limit Output in Direction when Clamped

● Action of the “Limit Output in Direction when Clamped”

When the block status of the PID and MLD-SW blocks are AUT, set the manipulated output value (MV) of a PID block to 0 % then the MLD-SW manipulated output (MV) become 0 %, clamped at the low limit, thus turns the data status of the manipulated output value (MV) to CLP-. This will also cause the PID block data status of the manipulated output value (MV) to CLP-, and restricts the PID manipulated output value (MV) changes on the clamped output direction. The restriction on the PID block will continue until a change in the PID output causes the manipulated output value (MV) of the MLD-SW to exceed 2 %, thus releasing the CLP-data status of the manipulated output value (MV).

● Precautions on “Limit Output in Direction when Clamped”

When changes on the direction of clamped output is restricted, the PID block output can not decrease to 0 % once it increased to 1.9 % under CPL- status. The MLD-SW is restricted on moving to the clamped direction kept it at 1.9 %. In other words, though a valve requires full close when the manipulated output value (MV) becomes 0 %, the valve is not fully closed in this case.

Turn off the “limit output in direction when clamped” at the upstream control block can avoid this state.

4.4 Preset Manipulated Output

The preset manipulated output is a function that, through an external command, forces a block to the manual mode (MAN) and output a preset value as the manipulated output value (MV).

The preset manipulated output function is available only for Regulatory Control Blocks.

■ Preset Manipulated Output

The preset manipulated output is a function that, through an external command, forces a block to the manual mode (MAN) and output a preset value as the manipulated output value (MV).

The command for the preset manipulated output is generated only upon switching the preset MV switch (PSW) from 0 to 1, 2 or 3. The value of the preset MV switch (PSW) will determine the manipulated output value (MV) as follows:

- PSW = 1: MV = MSL (the low limit of MV scale)
- PSW = 2: MV = MSH (the high limit of MV scale)
- PSW = 3: MV = PMV (the preset manipulated-output value)

The preset manipulated output value (PMV) is a value set as a tuning parameter from the operation and monitoring function, or from the General-Purpose Calculation Blocks.

■ Reset the Preset MV Switch

- The preset MV switch (PSW) value will be automatically reset to 0 when the preset manipulated output function is activated to set the manipulated output (MV) at a preset value. The block mode will remain manual (MAN), can not trace back the mode and value prior to the activation of preset manipulated output function.
- If the value of the preset MV switch (PSW) is set at 1 or 2, the output velocity limiter will not take effect on the preset manipulated output.
- If the value of the preset MV switch (PSW) is set at 3, and if MAN mode output velocity limiter bypass is set to off, the output velocity limiter will restrict the velocity when the MV tries to jump to the preset value.

IMPORTANT

- Since the preset MV switch (PSW) is automatically reset to 0, the PSW ≠ 0 state can not be referred by other function blocks. For example, the Sequence Table Blocks (ST16, ST16E) can not refer it as a condition for sequence control.
- If the preset MV output option is [Preset MV Valid by Preset Switch], PSW ≠ 0 cannot be grabbed by the sequence table.

■ Set Parameters for the Preset Manipulated Output Function

The set parameters for the preset manipulated output function is shown below:

Preset manipulated output value (PMV):	Data in an engineering unit and within the MV range. Default is the low limit of the MV scale.
Preset MV switch (PSW):	Selectable among “0,” “1,” “2,” “3.” Default is “0.”

■ Preset MV Valid Immediately : FFCS Series/KFCS2/LFCS2

▼ Preset MV valid immediately

Preset MV Valid immediately setting for the regulatory control blocks has two options. When [Preset MV valid by Preset Switch] is selected, after the preset MV switch (PSW) is activated by an external commands such as from a sequence control block, the preset MV will be output immediately.

When [Preset MV valid at next scan] is selected, after the preset MV switch (PSW) is activated by an external commands such as from a sequence control block, the preset MV will be output at the timing that the block is scaned again.

IMPORTANT

For pulse width output, [Preset MV valid immediately] Option cannot be set to [Preset MV valid immediately] .

TIP

In CENTUM V and CENTUM-XL, when a preset MV switch (PSW) is activated by an external commands such as from a sequence control block, the preset MV will be output immediately. For the system to compliant the CENTUM V and CENTUM-XL, [Preset MV valid by Preset Switch] option should be used.

Preset MV valid immediately setting can be specified on FCS properties sheet. When checking the option of [Preset MV valid by Preset Switch] , the preset MV will be immediately forced to the MV output instead of waiting for the next scan; while when checking the option of [Preset MV valid at next scan], the preset MV will be forced to the MV output at the next scan.The Default is [Preset MV valid at next scan].

4.5 Output Tracking

The output tracking is a function that forces the output value to match the value of the output destination or the value of the tracking-input signal.

■ Output Tracking

▼ Output Value Tracking

The output tracking behaves different between the Regulatory Control Blocks and Calculation Blocks.

- For the Regulatory Control Blocks, the manipulated output value (MV) is forced to match the value of the tracking signal input terminal (TIN) or output destination.
- In the Calculation Blocks, the calculated output value (CPV) is forced to match the value of the output destination when the data status of the destination block is conditional (CND).

■ Output Tracking in the Regulatory Control Block

The output tracking in the Regulatory Control Blocks is a function that forces the manipulated output value (MV) to match the value of the tracking signal input terminal (TIN) or output destination.

However, when the value of the tracking signal input terminal (TIN) or the output terminal connection destination falls outside the range of the manipulated output value (MV) scale, the manipulated output value (MV) is restricted to the MV scale low limit value (MSL) or MV scale high limit value (MSH).

In the Cascade Signal Distributor Block (FOUT), when each output point becomes cascade open, the manipulated output value (MV) is made to conform to the output destination data.

The output tracking in the Regulatory Control Block functions under the following conditions:

- In the tracking mode (TRK).
- In the initialization manual mode (IMAN)
- In a condition other than initialization manual mode where the initialization process is required (IMAN state).

When a Regulatory Control Block is defined to give pulse width output, the Remote/Local input signal is connected to the tracking switch input terminal (TSI), and the Valve opening feedback signal is connected to the tracking signal input terminal (TIN). If the data status of the tracking signal input terminal (TIN) becomes "PIO Not Ready" (NRDY), whether the block mode is tracking mode (TRK) or not, the block mode remains unchanged, the value immediately before the "PIO Not ready" (NRDY) is retained as the manipulated output value (MV), and an output open alarm (OOP) is activated.

The output tracking for Manual Loader Blocks (MLD, MLD-PVI and MLD-SW) can be set on Function Block Detail Builder.

Output Tracking: Selectable between "Yes" and "No."
Default is "No."

The output tracking for Enhanced Two-Position ONOFF Controller Block (ONOFF-E) and Enhanced Three-Position ONOFF Controller Block (ONOFF-GE) can be set on Function Block Detail Builder.

Output Tracking: Selectable between "Yes" and "No."
Default is "Yes."

TIP

For Regulatory Control Blocks with a remote manipulated output value (RMV), the remote manipulated value (RMV) data is made to conform to the manipulated output value (MV) when the block mode is other than remote output (ROUT) or service off (O/S).

■ Output Tracking of Calculation Blocks

The output tracking of the Calculation Blocks is a function that forces the calculated output value to match the value of the connected destination.

Even if calibration (CAL) is set as the data status of the calculated output value (CPV), the output tracking have a priority over it. The output tracking of the Calculation Blocks will operate when the data status of the output destination block status becomes conditional (CND). The conditional (CND) status are as follows:

- When the cascade connection is disconnected.
- When the downstream function block begins the operation in non-cascade mode.

The output tracking can be defined on the Function Block Detail Builder.

Output Value Tracking: Selectable from “Yes” or “No.”
Default is “No.”

Additionally, if the output tracking is set to “No,” and when the status of the output destination becomes CND, the previously calculated output value (CPV) will be held.

■ Output Tracking in the Tracking Mode

The Regulatory Control Blocks where the tracking (TRK) mode is valid are provided with the tracking switch (TSW). When the tracking switch (TSW) is turned to “ON,” the Regulatory Control Blocks operate in the tracking (TRK) mode.

If a function block operates in the tracking (TRK) mode, the output tracking is activated and the value of the tracking signal input terminal (TIN) from the external becomes its manipulated output value (MV). In this case, no output limiter and output velocity limiter functions on the value of the tracking signal input terminal (TIN).

Tracking switch (TSW) may be set either directly by an external data set action or by data reference through the tracking-switch input terminal (TSI).

● Occurrence of Data Errors in the Tracking Mode

When a data error (BAD) occurs, from either the tracking signal input terminal (TIN) or tracking-switch input terminal (TSI), an output open (OOP) alarm will be issued. Since the initialization manual condition is established at this time, the active mode switches from the tracking (TRK) mode to initialization manual (IMAN) mode. When the function block operates in the initialization manual mode (IMAN), the output tracking is activated to match the manipulated output value (MV) to the value of the output destination. For the Regulatory Control Blocks whose MV is pulse width output signal, the tracking (TRK) mode will prevail, and the previous manipulated output value (MV) will be held when above error occurs.

If the block is not in tracking (TRK) mode, the occurrence of a data error (BAD) in the signal at the tracking signal input terminal (TIN) or tracking-switch input terminal (TSI) does not invoke an alarm nor change the active mode. Control will continue regardless the occurrence of data error.

● PIO Not Ready Alarm and Output Tracking

When the block is in tracking (TRK) mode, if the tracking input signal on TIN terminal becomes PIO Not Ready (NRDY) status, the manipulated output (MV) will keep the previous good value and initiates an OOP alarm. However, the block mode is unchanged. PIO Not Ready is caused by the connected I/O module power failure, or by the initialization of Inter-

Station Data Link Block (ADL). When the causes of PIO Not Ready is resolved, the OOP alarm will vanish and the block will recover to its control activity.

When a Regulatory Control Block is defined to give pulse width output, if data status of the tracking signal input terminal (TIN) becomes “PIO Not Ready” (NRDY), whether the block mode is tracking mode (TRK) or not, the block mode remains unchanged, the value immediately before the “PIO Not ready” (NRDY) is retained as the manipulated output value (MV), and an output open alarm (OOP) is activated.

■ Output Tracking in the Initialization Manual Mode

When multiple Regulatory Control Blocks are in cascade connection, the manipulated output value (MV) of the upstream control loop is used as the setpoint value (SV) for downstream loop. If the cascade connection opens, the upstream loop enters to the initialization manual (IMAN) mode.

When a function block is in the initialization manual mode (IMAN), the output tracking function force its manipulated output value (MV) to match the value of the output destination.

In the case that the cascade connection is established via a selector switch, when the selector switch opens the cascade connection, the upstream loop changes to the initializing manual (IMAN) mode, when the connection restores, it tracks its MV to the destination only once. When the switch closes the cascade connection again, the downstream loop SV will change to bumpless cascade set value.

When the initialization manual mode is invoked from the tracking (TRK) mode, the initialization manual mode prevails.

■ Output Tracking in the IMAN Mode

In the mode other than the initialization manual mode, a situation calling for the initializing process is referred as IMAN status. A function block under the following circumstances is referred as in the IMAN status:

- When initialization manual condition established in a function block that has no initialization manual (IMAN) mode.
- When the mode of a block returns from off-service (O/S) mode to manual (MAN), automatic (AUT) or cascade (CAS) mode.
- On the first scan after returning from the initialization manual mode to the non-initialization manual mode.

As the function block operates in the IMAN state, the output tracking is activated to match the manipulated output value (MV) to the value of the connection destination.

4.6 Output Range Tracking

Output range tracking is a function that forces the scale High/Low limits of the manipulated output value (MV) to match those of the output destination, and the values of data items related to the manipulated output value (MV) are recalculated whenever there is a change in the scale High/Low limits.

■ Output Range Tracking

In a Regulatory Control Block, data are processed as engineering unit data. For this reason, whenever the manipulated output value (MV) of an upstream function block is used as the setpoint value (SV) of the downstream function block in a cascade connection, the scale High/Low limits of the both blocks must be identical. The output range tracking function matches these ranges automatically. Output range tracking function will force the scale High/Low-limit setpoints of the manipulated output value (MV) to match those of the output destination, and recalculates the values of the data items related to the manipulated output value (MV) whenever there is a change in the scale High/Low limits.

Output range tracking operates only when the OUT terminal of an upstream Regulatory Control Block is connected to the SET terminal of a downstream Regulatory Control Block. The following figure shows an example of connection in cascade where output range tracking works.

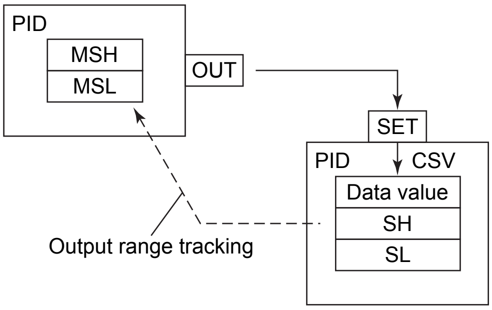


Figure 4.6-1 Output Range Tracking

For instance, if the output terminal's connected destination is a PID controller block (PID) with the PV range of 0 to 1500 m^3/h, the range of manipulated output values (MV) will also be 0 to 1500 m^3/h.

If the output destination is a process output, the range will be 0.0 to 100.0 % regardless of the output signal format. For instance, if the output destination is the analog output module, the manipulated output value (MV) will be 0.0 to 100.0 %.

■ Recalculation

The manipulated output value (MV)'s scale High/Low limits are applied to the following data items related to the manipulated output value (MV).

- Remote manipulated output value (RMV)
- Preset manipulated output value (PMV)
- Output High/Low limit indexes (OPHI and OPLO)
- Reset signal value (RLV1 and RLV2)
- Manipulated output High/Low limit setpoint values (MH and ML)

These data will be recalculated with the manipulated output value (MV) whenever there is a change in the MV scale High/Low limits. The formula for recalculation is:

$$\text{DATA.n} = \frac{\text{MSH.n-MSL.n}}{\text{MSH.o-MSL.o}} \cdot (\text{DATA.o-MSL.o})+\text{MSL.n}$$

Figure 4.6-2 The formula for recalculation

DATA.o	:	Target data before change
MSH.o	:	High limit of MV scale before change
MSL.o	:	Low limit of MV scale before change
DATA.n	:	Target data after change
MSH.n	:	High limit of MV scale after change
MSL.n	:	Low limit of MV scale after change

IMPORTANT

- In changing the output destination of the manipulated output value by means of switch blocks, etc., make sure that the downstream block does not have a conflict output. The engineering unit must be identical. If the range discrepancy exist, using Cascade Signal Distributor Block (FOUT) or Control Signal Splitter Block (SPLIT) is required.
- The output velocity limiter value is not automatically recalculated, even if there is a change in the MV scale High/Low limits. Use a Control Signal Splitter Block (SPLIT) for switching, if the destination of the manipulated output applied with velocity limiter in effect. Thus, the recalculation for output velocity limit is unnecessary.
- If a function block without output range tracking function, such as a Calculation Block, is placed in the middle of a cascade connection, it is necessary to set the MV scale High and Low limits for the upper stream regulatory control blocks. The MV scale High/Low limits can be defined on the Function Block Detail Builder.

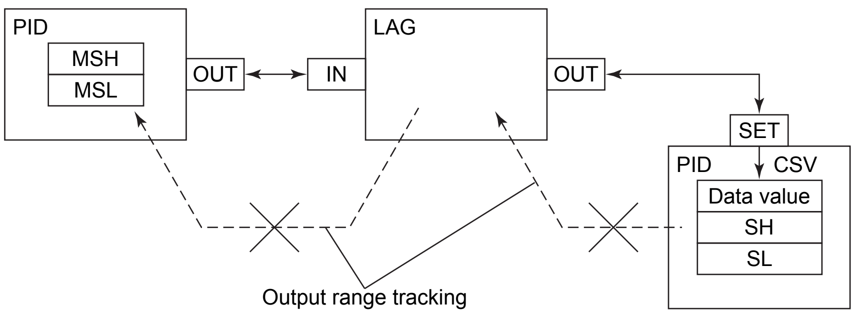


Figure 4.6-3 A Cascade Connection via a Calculation Block

- If multiple controllers are connected in parallel in the downstream of a cascade control loop, use a Cascade Signal Distributor Block (FOUT). Without using FOUT block multiple downstream control blocks cannot be chained by terminal connection.

4.7 Manipulated Output Index

This function displays indexes that show the permissible range of the manually manipulated values at the normal operation.

The manipulated output index is only available for Regulatory Control Blocks.

■ Manipulated Output Index

▼ Index

This function displays two indexes in the manipulated output value (MV) scale on the operation monitoring window of the operation and monitoring function. These are called the manipulated output indexes.

By setting these indexes at the operable limits of the manipulated output values (MV), they can be used as manipulation guides in the manual mode, or as guides for verifying normal status in the automatic mode.

For a Regulatory Control Block with manipulated output value (MV), both the high output limit (OPHI) and low output limit (OPLO) indexes can be set on the tuning view. These limits are displayed in the operation and monitoring window of the operation and monitoring function.

The indexes may be defined on the Function Block Detail Builder.

Set Indexes: Selectable from “Yes” and “No.”
Default is “Yes.”

■ Setting Parameters of Manipulated Output Index

The following items are the parameters of the manipulated output index:

High output limit index: In an engineering unit within MV scale range.
Default is the high limit of the MV scale.

Low output limit index: In an engineering unit within MV scale range.
Default is the low limit of the MV scale.

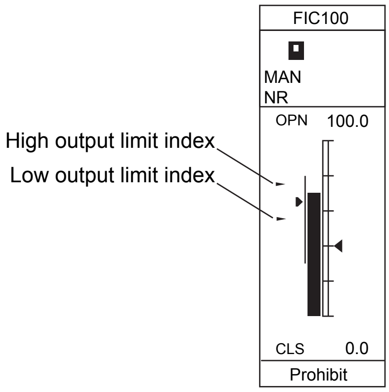


Figure 4.7-1 Example of Manipulated Output Index Display

4.8 Output Signal Conversion

This process converts the result of calculation process into an output format for the output modules or other function blocks.

■ Output Signal Conversion

▼ Output Signal Conversion

The output signal conversion may be used for the processes that are common to the Regulatory Control Blocks and the Calculation Blocks, and for the specific function blocks which have specific output process function.

SEE ALSO For more information about each type of the output signal conversion processes, refer to:

- 4.8.1, "No-Conversion" on page 4-27
- 4.8.2, "Pulse Width Output Conversion" on page 4-31
- 4.8.3, "Communication Output Conversion" on page 4-43
- 4.8.4, "Output Signal Conversion of Logic Operation Blocks" on page 4-44

■ Output Signal Conversion Process Common to Regulatory Control Blocks

Here is the outline of the output signal conversion processes that are common to the Regulatory Control Blocks:

● No-Conversion Output

The manipulated output value (MV) resulted from the control-calculation process is No-Conversion output.

● Pulse Width Output Conversion

The changes of manipulated output value (ΔMV) is output after converted into a pulse width signal.

● Communication Output Conversion

The manipulated output value (MV) resulted from the control-calculation process is converted into the format compatible with the destination subsystem.

■ Output Signal Conversion Process Common to Calculation Blocks

Here is the outline of the output conversion processes that are common to the Calculation Blocks:

● No-Conversion Outputs

The calculated output value (CPV) resulted from the control-calculation process is no-conversion output.

● Communication Output Conversion

The calculated output value (CPV) resulted from the control-calculation process is converted into the format compatible with the destination subsystem.

● Output Signal Conversion of the Logic Operation Blocks

In Logic Operation Blocks (*1), the output is unconverted.

*1: Logic Operation Block can be used in FCSs except PFCS.

■ Output Signal Conversion Processes for Specific Function Blocks

Here are the outlines of the conversion processes for different types of specific function blocks:

● Output Signal Conversion of the Motor Control Blocks (MC-2, MC-2E, MC-3, and MC-3E)

One of the following types of output may be specified: 2-position status output; 2-position pulsive output; 3-position status output or 3-position pulsive output.

- 2-position status output
  The process switches one contact point ON or OFF according to the manipulated output value (MV).
- 3-position status output
  The process switches two contact points ON or OFF according to the manipulated output value (MV).
- 2-position pulsive output
  The process switches one of the two contacts ON for one second according to the manipulated output value (MV).
- 3-position pulsive output
  The process switches one of the three contacts ON for one second according to the manipulated output value (MV).

SEE ALSO For more information about output processing specific to MC-2, MC-2E, MC-3, and MC-3E blocks, refer to:

1.17.3, “Output Processing of Motor Control Blocks (MC-2, MC-2E, MC-3 and MC-3E)” in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)

● Output Signal Conversion of the Two-Position ON/OFF Controller Block (ONOFF) and the Enhanced Two-Position ON/OFF Controller Block (ONOFF-E)

The Two-position status output is used.

SEE ALSO For more information about output processing specific to ONOFF, ONOFF-E block, refer to:

“■ Two-Position Status Output” in 1.8, “Two-Position ON/OFF Controller Block (ONOFF), Enhanced Two-Position ON/OFF Controller Block (ONOFF-E)” in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)

● Output Signal Conversion of the Three-Position ON/OFF Controller Block (ONOFF-G) and the Enhanced Three-Position ON/OFF Controller Block (ONOFF-GE)

The Three-position status output is used.

SEE ALSO For more information about output processing specific to ONOFF-G, ONOFF-GE block, refer to:

“■ Three-Position Status Output” in 1.9, “Three-Position ON/OFF Controller Block (ONOFF-G), Enhanced Three-Position ON/OFF Controller Block (ONOFF-GE)” in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)

● Output Signal Conversion, of the Time-Proportioning ON/OFF Controller Block (PID-TP)

This is applied for a time-proportioning ON/OFF output. Time-proportioning ON/OFF is a type of status output which set the contact output to ON via digital output module in proportional to the manipulated output value (MV) ON/OFF cycle.

SEE ALSO For more information about output processing specific to PID-TP block, refer to:

"■ Time-Proportioning ON/OFF Output" in 1.10, "Time-Proportioning ON/OFF Controller Block (PID-TP)" in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)

● Output Signal Conversion of Flow/Weight-Totalizing Batch Set Block (BESTU-2 and BESTU-3)

The 2-position or 3-position status output is used.

SEE ALSO For more information about output processing specific to BSETU-2, BSETU-3 blocks, refer to:

1.20.3, "Output Processing of Totalizing Batch Set Blocks (BSETU-2, BSETU-3)" in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)

● Output Signal Conversion of the Pulse Count Input Block (PTC)

A specific output process is used in the PTC block when it is connected to sequence blocks.

SEE ALSO For more information about output processing specific to PTC block, refer to:

"■ Output Signal Conversion of the Pulse Count Input Block (PTC)" in 1.32, "Pulse Count Input Block (PTC)" in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)

4.8.1 No-Conversion

The No-Conversion output is the data resulted from control computation process and given as output without signal conversion process.

This can be used in the Regulatory Control Blocks and Calculation Blocks. Regulatory Control Blocks and Calculation Blocks behave differently when they are defined to use the No-Conversion output.

Here describes the behaviors of the function blocks that are defined to use No-Conversion output:

■ No-Conversion in the Regulatory Control Block

The No-Conversion output is to be connected to another function block or an analog output module.

Here explains the both cases.

● Output to Another Function Block

The data output is carried out by data set to the other function blocks or by terminal connection to the other function blocks.

- Output by data set
  The manipulated output value (MV) is given from the OUT terminal can be used for data set, as well as the process variable (PV) from the OUT terminal of the input indicator block (PVI) or the input indicator block with deviation alarm (PVI-DV). The manipulated output value (MV) and process variable (PV) are no-conversion outputs, and in forms of engineering unit.

- Output by terminal connection
  The manipulated output value (MV) is set in the cascade setpoint value (CSV) of the connection destination function block via the SET terminal of the connection destination function block and the OUT terminal.

  When an inter-terminal connection is possible at the IN terminal of the connection destination function block, the manipulated output value (MV) can be passed to the process variable (PV) or calculated input value (RV) of the connection destination function block via the IN terminal of the connection destination function block and the OUT terminal.

● Output to the Analog Output Module

When connect the output to the analog output module, the tight-shut and full-open functions are automatically added to the manipulated output value (MV). The direction of analog output can also be defined.

Output to a analog output module is in term of data set output.

Manipulated output value (MV) of 0 to 100 % is given from the OUT terminal.

The analog output module converts the 0 to 100 % the manipulated output value (MV) data into a 4 to 20 mA (or 1 to 5 V) output to drive a final control element, such as a control valve.

● Analog Output Direction

The analog output module outputs 4 to 20 mA (or 1 to 5 V) against the 0 to 100 % range of manipulated output values (MV). However, it can also output in the reverse direction, e.g.20 to 4 mA (or 5 to 1 V). The following figure shows the relationship between the manipulated output value (MV) and output current. The reverse settings are shown in a dotted line.

The output direction can be defined on the Detailed setting items in the IOM Builder.

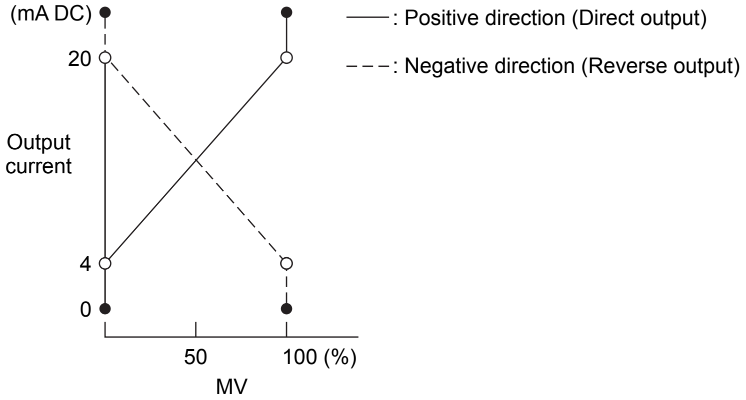


Figure 4.8.1-1 Relationship between the Manipulated Output Value and Output Current

● Tight-shut and Full-Open

In order to tightly shut a valve the manipulated output (MV) can be decreased to a value smaller than 0 % and fully open the valve, while to fully open the valve, the manipulated output (MV) can be increased to a value greater than 100 %.

All the regulatory function blocks with manipulated output (MV) and manual mode (MAN) support the Tight-Shut and Full-Open output except the 13-Zone Program Set block.

Tight-shut and Full-Open option can be defined on the detailed builder of the function block.

On the function block detailed builder, the Output Value for Tight-shut (Ms) should be set with a value smaller than 0 % and the Output Value for Full-Open (Mf) should be set with a value greater than 100 %.

IMPORTANT

- If the output destination of the manipulated value (MV) of a function block having the 'Tight shut/Full open' feature is a Fieldbus Faceplate block, set the 'Tight shut/Full open' specification to "No" in the Function Block Detail Builder.
- In the Function Block Detail Builder, the "Output Value for Tight-shut (Ms)" and "Output Value for Full-open (Mf)" settings are named assuming that it is the valves with reverse actions that fully closes when the manipulated value (MV) is 0%. For the valves that close tightly which fully close when the manipulated value (MV) is 100%, the "Output Value for Tight-shut (Ms)" setting is for decreasing the actual output value to less than 0%, when the manipulated value (MV) is 0%. Note that the "Output Value for Full-open (Mf)" setting is for increasing the actual output valve to more than 100%, when the manipulated value (MV) is 100%.

The following figure illustrates the relationship between the manipulated output value and the actual output value:

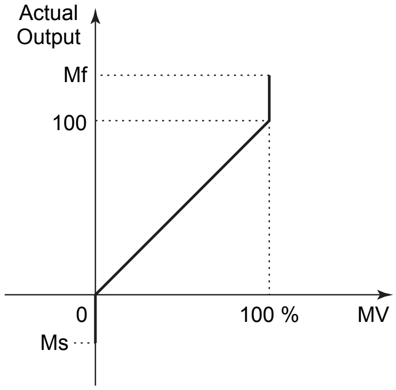


Figure 4.8.1-2 Tight-shut and Full-Open

On the Function Block Detail builder, the settings regarding tight-shut and full-open need to be defined.

- Tightly-shut/fully-open
  Choose “Yes” or “No.” The default setting is “Yes.”
- Output Value for tight-shut (Ms)
  The actual output value for tight-shut when the manipulated output (MV) indicates 0 %.

  Setting range is -17.19 to 117.19 % (5 significant figures). By default, the tight-shut value is -17.19 %.
- Output Value for full-open (Mf)
  The actual output value for full-open when the manipulated output (MV) indicates 100 %.

  Setting range is -17.19 to 117.19 % (5 significant figures). By default, the full-open value is 106.25 %.

The current and voltage (mA and V) correspond to the actual output of full-open and the actual current output of tight-shut, varies according to the output direction of the analog output module (direct or reversed output).

Now we use a current output module as an example to explain how the actual output current (mA) corresponds to the actual tight-shut output (Ms) and actual full-open output (Mf).

When the output direction of an analog output module is not reversed, the actual output current (mA) corresponds to the manipulated values (MV) 0 % and 100 % with 4 mA and 20 mA.

- Output Current = (0.16 • MV + 4) mA

In this case, we can find out that when the default tight-shut MV is -17.19 %, the current output for tight-shut (Ms) will become 1.25 mA, while when the default full-open MV is 106.25 %, the current output for full-open (Mf) will become 21 mA.

If the output direction of the analog output module is reversed, the actual output current (mA) corresponds to the manipulated values (MV) 0 % and 100 % with 20 mA and 4 mA.

- Output Current = (-0.16 • MV + 20) mA

In this case, we can find out that when the default tight-shut MV is -17.19 %, the current output for tight-shut (Ms) will become 22.75 mA (corresponds to 117.19 %), while when the default full-open MV is 106.25, the current the current output for full-open (Mf) will become 3 mA (corresponds to -6.25 %).

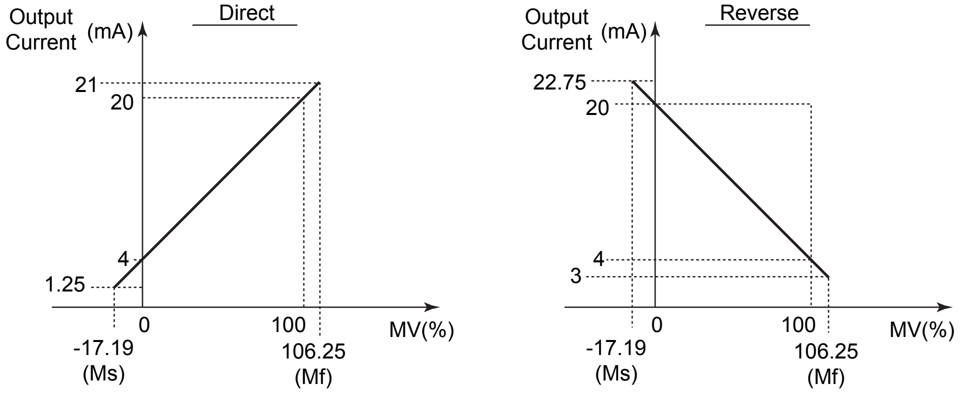


Figure 4.8.1-3 Actual Output Current for Tight-Shut and Full-Open

IMPORTANT

Even when the function block faceplate displays 0.0 % output, the actual output may not go to tight-shut. This phenomenon is caused due to the HIS trunks out the second digit after the decimal point of the displayed MV. Thus even the displayed MV is 0.0 % but the actual MV of FCS is still greater than 0.0 % since FCS does not truncate the MV.
When manipulate a function block to ensure the block gives a tight-shut output, i.e., to make sure the MV actually becomes zero, and the following operations can be performed:

- On the data entry dialog box, enter MV=0 directly
- Keep pushing the [DEC] key on the operation keyboard for one more second or even longer time after the MV of the function block becomes 0.0 %

SEE ALSO For more information about analog out direction , refer to:

"● Analog Output Direction" on page 4-27

■ No-Conversion in the Calculation Block

When the No-Conversion output is specified, the connection destination will be another function block or an analog output module.

Its use in different cases is explained below:

● Output to Another Function Block

The output is given unconverted to another function block either by data set or terminal connection.

The calculated output value (CPV) or the change in calculated output value (ΔCPV) is no-conversion output from the SUB terminal by data set.

● Output to an Analog Output Module

Calculation output value (CPV) is converted to 0 to 100 % on the CPV scale and output through the OUT or SUB terminal. The range of the output, converted in terms of 0 to 100 % will be limited to 0 to 100 %.

Output terminals other than OUT and SUB give the output value (CPVn) unconverted, regardless of the destination type.

4.8.2 Pulse Width Output Conversion

This conversion method converts the changes in manipulated output value (ΔMV) into a pulse width signal. It is used for the contact output module to open and close the motor-operated valve through the contacts.

The conversion to pulse width output is available only in the Regulatory Control Blocks.

■ Pulse Width Output Conversion

In this conversion method, the changes in manipulated variable (ΔMV) are converted into a pulse width signal. The degree of opening for the motor-operated valve can be manipulated by outputting this pulse width signal from the contact output module as two contact outputs (UP, DOWN) corresponding to the sign of the manipulated variable (ΔMV).

Furthermore, the displayed manipulated output value (MV) for the pulse width output conversion and the output action for increasing or decreasing the manipulated output value are different depending on the feedback input signal availability.

Table 4.8.2-1 Regulatory Control Blocks where Pulse width Output Conversion is Available

Type of regulatory control block	With feedback input(*1)	Without feedback input(*1)
PID Controller Block (PID)	x	x
Sampling PI Controller Block (PI-HLD)	x	x
PID Controller Block with Batch Switch (PID-BSW)	x	-
PD Controller Block with Manual Reset (PD-MR)	x	-
Blending PI Controller Block (PI-BLEND)	x	x
Self-Tuning PID Controller Block (PID-STC)	x	x
Ratio Set Block (RATIO)	x	-
Manual Loader Block (MLD, MLD-PVI and MLD-SW)	x	-
Feedforward Signal Summing Block (FFSUM)	x	-
Non-Interference Control Output Block (XCPL)	x	-

*1: x: available
-: not available

IMPORTANT

If a function block is applied with pulse-width output but no feedback of valve position, the control output action of the function should be specified as Velocity Type. If the function block control output action is specified as Positional Type, the output may not be correct. On the Function block detail builder, in the [Output] tab, the output type can be selected for the [Control Calculation Output Type] item.
For the pulse-width output, when the output is Velocity Type, the manipulated output change (ΔMV) will be output.

For the pulse-width output, when the output is Positional Type, the difference between the calculated manipulated output value (MV) and the feedback value will be output.

● Computational Expression for Pulse Width

The pulse width of pulse signal is given by the following computational expression:

$$\text{Tout}=\text{Pf} \cdot \frac{\Delta \text{MV}}{100}$$

Figure 4.8.2-1 The computational expression for pulse width

Tout : Output pulse width (sec.)

Pf : Pulse width stroke value (sec.)

ΔMV : Change in manipulated output (%)

The stroke of the pulse width is the pulse width in time that is required to operate the final control element from full-closed to full-opened state. In the automatic operation, the output pulse width can not be defined greater than the control period time.

The full stroke value is defined using the Function Block Detail Builder.

Full stroke value: The setting range is 0.00 to 7200.00 (sec.).
Default is 0.00 sec.

● Minimum Output Width

The final control element may not move if the pulse width signal is below a specific value because of the mechanical characteristics of the object. To prevent this happens, the minimum output width is utilized that, when a pulse width smaller than the minimum output width, this pulse is withheld to add to the next pulse, until the pulse width becomes wider than the minimum output value. A pulse width of fractional value below the output resolution (10 ms) will also be added to the next output.

The minimum pulse width is only valid in automatic operation.

The minimum pulse width is defined through the Function Block Detail Builder.

The minimum pulse width: Setting range is 0.00 to 7200.00 (sec.).
Default is 0.00 sec.

● Backlash Compensation

When the output reverses from the previous direction, a compensation value is added to the calculated output to compensate the backlash of the final control element.

The backlash compensation is set through the Function Block Detail Builder.

Backlash compensation value: Setting range is 0.00 to 7200.00 (sec.).
Default is 0.00 sec.

● Resetting Pulse Width

If the pulse width reset switch (RSW) is turned ON from the sequence control block or others, the pulse width signal being output is reset immediately. The pulse width reset switch returns to OFF after resetting the pulse width signal.

In the case of automatic operation (AUT, CAS and RCAS), no pulse width signal will be output until the next control period after reset the pulse width.

■ Output Contacts

The output contacts are the contacts that drive final control elements, such as an motor-operated valve. There are two types of contacts: the first contact and the second contact.

The terminal number for the first contact is defined through the Function Block Detail Builder. An odd terminal number in the digital output module should be assigned for the first contact. The second contact is automatically assigned with the succeeding terminal number of the first contact number.

The First Contact

The first contact is called “UP contact” and activated when the change in manipulated output (ΔMV) is a positive value.

If the feedback input signal is defined to indicate the valve opening, it increases when the UP contact is ON.

The Second Contact

The second contact is called “DOWN contact” and is activated when the change in manipulated output (ΔMV) is a negative value.

If the feedback input signal is defined to indicate the valve opening, it decreases when the DOWN contact is ON.

Remote/Local Switch

The on-site operation panel compatible with the pulse-width output may have an operation button for local control. If you wish to perform on-site operation, change the remote/local switch on the on-site operation panel to local to stop output from the FCS, then output the on-site operation button signal. When changing the switch between remote and local, the Regulatory Control Block receives the remote/local switching contact signal in the tracking switch input connection terminal (TSI). When this input turns ON, the tracking switch (TSW) turns ON and the Regulatory Control Block is set to the tracking (TRK) mode.

The operation upon switching between the remote and local modes is explained below.

Operation Upon Switching from Remote Mode to Local Mode

- The on-site operation is enabled.
- If the feedback input is provided, the feedback input value will be displayed as the manipulated output value (MV). If no feedback defined, a 50 % (fixed) value is displayed as the manipulated output value (MV).
- Only the pulse width being output at the time the mode is switched continuous till the pulse output completed.
- The block mode is switched to the tracking (TRK) mode.

Operation Upon Switching from Local Mode to Remote Mode (Other Than TRK Mode)

- The calculated output value of the Regulatory Control Block becomes valid.
- If the feedback input is provided, the calculated output value will be displayed as the manipulated output value (MV) immediately after switching, and thereafter, the feedback input value will be displayed as the manipulated output value (MV).
- The pulse width signal being output at the time the mode is switched will be reset.

Feedback Input

The feedback input communicates to the Regulatory Control Block to notify the absolute value of the manipulated output (MV), e.g. the opening of the valve connected as the final control element.

The pulse width output outputs the change in manipulated output (ΔMV). The absolute value of the manipulated output (MV) is not known.

The feedback signal from the final control element is connected to the tracking signal input terminal (TIN) of the Regulatory Control Block.

The feedback signal is provided or not provided decide how the manipulated output value (MV) display and how the manipulated output behaves to increase or decrease the output. The pulse width output behavior is described in the following table:

Table 4.8.2-2 Pulse width Output Operation

Item			With feedback input	Without feedback input
MV display	Remote mode	AUT mode	Feedback input value	During UP pulse output: Changes in +direction During DOWN pulse output: Changes in -direction While pulse is stopped: MV is held. (*1)
		MAN mode	During pulse output: Manual set point value While pulse is stopped: Feedback input value	During UP pulse output: Changes in +direction During DOWN pulse output: Changes in -direction While pulse is stopped: 50 %
	Local mode		Feedback input value	50 %
Pulse output status on depressing INC/DEC key			Pulse equivalent to ΔMV is output	Pulse equivalent to ΔMV is output. However, when MV reaches 0 or 100 %, pulse continues to output until the INC/DEC operation is ended.(*2)

*1: If the pulse width output of PID is specified to be CENTUM V compatible, the MV displays at the position of 50 % when no pulse is output.

*2: When MV reaches 0 or 100 %, the pulse output will stop right after the INC/DEC operation is ended. The actual pulse output is irrelevant to the ΔMV manipulated by the INC/DEC operation.

● No Feedback, Pulse Width Output MV at AUT Mode

▼ PID Pulse Width Output

If no feedback is set, the MV displays in the following two styles at the AUT mode.

- Hold Previous MV
  When no pulse is output (or the output pulse width smaller than threshold), the displayed MV keeps the previous MV. When pulse output restarts, if the restarted output is in the same direction of the previous output, the displayed MV equals to the previous MV plus the ΔMV for increment or previous MV minus the ΔMV for decrement. However, if the restarted output is in reversed direction, the displayed MV equals to 50 % plus or minus the ΔMV for increment or decrement.

- Display 50 % MV (CENTUM-V Compatible)
  When no pulse is output (or the output pulse width smaller than threshold), the displayed MV returns to 50 %. When pulse output restarts, the displayed MV equals to the 50 % plus the ΔMV for increment or 50 % minus the ΔMV for decrement. Whenever the ΔMV reverses from positive to negative or vice versa, the displayed MV returns to 50 % then plus or minus the ΔMV for increment or decrement respectively.

The MV display style when no feedback for pulse width output and the block is at AUT mode can be set on FCS properties sheet. Check the check box of [CENTUM-V compatible MV Display] in the column of [PID Pulse Width Output]. When this option is checked, the displayed MV returns to 50 % when no pulse is output. Otherwise, the displayed MV keeps the previous MV. This check box is unchecked by default.

● An Example of Pulse Width Output Conversion Operation with Feedback Signal

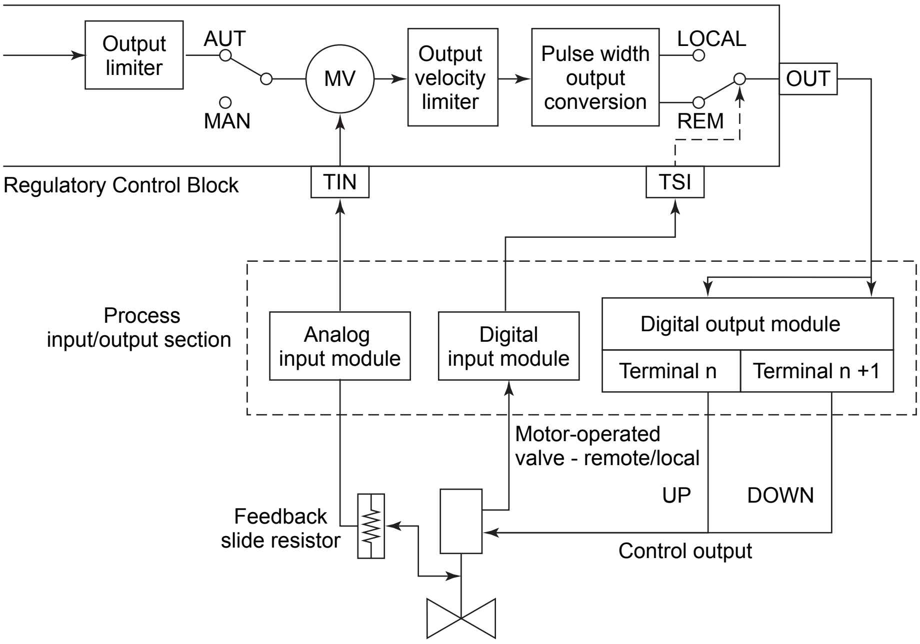


Figure 4.8.2-2 Schematic Diagram for Pulse Width Output Signal with Feedback Input

During automatic operation, the feedback input value which indicates how far the valve is open is displayed as the manipulated output value (MV).

When the manipulated output value (MV) from the operation and monitoring function is changed during manual operation, the pulse width corresponding to the change is output. The manipulated output value (MV) displays the manipulated output value (MV) set manually during pulse width output, and when output is complete it displays the feedback input value.

The timing chart for manual operation is shown below:

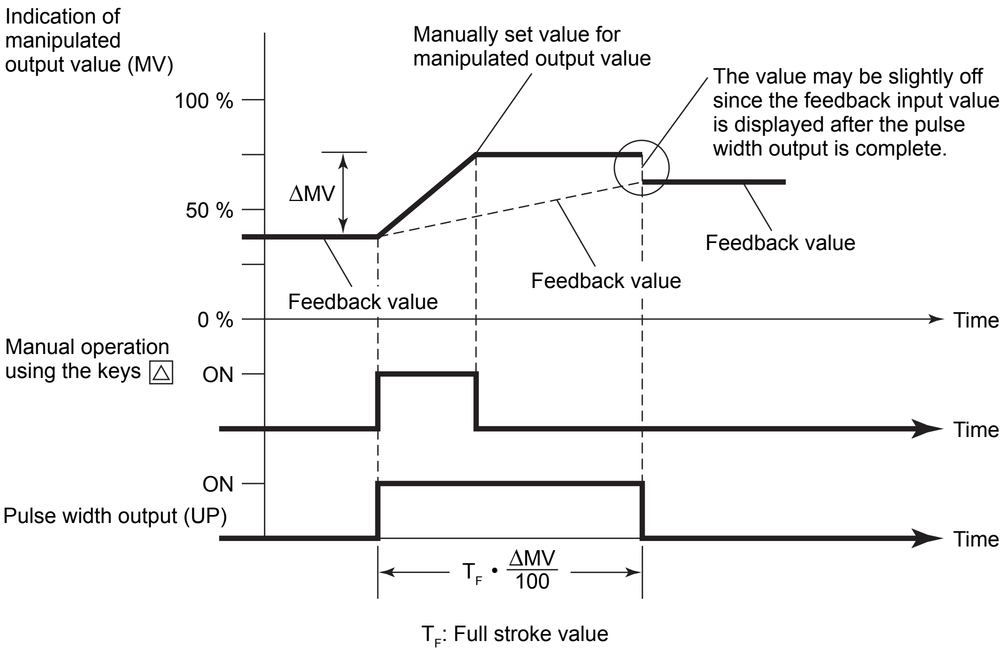


Figure 4.8.2-3 Timing Chart for Manual Operation

● An Example of Pulse Width Output Conversion Operation without Feedback

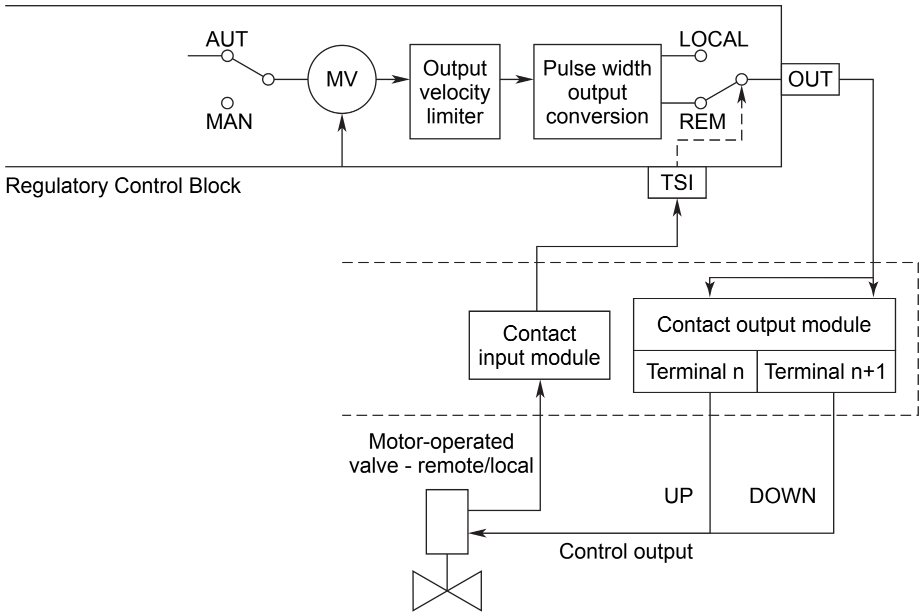


Figure 4.8.2-4 Schematic Diagram of Pulse Width Output Signal Conversion without Feedback Input

The valve opening will not be displayed since there is no feedback signal.

The manipulated output value (MV) in the automatic operation is displayed by UP/DOWN of the pulse width output.

When Up:	manipulated output value (MV) increases.
When DOWN:	manipulated output value (MV) decreases.
No output:	manipulated output value (MV) stays unchanged. (*1)

*1: When the PID pulse width output is specified as CENTUM V compatible, the displayed MV returns to 50 % when no pulse is output.

The timing chart for manual operation is shown below:

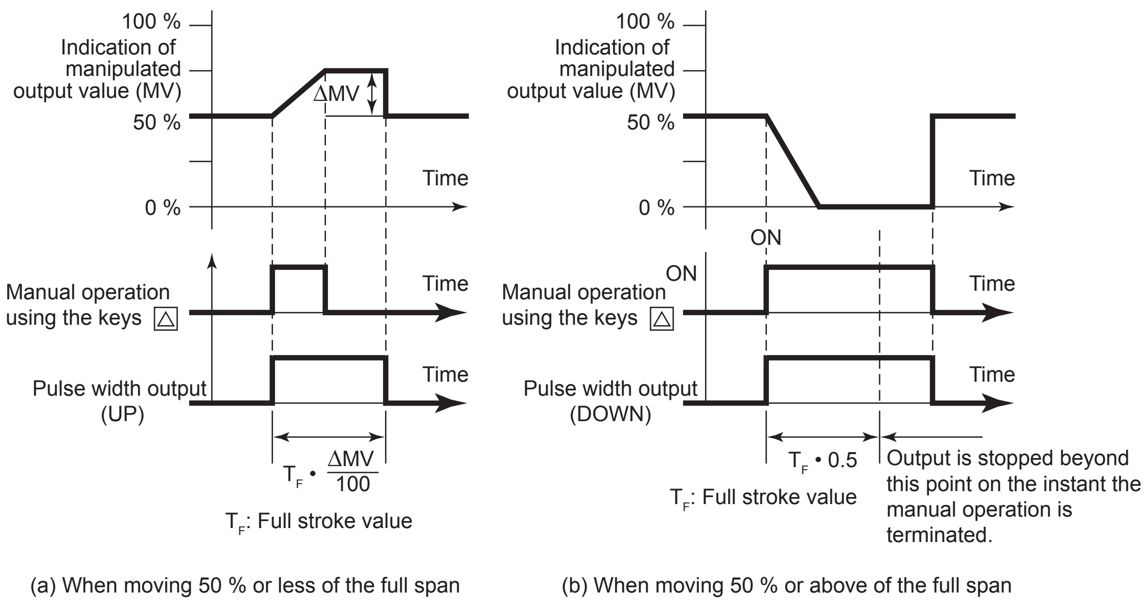
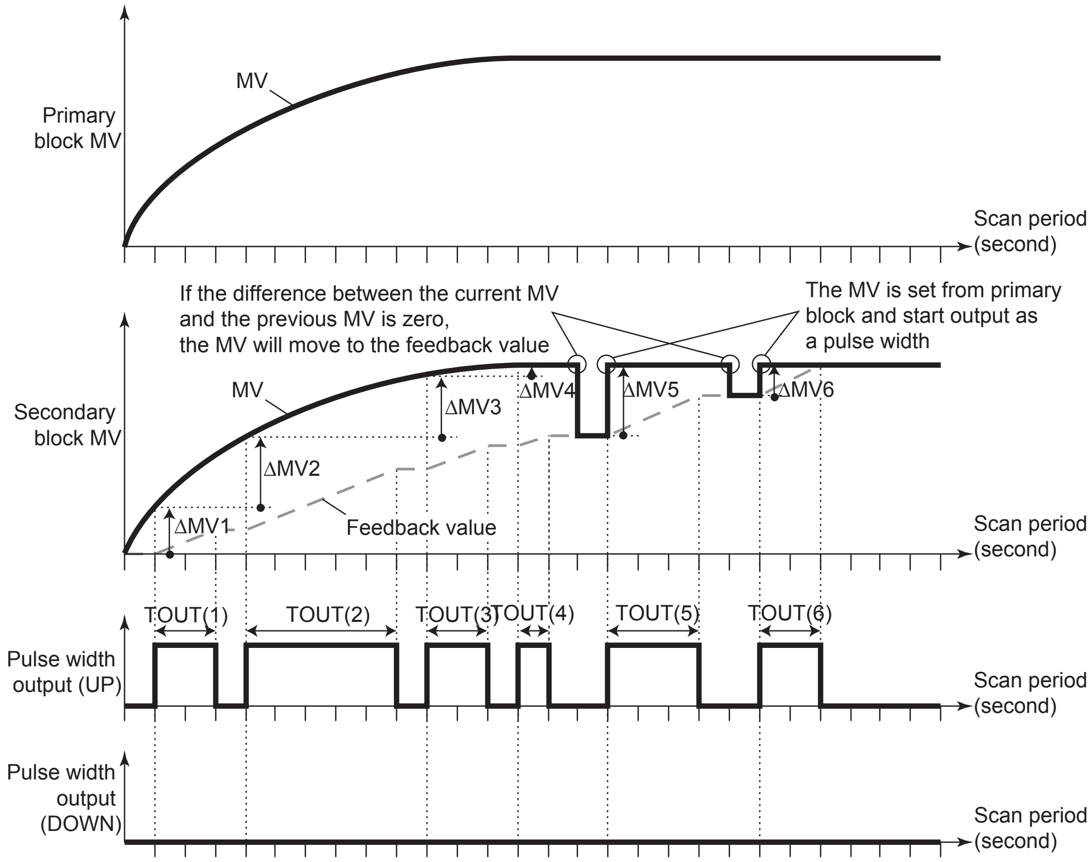


Figure 4.8.2-5 Timing Chart for Manual Operation

■ Pulse Width Output in PRD Mode

The behaviors of the pulse width output are different depending on the feedback input signal availability.

The behavior of the pulse width output when feedback input signal is available is shown as follows:



$$T_{OUT(n)} = Pf \cdot \frac{\Delta MVn}{100}$$ Pf: Full stroke value

Figure 4.8.2-6 Action of The Pulse Width Output when Feedback Input Signal is Available

The behavior of the pulse width output when feedback input signal is unavailable is shown as follows:

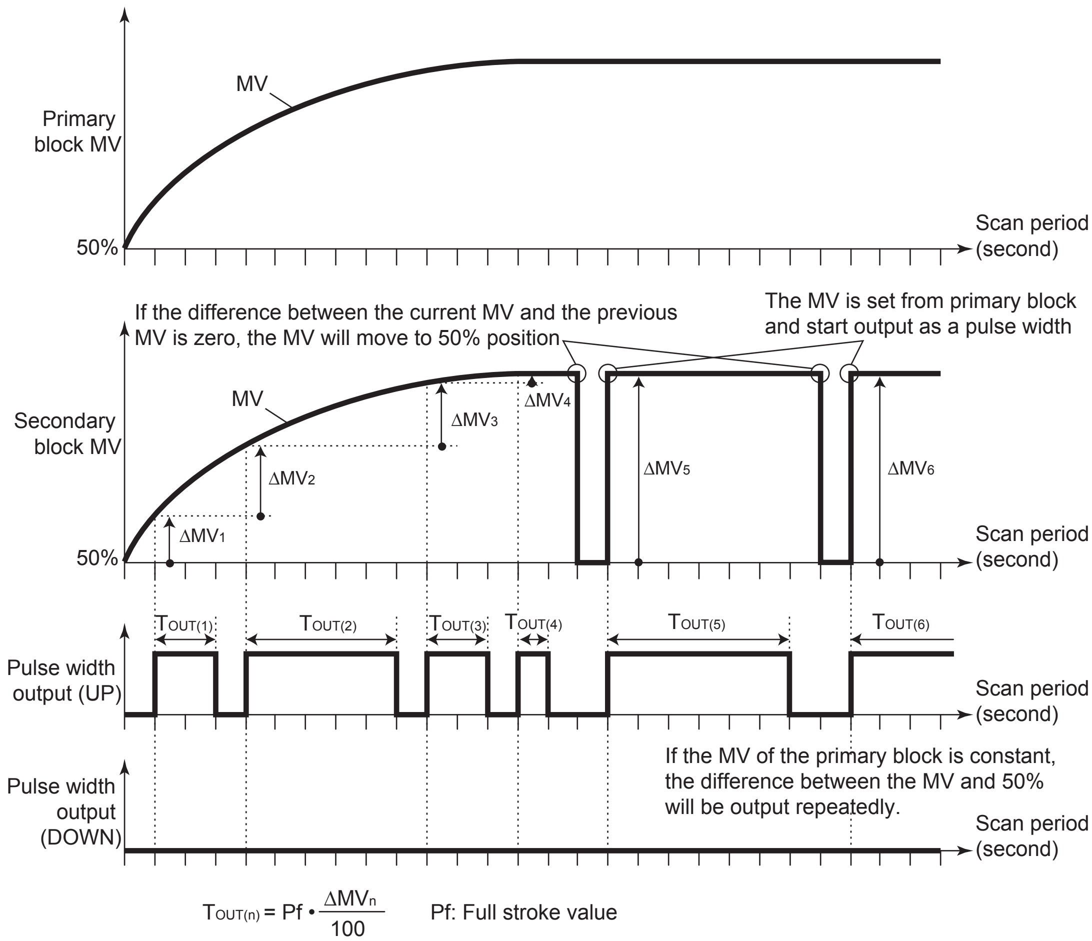


Figure 4.8.2-7 Action of The Pulse Width Output when Feedback Input Signal is Unavailable

Output Limiter in PRD Mode

The output limiter can be applied even when the block mode is PRD.

However, even if the output limiter is applied in PRD mode, for the pulse width output, when the output value is greater than the high limit of MV (MH) or smaller than the low limit of MV (ML) of the function block, the output limiter will not immediately function to limit the current pulse output. The output limiter will start function to limit the next pulse width output.

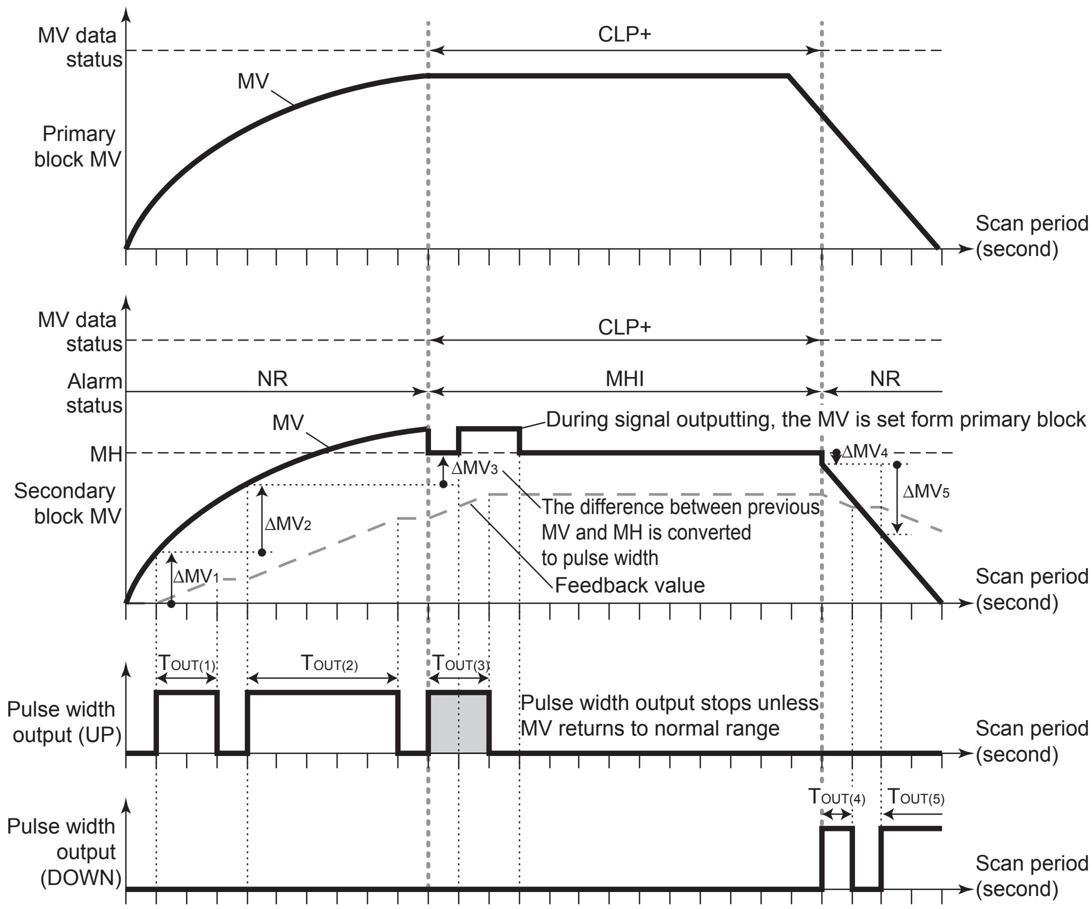
To limit the direct output value of primary loop, check the option of [Enable output limiter in PRD mode (XL compatible)] on [Constant 3] tab of FCS properties sheet.

This feature is available for the following function blocks:

PID, PI-HLD, PID-BSW, PD-MR, PID-STC

For the function block that the output limiter is functioning, when the manipulated output value (MV) of the primary loop becomes greater than the output high limit (MH) or lower than the output low limit (ML), and this phenomenon is prolonged to the next scan. The manipulated output high limit alarm (MHI) or manipulated output low limit alarm (MLO) will occur. The data status becomes high limit clamp (CLP+) or low limit clamp (CLP-). And the pulse width signal for the ΔMV, the difference between the current MV and the high or low limit is output. After the pulse signal is output, output value (MV) will be replaced by high limit (MH) or low limit (ML). When the output limiter is functioning, there will be no pulse width output unless the MV of primary loop returns to a value within the range between the output high limit (MH) and the output low limit (ML).

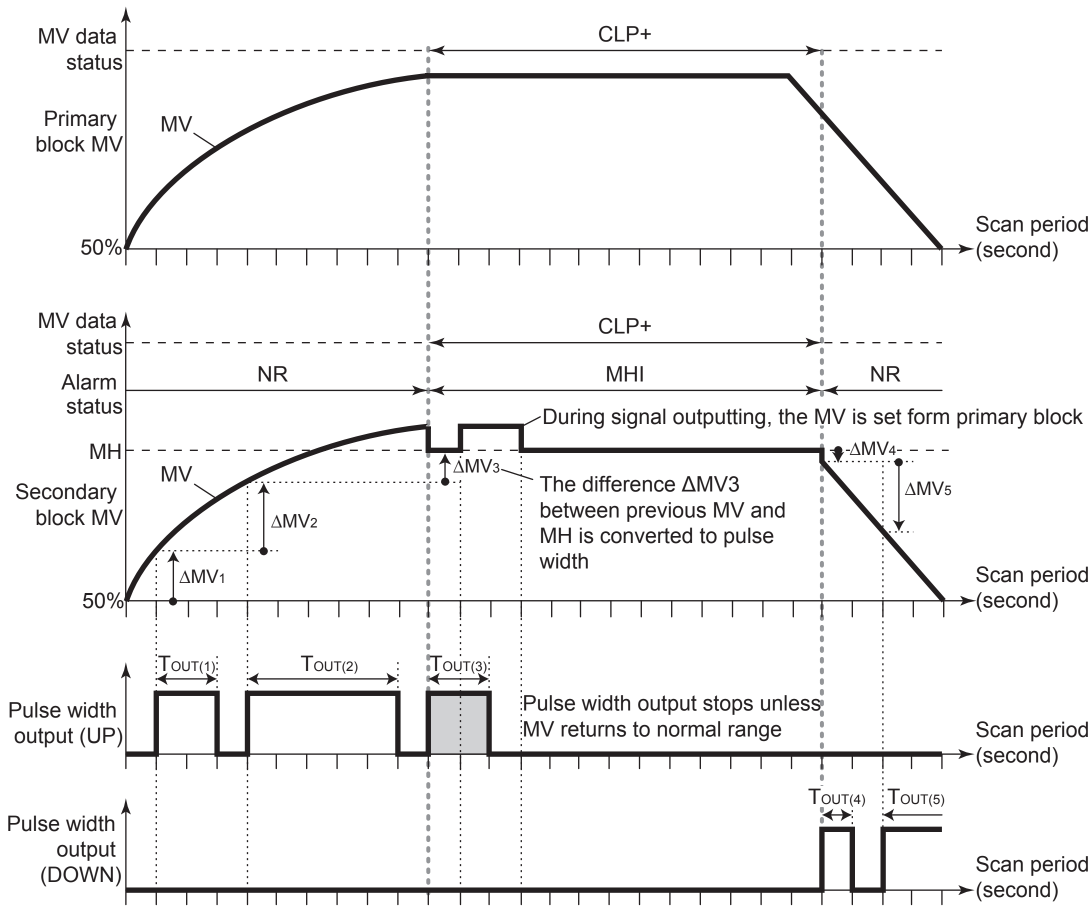
The behavior of the pulse width output when feedback input signal is available is shown as follows. Action when the option of [Limit output in direction when clamped] of the primary block is Yes.



$$T_{OUT(n)} = Pf \cdot \frac{\Delta MV_n}{100}$$ Pf: Full stroke value

Figure 4.8.2-8 Action of The pulse width output when feedback input signal is Available

The behavior of the pulse width output when feedback input signal is unavailable is shown as follows. Action when the option of [Limit output in direction when clamped] of the primary block is Yes.



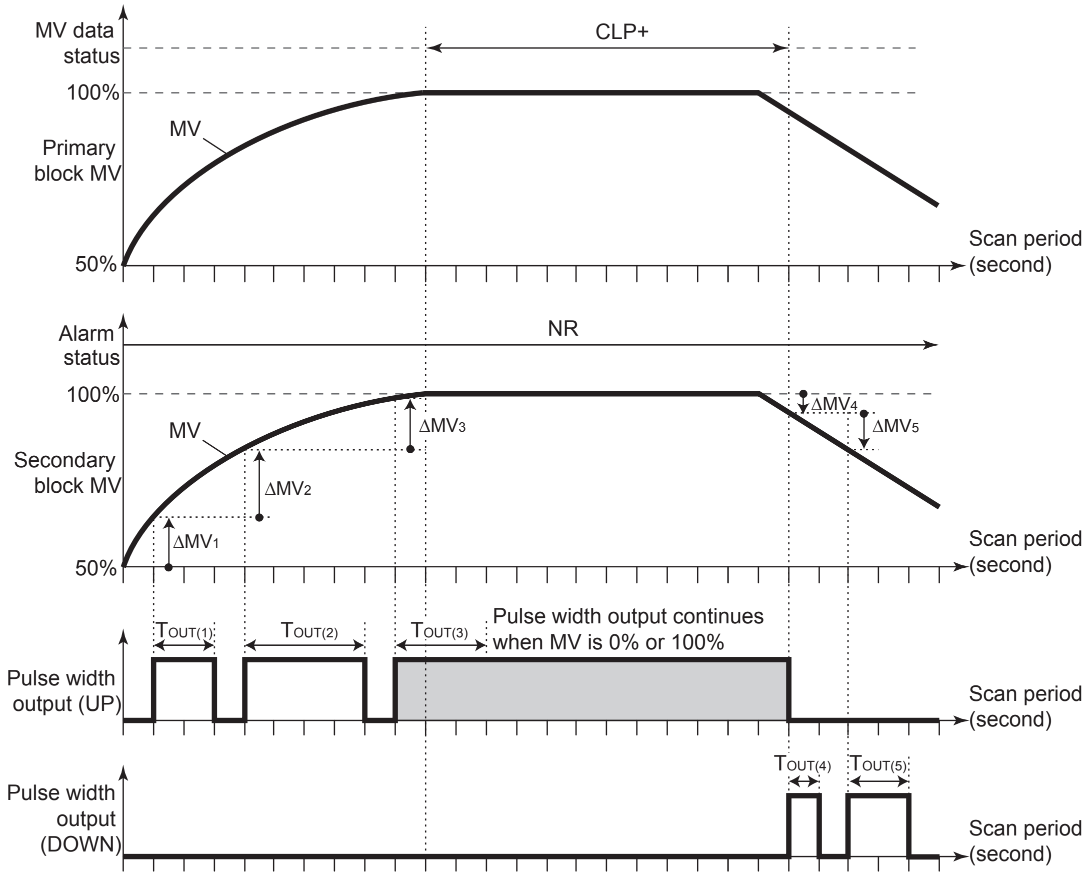
$T_{OUT(n)} = Pf \cdot \frac{\Delta MV_n}{100}$ Pf: Full stroke value

Figure 4.8.2-9 Action of The Pulse Width Output when Feedback Input Signal is Unavailable

Suppose the output high limit (MH) is set to 100% and the output low limit (ML) is set to 0% while the output feedback is not available, the actions are as follows:

For the function block in PRD mode, if the MV of primary block is 100%, the UP pulse width signal will be continuously output. If the MV of primary block is 0%, the DOWN pulse width signal will be continuously output.

The actions are as follows. Action when the option of [Limit output in direction when clamped] of the primary block is Yes.



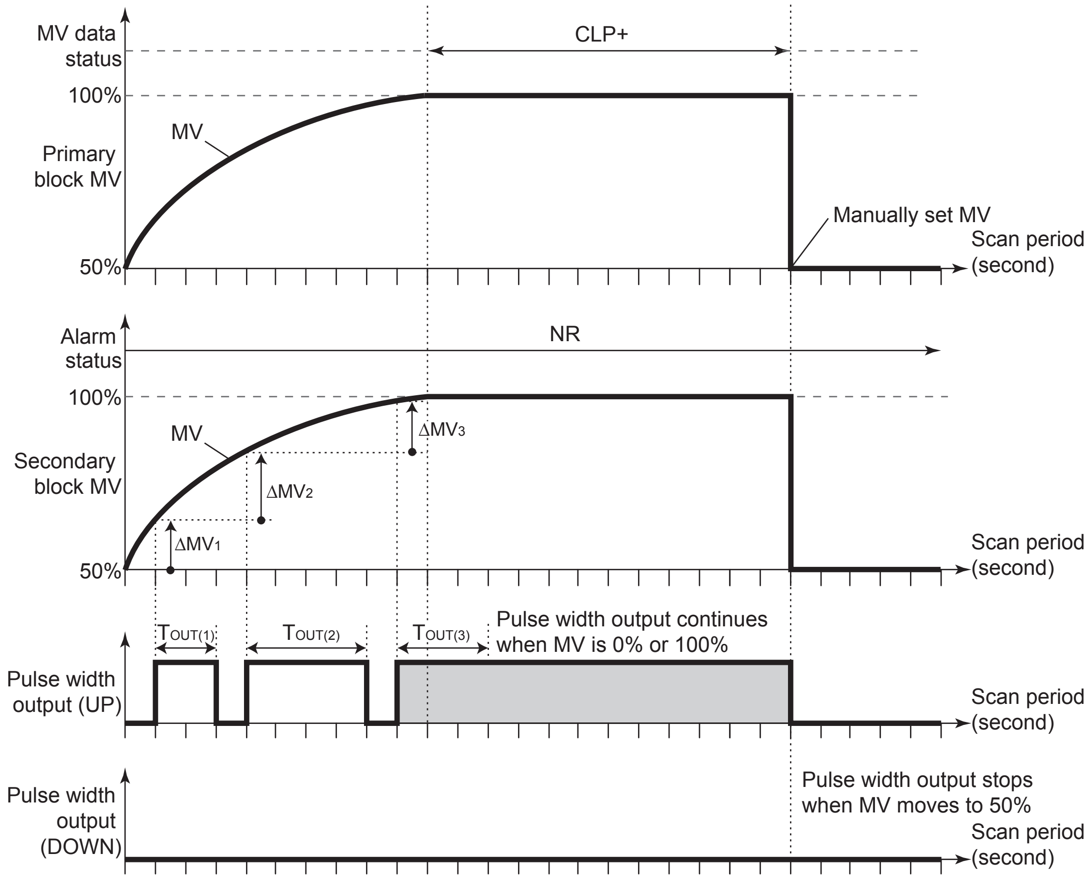
$$T_{OUT(n)} = Pf \cdot \frac{\Delta MV_n}{100}$$

Pf: Full stroke value

Figure 4.8.2-10 An instance when Manipulated Output Value (MV) reaches 100%

For the function block in PRD mode, if the MV is 50%, the pulse width output will stop.

The actions are as follows. Action when the option of [Limit output in direction when clamped] of the primary block is Yes.



$$T_{OUT(n)} = Pf \bullet \frac{\Delta MV_n}{100}$$ Pf: Full stroke value

Figure 4.8.2-11 An instance when Manipulated Output Value (MV) reaches 50%

4.8.3 Communication Output Conversion

The communication output conversion converts the data resulted from control computation into a format that can be output to the destination subsystem.

This conversion method is available for the Regulatory Control Blocks and Calculation Blocks. It behaves differently for Regulatory Control Block and Calculation Block.

■ Communication Output Conversion

The data value to be converted to the communication output will be the calculated output value (MV) and calculated output value (CPV) on the regulatory control and Calculation Blocks, respectively.

The computational expression for the communication output conversion is shown below:

$$OUT = \frac{1}{GAIN} \cdot (MV - BIAS)$$

Figure 4.8.3-1 The computational expression for the communication output conversion

OUT : Subsystem output value

GAIN : Data conversion gain

BIAS : Data conversion bias

The data conversion gain and bias are set through the function block detail definition builder.

Data Conversion Gain:	Set a 7-digit numeric value using a sign and a decimal point. Setting range is between -9999999 and 9999999 Default is 1.000.
Data Conversion Bias:	Set a 7-digit numeric value using a sign and a decimal point. Setting range is between -9999999 and 9999999 Default is 0.000.

TIP

If a function block is defined with communication output conversion, the following restrictions will be applied:

- Output reversal (i.e., to reverse the analog output signal) will not be supported.
- Though the output velocity limiter and the velocity type output can be applied to controller blocks, however the output velocity limiter and the velocity type output may not function properly in the subsystem since the subsystem communication takes longer time. For an example, when the output of a controller block is limited to 1 % per second, if the communication period is 3-seconds, the output limiter to the subsystem will become 3 % per three seconds.

4.8.4 Output Signal Conversion of Logic Operation Blocks

The following paragraph explains the output signal conversion of Logic Operation Blocks (*1) .

*1: Logic Operation Block can be used in FCSs except PFCS.

■ Output Signal Conversion of Logic Operation Blocks

The logic calculated value (CPV) for the output to the connection destination connected to the OUT terminal is passed to the destination block without any output processing. The data type of calculated value is integer type.

TIP

In bitwise logic operation blocks, CPV is displayed in 8 digits hexadecimal.

4.9 Auxiliary Output

The auxiliary output is used when output a signal to a destination other than the final control element. The signal is often used as compensation data to other function blocks, or to the indicator outside of the FCS, etc.

The operation of auxiliary output is different between the Regulatory Control Block and the Calculation Block.

This section explains the operation in the auxiliary output.

■ Auxiliary Output from the Regulatory Control Block

▼ Auxiliary Output

The auxiliary output is used when output a signal through the SUB terminal to a destination other than the final control element. The signal is often used as compensation data to other function blocks, or to the indicator outside of the FCS, etc. In the Regulatory Control Blocks, the process variable (PV), change in process variable (ΔPV), manipulated output value (MV), or the change in manipulated output value (ΔMV) is output via the SUB terminal. The connection method is the data setting.

The connection destinations of the SUB terminal are indicated below.

- Process output
- Data item of other function block

In function blocks with the output compensation function, the change of MV before the output compensation is considered to be the change in manipulated output value (ΔMV).

Even if the output of the auxiliary output becomes open, the alarm status does not change to an output open alarm (OOP).

The operation of the auxiliary output is determined by the settings for the auxiliary output builder definition items “Output Data” and “Output Type.”

The auxiliary output builder definition items “Output Data” and “Output Type” are set with the Function Block Detail Builder.

Table 4.9-1 Selection List and Default Values for Output Data of Auxiliary Output

Name of the subject function block	Selection list	Default value
Manual Loader Block (MLD) Manual Loader Block with Auto/Man SW (MLD-SW) Velocity Limiter Block (VELLIM)	[MV] [ΔMV]	[MV]
Input Indicator Block (PVI) Input Indicator Block with Deviation Alarm (PVI-DV) 2-Position ON/OFF Controller Block (ONOFF) Enhansed Two-Position ON/OFF Controller Block (ONOFF-E) 3-Position ON/OFF Controller Block (ONOFF-G) Enhansed Three-Position ON/OFF Controller Block (ONOFF-GE) Non-Interference Control Output Block (XCPL)	[PV] [ΔPV]	[PV]
Regulatory control blocks other than those listed above	[MV] [ΔMV] [PV] [ΔPV]	[PV]

- Output Type:
  Selectable from “Positional Output Action” and “Velocity Output Action.”

  Default is “Positional Output Action.”

When the output action for auxiliary output is set to “Positional Output Action,” the output values (MV, ΔMV, PV, or ΔPV) can be set in the connection destination as it is. Also, when set to the "Velocity Output Action" type, the value read back from the connection destination is added to the output value and set in the connection destination.

● When the Connection Destination is a Process I/O

The output value is converted to a percentage by the following arithmetic expression to output.

$$\text{Output value}=\frac{\text{PV-SL}}{\text{SH-SL}}\bullet 100.0$$

Figure 4.9-1 The arithmetic expression when the auxiliary output is process variable (PV)

$$\text{Output value}=\frac{\Delta\text{PV}}{\text{SH-SL}}\bullet 100.0$$

Figure 4.9-2 The arithmetic expression when the auxiliary output is change in process variable (ΔPV)

$$\text{Output value}=\frac{\text{MV-MSL}}{\text{MSH-MSL}}\bullet 100.0$$

Figure 4.9-3 The arithmetic expression when the auxiliary output is manipulated output value (MV)

$$\text{Output value}=\frac{\Delta\text{MV}}{\text{MSH-MSL}}\bullet 100.0$$

Figure 4.9-4 The arithmetic expression when the auxiliary output is change in manipulated output value (ΔMV)

● When the Connection Destination is a Data Item of the Function Block

The output value is output from the SUB terminal without any conversion.

When the output action is set to positional type, the output value (PV, ΔPV, MV or ΔMV) is set to the connection destination as it is, whereas for the velocity type, the output value is added to the readback value from the connection destination and set to the connection destination.

Table 4.9-2 Relationship between the I/O Connection Methods and Output Action

Output connection method	Output value	Output action(*1)	
		Positional type	Velocity type
Process output	PV, ΔPV MV, ΔMV	x	-
Data setting to the function block		x	x

*1: x: Allowed
-: Not allowed

■ Auxiliary Output from the Calculation Block

▼ Auxiliary Output

A Calculation Block outputs the calculated output variable (CPV) or change in the calculated output value (ΔCPV) through its SUB terminal.

A batch set block with input indicator (DSET-PVI) can output the data setpoint (SV) and the change in data setpoint (ΔSV) as well as the calculated output variable (CPV) and the change in calculated output variable (ΔCPV).

The output value and output action of auxiliary output can be set with the Function Block Detail Builder. The action of the auxiliary output is determined by the settings of the auxiliary output builder definition item "Output Data" and "Output Type."

The auxiliary output builder definition items “Output Data” and “Output Type” are set with the Function Block Detail Builder.

- Output Data
  Selectable from “CPV” and “ΔCPV.” However, in the case of the DSET-PVI block, it is selectable from “CPV,” “ΔCPV,” “SV” and “ΔSV.”

  Default is “CPV.”

- Output Type
  Selectable from “Positional Output Action” and “Velocity Output Action”

  Default is “Positional Output Action.”

When the “Positional Output Action” is defined for output action, the output value (CPV, ΔCPV, SV or ΔSV) is set to the connection destination without change, whereas the output value is added to the readback value from the connection destination and set to the connection destination when set to the “Velocity Output Action.”

4.10 Output Processing in Unsteady State

In the unsteady state, the Calculation Block and the Regulatoly Control Block execute different output processing from that in the usual status.

The unsteady state includes the calibration (CAL) status and the bad data status (BAD) .

■ Operation during Calibration (CAL)

When the data status of the calculated output value (CPV) of the calculation block is in calibration (CAL), output from the secondary terminals (CPV1 to CPVn) is stopped. Manual setting of the calculated output value (CPV) will be enabled and the calculated output value (CPV) output will be available as usual.

■ Auxiliary Output (∆PV) When PV Data Status is BAD : FFCS Series/KFCS2/LFCS2

▼ ΔPV/ΔCPV Output from SUB Becomes Zero Right After IOP

According to the [PV Overshoot] setting of the Regulatory Controller Block set on the Function Block Detail Builder, the PV will overshoot when the data status becomes bad (BAD). Under such circumstance, the ΔPV output from the SUB terminal can be set to output either 0 or the actual delta PV i.e. the increment or decrement between the current-scan PV and the previous-scan PV.

For the DPV output, an option [ΔPV/ΔCPV Output from SUB Becomes Zero Right After IOP] can be checked on FCS Properties sheet. When the option is checked, the DPV output becomes 0 immediately after IOP. Otherwise, the actual DPV will be output from the SUB terminal. By default, this option is not checked.

■ Calculation Output and Auxiliary Output (CPV, ΔCPV) When CPV Data Status is BAD

When the calculation input value is abnormal or when an error occurs during the calculation processing, the data status of the calculated output value (CPV) becomes BAD (bad data value) and the previous value is retained, and the connected destination of OUT terminal will hold this retained previous value. However, the ΔCPV output from the SUB terminal will become 0 immediately.

Regardless the [PV Overshoot] setting on the Function Block Detail Builder, the output from the OUT terminal will not be affected when the calculation input (RV) becomes abnormal. If the output from the SUB terminal is CPV, the output will be the value in accordance with the [PV Overshoot] setting. If the output from the SUB terminal is ΔCPV, the output will be either the actual ΔCPV.

■ Auxiliary Output (∆CPV) When CPV Data Status is Bad : FFCS Series/KFCS2/LFCS2

▼ ΔPV/ΔCPV Output from SUB Becomes Zero Right After IOP

According to the [PV Overshoot] setting of the Regulatory Controller Block set on the Function Block Detail Builder, when the data status becomes bad (BAD), the ΔCPV output from the SUB terminal can be set to output either 0 or the actual delta CPV i.e. the increment or decrement between the current-scan CPV and the previous-scan CPV.

For the DCPV output from SUB terminal, an option [dPV/dCPV Output from SUB Becomes Zero Right After IOP] can be checked on FCS Properties sheet. When the option is checked, the DCPV output from SUB becomes 0 immediately after IOP. Otherwise, the actual DCPV will be output from the SUB terminal. By default, this option is not checked.

However, for the calculation block that the setting item [Calculated Input Value Error Detected] is specified for no reaction on the error, the DCPV output terminal will be the actual delta CPV value in accordance with the [PV Overshoot] setting even if the option [ΔPV/ΔCPV Output from SUB Becomes Zero Right After IOP] is checked.

SEE ALSO For more information about input signal conversion on erroneous calculation inputs, refer to:

“■ Input Processing at Calculated Input Value Error Detection” on page 3-29

4.11 CPV Pushback

The CPV pushback is a function to prevent from the abrupt change of the output when a cascade loop connection switched from open to close.

■ CPV Pushback

The CPV pushback is a function that use the calculated output value (CPV) obtained from tracking the downstream function block in cascade to calculate back the calculated input value (RV) for the upstream function block to track.

The CPV pushback is used to prevent the process output from abrupt changes when the Analog Calculation Block receives output signal from a Controller Block (such as a PID block) via its IN terminal by terminal connection, and the calculated output value (CPV) outputs to a Manual Loader Block with Auto/Man SW (MLD-SW) or etc.

The CPV pushback operates only when the output value tracking is defined to “YES.”

SEE ALSO
For more information about the setting of the output value tracking, refer to:

4.5, “Output Tracking” on page 4-18

■ CPV Pushback Calculation

The calculations shown below are performed during a CPV pushback.

Table 4.11-1 CPV Pushback Calculations

Type	Calculation formula
SQRT	$RV=\left(\frac{CPV}{GAIN}\right)^2$ (*1)
EXP	$RV=\ln\left(\frac{CPV}{GAIN}\right)$ (*2)
LAG INTEG LD LDLAG DLAY	$RV=\frac{CPV}{GAIN}$ (*1)
FUNC-VAR	RV is the value of X axis coordinate calculated by inputting for the Y axis the value resulting from dividing CPV by GAIN (*1).

*1: When GAIN is 0, the CPV pushback calculation is bypassed and the previous calculated input value (RV) is retained.
*2: When (CPV/GAIN) ≤ 0, the previous calculated input value (RV) is retained.

■ Example of CPV Pushback

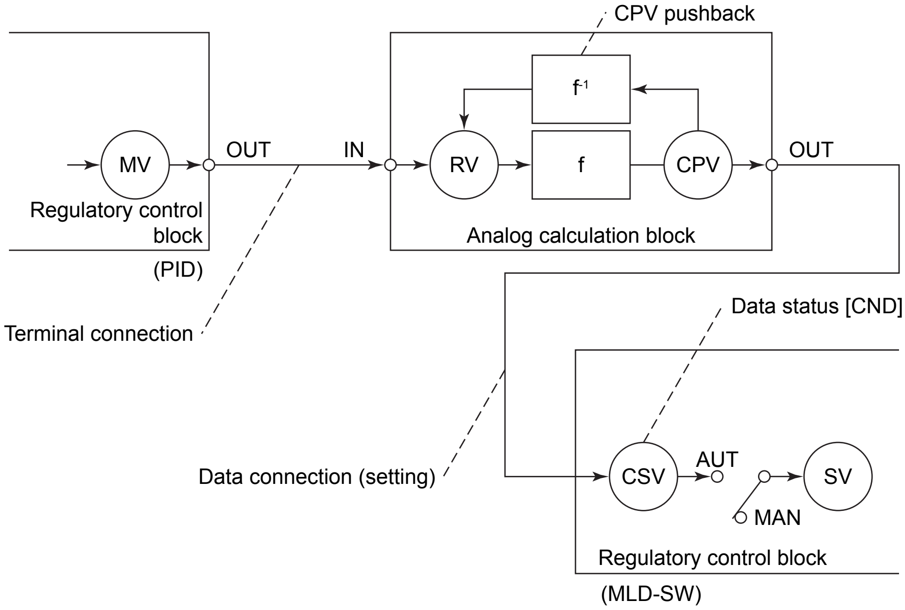


Figure 4.11-1 Example of CPV Pushback

In the above control loop, if the Calculation Block in the middle of the loop has not the CPV pushback, the upstream PID block can not track the downstream block MLD-SW when the MLD-SW is switched to MAN mode. Thus when the MLD-SW block is switched to AUT mode the bumps occurs to the cascade setting value (CSV) of the MLD-SW block.

The CPV pushback function monitors the down stream block status. When tracking is required, it uses calculated output value (CPV) tracked from the downstream block to calculate back the calculated input value (RV) for upstream block to track.

Examples of the CPV pushback operations are as follows:

- When the data status of the setting destination is CND, this function equalize the calculated output value (CPV) to the data value of the setting destination (Output tracking function).
- When tracking, a reverse calculation is carried out to calculate the calculated input value (RV) from the calculated output value (CPV) obtained via tracking.
- As the figure shows, the CPV pushback function is activated only when a loop is established via terminal connection between the IN terminal (such as a switch function block placed before the destination block) and the OUT terminal of a controller block.

Table 4.11-2 Calculation Blocks with CPV Pushback Function

Block type	Model name	Name
Analog calculation block	SQRT	Square Root Block
	EXP	Exponential Block
	LAG	First-Order Lag Block
	INTEG	Integration Block
	LD	Derivative Block
	LDLAG	Lead/Lag Block
	DLAY	Dead-Time Block
	FUNC-VAR	Variable Line-Segment Function Block

- The CPV pushback function is not available in Calculation Blocks with multiple calculation input values such as the ADD (Addition) block and TPCFL (Temperature and pressure correction) block, as well as those which cannot uniquely define a calculated output value from a calculation input value such as the RAMP block, since reverse calculation is impossible.

4.12 Output Processing in Sequence Connection

The pulse count block (PTC) as well as the blocks that can be connected in sequence connection may perform the status manipulation to the output destination function block specified in the connected OUT terminal when the logic value required becomes true.

■ Output Processing in Sequence Connection

The PTC block, logic operation block (*1) and CALCU, CALCU-C blocks can use sequence connection for the I/O connection method.

The "status manipulation" function based on output signals can be used as special output processing when the logic operation block and CALCU, CALCU-C blocks use sequence connection.

*1: Logic Operation Block can be used in FCSs except PFCS.

■ Status Manipulation

In the case of sequence connection, the I/O connection information indicated below is held in the output terminal (OUT, Jn):

- Information that identifies the connection destination, such as tag name, user definition label name, terminal number, and element number
- Information that identifies data item
- Information that shows action specifications

The information based on this I/O connection information and the logical value obtained by the PTC block, logic operation block (*1) or a CALCU, CALCU-C block, the status manipulations indicated by the output connection information that is written to the OUT terminal are performed, for the output destination function block that is also indicated by the output connection information when the required logic value in the block becomes true.

*1: Logic Operation Block can be used in FCSs except PFCS.

SEE ALSO

For more information about the action specification on the output connection information, refer to:

- 2.3.10, "Action Signal Description : Status Manipulation for Other Function Blocks and I/O Data" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)
- 2.3.11, "Syntax for Action Signal Description : Status Manipulation of Logic Chart" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)
- 2.3.12, "Syntax for Action Signal Description : Status Manipulation of Sequence Table from Logic Chart" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)

4.13 Output Processing during OOP

There two actions for the function block output processing during OOP.

- Function block tracks output module during OOP
- Function block writes to output module during OOP

■ Function Block Tracks Output Module during OOP: /FFCS Series/ KFCS2/LFCS2

▼ Function block tracks output module during OOP

On [Detailed Settings] tab of FCS Constants builder, select the option of [Function block tracks output module during OOP]. When OOP occurs, the function block MV will track the value in the output module.

TIP

For the function blocks that the output tracking is user-definable, the option of [Function block tracks output module during OOP] is valid only for the function blocks that the output tracking is enabled.

This feature is available for the following function blocks.

PID, PI-HLD, PID-BSW, PD-MR, PI-BLEND, PID-STC, MLD, MLD-PVI, MLD-SW, RATIO, PG-L13, BSETU-2(AO only), BSETU-3(AO only), VELLIM, AS-H, AS-M, AS-L, FFSUM, XCPL, SPLIT, FSBSET

SEE ALSO

For more information about the output tracking, refer to:

4.5, “Output Tracking” on page 4-18

■ Function Block Writes to Output Module during OOP: FFCS Series/ KFCS2

When OOP occurs, the function block can write the MV to the output module. This feature can be enabled or disabled by a system common switch.

When this feature is enabled, the function block will write the MV to the output module even OOP occurs. A system alarm will occur when function block writes the MV to the output module during OOP.

This feature is available for the following function blocks.

PID, PI-HLD, PID-BSW, PD-MR, PI-BLEND, PID-STC, MLD, MLD-PVI, MLD-SW, RATIO, PG-L13, BSETU-2(AO only), BSETU-3(AO only), VELLIM, AS-H, AS-M, AS-L, FFSUM, XCPL, FSBSET

IMPORTANT

his feature is available only for the R5.03.00 or newer FCS where the function block is hosted and HIS where the function block MV is manipulated.

When the connection recovers, the output module MV will be output to the destination device. The incorrect setting may apply bad affect to the plant. Please handle with cautions.

TIP

- The output module that supports this feature is FIO output module only.
- This feature is valid when OOP occurs where a channel of output module encounters a disconnection or malfunction. If OOP occurs due to output module failure, the function block cannot write MV to the output module.

● System Common Switch

This feature can be enabled or disabled by toggling a system common switch. In the table below, the system common switches used for enabling or disabling this feature for different FCSs are listed.

Table 4.13-1 Common switches used for enabling or disabling this feature

FCS Type	Number	Setting
FFCS/FFCS-L/KFCS2	%SW0015	0 (Disabled), 1 (Enabled)
FFCS-V	%SW8015	

● Operation for Writing MV to Output Module

When the option of [Function block writes to output module during OOP] is enabled, only the MV from the followings can be written to the output module.

- MV changed on Tuning View, Faceplate or Graphic data entry dialog box
- MV changed on Graphic menu dialog
- MV changed by Graphic instrument commend
- MV changed by INC/DEC buttons on Tuning View or Faceplate.

MV changed by the following means cannot be written to the output module.

- MV sent from an OPC client through the OPC server
- MV written by a FCS SEBOL program or from a ADL block.

TIP

If the MV change is performed from an ActiveX control or a .Net component through the OPC interface of Exaopc (either "Exaopc OPC interface package for HIS" or "Exaopc NTPF100"), the changed MV will not be written to the output module.

● Security Level of System Common Switch

The security level of the system common switch for toggling this feature is level 4. If a different security level is applied to the switch, this feature may not be properly toggled due to the operator's privilege level.

SEE ALSO

For more information about the security levels, refer to:

"■ Security Levels" in 3.4, "Function Block Security" in Engineering Reference Vol.1 (IM 33K03G21-50E)

● Messages Triggered when Writing MV to Output Module

When writing MV to the output module during OOP, a system alarm message will be generated. This system alarm message will not be displayed on System Alarm Viewer, however, it can be found on Message Monitor window.

IMPORTANT

Whether the MV was properly written to the output module cannot be confirmed on the display of the function block. Thus, it is necessary to check the system alarm on the Message Monitor window when the MV is changed during OOP.

TIP

- In order to check the system alarm messages on Message Monitor window, the system alarm message should be registered on the Message Registration dialog box of Message Monitor window.
- When using INC/DEC button to change MV, the system alarm may occur at every second while the button is pressed.
- In the HIS prior than R5.03.00, this system alarm message cannot be displayed.

SEE ALSO

For more information about the system alarm messages triggered by this operation, refer to:

"No.0468 Written to IOM in OOP Status" in 2.5, "Control Station Status Change Related Messages (Message Nos. 0400 to 0491)" in Operating Messages (IM 33K02D20-50E)

For more information about Message Registration dialog box of Message Monitor window, refer to:

"■ Message Registration Dialog Box" in 3.7.2, "Message Registration" in Human Interface Stations Reference Vol.1 (IM 33K03F21-50E)

● Priorities of this Feature and OOP Clear

This feature has higher priority than OOP Clear. When [OOP Clear] is enabled while [Function block writes MV to output module during OOP] is also enabled, if MV is changed before time-up of OOP Clear, writing MV to the output module will be performed, and OOP Clear will be aborted.

TIP

The above-mentioned action is definable for each terminal of signal channel. Therefore, for the channel in the same output module, if no MV is written to the channel, the OOP Clear will function.

SEE ALSO

For more information about the OOP clear, refer to:

"■ OOP Clear (OPCLS) - FIO Analog Input/Output : FFCS series/KFCS2/KFCS" in A3.4.1, "Parameters for FIO Analog Inputs/Outputs" in Field Control Stations Reference (IM33K03E10-50E)

● Actions Vary with [Function block writes to output module during OOP] and [Function block tracks output module during OOP] Settings

Under the following cases, the actions are performed differently:

- Condition A: [Function block writes to output module during OOP] is enabled
- Condition B: [Function block tracks output module during OOP] is enabled

The action will be performed according to the existence of condition A and condition B.

Table 4.13-2 Function block MV actions during OOP

Condition A	Condition B	Actions
Yes	Yes	Function block writes its MV to the output module. Function block MV equals to the value in the output module.
Yes	No	Function block writes its MV to the output module. Function block MV equals to the value in the output module.

Condition A	Condition B	Actions
No	Yes	Function block does not write its MV to the output module. Function block MV tracks the value in the output module so that function block MV equals to the value held in the output module when OOP occurred.
No	No	Function block does not write its MV to the output module. Function block MV does not track the value in the output module so that function block MV may be different from the value held in the output module.

TIP

Note that if the OOP occurred due to the output module failure, the function block cannot write its MV to the output module, thus only [Function block tracks output module during OOP] feature functions.

● How to Confirm that Function Block Writes to Output Module during OOP

Do the following to confirm that the function block writes its MV to the output module. After the confirmation, connect the destination device to the module to eliminate OOP.

1. Identify which FCS that the manipulated function block belongs to.
2. Confirm that the versions of HIS where the function block MV is manipulated and FCS where the function block is hosted are R5.03.00 or newer.

IMPORTANT

This feature is available only for the R5.03.00 or newer FCS where the function block is hosted and HIS where the function block MV is manipulated.

3. Toggle the system common switch (%SW0015 or %SW8015) to 1 on the FCS where the manipulated function block is hosted so as enable this feature.
4. Manipulate the MV of the target function block.
5. On the HIS message monitor window, confirm that the system alarm message indicating the function block has written a value to the output module exist.
6. Toggle the system common switch (%SW0015 or %SW8015) to 0 on the FCS where the manipulated function block is hosted so as disable this feature.

IMPORTANT

After writing MV to the output module, it is necessary to disable this feature.

7. On the history report, check the system alarm messages and the MV operation records to confirm that the MV has been properly written to the output module.

TIP

- Do one of the followings to identify which FCS that the function block is hosted.
  - Perform a name search on System View using the tag name of the function block to find its location.
  - Perform a name search on Process Report View using the tag name of the function block and refer the system tag name to find its location.
- The version number of the FCS where the target function block is hosted can be found on FCS Status Display window of HIS.
- The version number of the HIS where the operation is performed can be found on the [All] tab of Software Configuration Viewer.
- Changing common switch security level may be performed on Common Switch builder by entering a tag name and a security level. To manipulate a common switch, you need to call the common switch faceplate by typing the tag name. If a common switch faceplate is called by typing the element number of the common switch, the default security level 4 will be applied.
- In order to display the system alarm messages on Message Monitor window, the following settings need to be performed on the Message Registration dialog box of Message Monitor window.
  - [Details] check box: Check
  - [First]: BSTS
  - [Second]: MFCS
  - [Third]: *
  - [Color]: Designate a color
- If the MV is changed by INC/DEC buttons, the traces on system alarm messages and on the operation records are not identical.
- There is no user notification for manipulating a system common switch. If required, a control application may be created to notify the user on manipulating a system common switch.

SEE ALSO

For more information about process report view, refer to:

5.1, "Process Report View" in Human Interface Stations Reference Vol.2 (IM 33K03F22-50E)

For more information about system tag names, refer to:

1.1, "Names" in Engineering Reference Vol.1 (IM 33K03G21-50E)

For more information about the Message Registration dialog box of Message Monitor window, refer to:

"■ Message Registration Dialog Box" in 3.7.2, "Message Registration" in Human Interface Stations Reference Vol.1 (IM 33K03F21-50E)

5. Alarm Processing – FCS

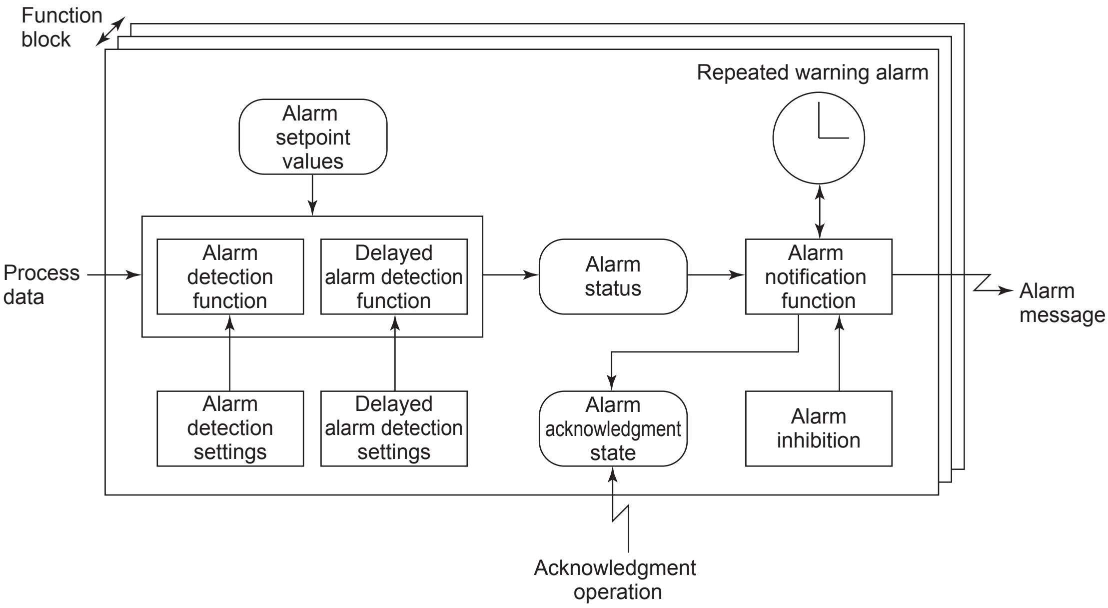
The FCS alarm processing includes an alarm detection function for detecting any abnormality in the process and an alarm notification function for giving notification of the detection results.

■ Functional Structure of the Alarm Processing

Alarm processing is a function that detects any abnormalities in the process from values such as the process variable (PV) and manipulated output value (MV), then reflects this in the function block alarm status while at the same time consolidating the detection results and giving notification of these to the operation and monitoring function as a message. Alarm processing is found in each function block.

Alarm processing consists mainly of the following two functions.

- “Alarm detection function,” which detects any abnormality in the process
- “Alarm notification function,” which notifies the operation and monitoring function of the detection result



Alarm setpoint values:	Individual data items relating to the alarm settings (PH, PL, etc.)
Alarm status:	Data item that indicates the status of the function blocks (ALRM)
Alarm acknowledgment state:	Data items that indicate the alarm flashing status (ALFS)

Figure 5-1 Function Structure of the Alarm Processing

The following functions act as auxiliary functions to the alarm function and alarm notification function.

- Alarm detection stop function
- Alarm inhibition function
- Alarm operation
- Delayed alarm detection function

■ Alarm Detection Functions

This is a function that detects any abnormality in the process from values such as process variables (PV) and manipulated output values (MV). In order to detect anomalies in the process, the alarm detection function performs the following alarm checks.

- Input open alarm check
- Input error alarm check
- Input high-high and low-low limit alarm check
- Input high and low limit alarm check
- Input velocity alarm check
- Deviation alarm check
- Output open alarm check
- Output failure alarm check
- Output high and low limit alarm check
- Connection failure alarm check

The alarm check can be executed among the detection functions varies by the function block.

■ Alarm Detection Functions of Certain Function Blocks

Certain function blocks perform a special alarm check that differs from other function blocks. These particular alarm checks are indicated below.

● Blending PI Controller Block (PI-BLEND)

- Cumulative deviation alarm check
- Control error alarm check

SEE ALSO

For more information about alarm check specific to PI-Blend block, refer to:

- "■ Cumulative Deviation Alarm Check" in 1.12, "Blending PI Controller Block (PI-BLEND)" in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)
- "■ Control Error Alarm Check" in 1.12, "Blending PI Controller Block (PI-BLEND)" in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)

● Flow/weight-Totalizing Batch Set Block (BSETU-2,BSETU-3)

- Pre-batch alarm check
- Batch end alarm check
- Cumulative deviation high and low limit alarm check
- Leak alarm check
- Missing pulse alarm check (BSETU-2 only)
- Flowrate alarm check (BSETU-2)
- Flowrate alarm check (BSETU-3)
- Priority order for alarm displays specific to the batch set block for flowrate measurement
- Priority order for alarm displays specific to the Weight-Totalizing Batch Set Block

SEE ALSO For more information about alarm check of BSETU-2 and BSETU-3 blocks, refer to:

1.20.4, “Alarm Processing of Totalizing Batch Set Blocks (BSETU-2, BSETU-3)” in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)

For more information about alarm check of BSETU-2 block, refer to:

- “■ Missing Pulse Alarm Check” in 1.21, “Flow-Totalizing Batch Set Block (BSETU-2)” in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)
- “■ Flowrate Alarm Check” in 1.21, “Flow-Totalizing Batch Set Block (BSETU-2)” in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)
- “■ Alarm Display Priority of the Flow-Totalizing Batch Set Block (BSETU-2)” in 1.21, “Flow-Totalizing Batch Set Block (BSETU-2)” in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)

For more information about alarm check of BSETU-3 block, refer to:

1.22.2, “Alarm Processing of Weight-Totalizing Batch Set Block (BSETU-3)” in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)

● Motor Control Block (MC-2, MC-2E, MC-3, and MC-3E)

- Feedback input high and low limit check
- Thermal trip alarm check
- Interlock alarm check
- Answerback inconsistency alarm
- Answerback error alarm

SEE ALSO For more information about alarm check of MC-2, MC-2E, MC-3, and MC-3E blocks, refer to:

1.17.4, “Alarm Processing of Motor Control Blocks (MC-2, MC-2E, MC-3, and MC-3E)” in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)

● Velocity Limiter Block (VELLIM)

- Deviation alarm check

SEE ALSO For more information about alarm check of VELLIM block, refer to:

“■ Deviation Alarm Check” in 1.23, “Velocity Limiter Block (VELLIM)” in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)

● Switch Instrument Block (SI-2, SIO-21, SIO-22, SIO-22P), Enhanced Switch Instrument Block (SI-2E, SIO-21E,SIO-22E,SIO-22PE)

- Answerback inconsistency alarm (Same function as the motor control operation block)
- Answerback error alarm (Same function as the motor control operation block)

SEE ALSO For more information about alarm check of Switch Instrument blocks, and Enhanced Switch Instrument Block, refer to:

- “■ Answerback Check” in 2.4,“Switch Instrument Block and Enhanced Switch Instrument Block” in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)
- “■ Actions of Answer-Back Inconsistency Alarm Check” in 2.4,“Switch Instrument Block and Enhanced Switch Instrument Block” in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)

● General-Purpose Calculation Block (CALCU, CALCU-C)

- Computation error alarm

SEE ALSO For more information about alarm check of CALCU and CALCU-C blocks, refer to:

"■ Computation Error Alarm Check" in 1.33, "General-Purpose Calculation Blocks (CALCU, CALCU-C)" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)

■ Alarm Notification Functions

This is a function that summarizes the detection made by the alarm detection function and reports the summary to the operation and monitoring function as a message. The types of messages reported to the operation and monitoring function are listed below.

- Process alarm messages
- System alarm messages

■ Alarm Detection Stop Function

This is a function that sets whether the alarm detection function for each process alarm is "Detect enabled" or "Detect disabled."

■ Alarm Inhibition Function

This is a function that temporarily inhibits the process alarm message operation with the alarm detection function still operative.

■ Alarm Operation

This is a function that enables the alarm settings to be specified by engineers or operators. The following categories can be set.

- Classification of the alarm operation based on alarm priority level
- Specification of the alarm processing level

■ Alarm Checks that are Possible for Each Function Block

The alarm checks that are possible differ for each function block.

SEE ALSO For more information about alarm check items of regulatory control blocks, refer to:

"■ Alarm Processing Possible for Each Regulatory Control" in 1.1.3, "Input Processing, Output Processing and Alarm Processing Possible for Each Regulatory Control Block" in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)

For more information about alarm check items of calculation blocks, refer to:

"■ Alarm Processing Possible in Each Calculation Block" in.1.3.1, "Input Processing, Output Processing and Alarm Processing Possible for Each Calculation Block" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)

For more information about alarm check items of sequence control blocks, refer to:

2.1.1, "Alarm Processing of Sequence Control Blocks" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)

■ Delayed Alarm Detection Function

You can use this function to reduce the alarms which are generated at the event of the momentary spike of process variable (PV) or the fluctuation of process variable (PV) around the alarm setpoint value.

This function is applicable to the following types of FCS, function blocks and alarms.

- Types of FCS
  FFCS-V
- Function blocks
  Input indicator blocks (PVI) and PID controller blocks (PID)
- Types of Alarm
  HH/LL, HI/LO, DV+/DV- and IOP/IOP-

SEE ALSO For more information about delayed alarm detection function, refer to:

5.17, “Delayed Process Alarm Detection: FFCS-V” on page 5-35

5.1 Input Open Alarm Check

The input open alarm check may generate an alarm to indicate that the input signal is in high limit or low limit input open alarming status (IOP, IOP-).

■ Operation of the Input Open Alarm Check

▼ Input Open Alarm

The input open alarm check is a function that determines whether the input values read from the field by the I/O module is out of the range of the high and low limit input open detection setpoint values.

The high-limit input open alarm (IOP) is initiated when it is determined that the input value exceeds the input open high detection setpoint value. Similarly, the low-limit input open alarm (IOP-) is initiated when the input value is below the low-limit input open detection set-point value.

The high and low limit input open alarm (IOP, IOP-) indicates that a failure such as severed wires in the detection terminal or transmitter has occurred.

The Input open alarm check is performed by the I/O module. The function blocks that are connected directly to the I/O module receives the check results from the I/O module as a data status, and the high and low limit input open alarm is activated or recovered. Even in the function blocks not directly connected to the I/O module, when the data for the cause of the high and low limit input open alarm is accessed, the high and low limit input open alarm is activated.

For a pair of redundant modules, the high and low limit input open alarm is initiated when a high and low limit input open alarm is detected from both modules. When the conditions for the alarm activation are not satisfied, the system recovers from the high and low limit input open alarm.

For the function blocks connected to an input/output module that is undergoing online maintenance, it is possible to specify whether to set the function blocks to input open (IOP) or not. This can be specified for each FCS. To set the function blocks to IOP, check the item “IOP Occurs in Connected Blocks” in the filed of “IOM Online Updating” on the Constant tab of the FCS property sheet. (*1)

*1: Can only be specified in FFCS series, KFCS2, KFCS, LFCS2, and LFCS. In FFCS series, KFCS2, and KFCS, the setting item “IOM Online Updating” specifies the action of the I/O module when the initial load setting item, among many other I/O module setting items, is changed online.

TIP

- In the Motor Control Blocks (MC-2, MC-2E, MC-3 and MC-3E), an input open alarm check is conducted for the feedback input.
- When the input terminal connection for a Logic Operation Block and General-Purpose Calculation Block is a sequence connection, no input open check is conducted. (*1)

*1: Logic Operation Block can be used in FCSs except PFCS.

SEE ALSO

For more information about FFCS series, KFCS2, and KFCS I/O Modules’ initial loading items, refer to:

“■ Operation of the I/O Module when Online Download to the I/O Module is Executed : FFCS series/ KFCS2/KFCS” in B3.5, “Operation of I/O Module when Downloading is Performed” in the Field Control Stations Reference (IM 33K03E10-50E)

■ The Operation of the Function Block during Alarm State Initiated by the Input Open Alarm Check

The behavior of the function block when the high and low limit input open alarm (IOP, IOP-) initiated by the Input open alarm check is described below.

- Analog input process such as square root extraction, pulse input conversion, digital filters, and totalizer functions are disabled.
- The value before alarms occurrence is latched as a process variable (PV). However, if the PV overshoot is defined, the process variables (PV) are overshot to the high or low limit of the PV range.
- For regulatory control blocks that have the MAN fallback function, the MAN fallback function is activated and the block mode is switched to manual (MAN) mode.

■ Settings for the Input Open Alarm Check

Input open alarm check types and the high and low limit input open detection level can be set.

● Input Open Alarm Check Types

The setting of the input open alarm check type can be defined in the “input open alarm” on the Function Block Detail Builder. The types of input open alarm checks are listed below. The default setting is “both input open alarms enabled.”

- Both input open alarms enabled
- High limit open alarms enabled
- Low limit input open alarms enabled
- Input open alarms disabled

● High and Low Limit Input Open Detection Level

The high and low limit input open detection level can be defined in the IOM Builder Detail Setting.

High-limit input open detection level:	The value shall be within the range between -1000.0 to 1000.0 % The default setting is 106.3 %
Low-limit input open detection level:	The value shall be within the range between -1000.0 to 1000.0 % The default setting is -6.3 %

■ Inhibit IOP Reactions : FFCS Series/KFCS2/LFCS2

▼ **Inhibit IOP Reactions**

When the Regulatory Controller Block encounters an input open alarm (IOP, IOP-), the block reactions upon IOP can be inhibited so as the block can ignore the IOP alarm and continues the current control actions without changing the data status into BAD or performing MAN Fallback. If the IOP alarm reactions are inhibited, the block will behave as follows upon IOP:

- The data status of process variable (PV) will be kept as normal (NR) not indicated as a bad value (BAD). However, this inhibition only valid for IOP and IOP- alarms, the data status will become bad by other abnormalities.
- The block will continue its control actions without performing MAN Fallback.
- The process variable (PV) will become the value in accordance with the setting of [PV Overshoot].
- If the process variable (PV) is in calibration status (CAL), the PV will keep the value set from HIS.
- Input processing and digital filter will not function.
- Control calculation and totalization will continue.

- Input velocity check will continue
- According to the Input Open Alarm setting, input open alarm as well as other process alarms will be initiated.

The inhibition of the IOP reactions can be set on the Function Block Detail Builder.

Inhibit IOP Reactions: Choose [Valid] or [Invalid].
The default is [Invalid].

The following function blocks can be set to inhibit the IOP alarm reactions:

PVI, PVI-DV, PID, PI-HLD, ONOFF-E, ONOFF-GE, PID-STC, RATIO

Moreover, this inhibition is valid only for the input open alarm (IOP, IOP-) of the process variable (PV), not other signals. For example, if the compensation input (VN) of a PID controller becomes open (IOP), the PID controller will perform MAN fallback since this inhibition is not valid to the IOP of VN.

5.2 Input Error Alarm Check

The input error alarm check determines whether the data status of the input value is invalid (BAD). When the data is invalid (BAD), the high-limit input open alarm (IOP) is activated.

■ The Actions of Input Error Alarm Check

The input error alarm check is a function that determines whether the data status of the input value is invalid (BAD). When it is determined that the data status of the input value is invalid (BAD), the high-limit input open alarm (IOP) is activated. The system recovers from the alarming state when the data status value is no longer invalid (BAD).

The possible causes of the invalid (BAD) data status of the input value are listed below.

- Input open detected
- I/O module failure
- Block mode of the block for data reference is disabled (O/S)
- Data status of the data for data reference is invalid (BAD)
- Data status of the input value fails to communicate (NCOM)

However, when the cause of the invalidity (BAD) data status is low-limit input open, the low-limit input open alarm (IOP-) is activated and the high-limit input open alarm (IOP) is not activated.

TIP

In the motor control blocks (MC-2, MC-2E, MC-3, and MC-3E), an input error alarm check is conducted for the feedback input and answer-back input.

■ The Operation of the Function Block During Alarm State Initiated by the Input Error Alarm Check

The actions of the function block during input open high alarm (IOP) state initiated by the Input error alarm check are described below.

- Analog input process such as square root extraction, pulse input conversion, digital filters, and totalizer functions are disabled.
- Process variables (PV) are latched at the value before the alarm occurred. However, when the PV overshoot is defined, the process variables (PV) are overshot to the high or low limit of the PV range.
- For regulatory control blocks that have the MAN fallback function, the MAN fallback function is activated and the block mode is switched to manual (MAN) mode.

■ Settings for the Input Error Alarm Check

The Input error alarm check operates when both the input open alarms enabled or the input open high alarm enabled on the input open alarm check is defined. If neither are defined, input error alarm check will not function. In this case, even if the data status is invalid (BAD), the alarm will not be activated.

5.3 Input High-High and Low-Low Limit Alarm Check

The input high-high limit and low-low limit alarm check may generate an alarm to indicate that the input signal is in high-high and low-low alarming status (HH, LL).

■ The Operation of the Input High-High and Low-low Limit Alarm Check

▼ PV High-High/Low-Low Limit Alarm

The input high-high and low-low limit alarm check is a function that determines whether the input process variable (PV) is out of the range of the high-high and low-low limit alarm set-point value (HH, LL).

When it is determined that the input process variable (PV) exceeds the high-high limit alarm setpoint value, the high-high limit alarm (HH) is activated. Similarly, when the process variable is below the low-low limit alarm setpoint value (LL), the low-low limit alarm (LL) is activated.

When in alarming state, if the process variable (PV) becomes smaller than the value obtained by subtracting the alarm hysteresis value (HYS) from the high-high limit alarm setpoint value (HH), the system recovers from the high-high limit alarm. Similarly, if the process variable (PV) becomes greater than the value obtained by adding the alarm hysteresis value (HYS) to the low-low limit alarm setpoint value (LL), the system recovers from the low-low limit alarm.

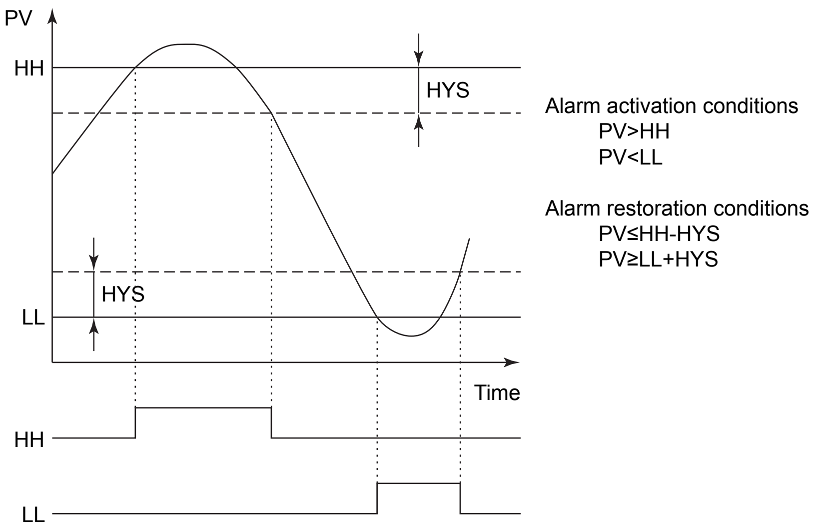


Figure 5.3-1 Input High-High and Low-Low Limit Alarm Check

■ Settings for the Input High-High Limit and Low-Low Limit Alarm Checks

The input high-high and low-low limit alarm check types, the high-high and low-low limit alarm setpoint values (HH, LL), and the alarm hysteresis values (HYS) can be defined.

● The Input High-High and Low-Low Limit Alarm Check Types

The input high-high and low-low limit alarm check type is defined on the “input high-high and low-low limit alarm” item in the Function Block Detail Builder. The types of the high-high and low-low limit alarm checks are listed below. The default is to check both the high-high and low-low limit alarms.

- Both high-high and low-low limit alarm
- High-high limit only

- Low-low limit only
- No alarm

However, with the Flow-Totalizing Batch Set Block (BSETU-2) and weight-Totalizing Batch Set Block (BSETU-3), only “enabled” or “disabled” can be set for the “input low-low limit alarm check.” The default in this case is “enabled.”

● High-High / Low-Low Limit Alarm Setpoint Value (HH, LL)

Settings of the high-high limit alarm setpoint values (HH) and the low-low limit alarm setpoint values (LL) are executed in the HIS.

High-high limit alarm setpoint values (HH):	Engineering unit data within PV scale Range The default setting is PV scale high limit value
Low-low limit alarm setpoint values (LL):	Engineering unit data within PV scale Range The default setting is PV scale low limit value

If the high-high limit alarm setpoint value (HH) is set to the same value as the PV scale high limit value, the high-high limit alarm will not be activated. Similarly, if the same value as the PV scale low limit value is set to the low-low limit alarm setpoint value (LL), the low-low limit alarm will not be activated.

● Alarm Hysteresis Value (HYS)

The alarm hysteresis value is define on each function block using the Function Block Detail Builder.

- Hysteresis
  Engineering unit data within the range of 0 to PV scale span, or percentage data for the PV scale span

  When specifying percentage data, add % after the numeric value.

  The default is 2.0 %.

This alarm hysteresis value is also used for the input high and low limit alarm check.

5.4 Input High and Low Limit Alarm Check

The input high limit and low limit alarm check may generate an alarm to indicate that the input signal is in high or low alarming status (HI, LO).

In the Motor Control Blocks (MC-2, MC-2E, MC-3, and MC-3E), it checks the high limit and low limit of the feedback input signal.

■ Input High and Low Limit Alarm Check

▼ PV High/Low Limit Alarm

The input high and low limit alarm check is a function that determines whether the process variable (PV) is within the range of the high and low limit alarm setpoint values (PH, PL).

When it is determined that the process variable (PV) exceeds the high limit setpoint value, the high limit alarm (HI) is activated. Similarly, when the process variable (PV) is below the low limit alarm setpoint value (PL), the low limit alarm (LO) is activated.

When in alarming state, if the process variable (PV) becomes smaller than the value obtained by subtracting the alarm hysteresis value (HYS) from the high limit alarm setpoint value (PH), the system recovers from the high limit alarming state. Similarly, if the process variable (PV) becomes greater than the value obtained by adding the alarm hysteresis value (HYS) to the low limit alarm setpoint value (PL), the system recovers from the low limit alarming state.

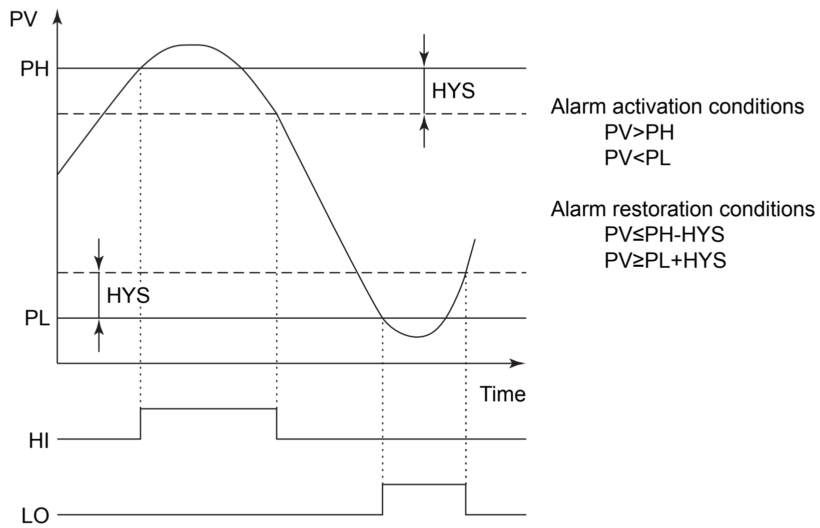


Figure 5.4-1 Input High and Low Limit Alarm Check

■ Settings for the Input High Limit and Low Limit Alarm Check

The types of input high and low limit alarm check, the high limit / low limit alarm setpoint values (PH, PL), and the alarm hysteresis values (HYS) can be set.

● Types of Input High and Low Limit Alarm Check

The type of the input high and low limit alarm check is defined on the “input high and low limit alarm” item in the Function Block Detail Builder. The types of input high and low limit alarm check are listed below. The default is set as “both high and low limit alarms.”

- Both high and low limit alarms
- High limit only
- Low limit only
- No alarm

● High Limit/Low Limit Alarm Setpoint Value (PH, PL)

Setting of the high limit alarm setpoint values (PH) and the low limit alarm setpoint values (PL) are performed on the HIS.

High limit alarm setpoint values (PH):	Engineering unit data within PV scale range The default setting is PV scale high limit value
Low limit alarm setpoint values (PL):	Engineering unit data within PV scale range The default setting is PV scale low limit value

If the same value as the PV scale high limit value is set to the high limit alarm setpoint value (PH), the input high limit alarm will not be activated. Similarly, if the same value as the PV scale low limit value is set to the low limit alarm setpoint value (PL), the input low limit alarm will not be activated.

● Alarm Hysteresis Value (HYS)

The alarm hysteresis value is defined on each function block using the Function Block Detail Builder.

- Hysteresis:
  Engineering unit data within the range of 0 to PV scale span, or percentage data for the PV scale span

  When specifying percentage data, add % after the numeric value.

  The default is 2.0 %.

This alarm hysteresis value is also used for the high-high limit/low-low limit alarm check.

5.5 Input Velocity Alarm Check

The input velocity alarm check may generate an alarm to indicate that the velocity in positive direction (VEL+) or velocity in negative direction (VEL-) is in alarming status.

■ Input Velocity Alarm Check

▼ Input Velocity Limit Alarm

The input velocity alarm check is a function that determines whether the changes in the process variable (PV) over a specified time exceed the velocity alarm setpoint value (VL).

When it is determined that the changes in the process variable (PV) over a specified time exceed the velocity alarm setpoint value (VL), the velocity alarm in the positive direction (VEL+) is activated if the changes are in the increasing direction. Similarly, if the changes are in the decreasing direction, the velocity alarm in the negative direction is activated (VEL-).

The input velocity alarm indicates abrupt changes in the process condition and abnormalities in the detectors or the transmitters.

The system will recover from the alarm status if the change in the process variable (PV) falls into the range in which the alarm hysteresis value (HYS) is subtracted from the velocity alarm setpoint value during the alarm status.

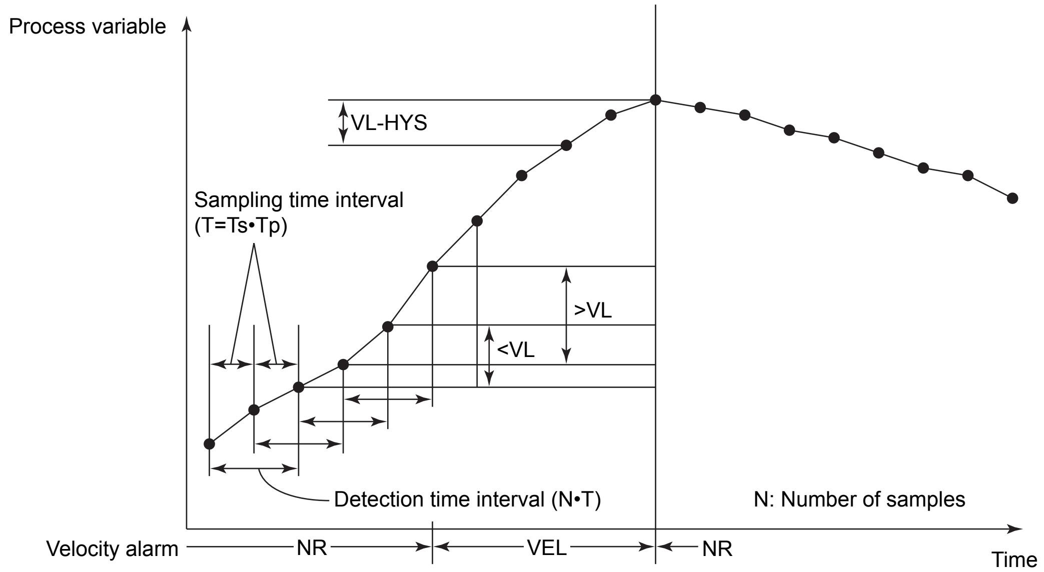


Figure 5.5-1 Input Velocity Alarm Check (When Sampling N=2)

The value of sampling time interval (T) is calculated using the set sampling interval (Tp) and the scan period of the function block (Ts).

```
T=Ts•Tp
```

Once the input velocity alarm is activated, the alarm will continue to ring for at least the sample time interval (T).

The most previously sampled process variable (PV) for input velocity alarm check in the buffer, i.e., the data sampled right before the Velocity – Reference Sample (PVP), may be monitored and referred to.

■ Settings for the Input Velocity Alarm

The types of input velocity alarm check, the velocity alarm setpoint values (VL), the alarm hysteresis value (HYS), the number of samples (N), and the sampling Intervals (Tp) can be defined.

● Types of Input Velocity Alarm Check

The type of input velocity alarm check can be set with the “input velocity alarm” item of the Function Block Detail Builder. The following types of input velocity alarm checks are available. The default is “detection of both directions.”

- Detection of both directions:
  Monitors velocity in both directions, positive and negative
- Detection of single direction:
  Monitors velocity only in positive direction or negative direction
- No Detection:
  Detection is not conducted

When single direction is selected for the input velocity alarm check type, only the velocity in the positive direction is monitored when the velocity alarm setpoint (VL) has a plus sign and only the velocity in the negative direction is monitored when the velocity alarm setpoint has a minus sign. However, when VL=0, the velocity of both directions is monitored regardless of the input velocity alarm check type.

Also, when “detection of both directions” is selected as the input velocity alarm check type, the absolute value of the velocity alarm setpoint (VL) is used for monitoring.

● Velocity Alarm Setpoint Value (VL)

The velocity alarm setpoint value (VL) can be set on the operation and monitoring function.

Velocity alarm setpoint value (VL):	Amount of change over the detection time interval (N•Tp•Ts) Engineering unit data within ±PV scale span The default setting is the PV scale Span

When single direction is set for the input velocity alarm check type, the sign (+ or -) of the direction to be detected is added to the engineering unit data of the velocity alarm setpoint (VL).

Also, when the same value (positive value) as the PV scale span is set to the velocity alarm setpoint (VL), neither a positive or negative direction velocity alarm occurs regardless of the input velocity alarm check type.

● Alarm Hysteresis Value (HYS)

Setting of alarm hysteresis conducted for each block using the Function Block Detail Builder.

- Hysteresis
  Engineering unit data within the range of 0 to PV scale span, or percentage data for the PV scale span

  When specifying percentage data, add % after the numeric value.

  The default is 2.0 %.

● Number Of Samplings (N) And Sampling Interval (Tp)

The sampling intervals (Tp) and the number of samplings (N) is defined in the Function Block Detail Builder.

Number of samplings (N):	1 to 12 points The default setting is 1 point
Sampling interval(Tp):	1 to 10,000 Unit is scan interval The default setting is 1

5.6 Deviation Alarm Check

The deviation alarm check may generate an alarm to indicate that the deviation in positive direction (DV+) or the deviation in negative direction (DV-) is in alarming status.

■ The Operation of the Deviation Alarm Check

▼ Deviation Alarm

The deviation alarm check is a function that determines whether the absolute value of the deviation (DV=PV-SV) between the process variable (PV) and the setpoint value (SV) exceeds the absolute value of the deviation alarm setpoint value (DL). When it is determined that the former exceeds the latter, a deviation alarm in the positive direction (DV+) is activated if the deviation is in the positive direction. Similarly, if the deviation is in the negative direction, a deviation alarm in negative direction (DV-) is activated.

During the alarm status, if the absolute value of the deviation (DV) falls into the range in which the alarm hysteresis value (HYS) is subtracted from the absolute value of the deviation alarm setpoint value (DL), the system will recover from the alarm status.

Further, when the same value (positive value) as the PV scale span is set to the deviation alarm setpoint (DL), neither a positive or negative direction deviation alarm occurs regardless of the deviation alarm check type.

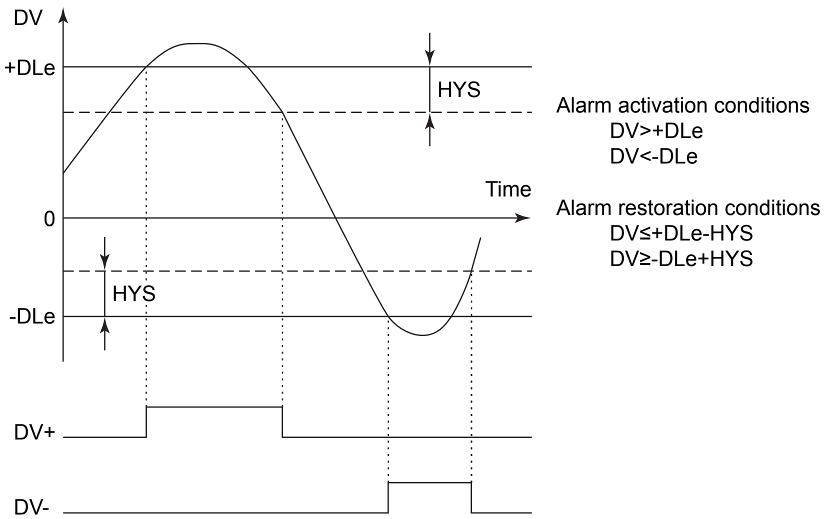


Figure 5.6-1 Action of Deviation Alarm Check

TIP

With the Dual-Redundant Signal Selector Block (SS-DUAL), the deviation alarm check is performed with respect to the deviation (DV=PV-SV) between the selected signal value (PV) and the non-selected signal value (SV).
In this case, if either the selected signal value (PV) or the non-selected signal value (SV) is a BAD value (BAD), the deviation alarm check is bypassed.

● Characterization of the Deviation Alarm Check

To prevent occurrence of the undesired alarm caused by abrupt set value change or set value ramp, the velocity change speed (derivative value) of the setpoint value (SV) is used as the deviation alarm setpoint value correction factor (r) to compensate the deviation alarm setpoint value (DL). The compensated deviation alarm setpoint value (DLe) is the value that adding the deviation alarm setpoint value correction factor (r) to the deviation alarm setpoint value (DL).

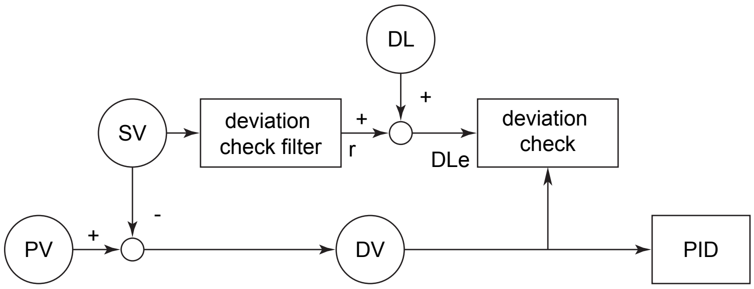


Figure 5.6-2 Characterization of Deviation Alarm Check

● The Operation of the Deviation Check Filter

The deviation check filter is a function that determines the deviation alarm setpoint value correction factor (r) by performing derivative calculations of setpoint values (SV).

The computational expression for the deviation check filter is shown below.

$$r(s)=\frac{K_{SV} \cdot T_{SV}s}{1+T_{SV}s} \cdot SV(s)$$

Figure 5.6-3 The computational expression for the deviation check filter

K_{SV} : deviation check filter gain. 0.000 to 10.000

T_{SV} : deviation check filter time constant. 0 to 10000 seconds

s : Laplacian

● Effects of the Deviation Check Filter

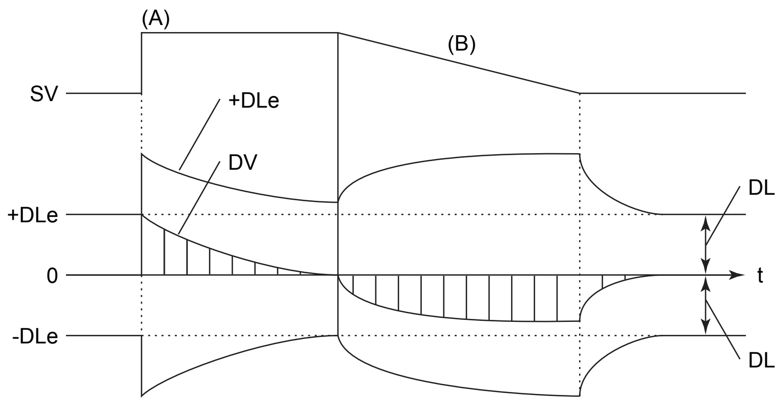


Figure 5.6-4 Effect of the Deviation Check Filter

The (A) and (B) shown in the above diagram assume the setpoint value operations below.

(A) Abruptly ramp to the setpoint value (SV)

(B) Gradually ramp to the setpoint value (SV)

When operation performed like this, the deviation (DV) temporarily increases to the points indicated as (A) and (B) in the diagram if the process variable is constant. The increase of the deviation is the result of the setpoint values' change performed by the operator who expected such a consequence. The deviation check filter has the effect to prevent the temporarily increased deviation (DV) caused by setpoint value changes from being activated. In the diagram, it shows how the corrected deviation alarm setpoint values (+/- DLe) changes according to the changes in the setpoint value (SV) under the conditions such as (A) and (B). If the deviation check filter is not provided, the deviation alarm always occurs whenever the deviation (DV) exceeds the limit of the deviation alarm setpoint value (+/- DLe), because the deviation alarm setpoint value (DLe) is the same value of the deviation alarm setpoint value (DL).

■ Settings for the Deviation Alarm Check

The types of the deviation alarm check, the deviation alarm setpoint values (DL), the alarm hysteresis value (HYS), and the deviation check filter can be set.

● Types of Deviation Alarm Check

The type of deviation alarm check can be defined on the “deviation alarm check” item of the Function Block Detail Builder. The types of deviation alarm checks are listed below. The default is “detect both directions.”

- Detection of both directions
  Monitors deviation in both directions
- Detection of single direction
  Monitors deviation in only one direction, positive or negative
- No detection
  Detection is not conducted

When single direction is selected for detection, if the deviation alarm setpoint value is plus symbol, only the deviation in the positive direction is detected and if the deviation alarm setpoint value is negative symbol, only the deviation in the negative direction is detected.

● Deviation Alarm Setpoint Value (DL)

The deviation alarm setpoint value (DL) is set by the operation and monitoring function.

Deviation alarm setpoint value (DL): Engineering unit data within ±PV scale span
Default is the ±PV scale span

When only one direction is selected for detection, the direction that is to be detected (+ or -) must be added to the engineering unit data of the velocity alarm setpoint value (DL).

● Alarm Hysteresis Value

The alarm hysteresis value is defined on each function block in the Function Block Detail Builder.

- Hysteresis
  Engineering unit data within the range of 0 to PV scale span, or percentage data for the PV scale span

  When specifying percentage data, add % after the numeric value.

  The default is 1.0 %.

● Deviation Check Filter

The deviation check filter gain and time constant can be defined in the Function Block Detail Builder.

- DV check filter gain: Deviation check filter gain
  Set a maximum 6-digit numeric value including a decimal point.

  0.000 to 10.000

  Default is 0.000
- DV check filter time constant: Deviation check filter time constant
  0 to 10,000 seconds

  Default is 0.

For Input Indicator Blocks with Deviation Alarm (PVI-DV), if scan coefficient is specified as 2 or greater on the Function Block Detail Builder, the deviation check filter should be multiplied by the specified scan coefficient.

TIP

The Dual-Redundant Signal Selector Block (SS-DUAL) and the velocity limiter block (VELLIM) do not have the deviation check filter function.

SEE ALSO

For more information about the deviation check filter gain and deviation check filter time constant, refer to:

“● The Operation of the Deviation Check Filter” on page 5-17

5.7 Output Open Alarm Check

The output open alarm check may generate an alarm to indicate that the output is open (OOP).

■ The Operation of the Output Open Alarm Check

▼ Output Open Alarm

The output open alarm check is a function that determines the data status received from the I/O module.

Based on the data status (OOP) received from the I/O module, an output open alarm (OOP) is activated.

This alarm indicates that the control output line has been physically severed.

The output open alarm check is performed by the I/O module. The function block receives the check results from the I/O module as a data status (OOP) and processes the activation or the restoration of the output open alarm. The output open alarm is activated only in the function block that is directly connected to the I/O module. The alarm will not be activated at function blocks that sends output through data connections to other function blocks.

When I/O modules are duplicated, an output open alarm is activated if both I/O modules are failed.

When the output signal is the pulse width output type and if data status of the tracking signal input terminal (TIN) becomes “PIO Not Ready” (NRDY), the block mode remains unchanged, the value immediately before the “PIO Not Ready” (NRDY) is retained as the manipulated output value (MV), and an output open alarm (OOP) is activated. If the block mode is tracking mode (TRK) and if the data status of the tracking signal input terminal (TIN) or the tracking switch input terminal (TSI) becomes “BAD value” (BAD), the block mode remains unchanged, the value immediately before the “BAD value” (BAD) is retained as the manipulated output value (MV), and an output open alarm (OOP) is activated.

If the output signal is not the pulse width output type and the block mode is tracking mode (TRK) and if the data status of the tracking signal input terminal (TIN) or the tracking switch input terminal (TSI) becomes “BAD value” (BAD), the block mode is changed to Initialization manual mode (IMAN).

■ The Function Block Reaction when Output Open Detected by Output Open Check Function

When an alarm output open alarm occurs, detected by output open alarm check function, the function blocks react as follows:

- For the regulatory control blocks that have MAN fallback functions, the MAN fallback action is initiated and the block mode is changed to manual (MAN) mode.
- The manipulated output value (MV) is frozen, and the current value is kept as manipulated output values (MV).

The reaction is the same as described above when the output fail alarm activates.

■ Settings for the Output Open Alarm Check

Setting of the output open alarm check is performed using the Function Block Detail Builder.

Output open alarm check: Choose from “Enabled” or “Disabled”
Default is “Enabled.”

5.8 Output Fail Alarm Check

The output fail alarm check may generate an alarm to indicate that the output is open (OOP).

■ The Operation of the Output Fail Alarm Check

The output fail alarm check is a function that determines whether the data status of the manipulated output value (MV) is output fail (PTPF).

When it is determined that the data status of the manipulated output value (MV) is output fail (PTPF), an output open alarm (OOP) is activated.

The system recovers from the alarming state when the data status is no longer output fail (PTPF).

The possible causes of the data status of the manipulated output value (MV) being output fail (PTPF) are listed below.

- Output open detected
- I/O module failure
- The mode of the destination block that data is being set is out of service (O/S).

■ The Function Block Reaction when Output Fail Alarm Detected by Output Fail Alarm Check Function

When an alarm output fail alarm occurs, detected by output fail alarm check function, the function blocks react as follows:

- For the regulatory control blocks that have MAN fallback functions, the MAN fallback action is initiated and the block mode is changed to manual (MAN) mode.
- The manipulated output value (MV) is frozen, and the current value is kept as manipulated output values (MV).

The reaction is the same as described above when the output open alarm activates.

■ Settings for the Output Fail Alarm Check

Enable or disable the output fail alarm check depends on whether the output open alarm check is enabled or not.

When the output open alarm check is enabled, the output fail alarm check is automatically enabled.

Vice versa, if the output open alarm check is disabled, the output fail alarm check is disabled too. In this case output fail alarm will not be activated even when the data status becomes output fail (PTPF).

5.9 Output High and Low Limit Alarm Check

The output high limit and low limit alarm check may generate an alarm to indicate that the output is in high limit or low limit alarming status (MHI, MLO).

■ Operation of the Output High and Low Limit Alarm Check

▼ Output High/Low Limit Alarm

The output high and low limit alarm check is a function that determines whether the manipulated output value (MV) exceeds the range of the manipulated output variable high-limit/low-limit setpoint (MH, ML) for the output limiter.

When it is determined that the manipulated output value (MV) exceeds the manipulated output variable high limit setpoint (MH), an output high limit alarm (MHI) is activated. Similarly, when it is lower than the manipulated variable low limit setpoint (ML), an output low limit alarm (MLO) is activated.

The system will recover from the alarming state when manipulated output value (MV) becomes lower than the value in which the alarm hysteresis value is subtracted from the manipulated variable high-limit setpoint. Similarly, the system recovers from the output low- limit alarm when the manipulated output value becomes greater than the value in which the alarm hysteresis value is added to the manipulated variable low-limit setpoint.

■ Settings for Output High and Low Limit Alarm Check

The types of output high and low limit alarm checks, the manipulated variable high limit/low limit setpoints (MH, ML), and the alarm hysteresis value (HYS) can be set.

● Types of Output High and Low Limit Alarm Checks

The types of output high and low limit alarm checks can be defined on the “output high and low limit alarm” item in the Function Block Detail Builder. The types of output high and low limit alarm checks are listed below. The default is set as “high and low limit alarms.”

- Both high and low limit alarms
  Output high and low limit alarm check enabled
- High-limit alarm only
  Output high-limit alarm check enabled
- Low-limit alarm only
  Output low-limit alarm check enabled
- Alarms disabled
  No alarm checking

● Manipulated Variable High-Limit / Low-Limit Setpoint (MH, ML)

The manipulated variable high-limit setpoint (MH) and the manipulated variable low-limit setpoint (ML) is set by the operation and monitoring function.

Manipulated variable high-limit setpoint (MH):	Engineering unit data within MV scale span Default is MV scale high-limit value
Manipulated variable low-limit setpoint (ML):	Engineering unit data within MV scale span Default is MV scale low-limit value

If the same value as the MV scale high-limit value is set to the manipulated variable high-limit setpoint (MH), the output high-limit alarm will not be activated. Similarly, if the same value as the MV scale low-limit value is set to the manipulated variable low-limit setpoint (ML), the output low-limit alarm will not be activated.

● Alarm Hysteresis Value (HYS)

Alarm hysteresis can be defined in each block in the Function Block Detail Builder.

- Hysteresis
  Engineering unit data within the range of 0 to PV scale span, or percentage data for the PV scale span

  When specifying percentage data, add % after the numeric value.

  Default is 2.0 %.

5.10 Bad Connection Status Alarm Check

Bad connection status alarm check may generate an alarm to indicate that the connection is bad (CNF).

■ Operation of the Bad Connection Alarm Check

▼ Bad Connection Alarm

The bad connection status alarm check is a function that determines whether there is a faulty connection to the function block or data at the I/O connection destination.

When it is determined that the connection status is bad, the bad connection alarm (CNF) is activated. When the connection is restored, the system recovers from the alarming state.

The bad connection is judged as follows.

- The connected destination function block is in out of service (O/S) mode.
- The connection information is abnormal, and the data reference or data set cannot be performed.
- The connected destination function block’s data type is invalid (cannot be convert to the appropriate data type).

■ Setting of the Bad Connection Alarm Check

The bad connection alarm check can be defined in the Function Block Detail Builder.

Bad connection alarm check: “Enabled” or “Disabled”
Default is “Enabled.”

5.11 Process Alarm Message

The process alarm messages are the process alarms that have been detected by the alarm detection functions and transmitted from the function block to the operation and monitoring function.

■ Process Alarm Messages

When a function block detected an abnormality of process variable (PV) or manipulated output value (MV) or vice versa, the function block will broadcast a message to notify the event. In the message text, the typical data item such as PV or MV will be appended. What alarms a function block detects and the alarm detection capabilities vary with the function block types.

For those alarm items that have been set to “AOF (Alarm Off),” when alarm occurs, the process alarm message is detected but activates no alarm message on the operation and monitoring function.

For those alarm items that have been as “no alarm detection,” when alarm occurs, it activates no alarm message on the operation and monitoring function because the function block does not detect the alarm.

SEE ALSO

For more information about each process alarm detection, refer to:

- 5.1, “Input Open Alarm Check” on page 5-6
- 5.2, “Input Error Alarm Check” on page 5-9
- 5.3, “Input High-High and Low-Low Limit Alarm Check” on page 5-10
- 5.4, “Input High and Low Limit Alarm Check” on page 5-12
- 5.5, “Input Velocity Alarm Check” on page 5-14
- 5.6, “Deviation Alarm Check” on page 5-16
- 5.7, “Output Open Alarm Check” on page 5-20
- 5.8, “Output Fail Alarm Check” on page 5-21
- 5.9, “Output High and Low Limit Alarm Check” on page 5-22
- 5.10, “Bad Connection Status Alarm Check” on page 5-24

5.12 System Alarm Message

The system alarm messages are the messages represent system abnormality occurred in the FCS and in the function blocks, those have been transmitted to the operation and monitoring function.

This section describes the messages related to the function blocks.

■ System Alarm Message

The system alarm messages are transmitted for abnormalities originating in the control station system, such as a malfunction in the I/O module or calculational errors caused in the user-defined computational expressions.

Regardless how the function block alarm detection defined, system alarm message will definitely broadcasts whenever it occurs or recovers.

The abnormalities that initiate the system alarm message are shown below.

● Abnormalities of the Input Modules

Abnormalities caused by the Input module detected by the Input error alarm check.

● Abnormalities of the Output Modules

Abnormalities caused by the output module detected by the output fail alarm check.

● Abnormalities in the User-Defined Computational Expression

Computational error caused in the user-defined computational expression in the general-purpose calculation block (CALCU).

● Abnormalities on One-Shot Initiation Exerted from the Sequence Control Block

Abnormalities occurred in a function block which terminal connected to a sequence control block such as sequence table, when the sequence block exerts a One-Shot initiation to the function block. This can be caused by any of the reasons below.

- The connected destination block is in out of service (O/S) mode.
- Restriction on the one-shot initiation from the nest.
  A function block that can be one-shot initiated is able to one-shot initiate another function block. When the chain one-shot initiation on multiple function blocks exceed the limit, an error occurs.

5.13 Deactivate Alarm Detection

The alarm detection function for each process alarm can be stopped in accordance with the alarm detection specification.

■ Alarm Detection

For each alarm detection function of the process alarms, the alarm detection status is changed by the function that sets “Detection enabled” or “Detection disabled”.

● Deactivate Alarm Detection

The alarms that are set as “Detection disabled,” the alarm detection function itself will be disabled.

When the alarm detection is being changed from “Detection enabled” to “Detection disabled”, if the process alarm is in alarm activating status, the alarm notification process is executed in the same manner as alarm recovery. Therefore, if there are no alarms in progress in the function block of the process alarm, that function block will be handled as a normal recovery (Alarm status: NR).

■ Alarm Detection Specification

▼ Do Not Allow Online Change of Alarm Detection (AF) Setting

The alarm detection settings of each function block can be defined on the Function Block Detail Builder. The setting item (AF) for detecting or not detecting the alarms can be switched by the command from the sequence table blocks or from the sequentially connected blocks.

However, online downloading the alarm detection setting item (AF) can also be restricted. On the properties sheet of a project, the following option for restricting the online downloading of alarm detection setting can be found:

- Do Not Allow Online Change of Alarm Detection (AF) Setting
  If this option is checked, online downloading the alarm detection will be restricted. When performing online maintenance, if the alarm detection settings are changed on the Function Block Detail Builder, during online downloading, the changes regarding the alarm detection setting (AF) will not be downloaded and a warning message will be displayed. By default, the option of “Do Not Allow Online Change of Alarm Detection (AF) Setting” is not checked, thus, the changes of alarm detection setting on the Function Block Detail Builder can be downloaded with no problem.

5.14 Alarm Inhibition (Alarm OFF)

The alarm inhibition can be used to temporarily inhibit the action of process alarm messages while the alarm detection is still active.

■ Alarm Inhibition (Alarm OFF)

The alarm inhibition is a function that temporarily inhibits the action of process alarm messages. During this time, the alarm detection is still active.

The alarm inhibition is abbreviated to alarm OFF (AOF).

The following table indicates the alarm action during alarm inhibition and when it is not inhibited, each alarms priority level.

Table 5.14-1 Standard Specification of Alarm Actions

Alarm priority	Normal alarm action		Alarm action when inhibited	
	Alarm display flashing action	Repeated warning alarm(*1)	Alarm display flashing action	Repeated warning alarm (*1)
High-priority	Locked (*2)	x	Self acknowledge	-
Medium-priority	Locked (*2)	-	Self acknowledge	-
Low-priority	Non-locked (*2)	-	Self acknowledge	-
Logging	Self acknowledge	-	Self acknowledge	-
Reference	Self acknowledge	-	Self acknowledge	-

*1: X: For repeated warning alarm
-: Not for repeated warning alarm
*2: Can be changed on Alarm Priority Builder.

When the alarm shifts from the normal mode to AOF mode, the active alarms will keep previous behaviors. For example, the alarm flashing state and the repeated-warning alarm actions will be remained. When the alarm status changes (alarm occurs or recovers in the same tag), the initial alarm inhibition actions will take place.

Since the alarm detection is conducted even in the alarm inhibition mode, the alarm status can be still referred from sequence control blocks as sequence condition. The conditions are functioning the same as in normal (not in AOF) mode.

SEE ALSO

For more information about alarm processing during alarm inhibition, refer to:

7.3.4, “Alarm Inhibition (AOF)” in the Human Interface Stations Reference Vol.2 (IM 33K03F22-50E)

For more information about alarm flashing, refer to:

5.15.1, “Alarm Display Flashing Actions” on page 5-31

■ Alarm Inhibition Setting

The alarm inhibition can be set for each function block or each annunciator message.

Alarms of a block can be inhibited individually or inhibited all together. All the process alarms, including the input open alarm (IOP) and the output open alarm (OOP), can be inhibited.

The alarm inhibition setting can be changed manually by the operator or changed by the sequence control blocks or the blocks in sequence connection. However, from an operation and monitoring console, the alarm cannot be inhibited individually, all the alarms of a block can only be inhibited all together.

■ Alarm Notification when Releasing Alarm Inhibition of Function Blocks

For all alarms in a function block, how they behave when releasing Alarm Inhibition can be specified. To notify the operation and monitoring console the existing alarms or not to notify the operation and monitoring console can be selected.

TIP

- The setting for notifying or not notifying the operation and monitoring console of the existing alarms only valid for releasing the AOF, it does not function when release alarm inhibition of each alarm items of a function block.
- For the alarms masked by Representative Alarm Block (ALM-R), when alarm mask is released by changing the SV or SW of the ALM-R block, the existing alarms will not initiate alarm message outputs.

The option of [Alarm Notify Action when All AOF Released] can be set on FCS properties sheet. The setting can be changed during offline and valid for the whole FCS.

- Alarm Notify Action when All AOF Released
  Check or uncheck the option. By default, this option is not checked.

■ Activate the Energized Annunciator when AOF Released

▼ Raised %AN sends out alarm when AOF released

An option needs to be checked or unchecked for whether to activate the energized annunciator as a new alarm on the operation and monitoring console when AOF of the annunciator is released.

The option of [Raised %AN sends out alarm when AOF released] can be set on FCS properties sheet. The setting can be changed during offline and valid for the whole FCS.

- Raised %AN sends out alarm when AOF released
  Check or uncheck the option. By default, this option is not checked.

5.15 Classification of Alarm Actions Based on Alarm Priority

There are five levels of alarm priority. The alarm action varies by the priority level.

■ Alarm Priority

There are five levels in alarm priority. They are high-priority alarms, medium-priority alarms, low-priority alarms, logging alarms and reference alarms.

Each level of alarm priority has its own specifications regarding to the following items:

- Enable/Disable display on windows
- Enable/Disable printouts
- Enable/Disable logging into file

Regarding to FCS, each level of alarm priority has the following different specifications:

- Alarm Action (Alarm flashing actions when alarm occurs)
- Enable/Disable repeated warning alarm
- Alarm Action when Returning to Normal (Alarm flashing when returns to normal)

The alarm actions corresponding to each level of alarm priority when alarm inhibition is deactivated are list in the following table.

Table 5.15-1 Alarm Actions and Levels of Alarm Priority

Alarm priority	Alarm display flashing	Repeated warning alarm	When returns to normal
High-priority	Locked (*1)	Yes (*1)	Depends on alarm type (*1)
Medium-priority	Locked (*1)	No	No action
Low-priority	Non-locked (*1)	No	No action
Logging	Self acknowledge	No	No action
Reference	Self acknowledge	No	No action

*1: The action may be defined on Alarm Priority Builder.

IMPORTANT

When enabling or disabling the alarm repeat action, the FCS offline downloading is required.

SEE ALSO

For more information about the behaviour of operation and monitoring console when an alarm returns to normal, refer to:

7.2, “Alarm Priority” in the Human Interface Stations Reference Vol.2 (IM 33K03F22-50E)

For more information about alarm display flashing, action when returns to normal (Alarm flashing when returns to normal), refer to:

5.15.1, “Alarm Display Flashing Actions” on page 5-31

For more information about the repeated warning alarm, refer to:

5.15.2, “Repeated Warning Alarm” on page 5-33

5.15.1 Alarm Display Flashing Actions

The alarm display flashing action is a function that changes the alarm display color or flashing status in the operation and monitoring window when a change occurs with respect to the alarm.

■ Alarm Display Flashing Actions

When an alarm occurs or when the system recovers from the alarm status, and when operator acknowledges the occurrence or recovery, the alarm marks on the operation and monitoring window change flashing status or colors.

The alarm acknowledgment action performed by the operator is transmitted to the FCS. The FCS indicates the alarm acknowledged status (if the alarm has been acknowledged or not) by alarm flashing status (AFLS) for each function block. Therefore, from any operation and monitoring console in the same project can perform the alarm acknowledgment action and monitor the alarm acknowledgment status.

■ Alarm Display Flashing Action When Alarm Occurs

The alarm display object starts to flash when the alarming activates and stops flashing when the alarm is acknowledged. There are three types of flashing actions behaves differently when the alarm status recovers. They are lock type, non-lock type, and self-acknowledge type.

● Locked Type (LK : Locked)

Even if alarm has recovered, the alarm flashing will continue until it is acknowledged.

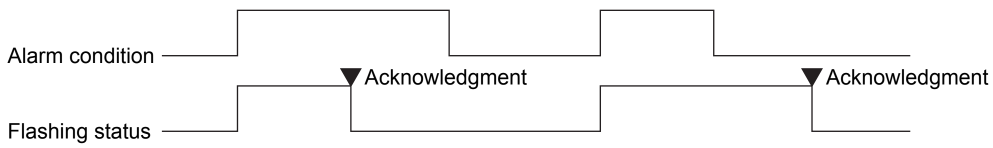


Figure 5.15.1-1 Locked Type Alarm Display Flashing Action

● Non-Locked Type (NL : Non-Locked)

When alarm status has recovered, the flashing stops regardless whether has been acknowledged or not.

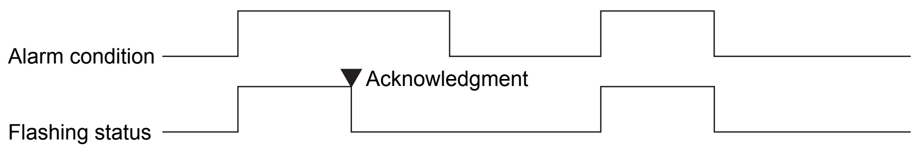


Figure 5.15.1-2 Non-Locked Type Alarm Display Flashing Action

● Self-Acknowledge Type (SA : Self-Acknowledge)

It is assumed that the alarm is acknowledged when it activates, so that the alarm display object does not flash.

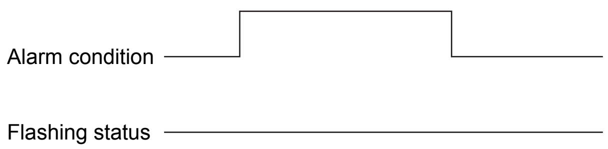


Figure 5.15.1-3 Self-Acknowledging Type Alarm Display Flashing Action

■ Alarm Action in Normal Status (Flashing Action in NR Status)

When a function block returns to normal status with all alarms vanished, the mark for normal (NR) will flash to notify the operator. The flashing action when normal status returns from alarm status contains two types, Alarm Dependent and Normal Notification. Only the alarms with high-priority can be specified with these options.

● Alarm Dependent (XL: eXception Lock)

When all alarms vanish and normal status returns, the alarm marks that have been flashing will continue to flash. The alarm marks that have stopped flashing will not flash.

● Normal Notification (NL: Non-Lock)

When all alarms vanish and the block returns to normal, the alarm mark starts to flash even though the alarming status flashing has been stopped. An acknowledge operation may stop the normal notification flashing. Without the acknowledge operation, the normal notification flashing will be kept when a new alarm occurs. When a function block returns to normal, it notifies it to operator by this Non-Lock type notification.

5.15.2 Repeated Warning Alarm

The repeated warning alarm retransmits a process alarm message to notify the operator that a critical alarm status is continuing.

■ Repeated Warning Alarm

The repeated warning alarm is a function that retransmits a process alarm message after a specified time has elapsed during the period between the alarm occurrence and recovery from the alarm, regardless of whether the alarm is acknowledged. The purpose of the repeated warning alarm is to notify the operator that a critical alarming state is continuing.

When multiple alarms originated from the same tag, the re-alarm requests are issued simultaneously to the same tag. In this case, only the one has the highest-priority among them can be reissued again.

Only the high-priority alarm can be reissued as repeated warning alarm. If the definition item of repeated warning alarm cycle is defined as zero second, repeated warning alarm does not effect.

■ Repeated Warning Alarm at Initial Cold Start

▼ Alarm Mask for Initial Cold Start

The repeated warning alarm has a function in which at initial cold start (include recovery from a long period of power failure) of the FCS, only the high-priority alarms are picked up from all the alarms and issued to the operation and monitoring consoles.

When the initial cold start for the FCS is executed, as a rule, only the alarms that were newly activated after start are transmitted to the operation and monitoring console by the alarm mask function. However, for the alarms that are set as repeated warning, if alarm is in process, the process alarm message is transmitted even though it was not newly activated.

However, if it is set as “alarm mask disabled” in the FCS Constants Builder, every alarm that is in progress will be classified as newly activated, and the process alarm messages will be issued at initial cold start.

■ Setting of Repeated Warning Alarm Period

▼ Interval of Repeated Warning Alarms

The period of the repeated warning alarm is defined for each control station in the FCS Constants Builder.

Repeated warning alarm period: The setting value is between 0 to 3600 seconds.
When 0 is set, repeated warning alarm function is disabled.

Default is 600 seconds

5.16 Alarm Processing Levels

With the alarm processing level, the alarm priority level can be specified for each function block or element.

■ Alarm Processing Levels

▼ Alarm Level

Designating an alarm processing level to a function block or an element, the alarms from the function block or the element will have the designated priority and display the designated color.

There are 16 levels, 1 to 16, for alarm processing. The alarm priority and alarm colors of all alarms occurred in a function block or an element are defined for each processing level.

The alarm priority defined for the first 4 levels of alarm processing are shown as follows.

- Level1
  All alarms initiated from the function block or the element are “High-priority alarms.”
- Level2
  All alarms initiated from the function block or the element are “Medium-priority alarms.”
- Level3
  All alarms initiated from the function block or the element are “Low-priority alarms.”
- Level4
  All alarms initiated from the function block or the element are “Logging alarms.”

The definitions for level1 to level4 are fixed for a whole system. The alarm priorities and colors for level5 to level16 can be defined by users.

The alarm processing levels for each function block can be defined on the Function Block Overview Builder or function block detail builder.

SEE ALSO For more information about the definitions of level5 to level16, refer to:

“■ Alarm Processing” in 7.4, “Alarm Status Character String and Alarm Processing” in the Human Interface Stations Reference Vol.2 (IM 33K03F22-50E)

5.17 Delayed Process Alarm Detection: FFCS-V

When alarm activation conditions or alarm restoration conditions are satisfied, this function provides a time delay for the state to be reflected into the alarm status and for an alarm message to be transmitted.

You can set the delay time for activation and recovery separately for each alarm.

IMPORTANT

When using the feature to delay the process alarm detection, you must carefully determine the appropriate delay time values for alarm detection and detection of recovery from alarms in your alarm design works for the plant.

Improper use of this function often fails to detect a true alarm because of the nature of this function.

■ Delayed Detection for Activation

Alarm activation is not detected immediately after the process variable (PV) goes beyond the range of alarm setpoint value; PV must stay out of this range for the time period specified by the user before the alarm activation is detected.

If PV goes back into the range of alarm setpoint value before the alarm activation is detected, then alarm activation is not detected.

■ Delayed Detection for Recovery

Recovery from an alarm is not detected immediately after the process variable (PV) goes back into the range of alarm setpoint value +/- hysteresis; PV must stay within this range for the time period specified by the user before the recovery is detected.

If PV goes beyond the range of alarm setpoint value +/- hysteresis again before the recovery is detected, then recovery is not detected.

■ The Effect of Using the Delayed Detection Function for Process Alarm

The following figures illustrate the effect of using the delayed process alarm detection function, showing a case when the process variable (PV) pulses and a case when the process variable (PV) fluctuates around the alarm setpoint value.

Example: The process variable pulses.

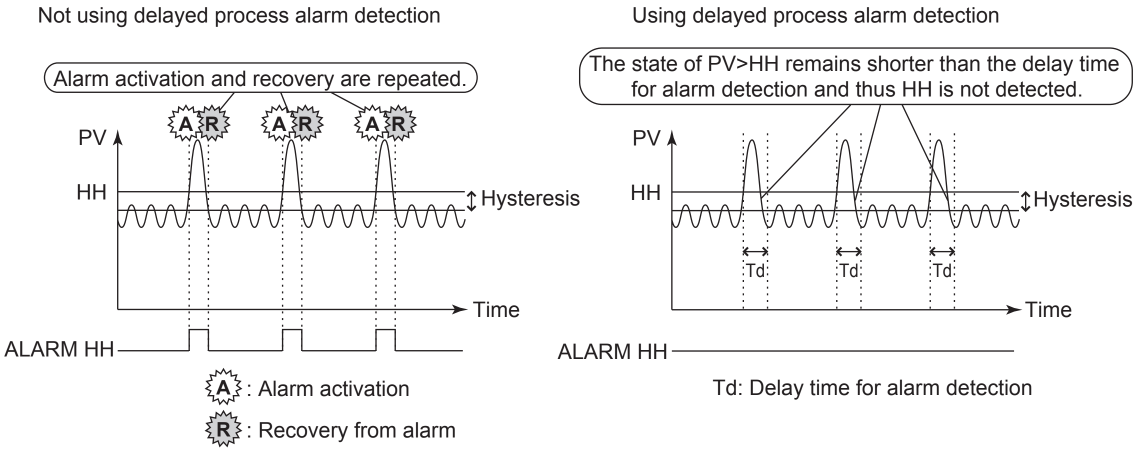


Figure 5.17-1 Effect of using the delayed detection function for process alarm 1: when process variable pulses

Example: The process variable fluctuates around the alarm setpoint value.

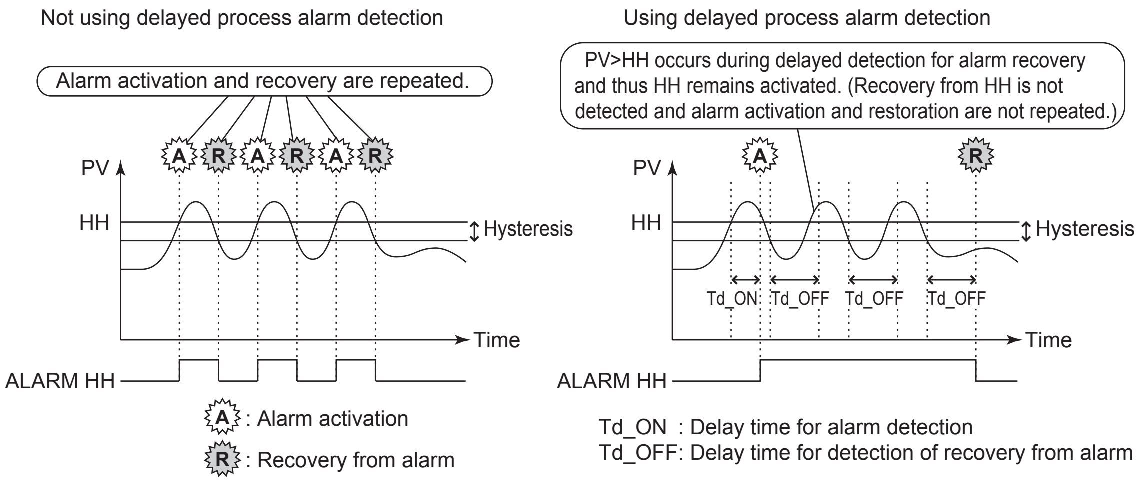


Figure 5.17-2 Effect of using the delayed detection function for process alarm 2: when the process variable fluctuates around the alarm setpoint value

■ Process Alarm Message Display

The time in a process alarm message is the time when the alarm activation or recovery is reflected into the alarm status.

The process variable (PV) in a process alarm message is the value when the alarm activation or recovery is reflected into the alarm status.

■ How to Set the Delay Time for Alarm Detection and the Delay Time for Detection of Recovery from Alarm

▼ Delay for alarm detection, Delay for detection of recovery from alarm

You can set the delay time to detect alarm activation and the delay time to detect recovery from alarm for indicator blocks (PVI) and PID controller blocks (PID). You can set the delay times on the Alarm tab of the Function Block Detail Builder. By default, the values are set to 0.

Table 5.17-1 Delay time settings for detection of alarm activation and recovery

Definition item	Setting range (unit: second)
Delay for HH alarm detection [sec.]	0 to 10

Definition item	Setting range (unit: second)
Delay for detection of recovery from HH alarm [sec.]	0 to 120
Delay for LL alarm detection [sec.]	0 to 10
Delay for detection of recovery from LL alarm [sec.]	0 to 120
Delay for HI alarm detection [sec.]	0 to 10
Delay for detection of recovery from HI alarm [sec.]	0 to 120
Delay for LO alarm detection [sec.]	0 to 10
Delay for detection of recovery from LO alarm [sec.]	0 to 120
Delay for DV+ alarm detection [sec.]	0 to 10
Delay for detection of recovery from DV+ alarm [sec.]	0 to 120
Delay for DV- alarm detection [sec.]	0 to 10
Delay for detection of recovery from DV- alarm [sec.]	0 to 120
Delay for IOP alarm detection [sec.]	0 to 10
Delay for detection of recovery from IOP alarm [sec.]	0 to 120

IMPORTANT

- When the delay time for alarm detection is set to the default value, the delayed detection will not function for activation.
- When the delay time for alarm recovery detection is set to the default value, the delayed detection will not function for recovery.

5.17.1 Delayed Detection for High-high Limit (HH), Low-low Limit (LL), High Limit (HI) and Low Limit (LO) Alarms

This section describes the delayed detection for activation of HH, LL, HI, and LO alarms and the delayed detection for recovery from HH, LL, HI, and LO alarms.

■ Delayed Detection for the Activation of HH, LL, HI or LO Alarm

When the process variable (PV) goes beyond the range of HH, LL, HI or LO alarm setpoint value, the delayed detection works for the activation of HH, LL, HI or LO alarm. If the value remains outside the range longer than the set delay time, the occurrence of HH, LL, HI or LO is reflected into the alarm status and a message for alarm activation is generated.

When the process variable (PV) falls within the range of alarm setpoint value before the set delay time elapses, the alarm status remains unchanged and a message for alarm activation is not generated.

TIP

The delay times are set using these builder definition items: Delay for HH alarm detection, Delay for LL alarm detection, Delay for HI alarm detection and Delay for LO alarm detection.

The following figure shows how the delay detection capability behaves for HH alarm activation.

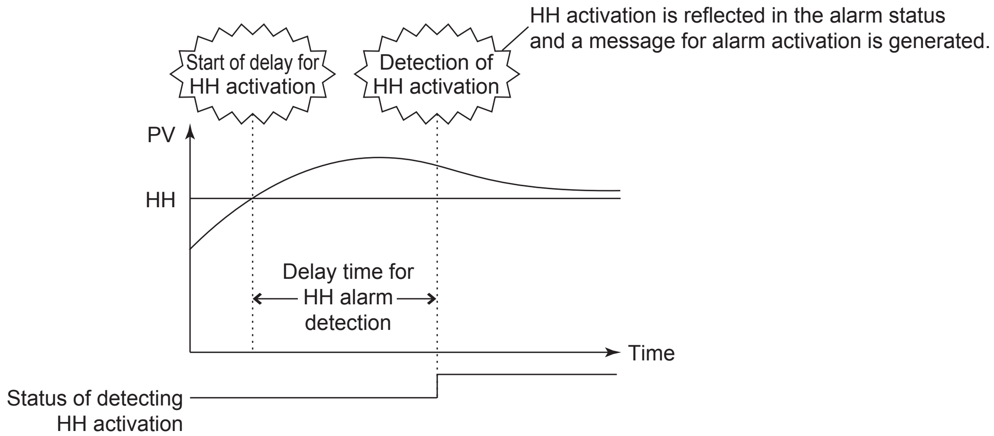


Figure 5.17.1-1 Behavior of delay delayed detection for HH alarm activation

TIP

If the time specified in Delay for HH alarm detection is shorter than the time specified in Delay for HI alarm detection, HH alarm may be detected first and then HI alarm is detected.

■ Delayed Detection for the Recovery from HH, LL, HI or LO Alarm

The delayed detection works for the recovery from HH or HI alarm when the process variable (PV) falls below "the alarm setpoint value - hysteresis."

If the PV remains below this level longer than set delay time, the recovery from, HH or HI is reflected into the alarm status and a recovery message is generated.

The delayed detection works for the recovery from LL or LO alarm when the process variable (PV) exceeds "the alarm setpoint value + hysteresis." If PV remains above this level longer than set delay time, the recovery from LL or LO is reflected into the alarm status and a recovery message is generated.

When the process variable (PV) goes outside the range of alarm setpoint value again before the set delay time elapses, the alarm status remains unchanged and a recovery message is not generated.

TIP

- Hysteresis is applied only to the starting of delaying the detection of recovery from alarm, and not applied to judging whether to continue the delay while detection of recovery is being delayed.
- The delay times are set using these builder definition items: Delay for detection of recovery from HH alarm, Delay for detection of recovery from LL alarm, Delay for detection of recovery from HI alarm, and Delay for detection of recovery from LO alarm.

The following figure shows how the delay detection capability behaves for recovery from HH alarm.

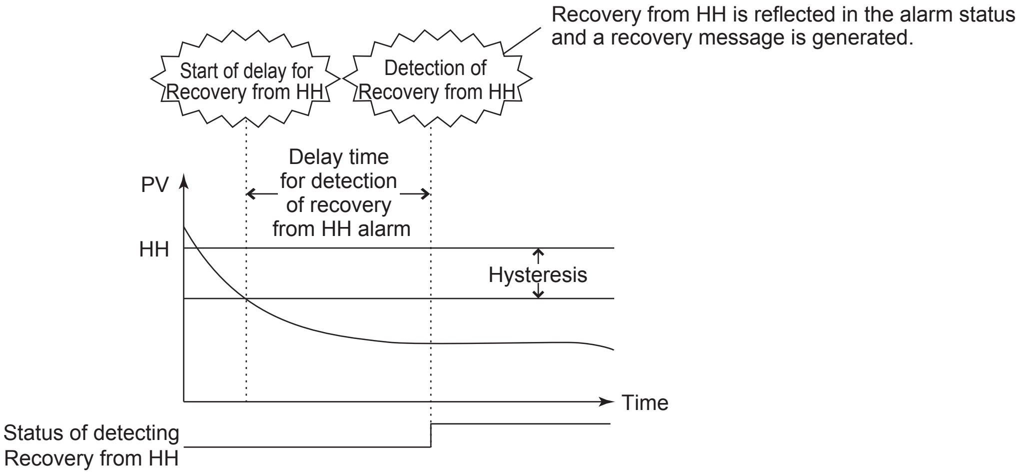


Figure 5.17.1-2 Behavior of delayed detection for recovery from HH alarm

5.17.2 Delayed Detection for Deviation (DV+/DV-) Alarm

This section describes the delayed detection for activation of deviation alarms and the delayed detection for recovery from deviation alarms.

■ Delayed Detection for the Activation of Deviation Alarm

When the absolute value of the deviation (DV) between the process variable (PV) and the setpoint value (SV) goes beyond the range of the deviation alarm setpoint (DL), the delayed detection works for the activation of deviation alarm. If the value remains beyond the range longer than the set delay time, the activation is reflected into the alarm status and a message for alarm activation is generated.

TIP

The delay times are set using the builder definition items Delay for DV+ alarm detection and Delay for DV- alarm detection.

■ Delayed Detection for the Recovery from Deviation Alarms

When the absolute value of the deviation (DV) between the process variable (PV) and the setpoint value (SV) falls within the range of "the deviation alarm setpoint +/- hysteresis", the delayed detection works for the recovery from deviation alarm. If the value remains within the range longer than the set delay time, the recovery is reflected into the alarm status and a recovery message is generated.

TIP

- Hysteresis is applied only to the starting of delaying the detection of recovery from alarm, and not applied to judging whether to continue the delay while detection of recovery is being delayed.
- The delay times are set using the builder definition items Delay for detection of recovery from DV+ alarm and Delay for detection of recovery from DV- alarm.

5.17.3 Delayed Detection for input Open (IOP) Alarm

This function is applicable to the input from IN terminals. The possible causes of the Input open (IOP) alarm are listed below:

- I/O module failure
- Disconnection
- Route error
- Data status of the data for data reference is invalid (BAD).

This section describes the delayed detection for activation of IOP alarm and the delayed detection for recovery from IOP alarm.

■ Delayed Detection for the Activation of Input Open (IOP) Alarm

This function works when the process variable (PV) goes outside the range of IOP detection level. If the value remains outside the range longer than the time set in Delay for IOP alarm detection, the data status of process variable (PV) turns to BAD, the occurrence of IOP is reflected into the alarm status and a message for alarm activation is generated.

The process variable (PV) moves according to the PV Overshoot setting.

If the PV goes back into the range of IOP detection level the period set in Delay for IOP alarm detection elapses, the data status of process variable (PV) and the alarm status remain unchanged. A message for alarm activation will not be generated.

During the period set in Delay for IOP alarm detection, the data status of process variable (PV) turns to QST and the process variable (PV) has the value immediately before it went outside the IOP detection level range. The function block that is referencing the process variable (PV) continues control calculation without performing MAN fallback.

● How the Delayed Detection for Activation Operates If Input Errors Concur on the IN Terminal and a Terminal Other Than the IN Terminal

This section provides example operations of the delayed detection for activation of IOP and input error alarm when input errors concur on the IN terminal and a terminal other than the IN terminal.

Example 1: Input error occurs on an input terminal other than the IN terminal first and then on the IN terminal.

- An IOP alarm is activated at the instant of input error on the input terminal other than the IN terminal. When input error occurs on the IN terminal later, this function works.
- If the input remains faulty longer than the time set in Delay for IOP alarm detection, the data status of process variable (PV) turns to BAD.
- The process variable (PV) moves according to the PV Overshoot setting.

Example 2: Input error occurs on the IN terminal first and then on an input terminal other than the IN terminal.

- This function works at the instant of input error on the IN terminal.
- If input error occurs on an input terminal other than the IN terminal while activation of IOP alarm is being delayed, an IOP alarm is activated at the instant of input error on the input terminal other than the IN terminal.
- If the input remains faulty longer than the time set in Delay for IOP alarm detection, the data status of process variable (PV) turns to BAD.
- The process variable (PV) moves according to the PV Overshoot setting.

■ Delayed Detection for Recovery from Input Open (IOP) Alarm

This function works when the process variable (PV) falls within the range of IOP detection level. If the value remains within the range longer than the period set inDelay for detection of recovery from IOP alarm, the BAD data status is canceled for process variable (PV), recovery from the IOP alarm is reflected into the alarm status and a recovery message is generated.

If the PV goes outside the IOP detection level range during the period set in Delay for detection of recovery from IOP alarm, the data status for process variable (PV) and the alarm status remain unchanged. A recovery message is not generated.

During the period set in Delay for detection of recovery from IOP alarm, the data status of process variable (PV) remains BAD. The process variable (PV) moves according to the PV Overshootsetting.

● How the Delayed Detection for Recovery Operates If Recovery of Input Concurs on the IN Terminal and a Terminal Other Than the IN Terminal

This section provides example operations of the delayed detection for recovery from IOP alarm and input error alarm when recovery of input concurs on the IN terminal and a terminal other than the IN terminal.

Example 1: Input on an input terminal other than the IN terminal returns to normal first and then on the IN terminal.

- The IOP alarm is not turned off immediately when the input terminal other than the IN terminal recovers from the input error.
- This function works when the IN terminal recovers from the input error.
- If the input on IN terminal remains good longer than the period set in Delay for detection of recovery from IOP alarm, the BAD data status is canceled for process variable (PV).

Example 2: Input on the IN terminal returns to normal first and then on an input terminal other than the IN terminal.

- This function works when the IN terminal recovers from the input error.
- When the input terminal other than the IN terminal recovers from the input error, detection of recovery from IOP is being delayed and thus the IOP alarm is not turned off.
- If the input on the IN terminal remains good longer than the period set in Delay for detection of recovery from IOP alarm, the BAD data status is canceled for process variable (PV).

5.17.4 Operations on a Change Related to Alarm Detection

This section describes how the delayed detection operates for process alarms when the following changes are made:

- Alarm setpoint value is changed.
- Alarm detection settings (AF) is changed.
- Calibration (CAL) is canceled.

■ Operations When an Alarm Setpoint Value Is Changed

When an alarm setpoint value is changed, the new setpoint value is used in conditional judgment for alarm activation or recovery and in the operation while detection of alarm activation/recovery is being delayed.

TIP

If you make the following changes and recovery from the relevant alarm occurs, detection of the recovery is not delayed. The recovery is immediately detected and a recovery message is generated.

- HH/LL and PH/PL are changed to SH/SL.
- DL is changed to SH-SL.

■ Operations When Alarm Detection Setting (AF) Is Changed

When the alarm detection settings (AF) are changed from "Detect disabled" to "Detect enabled", the delayed detection runs for activation as follows:

- Changed to "Detect enabled" when the alarm status is normal (NR);
  When the process variable (PV) goes outside the range of alarm setpoint value after AF is changed to "Detect enabled", the delayed detection runs for activation. If the value remains outside the range longer than the set delay time, the activation is reflected into the alarm status and an alarm message is generated.
- Changed to "Detect enabled" during the delayed detection for alarm activation;
  If the process variable (PV) remains outside the range for alarm setpoint value after AF is changed to "Detect enabled", the delayed detection works for activation immediately when AF is changed to "Detect enabled". If the value remains outside the range longer than the set delay time, the activation is reflected into the alarm status and an alarm message is generated.

■ Operations When Calibration (CAL) Is Canceled

If the conditions for alarm activation are satisfied when CAL is canceled, the delayed detection works for activation.

5.17.5 Delayed Process Alarm Detection in Unusual Operations

This section describes the behaviors of the delayed process alarm detection feature in the following unusual operations:

- Initial cold start
- Recovery from Out of Service (O/S) (including online maintenance)

■ Initial Cold Start

In initial cold start, the elapsed times of the delay times for alarm activation/recovery detection are cleared.

■ Recovery from Out of Service (O/S) (Including Online Maintenance)

If the block mode is set to O/S while the detection of alarm activation or recovery is being delayed, the elapsed time of the delay for alarm activation/recovery detection is cleared when the function block recovers from O/S mode.

The elapsed time of the delay times for alarm activation/recovery detection is cleared also when online maintenance is performed for the function block.

5.17.6 The Area Affected by Delayed Process Alarm Detection

This section describes how the delayed process alarm detection affects when repeated warning alarm or alarm inhibition (AOF) is specified.

■ How Alarms are Generated When Repeated Warning Alarm Is Specified

When repeated warning alarm is specified, a process alarm message is generated repeatedly with the set timing of repeated warning even if the delayed detection is working for recovery.

TIP

When detection of recovery is being delayed, the process variable (PV) has already returned to normal but the alarm status has not yet returned to normal. A recovery message is not generated, either. The reason for generating process alarms for repeated warning while the detection of recovery is being delayed is that the alarm state may recur while the detection of recovery is being delayed.

■ When Alarm Inhibition (AOF) Is Applied

Alarm inhibition (AOF) is not affected by the delayed process alarm detection feature.

Blank Page

6. Block Mode and Status

Block mode and status are the information that represent how a function block behaves under the current circumstances. Block mode and block status represent the operating state of the function blocks, while alarm status represents the alarm state of the process. Data status represents the reliability of the process data.

■ Block Mode

Block mode is the information that represents the control state and the output state of a function block. The different type of function blocks have different kinds of operation mode. In general there are 9 kinds of basic block mode, and some other block modes that are the composition of those basic block modes.

■ Block Status

A block status is the information that represents the operating state of a function block. Block status and block mode reflect the overall behavior of the function blocks. The different type of function block have different type of block status. Some function blocks do not have any block status.

■ Alarm Status

An alarm status is information that represents the alarm state of a process which was detected by the function block. The different type of function block have different type of alarm status. Some function blocks do not have any alarm status.

■ Data Status

A data status is the information that represents the reliability of data. For the data obtained from I/O module, their status pass from one function block to another. Data status is observed when various exceptional events occurred due to abnormality in the process or calculation.

■ Block Mode and Block Status of the Faceplate Block

The block mode and block status of the faceplate block are different from the block mode and block status of other function blocks.

SEE ALSO For more information about block mode and block status of the faceplate blocks, refer to:

3.3, “Block Mode and Status of Faceplate Blocks” in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)

■ Block Mode and Block Status of the SFC Block

The block mode and block status of the SFC block are different from the block mode and block status of other function blocks.

SEE ALSO For more information about block mode and block status of the SFC block, refer to:

1.6.13, “SFC Block Mode & Status” in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)

6.1 Block Mode

▼ **Block Mode**

Block mode is the information that represents the control state and the output state of a function block. The different type of function blocks have different kinds of operation mode. In general there are 9 kinds of basic block mode, and some other block modes that are the composition of those basic block modes.

■ Basic Block Mode

There are 9 kinds of basic block mode. These basic block modes are shared by all function blocks.

However, the basic block mode that can be applied to the particular function block varies from the types of the function block.

■ Compound Block Mode

Compound block mode refers to the state where multiple basic block modes are established simultaneously. In a compound block mode, the block mode with the highest priority among the basic block modes that are being simultaneously established will be executed.

Between any two block modes, there exists either a complementary relationship in which the two modes can be simultaneously established as a compound block mode, or a mutually exclusive relationship in which the two modes cannot be simultaneously established.

■ Transition of Block Mode

A transition of a block mode means that a specific block mode changes into a different block mode. A transition of block mode can be initiated by a mode change command from outside of the function block, or it may take place automatically when a mode transition condition is established inside.

■ Block Mode Transition Command

The block mode transition commands are the commands for switching the block mode performed outside of the function block, such as the operations performed by operator, or the action from the sequence control blocks and so on.

■ Block Mode Transition Condition

The block mode transition condition stand for the factors related to block mode change. When the condition for block mode change established, the block will automatically shift to the mode.

■ Block Mode of Function Blocks

The type of block mode applicable to the function blocks varies with type of block.

SEE ALSO

For more information about block mode of regulatory control blocks, refer to:

1.1.4, "Valid Block Modes for Each Regulatory Control Block" in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)

For more information about block mode of calculation blocks, refer to:

1.3.2, "Valid Block Modes for Each Calculation Block" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)

For more information about block mode of sequence control blocks, refer to:

2.1.2, "Block Mode of Sequence Control Blocks" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)

6.1.1 Basic Block Mode

There are 9 kinds of basic block mode. These basic block modes have common meanings when applied to all function blocks.

■ Basic Block Mode

The following table lists the basic block modes. The basic block mode that can be applied to the particular function block varies from the types of the function block.

Table 6.1.1-1 Basic Block Modes

Symbol	Name	Description
O/S	Out of Service	All functions of the function block are currently stopped.
IMAN	Initialization MANual	Calculation processing and output processing are currently stopped.
TRK	TRacKing	Calculation processing is currently stopped and the specified value is forced to be output.
MAN	MANual	Calculation processing is currently stopped and the manipulated output value, which is set manually, is output.
AUT	AUTomatic	Calculation processing is being executed and the calculation result is output.
CAS	CAScade	Calculation processing is being executed, the set value CSV is from the cascade connected upstream block, and the calculation result referred to this CSV is output.
PRD	PRimary Direct	Calculation processing is currently stopped, the set value CSV is from the cascade connected upstream block, this CSV is output directly.
RCAS	Remote CAScade	An control and calculation processing is being executed using the remote setpoint value (RSV) which is set remotely from a supervisory system computer, and the calculation results is output.
ROUT	Remote OUTput	Calculation processing is currently stopped, and the remote manipulated output value (RMV) which is set remotely from a supervisory system computer is output directly.

■ Relationships among Basic Block Modes

The initialization manual (IMAN) mode and the tracking (TRK) mode of the basic block modes cannot exist by themselves, and they represent certain operating states when they are established together with other basic block modes.

Any two of the modes such as automatic (AUT), manual (MAN), cascade (CAS), and primary direct (PRD) can not be established simultaneously. They have exclusive relationship and repel each other. The automatic (AUT) mode and tracking (TRK) mode can exist together. They have a complementary relationship.

The following figure shows the relationships among the basic block modes:

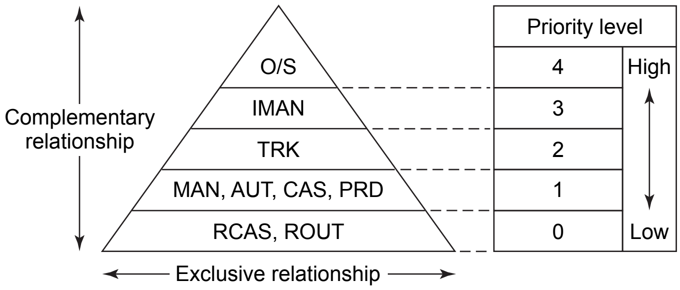


Figure 6.1.1-1 Relationships among Basic Block Modes

6.1.2 Compound Block Mode

The compound block mode refers to the state where the multiple basic block modes exist simultaneously.

■ Compound Block Mode

The state where multiple basic block modes are established simultaneously is called a compound block mode. The basic block modes that constituting a compound block mode have complementary relationships with each other so that they can exist together.

Among basic block modes, the initialization manual (IMAN) mode and the tracking (TRK) mode cannot exist by themselves. Only when they are combined with other basic block mode, they represent meaningful operating state. Therefore, the initialization manual (IMAN) mode or the tracking (TRK) mode is established only as part of a compound block mode.

Take the relationship between the tracking (TRK) mode and the automatic (AUT) mode for the PID controller block (PID) for example. When the tracking switch (TSW) is turned on when the operation in the automatic (AUT) mode, the operation of the tracking (TRK) is executed. When the tracking switch (TSW) is turned off, the operation of the original automatic (AUT) mode is executed. In this example, when the tracking switch (TSW) is on, the automatic (AUT) mode and the tracking (TRK) mode are established as the compound block mode.

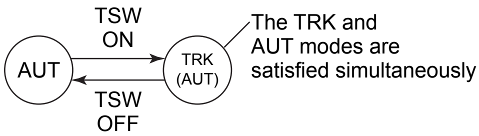


Figure 6.1.2-1 An Example of the Compound Block Mode

■ The Priority Levels of Block Mode

In a compound block mode, the basic modes that have complementary relationships each other are distinguished by the priority levels assigned to them. The basic block modes that have the same priority level but have exclusive relationship cannot exist together. On the other hand, the basic block modes that have different priority levels and complementary relationships each other can be established simultaneously.

In a compound block mode, the mode which has the highest priority level among all existing basic block modes is called the active mode of the compound block mode. In a compound block mode, the active mode takes effect.

■ Displaying Compound Block Mode on the Operation and Monitoring Function

Among all simultaneously established basic block modes, the block mode which has the highest priority level and the one which has the lowest priority level are displayed on the operation and monitoring function.

Example 1) AUT TRK

Example 2) ROUT AUT

■ References to Compound Block Mode Throughout This Document

In this chapter, a compound block mode is indicated by listing the simultaneously established basic block modes in a descending order starting from the highest priority level, and the block modes that have lower priority levels are in parentheses.

Example 1) TRK (AUT)

Example 2) IMAN (AUT (RCAS))

Combinations of Block Mode

The following figure shows the combinations of the basic block modes that constitute compound block mode.

Low ← Priority level → High

	0	1	2	3	4
Type 1		MAN, AUT, CAS			O/S
Type 2		MAN, AUT, CAS, PRD		IMAN	
Type 3		MAN, AUT, CAS, PRD	TRK		
Type 4		MAN, AUT, CAS, PRD	TRK	IMAN	
Type 5	RCS, ROUT			IMAN	
Type 6	RCS, ROUT		TRK		
Type 7	RCS, ROUT		TRK	IMAN	
Type 8	RCS, ROUT	MAN, AUT, CAS, PRD			
Type 9	RCS, ROUT	MAN, AUT, CAS		IMAN	
Type 10	RCS, ROUT	MAN, AUT, CAS	TRK		
Type 11	RCS, ROUT	MAN, AUT, CAS	TRK	IMAN	

Figure 6.1.2-2 Combinations of Block Modes

● When Transit from O/S to MAN, AUT or CAS Mode (Type 1)

Compound block mode that represents a course of transition state when transit a block mode from O/S to MAN, AUT or CAS mode (over transition mode). PRD mode can not be combined.

Example)Transit from O/S to MAN

Display on the operation and monitoring function: MAN O/S

Block mode: O/S (MAN)

SEE ALSO For more information about the over transition block mode, refer to:

"■ Over Transition Mode" on page 6-15

● When Operate in MAN, AUT, CAS or PRD Mode

Here is the explanation on the compound block modes established when a block mode transition condition is satisfied while the operation in MAN, AUT, CAS or PRD mode.

- When the Initialization Manual Condition is Satisfied (Type 2)
  Compound block modes established when the initialization manual condition is satisfied while operation in MAN, AUT, CAS or PRD mode.

Example) When the initialization manual condition is satisfied while operation in AUT mode

Display on the operation and monitoring function:	AUT IMAN
Block mode:	IMAN (AUT)

- When the Tracking Condition is Satisfied (Type 3)
  Compound block modes established when the tracking condition is satisfied while operation in MAN, AUT, CAS or PRD mode.

  Example) When the tracking condition is satisfied while operation in CAS mode:

Display on the operation and monitoring function:	CAS TRK
Block mode:	TRK (CAS)

- When the Initialization Manual Condition is Satisfied in Tracking Operation (Type 4)
  Compound block modes established when the initialization manual condition is satisfied while tracking is effect in the operation in MAN, AUT, CAS or PRD mode.

  Example) When the initialization manual condition is satisfied while tracking is effect in the operation in AUT mode:

Display on the operation and monitoring function:	AUT IMAN
Block mode:	IMAN (TRK (AUT))

SEE ALSO For more information about the mode transition conditions, refer to:

6.1.5, “Block Mode Transition Condition” on page 6-18

When Operation in RCAS or ROUT Mode

Here is the explanation of compound block modes established when a block mode transition condition is satisfied while operation in RCAS or ROUT mode.

- When the Initialization Manual Condition is Satisfied (Type 5)
  Compound block modes established when the initialization manual condition is satisfied while the operation in RCAS or ROUT mode.

  Example) When the initialization manual condition is satisfied while operation in RCAS mode:

Display on the operation and monitoring function:	RCAS IMAN
Block mode:	IMAN (RCAS)

- When the Tracking Condition is Satisfied (Type 6)
  Compound block modes established when the tracking condition is satisfied while operation in RCAS or ROUT mode.

  Example) When the tracking condition is satisfied while operation in ROUT mode:

Display on the operation and monitoring function:	ROUT TRK
Block mode:	TRK (ROUT)

- When the Initialization Manual Condition is Satisfied During the Tracking Operation (Type 7)
  Compound block modes established when the initialization manual condition is satisfied during the tracking operation in RCAS or ROUT mode.

  Example) When the initialization manual condition is satisfied during the tracking operation in RCAS mode:

Display on the operation and monitoring function: RCAS IMAN

Block mode: IMAN (TRK (RCAS))

- When Transit to a Computer Backup Mode (MAN, AUT or CAS) (Type 8)
When communication from the supervisory system fails while the operation is in RCAS or ROUT mode, transition to a compound block mode e.g. computer backup mode (MAN, AUT or CAS) takes place. A combination with the PRD mode [PRD (RCAS) and PRD (ROUT)] is a compound block mode that represents only an over transition state (over transition mode) during the block mode transition.

  Example) When transit from ROUT to a computer backup mode (MAN):

Display on the operation and monitoring function:	ROUT MAN
Block mode:	MAN (ROUT)

- When Transit from a Computer Backup Mode to RCAS or ROUT Mode (Type 8)
Compound block mode that represents only an over transition state during the block mode transition from a computer backup mode to the RCAS or ROUT mode (over transition mode). A combination with the PRD mode [PRD (RCAS) and PRD (ROUT)] is a compound block mode that represents an over transition state (over transition mode) during the mode transition.

  Example) When transiting from the ROUT mode to a computer backup mode (MAN):

Display on the operation and monitoring function:	ROUT MAN
Block mode:	MAN (ROUT)

SEE ALSO

For more information about block mode transition conditions, refer to:

6.1.5, “Block Mode Transition Condition” on page 6-18

For more information about the over transition mode, refer to:

“■ Over Transition Mode” on page 6-15

● During Computer Backup in RCAS or ROUT Mode

Here is the explanation of the combinations of block modes that are obtained when each block mode transition condition is satisfied during the computer backup in the RCAS or ROUT mode.

- When Initialization Manual Condition is Satisfied (Type 9)
Compound block modes that are obtained when the initialization manual condition is satisfied during computer backup operation (MAN, AUT or CAS mode) in RCAS or ROUT mode. PRD mode cannot be combined.

  Example) When the initialization manual condition is satisfied during the operation in the computer backup mode (AUT mode) and in the RCAS mode:

Display on the operation and monitoring function:	RCAS IMAN
Block mode:	IMAN (AUT (RCAS))

- When Tracking Condition is Satisfied (Type 10)
Compound block mode established when the tracking condition is satisfied while the computer backup operation (MAN, AUT or CAS mode) is in RCAS or ROUT mode. PRD mode cannot be combined.

  Example) When tracking condition is satisfied while the AUT backup operation in ROUT mode:

Display on the operation and monitoring function:	ROUT TRK
Block mode:	TRK (AUT (ROUT))

- When Initialization Manual Condition is Satisfied During Tracking Operation (Type 11)

Compound block mode established when the initialization manual condition is satisfied while the RCAS or ROUT mode and the computer backup mode (MAN, AUT or CAS mode) exist together and the tracking is in operation. PRD mode cannot be combined.

Example) When the initialization manual condition is satisfied during tracking operation in computer backup mode (AUT) and in RCAS or ROUT mode:

Display on the operation and monitoring function:	RCAS IMAN
Block mode:	IMAN (TRK (AUT (RCAS))

SEE ALSO For more information about the block mode transition conditions, refer to:

6.1.5, “Block Mode Transition Condition” on page 6-18

6.1.3 Block Mode Transition

Block mode transitions means to change a specific block mode to a different block mode. Block mode transition can be initiated by a block mode change command from other function block, or it may take place automatically when a block mode transition condition is satisfied inside.

When transit from a specific block mode to a different block mode, the state which both block modes exist temporarily is called transition mode.

■ Block Mode Transition

When change a specific block mode into a different block mode, the event is called Block Mode Transition. In principle, block mode transitions are triggered by the following two factors:

● Block Mode Change Commands

Block mode transition can be initiated by an external operation, such as an operation performed by the operator from the operation and monitoring function or block mode change command from the sequence control block. Operations from the outside of the function block are called block mode change commands.

● Block Mode Transition Conditions

Block mode transition can take place automatically corresponding to the condition inside of the function block itself. An example is that block mode changes when abnormality from connected I/O module is detected.

The conditions which initiates the function block transit its own mode are called block mode transition conditions.

■ Block Mode Transition Diagram

The following figure shows an overview of block mode transitions.

Each arrow in the figure denotes a block mode change command or a block mode transition condition.

Items designated with “command” such as in “MAN command” indicate transitions executed by a block mode change command. The other items indicate transitions executed by a block mode transition condition.

For example, the transition to the out-of-service (O/S) mode is executed by the O/S command. The MAN, AUT or CAS command executes transition from the out-of-service (O/S) mode to the respective block modes.

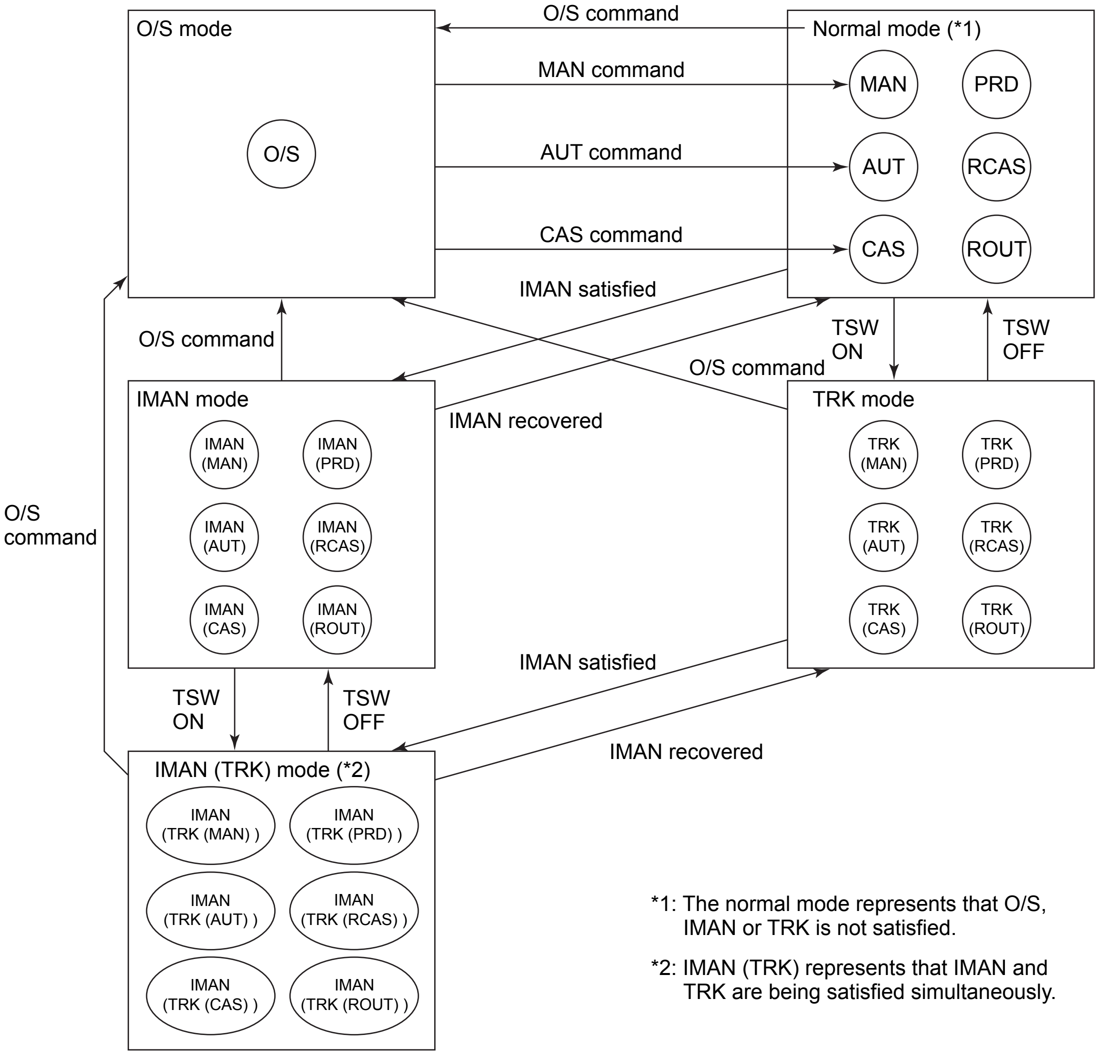


Figure 6.1.3-1 A Transition Diagram for Block Modes

SEE ALSO

For more information about block mode transitions in “normal mode,” “IMAN mode,” “TRK mode” and “IMAN (TRK) mode” shown in the above diagram, refer to:

“■ A Diagram for Basic Block Mode Transition in the Normal Mode” on page 6-12

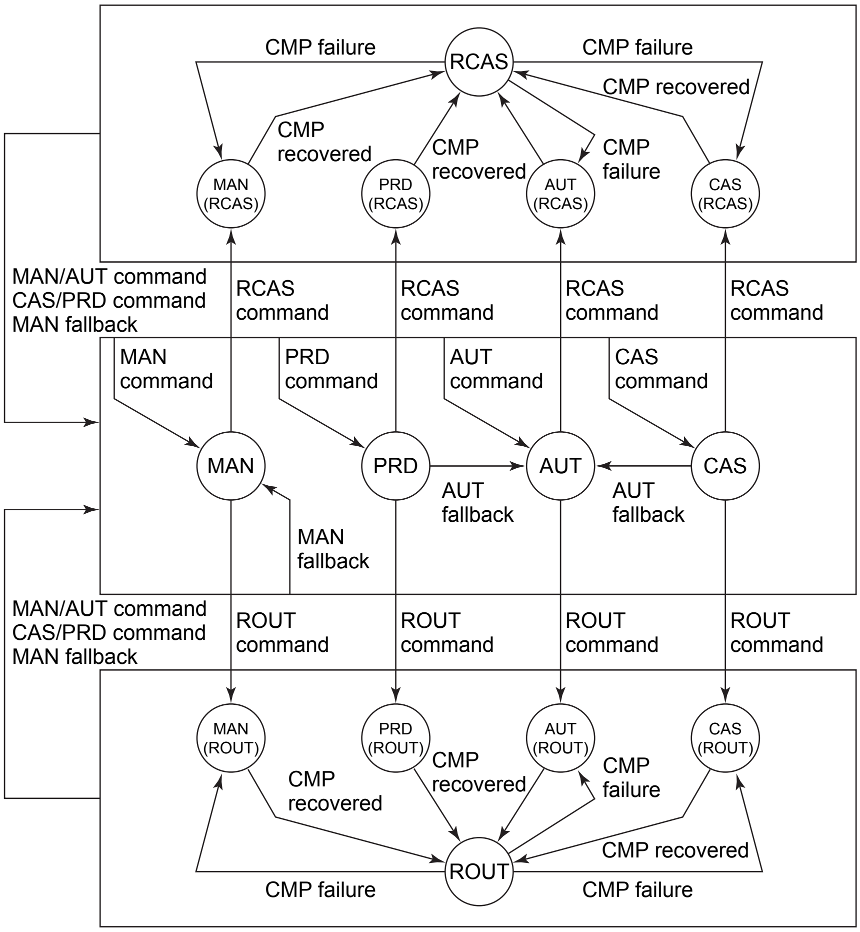
■ A Diagram for Basic Block Mode Transition in the Normal Mode

The following figure shows the details of block mode transitions for the segment indicated as “normal mode” in Block Mode Transition Diagram shown in the previous page.

The block modes transition is shown as follows:

- Priority level 0
  RCAS, ROUT
- Priority level 1
  MAN, AUT, CAS, PRD

Transition can take place from one block mode to the other if the two modes have the same priority level. For example, transition can take place from MAN to CAS.



CMP failure: A computer failure has occurred
CMP recovered: The computer failure has been recovered

Figure 6.1.3-2 A Transition Diagram for Block Modes in the Normal Mode

Even when the “IMAN mode,” “TRK mode” and “IMAN (TRK) mode” shown in “Block Mode Transition Diagram” in the previous page exist, transition of basic block mode whose priority level is 0 or 1 will be similar to the above figure. In these cases, IMAN or TRK, or both IMAN and TRK will exist in addition to the block modes shown in the above figure.

■ Examples of the Block Mode Transition

Normal mode transition and compound mode transition are show as follows:

1. In Automatic (AUT) mode which is a normal mode, when the tracking switch (TSW) is turned on, a compound mode consisting of the tracking (TRK) mode and the automatic (AUT) mode is established.
   “AUT TRK” is displayed as a block mode. The tracking (TRK) mode is the active mode.

2. When the connection with the cascade connected output destination opens, the initialization manual (IMAN) mode is established. A compound mode consisting of the initialization manual (IMAN) mode, tracking (TRK) mode, and automatic (AUT) mode are established together.
   “AUT IMAN” is displayed as the block mode. The initialization manual (IMAN) mode is the active mode.

3. When the tracking switch (TSW) is turned off, the tracking (TRK) mode no longer exists. The compound block mode only consists of the initialization manual (IMAN) mode and automatic (AUT) mode.
   “AUT IMAN” is displayed as a block mode. The initialization manual (IMAN) mode remains to be active mode.

4. When the connection with the cascade connected output destination recovers, the automatic (AUT) mode becomes the only block mode.
   "AUT" is displayed as a block mode. The automatic (AUT) mode becomes the active mode.

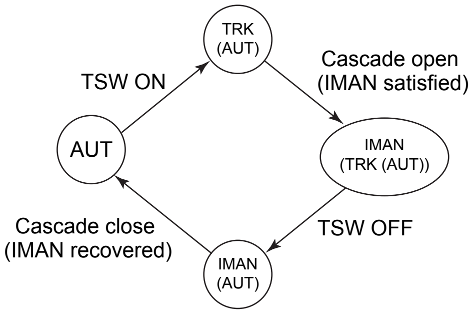


Figure 6.1.3-3 Examples of the Block Modes Transition

■ Rules for Block Mode Transitions

The block mode transition either by external mode change commands or by internal mode change conditions follows the same rules as explained below.

● Transition to an Exclusively Related Block Mode

When transit to a new block mode which has exclusive relationship with the current block mode, current block mode will be replaced by the new block mode.

The basic block modes that have exclusive relationship with each other:

- Among MAN, AUT CAS and PRD
- Between RCAS and ROUT

Example:

MAN
↓ AUT command
AUT

Under the same above condition, when transit from one inactive mode of a compound block mode to another inactive mode of the same compound block mode, the active mode is not affected, the function block's behavior remains unchanged too.

Example:

IMAN (AUT)
↓ MAN command
IMAN (MAN)

● Transition to a Complementary Block Mode

When transit to a new block mode which has complementary relationship with the current block mode, the new block mode will be added to the current block mode and a compound block mode is established. At this time, the block mode which has the highest priority level becomes the active mode.

Example:

AUT
↓ IMAN condition establish
IMAN (AUT)

The Invalidation of the Active Mode

When a new condition invalidates the current active mode, the block mode which has the next highest priority level becomes the new active mode.

Example:

IMAN (TRK (AUT))
↓ IMAN condition vanish
TRK (AUT)

Over Transition Mode

When a specific block mode transits to a different block mode, a state occurs in which both of the block modes are satisfied temporarily. This state is called an over transition mode.

An over transition occurs in the following mode transitions:

- A mode transition to the RCAS or ROUT mode when a computer failure has occurred.
- A mode transition from the O/S mode.

A Block Mode Transition to RCAS or ROUT Mode when a Computer Failure Occurs

When a remote cascade (RCAS) or remote output (ROUT) command is received during a computer failure (BSW=ON), the function block does not make a direct transition to the remote cascade (RCAS) or remote output (ROUT) mode but changes first to the over transition mode.

Subsequently, the computer failure status is determined during the scan period in that function block, and if there has been a recovery from the computer failure there is a transition to the remote cascade (RCAS) or remote output (ROUT) mode. If the computer failure occurrence status is unchanged, a transition is made to the backup mode for the computer.

The over transition mode refers to a compound block mode of the block mode just prior to the mode change command and the remote cascade (RCAS) or remote output (ROUT) mode.

Example:

The remote output (ROUT) command when the manual (MAN) mode has been specified as a computer backup mode, and the computer failure condition is being satisfied.

AUT
↓ Remote output (ROUT) command
AUT(ROUT) Over transition mode
↓ After one scan cycle
MAN(ROUT) Computer backup mode (when BSW=ON)

Example:

The remote cascade (RCAS) command when the computer failure condition is not satisfied

AUT
↓ Remote cascade (RCAS) command
AUT(RCAS) Over transition mode
↓ After one scan cycle
RCAS Remote cascade mode (when BSW=OFF)

A Block Mode Transition from the O/S Mode

When a block mode change command is received from the out-of-service (O/S) mode, the function block does not make a direct transition to the specified block mode but temporarily changes to an over transition mode such as O/S (MAN), O/S (AUT) or O/S (CAS).

In this case, when the function block which has undergone a block mode transition is a regulatory control block, the transition to the specified block mode (MAN, AUT, CAS) is made after initializing the output during the next scan period.

Example:

O/S
↓ AUT command
O/S(AUT)
↓ After one scan period
AUT

TIP

Output initialization processing includes the following operations:

- Output tracking in the IMAN state
- Reset the pulse width output
- Reset the time-proportional output

SEE ALSO

For more information about output tracking in the IMAN state, refer to:

4.5, "Output Tracking" on page 4-18

The Status Change Message

Status Change Message Bypass

When function blocks such as sequence control blocks change block mode, the event recording function sends the status change message from FCS to HIS to inform the operator.

To suppress this message can be defined on each function block in the Function Block Detail Builder.

- Status Change Message Bypass
  Select "Yes" or "No."

  The default is "No."

  In case of the sequence table block (ST16), the default is "Yes."

When status change message is set as Bypass, the message of status change is not recorded in the historical message file in HIS.

When the status change is performed manually on HIS, the status change message will all be recorded in the historical message file in HIS regardless the setting of bypass.

6.1.4 Block Mode Change Command

Block mode change commands are operations outside of the function block to transit the block mode, such as operations from the operation and monitoring function or status change command from sequence control blocks or from general-purpose calculation blocks (CALCU and CALCU-C).

■ Block Mode Change Commands

Operations and descriptions related to block mode change commands are shown as follows.

Table 6.1.4-1 Mode Change Commands

Command operation	Description
O/S command	Sets the O/S mode and resets all other modes.
MAN command	Sets the MAN mode and resets modes other than the O/S, IMAN and TRK modes.
AUT command	Sets the AUT mode and resets modes other than the O/S, IMAN and TRK modes.
CAS command	Sets the CAS mode and resets modes other than the O/S, IMAN and TRK modes.
PRD command	Sets the PRD mode and resets modes other than the O/S, IMAN and TRK modes (*1).
RCAS command	Sets the RCAS mode (*2).
ROUT command	Sets the ROUT mode (*3).

*1: The PRD command from the O/S mode is invalid.
*2: The RCAS command from the O/S or ROUT mode is invalid.
*3: The ROUT command from the O/S or RCAS mode is invalid.

When a function block is in calibration status, the command to change block mode to AUT, CAS, RCAS, ROUT mode given from an HIS will be ignored. If the option of [Allow to change function block to PRD mode while CAL is on (XL compatible)] is not enabled for the FCS, the command to change the block to primary direct (PRD) mode may force the function block to switch to PRD mode for a short moment, but the function block will fall to manual (MAN) mode right away.

6.1.5 Block Mode Transition Condition

Block mode transition conditions are conditions that internally initiate the function block to transit its mode from one to another.

■ Block Mode Transition Conditions

Block mode transition conditions are conditions that internally initiate a function block to transit its block mode from one to the other. When a specific condition for transiting to a certain block mode establishes, such as abnormality in I/O module, the function block itself changes the block mode automatically by itself according to the block mode transition conditions.

Block mode transition conditions are listed below:

- The tracking condition
- The MAN fallback condition
- The AUT fallback condition
- The initialization manual condition
- The computer failure condition
- The block mode change interlock condition

■ The Tracking Condition

For the function blocks that have tracking (TRK) function, a tracking switch (TSW) is provided. According to the status of the tracking switch (ON/OFF), the function blocks set or reset its tracking (TRK) mode.

A function block changes to the tracking (TRK) mode only when the tracking switch is turned on. The status of the tracking switch can be changed by the external contact input or set from other function blocks such as from the sequence control block.

Example:

AUT
↓ TSW = ON
TRK (AUT)
↓ TSW = OFF
AUT

■ The MAN Fallback Condition

When the MAN fallback condition establishes, the function block changes to the manual (MAN) mode regardless of the current operation status. Control calculation stops and the operation status changes to manual.

Example:

AUT → MAN
IMAN (CAS) → IMAN (MAN)

The MAN fallback condition establishes when following events occur.

- When the data status of the process variable (PV) is bad value (BAD). However, the MAN fallback condition does not establishes when the block mode is the primary direct (PRD) or remote output (ROUT) mode (excluding compound block modes that are obtained during computer backup operation).

- When the data status of the process variable (PV) is calibration (CAL).However, the MAN fallback condition does not establish when the block mode is the primary direct (PRD) mode if the option of [Allow to change function block to PRD mode while CAL is on (XL compatible)] is enabled.
- When the data status of the manipulated output value (MV) is output failure (PTPF).
- When the data status of the setpoint value (SV) is bad value (BAD).
- At the regulatory control block where the manipulated output value (MV) is connected to the process I/O, when the FCS has been started after initialization.
- When the block mode change interlock condition has been satisfied.
- At the regulatory control block where the manipulated output value (MV) is connected to the process I/O, when the I/O module that contains the I/O point of the connection destination has been changed due to online maintenance.

Even when the above events that constitute the MAN fallback condition have vanished, the block mode remains manual (MAN) and does not automatically return to the original mode.

■ The AUT Fallback Condition

When the AUT fallback condition establishes, the block changes to the automatic (AUT) mode, and then operates in automatic status. The mode transition is the same as enforced by the external AUT command.

Example:

CAS → AUT

IMAN (CAS) → IMAN (AUT)

The AUT fallback condition establishes when the AUT fallback item (in the builder) is specified "Yes" and the data status of the cascade setpoint value (CSV) is bad value (BAD) or no communication (NCOM).

The AUT fallback condition establishes only in the cascade (CAS) mode or primary direct (PRD) mode. The mode transition is the same as enforced by the external AUT command.

Even when the above events that constitute the AUT fallback condition have vanished, the block mode remains automatic (AUT) and does not automatically return to the original mode.

The setting of AUT fallback condition can be in the Function Block Detail Builder.

AUT Fallback: Select "Yes" or "No."
The default is "No."

■ The Initialization Manual Condition

When the initialization manual condition establishes, the block mode changes to the initialization manual (IMAN) mode, and control operation temporarily stops.

A regulatory control block changes to initialization manual (IMAN) mode only when the internal condition for the block mode establishes. When the condition vanishes, the initialization manual (IMAN) mode is reset and replaced by a new active mode.

Example:

AUT
↓ IMAN condition satisfied
IMAN (AUT)
↓ IMAN condition recovered
AUT

The initialization manual condition establishes when any of the following events occurs:

- When the manipulated output value (MV) connected destination block changes to conditional (CND) status (i.e. cascade loop open).
- When the manipulated output value (MV) connected destination block is in no communication (NCOM) or the output failure (PTPF) alarm status.
- When the manipulated output value (MV) connected destination block is a switch block, and the connection is switched off. (i.e., cascade loop open).
- When the manipulated output value (MV) is output to the process equipment, and the equipment is in abnormal status or the output open occurs.
- When the output signal is not the pulse width output type, and the data status of the input signal of the TIN or TSI terminal has changed to bad value (BAD) in the tracking (TRK) mode.

■ The Computer Failure Condition

When the computer failure condition establishes, operation in remote cascade (RCAS) or remote output (ROUT) mode temporarily stops, and then switches to computer backup mode.

Example:

When the automatic (AUT) mode is specified as a computer backup mode

RCAS

↓ Computer failure occurs

AUT (RCAS)

↓ Computer failure recovers

RCAS

In the remote cascade (RCAS) mode or remote output (ROUT) mode, a setpoint value or manipulated output value is transmitted from a supervisory system computer through remote setting communication via control bus.

If abnormality is detected in remote setting communication, the computer failure condition establishes. Operation in the remote cascade (RCAS) or remote output (ROUT) mode temporarily stops, switches to a computer backup mode. When the computer failure condition vanishes, the mode immediately returns to the original (RCAS or ROUT) mode.

The function blocks that have the remote cascade (RCAS) or remote output (ROUT) mode is provided with a backup switch (BSW). According to the status of this switch (ON/OFF), the function block determines whether the computer failure conditions exist or not.

- When BSW=ON, the computer failure conditions exist.
- When BSW=OFF, the computer failure conditions not exist

Determine as desired the status of the backup switch (BSW) in the control application such as a sequence table and set it for the backup switch (BSW). However, in a block mode other than remote cascade (RCAS) or remote output (ROUT), the block mode does not change even if the backup switch (BSW) is operated.

The computer backup mode can be set with the Function Block Detail Builder.

- Computer Backup Mode
  Status setting when the computer is down.

  Select from “MAN,” “AUT” and “CAS.”

  Default is “MAN.”

  Select from “AUT” and “CAS” in the case of a Control Signal Splitter Block (SPLIT).

  Default is “AUT.”

■ The Block Mode Change Interlock Condition

When the block mode change interlock condition establishes, the block changes to manual (MAN) mode, and the change command to automatic operation status (AUT, CAS, PRD, RCAS or ROUT mode) become invalid.

If automatic operation cannot continue due to abnormality in the plant, etc., it is necessary not only to stop the currently automatically operated control loops but also to prevent them from being put into AUT again.

To prevent the function blocks change to automatic operation mode, the automatic operation interlock switch can be used. The automatic operation interlock switch can be connected to the function blocks that have automatic operation mode.

When the status of the interlock switch connected to the terminal (INT) becomes ON, the block mode change interlock establishes, then succeeded by the following actions:

- The MAN fallback condition establishes, and the block mode changes to manual (MAN).
- Any mode change command to put the function block into an automatic operation status (AUT, CAS, PRD, RCAS or ROUT mode) is invalidated.

Any datum that has a logical value, such as a contact I/O or internal switch, can be specified as an automatic operation interlock switch regardless the type of the element.

6.2 Block Status

▼ **Block Status**

The operating state of a function block may be monitored via block status.

■ Block Status

Block status is the information that represents the operating state of a function block. Block status and block mode simultaneously reflect an overall operation state of a function block.

The different type of function blocks have different kinds of block status. Some function blocks do not have any block status.

A function block may check itself and display the result via block status.

Similar to block modes, the block status is also categorized by priority levels. A block status with same priority level exists exclusively. It is not possible that a block can be in two statuses with same priority level at the same time. A supplementary status can exists together with the main status, however, in this case, only the main status is displayed on HIS as the block status.

Table 6.2-1 Block Status of Each Function Block

Priority level	Function block name									
	Regulatory control block			Sequence control block						Arithmetic-calculation block
	MC-2 MC-2E MC-3 MC-3E	BSE-TU-1 BSE-TU-2	PTC	SO-1 SO-1E SO-2 SO-2E	SIO-11 SIO-11E SIO-12 SIO-12E SIO-21 SIO-21E SIO-22 SIO-22E SIO-12P SIO-12PE SIO-22P SIO-22PE	TM CTP	CTS	CI	CO	INTEG AVE-C
3	ANCK	-	-	-	ANCK	PAUS	-	ERR	-	-
2	OFF LOCK	-	PAUS	-	-	PALM CTUP NR	PALM CTUP NR	HI LO NR	HI LO NR	-
1	SIM NR	STRT IBCH STUP STDY ERLY PBCH END NCNT RSET EMST EEMS RSTR	CTUP PALM NR	SIM NR	SIM NR	RUN STOP	RUN STOP	-	-	RUN STOP

6.3 Alarm Status

The process alarm may be monitored and managed via alarm statuses of data items.

■ Alarm Status Common to Regulatory Control Blocks

An alarm status is information that represents the alarm state of a process which was detected by the function block. The alarm status that can be satisfied for a particular function block varies depending on the type of the function block.

Some function blocks do not have any alarm status.

The alarm status for the operation and monitoring function is displayed in the data item ALRM. When multiple alarms are occurring, the alarm status with the highest alarm display priority will be displayed.

The priority for alarm display is as follows:

OOP>IOP>IOP->HH>LL>HI>LO>DV+>DV->VEL+>VEL->MHI>MLO>CNF

Table 6.3-1 Alarm Status Common to Regulatory Control Blocks

Symbol	Name	Description
NR	Normal	Indicates a state in which no alarm has occurred.
OOP	Output OPen Alarm	Indicates a state in which the output data status has become output failure (PTPF) as a result of the failure or disconnection of an operation terminal or process I/O device or the abnormality of output destination data. Normally, the output function is stopped.
IOP	High Input Open Alarm	Indicates a state in which the input data status has become bad value (BAD) as a result of the failure or disconnection of a detection terminal or process I/O device or the abnormality of input destination data. Normally, any processing that uses input signals is stopped. If the input signal has been overshot due to disconnection, etc., this alarm indicates a state in which input is overshot to the high-limit direction.
IOP-	Low Input Open Alarm	Indicates a state in which the input signal has been overshot to the low-limit direction due to disconnection, etc. The input data status becomes bad value (BAD). Normally, any processing that uses input signals is stopped.
HH	High High Alarm	Indicates a state in which the process variable exceeds the high high-limit alarm setpoint.
LL	Low Low Alarm	Indicates a state in which the process variable falls below the low low-limit alarm setpoint.
HI	High Alarm	Indicates a state in which the process variable exceeds the high-limit alarm setpoint.
LO	Low Alarm	Indicates a state in which the process variable falls below the low-limit alarm setpoint.
DV+	Deviation Alarm +	Indicates a state in which the deviation between the process variable and the setpoint value exceeds the deviation alarm setpoint in the positive direction.
DV-	Deviation Alarm -	Indicates a state in which the deviation between the process variable and the setpoint value exceeds the deviation alarm setpoint in the negative direction.
VEL+	Velocity Alarm +	Indicates a state in which the change amount of the input signal within a specified time exceeds the velocity limit alarm setpoint in the positive direction.
VEL-	Velocity Alarm -	Indicates a state in which the change amount of the input signal within a specified time exceeds the velocity limit alarm setpoint in the negative direction.

Symbol	Name	Description
MHI	Output High Alarm	Indicate a state in which the output signal almost exceeded the output high-limit value. The actual output is limited to the output high-limit value.
MLO	Output Low Alarm	It indicates a state in which the output signal almost fell below the output low-limit value. The actual output is limited to the output low-limit value.
CNF	Connection Failure Alarm	Indicates a state in which a block mode of the function block in the I/O connection destination is in the out of service (O/S) mode. This alarm controls a temporary out of service state due to maintenance, and indicates a function block which is still in operation. Normally, IOP or OOP occurs simultaneously.

■ Alarm Status Common to Calculation Blocks

The alarm status for the operation and monitoring function is displayed in the data item ALRM. When multiple alarms are occurring, the alarm status with the highest alarm display priority will be displayed.

The priority for alarm display is as follows:

IOP>IOP->HH>LL>HI>LO>VEL+>VEL->CNF

The alarm status is NR if no alarm occurs.

Table 6.3-2 Alarm Status Common to Calculation Blocks

Symbol	Name	Description
NR	Normal	Indicates a state in which no alarm has occurred.
IOP	High Input Open Alarm	Indicates a state in which the input data status has become bad value (BAD) as a result of the failure or disconnection of a detection terminal or process I/O device or the abnormality of input destination data. Normally, any processing that uses input signals is stopped. If the input signal has been overshot due to disconnection, etc., this alarm indicates a state in which input is overshot to the high-limit direction.
IOP-	Low Input Open Alarm	Indicates a state in which the input signal has been overshot to the low-limit direction due to disconnection, etc. The input data status becomes bad value (BAD). Normally, any processing that uses input signals is stopped.
HH	High High Alarm	Indicates a state in which the calculated input value exceeds the high high-limit alarm setpoint.
LL	Low Low Alarm	Indicates a state in which the calculated input value falls below the low low-limit alarm setpoint.
HI	High Alarm	Indicates a state in which the calculated input value exceeds the high-limit alarm setpoint.
LO	Low Alarm	Indicates a state in which the calculated input value falls below the low-limit alarm setpoint.
VEL+	Velocity Alarm +	Indicates a state in which the change amount of the calculated input value within a specified time exceeds the velocity limit alarm setpoint in the positive direction.
VEL-	Velocity Alarm -	Indicates a state in which the change amount of the calculated input value within a specified time exceeds the velocity limit alarm setpoint in the negative direction.
CNF	Connection Failure Alarm	Indicates a state in which the block mode of a function block in the I/O connection destination is in the out of service (O/S) mode. This alarm controls a temporary out of service state due to maintenance, and indicates a function block which is still in operation.

Symbol	Name	Description
CERR	Computation Error Alarm	Indicates a state in which a computation error has occurred during a user-defined calculation processing. Calculation processing is stopped.

TIP

The one-shot processing initiated by one-shot start does not update the alarm status of the function block.

■ Alarm Status Common to Sequence Control Blocks

Table 6.3-3 Alarm Status Common to Sequence Control Blocks

Symbol	Name	Description
NR	Normal	Indicates a state in which no alarm has occurred.
OOP	Output Open	Indicates that the block is in an output open state due to the disconnection of a process I/O device of the output destination or other failure.
IOP	Input Open	Indicates a state in which the input data status has become bad value (BAD) as a result of the failure or disconnection of a detection terminal or process I/O device or the abnormality of input destination data.
CNF	Connection Failure	Indicates a state in which the block mode of a function block in the I/O connection destination is in the out of service (O/S) mode. This alarm controls a temporary out of service state due to maintenance, and indicates a function block which is still in operation.
PERR	Computation Error Answerback Inconsistency Alarm	Indicates a state in which an illegal input pattern has occurred, such as when the full-open and full-close signals are inputted simultaneously.
ANS+	Answerback Error +	Indicates a state in which answerback check is being performed and the manipulated output value (MV) of ON operation and the answerbacked process variable (PV) do not agree.
ANS-	Answerback Error -	Indicates a state in which answerback check is being performed and the manipulated output value (MV) of OFF operation and the answerbacked process variable (PV) do not agree.

■ Alarm Status of Each Function Block

Different model of function block supports different alarm status.

SEE ALSO

For more information about alarm status of regulatory control blocks, refer to:

"■ Alarm Processing Possible for Each Regulatory Control" in 1.1.3, "Input Processing, Output processing, and Alarm Processing Possible for Each Regulatory Control Block" in the Function Blocks Reference Vol.1 (IM 33K03E22-50E)

For more information about alarm status of calculation blocks, refer to:

"■ Alarm Processing Possible in Each Calculation Block" in 1.3.1, "Input Processing, Output processing, and Alarm Processing Possible for Each Calculation Block" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)

For more information about alarm status of sequence control blocks, refer to:

2.1.1, "Alarm Processing of Sequence Control Blocks" in the Function Blocks Reference Vol.2 (IM 33K03E23-50E)

6.4 Data Status

▼ **Data Status**

Data status is the information that represents the quality of data. It is used for judging the proper operations according to the reliability of the data.

■ Data Status

Data status is the information that represents the reliability of data. The data obtained from I/O module is passed from one function block to another with its data status information. Data status is observed when various exceptional events occurred due to abnormality in the process and it is helpful for making a proper decision for control operation.

Table 6.4-1 Data Status (1/2)

Symbol	Name	Description
BAD	BAD value	Indicates a state in which a normal data value cannot be obtained. The data value stored when this status occurs may be a meaningless value or the last normal value which has been stored.
QST	QueSTionable value	Indicates that the data value is questionable and cannot be determined whether it is normal or bad. The data value stored when this status occurs may be a value inputted from outside while it is in the QST status, a manually set value using the CAL function, or the last normal value which has been stored.
NCOM	No COMmunication	Indicates that when data is inputted or outputted through communication, the communication has been disconnected and the data has not been updated. Used only for I/O data that is exchanged with other control stations.
NFP	Not From Process	Indicates that the data value is not derived from a process I/O. The data value stored when this status occurs may be a value inputted from outside while it is in the NFP status, a calculated value, or a manually set value using the CAL function.
PTPF	Path To Process Failed	Indicates a state in which output is being disabled due to the abnormality of the block itself or the output destination. If the output destination is a PI/O, this status occurs when output open (OOP), not ready (NRDY) or power failure has occurred. If the output destination is a function block, this status occurs when the output destination block is in the out of service (O/S) mode.
CLP+	CLamP high	Indicates that output is clamped at the high-limit value. This status occurs when the block itself is limited by the output high limit or when the data status of the output destination is clamp high (CLP+).
CLP-	CLamP low	Represents that output is clamped at the low-limit value. This status occurs when the block itself is limited by the output low limit or when the data status of the output destination is clamp low (CLP-).
CND	CoNDitional	Indicates that cascade connection is open. This status occurs when a downstream function block changes to the non-cascade mode or the cascade connection path has been disconnected due to switching, etc. Used only for data that is the object of cascade connection (MV, CSV, etc.)
CAL	CALibration	Indicates a state in which the data value can be replaced manually as an emergency. This status occurs when a downstream function block changes to the non-cascade mode or value will not be updated until it is replaced manually.
NEFV	Not EFfecTive	Indicates a state in which the data value is invalid. This is a state in which no setpoint value has been set manually after the CAL status was obtained or the value is yet to be updated after the CAL status was turned off.

Table 6.4-2 Data Status (2/2)

Symbol	Name	Description
O/S	Out of Service	Indicates that the function block of the I/O destination is in O/S mode. If the operation is input, the data value is not updated.
MNT	MaiNTenance	Indicates that the function block of the I/O destination is undergoing online maintenance. If the operation is input, the data value is not updated. Normally, data reference is not performed while this data status in on, since online maintenance is performed as a group between function block executions and data access processing.
IOP+	Input Open high	Indicates that the process I/O of the input destination is in a high limit input open state due to disconnection or other failure. The data value is not updated. The PV value is forcibly set to a special value only when the PV overshoot function is activated.
IOP-	Input OPen low	Indicates that the process I/O of the input destination is in a low limit input open state due to disconnection or other failure. The data value is not updated. The PV value is forcibly set to a special value only when the PV overshoot function is activated.
OOP	Output OPen	Indicates that the process I/O of the output destination is in an output open state due to disconnection or other failure.
NRDY	PI/O Not ReaDY	Indicates that the process I/O of the I/O destination is in an operation disabled state due to power failure, maintenance or a failure. If the operation is input, the data value is not updated.
PFAL	PI/O Power FAiLure	Indicates that the process I/O of the I/O destination is not responding due to power failure or other reason and is in an operation disabled state. If the operation is input, the data value is not updated.
LPFL	PI/O Long Power FaiLure	Indicates that the process I/O of the I/O destination has been non-responsive for a long time due to power failure or other reason and is in an operation disabled state. If the operation is input, the data value is not updated.
MINT	Master INiTialize	Indicates that the upstream side of the cascade connection is in a state where a balance operation should be performed.
SINT	Slave INiTialize	Indicates that the downstream side of the cascade connection is in a state where a balance due to power failure or other reason and is in an operation disabled state. If the operation
SVPB	SV PushBack	Indicates that the downstream side of the cascade connection is in a state where the CSV should be made to match SV by the SV pushback operation.

Blank Page

7. Process Timing

The function block executes a process in accordance with the process timing.

This chapter describes the process timing for the regulatory control blocks, the sequence control blocks, the calculation blocks, and SEBOL programs.

■ Process Timing

The process timing is a timing at which a function block executes a process. An individual function block executes an input, a calculation or an output processing in accordance with the process timing defined in.

There are four types of the process timing.

- Periodic Execution
  This is repeatedly executed per preset period. The periodic execution can be used in the regulatory control block, sequence control block, and calculation block.
- One-Shot Execution
  This is executed only once when it is invoked from other function blocks. The one-shot Execution can be used in the sequence control block and the calculation block.
- Initial Execution/Restart Execution
  This is executed when the FCS executes the cold start or restart process. The initial execution can be used in the sequence control block.
- Restricted Initial Execution
  This is executed when the FCS executes the initial cold start process. The restricted initial execution can be used in the sequence control block.

■ Process Timing for Regulatory Control Block

The process timing for the regulatory control block is the periodic execution.

■ Process Timing for Calculation Block

There are following two types of the process timings for the calculation block.

- Periodic Execution
- One-Shot Execution

■ Process Timing for Sequence Control Block

The process timing for the sequence control block has following four different types of execution timing.

- Periodic Execution
- One-Shot Execution
- Initial Execution/Restart Execution
- Restricted Initial Execution

■ SEBOL Execution Timing

There are two types of SEBOL execution timings.

- Timeshare Execution Timing
  Timeshare execution timing denotes that the SEBOL programs are processed at the CPU idle time. This type SEBOL programs are referred to as “Timeshare SEBOL.”

- Periodic Execution Timing
  Periodic execution timing denotes that the SEBOL programs are processed at FCS basic scan. This type SEBOL programs are referred to as “Periodic SEBOL.”

TIP

Periodic SEBOL programs are available in FFCS-V only.

7.1 Process Timing for Regulatory Control Block

The process timing for the regulatory control block is the periodic execution.

■ Periodic Execution for Regulatory Control Block

The regulatory control block executes process repeatedly in a predetermined period. Normally, these periodic execution function blocks execute per scan period.

The timing that individual function blocks activate as well as control drawings are determined by following factors in the periodic execution of the regulatory control block.

● Scan Period

The scan period is the period at which the function block is executed periodically. The periodic execution function block executes a process based on the scan period.

There are three types of scan periods: the basic scan, the medium-speed scan (*1) and the high-speed scan. One of these scan periods can be selected for each individual function block. However, the medium-speed scan (*1) and high-speed scan cannot be selected for some function blocks.

*1: The medium-speed scan setting is available only for the FFCS series, KFCS2, KFCS, LFCS2, and LFCS.

● Order of Process Execution

The order of process execution refers to a sequence in which the control drawing and individual function block are executed in the periodic execution. The order of process execution determines the execution timing of the control drawing and individual function block within a scan period.

● Timing for Process I/O

The timing for the process I/O is a timing in which data input/output is executed between the function block and the process I/O.

The timing for the process input/output differs from the type of input/output is analog or contact.

● Control Period of Controller Block

Among the regulatory control blocks, the controller block has a control period that is dependent on process timing. The control period of the controller block is a period in which control calculation and output processing are executed when the controller block is performing automatic operation (AUT, CAS, RCAS). The control period of the controller block is always an integer multiple of the scan period.

The only processing that the controller block always performs for every scan period is input processing and alarm processing. Control calculation and output processing during automatic operation (AUT, CAS, RCAS) are performed during every control period.

7.1.1 Scan Period

The periodic execution function block performs processing based on this scan period.

■ Scan Period

▼ Scan Period, Fast-Scan Period, Medium-Speed Scan Period

Scan period determines a period for the periodic execution of the function block.

There are three types of scan periods: basic scan, medium-speed scan (*1) and high-speed scan.

*1: The medium-speed scan setting is available only for the FFCS series, KFCS2, KFCS, LFCS2, and LFCS.

● Basic Scan

The basic scan is a standard scan period which is common to function blocks.

The basic scan period is fixed to 1 second. This cannot be changed.

● Medium-Speed Scan : FFCS Series/KFCS2/KFCS/LFCS2/LFCS

The medium-speed scan is a scan period suited for the process control that requires quicker response than what can be achieved with the basic scan setting. Setting value of the medium-speed scan can be selected by each FCS according to its use.

Setting value of the medium-speed scan can be changed on FCS property sheet:

Medium-speed scan period:	Select “200 ms” or “500 ms.” “50 ms”, “100 ms” or “250 ms” can also be used by direct entry from keyboard. The default is “500 ms.”

● High-Speed Scan

The high-speed scan is a scan period suited for the process control that requires high-speed response. Setting value of the high-speed scan can be selected by each FCS according to its use.

Setting value of the high-speed scan can be changed on FCS property sheet:

High-speed scan period:	Select “200 ms” or “500 ms.” “50 ms ”, “100 ms” or “250 ms” can also be used by direct entry from keyboard. The default is “200 ms.”

● Set Scan Speed of Function Blocks

Scan speed of function blocks can be set on the Function Block Detail Builder. Basic scan, medium-speed scan or high-speed scan can be selected on the builder in accordance with requirement of response time.

Scan period:	Select [Basic Scan], [Medium-speed Scan](*1)or [High-speed Scan.] The default setting is [Basic Scan.]

*1: The medium-speed scan setting is available only for the FFCS series, KFCS2, KFCS, LFCS2, and LFCS.

Note that [Basic Scan], not [Medium-speed Scan] or [High-speed Scan], should be set for the following function blocks:

PID-TP, MC-2, MC-2E, MC-3, MC-3E, PG-L13, SLCD, SLPC, SLMC, SMST-111, SMST-121, SMRT, SBSD, SLBC, SLCC, STLD

With certain function blocks, processing can be executed in a scan longer than the basic scan by specifying the [Scan Coefficient] and [Scan Phase] parameters.

● Scan Coefficient, Scan Phase

▼ Scan Coefficient, Scan Phase

When a scan coefficient(*1) is specified in addition to the scan period in the Function Block Detail Builder, input indicator blocks (PVI), input indicator blocks with deviation alarm (PVI-DV) and general-purpose calculation blocks (CALCU, CALCU-C) can be executed based on their actual scan period being calculated as follows:

- Actual scan period = Scan period • Scan coefficient
- Scan coefficient: 1, 2, 4, 8, 16, 32 or 64

If the scan coefficient is represented by N, the function block is executed every N x scan period.

In addition, when a scan phase is specified in the Function Block Detail Builder, in which of the N times of scans the function block is executed can be defined. Specify the scan phase using a numeric value in the following range:

- Scan phase: 0 to ((Scan coefficient) - 1)

*1: Scan Coefficient, and Scan phase can be used in FCSs except PFCS.

7.1.2 Order of Process Execution

The order of process execution refers to a sequence in which the control drawing and individual function block are executed in the periodic execution. The process timing of a periodic execution regulatory function block is determined by the orders of execution of the control drawings and the function blocks.

The following section describes the orders in which the control drawings and individual function blocks are executed in the periodic execution.

■ Order of Process Execution for Control Drawings/Function Blocks

The diagram below shows an example of executing control drawings each consisting of function blocks being executed in the high-speed scan, medium-speed scan (*1) and basic scan. In this example, three control drawings are processed. The groups of high-speed scan function blocks in the respective drawings are indicated as A, B and C. Similarly, the groups of medium-speed scan (*1) function blocks in the respective control drawings are indicated as A', B' and C'; and the groups of basic scan function blocks, a, b and c. In the diagram below and the explanation that follows, the processing of the function blocks belonging to A, B and C is referred to as "high-speed processing"; processing of the function blocks belonging to A', B' and C', "medium-speed processing" (*2); and processing of the function blocks belonging to a, b and c, "basic processing."

*1: The medium-speed scan setting is available only for FFCS series, KFCS2, KFCS, LFCS2, and LFCS.
*2: The medium-speed processing function is available only for FFCS series, KFCS2, KFCS, LFCS2, and LFCS.

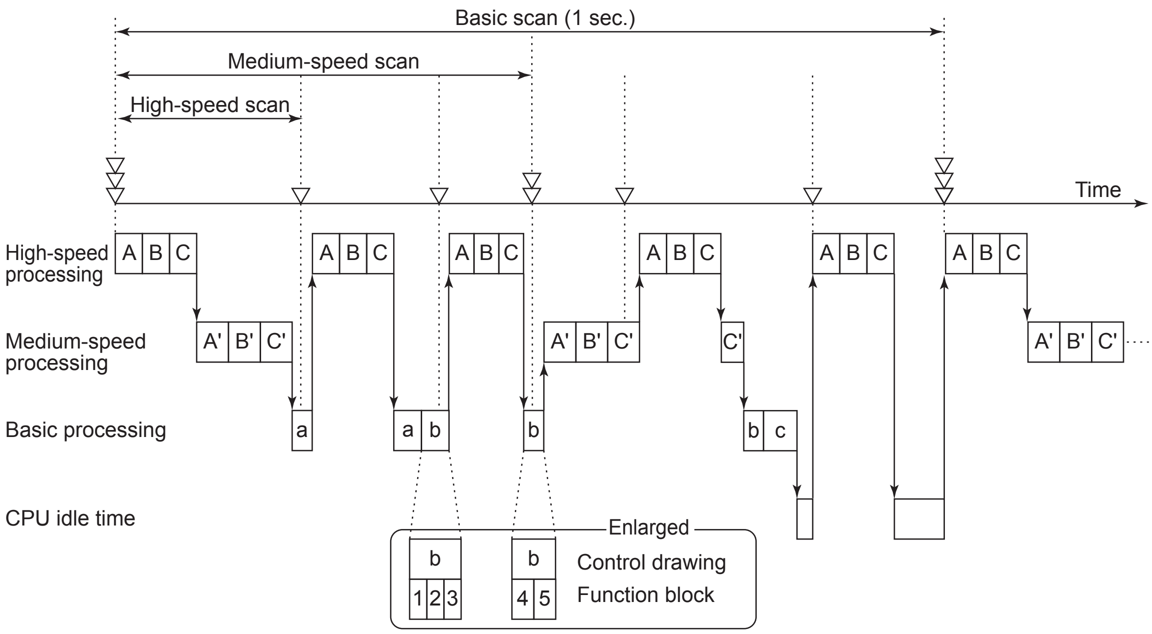


Figure 7.1.2-1 Example of Control Drawings/Function Blocks Process Execution

- The high-speed processing has priority over the medium-speed processing or basic processing. The medium-speed processing has priority over the basic processing.
- Once processing of all high-speed function blocks have been completed, the medium-speed processing is executed.
  When execution of all high-speed and medium-speed processes of function blocks have been completed, the basic processing is executed.
- In case that the high-speed processing gets its timing for execution during the basic processing or medium-speed processing is being executed, the high-speed processing interrupts the basic processing or medium-speed processing by making the basic processing or medium-speed processing pause at the gap among function blocks' basic processing or medium-speed processing. Once all function blocks of high-speed processing are

completed to execute, the basic processing or medium-speed processing is resumed from where it was interrupted.

- In case that the medium-speed processing its timing for execution during the basic processing is being executed, the medium-speed processing interrupts the basic processing by making the basic processing pause at the gap among function blocks' basic processing. Once all function blocks of medium-speed processing are completed to execute, the basic processing is resumed from where it was interrupted.
- The high-speed processing of function blocks are executed for each of the control drawings containing the function blocks and in the order of control drawing numbers. Function blocks having the same scan period within the same control drawing are executed in the set execution order (order of the function block numbers defined).
  The medium-speed processing and basic processing of function blocks are executed in the same order as applied to the high-speed processing.
- Processing of each function block is executed only once per single scan period.
- Other processes are executed in the idle time after the high-speed processing, medium-speed processing and basic processing are completed.

TIP

The processing of function blocks is executed asynchronously between different FCSs.

7.1.3 Timing of Process I/O

The timing of process I/O refers to the timing at which the data input/output is executed between the function block and the process I/O modules.

The timing of process I/O depends upon the type of the input/output data whether it is analog or digital.

The following section explains the timing of process I/O.

■ Data Flow in Process I/O

The flow of data in process input/output is different for analog data and for digital data. The following section explains the data flow of each type.

● Analog Data : PFCS

For analog input/output signals, the I/O module and the function block exchange data via the process I/O image in the main memory of processor unit.

The diagram below shows an image of analog data input/output processing.

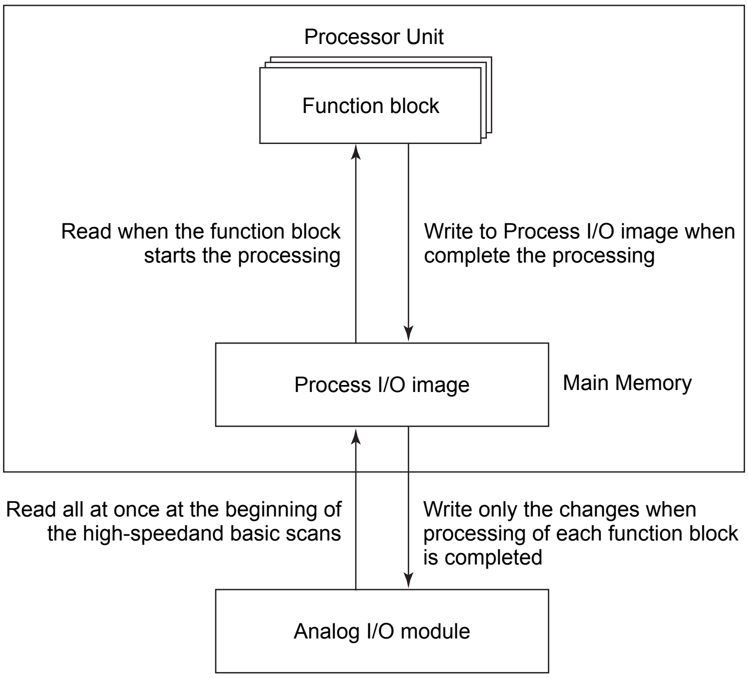
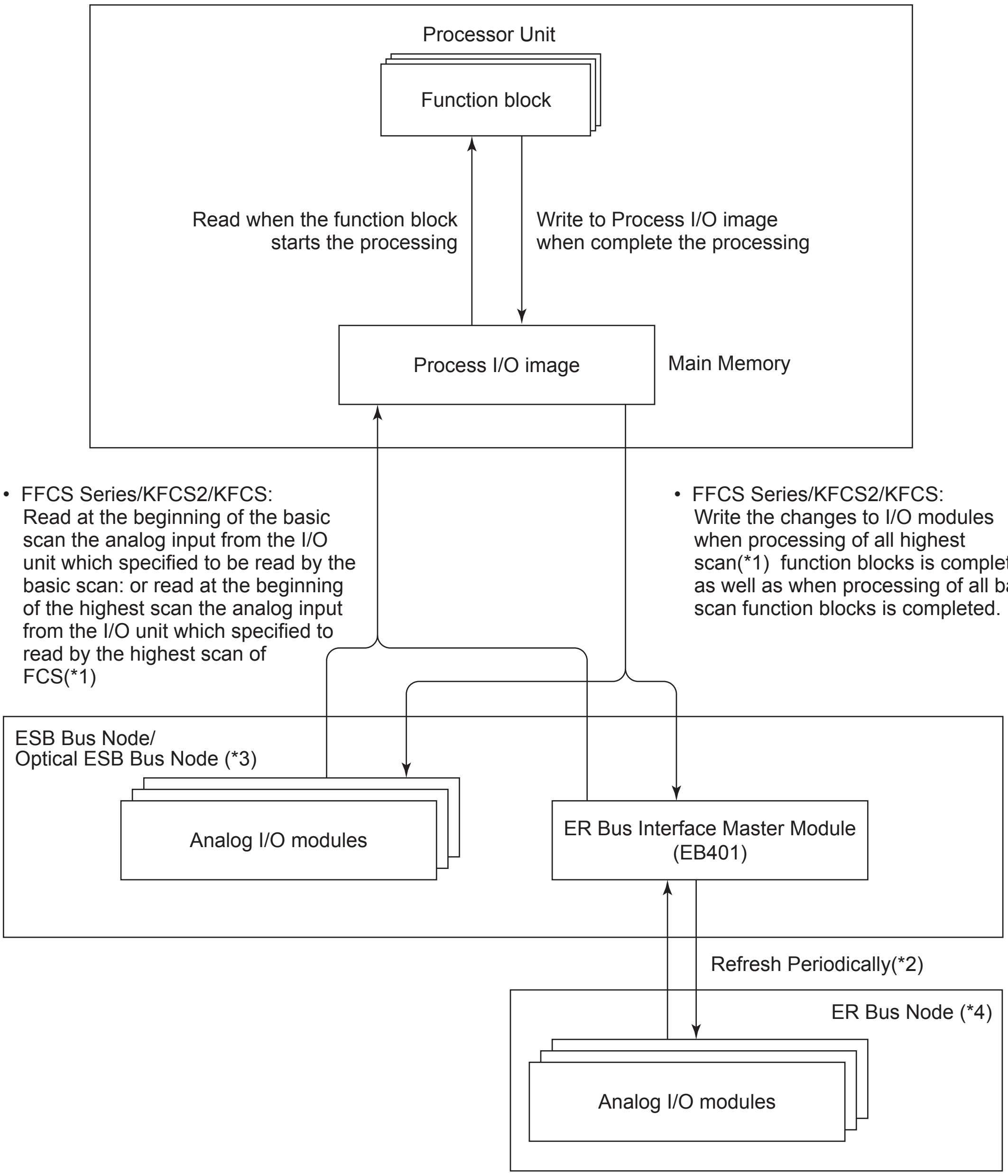


Figure 7.1.3-1 Image of Analog Data Input/Output : PFCS

● Analog Data : FFCS Series/KFCS2/KFCS

For analog input/output signals, the I/O module and the function block perform input/output of data via the process I/O image in the main memory of processor unit.

The diagram below shows an image of analog data input/output processing.



*1: The Highest Scan of FCS means that if the FCS is applied with high-speed scan, the High-Speed Scan is the highest scan, and otherwise the Medium-Speed Scan is. If the FCS is applied with basic scan only, the Basic Scan is the highest scan of FCS.
*2: The period of analog I/O data refresh between EB401 and ER bus node varies with the number of ER bus nodes connected.
*3: The input processing of the analog I/O modules and EB401 module inserted in the slots of FFCS series FCU are the same as the processing of the modules in the ESB bus node/Optical ESB bus node. Note that the Optical ESB bus nodes can be connected to only the FFCS-V.
*4: The ER bus nodes cannot be connected to the FFCS-V.

Figure 7.1.3-2 Image of Analog Data Input/Output : FFCS Series/KFCS2/KFCS

The period of data refresh between EB401 and ER bus nodes varies with the number of ER bus nodes connected. The data refresh period corresponds to the number of nodes are shown as follows.

Table 7.1.3-1 Data Refresh Period between EB401 and Input/Output Module

	Number of ER Bus Nodes		
	2	4	6
Analog I/O (None HART)	50 ms	100 ms	200 ms

	Number of ER Bus Nodes		
	2	4	6
Analog I/O (HART)	100 ms	200 ms	400 ms
HART Variables	1 to 2 seconds		

● Analog Data : LFCS2/LFCS/SFCS

For analog input/output signals, the I/O module and the function block perform input/output of data via the process I/O image in the main memory of processor unit.

The diagram below shows an image of analog data input/output processing.

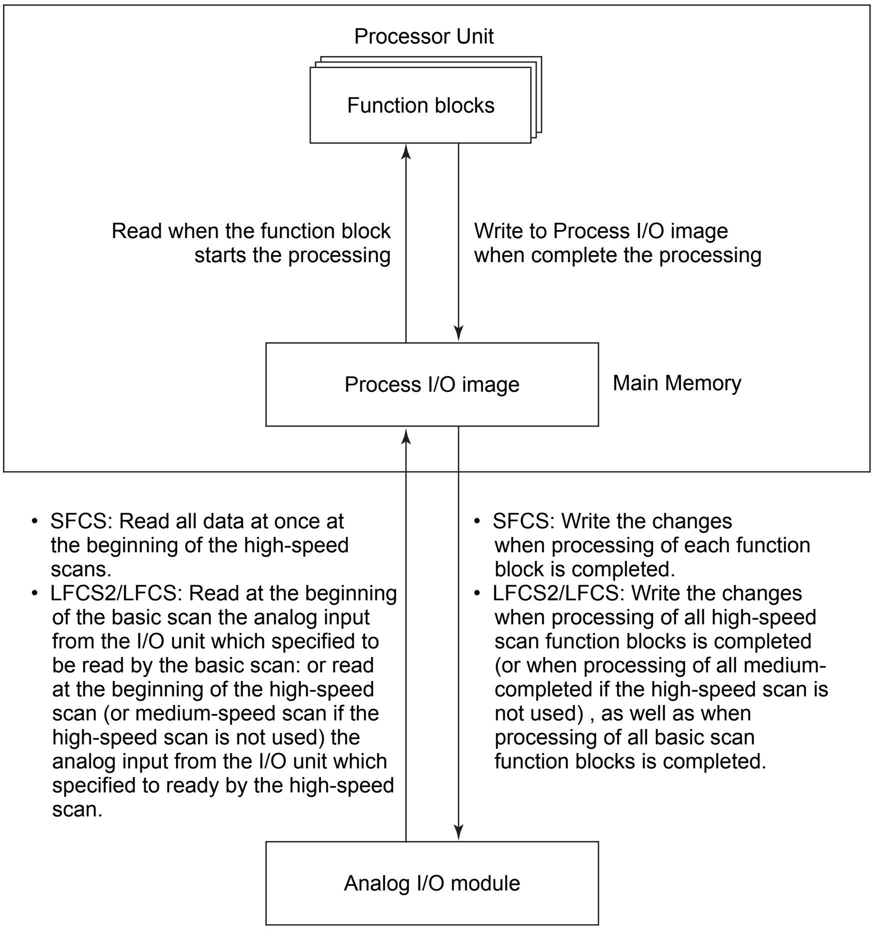


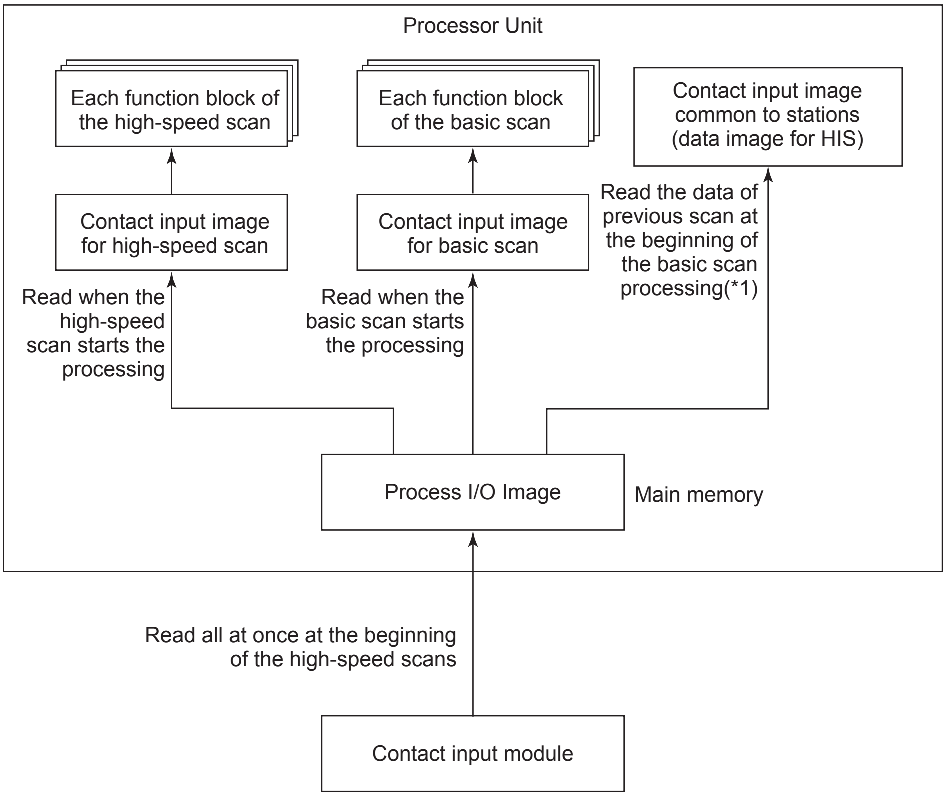
Figure 7.1.3-3 Image of Analog Data Input/Output : LFCS2/LFCS/SFCS

● Contact I/O Data : PFCS

For contact input, the contact input module and the function block perform data input processing via the process I/O image in the main memory of the processor unit and the contact input image of the corresponding scan period stored in the data buffer areas.

Since the contact input data are stored in the image area, the function blocks which are operated in the same period use the contact input's image read at the beginning of that scan period.

The diagram below shows an image of contact input data processing:



*1: The contact input image common to stations (data image for HIS) will be updated at the beginning of the next basic scan so as to match the data recognized by the function blocks in the FCS at the previous scan (before updating the process I/O image). Therefore, the contact input image for basic scan will be delayed by one scan.

Figure 7.1.3-4 Image of Contact Input Data Processing : PFCS

For contact output, the function blocks output data via the process I/O image, which is in the processor unit main memory, to contact output modules.

The diagram below illustrates the digital data output processing.

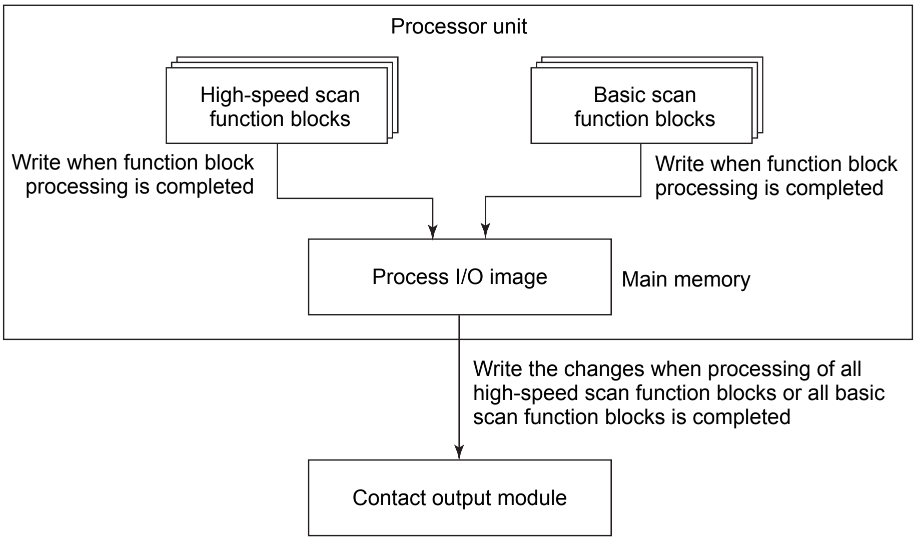


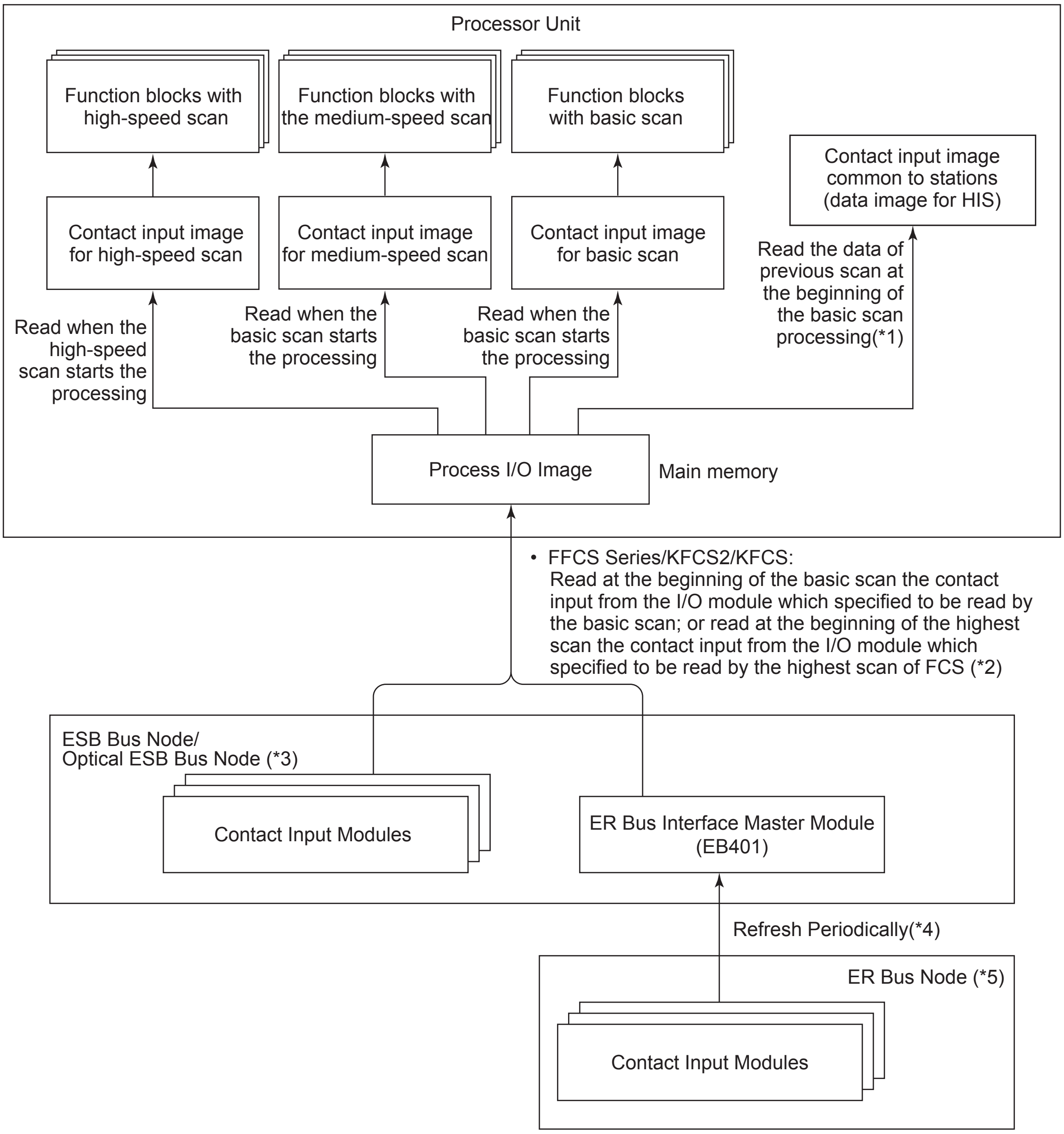
Figure 7.1.3-5 Image of Contact Output Data Processing : PFCS

● Contact I/O Data : FFCS Series/KFCS2/KFCS

For contact input, the contact input module and the function block perform data input processing via the process I/O image in the main memory of the processor unit and the contact input image of the corresponding scan period stored in the data buffer areas.

Since the contact input data are stored in the image area, the function blocks which are operated in the same period use the contact input's image read at the beginning of that scan period.

The diagram below shows an image of contact input data processing:



*1: The contact input image common to stations (data image for HIS) will be updated at the beginning of the next basic scan so as to match the data recognized by the function blocks in the FCS at the previous scan (before updating the process I/O image). Therefore, the contact input image for basic scan will be delayed by one scan.
*2: The Highest Scan of FCS means that if the FCS is applied with high-speed scan, the High-Speed Scan is the highest scan of FCS, and otherwise the Medium-Speed Scan is. If the FCS is applied with basic scan only, the Basic Scan is the highest scan of FCS.
*3: The input processing of the analog I/O modules and EB401 module inserted in the slots of FFCS series FCU are the same as the processing of the modules in the ESB bus node/Optical ESB bus node. Note that the Optical ESB bus nodes can be connected to only the FFCS-V.
*4: The period of contact I/O data refresh between EB401 and ER bus node varies with the number of ER bus nodes connected.
*5: The ER bus nodes cannot be connected to the FFCS-V.

Figure 7.1.3-6 Image of Contact Input Data Processing : FFCS Series/KFCS2/KFCS

The period of data refresh between EB401 and ER bus nodes varies with the number of ER bus nodes connected. The data refresh period corresponds to the number of nodes are shown as follows.

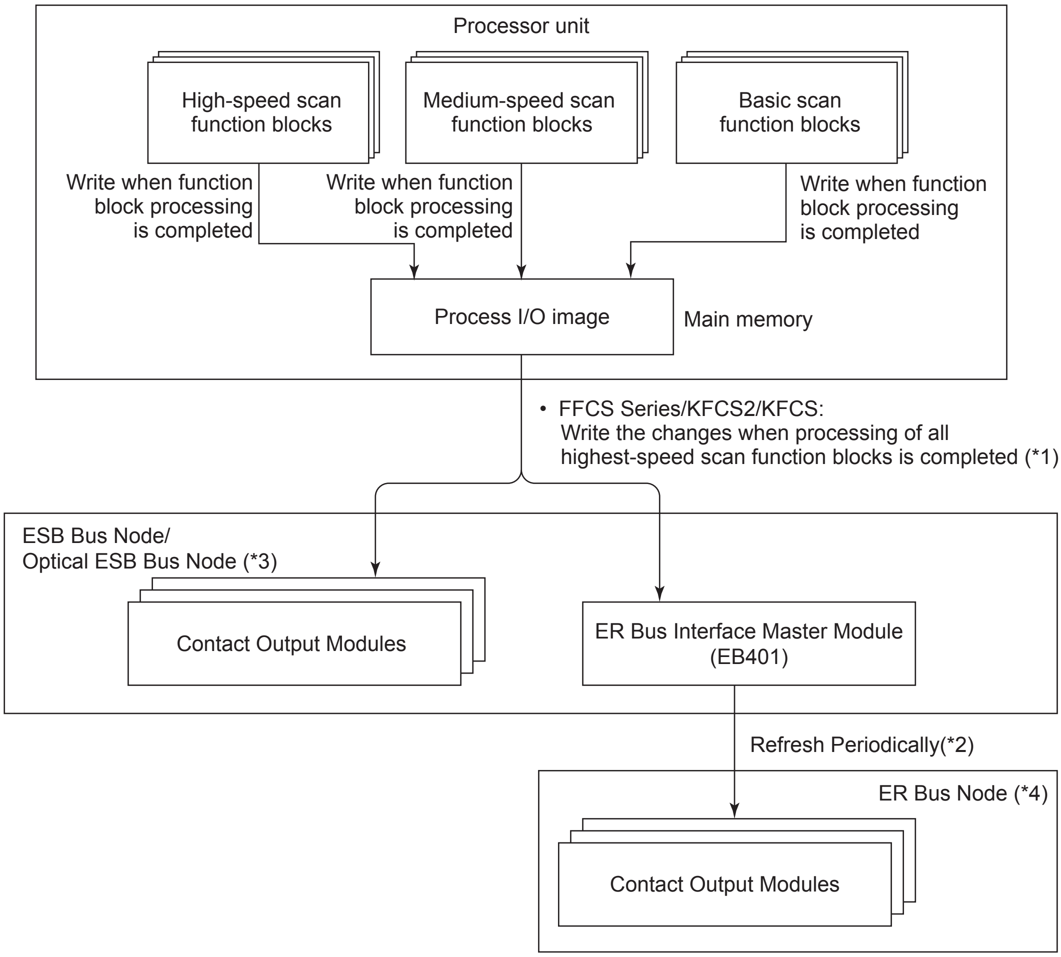
Table 7.1.3-2 Data Refresh Period between EB401 and Input/Output Module: FFCS/FFCS-L/KFCS2/KFCS

	Number of ER Bus Nodes		
	2	**4**	**6 (*1)**
Status Input	50 ms	100 ms	200 ms
ST Compatible (16-Point Input)	50 ms	100 ms	200 ms
Pushbutton Input	100 ms	200 ms	400 ms
ST Compatible (32-Point Input)	100 ms	200 ms	400 ms
ST Compatible (64-Point Input)	200 ms	400 ms	800 ms

*1: For FFCS-L/KFCS2/KFCS.

For contact output, the function blocks output data via the process I/O image, which is in the processor unit main memory to the I/O modules.

The diagram below illustrates the contact output data processing.



*1: The Highest Scan of FCS means that if the FCS is applied with high-speed scan, the High-Speed Scan is the highest scan of FCS, and otherwise the Medium-Speed Scan is. If the FCS is applied with basic scan only, the Basic Scan is the highest scan of FCS.
*2: The period of contact I/O data refresh between EB401 and ER bus node varies with the number of ER bus nodes connected.
*3: The output processing of the digital output modules and EB401 module inserted in the slots of FFCS series FCU are the same as the processing of the modules in the ESB bus node/Optical ESB bus node. Note that the Optical ESB bus nodes can be connected to only the FFCS-V.
*4: The ER bus nodes cannot be connected to the FFCS-V.

Figure 7.1.3-7 Image of Contact Output Data Processing : FFCS Series/KFCS2/KFCS

The period of data refresh between EB401 and ER bus nodes varies with the number of ER bus nodes connected. The data refresh period corresponds to the number of nodes are shown as follows.

Table 7.1.3-3 Data Refresh Period between EB401 and Input/Output Module : FFCS/FFCS-L/KFCS2/KFCS

	Number of ER Bus Nodes		
	2	4	6 (*1)
Status Output	50 ms	100 ms	200 ms
Pulse-Width Output (*2)	100 ms	200 ms	400 ms
ST Compatible (64-Point Output)	200 ms	400 ms	800 ms

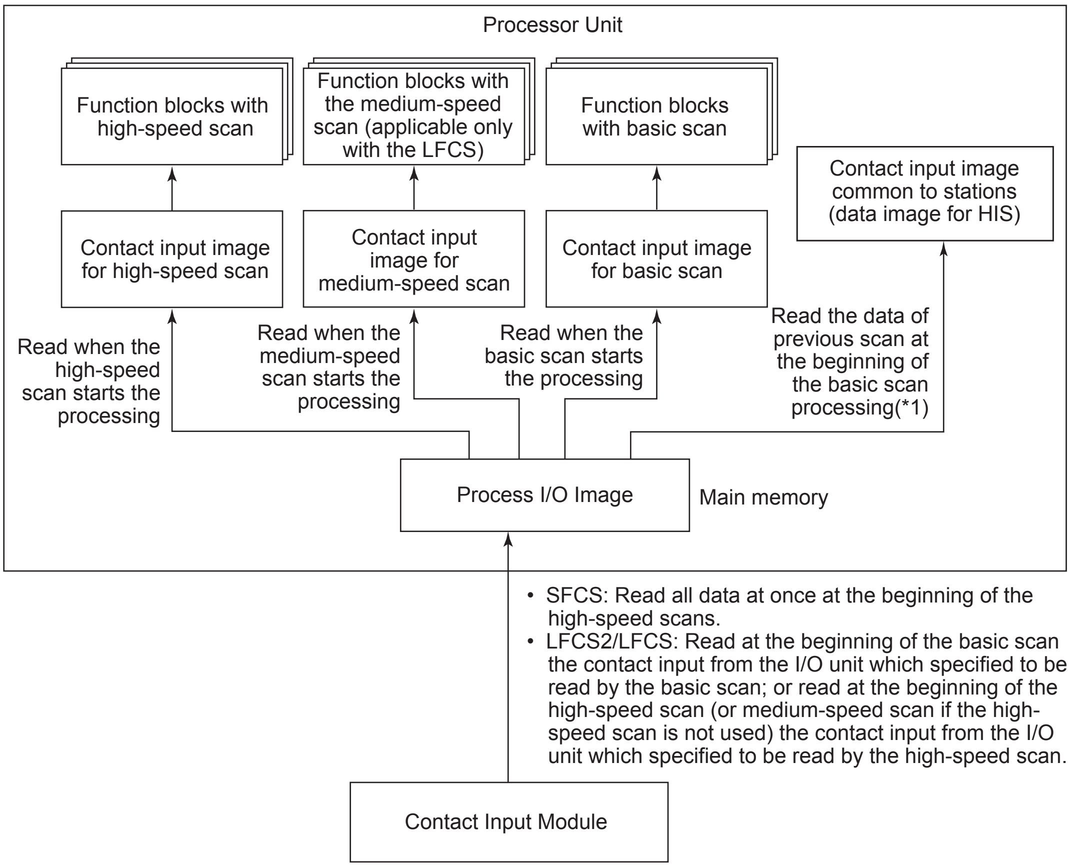
*1: For FFCS-L/KFCS2/KFCS.
*2: The pulse-width output between Input/Output Module and field devices behaves differently, the output timing is controlled in the output module.

● Contact I/O Data : LFCS2/LFCS/SFCS

For contact input, the contact input module and the function block perform data input processing via the process I/O image in the main memory of the processor unit and the contact input image of the corresponding scan period stored in the data buffer areas.

Since the contact input data are stored in the image area, the function blocks which are operated in the same period use the contact input's image read at the beginning of that scan period.

The diagram below shows an image of digital data input processing:



*1: The contact input image common to stations (data image for HIS) will be updated at the beginning of the next basic scan so as to match the data recognized by the function blocks in the FCS at the previous scan (before updating the process I/O image). Therefore, the contact input image for basic scan will be delayed by one scan.

Figure 7.1.3-8 Image of Contact Input Data Processing : LFCS2/LFCS/SFCS

For contact output, the function blocks output data via the process I/O image, which is in the processor unit main memory, to contact output modules.

The diagram below illustrates the contact output data processing.

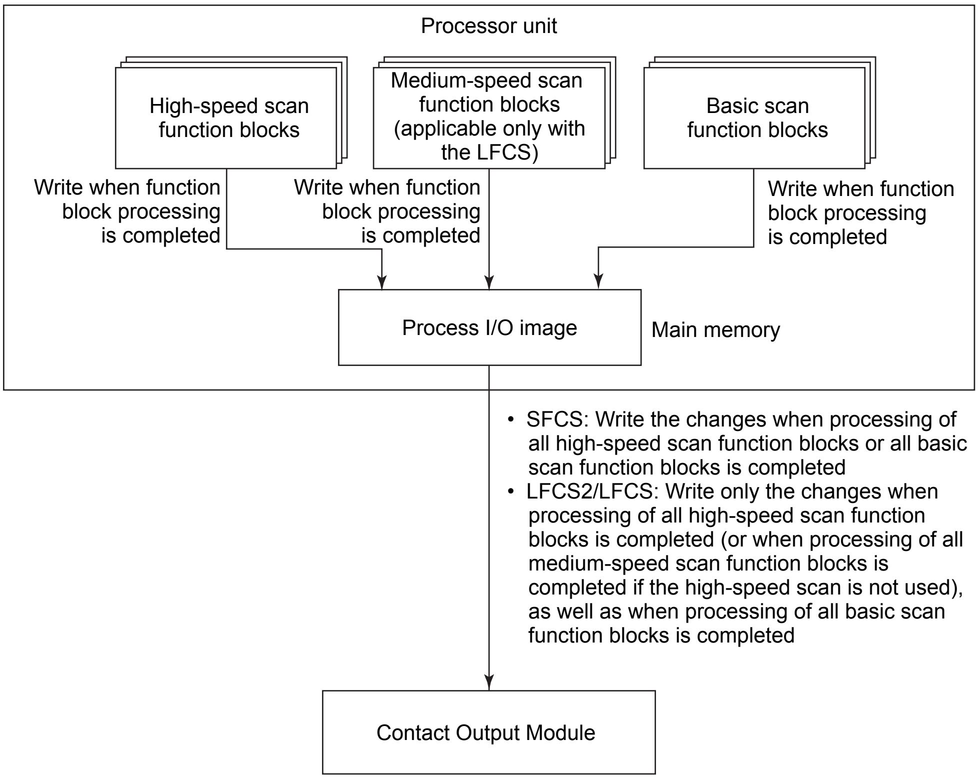


Figure 7.1.3-9 Image of Contact Output Data Processing : LFCS2/LFCS/SFCS

■ Timing of Process I/O : PFCS

The following section describes the timing of process I/O.

● Input Timing : PFCS

The input signals are read from the input module to the process I/O image in the processor unit, then to the contact input image in which they are grouped according to their scan period in the data buffer area. All the input data are accessed all together at the beginning of each “high-speed processing” scan.

The accessed data may be applied to either high scanned function blocks or basic scanned function blocks.

The function blocks perform their input and calculation process to the data they read from the process I/O image.

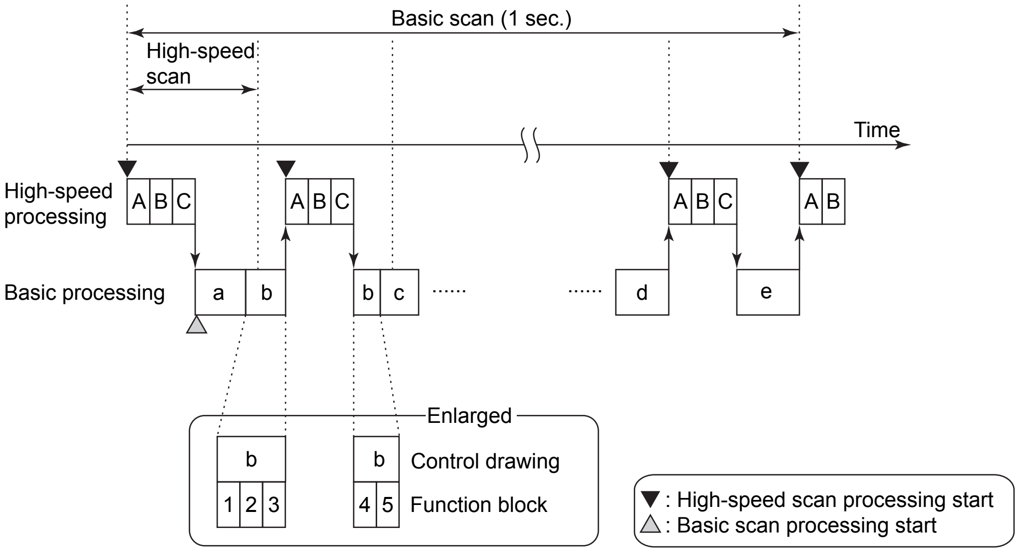


Figure 7.1.3-10 Input Timing : PFCS

● Output Timing : PFCS

The function blocks write output data to the specified area in the process I/O image when the function blocks are executed. Of the data written to the process I/O image, only the changes are written to the output module at the following timing:

- The contact output data is written when processing of all high-speed scan function blocks or all basic scan function blocks is completed.
- The analog output data are written at the end of each function block's processing.

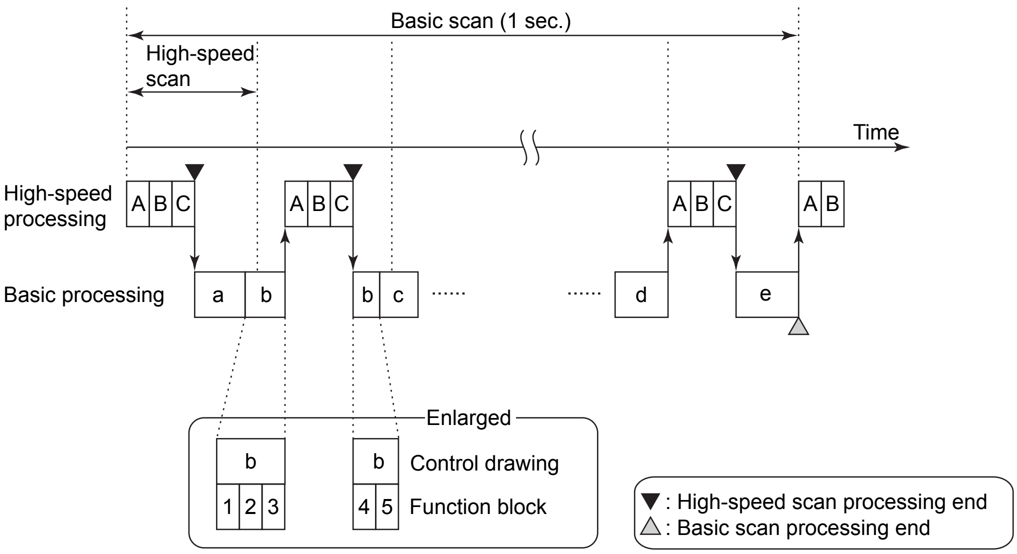


Figure 7.1.3-11 Output Timing : PFCS

■ Timing of Process I/O

Timing of Process I/O is explained in the following section.

● Input Timing : FFCS Series/KFCS2/KFCS

The processor unit reads input data from the input modules installed in ESB bus nodes/Optical ESB bus nodes (*1) and/or FFCS series FCU, and/or from the ER bus interface master modules (EB401) and then writes the data to the memory with the following timing:

The input data of ER bus nodes(*2) are acquired periodically by EB401. The input data from EB401 to memory are the data that EB401 periodically acquired from the ER bus nodes (*2).

- For the I/O modules not defined with high-speed scan, data are accessed at the beginning of basic processing scan period.
- For the I/O modules defined with high-speed scan or medium-speed scan, data are accessed at the beginning of highest processing scan period.

*1: The Optical ESB bus nodes can be connected to only the FFCS-V.
*2: The ER bus nodes cannot be connected to the FFCS-V.

The highest scan period means that if the FCS is applied with High-Speed scan, the High-Speed scan is the highest scan of FCS, and otherwise the Medium-Speed scan is. If the FCS is applied with basic scan only, the basic scan is the highest scan period.

Moreover, for contact input data, the data are sent from the process control I/O image tables to the contact input image tables categorized in accordance with the scan periods of the contact inputs. The timing to access the I/O data is at the beginning of the function block processing that corresponds to the scan periods.

Function blocks perform input processing, calculation processing to the analog data acquired from process I/O image. While the function blocks perform input processing, calculation processing to the contact input data acquired from the I/O images of various scan periods.

● Input Timing : LFCS2/LFCS/SFCS

The input signals are read from input module to the process I/O image in the processor unit, then to the contact input image in which they are grouped according to their scan period in the data buffer area, at the following timing:

LFCS2 or LFCS:

- The input from the I/O module which specified to be read by the high-speed scan is read when the high-speed scan processing of the function block is started (or when the medium-speed scan processing is started if the high-speed scan is not used).
- The input from the I/O module which specified to be read by the basic scan is read when the basic scan processing of the function block is started.

SFCS:

- Access data at the beginning of each “high-speed processing” scan.
  The accessed data may be applied to either high scanned function blocks or basic scanned function blocks.

The function blocks perform their input and calculation process to the data they read from the process I/O image.

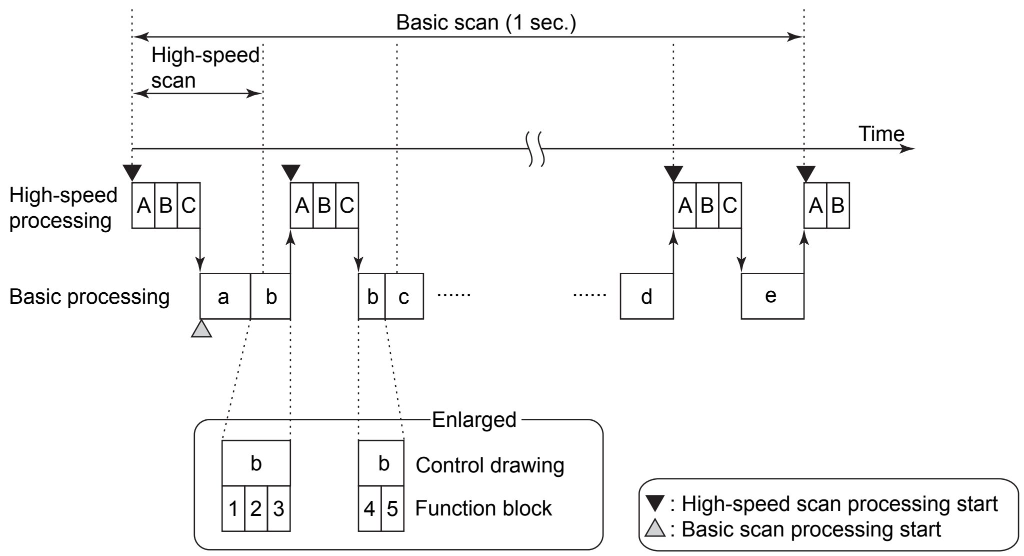


Figure 7.1.3-12 Input Timing

● Output Timing : FFCS Series/KFCS2/KFCS

▼ Output Type – Output in a Lump, Output Type – Output Immediately

Function blocks send their output data to process I/O image at the time of the function blocks performing output processing.

By comparing the outputs with the I/O memory image, the processor unit puts the difference to the output modules installed in ESB bus nodes/Optical ESB bus nodes(*1) and/or FFCS series FCU, and/or to the ER bus interface master modules (EB401) with the following timing:

The output data in the EB401 sent from process I/O image are periodically written to output modules of ER bus nodes(*2) .

*1: The Optical ESB bus nodes can be connected to only the FFCS-V.
*2: The ER bus nodes cannot be connected to the FFCS-V.

For analog output in FFCS series, KFCS2, and KFCS:

- The timings for writing the output data in process I/O image to I/O modules vary with the designated options [Output in a Lump] and [Output immediately].
  When option [Output in a Lump] is designated, the output data are written from process I/O image to I/O modules at the completion of the highest scan of function blocks.

  When option [Output Immediately] is designated, the output data are written from process I/O image to I/O modules right after the data are outputted from the function blocks.

For contact output in FFCS series, KFCS2, and KFCS:

- The contact output data are written from process I/O image to I/O modules at the completion of the highest scan of function blocks.

The highest scan period means that if the FCS is applied with High-Speed scan, the High-Speed scan is the highest scan of FCS, and otherwise the Medium-Speed scan is. If the FCS is applied with basic scan only, the basic scan is the highest scan period.

The output type may be specified on IOM property sheet.

Output Type: Selectable between “Output in a Lump” and “Output immediately.”
Default is “Output in a Lump”

IMPORTANT

When option [Output immediately] is designated, it takes 1 or 2 milliseconds of processor unit to perform the task of writing from process I/O image to I/O modules, thus the CPU load is added.
It is recommended to choose [Output in a Lump] option instead of choosing [Output immediately] option unless it is necessary. With [Output in a Lump] option, the output data can also be sent from process I/O image to I/O modules at high-speed scan period if the high-speed scan is specified.

TIP

For Flow-totalizing batch set block (BSETU-2) and Weight-totalizing batch set block (BSETU-3), if the output automatic prediction is applied, the output data are written from the function block to I/O modules or EB401 at the predicted time.
Furthermore, it may take up to 30 milliseconds more time for outputting to the I/O modules in the ER bus nodes.

● Output Timing : LFCS2/LFCS/SFCS

The function blocks write output data to the specified area in the process I/O image when being executed. Of the data written to the process I/O image, only the changes are written to the output module at the following timing:

LFCS2 or LFCS analog output/contact output:

- When processing of all high-speed scan function blocks (or all medium-speed scan function blocks if the high-speed scan is not used) is completed.
- When processing of all basic scan function blocks is completed.

SFCS analog output:

- When processing of each function block is completed.

SFCS contact output:

- When processing of all high-speed scan function blocks (or all medium-speed scan function blocks if the high-speed scan is not used) is completed.
- When processing of all basic scan function blocks is completed.

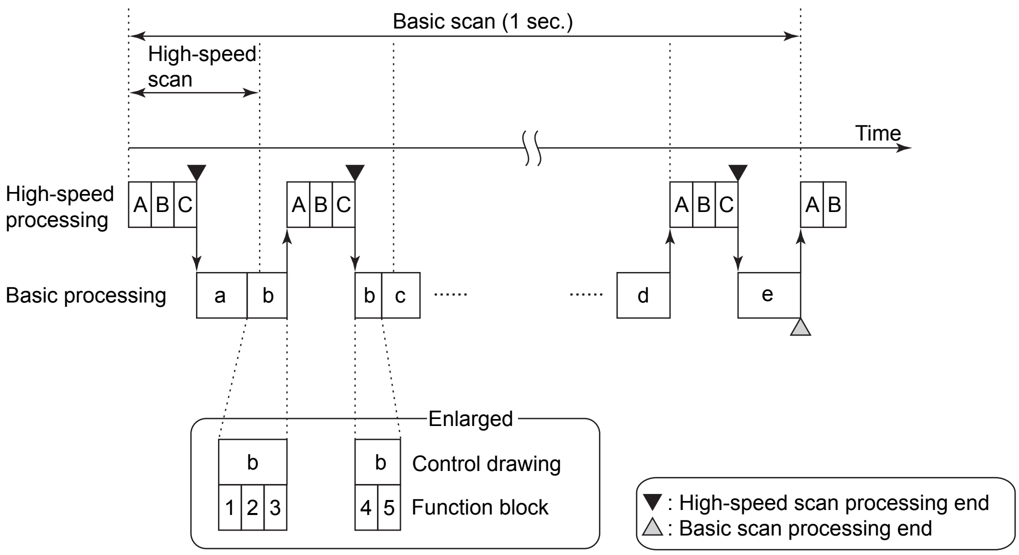


Figure 7.1.3-13 Output Timing : LFCS2/LFCS/SFCS

7.1.4 Control Period for Controller Block

Among the regulatory control blocks, the controller block has control period that is dependent on process timing. The control period of the controller block is a time period that the controller block executes control calculation and output processing during automatic operation (AUT, CAS, RCAS). The control period of the controller block is always an integer multiple of the scan period.

The only processing that the controller block always performs for each scan period is input processing and alarm processing. Control calculation and output processing are performed once per each control period.

■ Control Period of Controller Block

▼ Control Period

The controller block executes the input processing per scan period. However, the control calculation and output processing are executed per each control period.

The control period of controller block is always an integer-multiple of the scan period.

There are 2 types of the control periods of controller block as shown below:

- The control period of the regulatory control action.
- The control period of the intermittent control action.

● Control Period of Regulatory Control Action

Control calculation is executed at every control period in regulatory control action.

The figure below shows the controller block’s control period in the regulatory control action.

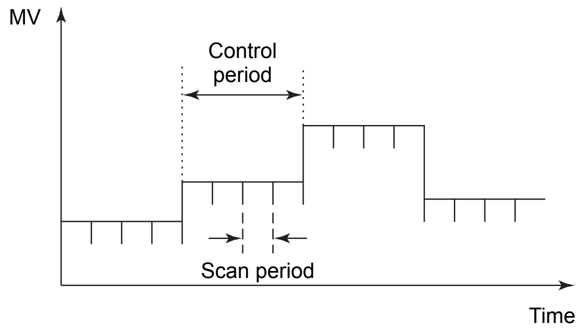


Figure 7.1.4-1 The Control Period in the Regulatory Control Action

● Control Period in Intermittent Control Action

In intermittent control action, control calculation and output processing are executed only once in the scan period in which the control switch (CSW) is turned ON during the automatic operation (AUT, CAS, RCAS). After the execution, the control switch (CSW) is turned OFF. The control switch (CSW) can be set to ON by other function blocks such as the sequence control blocks.

The figure below shows the control period of the controller block in the intermittent control action.

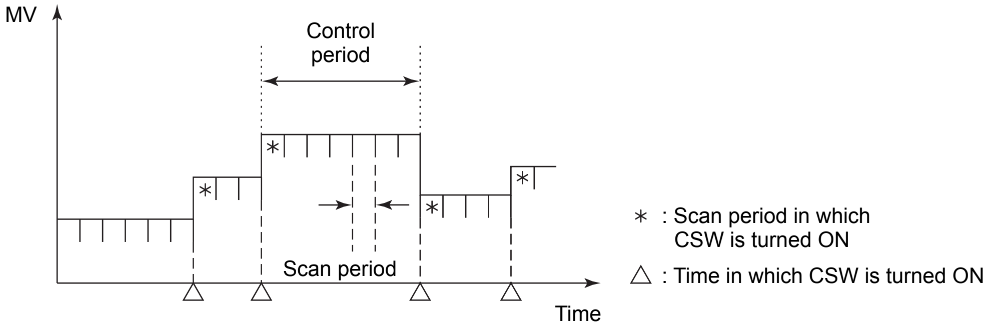


Figure 7.1.4-2 The Control Period of the Intermittent Control Action

The intermittent control action is used for the sampling control with a sampling value to be measured at the timing determined by outside of the controller block.

■ Setting for Control Period of Controller Block

The following section describes the method of setting the control period of a controller block, and how the function block behaves for the different settings.

● Setting for Control Period of Controller Block

Setting for control period of controller block can be defined in the Function Block Detail Builder.

- Control period
  Select from "fixed control period," "automatic determination" or "intermittent control action."

  In the case of the Blending PI Controller Block (PI-BLEND) or the sampling PI controller block (PI-HLD), select from "fixed control period" or "automatic determination."

  The default is "automatic determination."

TIP

The control period for the two-position ON/OFF controller block (ONOFF), Enhanced Two-Position ONOFF Controller Block (ONOFF-E), three-position ON/OFF controller block (ONOFF-G), Enhanced Three-Position ONOFF Controller Block (ONOFF-GE) and the PD Controller Block with Manual Reset (PD-MR) is the same as the scan period.

The fixed control periods is selected from the following:

1, 2, 4, 8, 16, 32 or 64 seconds.

When the automatic determination is selected, the control period is decided by the following rules:

Table 7.1.4-1 Control Period at the Automatic Determination

Integral Time (Sec.)	Control Period (Sec.)	
	Basic Scan	High-Speed Scan
1 to 31	1	Same as the scan period
32 to 63	2	
64 to 255	4	
256 to 1023	8	
1024 to 2047	16	
2048 to 10000	32	

When the fixed control period or the automatic determination is selected, it is operated with the control period of the regulatory control action. When the intermittent control action is selected, it is operated with the control period of the intermittent control action.

● Action of Controller Block Based on Control Period

Based on the control period defined, the controller blocks are executed as follows:

- When a fixed control period is selected
  The control calculation processing and output action during the automatic operation (AUT, CAS, RCAS) are executed with the preset fixed control period.
- When the automatic determination is selected
  If the scan period is the basic scan, the control period is determined automatically according to the parameter of the integral time (I). If the scan period is the high-speed scan, the control period is the same as the scan period.
- When an intermittent control action is selected
  The control calculation and output processings are executed only once with the scan period in which the control switch (CSW) is turned ON during the automatic operation (AUT, CAS, RCAS).

7.2 Process Timing of Calculation Block

The process timing for the calculation block has the following two execution types according to execution timing.

- Periodic Execution
- One-Shot Execution

■ Periodic Execution

▼ Start Timing

The periodic execution means that the calculation block is repeatedly executed in a preset cycle. The function blocks for which a periodic execution is defined is referred as the periodic-execution type function block.

Normally, the periodic-execution type function block is executed per scan period.

The timing that the control drawings and the individual function blocks are executed by the periodic execution of calculation block is determined by following factors.

● Scan Period

The scan period is a period of periodic execution for a function block. The periodic execution function block executes a process based on the scan period.

There are three types of scan periods: basic scan, medium-speed scan(*1) and high-speed scan. Select one of these scan periods for each function block. However, the medium-speed scan (*1) and high-speed scan cannot be selected for some function blocks.

*1: The medium-speed scan setting is available only for FFCS series, KFCS2, KFCS, LFCS2, and LFCS.

SEE ALSO For more information about the scan period, refer to:

7.1.1, “Scan Period” on page 7-4

● Order of Process Execution

The order of process execution refers to a sequence in which the control drawing and individual function block are executed in the periodic execution. This order of process execution determines the execution timing within a scan period of the control drawing and individual function block.

SEE ALSO For more information about the order of process execution, refer to:

7.1.2, “Order of Process Execution” on page 7-6

● Timing of Process I/O

The timing of the process input/output refers to a timing at which data I/O is executed between the function block and the process I/O.

The timing of the process I/O differs by whether the input/output data is analog or contact type.

SEE ALSO For more information about the process input/output timing, refer to:

7.1.3, “Timing of Process I/O” on page 7-8

■ One-Shot Execution

In one-shot execution, the calculation block is executed only once when it is invoked by other function blocks. The function block defined with one-shot execution is referred as the one-shot-execution (O) type function block.

When an one-shot function block is invoked by other function blocks in the same FCS, it starts its own process with interrupting the process from which it was invoked. When the one-shot block's process is completed, it hands back the process to the block from which it was invoked. The sequence control block cannot be invoked from other control stations.

The following diagram shows the one-shot processing:

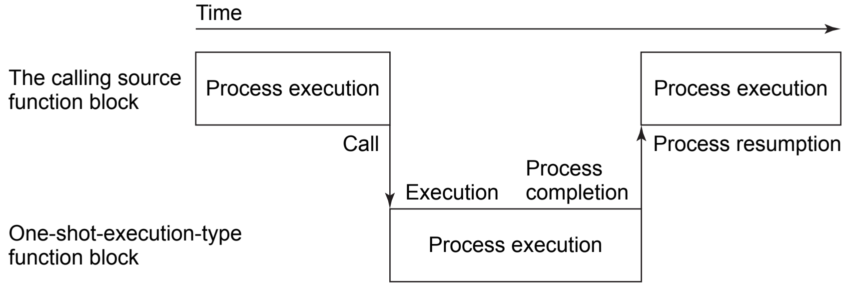


Figure 7.2-1 One-Shot Execution Conceptual Diagram

The one-shot execution is used when a calculation block is invoked from a sequence table block.

The following table lists the calculation blocks where one-shot start is possible.

Table 7.2-1 Calculation Blocks in Which One-Shot Execution is Possible

Block type	Type	Name
Arithmetic calcula-tions	ADD	Addition Block
	MUL	Multiplication Block
	DIV	Division Block
	AVE	Averaging Block
Logic operation (*1)	AND	Logical AND Block
	OR	Logical OR Block
	NOT	Logical NOT Block
	SRSI-S	Set-Dominant Flip-Flop Block with 1 Output
	SRSI-R	Reset-Dominant Flip-Flop Block with 1 Output
	SRS2-S	Set-Dominant Flip-Flop Block with 2 Outputs
	SRS2-R	Reset-Dominant Flip-Flop Block with 2 Outputs
	WOUT	Wipeout Block
	GT	Comparator Block (Greater Than)
	GE	Comparator Block (Greater Than or Equal)
	EQ	Equal Operator Block
	BAND	Bitwise AND Block
	BOR	Bitwise OR Block
	BNOT	Bitwise NOT Block

Block type	Type	Name
General-purpose calculations	CALCU	General-Purpose Calculation Block
	CALCU-C	General-Purpose Calculation Block (with character string data I/O terminal)
Auxiliary calculations	BDSET-1L	One-Batch Data Set Block
	BDSET-1C	One-Batch String Data Set Block
	BDSET-2L	Two-Batch Data Set Block
	BDSET-2C	Two-Batch String Data Set Block (strings only)
	BDA-L	Batch Data Acquisition Block
	BDA-C	Batch String Data Acquisition Block

*1: Logic Operation Block can be used in FCSs except PFCS.

TIP

When a calculation block is executed based on one-shot specification, the alarm status is not updated during the one-shot processing.

■ Setting for Processing Timing

Setting for processing timing can be defined in the Function Block Detail Builder.

Execution timing: Select from “0 (periodic execution)” or “1 (one-shot execution).”
The default is 0.

When the periodic execution is selected for execution timing, set a scan period in the function block detail definition builder. There are three types of scan periods: basic scan, medium-speed scan (*1) and high-speed scan.

*1: The medium-speed scan setting is available only for FFCS series, KFCS2, KFCS, LFCS2, and LFCS.

7.3 Process Timing for Sequence Control Block

This section explains the process timing of a sequence control block.

Process timing for a sequence table block (ST16, ST16E) and logic chart block (LC64, LC64-E) includes the execution timing, output timing subordinated to execution timing, control period, and control phase.

■ Start Timing of Sequence Control Block

▼ Processing Timing

A sequence control block and a logic chart block have the following four types of execution timing:

- Periodic Execution (T)
- One-Shot Execution (O)
- Initial Execution/Restart Execution (I)
- Restricted Initial Execution (B)

■ Output Timing of Sequence Control Block

The output timing of sequence control block and logic chart block indicates the conditions to execute the output processing when the sequence table is started periodically or as a one shot.

There are two types of output timing as below:

- Output only when conditions change (C)
- Output each time conditions are satisfied (E)

The output timing of function blocks excluding sequence control blocks is “Output each time conditions are satisfied (E).”

■ Combining Execution Timing and Output Timing

The execution timing and the output timing can be used in combination.

■ Control Period and Control Phase of a Sequence Control Block

The control period is a period at which Periodic Execution Type (T) sequence tables and logic charts are executed. The control phase is the timing for performing sequence table processing during a control period.

The control period and control phase of a sequence table and a logic chart are used when the sequence table and a logic chart are executed using a period longer than the basic scan for periodic execution.

7.3.1 Execution Timing for Sequence Control Blocks

The sequence control block is executed at the timing defined, then it performs data input processing. After the input processing, it executes processes such as the control calculation processing and output processing.

There are four types of execution timing to start the sequence control block

- Periodic Execution
- One-Shot Execution
- Initial Execution/Restart Execution
- Restricted Initial Execution

■ Periodic Execution (T)

▼ Scan Period

The periodic execution means that the sequence control block is repeatedly executed in a preset cycle. The function block for which a periodic execution is defined is referred as the periodic-execution (T) type function block.

The timing that the control drawings and individual function blocks are executed by the periodic execution is determined by following factors.

● Scan Period

The scan period is a time period of periodic execution for a function block. The periodic execution function block executes a process based on the scan period.

There are three types of scan periods: basic scan, medium-speed scan(*1) and high-speed scan. In the periodic-execution-type sequence table block and logic chart block, the basic scan, medium-speed scan (*1) or high-speed scan can be selected as the scan period. Note that basic scan, not medium-speed scan or high-speed scan, should be set for the switch instrument block, enhanced switch instrument block, and VLVM block.

*1: The medium-speed scan setting is available only for FFCS series, KFCS2, KFCS, LFCS2, and LFCS.

SEE ALSO For more information about the scan period, refer to:

7.1.1, “Scan Period” on page 7-4

● Order of Process Execution

The order of process execution refers to a sequence in which the control drawing and individual function block are executed in the periodic execution. The execution timing within a scan period of the control drawing and the individual function block is determined by the order of the execution.

SEE ALSO For more information about the order of process execution, refer to:

7.1.2, “Order of Process Execution” on page 7-6

● Timing of Process I/O

The timing of process input/output refers to a timing at which data input/output is executed between the function block and the process I/O.

The timing of process I/O differs by whether the input/output data is analog or contact type.

SEE ALSO For more information about the process input/output timing, refer to:

7.1.3, “Timing of Process I/O” on page 7-8

■ One-Shot Execution (O)

In one-shot execution, sequence block is executed only once when it is invoked by other function blocks. The function block defined with one-shot execution is referred as the one-shot-execution (O) type function block.

When an one-shot function block is invoked by other function blocks in the same FCS, it starts its own process with interrupting the process from which it was invoked. When the one-shot block’s process is completed, it hands back the process to the block from which it was invoked. The sequence control block cannot be invoked from other control stations.

The following diagram shows the one-shot processing:

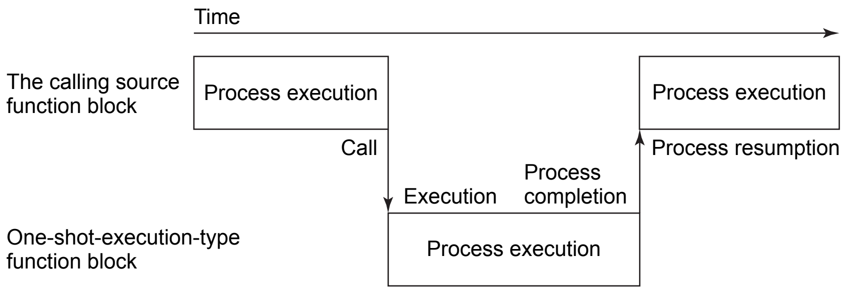


Figure 7.3.1-1 One-Shot Processing Conceptual Diagram

A one-shot function block can invoke another one-shot function block. However, such succession is limited to seven blocks.

The one-shot processing is used when a function block is executed from a sequence control block.

■ Initial Execution/Restart Execution (I)

In initial execution/restart execution, the sequence block executes its process when the FCS performs a cold start or a restart. The function blocks only work in such timing are referred as the initial-cold start/restart execution (I) style function block.

■ Restricted Initial Execution (B)

In restricted initial execution (B), the sequence control block executes only when the FCS performs a cold start, not include restart. The function blocks only work in such timing are referred as the initial-cold start (B) style function block.

■ Setting for Execution Timing

Setting for execution timing can be defined in the Function Block Detail Builder.

- Execution Timing
  Select from the “Periodic Execution (T),” “One-Shot Execution (O),” “Initial Execution/Restart Execution (I),” or “Restricted Initial Execution (B).”

When the periodic execution (T) is selected for execution timing, set a scan period in the Function Block Detail Builder.

7.3.2 Output Timing of Sequence Table Blocks (ST16, ST16E)

The output timing of a sequence table block indicates the conditions under which output processing is performed when the sequence table is executed periodically or as a one shot.

The two types of output timing are given below.

- Output only when conditions are changed (C)
- Output each time conditions are satisfied (E)

■ Output Only when Conditions are Changed (C)

The ST16, ST16E blocks output an operation signal only at the timing when the judged conditions are changed from unsatisfied to satisfied. The "output only when conditions are changed" can only be specified for ST16, ST16E blocks with periodic execution (T) or one-shot start (O).

■ Output Each Time Conditions are Satisfied (E)

The ST16, ST16E blocks output an operation signal every scan period as long as the judged conditions are satisfied.

■ Setting for Execution Timing

The output timing can be defined in the Function Block Detail Builder.

Output timing: Select from "Output only when conditions are change" or "Output each time conditions are satisfied."

7.3.3 Output Timing of Logic Chart Blocks (LC64, LC64-E)

Output timing of logic chart block shows output conditions when the logic chart block starts by Periodic/One-shot Execution. “Each Time Conditions are Satisfied (E)” can only be selected as output timing.

■ Output Each Time Conditions are Satisfied (E)

The LC64, LC64-E blocks output operation signals by scan period if the specified conditions are satisfied.

IMPORTANT

- Manipulated output is sent from the logic chart block every cycle. If the output data (ex. Printout messages) is not necessary every cycle, change output timing to output only when logical operators are changed.
- When the logic chart block starts by one-shot processing, do not use time data in internal logical operation. For example, if on-delay timer were used, output is in the initial state.
- Internal logic operators are reset upon recovery from O/S upon online maintenance.

7.3.4 Combination of Execution Timing and Output Timing

The following section describes combinations of the execution and output timings.

■ Combination of Execution Timing and Output Timing of Sequence Table Blocks (ST16, ST16E)

The execution timing and the output timing of the sequence table blocks can be used with the following.

No other combination is allowed. The default is TC.

Table 7.3.4-1 Combination of Execution Timing and Output Timing of Sequence Table Blocks

Execution Timing	Output Timing	Symbol
Periodic Execution (T)	Conditional Output (C)	TC
	Each time Output (E)	TE
One-Shot Execution (O)	Conditional Output (C)	OC
	Each time Output (E)	OE
Startup at Initial Cold Start/Restart (I)	-	I
Restricted Initial Execution (B)	-	B

■ Combination of Execution Timing and Output Timing of Logic Chart Blocks (LC64, LC64-E)

The execution timing and the output timing of the logic chart blocks can be used with the following. No other combination is allowed. The default is T.

Table 7.3.4-2 Combination of Execution Timing and Output Timing of Logic Chart Blocks

Execution Timing	Output Timing	Symbol
Periodic Execution (T)	Each time (E)	TE
One-Shot Execution (O)	Each time (E)	OE
Startup at Initial Cold Start/Restart (I)	-	I
Restricted Initial Execution (B)	-	B

7.3.5 Control Period and Control Phase for Sequence Table Blocks (ST16, ST16E)

For the periodic execution (T) type ST16 and ST16E blocks, the control period and control phase can be set by the Function Block Detail Builder. However, the control period and control phase can only be set when the scan period is set to the basic scan.

■ Control Period for Sequence Table Blocks (ST16, ST16E)

The control period for the ST16, ST16E blocks refers to the interval at which the periodic-execution-type ST16 or ST16E block executes the sequence table.

The control period can be set in the Function Block Detail Builder.

Control period: Set a value between 1 and 16 seconds.
Default is 1 second.

■ Control Phase for Sequence Table Blocks (ST16, ST16E)

The control phase for the ST16, ST16E blocks refers to the timing at which the sequence table is executed in the control period. It sets the execution timing relative to the execution timing of the phase-zero sequence table.

The control phase can be set in the Function Block Detail Builder.

Control phase: Set a value between 0 and 15 seconds.
Default is 0 second.

■ Example of Control Period and Control Phase for Sequence Table Blocks (ST16, ST16E)

For example, in case of that a ST16 or ST16E block's control period is 5 sec. and control phase is 3 sec., the ST16 or ST16E block is executed at every 5 seconds interval, 3 seconds after the phase zero's ST16 or ST16E block.

The following execution timing diagram shows the execution timing in case of the control period and control phase are set.

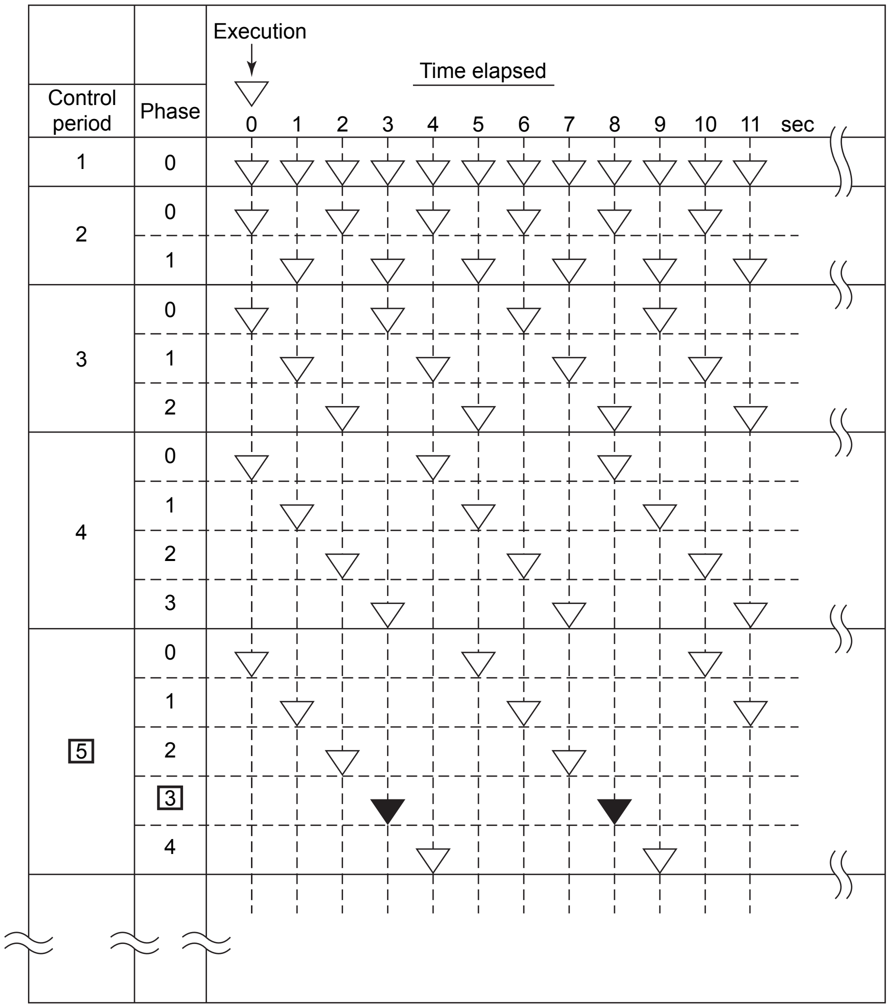


Figure 7.3.5-1 Control Period and Phase for Sequence Table Block (ST16, ST16E)

7.3.6 Control Period and Control Phase for Logic Chart Blocks (LC64, LC64-E)

For the periodic execution (T) type LC64 and LC64-E blocks, the control period and control phase can be set by the Function Block Detail Builder. However, the control period and control phase can only be set when the scan period is set to the basic scan.

■ Control Period for Logic Chart Blocks (LC64, LC64-E)

The control period for the LC64, LC64-E blocks refers to the interval at which the periodic-execution-type LC64 or LC64-E block executes the sequence table.

The control period can be set in the Function Block Detail Builder.

Control Period: Set a value between 1 to 16 seconds.
The default is 1.

■ Control Phase for Logic Chart Blocks (LC64, LC64-E)

The control phase for the LC64, LC64-E blocks refers to the timing at which the sequence table is executed in the control period. It sets the execution timing relative to the execution timing of the phase-zero sequence table.

The control phase can be set in the Function Block Detail Builder.

Control Phase: Set a value between 0 to 15 seconds.
The default is 0.

7.4 SEBOL Process Timing

SEBOL (SEquence and Batch Oriented Language) is a programming language oriented for sequential controls. There are two types of SEBOL vary with processes timings. They are respectively referred to as Periodic SEBOL and Timeshare SEBOL.

TIP

Periodic SEBOL programs are available in FFCS-V only.

SEE ALSO

For more information about SEBOL, refer to:

- 1.2, “Action Description Using SEBOL” in the Function Blocks Reference Vol.3 (IM 33K03E24-50E)
- 1., "Basics of SEBOL" in the SEBOL Reference (IM 33K03K10-50E)

■ Timeshare SEBOL

The timeshare SEBOL programs are processed at the CPU idle time; it is irrelevant to the high speed, medium speed, and basic scan processing periods. When multiple SFC blocks and operations exist, every 20 lines of the scripts in the SEBOL programs in the SFC blocks and operations will be sequentially executed at CPU idle time. For executing the tasks to access the other FCS data or running for…next (without @) loops and so on, the program may be interrupted even when 20 lines are not yet completed and then to be executed at the next scan period.

● Processing at CPU Idle Time

▼ SEBOL/User C Ratio[%]

The processing time shared by the timeshare SEBOL programs can be defined by the setting item of [SEBOL/User C Ratio[%]] on the FCS Constants Builder.

By default, this setting is set to 100 %. 100 % ratio means the 100 % of shares of the FCS CPU idle time is assigned for the timeshare SEBOL programs.

■ Periodic SEBOL: FFCS-V

▼ SEBOL Execution Type

The periodic SEBOL programs are basically executed at each basic scan.

The SFC blocks and operations that consist of periodic SEBOL programs are executed at each basic scan. If a step of SFC consists of more than 2000 lines of SEBOL scripts, the execution will be interrupted and then executed again at the next scan. Moreover, for executing the tasks to access the other FCS data or running for…next (without @) loops and so on, the tasks may be interrupted and then executed again at the next scan.

● Function Blocks that may Contain Periodic SEBOL Programs: FFCS-V

The function blocks that the contained SEBOL programs may be designated to periodic type are the following SFC blocks and operations.

- _SFCSW (3-Position Switch SFC Block)
- _SFCPB (Pushbutton SFC Block)
- _SFCAS (Analog SFC Block)
- OPSFC (SFC-Type Operation)
- OPSBL (SEBOL-Type Operation)
- OPSFCP1 (SFC-Type Operation with Floating-Data Parameters)
- OPSFCP2 (SFC-Type Operation with Character-Data Parameters)

- OPSFCP3 (SFC-Type Operation with Floating/Character-Data Parameters)
- OPSFCP4 (SFC-Type Operation with Integer/Character-Data Parameters)
- OPSFCP5 (SFC-Type Operation with Floating/Integer-Data Parameters)

Process Timing: FFCS-V

The SEBOL process timing of each function block is defined on the Basic tab of Function Block Detail Builder.If [PERIODIC: Periodic Execution] is selected for [SEBOL Execution Type] setting item on the Basic tab, the SEBOL programs in the corresponding SFC blocks or operations will be executed periodically. By default, [TIMESHARE: Timeshare Execution] is selected for [SEBOL Execution Type] setting item.

Periodic SEBOL Action after Step Transition : FFCS-V

For periodically executed SEBOL programs, after a step transition whether to immediately proceed or wait for one scan cycle can be specified on the [Basic] tab of function block details builder.

- After step transition [One cycle wait]:
  If [One cycle wait] is selected, after the step transition, the periodic SEBOL programs will wait for one scan cycle and then proceed.
- After step transition [Continue]:
  If [Continue] is selected, after the step transition, the periodic SEBOL programs will immediately proceed.

Number of Accessing Other Stations: FFCS-V

Accessing other station means to access the function block data of other FCSs, and to use signal statement, qsigcancel statement and qsigmove statement to communicate with other stations.

The limit for number of accessing other stations is 64 for all the periodic SEBOL programs per each basic scan. The 65th access from the SFC blocks or operations will be interrupted and executed at the next scan.

The above mentioned 64 accesses are not counting the accesses to the data in host station, and the accesses to the data of other stations from timeshare SEBOL programs. When timeshare SEBOL programs access the data of other stations, it is irrelevant to the limit of 64 accesses. The SEBOL statements that are subjected to the 64-accesse limit are as follows:

- Assignment Statement
- Group assignment statement
- “signal” statement
- “compare” statement (*1)
- “drive” statement
- Besides the above statements, all the SEBOL scripts destination to the tags of other stations.

*1: If either the right side or the left side item to be compared is a function block data of other stations, it is counted as two accesses.

Number of Accessing Common Blocks of Other Stations: FFCS-V

Accessing common blocks of other station means to access the common block data of other FCSs.

The limit for number of accessing common blocks of other stations is 16 for all the periodic SEBOL programs per each basic scan. The 17th access from the SFC blocks or operations will be interrupted and executed at the next scan.

The above mentioned 16 accesses are not counting the accesses to the common blocks of the host station and the accesses to the common blocks of other stations from timeshare SEBOL programs. When timeshare SEBOL programs access the data of other stations, it is irrelevant to the limit of 16 accesses.

■ Interrupting and Restarting SEBOL Programs

If SEBOL programs contain the tasks to access the other FCS data or running for…next (without @) loops and so on, the program may be interrupted and then restarted at the next scan period.

A SEBOL program may be interrupted by SEBOL statements and other reasons.

The causes that may interrupt and restart the timeshare and periodic SEBOL programs are listed below:

Table 7.4-1 SEBOL Statements Related to Interruption and Restart of SEBOL Programs

Cause of Interruption	Timeshare SEBOL	Periodic SEBOL
for...next@	Continues until all loops are completed.	
while...wend@	Continues until all loops are completed.	
repeat...until@	Continues until all loops are completed.	
for...next	Interrupts when 1 loop is completed and restarts at next scan.	
while...wend	Interrupts when 1 loop is completed and restarts at next scan.	
repeat...until	Interrupts when 1 loop is completed and restarts at next scan.	
goto	Continues	Interrupts and then restarts at next scan.
Delay	Interrupts and then restarts at next scan.	
Delaycycle	Interrupts and then restarts at next scan.	
Drive	Interrupts and then restarts at next scan.	
drive vpmon	Interrupts and then restarts at next scan.	
drive vpoff	Interrupts and then restarts at next scan.	
seqtable drivewait	Interrupts and then restarts at next scan.	
logicchart drivewait	Interrupts and then restarts at next scan.	
compare(Unsatisfied)	Interrupts and then restarts at next scan.	
wait...until (Unsatisfied)	Interrupts and then restarts at next scan.	
dialogue	Interrupts and then restarts at next scan.	
semlock wait	Interrupts and then restarts at next scan.	
wait for qsignal	Interrupts and then restarts at next scan.	

Table 7.4-2 Causes to Interrupt and Restarts SEBOL Programs Other Than SEBOL Statements

Cause of Interruption	Timeshare SEBOL	Periodic SEBOL
Number of SEBOL script lines per SFC step exceeds defined limit.	Interrupts after executing 20 lines and then restarts at next scan.	Interrupts after executing 2000 lines and then restarts at next scan.
Number of output SEBOL messages per FCS exceeds defined limit for one basic scan.	Interrupts and then restarts at next scan.	

Cause of Interruption	Timeshare SEBOL	Periodic SEBOL
SFC Step Transition	Interrupts and then restarts when CPU has idle time.	If there is only one transition condition, interrupts and then restarts at next scan. If there are multiple transitional conditions, interrupts at the timing when any of the condition is found established or not established, and then restarts at next scan.
Accessing other stations	Interrupts and then restarts if responses received when CPU is idle while CPU has idle capacity to process.	Interrupts and then restarts at next scan.(*1)
Accessing common blocks of other stations	Interrupts and then restarts if responses received when CPU is idle while CPU has idle capacity to process.	Interrupts and then restarts at next scan.

*1: If you do not want to your periodic SEBOL programs to be interrupted for accessing the data of other stations, you can define the data links for inter station communication (ADL) on the Control Drawing Builder. And then script the SEBOL program to access the linked data in the host station.

Blank Page

8. CENTUM-XL Compatibility

CENTUM-XL compatibility allows the function blocks in CENTUM VP to run the same as those in CENTUM-XL when IOP occurs or an I/O module fails.

This section describes the CENTUM-XL compatibility.

8.1 Reactions to IOP or I/O Module Failure: FFCS Series/KFCS2/LFCS2

This section describes the reactions of function blocks when IOP/IOP- occurs or an input module failed.

■ Parameters

▼ **XL compatibility for IOP reactions- Primary Input Signal, XL compatibility for IOP reactions - Secondary input and tag data referenced in the computational expressions of CALCU**

On the Constants 3 tab of FCS properties, compatibility with CENTUM-XL can be specified for primary input terminals and for secondary input terminals and data referenced in computational expressions.

If this option is selected, the behavior of function blocks when IOP/IOP- occurs on the IN terminal or an I/O module failed will be compatible with CENTUM-XL.

The applicable terminals are as follows:

- Qn terminals of CALCU
- Qn terminals of Logic Operation Block (except for CALCU)
- Jn terminals of BDA-L

● Different IOP Reactions of CENTUM-XL and CENTUM VP Function Blocks

The different reactions of the CENTUM-XL function blocks and the CENTUM VP function blocks will described below:

- Control activity when [No Alarm] is selected from Input Open Alarm drop-down list on Alarm tab
  CENTUM-XL Function Block:

  The control activity continues even when IOP (Input Open) occurs.

  CENTUM VP Function Block:

  When IOP occurs, the input signal becomes IOP status, though does not triggers IOP alarm message, the PV status becomes BAD and the block performs MAN Fallback. The other function blocks that using the bad PV will also perform MAN Fallback.

- PVs when IOP/IOP-/IOM Fail error occurs
  CENTUM-XL Function Block:

  The function block holds the PV read from I/O element (%Z) or the values of connected data.

  CENTUM VP Function Block:

  The PV of the function block acts according to the option selected on the Function Block Detail Builder. It may push the PV to an overshoot value if [PV Overshoot] is selected or holds the previous value if [Holding PV] is selected.

  The function block that connected with the bad PV will hold the previous PV (holds the last good value).

SEE ALSO For more information about PV overshooting, refer to:

"■ PV Overshoot" on page 3-22

■ Applicable Types of FCS

This feature is applicable to the following types of FCS.

FFCS/FFCS-L/FFCS-V/KFCS2/LFCS2/RFCS5

■ Applicable Function Blocks and Terminals

The following table shows the applicable regulatory control blocks and their relevant terminals.

Table 8.1-1 Applicable Regulatory Control Blocks

Function block model	Relevant terminal
PVI	IN
PVI-DV	IN
PID	IN
PID-HLD	IN
PID-BSW	IN
ONOFF	IN
ONOFF-E	IN
ONOFF-G	IN
ONOFF-GE	IN
PID-TP	IN
PID-MR	IN
PI-BLEND	IN
PID-STC	IN
MLD-PVI	IN
MC-2	FB
MC-2E	FB
MC-3	FB
MC-3E	FB
RATIO	IN
BSETU-2	IN
BSETU-3	IN/IN2/IN3/IN4
SS-H/M/L	IN1/IN2/IN3
AS-H/M/L	IN1/IN2/IN3
SS-DUAL	IN1/IN2

The following table shows the applicable computational blocks and relevant terminals.

Table 8.1-2 Applicable Computational Blocks

Function block model	Relevant terminal (*1)
ADD	IN
	Q01
MUL	IN
	Q01
DIV	IN
	Q01
AVE	-
	Q01 to Q08

Function block model	Relevant terminal (*1)
SQRT	IN
	-
EXP	IN
	-
LAG	IN
	-
INTEG	IN
	-
LD	IN
	-
RAMP	IN
	-
LDLAG	IN
	-
DLAY	IN
	-
DLAY-C	IN
	-
AVE-M	IN
	-
AVE-C	IN
	-
FUNC-VAR	IN
	-
TPCFL	IN
	Q01 to Q02
ASTM1	IN
	-
ASTM2	IN
	-
CALCU	IN
	Q01 to Q07
CALCU-C	IN
	Q01 to Q03
DSET-PVI	IN
	-
BDA-L	-
	J01 to J16

*1: - : No terminal is allowed.

8.1.1 Reactions at Process Input Error on IN Terminal

This section describes the behavior of function blocks when IOP/IOP- occurs or an I/O module fails, causing a process input error on the IN terminal.

An example of function block connection where CENTUM-XL compatibility is applied to IOP reactions is described as follows.

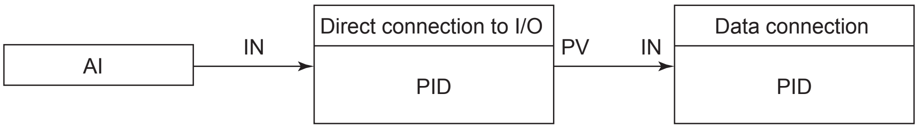


Figure 8.1.1-1 Example of Function Block Connection

■ IOP Reactions

The following table shows the reactions to IOP on the process input of IN terminal in the I/O connected function block and the data connected function block.

Table 8.1.1-1 Reactions to IOP

PV Over-shoot (builder definition item)	I/O-connected function block					Data-connected function block				
	Input Open Alarm (builder definition item) (*1)	PV			Alarm Status	Input Open Alarm (builder definition item) (*1)	PV			Alarm Status
		Data	Revised (*2)	Data Status			Data	Revised (*2)	Data Status	
YES	NO	Read value (*3)	YES	≠BAD	NR	NO	Read value (*4)	NO	≠BAD	NR
						YES	Read value (*4)	NO	≠BAD	NR
	YES	SH/SL (*5) (*6)	NO	BAD	IOP±	NO	Read value (*4)	YES	≠BAD	NR
						YES	Read value (*4) (*6)	YES	BAD	IOP±
NO	NO	Read value (*3)	YES	≠BAD	NR	NO	Read value (*4)	NO	≠BAD	NR
						YES	Read value (*4)	NO	≠BAD	NR
	YES	Hold previous value (*7)	NO	BAD	IOP±	NO	Read value (*4)	YES	≠BAD	NR
						YES	Hold previous value (*8)	NO	BAD	IOP±

*1: YES in the Input Open Alarm columns indicates the case when [Both] is specified for IOP alarm detection, and [Both] means that both high limit and low limit input open alarms are enabled. NO in the Input Open Alarm columns indicates the case when [No Alarm] is specified for IOP alarm detection.
*2: YES in Revised columns indicates that the CENTUM VP and CENTUM-XL can be defined with compatible actions.
*3: Process Input
*4: PV of connected Function Block

*5: PV Overshoot (PV is matched to SH or SL)
*6: BSETU-3 holds the previous value
*7: SS-DUAL holds (process input)
*8: SS-DUAL holds (PV of connected Function Block)

SEE ALSO For more information about PV overshooting, refer to:

"■ PV Overshoot" on page 3-22

■ Reactions at I/O Module Failure

The following table shows the reactions of both the I/O connected function block and CALCU block when the I/O module fails and an input error is caused in the I/O connected function block whose data is referenced by the computational expression of CALCU.

Table 8.1.1-2 Reactions to I/O module failure

PV Over-shoot (builder defini-tion item)	I/O-connected function block					Data-connected function block				
	Input Open Alarm (builder defini-tion item) (*1)	PV			Alarm Status	Input Open Alarm (builder defini-tion item) (*1)	PV			Alarm Status
		Data	Revised (*2)	Data Status			Data	Revised (*2)	Data Status	
YES	NO	Read value (*3) (*4) (*5)	YES	≠BAD (*6) (*5)	NR (*5)	NO	Read value (*7)	YES	≠BAD (*8) (*5)	NR (*5)
						YES	Read value (*7)	Yes (*6)	≠BAD (*8) (*5)	NR (*5)
	YES	SL (*9) (*10)	NO	BAD	IOP	NO	Read value (*7)	YES	≠BAD (*5)	NR (*5)
						YES	Read value (*7) (*10)	YES	BAD	IOP
NO	NO	Read value (*3) (*4) (*5)	YES	≠BAD (*6) (*5)	NR (*5)	NO	Read value (*7) (*5)	YES	≠BAD (*8) (*5)	NR (*5)
						YES	Read value (*7) (*5)	YES	≠BAD (*8) (*5)	NR (*5)
	YES	Hold previous value (*11)	NO	BAD	IOP	NO	Read value (*7) (*5)	YES	≠BAD (*5)	NR (*5)
						YES	Hold previous value (*12)	NO	BAD	IOP

*1: YES in the Input Open Alarm columns indicates the case when [Both] is specified for IOP alarm detection, and [Both] means that both high limit and low limit input open alarms are enabled. NO in the Input Open Alarm columns indicates the case when [No Alarm] is specified for IOP alarm detection.
*2: YES in Revised columns indicates that the CENTUM VP and CENTUM-XL can be defined with compatible actions.
*3: The last good value
*4: Pulse Input holds the previous value
*5: Different between CENTUM VP (XL Compatible: Yes) and CENTUM-XL.
*6: Pulse Input is not BAD (QST/NFP/NRDY)
*7: PV of connected Function Block
*8: For pulse input, changes to QST/NFP/NRDY
*9: PV Overshoot (PV is matched to SL)
*10: BSETU-3 holds the previous value

*11: SS-DUAL holds (The last good value)
*12: SS-DUAL holds (PV of connected Function Block)

8.1.2 Reactions at Process Input Error on Terminals Except for IN Terminal

This section describes the behavior of function blocks when IOP/IOP- occurs or an I/O module fails, causing process input errors on terminals other than the IN terminal.

■ Reactions at a process input error when the computational expression of CALCU is referencing PV of a function block directly connected to an I/O module

Reactions of function block PVI differ between the case when IOP occurs in the process input connected to the IN terminal and the case when the connected I/O module fails. As a result of this, the reaction on the data read into CALCU also differs between the case of IOP in the process input and the case of I/O module failure.

An example of function block connection where CENTUM-XL compatibility is applied to the reactions when IOP/IOP- occurs and when an I/O module fails is described as follows.

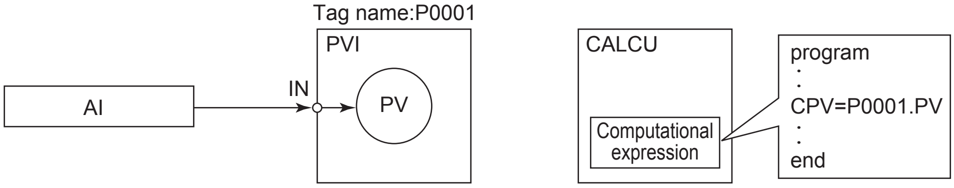


Figure 8.1.2-1 Example of Function Block Connection

● IOP Reactions

To apply CENTUM-XL compatibility, the settings of both the [Primary Input Signal] and [Secondary input and tag data referenced in the expressions of CALCU] options are required.

The following table shows the reactions of both the I/O connected function block and CALCU block when IOP occurs and an input error is caused in the I/O connected function block whose data is referenced by the computational expression of CALCU.

Table 8.1.2-1 Reactions to IOP

PV Overshoot in both CALCU and PVI (builder definition item)	I/O-connected function block				CALCU				
	Input Open Alarm (builder definition item) (*1)	PV		Alarm Status	Calculated input value error detected (builder definition item)	Input Open Alarm (builder definition item) (*1)	Value applied in the computational expression	CPV status	Alarm Status
		Data	Data Status						
YES	NO	Read value (process input)	≠BAD	NR	All-detection Type	NO	The value that the referenced block reads from process input is used to continue the computation.	≠BAD	NR
						YES			
					Compensation-computing Type	NO			
						YES			
					Non-detecting Type	NO			
						YES			
	YES	SH/SL (overshoot)	BAD	IOP±	All-detection Type	NO	The value that the referenced block overshoots is used to continue the computation.	≠BAD	
						YES	The value before IOP read by the referenced block is held to stop the computation.	BAD	
					Compensation-computing Type	NO	The value that the referenced block overshoots is used to continue the computation.	≠BAD	
						YES			
					Non-detecting Type	NO			
						YES			

PV Overshoot in both CALCU and PVI (builder definition item)	I/O-connected function block				CALCU				
	Input Open Alarm (builder definition item) (*1)	PV Data	PV Data Status	Alarm Status	Calculated input value error detected (builder definition item)	Input Open Alarm (builder definition item) (*1)	Value applied in the computational expression	CPV status	Alarm Status
NO	NO	Read value (process input)	≠BAD	NR	All-detection Type	NO	The value that the referenced block reads from process input is used to continue the computation.	≠BAD	NR
						YES			
					Compensation-computing Type	NO			
						YES			
					Non-detecting Type	NO			
						YES			
	YES	Hold previous value	BAD	IOP±	All-detection Type	NO	The value that the referenced block holds is used to continue the computation.	≠BAD	
						YES	The value before IOP read by the referenced block is held to stop the computation.	BAD	
					Compensation-computing Type	NO	The previous value that the referenced block holds is used to continue the computation.	≠BAD	
						YES			
					Non-detecting Type	NO			
						YES			

*1: YES in the Input Open Alarm columns indicates the case when [Both] is specified for IOP alarm detection, and [Both] means that both high limit and low limit input open alarms are enabled. NO in the Input Open Alarm columns indicates the case when [No Alarm] is specified for IOP alarm detection.

● Reactions at I/O Module Failure

To apply CENTUM-XL compatibility, you need to set both the [Primary Input Signal] and [Secondary input and tag data referenced in the expressions of CALCU] options.

The following table shows the reactions of both the I/O connected function block and CALCU block when the I/O module fails and an input error is caused in the I/O connected function block whose data is referenced by the computational expression of CALCU.

Table 8.1.2-2 Reactions to I/O module failure

<table>
<tr><th rowspan="3">PV Overshoot in both CALCU and PVI (builder definition item)</th><th colspan="4">I/O-connected function block</th><th colspan="5">CALCU</th></tr>
<tr><th rowspan="2">Input Open Alarm (builder definition item) (*1)</th><th colspan="2">PV</th><th rowspan="2">Alarm Status</th><th rowspan="2">Calculated input value error detected (builder definition item)</th><th rowspan="2">Input Open Alarm (builder definition item) (*1)</th><th rowspan="2">Value applied in the computational expression</th><th rowspan="2">CPV status</th><th rowspan="2">Alarm Status</th></tr>
<tr><th>Data</th><th>Data Status</th></tr>
<tr><td rowspan="12">YES</td><td rowspan="6">NO</td><td rowspan="6">Read value (The last good value)</td><td rowspan="6">≠BAD</td><td rowspan="6">NR</td><td rowspan="2">All-detection Type</td><td>NO</td><td rowspan="6">The value that the referenced block reads from process input is used to continue the computation.</td><td rowspan="6">≠BAD</td><td rowspan="12">NR</td></tr>
<tr><td>YES</td></tr>
<tr><td rowspan="2">Compensation-computing Type</td><td>NO</td></tr>
<tr><td>YES</td></tr>
<tr><td rowspan="2">Non-detecting Type</td><td>NO</td></tr>
<tr><td>YES</td></tr>
<tr><td rowspan="6">YES</td><td rowspan="6">SL (overshoot)</td><td rowspan="6">BAD</td><td rowspan="6">IOP</td><td rowspan="2">All-detection Type</td><td>NO</td><td>The value that the referenced block overshoots is used to continue the computation.</td><td>≠BAD</td></tr>
<tr><td>YES</td><td>The value before IOP read by the referenced block is held to stop the computation.</td><td>BAD</td></tr>
<tr><td rowspan="2">Compensation-computing Type</td><td>NO</td><td rowspan="4">The value that the referenced block overshoots is used to continue the computation.</td><td rowspan="4">≠BAD</td></tr>
<tr><td>YES</td></tr>
<tr><td rowspan="2">Non-detecting Type</td><td>NO</td></tr>
<tr><td>YES</td></tr>
</table>

<table>
<tr><th rowspan="3">PV Overshoot in both CALCU and PVI (builder definition item)</th><th colspan="4">I/O-connected function block</th><th colspan="5">CALCU</th></tr>
<tr><th rowspan="2">Input Open Alarm (builder definition item) (*1)</th><th colspan="2">PV</th><th rowspan="2">Alarm Status</th><th rowspan="2">Calculated input value error detected (builder definition item)</th><th rowspan="2">Input Open Alarm (builder definition item) (*1)</th><th rowspan="2">Value applied in the computational expression</th><th rowspan="2">CPV status</th><th rowspan="2">Alarm Status</th></tr>
<tr><th>Data</th><th>Data Status</th></tr>
<tr><td rowspan="12">NO</td><td rowspan="6">NO</td><td rowspan="6">Read value (The last good value)</td><td rowspan="6">≠BAD</td><td rowspan="6">NR</td><td rowspan="2">All-detection Type</td><td>NO</td><td rowspan="6">The value that the referenced block reads from process input is used to continue the computation.</td><td rowspan="6">≠BAD</td><td rowspan="12">NR</td></tr>
<tr><td>YES</td></tr>
<tr><td rowspan="2">Compensation-computing Type</td><td>NO</td></tr>
<tr><td>YES</td></tr>
<tr><td rowspan="2">Non-detecting Type</td><td>NO</td></tr>
<tr><td>YES</td></tr>
<tr><td rowspan="6">YES</td><td rowspan="6">Hold previous value</td><td rowspan="6">BAD</td><td rowspan="6">IOP</td><td rowspan="2">All-detection Type</td><td>NO</td><td>The value that the referenced block holds is used to continue the computation.</td><td>≠BAD</td></tr>
<tr><td>YES</td><td>The value before IOP read by the referenced block is held to stop the computation.</td><td>BAD</td></tr>
<tr><td rowspan="2">Compensation-computing Type</td><td>NO</td><td rowspan="4">The previous value that the referenced block holds is used to continue the computation.</td><td rowspan="4">≠BAD</td></tr>
<tr><td>YES</td></tr>
<tr><td rowspan="2">Non-detecting Type</td><td>NO</td></tr>
<tr><td>YES</td></tr>
</table>

*1: YES in the Input Open Alarm columns indicates the case when [Both] is specified for IOP alarm detection, and [Both] means that both high limit and low limit input open alarms are enabled. NO in the Input Open Alarm columns indicates the case when [No Alarm] is specified for IOP alarm detection.

■ Reactions at a process input error when the computational expression of CALCU is referencing the process input

The data read into CALCU differs between the case when IOP occurs in the process input and the case when the referenced I/O module fails.

An example of function block connection where CENTUM-XL compatibility is applied to the reactions when IOP/IOP- occurs and when the referenced I/O module fails is described as follows.

Figure 8.1.2-2 Example of Function Block Connection

● IOP Reactions

To apply CENTUM-XL compatibility, you need to set both the [Primary Input Signal] and [Secondary input and tag data referenced in the expressions of CALCU] options.

The following table shows the reaction of CALCU block when IOP occurs and an input error is caused in the I/O connected function block whose data is referenced by the computational expression of CALCU.

Table 8.1.2-3 Reactions to IOP

PV Overshoot in CALCU (builder definition item)	CALCU				
	Calculated input value error detected (builder definition item)	Input Open Alarm (builder definition item) (*1)	Value applied in the computational expression	CPV Data Status	Alarm Status
YES	All-detection Type	NO	The value read from process input is used to continue the computation.	≠BAD	NR
		YES	The value which is read before IOP occurs on the process input is held to stop the computation.	BAD	
	Compensation-computing Type	NO	The value read from process input is used to continue the computation.	≠BAD	
		YES			
	Non-detecting Type	NO			
		YES			
NO	All-detection Type	NO	The value read from process input is used to continue the computation.	≠BAD	
		YES	The value which is read before IOP occurs on the process input is held to stop the computation.	BAD	
	Compensation-computing Type	NO	The value read from process input is used to continue the computation.	≠BAD	
		YES			
	Non-detecting Type	NO			
		YES			

*1: YES in the Input Open Alarm columns indicates the case when [Both] is specified for IOP alarm detection, and [Both] means that both high limit and low limit input open alarms are enabled. NO in the Input Open Alarm columns indicates the case when [No Alarm] is specified for IOP alarm detection.

● Reactions at I/O Module Failure

To apply CENTUM-XL compatibility, you need to set both the [Primary Input Signal] and [Secondary input and tag data referenced in the expressions of CALCU] options.

The following table shows the reactions of CALCU block when the process input referenced by the computational expression of the CALCU becomes erroneous due to I/O module failure.

Table 8.1.2-4 Reactions to I/O module failure

PV Over-shoot in CAL-CU (builder definition item)	CALCU				
	Calculated input value error detected (builder definition item)	**Input Open Alarm (builder definition item) (*1)**	**Value applied in the computational expression**	**CPV Data Status**	**Alarm Status**
YES	All-detection Type	NO	The value read from process input is used to continue the computation.	≠BAD	NR
		YES	The value which is read before IOP occurs on the process input is held to stop the computation.	BAD	
	Compensation-computing Type	NO	The value read from process input is used to continue the computation.	≠BAD	
		YES			
	Non-detecting Type	NO			
		YES			
NO	All-detection Type	NO	The value read from process input is used to continue the computation.	≠BAD	
		YES	The value which is read before IOP occurs on the process input is held to stop the computation.	BAD	
	Compensation-computing Type	NO	The value read from process input is used to continue the computation.	≠BAD	
		YES			
	Non-detecting Type	NO			
		YES			

*1: YES in the Input Open Alarm columns indicates the case when [Both] is specified for IOP alarm detection, and [Both] means that both high limit and low limit input open alarms are enabled. NO in the Input Open Alarm columns indicates the case when [No Alarm] is specified for IOP alarm detection.

■ Reactions at a process input error when a calculation function block other than CALCU is referencing PV of an I/O connected function block through a Qn terminal

The data value read into the calculation block differs between the case when IOP occurs in the process input connected to the Qn terminal of the calculation block and the case when the connected I/O module fails.

An example of function block connection where CENTUM-XL compatibility is applied to the reactions when IOP/IOP- occurs and when the I/O module fails is described as follows.

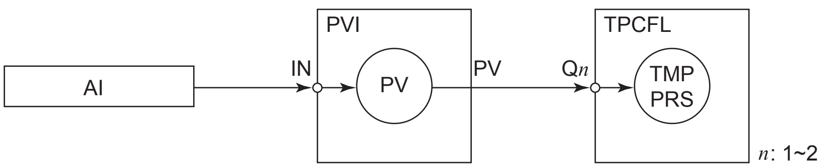


Figure 8.1.2-3 Example of Function Block Connection

● IOP Reactions

To apply CENTUM-XL compatibility, you need to set both the [Primary Input Signal] and [Secondary input and tag data referenced in the expressions of CALCU] options.

The following table shows the reactions of both the function block and CALCU block when the process input of the I/O connected function block that is connected to a Qn terminal of the calculation block becomes erroneous due to IOP.

Table 8.1.2-5 Reactions to IOP

PV Overshoot in both calculation block and PVI (builder definition item)	I/O-connected function block				Calculation block			
	Input Open Alarm (builder definition item) (*1)	PV		Alarm Status	Input Open Alarm (builder definition item) (*1)	Input value on Qn terminal (RVn/TMP/PRS)	CPV status	Alarm Status
		Data	Data Status					
YES	NO	Read value (process input)	≠BAD	NR	NO	The value that the referenced block reads from process input is used to continue the computation.	≠BAD	NR
					YES			
	YES	SH/SL (overshoot)	BAD	IOP±	NO	The value that the referenced block overshoots is used to continue or to stop the computation.	≠BAD or BAD	
					YES			
NO	NO	Read value (process input)	≠BAD	NR	NO	The value that the referenced block reads from process input is used to continue the computation.	≠BAD	
					YES			
	YES	Hold previous value	BAD	IOP±	NO	The previous value that the referenced block holds is used to continue or to stop the computation.	≠BAD or BAD	
					YES			

*1: YES in the Input Open Alarm columns indicates the case when [Both] is specified for IOP alarm detection, and [Both] means that both high limit and low limit input open alarms are enabled. NO in the Input Open Alarm columns indicates the case when [No Alarm] is specified for IOP alarm detection.

● Reactions at I/O Module Failure

To apply CENTUM-XL compatibility, you need to set both the [Primary Input Signal] and [Secondary input and tag data referenced in the expressions of CALCU] options.

The following table shows the reactions of both the I/O connected function block and CALCU block when the process input of the I/O connected function block that is connected to a Qn terminal of the calculation block becomes erroneous due to I/O module failure.

Table 8.1.2-6 Reactions to I/O module failure

PV Overshoot in both calculation block and PVI (builder definition item)	I/O-connected function block				Calculation block			
	Input Open Alarm (builder definition item) (*1)	PV Data	PV Data Status	Alarm Status	Input Open Alarm (builder definition item) (*1)	Input value on Qn terminal (RVn/TMP/PRS)	CPV status	Alarm Status
YES	NO	Read value (The last good value)	≠BAD	NR	NO	The last good value read by the referenced block is used to continue the computation.	≠BAD	NR
					YES			
	YES	SH/SL (overshoot)	BAD	IOP	NO	The value that the referenced block overshoots is used to continue or to stop the computation.	≠BAD or BAD	
					YES			
NO	NO	Read value (The last good value)	≠BAD	NR	NO	The last good value read by the referenced block is used to continue the computation.	≠BAD	
					YES			
	YES	Hold previous value	BAD	IOP	NO	The previous value that the referenced block holds is used to continue or to stop the computation.	≠BAD or BAD	
					YES			

*1: YES in the Input Open Alarm columns indicates the case when [Both] is specified for IOP alarm detection, and [Both] means that both high limit and low limit input open alarms are enabled. NO in the Input Open Alarm columns indicates the case when [No Alarm] is specified for IOP alarm detection.

■ Reactions at a process input error when the process input is connected to a Qn terminal of a calculation function block other than CALCU

The process input data value read into a calculation block other than CALCU differs between the case when IOP occurs in the process input connected to the Qn terminal of the calculation block and the case when the connected I/O module fails.

An example of function block connection where CENTUM-XL compatibility is applied to the reactions when IOP/IOP- occurs and when the I/O module fails is described as follows.

Figure 8.1.2-4 Example of Function Block Connection

● IOP Reactions

To apply CENTUM-XL compatibility, you need to set both the [Primary Input Signal] and [Secondary input and tag data referenced in the expressions of CALCU] options.

The following table shows the reactions of the calculation block when the process input connected to a Qn terminal of the calculation block becomes erroneous due to IOP.

Table 8.1.2-7 Reactions to IOP

PV Overshoot in calculation block (builder definition item)	Calculation block			
	Input Open Alarm (builder definition item) (*1)	Input value on Qn terminal (RVn/TMP/PRS)	CPV status	Alarm Status
YES	NO	The value read from process input is used to continue the computation.	≠BAD	NR
	YES	The value read from process input is used to continue or to stop the computation.	≠BAD or BAD	
NO	NO	The value read from process input is used to continue the computation.	≠BAD	
	YES	The value read from process input is used to continue or to stop the computation.	≠BAD or BAD	

*1: YES in the Input Open Alarm columns indicates the case when [Both] is specified for IOP alarm detection, and [Both] means that both high limit and low limit input open alarms are enabled. NO in the Input Open Alarm columns indicates the case when [No Alarm] is specified for IOP alarm detection.

● Reactions at I/O Module Failure

To apply CENTUM-XL compatibility, you need to set both the [Primary Input Signal] and [Secondary input and tag data referenced in the expressions of CALCU] options.

The following table shows the reactions of the calculation block when the process input from the I/O module connected to a Qn terminal of the calculation block becomes erroneous due to I/O module failure.

Table 8.1.2-8 Reactions to I/O module failure

PV Overshoot in calculation block (builder definition item)	Calculation block			
	Input Open Alarm (builder definition item) (*1)	Input value on Qn terminal (RVn/TMP/PRS)	CPV status	Alarm Status
YES	NO	The value read from process input is used to continue the computation.	≠BAD	NR
	YES	The value read from process input is used to continue or to stop the computation.	≠BAD or BAD	
NO	NO	The value read from process input is used to continue the computation.	≠BAD	
	YES	The value read from process input is used to continue or to stop the computation.	≠BAD or BAD	

*1: YES in the Input Open Alarm columns indicates the case when [Both] is specified for IOP alarm detection, and [Both] means that both high limit and low limit input open alarms are enabled. NO in the Input Open Alarm columns indicates the case when [No Alarm] is specified for IOP alarm detection.

■ Reactions at a process input error when BDA-L is referencing PV of an I/O connected function block through a Jn terminal

The data value read into BDA-L differs between the case when IOP occurs in the process input connected to the Jn terminal of BDA-L and the case when the connected I/O module fails.

An example of function block connection where CENTUM-XL compatibility is applied to the reactions when IOP/IOP- occurs and when the I/O module fails is described as follows.

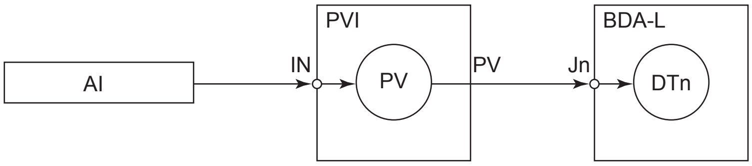


Figure 8.1.2-5 Example of Function Block Connection

● IOP Reactions

To apply CENTUM-XL compatibility, you need to set both the [Primary Input Signal] and [Secondary input and tag data referenced in the expressions of CALCU] options.

The following table shows the reactions of both the I/O connected function block and BDA-L block when the process input of the I/O connected function block that is connected to a Jn terminal of the BDA-L block becomes erroneous due to IOP.

Table 8.1.2-9 Reactions to IOP

PV Overshoot in PVI (builder definition item)	I/O-connected function block				BDA-L		
	Input Open Alarm (builder definition item) (*1)	PV		Alarm Status	Value acquired by DTn	DTn status	Alarm Status
		Data1	Data Status				
YES	NO	Read value (process input)	≠BAD	NR	Process input value read by the I/O-connected function block	≠BAD	NR
	YES	SH/SL (overshoot)	BAD	IOP±	Value that the I/O-connected function block overshoots	BAD	
NO	NO	Read value (process input)	≠BAD	NR	Process input value read by the I/O-connected function block	≠BAD	
	YES	Hold previous value	BAD	IOP±	Previous PV value that is held by the I/O-connected block	BAD	

*1: YES in the Input Open Alarm columns indicates the case when [Both] is specified for IOP alarm detection, and [Both] means that both high limit and low limit input open alarms are enabled. NO in the Input Open Alarm columns indicates the case when [No Alarm] is specified for IOP alarm detection.

● Reactions at I/O Module Failure

To apply CENTUM-XL compatibility, you need to set both the [Primary Input Signal] and [Secondary input and tag data referenced in the expressions of CALCU] options.

The following table shows the reactions of both the I/O connected function block and BDA-L block when the process input of the I/O connected function block that is connected to a Jn terminal of the BDA-L block becomes erroneous due to I/O module failure.

Table 8.1.2-10 Reactions to I/O module failure

PV Over-shoot in PVI (builder defini-tion item)	I/O-connected function block				BDA-L		
	Input Open Alarm (builder defini-tion item) (*1)	PV		Alarm Status	Value acquired by DTn	DTn sta-tus	Alarm Status
		Data1	Data Sta-tus				
YES	NO	Read value (The last good value)	≠BAD	NR	Process input value read by the I/O-con-nected function block	≠BAD	NR
	YES	SL (overshoot)	BAD	IOP	Value that the I/O-connected function block overshoots	BAD	
NO	NO	Read value (The last good value)	≠BAD	NR	Process input value read by the I/O-con-nected function block	≠BAD	
	YES	Hold previous value	BAD	IOP	Previous PV value that is held by the I/O-connected block	BAD	

*1: YES in the Input Open Alarm columns indicates the case when [Both] is specified for IOP alarm detection, and [Both] means that both high limit and low limit input open alarms are enabled. NO in the Input Open Alarm columns indicates the case when [No Alarm] is specified for IOP alarm detection.

8.2 Reactions at DI Module Failure: FFCS Series/KFCS2

In case when a digital input module fails, you can determine whether the PV values of the relevant process input terminals are held or set to zeros.

■ Parameters

▼ Set PV of %Z to zero upon failure of a DI module

To apply CENTUM-XL compatibility to the reactions to DI module failure, select the [Set PV of %Z to zero upon failure of a DI module] check box on the Constant 3 tab of FCS properties.

■ Applicable FCS Types

The types of applicable FCS are FFCS-V, FFCS, FFCS-L, KFCS2, and RFCS5.

■ Applicable Function Blocks

The following table shows the function blocks to which this feature is applicable.

Table 8.2-1 Applicable function blocks

Function block type	Model
Sequence table	ST16
Logic chart	LC64, LC64-E
Switch instrument	SI-1, SI-2, SIO-11, SIO-12, SIO-21, SIO-22, SIO-12P, SIO-22P
Enhanced switch instrument	SI-1E, SI-2E, SIO-11E, SIO-12E, SIO-21E, SIO-22E, SIO-12PE, SIO-22PE
Code input	CI
Valve monitor	VLVM
Logic operation	AND, OR, NOT, SRS1-S, SRS1-R, SRS2-S, SRS2-R, WOUT, OND, OFFD, TON, TOFF, BAND, BOR, BNOT
Calculation	CALCU, CALCU-C
SFC	_SFCSW, _SFCPB, _SFCAS
SEBOL	OPSFC, OPSBL, OPSFCP1, OPSFCP2, OPSFCP3, OPSFCP4, OPSFCP5
Unit instrument	_UTSW, _UTPB, _UTAS, _UTSW-N, _UTPB-N, _UTAS-N, _UTSW-SN, _UTPB-SN, _UTAS-SN
Unit operation	UTOP-SN

PV values of process I/O terminals are displayed in the process report and faceplates. When a digital input module fails, the PV values will be displayed as follows according to the status of the check box for this feature:

- Unchecked: PV holds the last good value.
- Checked: PV is set to zero.

● Sequence Table Block

When the condition signal contains ON/OFF judgment for DI element, the system runs as follows by checking/unchecking [Set PV of %Z to zero upon failure of a DI module] checkbox.

- Unchecked: DI value before DI module failure is used to judge whether the condition is satisfied or not.

- Checked: Whether the condition is satisfied or not is judged when DI turns off.

The following table shows how the sequence table behaves when a DI module fails.

Table 8.2-2 Behavior of sequence table at DI module failure

Condition signals and rule descriptions	Event	[Set PV of %Z to zero upon failure of a DI module] checkbox	
		Unchecked	Checked
Element symbol.PV.ON:Y	I/O module fails while DI turns off	Not satisfied	Not satisfied
	I/O module fails while DI turns on	Satisfied	Not satisfied
Element symbol.PV.ON:N	I/O module fails while DI turns off	Satisfied	Satisfied
	I/O module fails while DI turns on	Not satisfied	Satisfied
Element symbol.PV.OFF:Y	I/O module fails while DI turns off	Satisfied	Satisfied
	I/O module fails while DI turns on	Not satisfied	Satisfied
Element symbol.PV.OFF:N	I/O module fails while DI turns off	Not satisfied	Not satisfied
	I/O module fails while DI turns on	Satisfied	Not satisfied
Element symbol.PV=BAD:Y	I/O module fails on DI	Satisfied	Satisfied
Element symbol.PV=BAD:N	I/O module fails on DI	Not satisfied	Not satisfied

● Logic Chart Blocks

When the condition signal contains ON/OFF judgment for DI element, the system runs as follows by checking/unchecking [Set PV of %Z to zero upon failure of a DI module] checkbox.

- Unchecked: DI value before DI module failure is used to judge whether the condition is satisfied or not.
- Checked: Whether the condition is satisfied or not is judged when DI turns off.

The following table shows how the logic chart behaves when a DI module fails.

Table 8.2-3 Behavior of logic chart at DI module failure

Description of condition signal	Event	[Set PV of %Z to zero upon failure of a DI module] checkbox	
		Unchecked	Checked
Element symbol.PV.ON	I/O module fails while DI turns off	Not satisfied	Not satisfied
	I/O module fails while DI turns on	Satisfied	Not satisfied
Element symbol.PV.OFF	I/O module fails while DI turns off	Satisfied	Satisfied
	I/O module fails while DI turns on	Not satisfied	Satisfied
Element symbol.PV=BAD	I/O module fails on DI	Satisfied	Satisfied

● Switch Instrument and Enhanced Switch Instrument Blocks

When DI element is connected to IN terminal, the system runs as follows by checking/unchecking [Set PV of %Z to zero upon failure of a DI module] checkbox.

- Unchecked: PV value holds the previous value.
- Checked: PV takes the value when the raw input data becomes zero.

The following table shows how the switch instrument behaves at DI module failure.

Table 8.2-4 Behavior of switch instrument at DI module failure

Applicable function block models	Answerback direction	Event	[Set PV of %Z to zero upon failure of a DI module] checkbox	
			Unchecked	Checked
SI-1, SIO-11,SIO-12, SIO-12P, SI-1E, SIO-11E, SIO-12E, SIO-12PE	Direct action	I/O module fails while DI turns off	PV=0 is maintained	PV=0 is maintained
		I/O module fails while DI turns on	PV=2 is maintained	PV=0
	Reverse action	I/O module fails while DI turns off	PV=2 is maintained	PV=2 is maintained
		I/O module fails while DI turns on	PV=0 is maintained	PV=2
SI-2, SIO-21, SIO-22, SIO-22P, SI-2E, SIO-2E, SIO-22E, SIO-22PE	Direct action	I/O module fails when the first contact and the second contacts turn off for DI	PV=1 is maintained	PV=1 is maintained
		I/O module fails when the first contact turns on and the second contacts turns off for DI	PV=2 is maintained	PV=1
		I/O module fails when the first contact turns off and the second contacts turns on for DI	PV=0 is maintained	PV=1
		I/O module fails when the first contact and the second contacts turn on for DI	PV=0/1//2 (previous value) ALRM=PERR	PV=1 Recovered from PERR
	Reverse action	I/O module fails when the first contact and the second contacts turn off for DI	PV=0/1//2 (previous value) ALRM=PERR	PV=0/1//2 (previous value) ALRM=PERR
		I/O module fails when the first contact turns on and the second contacts turns off for DI	PV=0 is maintained	PV=0 is maintained ALRM=PERR
		I/O module fails when the first contact turns off and the second contacts turns on for DI	PV=2 is maintained	PV=2 is maintained ALRM=PERR
		I/O module fails when the first contact and the second contacts turn on for DI	PV=1 is maintained	PV=1 is maintained ALRM=PERR

● Code Input Block

When DI element is connected to IN terminal, the system runs as follows at One-Shot execution by checking/unchecking [Set PV of %Z to zero upon failure of a DI module] checkbox.

- Unchecked: PV value holds the previous value.
- Checked: PV is set to zero.

Valve Monitor Block

When DI element is connected to the external input terminals (Q01 to Q32), the system runs as follows by checking/unchecking [Set PV of %Z to zero upon failure of a DI module] checkbox.

- Unchecked: The valve output monitoring feature stops and the valve error PVnn holds the previous value.
- Checked: The valve error PVnn is refreshed.

Logic Operation Block

Logic Operation Block behaves as follows by checking/unchecking [Set PV of %Z to zero upon failure of a DI module] checkbox.

- Unchecked: The corresponding RVn holds the previous value.
- Checked: DI is considered to be off and the corresponding RVn judges whether the condition is satisfied or not.

The following table shows how Logic Operation Block behaves when a DI module fails.

Table 8.2-5 Behavior of Logic Operation Block at DI module failure

Relevant function block models	Connection method	Event	[Set PV of %Z to zero upon failure of a DI module] checkbox	
			Unchecked	Checked
AND, OR, SRS1-S, SRS1-R, SRS2-S, SRS2-R, WOUT, BAND (*1), BOR (*1)	Sequence connection Element symbol.PV.ON	I/O module fails while DI turns off	RVn=0 is maintained CPV: The value acquired by the specified detection of an error in calculated input value	RVn=0 CPV: The value acquired by the specified detection of an error in calculated input value
		I/O module fails while DI turns on	RVn=1 is maintained CPV: The value acquired by the specified detection of an error in calculated input value	RVn=0 CPV: The value acquired by the specified detection of an error in calculated input value
	Sequence connection Element symbol.PV.OFF	I/O module fails while DI turns off	RVn=1 is maintained CPV: The value acquired by the specified detection of an error in calculated input value	RVn=1 CPV: The value acquired by the specified detection of an error in calculated input value
		I/O module fails while DI turns on	RVn=0 is maintained CPV: The value acquired by the specified detection of an error in calculated input value	RVn=1 CPV: The value acquired by the specified detection of an error in calculated input value
	Data connection Element symbol.PV	I/O module fails while DI turns off	RVn=0 is maintained CPV: The value acquired by the specified detection of an error in calculated input value	RVn=0 is maintained CPV: The value acquired by the specified detection of an error in calculated input value
		I/O module fails while DI turns on	RVn=1 is maintained CPV: The value acquired by the specified detection of an error in calculated input value	RVn=1 is maintained CPV: The value acquired by the specified detection of an error in calculated input value

Relevant function block models	Connection method	Event	[Set PV of %Z to zero upon failure of a DI module] checkbox	
			Unchecked	Checked
NOT, OND, OFFD, TON, TOFF, BNOT (*1)	Sequence connection Element symbol.PV.ON	I/O module fails while DI turns off	RVn=0 is maintained CPV is maintained	RVn=0 CPV is maintained
		I/O module fails while DI turns on	RVn=1 is maintained CPV is maintained	RVn=0 CPV is maintained
	Sequence connection Element symbol.PV.OFF	I/O module fails while DI turns off	RVn=1 is maintained CPV is maintained	RVn=1 CPV is maintained
		I/O module fails while DI turns on	RVn=0 is maintained CPV is maintained	RVn=1 CPV is maintained
	Data connection Element symbol.PV	I/O module fails while DI turns off	RVn=0 is maintained CPV is maintained	RVn=0 CPV is maintained
		I/O module fails while DI turns on	RVn=1 is maintained CPV is maintained	RVn=1 CPV is maintained

*1: Sequence connection is not allowed in BAND, BOR and BNOT. Only data connection is allowed.

● Calculation Block

This section explains the behaviors when I/O terminal of CALCU block applies data connection or sequence connection.

- Data Connection
  When IN terminal of CALCU intends data connection to DI, the system runs according to the settings for [Primary Input Signal] at the event of DI module failure.

  When data connection is intended to DI for DI data reference in the computational expression of CALCU or external input terminals (Q01 to Q32), the system runs according to the settings for [Secondary input and tag data referenced in the expressions of CALCU] at the event of DI module failure.

- Sequence Connection
  When IN terminal of CALCU or external input terminals (Q01 to Q32) intend sequence connection to DI or when a computational expression browses the sequence-connected DI, the system runs as follows by checking/unchecking [Set PV of %Z to zero upon failure of a DI module] checkbox.

  - Unchecked: The corresponding RVn holds the previous value.
  - Checked: DI is considered to be off and the corresponding RVn judges whether the condition is satisfied or not.

● SFC/SEBOL/Unit Instrument /Unit Operation Blocks

When DI is browsed in SFC transition, SEBOL statement or unit instrument transition, the system runs as follows by checking/unchecking [Set PV of %Z to zero upon failure of a DI module] checkbox.

- Unchecked: PV holds the last good value.
- Checked: PV is set to zero.

8.3 Differences from CENTUM -XL

This section describes differences from CENTUM-XL in the following items:

- Behavior when input open (IOP) alarm check is enabled and the function block is in Calibration (CAL) mode(*1)
- Alarm Notification when IOP- is detected
- PV Range limit
- Connection to SS-DUAL
- IOP of SS-H/SS-M/SS-L and AS-H/AS-M/AS-L
- RV of SS-H/SS-M/SS-L, AS-H/AS-M/AS-L, and SS-DUAL

*1: By using Input Open Alarm drop-down list on Alarm tab on the Function Block Detail Builder, select the type of input open (IOP) alarm check.

■ Discrepancies between CENTUM-XL and CENTUM VP – When input open (IOP) alarm check is enabled and the function block is in Calibration (CAL) mode

When input open (IOP) alarm check-enabled function blocks are connected with process I/O through terminal connection or data connection and the function blocks are in calibration (CAL) mode, the PV data status of the function block that is directly connected to I/O acts differently from CENTUM-XL on IOP/IOP- occurrence.

An example is shown below:

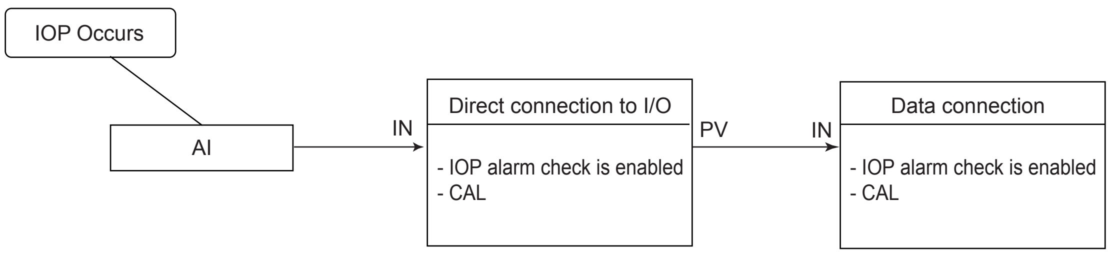


Figure 8.3-1 Difference between CENTUM-XL Compatible and CENTUM-XL Reactions (Function Block Directly Connected to Process I/O, IOP alarm check is enabled and in CAL mode)

In the above figure, when a function block that directly connected to the process I/O encounter IOP, the PV status of the function block changes differently in CENTUM-XL and CENTUM VP. Refer the following table.

Table 8.3-1 Difference (Function Block Directly Connected to Process I/O, IOP alarm check is enabled and in CAL mode)

System	PV: Directly Connected to Process I/O	PV: PV of referencing function block	Remarks
CENTUM-XL	≠BAD	≠BAD	PVs of both function blocks do not change to BAD.
CENTUM VP	BAD	≠BAD	PV of the function block that directly connected to process I/O changes to BAD status. Since the function block that directly connected to the process I/O is in CAL mode, so that the PV of the function block that referencing that function block' s PV does not change to BAD status. (If the function block that directly connected to the process I/O is not in CAL mode but only the referencing function block is CAL mode, the PV of referencing function block will become BAD.)

Discrepancies between CENTUM-XL and CENTUM VP – Alarm Notification when IOP- is detected

- CENTUM-XL: When Detect IOP is enabled, IOP- occurrence will trigger an IOP alarm and send a notification to the operator console.
- CENTUM VP: When Detect IOP is enabled, IOP- occurrence will trigger an IOP- alarm and send a notification to the operator console.

Discrepancies between CENTUM-XL and CENTUM VP – PV Range limit

- CENTUM-XL: PV range limit is always running and the PV value is limited within the range between scale high-limit (SH) and scale low-limit (SL). When the raw value of the PV is bigger than SH or smaller than SL, the SH value or the SL value will be used for the PV.
- CENTUM VP: Since "PV Range Limit" is disabled by default, the PV value is not limited within the range between scale high-limit (SH) and scale low-limit (SL).
  However, after enabling "PV Range Limit," the PV value in CENTUM VP system can also be limited within the range between SH and SL.

  Enabling or disabling the PV Range Limit can be defined on the Function Block Detail Builder.

  PV Range Limit: Select [Yes] or [No].

Discrepancies between CENTUM-XL and CENTUM VP – Connected to SS-DUAL

- CENTUM-XL: SS-DUAL cannot be directly connected to an process I/O.
- CENTUM VP: SS-DUAL can be directly to an process I/O.

Discrepancies between CENTUM-XL and CENTUM VP – SS-H/M/L and AS-H/M/L IOP Reactions

- CENTUM-XL: SS-H/M/L and AS-H/M/L are irrelevant from IOP.
- CENTUM VP: SS-H/M/L and AS-H/M/L can define input open (IOP) alarm check type. When migrating from CENTUM-XL and CENTUM VP, it is necessary to select [No Alarm] from Input Open Alarm drop-down list on the Alarm tab so as to maintain compatibility with CENTUM-XL.

Discrepancies between CENTUM-XL and CENTUM VP – RV of SS-H/M/L, AS-H/M/L and SS-DUAL

- CENTUM-XL: SS-H/M/L, AS-H/M/L and SS-DUAL do not have RV
- CENTUM VP: SS-H/M/L, AS-H/M/L and SS-DUAL have RV

Blank Page

Function Blocks Overview

IM 33K03E21-50E 5th Edition

INDEX

Revision Information

Title : Function Blocks Overview

Manual No. : IM 33K03E21-50E

Aug. 2014/5th Edition/R5.04 or later*

* : Denotes the release number of the Software Product corresponding to the contents of this Manual. The revised contents are valid until the next edition is issued.

Preface	Descriptions of grounding are modified.
3.1.1	Descriptions of "● Data Conversion" in "■ Communication Input Conversion" are modified.
4.8.3	Descriptions of "■ Communication Output Conversion" are modified.
5.	Descriptions of "■Functional Structure of the Alarm Processing" are modified.
5.	"■ Delayed Alarm Detection Function" is added.
5.1	Descriptions about compatibility with CENTUM-XL are removed.
5.6	Descriptions of "● Deviation Check Filter" in "■ Settings for the Deviation Alarm Check" are modified.
5.17	Newly added.
8	Newly added.

Apr. 2014/4th Edition/R5.03.20 or later

Preface	Description of the caution symbol is modified.
Postface	The "■ For Questions and More Information" is changed.
	The "■ Printed by" is deleted.

Jun. 2013/3rd Edition/R5.03 or later

Preface	The preface text is modified.
3.6.2	"Input Processing of the Calculation Block in Unsteady State" is added.
4.	"● Output Processing During OOP" is added.
4.1	Descriptions on "■ Output Limiter" are modified.
4.1	Descriptions on "■ High/Low Limit Bumpfree Capability" are modified.
4.1	Descriptions on "■ Disable High/Low Limit Bumpfree Capability" are modified.
4.8.1	Descriptions on "● Tight-shut and Full-Open" are modified.
4.8.2	Descriptions on "■ Pulse Width Output in PRD Mode" are modified.
4.8.2	Descriptions on "● Output Limiter in PRD Mode" are modified.
4.13	Newly added.
7.4	Descriptions on "● Periodic SEBOL Action after Step Transition" are modified.

Jun. 2012/2nd Edition/R5.02 or later

7.3	Added a description for logic chart blocks.

Sep. 2011/1st Edition/R5.01 or later

Newly published

■ For Questions and More Information
Online Query: A query form is available from the following URL.
http://www.yokogawa.com/dcs/
■ Written by Yokogawa Electric Corporation
■ Published by Yokogawa Electric Corporation
2-9-32 Nakacho, Musashino-shi, Tokyo 180-8750, JAPAN