

SIMATIC

Process Control System PCS 7 SIMATIC BATCH Blocks V9.1 SP2

Function Manual

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

 DANGER
indicates that death or severe personal injury will result if proper precautions are not taken.
 WARNING
indicates that death or severe personal injury may result if proper precautions are not taken.
 CAUTION
indicates that minor personal injury can result if proper precautions are not taken.
NOTICE
indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

 WARNING
Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Table of contents

1	Security information	7
2	Interface blocks	9
2.1	General information on the block description	9
2.2	BATCH interface blocks.....	13
2.2.1	IEPH	13
2.2.1.1	IEPH: Equipment Phase - BATCH Interface	13
2.2.1.2	I/O of IEPH	18
2.2.1.3	IEPH message texts and associated values	20
2.2.1.4	Controlling and monitoring IEPH	21
2.2.2	IEOP	23
2.2.2.1	IEOP: Equipment operation - BATCH interface	23
2.2.2.2	I/Os of IEOP.....	29
2.2.2.3	IEOP message texts and associated values	30
2.2.2.4	Operator Control and Monitoring of IEOP	32
2.2.3	TAG_COLL.....	34
2.2.3.1	TAG_COLL: TAG_Collect interface BATCH.....	34
2.2.3.2	I/O of TAG_COLL.....	35
2.2.4	UNIT_PLC.....	35
2.2.4.1	UNIT_PLC: Unit allocation - BATCH interface.....	35
2.2.4.2	Occupy and release.....	36
2.2.4.3	Status word	37
2.2.4.4	User Status Word	38
2.2.4.5	Deactivating a Unit	38
2.2.4.6	Command interface for RPEs in a unit.....	39
2.2.4.7	Command interface to influence the entire unit.....	40
2.2.4.8	Settings for communication monitoring and communication buffer	41
2.2.4.9	EQM DB consistency check.....	43
2.2.4.10	Recipe processing	45
2.2.4.11	Monitoring time of a recipe operation	45
2.2.4.12	Diagnostics Messages	45
2.2.4.13	UNIT_PLC connectors	46
2.2.4.14	UNIT_PLC message texts and associated values.....	52
2.2.4.15	Operating and Monitoring of UNIT_PLC	54
2.3	Equipment parameter module.....	59
2.3.1	IEPAR_BOOL.....	59
2.3.1.1	IEPAR_BOOL: Equipment parameter module for data type BOOL	59
2.3.1.2	I/Os of IEPAR_BOOL	59
2.3.2	IEPAR_DEST.....	60
2.3.2.1	IEPAR_DEST: Equipment parameter module for data type DEST	60
2.3.2.2	I/Os of IEPAR_DEST	60
2.3.3	IEPAR_DINT.....	61
2.3.3.1	IEPAR_DINT: Equipment parameter module for data type Double Integer.....	61
2.3.3.2	I/Os of IEPAR_DINT	61
2.3.4	IEPAR_ENUM.....	62

2.3.4.1	IEPAR_ENUM: Equipment parameter module for parameter type ENUM.....	62
2.3.4.2	I/Os of IEPAR_ENUM	63
2.3.5	IEPAR_PI.....	64
2.3.5.1	IEPAR_PI: Equipment parameter module for process inputs	64
2.3.5.2	I/Os of IEPAR_PI	64
2.3.6	IEPAR_PO	65
2.3.6.1	IEPAR_PO: Equipment parameter module for process outputs	65
2.3.6.2	I/Os of IEPAR_PO	66
2.3.7	IEPAR_REAL.....	67
2.3.7.1	IEPAR_REAL: Equipment parameter module for data type REAL	67
2.3.7.2	I/Os of IEPAR_REAL	67
2.3.8	IEPAR_SOURCE.....	68
2.3.8.1	IEPAR_SOURCE: Equipment parameter module for data type SOURCE.....	68
2.3.8.2	I/Os of IEPAR_SOURCE	68
2.3.9	IEPAR_STR.....	69
2.3.9.1	IEPAR_STR: Equipment parameter module for parameter type STRING	69
2.3.9.2	I/Os of IEPAR_STR	69
2.3.10	IEPAR_VIA	70
2.3.10.1	IEPAR_VIA: Equipment parameter module for data type VIA	70
2.3.10.2	I/Os of IEPAR_VIA	71
3	System blocks	73
3.1	Function blocks.....	73
3.1.1	RecipeControl	73
3.1.2	MemIDB64	73
3.1.3	EventSend	73
3.1.3.1	EventSend	73
3.1.3.2	EventSend installation	74
3.1.3.3	Collect EventSend RPE events and communication with BCS	75
3.1.3.4	EventSend diagnostic messages	76
3.2	Function calls.....	77
3.2.1	CallFB	77
3.2.2	CheckCmdPriority	77
3.2.3	CheckEQMDB.....	77
3.2.4	CmdControl	77
3.2.5	CommRPEEvent	78
3.2.6	CommSPDataToDB	78
3.2.7	ContControl.....	78
3.2.8	ContGetRPEListSW	79
3.2.9	ContPropagateRef	79
3.2.10	ContSetSW.....	79
3.2.11	CopyBlock.....	80
3.2.12	DebugTrace	80
3.2.13	EPARGet.....	80
3.2.14	EPARGetValue	81
3.2.15	EPHGetHeader	81
3.2.16	EPHGetSW	81
3.2.17	EPHManualControl.....	81
3.2.18	EPHRun.....	82
3.2.19	LookUp_ID.....	82
3.2.20	NOPControl	82
3.2.21	ReliableCmd	83

3.2.22	RPEEnumEleListCW	83
3.2.23	RPEGetSP	83
3.2.24	RPEGetSPList	84
3.2.25	RPEGetSW	84
3.2.26	RPEPrepareOccupy	84
3.2.27	RPEPrepareToCmpl	84
3.2.28	RPESetCW	85
3.2.29	RPESetLinkCW	85
3.2.30	RPHControl	85
3.2.31	RPHGetAllRef	86
3.2.32	RPHGetAllSP	86
3.2.33	RPHGetRef	86
3.2.34	RPHSetRef	87
3.2.35	TagCollGetHeader	87
3.2.36	TransBCSControl	87
3.2.37	TransControl	88
3.2.38	TransGetAllPin	88
3.2.39	TransGetPin	88
3.2.40	TransGetPinOper	89
3.2.41	TransGetPinVal	89
3.2.42	TransGetVal	89
3.2.43	TransSetGateVal	90
3.2.44	UnitParamControl	90
Index		91

Tables

Table 2-3	Parameter properties	11
Table 2-2	Block algorithm	11
Table 2-1	I/Os of	10

Figures

Figure 2-1	Assign enumeration type	63
------------	-------------------------------	----

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For additional information on industrial security measures that may be implemented, please visit

<https://www.siemens.com/industrialsecurity>.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under

<https://www.siemens.com/cert>.

Interface blocks

2.1 General information on the block description

Introduction

The block descriptions always organized in the following form:

- Overview of the block description
- Object name (class + number)
- Function
- Operating principle
- Calling OBs
- Error handling
- I/Os of ...
- Operating control and monitoring of ...

Overview of the block description

Example: IEPH: Equipment phase - BATCH interface

The title begins with the type name of the block (IEPH). This symbol name is entered in the symbol table and must be unique in the project.

You will find the keyword for the task/function of the block (equipment phase - BATCH interface) in the type name.

Object name (class + number)

FBx

The object name of the block type is formed by the type of implementation:

Function block = FB, function = FC and the block number = x.

Function

Brief description of the block function. You can find additional information about complex blocks in the section "Operating principle".

Operating principle

Additional information about the functions of individual inputs, modes, time sequences, etc. You learn about the overall relationships to use the block effectively.

Calling OBs

Information about the organization blocks (OBs) in which the block being described must be installed. When using CFC, the block is installed in the cyclic OBs (cyclic interrupt) and automatically in the OBs listed in the taskbar of the block (for example, in OB100 for a hot restart).

During compilation, CFC generates the required OBs. When using the blocks without CFC, you must program these OBs and call the block instance in them.

Error handling

You can find the error code in the CFC chart at the **ENO** Boolean block output. The value corresponds to the **BR** (binary result in STEP 7 SCL on completion of the block) or the **OK** bit (in SCL notation) has the following meaning:

- ENO=BR=OK=1 (TRUE) → The result of the block is correct.
- ENO=BR=OK=0 (FALSE) → The result or the conditions for calculating the result (for example, input values, modes) are not valid.

With FBs, you will also find the inverted BR at the output **QERR** of the instance DB.

- QERR=NOT ENO

The error code is produced in two independent ways:

- The operating system detects a processing error (for example, value overflow, called system functions supply error ID with BR=0). This is a system function and is not described in detail in the individual block descriptions.
- The block algorithm checks whether values and modes are functionally permitted. These errors are documented in the description of the block.

You can evaluate the error code, for example, to create messages or to work with substitute values instead of incorrect results.

I/Os of ...

The I/Os represent the data interface of the block. Via these I/Os, you can transfer data to the block and fetch results from it.

Table 2-1 I/Os of ...

I/O (parameter)	Meaning	Type	Default	Class	Attributes	OCM	Permitted values
SP_VAL	Setpoint (input)	DInt	0	I	QU	+	

The "I/Os" table lists all the input and output parameters of the block type that the user can access with configuration tools. They are sorted alphabetically. I/Os that can only be accessed by the

algorithm of the block, so-called internal variables, are not listed. The columns have the following meaning:

- **I/O**
Name of the parameter derived from the English name.
Example: PV_IN = **P**rocess **V**ariable **I**Nput (process variable, control variable). As far as permitted by SIMATIC conventions, the same names were used.
- **Meaning**
Function (possibly a brief description)
- **Type**
S7 data type of the parameter (e.g. BOOL, REAL)
- **Default**
The value of the parameter before the block is run for the first time, unless this was changed during configuration
- **Class**
Type of access of the block algorithm to the parameter; differentiates between inputs, reacting inputs and outputs (table below).

Table 2-2 Block algorithm

Abbrevia- tion	Class
I	Input Value supply of the block (representation in the CFC: left side of block)
O	Output Output value. (representation in the CFC: right side of the block)
IO	Input / output Reacting input that can be written by the OS and written back by the block (representation in the CFC: left side of block)

- **Attr. (Attributes)**
Additional characteristics of the parameter when used in CFC. Input and output parameters that are not interconnected can have parameters assigned (input/output parameters only with online FCs).
Output parameters cannot have values assigned and can be transferred in CFC by interconnecting them to an input of the same data type.

Additional or deviating parameter properties are specified as follows:

Table 2-3 Parameter properties

Abbrevia- tion	Attribute
Q	Interconnectable The I/O can be interconnected with another output of the same type.
U	Hidden in the CFC The I/O is not displayed in CFC (for example, the message ID), because it is supplied by the block itself or by the OS block. This default setting can be changed in the CFC.

2.1 General information on the block description

OCM

Parameters identified by the "+" character can be monitored and controlled using the corresponding OS block.

Permitted values

Valid limit within the range of values of the data type.

Operating control and monitoring of ...

If an OS block exists for the PLC block, the fields on the faceplate are described in the table.

Note

Detail technical version of the Blocks might be different to the version of the product.

2.2 BATCH interface blocks

2.2.1 IEPH

2.2.1.1 IEPH: Equipment Phase - BATCH Interface

Object name (class + number)

FB 254

Calling OBs

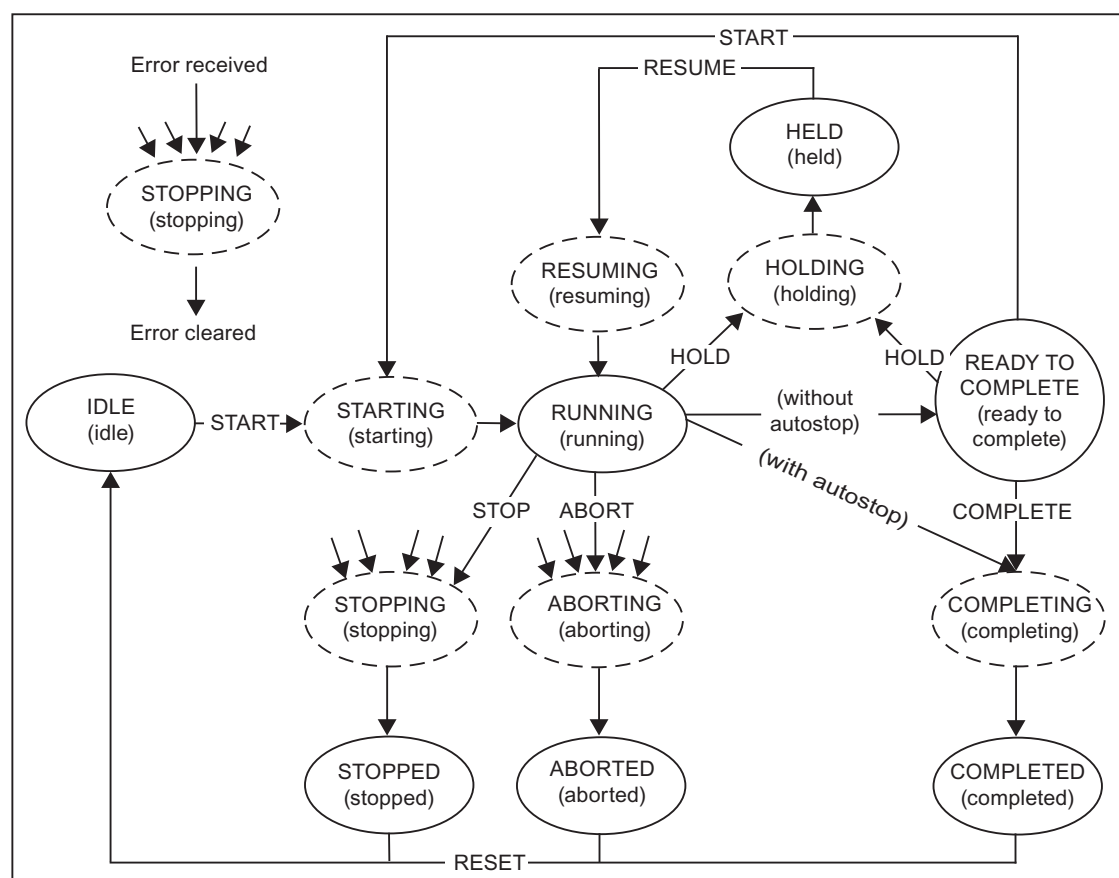
The OB in which you install the block (for example, OB1).

Function

The IEPH block serves as the interface between a BATCH recipe step of the type RPH (recipe phase) and the user programs on the automation system for control and status information. The IEPH block is used to pass on the batch control commands to the user program (basic function blocks) and the status messages of the user encoded in the status word of the IEPH block for batch control.

The setpoints are set and the process values adopted via the parameter blocks (IEPAR_...) underlying the IEPH block. To allow this, the IEPAR blocks are interconnected with the EPE_CONN output of IEPH. Via this output, there is also check to establish whether the configured IEPH block instance and the IEPAR block instances match the definition of the phase type. The name of the phase type is located in the F_TYPE input parameter of the IEPH block.

The IEPH block allows the following status changes:



Legend for image

Object in image	Meaning
Closed objects	Final state
Dashed objects	Intermediate state
Several arrows at one object	Transition is possible from all active states except from "Idle" and "Completed" states.
Arrows without text	Automatic state transitions
Arrows with text in capital letters	State transitions by command, e.g., ABORT
Arrows with text in lower case letters and in parentheses	Block behavior

Operating principle

- **Allocate / Enable**
A recipe step can occupy the block by setting the identifier OCCUPIED = 1. It also enters an occupied information based on the batch name, batch number, the step number (UBA_NAME, UBA_ID, USTEP_NO) and, if necessary, the step monitoring time (STEP_T). The block is only occupied if enabled for BATCH (BA_EN = 1), if set to auto mode (status word bit 17 = 1), and if there is no active group error (status word bit 31 = 0).
- **Control commands**
The block is controlled by way of the ISTART, IHOLD, IABORT, ISTOP, ITERM, IRESET (for interconnection in CFC) control inputs, or by means of control word ICTRL (used only in BATCH). The active control commands can be read from the corresponding outputs, for example, QSTART and QHOLD. Control word ICTRL enables the consistent writing of control commands in BATCH. Example: ICTRL = 1 ensures that ISTART = 1 and IHOLD = 0 are written simultaneously. The control inputs are reset automatically after the corresponding end status was reached (reset of ISTART at the positive of the RUNNING bit in USTAT_L, for example).
- **Status word**
The user block connected to the IEPH block signals the reaction to operator input via the status input USTAT_L. The status – along with other block information – is available at the QUSTAT_L output and is monitored by BATCH. The bit assignment is synchronized with the assignment at the SFC_CTRL block.
- **No automatic completion**
The BATCH block sets the READY bit in the status word to report that it is not automatically completing and that BATCH is to handle completion. If using the SFC_CTRL block which does not provide a READY status you can set the READY bit using input IREADY.
- **Runtime monitoring**
The step time is incremented at intervals of one second with a positive edge at the RUNNING status. The COMPLETED status is used to stop the time count. The "Runtime exceeded" identifier is set (bit 19) if the actual value (Q_STEP_T) exceeds the setpoint (STEP_T). Runtime monitoring is taken into account as message in WinCC and does **not** generate a group error.

Note

Identifier "SFC: Step runtime error" is used internally by the SFC and leads to a group error.

- **Trigger for reading setpoints/actual values**
If the block is active the user program can set the trigger signal IREFRESH in QUSTAT_L by generating a positive edge at input IREFRESH. This action instructs BATCH to immediately read all setpoints and actual values and to archive these in the batch data for reporting.
- **Trigger for checking and activating new setpoints**
BATCH sets input IPARAM when reading new setpoints. The basic function detects this action by monitoring the corresponding QPARAM output and can then check and activate the setpoints. If the check returns a negative result the basic function can either signal an error or disable the start by setting input "ILOCK". BATCH resets input IPARAM during startup of the equipment phase. This can be done, for example, in conjunction with a change of the mode of operation.

- Start lock
The "Start lock" identifier in the status word or input "ILOCK" can be used to inform the BATCH block that start or resume is not desired.

Note

The start lock must have been set in IDLE state if the BATCH is to comply with the start lock. If start lock is set the BATCH executes the occupation it initially writes the setpoints and then waits for the block to reset the start lock.

- Continuous operation or bumpless changeover. In continuous operation, two recipe steps (time-based) successively use the same block without resetting this block between the steps. The block only reports the READY state at the end of the first cycle. BATCH then sets the status ID for continuous mode by setting input ICONT instead of terminating the block by setting ITERM. You configure this functionality when creating the recipes. ICONT is reset when the second recipe step is activated.
"Continuous operation or bumpless changeover" is only supported by recipe functions which are not automatically completed.

Allocation of ICTRL

Bit	Meaning
0	ISTART
1	IHOLD
2	IABORT
3	IRESET
4	ITERM
5	Reserved
6	ISTOP
7	Reserved
8	ICONT
9	IPARAM

ICTRL contains all control inputs which are written by BATCH.

IREADY, ILOCK and IREFRESH are used by the **user program** and are therefore not available in ICTRL.

Allocation of the status words USTAT_L and QUSTAT_L

Bit	Meaning	Remark	Set if
0	IDLE	Static state is "inactive"	
1	RUNNING	Static state is "running"	
2	COMPLETED	Static state is "completed"	
3	HELD	Static state is "held"	
4	ABORTED	Static state is "aborted"	
5	READY	Static state is "ready to complete"	IREADY = 1
6	STOPPED	Static state is "stopped"	

Bit	Meaning	Remark	Set if
7	Free	-	
8	Free	-	
9	STARTING	Transient state is "starting"	
10	RESUMING	Transient state is "resuming"	
11	COMPLETING	Transient state is "completing"	
12	HOLDING	Transient state is "holding"	
13	ABORTING	Transient state is "aborting"	
14	STOPPING	Transient state is "stopping"	
15	Free	-	
16	Reserved	(for SFC: Manual/Automatic mode requested)	
17	QCMOD	Identifier: Manual/Automatic (1 = automatic)	
18	BA_EN	Identifier: BATCH enable (1 = enabled)	BA_EN = 1
19	STEP_RT	Runtime exceeded identifier	Q_STEP_T > STEP_T
20	Reserved	(for SFC: Operator request transition)	
21	REFRESH	Trigger for reading setpoints / actual values	IREFRESH = 1
22	LOCK	Identifier: Start lock (1 = locked)	ILOCK = 1
23	CONTINUOUS	Identifier: Continuous operation	ICONT = 1
24	OCCUPIED	Identifier: BATCH allocation (1 = occupied)	OCCUPIED = 1
25	PROC_ERR	(for SFC: Process error)	
26	Reserved	(for SFC: Execution error)	
27	Reserved	(for SFC: Step runtime error)	
28	Reserved	(for SFC: Configuration error)	
29	Reserved	(for SFC: Operator input error)	
30	Reserved	(for SFC: External error)	
31	ERROR	Group error	

Rules for setting the status bits

- Static states are of highest importance in BATCH. The transition states are optional. The RUNNING state may be activated directly after the IDLE state, for example.
- Only one static state and one transient state may be active simultaneously.
Example:

Allowed	HELD on its own
Allowed	ABORTING on its own
Allowed	HELD and ABORTING (useful in order to be able to detect the original state)
Not allowed	HELD and ABORTED
Not allowed	HOLDING and ABORTING

Note

Exception: The READY state may be generated at another static state if the IREADY input is used.

- A group error (bit 31) may be set in addition to any other state.
- The corresponding USTAT_L bit does not have to be set for entries in the "set if" column.

Batch messages

The associated values are occupied by the block as follows:

Associated value 1 = batch name	UBA_NAME
Associated value 2 = step number	USTEP_NO
Associated value 3 = batch ID	UBA_ID

Block parameters	Value	Event	Message class
QUSTAT_L bit 1	1	Function is active	Status message - AS
QUSTAT_L bit 2	1	Function terminated	Status message - AS
QUSTAT_L bit 3	1	Function in hold	Status message - AS
QUSTAT_L bit 4	1	Function aborted	Status message - AS
QUSTAT_L bit 5	1	Phase ready to complete	Status message - AS
QUSTAT_L bit 6	1	Function stopped	Status message - AS
QUSTAT_L bit 17	0	Function in manual mode	Status message - AS
QUSTAT_L bit 18	0	Function not enabled for BATCH	Status message - AS
QUSTAT_L bit 18	1	Operation not enabled for BATCH	Status message - AS
QUSTAT_L bit 19	1	Function exceeded runtime	Status message - AS
QUSTAT_L bit 31	1	Function error	AS control system message

See also

IEPH message texts and associated values (Page 20)

2.2.1.2 I/O of IEPH**I/O**

I/O (parameter)	Meaning	Type	Default	Type	Attributes	OCM	Permitted values
BA_EN	BATCH - occupy enable	Bool	1	I		+	
EPE_CONN	Interconnection with IE-PAR blocks	DInt	0	O		-	
F_TYPE	Name of the phase type	String[16]	"	I		+	
IABORT	Control input: Cancel	Bool	0	IO		+	

I/O (parameter)	Meaning	Type	Default	Type	Attributes	OCM	Permitted values
ICONT	Sets the CONTINUOUS ID in the status word	Bool	0	IO		+	
ICTRL	Control word for control inputs	DWord	0	I	Q	+	
IHOLD	Control input: Hold	Bool	0	IO		+	
ILOCK	Sets the start lock ID in the status word	Bool	0	IO		+	
IPARAM	New batch control parameter set	Bool	0	IO		+	
IREADY	Sets the READY status bit in the status word	Bool	0	IO		+	
IREFRESH	Sets the REFRESH ID in the status word	Bool	0	IO	QU	+	
IRESET	Control input: Reset	Bool	0	IO		+	
ISTART	Control input: Start	Bool	0	IO		+	
ISTOP	Control input: Stop	Bool	0	IO		+	
ITERM	Control input: Complete	Bool	0	IO		+	
OCCUPIED	BATCH occupied ID	Bool	0	I	U	+	
Q_OCCUPI	Copy of OCCUPIED	Bool	0	O		-	
Q_STEP_T	Actual value of the step runtime	DInt	0	O	U	+	
QABORT	Copy of IABORT	Bool	0	O		-	
QBA_EN	Copy of BA_EN	Bool	1	O		+	
QCONT	Copy of ICONT	Bool	0	O		-	
QHOLD	Copy of IHOLD	Bool	0	O		-	
QLOCK	Copy of ILOCK	Bool	0	O		-	
QPARAM	Copy of IPARAM	Bool	0	O		-	
QREADY	Copy of IREADY	Bool	0	O		-	
QREFRESH	Copy of IREFRESH	Bool	0	O		-	
QRESET	Copy of IRESET	Bool	0	O		-	
QSTART	Copy of ISTART	Bool	0	O		-	
QSTOP	Copy of ISTOP	Bool	0	O		-	
QTERM	Copy of ITERM	Bool	0	O		-	
QSTAT_L	Status word output	DWord	0	O		+	
STEP_T	Setpoint for step run time in seconds	DInt	0	I	U	+	
UBA_ID	Batch ID	DWord	0	I		+	
UBA_NAME	Batch name	String[32]	"	I		+	
ULOOP_I	Number of step activations (for loops)	Word	1	I	U	+	
USTAT_L	Status word input	DWord	0	I	Q	-	
USTEP_NO	Step number in the recipe	DWord	0	I	U	+	
VBA_ID	Copy of UBA_ID	DWord	0	O		+	

I/O (parameter)	Meaning	Type	Default	Type	Attributes	OCM	Permitted values
VBA_NAME	Copy of UBA_NAME	String[32]	"	O		+	
VLOOP_I	Copy of ULOOP_I	Word	0	O		+	
VSTEP_NO	Copy of USTEP_NO	DWord	0	O		+	

See also

General information on the block description (Page 9)

2.2.1.3 IEPH message texts and associated values**Assigning a message text and message class to the block parameters**

Message no. MSG_EVID_1	Block parameters	Predefined message text	Message class	Can be suppressed by
1		-	no message	-
2	QUSTAT_L AND 0x2 = 0x2	Function is active	AS status (Status Message)	-
3	QUSTAT_L AND 0x4 = 0x4	Function terminated	AS status (Status Message)	-
4	QUSTAT_L AND 0x8 = 0x8	Function held	AS status (Status Message)	-
5	QUSTAT_L AND 0x10 = 0x10	Function aborted	AS status (Status Message)	-
6	((QUSTAT_L AND 0x20) = 0x20) OR IREADY	Phase ready to complete	AS status (Status Message)	-
7	QUSTAT_L AND 0x40 = 0x40	Function stopped	AS status (Status Message)	-
8	QUSTAT_L AND 0x10000 = 0x10000	Function in manual mode	AS status (Status Message)	-

Message no. MSG_EVID_2	Block parameters	Predefined message text	Message class	Can be suppressed by
1	NOT BA_EN	Function not enabled for BATCH	AS status (Status Message)	-
2	Q_STEP_T > STEP_T	Function runtime exceeded	AS status (Status Message)	-
3		-	No message	-
4	QUSTAT_L AND 0x80000000 = 0x80000000	Function error	Process control message F (Failure)	-
5	BA_EN	Function enabled for BATCH	AS status (Status Message)	-

Assigning the associated values to the block parameters MSG_EVID_1

Of the associated values of the message block, the first three have SIMATIC BATCH data assigned to them.

Associated value	Block parameters
1	BA_NA
2	STEP_NO
3	BA_ID
4	AUX_PR04
5	AUX_PR05
6	AUX_PR06
7	AUX_PR07
8	AUX_PR08
9	AUX_PR09
10	AUX_PR10

Assigning the associated values to the block parameters MSG_EVID_2

Associated value	Block parameters
1	BA_NA
2	STEP_NO
3	BA_ID
4	AUX_PR04
5	AUX_PR05
6	AUX_PR06
7	AUX_PR07
8	AUX_PR08
9	AUX_PR09
10	AUX_PR10

2.2.1.4 Controlling and monitoring IEPH

Operating and monitoring

The following table shows the assignment of the parameters of the IEPH block to the input / output fields of the corresponding faceplate for the following views:

- Function
- Batch
- Setpoints
- Materials

2.2 BATCH interface blocks

Display	Input/output field	User authorization	Block parameters
Function	"Status" display	-	QUSTAT_L
	LED "Runtime error"	-	QUSTAT_L bit 19
	"Error" LED	-	QUSTAT_L bit 31
	LED "Manual"	-	QUSTAT_L bit 17 (if = 0)
	Buttons "T", "H", "B": Status in string, hexadecimal, and binary format	-	-
	Operation and display "Function"		
	Enabled for BATCH		BA_EN bit 18
	Occupy		OCCUPIED bit 24
	"Start" button	Can only be operated if no batch is enabled and occupied and if WinCC has set user authorization level 5 or 6.	ISTART or ICTRL bit 0
	"Cancel" button		IABORT or ICTRL bit 2
	"Stop" button		ISTOP or ICTRL bit 6
	"Hold" button		IHOLD or ICTRL bit 1
	"Complete" button		ITERM or ICTRL bit 4
	"Reset" button		IRESET or ICTRL bit 3
	"Resume" button		ISTART or ICTRL bit 0
Batch	"Status" display	-	QUSTAT_L
	Buttons "T", "H", "B" Status in string, hexadecimal, and binary format	-	-
	"Batch" Operator and Display		
	Enabled for BATCH	-	BA_EN bit 18
	Occupy	-	OCCUPIED bit 24
	Batch no.	-	VBA_ID
	Batch name	-	VBA_NAME
	Step no.	-	VSTEP_NO
	Step time	-	STEP_T
	Number of loops	-	VLOOP_I
	Average number of steps	-	Q_STEP_T
Setpoints	Enabled for BATCH	-	BA_EN bit 18
	Tabular display of setpoints, for example, setpoint, actual value, unit and parameter name.	-	Each row of the list box represents an IEPAR_block which is interconnected at input EPE_CONN.

Materials	Enabled for BATCH	-	BA_EN bit 18
	Tabular display of materials: lists all outputs of the IEPAR_PI and IEPAR_PO blocks	-	Each list box represents an insertion material or a material yield.

Creating user authorizations

The operator enable option must be set to "Yes" in the "Other" section of the "Properties" tab in the object properties dialog of the IUNIT and IEPH/IEOP faceplates.

In order to operate the object, the operator must additionally have the authorization. The settings for the authorization level for each operator are made with the WinCC Explorer in the User Administrator:

- Operator input must be enabled at authorization level 5 (process control) or 6 (higher-priority process control).
- The authorization levels can be enabled as follows:
 - generally for all units (column "Enable" in the User Administrator)
 - unit specific (column "<Unit name>" in the User Administrator)

Create Batch faceplate from Template

In the Graphics Designer (WinCC Explorer) there is a Template available for selecting Batch faceplates in PCS 7 OS pictures.

1. Open the desired process picture in the Graphics Designer.
2. Open the template file "@Template_Batch.pdl" via File > Open.
3. Use the "IEPH" block icon to visualize technical functions in the faceplate.
4. Select the block symbol in the template file and copy it to your process picture.
5. Select the block symbol in your process picture.
6. In the window of the Dynamic Wizard, double click on the PCS 7 function "Connect faceplate to process tag" under "Standard Dynamics".
7. In the Dynamic Wizard click "Next" and then the search function "...".
8. In the tag selection dialog all instanced blocks are displayed. Select the desired block and click OK > Next > Finish.

Result: You have successfully installed the "IEPH" block icon for calling the corresponding faceplate in the process picture you edited.

2.2.2 IEOP

2.2.2.1 IEOP: Equipment operation - BATCH interface

Object name (class + number)

FB 253

Calling OBs

The OB in which you install the block (for example, OB1).

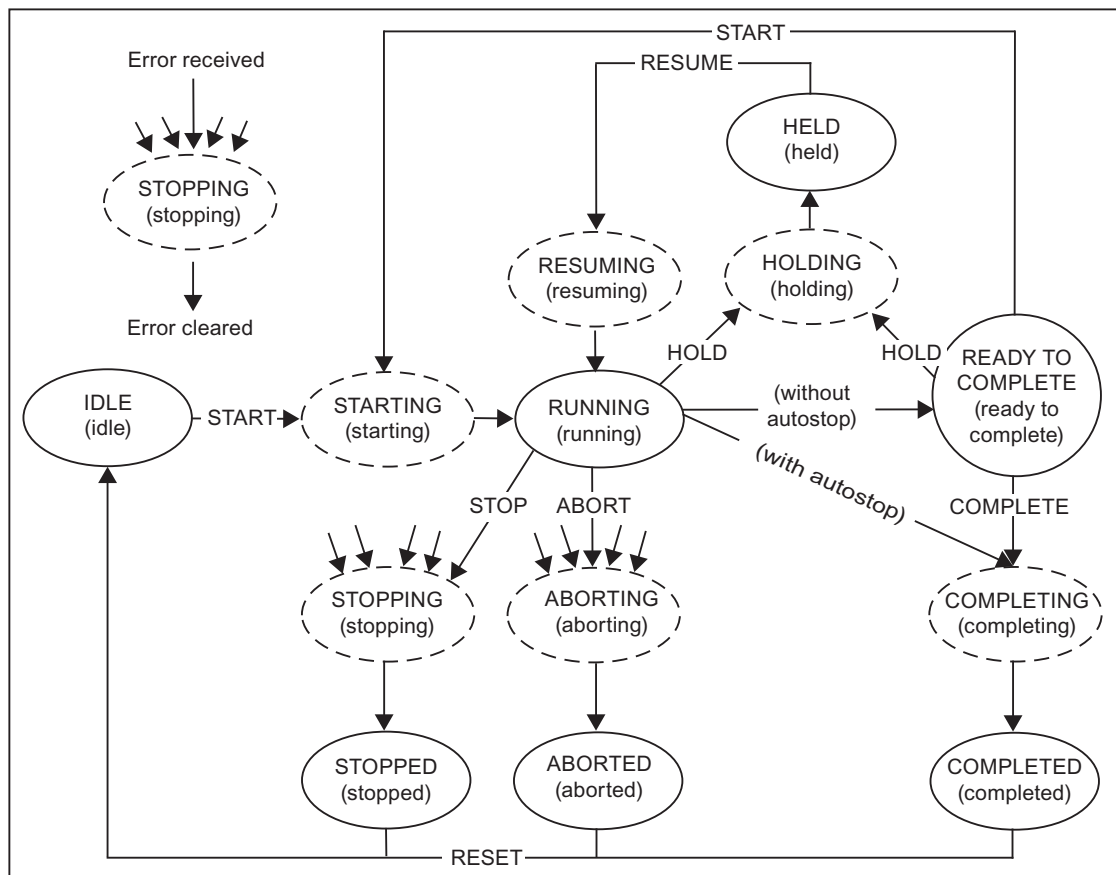
Function

The IEOP block is used when complex functions must be implemented in the automation system and these are to be executed in the BATCH recipe as a recipe operation step.

The IEOP block serves as an alternative to the BATCH recipe step of the ROP (recipe operation) type; a ROP thus runs completely separate in the automation system. The IEOP is installed directly at the position of the ROP in the recipe. On the one hand, the IEOP block is used to pass on the batch control commands to the user program, and on the other hand, the process values from the process are collected and passed on to batch control.

Setpoints are set and the process values adopted via the parameter blocks (IEPAR_...) underlying the IEOP block. To allow this, the IEPAR blocks are interconnected with the EPE_CONN output of IEOP. Via this output, there is also check later to establish whether the configured IEOP block instance and the IEPAR block instances match the definition of the phase type. The name of the phase type is located in the F_TYPE input parameter of the IEOP block.

The block allows the following status changes:



Legend for image

Object in image	Meaning
Closed objects	Final state
Dashed objects	Intermediate state

Object in image	Meaning
Many arrows at one state object	Transition is possible from all active states except from "Idle" and "Completed" states.
Arrows without text	Automatic state transitions
Arrows with text in capital letters	State transitions by command, e.g., ABORT
Arrows with text in lower case letters and in parentheses	Block behavior

Operating principle

- **Allocate / Enable**
A recipe step can occupy the block by setting the identifier OCCUPIED = 1. It also enters an occupied information based on the batch name, batch number, the step number (UBA_NAME, UBA_ID, USTEP_NO) and, if necessary, the step monitoring time (STEP_T). The block is only occupied if enabled for BATCH (BA_EN = 1), if set to auto mode (status word bit 17 = 1), and if there is no active group error (status word bit 31 = 0).
- **Control commands**
The block is controlled by way of the ISTART, IHOLD, IABORT, ISTOP, ITERM, IRESET (for interconnection in CFC) control inputs, or by means of control word ICTRL (used only in BATCH). The active control commands can be read from the corresponding outputs, for example QSTART. Control word ICTRL enables the consistent writing of control commands in BATCH. Example: ICTRL = 1 ensures that ISTART = 1 and IHOLD = 0 are written simultaneously. The control inputs are reset automatically after the corresponding end status was reached (reset of ISTART at the positive of the RUNNING bit in USTAT_L, for example).
- **Status word**
The user block connected to the IEOP signals the reaction to operator input via the status input USTAT_L. The status – along with other block information – is available at the QUSTAT_L output and is monitored by BATCH. The bit assignment is synchronized with the assignment at the SFC_CTRL block.
- **No automatic completion**
The BATCH block sets the READY bit in the status word to report that it is not automatically completing and that BATCH is to handle completion. If using the SFC_CTRL block (which does not provide a READY status signal) you can set the READY bit using input IREADY.
- **Runtime monitoring**
The step time is incremented at intervals of one second with a positive edge at the RUNNING status. The COMPLETED status is used to stop the time count. The "Runtime exceeded" identifier is set (bit 19) if the actual value (Q_STEP_T) exceeds the setpoint (STEP_T). Runtime monitoring is taken into account as message in WinCC and does **not** generate a group error.

Note

Identifier "SFC:
Step runtime error" is used internally by the SFC and leads to a group error.

- **Trigger for reading setpoints/actual values**
If the block is active the user program can set the trigger signal IREFRESH in QUSTAT_L by generating a positive edge at input IREFRESH. This action instructs BATCH to immediately read all setpoints and actual values and to archive these in the batch data for reporting.

- Trigger for checking and activating new setpoints
BATCH sets input IPARAM when reading new setpoints. The basic function detects this action by monitoring the corresponding QPARAM output and can then check and activate the setpoints. If the check returns a negative result the basic function can either signal an error or disable the start by setting input "ILOCK". BATCH resets input IPARAM during startup of the equipment phase. This can be done, for example, in conjunction with a change of the mode of operation.
- Start lock
The "Start lock" identifier in the status word or input "ILOCK" can be used to inform the BATCH block that start or resume is not desired.

Note

The start lock must have been set in IDLE state if the BATCH is to comply with the start lock. If start lock is set the BATCH executes the occupation it initially writes the setpoints and then waits for the block to reset the start lock.

- Continuous operation or bumpless changeover
When operating in continuous operation, two recipe steps (time-based) successively use the same block without resetting this block between the steps. The block only reports the READY state at the end of the first cycle. BATCH then sets the status ID for continuous mode by setting input ICONT instead of terminating the block by setting ITERM. ICONT is reset at the start of the second recipe step.
"Continuous operation or bumpless changeover" is only possible for recipe functions which are not automatically completed.

Allocation of ICTRL

Bit	Meaning
0	ISTART
1	IHOLD
2	IABORT
3	IRESET
4	ITERM
5	Reserved
6	ISTOP
7	Reserved
8	ICONT

ICTRL contains all control inputs which are written by batch.

IREADY, ILOCK and IREFRESH are used by the **user program** and are therefore not available in ICTRL.

Allocation of the status words USTAT_L and QUSTAT_L

Bit	Meaning	Remark	Set if
0	IDLE	Static state is "inactive"	
1	RUNNING	Static state is "running"	

Bit	Meaning	Remark	Set if
2	COMPLETED	Static state is "completed"	
3	HELD	Static state is "held"	
4	ABORTED	Static state is "aborted"	
5	READY	Static state is "ready to complete"	IREADY = 1
6	STOPPED	Static state is "stopped"	
7	Free	-	
8	Free	-	
9	STARTING	Transient state is "starting"	
10	RESUMING	Transient state is "resuming"	
11	COMPLETING	Transient state is "completing"	
12	HOLDING	Transient state is "holding"	
13	ABORTING	Transient state is "aborting"	
14	STOPPING	Transient state is "stopping"	
15	Free	-	
16	Reserved	(for SFC: Manual/Automatic mode requested)	
17	QCMOD	Identifier: Manual/Automatic (1 = automatic)	
18	BA_EN	BATCH enable ID (1 = enabled)	BA_EN = 1
19	STEP_RT	Identifier: Runtime exceeded	Q_STEP_T > STEP_T
20	Reserved	(for SFC: operator request transition)	
21	REFRESH	Trigger for reading setpoints / actual values	IREFRESH = 1
22	LOCK	Identifier: Start lock (1 = locked)	ILOCK = 1
23	CONTINUOUS	Identifier: Continuous operation	ICONT = 1
24	OCCUPIED	Identifier: BATCH allocation (1 = occupied)	OCCUPIED = 1
25	PROC_ERR	(for SFC: Process error)	
26	Reserved	(for SFC: Execution error)	
27	Reserved	(for SFC: Step runtime error)	
28	Reserved	(for SFC: Configuration error)	
29	Reserved	(for SFC: Operator input error)	
30	Reserved	(for SFC: External error)	
31	ERROR	Group error	

Rules for setting the status bits

- Static states are of highest importance in BATCH. The transition states are optional. The RUNNING state may be activated directly after the IDLE state, for example.
- Only one static state and one transient state may be active simultaneously.
Example:

Allowed	HELD on its own
Allowed	ABORTING on its own

Allowed	HELD and ABORTING (useful in order to be able to detect the original state)
Not allowed	HELD and ABORTED
Not allowed	HOLDING and ABORTING

Note

Exception: The READY state may be set additive at another static state if the IREADY input is used.

- A group error (bit 31) may be set in addition to any other state.
- The corresponding USTAT_L bit does not have to be set for entries in the "set if" column.

Batch messages

The associated values are occupied by the block as follows:

Associated value 1 = batch name	UBA_NAME
Associated value 2 = step number	USTEP_NO
Associated value 3 = batch ID	UBA_ID

Block parameters	Value	Event	Message class
QUSTAT_L bit 1	1	Operation in active state	Status message - AS
QUSTAT_L bit 2	1	Operation completed	Status message - AS
QUSTAT_L bit 3	1	Operation in hold state	Status message - AS
QUSTAT_L bit 4	1	Operation aborted	Status message - AS
QUSTAT_L bit 5	1	Operation ready to complete	Status message - AS
QUSTAT_L bit 6	1	Operation stopped	Status message - AS
QUSTAT_L bit 17	0	Operation in manual mode	Status message - AS
QUSTAT_L bit 18	0	Operation not enabled for BATCH	Status message - AS
QUSTAT_L bit 18	1	Operation enabled for BATCH	Status message - AS
QUSTAT_L bit 19	1	Operation runtime exceeded	Status message - AS
QUSTAT_L bit 31	1	Operation error	AS process control system message

See also

IEOP message texts and associated values (Page 30)

2.2.2.2 I/Os of IEOP

I/Os

I/O (parameter)	Meaning	Type	Default	Type	Attributes	OCM	Permitted values
BA_EN	BATCH - occupy enable	Bool	1	I		+	
EPE_CONN	Interconnection with IEPAR blocks	DInt	0	O		-	
F_TYPE	Name of the phase type	String[16]	"	I		+	
IABORT	Control input: Cancel	Bool	0	IO		+	
ICONT	sets the CONTINUOUS ID in the status word	Bool	0	IO		+	
ICTRL	Control word for control inputs	DWord	0	I	Q	+	
IHOLD	Control input: Hold	Bool	0	IO		+	
ILOCK	sets the start lock ID in the status word	Bool	0	IO		+	
IPARAM	New batch control parameter set	Bool	0	IO		+	
IREADY	sets the READY status bit in the status word	Bool	0	IO		+	
IREFRESH	sets the REFRESH ID in the status word	Bool	0	IO	QU	+	
IRESET	Control input: Reset	Bool	0	IO		+	
ISTART	Control input: Start	Bool	0	IO		+	
ISTOP	Control input: Stop	Bool	0	IO		+	
ITERM	Control input: Complete	Bool	0	IO		+	
OCCUPIED	BATCH occupied ID	Bool	0	I	U	+	
Q_OCCUPI	Copy of OCCUPIED	Bool	0	O		-	
Q_STEP_T	Actual value of the step run-time	DInt	0	O	U	+	
QABORT	Copy of IABORT	Bool	0	O		-	
QBA_EN	Copy of BA_EN	Bool	1	O		+	
QCONT	Copy of ICONT	Bool	0	O		-	
QHOLD	Copy of IHOLD	Bool	0	O		-	
QLOCK	Copy of ILOCK	Bool	0	O		-	
QPARAM	Copy of IPARAM	Bool	0	O		-	
QREADY	Copy of IREADY	Bool	0	O		-	
QREFRESH	Copy of IREFRESH	Bool	0	O		-	
QRESET	Copy of IRESET	Bool	0	O		-	
QSTART	Copy of ISTART	Bool	0	O		-	
QSTOP	Copy of ISTOP	Bool	0	O		-	
QTERM	Copy of ITERM	Bool	0	O		-	
QSTAT_L	Status word output	DWord	0	O		+	

I/O (parameter)	Meaning	Type	Default	Type	Attributes	OCM	Permitted values
STEP_T	Step runtime setpoint in seconds	DInt	0	I	U	+	
UBA_ID	Batch ID	DWord	0	I		+	
UBA_NAME	Batch name	String[32]	"	I		+	
ULOOP_I	Number of step activations (for loops)	Word	1	I	U	+	
USTAT_L	Status word input	DWord	0	I	Q	-	
USTEP_NO	Step number in the recipe	DWord	0	I	U	+	
VBA_ID	Copy of UBA_ID	DWord	0	O		+	
VBA_NAME	Copy of UBA_NAME	String[32]	"	O		+	
VLOOP_I	Copy of ULOOP_I	Word	0	O		+	
VSTEP_NO	Copy of USTEP_NO	DWord	0	O		+	

See also

General information on the block description (Page 9)

2.2.2.3 IEOP message texts and associated values

Assigning a message text and message class to the block parameters

Message no. MSG_EVID_1	Block parameters	Predefined message text	Message class	Can be suppressed by
1	-	-	no message	-
2	QUSTAT_L AND 0x2 = 0x2	Operation in active state	AS status (Status Message)	-
3	QUSTAT_L AND 0x4 = 0x4	Operation completed	AS status (Status Message)	-
4	QUSTAT_L AND 0x8 = 0x8	Operation held	AS status (Status Message)	-
5	QUSTAT_L AND 0x10 = 0x10	Operation aborted	AS status (Status Message)	-
6	((QUSTAT_L AND 0x20) = 0x20) OR IREADY	Operation ready to complete	AS status (Status Message)	-
7	QUSTAT_L AND 0x40 = 0x40	Operation stopped	AS status (Status Message)	-
8	QUSTAT_L AND 0x10000 = 0x10000	Operation in manual mode	AS status (Status Message)	-

Message no. MSG_EVID_2	Block parameters	Predefined message text	Message class	Can be suppressed by
1	NOT BA_EN	Operation not enabled for BATCH	AS status (Status Message)	-
2	Q_STEP_T > STEP_T	Operation runtime exceeded	AS status (Status Message)	-
3		-	No message	-
4	QUSTAT_L AND 0x80000000 = 0x80000000	Operation error	Process control message F (Failure)	-
5	BA_EN	Operation enabled for BATCH	AS status (Status Message)	-
6		-	No message	-
7		-	No message	-
8		-	No message	-

Assigning the associated values to the block parameters MSG_EVID_1

Of the associated values of the message block, the first three have SIMATIC BATCH data assigned to them.

Associated value	Block parameters
1	BA_NA
2	STEP_NO
3	BA_ID
4	AUX_PR04
5	AUX_PR05
6	AUX_PR06
7	AUX_PR07
8	AUX_PR08
9	AUX_PR09
10	AUX_PR10

Assigning the associated values to the block parameters MSG_EVID_2

Associated value	Block parameters
1	BA_NA
2	STEP_NO
3	BA_ID
4	AUX_PR04
5	AUX_PR05
6	AUX_PR06
7	AUX_PR07
8	AUX_PR08

Associated value	Block parameters
9	AUX_PRO9
10	AUX_PR10

2.2.2.4 Operator Control and Monitoring of IEOP

Operating and monitoring

The following table shows the assignment of the parameters of the IEOP block to the input/output fields of the corresponding faceplates for the following views:

- Function
- Batch
- Setpoints
- Materials

Display	Input/output field	User authorization	Block parameters
Function	"Status" display	-	QUSTAT_L
	LED "Runtime error"	-	QUSTAT_L bit 19
	"Error" LED	-	QUSTAT_L bit 31
	LED "Manual"	-	QUSTAT_L bit 17 (if = 0)
	"T", "H" and "B" buttons: Status in string, hexadecimal, and binary format	-	-
	Operation and display "Function"		
	Enabled for BATCH		BA_EN bit 18
	Occupy		OCCUPIED bit 24
	"Start" button	Can only be operated if no batch is enabled and occupied and if WinCC has set user authorization level 5 or 6.	ISTART or ICTRL bit 0
	"Cancel" button		IABORT or ICTRL bit 2
	"Stop" button		ISTOP or ICTRL bit 6
	"Hold" button		IHOLD or ICTRL bit 1
	"Complete" button		ITERM or ICTRL bit 4
	"Reset" button		IRESET or ICTRL bit 3
	"Resume" button		ISTART or ICTRL bit 0

Display	Input/output field	User authorization	Block parameters
Batch	"Status" display	-	QUSTAT_L
	Buttons "T", "H", "B" Status in string, hexadecimal, and binary format	-	-
	"Batch" Operator and Display		
	Enabled for BATCH	-	BA_EN bit 18
	Occupy	-	OCCUPIED bit 24
	Batch no.	-	VBA_ID
	Batch name	-	VBA_NAME
	Step no.	-	VSTEP_NO
	Step time	-	STEP_T
	Number of loops	-	VLOOP_I
	Average number of steps	-	Q_STEP_T
Setpoints	Enabled for BATCH	-	BA_EN bit 18
	Tabular display of setpoints: For example, the setpoint, actual value, unit and the parameter name.	-	Each row of the list box represents an IEPAR_ block which is interconnected at input EPE_CONN.
Materials	Enabled for BATCH	-	BA_EN bit 18
	Tabular display of materials: lists all outputs of the IEPAR_PI and IEPAR_PO blocks.	-	Each list box represents an insertion material or a material yield.

Setting user authorizations

You can generally enable the control of objects for yourself in the "Properties/Other" tab of the object properties dialog of the UNIT and EPH/EOP faceplates.

- Yes - the object can be operated in process mode.
- No - the object can be controlled in process mode.
The operator must also be authorized in order to be able to control an object. The settings for the authorization level for each operator are made with the WinCC Explorer in the User Administrator:
- You can only select access levels 5 (process control) and 6 (higher-priority process control).
- The authorization levels can be enabled as follows:
 - generally for all units (column "Enable" in the User Administrator)
 - unit specific (column "<Unit name>" in the User Administrator)

Create Batch faceplate from Template

In the Graphics Designer (WinCC Explorer) there is a Template available for selecting Batch faceplates in PCS 7 OS pictures.

1. Open the desired process picture in the Graphics Designer.
2. Open the template file "@Template_Batch.pdl" via File > Open.

3. Use the "IEOP" block icon to visualize technical operations in the faceplate.
4. Select the block symbol in the template file and copy it to your process picture.
5. Select the block symbol in your process picture.
6. In the window of the Dynamic Wizard, double click on the PCS 7 function "Connect faceplate to process tag" under "Standard Dynamics".
7. In the Dynamic Wizard click "Next" and then the search function "...".
8. In the tag selection dialog all instanced blocks are displayed. Select the desired block and click OK > Next > Finish.

Result: You have successfully installed the "IEOP" block icon for calling the corresponding faceplate in the process picture you edited.

2.2.3 TAG_COLL

2.2.3.1 TAG_COLL: TAG_Collect interface BATCH

Object name (class + number)

FB 252 (Page 35)

Calling OBs

The OB in which you install the block (e.g. OB1).

Function

The block serves as a collector block for the IEPARs that are used in transition conditions in the BATCH recipe editor and that are not available as IEPAR in the IEPH. Process tags can also be included as parameters that are used in the batch measured value acquisition. The parameters of TAG_COLL are purely process values and the SP_VAL inputs are therefore not written. Only ACT_VAL and QACT_VAL are significant.

Operating principle

Display only

2.2.3.2 I/O of TAG_COLL

I/O

I/O (parameter)	Meaning	Type	Default	Type	Attributes	OCM
EPE_CONN	Interconnection with IEPAR blocks	DInt	0	O		
F_TYPE	Name of the process tag type	String[16]	"	I		+

See also

General information on the block description (Page 9)

2.2.4 UNIT_PLC

2.2.4.1 UNIT_PLC: Unit allocation - BATCH interface

Object name (class + number)

FB 266 (Page 46)

Calling OBs

The OB in which you install the block (for example, OB 32).

Function

Using UNIT_PLC block, batch control coordinates the allocation of individual units to the active batches. One instance of the UNIT_PLC block in one of the hierarchy folder charts is required for each unit.

The "Compile" function in the BATCH configuration dialog copies the unit name from the plant hierarchy in SIMATIC Manager to the block input "UNIT_NAME".

Batch messages

Event	Message class
Unit allocated	Status message - AS
Unit enabled	Status message - AS
Unit not enabled for BATCH	Status message - AS
Unit enabled for BATCH	Status message - AS
Unit registered for maintenance	Status message - AS
Unit error	AS process control system message

Event	Message class
Unit batch operation requested	Operating message
Missing equipment DB	AS process control system message - Problem
Command entry with: e.g. ISA 88 command	AS process control system message - Problem
The time stamps from the EQM DB (data block for equipment properties) and SP DB (setpoint data block) are not the same.	AS process control system message - Problem
The batch processing time ("Time calculated by system") ms exceeds the max. limit ("Input: Max_CycleRun") ms.	Process message with acknowledgement
L_ICTRL command, e.g. ISA 88 command, activated for STEP_NO: Batch step number	Status message - AS
L_ICTRL command, "e.g. ISA 88 command", not permitted for STEP_NO: Batch step number	Status message - AS

See also

Operating and Monitoring of UNIT_PLC (Page 54)
 Occupy and release (Page 36)
 Status word (Page 37)
 User Status Word (Page 38)
 Deactivating a Unit (Page 38)
 Command interface for RPEs in a unit (Page 39)
 Command interface to influence the entire unit (Page 40)
 Settings for communication monitoring and communication buffer (Page 41)
 EQM DB consistency check (Page 43)
 Recipe processing (Page 45)
 Diagnostics Messages (Page 45)
 UNIT_PLC message texts and associated values (Page 52)

2.2.4.2 Occupy and release**Operating principle**

The batch control sets the identifier OCCUPIED = 1 to occupy a unit. It also enters the occupation information as follows:

- Batch name, batch ID (UBA_NAME, UBA_ID),
- Recipe name, recipe version, formula (URP_NAME, URP_REL, FORMULA),
- Material name and material ID (MAT_NAME, MAT_SP) of the product made with the batch.

The block is only occupied if enabled for BATCH (BA_EN = 1 and L_BA_EN=1) and if there is no active group error (status word bit 31 = 0).

When enabling, the code OCCUPIED = 0 is set. In addition, the batch name and ID are reset.

The other occupation information for the product is retained and can be taken into account both in the user program and when planning additional batches, unless you have activated the "Delete all text entries in the unit block upon release" option that appears under "Units" on the "System characteristics" tab under "Settings" in the BATCH configuration dialog. If the occupation information is retained, you may not need to include any cleaning batches if the product remains the same, for example.

See also

UNIT_PLC: Unit allocation - BATCH interface (Page 35)

2.2.4.3 Status word

Operating principle

The status of UNIT_PLC is entered in the Q_STATUS status word and is monitored by BATCH.

Definition of the Q_Status

Bit	Meaning	Remark
0	Step control by operator	
1	Step control by transition	
2	Step control by operator and transition	
3	Step control by condition or transition	
4	Not used	
5	Not used	
6	Step control mode affects the phase level	
7	Step control mode affects the operation level	
8		Bits 8 to 15 as byte
9		The following RUP status values are possible:
10		RPE_STATE_UNDEFINED = 0
11		RPE_STATE_EDITED = 1
12		RPE_STATE_IDLE = 2
13		RPE_STATE_STARTING = 3
14		RPE_STATE_RUNNING = 4
15		RPE_STATE_READY = 5
		RPE_STATE_COMPLETING = 6
		RPE_STATE_COMPLETED = 7
		RPE_STATE_CONTINUE = 8
		RPE_STATE_ABORTED = 9
		RPE_STATE_ABORTING = 10
		RPE_STATE_STOPPED = 11
		RPE_STATE_STOPPING = 12
		RPE_STATE_HELD = 13
		RPE_STATE_HOLDING = 14
		RPE_STATE_RESUMING = 15
		RPE_STATE_PAUSED = 16
		RPE_STATE_PAUSING = 17

Bit	Meaning	Remark
16	Reserved	
17	Reserved	
18	Reserved	
19	Reserved	
20	Reserved	
21	Reserved	
22	Free	
23	Free	
24	OCCUPIED	UNIT_PLC is allocated
25	BA_EN	Enabled for BATCH
26	PEND_OOS	Pending out of Service; BA_EN = 0 is set at the earliest opportunity (allocation is canceled).
27	Free	
28	Free	
29	Free	
30	Free	
31	ERROR	Group error

See also

UNIT_PLC: Unit allocation - BATCH interface (Page 35)

2.2.4.4 User Status Word**Operating principle**

The USER_STATUS is a status word that can be freely set by other PCS 7 applications and is available on the PCS 7 OS as QUSER_STATUS. The status can be requested by the corresponding application programs on this OS. The status word may not be used by a user program.

See also

UNIT_PLC: Unit allocation - BATCH interface (Page 35)

2.2.4.5 Deactivating a Unit**Operating principle**

UNIT_PLC allows the deliberate deactivation of the respective unit for use by batch control. Input PEND_OOS (pending out of service) sets an internal trigger. BA_EN is immediately set to 0 if the unit is not occupied. If a batch is currently occupied (OCCUPIED = 1) UNIT_PLC waits for the batch control to cancel this allocation and then sets BA_EN = 0.

See also

UNIT_PLC: Unit allocation - BATCH interface (Page 35)

2.2.4.6 Command interface for RPEs in a unit

Note

The following functionality is only supported in AS-based mode.

"L_ICTRL" / "L_ICTRL_STEP_NO" and "OS_ICTRL" / "OS_ICTRL_STEP_NO" operating principle

Use the interface to set an ISA-88 command at an RPE in the associated unit that is currently being processed by the block "UNIT_PLC". Both input parameters "L_ICTRL" / "L_ICTRL_STEP_NO" serve as interfaces for the automation program. Both input parameters "OS_ICTRL" / "OS_ICTRL_STEP_NO" serve as interfaces for the Operator Station (OS).

You can transfer ISA-88 commands to both inputs "L_ICTRL" or "OS_ICTRL". An ISA-88 command is transferred to the RPE with the step number that you set at the inputs L_ICTRL_STEP_NO or OS_ICTRL_STEP_NO. The transferred commands with the step numbers can only be transferred in a block processing cycle. A new transfer may only take place after a feedback via L_ICTRL_OUT.

It may happen in a block processing cycle that the OS interface (OS_ICTRL and OS_ICTRL_STEP_NO) and the CFC interface receive commands at the same time. In this case, the commands for the CFC interface have a higher priority and the commands for the OS interface are deleted.

The result is returned to the output "L_ICTRL_OUT". If "L_ICTRL_OUT" < 0, the command at the current RPE status was not permitted or the RPE had already received commands with a higher priority. If the necessary command is set, the block will transfer the command from "L_ICTRL" or "OS_ICTRL" to "L_ICTRL_OUT" (always > 0). When the RPE reaches the required status, all "L_ICTRL_x" and all "OS_ICTRL_x" inputs and output are set to 0. A PCS 7 OS message with the result for each command is issued.

You receive the status of the respective unit via bits 0 to 7 of the 32 bit output via "QContainerID_SW" in the table in section "Command interface to influence the entire unit (Page 40)".

Consistency check at START command

The START command in recipe elements is permitted only if all RPEs in the RUP have the status "completed" or "inactive". The target object must also have the status "inactive".

Note

During the consistency check of the START command by the block "UNIT_PLC", it cannot be determined - in contrast to operation in BCC - whether the command should be set for an element in a parallel branch. This may lead to the rejection of the START command during the consistency check, which does not happen in BCC.

Table for inputs "L_ICTRL" or "OS_ICTRL"

The ISA-88 commands are transferred with the values stated in the table.

ISA-88 command	Value
No command	0
Start	1
Stop	2
Abort	3
Hold (immediately)	4
Reset (not possible for RUP)	5
Resume	6
Complete	7
Pause (after step)	8

See also

UNIT_PLG: Unit allocation - BATCH interface (Page 35)

2.2.4.7 Command interface to influence the entire unit**Note**

The following functionality is only supported in AS-based mode.

Operating principle

Influence on the entire unit via the command interface inputs "CmdPropagation" and "TriggerCmdPropagation"

You can configure the following commands at the "CmdPropagation" parameter:

- No command: No command for the unit that is just being processed.
- Hold: The unit currently being processed enters the "Hold" status.
- Resume: The unit currently being processed enters the "Run" status.
- Stop: The unit currently being processed enters the "Stop" status.
- Cancel: The unit currently being processed enters the "Cancel" status.

Note

The value for the input "CmdPropagation" is available in the command enumeration table in the chapter "Command interface for RPEs in a unit (Page 39)".

The command configured at the "CmdPropagation" parameter will be run only if the "TriggerCmdPropagation" parameter detects a positive edge. The status of the respective unit will be reported via bits 0 to 7 of the 32 bit output via "QContainerID_SW".

Bit	Name	Data type	Comment
0-7	RPE status	BYTE	RPE_STATE_UNDEFINED = 0 RPE_STATE_EDITED= 1 RPE_STATE_IDLE= 2 RPE_STATE_STARTING = 3 RPE_STATE_RUNNING = 4 RPE_STATE_READY = 5 RPE_STATE_COMPLETING = 6 RPE_STATE_COMPLETED = 7 RPE_STATE_CONTINUE = 8 RPE_STATE_ABORTED = 9 RPE_STATE_ABORTING = 10 RPE_STATE_STOPPED = 11 RPE_STATE_STOPPING = 12 RPE_STATE_HELD = 13 RPE_STATE_HOLDING = 14 RPE_STATE_RESUMING = 15 RPE_STATE_PAUSED = 16 RPE_STATE_PAUSING = 17

RPE status enumeration

2.2.4.8 Settings for communication monitoring and communication buffer

Introduction

Communication monitoring helps the user program respond to a missing connection to batch control.

The communication buffer saves all BATCH data (for example, status changes).

Communication monitoring

The unit block provides outputs to determine the communication status between PCS 7 OS and the BATCH Control Server. The outputs are called:

- AS-OS_ERR: Lifebeat monitoring for the communication between AS and the BATCH Control Server via PCS 7 OS.
- Q_ERR_COM: Communication status monitoring between AS and BATCH Control Server.

Operating principle of output "AS_OS_ERR"

If the lifebeat does not start after an interval you set with BATCH, then the block UNIT_PLC sets the output AS_OS_ERR = 0. The user program can then respond to the missing connection to batch control. Output AS_OS_ERR is reset to 1 after the connection is re-activated (and transmission of the Lifebeat" signal has resumed).

- SP_COUNT = 0: deactivated/VA_COUNT constant 1
- SP_COUNT > 0: VA_COUNT is decremented at intervals of seconds, starting at SP_COUNT.

If VA_COUNT = 0, the monitoring time has elapsed; in other words, there is no contact to the PCS 7 OS or batch control.

If LIFE = TRUE, monitoring is restarted at SP_COUNT.

Note

Time monitoring is active only when the unit is occupied by a batch. If a unit is occupied manually using the UNIT faceplate, the UNIT watchdog is not activated. You will find more information in the online help for SIMATIC BATCH in the chapter Redundancy "Time monitoring in the PCS 7 OS."

Operating principle of output "Q_ERR_COM"

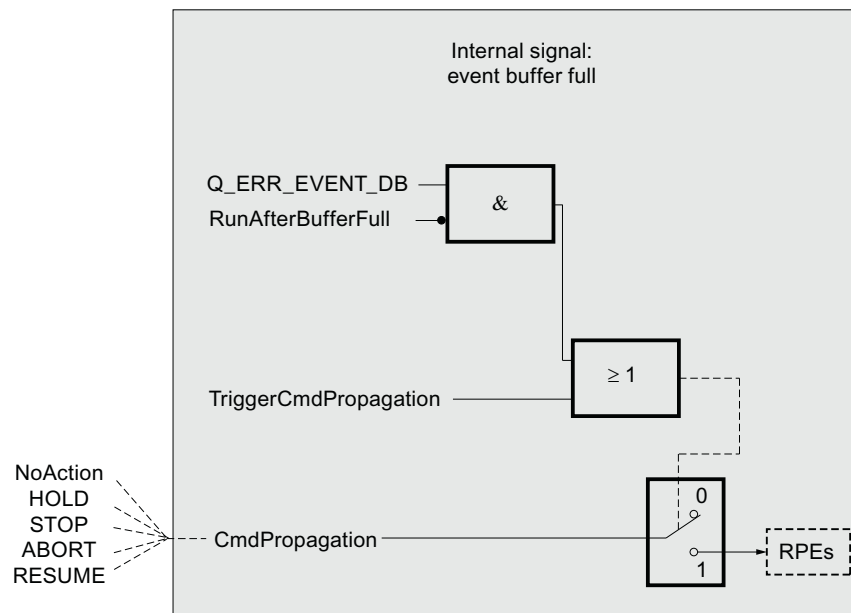
The output indicates if there is direct communication with the BATCH Control Server. If the output is set to "1" by the system, communication will be interrupted; the value "0" indicates that communication is established. In case of a BATCH Control Server failure, a started RUP will be processed until the end or up to the recipe instance requiring a request from the BATCH Control Server (for example, synchronization, operator dialog). All status changes of the recipe elements will be saved in their own AS communication buffer for a certain time. The following paragraph describes the further processing of the RUP in case of an error of the BATCH Control Server.

Operating principle of the communication buffer

The status messages of a recipe element (RUP, ROP, RPH, etc.) are collected in a special buffer in the AS and sent to the BATCH Control Server (BCS). If this buffer is full, for example, because communication to the BCS has failed, the output parameter "Q_ERR_EVENT_DB" is set to 1 and the parameter "RunAfterBufferFull" determines further recipe processing:

- RunAfterBufferFull = 0
The current RUP will execute the command configured at the parameter "CmdPropagation" immediately.
- RunAfterBufferFull = 1
The current batch will be executed until the end, even if this runs the potential risk of status messages (e.g. ReadyToComplete, Completed etc.) being lost.

The following picture shows the associated signal logic:



See also

UNIT_PLG: Unit allocation - BATCH interface (Page 35)

2.2.4.9 EQM DB consistency check

Operating principle

The EQM DB will be checked for consistency every time it is loaded to the AS. If the data in the EQM DB are inconsistent, the unit will be locked (block input "BA_EN" reset to 0).

What results in inconsistencies?

If a unit is allocated by a batch and batch-related data are altered in the CFC, for example by adding or deleting a new recipe function (SFCTyp / IEPH) or a new setpoint (IEPAR) with subsequent CFC loading, the content of the EQM-DB no longer corresponds to the block instances in the AS.

The running batch in the unit becomes faulty (BA_EN=0 unit no longer enabled for BATCH) and will be stopped.

Consequence of the inconsistency

If a batch is currently being processed, recipe processing will be stopped and a PCS 7 OS message generated. The block output "MissingDBCount" contains information about the incorrect data in the EQM DB.

Correcting the inconsistency

Note

The following function is reserved for service technicians.

- Trigger: Proceed as follows when adding new recipe functions or setpoints to release the unit to continue its batch.
 - Trigger: Deleting existing recipe functions or setpoints that are not to be used any longer when continuing the batch (no longer present in recipe).
1. Open the BATCH configuration dialog in the SIMATIC manager and compile the Batch instances once again. Then upload the data to SIMATIC BATCH.
 2. Compile the CFC charts and load them to the AS. Make sure that the option "Include user data blocks when loading" is active.
 3. Open the corresponding unit block in the CFC Editor and set the invisible block input "CopyEQMDB" from default "0" to "1".
 4. Verify that the unit has been enabled for BATCH. The input/output "BA_EN" is set to the value "1". The unit is once again released for BATCH.
 5. Navigate to the incorrect recipe element in BCC and click on "Resume step" in the shortcut menu.

Deleting recipe functions while continuing the batch

When deleting existing recipe functions or setpoints and subsequent loading to SIMATIC BATCH will cause the batch or unit to fail. Use "Cancel" command to terminate.

See also

UNIT_PLC: Unit allocation - BATCH interface (Page 35)

2.2.4.10 Recipe processing

Operating principle

- **Recipe processing**
Each UNIT_PLC block manages exactly one unit in the AS. This block is the central block for batch control. Every ROP, RF or transition in the assigned unit is checked by this block and the specifications for further processing steps are defined. The more ROPs, RFs or transitions included in a recipe unit procedure (RUP), the longer the time needed to process the recipe. To allow the plant configuration staff to optimally divide the time between processing time (actuators, sensors, process logic, etc.) and time for processing recipes, there are two parameters on the UNIT_PLC:
- **MaxCyclRun**
Maximum time per processing cycle that may be used to process batch control.
- **QAvgRecProcTime**
Average time per processing cycle that is currently used to process batch control. Two measured values (average calculated time from the last cycle + measurement from the present cycle) are taken into account for each calculation.

Additional information is available in the chapter "Installation of the UNIT_PLC block" in the online help or the manual under SIMATIC BATCH AS based > Batches.

See also

UNIT_PLC: Unit allocation - BATCH interface (Page 35)

2.2.4.11 Monitoring time of a recipe operation

Operating principle

You have the option to enter a monitoring time for a recipe operation (ROP) in the Recipe Editor. You can monitor this time in the BATCH Control Center or in BATCH Control (OCX) "SIMATIC BATCH OS Properties" within an PCS 7 OS process picture.

If the time required for processing the recipe operation is greater than the set monitoring time, the status "Time has elapsed" is set in the BatchCC. In addition, the block output "Q_ErrContMonTime" is set in AS-based operation if at least one active recipe operation exceeds the set monitoring time.

2.2.4.12 Diagnostics Messages

Operating principle

Diagnostic messages are reserved for service technicians.

While a recipe is being processed, it is possible to transfer diagnostic messages from a block to the BATCH Server. You turn on the diagnostic messages with the two block inputs "DebugTrace" and "TraceSTEP_NO".

Note

Use of this function will make high demands on recipe processing as well as system performance. During standard production, the block input "DebugTrace" must be set to 0.

Bit number	Name	Diagnostic type	Comment
0	Spare	-	Not used
1	All	1	All blocks generating diagnostic messages write messages.
2	OneRPEOnly	2	Only an RPE with specified step number (input "TraceSTEP_NO" at UNIT_PLC block) generates the diagnostic message.
3	NOPControl	1	All NOP generate diagnostic messages.
4	RPHControl	1	All RF instances generate diagnostic messages.
5	TransControl	1	All transitions generate diagnostic messages.
6	ContControl	1	All structures (ROP, RUP) generate diagnostics messages.
7	SubFunctions	1	All substructures generate diagnostic messages.

Diagnostic settings at the block input "DebugTrace"

See also

UNIT_PLC: Unit allocation - BATCH interface (Page 35)

2.2.4.13 UNIT_PLC connectors

I/O

I/O (parameter)	Meaning	Type	Default	Class	Attributes	OCM	Permitted values
AS_ID	Configured AS ID	Int		I	U	-	
AS_OS_ERR	Lifebeat monitoring	Bool	0	O	U	+	
BA_CTRL	Not used in this version.	DWord	16#0	IO	U	+	
BA_EN	BATCH occupied enable	Bool	1	IO		+	
BA_INFO	BATCH data structure	Struct	-	O	QU	-	

I/O (parameter)	Meaning	Type	Default	Class	Attributes	OCM	Permitted values
BA_INFO.Q_Status	Block status for SIMATIC BATCH	DWord	0	O	U	-	
BA_INFO.STEP_NO	RUP step number	DWord	0	O	U	-	
BA_INFO.UNIT_NAME	Unit name	String[24]	"	O	U	-	
BA_INFO.VBA_ID	Batch ID (output)	DWord	0	O	U	-	
BA_INFO.VBA_NAME	Batch name (output)	String[32]	"	O	U	-	
CmdPropagation	Predefinition for command	Int	Hold	I	Q	-	HOLD, ABORT, STOP, RESUME, NoCommand
CopyEQMDB	Copies EQM_DB to IDB	Bool	0	I/O	U	-	
EN	1= Called block is processed	Bool	1	I		-	
EN_LinkedUnitStatus	Value = 0, then UnitStatus at Q_UnitStatus; value = 1, then L_UnitStatus at Q_UnitStatus	Bool	0	I	Q	+	
ENO	1= Block algorithm executed without errors	Bool	0	I	U	-	
FastSFC	Activates fast SFC mode	Bool	1	I	QU	-	
FORMULA	Formula	String[32]	"	I	U	+	
FORMULA_REL	Formula version Not used in this version.	String[32]	"	I	U	+	
ICTRL	Batch Control WORD	Word	"	I/O	U	+	
L_BA_EN	Interconnectable BATCH occupied enable	Bool	1	I/O	QU	+	
L_ICTRL	Interconnectable RPE command	Int	0	I/O	U	+	
L_ICTRL_STEP_NO	Interconnectable step number	DInt	0	I/O	QU	+	

I/O (parameter)	Meaning	Type	Default	Class	Attributes	OCM	Permitted values
L_ICTRL_OUT	Output of the results from the inputs L_ICTRL / OS_ICTRL	Int	0	O	QU	+	
L_UnitStatus	Interconnectable unit status	DWord	0	I	Q		
LIFE	Lifebeat bit (SP_COUNT is applied to VA_COUNT)	Bool	0	I/O	QU	+	
MAT_NAME	Material name	String[32]	"	I		+	
MAT_SP	Material ID of the batch product	String[16]	"	I		+	
MSG_OR	Trigger operator prompt	Bool	0	I	U	+	
OCCUPIED	Unit allocated	Bool	0	I		+	
OS_ICTRL	PCS 7 OS interface for the RPE command	Int	0	I/O	U	+	
OS_ICTRL_STEP_NO	PCS 7 OS interface for the step command	DInt	0	I/O	QU	+	
PEND_OOS	Pending out of service: This unit can no longer be allocated after the current allocation is completed	Bool	0	I		+	
QAvgRecProcTime	Average time needed for batch control in the AS (per execution cycle)	Time		I/O	Q	+	
QContainerID	Unit procedure ID (RUP-ID)	Word	-	O	U	+	
QContainerID_SW	Status word of the unit procedure.	DWord	-	O	U	+	
Q_ERR_AS	Totals display: "Recipe error"	Int		O	Q	-	
Q_ERR_COMM	Communication error between AS and BCS	Bool		O	Q	-	
Q_ERR_EVENT_DB	The event buffer is allocated	Bool		O	Q	-	

I/O (parameter)	Meaning	Type	Default	Class	Attributes	OCM	Permitted values
QAvgRecProcTime	Average time needed for batch control in the AS (per execution cycle)	Time		I/O	Q	+	
Q_ERRContRunTime	At least one RUP / ROP / SUB has violated the monitoring time limit. Applies only to AS-based mode.	BOOL	-	O	Q	+	
Q_OCCUPI	Allocated	Bool	0	O	Q	-	
Q_STATUS	Block status for SIMATIC BATCH	DWord	0	O	U	+	
Q_UnitStatus	Unit status with user enumeration. The value depends on EN_LinkedUnit-Status	DWord	-	O		+	
QMAT_SP	Copy of MAT_SP	String[16]	-	O	Q	+	
QUSER_STATUS	Acquisition of USER_STATUS	Word	0	O	U	+	
RUP_STAT	(not used)	Byte	0	I	Q	+	
SP_COUNT	Lifebeat interval setpoint	Int	0	I		+	
STEPMODE	(not used)	Byte	0	I	Q	+	
STEP_NO	RUP step number	DWord	0	I	U	+	
UBA_ID	Batch ID (input)	DWord	0	I		+	
UBA_NAME	Batch name (input)	String[32]	"	I		+	
Unit_Attribute	Unit_Attribute	DWord	0	I	Q	+	
UNIT_ID	Configured UNIT_ID	Word	0	I			
UNIT_NAME	Unit name (Page 54)	String[24]	"	I		+	
UnitStatus	Unit status set by SIMATIC BATCH or PCS 7 OS	DWord	0	I		+	
URP_NAME	Recipe name, version	String[32]	"	I		+	
URP_REL	Recipe version	String[32]	"	I		+	

I/O (parameter)	Meaning	Type	Default	Class	Attributes	OCM	Permitted values
USER_STATUS	Free for use by PCS 7 programs	Word	0	I	U	+	
USTAT_PRIO	(not used)	Byte	0	I	U	-	
VA_COUNT	Actual Value: Lifebeat interval (counts down to 0)	Int	0	O	Q	+	
VBA_ID	Batch ID (output)	DInt	0	O	Q	+	
VBA_NAME	Batch name (output)	String[32]	"	O		+	
TriggerCmdPropagation	Run command from "CmdPropagation" parameter	Bool	0	I	Q	-	
RunAfterBufferFull	Run batch even if the event buffer is allocated.	Bool	1	I	Q	-	
UNITASID	The output contains the UNIT_ID in the higher value word and the AS_ID in the lower value word.	DWord		O	-	-	

Error codes on the "Q_ERR_ASB" parameter

The last error to occur during a batch process is shown on the parameter. Sometimes, the error codes can only be interpreted with specific knowledge of the batch process in the AS:

Error code	Meaning
0	No errors
-1	Undefined error
-2	An input parameter is invalid
-3	A data block needed is not available
-4	Internal system recognition
-10	A recipe list needed for the batch process is empty
-11	An ID needed for the batch process is not available
-12	After copying, the target list contains more entries than the associated source list (e.g. list of setpoints)
-13	After copying, the source string is bigger than the target string
-15	The phase (EPH) is already allocated
-16	The EPH command has not been run
-17	A buffer needed for the batch process is too small
-18	The EQM data block contains invalid entries

Error code	Meaning
-30	The event buffer is full
-37	The installed version of the blocks does not match the installed version of SIMATIC BATCH

System parameters

The following system parameters must not be changed by the user.

I/O (parameter)	Meaning	Type	Default	Class	Attributes	OCM	Permitted values
MSG_EVID_1	Unique message number	DWord	16#000000D2	I	U	-	
MSG_EVID_2	Unique message number	DWord	16#0000005A	I	U	-	
UNIT_ID	Unique unit ID within an AS	Word	16#0011	I	U	-	
AS_ID	Unique AS ID within the project	Int	1	I	U	-	
SampleTime	Sampling time	Real	1.0	I	U	-	
CFC_POOL_DB	Internal system parameter	Word	16#0584	I	U	-	
EVENT_DB	Result buffer for storage of status of recipe procedure elements	Word	16#0497	I	U	-	
EPE_CONN	Interconnection with IEPAR blocks	DInt	0	O	QU	-	
EQM_DB	Number of the equipment data block	Word	16#0000	I	U	-	
FirstSP_DB	Number of the first data block containing the setpoint information	Word	16#0000	I	U	-	
FirstRecipe_DB	Number of the first data block containing the recipe information	Word	16#0000	I	U	-	
ContainerID	Unique unit procedure ID within a recipe	Word	16#0000	I	U	+	
Max_CycleRun	Time limitation for batch control in the AS (per processing cycle)	Time	25 ms	I	U	-	

2.2 BATCH interface blocks

I/O (parameter)	Meaning	Type	Default	Class	Attributes	OCM	Permitted values
PLCRecipeExecution	Unit release for batch control in the AS	Bool	1	I	U	-	
MissingDBCount	Consistency check: Instance data block	Int	0	O	QU	-	
RUPReset	The AS-based recipe processing is aborted and the recipe data deleted	Bool	0	O	U	+	
RetVal	Detailed view: "Recipe error"	Int	0	O	QU	-	
FastSFC	Enable of fast SFC mode	Bool	activated	I	QU	-	
CopyEQMDB	For internal use or service technician only. Copy EQM data block to instance data block	Bool	deactivated	I	QU	-	
DebugTrace	For internal use or service technician only.	Word	16#0	I	QU	-	
TraceSTEP_NO	For internal use or service technician only.	DInt	L#0	I	QU	-	

2.2.4.14 UNIT_PLC message texts and associated values

Assigning a message text and message class to the block parameters.

Message no. MSG_EVID_1	Block parameters	Predefined message text	Message class	Can be suppressed by
1	OCCUPIED	Unit allocated	AS status (Status Message)	-
2	(Not BA_EN OR NOT L_BA_EN)	Unit not enabled for BATCH	AS status (Status Message)	-
3	PEND_OOS	Unit registered for maintenance	AS status (Status Message)	-
4	Q_STATUS AND 0x80000000 = 0x80000000	Unit error	Process control system message F (Fault)	-
5	MSG_OR	BATCH operator input required	OR (Operator Request)	-
6	NOT OCCUPIED	Unit enabled	AS	-

Message no. MSG_EVID_1	Block parameters	Predefined message text	Message class	Can be suppressed by
7	BA_EN AND L_BA_EN	Unit enabled for BATCH	AS	-
8			No message	-

Message no. MSG_EVID_2	Block parameters	Predefined message text	Message class	Can be sup- pressed by
1	Q_ERR_EQM_DB	Missing equipment DB	Process control sys- tem message S (Sys- tem)	-
2	(TriggerCmdPropagation OR (Q_ERR_EVENT_DB AND NOT RunAfterBufferFull))	Command entry with: @4%s@	AS status (Status Message)	-
3		The time stamps of EQM DB and SP DB are not the same. Unit not enabled for BATCH.	Process control sys- tem message S (Sys- tem)	-
4	QAvgRecProcTime >= Max_CycleRun	Recipe processing time @6%d@ ms ex- ceeds the max. limit @5%d@ ms.	Process message PM (Process Message)	-
5	L_ICTRL_OUT > 0	L_ICTRL command @4%s@ activated for STEP_NO: @7%d@	Process control sys- tem message S (Sys- tem)	-
6	L_ICTRL_OUT < 0	L_ICTRL command @4%s@ not permitted for STEP_NO: @7%d@	Process control sys- tem message S (Sys- tem)	-
7		-	No message	-
8		-	No message	-

Assigning the associated values to the block parameters MSG_EVID_1

Of the associated values of the message block, the first three have SIMATIC BATCH data assigned to them.

Associated value	Block parameter
1	BA_NA
2	STEP_NO
3	BA_ID
4	AUX_PR04
5	AUX_PR05
6	AUX_PR06
7	AUX_PR07
8	AUX_PR08
9	AUX_PR09
10	AUX_PR10

Assigning the associated values to the block parameters MSG_EVID_2

Associated value	Block parameter
1	BA_NA
2	STEP_NO
3	BA_ID
4	CmdPropagString
5	Max_CycleRun
6	(ProcessCurrentTime - ProcessStartTime)
7	L_ICTRL_STEP_NO
8	AUX_PRO8
9	AUX_PRO9
10	AUX_PR10

2.2.4.15 Operating and Monitoring of UNIT_PLC

UNIT_PLC views

The UNIT_PLC block offers the following views:

- Unit
- Functions
- Batch
- Parameter
- Messages

The tables below show the assignment of the parameters of the UNIT_PLC block to the input/output fields of the associated faceplate for the corresponding views.

Unit

The "Unit" views displays data for the corresponding unit. You operate and monitor the corresponding unit here.

Input/output field	Block parameter	Comment
"Name" display	UNIT_NAME	
"Status" display	QUSTAT_L	
"Release for BATCH" check box	BA_EN Bit 25	
"Allocate" check box	OCCUPIED	Display for occupied status of unit.
"User status" display	QUSER_STATUS	Status which is freely available to you. Click the display field to enter a value. The data format is INTEGER and values from 0 to 65535 are permitted.

Input/output field	Block parameter	Comment
"Unit status" check box	EN_LinkedUnitStatus	Cleared check box ("EN_LinkedUnitStatus" = 0): Value of "UnitStatus" input is switched through to output "Q_UnitStatus". Selected check box ("EN_LinkedUnitStatus" = 1): Value of "L_UnitStatus" input is switched through to output "Q_UnitStatus".
"Unit status" display	UnitStatus	You can enter the value of the unit status here. This field is only visible if you have created an enumeration data type for this purpose.
"Logon for maintenance" check box	PEND_OOS	This check box can only be selected if the unit has been released for BATCH. If the allocation is canceled afterwards, the unit is no longer released for BATCH and is not available for maintenance work.

Batch

The "Batch" view displays the data for the currently running batch or the last run batch.

Input/output field	Block parameter	Comment
"Name" display	UNIT_NAME	
"Status" display	QUSTAT_L	
Batch name	UBA_NAME	
Batch ID	UBA_ID	
Product	MAT_NAME	
Product code	MAT_SP	
Recipe	URP_NAME	
Recipe version	URP_REL	
Formula	FORMULA	
Formula version	FORMULA_REL	
"BatchCC" button		To open the control recipe window in BatchCC. BATCH Runtime must be started.

Functions

The "Functions" view displays all equipment phases of this unit in a table. Each equipment phase is arranged in a table row.

Input/output field	Block parameter	Comment
"Equipment phase" buttons		Opens the faceplate of this equipment phase
"Reset" button		Use this button to reset finished/aborted/stopped functions. Caution: This aborts and resets running equipment phases and removes the batch allocation.
Display "Status" of equipment phase	QUSTAT_L BA_STATE	Depending on the configuration of the equipment phase, as SFC type or by means of IEPH or IEOP.
"BatchCC" button		To open the control recipe window in BatchCC. BATCH Runtime must be started.
"Scroll backwards" button		
"Reset all functions" button	BA_EN from all IEPH/IEOP	
"Scroll forward" button	"Scroll forward" button	

Parameter

The "Parameter" view displays all configured parameters on the unit block "UNIT_PLC". An example for interconnecting the block with parameter blocks (IEPAR_XXX) is included in the SIMATIC BATCH help under "Best practice".

Input/output field	Block parameter	Comment
Control strategy	Control strategy configured at "UNIT_PLC" block	Value of set control strategy. Changes are possible if the unit is not occupied and is not released for BATCH.
Parameter		Name of the IEPAR_XXX
Setpoint	Q_SP_VAL	
Process value	QACT_VAL	
Unit of measure		Derived from the configuration in the SIMATIC Manager.

Setting user authorizations

Operator input can generally be enabled in the tab "Properties/Others" of the object properties of the UNIT_PLC and IEPH/IEOP faceplates.

- Yes - the object can be operated in process mode.
- No- the object cannot be operated in process mode.

In order to operate the object, the operator must additionally have the authorization. The settings for the authorization level for each operator are made with the WinCC Explorer in the User Administrator:

- Operator control requires the operator authorization level 5 (Process controlling) or level 6 (Higher process controlling).
- The authorization levels can be enabled as follows:
 - generally for all units (column "Enable" in the User Administrator)
 - unit specific (column "<Unit name>" in the User Administrator)

Create Batch faceplate from Template

In the Graphics Designer (WinCC-Explorer) there is a Template available for selecting Batch faceplates in PCS 7 OS pictures.

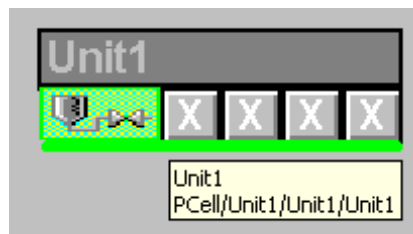
1. Open the desired process picture in the Graphics Designer.
2. Open the template file "@Template_Batch.pdl" via File > Open.
3. Apply the block icon "UNIT_PLC" in order to display the unit allocations and the operations and functions assigned to this unit.
4. Select the block symbol in the the template file and copy it to your process picture.
5. Select the block symbol in your process picture.
6. In the Dynamic Wizard window under "Standard dynamics" double click the PCS 7 function "Connect faceplate with process tag".
7. In the Dynamic Wizard click "Next" and then the search function "...".
8. In the tag selection dialog all instanced blocks are displayed. Select the desired block and click OK > Next> Finish.

Result

The process picture edited by you, is supplied with the block symbol "UNIT_PLC", to call the corresponding faceplate.

Unit status display by means of colored bar in the UNIT_PLC faceplate

In the PCS 7 OS, the status of a unit is indicated by means of a colored bar in the UNIT_PLC faceplate. The image below shows a unit whose status is "RUN".



You can tell whether a unit has this status if the bar is light green.

You can select or adapt the status colors in the dialog "Properties" of the object in the Graphics Designer.

Unit name

The input "UNIT_NAME" of the "UNIT_PLC" block is managed using SIMATIC BATCH. When creating the BATCH process cell data, the input is updated with the name of the corresponding units hierarchy folder. The name is thus displayed in the faceplate, like the name "Unit1" in the example image.

Other statuses

The table below lists other statuses that may apply.

Color	Status	State
Gray	Idle	idle
Dark green	Completed	complete
Mauve	Aborted	abort
Yellow	Held	hold
Light green	starting / run	starting / run
Gray	Occupied	occupied
Olive green	Not occupied	frame
Gray/green	Finished	ready
white	Released	released

2.3 Equipment parameter module

2.3.1 IEPAR_BOOL

2.3.1.1 IEPAR_BOOL: Equipment parameter module for data type BOOL

Object name (class + number)

FB 256 (Page 59)

Calling OBs

The OB in which you install the block (e.g. OB1).

Function

The block is used to set the setpoints and to acquire the process values of the BOOL data type (bit variable).

Operating principle

The bit signal applied to the inputs SP_VAL or ACT_VAL is transferred to the outputs Q_SP_VAL or QACT_VAL respectively. SP_VAL is usually set by the batch control or the IEPH faceplate; ACT_VAL is set by the user program.

2.3.1.2 I/Os of IEPAR_BOOL

I/Os

I/O (parameter)	Meaning	Type	Default	Type	Attributes	OCM	Permitted values
ACT_VAL	Actual value (input)	Bool	0	I			
EPE_CONN	IEPH/IEOP connection	DInt	0	I			
Q_SP_VAL	Setpoint (output)	Bool	0	O		+	
QACT_VAL	Actual value (output)	Bool	0	O		+	
SP_VAL	Setpoint (input)	Bool	0	I		+	

See also

General information on the block description (Page 9)

2.3.2 IEPAR_DEST

2.3.2.1 IEPAR_DEST: Equipment parameter module for data type DEST

Object name (class + number)

FB263 (Page 60)

Calling OBs

The OB in which you install the block (e.g. OB1).

Function

The block is used to set the setpoints and to acquire the process values of the data type double integer for the target destination (location ID).

Operating principle

The active signal at input SP_VAL or ACT_VAL is transferred to output Q_SP_VAL or QACT_VAL. SP_VAL is usually set by the batch control or the IEPH faceplate; ACT_VAL is set by the user program.

2.3.2.2 I/Os of IEPAR_DEST

I/Os

I/O (parameter)	Meaning	Type	Default	Type	Attributes	OCM	Permitted values
ACT_VAL	Actual value (input)	DInt	0	I			
EPE_CONN	IEPH/IEOP connection	DInt	0	I			
Q_SP_VAL	Setpoint (output)	DInt	0	O		+	
QACT_VAL	Actual value (output)	DInt	0	O		+	
SP_VAL	Setpoint (input)	DInt	0	I		+	

See also

General information on the block description (Page 9)

2.3.3 IEPAR_DINT

2.3.3.1 IEPAR_DINT: Equipment parameter module for data type Double Integer

Object name (class + number)

FB 255 (Page 61)

Calling OBs

The OB in which you install the block (e.g. OB1).

Function

The block is used to set the setpoints and to acquire the process values of the data type double integer.

Operating principle

The active signal at input SP_VAL or ACT_VAL is transferred to output Q_SP_VAL or QACT_VAL. SP_VAL is usually set by the batch control or the IEPH faceplate; ACT_VAL is set by the user program.

2.3.3.2 I/Os of IEPAR_DINT

I/Os

I/O (parameter)	Meaning	Type	Default	Type	Attributes	OCM
ACT_VAL	Actual value (input)	DInt	0	I		
EPE_CONN	IEPH/IEOP connection	DInt	0	I		
LOWLIMIT	Low limit	DInt	0	I		+
Q_SP_VAL	Setpoint (output)	DInt	0	O		+
QACT_VAL	Actual value (output)	DInt	0	O		+
SP_VAL	Setpoint (input)	DInt	0	I		+
UPLIMIT	Upper limit	DInt	100	I		+

See also

General information on the block description (Page 9)

2.3.4 IEPAR_ENUM

2.3.4.1 IEPAR_ENUM: Equipment parameter module for parameter type ENUM

Object name (class + number)

FB 259 (Page 63)

Calling OBs

The OB in which you install the block (for example, OB 32).

Function

The block is used to set the setpoints and acquire the actual values of enumeration types. In the properties dialog of the "SP_VAL" input in the CFC Editor you can select a configured enumeration type in SIMATIC Manager. You can also select the standard values.

Note

Migration note

The block input connector "ENUM" may no longer be configured; for this reason, the connector properties have been made invisible.

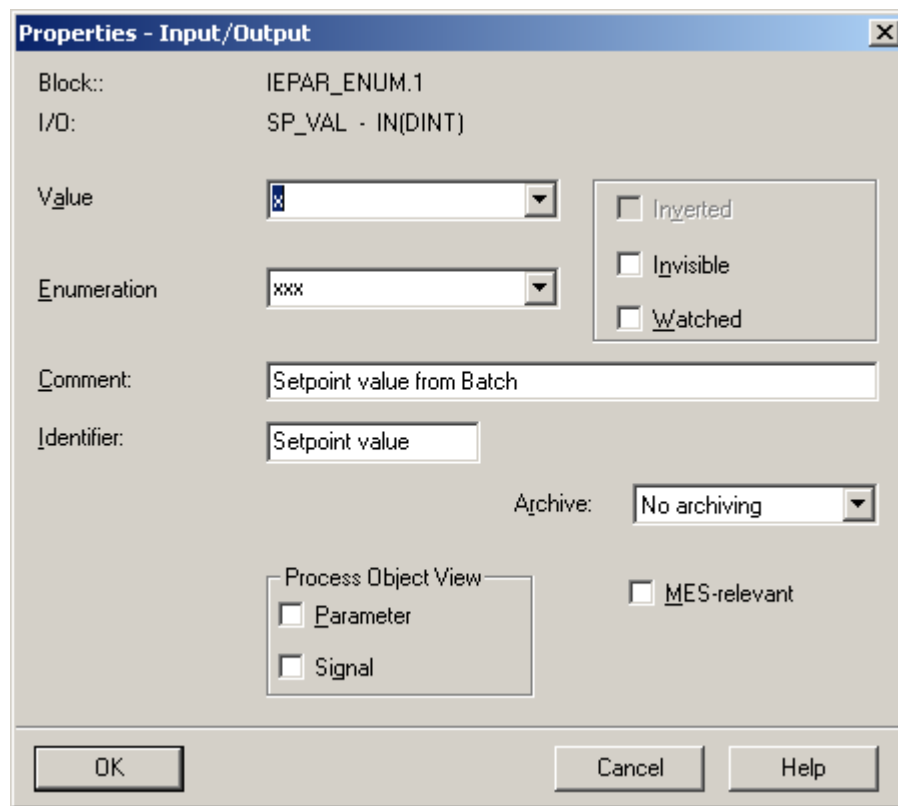


Figure 2-1 Assign enumeration type

Operating principle

The setpoints activated at the inputs are transferred to the outputs.

2.3.4.2 I/Os of IEPAR_ENUM

I/Os

I/O (parameter)	Meaning	Type	Default	Type	Attributes	OCM	Permitted values
ACT_VAL	Actual value (input)	DInt	0	I			
ENUM	Text string (input)	String[16]	"	I		+	
EPE_CONN	IEPH/IEOP connection	DInt	0	I			
Q_SP_VAL	Setpoint (output)	DInt	0	O		+	
QACT_VAL	Actual value (output)	DInt	0	O		+	
SP_VAL	Setpoint (input)	DInt	0	I		+	

See also

General information on the block description (Page 9)

2.3.5 IEPAR_PI**2.3.5.1 IEPAR_PI: Equipment parameter module for process inputs****Object name (class + number)**

FB 260 (Page 64)

Calling OBs

The OB in which you install the block (e.g. OB1).

Function

The block is used to set the setpoints and acquire process values of input materials (quantity, material ID, batch ID of origin)

Operating principle

The setpoints activated at the inputs are transferred to the outputs.

2.3.5.2 I/Os of IEPAR_PI**I/Os**

I/O (parameter)	Meaning	Type	Default	Type	Attributes	OCM	Permitted values
ACT_VAL	Actual value (input)	Real	0	I			
BA_ID_AC	Batch ID actual value (input)	DInt	0	I			
BA_ID_SP	Batch ID setpoint (input)	DInt	0	I		+	
EPE_CONN	IEPH/IEOP connection	DInt	0	I			
LOWLIMIT	Low limit	Real	0	I			LOWLIMIT < UPLIMIT
MAT_ACT	Material actual value (input)	String[16]	"	I			
MAT_SP	Material setpoint (input)	String[16]	"	I		+	

I/O (parameter)	Meaning	Type	Default	Type	Attributes	OCM	Permitted values
Q_SP_VAL	Setpoint (output)	Real	0	O			
QACT_VAL	Actual value (output)	Real	0	O		+	
QBAID_AC	Batch ID actual value (output)	DInt	0	O	U	+	
QBAID_SP	Batch ID setpoint (output)	DInt	0	O		+	
QMAT_ACT	Material actual value (output)	String[16]	"	O		+	
QMAT_SP	Material setpoint (output)	String[16]	"	O		+	
QTOL_ACT	Actual value tolerance (output)	Real	0	O	U		
QTOL_SP	Tolerance setpoint (output)	Real	0	O	U		
SP_VAL	Setpoint (input)	Real	0	I		+	
UPLIMIT	Upper limit	Real	100.0	I			

See also

General information on the block description (Page 9)

2.3.6 IEPAR_PO**2.3.6.1 IEPAR_PO: Equipment parameter module for process outputs****Object name (class + number)**

FB 261 (Page 66)

Calling OBs

The OB in which you install the block (e.g. OB1).

Function

The block is used to set the setpoints and collect the process values of main, secondary, intermediate and waste products (quantity, material, batch ID of origin).

Operating principle

The setpoints activated at the inputs are transferred to the outputs.

2.3.6.2 I/Os of IEPAR_PO

I/Os

I/O (parameter)	Meaning	Type	Default	Type	Attributes	OCM	Permitted values
ACT_VAL	Actual value (input)	Real	0	I			
BA_ID_AC	Batch ID actual value (input)	DInt	0	I			
BA_ID_SP	Batch ID setpoint (input)	DInt	0	I		+	
EPE_CONN	IEPH/IEOP connection	DInt	0	I			
LOWLIMIT	Low limit	Real	0	I		+	LOWLIMIT < UPLIMIT
MAT_ACT	Material actual value (input)	String[16]	"	I			
MAT_SP	Material setpoint (input)	String[16]	"	I		+	
Q_SP_VAL	Setpoint (output)	Real	0	O			
QACT_VAL	Actual value (output)	Real	0	O		+	
QBAID_AC	Batch ID actual value (output)	DInt	0	O	U	+	
QBAID_SP	Batch ID setpoint (output)	DInt	0	O		+	
QMAT_ACT	Material actual value (output)	String[16]	"	O		+	
QMAT_SP	Material setpoint (output)	String[16]	"	O		+	
QTOL_ACT	Actual value tolerance (output)	Real	0	O	U		
QTOL_SP	Tolerance setpoint (output)	Real	0	O	U		
SP_VAL	Setpoint (input)	Real	0	I		+	
TOL_ACT	Actual value tolerance (input)	Real	0	I	U		
TOL_SP	Tolerance setpoint (input)	Real	0	I	U		
UPLIMIT	Upper limit	Real	100.0	I		+	

See also

General information on the block description (Page 9)

2.3.7 IEPAR_REAL

2.3.7.1 IEPAR_REAL: Equipment parameter module for data type REAL

Object name (class + number)

FB 257 (Page 67)

Calling OBs

The OB in which you install the block (e.g. OB1).

Function

The block is used to set the setpoints and to acquire the process values of the data type real.

Operating principle

The signal applied to the inputs SP_VAL or ACT_VAL is transferred to the outputs QSP_VAL or QACT_VAL respectively.

2.3.7.2 I/Os of IEPAR_REAL

I/Os

I/O (parameter)	Meaning	Type	Default	Type	Attributes	OCM	Permitted values
ACT_VAL	Actual value (input)	Real	0	I			
EPE_CONN	IEPH/IEOP connection	DInt	0	I			
LOWLIMIT	Low limit	Real	0	I		+	LOWLIMIT < UPLIMIT
Q_SP_VAL	Setpoint (output)	Real	0	O		+	
QACT_VAL	Actual value (output)	Real	0	O		+	
SP_VAL	Setpoint (input)	Real	0	I		+	
UPLIMIT	Upper limit	Real	100.0	I		+	

See also

General information on the block description (Page 9)

2.3.8 IEPAR_SOURCE

2.3.8.1 IEPAR_SOURCE: Equipment parameter module for data type SOURCE

Object name (class + number)

FB 262 (Page 68)

Calling OBs

The OB in which you install the block (e.g. OB1).

Function

The block is used to set the setpoints and to acquire the process values of the data type double integer for the source destination (location ID).

Operating principle

The active signal at input SP_VAL or ACT_VAL is transferred to output Q_SP_VAL or QACT_VAL. SP_VAL is usually set by the batch control or the IEPH faceplate; ACT_VAL is set by the user program.

2.3.8.2 I/Os of IEPAR_SOURCE

I/Os

I/O (parameter)	Meaning	Type	Default	Type	Attributes	OCM	Permitted values
ACT_VAL	Actual value (input)	DInt	0	I			
EPE_CONN	IEPH/IEOP connection	DInt	0	I			
Q_SP_VAL	Setpoint (output)	DInt	0	O		+	
QACT_VAL	Actual value (output)	DInt	0	O		+	
SP_VAL	Setpoint (input)	DInt	0	I		+	

See also

General information on the block description (Page 9)

2.3.9 IEPAR_STR

2.3.9.1 IEPAR_STR: Equipment parameter module for parameter type STRING

Object name (class + number)

FB 258 (Page 69)

Calling OBs

The OB in which you install the block (e.g. OB1).

Function

The block is used to set setpoints and collect process values of the string parameter type.

Operating principle

The bit signal applied to the inputs SP_VAL or ACT_VAL is transferred to the outputs Q_SP_VAL or QACT_VAL respectively. At the same time, SP_VAL can be converted to a number. To allow this, the string must be a numeric string and be representable with the DINT data type (+-2147483647). The number is applied to the Q_SP output. To activate the conversion, LOC_ID_ON = 1 must be set. With SP_ON_ERR, a number can be configured that is written to the Q_SP output when there is a conversion error.

A 1 signal is always applied to the QRESULT output when conversion is not active. If LOC_ID_ON = 1 (conversion to number), QRESULT = 0 as soon as there is a conversion error.

2.3.9.2 I/Os of IEPAR_STR

I/Os

I/O (parameter)	Meaning	Type	Default	Type	Attributes	OCM	Permitted values
ACT_VAL	Actual value (input)	String[16]	"	I			
EPE_CONN	IEPH/IEOP connection	DInt	0	I			
LOC_ID_ON	Enable string conversion	Bool	0	I	U		
Q_SP	Setpoint converted by string => DInt	DInt	0	O			
Q_SP_VAL	Setpoint (output)	String[16]	"	O		+	

2.3 Equipment parameter module

I/O (parameter)	Meaning	Type	Default	Type	Attributes	OCM	Permitted values
QACT_VAL	Actual value (output)	String[16]	"	O		+	
QRESULT	Result bit	Bool	0	O	U	+	
SP_ON_ERR	Value returned if the STRING => DInt conversion has failed	DInt	0	I			
SP_VAL	Setpoint (input)	String[16]	"	I		+	

See also

General information on the block description (Page 9)

2.3.10 IEPAR_VIA**2.3.10.1 IEPAR_VIA: Equipment parameter module for data type VIA****Object name (class + number)**

FB 264 (Page 71)

Calling OBs

The OB in which you install the block (e.g. OB1).

Function

The block is used to set the setpoints and to acquire the process values of the data type double integer for the via destination (location ID).

Operating principle

The active signal at input SP_VAL or ACT_VAL is transferred to output Q_SP_VAL or QACT_VAL. SP_VAL is usually set by the batch control or the IEPH faceplate; ACT_VAL is set by the user program.

2.3.10.2 I/Os of IEPAR_VIA

I/Os

I/O (parameter)	Meaning	Type	Default	Type	Attributes	OCM	Permitted values
ACT_VAL	Actual value (input)	DInt	0	I			
EPE_CONN	IEPH/IEOP connection	DInt	0	I			
Q_SP_VAL	Setpoint (output)	DInt	0	O		+	
QACT_VAL	Actual value (output)	DInt	0	O		+	
SP_VAL	Setpoint (input)	DInt	0	I		+	

See also

General information on the block description (Page 9)

System blocks

3.1 Function blocks

3.1.1 RecipeControl

RecipeControl: Runtime-system block

Object name (class + number)

FB 267

Function

This internal system block is called by UNIT_PLC and controls units and recipe operations.

3.1.2 MemIDB64

MemID64: Runtime-system block

Object name (class + number)

FB 273

Function

This internal system block provides a 64 KB buffer for internal data management.

3.1.3 EventSend

3.1.3.1 EventSend

EventSend: Interface block of communication

Object name (class + number)

FB 274

Calling OBs

The block will be created by the SIMATIC BATCH configuration dialog in OB 35.

Function

The internal system block manages RPE events and is responsible for communication between AS and BCS.

Batch messages

Event	Message class
Communication error to BCS	AS process control system message - Problem
Event buffer full	AS process control system message - Problem
Communication DB does not exist	AS process control system message - Problem
Invalid BCS command frame	Status message - AS

Additional information

Installation (Page 74)

Collect RPE events and communication with BCS (Page 75)

Diagnostic message (Page 76)

3.1.3.2 EventSend installation

Installation

The block will be created by the SIMATIC BATCH configuration dialog in the "@ASB_Event" chart. The following block entries will be set during installation:

Input	Comment
ID_M_1	NetPro connection ID to BATCH master server
ID_S_1	NetPro connection ID to BATCH standby server
MRG_SND_DB	Instance DB number of MRG_SND instance
MRG_RCV_DB	Instance DB number of MRG_RCV instance
EVENT_DB_30	Instance DB number of EVENT_DB_30 instance
EVENT_DB_31	Instance DB number of EVENT_DB_31 instance
EVENT_DB_32	Instance DB number of EVENT_DB_32 instance
EVENT_DB_33	Instance DB number of EVENT_DB_33 instance
EVENT_DB_34	Instance DB number of EVENT_DB_34 instance
EVENT_DB_35	Instance DB number of EVENT_DB_35 instance
EVENT_DB_36	Instance DB number of EVENT_DB_36 instance
EVENT_DB_37	Instance DB number of EVENT_DB_37 instance

Note

You can only use one instance of the block in the AS.

3.1.3.3 Collect EventSend RPE events and communication with BCS

Event management and communication with SIMATIC BATCH server

All RPE events saved in the "UNIT_PLC" block in the send buffer of the OB (implemented as instances of the "MenID64" block and placed in the "@ASB_Event" chart) are collected in a send buffer and sent as frame to the SIMATIC BATCH server. The server returns a "logical acknowledgment" once the frame has been processed. While the block is waiting for the acknowledgment, the new events are saved in a second send buffer while the data in the first send buffer remain unchanged. There are three send buffers. If all three send buffers contain data that was not sent, then the events will remain in the event buffer of the OB. You can use the buffer length limit 16KB to trigger the "unit command" with the "UNIT_PLC" block. Additional information is available in the section UNIT_PLC, Settings for communication monitoring and communication buffer. Once the send buffer length 46.5 KB of the OB has been reached, then no additional new event are saved. The block will receive frames from the server for each OB35 block call (100 ms). Frames to the server will be sent via the time available at the input "MaxFillT" (standard value = 1s) or if the data in the send buffer exceed the value that you have set at the "MinBuffSND" input (standard value is 10%). BSEND and BRCV blocks are used as transport interface.

The status of each OB event buffer can be monitored via the following output set:

Output	Comment
EVENTDB30FILL 1)	Used memory in %
EVENTDB30STAT 1)	Status: <ul style="list-style-type: none"> • Bit 8 = buffer size $\geq$ 31 kB (trigger "command propagation" at the UNIT_PLC block) • Bit 9 = BSEND/BRCV error • Bit 10 = buffer size $\geq$ 46.5kB (no additional writing) • Bit 11 = buffer size $\geq$ 16kB

1) Example for OB30

The status of each send buffer can be monitored via the following output set:

Output	Comment
SNDBuf1State 2)	Status: <ul style="list-style-type: none"> • 0=EMPTY • 1=WRITING • 2=RDYTOSEND • 3=SENDING • 4=WAITFORACK
SNDBuf1_FILL 2)	Used buffer memory in %

2) Example for send buffer 1

You can monitor the communication status of the BSEND / BRCV block via the following outputs:

Output	Comment
QERRSTAT_BRCV	The low-value byte indicates the BRCV status to the standby server. The higher-value byte indicates the BRCV status to the master server.
QERRSTAT_BSEND	The low-value byte indicates the BSEND status to the standby server. The higher-value byte indicates the BSEND status to the master server.
QERR_BSEND	TRUE, if all BSEND instances have set their "ERROR" outputs.
QERR_BUFFER	One of the event buffers is full (Bit0 = OB30, Bit1 = OB31, etc.).

3.1.3.4 EventSend diagnostic messages

Principle behavior

Diagnostic messages are intended for service engineers.

A copy of the first 40 bytes of each received frame will be returned to the SIMATIC BATCH server for diagnostic purposes, if the block input "DebugTel" has been set to "true".

Note

Use of this function will slow down the system tremendously. In process operation, you will have to set the input "DebugTel" to "false".

3.2 Function calls

3.2.1 CallFB

CallFB: Runtime-system block

Object name (class + number)

FC 190

Function

This internal system function calls a function block outside of the standard processing cycle.

3.2.2 CheckCmdPriority

CheckCmdPriority: Runtime-system block

Object name (class + number)

FC 224

Function

This internal system function checks whether a pending command can be executed or not.

3.2.3 CheckEQMDB

CheckEQMDB: Runtime-system block

Object name (class + number)

FC 192

Function

This internal system function checks whether the EQM DB is present and, if so, whether it is valid.

3.2.4 CmdControl

CmdControl: Runtime-system block

Object name (class + number)

FC 221

Function

The internal system function is called up by ContControl and controls the command to all recipe procedure elements listed in the command list. For a START command, the status of the recipe procedure elements is checked in the existing command list to ensure that no other recipe procedure element is active. The function starts a runtime counter. If the process value is greater than the setpoint, the status of the recipe procedure element changes to "Completed" and the next recipe procedure element is started.

3.2.5 CommRPEEvent

CommRPEEvent: Runtime-system block

Object name (class + number)

FC 211

Function

This internal function creates a data record with current status information and adds it to the system list.

3.2.6 CommSPDataToDB

CommSPDataToDB: Runtime-system block

Object name (class + number)

FC 212

Function

This internal function determines a setpoint and adds it to a status data record.

3.2.7 ContControl

ContControl: Runtime-system block

Object name (class + number)

FC 219

Function

This internal system block is called by RecipeControl and controls units and recipe operations.

3.2.8 ContGetRPEListSW

ContGetRPEListSW: Runtime-system block

Object name (class + number)

FC 189

Function

This internal system function returns the status word of a recipe element (a recipe operation or recipe phase, for example).

3.2.9 ContPropagateRef

ContPropagateRef: Runtime-system block

Object name (class + number)

FC 213

Function

This internal function passes a reference value of a recipe operation on to an upstream recipe operation.

3.2.10 ContSetSW

ContSetSW: Runtime-system block

Object name (class + number)

FC 216

Function

The ContSetSW internal system block generates the current status of a recipe element (unit, recipe operation).

3.2.11 CopyBlock

CopyBlock: Runtime-system block

Object name (class + number)

FC 191

Function

This internal system function copies data blocks from one memory location (e.g. data block, instance data block) to another.

3.2.12 DebugTrace

DebugTrace: Runtime-system block

Object name (class + number)

FC 214

Function

The function will be activated via the inputs of the UNIT_PLC block. Each time the function is called, a diagnostic message (sequence tracking) is saved in a send buffer.

3.2.13 EPARGet

EPARGet: Runtime-system block

Object name (class + number)

FC 183

Function

This internal system function uses the EQM DB to determine the data type and data length of a setpoint for an equipment phase.

3.2.14 EPARGetValue

EPARGetValue: Runtime-system block

Object name (class + number)

FC 186

Function

This internal system function determines the current value of a process parameter.

3.2.15 EPHGetHeader

EPHGetHeader: Runtime-system block

Object name (class + number)

FC 184

Function

This internal system function uses the EQM DB to determine the setpoint list and the management data required by it for an equipment phase.

3.2.16 EPHGetSW

EPHGetSW: Runtime-system block

Object name (class + number)

FC 187

Function

This internal system function returns the status word of an IEPH block (QUSTAT_L) or an SFC type (BA_STAT and BA_ADDSTATE).

3.2.17 EPHManualControl

EPHMan: Runtime-system block

Object name (class + number)

FC 223

Function

This internal system function monitors all technical functions within a unit. If an equipment phase is not in "IDLE", "COMPLETED", "ABORTED", "STOPPED" state and is controlled in manual mode, the status change is detected and the BCS notified.

3.2.18 EPHRun

EPHRun: Runtime-system block

Object name (class + number)

FC 188

Function

This internal system function writes S88 commands to an equipment phase and calls this equipment phase outside of the processing cycle.

3.2.19 LookUp_ID

LookUp_ID: Runtime-system block

Object name (class + number)

FC 182

Function

This internal function determines elements within a list based on a defined parameter.

3.2.20 NOPControl

NOPControl: Runtime-system block

Object name (class + number)

FC 215

Function

This internal system block is called by ContControl and controls null instructions (NOPs). The block starts the runtime. If the time is longer than the specified runtime, the status changes to "Ready to Complete" and then to "Completed".

3.2.21 ReliableCmd

Reliable Command: Runtime-system block

Object name (class + number)

FC 225

Function

This internal system function is called by each RPE object. The function notifies the SIMATIC Batch Control Server about the execution of a command by a command step.

3.2.22 RPEEnumEleListCW

RPEEnumEleListCW: Runtime-system block

Object name (class + number)

FC 193

Function

This internal system function sets an S88 command for a recipe phase element (a recipe operation or recipe phase, for example) in a recipe element list.

3.2.23 RPEGetSP

RPEGetSP: Runtime-system block

Object name (class + number)

FC 195

Function

This internal function returns the setpoints determined (see RPEGetSPList).

3.2.24 RPEGetSPList

RPEGetSPList: Runtime-system block

Object name (class + number)

FC 194

Function

This internal function uses the setpoint DB to determine setpoints.

3.2.25 RPEGetSW

RPEGetSW: Runtime-system block

Object name (class + number)

FC 207

Function

This internal function returns the current status word of a recipe element (a recipe operation or recipe phase, for example).

3.2.26 RPEPrepareOccupy

RPEPrepareOccupy: Runtime-system block

Object name (class + number)

FC 206

Function

This function is called by RecipeControl and occupies a recipe element. If the recipe element is a unit or a recipe operation, all associated recipe elements are also occupied.

3.2.27 RPEPrepareToCmpl

RPEPrepareToCmpl: Runtime-system block

Object name (class + number)

FC 209

Function

If the recipe element (recipe operation or recipe phase, for example) is in "ready to complete" state, this internal function is called.

Taking the states of the recipe elements that are related to this particular recipe element into account, the previous and current recipe elements are set to "Completed" and the next recipe element is started.

3.2.28 RPESetCW

RPESetCW: Runtime-system block

Object name (class + number)

FC 208

Function

This internal function sets an S88 command at a recipe procedure element.

3.2.29 RPESetLinkCW

RPESetLinkCW: Runtime-system block

Object name (class + number)

FC 210

Function

This internal function sets an SA 88 command which was transferred to the inputs "L_ICTRL" or "OS_ICTRL" at a recipe procedure element. The recipe procedure element is defined by the inputs "L_ICTRL_STEP_NO" or "OS_ICTRL_STEP_NO".

3.2.30 RPHControl

RPHControl: Runtime-system block

Object name (class + number)

FC 218

Function

This internal system block is called by ContControl and controls the recipe phases. The block reads the recipe phase setpoints from the setpoint DB and controls the equipment phase. If the equipment phase is allocated, the S88 commands can be executed. Otherwise, the status of the recipe phase corresponds to the status of the equipment phase.

3.2.31 RPHGetAllRef

RPHGetAllRef: Runtime-system block

Object name (class + number)

FC 204

Function

This internal function determines within the list of recipe phase setpoints, whether a setpoint is a reference value for another unit or recipe operation. If a reference value is present, this reference is updated with the current setpoint value.

3.2.32 RPHGetAllSP

RPHGetAllSP: Runtime-system block

Object name (class + number)

FC 199

Function

This internal function copies the recipe phase setpoints from the setpoint DB to the equipment phase.

3.2.33 RPHGetRef

RPHGetRef: Runtime-system block

Object name (class + number)

FC 197

Function

This internal function returns reference values and their lengths for units and recipe operations.

3.2.34 RPHSetRef

RPHSetRef: Runtime-system block

Object name (class + number)

FC 198

Function

This function copies the current setpoint-actual value from the process to the reference list associated with the units and recipe operations.

3.2.35 TagCollGetHeader

TagCollGetHeader: Runtime-system block

Object name (class + number)

FC 185

Function

This internal system function uses the EQM DB to determine the process values and the management data required by them for the TagColl blocks.

3.2.36 TransBCSControl

TransBCSControl: Runtime-system block

Object name (class + number)

FC 220

Function

This internal system block is called by ContControl and controls transitions. The block starts and executes transitions. If a transition is "true" (i.e. the status is sent by the BCS), the status first changes to "ready to complete" and then to "completed".

3.2.37 TransControl

TransControl: Runtime-system block

Object name (class + number)

FC 217

Function

This internal system block is called by ContControl and controls transitions. The block starts and executes transitions. If a transition is "true", the status first changes to "ready to complete" and then to "completed".

3.2.38 TransGetAllPin

TransGetAllPin: Runtime-system block

Object name (class + number)

FC 200

Function

This internal function copies the comparison values for the transition from the setpoint DB to the recipe DB.

3.2.39 TransGetPin

TransGetPin: Runtime-system block

Object name (class + number)

FC 201

Function

This internal function returns the individual transition conditions (both operands and the associated condition) of a transition.

3.2.40 TransGetPinOper

TransGetPinOper: Runtime-system block

Object name (class + number)

FC 202

Function

This internal function cancels transition conditions and returns the current operand (constant or reference, for example).

3.2.41 TransGetPinVal

TransGetPinVal: Runtime-system block

Object name (class + number)

FC 203

Function

This internal function returns the current result value of a condition within a transition.

3.2.42 TransGetVal

TransGetVal: Runtime-system block

Object name (class + number)

FC 196

Function

This internal function cancels transition conditions.

3.2.43 TransSetGateVal

TransSetGateVal: Runtime-system block

Object name (class + number)

FC 205

Function

This internal function compares the binary input values of one logic with the result value of the previous logic.

3.2.44 UnitParamControl

UnitParamControl: Runtime-system block

Object name (class + number)

FC 222

Function

This internal system function is called by ContControl and controls the parameter step. The parameter step supplies the dynamic parameters of the unit with the values entered in the recipe. The function starts a runtime counter. If the process value is greater than the setpoint, the status of the RPE changes to "COMPLETED" and the next RPE is started.

Index

A

Associated values, 20

B

Batch messages, 35
Block parameters, 30

C

CallFB, 77
CheckCmdPriorityI, 77
CheckEQMDB, 77
CmdControl, 77
Comm_RPEEvent, 78
CommSPDataToDB, 78
Consistency check, 39
ContControl, 78
ContGetRPEListSW, 79
ContPropagateRef, 79
ContSetSW, 79
CopyBlok, 80

D

DebugTrace, 80

E

EPARGet, 80
EPARGetValue, 81
EPHGetHeader, 81
EPHGetSW, 81
EPHMan, 81
EPHRun, 82
EventSend, 73, 75
 Installation, 74
EventSend:diagnostic messages, 76

F

Function, 35

G

General information on the block description, 9
 Calling OBs, 10
 Error handling, 10
 Function, 9
 I/Os, 10
 Object name, 9
 OCM, 12
 Operating and monitoring, 12
 Operating principle, 9
 Permitted values, 12
 Type name, 9

I

IEOP

Allocation of ICTRL, 26
Batch messages, 28
Creating a Batch faceplate, 33
Equipment operation - BATCH interface, 24
Function, 24
I/Os, 29
Message texts and message class, 30
Operating and monitoring, 32
Operating principle, 25
Rules, 27
Setting user authorizations, 33

IEPAR_BOOL

Equipment parameter module for data type
BOOL, 59
Function, 59
I/Os, 59
Operating principle, 59

IEPAR_DEST

Equipment parameter module for data type
DEST, 60
Function, 60
I/Os, 60
Operating principle, 60

IEPAR_DINT

Equipment parameter module for data type Double
Integer, 61
Function, 61
I/Os, 61
Operating principle, 61

IEPAR_ENUM
Equipment parameter module for parameter type
ENUM, 62
Function, 62
I/Os, 63
Operating principle, 63

IEPAR_PI
Equipment parameter module for process
inputs, 64
Function, 64
I/Os, 64
Operating principle, 64

IEPAR_PO
Equipment parameter module for process
outputs, 65
Function, 65
I/Os, 66
Operating principle, 65

IEPAR_REAL
Equipment parameter module for data type
REAL, 67
Function, 67
I/Os, 67
Operating principle, 67

IEPAR_SOURCE
Equipment parameter module for data type
SOURCE, 68
Function, 68
I/Os, 68
Operating principle, 68

IEPAR_STR
Equipment parameter module for parameter type
STRING, 69
Function, 69
I/Os, 69
Operating principle, 69

IEPAR_VIA
Equipment parameter module for data type
VIA, 70
Function, 70
I/Os, 71
Operating principle, 70

IEPH
Allocation of ICTRL, 16
Allocation of the status words USTAT_L and
QUSTAT_L, 16
Batch messages, 18
Creating user authorizations, 23
Equipment phase - BATCH interface, 13
Function, 13
I/Os, 18
Message texts and message class, 20

Operating and monitoring, 21
Operating principle, 15
Setting status bits, 17
Using templates to create a Batch faceplates, 23

ISA-88
Commands, 40

L

LookUp_ID, 82

M

MemID64, 73
Message texts, 20, 30
MSG_EVID_1, 21, 31, 53
MSG_EVID_2, 21, 31, 54

N

NOPControl, 82

Q

Q_ERR_COM, 42

R

RecipeControl, 73
ReliableCmd, 83
RPE events, 75
RPEEnumEleListCW, 83
RPEGetSP, 83
RPEGetSPList, 84
RPEGetSW, 84
RPEPrepareOccupy, 84
RPEPrepareToCmpl, 84
RPESetCW, 85
RPESetLinkCW, 85
RPHControl, 85
RPHGetAllRef, 86
RPHGetAllSP, 86
RPHGetRef, 86
RPHSetRef, 87

S

Settings of the communication buffer
"RunAfterBufferFull", 42

START command, 39

T

TAG_COLL

- BATCH interface, 34

- Function, 34

- I/Os, 35

- Operating principle, 34

TagCollGetHeader, 87

TransBCSControl, 87

TransControl, 88

TransGetAllPin, 88

TransGetPin, 88

TransGetPinOper, 89

TransGetPinVal, 89

TransGetVal, 89

TransSetGateVal, 90

U

Unit allocation - BATCH interface, 35

UNIT_PLC

- Function and operating principle, 35

- Message texts and message class, 52

- Operating and monitoring, 54

UNIT_PLC function

- Deactivating a Unit, 38

- Diagnostic Messages, 45

UNIT_PLC operating principle

- Command interface for RPEs, 39

- Command interface to influence the entire unit, 40

- Communication buffer, 41

- Communication monitoring, 41

- EQM DB consistency check, 43

- Monitoring time of a recipe operation, 45

- Occupy and release, 36

- Recipe processing, 45

- Status Word, 37

- User Status Word, 38

UnitParamControl, 90

