Example Blocks for WinCC V7 and STEP 7 V5

(for S7-300 in SCL and S7-400 with CFC, SCL)

WinCC

Configuration Example • April 2010

Applications & Tools

Answers for industry.

SIEMENS

Warranty, Liability and Support

Note The Application Examples are not binding and do not claim to be complete regarding the configuration, equipping and any eventuality. The application examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for ensuring that the described products are used correctly. These application examples do not relieve you of the responsibility of safely and professionally using, installing, operating and servicing equipment. When using these application examples, you recognize that we cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these application examples at any time without prior notice. If there are any deviations between the recommendations provided in these application examples and other Siemens publications – e.g. Catalogs – the contents of the other documents have priority.

We do not accept any liability for the information contained in this document.

Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this Application Example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act (“Produkthaftungsgesetz”), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of a condition which goes to the root of the contract (“wesentliche Vertragspflichten”). However, claims arising from a breach of a condition which goes to the root of the contract shall be limited to the foreseeable damage which is intrinsic to the contract, unless caused by intent or gross negligence or based on mandatory liability for injury of life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.

It is not permissible to transfer or copy these application examples or excerpts of them without having prior authorization from Siemens Industry Sector in writing.

For questions about this document please use the following e-mail address:

online-support.automation@siemens.com

Preface

The example blocks for STEP 7 and WinCC enable the user to utilize various automation functions or to use these blocks as templates for the configuration of individual blocks.

Configuration of the application

This application includes:

- an example project that demonstrates the use of STEP 7 blocks and faceplates and
- the individual configuration examples of the blocks for usage in your own projects.

The table below shows all blocks included in this application:

Table 1-1

Block	No	Description
BST_DIGITAL	FB650	Visualization and operator control of digital signals
BST_ANALOG	FB640	Visualization and operator control of analog signals
BST_COUNT	FB654	Numerical rectangle integrator, e.g. simulation of a fill level (without faceplate)
BST_FF	FB653	Visualization and operator control of a FlipFlop
BST_ILOCK	FB651	Visualization and operator control of a logic AND/OR gate with 8 inputs
BST_VALVE	FB630	Visualization and operator control of a binary valve
BST_MOTOR	FB620	Visualization and operator control of a motor with fixed speed and rotational direction
BST_SIMODIR	FB611	Visualization and control of the motor management system SIMOCODE Pro as direct starter.
BST_SIMOREV	FB610	Visualization and control of the motor management system SIMOCODE Pro as reversing starter.
BST_MM4	FB681	Visualization and control of the frequency converter MICROMASTER 440
BST_SINAG120	FB680	Visualization and control of the frequency converter SINAMICS G120
BST_PIDCTRL	FB670	Visualization and operator control of a software-implemented PID controller for regulating technical parameters with continuous input and output values.
BST_LAG	FB656	Simulation of a controlled system (without faceplate)
BST_SPLITR	FB657	Splitting a control signal for two actuators with opposite functions, e. g. heating and cooling (without faceplate)
BST_SIPART	FB690	Visualization and control of the positioner SIPART PS2 PA

Industry Automation and Drives Technologies Service & Support Portal

This article is taken from the Service Portal of Siemens AG, Industry Automation and Drives Technologies. The following link takes you directly to the download page of this document.

http://support.automation.siemens.com/WW/view/en/68679830

Table of Contents

Application Description

1 Task

Technological task description/overview

This application describes the configuration of the technology block BST_SINAG120 (FB680) and the associated block icon (faceplate type) with faceplate. It is used for the visualization and control of the SINAMICS G120 frequency converter with a PROFINET interface.

The block offers the following functions:

- Start / stop
- Switchover of rotational direction
- Setpoint specification
- Manual / automatic mode
- Local / remote mode
- Simulation

The following operating modes of the frequency converter are displayed:

- Stop
- Left-hand rotation
- Right-hand rotation
- Locked
- Illegal
- Overload

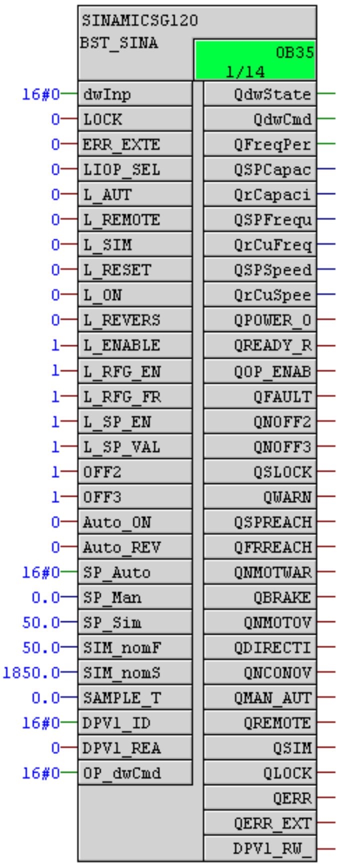
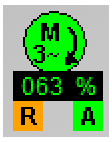
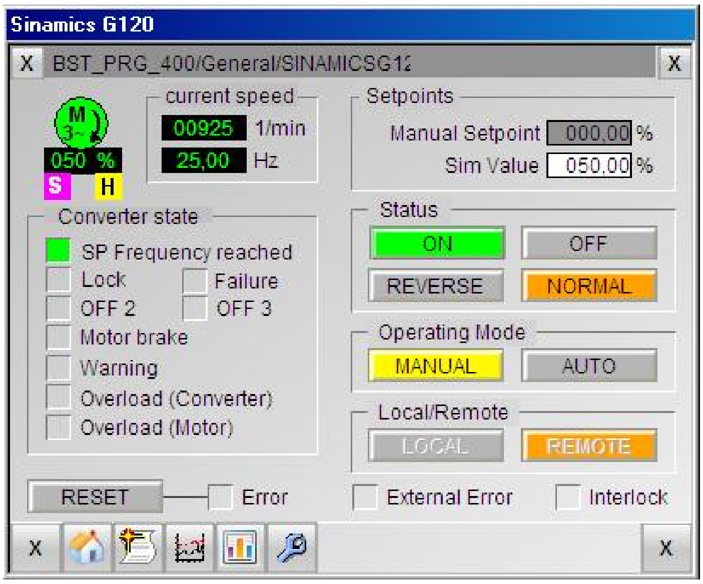
ATTENTION	**Before using the block in your own projects, check the proper functioning of the block and adjust it to your individual requirements where necessary. The block described in this application is just intended as a template for creating your own blocks.**

2 Structure of the Automation Solution

2.1 Overview

The illustrations below show the block, the block icon and the faceplate for the control, operation and monitoring of frequency converters type SINAMICS G120.

Table 2-1

<table>
<tr><th colspan="2"></th></tr>
<tr><td rowspan="2">
SINAMICSG120
BST_SINA
OB35
1/14
16#0 dwInp QdwState
0 LOCK QdwCmd
0 ERR_EXTE QFreqPer
0 LIOP_SEL QSPCapac
0 L_AUT QrCapaci
0 L_REMOTE QSPFrequ
0 L_SIM QrCuFreq
0 L_RESET QSPSpeed
0 L_ON QrCuSpee
0 L_REVERS QPOWER_O
1 L_ENABLE QREADY_R
1 L_RFG_EN QOP_ENAB
1 L_RFG_FR QFAULT
1 L_SP_EN QNOFF2
1 L_SP_VAL QNOFF3
1 OFF2 QSLOCK
1 OFF3 QWARN
0 Auto_ON QSPREACH
0 Auto_REV QFRREACH
16#0 SP_Auto QNMOTWAR
0.0 SP_Man QBRAKE
50.0 SP_Sim QNMOTOV
50.0 SIM_nomF QDIRECTI
1850.0 SIM_nomS QNCONOV
0.0 SAMPLE_T QMAN_AUT
16#0 DPV1_ID QREMOTE
0 DPV1_REA QSIM
16#0 OP_dwCmd QLOCK
QERR
QERR_EXT
DPV1_RW_</td><td>
 </td></tr>
<tr><td>
Sinamics G120
BST_PRG_400/General/SINAMICSG12
current speed
00925 1/min
25,00 Hz
Setpoints
Manual Setpoint 000,00 %
Sim Value 050,00 %
Converter state
🟩 SP Frequency reached
□ Lock □ Failure
□ OFF 2 □ OFF 3
□ Motor brake
□ Warning
□ Overload (Converter)
□ Overload (Motor)
□ Error
□ External Error
□ Interlock

X
M
3~
050 %
S
H
Status
ON
OFF
REVERSE
NORMAL
Operating Mode
MANUAL
AUTO
Local/Remote
LOCAL
REMOTE
RESET
X
X

</td></tr>
</table>

2.2 Required components

Hardware

Table 2-2

Component	Qty	Note
Development system	1	PC for the configuration of the control functions and of WinCC. The usual hardware requirements for STEP 7 and WinCC apply.
Frequency converters SINAMICS G120	1	This block has been developed specifically for use of the SINAMICS G120 frequency converter with the PROFINET option module. In the Appendix you will find hyperlinks to further information of the SINAMICS G120 frequency inverter.
S7-400 CPU or S7-300 CPU	1	This technology block is available in 2 variants. • The block for the S7-400 CPU uses a chronological message procedure (ALARM_8P). • The block for the S7-300 CPU uses a bit message procedure. As an alternative, you can also simulate system control with PLCSIM. **Note:** Under the following link you will find an FAQ list of all parameterizable modules: http://support.automation.siemens.com/WW/view/en/23678970

Software

Table 2-3

Component	Note
STEP 7 V5.4 SP3 Professional	**S7-PLCSIM** and **S7-SCL** are included in the scope of delivery. • S7-PLCSIM can be used for simulation. • S7-SCL is used for the creation of control blocks.
CFC V7.0 SP1	CFC is used for the easy interconnection of blocks. Additionally, CFC offers the transfer of texts (process tag comment, units of notation, interlock information, …) to WinCC.
WinCC V7.0 SP1	
WinCC WebNavigator V7.0 SP1	[Optional]

3 Function Mechanisms and Project Description

3.1 Alarms and messages

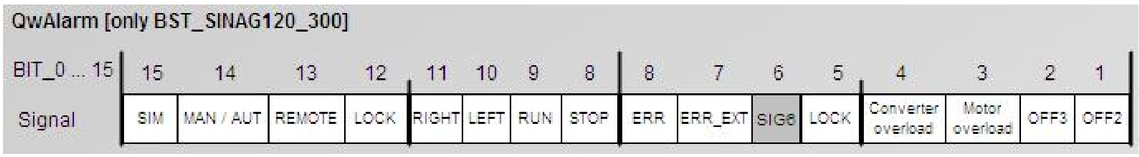
In this application, there are two options described for transferring alarms and messages from the AS to WinCC. The block for S7-300 CPUs uses the bit message principle, the block for S7-400 CPUs uses the chronological message procedure (“ALARM_8P” and “NOTIFY_8P”).

Compared to the bit message procedure, the use of the STEP 7 message blocks “ALARM_8P” and “NOTIFY_8P” considerably reduces the time and effort required for WinCC message configuration. The messages are generated in the course of OS compilation with WinCC Alarm Logging. However, they are only available within the scope of functions of S7-400 CPUs.

3.1.1 Block for S7-300 CPUs

This block uses the bit message procedure to store messages in WinCC Alarm Logging. The messages must be configured in the WinCC Alarm Logging Editor. In the WinCC tag management, a “QwAlarm” tag is generated as message tag for each block.

Figure 3-1

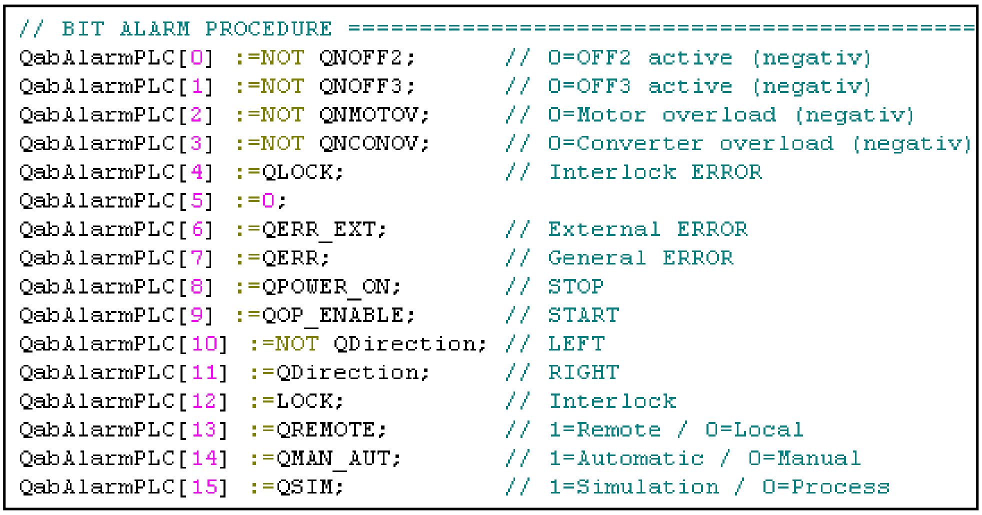


QwAlarm [only BST_SINAG120_300]

BIT_0 ... 15	15	14	13	12	11	10	9	8	8	7	6	5	4	3	2	1
Signal	SIM	MAN / AUT	REMOTE	LOCK	RIGHT	LEFT	RUN	STOP	ERR	ERR_EXT	SIG6	LOCK	Converter overload	Motor overload	OFF3	OFF2

The status signals are interlinked with the corresponding signals of the message tag in the S7 program.

Figure 3-2

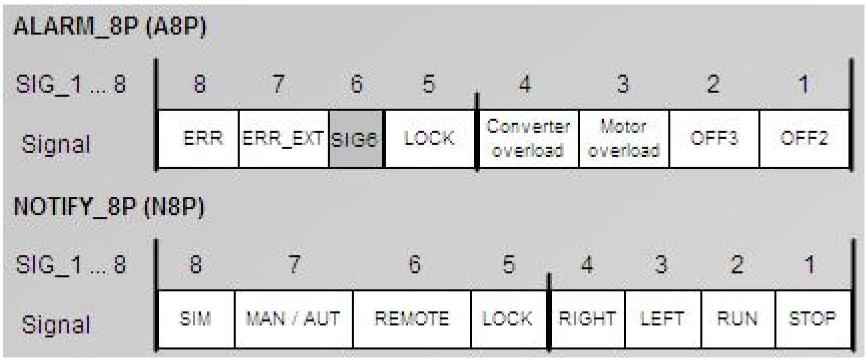


```
// BIT ALARM PROCEDURE ===========================================
QabAlarmPLC[0]  :=NOT QNOFF2;       // 0=OFF2 active (negativ)
QabAlarmPLC[1]  :=NOT QNOFF3;       // 0=OFF3 active (negativ)
QabAlarmPLC[2]  :=NOT QNMOTOV;      // 0=Motor overload (negativ)
QabAlarmPLC[3]  :=NOT QNCONOV;      // 0=Converter overload (negativ)
QabAlarmPLC[4]  :=QLOCK;            // Interlock ERROR
QabAlarmPLC[5]  :=0;
QabAlarmPLC[6]  :=QERR_EXT;         // External ERROR
QabAlarmPLC[7]  :=QERR;             // General ERROR
QabAlarmPLC[8]  :=QPOWER_ON;        // STOP
QabAlarmPLC[9]  :=QOP_ENABLE;       // START
QabAlarmPLC[10]  :=NOT QDirection; // LEFT
QabAlarmPLC[11]  :=QDirection;      // RIGHT
QabAlarmPLC[12]  :=LOCK;            // Interlock
QabAlarmPLC[13]  :=QREMOTE;         // 1=Remote / 0=Local
QabAlarmPLC[14]  :=QMAN_AUT;        // 1=Automatic / 0=Manual
QabAlarmPLC[15]  :=QSIM;            // 1=Simulation / 0=Process
```

3.1.2 Block for S7-400 CPUs

This block uses the chronological message procedure to store messages in WinCC Alarm Logging. These messages can be displayed and acknowledged in runtime under WinCC Alarm Control.

Figure 3-3



ALARM_8P (A8P)

SIG_1 ... 8	8	7	6	5	4	3	2	1
Signal	ERR	ERR_EXT	SIG6	LOCK	Converter overload	Motor overload	OFF3	OFF2

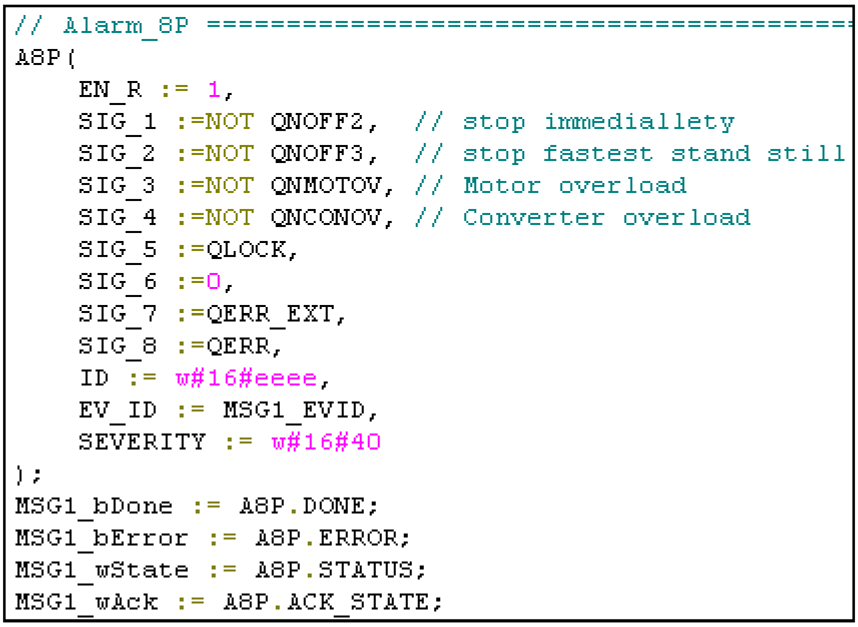
NOTIFY_8P (N8P)

SIG_1 ... 8	8	7	6	5	4	3	2	1
Signal	SIM	MAN / AUT	REMOTE	LOCK	RIGHT	LEFT	RUN	STOP

The “ALARM_8P” block for messages requiring acknowledgement

This block calls the block “ALARM_8P” (SFB 35) to create messages requiring acknowledgement. To do so, the signal inputs of the “ALARM_8P” block are linked to different status signals in the S7 program.

Figure 3-4

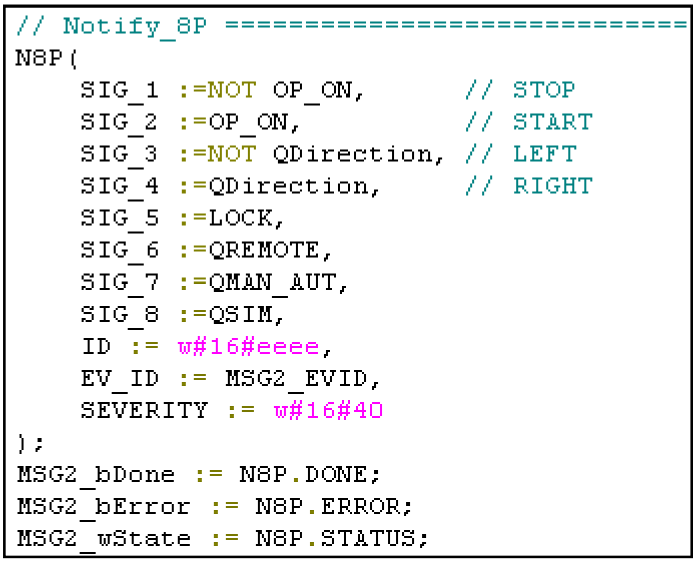


```
// Alarm_8P ============================================
A8P(
    EN_R := 1,
    SIG_1 :=NOT QNOFF2,   // stop immediallety
    SIG_2 :=NOT QNOFF3,   // stop fastest stand still
    SIG_3 :=NOT QNMOTOV,  // Motor overload
    SIG_4 :=NOT QNCONOV,  // Converter overload
    SIG_5 :=QLOCK,
    SIG_6 :=0,
    SIG_7 :=QERR_EXT,
    SIG_8 :=QERR,
    ID  := w#16#eeee,
    EV_ID := MSG1_EVID,
    SEVERITY := w#16#40
);
MSG1_bDone  := A8P.DONE;
MSG1_bError  := A8P.ERROR;
MSG1_wState  := A8P.STATUS;
MSG1_wAck := A8P.ACK_STATE;
```

Block “NOTIFY_8P” for messages not requiring acknowledgement

This block calls the block "NOTIFY_8P" (SFB 31) to create messages that do not require acknowledgement. To do so, the signal inputs of the NOTIFY_8P block are linked to different status signals in the S7 program.

Figure 3-5



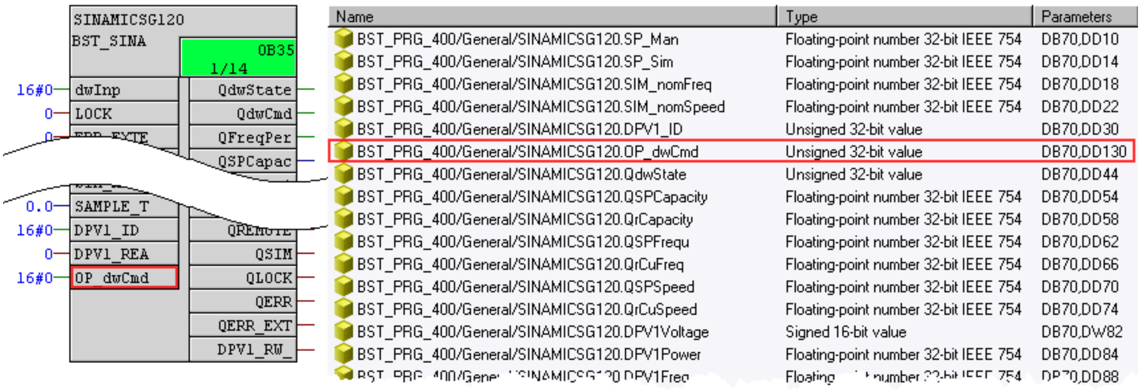
```
// Notify_8P =============================
N8P(
    SIG_1 :=NOT OP_ON,       // STOP
    SIG_2 :=OP_ON,           // START
    SIG_3 :=NOT QDirection, // LEFT
    SIG_4 :=QDirection,      // RIGHT
    SIG_5 :=LOCK,
    SIG_6 :=QREMOTE,
    SIG_7 :=QMAN_AUT,
    SIG_8 :=QSIM,
    ID  := w#16#eeee,
    EV_ID := MSG2_EVID,
    SEVERITY := w#16#40
);
MSG2_bDone := N8P.DONE;
MSG2_bError := N8P.ERROR;
MSG2_wState := N8P.STATUS;
```

3.2 Control and status signals

3.2.1 Processing of switching commands in the control system

The WinCC control tag **“OP_dwCmd”** of a block is used to transmit commands from WinCC to the PLC. The PLC block analyses the control command, executes the desired action and then initiates a reset of all pending control commands. For this purpose, the value of the control parameter is set to “0”.

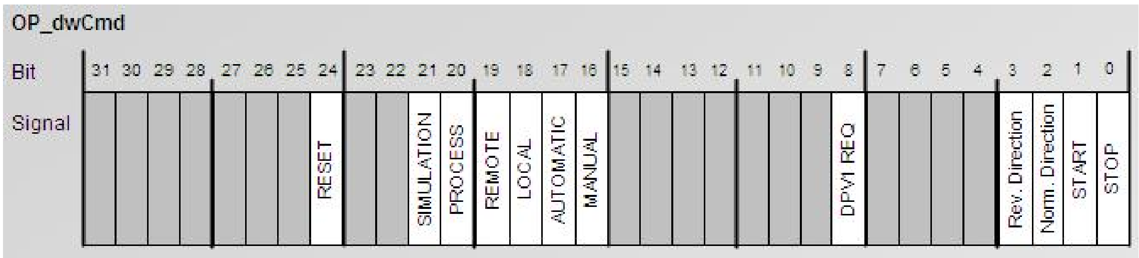
Figure 3-6



Name	Type	Parameters
BST_PRG_400/General/SINAMICSG120.SP_Man	Floating-point number 32-bit IEEE 754	DB70,DD10
BST_PRG_400/General/SINAMICSG120.SP_Sim	Floating-point number 32-bit IEEE 754	DB70,DD14
BST_PRG_400/General/SINAMICSG120.SIM_nomFreq	Floating-point number 32-bit IEEE 754	DB70,DD18
BST_PRG_400/General/SINAMICSG120.SIM_nomSpeed	Floating-point number 32-bit IEEE 754	DB70,DD22
BST_PRG_400/General/SINAMICSG120.DPV1_ID	Unsigned 32-bit value	DB70,DD30
BST_PRG_400/General/SINAMICSG120.OP_dwCmd	Unsigned 32-bit value	DB70,DD130
BST_PRG_400/General/SINAMICSG120.QdwState	Unsigned 32-bit value	DB70,DD44
BST_PRG_400/General/SINAMICSG120.QSPCapacity	Floating-point number 32-bit IEEE 754	DB70,DD54
BST_PRG_400/General/SINAMICSG120.QrCapacity	Floating-point number 32-bit IEEE 754	DB70,DD58
BST_PRG_400/General/SINAMICSG120.QSPFrequ	Floating-point number 32-bit IEEE 754	DB70,DD62
BST_PRG_400/General/SINAMICSG120.QrCuFreq	Floating-point number 32-bit IEEE 754	DB70,DD66
BST_PRG_400/General/SINAMICSG120.QSPSpeed	Floating-point number 32-bit IEEE 754	DB70,DD70
BST_PRG_400/General/SINAMICSG120.QrCuSpeed	Floating-point number 32-bit IEEE 754	DB70,DD74
BST_PRG_400/General/SINAMICSG120.DPV1Voltage	Signed 16-bit value	DB70,DW82
BST_PRG_400/General/SINAMICSG120.DPV1Power	Floating-point number 32-bit IEEE 754	DB70,DD84
BST_PRG_400/General/SINAMICSG120.DPV1Freq	Floating-point number 32-bit IEEE 754	DB70,DD88

The bits of the “OP_dwCmd” tag are assigned as follows:

Figure 3-7



OP_dwCmd

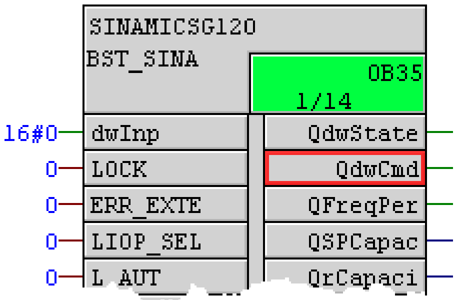
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Signal								RESET			SIMULATION	PROCESS	REMOTE	LOCAL	AUTOMATIC	MANUAL								DPV1 REQ					Rev. Direction	Norm. Direction	START	STOP

3.2.2 Transmission of Switching Commands to SINAMICS G120

Control signals to SINAMICS G120 are transmitted via the output “QdwCmd”. During hardware configuration, the standard message frame 1 for PROFINET was used. For information on the bit assignment of this standard message frame, please refer to the "Control Unit" manual under the following link:

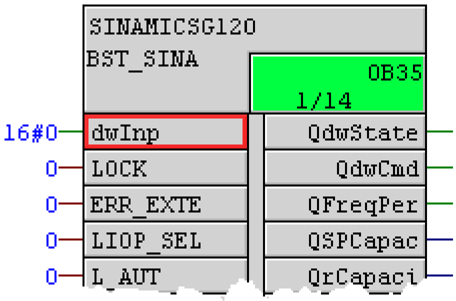
http://support.automation.siemens.com/WW/view/en/27864729

Figure 3-8



Status information of the SINAMICS G120 is read in via the input “dwInp”.

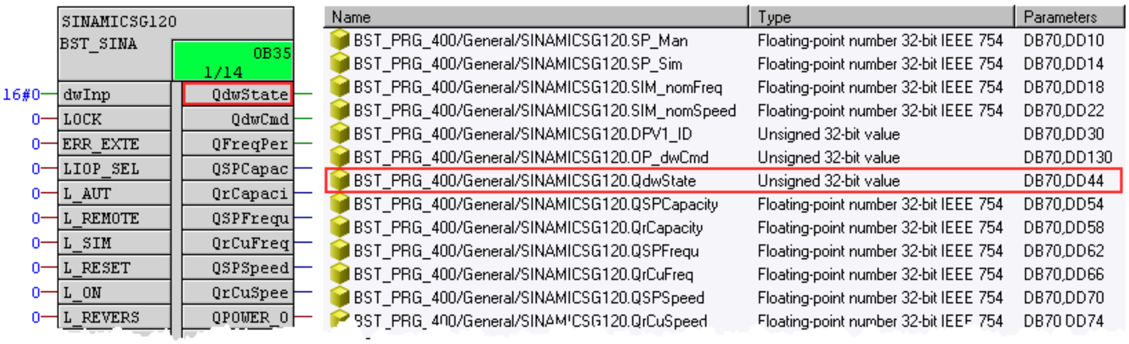
Figure 3-9



3.2.3 Analysis of process states and their output in the WinCC process display

The technology block of this application uses the output **“QdwState”** (DWORD) to show the different states in the WinCC process display.

Figure 3-10



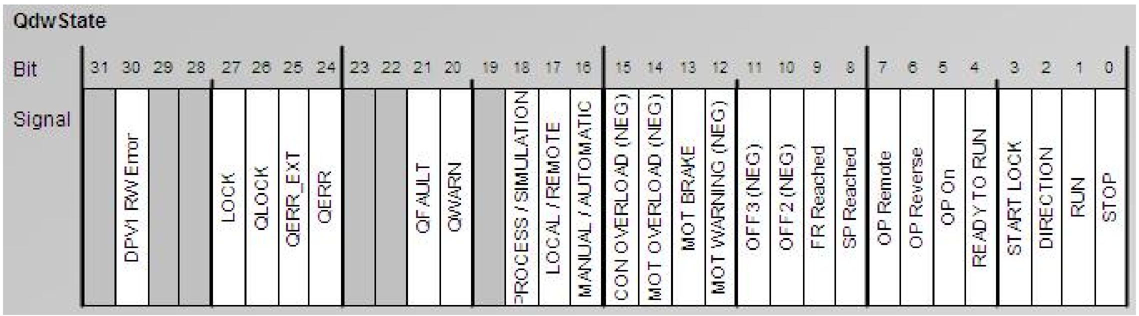
SINAMICSG120
BST_SINA
OB35
1/14

	Input	Output
16#0	dwInp	QdwState
0	LOCK	QdwCmd
0	ERR_EXTE	QFreqPer
0	LIOP_SEL	QSPCapac
0	L_AUT	QrCapaci
0	L_REMOTE	QSPFrequ
0	L_SIM	QrCuFreq
0	L_RESET	QSPSpeed
0	L_ON	QrCuSpee
0	L_REVERS	QPOWER_O

Name	Type	Parameters
BST_PRG_400/General/SINAMICSG120.SP_Man	Floating-point number 32-bit IEEE 754	DB70,DD10
BST_PRG_400/General/SINAMICSG120.SP_Sim	Floating-point number 32-bit IEEE 754	DB70,DD14
BST_PRG_400/General/SINAMICSG120.SIM_nomFreq	Floating-point number 32-bit IEEE 754	DB70,DD18
BST_PRG_400/General/SINAMICSG120.SIM_nomSpeed	Floating-point number 32-bit IEEE 754	DB70,DD22
BST_PRG_400/General/SINAMICSG120.DPV1_ID	Unsigned 32-bit value	DB70,DD30
BST_PRG_400/General/SINAMICSG120.OP_dwCmd	Unsigned 32-bit value	DB70,DD130
BST_PRG_400/General/SINAMICSG120.QdwState	Unsigned 32-bit value	DB70,DD44
BST_PRG_400/General/SINAMICSG120.QSPCapacity	Floating-point number 32-bit IEEE 754	DB70,DD54
BST_PRG_400/General/SINAMICSG120.QrCapacity	Floating-point number 32-bit IEEE 754	DB70,DD58
BST_PRG_400/General/SINAMICSG120.QSPFrequ	Floating-point number 32-bit IEEE 754	DB70,DD62
BST_PRG_400/General/SINAMICSG120.QrCuFreq	Floating-point number 32-bit IEEE 754	DB70,DD66
BST_PRG_400/General/SINAMICSG120.QSPSpeed	Floating-point number 32-bit IEEE 754	DB70,DD70
BST_PRG_400/General/SINAMICSG120.QrCuSpeed	Floating-point number 32-bit IEEE 754	DB70,DD74

The bits of the “QdwState” tag are assigned as follows:

Figure 3-11



QdwState

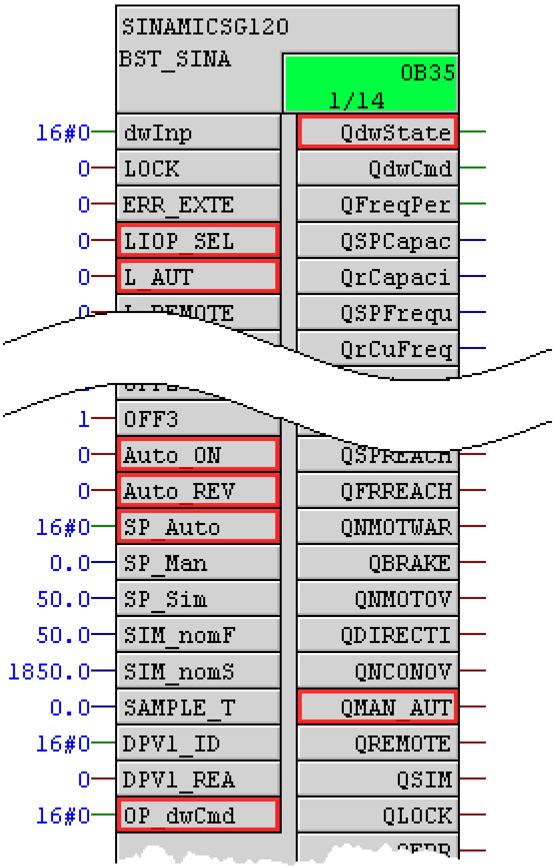
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Signal		DPV1 RW Error			LOCK	QLOCK	QERR_EXT	QERR			QFAULT	QWARN		PROCESS / SIMULATION	LOCAL / REMOTE	MANUAL / AUTOMATIC

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Signal	CON OVERLOAD (NEG)	MOT OVERLOAD (NEG)	MOT BRAKE	MOT WARNING (NEG)	OFF3 (NEG)	OFF2 (NEG)	FR Reached	SP Reached	OP Remote	OP Reverse	OP On	READY TO RUN	START LOCK	DIRECTION	RUN	STOP

3.2.4 Manual/automatic mode

For “manual/automatic” switchover, the following signals are used.

Figure 3-12



- ***“LIOP_SEL”***
  If the “LIOP_SEL” input is set, the mode is determined via the control input “L_AUT”.
  If the “LIOP_SEL” input is not set, the mode is determined by the operator (“OP_dwCmd” [Bit 16 and 17]).
- ***“L_AUT”***
  The “L_AUT” input is only active if the “LIOP_SEL” input is set.

“LIOP_SEL” = **1**	AND	“L_AUT” = **0**	➔	Manual
“LIOP_SEL” = **1**	AND	“L_AUT” = **1**	➔	Automatic

- ***"OP_dwCmd"***
  The operator commands of the control word "OP_dwCmd" are active only if the "LIOP_SEL" input is not set.

"LIOP_SEL" = **0**	AND	"OP_dwCmd [Bit 16]" = **1** ➔ Manual
"LIOP_SEL" = **0**	AND	"OP_dwCmd [Bit 17]" = 1 ➔ Automatic

- **"*QMAN_AUT*"**
  The operating mode is indicated at the "QMAN_AUT" block output.

"QMAN_AUT" = **0**	➔	Manual
"QMAN_AUT" = **1**	➔	Automatic

- ***"QdwState"***
  The current mode is transferred to WinCC in the status word "QdwState [Bit 16]".

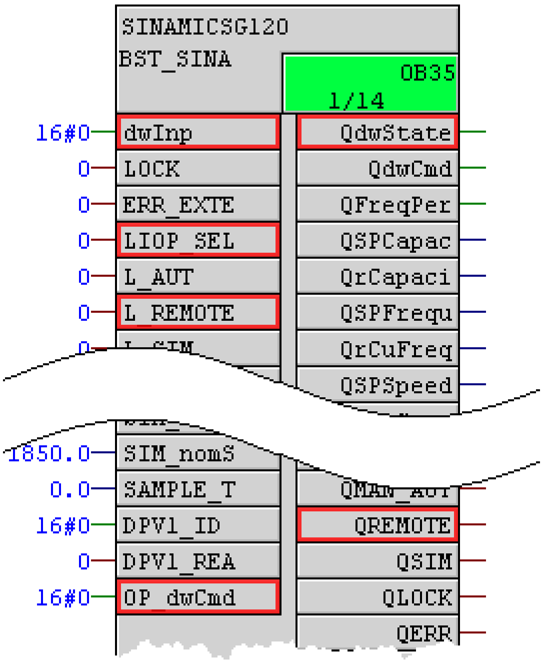
"QdwState [Bit 16]" = **0**	➔	Manual
"QdwState [Bit 16]" = **1**	➔	Automatic

- ***"Auto_ON", "Auto_REV", "SP_Auto" and "SP_Man"***
  If the automatic mode is selected, the inputs "Auto_ON", "Auto_REV" and "SP_Auto" are active.
  If the manual mode is selected, the operation via WinCC is enabled and the "SP_Man" input is active.

3.2.5 Local/remote switchover

In “Local” mode, the control commands of the user are enabled directly at the device. In “Remote” mode, the control commands of an automatic function (e.g. control, step sequence ...) or the control commands of the WinCC-Runtime operation are active.

Figure 3-13



- ***“LIOP_SEL”***
  If the “LIOP_SEL” input is set, the mode is determined via the control input “L_REMOTE”.
  If the “LIOP_SEL” input is not set, the mode is determined by the operator (“OP_dwCmd” [Bit 18 and 19]).
- ***“L_REMOTE”***
  The “L_REMOTE” input is only active if the “LIOP_SEL” input is set.

“LIOP_SEL” = **1**	AND	“L_REMOTE” = **0**	➔	Local
“LIOP_SEL” = **1**	AND	“L_REMOTE” = **1**	➔	Remote

- ***“OP_dwCmd (s7_m_c = true)”***
  The operator commands in control word “OP_dwCmd” become effective only, if input “LIOP_SEL” is not enabled.

“LIOP_SEL” = **0**	AND	“OP_dwCmd [Bit 18]” = **1**	➔	Local
“LIOP_SEL” = **0**	AND	“OP_dwCmd [Bit 19]” = **1**	➔	Remote

- ***“dwInp”***
  The “dwInp” input is used to evaluate the status word of MICROMASTER and set the “QREMOTE” output accordingly.

“dwInp [Bit 9]” = **0**	➔	Local
“dwInp [Bit 9]” = **1**	➔	Remote

- **“*QREMOTE*”**
  The operating mode is indicated at the “QREMOTE” block output.

“QREMOTE” = **0**	➔	Local
“QREMOTE” = **1**	➔	Remote

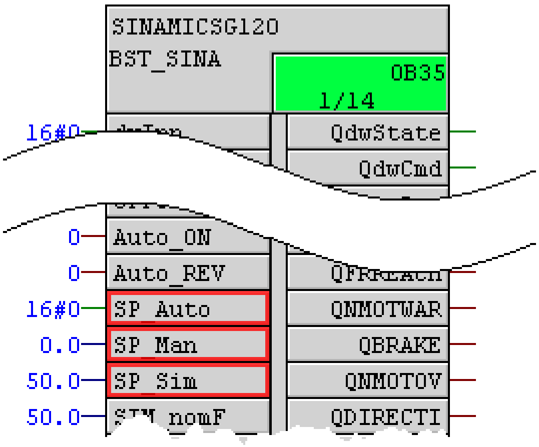
- ***"QdwState"***
  The status word for WinCC shows the switching command at [Bit 6] and the SINAMICS status for remote control at [Bit 17].

“QdwState [Bit 7]” = **0**	➔	Command for local operation
"QdwState [Bit 7]" = **1**	➔	Command for remote operation
"QdwState [Bit 17]" = **0**	➔	Local
"QdwState [Bit 17]" = **1**	➔	Remote

3.2.6 Setpoint specification

Setpoints can be linked to the inputs “SP_Auto” and “SP_Man”. If no peripheral devices are connected, the block can be used in simulation mode. For this purpose, the “SP_Sim” input is activated.

Figure 3-14



- ***“SP_Auto”***
  The “SP_Auto” input is used to set the setpoint in the peripheral format. (16#0000 to 16#4000)
- ***“SP_Man”***
  The “SP_Man” input is used to set the setpoint in the floating point format. (0.0 % to 100.0 %)
- ***“SP_Sim”***
  The “SP_Sim” input is used to set the setpoint in the simulation mode.

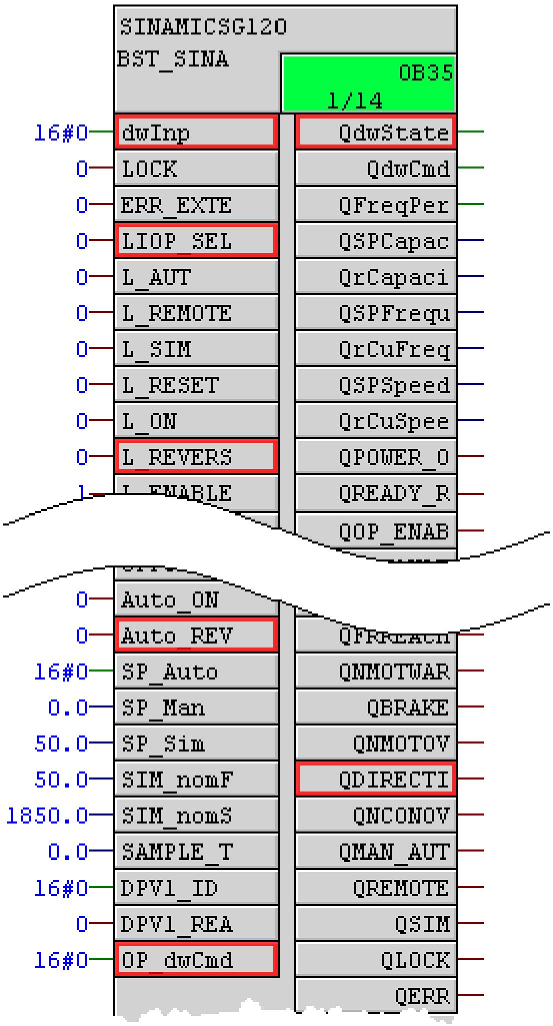
Depending on the current “manual”, “automatic” or “simulation” mode, one of the inputs “SP_Auto”, “SP_Man” or “SP_Sim” is active.

“QSIM” = **0** AND “QMAN_AUT” = **0**	➔	Input “SP_Man” is active
“QSIM” = **0** UND “QMAN_AUT” = **1**	➔	Input “SP_Auto” is active
“QSIM” = **1**	➔	Input “SP_Sim” is active

3.2.7 Specification of rotational direction

The control of the rotational direction is implemented by means of setpoint inversion. A positive setpoint value means rotation in a clockwise direction.

Figure 3-15



- ***"LIOP_SEL"***
  If the "LIOP_SEL" input is set, the rotational direction is determined via the control input "L_REVERSE".
  If the "LIOP_SEL" input is not set, the mode is determined by the operator ("OP_dwCmd" [Bit 2 and 3]).
- ***"L_REVERSE"***
  The "L_REVERSE" input is only active if the "LIOP_SEL" input is set.

  "LIOP_SEL" = **1** AND "L_REVERSE" = **0** ➔ Right for pos. setpoint
  "LIOP_SEL" = **1** AND "L_REVERSE" = **1** ➔ Left for pos. setpoint

- ***"OP_dwCmd (s7_m_c = true)"***
  The operator commands in control word "OP_dwCmd" become effective only, if input "LIOP_SEL" is not enabled.

  "LIOP_SEL" = **0** AND "OP_dwCmd [Bit 2]" = **1** ➔ Right for pos. setpoint
  "LIOP_SEL" = **0** AND "OP_dwCmd [Bit 3]" = **1** ➔ Left for pos. setpoint
- ***"Auto_REV"***
  In automatic mode, the rotational direction is controlled via the "Auto_REV" input.

  "QMAN_OUT"= **1** AND "Auto_REV" = **0** ➔ Right for pos. setpoint
  "QMAN_OUT"= **1** AND "Auto_REV" = **1** ➔ Left for pos. setpoint

- ***"dwInp"***
  The "dwInp" input is used to evaluate the status word of MICROMASTER and set the "QDIRECTION" output accordingly. For this, the status signal of MICROMASTER is negated in case of a positive setpoint.

"dwInp [Bit 14]" = **0**	➔	Rotational direction left
"dwInp [Bit 14]" = **1**	➔	Rotational direction right

- **"*QDIRECTION*"**
  The current rotational direction is indicated at the "QDIRECTION" block output.

"QDIRECTION" = **0**	➔	Rotational direction left
"QDIRECTION" = **1**	➔	Rotational direction right

- ***"QdwState"***
  The status word for WinCC indicates with "bit 6" the switching command and with "bit 2" the MICROMASTER status for the inversion of the rotational direction.

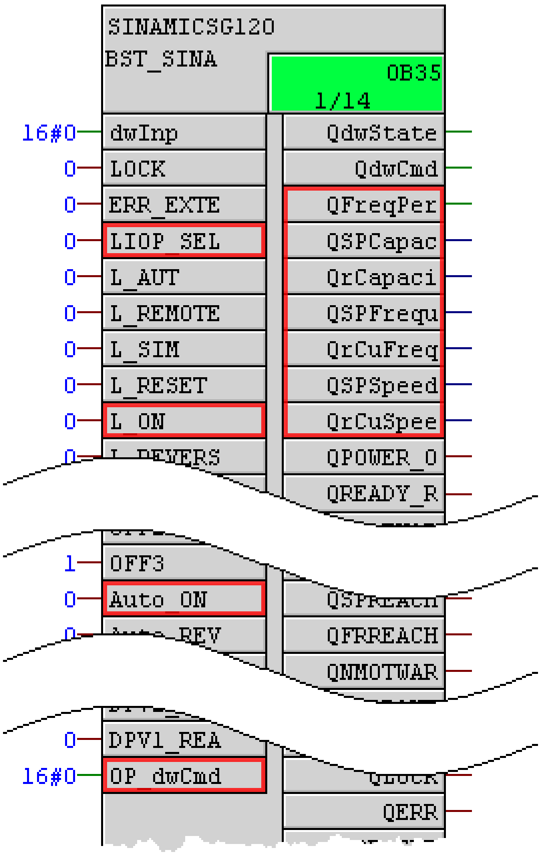
"QdwState [Bit 6]" = **0**	➔	Command for "clockwise"
"QdwState [Bit 6]" = **1**	➔	Command for "anti-clockwise"
"QdwState [Bit 2]" = **0**	➔	Anticlockwise rotation
„QdwState [Bit 2]" = **1**	➔	Rotational direction right

Note:
At the WinCC faceplate, the specification of a negative setpoint is not allowed. Here, the change of rotational direction is performed by means of the "reverse" switch.

3.2.8 Start/stop motor

Switchover is influenced by other functions, such as “manual / automatic” switchover, “local / remote” switchover and “simulation”.

Figure 3-16



- ***“LIOP_SEL”***
  If the “LIOP_SEL” input is set, the motor is started via the control input “L_ON”. If the “LIOP_SEL” input is not set, the motor is started (“OP_dwCmd [Bit 1]”) and stopped (“OP_dwCmd [Bit 0]”) by the operator or is controlled via the “Auto_ON” input in automatic mode.
- ***“L_ON”***

“LIOP_SEL” = **1**	AND	“L_ON” = **0**	➔	Stop
“LIOP_SEL” = **1**	AND	“L_ON” = **1**	➔	Start

- ***“OP_dwCmd”***
  Bits 0 and 1 in the control word “OP_dwCmd” are used for WinCC Runtime operation.

“LIOP_SEL” = **0**	AND	“OP_dwCmd [Bit 0]” = **1**➔	Stop
“LIOP_SEL” = **0**	AND	“OP_dwCmd [Bit 1]” = **1**➔	Start

- ***"Auto_ON"***
  The "AUTO_ON" input serves for automatic control of the block. The input can be linked to other blocks.
- **Current setpoint and actual value**
  At the following outputs, the current setpoint is indicated:

- QSPCapacity	➔	Setpoint in percent
- QSPFrequency	➔	Setpoint for converter frequency
- QSPSpeed	➔	Setpoint for motor speed

At the following outputs, the current actual value is indicated

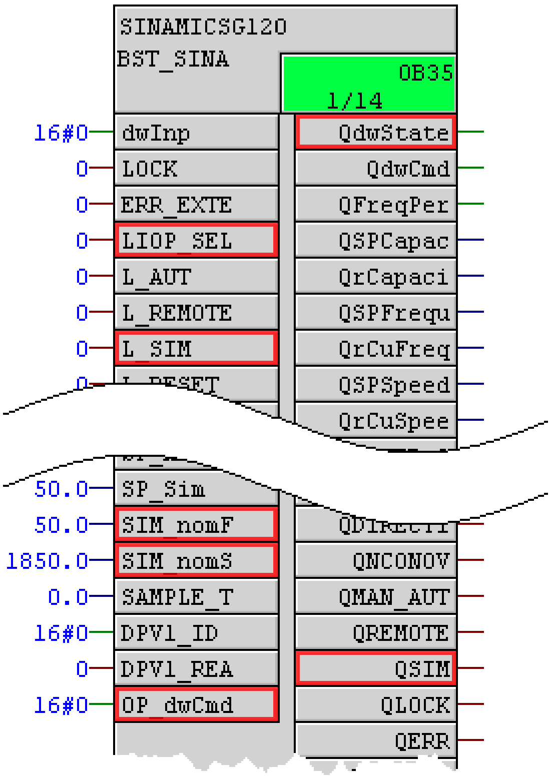
- QFreqPeri	➔	Actual value in peripheral format
- QrCapacity	➔	Actual value in percent
- QrCuFrequency	➔	Actual value for converter frequency
- QrCuSpeed	➔	Actual motor speed value

Correct display of the motor speed and converter frequency depends on the successful read-out of the parameters "P0310" and "P0311" from SINAMICS G120 by means of a DPV1 diagnosis.

3.2.9 Simulation On/Off

The operating mode "Simulation ON" allows simulation of the frequency converter. This is an important function, e. g. to test automatic functions already during the configuration phase when a connection to the process is not yet available. Without feedback simulation, many automation functions (e.g. step sequences) cannot be successfully performed, since the feedback signals are often required for switching conditions in case of step sequences or for triggering error messages. The "dwInp" input and the "QdwCmd" output are not evaluated in this mode.

Figure 3-17



- ***"LIOP_SEL"***
  If the "LIOP_SEL" input is set, the simulation is switched on/off via the control input "L_SIM".
  If the "LIOP_SEL" input is not set, the simulation is switched on/off by the operator ("OP_dwCmd" [Bit 20 and 21]).
- *"L_SIM"*

"LIOP_SEL" = 1	AND	"L_SIM" = 0	➔	Simulation **OFF**
"LIOP_SEL" = 1	AND	"L_SIM" = 1	➔	Simulation **ON**

- ***"SIM_nomFreq"***

  At the "SIM_nomFreq" input, the simulated nominal frequency can be parameterized. It is used for the calculation of the simulated value of the output.
- ***"SIM_nomSpeed"***

  At the "SIM_nomSpeed" input, the simulated nominal speed can be parameterized. It is used for the calculation of the simulated value of the output.

- ***"OP_dwCmd"***
  The operator commands of the control word "OP_dwCmd" are active only if the "LIOP_SEL" input is not set.

  "LIOP_SEL" = **0** AND "OP_dwCmd [Bit 20]" = **1** ➔ Simulation **ON**
  "LIOP_SEL" = **0** AND "OP_dwCmd [Bit 21]" = **1** ➔ Simulation **OFF**

- **"*QSIM*"**
  The "Simulation" mode is indicated at the "QSIM" block output.

  "QSIM" = **0** ➔ Simulation **OFF**
  "QSIM" = **1** ➔ Simulation **ON**

- ***"QdwState"***
  The current "Simulation" mode is indicated in the status word by means of bit 10.

  "QdwState [Bit 10]" = **0**➔ Simulation **OFF**
  "QdwState [Bit 10]" = **1**➔ Simulation **ON**

3.2.10 Interlock

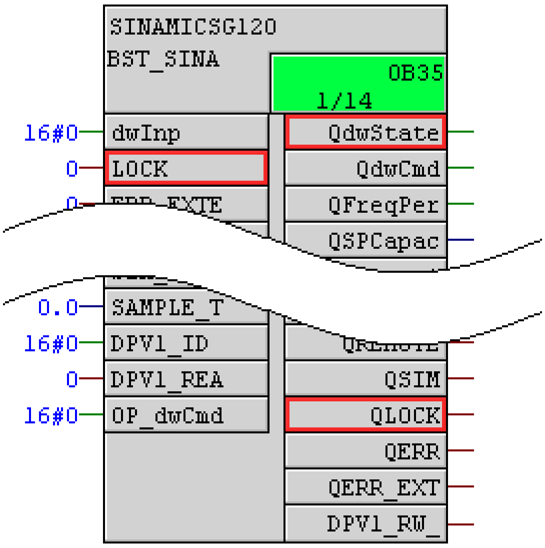
Blocks that influence the plant process as actuators (e.g. valves, pumps, controllers etc.) usually have signals for the interlock function (INTERLOCK).

If the interlock condition of a block is active, the specific function of the block (e.g. open valve, switch on motor etc.) cannot be implemented.

If the specific function of the block is already active (e.g. valve open, motor on etc.) when the interlock condition applies, the respective function of the block is deactivated (e.g. valve closed, motor switched off).

This function is used to avoid critical plant conditions.

Figure 3-18



The “LOCK” signal transfers the “OFF2” command to MICROMASTER (OFF2 – impulse lock, drive coasts down). If the interlock is not longer present, the drive can be activated again only, if a stop signal is applied.

- ***“LOCK”***
  The input signal “LOCK” contains the interlock condition. If the interlock condition is active, the specific block function cannot be performed.

  LOCK = **0** ➔ Interlock not pending
  LOCK = **1** ➔ Interlock pending

- ***“QLOCK”***
  The interlock is displayed at the “QLOCK” output.

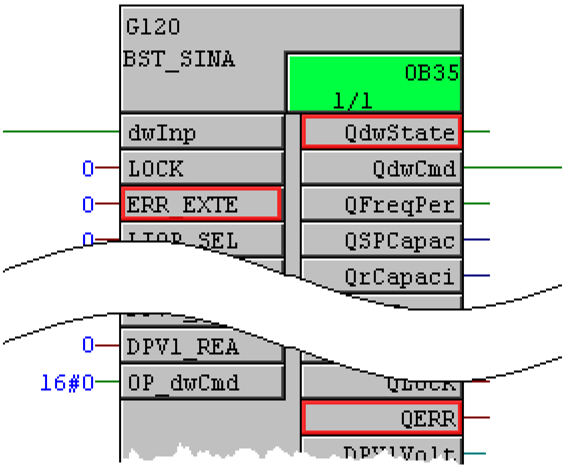
  “QLOCK” = **0** ➔ Interlock not pending
  “QLOCK” = **1** ➔ Interlock pending

- ***“QdwState”***
  The states “Interlock condition active” (LOCK=1) and “Error Interlock” (QLOCK=1) are displayed in the status word by means of bits 26 and 27.

  “QdwState [Bit 27]” = **1** ➔ LOCK = **1** (Interlock condition active)
  “QdwState [Bit 26]” = **1** ➔ QLOCK = **1** (Error Interlock active)

3.2.11 External error

Figure 3-19



The error signal sends the OFF3 command to MICROMASTER (OFF3 – fast stop, stop with shortest deceleration time). If the error is not longer present, the drive can be activated again only, if a stop signal is applied.

- ***"ERR_EXTERN"***
  The input signal "LOCK" includes the interlock condition. If the interlock condition is active, the specific block function cannot be performed.

  "ERR_EXTERN" = **0** ➔ No error pending
  "ERR_EXTERN" = **1** ➔ Error pending

- ***"QERR"***
  An interlock is indicated at output "QERR".

  "QERR" = **0** ➔ No error pending
  "QERR" = **1** ➔ Error pending

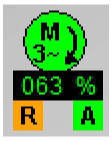
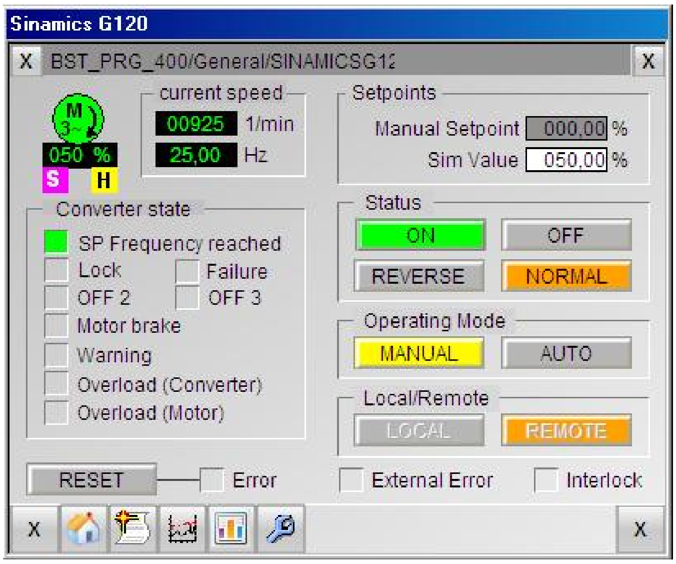
- ***"QdwState"***
  The error is transferred to WinCC with "bit 15" of the status word.
  "QdwState [Bit 15]" = **0**➔ No error pending
  "QdwState [Bit 15]" = **1**➔ Error pending

3.3 Block icon and faceplate

This example uses WinCC faceplate types. These faceplates are available for WinCC V7 or a later version and they offer the advantage of central changeability. This relieves you from the task of revising all process displays when modifying the block icon. If the faceplate types are not available, you can also configure WinCC user object.

Layout of the block icon and the associated faceplate

Table 3-1

Block icon	Faceplate
	

The technological faceplate provides the following functionality

- **Displayed operating states:**
  - Status stop/run
  - Setpoint / actual value
  - Setpoint inversion (rotational direction)
  - Local / remote mode
  - Manual / automatic mode
  - Simulation mode
  - Current speed and converter frequency
  - Converter status
- **Displayed error messages:**
  - External error
  - Interlock

- **Operation:**
  - Start / stop
  - Setpoint specification (in %)
  - Setpoint inversion
  - Resetting errors
  - Local / remote switchover
  - Manual / automatic switchover
  - Simulation On/Off
- **Further information:**
  - DPV1 data for SINAMICS G120
  - Trend view of setpoint and actual value (in %)
  - Alarms and messages of the block

3.3.1 Block icon

For each of the individual operating modes “Stop/Run/Illegal”, a picture has been created. The operating mode is transferred to WinCC via the “QdwState” tag [bit 0-3]. The display of the block icon is defined in the WinCC picture file "BST_SINAG120_ICON_Define.pdl".

- M 3~ Stop
- M 3~ Left-hand rotation started
- M 3~ Right-hand rotation started
- M 3~ Locked
- M 3~ Illegal status

3.3.2 Status displays

For the various characteristics (states) of a block, there are separate status displays configured.

- Display local / remote operation
  L Operating mode “local”
  R Operating mode “remote”
- Display manual / automatic operation
  H Operating mode “manual”
  A Operating mode “automatic”
- Display simulation on / off
  S Simulation “on”

- Display failure / warning

  F General failure

  W General warning

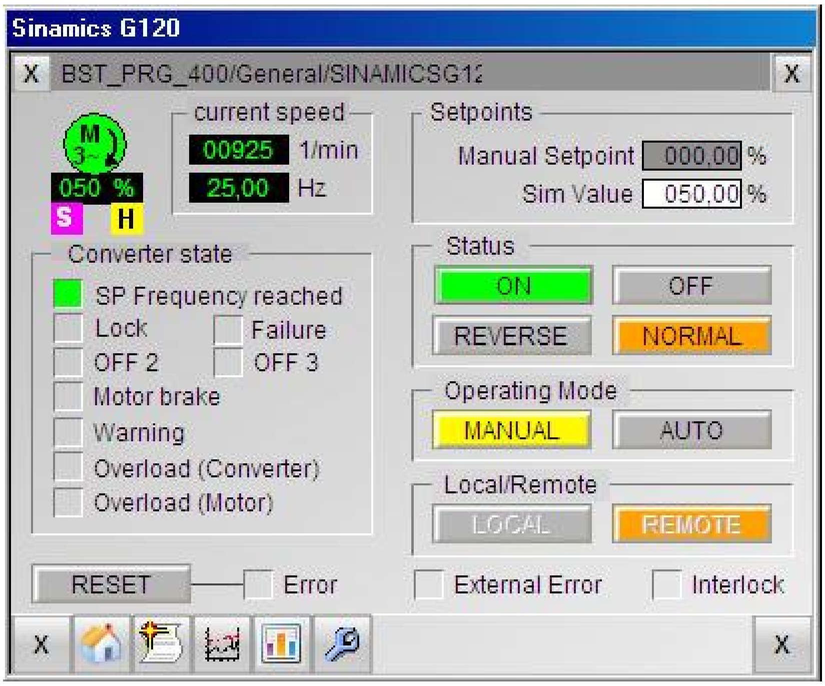
- Display interlock

  IL Interlock active

The icon **"BST_SINAG120_ICON_Define.pdl"** for the configuration of status displays of block icons has been created in the Graphics Designer program. In this picture, all states of the status displays of a block type are represented (configured). In addition to this, the names of the picture files and the status values are specified.

3.3.3 Faceplate – Standard view

Figure 3-20



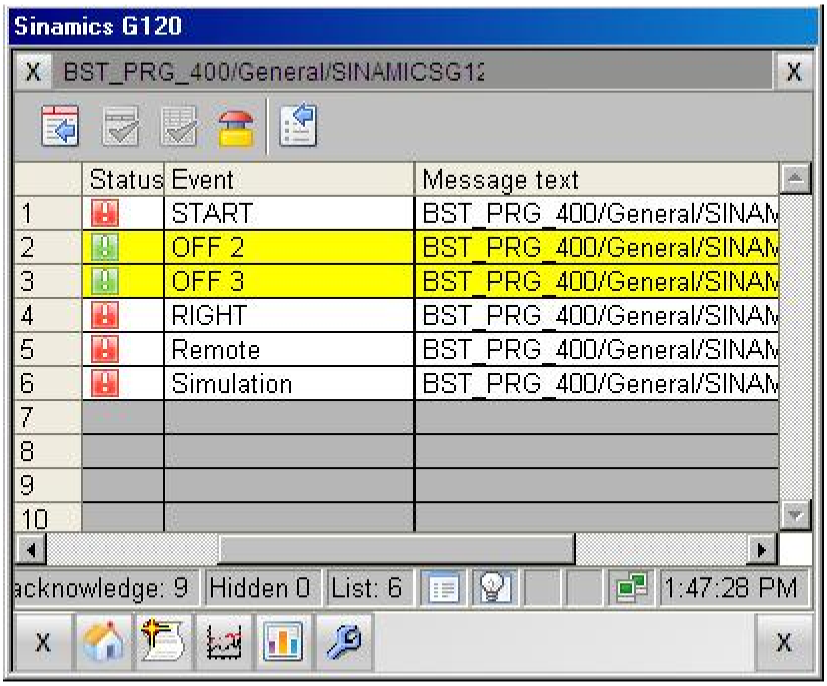
Click the following icon to open the standard view:

It provides the following functions:

- Display of SINAMICS G120 status information
- Switchover between “Local” (operation at the device) and “Remote” (operation at the visualization system)
- Manual / automatic switchover.
- Setpoint specification, setpoint inversion
- Start / stop drive
- Resetting of error states

3.3.4 Faceplate – Alarms and messages view

Figure 3-21



Sinamics G120

X BST_PRG_400/General/SINAMICSG12 X

	Status	Event	Message text
1	🟥	START	BST_PRG_400/General/SINAM
2	🟩	OFF 2	BST_PRG_400/General/SINAM
3	🟩	OFF 3	BST_PRG_400/General/SINAM
4	🟥	RIGHT	BST_PRG_400/General/SINAM
5	🟥	Remote	BST_PRG_400/General/SINAM
6	🟥	Simulation	BST_PRG_400/General/SINAM
7			
8			
9			
10			

acknowledge: 9 Hidden 0 List: 6 1:47:28 PM

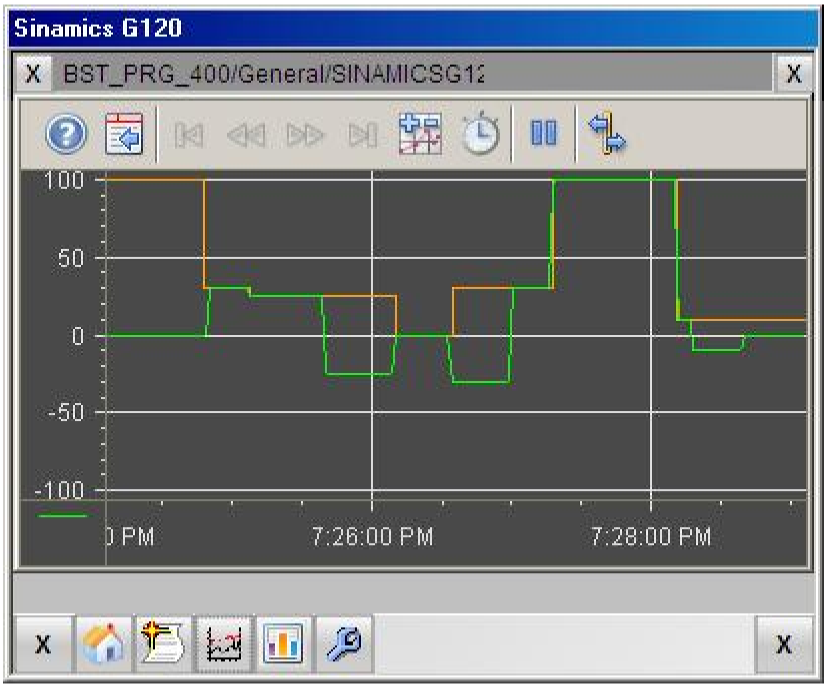
Click the following button to open the alarms and messages view:

In this view you can perform the following actions:

- Display of block-related alarms and messages.
- Acknowledgement of alarms.

3.3.5 Faceplate – Trend view

Figure 3-22

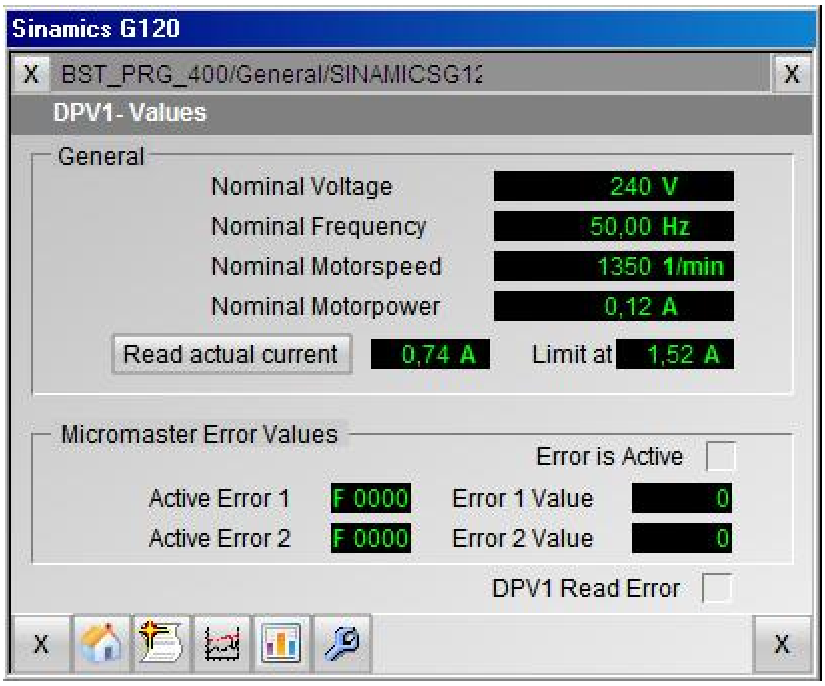


You can go to the trend view via the following button:

The curves show the specified setpoint and actual value in percent.

3.3.6 Faceplate– DPV1 values

Figure 3-23



Sinamics G120

X BST_PRG_400/General/SINAMICSG12 X

DPV1- Values

General

Nominal Voltage	240 V
Nominal Frequency	50,00 Hz
Nominal Motorspeed	1350 1/min
Nominal Motorpower	0,12 A

Read actual current 0,74 A Limit at 1,52 A

Micromaster Error Values

Error is Active □

Active Error 1	F 0000	Error 1 Value	0
Active Error 2	F 0000	Error 2 Value	0

DPV1 Read Error □

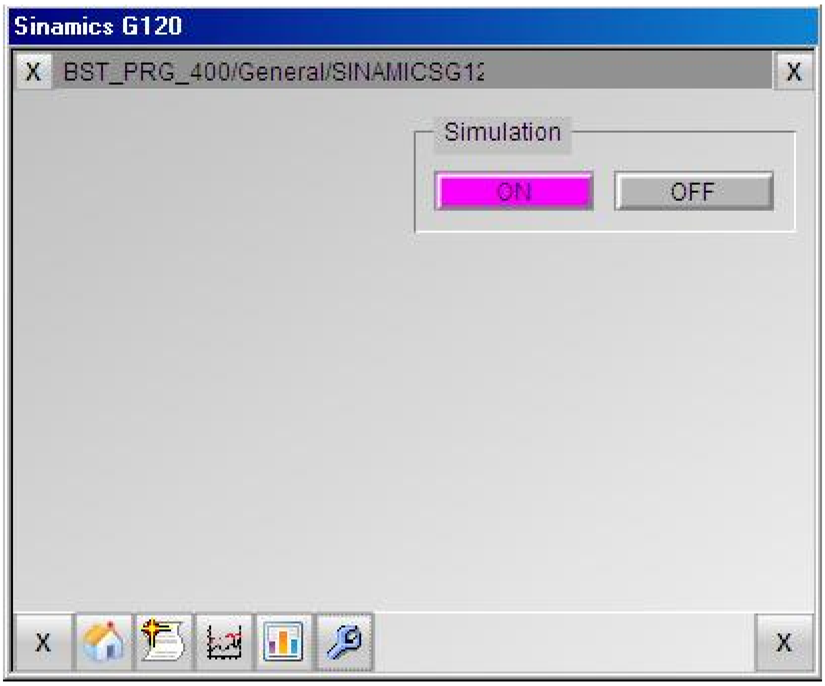
You can display the DPV1 data via the following button:

The data is read out from SINAMICS G120 asynchronously via the standard blocks SFB52 (RDREC) and SFB53 (WRREC). This is only an extract from all possible information. For details on the data provided by the SINAMICS G120 frequency converter, please refer to the system manual.

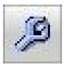
http://support.automation.siemens.com/WW/view/en/32465038

3.3.7 Faceplate – Service view

Figure 3-24



Click the following button to open the service view:



The service view provides the following functions:

- Switch between simulation / process operation.

4 Installation and Startup

Here you can learn…

how to integrate the “BST_SINAG120” block into an empty or a previous project. This example uses the unconnected STEP 7 block, the WinCC block icon (faceplate type) and the WinCC faceplate, which is embedded in a static picture window.

The document “Example blocks for WinCC and STEP 7” describes how to configure a dynamic call of the WinCC faceplates.

Requirements:

- STEP 7 project has been created
- The hardware has been configured, compiled and loaded
- The SINAMICS G120 frequency converter has been configured with “Standard Message Frame 1” as defined in the system manual.

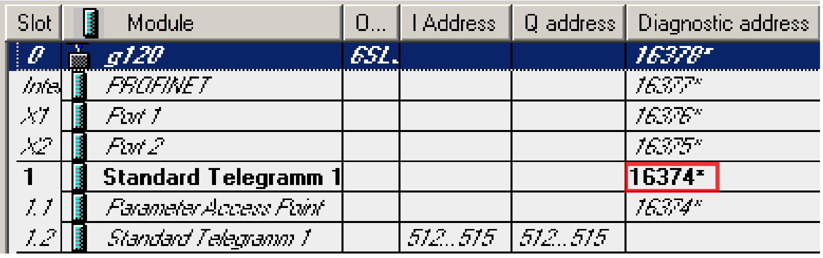
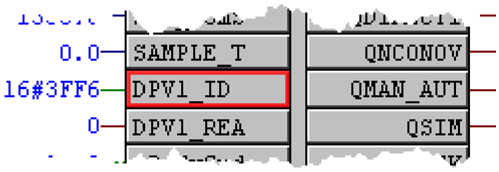
4.1 Configuring the STEP 7 block

First you create a new STEP 7 project with “S7 4xx” control or with “S7 3xx” control and a PC station (WinCC application), or open a previously created project.

Table 4-1

Step	Description
1.	**Insert blocks** • Open the delivered “BST_LIBRARY” library with the SIMATIC Manager. • Copy the FB 680 block and the SCL source “BST_SINAG120_xxx” into the respective directory of the S7 program or the project library. The block uses several standard blocks which also need to be added to the project. • Open the “Standard Library” and insert the following blocks in your S7 program: - System Function Blocks > Blocks > SFC6 (RD_SINFO) - System Function Blocks > Blocks > SFB52 (RDREC) - System Function Blocks > Blocks > SFB53 (WRREC) Only S7-400: - System Function Blocks > Blocks > SFB31 (Notify_8P) - System Function Blocks > Blocks > SFB35 (Alarm_8P)
2.	**Create CFC chart** • Add the block to a new or to a previous CFC chart. • Interlink the process output word of SINAMICS G120 with the block input “dwInp”. • Interlink the process input word of SINAMICS G120 with the block output “QdwCmd”.

<table>
<tr><th>Step</th><th>Description</th></tr>
<tr><td>3.</td><td>
<b>DPV1 – Diagnosis</b>

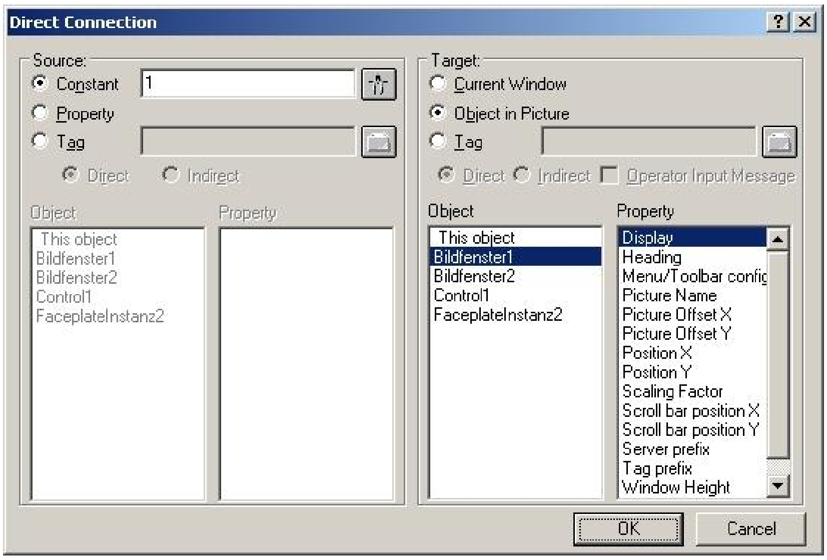
For the DPV1 data to be read out correctly, you have to parameterize the diagnosis address at the “DPV1_ID” input of the block. Parameterization and readout can be performed in the properties of the DP slave.

<table>
<tr><th>Slot</th><th>Module</th><th>O...</th><th>I Address</th><th>Q address</th><th>Diagnostic address</th></tr>
<tr><td>0</td><td>g120</td><td>6SL.</td><td></td><td></td><td>16378*</td></tr>
<tr><td>Inte</td><td>PROFINET</td><td></td><td></td><td></td><td>16377*</td></tr>
<tr><td>X1</td><td>Port 1</td><td></td><td></td><td></td><td>16376*</td></tr>
<tr><td>X2</td><td>Port 2</td><td></td><td></td><td></td><td>16375*</td></tr>
<tr><td>1</td><td>Standard Telegramm 1</td><td></td><td></td><td></td><td>16374*</td></tr>
<tr><td>1.1</td><td>Parameter Access Point</td><td></td><td></td><td></td><td>16374*</td></tr>
<tr><td>1.2</td><td>Standard Telegramm 1</td><td></td><td>512...515</td><td>512...515</td><td></td></tr>
</table>
Please note that the value has to be entered for the block in a hexadecimal format. The value 16374 corresponds to 3FF6 Hex.

0.0 — SAMPLE_T | QNCONOV
16#3FF6 — DPV1_ID | QMAN_AUT
0 — DPV1_REA | QSIM
</td></tr>
</table>

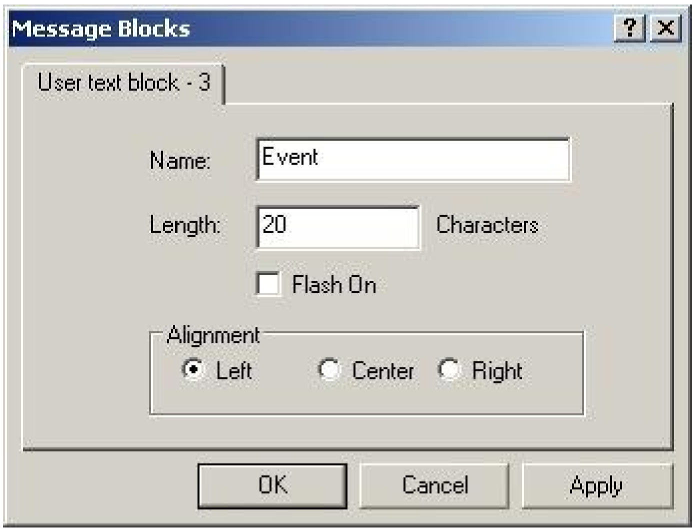
<table>
<tr><th>Step</th><th>Description</th></tr>
<tr><td>4.</td><td>
<b>Configure messages (only with S7-400)</b>
Enter the following messages in the “message configuration” dialog box “Object properties…” > tab: “General” > Special properties: “Messages…”):
<table>
<tr><th>Message ID</th><th>Message text</th><th>Message class</th></tr>
<tr><td colspan="3"><b>Alarm 8P</b></td></tr>
<tr><td>SIG1</td><td>OFF2 active</td><td>PLC Process Control Message - Failure</td></tr>
<tr><td>SIG2</td><td>OFF3 active</td><td>PLC Process Control Message – Failure</td></tr>
<tr><td>SIG3</td><td>Motor overload</td><td>PLC Process Control Message – Failure</td></tr>
<tr><td>SIG4</td><td>Converter overload</td><td>PLC Process Control Message – Failure</td></tr>
<tr><td>SIG5</td><td>Interlock Error</td><td>PLC Process Control Message – Error</td></tr>
<tr><td>SIG7</td><td>External Error</td><td>PLC Process Control Message – Error</td></tr>
<tr><td>SIG8</td><td>General Error</td><td>PLC Process Control Message – Error</td></tr>
<tr><td colspan="3"><b>Notify 8P</b></td></tr>
<tr><td>SIG1</td><td>STOP</td><td>Status Message – PLC</td></tr>
<tr><td>SIG2</td><td>RUN</td><td>Status Message – PLC</td></tr>
<tr><td>SIG3</td><td>LEFT</td><td>Status Message – PLC</td></tr>
<tr><td>SIG4</td><td>RIGHT</td><td>Status Message – PLC</td></tr>
<tr><td>SIG5</td><td>Interlock</td><td>Status Message – PLC</td></tr>
<tr><td>SIG6</td><td>Remote</td><td>Status Message – PLC</td></tr>
<tr><td>SIG7</td><td>Automatic</td><td>Status Message – PLC</td></tr>
<tr><td>SIG8</td><td>Simulation</td><td>Status Message – PLC</td></tr>
</table>
<b>Notes:</b>
<ul>
<li>The texts under “Message texts” are transferred by default to WinCC into “user text block 3”. Leave position 1 under “More>>” empty for message texts. The process tag for “user text block 1” will then be transferred automatically. This is important for the further processing of alarm messages in WinCC.</li>
<li>The following document describes how to use several message texts in the STEP 7-integrated project: http://support.automation.siemens.com/WW/view/en/30550240</li>
<li>The block for the 300-type CPU uses a bit message procedure. The messages are configured in the WinCC Alarm Logging Editor.</li>
</ul>
</td></tr>
<tr><td>5.</td><td><b>Compile and load program</b>
Compile the program and load it into the control system.</td></tr>
<tr><td>6.</td><td><b>Compile WinCC station</b>
Compile the WinCC station.</td></tr>
</table>

4.2 Configuration of the WinCC faceplate

Table 4-2

Step	Description
1.	**Copy files** First, copy all files from the supplied directory "WINCC_GRAPHICS/BST_SINAG120" to the directory “GraCS” and all files from the directory “WINCC_SCRIPTS” to the “library” directory of the WinCC project.
2.	**Configure the start-up parameters** Open the “WinCC-Project” from the “SIMATIC Manager”. Open the Properties of the WinCC station and select the “Startup” tab. Activate the following “Runtimes”: • Global Script Runtime • Tag Logging Runtime • Alarm Logging Runtime • Graphics Runtime
3.	**Register the scripts** Before you can use the provided scripts, they are to be registered. • Open the Global Script C-Editor • Click the “Generate headers” button. The previously copied scripts are now available for your project.
4.	**Create a start-up screen** • Create an empty WinCC picture, name it “Start.pdl” and define this picture as start-up screen. • Open “start.pdl” in the Graphics Designer.

5.	**Insert objects** - Insert an object of type “Smart-objects> Faceplate Instance” in the start picture. Select the file “BST_SINAG120_ICON.FPT”. - Insert an object of type “picture window” in the start picture.
6.	**Configure the properties of the faceplate instance object** Define the following parameters in the Properties dialog box of the faceplate instance object: - Others > Scaling mode = “1:1” Define the following tag assignments in the Properties dialog box of the faceplate instance object: - User-defined2 > QdwState = “<tag prefix > QdwState” [1)] - User-defined2 > QrCapacity = “<tag prefix> QrCapacity” [1)] Configure the following direct connection for “mouse click”: - Source: Constant = “1” - Target: Object in picture = “Picture window1 > Display” Direct Connection Source: ● Constant: 1 ○ Property ○ Tag ● Direct ○ Indirect Object: This object, Bildfenster1, Bildfenster2, Control1, FaceplateInstanz2 Property Target: ○ Current Window ● Object in Picture ○ Tag ● Direct ○ Indirect □ Operator Input Message Object: This object, Bildfenster1, Bildfenster2, Control1, FaceplateInstanz2 Property: Display, Heading, Menu/Toolbar confic, Picture Name, Picture Offset X, Picture Offset Y, Position X, Position Y, Scaling Factor, Scroll bar position X, Scroll bar position Y, Server prefix, Tag prefix, Window Height [1)] The tag prefix consists of program name, CFC chart and block name, followed by a dot.

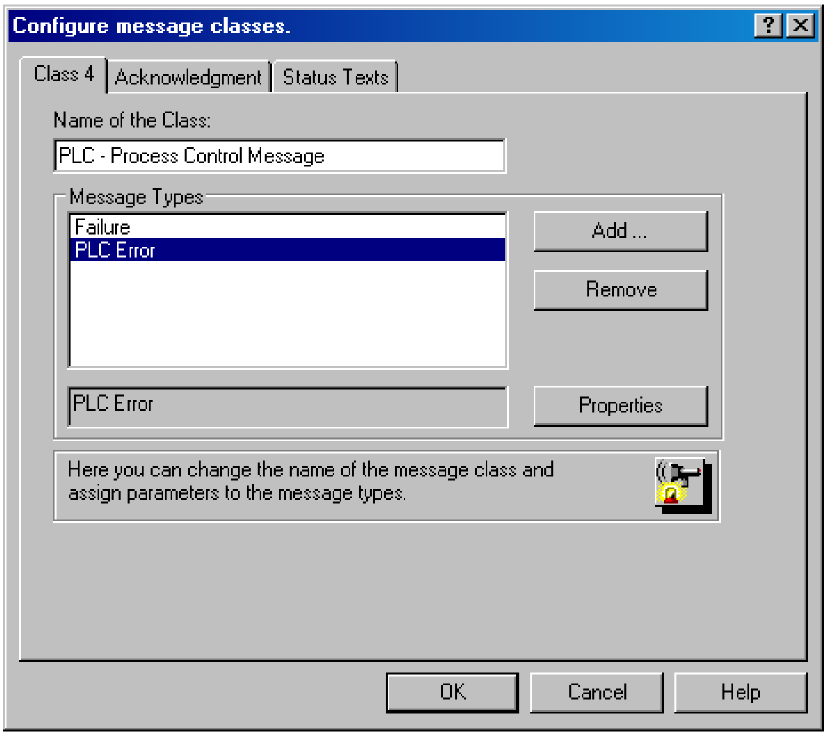
7.	**Configure the properties of the picture window object** Define the following parameters in the Properties dialog box of the picture window object: • Others > Picture name = "BST_SINAG120_MAIN.pdl" • Others > Tag prefix = "<tag prefix>" [2)] • Others > Display = "No" • Others > Moveable = "Yes" • Others > Frame = "Yes" • Others > Adjust window = "Yes" [2)] Same tag prefix as in step 5. This prefix will also be entered via script in the "MsgFilterSQL" property of the Alarm Control when the message window is called. This ensures that only messages of the relevant block will be displayed.
8.	**Create a user text block** For the output of messages you have configured for the block it is necessary to create a user text block and to define the relevant message class with the corresponding message type in WinCC. • Open the Alarm Logging Editor and add User text block no. 3. Define a reasonable name.  Message Blocks User text block - 3 Name: Event Length: 20 Characters ☐ Flash On Alignment ● Left ○ Center ○ Right **Note:** If you create your project with the OS project editor, this step is dropped, since this editor automatically creates the text blocks, message classes and message types.

9.

Create message classes and define message types

Configure the following message classes and message types:

- Message class 4 with message type 55 (corresponds to STEP 7 message class: “PLC process control message – Failure”).
- Message class 4 with message type 56 (corresponds to STEP 7 message class: “Process control message – Error”).
- Message class 16 with message type 253 (corresponds to STEP 7 message class: “PLC status message – without acknowledgement”).



Configure message classes.

Name of the Class: PLC - Process Control Message

Message Types: Failure, PLC Error

PLC Error

Here you can change the name of the message class and assign parameters to the message types.

Assign reasonable names for all message classes and types.

- For message class 4 you activate the selection box “Acknowledgement coming” in the “Acknowledgement” tab

Notes:

- The following document provides an overview of the WinCC message classes and the corresponding STEP 7 message classes: http://support.automation.siemens.com/WW/view/en/31622971
- The following document describes how to use several message texts in the STEP 7-integrated project: http://support.automation.siemens.com/WW/view/en/30550240

10. **Configure messages (bit message procedure – 300 CPU only)**

Create the following messages using the Alarm Logging Editor. Select the message class and type you have created in step 8:

Message 1:

- Class: PLC – Process Control Message (class: 4)
- Type: PLC Failure (type: 55)
- Message tag: <tag prefix>QwAlarm
- Message bit: 0
- Message text: <tag prefix>
- Event: OFF 2

Message 2:

- Class: PLC – Process Control Message (class: 4)
- Type: PLC Failure (type: 55)
- Message tag: <tag prefix>QwAlarm
- Message bit: 1
- Message text: <tag prefix>
- Event: OFF 3

Message 3:

- Class: PLC – Process Control Message (class: 4)
- Type: PLC Failure (type: 55)
- Message tag: <tag prefix>QwAlarm
- Message bit: 2
- Message text: <tag prefix>
- Event: Motor overload

Message 4:

- Class: PLC – Process Control Message (class: 4)
- Type: PLC Failure (type: 55)
- Message tag: <tag prefix>QwAlarm
- Message bit: 3
- Message text: <tag prefix>
- Event: Converter overload

Message 5:

- Class: PLC – Process Control Message (class: 4)
- Type: PLC Error (type: 56)
- Message tag: <tag prefix>QwAlarm
- Message bit: 4
- Message text: <tag prefix>
- Event: Interlock Error

Message 6:

- Class: PLC – Process Control Message (Class 4)
- Type: PLC Error (type 56)
- Message tag: <tag prefix>QwAlarm
- Message bit: 6
- Message text: <tag prefix>

	• Event: External Error Message 7: • Class: PLC – Process Control Message (Class 4) • Type: PLC Error (type 56) • Message tag: <tag prefix>QwAlarm • Message bit: 7 • Message text: <tag prefix> • Event: General Error Message 8: • Class: PLC – Status Message (class 16) • Type: Status PLC (type 253) • Message tag: <tag prefix>QwAlarm • Message bit: 8 • Message text: <tag prefix> • Event: STOP Message 9: • Class: PLC – Status Message (class 16) • Type: Status PLC (type 253) • Message tag: <tag prefix>QwAlarm • Message bit: 9 • Message text: <tag prefix> • Event: START Message 10: • Class: PLC – Status Message (class 16) • Type: Status PLC (type 253) • Message tag: <tag prefix>QwAlarm • Message bit: 10 • Message text: <tag prefix> • Event: LEFT Message 11: • Class: PLC – Status Message (class 16) • Type: Status PLC (type 253) • Message tag: <tag prefix>QwAlarm • Message bit: 11 • Message text: <tag prefix> • Event: RIGHT Message 12: • Class: PLC – Status Message (class 16) • Type: Status PLC (type 253) • Message tag: <tag prefix>QwAlarm • Message bit: 12 • Message text: <tag prefix> • Event: Interlock Message 13:

	• Class: PLC – Status Message (class 16) • Type: Status PLC (type 253) • Message tag: <tag prefix>QwAlarm • Message bit: 13 • Message text: <tag prefix> • Event: Remote Message 14: • Class: PLC – Status Message (class 16) • Type: Status PLC (type 253) • Message tag: <tag prefix>QwAlarm • Message bit: 14 • Message text: <tag prefix> • Event: Automatic Message 15: • Class: PLC – Status Message (class 16) • Type: Status PLC (type 253) • Message tag: <tag prefix>QwAlarm • Message bit: 15 • Message text: <tag prefix> • Event: Simulation **Note:** Create the messages for each BST_SINAG120_300 block in your S7 program.

Appendix

5 Block Connectors

5.1 Input parameters

Table 5-1

Parameter	Data type	Default setting	Description
dwInp	DWORD	16#0	Status word SINAMICS G120
LOCK	BOOL	FALSE	Interlock
ERR_EXTERN	BOOL	FALSE	External error
LIOP_SEL	BOOL	FALSE	Interconnection active
L_AUT	BOOL	FALSE	0 = Manual 1 = Automatic
L_REMOTE	BOOL	TRUE	0 = Local 1 = Remote (G120 Cmd-Bit [15])
L_SIM	BOOL	FALSE	0 = Process 1 = Simulation
L_RESET	BOOL	FALSE	Error reset
L_ON	BOOL	FALSE	0 = OFF 1 = ON (G120 Cmd-Bit [0])
L_REVERSE	BOOL	FALSE	1 = Setpoint inversion (G120 Cmd-Bit [11])
L_ENABLE	BOOL	TRUE	1 = Enable operation (G120 Cmd Bit [3])
L_RFG_EN	BOOL	TRUE	1 = RFG condition (G120 Cmd Bit [4])
L_RFG_FREE	BOOL	TRUE	1 = RFG enable (G120 Cmd Bit [5])
L_SP_EN	BOOL	TRUE	1 = Setpoint enable (G120 Cmd Bit [6])
L_SP_VALID	BOOL	TRUE	1 = Setpoint valid (G120 Cmd-Bit [10])
OFF2	BOOL	TRUE	0 = OFF2 (pulse inhibit) (G120 Cmd-Bit [1])
OFF3	BOOL	TRUE	0 = OFF3 (fast stop) (G120 Cmd-Bit [2])
Auto_ON	BOOL	FALSE	1 = ON
Auto_REV	BOOL	FALSE	1 = Change rotation direction
SP_Auto	WORD	16#0	Setpoint specification I/O
SP_Man	REAL	0.0	Setpoint specification, manual
SP_Sim	REAL	0.0	Setpoint specification, simulation
SIM_nomFreq	REAL	0.0	Rated motor frequency (simulation)
SIM_nomSpeed	REAL	0.0	Rated motor speed (simulation)
SAMPLE_T	REAL	0.0	Sampling time
DPV1_ID	WORD	16#0	DP slave, diagnosis address
DPV1_READ	BOOL	FALSE	Acyclic data request
OP_dwCmd	DWORD	16#0	Command word WinCC

5.2 Output parameters

Table 5-2

Parameter	Data type	Default setting	Description
QdwState	DWORD	16#0	Status tag WinCC
QdwCmd	DWORD	16#0	Command word SINAMICS G120
QFreqPeri	WORD	16#0	Actual value, I/O
QSPCapacity	REAL	0.0	Setpoint in percent
QrCapacity	REAL	0.0	Actual value in percent
QSPFrequ	REAL	0.0	Setpoint, frequency
QrCuFreq	REAL	0.0	Actual value, frequency
QSPSpeed	REAL	0.0	Setpoint, rotation speed
QrCuSpeed	REAL	0.0	Actual value, rotation speed
QPOWER_ON	BOOL	FALSE	1=Stop (ready to run) (G120 State Bit[0])
QREADY_RUN	BOOL	FALSE	1=Ready to run (G120 State Bit[1])
QOP_ENABLE	BOOL	FALSE	1=Run (operation enabled) (G120 State Bit[2])
QFAULT	BOOL	FALSE	1=Fault (G120 State Bit[3])
QNOFF2	BOOL	FALSE	0=OFF2 Command (G120 State Bit[4])
QNOFF3	BOOL	FALSE	0=OFF3 Command (G120 State Bit[5])
QSLOCK	BOOL	FALSE	1=Activation block (G120 State Bit[6])
QWARN	BOOL	FALSE	1=Warning active (G120 State Bit[7])
QSPREACH	BOOL	FALSE	1=No setpoint/actual deviation (G120 State Bit[8])
QFRREACH	BOOL	FALSE	1=max. converter frequency reached (G120 State Bit[10])
QNMOTWARN	BOOL	FALSE	0=Motor power limit reached (G120 State Bit[11])
QBRAKE	BOOL	FALSE	0=Motor holding brake (G120 State Bit[12])
QNMOTOV	BOOL	FALSE	0=Motor overload (G120 State Bit[13])
QDIRECTION	BOOL	FALSE	0=anticlockwise 1=clockwise rotation (G120 State Bit[14])
QNCONOV	BOOL	FALSE	0=Converter overload (G120 State Bit[15])
QMAN_AUT	BOOL	FALSE	0=Manual 1=Automatic
QREMOTE	BOOL	FALSE	1=Control requested (G120 State Bit[9])
QSIM	BOOL	FALSE	0=Process 1=Simulation
QLOCK	BOOL	FALSE	1=Interlock error

Parameter	Data type	Default setting	Description
QERR	BOOL	FALSE	1=General error active
QERR_EXT	BOOL	FALSE	1=External error
DPV1_RW_ERR	BOOL	FALSE	DPV1 - Read or write error (SFB52 or SFB53)

6 Links & Literature

6.1 Bibliography

This list is not complete and only represents a selection of relevant literature.

Table 6-1

	Topic	Title
/1/	STEP7 SIMATIC S7-300/400	Automating with STEP7 in STL and SCL Author: Hans Berger Publicis Publishing ISBN: 978-3895784125

6.2 Internet links

The following list is by no means complete and only provides a selection of appropriate sources.

Table 6-2

	Topic	Title
\1\	Reference to this entry	http://support.automation.siemens.com/WW/view/en/68679830
\2\	Siemens Industry Online Support	http://support.automation.siemens.com
\3\	Reference to documentation on frequency converter SINAMICS G120	SINAMICS G120 Function Manual http://support.automation.siemens.com/WW/view/en/31676845 Control Units CU240E / CU240S List of Parameters: http://support.automation.siemens.com/WW/view/en/32465038 Manual: Control Units CU240 http://support.automation.siemens.com/WW/view/en/27864729 Reading and writing of acyclic data http://support.automation.siemens.com/WW/view/en/29157692
\4\	Reference to this entry	Which PROFINET nodes support the extended PN diagnostics and what do you have to configure? http://support.automation.siemens.com/WW/view/en/23678970
\5\	Reference to this entry	How are message classes used if WinCC is integrated in the STEP 7 project? http://support.automation.siemens.com/WW/view/en/31622971
\6\	Reference to this entry	How are message texts used if WinCC is integrated in the STEP 7 project? http://support.automation.siemens.com/WW/view/en/30550240
\7\	SENTRON PAC faceplates for WinCC Runtime Professional	http://support.automation.siemens.com/WW/view/en/67318600
\8\	SITOP UPS 1600: Faceplates and STEP 7 Communication Blocks	http://support.automation.siemens.com/WW/view/en/85668144

	Topic	Title
\9\	Measuring and Visualizing Energy Data	http://support.automation.siemens.com/WW/view/en/86299299
\10\	Configuration instruction for creating faceplates for Comfort Panels and WinCC Runtime Advanced	http://support.automation.siemens.com/WW/view/en/68014632

6.3 Further Example Blocks

The following list is by no means complete and only provides a selection of appropriate sources.

Tabelle 6-3

	Topic	Title
\1\	Example Blocks for WinCC (TIA Portal) and STEP 7 (TIA Portal)	http://support.automation.siemens.com/WW/view/en/66839614
\2\	Example Blocks for WinCC V7 and STEP 7 (TIA Portal)	http://support.automation.siemens.com/WW/view/en/31624179
\3\	Example Blocks for WinCC flexible and STEP 7 V5	http://support.automation.siemens.com/WW/view/en/36435784

7 History

Table 7-1 History

Version	Date	Modifications
V1.0	27.04.2009	First issue
V2.0	14.09.2009	Reworking the block and adaptation of documentation
V2.0.1	29.04.2010	Changing some typing errors