engineering
mannesmann
Rexroth

SPS
Function Blocks

Application Manual

DOK-CONTRL-SPS*FUN*V16-ANW1-EN-P

274536

Indramat

Title SPS Function Blocks

Type of document Application Manual and Functional Description

Documentation type DOK-CONTRL-SPS*FUN*V16-ANW1-EN-P

Internal filing
- Folder 3 / Register 6
- Drawing number 109-0668-4192-EN/04.97

Purpose of the document Informing the user of the utilization and the setting of the parameter values of the function blocks.

Configuration control

Documentation identification of previous releases	Release date	Comment
109-0668-4192-00	11.93	New issue
109-0668-4192-01	04.97	Revision

Publisher INDRAMAT GmbH • Bgm.-Dr.-Nebel-Str. 2 • D-97816 Lohr a. Main

Phone 09352/40-0 • Tx 689421 • Fax 09352/40-4885

Dept. ENC (PG)

Contents

1 General

1.1 Transfer Parameters of Function Blocks (FB) and Functions (FK)

The following table shows the data types that can be processed by FBs or FKs.

Data type	Declaration	Bit length	Range	Comment
Boolean	BOOL	1	0,1	
Byte	BYTE	8	0h - FFh	
Word	WORD	16	0h - FFFFh	
Double word	DWORD	32	0h - FFFFFFFFh	
Time value	TIME	30	0 ms - 24 days	
Short Integer	SINT	8	-128 - +127	
Integer	INT	16	-32768 - +32767	
Unsigned Short Integer	USINT	8	0- 255	
Unsigned Integer	UINT	16	0 - 65535	
Character	CHAR	8		
Character string	STRING		max. 255 chars.	
User types	TYP name			cannot be used in functions
Structured process signals from SPS to CNC	qPROC			cannot be used in functions
Structured process signals from CNC to SPS	iPROC			cannot be used in functions
Structured mechanism signals from SPS to CNC	qMECH			cannot be used in functions
Structured mechanism signals from CNC to SPS	iMECH			cannot be used in functions
Structured axis signals from SPS to CNC	qAXIS			cannot be used in functions
Structured axis signals from CNC to SPS	iAXIS			cannot be used in functions
Structured digital outputs	qDEA			cannot be used in functions
Structured digital inputs	iDEA			cannot be used in functions
Structure for interface initialization	COM			cannot be used in functions

1.2 List of Function Blocks and Functions

Function blocks

FB name		Application
Stan-dard		
*	HIBIT_xx	Outputting the bit number of the most significant '1' bit in the byte
*	OVGY0_xx	Override setting via GRAY switches (TYP 1)
*	OVGY1_xx	Override setting via GRAY switches (TYP 2)
*	OVKEY_xx	Override setting via keyboard
*	OVPOT_xx	Override setting via potentiometer
*	OPMOD_xx	Selecting the NC modes
*	RNMOD_xx	Selecting the NC program execution modes
*	JGMOD_xx	Selecting the NC jog modes
*	SOTOP_xx	Initializing and opening the SOT interface
*	POWER_xx	Power activation
*	PRGST_xx	NC program start
*	AXIS_xx	Axis movement
*	MU4<yy>_xx	4-to-1 multiplexer; available for different data types
*	ERRES_xx	NC error/reset handling
*	SEL4_xx	Selection of four different outputs as a function of four inputs
*	STOPP_xx	Stop-watch

1.3 Procedure of Linking the Function Blocks

The function blocks that are marked as "standard" are available after the user interface has been installed..

Function blocks that are not marked as "standard" may be ordered from INDRAMAT. Please specify the corresponding software type identification in your order.

A function block that shall be used in a SPS program must be transferred to the related plant directory. Use the menu item 'File', 'File Manager' 'FETCH' for this purpose.

Afterwards, those function blocks must be handled like user-created ones. This means that the function blocks must be imported and declared in the program organization unit in which they are to be used.

Function blocks are encoded when they are stored. Thus 'zooming' into such a function block, or printing or modifying its program code is not possible.

2 Function Block Data Sheet: HIBIT_xx

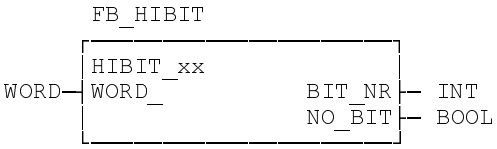
2.1 Description

The 'HIBIT_xx' function block may be used for determining the number of the most significant bit position that is logic '1' in the transferred input word.

As a result, the number of the bit position is output as an INTEGER value.

The number of the bit position is only valid if the output NO_BIT = 0. No bit is set if this is not true.

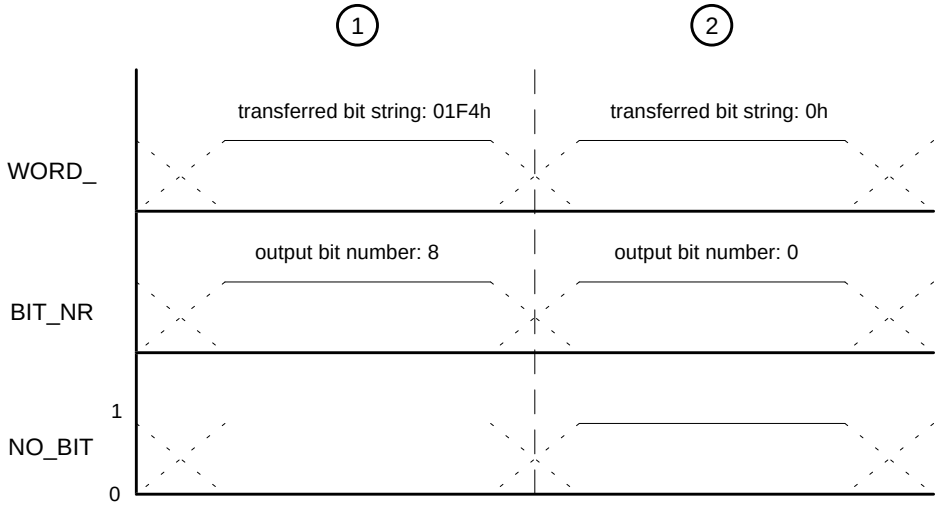
2.2 Interface Overview



2.3 Description of the Identifiers

Identifier	Application	Type
	Input variable	
WORD_	Input value whose contents is verified.	WORD
	Output variables	
BIT_NR	Number of the most significant '1' bit position.	INT
NO_BIT	Flag if no bit of the transferred bit string has been set to logic '1'. 0: Input word > 0000h 1: No bit in the bit string set to logic '1'	BOOL

2.4 Output Diagram



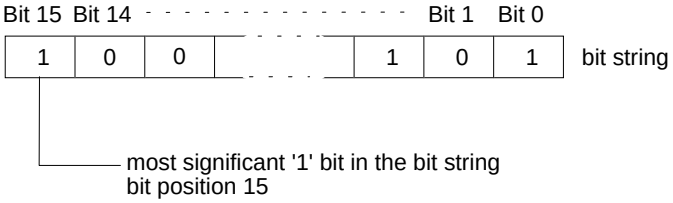
①: The most significant '1' bit position in the transferred bit string is 8.
The transferred bit string is greater than 0h. The 'NO_BIT' output is therefore set to 0.

②: The transferred bit string does not contain a bit position that is '1'.
To distinguish from the state that bit position 0 is the most significant '1' bit position, the 'NO_BIT' output is set to 1.

2.5 Application

Example: Determining the highest priority within a message register

Requirements:
Determining the message priority via offset and bit number. Message output shall start from number 585. This means that messages can be output in the range from 585 through 600; depending on the bit in the transferred register that has the highest '1' significance.



Identifier names that are used in the SPS:

MELDREG	Register that contains the message output error bits
MELD_IDX	Number of the most significant bit in the message register; is used as an index for message number output.
NO_MELD	Enabling input for message output. A message is not output if no bit in the message register is '1'.
MELD_NR	Message number; computed from offset plus index.
MELD_OK	The message has been written

Setting the parameter values of the function block:

```
(*INTERROGATION OF THE MOST SIGNIFICANT BIT IN THE MESSAGE REGISTER ----------------*)
          FB_HIBIT
          ┌──────────────────┐
          │HIBIT_xx          │
  MELDREG─┤WORD_       BIT_NR├─MELD_IDX
          │            NO_BIT├─NO_MELD
          └──────────────────┘

(*COMPUTING THE MESSAGE NUMBER OUTPUT OF THE MESSAGES FROM NO. 585 ONWARDS ------*)
          ┌────────┐
          │+       │
       585─┤        │
 MELD_IDX─┤        ├─MELD_NR
          └────────┘

(*OUTPUTTING THE MESSAGE OF THE HIGHEST PRIORITY --------------------------*)
(*                                                                        *)
(*PRIORITY SEQUENCE:                                                     *)
(*                                                                        *)
(* MESSAGE NUMBER   | MESSAGE                                            *)
(*  ────────────────┼──────────────────────────────────────────────────  *)
(*       600         | MOTOR PROTECTION SWITCH HYDRAULIC TRIGGERED        *)
(*       599         | PROTECTIVE DEVICE BYPASSED                        *)
(*       598         | RELEASE CLUTCH ACTIVATED DURING OPERATION         *)
(*       597         | CHUCK RELEASED                                    *)
(*       596         | AIR PRESSURE NOT AVAILABLE                        *)
(*       595         | COOLING NOT AVAILABLE N                           *)
(*       594         | GRIPPER RELEASED                                  *)
(*       593         | GRIPPER PRESSURE SWITCH TRIPPED                   *)
(*       592         | MINIMUM HYDRAULIC OIL LEVEL                       *)
(*       591         | MINIMUM LUBRICATION OIL LEVEL                     *)
(*       590         | CHIP CONVEYOR OPERATION MONITORING                *)
(*       589         | OVERTEMPERATURE AT DRIVE                          *)
(*       588         | ERROR MONITORING MEASURING PROBE                  *)
(*       587         | PALETTE FEEDER FLAP CLOSED                        *)
(*       586         | CONVEYOR BELT SHUT DOWN                           *)
(*       585         | NO PARTS AVAILABLE                                *)

          ┌────────┐
          │MSG_WR  │
  NO_MELD─OWRITE   │
        0─┤PROC    │
  MELD_NR─┤NR      ├─MELD_OK
          └────────┘
```

Message number 600 and the message

`'Motor protection switch hydraulic triggered',`

would be output if the bit string shown above were in the message register.

The message is computed from the most significant '1' bit in the message register. In this case: Bit number 15 plus the message offset of 585 yield the message number 600.

3 Function Block Data Sheet: OVGY0_xx

3.1 Description

The 'OVGY0_xx' function block interprets the override value that has been selected via a Gray code switch.

One byte is used for transferring the four-bit-encoded Gray code value to the function block.

Within the function block, the bits of the bytes are moved such that the Gray code saved from the switch is in the lower nibble. The number of digits by which the byte must be moved is allocated to the corresponding input parameter of the function block.

The allocation of an override value to the related Gray code can be programmed as required.

After an additional key has been pressed, the override value shall be set to 0%, 100% or to the selected switch position, without being influenced by the position of the Gray code switch.

If none of those inputs is connected, the switch position will always be interpreted.

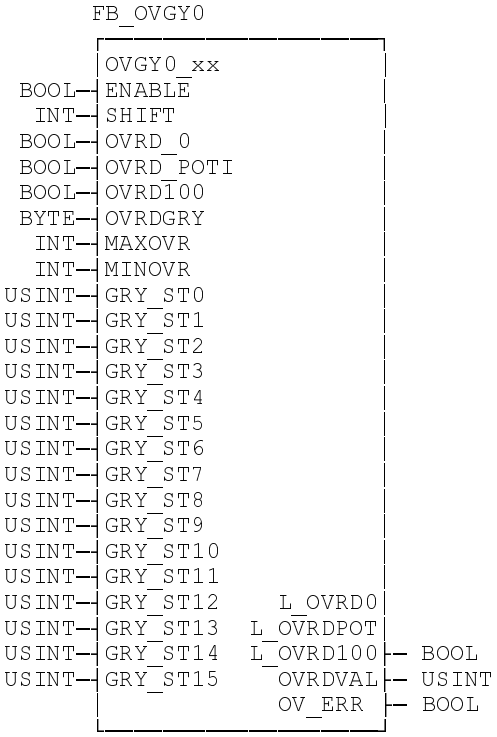
The function block has two parameter that permit the override value to be delimited to a minimum or maximum value. If the override value exceeds or falls below those values, the override output of the Gray code switch will be delimited to the respective limit value. The limitation does not have an influence when a 0% or 100% key is selected.

Removing the processing enabling signal permits the override value to be frozen at its last selected value, irrespective of any changes of the Gray switch during that time. The currently selected value will be output when the processing enabling signal is re-allocated.

Note: If the BTM 03/04 Gray code switch (VA-VB-VE-SA modules) is used, an additional bit (bit number 5 in the byte) is provided for checking for an overflow. If the bit is set, the override value at the function block is set to 0%, and the error output is set to logic '1'.

This functionality is available from Version 04.15/xx or 05.15/xx onwards.

3.2 Interface Overview



3.3 Description of the Identifiers

Identifier	Application	Type
	Input variables	
ENABLE	Block enabling 0: No interpretation of the inputs. Outputs retain their states. 1: Block processing is enabled; i.e. the inputs of the FB are interpreted.	BOOL
SHIFT	Number of digits by which the byte that contains the Gray code must be moved in order to permit the Gray code to be in the lower nibble of the input byte.	INT
OVRD_0	Input of the 0% override value pre-selection.	BOOL
OVRD_POTI	Select input for override pre-selection value via switch. A positive edge at the input derives the override value from the current Gray code switch setting.	BOOL
OVRD100	Input of the 100% override value A positive edge at the input sets the override to 100%.	BOOL
OVRDGRY	Byte that contains the Gray code that is to be interpreted.	BYTE
MAXOVR	Upper limit of override value.	INT
MINOVR	Lower limit of override value.	INT
GRY_ST0	Override value of Gray code 0	USINT
GRY_ST1	Override value of Gray code 1	USINT
GRY_ST2	Override value of Gray code 2	USINT
GRY_ST3	Override value of Gray code 3	USINT
GRY_ST4	Override value of Gray code 4	USINT
GRY_ST5	Override value of Gray code 5	USINT
GRY_ST6	Override value of Gray code 6	USINT
GRY_ST7	Override value of Gray code 7	USINT
GRY_ST8	Override value of Gray code 8	USINT

GRY_ST9	Override value of Gray code 9	USINT
GRY_ST10	Override value of Gray code 10	USINT
GRY_ST11	Override value of Gray code 11	USINT
GRY_ST12	Override value of Gray code 12	USINT
GRY_ST13	Override value of Gray code 13	USINT
GRY_ST14	Override value of Gray code 14	USINT
GRY_ST15	Override value of Gray code 15	USINT
Output variables		
L_OVRD0	Indication - 0% override value selection is active 0: selection is not active 1: selection is active	BOOL
L_OVRDPOT	Indication - override value is selected via Gray switch 0: selection is not active 1: selection is active	BOOL
L_OVRD100	Indication - 100% override value selection is active 0: selection is not active 1: selection is active	BOOL
OVRDVAL	Converted override value	USINT
OV_ERR	Overflow at the Gray code switch 0: no overflow 1: overflow; override value is set to 0%	BOOL

3.4 Application

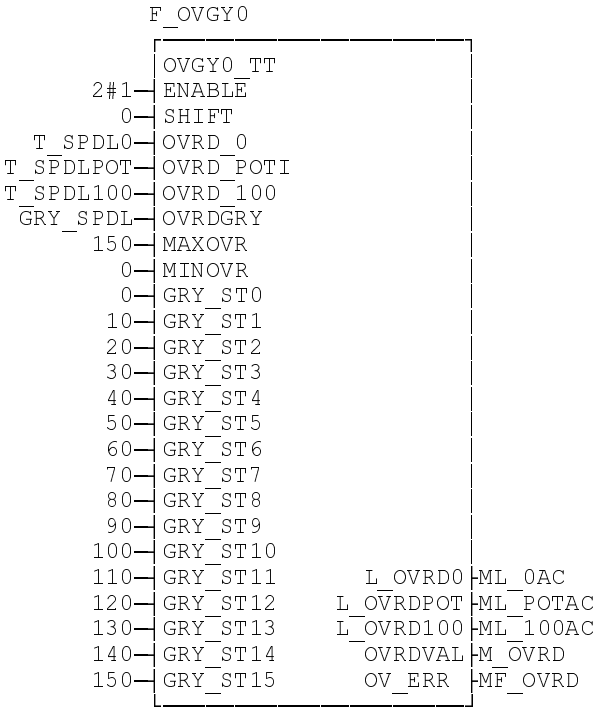
Example: Use Gray code switch for allocating spindle override.

Feed override via Gray code switch with the following steps:

Step 0: 0%	Step 4: 40%	Step 8: 80%	Step 12: 120%
Step 1: 10%	Step 5 50%	Step 9: 90%	Step 13: 130%
Step 2: 20%	Step 6: 60%	Step 10: 100%	Step 14: 140%
Step 3: 30%	Step 7: 70%	Step 11: 110%	Step 15: 150%

In addition to the Gray code switch, three other keys are used for selecting between 0%, 100%, and Gray code switch setting.

Setting the function block parameters:



4 Function Block Data Sheet: OVGY1_xx

4.1 Description

The 'OVGY1_xx' function block interprets the override value that has been selected via a Gray code switch.

One byte is used for transferring the four-bit-encoded Gray code value to the function block.

Within the function block, the bits of the bytes are moved such that the Gray code from the switch is in the lower nibble. The number of digits by which the byte must be moved is allocated to the corresponding input parameter of the function block.

The allocation of an override value to the related Gray code is subjected to invariable increments. This means that the difference between the override values of Gray code 0 and Gray code 1 is the same as the difference between the override values of Gray code 4 and Gray code 5. The increment value, however, may be allocated to an input parameter.

After an additional key has been pressed, the override value shall be set to 0%, 100% or to the selected switch position, without being influenced by the position of the Gray code switch.

If none of those inputs is connected, the switch position will always be interpreted.

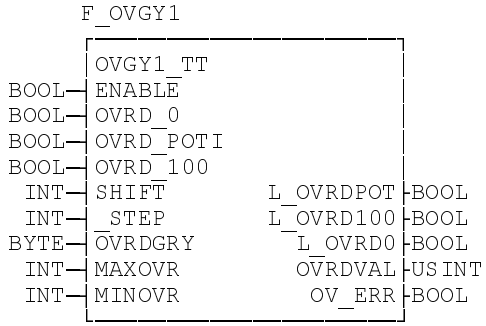
The function block has two parameter that permit the override value to be delimited to a minimum or maximum value. If the override value exceeds or falls below those values, the override output of the Gray code switch will be delimited to the respective limit value. The limitation does not have an influence when a 0% or 100% key is selected.

Removing the processing enabling signal permits the override value to be frozen at its last selected value, irrespective of any changes of the Gray switch during that time. The currently selected value will be output when the processing enabling signal is re-allocated.

Note: If the BTM 03/04 Gray code switch (VA-VB-VE-SA modules) is used, an additional bit (bit number 5 in the byte) is provided for checking for an overflow. If the bit is set, the override value at the function block is set to 0%, and the error output is set to logic '1'.

This functionality is available from Version 04.15/xx or 05.15/xx onwards.

4.2 Interface Overview



4.3 Description of the identifiers

Identifier	Application	Type
	Input variables	
ENABLE	Block enabling 0: No interpretation of the inputs. Outputs retain their states. 1: Block processing	BOOL
OVRD_0	Input of the 0% override value pre-selection. A positive edge at the input sets the override value to 0%.	BOOL
OVRD_POTI	Select input for override pre-selection value via switch. A positive edge at the input derives the override value from the current Gray code switch setting.	BOOL
OVRD_100	Input of the direct 100% override value pre-selection. A positive edge at the input sets the override to 100%.	BOOL
SHIFT	Number of digits by which the byte that contains the Gray code must be moved.	INT
_STEP	Increment between the individual override values that must be allocated to a Gray code.	INT
OVRDGRY	Byte that contains the Gray code that is to be interpreted.	BYTE
MAXOVR	Upper limit of override value.	INT
MINOVR	Lower limit of override value.	INT
	Output variables	
L_OVRD0	Indication - 0% override value selection is active 0: selection is not active 1: selection is active	BOOL
L_OVRDPOT	Indication - override value is selected via Gray switch 0: selection is not active 1: selection is active	BOOL
L_OVRD100	Indication - 100% override value selection is active 0: selection is not active 1: selection is active	BOOL
OVRDVAL	Converted override value	USINT
OV_ERR	Overflow at the Gray code switch 0: no overflow 1: overflow; override value is set to 0%	BOOL

4.4 Application

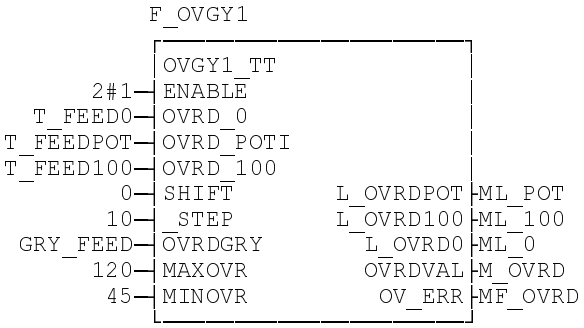
Example: Use Gray code switch for allocating feed override.

Feed override evaluation via Gray code switches with a 10%-increment between the individual Gray code positions.

The Gray code (4 bits) is located in the low nibble of the transferred input byte. The function block is continually processed.

In addition, there are three other keys for selecting between 0%, 100%, and Gray code switch setting.

Setting the function block parameters:



Gray code	0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001	1000
Override	0	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150

5 Function Block Data Sheet: OVKEY_xx

5.1 Description

The 'OVKEY_xx' function block can be used for selecting the override value via keys. Four keys are provided for selection. There is one key for directly selecting the 0% override value, and one for the 100% override value. Furthermore, an increment key permits the override value to be changed at a fixed incremental rate. Another key must be provided for decrementing the value at the same rate.

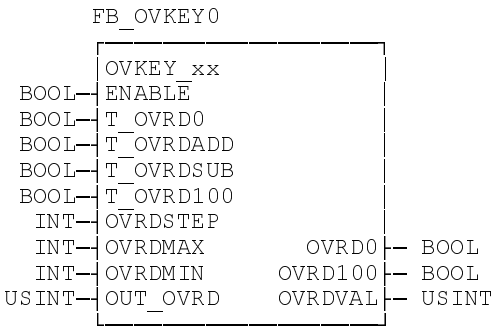
The step size value required for incrementing and decrementing is entered into the function block via an input parameter.

To delimit the override value, it is checked for remaining inside a minimum and a maximum limit. The value is adjusted accordingly if it lies outside the checked limit.

To signal the override values of 0% and 100%, there are output signals that become active (logic '1') when the related override value is reached.

Removing the processing enabling signal enables changes of the input to be ignored. This means that the output remains at its last state. The outputs will be adjusted to the current input signals when the block enabling signal is re-applied.

5.2 Interface Overview



5.3 Description of the Identifiers

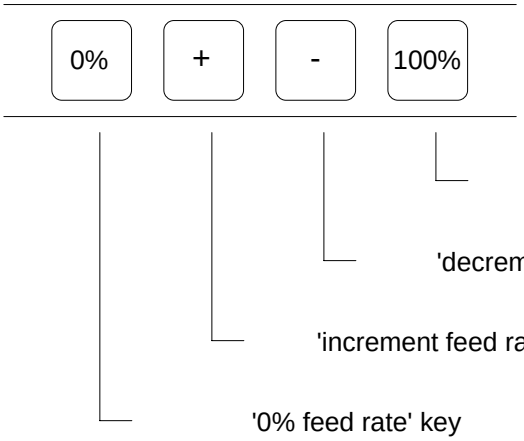
Identifier	Application	Type
	Input variables	
ENABLE	Block enabling 0: No interpretation of the inputs. Outputs retain their states. 1: Block processing	BOOL
T_OVRD0	A positive edge at the input sets the override value to 0%.	BOOL
T_OVRDADD	A positive edge at the input increments the override value.	BOOL
T_OVRDSUB	A positive edge at the input decrements the override value.	BOOL
T_OVRD100	A positive edge at the input sets the override value to 100%.	BOOL
OVRDSTEP	Step size used for decrementing and incrementing.	INT
OVRDMAX	Upper limit of override value.	INT
OVRDMIN	Lower limit of override value.	INT
OUT_OVRD	Returning the override value	USINT
	Output variables	
OVRD0	Indication - 0% override value 0: override value is not equal to 0% 1: override value is equal to 0%	BOOL
OVRD100	Indication - 100% override value 0: override value is not equal to 100% 1: override value is equal to 100%	BOOL
OVRDVAL	Output of the selected override value	USINT

5.4 Application

Example: Setting the feed override via an operator keyboard.

Select the feed override via the keyboard. The step size for incremental selection should be 10%. The override value should be delimited to a range between 0% and 150%. The function block is continually processed.

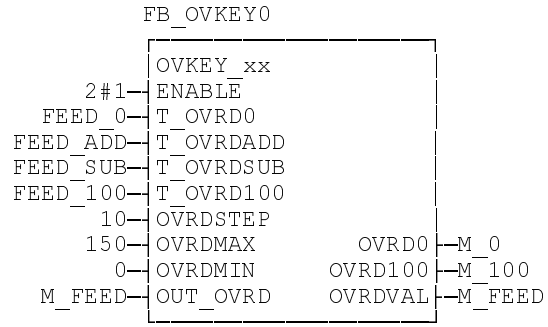
key pad



dentifier names that are employed in the SPS:

FEED_0	Set feed rate to 0%
FEED_ADD	Increment feed rate
FEED_SUB	Decrement feed rate
FEED_100	Set feed rate to 100%
M_0	Flag for 0% feed rate
M_100	Flag for 100% feed rate
M_FEED	Flag for feed override

Setting the function block parameters:



6 Function Block Data Sheet: OVPOT_xx

6.1 Description

The 'OVPOT_xx' function block implements override interpretation via a potentiometer. There is an additional function block input for selecting between potentiometer setting and a direct 100% override rate.

To set the override value to 100% upon the SPS program start, a preset input at the function block must be connected. Depending on the signal applied to that input, the override output will correspond to the potentiometer setting or to a fixed 100% value.

The 100% override selection via the related key is indicated by a corresponding function block output.

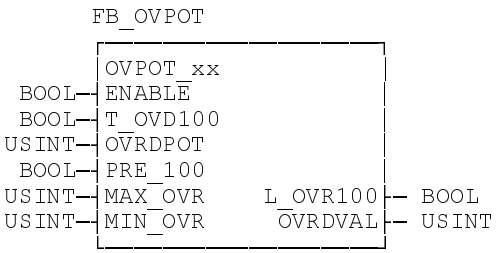
To delimit the override value, it is checked for remaining inside a minimum and a maximum limit. The value is adjusted to the limit values transferred to the function block if it lies outside the checked limit.

There is an output that permits the activation of the 100% value via the key to be indicated.

Removing the processing enabling signal enables changes of the input to be ignored. This means that the output remains at its last state. The outputs will be adjusted to the current input signals when the block enabling signal is re-applied.

Note: The preset input must have logic '0' assigned if the additional input for toggling between 100% and the potentiometer value has not been connected.

6.2 Interface Overview



6.3 Description of the identifiers

Identifier	Application	Type
	Input variables	
ENABLE	Processing enabled 0: No interpretation of the inputs. Outputs retain their states. 1: Processing enabled	BOOL
T_OVD100	A positive edge at the input toggles between 100% and the potentiometer setting.	BOOL
OVRDPOT	Read potentiometer value	USINT
PRE_100	Preset input 0: Override value pre-selection corresponds to current potentiometer value 1: Preset value is 100%	BOOL
MAX_OVR	Upper limit of override value.	USINT
MIN_OVR	Lower limit of override value.	USINT
	Output variables	
L_OVR100	Indication - 100% override selected via key 0: Override value via potentiometer 1: 100% override value activated via key	BOOL
OVRDVAL	Output of the selected override value.	USINT

6.4 Application

Example: Selecting feed override rate via potentiometer.

Select the feed override rate via a potentiometer. The potentiometer range is between 0 and 255. The output range of the override value shall be between 0 and 150.

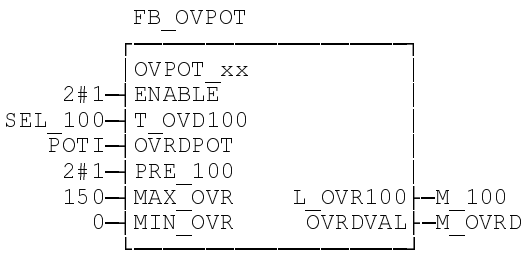
An additional key enables the override to be set to 100%, irrespective of the current potentiometer setting.

An override value of 100% shall be output as preset value. The function block is continually processed.

Identifier names that are employed in the SPS:

SEL_100	100% override pre-selection key
POTI	Override potentiometer
M_100	Flag for 100% override
M_OVRD	Flag for the selected override value

Setting the function block parameters:



7 Function Block Data Sheet: OPMOD_xx

7.1 Description

To be able to select one of the four possible NC modes, there are two interface signals (PxxCMODE0 and PxxCMODE1) available at the interface between SPS and CNC. These signals are used for encoding the modes. Please refer to Chapter 'Modes' in 'SPS-CNC Interfaces to IEC 1131-3' for details about executing the NC program in the respective mode.

The 'OPMOD_xx' function block can be used for selecting the NC modes via selector keys that are transferred. The function block provides the CNC interface signals as output signals.

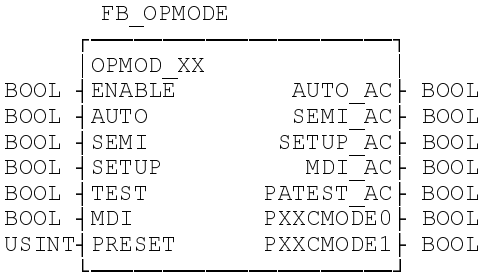
The mode bits are encoded according to the requested NC mode after a positive edge has been detected at the selected key. Furthermore, an additional output is activated that shows the currently active mode.

The 'PRESET' input is used for defining the mode pre-selection at the beginning of the program.

Removing the processing enabling signal enables changes of the input to be ignored. This means that the outputs remain at their last states. The outputs will be adjusted to the current input signals when the block enabling signal is re-applied.

Note: When different selector keys are pressed at the same time, the NC mode is preselected whose positive edge was last to be detected.

7.2 Interface overview



7.3 Description of the identifiers

Identifier	Application	Type
	Input variables	
ENABLE	Block enabling 0: No interpretation of the inputs. Outputs retain their states. 1: Block processing	BOOL
AUTO	Selector key for 'automatic mode' Automatic mode is selected when a positive edge is detected.	BOOL
SEMI	Selector key for 'semi-automatic mode' Semi-automatic mode is selected when a positive edge is detected.	BOOL
SETUP	Selector key for 'setup mode' Setup mode is selected when a positive edge is detected.	BOOL
TEST	Selector key for 'automatic parameter test mode' Automatic parameter test mode is selected when a positive edge is detected.	BOOL
MDI	Selector key for 'MDI mode' MDI mode is selected when a positive edge is detected.	BOOL
PRESET	This input is used for defining the pre-selection of the NC mode in the SPS program sequence. 0: Automatic mode 1: Semi-automatic mode 2: Setup mode 3: Automatic parameter test 4: MDI mode	USINT
	Output variables	
AUTO_AC	Indication of 'automatic mode' 0: no automatic mode 1: automatic mode pre-selected	BOOL
SEMI_AC	Indication of 'semi-automatic mode' 0: no semi-automatic mode 1: semi-automatic mode pre-selected	BOOL
SETUP_AC	Indication of 'setup mode' 0: no setup mode 1: setup mode pre-selected	BOOL
MDI_AC	Indication of 'MDI mode' 0: no MDI mode 1: MDI mode pre-selected	BOOL
PATEST_AC	Indication of 'automatic parameter test mode' 0: no automatic parameter test mode 1: automatic parameter test mode active	BOOL
PxxCMODE0	CNC mode bit 2^0 This bit is '1' or '0', depending on the currently active mode	BOOL
PxxCMODE1	CNC mode bit 2^1 This bit is '1' or '0', depending on the currently active mode	BOOL

7.4 Encoding the NC Modes

The two mode bits 'PxxCMODE0' and 'PxxCMODE1' are used for selecting the four possible NC modes in the NC. These bits are transferred via the SPS to the CNC.

The modes are encoded as follows:

Mode	PxxCMODE1	PxxCMODE0
Automatic	0	0
Semi-automatic	0	1
Setup	1	0
Automatic parameter test	1	1

Note: Changing the NC mode during NC program execution results in an immediate stop.

7.5 Application

Example: Mode selection for a process.

Four keys are available for mode selection. These keys shall be used for selecting 'automatic mode', 'semi-automatic mode', 'MDI mode' or 'Setup mode' of the NC process '0'. Each key has an indicator lamp allocated that shows the currently selected mode. Changing between modes shall be possible at any time. This means that the FB is continuously enabled for processing.

The CNC-SPS interface signals for the mode selection of process '0' are:

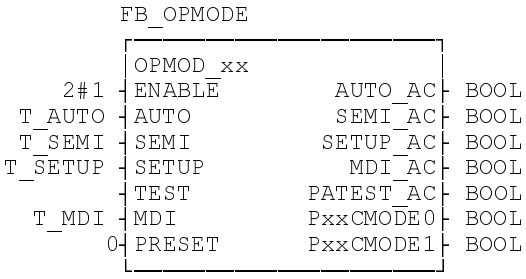
- P00CMODE0
- P00CMODE1

Those two signals must be connected to the FB outputs that are provided for this purpose.

Identifier names that are used in the SPS:

T_AUTO	Key for selecting 'automatic mode'
T_SEMI	Key for selecting 'semi-automatic mode'
T_SETUP	Key for selecting 'setup mode'
T_MDI	Key for selecting 'MDI mode'
ML_AUTO	Flag for 'automatic mode' indicator lamp
ML_SEMI	Flag for 'semi-automatic mode' indicator lamp
ML_SETUP	Flag for 'setup mode' indicator lamp
ML_MDI	Flag for 'MDI mode' indicator lamp
P00CMODE0	SPS-CNC interface signal mode bit 2^0
P00CMODE1	SPS-CNC interface signal mode bit 2^1

Setting the function block parameters:



8 Function Block Data Sheet: RNMOD_xx

8.1 Description

At the SPS side, distinction is made between three different program execution methods.

An NC program may be executed

- in single block mode,
- in single cycle mode, or
- in continuous operation.

Internally (i.e. within the NC), NC program execution only distinguishes between single block and single cycle. The generation of continuous execution mode for NC programs must be programmed in the SPS.

Single block mode

In 'single block mode', the individual NC blocks will only be executed after a start impulse has been issued via the SPS. The next NC block can only be executed after the previous one has been completed and a new start impulse has been issued. Please refer to Chapter 'Single Step Processing' of 'SPS Interface Description to IEC 1131-3' for details.

The 'PxxCSINGL' CNC-SPS interface signal is set to inform the NC about the selection proper of 'single step mode'. This signal must be connected as an output signal at the function block.

Single cycle mode

The NC is in 'single cycle mode' when the 'PxxCSINGL' signal is '0'. In that mode, a single start impulse causes a complete NC program to be processed in the selected mode (automatic mode, semi-automatic mode, setup mode). Please refer to Chapter 'Modes' of 'SPS Interface Description to IEC 1131-3' for details.

Continuous Operation

'Continuous operation mode' must be programmed in the SPS. It is based on the 'single cycle' execution method. If 'continuous operation' mode has been activated, the FB provides an output signal that must be interconnected with the 'PxxSREADY' in an additional logic element. This interconnection will then be used for influencing the 'PxxCADV' start signal. This causes a forward start to be generated after each successfully terminated 'single cycle', thus producing continuous operation. If the ‘PRGST_xx’ block is employed, the output signal for 'continuous operation active' only is to be passed on to that block.

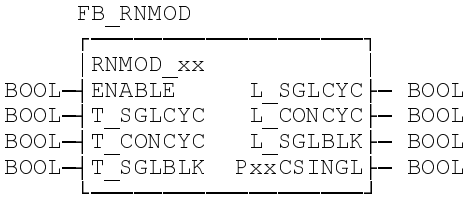
The 'RNMOD_xx' function block provides an output signal that is related to the active mode, and may be used for further processing.

Note: If the ‘PRGST_xx’ block is not used, a logic must be programmed that is able to abort and/or interrupt 'continuous operation'. The function block merely provides the signal that 'continuous operation' shall be activated.

Removing the processing enabling signal enables changes of the input to be ignored. This means that the outputs remain at their last states. The outputs will be adjusted to the current input signals when the block enabling signal is re-applied.

Note: When different selector keys are pressed at the same time, the NC mode is preselected whose positive edge was last to be detected.

8.2 Interface Overview



8.3 Description of the Identifiers

Identifier	Application	Type
	Input variables	
ENABLE	Block enabling 0: No interpretation of the inputs. Outputs retain their states. 1: Block processing	BOOL
T_SGLCYC	Selector key for 'single cycle mode' Single cycle mode is selected when a positive edge is detected.	BOOL
T_CONCYC	Selector key for 'continuous operation' Continuous operation is selected when a positive edge is detected. The signal must be interconnected in an additional SPS logic element for continuous operation generation.	BOOL
T_SGLBLK	Selector key for 'single step mode' Single step mode is selected when a positive edge is detected.	BOOL
	Output variables	
L_SGLCYC	Indication - 'single cycle mode' has been pre-selected 0: no single cycle mode 1: single cycle mode is pre-selected	BOOL
L_CONCYC	Indication - 'continuous operation' has been pre-selected The signal may be used for programming the continuous operation program sequence in the SPS. 0: continuous operation is not active 1: continuous operation is pre-selected	BOOL
L_SGLBLK	Indication - single step mode has been activated 0: no single step mode 1: single step mode is pre-selected	BOOL
PxxCSINGL	CNC-SPS interface signal for single step selection 0: single step mode is not active 1: single step mode is active	BOOL

8.4 Application

Example: Selection of program execution.

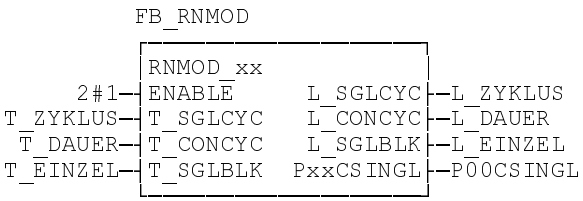
There are three keys in a control panel that permit the program execution mode of process '0' to be selected (single block, single cycle, continuous operation). A lamp indicates the currently selected mode. The selection may be changed at any time.

'P00CSINGL' is the interface signal between CNC and SPS that is used for single block selection.

Identifier names that are used in the SPS:

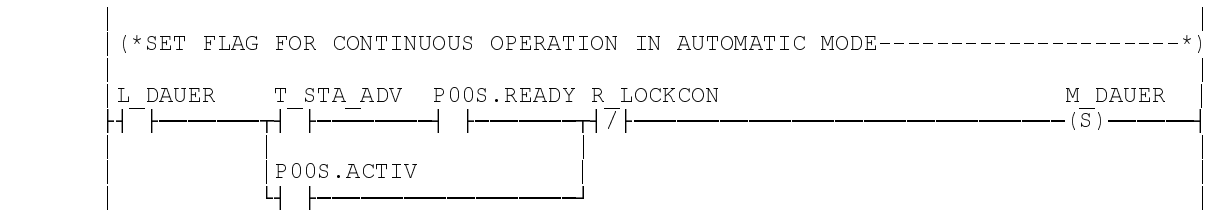
T_ZYKLUS	Key for selecting 'single cycle mode'
T_DAUER	Key for selecting 'continuous operation'
T_EINZEL	Key for selecting 'single block mode'
L_ZYKLUS	'Cycle mode' indicator lamp
L_DAUER	'Continuous operation' indicator lamp
L_EINZEL	'Single step mode' indicator lamp
P00CSINGL	SPS-CNC interface signal for 'single block mode'

Setting the function block parameters:

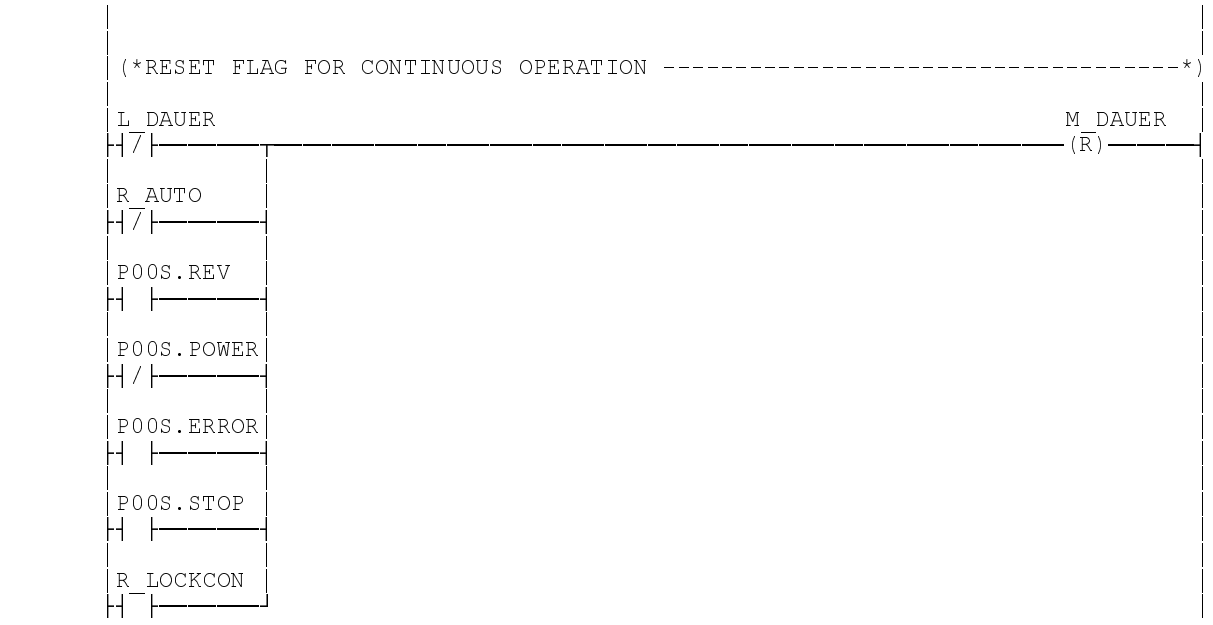


8.5 Excerpt from the SPS Program for Advance Start

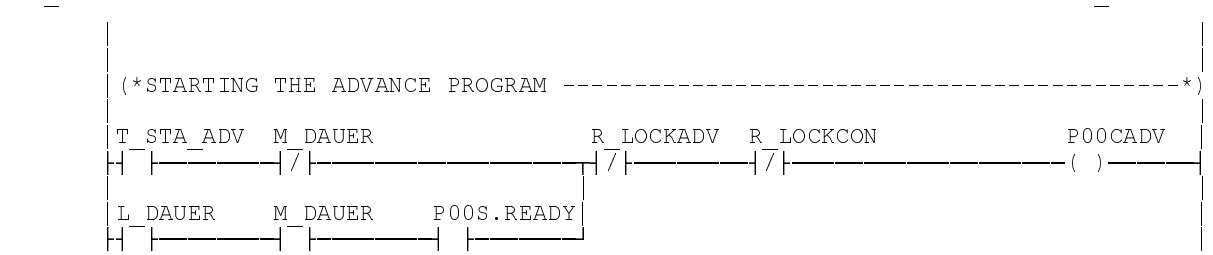
The following excerpt shows a typical program of the continuous operation sequence for process '0'.




```
L_DAUER..................... CONTINUOUS OPERATION SELECTED.......... VAR_INPUT. BOOL
T_STA_ADV................... ADVANCE KEY STATION.................... VAR_INPUT. BOOL
P00S.READY.................. ....................................... VAR_INPUT. BOOL
R_LOCKCON................... CONTINUOUS OPERATION INTERLOCK......... VAR_INPUT. BOOL
M_DAUER..................... FLAG FOR CONTINUOUS OPERATION.......... .......... BOOL
P00S.ACTIV.................. ....................................... VAR_INPUT. BOOL
```




```
L_DAUER..................... CONTINUOUS OPERATION SELECTED.......... VAR_INPUT. BOOL
M_DAUER..................... FLAG FOR CONTINUOUS OPERATION.......... .......... BOOL
R_AUTO...................... AUTOMATIC MODE......................... VAR_INPUT. BOOL
P00S.REV.................... ....................................... VAR_INPUT. BOOL
P00S.POWER.................. ....................................... VAR_INPUT. BOOL
P00S.ERROR.................. ....................................... VAR_INPUT. BOOL
P00S.STOP................... ....................................... VAR_INPUT. BOOL
R_LOCKCON................... CONTINUOUS OPERATION INTERLOCK......... VAR_INPUT. BOOL
```




```
T_STA_ADV................... ADVANCE KEY STATION.................... VAR_INPUT. BOOL
M_DAUER..................... FLAG FOR CONTINUOUS OPERATION.......... .......... BOOL
R_LOCKADV................... PROGRAM START INTERLOCK................ VAR_INPUT. BOOL
R_LOCKCON................... CONTINUOUS OPERATION INTERLOCK......... VAR_INPUT. BOOL
P00CADV..................... START SIGNAL FOR PROCESS............... .......... BOOL
L_DAUER..................... CONTINUOUS OPERATION SELECTED.......... VAR_INPUT. BOOL
M_DAUER..................... FLAG FOR CONTINUOUS OPERATION.......... .......... BOOL
P00S.READY.................. ....................................... VAR_INPUT. BOOL
.
.
.
```

9 Function Block Data Sheet: JGMOD_xx

9.1 Description

The selection of the CNC jog modes requires that the SPS controls interface signals in the interface between SPS and CNC. The ‘PxxCJOGM0’, ‘PxxCJOGM1’ and ‘PxxCJOGM2’ interface signals must be used for encoding the eight modes. Please refer to Chapter 'Modes', Section 'Jog Modes' of 'Interface Description to IEC 1131-3' for details about jog mode.

The 'JGMOD_xx' function block provides for the encoding of the jog modes that are selected via corresponding keys. The function block inputs are provided with the key signals; the outputs with interface signals and lamp signals.

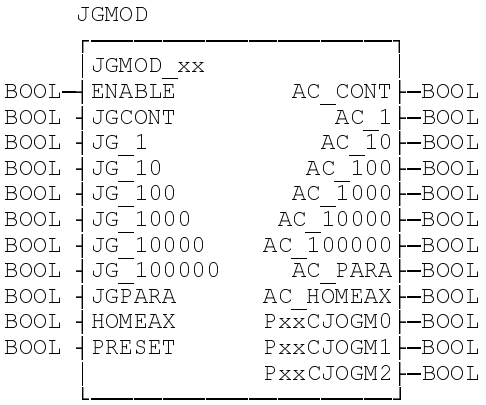
The 'single-axis homing' mode can be selected via the function block. There is **n o** mode interface signal between SPS and CNC for that mode. The generated output signal is provided for enabling the single axis homing move command (AxxCHOME, xx stands for the axis number).

When a positive edge is detected at a key input of the function block, the corresponding jog mode is encoded and an output is set that indicates the currently selected mode.

A preset input can be used for defining the jog mode that is selected at the beginning. 'Continuous jogging' mode is pre-selected if the input is not connected.

Removing the processing enabling signal enables changes of the input to be ignored. This means that the outputs remain at their last states. The outputs will be adjusted to the current input signals when the block enabling signal is re-applied.

9.2 Interface overview



9.3 Description of the Identifiers

Identifier	Application	Type
	Input variables	
ENABLE	Block enabling 0: No interpretation of the inputs. Outputs retain their states. 1: Block processing	BOOL
JGCONT	Selector key for 'continuous jog mode' Continuous jog mode is selected when a positive edge is detected.	BOOL
JG_1	Selector key for '1 AE jog distance mode' '1 AE jog distance mode is selected when a positive edge is detected.	BOOL
JG_10	Selector key for '0.001 mm jog distance mode' '10 AE jog distance mode is selected when a positive edge is detected.	BOOL
JG_100	Selector key for '0.01 mm jog distance mode' '100 AE jog distance mode is selected when a positive edge is detected.	BOOL
JG_1000	Selector key for '0.1 mm jog distance mode' '1000 AE jog distance mode is selected when a positive edge is detected.	BOOL
JG10000	Selector key for '1 mm jog distance mode' '10000 AE jog distance mode is selected when a positive edge is detected.	BOOL
JG100000	Selector key for '10 mm jog distance mode' '100000 AE jog distance mode is selected when a positive edge is detected.	BOOL
JGPARA	Selector key for 'jog mode' via specification in the axis parameters. The 'jog distance via parameter specification mode' is selected when a positive edge is detected.	BOOL
HOMEAX	Selector key for 'single axis homing mode' 'Single axis homing mode' is selected when a positive edge is detected.	BOOL
PRESET	This input permits the following jog mode pre-selections to be made at the beginning of a program: 0: continuous jogging 1: jogging via parameter specification 2: single axis homing 3: jog distance: 100 000 AE 4: jog distance: 10 000 AE 5: jog distance: 1000 AE 6: jog distance: 100 AE 7: jog distance: 10 AE 8: jog distance: 1 AE	USINT
	Output variables	
AC_CONT	'Continuous jogging mode' has been activated 0: no continuous jogging 1: continuous jogging has been activated	BOOL
AC_1	'1 AE jog distance mode' has been activated 0: no jog distance mode 1: '1 AE jog distance mode' has been pre-selected	BOOL
AC_10	'10 AE jog distance mode' has been activated 0: no jog distance mode 1: '10 AE jog distance mode' has been pre-selected	BOOL

Identifier	Application	Type
AC_100	'100 AE jog distance mode' has been activated 0: no jog distance mode 1: '100 AE jog distance mode' has been pre-selected	BOOL
AC_1000	'1000 AE jog distance mode' has been activated 0: no jog distance mode 1: '1000 AE jog distance mode' has been pre-selected	BOOL
AC_10000	'10000 AE jog distance mode' has been activated 0: no jog distance mode 1: '10000 AE jog distance mode' has been pre-selected	BOOL
AC_100000	'100000 AE jog distance mode' has been activated 0: no jog distance mode 1: '100000 AE jog distance mode' has been pre-selected	BOOL
AC_PARA	'Jog distance via parameter value' has been activated 0: no jogging via parameter value 1: 'jog distance via parameter value' mode	BOOL
AC_HOMEAX	Single axis homing has been activated 0: single axis homing has not been activated 1: single axis homing has been activated	BOOL
PxxCJOGM0	CNC job mode bit 2^0 The bit is '1' or '0', depending on the activated jog mode.	BOOL
PxxCJOGM1	CNC job mode bit 2^1 The bit is '1' or '0', depending on the activated jog mode.	BOOL
PxxCJOGM2	CNC job mode bit 2^2 The bit is '1' or '0', depending on the activated jog mode.	BOOL

AE: resolution unit

9.4 Encoding the Jog Modes

The jog modes are selected by encoding the jog mode bits:

- PxxCJOGM0,
- PxxCJOGM1 and
- PxxCJOGM2.

Jog mode	SPS control signals		Jog distances		Unit			
	PxxCJOGM0	PxxCJOGM1	PxxCJOGM2	progr. fract. part digit=4	progr. fract. par t digit=5	Linear axis	Rotary axis	Spindle
continuous	0	0	0	any	any	mm or inch	units	degrees
1 AE	0	0	1	0.0001	0.00001	mm or inch	units	degrees
10 AE	0	1	0	0.001	0.0001	mm or inch	units	degrees
100 AE	0	1	1	0.01	0.001	mm or inch	units	degrees
1000 AE	1	0	0	0.1	0.01	mm or inch	units	degrees
10 000 AE	1	0	1	1	0.1	mm or inch	units	degrees
100 000 AE	1	1	0	10	1	mm or inch	units	degrees
var. jog distance-	1	1	1	any	any	mm or inch	units	degrees

AE: resolution unit

9.5 Application

Example: Jog mode selection for a process.

Requirements:
There are six keys in a control panel that permit the jog mode to be selected. Another key is provided that shall be used for enabling single axis homing. The selectable jog modes are:

- continuous jogging
- jogging a distance of 0.001 mm, 0.01 mm, 01 mm, 1 mm, and
- jogging a distance that is specified in the axis parameters.

The jog modes are selected for process '0'.

Furthermore, each key has a signal lamp allocated that is used for indicating the selected mode. Changing between modes is only possible in 'manual mode'. This means that processing of the function block will only be enabled after transition to 'setup mode'.

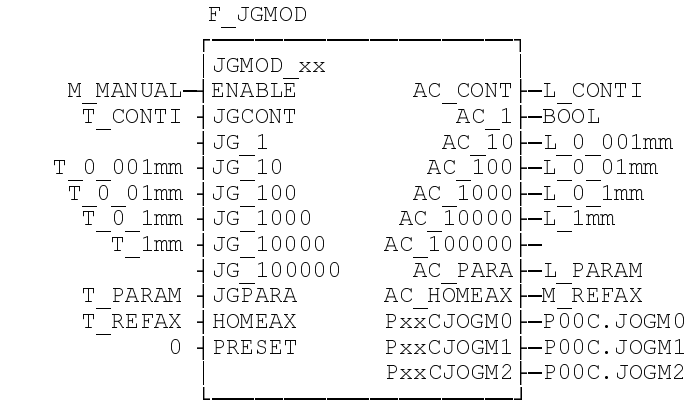
The CNC-SPS interface signals for jog mode selection of process '0' are:

- P00CJOGM0,
- P00CJOGM1,
- P00CJOGM2.

Identifier names that are used in the SPS:

M_MANUAL	Flag for recognizing 'Setup mode'
T_CONTI	Key for selecting 'continuous jogging'
T_0_001mm	Key for selecting 'jog distance 0.001 mm'
T_0_01mm	Key for selecting 'jog distance 0.01 mm'
T_0.1mm	Key for selecting 'jog distance 0.1 mm'
T_1mm	Key for selecting 'jog distance 1 mm'
T_PARAM	Key for selecting the 'parameter jog distance'
T_REFAX	Key for selecting 'single axis homing'
continuous jogging: 0	Default jog mode upon program start
L_CONTI	Lamp for 'continuous jogging'
L_0_001mm	Lamp for 'jog distance 0.001 mm'
L_0_01mm	Lamp for 'jog distance 0.01 mm'
L_0_1mm	Lamp for 'jog distance 0.1 mm'
L_1mm	Lamp for 'jog distance 1 mm'
L_PARAM	Lamp for 'parameter jog distance'
M_REFAX	Flag for enabling 'single axis homing'
P00C.JOGM0	SPS-CNC interface signal for jog mode bit 2^0
P00C.JOGM1	SPS-CNC interface signal for jog mode bit 2^1
P00C.JOGM2	SPS-CNC interface signal for jog mode bit 2^2

Setting the function block parameters:



'Single axis homing mode' is not a mode in the sense of the CNC.

⇒ The function block output must be interconnected in the enabling signal that triggers the axis movement. The ‘M_REFA’ flag of the example may be used for that purpose.

10 Function Block Data Sheet: SOTOP_xx

10.1 Description

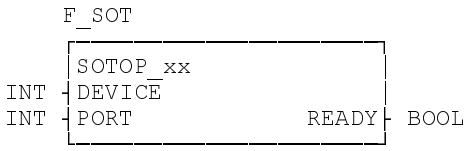
The corresponding interface must be provided with parameter values and be opened via the SPS if the SOT is connected via a SPS interface.

In that process, the function block is provided with the number of the interface and the device number.

An output parameter indicates that the interface has successfully been opened.

The interface is open and ready for communication when the function block output shows the readiness.

10.2 Interface Overview



10.3 Description of the identifiers

Identifier	Application	Type
DEVICE	Device number of the I/O unit to which the interface is allocated.	INT
PORT	Number of the interface	INT
Output variables		
READY	Output signal for successful initialization and opening of the interface. An impulse is output.	BOOL

10.4 Application

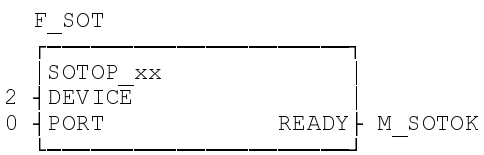
Example: Preparation for initializing an RS 232 interface that is used for connecting an SOT to a ReCo.

Requirement:
An SOT shall be connected to the RECO I/O unit with device number 2.

Identifier names that are used in the SPS:

M_SOTOK Flag used for inquiring whether or not the SOT interface has been initialized.

Setting the function block parameters:



11 Function Block Data Sheet: POWER_xx

11.1 Description

The NC performs the diagnoses for monitoring the EMERGENCY STOP circuit and the power activation. The SPS transfers the individual requirements via corresponding interface signals to the NC.

The validity of the diagnoses requires proper wiring of the EMERGENCY STOP circuit. The requirements of the 'Technical Documents' description must be fulfilled.

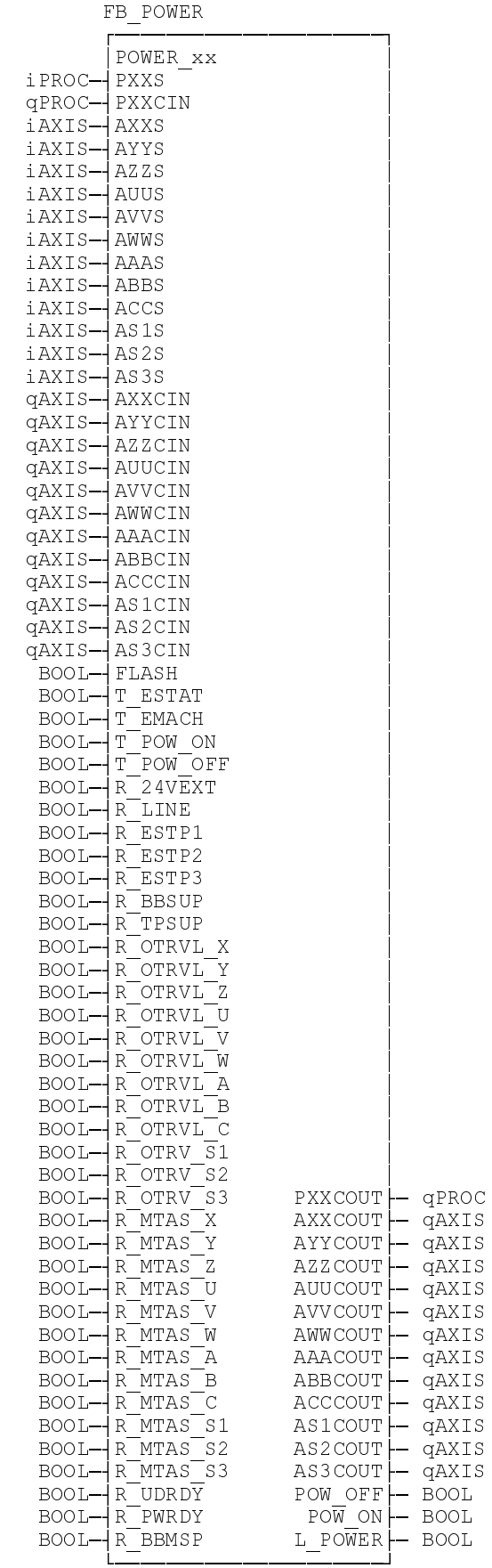
Please refer to the Chapter 'Power Activation' in 'SPS Interface Description to IEC 1131-3' for further explanations of the individual signals of the EMERGENCY STOP circuit and power activation.

If several processes are involved in processing, the power activation parameter values must be assigned for each individual process. This means that the function block must be declared and used according to the number of processes.

If not all signals exist that are required in the description, the parameter values must be set according to the instructions given in 'SPS Interface Description to IEC 1131-3'.

Note: The axis signals must only be connected for the axes that are allocated to the currently transferred process.

11.2 Interface overview



11.3 Description of the identifiers

Identifier	Application	Type
	Input variables	
PXXS	Process status signals for the process whose power shall be switched on	iPROC
PXXCIN	Process control signals for the process whose power shall be switched on	qPROC
AXXS	Status signals of the X axis	iAXIS
AYYS	Status signals of the Y axis	iAXIS
AZZS	Status signals of the Z axis	iAXIS
AUUS	Status signals of the U axis	iAXIS
AVVS	Status signals of the V axis	iAXIS
AWWS	Status signals of the W axis	iAXIS
AAAS	Status signals of the A axis	iAXIS
ABBS	Status signals of the B axis	iAXIS
ACCS	Status signals of the C axis	iAXIS
AS1S	Status signals of the 1st spindle	iAXIS
AS2S	Status signals of the 2nd spindle	iAXIS
AS3S	Status signals of the 3rd spindle	iAXIS
AXXCIN	Control signals of the X axis	qAXIS
AYYCIN	Control signals of the Y axis	qAXIS
AZZCIN	Control signals of the Z axis	qAXIS
AUUCIN	Control signals of the U axis	qAXIS
AVVCIN	Control signals of the V axis	qAXIS
AWWCIN	Control signals of the W axis	qAXIS
AAACIN	Control signals of the A axis	qAXIS
ABBCIN	Control signals of the B axis	qAXIS
ACCCIN	Control signals of the C axis	qAXIS
AS1CIN	Control signals of the 1st spindle	qAXIS
AS2CIN	Control signals of the 2nd spindle	qAXIS
AS3CIN	Control signals of the 3rd spindle	qAXIS
FLASH	Input for transferring a blink contact	BOOL
T_ESTAT	Station EMERGENCY STOP button	BOOL
T_EMACH	Machine EMERGENCY STOP button	BOOL
T_POWON	Power activation input	BOOL
T_POWOFF	Power de-activation input	BOOL
R_24VEXT	Feedback for external 24-V supply	BOOL
R_LINE	Feedback for auxiliary mains voltage	BOOL
R_ESTP1 - R_ESTP3	Feedback - protective device 1 through protective device 3	BOOL
R_BBSUP	Feedback - power supply unit operational	BOOL
R_TPSUP	Feedback - power supply temperature monitoring	BOOL
R_OTRVL_X	Safety limit switch of the X axis	BOOL
R_OTRVL_Y	Safety limit switch of the Y axis	BOOL
R_OTRVL_Z	Safety limit switch of the Z axis	BOOL
R_OTRVL_U	Safety limit switch of the U axis	BOOL
R_OTRVL_V	Safety limit switch of the V axis	BOOL
R_OTRVL_W	Safety limit switch of the W axis	BOOL

Identifier	Application	Type
	Input variables	
R_OTRVL_A	Safety limit switch of the A axis	BOOL
R_OTRVL_B	Safety limit switch of the B axis	BOOL
R_OTRVL_C	Safety limit switch of the C axis	BOOL
R_OTRVL_S1	Safety limit switch of the 1st spindle	BOOL
R_OTRVL_S2	Safety limit switch of the 2nd spindle	BOOL
R_OTRVL_S3	Safety limit switch of the 3rd spindle	BOOL
R_MTAS_X	Temperature-controlled safety switch of the X axis	BOOL
R_MTAS_Y	Temperature-controlled safety switch of the Y axis	BOOL
R_MTAS_Z	Temperature-controlled safety switch of the Z axis	BOOL
R_MTAS_U	Temperature-controlled safety switch of the U axis	BOOL
R_MTAS_V	Temperature-controlled safety switch of the V axis	BOOL
R_MTAS_W	Temperature-controlled safety switch of the W axis	BOOL
R_MTAS_A	Temperature-controlled safety switch of the A axis	BOOL
R_MTAS_B	Temperature-controlled safety switch of the B axis	BOOL
R_MTAS_C	Temperature-controlled safety switch of the C axis	BOOL
R_MTAS_S1	Temperature-controlled safety switch of the 1st spindle	BOOL
R_MTAS_S2	Temperature-controlled safety switch of the 2nd spindle	BOOL
R_MTAS_S3	Temperature-controlled safety switch of the 3rd spindle	BOOL
R_UDRDY	Feedback of the DC bus voltage	BOOL
R_PWRDY	Feedback - the master circuit breaker has picked up	BOOL
R_BBMSP	Feedback - the main spindle is operational	BOOL
	Output variables	
PXXCOUT	Process control signals for the process for which power is switched on	qPROC
AXXCOUT	Axis control signals for the X axis	qAXIS
AYYCOUT	Axis control signals for the Y axis	qAXIS
AZZCOUT	Axis control signals for the Z axis	qAXIS
AUUCOUT	Axis control signals for the U axis	qAXIS
AVVCOUT	Axis control signals for the V axis	qAXIS
AWWCOUT	Axis control signals for the W axis	qAXIS
AAACOUT	Axis control signals for the A axis	qAXIS
ABBCOUT	Axis control signals for the B axis	qAXIS
ACCCOUT	Axis control signals for the C axis	qAXIS
AS1COUT	Axis control signals for the 1st spindle	qAXIS
AS2COUT	Axis control signals for the 2nd spindle	qAXIS
AS3COUT	Axis control signals for the 3rd spindle	qAXIS
POW_OFF	De-activation relay for power shutdown 0: power is switched off 1: power is switched on	BOOL
POW_ON	Activation relay for power activation 0: no power activation or power has been activated 1: power is being activated The activation relay is only triggered until the main circuit breaker has picked up.	BOOL
L_POWER	Indication that power is switched on. 0: blinks if power has not been switched on process is ready for power being switched on 1: power has been switched on	

11.4 Application

Example: Activating the power for a process.

Requirements:
The power activation according to 'Interface Description to IEC 1131-1' shall be programmed for process '0'.

The process has two axes allocated (X axis and spindle). The X axis is entered in the parameters with axis number 1, the spindle with axis number 2.

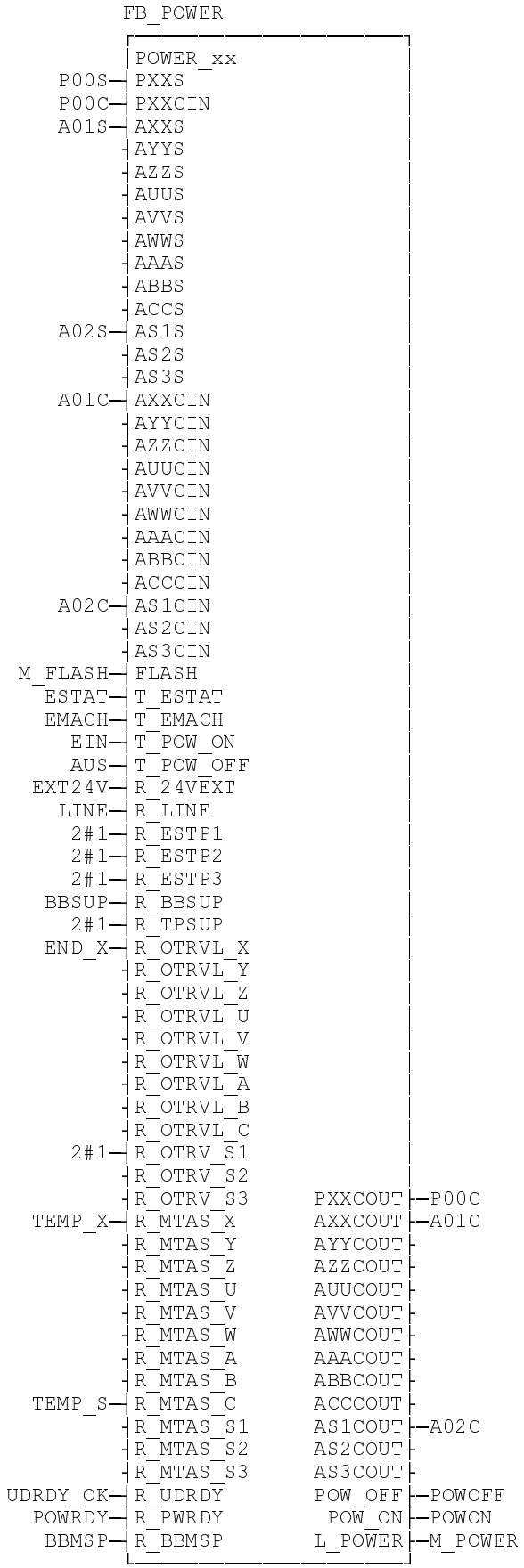
The 'ready for power activation' state shall be indicated by a blinking lamp. The lamp shall continuously be on when the power is switched on.

Power activation signals:

ESTAT	EMERGENCY STOP button at station
EMACH	EMERGENCY STOP button for complete machine
EIN	Button for activating power
AUS	Button for power shutdown
EXT24V	Feedback of the external 24-V power supply
LINE	Auxiliary mains voltage
BBSUP	Power supply ready
END_X	Safety limit switch of the X axis
TEMP_X	Temperature-controlled safety switch of the X axis
TEMP_S	Temperature-controlled safety switch of the spindle
UDRDY_OK	DC bus voltage is OK
POWRDY	Main circuit breaker has picked up
BBMSP	Main spindle is operational

Non-existent signals must be connected at the function block according to Chapter 'Power Activation' of 'SPS Interface Description to IEC 1131-3'.

Setting the function block parameters for power activation:



12 Function Block Data Sheet: PRGST_xx

12.1 Description

The function block provides the logic that is required for starting an NC machining program.

Depending on the number of processes that participate in machining and on the interdependencies between the individual processes, different logic elements are required for starting the program. For this purpose, the function block must be informed of the allocation of the process to the overall system.

The input parameters are used for distinguishing between

- a process that runs independently of other processes (master process or single process); and
- a SUB process that is started by a higher-level process (a station).

If the process works as a SUB process, start in 'automatic mode' is performed from a higher-level process (master process). Within the higher-level NC program, that start is triggered by an AP (advance process), i.e. advance program start, or an RP (reverse process),, i.e. a reverse program start.

A restart is possible after an interruption.

Please refer to Chapter 'Process Synchronization' in 'SPS Interface Description to IEC 1131-3' for details about process synchronization.

In 'semi automatic mode' and 'setup mode', the start for an NC advance or NC reverse program is only possible via the process-related start keys. With process synchronization, a start from the higher-level unit is not possible.

If the process has been defined as a master process or a single process, the start for an NC advance or NC reverse program is possible in any mode.

Please refer to the Chapters 'Advance Program Start' and 'Reverse Program Start' in 'SPS Interface Description to IEC 1131-3' for a description of the differences in NC program execution in the individual modes.

Interlocks for 'advance program start' or 'continuous execution' can be transferred to the function block via the corresponding parameters.

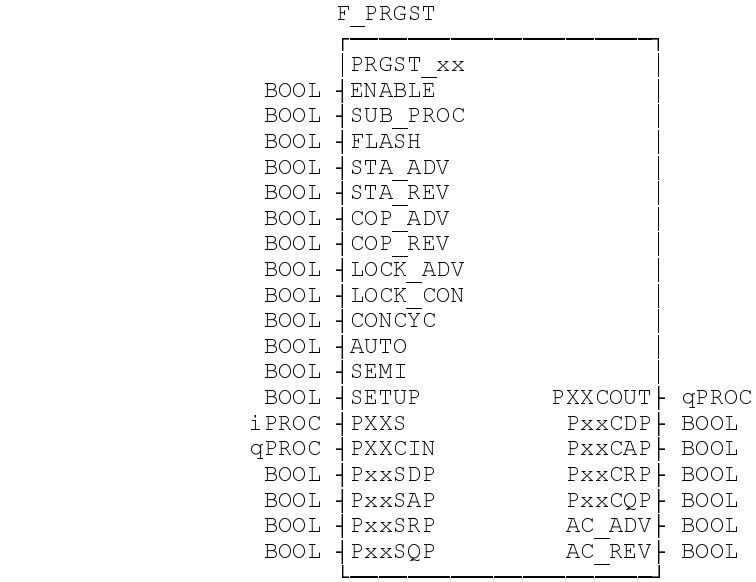
The state of the NC machining program that is to be started is indicated via function block outputs.

The following states are reported:

- no start possible
- ready to start
- NC program execution in progress

The processing enabling signal permits changes of the input to be ignored. The outputs will only be adapted to the existing input signals when the enabling signal is activated.

12.2 Interface Overview



12.3 Description of the Identifiers

Identifier	Application	Type
	Input variables	
ENABLE	Block enabling 0: Input changes are ignored 1: Input changes are processed	BOOL
SUB_PROC	Specification of the process definition 0: The process is a single process or a master process. This means that the NC program is started by process-related keys. 1: The process is a SUB process that is started from a higher-level NC process via the 'AP' or 'RP' command in 'automatic mode'.	BOOL
FLASH	Blink contact that is generated on the level that is above the function block.	BOOL
STA_ADV	The advance program of a single process or a master process is started via this input. Furthermore, it permits a SUB process in 'semi-automatic mode' or 'setup mode' to be started.	BOOL
STA_REV	The reverse program of a single process or a master process is started via this input. Furthermore, it permits a SUB process in 'semi-automatic mode' or 'setup mode' to be started.	BOOL
COP_ADV	The advance program of an interrupted SUB process can be re-started via this input.	BOOL
COP_REV	The reverse program of an interrupted SUB process can be re-started via this input.	BOOL
LOCKADV	This input permits advance program enabling to be influenced. 0: advance program start is enabled 1: advance program start is disabled	BOOL
LOCKCON	This input permits continuous execution enabling to be influenced. 0: continuous execution is enabled 1: continuous execution is disabled	BOOL
CONCYC	Transfer to the function block that the NC advance program shall be executed in continuous operation. 0: no continuous execution to be performed 1: continuous execution	BOOL

Identifier	Application	Type
	Input variables	
AUTO	Transfer of 'automatic mode' 0: no automatic operation 1: automatic operation is active	BOOL
SEMI	Transfer of 'semi-automatic mode' 0: no semi-automatic operation 1: semi-automatic operation is active	BOOL
SETUP	Transfer of 'setup mode' 0: 'setup mode' is not active 1: 'setup mode' is active	BOOL
PXXS	Input for transferring the process status signals. Input for status signals of the process that shall be started.	BOOL
PXXCIN	Input for transferring the process control signals. Input for control signals of the process that shall be started.	BOOL
PxxSDP	Process is defined - only to be used for SUB process	BOOL
PxxSAP	Request for advance program start - only to be used for SUB process	BOOL
PxxSRP	Request for reverse program start - only to be used for SUB process	BOOL
PxxSQP	Process is acknowledged - only to be used for SUB process	BOOL
	Output variables	
PXXCOUT	Output for transferring the control signals Output for control signals of the process that shall be started	BOOL
PxxCDP	Handshake for process defined	BOOL
PxxCAP	Handshake for advance program start request	BOOL
PxxCRP	Handshake reverse program start request	BOOL
PxxCQP	Handshake for process acknowledged	BOOL
AC_ADV	Indication of the advance program status 0: no advance program active 1: advance program active blinking: advance program ready for execution, or stopped	BOOL
AC_REV	Indication of the reverse program status 0: no reverse program active 1: reverse program active blinking: reverse program ready for execution, or stopped	BOOL

12.4 Application

Example: Performing process synchronization between two processes. The NC program start logic of the SUB process that is started via the master process is shown. The start sequence for the master process is not shown here.

Requirements:
Starting the NC advance and/or reverse programs of a SUB process.

The process that is started has the process number '1'. In 'automatic mode', the SUB process is started by the master process (process no. 0) via the NC command 'AP'. After an interruption, the stopped SUB process will be restarted from the master control panel.

The SUB process controls a station within the machining center. To improve handling, that station is provided with an independent control panel that permits the SUB process to be controlled outside 'automatic mode'.

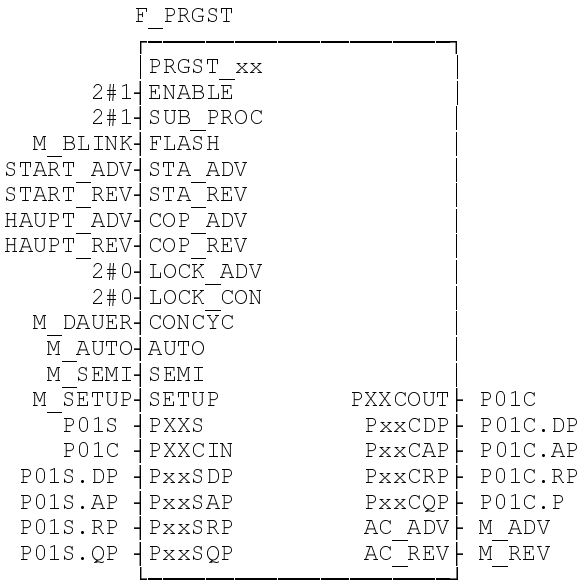
Signal lamps shall be used for indicating the current program execution state.

The function block is continually processed. An inhibition for advance program start or continuous execution is not provided.

Identifier names that are used in the SPS:

M_BLINK	Blinking flag
START_ADV	Key 'advance program start from station control panel'
START_REV	Key 'reverse program start from station control panel'
HAUPT_ADV	Key 'advance program start from master control panel'
HAUPT_REV	Key 'reverse program start from master control panel'
M_DAUER	Flag showing that NC continuous operation is selected.
M_AUTO	Flag for NC 'automatic mode'
M_SEMI	Flag for NC 'semi-automatic mode'
M_SETUP	Flag for NC 'setup mode'
P01S	Process status signals for process 1 (structure)
P01C	Process control signals from process 1 (structure)
M_ADV	Flag for indicator - forward program status
M_REV	Flag for indicator - reverse program status

Setting the function block parameters:



13 Function Block Data Sheet: AXIS_xx

13.1 Description

The ‘AXIS_xx’ function block is used for providing the base functions of an axis. The functions include

- moving the axis in setup mode (jog mode),
- manual homing,
- transferring the axis control signals for controller enabling, motion disabling, jog enabling, and homing switch.

The movements of jog mode are performed according to the pre-selected jog mode. Please refer to Chapter 'Jog Modes' in 'SPS Interface Description to IEC 1131-3' for details about the definition of the jog mode.

The axis type (analog or digital drive) is specified by an input parameter at the function block. Depending on that selection, certain axis signals are processed differently (e.g. controller enabling, homing switch).

In digital drives, the homing switch inputs are wired on the drive. In this case, the reference point input need not be transferred to the function block.

With respect to controller enabling, the specifications in Chapter 'Controller Enabling' of 'SPS Interface Description to IEC 1131-3' must be observed.

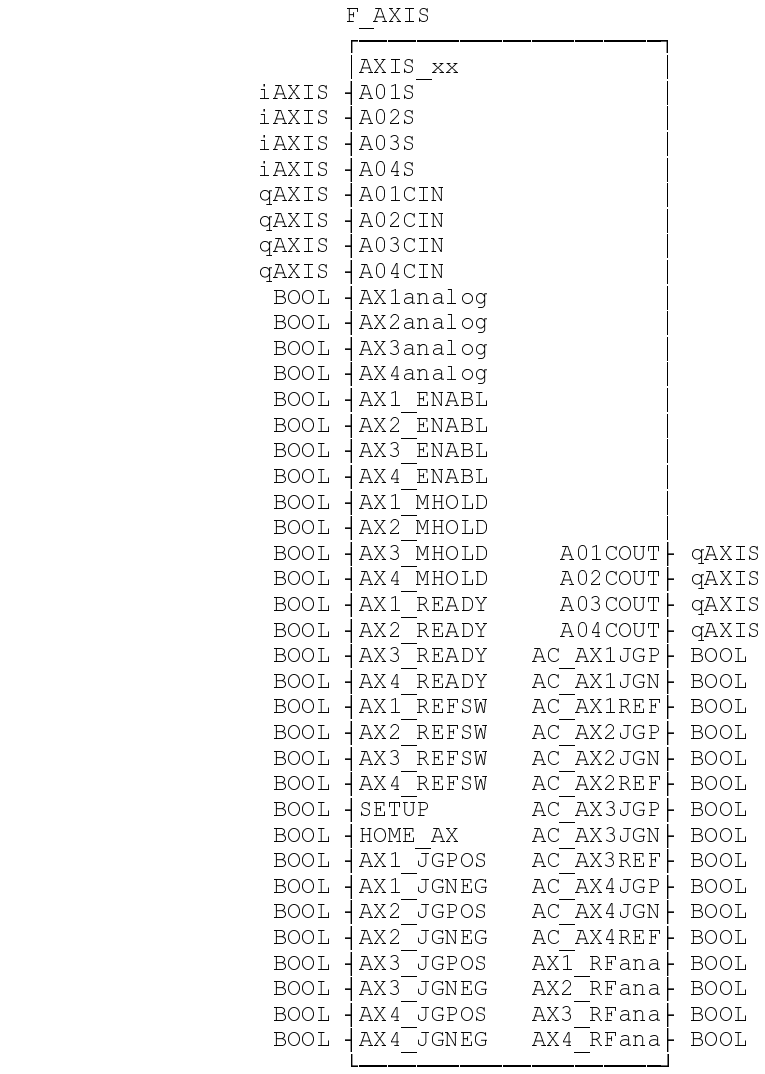
Controller enabling, axis inhibit, and jog enabling can be influenced via the corresponding function block inputs. The parameter values of the individual axes can be selected independently.

The request to the drive of performing its homing is executed after the 'single axis homing' mode has been selected and one of the two jog keys has been pressed.

The function block reports the motion state of an axis (i.e. positive/negative direction of movement, or homing) to the outside.

The function block is designed for controlling up to four axes. If fewer axes are required, merely the parameter values of the existing axes must be selected. If there are more than four axis, the function block must be declared and programmed repeatedly.

13.2 Interface Overview



13.3 Description of the Identifiers

Identifier	Application	Type
	Input variables	
A01S	Status signals of the 1st axis	iAXIS
A02S	Status signals of the 2nd axis	iAXIS
A03S	Status signals of the 3rd axis	iAXIS
A04S	Status signals of the 4th axis	iAXIS
A01CIN	Control signals of the 1st axis	qAXIS
A02CIN	Control signals of the 2nd axis	qAXIS
A03CIN	Control signals of the 3rd axis	qAXIS
A04CIN	Control signals of the 4th axis	qAXIS
AX1_analog	Definition of the drive type of axis 1, analog or digital 0: digital axis 1: analog axis	BOOL
AX2_analog	Definition of the drive type of axis 2, analog or digital 0: digital axis 1: analog axis	BOOL
AX3_analog	Definition of the drive type of axis 3, analog or digital 0: digital axis 1: analog axis	BOOL
AX4_analog	Definition of the drive type of axis 4, analog or digital 0: digital axis 1: analog axis	BOOL
AX1_ENABL	Jog enabling for the 1st axis 0: Jogging of axis not enabled 1: Axis is enabled for jog mode	BOOL
AX2_ENABL	Jog enabling for the 2nd axis 0: Jogging of axis not enabled 1: Axis is enabled for jog mode	BOOL
AX3_ENABL	Jog enabling for the 3rd axis 0: Jogging of axis not enabled 1: Axis is enabled for jog mode	BOOL
AX4_ENABL	Jog enabling for the 4th axis 0: Jogging of axis not enabled 1: Axis is enabled for jog mode	BOOL
AX1_MHOLD	Motion inhibit for the 1st axis, superimposes jog enabling 0: no motion inhibit 1: motion inhibit activated	BOOL
AX2_MHOLD	Motion inhibit for the 2nd axis, superimposes jog enabling 0: no motion inhibit 1: motion inhibit activated	BOOL
AX3_MHOLD	Motion inhibit for the 3rd axis, superimposes jog enabling 0: no motion inhibit 1: motion inhibit activated	BOOL
AX4_MHOLD	Motion inhibit for the 4th axis, superimposes jog enabling 0: no motion inhibit 1: axis is operational	BOOL
AX1_READY	Operational for the 1st axis; is used for influencing controller enabling. 0: axis is not operational; controller is not enabled 1: axis is operational	BOOL

Identifier	Application	Type
	Input variables	
AX2_READY	Operational for the 2nd axis, is used for influencing controller enabling. 0: axis is not operational; controller is not enabled 1: axis is operational	BOOL
AX3_READY	Operational for the 3rd axis, is used for influencing controller enabling. 0: axis is not operational; controller is not enabled 1: axis is operational	BOOL
AX4_READY	Operational for the 4th axis, is used for influencing controller enabling. 0: axis is not operational; controller is not enabled 1: axis is operational	BOOL
AX1_REFSW	Homing switch of the 1st axis is only required if the switch is not wired to the drive. 0: homing switch not actuated 1: homing switch actuated	BOOL
AX2_REFSW	Homing switch of the 2nd axis is only required if the switch is not wired to the drive. 0: homing switch not actuated 1: homing switch actuated	BOOL
AX3_REFSW	Homing switch of the 3rd axis is only required if the switch is not wired to the drive. 0: homing switch not actuated 1: homing switch actuated	BOOL
AX4_REFSW	Homing switch of the 4th axis is only required if the switch is not wired to the drive. 0: homing switch not actuated 1: homing switch actuated	BOOL
SETUP	'Setup mode' is active 0: 'Setup mode' has not been pre-selected 1: 'Setup mode' has been pre-selected	BOOL
HOME_AX	'Single axis homing mode' has been selected 0: homing has not been selected 1: homing has been selected	BOOL
AX1_JGPOS	Positive motion request to axis 1; taking the pre-selected jog mode into account 0: no motion request 1: motion request	BOOL
AX1_JGNEG	Negative motion request to axis 1; taking the pre-selected jog mode into account 0: no motion request 1: motion request	BOOL
AX2_JGPOS	Positive motion request to axis 2; taking the pre-selected jog mode into account 0: no motion request 1: motion request	BOOL
AX2_JGNEG	Negative motion request to axis 2; taking the pre-selected jog mode into account 0: no motion request 1: motion request	BOOL
AX3_JGPOS	Positive motion request to axis 3; taking the pre-selected jog mode into account 0: no motion request 1: motion request	BOOL
AX3_JGNEG	Negative motion request to axis 3; taking the pre-selected jog mode into account 0: no motion request 1: motion request	BOOL

Identifier	Application	Type
	Input variables	
AX4_JGPOS	Positive motion request to axis 4; taking the pre-selected jog mode into account 0: no motion request 1: motion request	BOOL
AX4_JGNEG	Negative motion request to axis 4; taking the pre-selected jog mode into account 0: no motion request 1: motion request	BOOL
	Output variables	
A01COUT	Control signals for the 1st axis	qAXIS
A02COUT	Control signals for the 2nd axis	qAXIS
A03COUT	Control signals for the 3rd axis	qAXIS
A04COUT	Control signals for the 4th axis	qAXIS
AC_AX1JGP	Movement of axis 1 in positive direction 0: no movement in positive direction 1: positive movement	BOOL
AC_AX1JGN	Movement of axis 1 in negative direction 0: no movement in negative direction 1: negative movement	BOOL
AC_AX1REF	Axis 1 performs homing 0: no homing 1: homing active	
AC_AX2JGP	Movement of axis 2 in positive direction 0: no movement in positive direction 1: positive movement	BOOL
AC_AX2JGN	Movement of axis 2 in negative direction 0: no movement in negative direction 1: negative movement	BOOL
AC_AX2REF	Axis 2 performs homing 0: no homing 1: homing active	
AC_AX3JGP	Movement of axis 3 in positive direction 0: no movement in positive direction 1: positive movement	BOOL
AC_AX3JGN	Movement of axis 3 in negative direction 0: no movement in negative direction 1: negative movement	BOOL
AC_AX3REF	Axis 3 performs homing 0: no homing 1: homing active	
AC_AX4JGP	Movement of axis 4 in positive direction 0: no movement in positive direction 1: positive movement	BOOL
AC_AX4JGN	Movement of axis 4 in negative direction 0: no negative jogging of axis 4 1: negative jogging of axis 4	BOOL
AC_AX4REF	Axis 4 performs homing 0: no homing 1: homing active	
AX1_RFana	Controller enabling of axis 1 for analog drives. The output must be wired at the drive without any further interconnections. 0: no controller enabling 1: controller enabling	BOOL

Identifier	Application	Type
	Output variables	
AX2_RFana	Controller enabling of axis 2 for analog drives. The output must be wired at the drive without any further interconnections. 0: no controller enabling 1: controller enabling	BOOL
AX3_RFana	Controller enabling of axis 3 for analog drives. The output must be wired at the drive without any further interconnections. 0: no controller enabling 1: controller enabling	BOOL
AX4_RFana	Controller enabling of axis 4 for analog drives. The output must be wired at the drive without any further interconnections. 0: no controller enabling 1: controller enabling	BOOL

13.4 Application

Example: Providing the base axis functionality for three axes.

Requirements:
A process has three axes allocated:

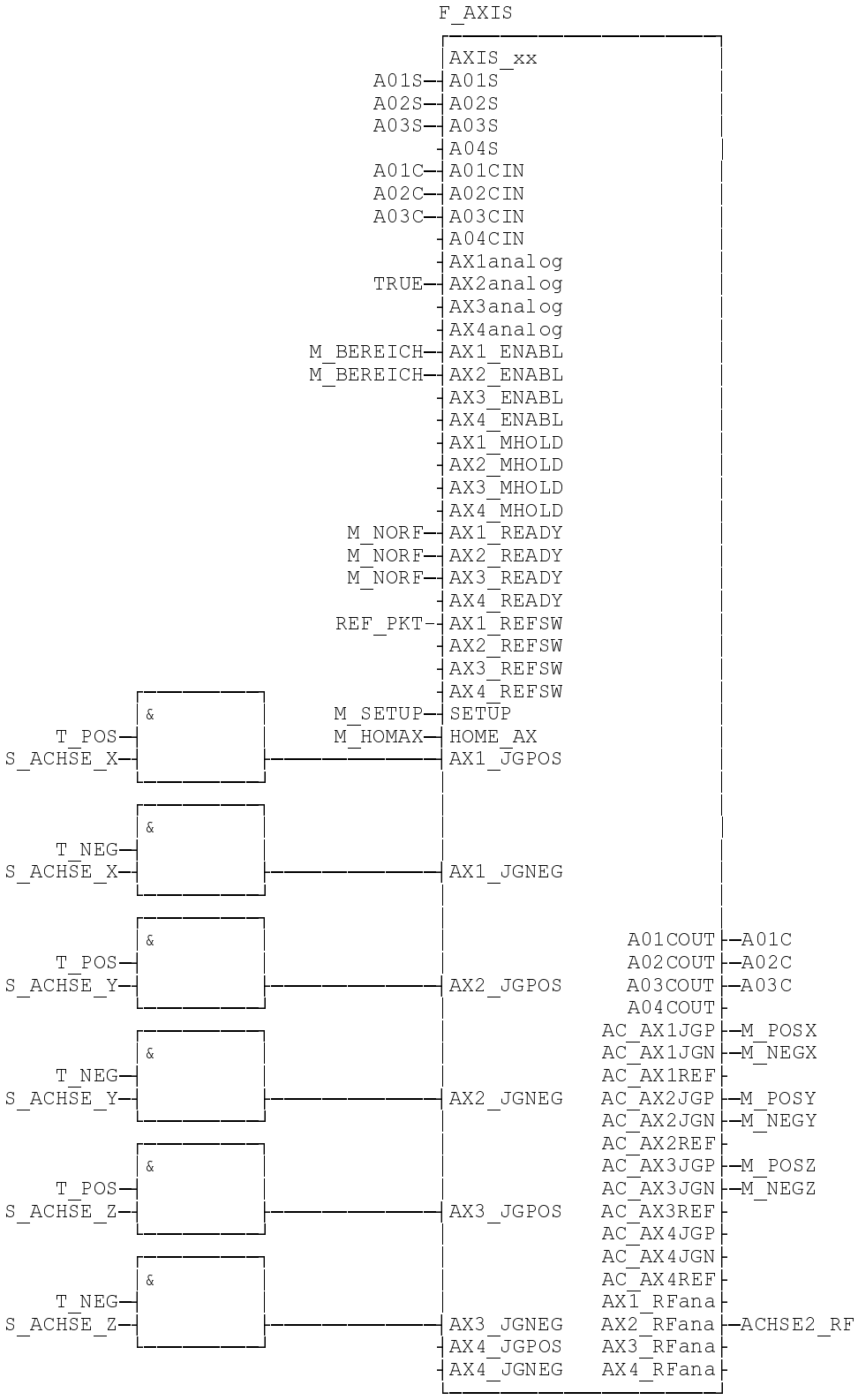
- Axis 1: digital rotary axis with axis number 1
- Axis 2: analog linear axis with axis number 2
- Axis 3: digital linear axis with axis number 3

Axis selection is pre-selected via a selector switch. There is one key for 'positive jogging' and one for 'negative jogging'. The controller enabling signals of the axes are influenced by external events. Jogging the axes is only permitted within a defined travel range. The axis movement is blocked when the range is left.

Identifier names that are used in the SPS:

S_ACHSE_X	Selector switch position X axis
S_ACHSE_Y	Selector switch position Y axis
S_ACHSE_Z	Selector switch position Z axis
T_POS	Axis 'positive jogging'
T_NEG	Axis 'negative jogging'
M_NORF	Switch off controller enabling signal
M_BEREICH	Travel range flag
M_NEGX	Lamp flag neg. movement X axis
M_NEGY	Lamp flag neg. movement Y axis
M_NEGZ	Lamp flag neg. movement Z axis
M_POSX	Lamp flag pos. movement X axis
M_POSY	Lamp flag pos. movement Y axis
M_POSZ	Lamp flag pos. movement Z axis
REF_PKT	Homing point switch axis 2
ACHSE2_RF	Controller enabling signal for analog axis
A01C	Control signals X axis
A01S	Status signals X axis
A02C	Control signals Y axis
A02S	Status signals Y axis
A03C	Control signals Z axis
A03S	Status signals Z axis

Setting the function block parameters:



14 Function Block Data Sheet: MU4<yy>_xx

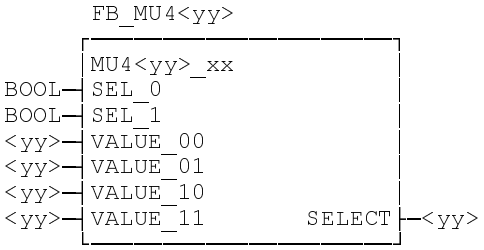
14.1 Description

The functionality of a 4-to-1 <yy> multiplexer is implemented here (<yy> stands for the different valid data types). There is one function block for each data type.

The following data types are supported:

- BYTE
- WORD
- INTEGER
- TIME
- UNSIGNED SHORT INTEGER
- UNSIGNED INTEGER
- SHORT INTEGER

14.2 Interface Overview



14.3 Description of the Identifiers

Identifier	Application	Type
Input variables		
SEL_0	Selector input of significance 2^0	BOOL
SEL_1	Selector input of significance 2^1	BOOL
VALUE_00	Output value if SEL_1 = 0 and SEL_0 = 0	<yy>
VALUE_01	Output value if SEL_1 = 0 and SEL_0 = 1	<yy>
VALUE_10	Output value if SEL_1 = 1 and SEL_0 = 0	<yy>
VALUE_11	Output value if SEL_1 = 1 and SEL_0 = 1	<yy>
Output variables		
SELECT	Output of the input selection	<yy>

14.4 Encoding the Selector Inputs

SEL_1	SEL_0	SELECT
0	0	VALUE_00
0	1	VALUE_01
1	0	VALUE_10
1	1	VALUE_11

14.5 Application

Selecting an input byte as a function of output flags. The ‘MU4BY_xx’ function block shall be used for that purpose.

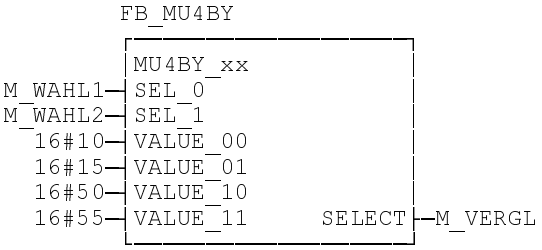
Identifier names that are used in the SPS:

M_WAHL1 Selector flag for significance 2^0

M_WAHL2 Selector flag for significance 2^1

M_VERGL Storage register for the comparison value

Setting the function block parameters:



15 Function Block Data Sheet: ERRES_xx

15.1 Description

The 'ERRES_xx' function block acknowledges a pending NC error and performs a control-reset. It acts directly upon the ‘PxxCCLEAR’ interface signal.

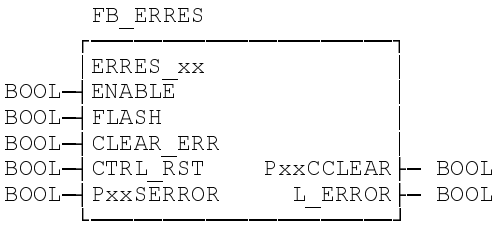
The ‘PxxSERROR’ signal (error is pending) is used for deciding whether the next action to be performed will be 'clear error' or 'control-reset'.

The block indicates a pending error via an output.

The controller only interprets a control-reset if the currently processed NC program is stopped.

Please refer to Chapter 'Clear Errors' in 'SPS Interface Description to IEC 1131-3' for details about clearing errors and control-reset.

15.2 Interface Overview



15.3 Description of the Identifiers

Identifier	Application	Type
ENABLE	Block enabling 0: no enabling for processing. Input changes are ignored 1: enabled for processing	BOOL
FLASH	Blinking contact for error indication	BOOL
CLEAR_ERR	Clear error key A positive edge at the input acknowledges a pending error.	BOOL
CTRL_RST	Control-Reset key Provided that the NC program is stopped, a positive edge at the input triggers a control-reset.	BOOL
PxxSERROR	Interface signal for 'error pending' 0: no error 1: error is pending	BOOL
Output variables		
PxxCCLEAR	Interface signal for 'clear error' and/or 'control-reset'. If there is a pending error, it will be cleared by a positive edge at the output. If there is no pending error, there will be a control-reset.	BOOL
L_ERROR	This output blinks if there is an error pending.	BOOL

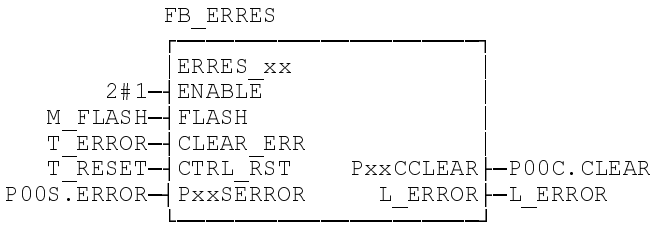
15.4 Application

Provides the logic for clearing errors and resetting the program for process 0.

Identifier that are used by the SPS:

M_FLASH	Blinking flag
T_ERROR	Key for clearing errors
T_RESET	Key for performing control-reset
P00S.ERROR	CNC-SPS interface signal: NC error is pending
P00C.CLEAR	SPS-CNC interface signal: Clear NC error or perform control-reset
L_ERROR	Indicator lamp for NC error

Setting the function block parameters:



16 Function Block Data Sheet: SEL4_xx

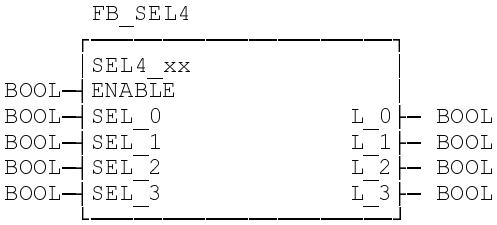
16.1 Description

The function block permits four different outputs to be selected as a function of four invariably allocated inputs. If a positive edge is recognized at one of the four inputs, the corresponding output will be set to logic '1'.

At the beginning of the processing (when a selector input has not yet been actuated), output 0 is set to the default value logic '1'.

Note: When different keys are pressed at the same time, the last positive edge is interpreted.

16.2 Interface Overview



16.3 Description of the Identifiers

Identifier	Application	Type
ENABLE	Enabling input for processing 0: no enabling, changes of the inputs are not interpreted 1: normal block processing	BOOL
SEL_0	a positive edge at this input causes output L_0 to be set	BOOL
SEL_1	a positive edge at this input causes output L_1 to be set	BOOL
SEL_2	a positive edge at this input causes output L_2 to be set	BOOL
SEL_3	a positive edge at this input causes output L_3 to be set	BOOL
Output variables		
L_0	Indicates that a positive edge has been detected at the SEL_0 selector key. The output is reset when a positive edge occurs at one of the other selector inputs.	BOOL
L_1	Indicates that a positive edge has been detected at the SEL_1 selector key. The output is reset when a positive edge occurs at one of the other selector inputs.	BOOL
L_2	Indicates that a positive edge has been detected at the SEL_2 selector key. The output is reset when a positive edge occurs at one of the other selector inputs.	BOOL
L_3	Indicates that a positive edge has been detected at the SEL_3 selector key. The output is reset when a positive edge occurs at one of the other selector inputs.	BOOL

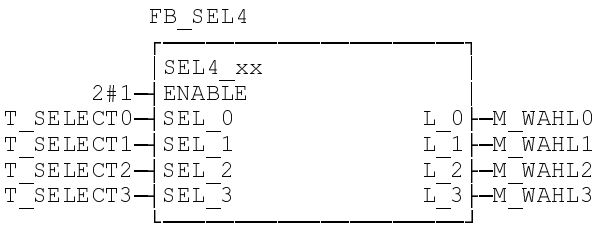
16.4 Application

Function key selection.

Identifier names that are used in the SPS:

T_SELECT0	Function key 1
T_SELECT1	Function key 2
T_SELECT2	Function key 3
T_SELECT3	Function key 4
M_WAHL0	Flag for function key 1
M_WAHL1	Flag for function key 2
M_WAHL2	Flag for function key 3
M_WAHL3	Flag for function key 4

Setting the function block parameters:



17 Function Block Data Sheet: STOPP_xx

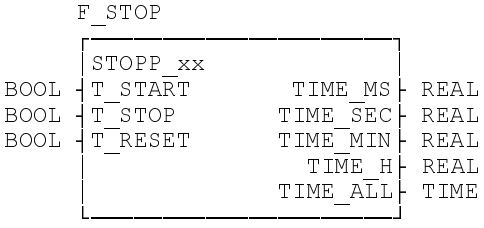
17.1 Description

The function block performs the function of a stop-watch. The inputs are used for starting, stopping, and resetting time sequences.

The elapsed time interval is provided in different resolutions. The time is output in milliseconds, seconds, minutes, hours, and in the general SPS 'TIME' data type.

The stop-watch wraps around after 23 days. The accuracy depends on the SPS cycle time.

17.2 Interface Overview



17.3 Description of the Identifiers

Identifier	Application	Type
	Input variables	
T_START	A positive edge at the input starts time measurement	BOOL
T_STOP	A positive edge at the input stops time measurement	BOOL
T_RESET	A positive edge at the input resets the time to 0.	BOOL
	Output variables	
TIME_MS	Elapsed time in milliseconds	REAL
TIME_SEC	Elapsed time in seconds	REAL
TIME_MIN	Elapsed time in minutes	REAL
TIME_H	Elapsed time in hours	REAL
TIME_ALL	Elapsed time in SPS TIME format	TIME

18 Index

List of Customer Service Points

Germany

Sales area Center	Sales area East	Sales area West	Sales area North
INDRAMAT GmbH D-97816 Lohr am Main Bgm.-Dr.-Nebel-Str. 2 Phone: 09352/40-0 Fax: 09352/40-4885	INDRAMAT GmbH D-09120 Chemnitz Beckerstraße 31 Phone: 0371/3555-0 Fax: 0371/3555-230	INDRAMAT GmbH D-40849 Ratingen Hansastraße 25 Phone: 02102/4318-0 Fax: 02102/41315	INDRAMAT GmbH D-22085 Hamburg Fährhausstraße 11 Phone: 040/227126-16 Fax: 040/227126-15
Sales area South	**Sales area South-West**		**INDRAMAT Service Hotline**
INDRAMAT GmbH D-80339 München Ridlerstraße 75 Phone: 089/540138-30 Fax: 089/540138-10	INDRAMAT GmbH D-71229 Leonberg Böblinger Straße 25 Phone: 07152/972-6 Fax: 07152/972-727		INDRAMAT GmbH Phone: D-0172/660 040 6 -or- Phone: D-0171/333 882 6

Customer service points in Germany

Europa

Austria	Austria	Belgium	Denmark
G.L.Rexroth Ges.m.b.H. Geschäftsbereich INDRAMAT A-1140 Wien Hägelingasse 3 Phone: 1/9852540-400 Fax:1/9852540-93	G.L.Rexroth Ges.m.b.H. Geschäftsbereich INDRAMAT A-4061 Pasching Randlstraße 14 Phone: 07229/4401-36 Fax: 07229/4401-80	Mannesmann Rexroth N.V.-S.A. Geschäftsbereich INDRAMAT B-1740 Ternat Industrielaan 8 Phone: 02/5823180 Fax: 02/5824310	BEC Elektronik AS DK-8900 Randers Zinkvej 6 Phone: 086/447866 Fax: 086/447160
England	**Finland**	**France**	**France**
Mannesmann Rexroth Ltd. INDRAMAT Division Cirencester, Glos GL7 1YG 4 Esland Place, Love Lane Phone: 01285/658671 Fax: 01285/654991	Rexroth Mecman OY SF-01720 Vantaa Riihimiehentie 3 Phone: 0/848511 Fax: 0/846387	Rexroth - Sigma S.A. Division INDRAMAT F-92632 Gennevilliers Cedex Parc des Barbanniers 4, Place du Village Phone: 1/41475430 Fax: 1/47946941	Rexroth - Sigma S.A. Division INDRAMAT F-69634 Venissieux - Cx 91, Bd 1 Joliot Curie Phone: 78785256 Fax: 78785231
France	**Italy**	**Italy**	**Netherlands**
Rexroth - Sigma S.A. Division INDRAMAT F-31100 Toulouse 270, Avenue de lardenne Phone: 61499519 Fax: 61310041	Rexroth S.p.A. Divisione INDRAMAT I-20063 Cernusco S/N.MI Via G. Di Vittoria, 1 Phone: 02/92365-270 Fax: 02/92108069	Rexroth S.p.A. Divisione INDRAMAT Via Borgomanero, 11 I-10145 Torino Phone: 011/7712230 Fax: 011/7710190	Hydraudyne Hydrauliek B.V. Kruisbroeksestraat 1a P.O. Box 32 NL-5280 AA Boxtel Phone: 04116/51951 Fax: 04116/51483
Spain	**Spain**	**Sweden**	**Switzerland**
Rexroth S.A. Centro Industrial Santiago Obradors s/n E-08130 Santa Perpetua de Mogoda (Barcelona) Phone: 03/718 68 51 Telex: 591 81 Fax: 03/718 98 62	Goimendi S.A. División Indramat Jolastokieta (Herrera) Apartado 11 37 San Sebastion, 20017 Phone: 043/40 01 63 Telex: 361 72 Fax: 043/39 93 95	AB Rexroth Mecman INDRAMAT Division Varuvägen 7 S-125 81 Stockholm Phone: 08/727 92 00 Fax: 08/64 73 277	Rexroth SA Département INDRAMAT Chemin de l`Ecole 6 CH-1036 Sullens Phone: 021/731 43 77 Fax: 021/731 46 78
Switzerland	**Russia**		
Rexroth AG Geeschäftsbereich INDRAMAT Gewerbestraße 3 CH-8500 Frauenfeld Phone: 052/720 21 00 Fax: 052/720 21 11	Tschudnenko E.B. Arsenia 22 153000 Ivanovo Rußland Phone: 093/22 39 633		

European customer service points without Germany

Outside Europe

Argentina Mannesmann Rexroth S.A.I.C. Division INDRAMAT Acassusso 48 41/7 1605 Munro (Buenos Aires) Argentina Phone: 01/756 01 40 01/756 02 40 Telex: 262 66 rexro ar Fax: 01/756 01 36	Argentina Nakase Asesoramiento Tecnico Diaz Velez 2929 1636 Olivos (Provincia de Buenos Aires) Argentina Argentina Telefon 01/790 52 30	Australia Australian Industrial Machenery Services Pty. Ltd. Unit 3/45 Horne ST Campbellfield VIC 2061 Australia Phone: 03/93 59 0228 Fax: 03/93 59 02886	Brazil Mannesmann Rexroth Automação Ltda. Divisão INDRAMAT Rua Georg Rexroth, 609 Vila Padre Anchieta BR-09.951-250 Diadema-SP Caixa Postal 377 BR-09.901-970 Diadema-SP Phone: 011/745 90 65 011/745 90 70 Fax: 011/745 90 50
Canada Basic Technologies Corporation Burlington Division 3426 Mainway Drive Burlington, Ontario Canada L7M 1A8 Phone: 905/335-55 11 Fax: 905/335-41 84	China Rexroth (China) Ldt. Shanghai Office Room 206 Shanghai Intern. Trade Centre 2200 Yanan Xi Lu Shanghai 200335 P.R. China Phone: 021/627 55 333 Fax: 021/627 55 666	China Rexroth (China) Ldt. Shanghai Parts & Service Centre 199 Wu Cao Road, Hua Cao Minhang District Shanghai 201 103 P.R. China Phone: 021/622 00 058 Fax: 021/622 00 068	China Rexroth (China) Ldt. 1430 China World Trade Centre 1, Jianguomenwai Avenue Beijing 100004 P.R. China Phone: 010/50 50 380 Fax: 010/50 50 379
China Rexroth (China) Ldt. A-5F., 123 Lian Shan Street Sha He Kou District Dalian 116 023 P.R. China Phone: 0411/46 78 930 Fax: 0411/46 78 932	Hong-Kong Rexroth (China) Ldt. 19 Cheung Shun Street 1st Floor, Cheung Sha Wan, Kowloon, Honkong Phone: 741 13 51/-54 und 741 14 30 Telex: 3346 17 GL REX HX Fax: 786 40 19 786 07 33	India Mannesmann Rexroth (India) Ltd. INDRAMAT Division Plot. 96, Phase III Peenya Industrial Area Bangalore - 560058 Phone: 80/839 21 01 80/839 73 74 Telex: 845 5028 RexB Fax: 80/839 43 45	Japan Rexroth Co., Ltd. INDRAMAT Division I.R. Building Nakamachidai 4-26-44 Tsuzuki-ku, Yokohama 226 Japan Phone: 045/942-72 10 Fax: 045/942-03 41
Korea Rexroth-Seki Co Ltd. 1500-12 Da-Dae-Dong Saha-Gu, Pusan, 604-050 Phone: 051/264 90 01 Fax: 051/264 90 10	Korea Seo Chang Corporation Ltd. Room 903, Jeail Building 44-35 Yoido-Dong Youngdeungpo-Ku Seoul, Korea Phone: 02/780-82 07 ~9 Fax: 02/784-54 08	Mexico Motorización y Diseño de Controles, S.A. de C.V. Av. Dr. Gustavo Baz No. 288 Col. Parque Industrial la loma Apartado Postal No. 318 54060 Tlalnepantla Estado de Mexico Phone: 5/397 86 44 Fax: 5/398 98 88	
USA Rexroth Corporation INDRAMAT Division 5150 Prairie Stone Parkway Hoffman Estates, Illinois 60192 Phone: 847/645-36 00 Fax: 857/645-62 01	USA Rexroth Corporation INDRAMAT Division 2110 Austin Avenue Rochester Hills, Michigan 48309 Phone: 810/853-82 90 Fax: 810/853-82 90		

Customer service points outside Europe

Notes


Indramat