

Description of PLC Examples for SINUMERIK 828D

Important:

The sample programs are freeware. Any user can use, copy and forward this program free of charge. The authors and owners of this program take no responsibility whatsoever for the functionality and compatibility of this software. Use of the software is at the user's own risk.

Since this software is free of charge, there is no warranty and no entitlement to error correction and hotline support.

General considerations when configuring a new PLC project

Use only PLC subroutines and data blocks from the same toolbox CD. Don't replace or add single subroutines or data blocks.

MCP 310 and MCP483 got different input / output images. Further is the arrangement of feed control different in milling and turning.

The symbol table IS_MCP in examples MCP_xx is adapted to the arrangement for the respective technology.

Please begin your project with the selection of the MCP and add in this project other subroutines.

All SBRs for magazine or tool management relate to first magazine, first tool holder and first loading point.

SBR 217 ALM control

PLC-controlled OFF1 for ALM (former X122.1).

SBR 218 Switch HMI Channel

Command interface can be used for selection of the active channel on HMI via PLC user program.

SBR 219 JOG USB MCP T

ax signals for turning machines, call ever after SBR USB_MCP

This subroutine needs data block DB_Common (DB9053). Data block DB9053 is contained in this project.

Please begin a new project with one MCP example. Symbol Table USB_MCP_Turn matches only for this example.

This example manages on base Manual PPU, chapter 7.2, signals NC-->PLC and PLC -->NC.

Signals:

ax selection

Feed and spindle control with override switch

Spindle start/stop (send to DB380x.DBB5006 and DB9053, interface is managed in SBR Axis_enable)

Direction keys with rapid traverse

SBR input Spindle1_Axis: enter here the machine axis number of the first spindle
(to manage the spindle override).

SBR 220 JOG USB MCP M

ax signals for milling machines, call ever after SBR USB_MCP

This subroutine needs data block DB_Common (DB9053). Data block DB9053 is contained in this project. Please begin a new project with one MCP example. Symbol Table USB_MCP_Mill matches only for this example.

This example manages on base Manual PPU, chapter 7.2, signals NC-->PLC and PLC -->NC.

Signals:

ax selection

Feed and spindle control with override switch

Spindle start/stop (send to DB380x.DBB5006 and DB9053, interface is managed in SBR Axis_enable)

Direction keys with rapid traverse

SBR input Spindle1_Axis: enter here the machine axis number of the first spindle
(to manage the spindle override).

SBR 221 USB MCP

This subroutine needs data block DB_Common (DB9053). Data block DB9053 is contained in this project. Please begin a new project with one MCP example. Symbol Table USB_MCP_xx matches only for this example.

This example manages on base Manual PPU, chapter 7.2, signals NC-->PLC and PLC -->NC.

Signals:

Operating Mode (Jog, Jog-Ref, MDA, Auto) and machine functions (Inc, Repos, Teach In)

Program control (NC Start, NC Stop, NC Reset, Single Block)

Key-operated switch

WCS/MCS

Customer key S7: by pressing the button it switch on/off. You can use the LED output (DB1100.DBX4.5) to control any machine function (on MCP 310 USB Coolant)

SBR 222 Run in Prog Control

This subroutine needs data block Prog_control (DB9051). Data block DB9051 is contained in this project.

SBR Run_in_Prog_Control manages the signals from HMI program control / Run In for two channels with master and counter spindle. Please check machine data 52206 \$MCS_AXIS_USAGE.

SBR 223 Log_out

SBR Log_out starts PI-service log out.

SBR 224 ASUP CALL 2C

This subroutine needs data block DB_ASUP (DB9060) from this projekt.

Please copy this data block into your project.

This example manages the user interface between NC and PLC for initialization and starting the PLCASUP1_SPF to PLCASUP4_SPF.

The interface and handshaking requirements are described in:

Function Manual Basic Functions Sinumerik 840Dsl/ 828D , chapter

14.6.2 Program instance services and

14.6.5 Starting ASUBs.

The example call from SBR ASUP occupied Alarm 229 till Alarm 236 (DB1600.DBX28.5 - DB1600.DBX29.4)

program flow:

The SBR ASUP contains for each PLC ASUP six input signals and five output signals.

Signal description for ASUP1 initialization:

ASUP1_initialize: starts the initialization via PI service. In the example call the initialization is started after the first PLC cycle.

ASUP1_CHAN: assignet the interrupt to channel 1 or 2

ASUP1_PRIO: assignet the priority level

ASUP1_LIFTFAST: when the interrupt signal is received a "fast retraction of the tool from the contour" will be performed before the start of the interrupt routine.

ASUP1_BLSYNC: when the interrupt signal is received the program block which is in progress will continue to be processed; only once this is complete will the interrupt routine be launched

ASUP1_ini_Error: Error in Job, initialization is failed

ASUP1_ini_Done: Job completed, ASUP1 is initialized

Signal description for ASUP1 start:

ASUP1_Start: starts PLCASUP1_SPF, please note the necessary conditions for start (see Function Manual chapter 14.6.5), example call encloses no interlock conditions!

ASUP1_Run_out: signal is 1 during PLCASUP1_SPF is running

ASUP1_Error_out: signal is 1 if ASUP cannot be executed or the ASUP is not initialized.

ASUP1_done_out: PLCASUP1 is finished without error, ASUP completed

same signals exist for PLCASUP2-4

SBR 225 JOG MCP483_2C

MCP483 for machines with two channels, call ever after SBR MCP483_2C.

This subroutine needs data block DB_Common (DB9053). Data block DB9053 is contained in this project. Please begin a new project with one MCP example. Symbol Table IS_MCP matches only for this example.

This example manages on base Manual PPU, chapter 7.1, signals NC-->PLC and PLC -->NC.

axis signals for the first nine machine axis

Signals:

Feed start/stop (send to DB9053, interface is managed in SBR Axis_enable)

Feed control with combined ax and direction key and override feed/rapid traverse

Spindle start/stop (send to DB9053, interface is managed in SBR Axis_enable)

Spindle control with override switch

rapid traverse key

SBR input Spindle1_Axis: enter here the machine axis number of the first spindle

(to manage the spindle override).

SBR 226 MCP483_2C

MCP483 for machines with two channels

This subroutine needs data block DB_Common (DB9053). Data block DB9053 is contained in this project.
Please begin a new project with one MCP example. Symbol Table IS_MCP matches only for this example.

This example manages on base Manual PPU, chapter 7.1, signals NC-->PLC and PLC -->NC.

Signals:

Operating Mode (Jog, Jog-Ref, MDA, Auto) and machine functions (Inc, Repos, Teach In) for mode group one
Program control (NC Start, NC Stop, NC Reset, Single Block) for the selected channel on HMI

Key-operated switch

WCS/MCS

Mode group reset (DB3000.DBX0.7) is not part of this example. Please use any customer key for this function.

SBR 228 Hand_wheel_2C

This example manages the interface between HMI and PLC to activate hand-wheel in WCS (DB3200.DBX100x.x)
or MCS (DB380x.DBB4) for two channels.

SBR Hand_wheel_2C doesn't support hand-wheel activation from HHU.

SBR 229 Hand-wheel HW_HHU_Mill

This example manages the interface between HMI and PLC to activate hand-wheel in WCS (DB3200.DBX100x.x)
or MCS (DB380x.DBB4) and jog keys on hand held unit (HHU).

Please see following conditions if you use a HHU:

The hand-wheel signals need to be taken to X143 of the PPU as first hand-wheel.

Keys for axis selection, jog, rapid movement and function keys needs to interfacing on X51, X52 and X55 on MCP
with standard addresses.

SIEMENS AG

The emergency and enabling buttons should be used in safety/enabling circuit of the machine.

Call SBR HW_HHU_Mill always after MCP and JOG_MCP SBR.

Function description:

Position 0 on axis selection key set HHU inactive. Hand-wheel functions may be selected on HMI.

Switch on axis Z...5 activates HHU. The selected axis can be displayed with PLC alarm (in example alarm 700236).

Machine function 1 INC is selected.

Rapid traverse button switch to machine function 100 INC. Other INC functions are not realized.

With traversing keys + / - the selected axis can be traversed continuously.

Switchover the hand-wheel for machine axis (MCS) or geometry axis (WCS) is achieved by switchover MCS/WCS on HMI.

Function keys F1-F3 are disposal for free using.

Axis selection, traversing keys and switchover MCS/WCS on MCP are blocked if HHU is active

SBR 230 TMM Cancel Job

This Subroutine manages cancel instruction to tool management and reset acknowledgment errors. SBR TMM_Cancel_Job is part of example TMM_Mill. If you use this SBR in other projects make sure that DB9902 TM_ACK has acknowledgment step 2 with status 3 or change the acknowledgment bits inside the SBR.

SBR 231 Emergency stop

This subroutine needs data block EMG_STOP (DB9052). Data block DB9052 is contained in this project.

Please copy this data block into your project.

CAUTION

Please check whether this subroutine complies with all the relevant safety requirements.

This subroutine handles the emergency stop sequence, as well as the power on and off sequence of the drive SINAMICS S120.

For detailed information about drive SINAMICS S120, please refer to SINAMICS S120 documents.

Signals:

Input E_KEY: PLC Input from emergency stop key

Input T_RDY_LM: PLC Input from line module X21 Pin 1

Input HWL_ON: any hardware limit is active

Input Axis_stopped: all axis stand still, can be used to make sure that spindle standstill before disable the drive system

Output OFF1: Output to set OFF1 and give pulse enable

Output OFF3: Output to set OFF3

Outputs E_Key_pressed, Drive_not_ready and HW_Limit_active: Outputs for set alarms.

SBR 232 SR_EMG_STOP

This subroutine needs data block EMG_STOP (DB9052). Data block DB9052 is contained in this project.
Please copy this data block into your project.

CAUTION

Please check whether this subroutine complies with all the relevant safety requirements.

This subroutine handles the emergency stop sequence by using a safety relay combination with undelayed and delayed relay contacts (hardware solution).

Only one emergency stop subroutine may be active in the OB1, either Emergency_stop or SR_EMG_STOP.

The spindle stop signal is not monitoring. The delayed time from the safety relay contact must have a value > drive stop time from the most powerful drive (e.g. time spindle stop time).

The SINAMICS controlled the line contactor via the Active Line Module.

By using a Smart Line Module without DRIVE CLiQ the line contactor is controlled by the safety relay.

Signals:

Input LM_RDY: Line module with DRIVE CLiQ 828D X132.8 ready to power up; by using line module without DRIVE CLiQ must be ONE

Input OFF3_K1_Cont: OFF3 quick stop from 828D X122.2

Input OFF1_K2_Cont: OFF1 from 828D X122.1

Input HWL_active: any of hardware limit switches is active (NO)

Input ON_Key_cust: Customer Key for flag ON safety relay

Input LM_Infp: infeed operation line module with (without line module must be one)

Input E_Key_cust: Emergency Stop key (NO)

Input LC_Replay_Cont: Replay from Line contactor

Output SR_ON: On flag for safety relay

Output OFF1_T: Terminal output to subroutine Axis_enable

Output OFF3_T: Terminal output

Output Drive_Power_Alarm: Alarm from Drive power up

Output EMG_button_pushed: Emergency stop button is pushed

Output LC_signal_error: No checkback signal from line contactor

SBR 233 Axis_enable

SBR Axis_enable matches to examples MCP_483/310... and Emergency_stop.

Signals OFF1, Feed Stop active and Spindle Stop active are taken from DB9052 and DB9053.

Please copy this data blocks into your project or use your own signals.

SBR manages ax enabling, feed stop /spindle stop for all axes and first spindle.

If more spindles exist please manage in a separate SBR.

SBR 234 Prog_control

This subroutine needs data block Prog_control (DB9051).

Please copy this data block into your project.

This subroutine manages program control signals from HMI to NC interface.

Signals:

Program test

Dry run

Programmed stop

Handwheel Offset

Skip Block

SBR 235 MCP_483

uses additional one SBR JOG_MCP483_x

This subroutine needs data block DB_Common (DB9053). Data block DB9053 is contained in this project.

Please begin a new project with one MCP example. Symbol Table IS_MCP matches only for this example.

This example manages on base Manual PPU, chapter 7.1, signals NC-->PLC and PLC -->NC.

Signals:

Operating Mode (Jog, Jog-Ref, MDA, Auto) and machine functions (Inc, Repos, Teach In)

Program control (NC Start, NC Stop, NC Reset, Single Block)

Key-operated switch

WCS/MCS

SBR 236 Jog MCP 483 M

MCP483 for milling machines

This subroutine needs data block DB_Common (DB9053). Data block DB9053 is contained in this project. Please begin a new project with one MCP example. Symbol Table IS_MCP matches only for this example.

This example manages on base Manual PPU, chapter 7.1, signals NC-->PLC and PLC -->NC.

Signals:

Feed start/stop (send to DB9053, interface is managed in SBR Axis_enable)

Feed control with override switch

Spindle start/stop (send to DB9053, interface is managed in SBR Axis_enable)

Spindle control with override switch

Direction keys with rapid traverse override

SBR input Spindle1_Axis: enter here the machine axis number of the first spindle (to manage the spindle override).

SBR 237 Jog MCP 483 T f

MCP483 for turning machines in front of turning center

This subroutine needs data block DB_Common (DB9053). Data block DB9053 is contained in this project. Please begin a new project with one MCP example. Symbol Table IS_MCP matches only for this example.

This example manages on base Manual PPU, chapter 7.1, signals NC-->PLC and PLC -->NC.

Signals:

Feed start/stop (send to DB9053, interface is managed in SBR Axis_enable)

Feed control with combined ax and direction key and override feed/rapid traverse

Spindle start/stop (send to DB9053, interface is managed in SBR Axis_enable)

Spindle control with override switch

rapid traverse key

SBR input Spindle1_Axis: enter here the machine axis number of the first spindle
(to manage the spindle override).

SBR 238 Jog MCP 483 T b

MCP483 for turning machines in behind the turning center

This subroutine needs data block DB_Common (DB9053). Data block DB9053 is contained in this project.
Please begin a new project with one MCP example. Symbol Table IS_MCP matches only for this example.

This example manages on base Manual PPU, chapter 7.1, signals NC-->PLC and PLC -->NC.

Signals:

Feed start/stop (send to DB9053, interface is managed in SBR Axis_enable)

Feed control with combined ax and direction key and override feed/rapid traverse

Spindle start/stop (send to DB9053, interface is managed in SBR Axis_enable)

Spindle control with override switch

rapid traverse key

SBR input Spindle1_Axis: enter here the machine axis number of the first spindle
(to manage the spindle override).

SBR 239 MCP 310

uses additional one SBR JOG_MCP310_x

This subroutine needs data block DB_Common (DB9053). Data block DB9053 is contained in this project.
Please begin a new project with one MCP example. Symbol Table IS_MCP matches only for this example.

This example manages on base Manual PPU, chapter 7.1, signals NC-->PLC and PLC -->NC.

Signals:

Operating Mode (Jog, Jog-Ref, MDA, Auto) and machine functions (Inc, Repos, Teach In)

Program control (NC Start, NC Stop, NC Reset, Single Block)

Key-operated switch

WCS/MCS

SBR 240 Jog MCP 310 M

MCP310 for milling machines

This subroutine needs data block DB_Common (DB9053). Data block DB9053 is contained in this project.

Please begin a new project with one MCP example. Symbol Table IS_MCP matches only for this example.

This example manages on base Manual PPU, chapter 7.2, signals NC-->PLC and PLC -->NC.

Signals:

Feed start/stop (send to DB9053, interface is managed in SBR Axis_enable)

Feed control with override switch

Spindle start/stop (send to DB380x.DBB5006 and DB9053, interface is managed in SBR Axis_enable)

Spindle control with toggle keys

Direction keys with rapid traverse override

SBR input Spindle1_Axis: enter here the machine axis number of the first spindle

(to manage the spindle override).

SBR 241 Jog MCP 310 T f

MCP310 for turning machines in front of turning center

This subroutine needs data block DB_Common (DB9053). Data block DB9053 is contained in this project.

Please begin a new project with one MCP example. Symbol Table IS_MCP matches only for this example.

This example manages on base Manual PPU, chapter 7.2, signals NC-->PLC and PLC -->NC.

Signals:

Feed start/stop (send to DB9053, interface is managed in SBR Axis_enable)

Feed control with combined ax and direction key and override feed/rapid traverse

Spindle start/stop (send to DB380x.DBB5006 and DB9053, interface is managed in SBR Axis_enable)

Spindle control with toggle keys

rapid traverse key

SBR input Spindle1_Axis: enter here the machine axis number of the first spindle
(to manage the spindle override).

SBR 242 Jog MCP 310 T b

MCP310 for turning machines in behind the turning center

This subroutine needs data block DB_Common (DB9053). Data block DB9053 is contained in this project.
Please begin a new project with one MCP example. Symbol Table IS_MCP matches only for this example.

This example manages on base Manual PPU, chapter 7.2, signals NC-->PLC and PLC -->NC.

Signals:

Feed start/stop (send to DB9053, interface is managed in SBR Axis_enable)

Feed control with combined ax and direction key and override feed/rapid traverse

Spindle start/stop (send to DB380x.DBB5006 and DB9053, interface is managed in SBR Axis_enable)

Spindle control with toggle keys

rapid traverse key

SBR input Spindle1_Axis: enter here the machine axis number of the first spindle
(to manage the spindle override).

SBR 243 TMM Mill

SBR Tool_Change_mill needs data blocks Toolchange(DB9056), TM_CTS(DB9900), TM_VTS(DB9901) and TM_ACK(DB9902) out this example.

This SBR match to flow chart and description of the procedure in commissioning manual chapter 9.8 Application example for milling machine...

The program steps in flow chart correspond to the labels in SBR Tool_Change_mill. The subroutine will set all necessary acknowledgments.

Please insert on inputs Step_x_done the signals, which reports the mechanical movements.

The acknowledgments will set with a positive transition on SBR input. SBR output Step_x_prepared means: start respective step.

In this example are the outputs Step_x_prepared direct wired to the inputs. This circuit makes a short run through the sequence.

SBR 243 TMM_Mill don't manages manual tools.

Signals:

Input Step₂_done till Step₁₄_done: wait for report step is done

Output Step: actual step number

Output Error_status: Feedback from DB4100.DBW104. see commissioning manual chapter 9.2.2

Output Ack_error: Signal from TMM to PLC acknowledgment error

Output Change_in_run: Tool change is running, on DB4300.DBX0.0 is a job pending

Output Step₄_prepared till Step₁₄_prepared: start respective step

SBR 244 Tool relocate

This subroutine needs data blocks DB TM_ACK (DB9902).

Please add this Data Block out Instructions/Libraries/Special Data Blocks with double click into your project.

SBR Tool_relocate manages the acknowledgement to tool management for load, unload, relocate and position magazine.

In DB9902 (TM_ACK) is as second acknowledgment step needed:

Transfer step number of the new tool = 0

Transfer step number of the old tool = 0

Status at NCK = 3

Please see second transfer step in this example.

Signals:

Input Job: job on interface from first loading point

Input Job_done: user program send to SBR job is done

Input Job_cancel: PLC-Input to cancel the job

Output Ack_ok: Signal from TMM to PLC acknowledgment ok

Output Ack_error: Signal from TMM to PLC acknowledgment error

Output Error_status: Feedback from DB4100.DBW104. see commissioning manual chapter 9.2.1

The acknowledgment from tool change is to manage in a separate SBR.

SBR 245 Mag_Dir

This subroutine needs data blocks MAG_DIR (DB9057).

Please copy this data block into your project.

SBR Mag_Dir calculates when a positioning of a new magazin place is required the pre-index position and the direction for short way for a turret or chain magazin.

Signals:

Input Num_place: number of magazine locations

Input Required_Place: target position in positioning job from TMM, you can use here DB9058.DBW4 from SBR Turret if this SBR is also used

Input current_Place: here is the actual location number as integer needed.

Input Calc_new: starts a new calculation

Output P_INDXo: location number one place before the target position is reached, can be used to start indexing process

Output DIR: direction for the short way to target position, = 1: for CW; 0: for CCW

Output Calc_is_done: signal to start magazine movement, if SBR Turret is used: connect to Rev_can_Start Input

SBR 246 Turret

This subroutine needs data blocks DB_Turret (DB9058).

Please copy this data block into your project. In DB9058.DBW0 have you set the number of magazine locations (is not provided in SBR).

DB9058.DBW2 must be provided with the actual location number from the magazine. DB9058.DBW4 is set in this SBR with the target position out of the TMM job.

SBR Turret controls by plc output controlled turret drive. In case the turret gets a request to turn from a tool management job and the enabling signal to start movement (Rev_can_Start) is true the output signal Turn_cw or Turn_ccw is true till target position is reached. The output signal In_Pos can be used to set the acknowledgement to tool management. With input Dir_short_way may be the direction get selected. Dir_short_way = 1 turns CW, Dir_short_way = 0 turns CCW. You can use to decide the direction the SBR MAG_DIR. The output DIR can provide input Dir_short_way from SBR Turret. Output Calc_is_done from MAG_DIR can provide the Rev_can_Start input from SBR Turret.

Input Jog_Rev_CW or Jog_Rev_CCW start manually revolver movement in case no request from TMM is pending.

Signals:

Input Rev_can_Start: magazine is ready to start positioning, direction was judged

Input Jog_Rev_CW and Jog_Rev_CCW are PLC Inputs for manually movement

Input Dir_short_way: = 1: for CW; = 0: for CCW, can get calculated from SBR MAG_DIR

Input Curr_Place: the SBR needs here the information about the actual location number as integer

Output TMM_new_pos_req: there is an new job from TMM with positioning request, signal to start in SBR MAG_DIR a new calculation

Output Turn_cw: output to plc, turn magazine clockwise

Output Turn_ccw: output to PLC, turn magazine counterclockwise

Output In_Pos: End position out TMM job is reached, with this signal you can set acknowledgement to TMM

SBR 247 Hand-wheel

This example manages the interface between HMI and PLC to activate hand-wheel in WCS (DB3200.DBX100x.x) or MCS (DB380x.DBB4).

SBR Handweel doesn't support hand-wheel activation from HHU.

SBR248 Star_Delta

This building block is at use SW4.5SP1 !

This subroutine needs data block Val_Star_Delta (DB9059).

Please copy this data block into your project.

He automatically recognizes from new DB390x.dbb4008 the availed MDS.

Analog FC17 switches he only 2 MDS. (Star /Delta).

With MDS>2 must the OEM manually switching.

The building block serves to case shift of the Contactor for Star"Y" - Delta motors and the DDS.

Signals:

Spindel_No: Here will the ax number for the spindle entered.

Standard for Turning is "3"

Standard for Milling is "4"

TimerVal: Here will the delay time between disconnection "Y" and connection "Delta" in "ms" entered.

Typing "0" are 100ms activates

YDelta: The switchover is done with the signal "YDelta". Standardsignal is "DB390[x].4002.5

speed threshold 2 (p2155.[0-15]) and

hysteresis speed 2 (p2140.[0-15])

No_Param: "0" is standard ---->DB380[x].4001.3 motor_A

Y: output for contactor of Star

Delta: output for contactor of delta

SBR 249 ASUP

This subroutine needs data block DB_ASUP (DB9060).

Please copy this data block into your project.

This example manages the user interface between NC and PLC for initialization and starting the PLCASUP1_SPF and PLCASUP2_SPF.

The interface and handshaking requirements are described in:

Function Manual Basic Functions Sinumerik 840Dsl/ 828D , chapter

14.6.2 Program instance services and

14.6.5 Starting ASUBs.

The example call from SBR ASUP_INTERFACE occupied Alarm 243 till Alarm 246 (DB1600.DBX30.2 - DB1600.DBX30.5)

program flow:

The SBR ASUP_INTERFACE contains for each PLC ASUP two input signals and five output signals.

Signal description for ASUP1 initialization:

SIEMENS AG

ASUP1_initialize: starts the initialization via PI service. In the example call the initialization is started after the first PLC cycle.

ASUP1_ini_Error: Error in Job, initialization is failed

ASUP1_ini_Done: Job completed, ASUP1 is initialized

Signal description for ASUP1 start:

ASUP1_Start: starts PLCASUP1_SPF, please note the necessary conditions for start

(see Function Manual chapter 14.6.5), example call encloses no interlock conditions!

ASUP1_Run_out: signal is 1 during PLCASUP1_SPF is running

ASUP1_Error_out: signal is 1 if ASUP cannot be executed or the ASUP is not initialized.

ASUP1_done_out: PLCASUP1 is finished without error, ASUP completed

same signals exist for PLCASUP2

SBR 250 Service Planner

This subroutine needs data blocks SP_INI(DB9903) and SP_ACT(DB9904).

Please add this Data Blocks out Instructions/Libraries/Special Data Blocks with double click into your project.

This example manages the user interface between HMI and PLC for service planner.

The interface and handshaking requirements are described in:

Commissioning Manual Sinumerik 828D Turning and Milling, chapter 8.1 Service Planner.

SBR Service_Planner occupied Alarm 100 till Alarm 163 (DB1600.DBX12.3 - DB1600.DBX20.2)

SBR 251 Easy Extend

This subroutine needs data block EE_IFC(DB9905).

Please add this Data Block out Instructions/Libraries/Special Data Blocks with double click into your project.

Further are user Data Blocks EE_IFC_Passwd_Option_1 and EE_IFC_Passwd_Option_2 necessary.

Please copy this data blocks out this project into your project.

SIEMENS AG

This example manages the communication between HMI and PLC for easy extend script. The SBR supports the password tag.

The example matches to easy extend configuration as:

```
<AGM>
<!-- Option MD -->
<OPTION_MD name = "plc/db9061.dbw0" index="0" />
<OPTION_MD name = "plc/db9061.dbw2" index="1" />
<OPTION_MD name = "plc/db9061.dbw4" index="2" />
<OPTION_MD name = "plc/db9061.dbw6" index="3" />
....
....
<DEVICE>
<list_id>1</list_id>
....
<!-- Password input required to protect Option -->
<password refvar="plc/db9062.dbd0" />
....
....
<DEVICE>
<list_id>2</list_id>
....
<!-- Password input required to protect Option -->
<password refvar="plc/db9062.dbd8" />
....
....
<SET_INACTIVE>
....
<!-- Delete saved Password -->
<DATA name = "plc/db9061.dbx8.0"> 1 </DATA>
....
....
```

See also the structure of DB9061 and DB9062.

You find for every device one network.

If you want to define a numerical password, change in first row the default password 123456.

The network compares on HMI supplied password with the value in first row and set in case of correct input the option bit in DB9061.

The dialog between HMI and PLC for set active and set inactive is without any condition realised. PLC gives the data handshaking requirements only back to HMI.

You can add there your own logic if necessary.

With DB9061.DBX8.0 and up you can delete the saved password if you set the associated bits from easy extend script.

SBR 252 Write NC Parameter

This subroutine needs data block NC_Parameter (DB9063).

Please copy this data block into your project.

The PLC user program can read or write a maximum of eight variables from the NCK area via the PLC/NCK interface "Read/write NC variable".

This example writes one NC-variable (parameter R0=99)

You find a description in: Function Manual Basic Functions / 14.6.1 Read/write NC variables

SBR 253 Read NC Parameter

This subroutine needs data block NC_Parameter (DB9063).

Please copy this data block into your project.

The PLC user program can read or write a maximum of eight variables from the NCK area via the PLC/NCK interface "Read/write NC variable".

This example reads one NC-variable (parameter R0)

You find a description in: Function Manual Basic Functions / 14.6.1 Read/write NC variables

Signals:

SIEMENS AG

Input Read_word: Reading Data is Word and not REAL

Output Done: Job reading NC Parameter is done without error, reading data are valid, wait with next call from SBR253 till this signal is true.

Output Error: Error in Job

SBR 254 Pos PLC Axis

Description from PLC axis control in: Function Manual Base Functions, chapter 14.6.4 PLC axis control

SBR 255 Turn PLC Spindle

Description from PLC axis control in: Function Manual Base Functions, chapter 14.6.4 PLC axis control

Subroutine Numbers and occupied memory

SBR Nummer	Name	DB	Timer	Alarm
255	Turn_PLC_Spindle			
254	Pos_PLC_Axis			
253	Read_NC_Parameter	9063		248
252	Write_NC_Parameter	9063		247
251	Easy_Extend	9062 9061 9905		
250	Service_Planner	9903 9904		100-163
249	ASUP	9060		246-243
248	Star_Delta	9059	127	
247	Hand-wheel			
246	Turret	9058		
245	Mag_Dir	9057		
244	Tool_relocate	9902		242
243	TMM_Mill	9056 9900 9901 9902		241
242	Jog_MCP_310_T_b	9053		
241	Jog_MCP_310_T_f	9053		
240	Jog_MCP_310_M	9053		
239	MCP_310	9053		
238	Jog_MCP_483_T_b	9053		
237	Jog_MCP_483_T_f	9053		
236	Jog_MCP_483_M	9053		
235	MCP_483	9053		
234	Prog_control	9051		
233	Axis_Enable	9053 9052	125	
232	SR_EMG_STOP	9052	126	239 238 237
231	Emergency_stop	9052		240 239 238
230	TMM_Cancel_Job	9902		
229	Hand-wheel_HHU_Mill	9053	124	237
228	Hand-wheel_2C			
226	MCP483_2C	9053		
225	Jog_MCP483_2C	9053		
224	ASUP_CALL_2C	9060		229-236
223	Log_out			
222	Run_In_Prog_control	9051		
221	USB_MCP	9053		
220	JOG_USB_MCP_M	9053		
219	JOG_USB_MCP_T	9053		
218	Switch_HMI_Channel			228

217	ALMcontrol			
-----	------------	--	--	--