Positing of a SINAMICS S110 with S7-1200 via USS protocol

with S7-1200, SINAMICS S110 and KTP600

Configuration Example x9 • December 2009

Applications & Tools

Answers for industry.

SIEMENS

Industry Automation and Drives Technologies Service & Support Portal

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

http://support.automation.siemens.com/WW/view/en/40616058

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

online-support.automation@siemens.com

SIEMENS

SIMATIC

CE-X9 - Positioning of a SINAMICS S110 with S7-1200 via USS Protocol

Warranty and Liability

Note

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

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

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

Table of Contents

1 Automation Task

1.1 Application automation task

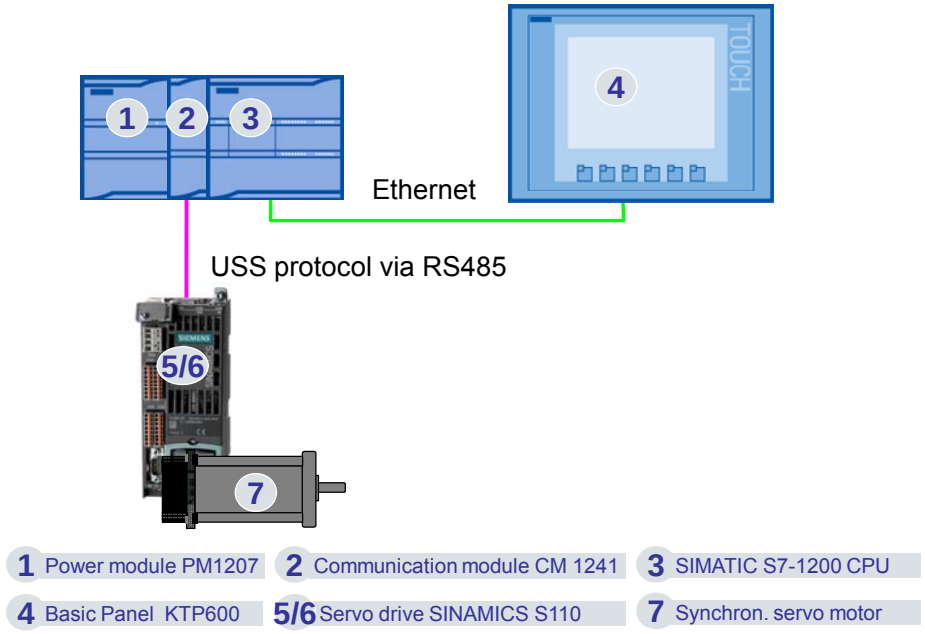
A synchronous motor that is connected with a servo drive type SINAMICS S110 is to be positioned via the CM1241 (RS485) communication module of a S7-1200 controller (CPU1214C).

Communication between controller and servo drive will be performed by USS protocol.

Operation and visualization takes place via a KTP600 touch panel which is to be connected via an Ethernet connection to the S7-1200 controller.

STEP7 Basic v10.5 is used as configuration tool for the control program and the HMI. The servo drive is configured using the STARTER startup tool.

Figure 1-1



The task consists of the following scenarios:

- moving the motor in jog mode
- referencing
- absolute and relative positioning
- moving the motor according to defined motion profiles

In order to position the motor, setpoint position and the setpoint velocity have to be transferred via the USS protocol to SINAMICS S110, as well as various command bits. SINAMICS S110 will then independently control the positioning or moving of the motor. The SINAMICS S110 will give a feedback to the controller on the various status bits, the actual position the actual velocity and also of error messages.

1.2 Component list

Products

Table 1-1

	Components	Qty	MLFB / Order number	Note
1.	PM1207 Power supply	1	6EP1332-1SH71	
2.	RS485 communication module CM1241	1	6ES7241-1CH30-0XB0	
3.	S7-1200 CPU1214C	1	6ES7214-1AE30-0XB0	DC/DC/DC
4.	Basic KTP600 panel (color, PN)	1	6AV6647-0AD11-3AX0	optional
5.	Power module PM340	1	6SL3210-1SB12-3AA0	230V
6.	Control unit CU305 DP	1	6SL3040-0JA00-0AA0	
7.	Synchron servo motor 1FK7	1	1FK7032-5AF21-1UA0	DRIVE-CLiQ
8.	SINAMICS S110 MMC incl. firmware v4.3 and licencing	1	6SL3054-4ED00-0AA0	Optional, if CU305 already existed with old firmware

Note A KTP600 is not absolutely necessary. To simulate the user interface, PC runtime from STEP7 Basic can be used.

Accessories

Table 1-2

	Component	Qty	MLFB / Order number	Note
9.	Power cable	1	6FX5002-5CG01-1AB0	
10.	Signal line DRIVE-CLiQ	1	6FX5002-2DC00-1AB0	
11.	Profibus cable	1	6XV1830-0EH10	
12.	Profibus connector with PG port	2	6ES7972-0BB12-0XA0	
13.	Commutation inductor	1	6SE6400-3CC00-4AB3	Optional
14.	Ethernet connection cables between KTP600, S7-1200 CPU and PC	1	6XV1870-3QH20	
15.	Serial null modem cable to commission the SINAMICS S110	1	Specialist dealer	RS232 (pin 2 and 3 rotated)
16.	Limit switch (make contact)	2	Specialist dealer	Mechanically operated
17.	Reference end-position switch	1	Specialist dealer	Inductive

Note The configuration, as it is, is intended for industrial application. For energy supply, industrial networks are usually implemented. It is therefore not necessary to use special filters/inductors with low leakage currents.
If the configuration is used in sensible electricity networks (e.g. PCs on the same network), filters or inductors should be used.

More information on the SINAMICS S110 can be found under:
http://www.siemens.com/sinamics-s110

Programming package

Table 1-3

	Component	Qty	MLFB / Order number	Note
18.	STEP 7 Basic V10.5	1	6ES7 822-0AA00-0YA0	
19.	STARTER startup tool on DVD	1	6SL3072-0AA00-0AG0	As of version 4.1.5 for firmware v4.3

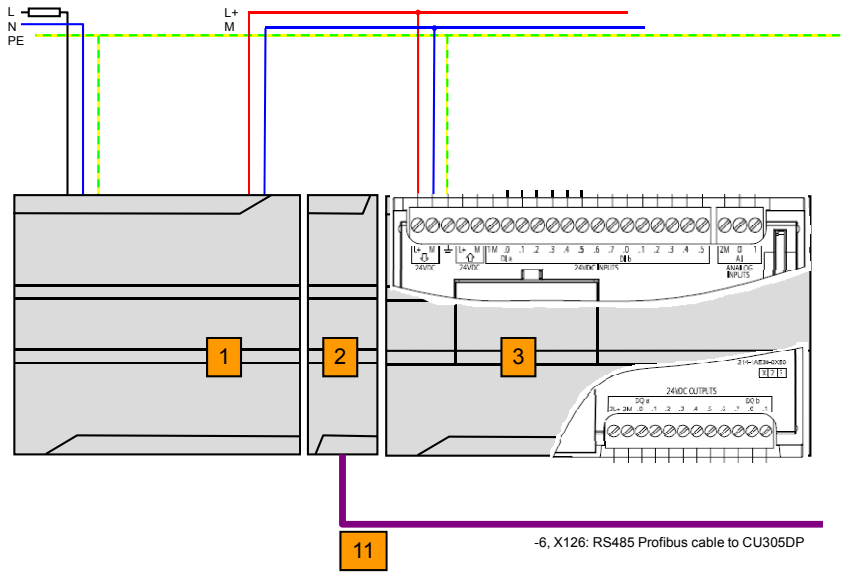
Note The current STARTER version can be downloaded here:
http://support.automation.siemens.com/WW/view/en/26233208

2 Automation Solution

2.1 Connection diagram

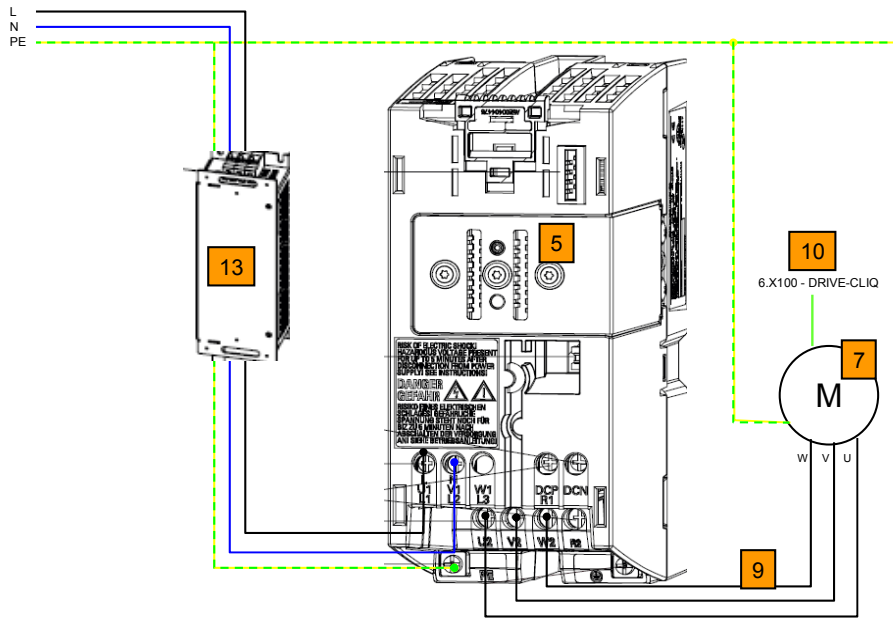
S7-1200 PM1270 + CPU1214C

Figure 2-1



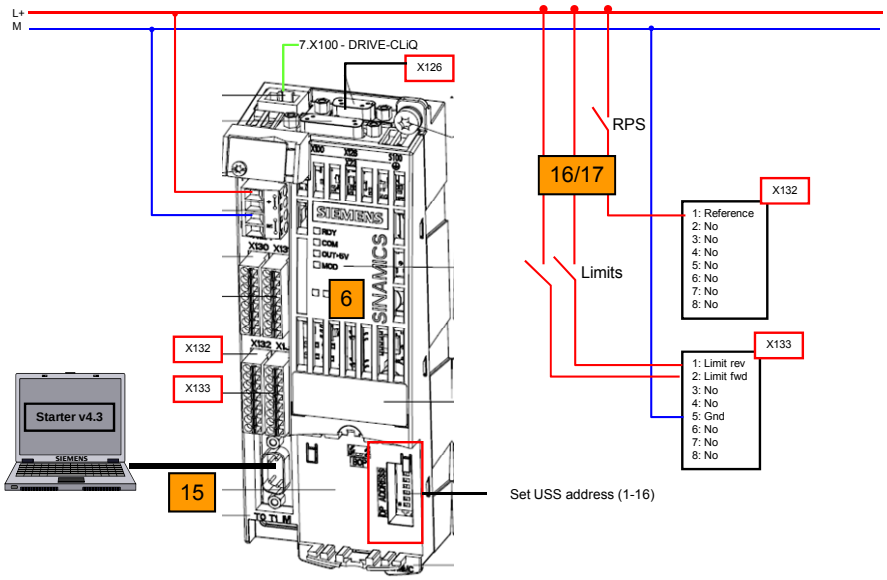
PM340

Figure 2-2



CU305DP

Figure 2-3



Note

Please observe all valid safety regulation and pay attention to the instructions from the handbook when connecting the AC 230V power supply of the SINAMICS S110.

http://support.automation.siemens.com/WW/view/en/33940571

ATTENTION

Notes on preventing electromagnetic interference:

- Make sure a good conductive connection between the servo drive and the (grounded) metal mounting plate is provided.
- Ensure all devices in the cabinet are earthed using short earthing lines with a large diameter and that they are connected to a common earthing point or earthing bar.
- Ensure that the S7-1200 CM connected to the frequency converter is connected to the same earthing or earthing point as the servo drive using a short line with a large diameter.
- Please use shielded control lines, e.g. a SIEMENS Profibus cable for setting up the RS485 bus.
- Run control lines as far separated from power cables in separate installation channels as possible. Crossings between power and control lines should be at a 90° angle.
- Connect the protective conductor of the motor to the earth connection (PE) of the respective servo drive.
- The line ends should be properly terminated, making sure that unshielded lines are kept as short as possible.

Use shielded lines for motor connections; earth the shielding both on the converter and the motor side using cable clamps.

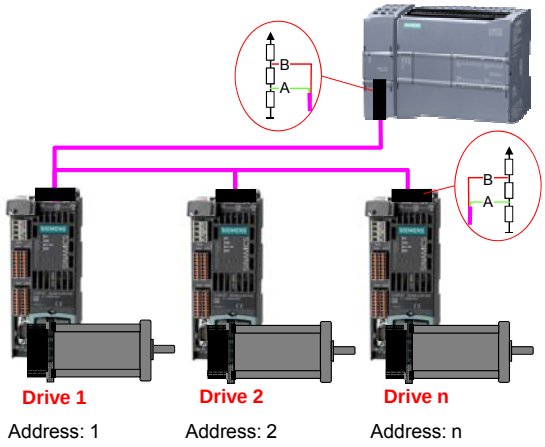
2.2 Addressing of drives and terminating a RS485 bus with USS protocol

A RS485 bus makes it possible to transmit data using the USS protocol via a 2-wire connection between a master (e.g. CPU 1214C) and up to 16 slaves (e.g. SINAMICS G110) per communication module. It is necessary to identify each slave via a unique address between 1 and 16.

The USS protocol allows only one master which does not require an assigned address.

In order to avoid reflections at the bus start or end, which may cause a falsified data signal, the bus must be closed off with terminating resistors as illustrated in Figure 2-4.

Figure 2-4



In this case, as illustrated in Figure 2-5, this is done bilaterally via the switch on the PROFIBUS connector.

Figure 2-5

2.3 Structure the USS protocol

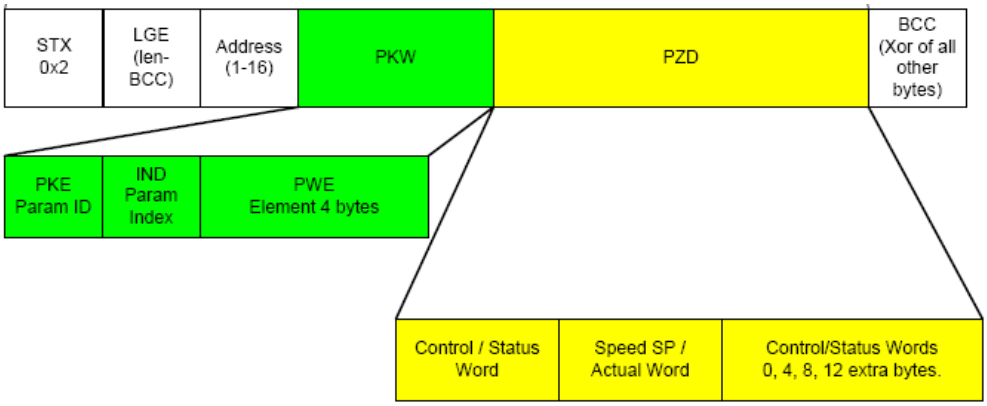
The USS protocol was developed in order to exchange process data between a central controller and bus stations (below called drive) on a RS485 bus. Each drive is identified via a unique bus address.

Even if PROFIBUS uses the same physical RS485 technology, PROFIBUS and USS protocol differ considerably.

A USS message consists of (Figure 2-6):

- STX: Start text
- LGE: Length of message
- ADR Slave address
- PKW: Parameter identification value
- PZD: Process data
- BCC: Control block (check total)

Figure 2-6



The PZD component transfers control commands and setpoints to the drive. The drive replies with status information and actual values. By default the control or status word is in the 1st PZD word. The 2nd PZD word has the main setpoint or the actual value. Another 6 words (12 bytes) are freely available. Therefore the PZD length can vary between 2, 4, 6 or 8. The PZD length on the drive and in the controller has to match.

The PKW component is used to read or write the parameter values in the drive. This makes it possible to change or read the individual parameters in the drive during runtime. The PKW component consists of the parts:

- PKE: parameter identification
- IND: parameter index
- PWE: parameter value

PKE and IND are each 1 word. The length of PWE can vary between 1 or 2 words, depending on what data type is to be transferred (word, double word, real). The overall length of PKW can be adjusted for most drives and has to be set to a fixed length of 4 words for the communication with a S7-1200. This results in the PWE component being 2 words.

Note Further information on the USS protocol can be found under the following link:

http://support.automation.siemens.com/WW/view/en/24178253

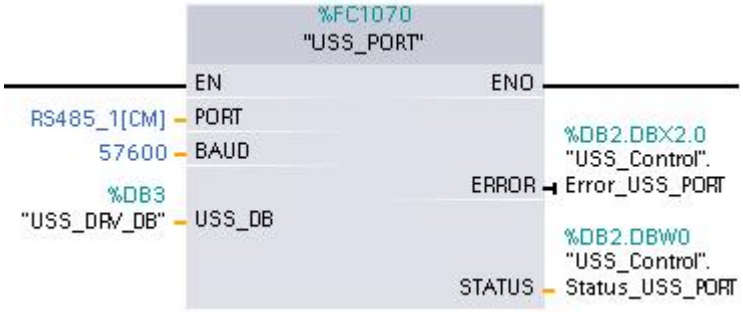
2.4 Communication with the drive

The S7-1200 controller communicates with the drive via the communication module by integrating the STEP7 Basic library, included in delivery.

Communicating with USS_PORT

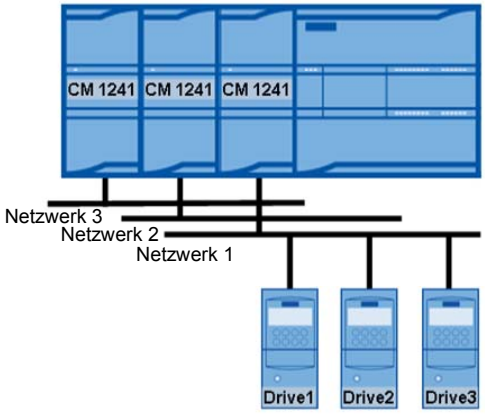
For a command to be sent from control to the drive, a function is necessary that controls the communication between CPU and the drive via the PtP communication module. This is achieved using the “USS_PORT” block (Figure 2-7).

Figure 2-7



The “Port” block parameter specifies the communication module via which the drives are connected. Per communication module a maximum of 16 drives can be operated. Since the S7-1200 supports a maximum of 3 communication modules, it is possible to connect up to 48 drives in 3 different networks.

Figure 2-8



At each call of the block, the communication with a drive is processed. Communication with the drive is asynchronous. This means the S7-1200 controller runs through several cycles before the data exchange with a drive is completed.

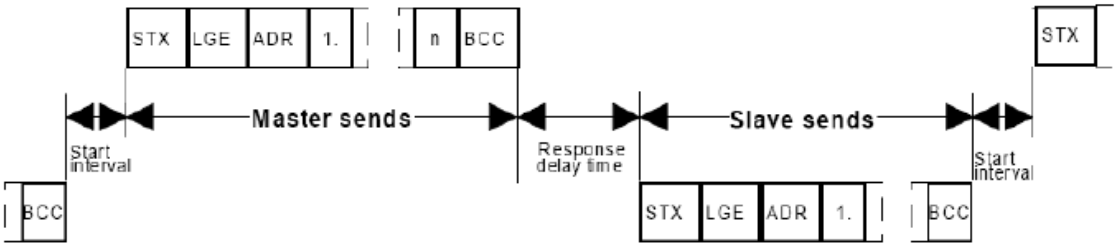
This is why the “USS_PORT” block is usually called out of a delay interrupt OB with a defined time interval. The call interval corresponds with the time which is necessary for a transaction with a drive.

The block can also be called up cyclically; however, this does not increase the number of transactions. If there is a call whilst the block is still active, the call will be ignored. At a high cycle time the interval between the calls increases and communication may break down.

Depending on the baud rate there is a “Worst Case Message Time” (WCMT) per transaction, meaning the time which may be needed for a transaction in the worst case. It is available in Table 2-1. It is made up of the length of a send and receive message as well as the corresponding waiting times (Figure 2-9). The individual profiles are listed below:

- The start interval is the time which has to pass before the USS master is allowed to send a request (formula: (2*11) / baud rate [bits per second])
- The request message of the master
- The maximum response delay time of 20ms
- The response message of the slave

Figure 2-9



By default, the library for the USS protocol carries out up to two repetitions per transaction. This results in a minimum call interval for USS_PORT which can be calculated according to the following formula:

Minimum call interval USS_PORT [ms] = 2 * WCMT

Furthermore, it has to be ensured that the “USS_PORT” block will be called within the timeout interval of the drive. The timeout interval of a drive is the time available for a transaction when due to a communication error, the transaction needs 3 attempts to complete the transaction. To calculate the timeout interval of a drive the following formula is used:

Timeout intervall per drive [ms] = 3 repetitions * minimum call interval USS_PORT

If several drives are located in the network, the number of drives in the network has to be multiplied in order to account for the timeout interval of the drives. The timeout interval per drive increases by this value calculated.

Timeout-Intervall per drive [ms] =
(3 repetitions * minimum call interval USS_PORT) *
Number of drives in the network

Table 2-1

Baud rate	WCMT [ms]	Minimum call - USS_PORT [ms] interval	Timeout interval per drive [ms]
1200	405.00	790	2370
2400	212.50	405	1215
4800	116.25	213	639
9600	68.13	117	351
19200	44.06	69	207
38400	32.03	45	135
57600	28.02	37	111
115200	24.01	29	87

Model calculation:

There are 2 drives in the network. The transmission speed is 57600 baud.

- Minimum call interval USS_PORT = (2 * 28.02) = 37ms
- Timeout interval per drive = (3 * 37) * 2 = 222ms

Result:

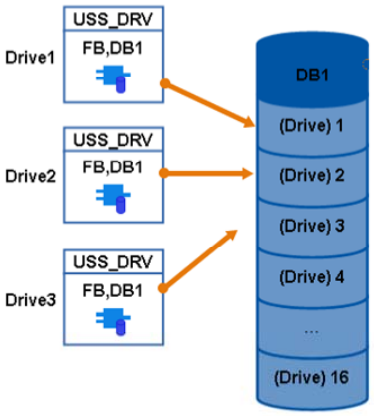
- The delay interrupt OB, in which the USS_Port block is called, has to be configured at an interval of min. 37s.
- The communication monitoring on the drive has to be configured with at at least 222ms, better at 230ms.

Transferring and reading process data on the drives with USS_DRV

The “USS_DRV” block exchanges data with the drives by creating request messages and evaluating answer messages. The instance data block is used as storage for the data. For each drive in the network a separate USS_DRV block has to be used. In the process, up to 16 called USS_DRV blocks share one and the same instance data block.

Once the first USS_DRV block was inserted in the STEP7 Basic editor, the instance data block is created automatically. The same instance data block has to be specified for all other USS_DRV blocks.

Figure 2-10



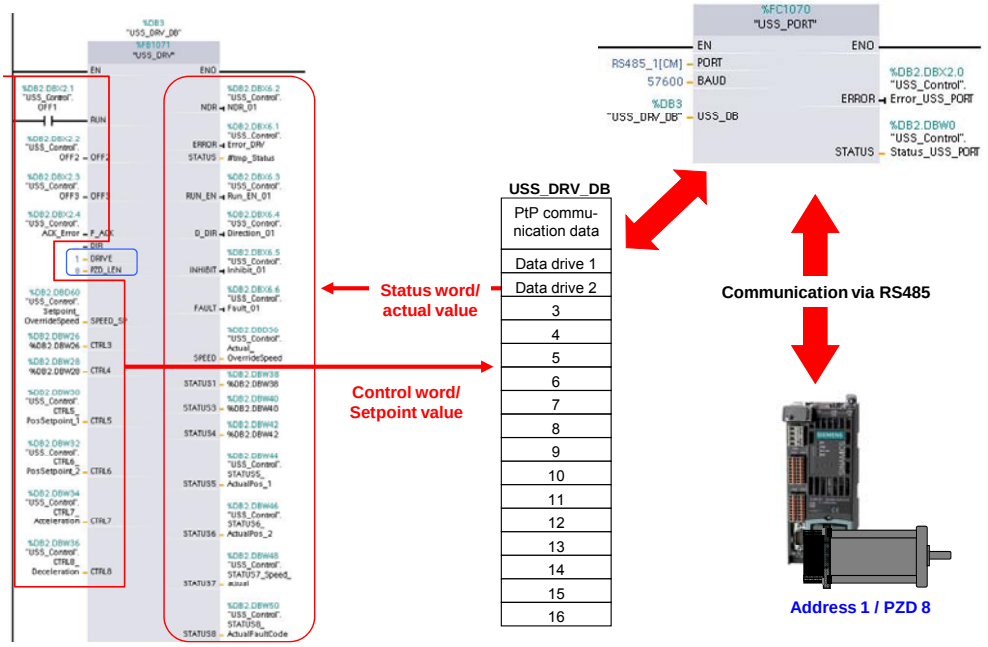
The call of the USS_DRV block has to be cyclically. When the first of execution of the block is performed, the drive indicated on the “Drive” parameter is initialized in the instance data block. Only after this initialization can the USS_PORT start communicating with this drive. This is why it is essential that the USS_DRV block is called at least once per drive. If the drive number is changed during runtime, first of all the instance data block needs to be reinitialized by setting the controller to STOP and then again to RUN.

Using the parameters on the left side of the USS_DRV block, the control word (RUN, OFF2, OFF3, F_ACK, DIR) and the main setpoint (SPEED_SP) of the assigned drive are configured. CTRL3 – CTRL8 constitute the freely available process data words of the send message. These configured parameters are stored in the send buffer of the instance data block.

The status word (STATUS1) and the actual value (SPEED) of the drive are read from a previously valid response buffer and are provided at the outputs of the USS_DRV block. STATUS3 – STATUS8 constitute the freely available process data words of the response message. The individual bits RUN_EN, D_DIR, INHIBIT and FAULT are a selection from the 1st status word.

During the execution of USS_DRV no data transmission takes place. Communication with the drives only takes place after the execution of USS_PORT. USS_DRV only configures the messages to be sent and evaluates data which was previously received via USS_PORT. (Figure 2-11)

Figure 2-11



Evaluation of communication errors

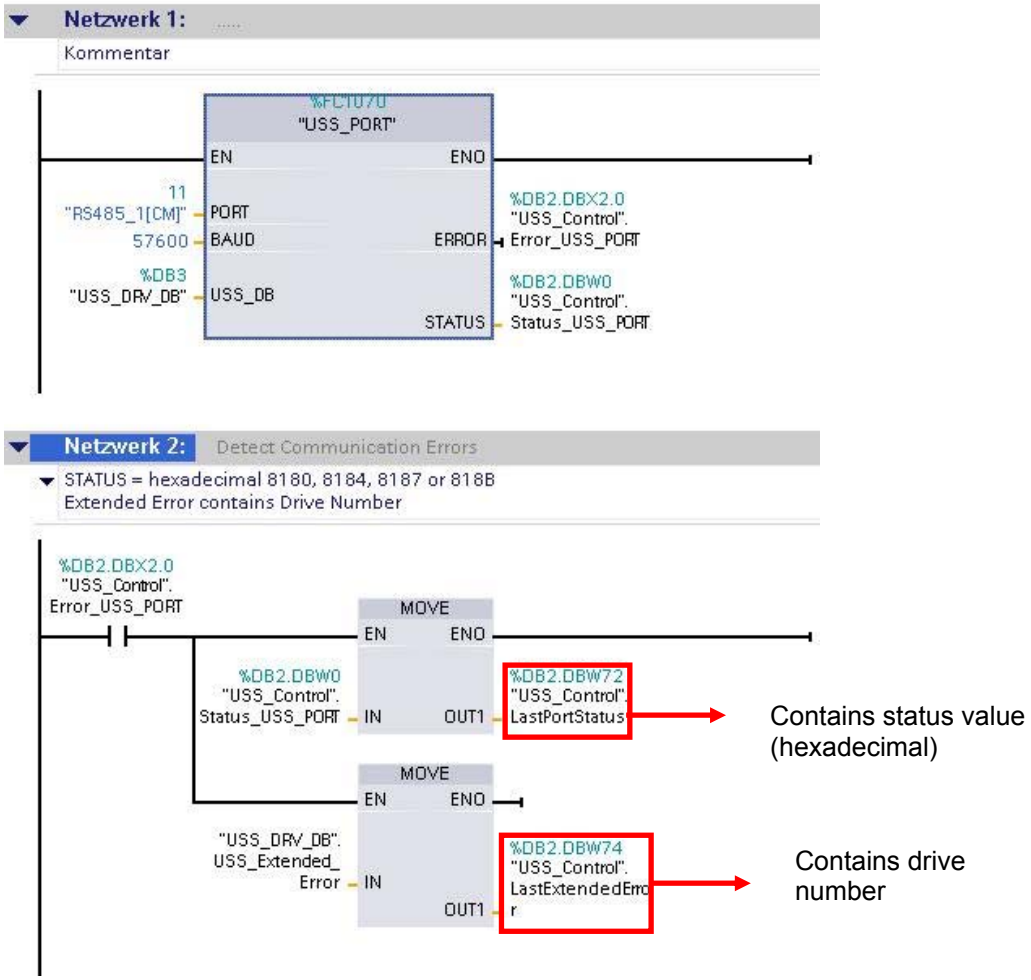
Communication errors are only output on the USS_PORT block and not on USS_DRV. They have the status value hex 8180, 8184, 8187 or 818B.

To be able to find out which drive gave the error message, there is a variable with the name “USS_Extended_Error” in the instance data block. In the case of communication errors the address of the faulty drive will be stored in this variable.

Since the status message is always only pending for the period of one cycle on the output of USS_PORT in the case of error, it has to be stored separately when an error occurs (Figure 2-12).

An overview of the status messages can be found in the S7-1200 System Manual, chapter 6.3.1.

Figure 2-12



2.5 Usage and configuration of all available control and status words

SINAMICS S110 offers the powerful “Basic positioner” (EPOS) function module. Using EPOS it is possible to fulfill all requirements of this configuration example. To be able to control EPOS with a S7-1200, a multitude of control and status signals is needed which can be transferred to the drive via the USS protocol or received by the drive via the USS protocol. The drive is configured in a way so that full functionality of the EPOS can be realized with a minimum number of process data words. As a result there are overall 8 process data words whose configuration on the USS_DRV block is described below.

Note

Before you deal with the configuration of the USS_DRV block it is recommended to get information on the functionality of the Basic positionier.

Sinamics S110 Function Manual (Chapter 7.3.6)

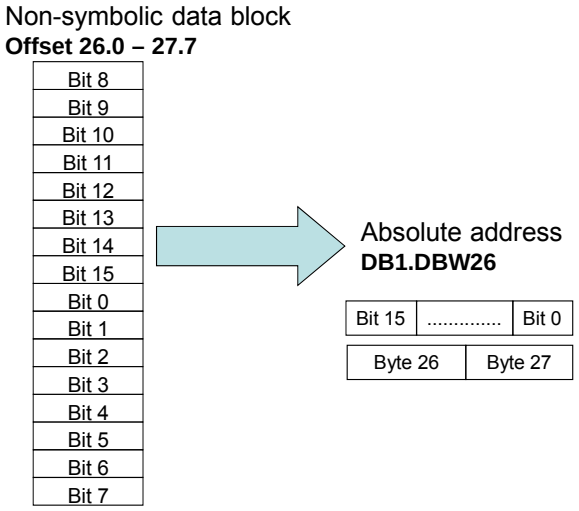
Configuration of USS_DRV block for 8 PZD words

Usual speed-controlled applications such as described in the Configuration Example CE-X11, manage with 2 PZD words. The USS_DRV block is conveniently designed for this kind of application. For the use of other PZD words the inputs CTRL3 – CTRL8 and STATUS3 – STATUS 8 are available. The length of the PZD words used is specified at the PZD_LEN input and can take on the values 2, 4, 6 and 8 (s. chapter 2.2).

The expanded input/outputs require the specification of variables of the data type word (16 bit). However, the majority of the values to be transferred consist of individual bits, double words or real numbers (32 bit).

For this reason it is necessary to insert the individual bits in the data block and to specify them with absolute addressing on the USS_DRV block. Please note that the lower-order bits 0-7 are located in the right part of the word and the higher-order bits 8-15 in the left part. In order to be able to address values from a data block absolutely, the data block must be declared “non-symbolic”. Figure 2-13 illustrates this.

Figure 2-13



32 bit values of the double word or real data type have to processed in a way so that the can be divided to two successive words whilst taking the lower-order and higher-order parts into account. This is the only way they can be correctly configured by the drive.

Note In the control program individual function blocks were created for the processing of PZD words which are classified in the “USS_Helpers” program group.

Overview of control signals

Control signals are known as signals that are transferred from the controller to the drive. The assignment of the individual PZD words is as shown below.

Table 2-2

PZD	Assignment of process data
PZD1	Control word 1
PZD2	Speed override (setpoint) – 0-100% (4000HEX = 100%)
PZD3	Control word 3
PZD4	Control word 2
PZD5	Position setpoint in [Unit of length LU, e.g. mm] for relative/absolute positioning
PZD6	
PZD7	Acceleration override 0-100%
PZD8	Deceleration override 0-100%

Assignment control word 1 – PZD1

Table 2-3

Bit	Name	Designation
0	RUN (OFF1)	ON - command 0 = OFF1 active 1 = ON OFF2 and OFF3 have to be set to “1” first before the ON command can be set.
1	OFF2	OFF2 – command 0 = OFF2 active 1 = no coasting active
2	OFF3	OFF3 – command 0 = OFF3 active 1 = no fast stop active
3	ENC	Enable operation (automatically set)
4	- not assigned -	
5	- not assigned -	
6	- not assigned -	
7	F_ACK (ACK_Error)	Acknowledging the error
8	- not assigned -	
9	- not assigned -	
10	LB	Life bit (PLC requests control)
11	- not assigned -	
12	- not assigned -	
13	- not assigned -	
14	- not assigned -	
15	- not assigned -	

Assignment control word 2 – PZD4

Table 2-4

Bit	Name	Designation
0	RefStart	Start referencing
1	RefPSet	Set reference point
2	RefTyp	Selection reference type 0 = reference point approach 1 = flying referencing
3	RefStDi	Reference point approach, start direction 0 = positive start direction 1 = negative start direction
4	- not assigned -	
5	- not assigned -	
6	- not assigned -	
7	- not assigned -	
8	MdiStart	Start MDI mode / start direct setpoint specification

Bit	Name	Designation
9	MdiSetup	MDI – Selection set MDI mode 0 = positioning 1 = set up
10	MdiPsTy	MDI – positioning type 0 = relative positioning 1 = absolute positioning
11	MdiPosDir	MDI – selection of direction for setup or Absolute positioning of round axes, in positive direction
12	MdiNegDir	MDI – selection of direction for setup or absolute positioning of round axes, in negative direction
13	MdiEdge	MDI – transfer of setpoint by positive edge if MdiTrTyp = 0
14	MdiTrTyp	MDI – setpoint transfer type 0 = value transfer by positive edge on MdiEdge 1 = constant setpoint transfer
15	- not assigned -	

Assignment control word 3 – PZD3

Table 2-5

Bit	Name	Designation
0	TrvStart	TRV – activate traversing task (by positive edge)
1	TrvBit0	TRV – block selection bit 0
2	TrvBit1	TRV – block selection bit 1
3	TrvBit2	TRV – block selection bit 2
4	TrvBit3	TRV – block selection bit 3
5	TrvBit4	TRV – block selection bit 4
6	TrvBit5	TRV – block selection bit 5
7	IntMStp	TRV/MDI – intermediate stop 0 = active travel command interrupted / axis decelerates at specified delay override 1 = no intermediate stop (axis can be moved)
8	RejTask	TRV/MDI – reject traversing task 0 = active travel command interrupted / axis decelerates at 100% delay override 1 = do not reject traversing task (axis can be moved)
9	Jog1	Jog backwards
10	Jog2	Jog forwards
11	- not assigned -	
12	JogInc	Jog mode 0 = continuous motion 1 = travel by configured path
13	SftLimAct	Activation of software limit switch
14	StpCamAct	Activation of the hardware limit switches

Bit	Name	Designation
15	- not assigned -	

Overview of status signals

Status signals are known as signals that are transferred from the drive to the controller. The assignment of the individual PZD words is as shown below.

Table 2-6

PZD	Assignment of process data
PZD1	Status word 1
PZD2	Speed actual value – 0-100% (4000HEX = 100%) in relation to maximum speed
PZD3	Status word 3
PZD4	Status word 2
PZD5	Position actual value in [unit of length LU, e.g. mm]
PZD6	
PZD7	Speed actual value [revolutions per minute]
PZD8	Current fault code

Assignment status word 1 – PZD1

Table 2-7

Bit	Name	Designation
0	RTS	Ready to start
1	RDY	Ready to operate
2	IOP	Drive in operation
3	Fault	Fault active
4	OFF2_inactiv	OFF2 not enabled
5	OFF3_inactiv	OFF3 inactive
6	Inhibit	On inhibit active
7	Alarm	Alarm active
8	Standstill	Actual speed < speed threshold value 3 (standstill detection)
9	LB_CR	Life bit control request
10	JogAct	Jog mode active
11	RefAct	Referencing mode active
12	TrvBlAct	Traversing block mode active
13	MdiPosAct	Positioning in the MDI/direct setpoint specification mode is active
14	MdiSetupAct	Setup in the MDI/direct setpoint specification mode is active
15	FlyRefAct	Flying referencing active

Assignment status word 2 – PZD4

Table 2-8

Bit	Name	Designation
0	RefDone	Refernce point set
1	CmdAct	Travel command active
2	TargPos	Target position reached
3	NoFlwErr	Following error within tolerance
4	SftSwNegAct	Software limit switch “reverse” approached
5	SftSwPosAct	Software limit switch “forward” approached
6	StpCamNegAct	Hardware limit switch “reverse” approached
7	StpCamPosAct	Hardware limit switch “forward” approached
8	AckTrvBl	For the traversing block mode or MDI/direct setpoint specification at triggered setpoint transfer (MdiTrTyp = 0) the bit is used to acknowledge the traversing block.
9	SetPStatic	Setpoint static
10	Fwd	Axis moving forward
11	Rev	Axis reversing
12	Accel	Axis accelerating
13	Decel	Axis decelerating
14	PrntMrkOut	Print mark outside outer window
15	VelctyLimit	Velocity limit active

Assignment status word 3 – PZD3

Table 2-9

Bit	Name	Designation
0	AckTrvBit0	Active traversing bit 0
1	AckTrvBit1	Active traversing bit 1
2	AckTrvBit2	Active traversing bit 2
3	AckTrvBit3	Active traversing bit 3
4	AckTrvBit4	Active traversing bit 4
5	AckTrvBit5	Active traversing bit 5
6	TrvOut1	Direct output 1 via traversing block
7	TrvOut2	Direct output 2 via traversing block
8	- not assigned -	
9	- not assigned -	
10	- not assigned -	
11	- not assigned -	
12	- not assigned -	
13	TrckMode	Tracking mode active
14	PosSmCam1	Position actual value <= Cam switch position 1
15	PosSmCam2	Position actual value <= Cam switch position 2

2.6 Functionalities used in the Basic positionier

Operating modes

For this configuration example all four modes of the EPOS were used:

- Jog mode
- Referencing
- MDI/direct setpoint specification (absolute/relative positioning)
- Traversing blocks

The priority of the interconnected operating modes during simultaneous selection is as follows:

Jog > reference point approach > MDI > traversing blocks

If another operating mode is selected during an active one, a warning message will appear.

Each mode has different functions although they are not all used here. The following pages explain which functions are relevant in the individual operating modes for this configuration example, how they are configured and what additional parameters are used.

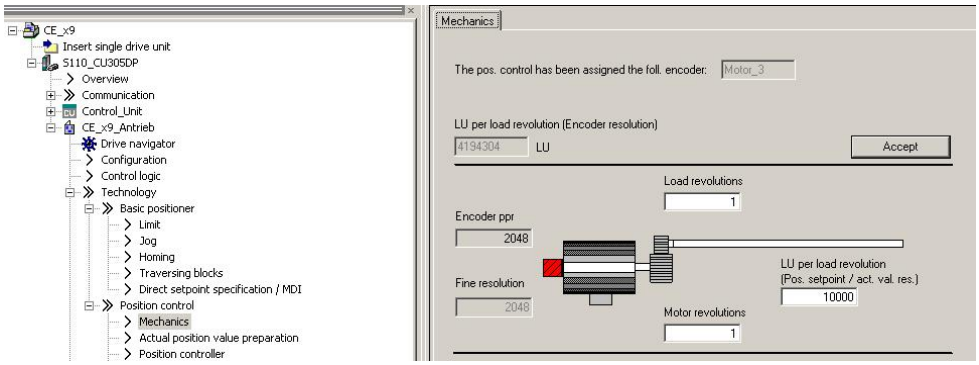
Mechanism

The mechanical axis is usually connected with the motor via a gear or a toothed belt. In order to define by how many units the axis is moved per motor revolution, the ratio of load revolutions to motor revolutions has to be specified.

For linear axis most of the time the normalization 1LU = 1µm is selected. Then, in your mind, you only need to replace the abbreviation LU by the unit µm.

In this configuration example a load revolution of 10,000 LU is assumed. Furthermore, a load revolution corresponds with a motor revolution. This means that the position of the axis is changed by 10,000 LU = 10,000 µm = 10mm per motor revolution.

Figure 2-14



Mechanical and dynamical limits

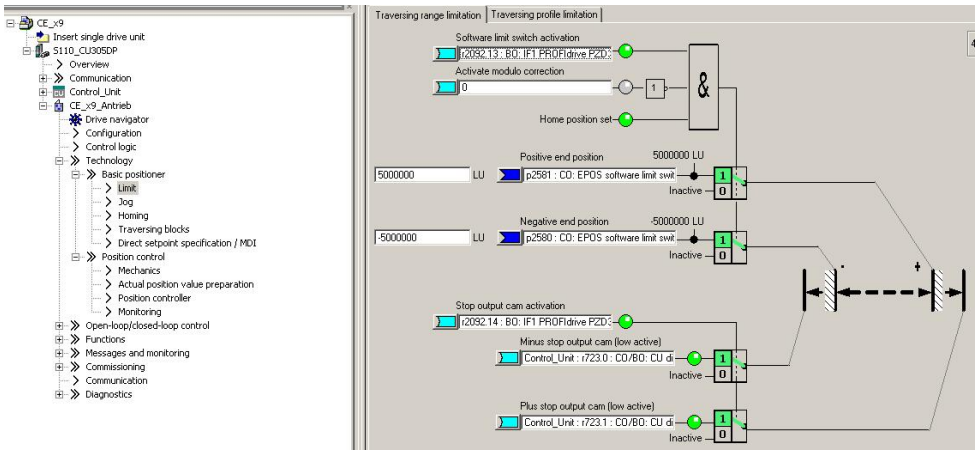
To avoid moving the axis beyond the mechanical limit, software and hardware limit switches (stop cams) can be applied.

Software limit switches are fixed position values. When the axis is referenced, these positions cannot be overrun. Before reaching this position the axis is brought to a standstill with delay.

Hardware limit switches are wired switches or sensors. When this switch is triggered by overrunning the axis, the axis will be abruptly brought to a standstill.

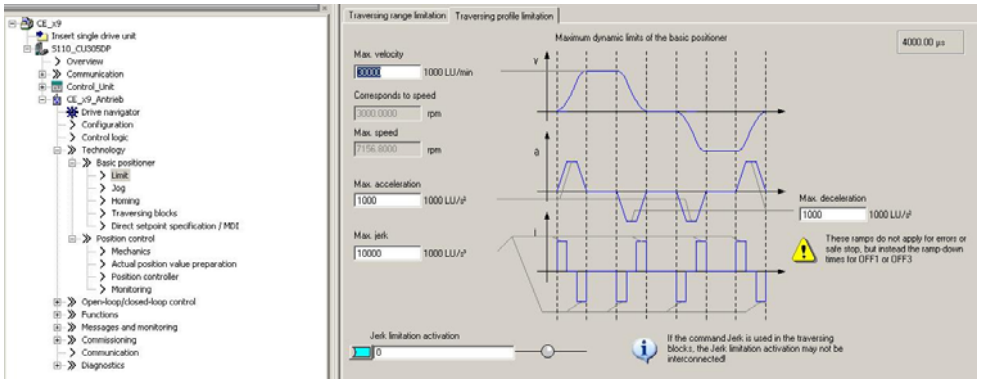
Both limit switches are activated in this configuration example. The hardware limit switches are also used as reversing cams in homing mode.

Figure 2-15



The maximum velocity, acceleration and deceleration values also have to be configured. They usually depend on the nominal speed and inertia of the motor. The maximum velocity in this configuration example corresponds with the nominal speed of the motor. Acceleration and deceleration values are adjusted in a way so that the motor quickly reaches its final speed without creating a following error at 100% acceleration and without connected axis.

Figure 2-16



Switching on of drive

The drive has to be brought to the “Ready to start” state to be able to switch it on. The following conditions are necessary:

- No fault pending
- OFF2 and OFF3 are not active (TRUE)
- RUN is not active (FALSE)

Only when these conditions apply, can the drive be operated by setting the “RUN” input on the “USS_DRV” block.

Acknowledging the error

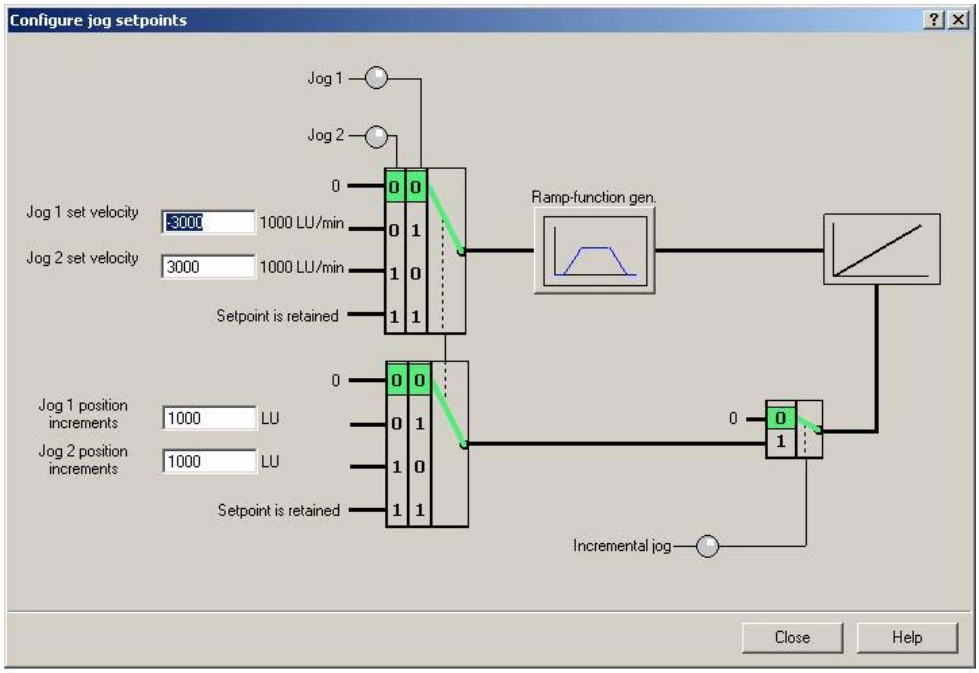
If an acknowledgeable error is pending, it has to be reset by setting the “F_ACK” input on the “USS_DRV” block. Afterwards the drive can be switched on again according to the above mentioned conditions.

Jog mode – manual moving

The jog mode is used for slowly moving the axis, for example, during commissioning. This is why the maximum velocity of the axis can be limited during jog mode. The setpoint velocity for forward and reverse may differ.

In this configuration example the axis is moved with only 10% of the nominal speed of the motor at an indicated setpoint of 100%.

Figure 2-17



Two modes are available in jog mode:

- Continuous jog, i.e. as long as the control bit “Jog” is active, the axis is moved in the selected direction
- Incremental jog, i.e. the axis is only moved at a defined increment when the “Jog” control bit is active

As long as the axis is moved the “CmdAct” bit is active.

Homing – reference point approach

The drive has to know the physical position of the axis before the axis can be moved in a defined way. This can be realized by the EPOS via different routes (see function manual).

An external switch (sensor) is used as reference point switch (zero mark) in this configuration example, which has to be connected via a digital input of the drive. The two hardware limit switches are used as reverse cams if the reference point is not found.

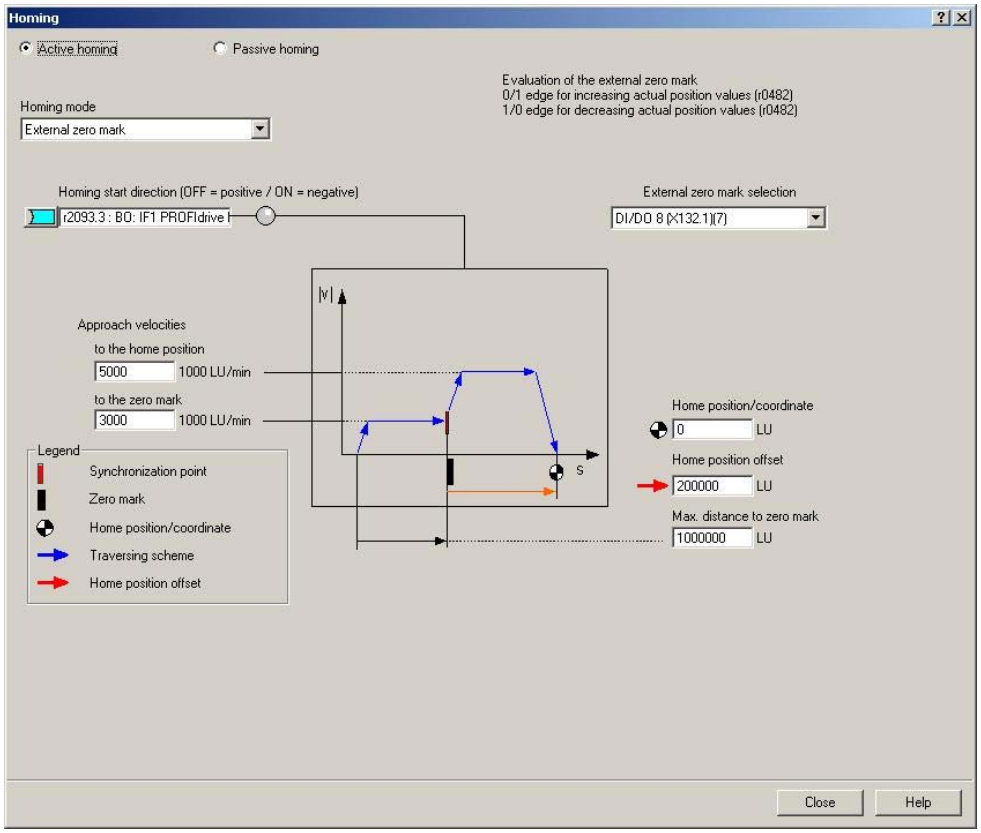
The axis is moved at a defined velocity and in a defined direction until it reaches the zero mark by setting the “RefStart” bit.

When the positive edge of the zero mark is detected, the referene point is approached at a defined velocity.

If the zero mark is not detected but the hardware limit switch is detected instead, then the direction is reversed and the search continues.

The value of the reference point shift indicates how far the reference point is away from the zero mark. Once the reference point has been reached, the configured reference point coordinate is applied as the actual position of the axis.

Figure 2-18



As long as the axis is moved, the “CmdAct” status bit is active. Once homing was successful the “RefDone” status bit is set to “TRUE”.

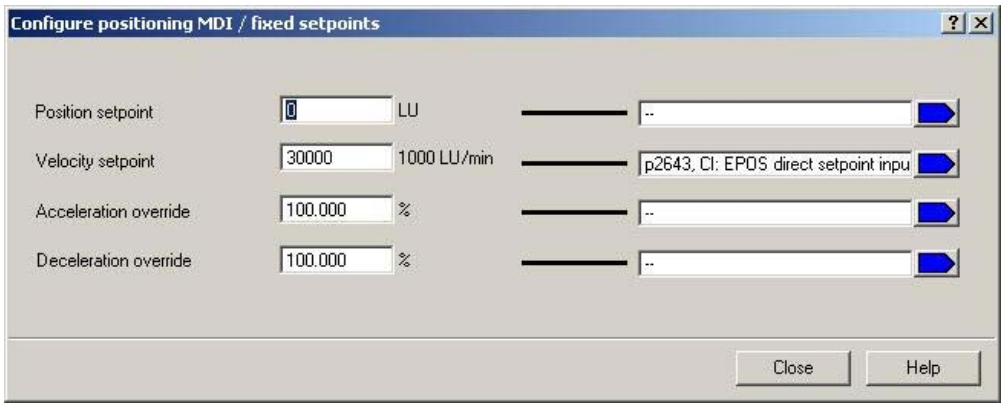
Direct setpoint specification/MDI – absolute and relative positioning

Due to the homing the axis, the current position is known. Now any position within the mechanical limits can be approached by specifying the setpoint position in [mm] and the setpoint velocity in [%].

Note Relative positioning is also possible when the axis is not referenced.

100% setpoint velocity corresponds with the nominal speed of the motor. Acceleration and deceleration values are related to the parameters of the traversing profile limit.

Figure 2-19



Before a MDI job can be started the control bits “IntMStp” and “RejTask” have to be active. It has to be selected between absolute or relative positioning.

As long as the axis is moved, the “CmdAct” status bit is active. A traversing task can be rejected be resetting the “RejTask” control bit. Or it can be interrupted by resetting the “IntMStp” control bit and continuing later. A new setpoint transfer during axis motion is also possible.

Traversing blocks

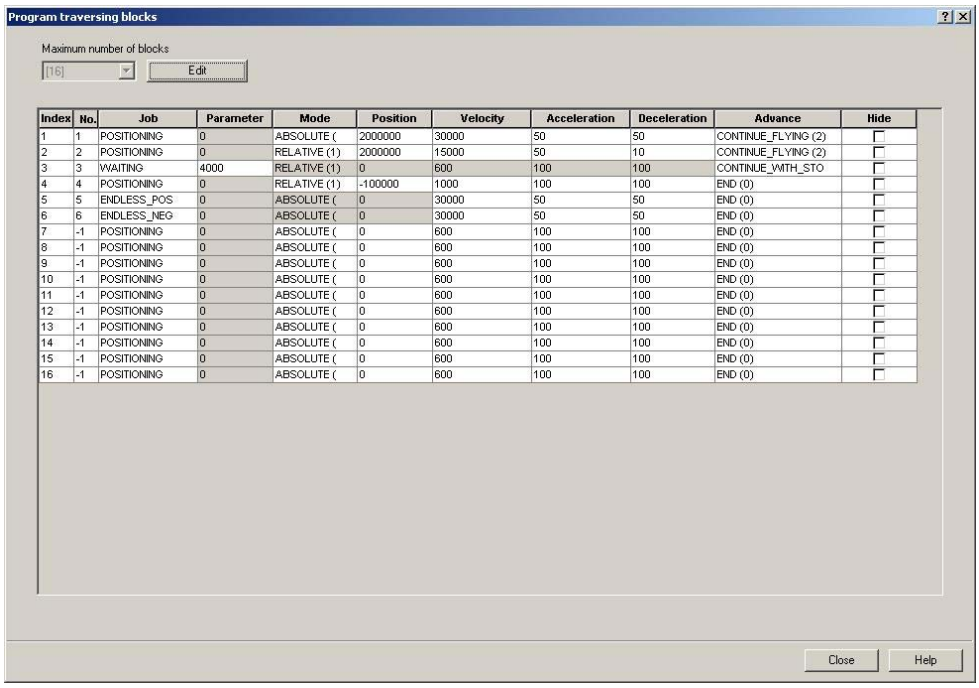
Once the axis was referenced, you can approach up to 16 stored traversing blocks in this mode. Per block position, velocity, acceleration and deceleration can be adjusted separately.

The jobs can be:

- absolute/relative positioning
- continuous positive/negative traversing
- waiting with wait time
- jumping blocks
- simultaneous setting/resetting of up two digital outputs

The use of traversing blocks is advisable, e.g. in machines where one and the same workstep has to be repeated again and again with firmly defined values.

Figure 2-20



Index	No.	Job	Parameter	Mode	Position	Velocity	Acceleration	Deceleration	Advance	Hide
1	1	POSITIONING	0	ABSOLUTE (	2000000	30000	50	50	CONTINUE_FLYING (2)	
2	2	POSITIONING	0	RELATIVE (1)	2000000	15000	50	10	CONTINUE_FLYING (2)	
3	3	WAITING	4000	RELATIVE (1)	0	600	100	100	CONTINUE_WITH_STO	
4	4	POSITIONING	0	RELATIVE (1)	-100000	1000	100	100	END (0)	
5	5	ENDLESS_POS	0	ABSOLUTE (	0	30000	50	50	END (0)	
6	6	ENDLESS_NEG	0	ABSOLUTE (	0	30000	50	50	END (0)	
7	-1	POSITIONING	0	ABSOLUTE (	0	600	100	100	END (0)	
8	-1	POSITIONING	0	ABSOLUTE (	0	600	100	100	END (0)	
9	-1	POSITIONING	0	ABSOLUTE (	0	600	100	100	END (0)	
10	-1	POSITIONING	0	ABSOLUTE (	0	600	100	100	END (0)	
11	-1	POSITIONING	0	ABSOLUTE (	0	600	100	100	END (0)	
12	-1	POSITIONING	0	ABSOLUTE (	0	600	100	100	END (0)	
13	-1	POSITIONING	0	ABSOLUTE (	0	600	100	100	END (0)	
14	-1	POSITIONING	0	ABSOLUTE (	0	600	100	100	END (0)	
15	-1	POSITIONING	0	ABSOLUTE (	0	600	100	100	END (0)	
16	-1	POSITIONING	0	ABSOLUTE (	0	600	100	100	END (0)	

Before a traversing block can be started the control bits “IntMStp” and “RejTask” have to be active. A valid traversing block number has to be selected via the six “block selection” bits.

As long as the axis is moved, the “CmdAct” status bit is active. A traversing task can be rejected be resetting the “RejTask” control bit. Or it can be interrupted by resetting the “IntMStp” control bit and continuing later. By activating a new traversing block a running traversing block can be interrupted and it can be quickly changed to the new traversing block.

DANGER

To ensure that positioning and traversing blocks are only approached within the permitted limits, the axis must have been homed beforehand.

3 Configuration

3.1 Installing and wiring hardware

Table 3-1

No	Instruction	Comment/picture
1.	Mount • the fuse • the PM 1207 power supply unit • the S7-1200 CPU 1211C • the CM1241 on a top hat rail. Mount the PM340 onto a suitable fixture	
2.	Mount the motors onto a suitable fixture.	
3.	Connect the controller to the 24 V DC supply voltage of the PM 1207	See Figure 2-1.
4.	Connect the PM340 with the motor	See Figure 2-2.
5.	Plug the CU305 DP into the PM340 and connect the digital inputs	See Figure 2-3.
6.	Connect the PROFIBUS cable with the S7-1200 CM1241 and the CU305 DP	See Figure 2-2.
7.	Connect all earth connections with earth.	

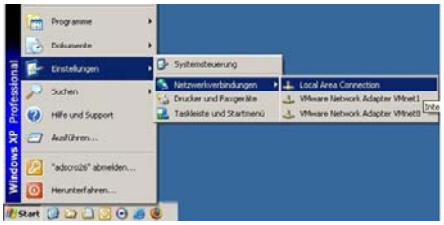
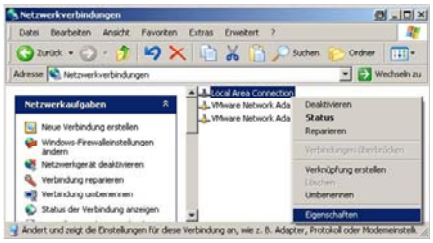
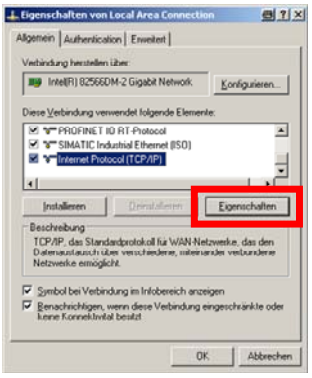
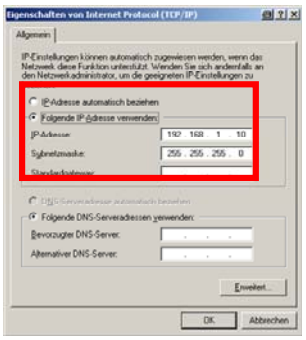
Note At this point it is assumed that the necessary software has been installed on your computer and that you are familiar with handling the software.

WARNING

Please carefully read all safety and warning notices given in the operating instructions on the frequency converter and all warning labels attached to the frequency converter before carrying out any installation and commissioning procedures. Please maintain warning labels in a legible condition and do not remove them from the device.

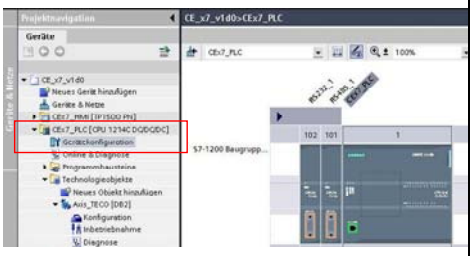
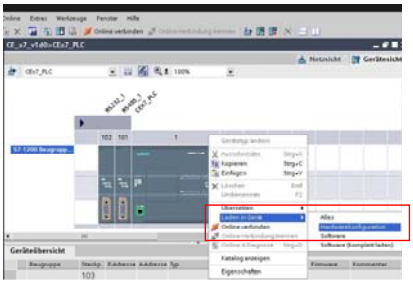
3.2 Connect S7-1200 with PG/PC

Table 3-2

No	Instruction	Comment/picture
1.	Connect your development system (PG/PC) to the S7-1200 CPU, using an Ethernet cable.	
2.	Open the network connection in the control panel of the PG/PC.	
3.	Open the properties of the network connection.	
4.	Open the Internet Protocol (TCP/IP) Properties	
5.	In the “General” tab mark “Use the following IP address” and enter the address: • IP address: 192.168.0.100 • Subnet: 255.255.255.0	
6.	Confirm with OK to close all windows	

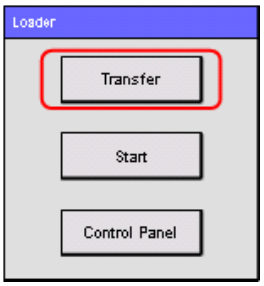
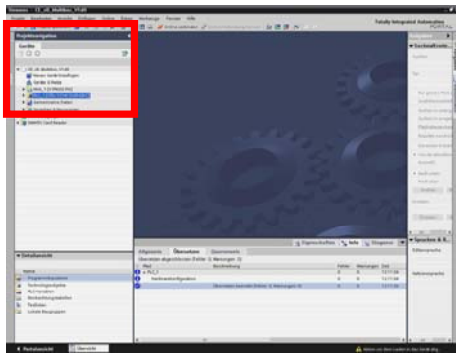
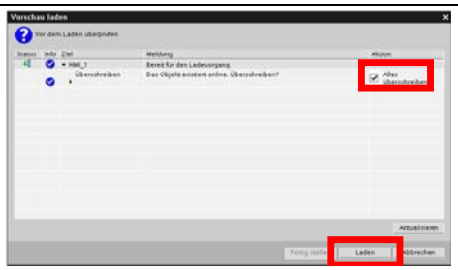
3.3 Load project in the S7-1200 CPU1214C

Table 3-3

No	Instruction	Comment/picture
1.	Unzip the file from Table 4-1 no. 1	CE_x9_S7-1200_v1d0.zip
2.	Open the unzipped project with STEP7 Basic v10.5	*.ap10
3.	Select the “PLC_1” device in project navigation and open the device configuration	
4.	Check the device configuration and if necessary adjust your hardware	• additional module • IP address
5.	Load the project into the CPU • Select CPU, press right mouse button and select “Download to device” → “All” • After loading set CPU to “RUN”	

3.4 Load project into the SIMATIC Panel KTP600

Table 3-4

No	Instruction	Comment/picture
1.	• Connect KTP600 with a DC 24V power supply • Connect your development system (PG/PC) with the KTP600 using an Ethernet cable	
2.	After the “bootloader” sequence, select the “Transfer” button. Beforehand, make sure that the IP addresse entered in the panel corresponds with the assigned IP address in the project. The panel is now ready for loading the project.	
3.	Click HMI_1 [KTP600] with the right mouse in the project navigation	
4.	Select “Download to device” → “All”	
5.	Tick the “Action” column and click “Download”	
6.	The HMI project is loaded. The panel starts automatically after completion.	
7.	Once the S7-1200 project was successfully loaded, connect the KTP600 with the S7-1200 CPU1214C, using an Ethernet cable.	

3.5 Using PC runtime instead of the panel

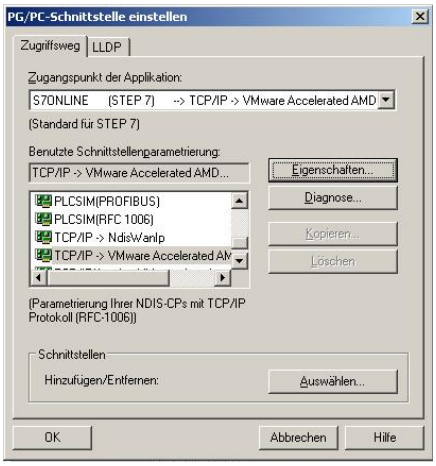
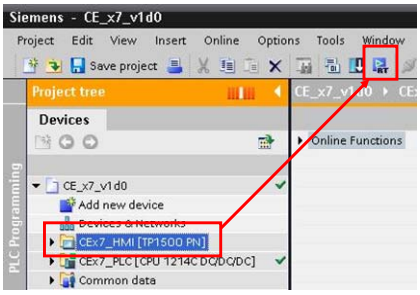
Apart from programming a controller, STEP7 Basic V10.5 also offers the visualization of the project. The software supports all currently available Basic Panels with Ethernet interface.

If no panel is available, the panel can also be simulated by the integrated PC runtime.

For convenient operation of the project, a HMI project was integrated which can also be simulated via PC runtime.

To make the simulation executable please proceed as follows:

Table 3-5

No	Instruction	Comment/picture
1.	Go to the control panel of your programming device and set the PG/PC interface as follows: • Access point: S7-Online • Interface: TCP/IP -> “Your network adapter”	
2.	Go back to the STEP7 Basic project	*.al10
3.	• Mark “HMI_1 [KTP600]” in project navigation • Subsequently click the “Start runtime “ icon	

3.6 Terminating RS 485 bus with terminating resistors

Table 3-6

No	Instruction	Comment/picture
1.	• To activate the terminating resistors put the switch in the “ON” position	

3.7 Configuring the SINAMICS S110

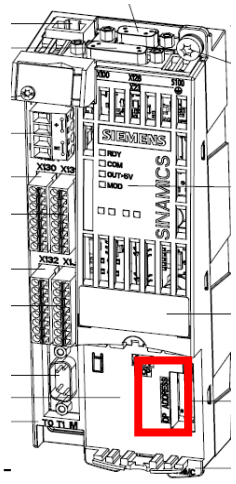
SINAMICS S110 can be easily and quickly configured with the STARTER startup tool. Basic knowledge of the software is assumed.

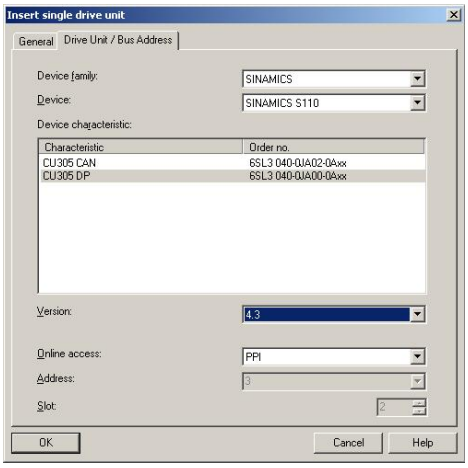
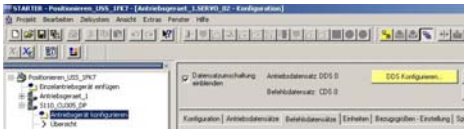
Below, it is explained how the SINAMICS S110 is to be configured for the use of the USS interface and the EPOS, so that this configuration example is executable.

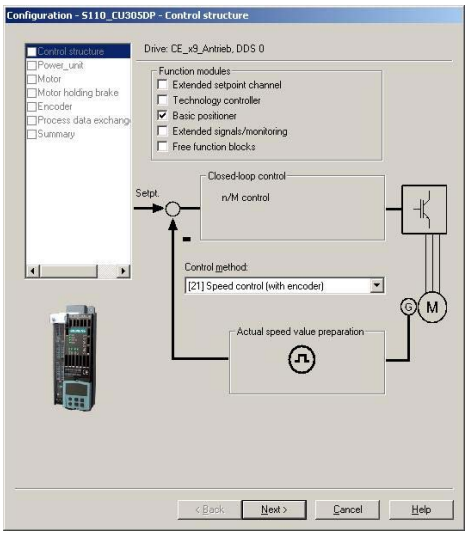
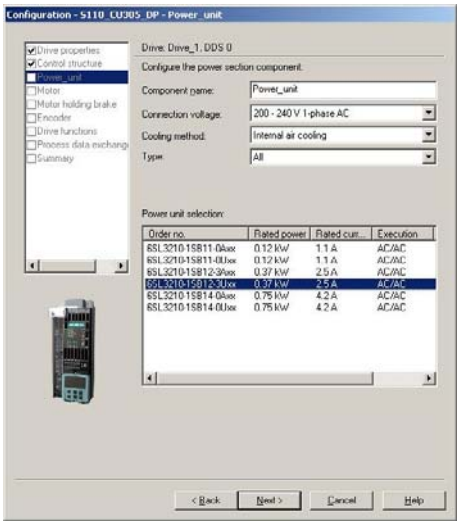
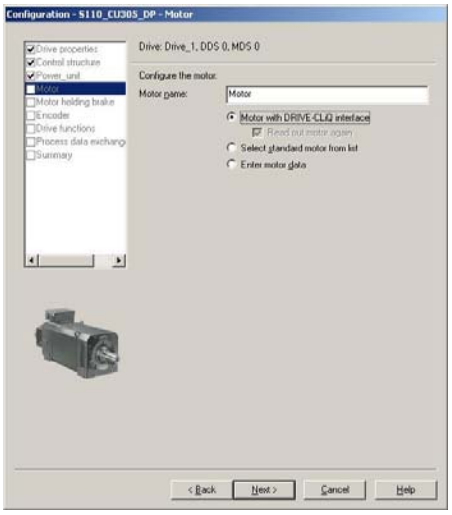
Note

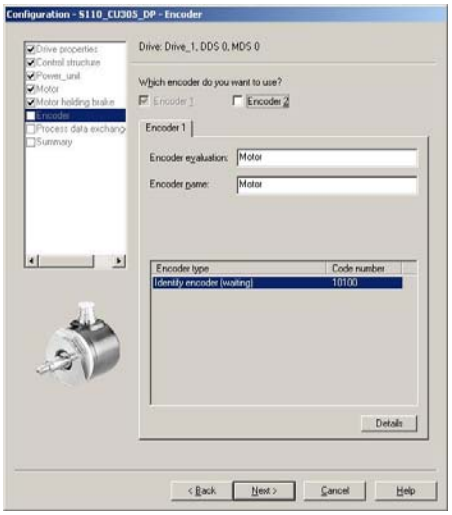
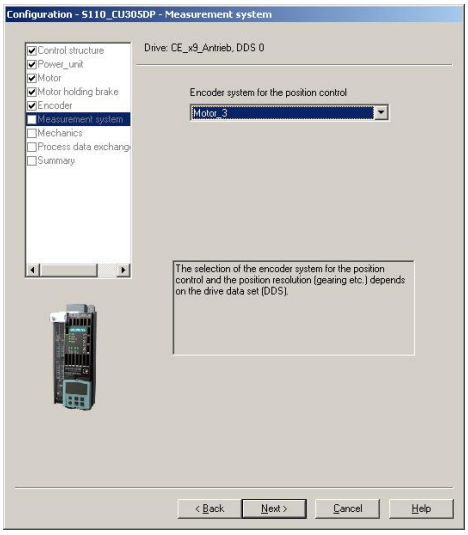
On the Internet site on which you have downloaded this documentation you will find a STARTER project in which the configuration for the SINAMCIS S110 incl. motor, listed in the component list (chapter 1.2), has already been performed. It only has to be loaded to the device. In this case, reconfiguration is not necessary.

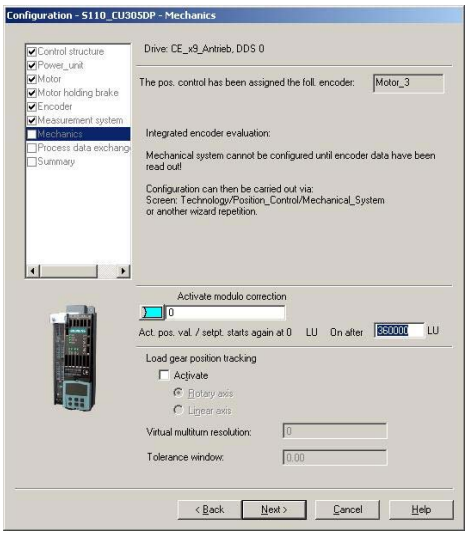
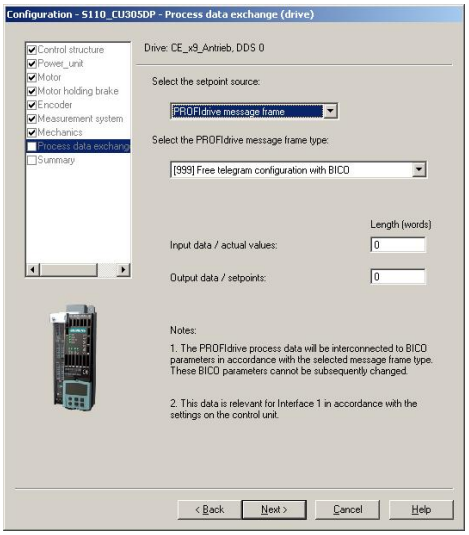
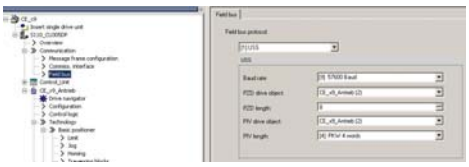
Table 3-7

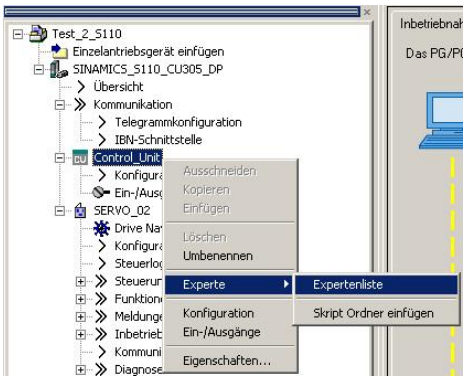
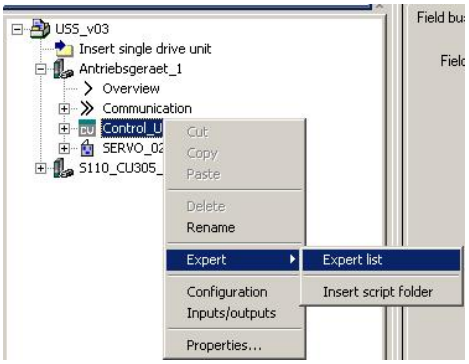
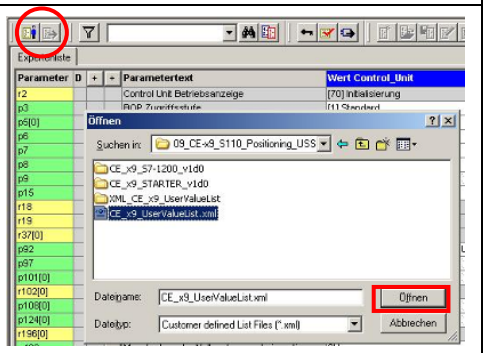
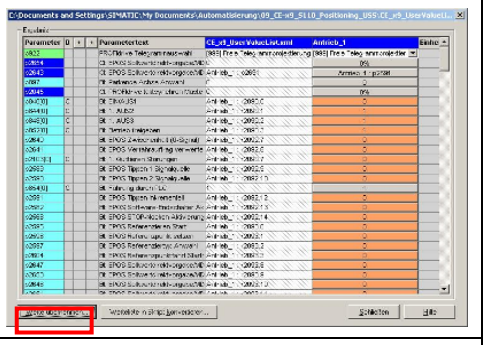
No	Instruction	Comment/picture
1.	Open the STARTER program	
2.	Connect the PC with the RS232 interface of the SINAMICS S110 using the serial null modem cable	
3.	Set address 1 of the drive via the DIP switches on the CU305 DP. Afterwards, switch the drive on	

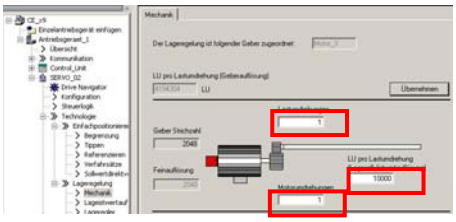
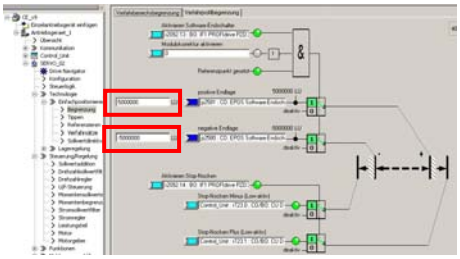
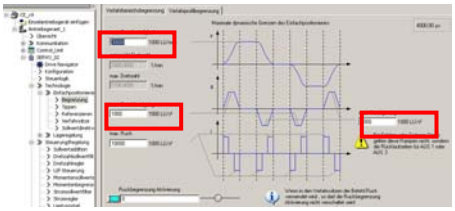
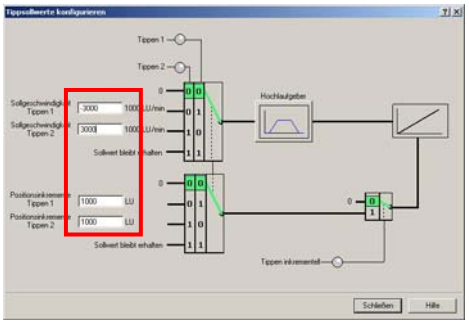
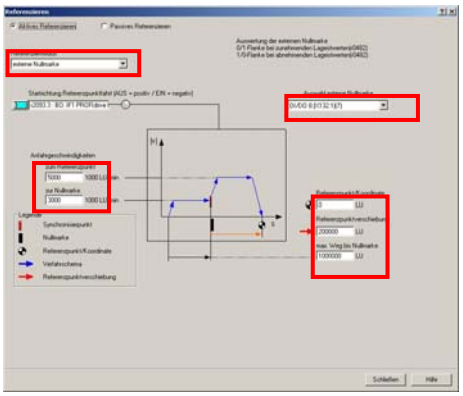
No	Instruction	Comment/picture
4.	Create a new project	
5.	Insert a new single drive unit with the following characteristics: • SINAMICS S110 • CU305 DP • Version 4.3 • Online access PPI	
6.	**Configuring drive** Doubleclick “Configure drive unit”	
7.	Select an object name Click Next.	

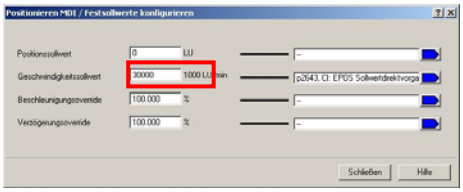
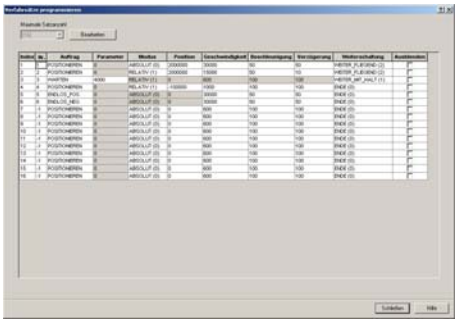
No	Instruction	Comment/picture
8.	Specify the control structure • Basic positionier • Speed control with encoder Click Next.	
9.	Select a power unit: • 6SL3210-1SB12-3Axx, 0.37kW, 2.5A, AC/AC Click Next.	
10.	Select a motor: • Motor with DRIVE-CLiQ interface Click Next.	

No	Instruction	Comment/picture
11.	Select: • No motor holding brake Click Next.	
12.	Encoder 1 is selected by default (motor encoder) Click Next.	
13.	By default encoder system for position control: Motor Click Next.	

No	Instruction	Comment/picture
14.	The mechanic can only be configured when the encoder data was read. Click Next.	
15.	As setpoint source select the PROFIDrive message frame with telegram type: • [999] free telegram configuration with BICO • Click Next, afterwards click Finish.	
16.	**Setting USS protocol** Select: “Your drive unit” → Communication → Field bus. Enter the following values: • Field bus protocol. USS • Baud rate: 57600 • PZD drive object: “Your drive” (e.g.: SERVO or Drive_1) • PZD length: 8 • PIV drive object: “Your drive” (e.g.: SERVO or Drive_1) • PIV length: 4	

No	Instruction	Comment/picture
17.	Open the expert list of the control unit by clicking the right mouse button in the project tree: "Control Unit" → Expert → Expert list	
18.	Search the parameter P2040 COMM_INT monitoring time and set the value to 230ms.	TimeOut monitoring of drive
19.	**Creating interconnections for send and receive direction** Open the expert list of the drive by clicking the right mouse button in the project tree: "Your drive" → Expert → Expert list	
20.	• In the expert list click the "Open user defined value list" tab • look for the storage location of the "CE_x9_UserValueList.xml" list, included in delivery • Click "Open"	
21.	Now the differences between the current project (right column) and the user-defined list (left column) is displayed in two columns • Click "Apply values" • When the values have been applied, close the list → The EPOS function model is now wired in a way so that it operates the process data words, as described in chapter 2.5	
22.	**Application-specific settings** Below other relevant assignments are set, such as e.g. the configuration of the acceleration, deceleration, speeds, limit switches, positions, etc.	See chapter 2.6

No	Instruction	Comment/picture
23.	• Click “Your drive” → Technology → Position control → Mechanics • Click Process. • Set the load revolution and motor revolution to “1” • Set the “LU per load revolution” to “10000”	
24.	• Click “Your drive” → Technology → Basic positionier → Limit • Select the “Traversing range limitation” tab • Set the limit of the software limit switch to 5,000,000 and -5,000,000	
25.	• Click “Your drive” → Technology → Basic positionier → Limit • Select the “Traversing profile limitation” tab • Set the max. speed to the nominal speed of the motor, e.g. “30,000” for 3,000min^{-1} • Set the max. acceleration and deceleration to “1,000”	
26.	• Click “Your drive” → Technology → Basic positionier → Jog • Click Configure jog actual values • Set the actual speed “Jog 1” to -3,000 (reverse) • Set the actual speed “Jog 1” to 3,000 (forward) • Set the position increment to “Jog 1 and 2” to 1,000	
27.	• Click “Your drive” → Technology → Basic positionier → Referencing • Click Referencing Configuration • Set “External zero mark” as referencing mode • Set the approach speed to the reference point at 5,000 and to zero mark at 3,000 • Select the digital input of the external zero mark: X132.1 • Assign a reference point coordinate, reference point shift and the max. path to the zero mark	 When using absolute value encoder it can be calibrated here. Referencing is then no longer necessary

No	Instruction	Comment/picture
28.	• Click “Your drive” → Technology → Basic positionier → direct setpoint specification/MDI • Click Configure setpoint values • Set the speed setpoint to a max of, e.g.: 30000	
29.	• Click “Your drive” → Technology → Basic positionier → Limit • Click Program traversing blocks • Program some traversing blocks with valid number	
30.	Connect with the target system	
31.	Load your project in the target system and select “Copy from RAM to ROM“	

3.8 Operation of the application via HMI

The application can be easily and intuitively operated via the HMI user interface, included in delivery. All functions are available, such as in chapter 2.5 and chapter 2.6. All relevant status messages are displayed. Four operating modes can be selected in sequence.

Structure of the HMI surface

Soft keys

- F1 soft key changes to start screen
- F6 soft key changes between German and English

Left bar

- Switching drive on/off
- Acknowledging the error
- Global status messages
- Currently pending fault code

Lower section

- Change between operating modes

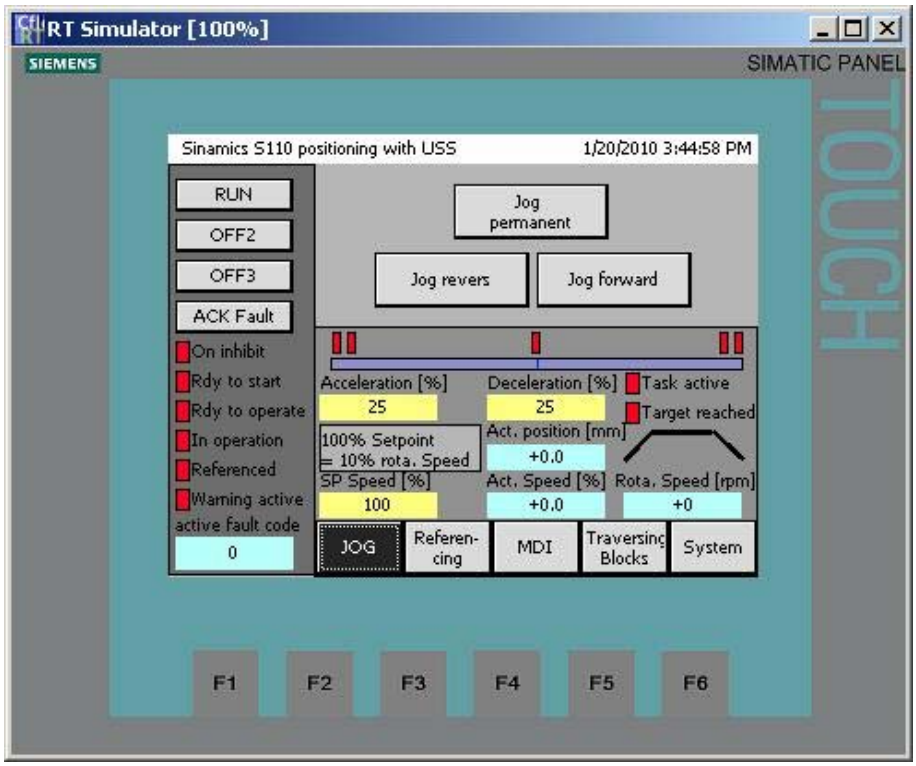
Middle section

- Display of axis in the form of a bar
- Status of the software and hardware limit switches with respect to the axis and the reference point switch
- Display of job status
- Display whether axis is currently accelerating, moving at constant speed or decelerating
- Yellow fields: Possibility to enter
  - acceleration
  - deceleration
  - setpoint speed, depending on mode
  - in the MDI mode additional entry of setpoint position
- Blue fields:
  - status display of actual position
  - actual speed
  - actual speed

Upper section

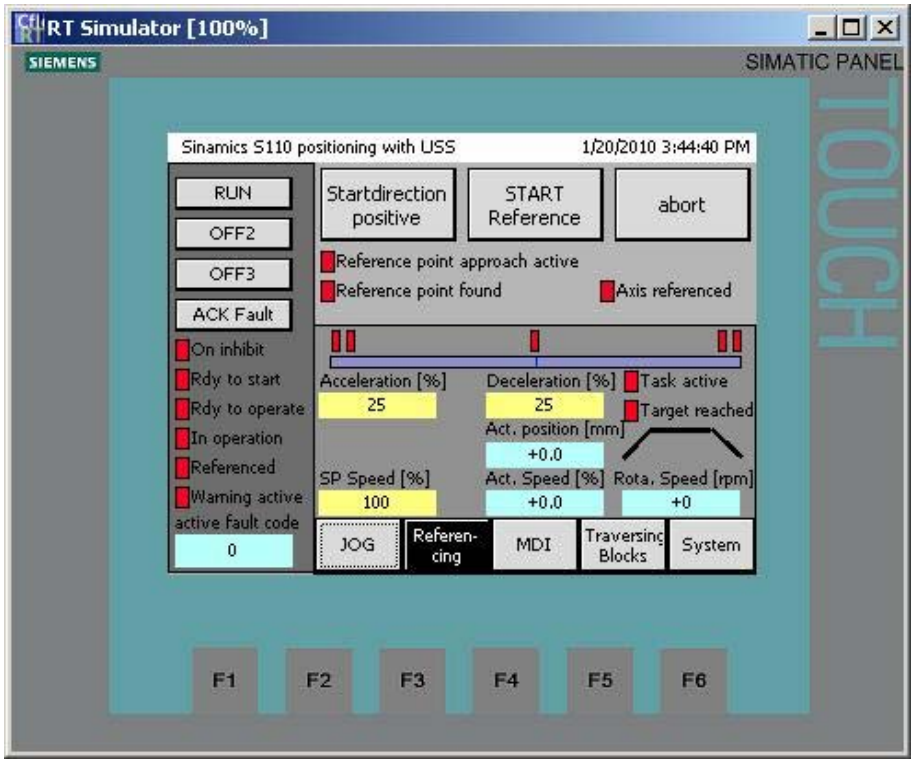
- control possibilities, depending on mode
- Jog mode:
  - Jogging continuous forward and reverse
  - Jogging incremental forward and reverse

Figure 3-1



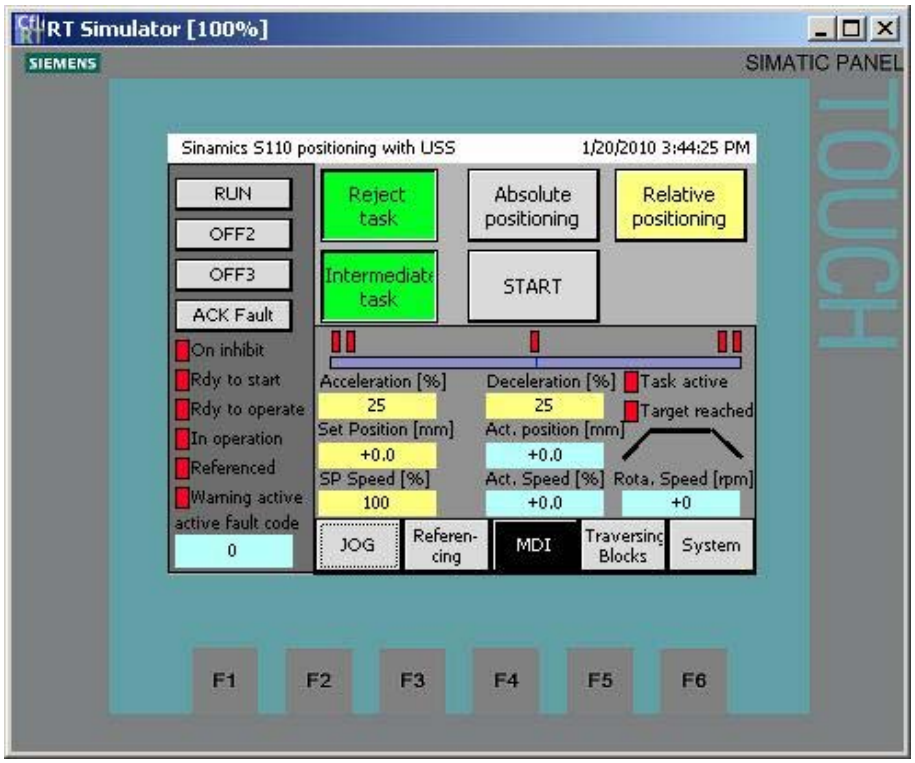
- Referencing:
  - Specification of start direction
  - Start and abort reference points search
  - Display reference status

Figure 3-2



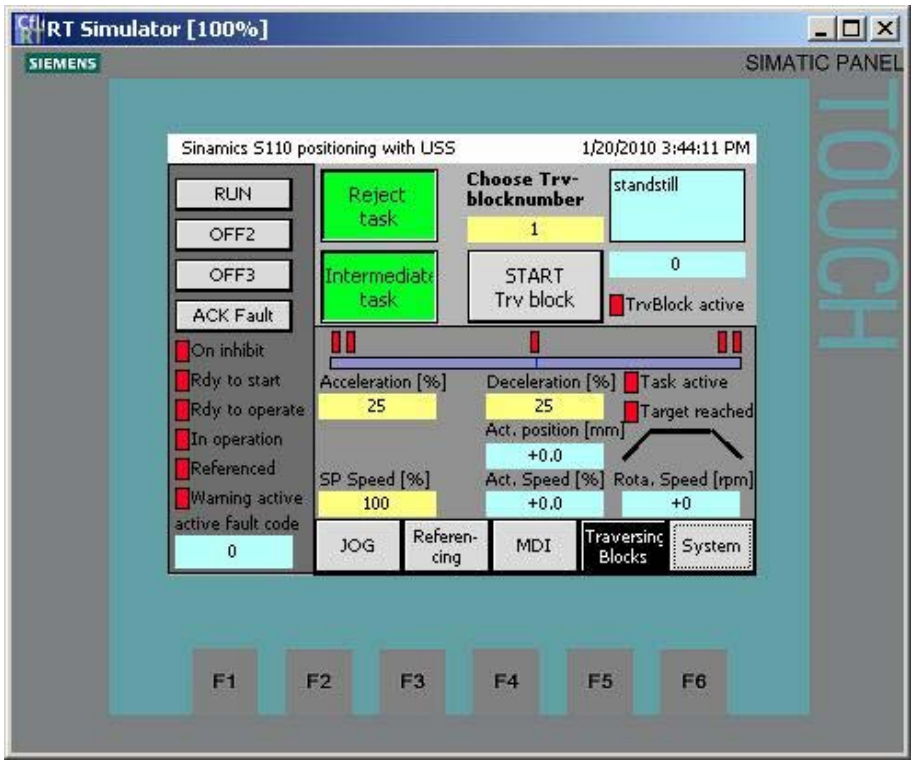
- MDI:
  - Abort job and intermediate stop
  - Select whether relative or absolute positioning
  - Start positioning

Figure 3-3



- Traversing blocks:
  - Abort job and intermediate stop
  - Select traversing block number
  - Start traversing block
  - Display which traversing block is currently active and what is set in this traversing block

Figure 3-4



4 Code Elements

The software examples are available on the HTML page from which you have downloaded this document.

Table 4-1

No	File name	Contents
1.	CE_x9_S7-1200_v1d0.zip	STEP 7 Basic V10.5 project
2.	CE_x9_STARTER_v1d0.zip	STARTER project
3.	CE_x9_UserValueList.zip	Value list for interconnection of EPOS

5 History

Version	Date	Modification
V1.0	27.01.2010	First issue