Position Control of a Drive via Pulse/Direction Interface

S7-1200, Sinamics S110 and KTP1500

Configuration Example x7 • January 2010
SIEMENS

SIMATIC
CE-X7 - Positioning a Sinamics S110 Servo Drive with S7-1200 Motion Control
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.

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

Warranty and Liability .................................................................................................................. 4

1 Automation Task .......................................................................................................................... 6
   1.1 Application environment ..................................................................................................... 6
   1.2 Component list ................................................................................................................... 7

2 Automation Solution ..................................................................................................................... 9
   2.1 Wiring diagram ................................................................................................................... 9
   2.2 Control signals between S7-1200 and servo drive .............................................................. 11
   2.3 Moving the servo motor with the aid of the pulse interface ................................................. 15
   2.4 Managing the position in the S7-1200 and the servo drive ................................................... 16
   2.5 Calculating the maximum motor frequency ......................................................................... 17
   2.6 Technology object “axis” and “motion control” function blocks ........................................... 20
   2.7 Enabling/disabling of the axis (MC_Power) ........................................................................ 21
   2.8 Acknowledgment of error (MC_Reset) ................................................................................ 21
   2.9 Manual moving – jog mode (MC_MoveJOG) ...................................................................... 22
   2.10 Manual moving – with preset velocity (MC_Velocity) ........................................................ 22
   2.11 Homing (MC_Home) ........................................................................................................ 23
   2.12 Interrupting jobs (MC_Halt) ............................................................................................. 25
   2.13 Absolute positioning (MC_MoveAbsolute) ......................................................................... 26
   2.14 Relative positioning (MC_MoveRelative) ........................................................................... 26
   2.15 Reset position (Clear Position) ........................................................................................... 27

3 Configuration ............................................................................................................................... 28
   3.1 Installing and wiring hardware ........................................................................................... 28
   3.2 Configuring the servo drive ................................................................................................. 29
   3.3 Configuring the S7-1200 CPU and downloading the hardware ........................................... 32
   3.4 Configuration of technology object “Axis” ......................................................................... 34
   3.5 Loading software ................................................................................................................ 36
   3.6 Commissioning and diagnoses of axis via the technology object ......................................... 37
   3.7 Operator control with WinCC Runtime HMI ....................................................................... 40

4 Code Elements ........................................................................................................................... 42

5 History ......................................................................................................................................... 43

6 Appendix .................................................................................................................................... 44
1 Automation Task

1.1 Application environment

A servo motor is to be moved with a servo drive by Siemens Sinamics S110 and using the pulse interface of a S7-1200 CPU1214C. Both the servo drive and the S7-1200 CPU have an individual internal pulse counter each, whose count represents the current position. Before moving to an absolute position, the counter of the S7-1200 CPU has to be synchronized with the physical position of the axis.

The task consists of absolute positioning independent of start position and velocity (Figure 1-2). The S7-1200 technology object “axis” with the respective “PLCopen - Motion Control” function block provides the necessary functions.

On the basis of the stored
- acceleration “a” and deceleration “d” [mm/s²]
- velocity in „v“ [mm/s]
- target position “•” in [mm],
the moved distance s in [mm] is calculated based on the current start position “••” and the target position “•” is approached.
## 1.2 Component list

### Products

Table 1-1

<table>
<thead>
<tr>
<th>Components</th>
<th>Qty</th>
<th>MLFB / Order number</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>PM1207 Power supply</td>
<td>1</td>
<td>6EP1332-1SH71</td>
<td></td>
</tr>
<tr>
<td>S7-1200 CPU1214C</td>
<td>1</td>
<td>6ES7214-1AE30-0XB0</td>
<td>DC</td>
</tr>
<tr>
<td>Basic panel KTP1500 (color, PN)</td>
<td>1</td>
<td>6AV6647-0AG11-3AX0</td>
<td>optional</td>
</tr>
<tr>
<td>SINAMICS Power Module PM340</td>
<td>1</td>
<td>6SL3210-1SB12-3AA0</td>
<td>230V</td>
</tr>
<tr>
<td>SINAMICS Control Unit CU305 DP</td>
<td>1</td>
<td>6SL3040-0JA00-0AA0</td>
<td>Pulse/direction variant from firmware v4.3</td>
</tr>
<tr>
<td>Synchronous servo motor 1FK7</td>
<td>1</td>
<td>1FK7023-5AF21-1UA0</td>
<td>DRIVE-CLiQ</td>
</tr>
<tr>
<td>SINAMICS S110 MMC incl. firmware v4.3 and licensing</td>
<td>1</td>
<td>6SL3054-4ED00-0AA0</td>
<td>Optional, if CU305 already existed with old firmware</td>
</tr>
</tbody>
</table>

### Accessories

Table 1-2

<table>
<thead>
<tr>
<th>Components</th>
<th>Qty</th>
<th>Order number</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power cable</td>
<td>1</td>
<td>6FX5002-5CG01-1AB0</td>
<td></td>
</tr>
<tr>
<td>Signal line DRIVE-CLiQ</td>
<td>1</td>
<td>6FX5002-2DC00-1AB0</td>
<td></td>
</tr>
<tr>
<td>Commutation inductor</td>
<td>1</td>
<td>6SE6400-3CC00-4AB3</td>
<td>Optional</td>
</tr>
<tr>
<td>230 V connection with fusing</td>
<td>1</td>
<td>6SE6400-3CC00-4AB3</td>
<td>L.N</td>
</tr>
<tr>
<td>Limit switch</td>
<td>2</td>
<td>Specialist dealer</td>
<td>Mechanically operated</td>
</tr>
<tr>
<td>Reference point switch</td>
<td>1</td>
<td>Specialist dealer</td>
<td>Inductive</td>
</tr>
<tr>
<td>Emergency stop circuit-breaker</td>
<td>1</td>
<td>Specialist dealer</td>
<td>Make contact</td>
</tr>
<tr>
<td>15 pin sub-D plug with cable</td>
<td>1m</td>
<td>Specialist dealer</td>
<td>Connection of pulse/direction signals to encoder interface of CU305 DP</td>
</tr>
<tr>
<td>330 ohm resistor 2W</td>
<td>1</td>
<td>Specialist dealer</td>
<td>Load resistor</td>
</tr>
<tr>
<td>Serial null modem cable to commission the Sinamics S110</td>
<td>1</td>
<td>Specialist dealer</td>
<td>RS232 (pin 2 and 3 rotated)</td>
</tr>
</tbody>
</table>

### Notes

- A KTP1500 is not absolutely necessary. To simulate the user interface, PC runtime from STEP7 Basic can be used.
- More information on the Sinamics S110 can be found under: [http://www.siemens.com/sinamics-s110](http://www.siemens.com/sinamics-s110)
### Programming package

**Table 1-3**

<table>
<thead>
<tr>
<th>Component</th>
<th>Qty</th>
<th>MLFB / Order number</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>18. STEP 7 Basic V10.5</td>
<td>1</td>
<td>6ES7822-0AA00-0YA0</td>
<td></td>
</tr>
<tr>
<td>19. STARTER startup tool on DVD</td>
<td>1</td>
<td>6SL3072-0AA00-0AG0</td>
<td>As of version 4.1.5 for firmware v4.3</td>
</tr>
</tbody>
</table>

**Note**

2 Automation Solution

2.1 Wiring 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.


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.
- Use shielded control lines
- 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 Control signals between S7-1200 and servo drive

Digital inputs used on the servo drive (outputs on the S7-1200)

The drive is designed to be controlled only by NPN signals. For this purpose it is essential that the X133.5 terminal is connected with ground.

The S7-1200 CPU1214C provides only PNP outputs. If the symbolically represented switch is closed by a logic “1” on the Q0.4 output of the S7-1200, the current “I” will flow. The current flow is detected by the drive as a logic “1” (Figure 2-4).

![Figure 2-4](image)

To operate the servo drive, the following input signals are used:
- Enabling/disabling of drive - Enable Servo
- Resetting of alarm – Alarm Reset
- Setting setpoint- and actual position in drive to “0” (reset position) – Clear Position

Using the encoder interface of the servo drive for pulse/direction signals

The internal X23 encoder interface of the CU305 is used to control the drive with pulse/direction signals. Only the pins 7, 13 and 15 are used according to Table 2-1.

<table>
<thead>
<tr>
<th>Pin</th>
<th>Signal name</th>
<th>Technical details</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-6</td>
<td>not relevant</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>M</td>
<td>Ground</td>
</tr>
<tr>
<td>8-12</td>
<td>not relevant</td>
<td>-</td>
</tr>
</tbody>
</table>
| 13    | BP  
Pulse/direction interface: direction | B track positive      |
| 14    | not relevant                       | -                      |
| 15    | AP_DAT  
Pulse/direction interface: pulse | A track positive    |
Control is via PNP signals, just like for digital inputs.

Figure 2-5

Furthermore, a resistor that is switched parallel to ground has to be used so that the pulses are not distorted at high frequency and that they can be clearly detected by the servo drive. Figure 2-6 shows the pulse signal without resistor. In Figure 2-7 the load resistor is present.

Figure 2-6
Incorrect wiring of the digital outputs of the S7-1200 CPU can lead to the destruction of the outputs.

**Outputs used on the servo drive (inputs on the S7-1200)**

The outputs of the servo drive can only be connected as PNP. They can be read with the S7-1200 CPU via a ground connection of the digital inputs (terminal 1M is supplied with M).

The current flows if a logic “1” is pending at the digital output of the drive (the switch symbolically represented in Figure 2-8 is closed). The current is interpreted by the S7-CPU as a logic “1”.

**Note**

All digital inputs of the S7-1200 CPU which are connected to the common potential 1M can only read PNP signals. Please observe this when wiring the hardware switches.
The following output signals are used for the servo drive feedbacks:

- Servo ready
- Servo alarm (active fault) – Alarm
- Drive stopped – standstill / In position
2.3 **Moving the servo motor with the aid of the pulse interface**

Depending on the servo drive settings, each pulse causes the servo motor to move by a defined angle.

If the drive is set, for instance, at 1000 pulses per revolution, the motor moves by 0.36° per pulse.

![Figure 2-9](image)

0.36°/per pulse – at 1000 pulses per revolution

1 revolution = 360°

The velocity of the motor is determined by the number of pulses output per second. Using the S7-1200 CPU1214C, a maximum of 100,000 pulses per second (pps) can be output.

![Figure 2-10](image)

**Correlation between velocity and distance**

The correlation between velocity and distance is explained in Figure 2-11. The moved distance in the diagram is represented by the enclosed area of both curves. The area and thus the number of output pulses is identical in both cases. Since the blue curve is moved slower than the red curve it takes more time to travel the distance.

![Figure 2-11](image)
Meaning of start/stop velocity, acceleration and deceleration

Due to the inertia of the motor it is not possible to move smoothly close to velocity “0”. To avoid a jerking of the motor, a minimum velocity is defined (start/stop velocity).

If the pulse interface is activated, the start/stop velocity is moved first. From there, the motor is accelerated to the specified velocity. Before reaching the end position, the motor is decelerated until start/stop velocity is reached. Subsequently, the pulse interface is disabled.

Figure 2-12

2.4 Managing the position in the S7-1200 and the servo drive

The pulses output by the S7-1200 are evaluated in the servo drive, independently of the S7-1200. Internally, the S7-1200 counts the number of output pulses via a high-speed counter, but receives no feedback on the actual position of the servo drive.

To be able to correctly evaluate the output pulses of the S7-1200, the maximum frequency of the S7-1200 has to be adjusted to the nominal speed of the servo motor (s. chapter 2.5).

The servo drive then controls the motion of the servo motor.

Figure 2-13 illustrates the sequence.
2.5 Calculating the maximum motor frequency

To ensure that the motor is not moved at a speed that is higher than its nominal speed, the maximum motor frequency must be determined that may finally be output by the pulse interface of the S7-1200 CPU. For this purpose the nominal speed of the motor and the number of pulses per revolution has to be known.

A special feature of the Sinamics S110 allows that the number of pulses per revolution (number of pulses per revolution) can be set variably in the drive. It can be selected between greater position accuracy and greatest possible dynamic.

This means that when the position accuracy is greater (greater number of pulses per revolution), this results in the motor being moved with a small angle per pulse. The nominal speed is limited.

If the dynamic is greater (smaller number of pulses per revolution), the nominal speed of the motor can be reached or it can even be exceeded. However, the moving of the motor is performed with a greater angle per pulse. Positioning is less accurate.

In this configuration example, it is aimed to reach the nominal speed of the motor. So therefore the number of pulses per revolution has to be calculated. This results in a maximum motor frequency (nominal speed) of 3000 revolutions per minute, corresponding to the maximum possible pulse frequency of the S7-1200 CPU of 100,000 pulses per second.

In this case, calculating the number of pulses for the drive looks like this:
Sample calculation for greatest possible dynamic

**Given variables:**
- Nominal speed of the servo motor \( (T_{Motor}) = 3000 \text{ rpm} \)
- Maximum pulse frequency of the CPU \((f_{CPU}) = 100,000 \text{ pps}\)

**Calculation:**

\[
P_{Motor} = \frac{f_{CPU} \times 60s}{T_{Motor}}
\]

\[
P_{Motor} = \frac{100000 \text{ pps} \times 60s}{3000 \text{ rpm}}
\]

\[
P_{Motor} = 2000 \text{ ppr}
\]

**Result:**
To reach the nominal speed of the motor, whilst taking into account that the maximum frequency is 100,000 pulses per second, a number of 2000 pulses per revolution has to be set. This results in a position accuracy of 0.18° per pulse.

A smaller number of pulses per revolution would mean that the nominal speed is exceeded. Limiting the maximum frequency of the S7-1200 prevents the exceeding of the nominal speed at a lower number of pulses per revolution.

**Note**
Using the additional SB 1222 DC signal board enables the PLC to increase the maximum control frequency to 200,000 pulses per second

Sample calculation for greater position accuracy

If the position accuracy is to be increased, then the number of pulses per revolution has to be increased. At a consistent maximum frequency of 100,000 pulses per second this means the following for the nominal speed of the motor:

\[
T_{Motor} = \frac{f_{CPU} \times 60s}{P_{Motor}}
\]

**Calculation:**
Increase of the number of pulses per revolution to 4000 ppr.

\[
T_{Motor} = \frac{100000 \text{ pps} \times 60s}{4000 \text{ ppr}}
\]

\[
T_{Motor} = 1500 \text{ rpm}
\]
Result:
At double position accuracy of 0.09° per pulse the nominal speed of the motor increases to 1500 revolutions per minute at a constant maximum frequency of 100,000 pulses per second.
2.6 Technology object “axis” and “motion control” function blocks

The technological object “axis” represents an axis in the control and facilitates the control of the servo drive via the pulse interface of the S7-1200 CPU1214C. The technology object “axis” is controlled via the “motion control” instruction.

The configuration of the technology object “axis” is described in more detail in chapter 3.4.

Figure 2-14

To meet all functions of this configuration example, the following program blocks are required which must be called cyclically in the user program.

Table 2-2

<table>
<thead>
<tr>
<th>No.</th>
<th>Program block</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>MC_Power</td>
<td>Enabling/disabling of the axis</td>
</tr>
<tr>
<td>2.</td>
<td>MC_Reset</td>
<td>Acknowledgement of all pending errors</td>
</tr>
<tr>
<td>3.</td>
<td>MC_MoveJog</td>
<td>Jog mode</td>
</tr>
<tr>
<td>4.</td>
<td>MC_MoveVelocity</td>
<td>Moving of axis at specified velocity and direction</td>
</tr>
<tr>
<td>5.</td>
<td>MC_Home</td>
<td>Homing the axis</td>
</tr>
<tr>
<td>6.</td>
<td>MC_Halt</td>
<td>Cancelling all movements, stopping of axis</td>
</tr>
<tr>
<td>7.</td>
<td>MC_MoveAbsolute</td>
<td>Absolute positioning of axis</td>
</tr>
<tr>
<td>8.</td>
<td>MC_MoveRelative</td>
<td>Relative positioning of axis</td>
</tr>
</tbody>
</table>
2.7 Enabling/disabling of the axis (MC_Power)

Before the axis can be moved it has to be enabled. When the “TRUE” signal is applied on the “enable” input of the “MC_Power” block, the output of the technology object “axis” of S7-1200 CPU is set in the configuration and the servo drive is switched on.

The “StopMode” input indicates whether the axis is to be decelerated at the configured “emergency stop” deceleration when it is disabled and turned off afterwards (“0”) or whether the axis is to be stopped instantly (“1”).

The servo drive will receive the feedback whether it is ready, on the “Status” output of the block. Errors during the operation are displayed on the “Error” output and the respective error identification on the “ErrorID” output. A list of the ErrorIDs can be found in the online help of STEP7 Basic.

Figure 2-15

2.8 Acknowledgment of error (MC_Reset)

If an acknowledgeable error is pending, it has to be reset by a positive edge on the “Execute” input on the “MC_Reset” block.

Figure 2-16
2.9 Manual moving – jog mode (MC_MoveJOG)

To move in "jog mode" the "MC_MoveJog" block is available. Once a speed was indicated at the "Velocity" input and the "JogForward" or "JogBackward" input was set, a pulse sequence will be output on the pulse output of the control until "JogForward" or "JogBackward" is reset.

The "Busy" output is active as long as the axis is moved via this block.

Figure 2-17

2.10 Manual moving – with preset velocity (MC_Velocity)

To move with preset velocity the "MC_MoveVelocity" block is available. Once a speed was indicated at the "Velocity" input and by a positive edge on the "Execute" input, a pulse sequence is output at the pulse output of the control until the "MC_Halt" block is executed.

The "Direction" input is used to specify the rotation direction and can contain the following three values:

- 0: the rotation direction is controlled via the sign (+/-) of the speed indicated
- 1: positive rotation direction (unsigned velocity value)
- 2: negative rotation direction (unsigned velocity value)

The "Busy" output is active as long as the axis is moved via this block.
2.11 Homing (MC_Home)

The controller has to know the physical position of the axis before the servo motor may be moved defined via a pulse sequence.

Learning the physical position (homing) shall be explained using a linear axis. This axis consists, for example, of a spindle that is connected to the servo motor. One revolution of the motor is to correspond to 2000 pulses and one unit length [LU] of the spindle.

It is assumed that the axis depicted green in the picture is by default located left of the reference point switch on position “0”.

The axis is moved by a positive edge on the “Execute” input of the “MC_Home” block at a defined speed and in a defined direction.

In the configuration of the technology object you define in which direction and with what speed the axis is to be moved. (Chapter 3.4, Configuration). The axis is only moved in accordance with this configuration when the value “3” is pending at the “Mode” input of the “MC_Home” block.
Three different cases can occur which have an influence on homing the axis.

**Case 1: starting position left of reference point; deceleration to slow speed is complete before reaching the negative edge**

At a positive edge of the reference point switch, the motor is decelerated to a slower velocity. The axis is now moved to the falling edge of the reference point switch and is then stopped. The position counter is set to the absolute value pending on the “Position” input.

**Figure 2-21**

**Case 2: starting position left of reference point; deceleration to slow speed is not complete before reaching the negative edge**

In case deceleration to slower velocity is not achieved before reaching the negative edge of the reference point switch, the axis is stopped. Subsequently, the axis is moved backwards at slow velocity until the positive edge of the reference point switch. The axis is stopped again and then moved forward at slow velocity, up to the negative edge.

**Figure 2-22**
**Case 3: starting position to the right of the reference point**

If the axis is behind or on the reference point switch, the axis is not detected by the reference point switch but by the forward limit switch. Axis movement will be stopped. Once it has come to standstill, it is moved backward at a defined speed until it reaches the reference point. Afterwards normal homing starts again.

Figure 2-23

The “Busy” output is active as long as the axis is moved via this block. Once the block was successfully run through, the “HomingDone” status bit is set to “TRUE” in the data block of the “axis” technology object.

### 2.12 Interrupting jobs (MC_Halt)

Each active job, i.e. each active movement of the axis can be stopped by the "MC_Halt" block. The axis is brought to a standstill with delay by a positive edge on the "Execute" input. The position, where the axis stops is not defined.

Figure 2-24

Additionally, every active job can be interrupted by triggering a new job. It is always only the job triggered last that is active.

Example: the axis is currently moved at preset velocity. If the jog mode is now activated, the job with preset velocity is deleted and jog mode is active.
2.13 Absolute positioning (MC_MoveAbsolute)

Due to the homing, the current position of the axis is known. With the aid of the "MC_MoveAbsolute" block, any position within the mechanical limits can be approached in [mm] by specifying the real position. In addition, the traversing velocity has to be specified.

Figure 2-25

If the block is started by a positive edge at the "EXECUTE" input, the number of pulses required for reaching the target position is calculated on the basis of the current position and the target position. The motor is then, if possible, accelerated up to the indicated velocity and is then stopped with delay at the target position.

2.14 Relative positioning (MC_MoveRelative)

Apart from the absolute positioning there is also the option of relative moving at any distance, direction and velocity, using the "MC_MoveRelative" block.
When the block is started by a positive edge at the “EXECUTE” input, the axis is moved by the set distance at the selected velocity. The direction results from the sign (+/-) of the distance.

**DANGER**

To ensure that the positioning only occurs within the permitted boundaries, the axis must have been homed beforehand.

### 2.15 Reset position (Clear Position)

Resetting the position in the drive is used for round axis to avoid a moving above the maximum possible count and the thus connected interference. If the “CLR” output is set, the counter for the setpoint- and actual position in the drive is set to “0”. This is necessary to eliminate a possible offset.

Is there a following error in drive because of a big difference between setpoint position and actual position, the only possibility to reset this fault is “reset position”. During switching on the drive, the digital output “CLR” is set automatically for one second, because depending on the type of encoder there can be a difference between setpoint- and actual position after switching on.
## 3 Configuration

### 3.1 Installing and wiring hardware

**Table 3-1**

<table>
<thead>
<tr>
<th>No</th>
<th>Instruction</th>
<th>Note/picture</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Mount the fuse, PM1207 and the S7-1200 CPU1214C onto a top hat rail.</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>Connect the PM1207 to the 230 V AC supply voltage. Connect the controller to the 24 V DC supply voltage of the PM1207.</td>
<td>See chapter &quot;Wiring diagram&quot;</td>
</tr>
<tr>
<td>3.</td>
<td>Mount the PM340 to the commutation inductor and install both, paying attention to the installation instructions.</td>
<td>See Sinamics S110 manual.</td>
</tr>
<tr>
<td>4.</td>
<td>Connect the inductor with the 230V AC supply voltage with the PM340.</td>
<td>See chapter &quot;Wiring diagram&quot;</td>
</tr>
<tr>
<td>5.</td>
<td>Connect the PM340 with the motor, using the power cable.</td>
<td>See chapter &quot;Wiring diagram&quot;</td>
</tr>
<tr>
<td>6.</td>
<td>Insert the CU305 to the PM340.</td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>Connect the digital inputs/outputs of the CU305 with the S7-1200.</td>
<td>See chapter &quot;Wiring diagram&quot;</td>
</tr>
<tr>
<td>8.</td>
<td>Connect the DRIVE-CLiQ interface of the CU305 and the encoder of the motor.</td>
<td>See chapter &quot;Wiring diagram&quot;</td>
</tr>
<tr>
<td>9.</td>
<td>Prepare the signal cable for the pulse/direction interface.</td>
<td>See Table 2-1.</td>
</tr>
<tr>
<td>10.</td>
<td>Connect the signal cable with the encoder interface of the CU305.</td>
<td>See chapter &quot;Wiring diagram&quot;</td>
</tr>
<tr>
<td>11.</td>
<td>Connect all earth connections with earth.</td>
<td></td>
</tr>
</tbody>
</table>
3.2 Configuring the servo drive

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

Below, the servo drive is configured in a way so that it can be moved via the external pulse/direction signals, allowing for the greatest possible dynamic.

**Note**

On the Internet site on which you have downloaded this documentation you will find a STARTER project in which the configuration for the SINAMICS S110 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>
<thead>
<tr>
<th>No</th>
<th>Instruction</th>
<th>Comment/picture</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Open the STARTER program</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>Connect the PC with the RS232 interface of the Sinamics S110 using the serial null modem cable</td>
<td>A connection with the PC is also possible via the profibus interface. A respective Profibus adapter for the PC is necessary.</td>
</tr>
<tr>
<td>3.</td>
<td>Create a new project</td>
<td></td>
</tr>
</tbody>
</table>
| 4. | Insert a new single drive unit with the following characteristics:  
  - SINAMICS S110  
  - CU305 DP  
  - Version 4.3  
  - Online access PPI (or Profibus) | ![Insert single drive unit](image) |
| 5. | Double click “Configure drive unit” | ![Configure drive unit](image) |
6. Select an object name
   Click Next.

7. Specify the control structure
   - Speed control with encoder
   
   Click Next.

8. Select a power unit:
   - 6SL3210-1SB12-3Axx, 0.37kW, 2.5A, AC/AC
   
   Click Next.
<table>
<thead>
<tr>
<th>No</th>
<th>Instruction</th>
<th>Comment/picture</th>
</tr>
</thead>
</table>
| 9. | Select a motor:  
• Motor with DRIVE-CLiQ interface | ![Configuration Example](image)  
Click Next. |
| 10. | Select:  
• No motor holding brake | ![Configuration Example](image)  
Click Next. |
| 11. | Encoder 1 is selected by default (motor encoder) | ![Configuration Example](image)  
Click Next. |
3 Configuration

12. Select the pulse/direction interface as setpoint source. Select the control type:
   - Position control

Configure the pulse/direction interface:
   - Encoder channel: 2
   - Encoder evaluation: CU305 DP
   - Pulses per revolution: 2000
   - Signal shape: Pulse/direction, positive logic

Click Next, afterwards click Finish.

13. Connect with the target system

14. Load your project in the target system and select "Copy from RAM to ROM"

### 3.3 Configuring the S7-1200 CPU and downloading the hardware

<table>
<thead>
<tr>
<th>No</th>
<th>Instruction</th>
<th>Comment/picture</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Extract the file from Table 4-1 no. 1 *.zip</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Open the extracted project with STEP7 Basic v10.5</td>
<td>*.ap10</td>
</tr>
<tr>
<td>3</td>
<td>Select the device &quot;CEx7_PLC&quot; in project navigation and open the device configuration</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Check the device configuration and if necessary adjust it to your hardware</td>
<td>additional module, IP address</td>
</tr>
<tr>
<td>No</td>
<td>Instruction</td>
<td>Note/picture</td>
</tr>
<tr>
<td>----</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>5.</td>
<td>Check whether the “PTO1” pulse generator is enabled.</td>
<td><img src="https://example.com/image1.png" alt="Image" /></td>
</tr>
<tr>
<td></td>
<td>- For this purpose click CPU (1) and then properties (2).</td>
<td><img src="https://example.com/image2.png" alt="Image" /></td>
</tr>
<tr>
<td></td>
<td>- Then select “Pulse generator” (PTO/PWM) (3).</td>
<td><img src="https://example.com/image3.png" alt="Image" /></td>
</tr>
<tr>
<td></td>
<td>- Check the settings for PTO1/PWM1 (4-5)</td>
<td><img src="https://example.com/image4.png" alt="Image" /></td>
</tr>
<tr>
<td>6.</td>
<td>Check whether the clock memory byte 2 is active</td>
<td><img src="https://example.com/image5.png" alt="Image" /></td>
</tr>
<tr>
<td></td>
<td>- For this purpose click CPU (1) and then properties (2).</td>
<td><img src="https://example.com/image6.png" alt="Image" /></td>
</tr>
<tr>
<td></td>
<td>- Then select „System and clock memory“ (3).</td>
<td><img src="https://example.com/image7.png" alt="Image" /></td>
</tr>
<tr>
<td></td>
<td>- Enable the clock memory byte and set as location MB2 (4)</td>
<td><img src="https://example.com/image8.png" alt="Image" /></td>
</tr>
<tr>
<td>7.</td>
<td>Load the hardware into the CPU</td>
<td><img src="https://example.com/image9.png" alt="Image" /></td>
</tr>
<tr>
<td></td>
<td>- Select CPU and click the “Download to device” icon or</td>
<td><img src="https://example.com/image10.png" alt="Image" /></td>
</tr>
<tr>
<td></td>
<td>- Right mouse on CPU and select “Download to device” → “Hardware configuration”</td>
<td><img src="https://example.com/image11.png" alt="Image" /></td>
</tr>
<tr>
<td></td>
<td>- After loading set CPU to “RUN”</td>
<td><img src="https://example.com/image12.png" alt="Image" /></td>
</tr>
</tbody>
</table>
3.4 Configuration of technology object “Axis“

The technology object “Axis” is already fully configured in this project. For better understanding the configuration of the object is described in more detail in the table below.

Note

The parameters “Pulses per motor revolution”, “maximum velocity” and “Start/stop velocity” may have to be individually adjusted, depending on servo drive or motor used (see chapter 2.5)

Depending on the real axis used, the mechanical limits of the axis also have to be adjusted.

Table 3-4

<table>
<thead>
<tr>
<th>No</th>
<th>Instruction</th>
<th>Note/picture</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Select the &quot;Axis_Servo&quot; technological object in the project navigation and double click “Configuration”</td>
<td></td>
</tr>
</tbody>
</table>
| 2. | Click “Basic parameters” → “General”  
- Defining the name of the axis: **Axis_Servo**  
- Select pulse interface according to device configuration: **Servo**  
- Select length unit: **mm** | |
| 3. | “Extended parameters” → Drive signals  
Used to enable/block the servo drive and is managed by “MC_Power”  
- Selecting the enable output according to wiring diagram: **Q0.4 Servo_ON**  
- Selecting of ready input according to wiring diagram: **I1.0 Servo_Ready**  
If servo drive does not provide “Ready” signal, the value **TRUE** is to be entered here | |
<table>
<thead>
<tr>
<th>No</th>
<th>Instruction</th>
<th>Note/picture</th>
</tr>
</thead>
</table>
| 4. | “Extended parameters” → Mechanics Specifying the limits of the motor and converting pulses into a length unit  
  - Pulses per motor revolution: **2000**  
  - Path per motor revolution: This is where you enter the distance which, e.g. a slide covers on a spindle per motor revolution (e.g. 10 mm)  
  - Inverting direction: Exchanges “forward” with “reverse” | ![Mechanics Diagram](image1.png) |
| 5. | “Extended parameters” → Position monitoring Defining hardware and software limit switches, their position and switching behavior  
  - Enable both, hardware and software limit switches.  
  - Define the hardware limit switches according to the wiring diagram and specify whether they are designed as make or break contacts. (e.g.: I0.2 and I0.1, Upper level → break contact)  
  - Define the position of the software limit switches according to the mechanical limits of your axis (e.g. -5000 mm to 5000 mm) | ![Position Monitoring Diagram](image2.png) |
| If the axis has been homed it will be moved within the limits of the software limit switches. When reaching the software limit switches the axis is decelerated until standstill.  
If the axis is not homed, the hardware limit switches will bring the axis to a standstill with emergency stop deceleration when it moves past the limits. |
| 6. | “Extended parameters” → Dynamic general Setting the velocity limits, acceleration, deceleration (see chapter 2.3)  
  - Enter the maximum velocity in pulses/seconds: **100,000 pps**  
  - Enter a permissible start/stop velocity (pulses/second): **1000 pps**  
  - Enter the acceleration and deceleration in mm/s², alternatively you can also enter the startup and ramp-down time in seconds: Examples: 2 s → 247.5 mm/s² Automatic conversion to mm/s². | ![Dynamic General Diagram](image3.png) |
| 7. | “Extended parameters” → Dynamic emergency stop  
  - Enter an emergency stop deceleration or ramp-down time to stop the axis when going past the hardware limit switches or to stop the axis when disabling through "MC_Power" (e.g.: 0.01 s → 49500 mm/s²) | ![Dynamic Emergency Stop Diagram](image4.png) |
8. "Extended parameters" → Homing
   - Define the reference point switch according to wiring diagram (I0.0 RPS)
   - Permit the change of direction at the hardware limit switch (see chapter 2.11 – case 3)
   - Determine the approach direction: positive
   - Specify the right side as detection point of the reference point switch
   - Define the approach speed (fast velocity for reference point switch search): 200 mm/s
   - Define the entry speed (slow velocity for the falling edge of the reference point): 10 mm/s
   - Define the reference point shift: 0 mm
   - The reference point coordinate (position to be assumed when homing successful) is configured on the "MC_Home" block

3.5 Loading software

1. Load the fully parameterized project into the controller.
   - Select program blocks and click the "Download to device" icon or
   - Right mouse on CEx7_PLC and select "Download to device" → "Software"
   - After loading set CPU to "RUN"
### 3.6 Commissioning and diagnoses of axis via the technology object

This chapter describes how you can test and diagnose the operating capability of the servo drive via the online function of the technological object "axis".

<table>
<thead>
<tr>
<th>No</th>
<th>Instruction</th>
<th>Note/picture</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Doubleclick “Technological object” → “Axis_Servo” → “Commissioning”</td>
<td><img src="image1.png" alt="Image" /></td>
</tr>
</tbody>
</table>
| 2. | - Click “Manual” → the CPU will automatically be online  
   - Then click “Enable” to activate the servo drive | ![Image](image2.png) |
| 3. | You are now in Jog mode  
   - Specify a velocity as well as acceleration/deceleration  
   - Click “Jog backwards” or “Jog forward” | The axis accelerates at the acceleration indicated and will move at the specified velocity as long as the button remains pressed. Afterwards the axis is brought to a standstill at the specified deceleration. |
| 4. | Go to “Homing” mode  
   - Specify a home position as well as acceleration/deceleration  
   - Start homing | The axis will move as long in the defined direction until the reference point switch or the hardware limit switch is detected. When the negative edge of the reference point switch is detected, the axis is stopped and the specified position of the reference point will be taken on in the current position. |
| 5. | Go to “Positioning” mode  
   - Specify a velocity as well as acceleration/deceleration  
   - Move the axis “Relative” by specifying a “Path” (+/-)  
   - Move the axis “Absolute” by specifying a “Target” (+/-)  
   - Please note: the axis can only be moved absolute when it has been homed. | The axis moves from the current position to the specified path  
The axis moves to the position specified. |
### Diagnostics

<table>
<thead>
<tr>
<th>No</th>
<th>Instruction</th>
<th>Note/picture</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Doubleclick “Technological objects” → “Axis_Servo” → “Diagnostics”.</td>
<td><img src="image1.png" alt="Diagnostics" /></td>
</tr>
<tr>
<td>2.</td>
<td>When the CPU is online, you can see all currently pending status and error messages. <strong>Software errors</strong> can be acknowledged via “MC_RESET” once they are fixed. A list of possible software errors can be found in STEP7 Basic online help.</td>
<td><img src="image2.png" alt="Diagnostics" /></td>
</tr>
<tr>
<td>3.</td>
<td><strong>Alarm drive</strong> After an error, which the servo drive detects, a restart is necessary.</td>
<td><img src="image3.png" alt="Diagnostics" /> RDY-LED on the CU305 flashes red. An error code is generated. It can be displayed using a BOP or with the STARTER software (see S110 manual for more detailed explanations) The servo drive is disabled and an alarm message bit is set</td>
</tr>
</tbody>
</table>
| 4. | Remove the disruption.  
   - Reset the alarm  
   - Reactivate the servo drive  
   - Reference the axis again | |

### Error

If an error is pending, you can reset it by clicking “Acknowledge”. For example, you can simulate an error by actuating a hardware limit switch. The respective last error message is displayed in the bottom line.
### 3 Configuration

<table>
<thead>
<tr>
<th>No</th>
<th>Instruction</th>
<th>Note/picture</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.</td>
<td>Some serious errors can only be removed by a “Power Off Reset”. Switch off the servo drive and then switch it back on again after a short while.</td>
<td>Note</td>
</tr>
</tbody>
</table>

**Note**

The technology object "axis" creates a global data block in which all parameters and the current state of the axis is stored. By entering the symbolic name of the axis, these values can be accessed in the user program during runtime.
3.7 Operator control with WinCC Runtime HMI

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

<table>
<thead>
<tr>
<th>No</th>
<th>Instruction</th>
<th>Comment/picture</th>
</tr>
</thead>
</table>
| 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” | ![Image of control panel settings] |
| 2. | Go back to the STEP7 Basic project *.al10 | ![Image of project navigation] |
| 3. | • Mark “CEx7_HMI” in project navigation  
   • Subsequently click the “Start runtime” icon | ![Image of project runtime] |

Via PC runtime you can test all the features described in this documentation. All important status messages are displayed. In case of an error a message text is displayed.

I/O fields highlighted in blue, provide only read access. In the yellow I/O fields values can also be written.

The bar diagram indicates the position of the axis. The ramp displayed shows whether the axis accelerates, decelerates or whether it moves at constant velocity. Flashing buttons indicate that an action must be performed.
In addition there is also a project for a KTP600 Basic Panel available.
4 Code Elements

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

Table 4-1

<table>
<thead>
<tr>
<th>No</th>
<th>File name</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>CE_x7_S7-1200_v2d0.zip</td>
<td>STEP 7 Basic V10.5 project</td>
</tr>
<tr>
<td>2.</td>
<td>CE_x7_STARTER_v2d0.zip</td>
<td>STARTER project</td>
</tr>
</tbody>
</table>
## History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1.0</td>
<td>03.10.2009</td>
<td>First version with universal servo drive</td>
</tr>
<tr>
<td>V2.0</td>
<td>19.01.2010</td>
<td>Second version with Sinamics S110</td>
</tr>
</tbody>
</table>
Certificate of Compliance

PLCopen hereby certifies that the product

SIMATIC S7-1200
Version V01.00.00
Release date 10-6-2009

as supplied by Siemens AG
fulfills the requirements as specified in
PLCopen Compliance Procedure for Motion Control Library V. 1.1 of April 9, 2005

Check www.plcopen.org for details on compliance of this product

E. van der Wal, Managing Director