SIEMENS

SIMATIC

S7-300
SM331;AI 8x12 Bit Getting Started
Part 2: Voltage and PT100

Getting Started

11/2006
A5E00264161-02

Safety Guidelines

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠ **Danger**

indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠ **Warning**

indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠ **Caution**

with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

Caution

without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.

Notice

indicates that an unintended result or situation can occur if the corresponding information is not taken into account.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The device/system may only be set up and used in conjunction with this documentation. Commissioning and operation of a device/system may only be performed by **qualified personnel**. Within the context of the safety notes in this documentation qualified persons are defined as persons who are authorized to commission, ground and label devices, systems and circuits in accordance with established safety practices and standards.

Prescribed Usage

Note the following:

⚠ **Warning**

This device may only be used for the applications described in the catalog or the technical description and only in connection with devices or components from other manufacturers which have been approved or recommended by Siemens. Correct, reliable operation of the product requires proper transport, storage, positioning and assembly as well as careful operation and maintenance.

Trademarks

All names identified by ® are registered trademarks of the Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Automation and Drives
Postfach 48 48
90437 NÜRNBERG
GERMANY

Order No.: A5E00264161-02
11/2006

Copyright © Siemens AG 2006.
Technical data subject to change

Table of contents

Preface 1

1.1 General information

Purpose of the Getting Started manual

The Getting Started guide gives you a complete overview of the commissioning of analog module SM331. It supports you when installing and parameterizing the hardware of a voltage measuring transducer and a PT100 resistance thermometer. In addition, you will receive an introduction for configuring the analog module with the SIMATIC S7 manager.

The intended readership of Getting Started is a novice with only basic experience in configuring, commissioning, and servicing automation systems.

What to expect

The procedures, from mounting the module to storing analog values in the STEP7 user program, are explained step-by-step and in detail based on an example. In the following sections you will be introduced to:

- Problem analysis
- Mechanical setup of the example station
- Electrical connection of the example station using conventional wiring
- Electrical connection of the example station using SIMATIC TOP connect system cabling
- Configuration using the SIMATIC Manager
- Creating a small user program with STEP7 which stores the read analog value in a data block
- Triggering and interpreting diagnostic and hardware interrupts

Prerequisites 2

2.1 Basics

Basic knowledge

No special knowledge of the field of automation technology is required in order to understand the Getting Started guide. As the configuration of the analog module is done with the STEP7 software, proficiency in STEP7 would be advantageous.

Further information on STEP7 can be found in the electronic manuals that are supplied with STEP7.

You will also need to know how to use computers or PC-like equipment (such as programming devices) with the Windows 95/98/2000/NT or XP operating systems.

Required hardware and software

The scope of delivery of the analog module consists of 2 parts:

- SM331 module
- Front panel connector for convenient connection of the power supply and data cables.

Analog module components

Quantity	Item	Order no.
1	SM 331, Electrically ISOLATED 8 AI, ALARM DIAGNOSTICS	6ES7331-7KF02-0AB0
1	20-pin FRONT CONNECTOR with spring contacts Alternatives: - FRONT CONNECTOR WITH 20-PIN SCREW-TYPE CONTACTS - FRONT CONNECTOR MODULE W/ FLAT ROUND CABLE CONNECTION (TOP connect system cabling)	6ES7392-1BJ00-0AA0 6ES7392-1AJ00-0AA0 6ES7921-3AF00-0AA0
1	SIMATIC S7 SHIELDED SUPPORT ELEMENT	6ES7390-5AA00-0AA0
2	SIMATIC S7, SHIELDED CONNECTION TERMINAL F. 1 CABLE W/ 4...13MM DIAM	6ES7390-5CA00-0AA0

The general SIMATIC components required for the example are as follows:

SIMATIC material for the example station

Quantity	Item	Order no.
1	POWER SUPPLY LOAD PS 307 AC 120/230V, DC 24V, 5A (incl. power supply jumper)	6ES7307-1EA00-0AA0
1	CPU 315-2 DP	6ES7315-2AG10-0AB0
1	MICRO MEMORY CARD, NFLASH, 128KBYTES	6ES7953-8LG00-0AA0
1	SIMATIC S7-300, MOUNTING RAIL L=530MM	6ES7390-1AF30-0AA0
1	Programming device (PD) with MPI interface and MPI cable PC with suitable interface card	depending on the configuration

If you would like to carry out the example station using SIMATIC TOP connect, you will need the following additional components:

SIMATIC Top connect components

Quantity	Item	Order no.
1	FRONT CONNECTOR MODULE W/ FLAT ROUND CABLE CONNECTION FOR ANALOG MODULES OF THE S7-300 POWER SUPPLY VIA SPRING CLIPS	6ES7921-3AF00-0AA0
2	TERMINAL BLOCK TPA,3-ROW FOR ANALOG MODULES OF THE SIMATIC S7;CONNECTION USING FLAT ROUND CABLE CONNECTIONS VIA SPRING CLIPS	6ES7924-0CC00-0AB0
2	SHIELDING PLATE FOR ANALOG TERMINAL BLOCK	6ES7928-1BA00-0AA0
4	CONNECTOR (RIBBON CABLE CONNECTOR) AS PER DIN 41652, 16-PIN, SNAP-ON/SCREW-ON	6ES7921-3BE10-0AA0
2	SIMATIC S7, SHIELDED CONNECTION TERMINAL F. 1 CABLE W/ 4...13MM DIAM	6ES7390-5CA00-0AA0
2	SIMATIC S7, SHIELDED CONNECTION TERMINAL. F. 2 CABLES W/ 2...6MM DIAM EACH	6ES7390-5AB00-0AA0
1	FLAT ROUND CABLE WITH 16 WIRES 0.14 MM^2 LENGTH: 30 M SHIELDED	6ES7923-0CD00-0BA0

STEP 7 software

Quantity	Item	Order no.
1	STEP 7 Software version 5.2 or later, installed on the programming device.	6ES7810-4CC06-0YX0

You can use the following resistance-based sensors and voltage measuring transducers to detect the analog signals:

Resistance-based sensors and voltage measuring transducers

Quantity	Item	Order no.
1	±5V voltage measuring transducer	depends on the manufacturer
3	PT100 Standard	depends on the manufacturer

Note

"Getting Started" only describes the handling of PT100 standard voltage measuring transducers and resistance thermometers. If you wish to use other transducers, you will need to wire and configure an SM331 differently.

The following tools and materials will also be needed:

General tools and materials

Quantity	Item	Order no.
Various	M6 bolts and nuts (length depends on the mounting position)	commonly available
1	Screwdriver with 3.5 mm blade	commonly available
1	Screwdriver with 4.5 mm blade	commonly available
1	Side cutter and cable stripper	commonly available
1	Crimp tool	commonly available
X m	Cable for grounding the mounting rail with 10 mm^2 cross-section, ring terminal with 6.5 mm hole, length appropriate for local requirements.	commonly available
X m	Flexible wire with $1mm^2$ diameter with fitting wire end sleeves, form A in 3 different colors – blue, red, and green	commonly available
X m	3-wire power cord (AC 230/120V) with protective contact socket, length according to local conditions	commonly available
1	Calibration device (measuring instrument for commissioning, which can measure and supply current)	depends on the manufacturer

Task

3

3.1 Example of an application

Overview

The Getting Started manual leads you all the way through an example application during which you will connect the following four sensors:

- A pressure sensor, which is connected to a voltage measuring transducer (±5V).
- Three resistance thermometers, type PT100

They will initiate fault diagnostics and hardware interrupts. Analog input module SM331, AI8x12 Bit (order number 6ES7 331-7KF02-0AB0) is available to you.

The module can initiate diagnostic alarms and hardware interrupts. It can process up to 8 analog inputs. Various types of measurements can be set for each module (e.g. current measurements, a voltage measurement, PT 100, thermocouple).

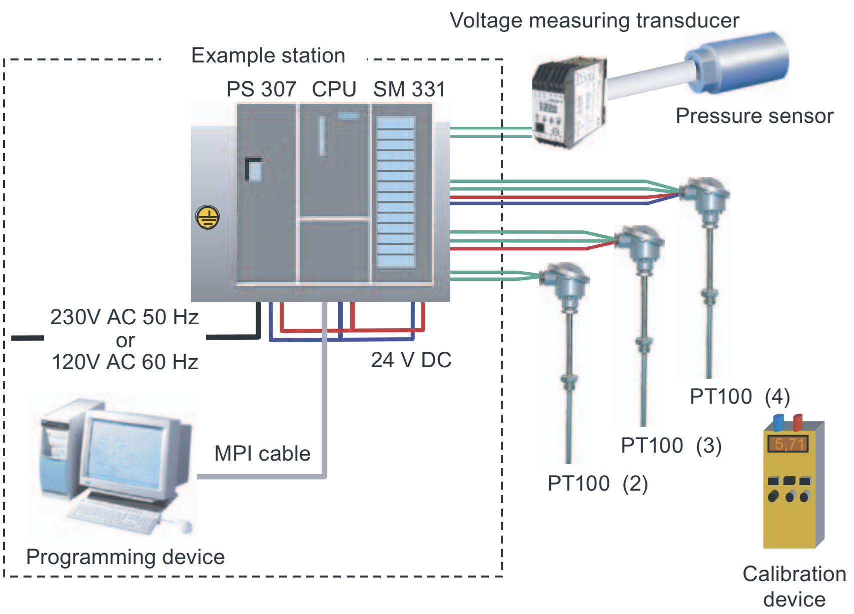


Figure 3-1 Components of the example station

In the following sections you will be introduced to:

- Mechanical setup of the example station
  - General mounting instructions for S7-300 modules
  - Configuration of the SM331 for the two selected measurement transducer types
- Electrical connection of the example station
  - Wiring the power supply module and the CPU
  - Connecting the analog module in the conventional manner
  - Connecting the analog module using the SIMATIC TOP connect system cabling
- Configuration of the SIMATIC Manager
  - Using the project wizard
  - Completing the automatically generated hardware configuration
  - Integrating the supplied user program source
- Testing the user program
  - Interpreting the read values
  - Converting the measured values into readable analog values
- Utilizing the diagnostic capabilities of the SM331 module
  - Triggering a diagnostic interrupt
  - Evaluating the diagnostics:
- Application of hardware interrupts
  - Configuration of hardware interrupts
  - Configuration and analysis of hardware interrupts

See also

Assembling the example station (Page 4-1)

Wiring the power supply and the CPU (Page 5-1)

Create a new project (Page 6-1)

Downloading the system data and user program (Page 7-1)

Initiating the diagnostic interrupt (Page 8-1)

Hardware interrupt (Page 9-1)

4 Mechanical setup of the example station

4.1 Assembling the example station

Overview

The setup of the example station is divided into two steps. First, the setup of the power supply and the CPU is explained. After you become acquainted with analog module SM331, the mounting of it is described.

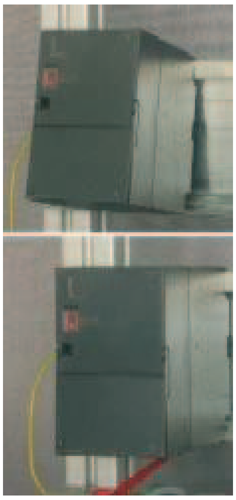
Prerequisites

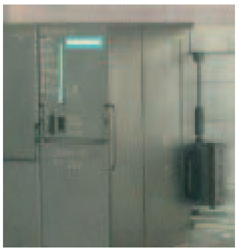
Before you can use analog input module SM331, you need a basic setup of general SIMATIC S7-300 components.

The order of the assembly takes place from left to right:

- Power supply PS307
- CPU 315-2 DP
- Analog module SM331

Instructions (without SM331)

Step	Graphic	Description
1		Screw on the mounting rail (screw size: M6) so that at least 40 mm space remains above and below the rail. When mounting it on a grounded metallic panel or on a grounded device mounting panel made of steel sheet, make sure you have a low impedance connection between the mounting rail and the mounting surface. Connect the rail to the protective conductor. An M6 screw is provided on the rail for this purpose.
2		Mounting the power supply: : • Hang the power supply on to the top end of the rail • Screw it tight to the rail underneath

Step	Graphic	Description
3		Connect the bus connector (delivered with the SM331) to the **left** connector on the back of the CPU
4		Mounting the CPU: : • Hang the CPU on to the top end of the rail • Push it all the way left to the power supply • Push it down • Screw it tight to the rail underneath

4.2 Mounting of the analog module

4.2.1 General information

Overview

Prior to mounting the SM331, you must connect the measuring range modules accordingly.

In this section, you will learn about:

- The components you need
- The properties of the analog input module
- What a measuring range module is and how it is configured
- Mounting a configured module

See also

Measuring range modules (Page 4-7)

4.2.2 Components of the SM331 with conventional connecting plug

Overview

A functional analog module consists of the following components:

- Module SM331 (in our example 6ES7331-7KF02-0AB0)
- 20-pin front connector There are two different types of front connectors:
  - With spring clamp (order number 6ES7392-1BJ00-0AA0)
  - With screw contacts (order number 6ES7392-1AJ00-0AA0)

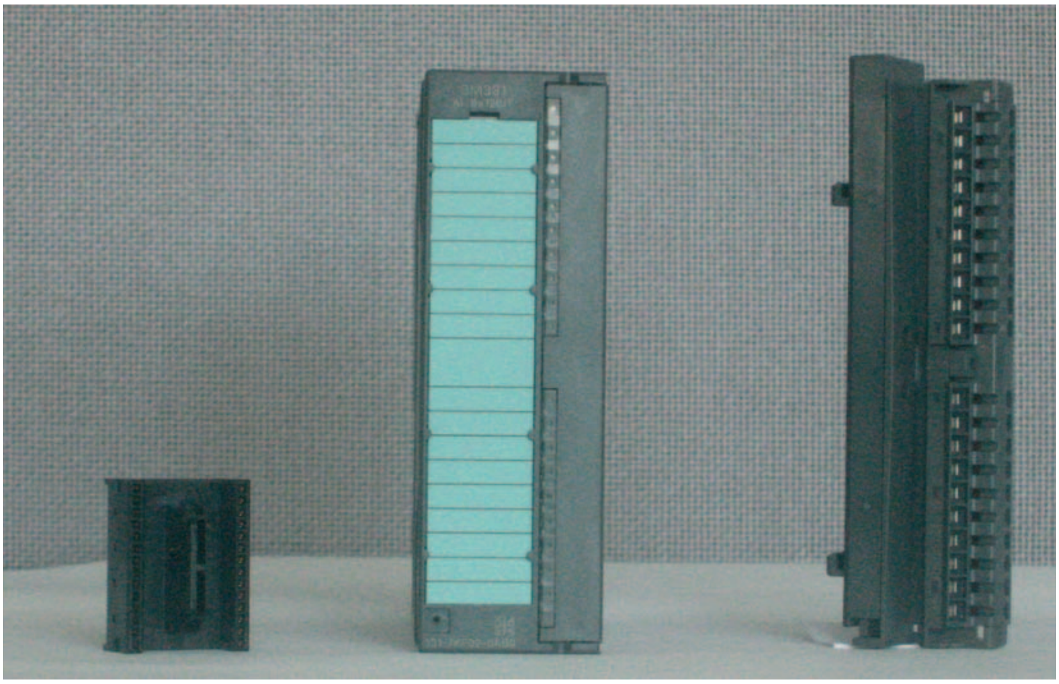


Figure 4-1 Components of the SM331

4.2.3 SM331 with the SIMATIC TOP connect system cabling

Overview

The SIMATIC TOP connect system cables for the SM331 module consist of the following components

- Front connector module (order number 6ES7921-3AF00-0AA0)
- Terminal block TPA (order number 6ES7924-0CC00-0AB0)
- Various small components

Figure 4-2 Components of the SM331 with the SIMATIC TOP connect system cables

See also

Basics (Page 2-1)

4.2.4 Features of the analog module

Features

The module is a universal analog module that can be used with the most commonly used applications.

The desired measuring mode should be set up directly on the module with the measuring range modules.

- 8 inputs in 4 channel groups (each group with two inputs of same type)
- Measurement resolution adjustable for each channel group
- User-defined measuring mode per channel group:
  - Voltage
  - Current
  - Resistor
  - Temperature
- Programmable diagnostic interrupt
- Two channels with limit alarms (only channel 0 and channel 2 are configurable)
- Electrically isolated against backplane bus connection
- Electrically isolated against load voltage (exception: at least one measuring range module is set to position D)

Scope of supply of module SM331 (order no.: 6ES7331-7KF02-0AB0):

Components
Analog module SM331
Labeling strips
Bus connectors
2 cable ties (not in the picture above) to mount the external wiring

See also

Measuring range modules (Page 4-7)

4.2.5 Measuring range modules

Connection

Module SM331 has 4 measuring range modules (one measuring range module per channel group). You can plug each measuring range module into 4 different positions (A, B, C, or D). The position enables you to specify the transducer to be connected to the respective channel group.

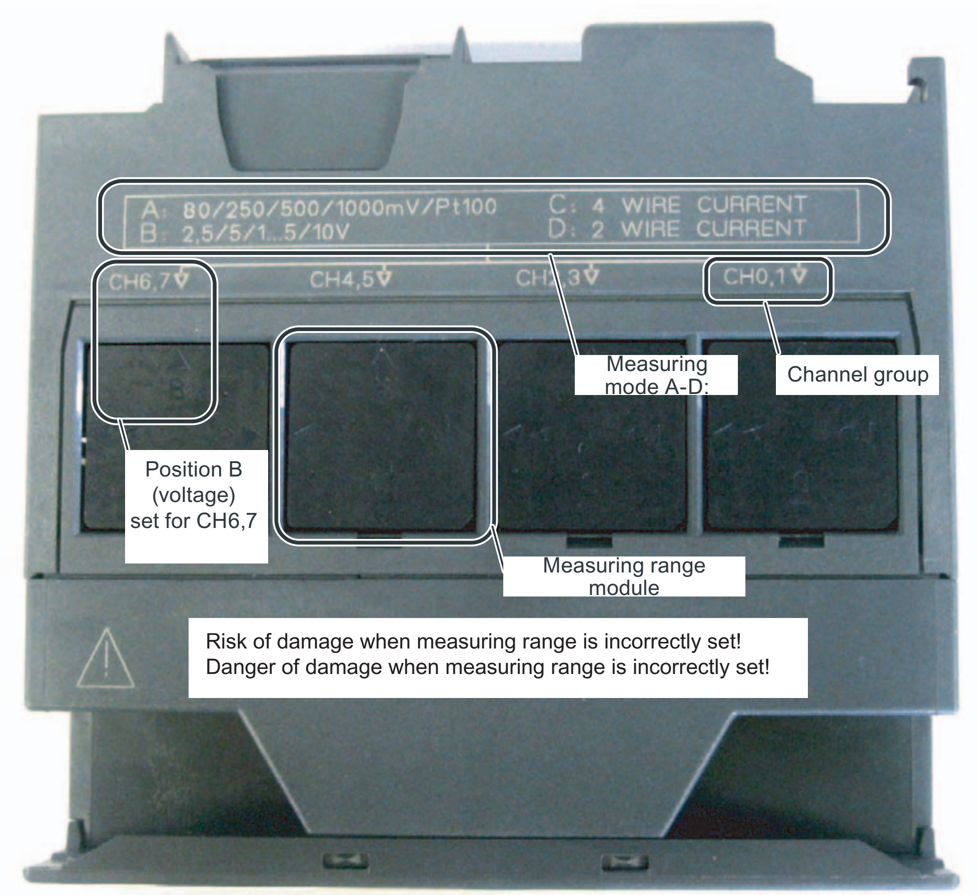


Figure 4-3 4 measuring range modules with default setting B (voltage)

Positions of the measuring range modules

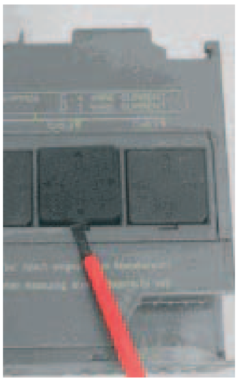
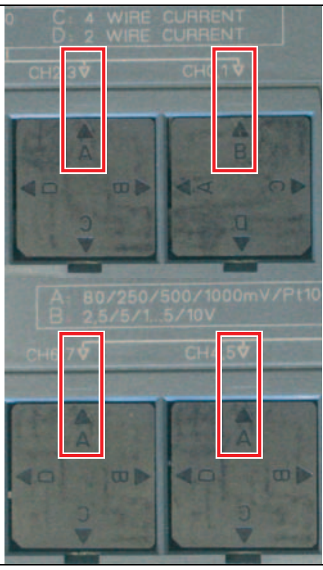
Position	Type of measurement
A	Thermocouple / resistance measurement
B	Voltage (factory setting)
C	Current (4-wire transducer)
D	Current (2-wire transducer)

In our example task, a sensor with a ±5V voltage measuring transducer is connected to channel group CH0,1 at input 0.

For connecting the three type PT100 resistance thermometers, you need a complete channel group (CH2,3/CH4,5/CH6,7) for each PT100.

The first measuring range module of channel group CH0,1 therefore contains position B (default setting). You must switch the other modules to position A.

Positioning of the measuring range modules

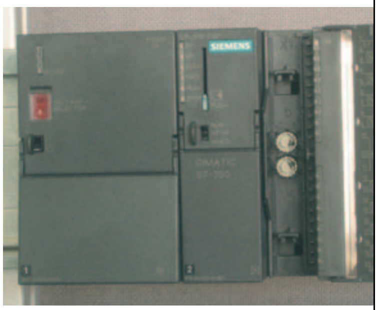
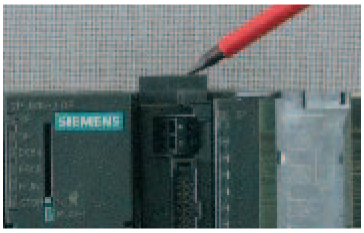
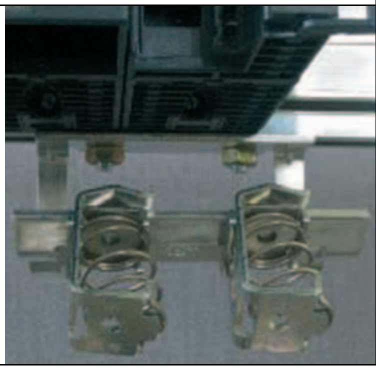
Step	Graphic	Description
1		Using a screwdriver, remove the two measuring range modules
2		Turn the measuring range modules to the desired position:
3		Plug the measuring range modules back into the module In our example, the modules should have the following positions: CH0,1: B CH2,3: A CH4,5: A CH6,7: A

4.2.6 Mounting the SM331 module

Mounting the SM331

After you have prepared the analog module accordingly, mount it to the rail as well.

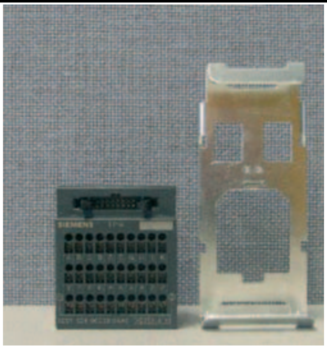
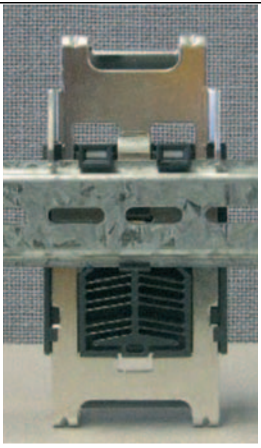
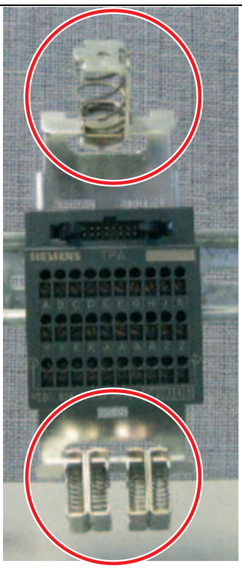
Mounting the SM331 module

Step	Graphic	Description
1		Mounting the SM331: • Hang the SM331 to the top end of the rail • Push it all the way to the left up to the CPU • Push it down • Screw it tight to the rail underneath
2		Mounting the front connector: • Press the upper front connector button. • Insert the connector into the module until the connector button snaps into the top position.
3		Mounting the shielded plate: • Tighten the shielded plate to the underside of the mounting rail. • Connect the two shielded connection terminals to the shielded plate.

4.2.7 Mounting the TOP connect terminal block

Procedure

The TOP connect system cables need a system-specific terminal block.

Step	Graphic	Description
1		Connect the terminal block to the shielded support element.
2		Latch the terminal block with the shielded support element onto a top-hat rail.
3		Install the shielded element terminals onto the shielded support element.

The example station is now mechanically mounted.

Electric connection 5

5.1 Wiring the power supply and the CPU

Overview

> **Warning**
> You might get an electrical shock if power supply module PS307 is turned on or the power cord is connected to the main power supply.
> Always switch off power before you start wiring the S7-300.

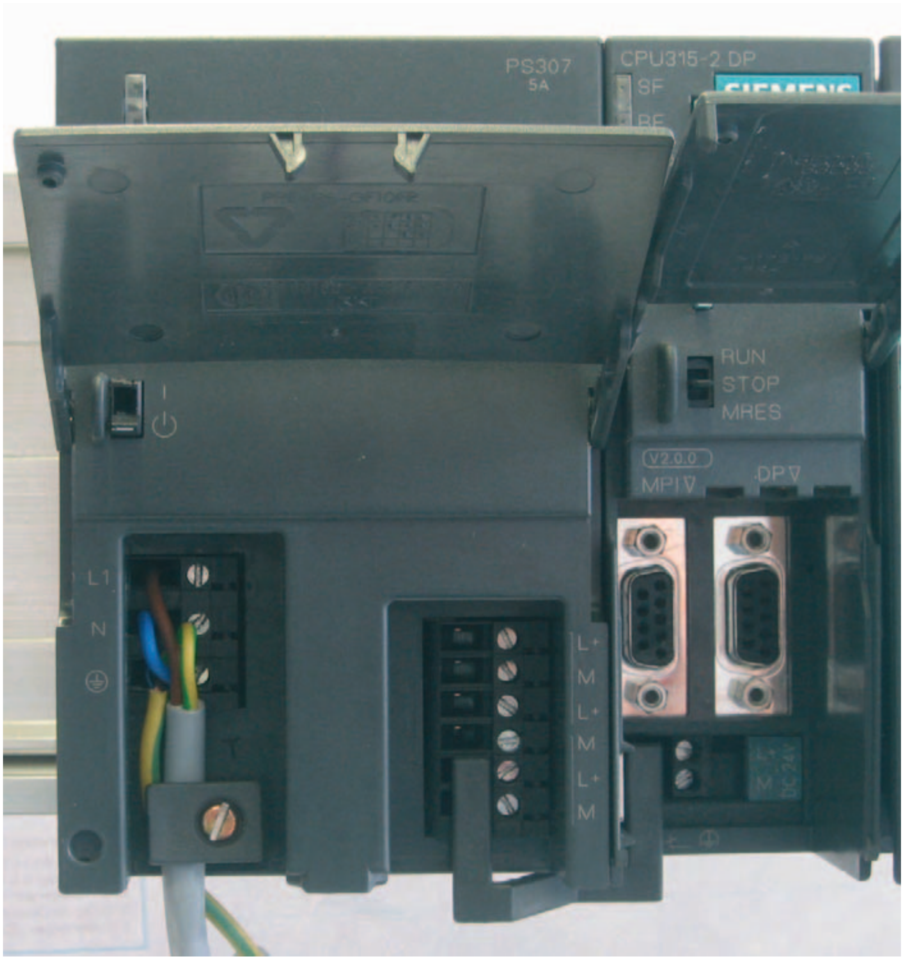
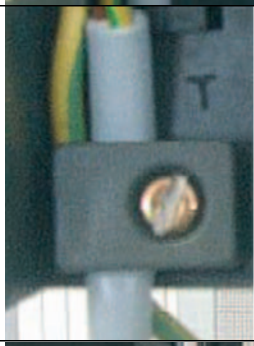


Figure 5-1 Wiring the power supply and the CPU

The example station requires a power supply. The wiring is done as follows:

Wiring the power supply and the CPU

Step	Graphic	Description
1		Open the front panel covers of the power supply module and CPU.
2		Unscrew the strain relief clamp on the power supply module.
3		Remove the insulation from the power cord, attach the cable end sleeves (for stranded conductors), and connect them to the power supply.
4		Screw the strain relief clamp in place.
5		Insert the power supply jumper between the power supply and the CPU and tighten it securely. Info on the grounding slide switch of the CPU: • Insert: non-isolated (as-delivered state) • Remove: Isolated You do not have to change the grounding slide switch of the CPU, because the SM331 is already isolated.
6		Check that the line voltage selector switch is set to the correct line voltage. The default line voltage setting for the power supply module is 230 VAC. To change the setting, proceed as follows: Remove the protective cap using a screwdriver, set the switch to the existing main power supply, and re-insert the protective cap. .

5.2 Wiring of the analog module

5.2.1 Shielded lines for analog signals

General information

The wiring of analog module SM331 is independent of the type of the analog measuring transducer.

Cables

You should use shielded and twisted-pair lines for analog signals. This reduces the effect of interference. You should ground the shield of the analog lines at both ends of the line.

If there are differences in potential between the ends of the cables, equipotential current may flow across the shield, which could disturb the analog signals. In this case, the shield should only be grounded at one end of the line or an adequately dimensioned equipotential bonding conductor should be routed.

5.2.2 Voltage measuring transducer wiring principle

Wiring the voltage measuring transducer:

A voltage measuring transducer should be wired as follows:

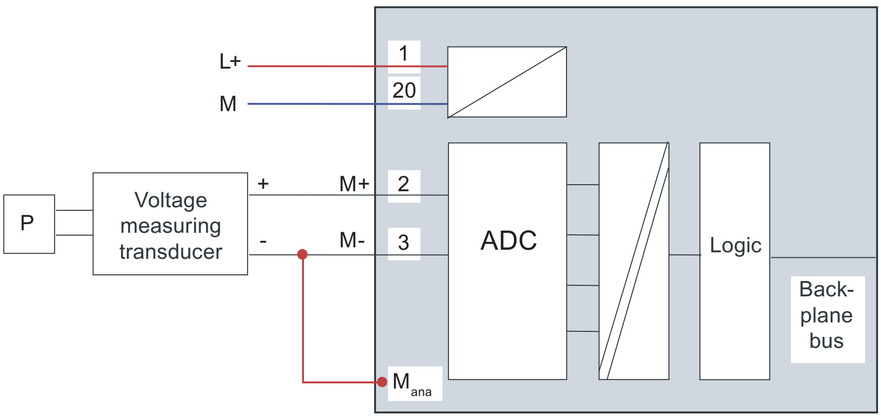


Figure 5-2 Wiring principle: Voltage measuring transducer for an isolated SM331

If you use the SM331 module in areas of heavy EMC interference, you should connect M- to M_{ana}. In this way, the potential difference between the inputs and the M_{ana} reference potential will not exceed the permitted value.

5.2.3 Wiring principle of the resistance thermometer (PT100)

Options

There are three options for wiring the resistance thermometer:

- Four-wire connection
- Three-wire connection
- Two-wire connection

For a 4-wire and 3-wire connection, the module provides a constant current via terminals Ic+ and Ic-, which compensates for the voltage drop of the measuring lines.

It is imperative at this point to wire the constant current cables directly to the resistance thermometer.

Note

Compensated measurements with 3- or 4-wire elements return a more precise result than that of 2-wire measurements.

4-Wire Connection of a Resistance Thermometer

The voltage arising at the resistance thermometer is measured at connections M + and M -.

When connecting, observe the polarity of the connected lines Ic+ / M+ and Ic- / M- and ensure that the lines are connected directly to the resistance thermometer.

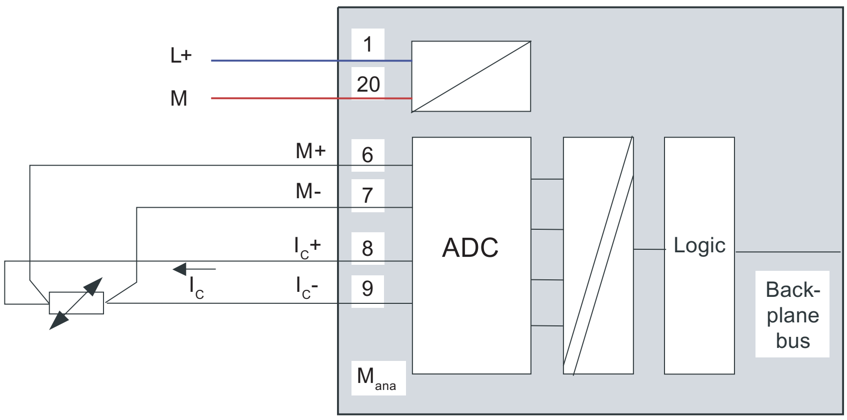


Figure 5-3 Wiring: 4-Wire Connection of a Resistance Thermometer

3-Wire Connection of a Resistance Thermometer

For a 3-wire connection, you must generally place a jumper between M- and Ic-.

When connecting, ensure that connected lines Ic+ and M+ are connected directly to the resistance thermometer.

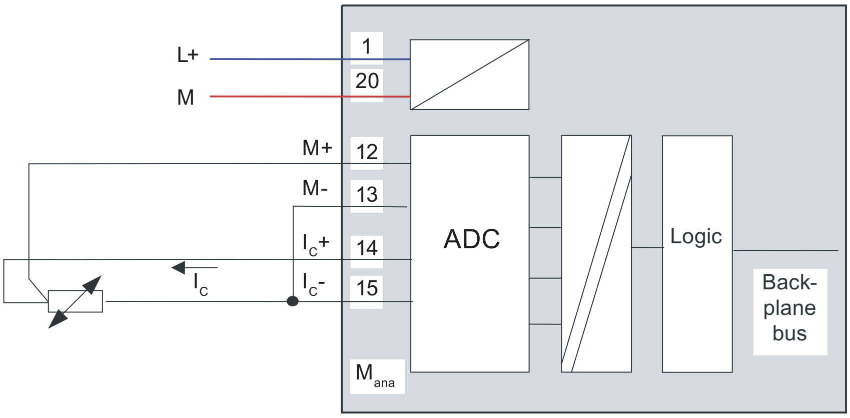


Figure 5-4 Wiring: 3-Wire Connection of a Resistance Thermometer

2-Wire Connection of a Resistance Thermometer

For a 2-wire connection, you must insert a jumper between terminals M+ and Ic+ on the front connector of the module and another jumper between terminals M- and Ic-.

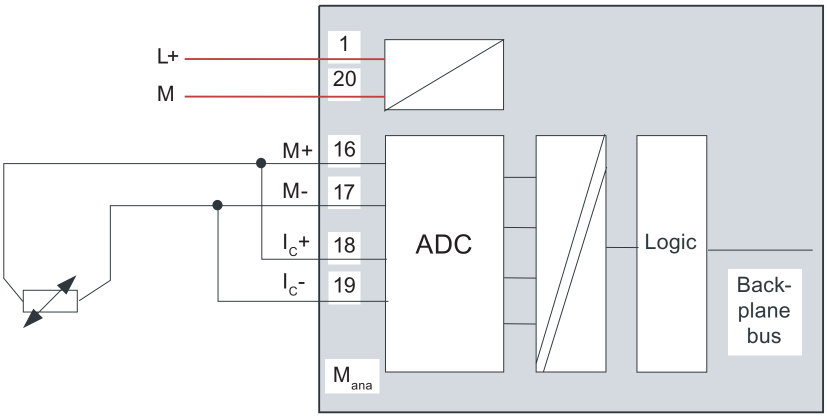


Figure 5-5 Wiring: 2-Wire Connection of a Resistance Thermometer

5.2.4 Connecting the analog module in the conventional manner

General information

This section encompasses the wiring of the analog modules to individual lines in the conventional manner. The type of connection using TOP connect system cables can be found in the corresponding section.

Tasks

The wiring of the analog module consists of the following tasks:

- Connecting the power supply (red cable)
- Connecting the voltage measuring transducer (green cable)
- Connecting the unused channel of a channel group in parallel
- Wiring of the first PT100 with a 4-wire connection (green cable)
- Wiring of the first PT100 with a 3-wire connection (green cable)
- Wiring of the first PT100 with a 3-wire connection (green cable)
- Wiring the ground (blue cable)

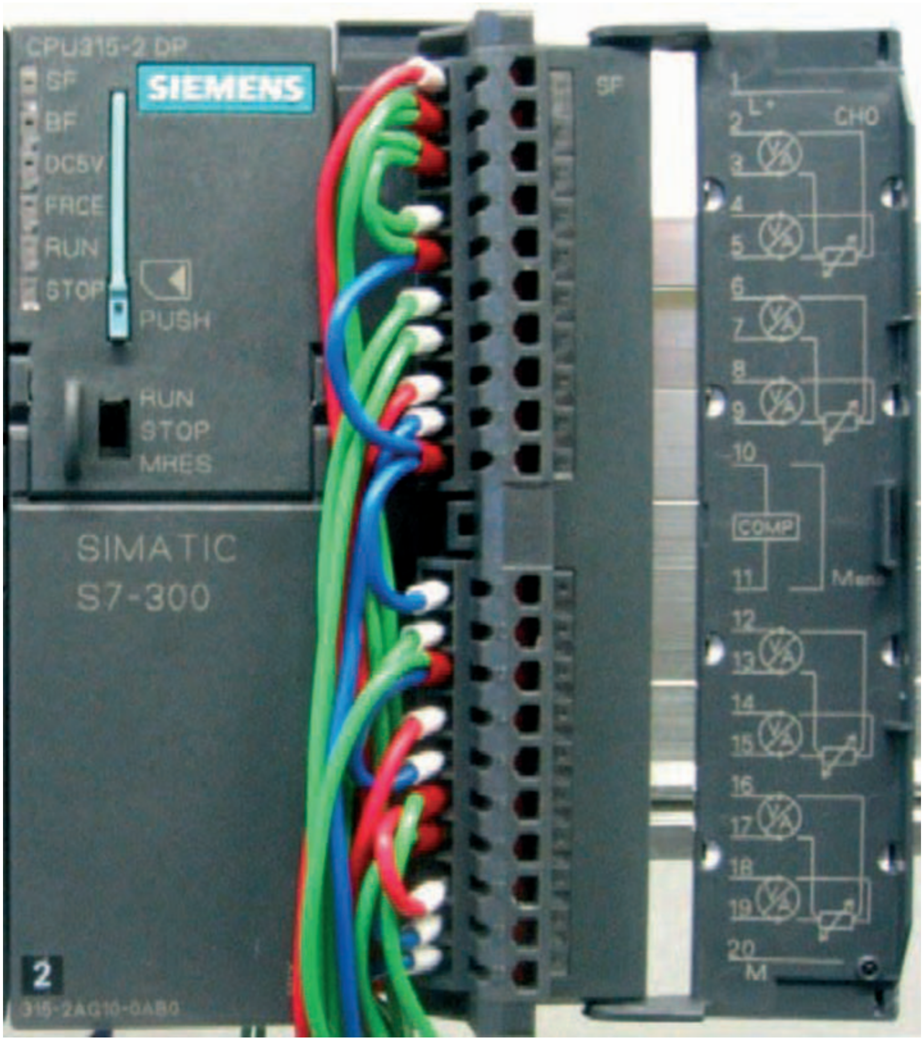
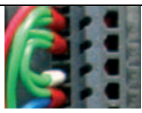
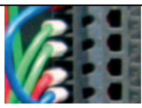
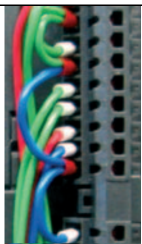
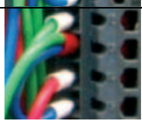


Figure 5-6 SM331 Front connector wiring

Procedure

The required wiring tasks are explained below step-by-step:

SM331 Front connector wiring

Step	Graphic	wiring	Comments
1		Open the front door of the SM331	The terminals are printed on the front door
2		Remove 6 mm of the insulation from the ends of the wires that go into the front connector. Attach cable end sleeves to these ends.	
3		Wire the front connector as follows: Terminal 1: L+	Power supply of the module
4		Terminal 2: M+ sensor 1 Terminal 3: M- sensor 1 Connect inputs in parallel: Connect terminal 2 to 4 Connect terminal 3 to 5	Standard wiring for voltage measuring transducers to an isolated module To maintain the diagnostic functionality of channel group 0, you must connect the second unused input to the first in parallel
5		Terminal 6: M+ PT100 (4-wire) Terminal 7: M- PT100 (4-wire) Terminal 8: Ic+ PT100 (4-wire) Terminal 9: Ic- PT100 (4-wire)	Standard wiring of a PT100 with a 4-wire connection
6		Connect terminal 10 (Comp) to Mana Connect terminal 11 (Mana) to terminals 3 and 5	Comp is not used for voltage measuring and PT100 Recommended for voltage measuring transducers
7		Terminal 12: M+ PT100 (3-wire) Terminal 13: M PT100 (3-wire) Terminal 14: PT100 (3-wire) Connect terminal 15 (Ic-) to 13 (M-)	Standard wiring of a PT100 with a 3-wire connection
8		Terminal 16: M+ PT100 (2-wire) Terminal 17: M PT100 (2-wire) Connect terminal 18 (Ic+) to 16 M+ Connect terminal 19 (Ic-) to 17 (M-)	Standard wiring of a PT100 with a 2-wire connection
9		Terminal 20: G	Ground

See also

Wiring the analog module using the system cables of TOP connect (Page 5-10)

Measuring range modules (Page 4-7)

5.2.5 Wiring the connection terminals

Overview

In our example, a terminal strip replaces the connections of the voltage sensor or of the resistance thermometer. The voltages are specified using a calibration device; the resistance thermometer is simulated by a potentiometer.

Voltage measurement

In our example, we are simulating the voltage measuring transducer via the following connection:

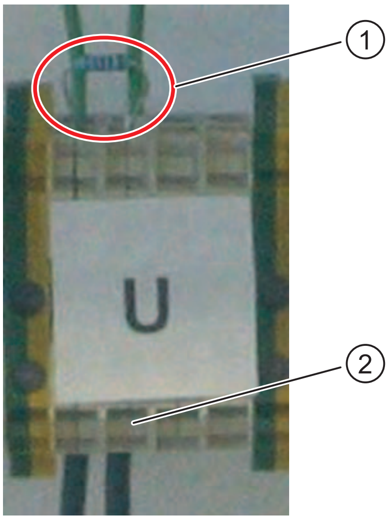


Figure 5-7 Terminal connection of the voltage measuring transducer

(1) 750 Ohms resistance
(2) Calibration device terminal for specifying the voltage

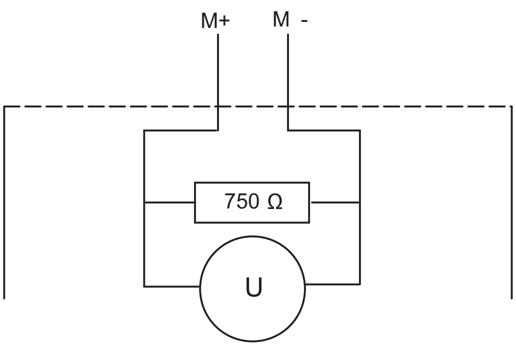


Figure 5-8 Block diagram of the voltage measuring transducer

The type of protective circuit required for your voltage measuring transducer can be found in the manual for your voltage sensor.

PT100 resistance thermometers

If you want to connect a PT100, then you must also wire the terminals to the

resistance thermometers as explained in the corresponding section.

In our example station, a terminal strip replaces the terminals of the resistance thermometer. The desired resistance value is set using a potentiometer.

To simulate the lines, we use resistors. The 5 Ohm resistor simulates a copper wire with a cross-section of 0.6 mm2 and a length of 171.4 m.

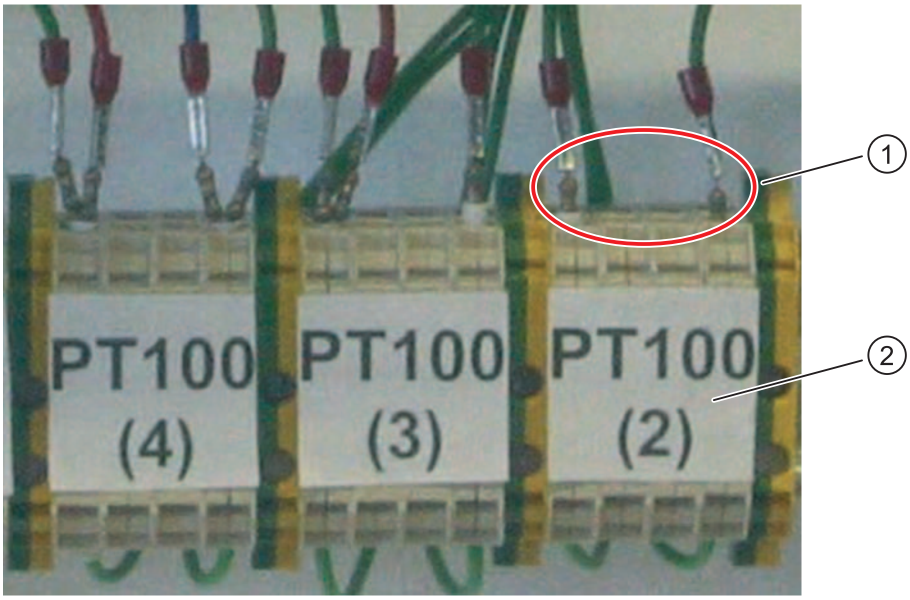
Using the following formula, the length of the conductor is calculated from the resistance:

$R = (\rho * l) : q$

$l = (R * q) : \rho$

R:	Line resistance
ρ:	Specific resistance of the conductor material (copper, 0.0178Ω mm²/m)
q:	Conductor cross-section
l:	Conductor length

M+ Ic+ Ic - M - M+ Ic+ M - M+ M -



Terminal connection of the PT100

(1)	5-Ohm resistors for simulating the conductor length
(2)	PT100 simulated

See also

Wiring principle of the resistance thermometer (PT100) (Page 5-4)

5.2.6 Wiring the analog module using the system cables of TOP connect

Overview

Using the SIMATIC TOP connect system cables, route the sensor-specific wiring from the analog module to the TOP connect terminal block.

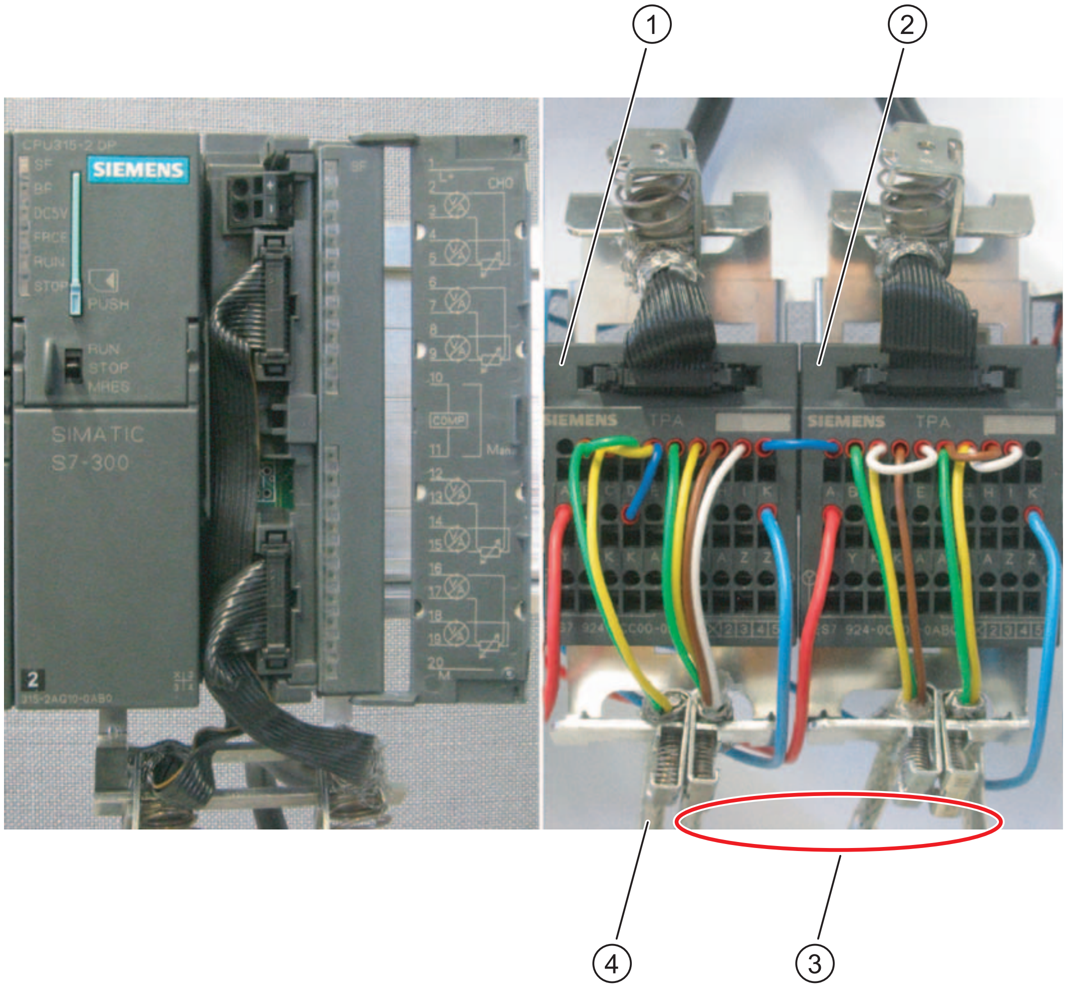
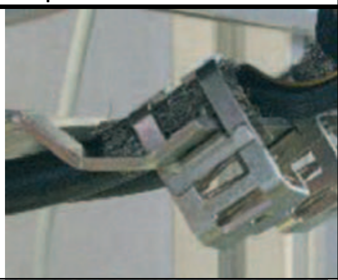
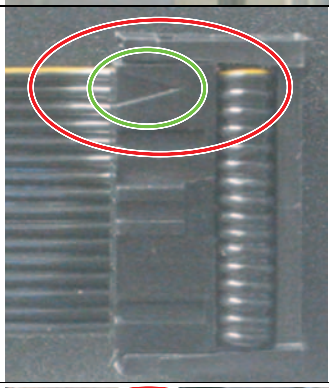
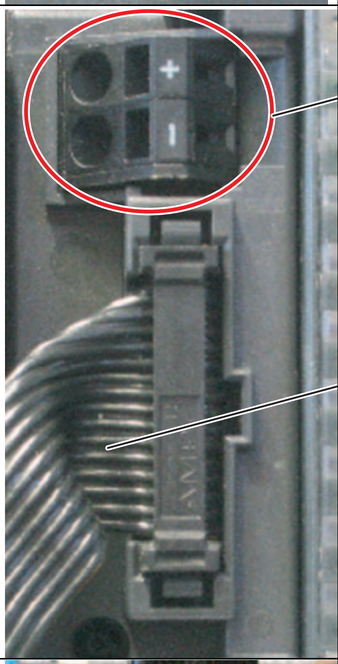
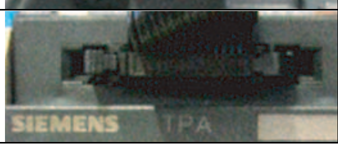


Figure 5-9 TOP connect connection

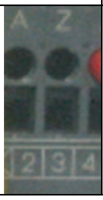
(1)	Terminal block 1
(2)	Terminal block 2
(3)	For resistance thermometer PT 100
(4)	For the voltage measuring transducer

The following table describes the individual tasks of the wiring for the connection to terminal block 1 step-by-step. The connection of terminal block 2 is done in the same manner.

SM331 Front connector wiring

Step	Graphic	wiring
1		Strip the protective jacket from the Top connect flat round cable to the appropriate length and expose the 16-pin flat ribbon cable. Shorten the shielded cable to approx. 15 mm and turn it up and back. Insert the flat round cable into the shielded connection terminal.
2		Guide the exposed flat ribbon cable into the snap-on/screw-on connector and press on it lightly. Make sure that the identification triangle of the plug (green circle) and the wire marked in yellow are on the same side.
3		Now plug the 16-pin flat connector into the front connector of the analog module. (1) If you need more than 4A current (this is not the case in our example), you must establish the power supply of the module directly via the terminals of the front connector of the SM331 (see red circle). Cabling to terminal block 1
4		Plug the other end of the flat round cable into the terminal block

Wiring the SM331 TOP connect terminals

Step	Graphic	wiring	Comments
1		Terminal block 1 and 2: Terminal Y: Power supply of the module	For a requirement of up to 4A of current, the power supply of the module can be routed across the terminal blocks. For higher requirements of current, the power supply must be directly connected to the front connector of the module.
2		Terminal block 1: Terminal B: M+ Voltage Transducer Terminal C: M- Voltage Transducer Connect terminals E and K Connect inputs in parallel: Connect terminal B to D Connect terminal C to E	Standard wiring for voltage measuring transducers on an isolated module To maintain the diagnostic functionality of channel group 0, you must connect the second unused input to the first in parallel
3		Terminal block 1: Terminal F: M+ PT100 (4-wire) Terminal G: M- PT100 (4-wire) Terminal H: Ic+ PT100 (4-wire) Terminal I: Ic+ PT100 (4-wire)	Standard wiring of a PT100 with a 4-wire connection
4		Terminal block 1: Connect terminal K Comp to A Mana	Comp is not used for voltage measuring and PT100 Recommended for voltage measuring transducers
5		Terminal block 2: Terminal B: M+ PT100 (3-wire) Terminal C: M- PT100 (3-wire) Terminal D: Ic+ PT100 (3-wire) Terminal E: Connect Ic- to terminal C M-	Standard wiring of a PT100 with a 3-wire connection
6		Terminal block 2: Terminal F: M+ PT100 (2-wire) Terminal G: M- PT100 (2-wire) Terminal H: Connect Ic+ to F M+ Terminal I: Connect Ic to G M-	Standard wiring of a PT100 with a 2-wire connection
7		Terminal block 2: Terminal Z: M	Grounding terminal

Note

If you need electrical isolation between the CPU and the analog module, you must supply the analog module with a separate power supply.

5.2.7 Wiring of a PT100

wiring

The figure clarifies the connection of a PT100 with a 4-wire connection. The cables are wired together inside the PT100 itself.

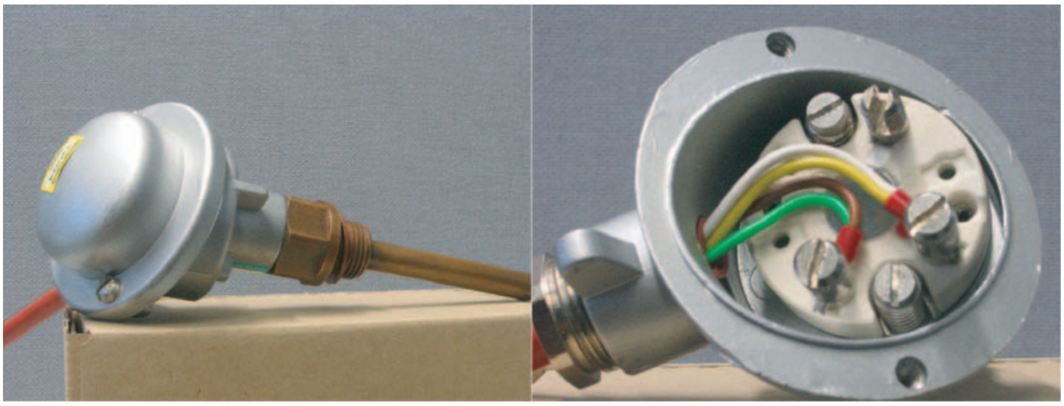


Figure 5-10 Wiring of the PT100 with a 4-wire connection

5.2.8 Checking the wiring

Procedure

If you want to test your wiring, switch the power supply on.

Do not forget to set the CPU to STOP (see the red circle).

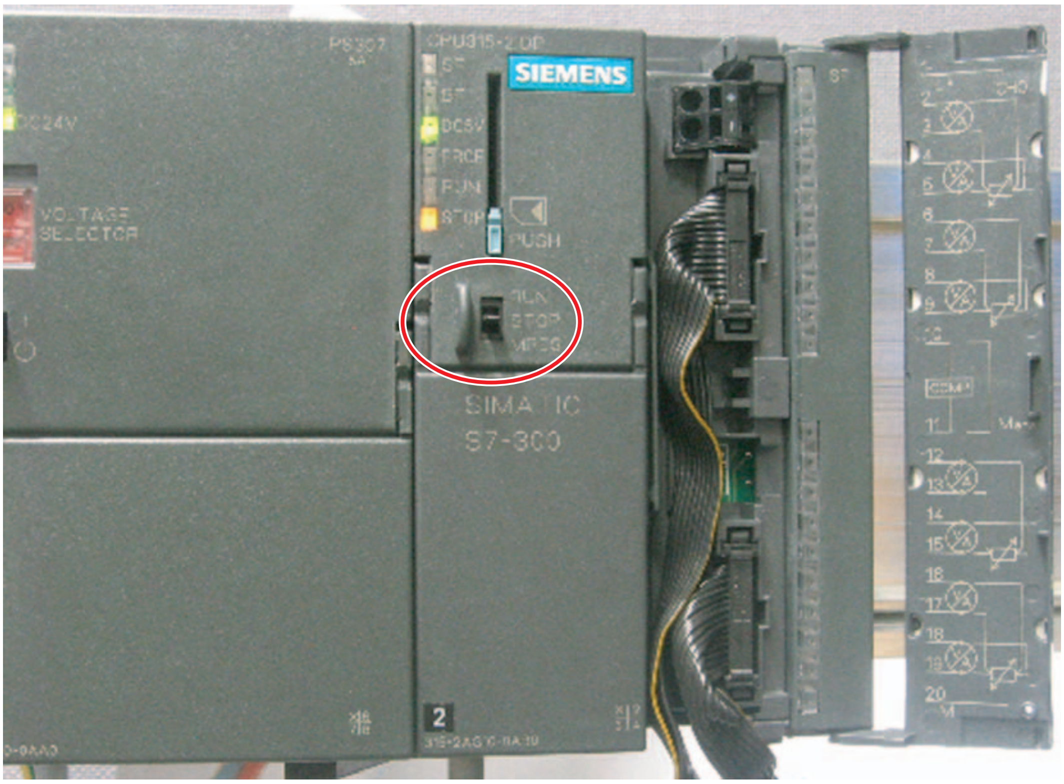


Figure 5-11 Successful wiring, CPU in STOP position

If a red LED is lit, an error has occurred in the wiring. Verify your wiring..

6 Configuration of the SIMATIC Manager

6.1 Creating a new STEP 7 project

6.1.1 Create a new project

"New Project..." wizard

Use SIMATIC Manager STEP7 V5.2 or later for configuring the new CPU 315-2 DP.

Start the SIMATIC Manager by clicking the "SIMATIC Manager" icon on your Windows Desktop and create a new project with the "New Project" wizard.

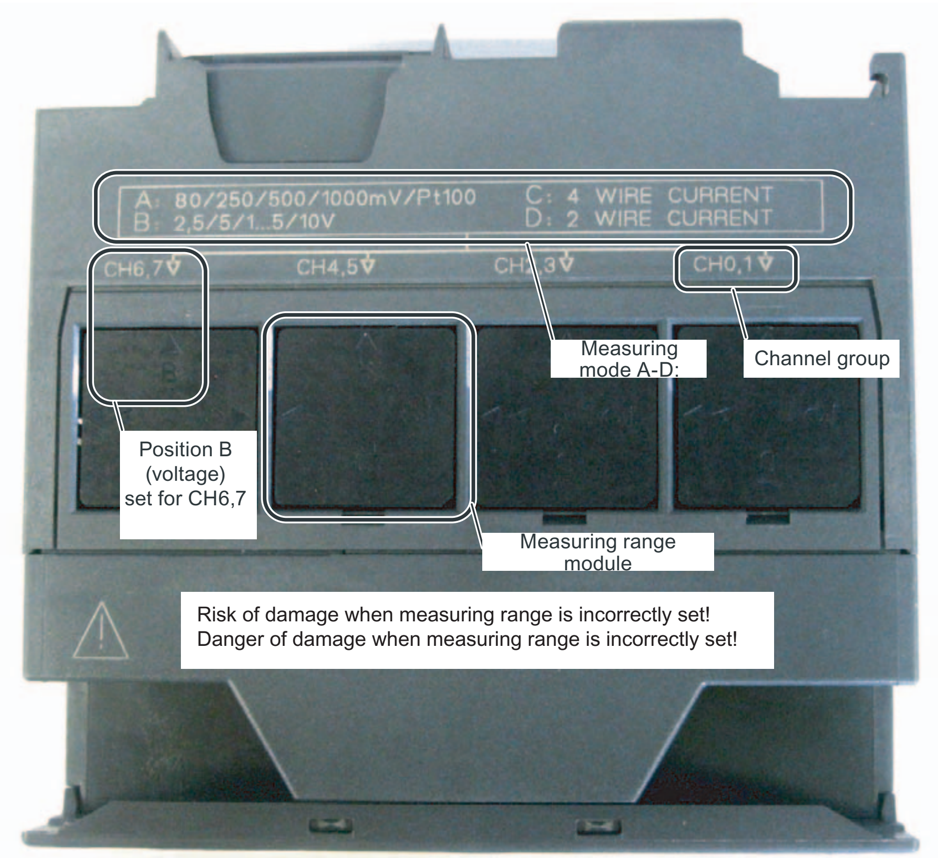


Figure 6-1 Starting the "New Project..." wizard

A project wizard introduction window will appear. The wizard guides you through the procedure for creating a project.

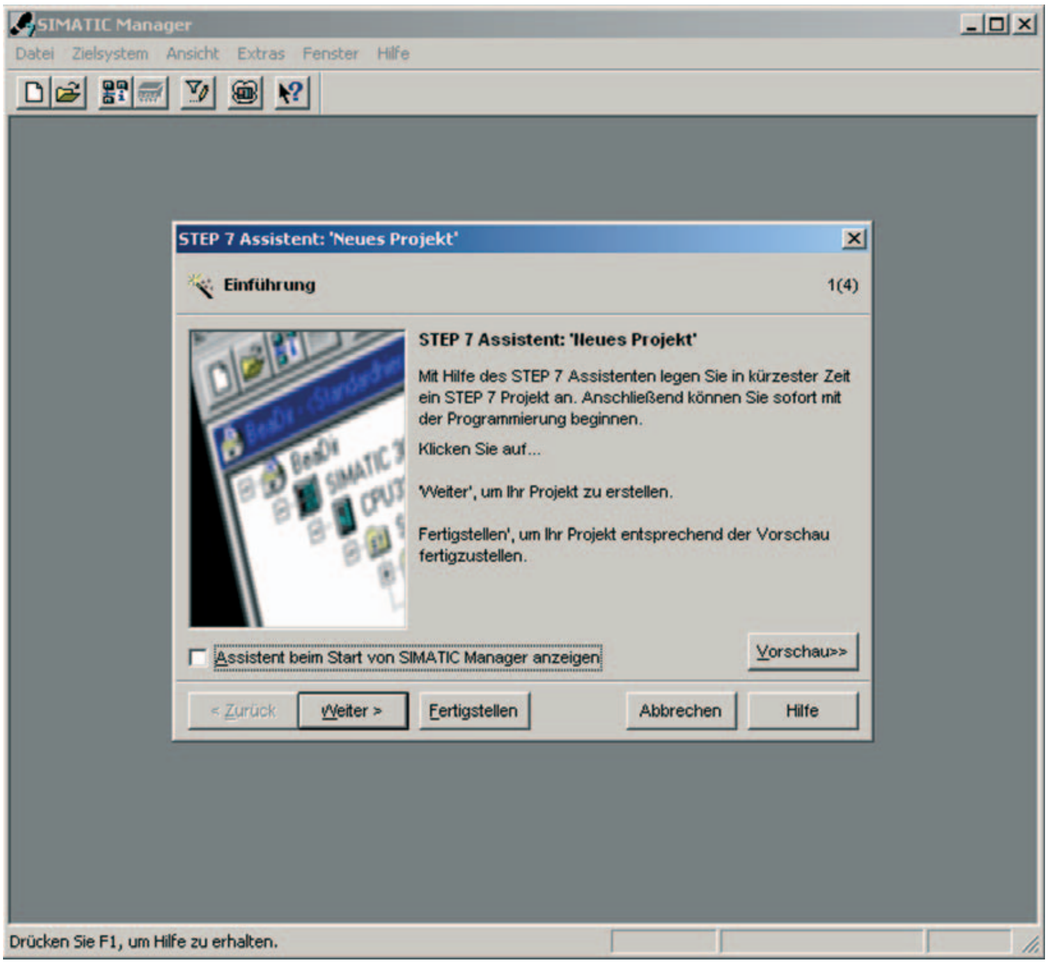


Figure 6-2 "New Project" wizard, start

The following must be specified during the creation procedure:

- The CPU type
- The basic user program
- The organization blocks
- Project name

Click "Next."

6.1.2 CPU selection

Procedure

Choose CPU 315-2DP for the example project. (You can also use our example for a different CPU. Then select the corresponding CPU.)

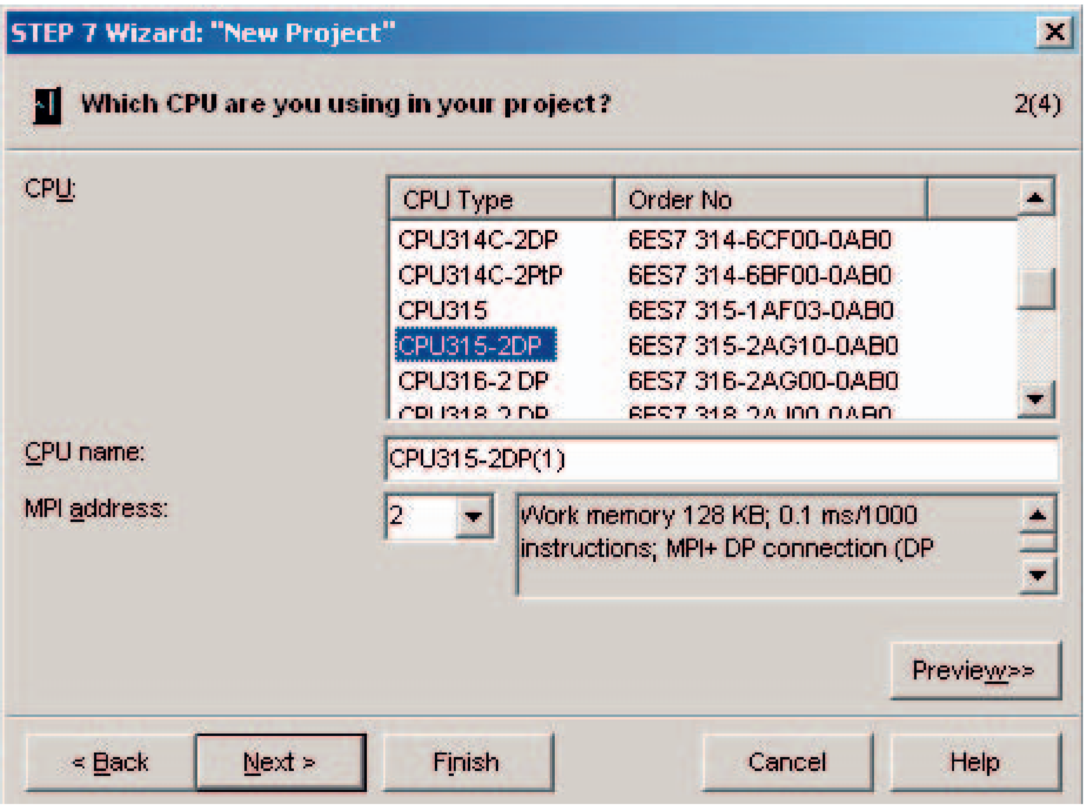


Figure 6-3 "New Project" wizard: selecting a CPU

Click "Next".

6.1.3 Defining the basic user program

Procedure

Choose the configuring language STL and select the following organization blocks (OBs):

- OB1 cyclically executed block
- OB40 hardware interrupt
- OB82 diagnostic interrupt

OB1 is required in every project and is called up cyclically.

OB40 is called up when a hardware interrupt occurs.

OB 82 is called up when a diagnostic interrupt occurs.

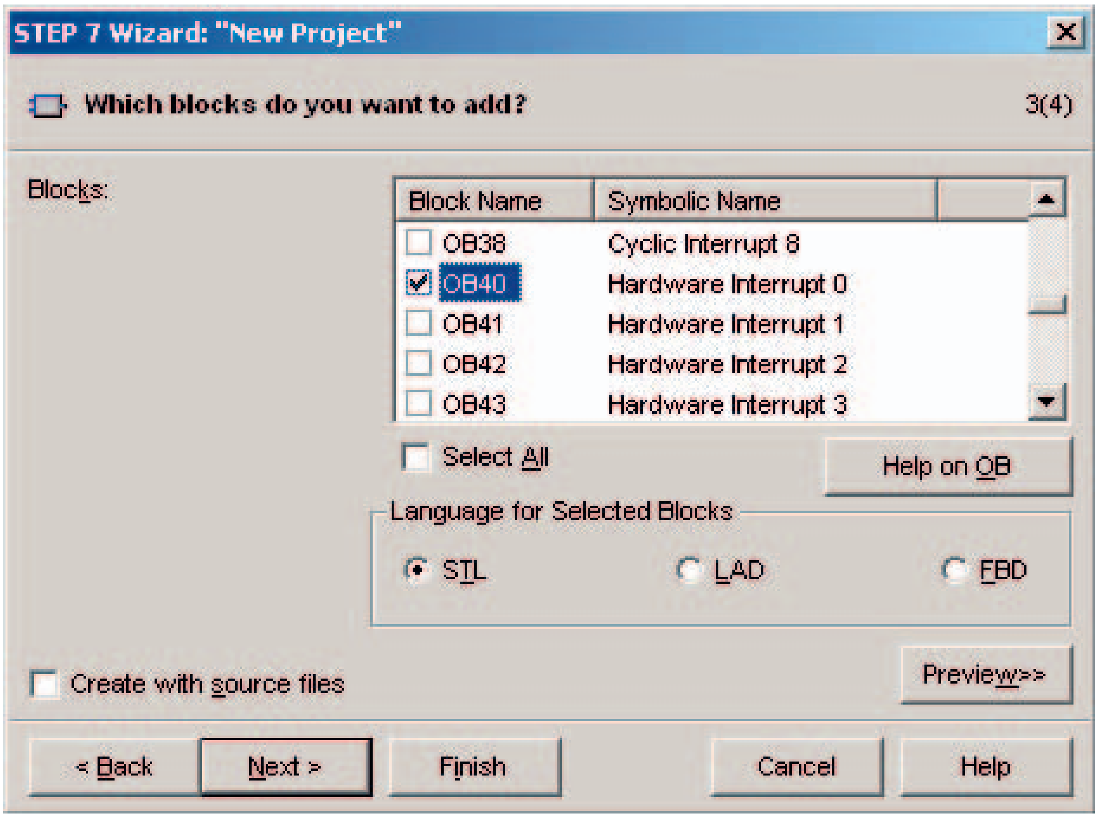


Figure 6-4 "New Project" wizard: Inserting organization blocks

Click "Next."

6.1.4 Assigning the project name

Procedure

Select the “Project name” text box and overwrite the existing name in it with "Getting Started S7 SM331".

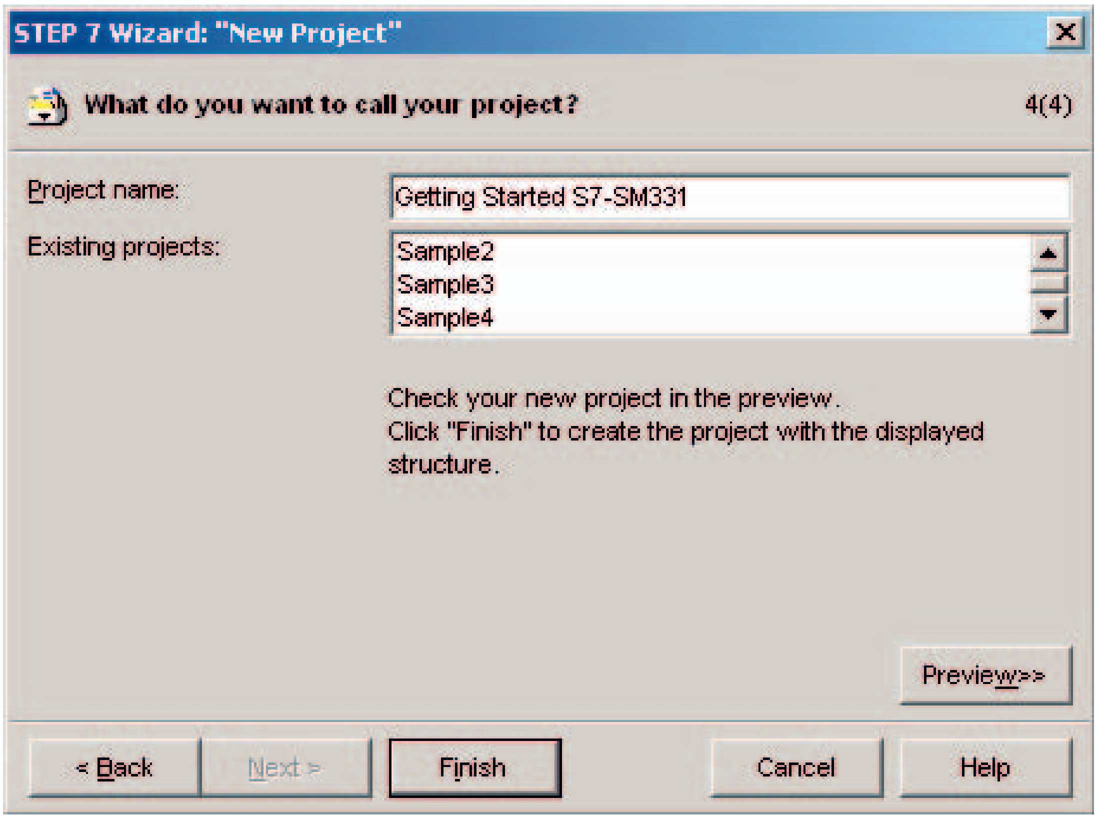


Figure 6-5 "New Project" wizard: Naming the project name

Click "Finish". The basic STEP7 project is created automatically.

6.1.5 Result S7 project is created

Result

The wizard has created the "Getting Started S7-SM331" project. You can see the inserted organization blocks in the right window.

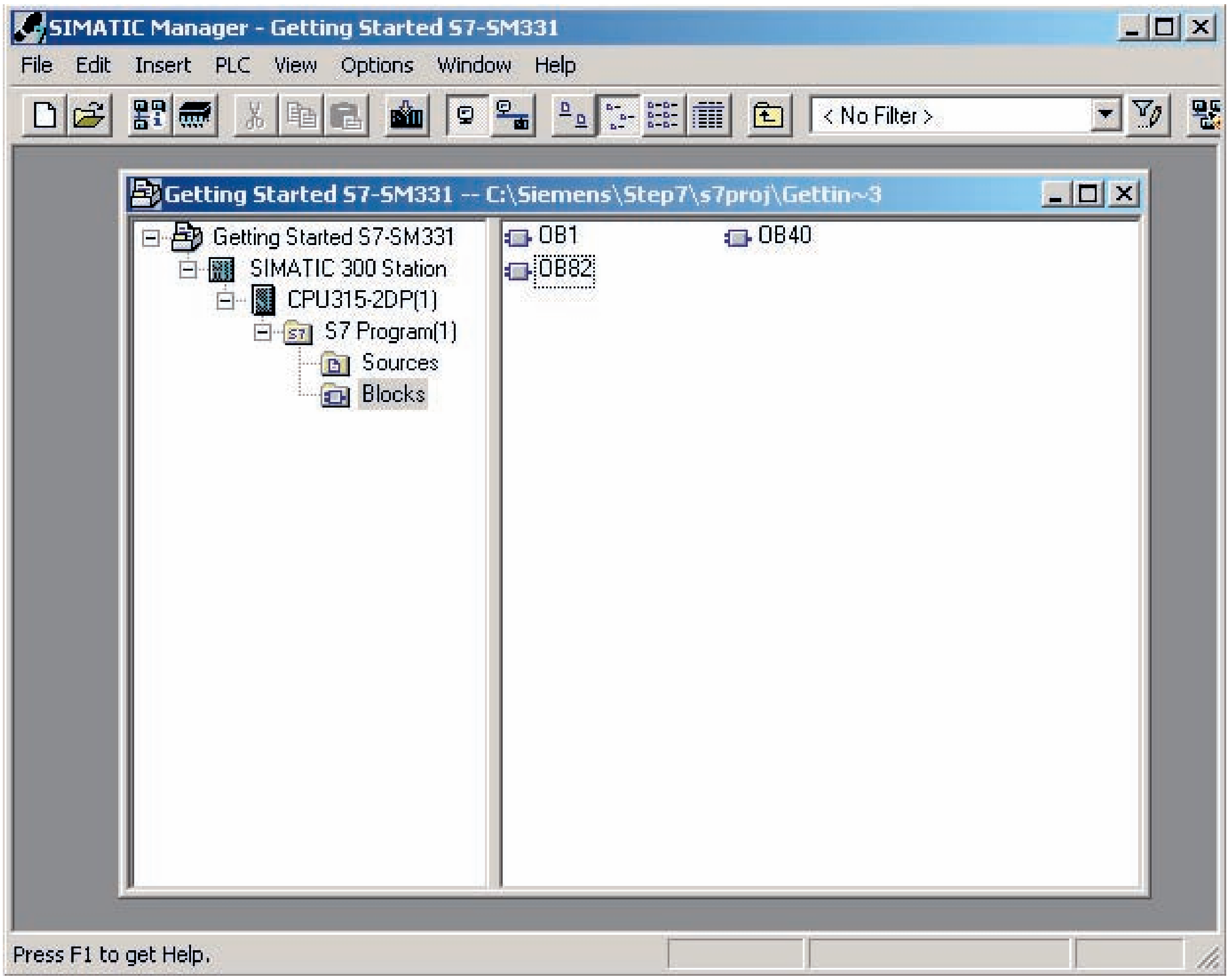


Figure 6-6 "New Project" wizard results

6.2 Configuring the hardware configuration

6.2.1 Creating the hardware configuration

Requirement

The STEP 7 wizard has created a basic S7 project. You also need complete hardware configuration in order to create the system data for the CPU.

Procedure

You can create the hardware configuration of the example station with SIMATIC Manager. .

To do this, select the "SIMATIC 300 Station" folder in the left window. Start the hardware configuration by double clicking the "Hardware" folder in the right window. .

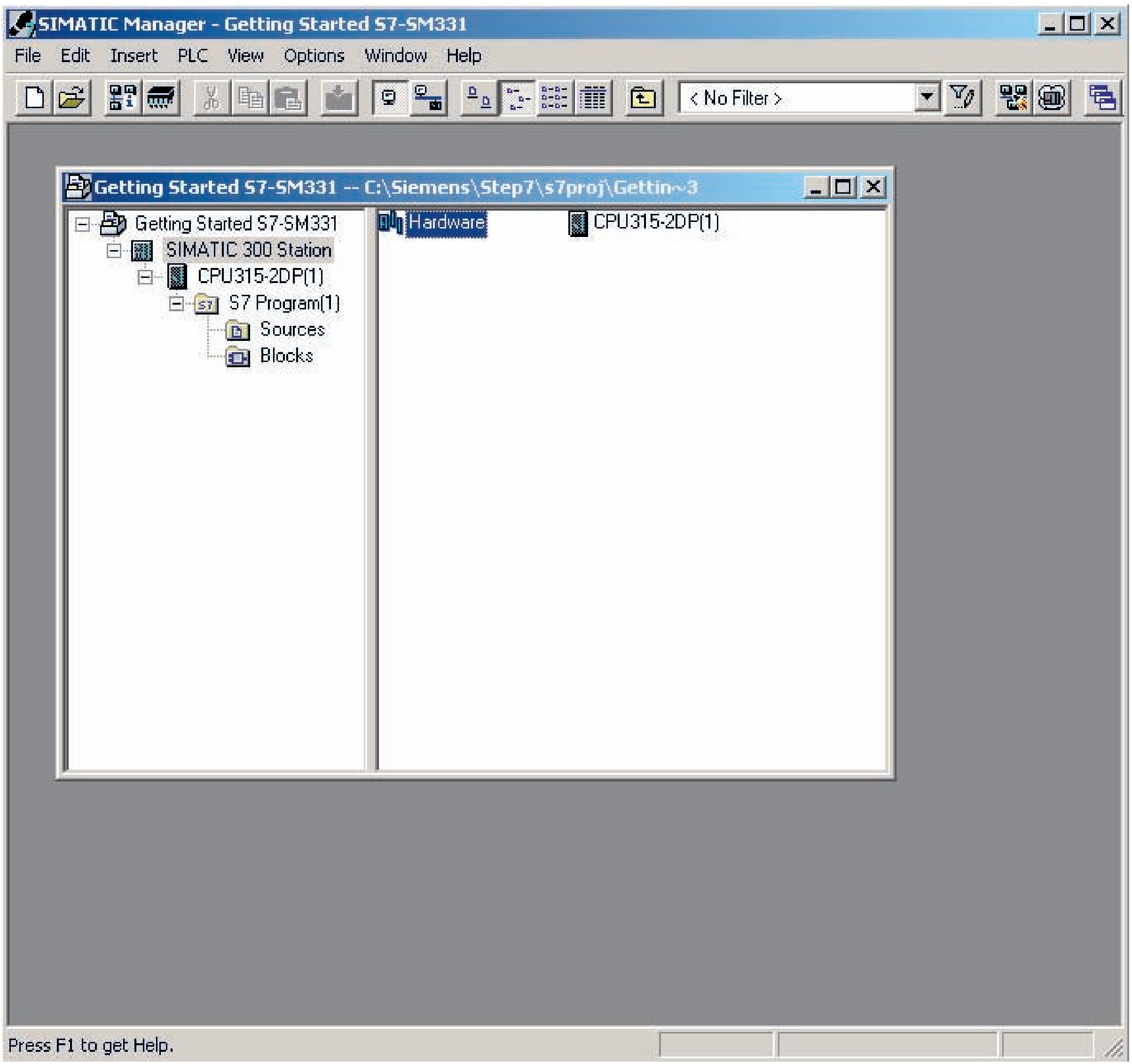


Figure 6-7 Calling up the hardware configuration

6.2.2 Adding SIMATIC components

Procedure

First select a power supply module from the hardware catalog.

If the hardware catalog is not visible, open it with the shortcut key combination Ctrl+K or by clicking the catalog icon (blue arrow).

In the hardware catalog you can browse through the SIMATIC 300 folder to the PS-300 folder.

Select PS307 5A and drag it into slot 1 (see red arrow).

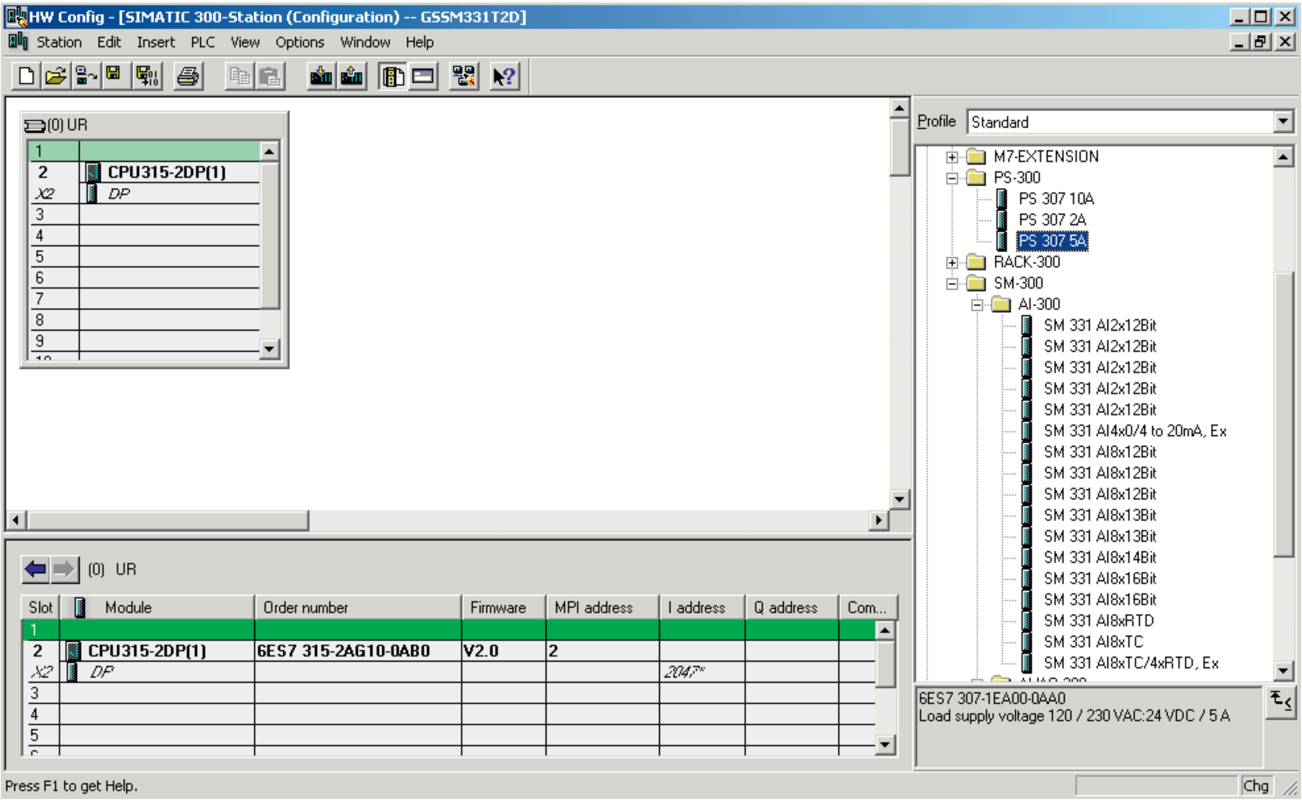


Figure 6-8 Hardware configuration: Basic configuration

Result: PS 307 5A appears in the configuration of your rack.

Inserting an analog module

There are many SM331 analog modules. For this project we use an SM331, AI8x12 bit with order number 6ES7 331-7KF02-0AB0.

The order number is displayed at the bottom of the hardware catalog (see blue arrow).

In the right window, click on the SM331 AI8x12Bit and drag it to the first free field in slot 4 (see red arrow) in the configuration table.

Now you have inserted all the modules into the hardware configuration. In the next step, you configure the module.

Result: You can now configure the SM331.

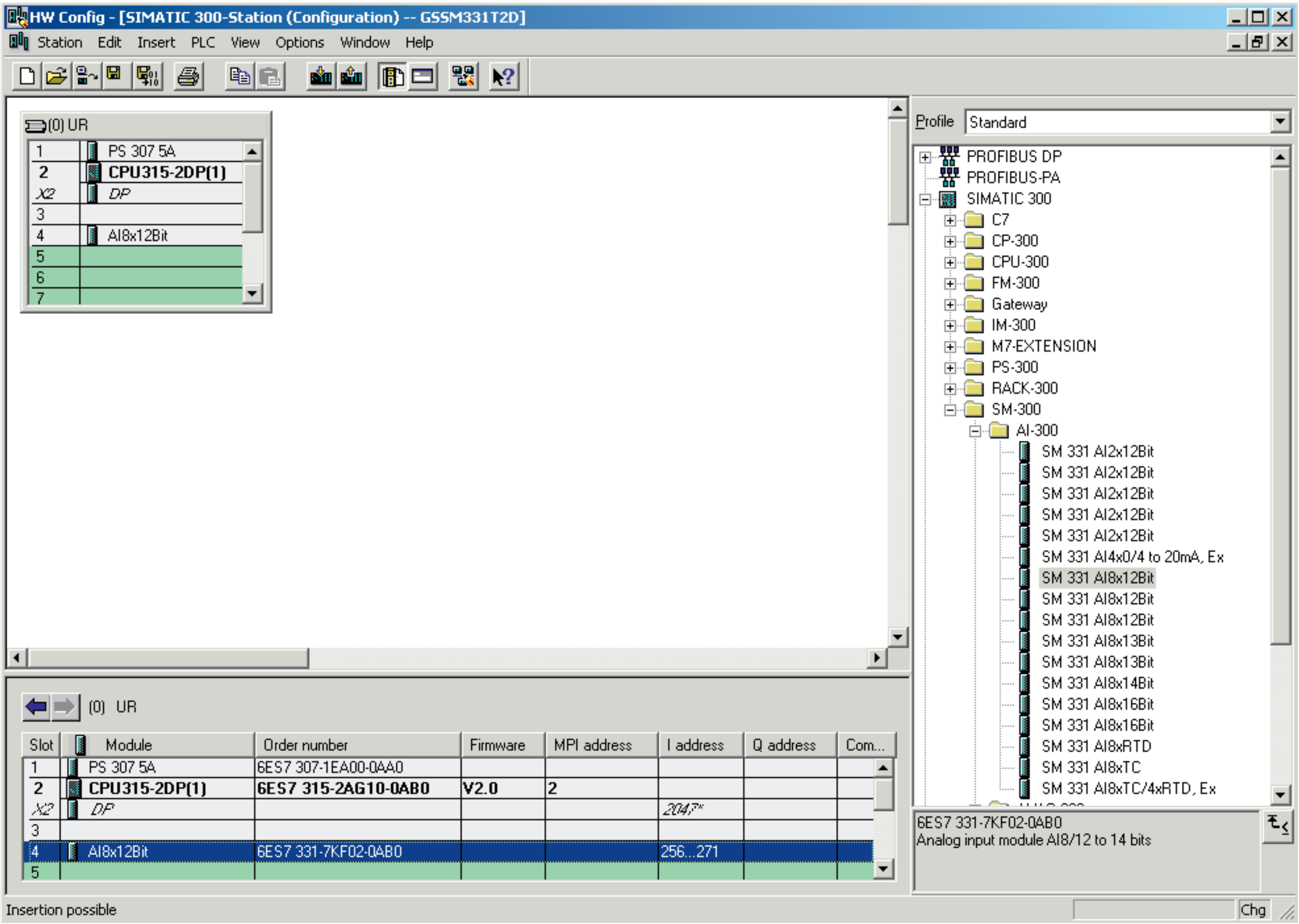


Figure 6-9 Hardware configuration: Inserting SM331

6.2.3 Configuring the analog module

Overview

The SIMATIC Manager inserts the analog module with the default settings. You can now modify the parameters to change the sensor types, diagnostics, and interrupt capabilities.

Functionalities of the example station

The table shows which parameters have to be set for our example station.

SM331 functions of the example station

Functionalities	Description
Process reactions	• Diagnostics – enabled • Hardware interrupt when limit value exceeded - enabled
Sensor 1	• Voltage measuring transducer • Group diagnostics • Measuring range ±5V • Limit values -3 Volts and +3 Volts
Sensor 2	• PT100 resistance thermometers • Group diagnostics • Check for wire break • Limit values -20 °C and +50 °C
Sensor 3	• PT100 resistance thermometers • Group diagnostics • Check for wire break
Sensor 4	• PT100 resistance thermometers • Group diagnostics • Check for wire break

Calling up the configuration

Double-click on slot 4 that has the SM331 in it.
Select the "Inputs" tab.
Configure the following functions:

- Diagnostic interrupt enabled
- Hardware interrupt enabled
- Input 0-1:
  - Type of measurement: U
  - Measuring range ±5
  - Group diagnostics enabled
- Input 2-3, 4-5, and 6-7
  - Type of measurement: RT
  - Measuring range PT100 standard
  - Group diagnostics enabled
  - Wire break enabled
- Interference frequency
  - Select your power frequency (50 Hz or 60 Hz)
- Trigger for hardware interrupt channel 0
  - Upper limit value +3 V
  - Lower limit value -3V
- Trigger for hardware interrupt channel 2
  - Upper limit value +50 °C
  - Lower limit value -20 °C

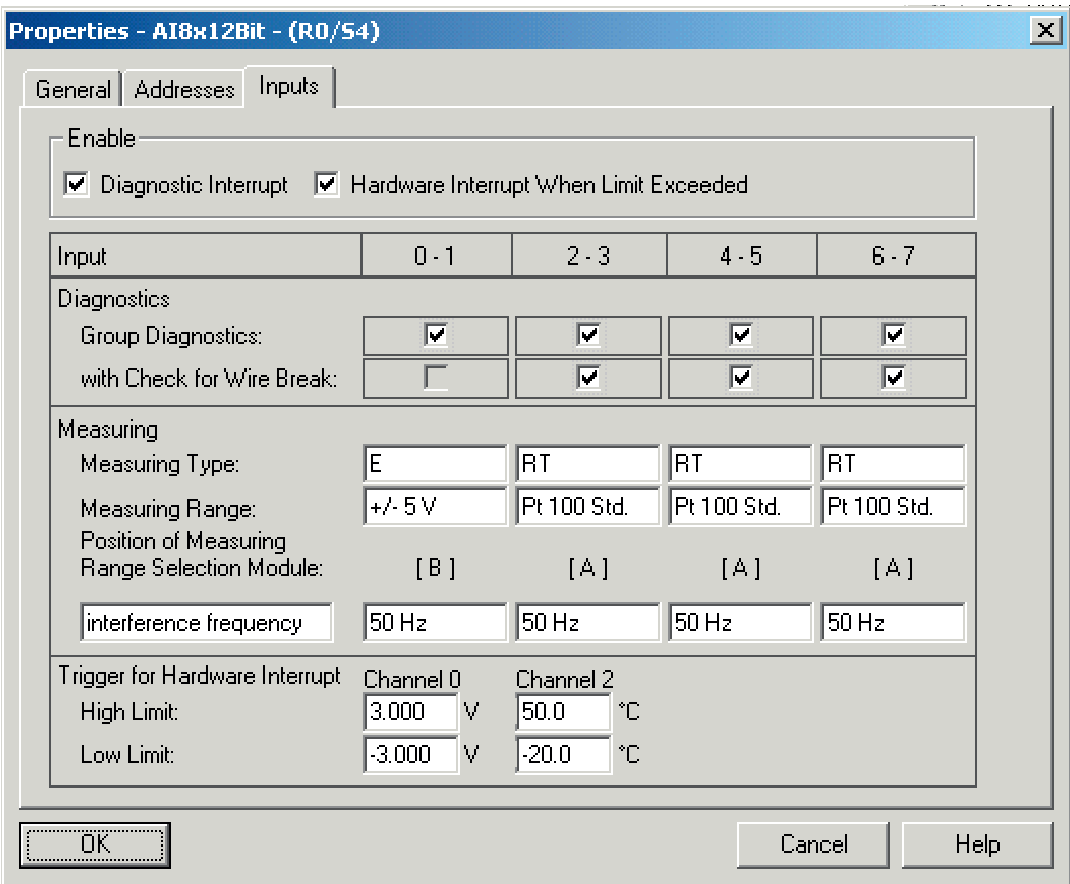


Figure 6-10 SM331: Configuration

Explanation of the settings of the SM331

Diagnostic interrupt:

If the diagnostic interrupt is enabled, diagnostic OB86 is called up when the ground or the power supply is missing.

Hardware interrupt:

If the parameter "Hardware interrupt when limit value exceeded" is enabled, hardware interrupt OB40 is called up when the set limit value is exceeded or undershot.

You can configure the limit values in the same window under "hardware interrupt triggers".

Group diagnostics:

If group diagnostics is selected, channel-linked diagnostic messages are enabled. If a diagnostic event occurs, OB86 is called up.

Wire break check:

If wire break check is enabled, a wire breakage is diagnosed. Diagnostic OB86 is called up.

Type of measurement:

U stands for voltage.

RT stands for resistance (resistor, temperature).

Measuring range:

Specifying the measuring range of the voltage sensor and type PT100.

Position of the measuring range module:

The required setting of the measuring range module is displayed.

Interference frequency (Interference frequency suppression):

You set the interference frequency on your existing network frequency.

Hardware interrupt trigger:

If the parameter "hardware interrupt when limit is exceeded" is enabled, you can enter your required limit values. If a fixed value is exceeded or undershot, hardware interrupt OB40 is called up.

Only channels (inputs) 0 and 2 have hardware interrupt capabilities.

Finishing the hardware configuration:

Close the window with the configuration.

Compile and save the project with the command "Station > Save and Compile" (Ctrl+S).

This completes your hardware configuration for the project.

See also

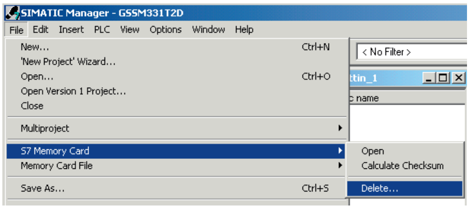
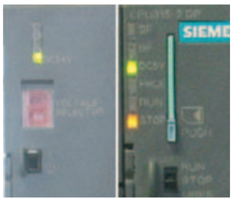
Types of diagnostic messages (Page 8-4)

Measuring range modules (Page 4-7)

6.2.4 Power-up test

Procedure

For testing, do a power-up test and download the system data.

Step	Graphic	Description
1		Erase your Micro Memory Card with a Power PG or a PC with external programming device: In SIMATIC Manager, select "File -> S7 Memory Card > Delete …". The MMC will be deleted.
2		Switch off the power supply to the CPU Insert the MMC into the CPU. Switch on the power supply.
3		If the CPU is in RUN mode, set it to STOP mode.
4		Switch the power supply back on. If the STOP LED flashes, the CPU is requesting a reset. . Acknowledge this by turning the mode switch to MRES for a moment.
5		Connect the CPU to the programming device using an MPI cable. . To do this, connect the MPI cable with the CPU's MPI port. Connect the other end to the programming device interface of your programming device.

Downloading hardware configuration

Load the hardware configuration into the CPU with HW Config.

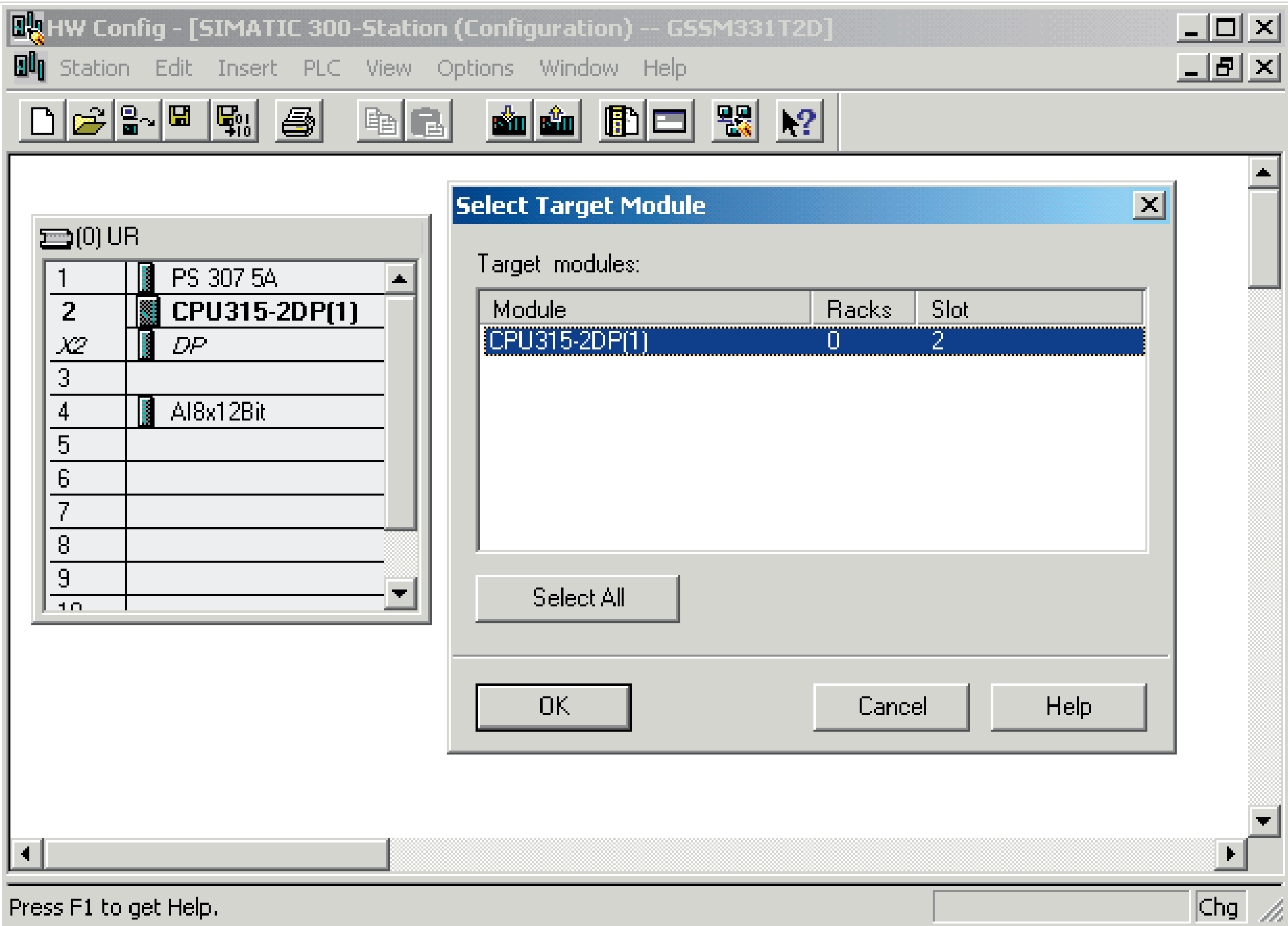


Figure 6-11 Download the CPU hardware configuration (1)

Click the "Load to module" icon (see the red circle).

When the dialog window "Select target module" appears, click OK (see the red arrow).

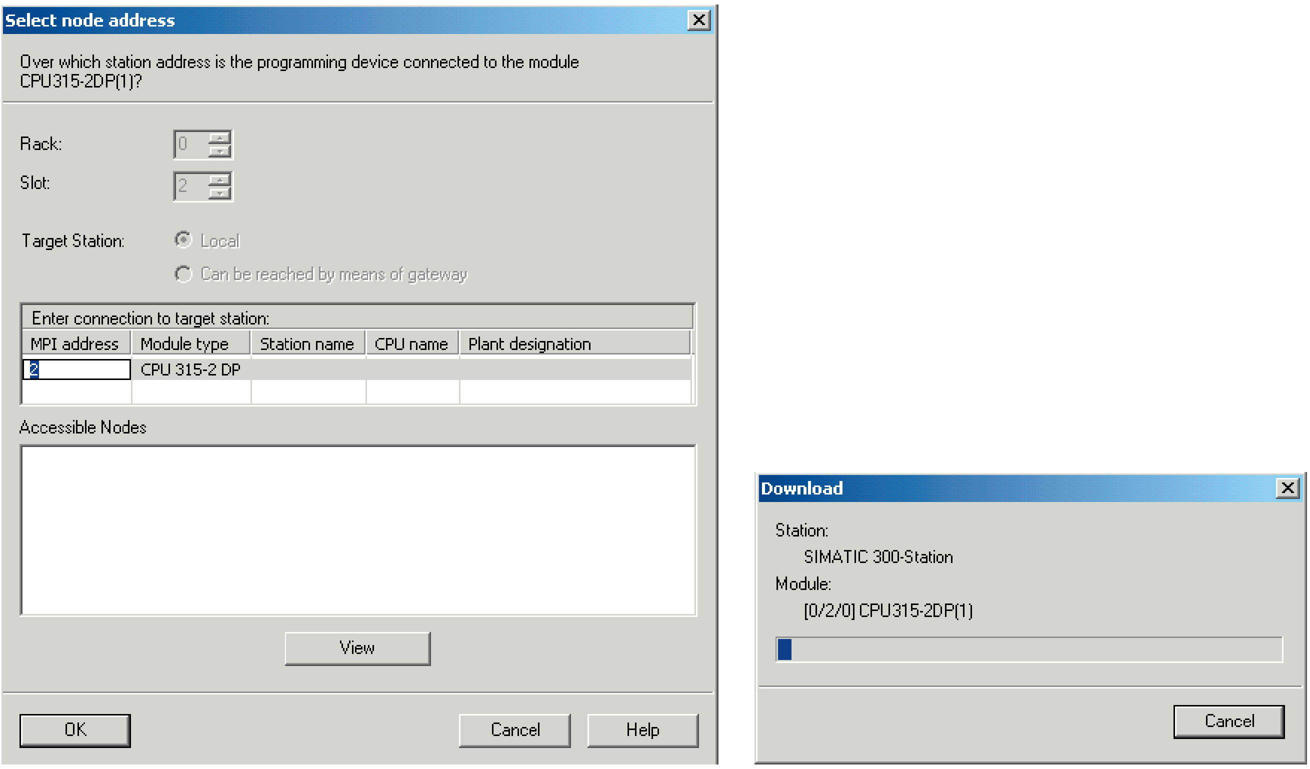


Figure 6-12 Download the CPU hardware configuration (2)

The dialog window "Select target address" will be shown. Click "OK." The system data will now be transferred to the CPU.

Starting the CPU

Switch the CPU to RUN.

If you have performed the hardware configuration correctly, two green LEDs (RUN and DC5V) should be lit on the CPU.

Figure 6-13 CPU in error free state

If the RUN LED does not light up, there is an error.

Read out the diagnostic buffer using the programming device to localize the error. Possible causes of error:

- The wiring was not done correctly
- The coding cube is improperly plugged in
- You have incorrectly input the parameters of the SM331.

6.3 STEP 7 user program

6.3.1 Tasks of the user program

Overview

The example user program

- stores sensor values in a data block and
- saves the status information regarding the hardware interrupts in a marker word.

The status information is acknowledged by means of a bit. Furthermore, the channel values (values of the input words) are saved in another data block.

The following tasks have to be performed in the user program:

1. Cyclical storage of the analog inputs in a data block (DB1)
2. Cyclical conversion of the sensor values in floating point values (FC1) and storage in a data block (DB2)
3. Acknowledgement of the hardware interrupt status when the acknowledge marker (M200.0) is TRUE.
4. Save the status in a marker word (MW100) when a hardware interrupt occurs

Structure of the user program

Call-up type	Responsible organization block	Task to be programmed	Used block or marker
Cyclic execution	OB1	Save analog inputs	DB1
		Convert and store the sensor values	FC1, DB2
		Acknowledge hardware interrupt	M200.0
Hardware interrupt triggered call-up	OB40	Save status	MW100
Diagnostic interrupt triggered call-up	OB82	Has to be implemented because a module with diagnostic capabilities is used	---

OB82 diagnostic interrupt

In the STEP7 program, the OB82 is used for modules with diagnostic capabilities.

If the module detects an error (both for incoming and outgoing events), the module makes a diagnostic interrupt request to the CPU. The operating system then calls up OB 82.

In our example, we use OB82 merely to prevent the CPU from changing to STOP mode. You can program reactions to diagnostic interrupts in OB82.

6.3.2 Creating a user program

Procedure

There are two ways to create a user program.

- If you know how to program STEP7 STL, then you can create and program the necessary blocks and functions in the blocks folder.
- You can insert the user program from an STL source into your project. We describe this method in the "Getting started" manual.

Creating a user program in STEP7 requires three steps:

1. Downloading the source file directly from the HTML page
2. Importing a source file
3. Compiling the source

Downloading the source file

You can download the source file directly from the HTML page from which you loaded this "Getting Started" manual. Click on "Info" and the download window will open.

- Make note of the name of the source file.
- Save the source file to your hard drive.

Importing a source file

You can import the source file into SIMATIC Manager as follows:

- Right click the "Sources" folder.
- Select "Insert new Object > External Source...".

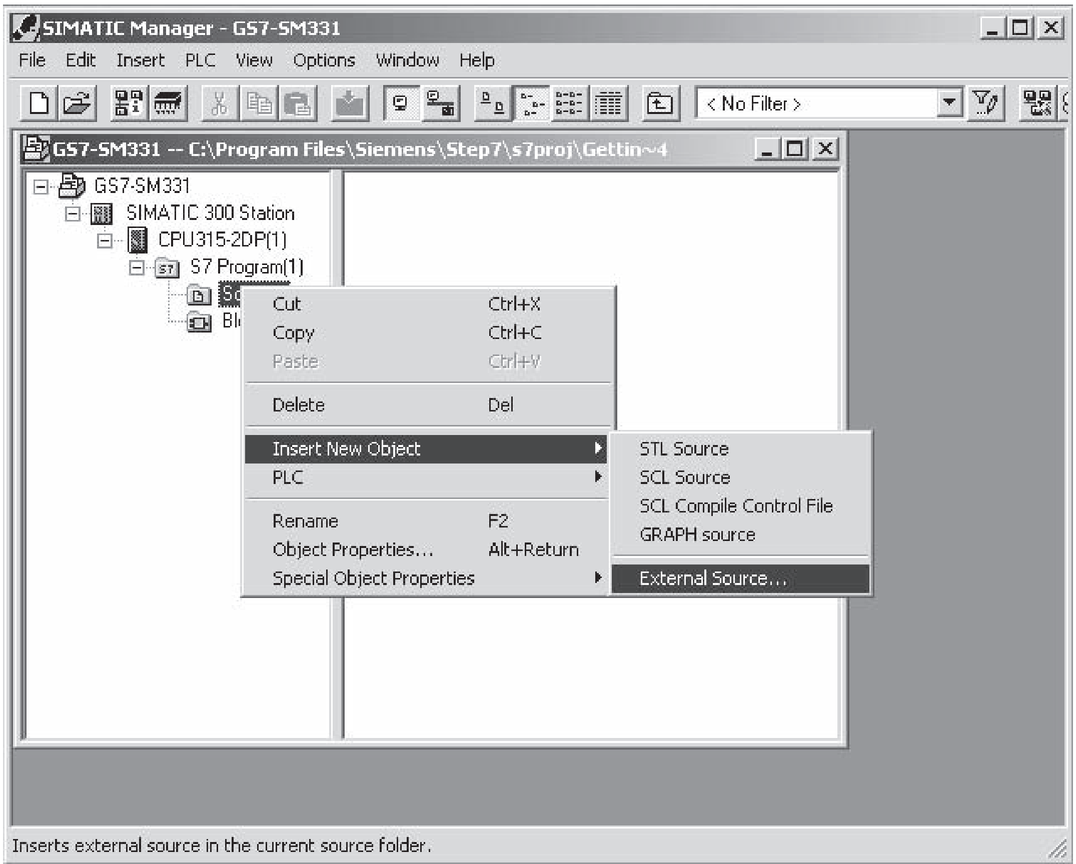


Figure 6-14 Importing an external source

In the "Insert external source" dialog, browse for the source file, which you have already downloaded and saved on your hard disk.

Select source file GSSM331T2DE.AWL (see red arrow).

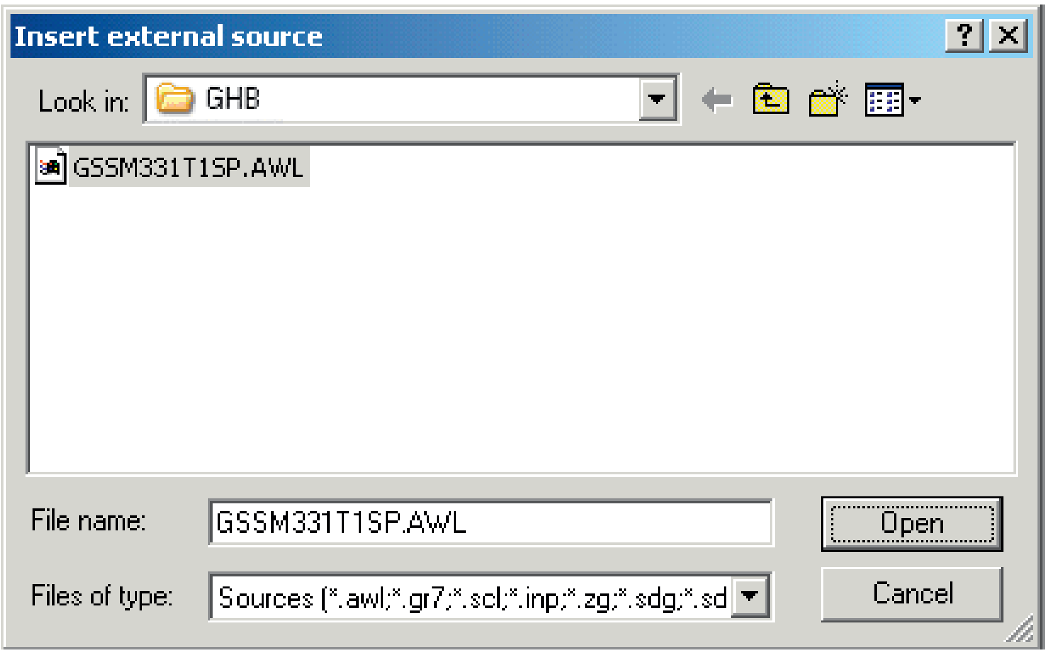


Figure 6-15 Importing an external source

Click "Open".

SIMATIC Manager has read the source file. In the right window you can see the source file inserted.

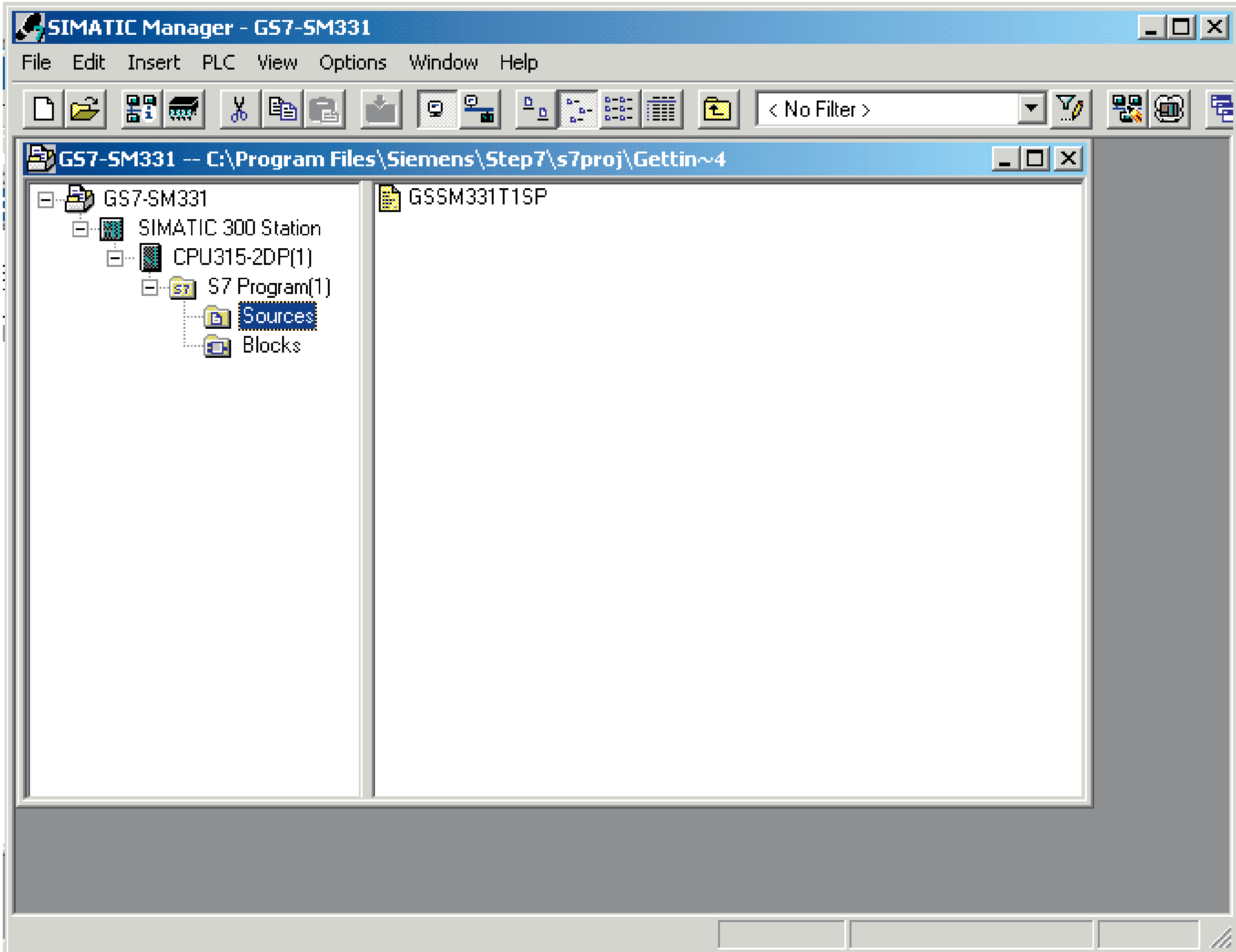


Figure 6-16 Storing the source file

Compiling the source code

In order to create an executable STEP7 program, you must compile the STL source file.

Double-click on the source file in the source folder (see red arrow). The source code editor will be called up.

In the window of the source code editor you can view the source code.

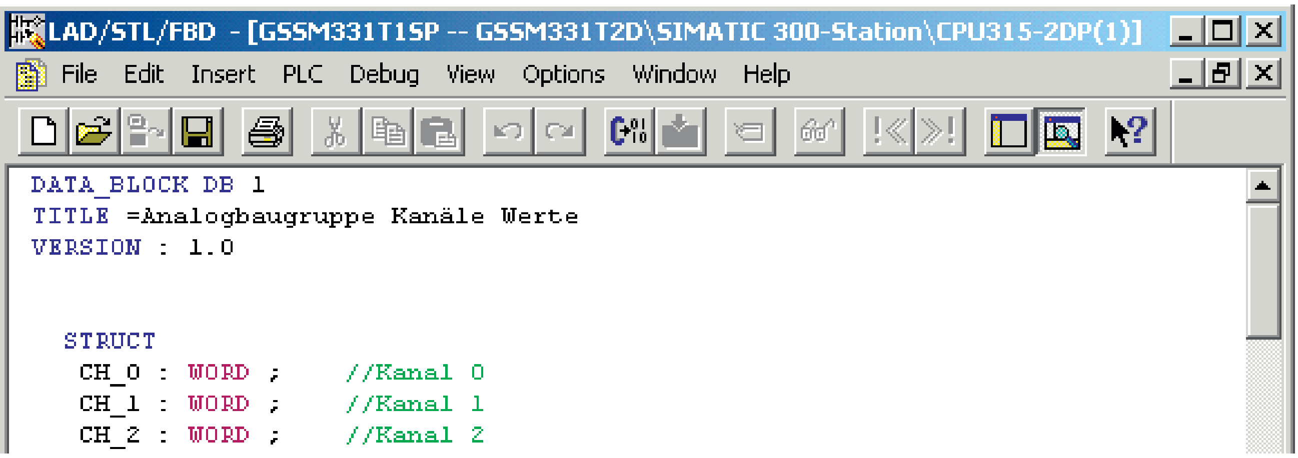


Figure 6-17 Source code editor

After the source code is loaded, start the compilation.

Press shortcut key combination Ctrl+K or select "File > Compile". The compilation will start immediately.

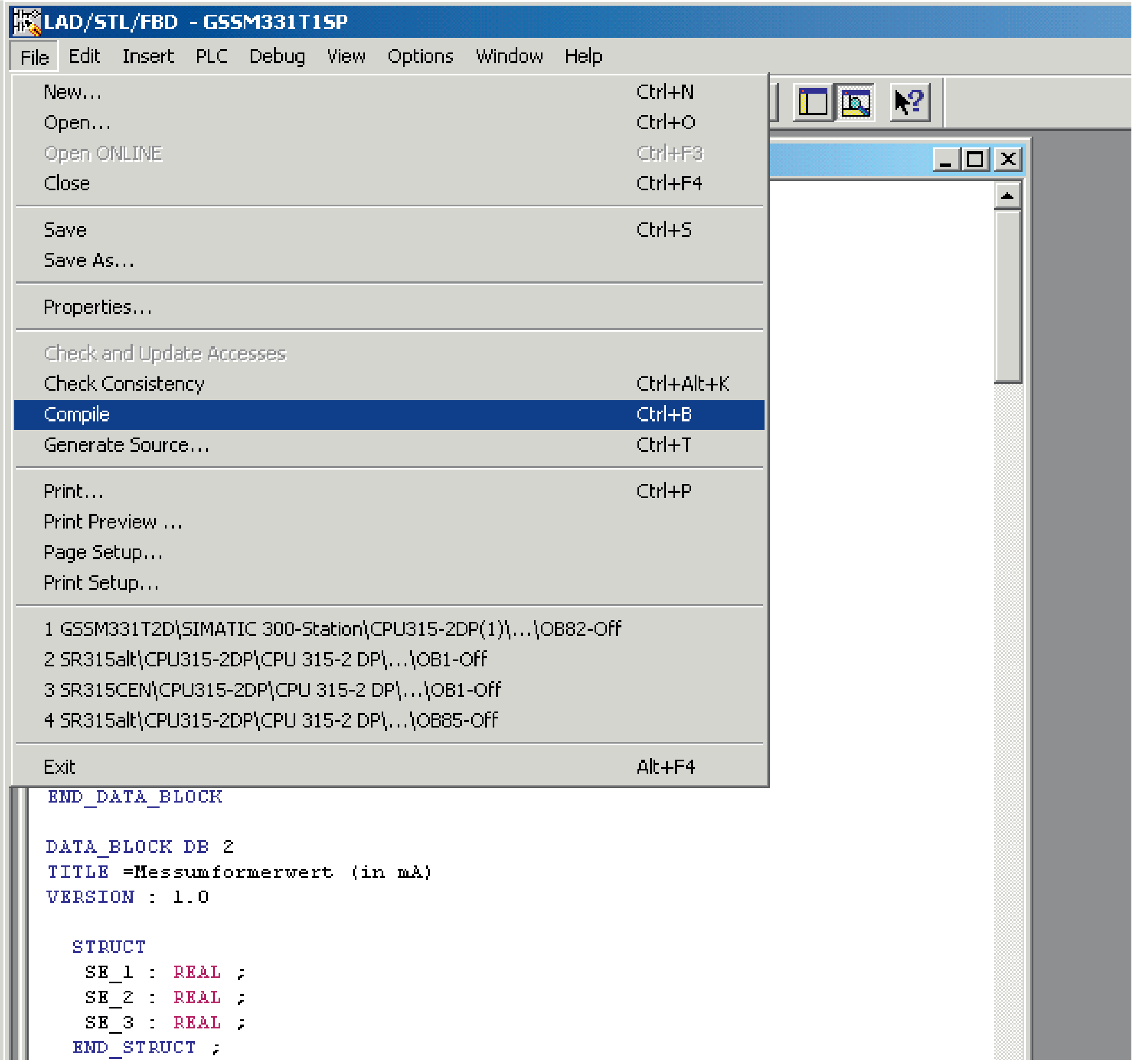


Figure 6-18 Compiling STL source files

In the event of warning or error messages, check the source.

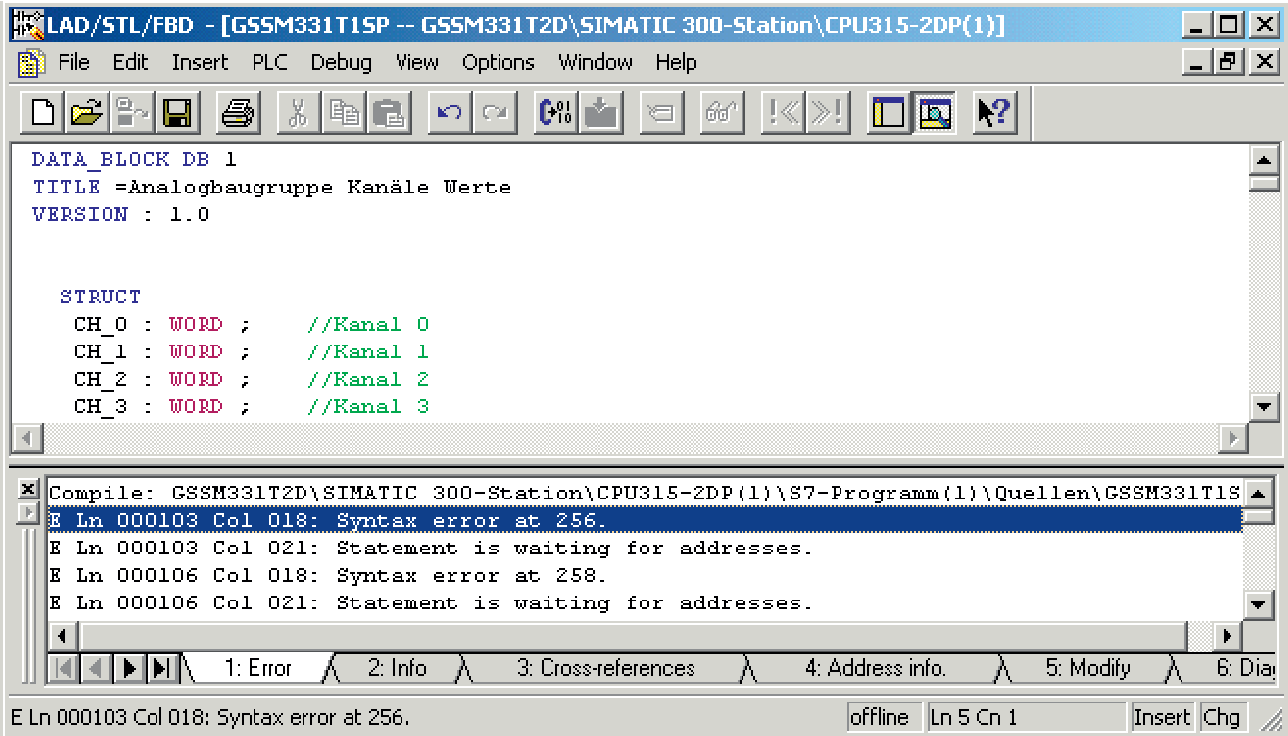


Figure 6-19 Source code editor, messages after compilation

Close the source editor.

After compiling the STL source without errors, the following blocks should appear in the block folder:

OB1, OB40, OB82, FC1, DB1, and DB2

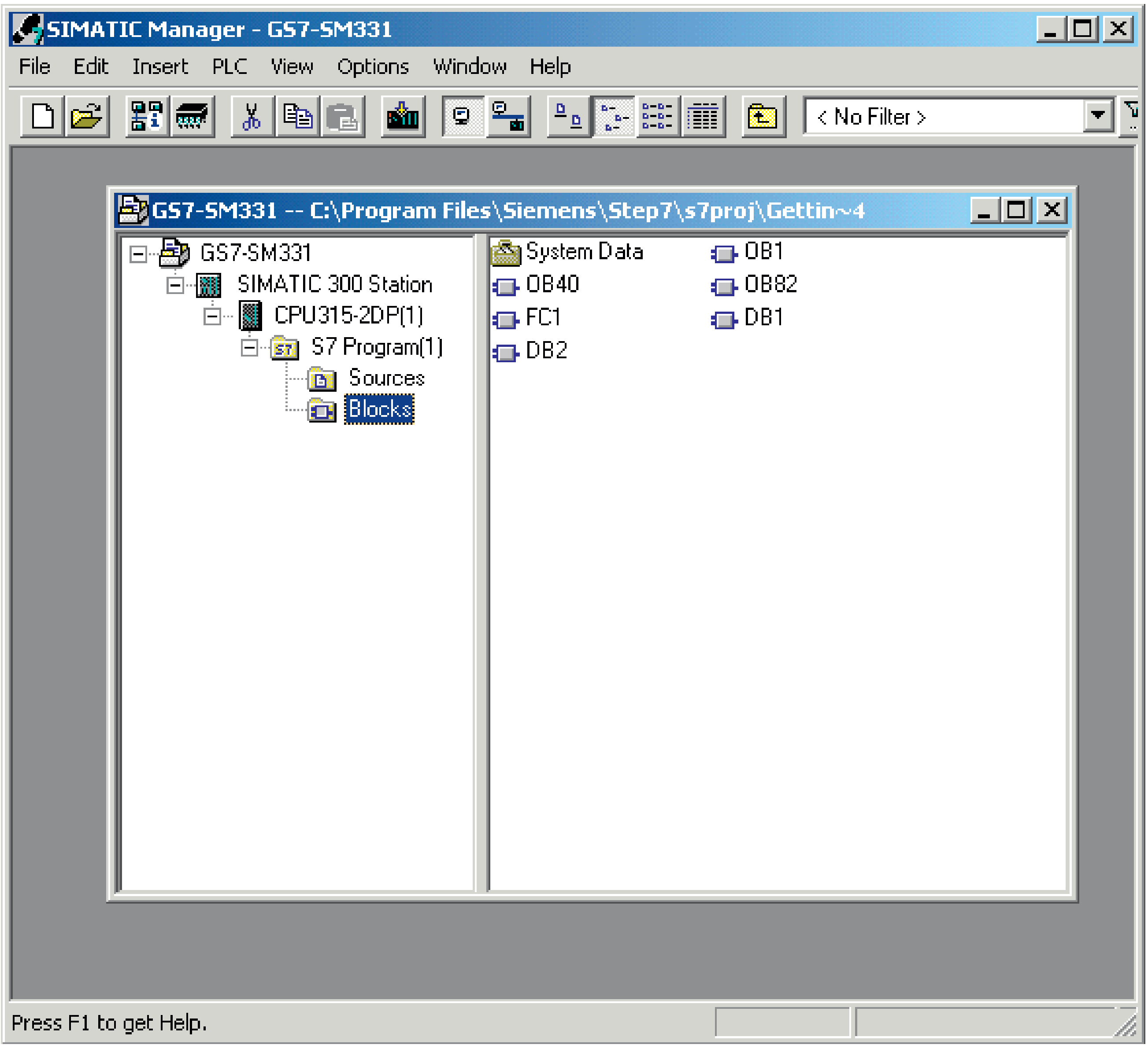


Figure 6-20 Generated blocks

See also

Source code of the user program (Page A-1)

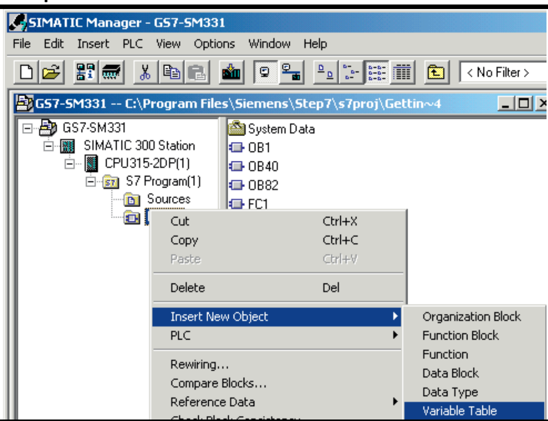
Testing the user program 7

7.1 Downloading the system data and user program

Procedure

The hardware and software are now ready. The next step is to download the system data and the user program into the automation system. To do this, proceed as follows:

Downloading the system data and user program

Step	Graphic	Description
1		Using the SIMATIC Manager, download the user program and the system data (containing the hardware configuration) into the CPU.
2		Follow the instructions displayed on the screen. If all sensors are properly connected, the CPU and the SM331 do not display a red error light. The error-free operation of the CPU is displayed by the green "RUN" light.

Smart Label

The labeling strips for the modules were created with Siemens S7 Smart Label (order no: 2XV9 450-1SL01-0YX0).

A labeling strip in its actual size:

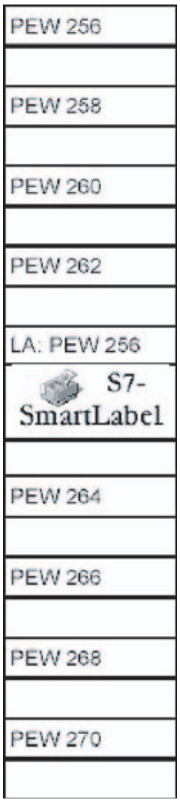


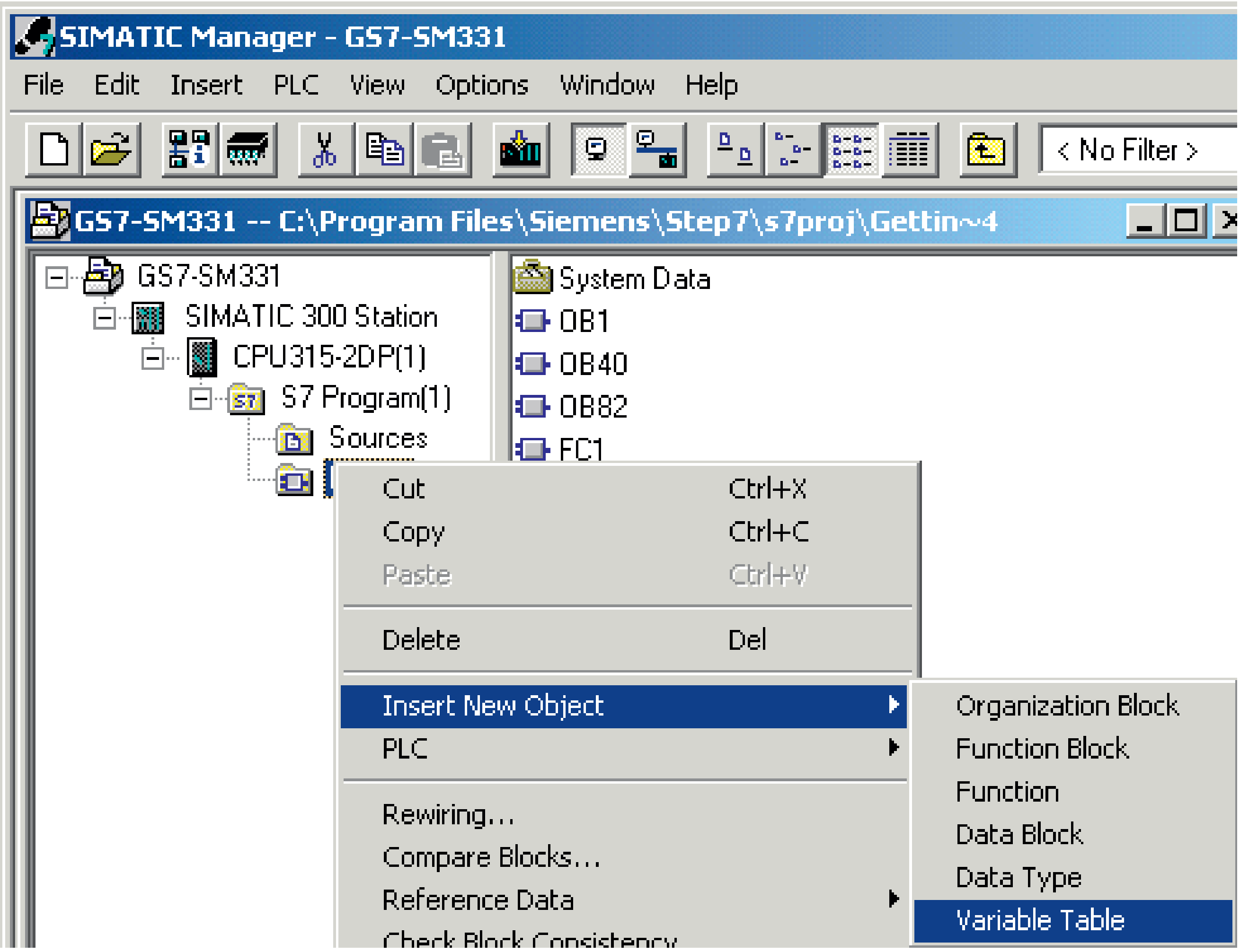
Figure 7-1 S7-SmartLabel labeling strip for the example

7.2 Visualizing the sensor values

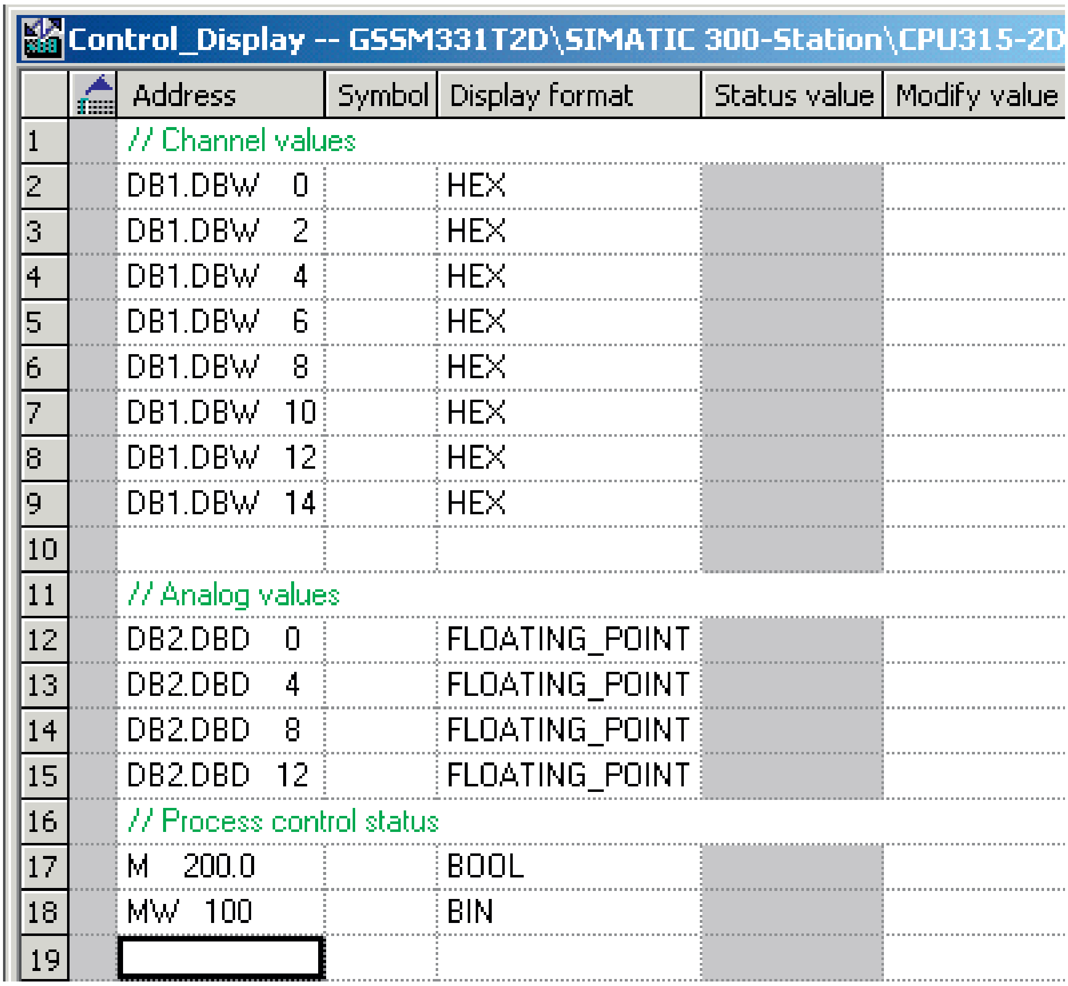
Procedure

In order to visualize the sensor values, insert a variable table as follows into the project. To do this, select from the context menu of the blocks folder:

Insert new object > Variable Table



Complete the newly created variable table as follows: :


Control_Display -- GSSM331T2D\SIMATIC 300-Station\CPU315-2D

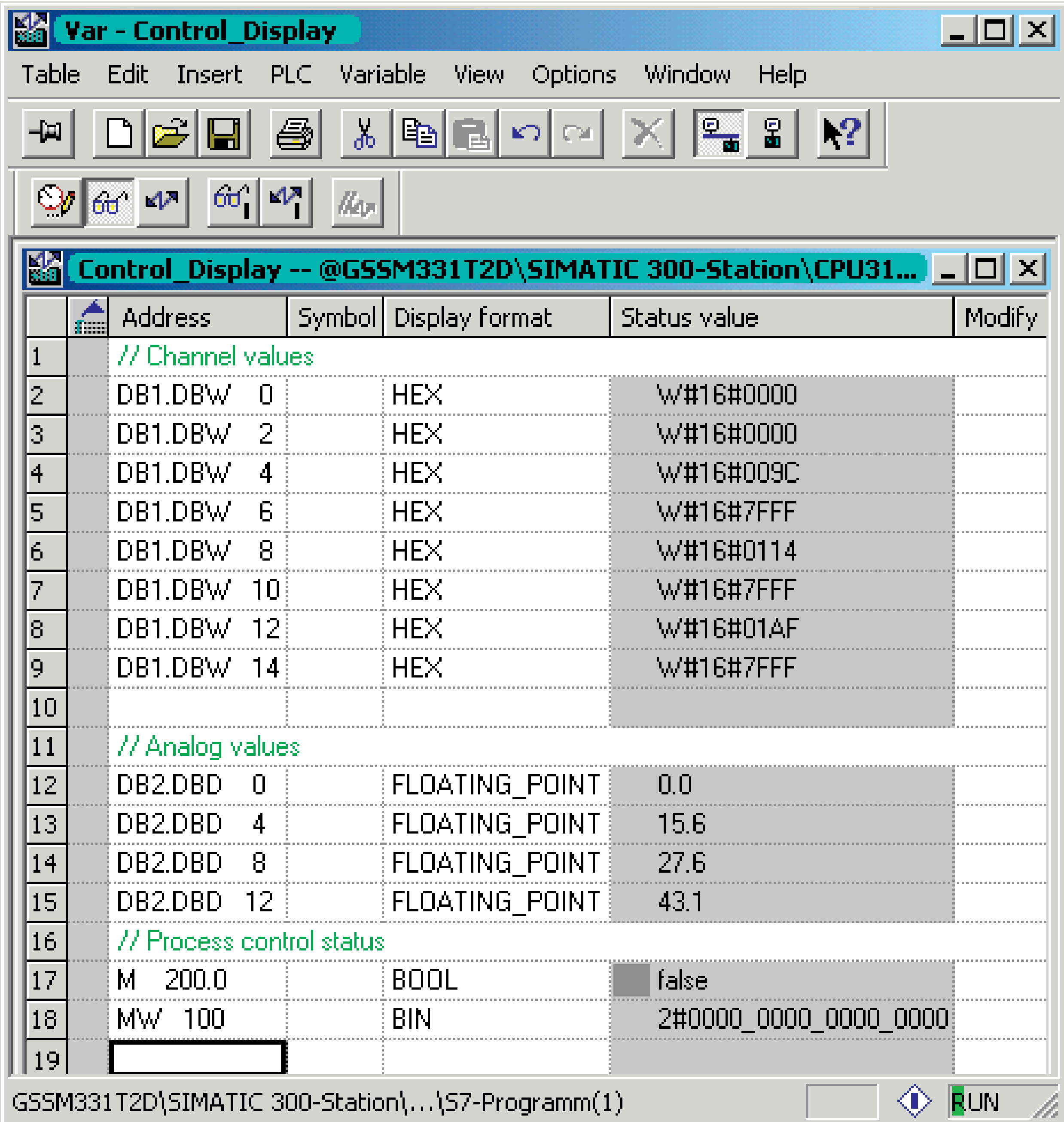
	Address	Symbol	Display format	Status value	Modify value
1	// Channel values				
2	DB1.DBW 0		HEX		
3	DB1.DBW 2		HEX		
4	DB1.DBW 4		HEX		
5	DB1.DBW 6		HEX		
6	DB1.DBW 8		HEX		
7	DB1.DBW 10		HEX		
8	DB1.DBW 12		HEX		
9	DB1.DBW 14		HEX		
10					
11	// Analog values				
12	DB2.DBD 0		FLOATING_POINT		
13	DB2.DBD 4		FLOATING_POINT		
14	DB2.DBD 8		FLOATING_POINT		
15	DB2.DBD 12		FLOATING_POINT		
16	// Process control status				
17	M 200.0		BOOL		
18	MW 100		BIN		
19					

Figure 7-2   Variable table Control_Display

(1) In this area you can monitor the channel values.
(2) In this area you can monitor and control the status signals.
(3) In this area you can see the analog values.

Monitoring values

In order to monitor values, open the online view of the controller by clicking the eye glasses icon. Now you can monitor the values in the data blocks and markers.



	Address	Symbol	Display format	Status value	Modify
1	// Channel values				
2	DB1.DBW 0		HEX	W#16#0000	
3	DB1.DBW 2		HEX	W#16#0000	
4	DB1.DBW 4		HEX	W#16#009C	
5	DB1.DBW 6		HEX	W#16#7FFF	
6	DB1.DBW 8		HEX	W#16#0114	
7	DB1.DBW 10		HEX	W#16#7FFF	
8	DB1.DBW 12		HEX	W#16#01AF	
9	DB1.DBW 14		HEX	W#16#7FFF	
10					
11	// Analog values				
12	DB2.DBD 0		FLOATING_POINT	0.0	
13	DB2.DBD 4		FLOATING_POINT	15.6	
14	DB2.DBD 8		FLOATING_POINT	27.6	
15	DB2.DBD 12		FLOATING_POINT	43.1	
16	// Process control status				
17	M 200.0		BOOL	false	
18	MW 100		BIN	2#0000_0000_0000_0000	
19					

Figure 7-3 Online view of the variable table

Peculiarity in monitoring the values

While monitoring the values you will surely notice that the channel values are different from the analog values. The reason for this is that the analog module only supports the "Word" binary format (16 bits). Therefore, the values of the analog module have to be converted.

Controlling values

To control the process acknowledgement, enter the desired value ("TRUE or FALSE", depending on whether you want to activate or deactivate acknowledgement) into the "Control Value" column and click the icon with the two arrows.

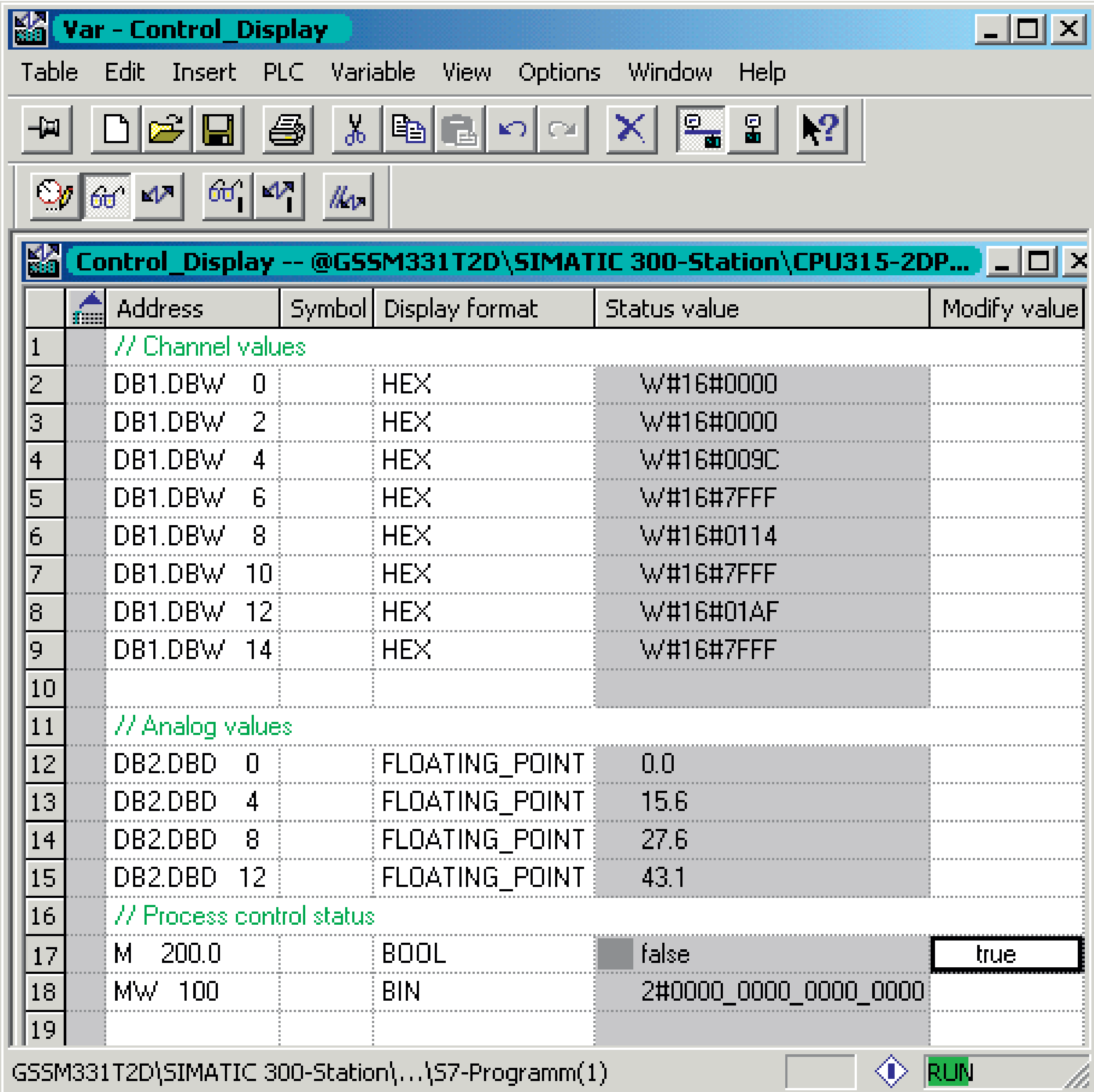


Figure 7-4 Controlling variables

7.3 Analog value representation

7.3.1 Introduction

Introduction

Analog input modules convert the analog process signal into a digital format (16-bit word).

If you want to display analog process values, you must convert the digital values of the module into decimal values.

In our example program, the process value is displayed in the readable format, Volts (V), or in °C for the PT100. The conversion of the digital values into a decimal value is implemented by the programmed function in the FC1.

For converting from the digital value to the analog value, you must take five areas of application into consideration. These areas are described in the following tables.

7.3.2 Display of analog value of a ±5V voltage measuring transducer

Display of analog value of a ±5V voltage measuring transducer

Analog value representation		Voltage measuring range	Area of application	Remark
Decimal	Hexadecimal			
32767	7FFF	5.926 V	Overflow	From the hex value 16#7F00 on, the sensor value is above the overload range and is no longer valid.
32512	7F00			
32511	7EFF	5.879 V	Overload range	This range corresponds to a tolerance band before the overflow is reached. Within this measuring range, the resolution is no longer optimal
27644	6C01			
27648	6C00	5 V	Nominal range	The nominal range is the normal range for recording measurement values. This range guarantees optimal resolution
20736	5100	3.75 V		
1	1	180.8µV		
0	0	0V		
-1	FFFF			
-20736	AF00	-3.75 V		
-27648	9400	-5 V		
-27649	93FF		Undershoot range	Range corresponding to the overload range but for low values
-32512	8100	-5.879V		
-32513	80FF		Underflow	From hex value 16#80FF on, the sensor value is below the configured measurement value range and is no longer valid.
-32768	8000	-5.926V		

With the aid of a voltage sensor (calibration device), you can now compare the specified values to the values in the analog value display in the table. The values will be identical.

7.3.3 Display of analog value of a ±10V voltage measuring transducer

Analog value display in the ±10V voltage measuring range

Analog value representation		Voltage measuring range	Area of application	Remark
Decimal	Hexadecimal			
32767	7FFF	11.851V	Overflow	From the hex value 16#7F00 on, the sensor value is above the overload range and is no longer valid.
32512	7F00			
32511	7EFF	11.759V	Overload range	This range corresponds to a tolerance band before the overflow is reached. Within this measuring range, the resolution is no longer optimal
27644	6C01			
27648	6C00	10V	Nominal range	The nominal range is the normal range for recording measurement values. This range guarantees optimal resolution
20736	5100	7.5V		
1	1	361.7µV		
0	0	0V		
-1	FFFF			
-20736	AF00	-7.5V		
-27648	9400	-10 V		
-27649	93FF		Undershoot range	Range corresponding to the overload range but for low values
-32512	8100	-11.759 V		
-32513	80FF		Underflow	From hex value 16#80FF on, the sensor value is below the configured measurement value range and is no longer valid.
-32768	8000	-11.851V		

7.3.4 Display of analog value of a 0-10V voltage measuring transducer

Analog value display in the 0-10V voltage measuring range

Analog value representation		Voltage measuring range	Area of application	Remark
Decimal	Hexadecimal			
32767	7FFF	11.851V	Overflow	From the hex value 16#7F00 on, the sensor value is above the overload range and is no longer valid.
32512	7F00			
32511	7EFF	11.759V	Overload range	This range corresponds to a tolerance band before the overflow is reached. Within this measuring range, the resolution is no longer optimal
27644	6C01			
27648	6C00	10V	Nominal range	The nominal range is the normal range for recording measurement values. This range guarantees optimal resolution
20736	5100	7.5V		
1	1	361.7µV		
0	0	0V		
			Undershoot range	Negative values are not supported

7.3.5 Analog value display of a standard PT100

Analog value display for a standard PT100 resistance thermometer

Analog value representation		Voltage measuring range	Area of application	Remark
Decimal	Hexadecimal			
32.767	7FFF	> 1,000°C	Overflow	From the hex value 16#2711 on, the sensor value is above the overload range and is no longer valid.
10.000	2710	1,000°C	Overload range	This range corresponds to a tolerance band before the overflow is reached. Within this measuring range, the resolution is no longer optimal
	...	...		
8.501	2135	850.1°C		
8.500	2134	850°C	Nominal range	The nominal range is the normal range for recording measurement values. This range guarantees optimal resolution
	…	…		
	…	…		
-2.000	F830	-200 °C		
-2.001	F82F	-200.1 °C	Undershoot range	Range corresponding to the overload range but for low values
-2.430	F682	-243 °C		
-2431	F681		Underflow	From hex value 16#F681 on, the sensor value is below the configured measurement value range and is no longer valid.
-32.768	8000	< -243 °C		

7.3.6 Effect of the PT100 wiring on the analog value display

Overview

The PT 100 wiring significantly influences the measured value acquisition.

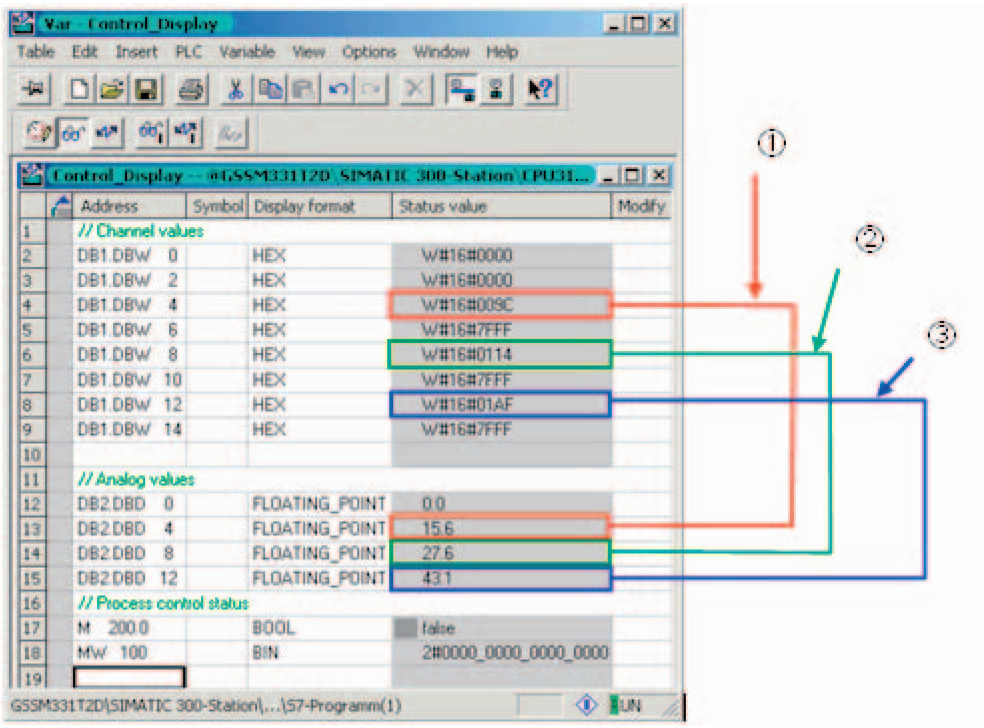
The connecting cables from the SM331 module to the PT100 thermometer have a resistance that is independent of the conducting material, the length, and the cable cross-section.

If you want to compensate for this cable resistance, you must select the 4-wire connection or 3-wire connection.

Measure the ambient temperature with the 3 PT100s and a cable resistance of 5 Ohms (170m copper cable, 0.6mm2). The following measured values will be displayed:

Influence of the wiring on the temperature acquisition

PT100 connection variants	Ambient temperature	Analog value representation	measured temperature	Absolute error
4-cables	17.0°C	00AA Hex.	17.0°C	0°C
3-cables	17.0°C	013C Hex.	31.6°C	14.6°C
2-cables	17.0°C	01BD Hex.	44.5°C	27.5°C



Direct comparison of the three wiring options

(1) PT100 4-wire connection
(2) PT100 3-wire connection
(3) PT100 2-wire connection

Note

The cable resistance does not rise along with the temperature. It remains constant. If you are measuring high temperatures, the inaccuracy is reduced by a percentage value.

8 Diagnostic interrupt

8.1 Initiating the diagnostic interrupt

General information

Diagnostic interrupts enable the user program to react to hardware errors.

Modules must have diagnostic capabilities in order to also be able to generate diagnostic interrupts.

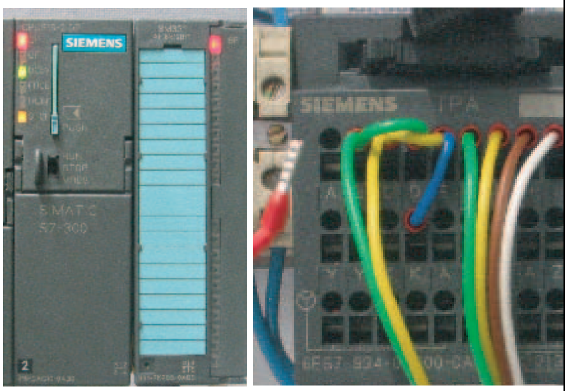
In OB82, you program the reactions to diagnostic interrupts.

Diagnostic interrupt

Analog input module SM331 AI8x12bit has diagnostic capabilities.

Diagnostic interrupts that occur are signaled by the red "SF" LED on the SM331 module and on the CPU (see red circle).

Generating a hardware error

Graphic	Description
	Remove the power supply at terminal 1 on the front connector of the module or terminal Y on the TOP connect terminal block. Result: A diagnostic interrupt is initiated; the red "SF" LED lights up.

The cause of the error can be determined "online" by requesting the hardware status. .

In order to determine the state of module "online", proceed as follows:

- In the hardware configuration, click on the SM331
- In the target system / module status menu, call up the hardware diagnostics

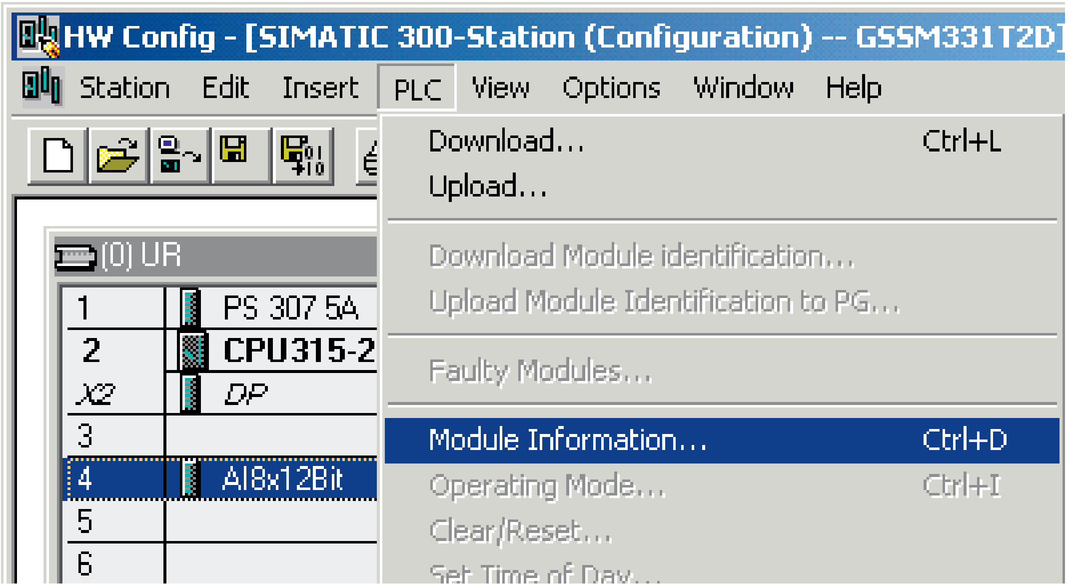


Figure 8-1 Module status

8.2 General diagnostic message

Diagnostic interrupt tab

On the Diagnostic Interrupt tab, you will find information about the reported error.

Any interrupts that occur are not channel dependent and apply to the entire module.

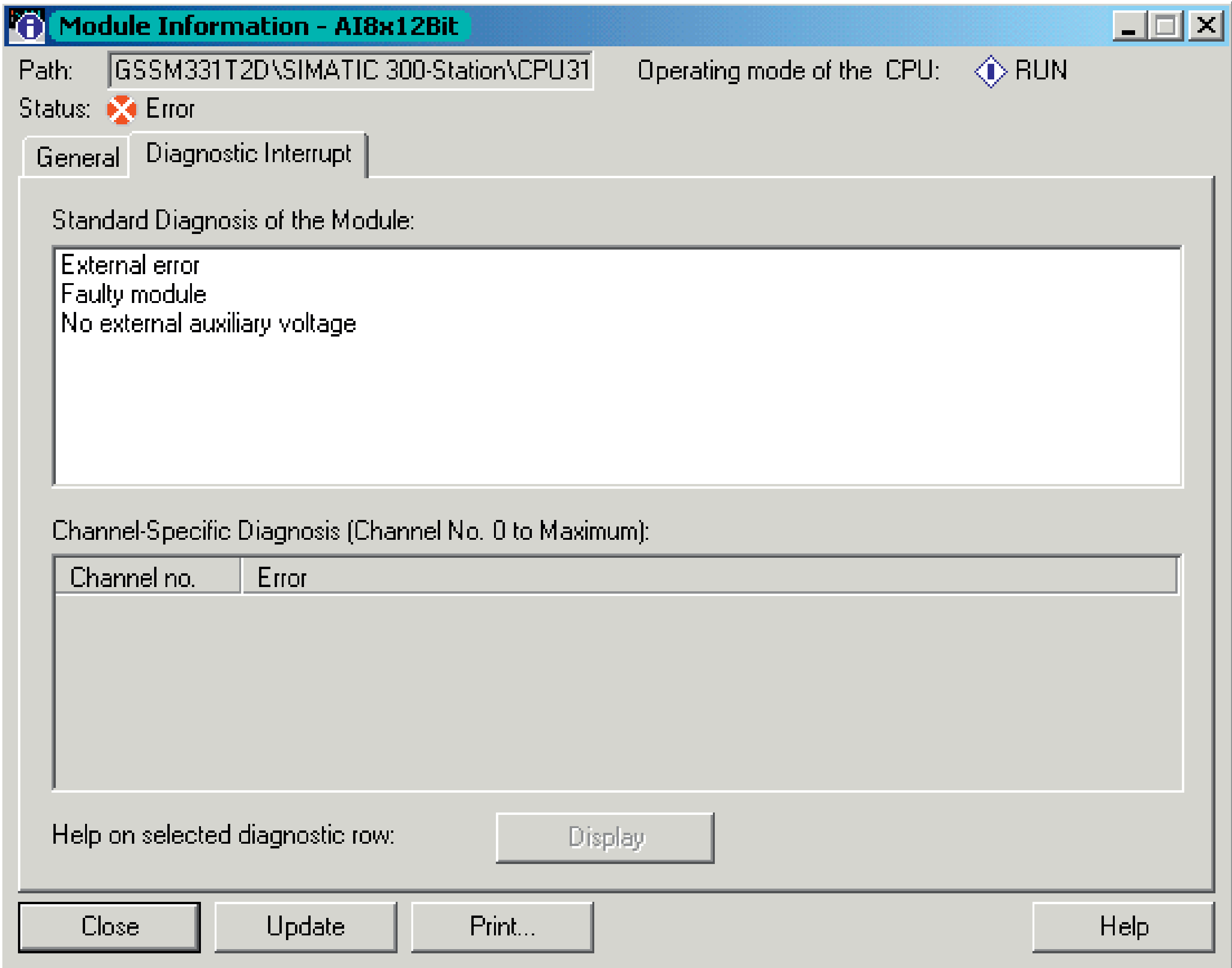


Figure 8-2 Diagnostics for SM331

8.3 Channel-linked diagnostic messages

8.3.1 Types of diagnostic messages

Channel-linked diagnostic messages

There are five types of channel-linked diagnostic messages:

- Configuration / programming error
- Common mode error
- Wire break
- Underflow
- Overflow

Note

This only shows you the channel-linked diagnostics for measuring modes of 2 or 4-wire current transducers. Other measuring modes are similar but are not described here.

8.3.2 Configuration / programming error

Meaning

The position of the measuring range modules does not match the measuring mode set in the hardware configuration.

8.3.3 Common mode error

Meaning

The voltage difference U_{cm} between the inputs (M-) and the common voltage potential of the measuring circuit (M_{ana}) is too high.

In the example, this error cannot occur because M_{ana} was connected to M for the voltage measuring transducer.

8.3.4 Wire breakage (only for the PT100 measuring mode)

Meaning

If the measuring mode is set to voltage, there is no way to check a wire breakage. You also cannot select it in SIMATIC Manager. In PT100 measuring mode, a wire breakage is detected and reported.

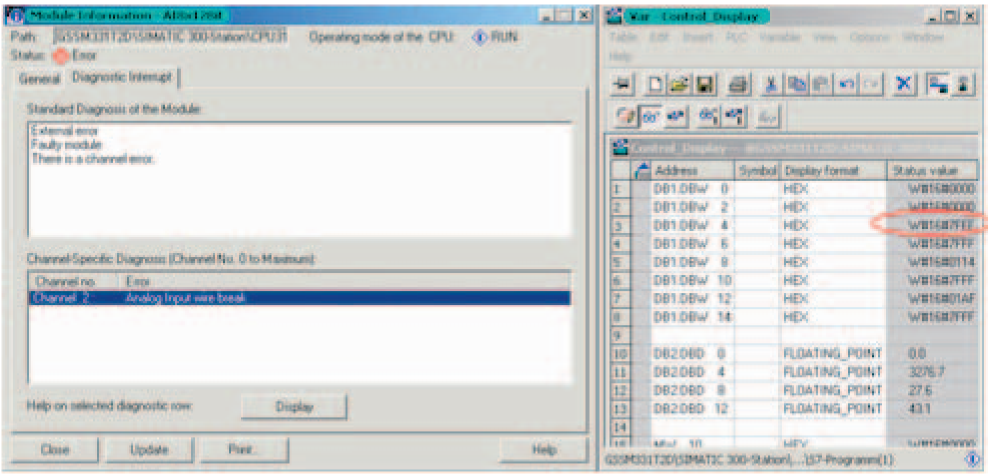


Figure 8-3 Left: Diagnostic message with wire break / Right: Variable table

The analog value display immediately goes into the overflow range (HEX 7FFF), since the channel measures infinite resistance.

See also

Configuring the analog module (Page 6-10)

8.3.5 Underflow

Meaning

The two measuring modes, voltage and PT100, can initiate the diagnostic message "Analog input measuring range/lower limit value undershot".

Voltage

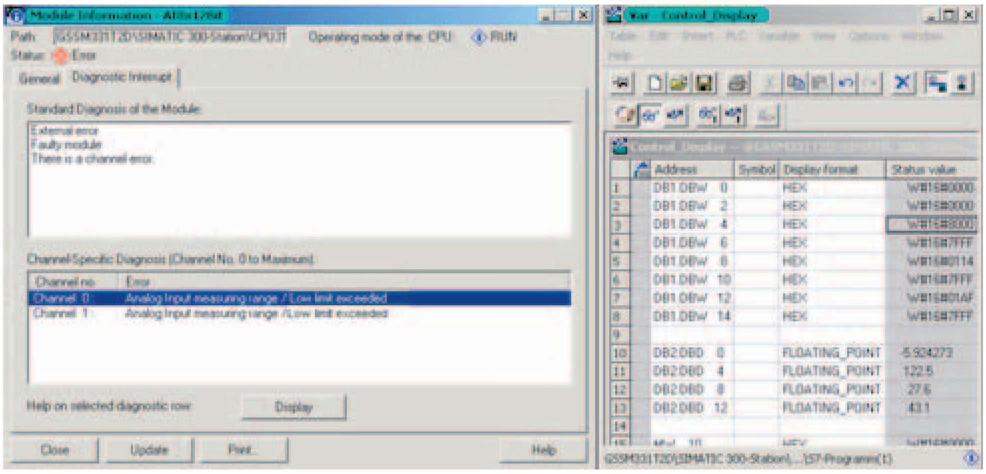


Figure 8-4 Left: Diagnostic display in the underflow range/Right: Variable table

We have connected the 2 channels in parallel in order to retain the diagnostic capability of the channel group. Logically, we also receive the diagnostic message for the second channel.
If you receive this message during commissioning, check to see whether the measuring range of the measuring transducer and its configuration match each other.

PT100

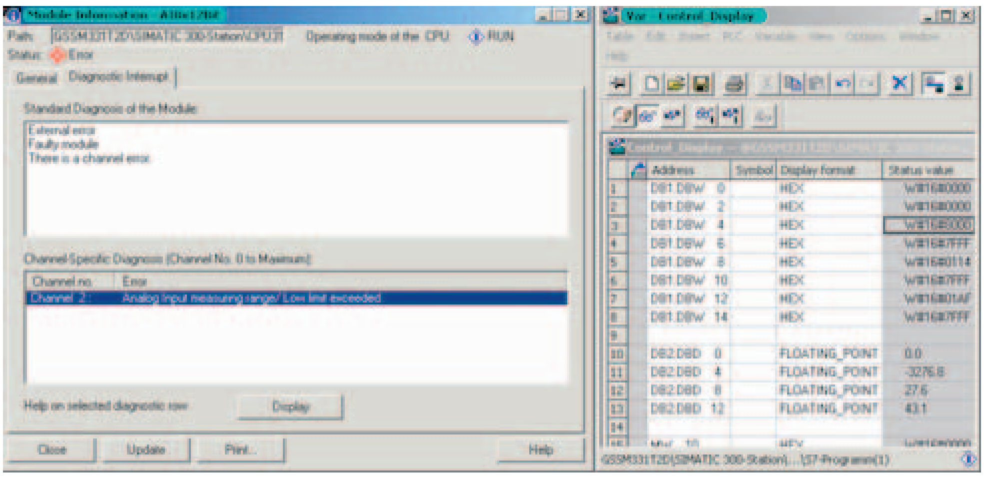


Figure 8-5 Left: Diagnostic display in the underflow range/Right: Variable table

This message is generated if the temperature is under -243 °C or if the PT100 has very low resistance. Most probably, the PT100 connected has a short-circuit or the PT100 is defective.

8.3.6 Overflow

Meaning

In the two measuring modes, voltage and PT100, the diagnostic message "Analog input measuring range/upper limit value exceeded" can be initiated.

"Voltage" measuring mode

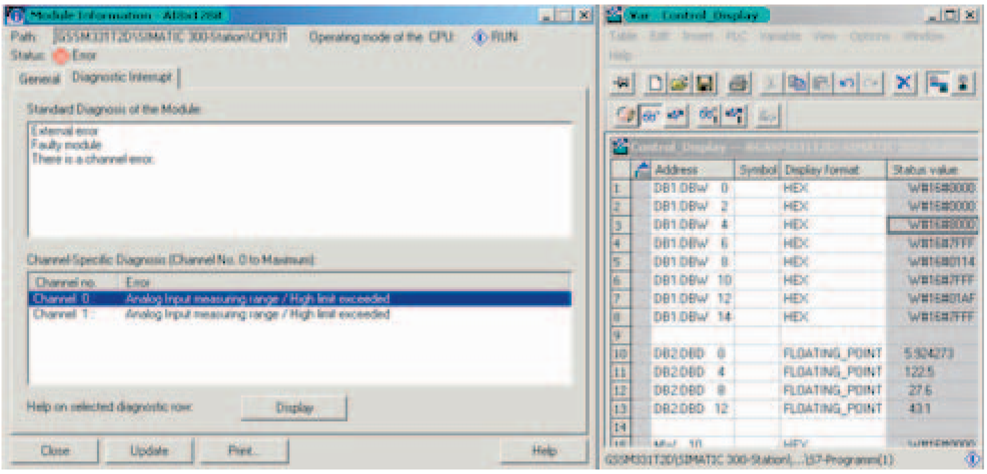


Figure 8-6 Left: Diagnostic message with overflow / Right: Variable table

PT100 measuring mode

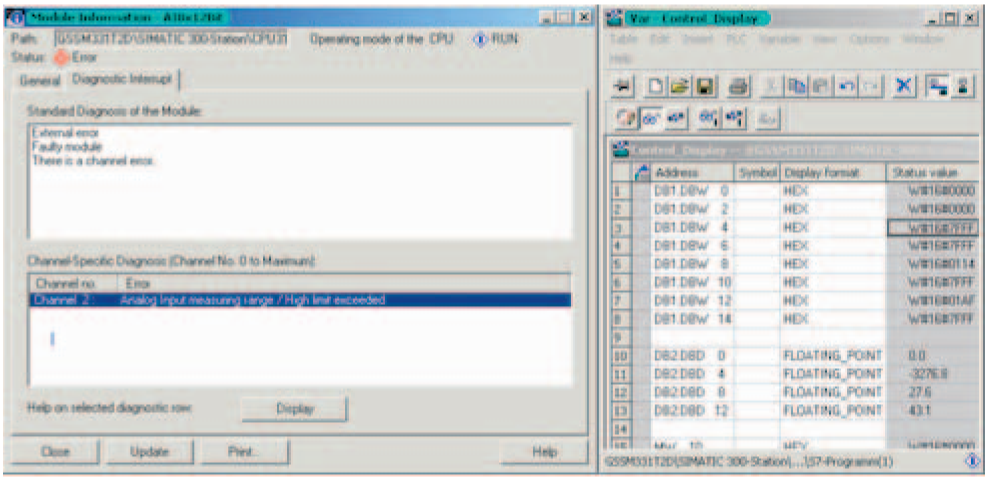


Figure 8-7 Left: Diagnostic message with overflow / Right: Variable table

Hardware interrupt 9

9.1 Hardware interrupt

Overview

A special feature of the SM331 AI8x12bit is the capability to also trigger hardware interrupts. The two channels, 0 and 2, can be correspondingly configured.

Defining the limit values for hardware interrupts:

For the PT100 resistance thermometer, you must define the limit values in °C and not in °F or K.

For the voltage measuring transducer, you must define the limit values in Volts (V) and not in the unit of the connected sensor.

Example:

You have a pressure sensor with the physical unit in Pascals (Pa). You do not, however, specify the limit values in Pascals. Instead, you use the corresponding value of the voltage measuring transducer in Volts.

Features of hardware interrupt triggering

In order to trigger a hardware interrupt, the limit values have to be within the nominal range of the measuring mode.

Example:

You use a voltage measuring transducer (±5V) with a nominal range of -5V and +5V. If you have entered -6V as the lower limit value, these settings are accepted by the system, but the hardware interrupt is never triggered, because the diagnostic interrupt (underflow of the nominal range) is always activated beforehand.

In our example, we have configured channel 0 (voltage measuring transducer) with the following limit values:

- Lower limit value -3V
- Upper limit value +3V

If these values are undershot or exceeded within the nominal value range, hardware interrupt OB40 is triggered.

OB40 hardware interrupt

Hardware interrupts generally trigger alarm organization blocks in the CPU. In our example, OB40 is called up.

In the STEP 7 program, OB40 is used for hardware interrupts. Depending on the CPU, several hardware interrupts can be configured.

If a hardware interrupt occurs, OB40 is called up. In the OB40 user program, you can program the reaction of the automation system to hardware interrupts.

In the example user program, OB40 reads the cause of the hardware interrupt. This can be found in temporary variable structure OB40_POINT_ADDR (local bytes 8 to 11).

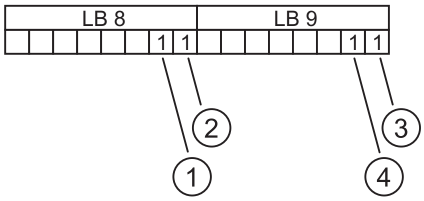


Figure 9-1 OB40 start information: Which event has violated limits and triggered a hardware interrupt

(1) Value exceeds upper limit in channel 1
(2) Value exceeds upper limit in channel 0
(3) Violation of the lower limit in channel 0
(4) Violation of the lower limit in channel 1

In the example, OB40 only transfers local data variables LD8 and LD9 into a marker word (MW100). The marker word is monitored in the existing variable table.

You can acknowledge the marker word in OB1 by setting marker bit M200.0 or by setting it to TRUE in the variable table.

Simulation of a hardware interrupt

If you provide channel 0 with 4V using a calibration device, you will receive the binary value 0000 0001 0000 0000 in MW100. This means that OB40 was called up and the upper limit of >4V has been exceeded in channel 0.

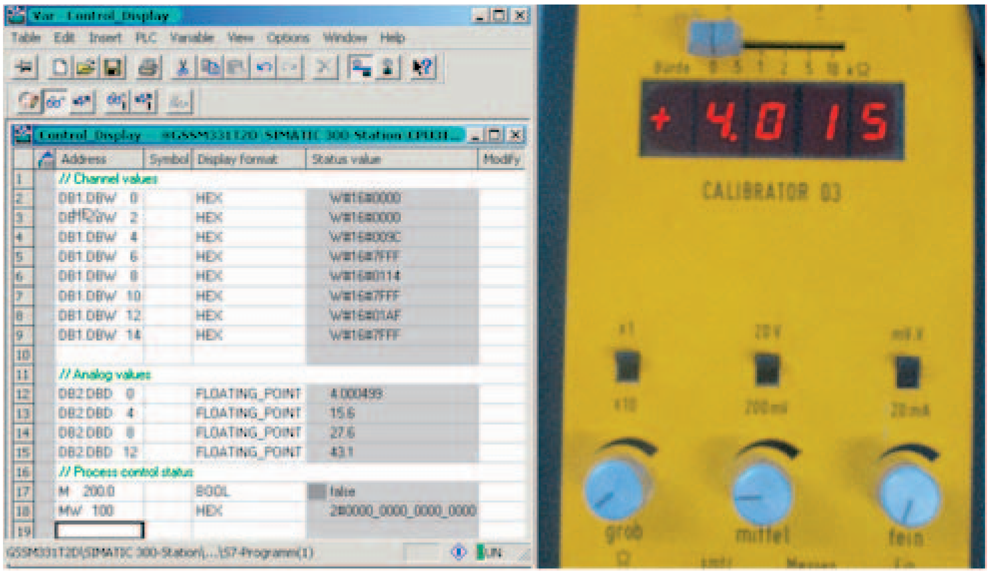


Figure 9-2 Process interrupt: Violation of lower limit in channel 0

Appendix

A

A.1 Source code of the user program

Overview

In this section, you can get a quick overview of the functions of the user program for the example station. A flowchart shows you the rough outline of the program structure. You will find the complete program in detail in the STL source code.

For your own use, you can also download the STL source code directly in the form of an STL file from the HTML page from which you loaded this "Getting Started" manual.

Flowchart

The texts marked in red correspond to the source code in the user program.

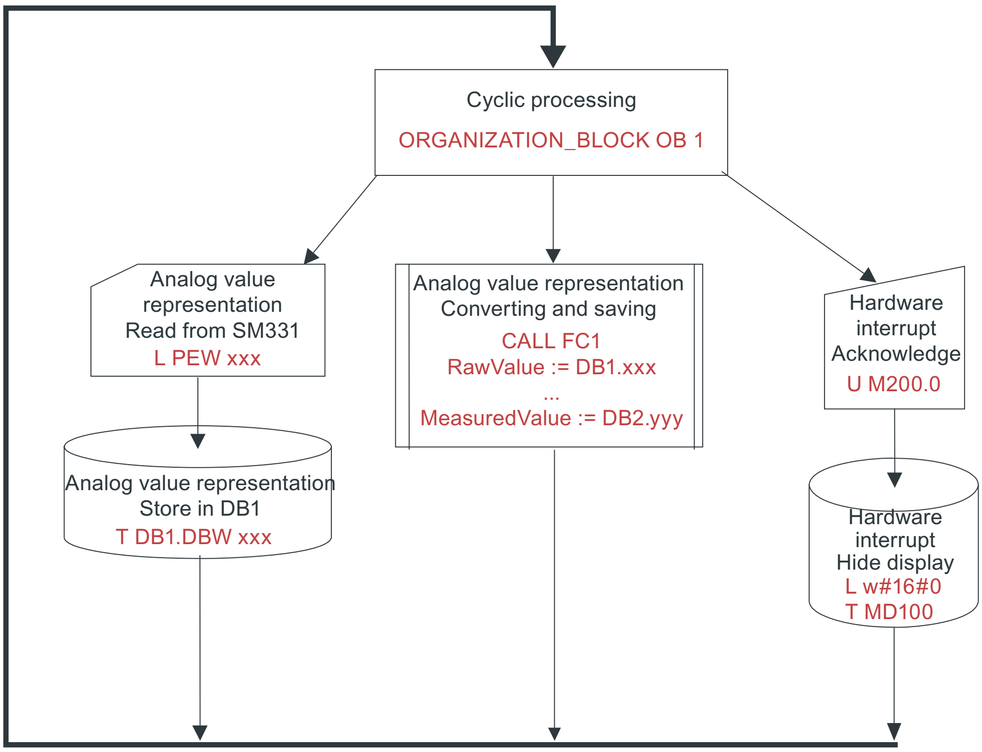


Figure A-1 OB1 Flowchart

Variable description

Variables	Description
DB1.DBW 0	Channel 0 Display of analog value
DB1.DBW 2	Channel 1 Display of analog value
DB1.DBW 4	Channel 2 Display of analog value
DB1.DBW 6	Channel 3 Display of analog value
DB1.DBW 8	Channel 4 Display of analog value
DB1.DBW 10	Channel 5 Display of analog value
DB1.DBW 12	Channel 6 Display of analog value
DB1.DBW 14	Channel 7 Display of analog value
DB2.DBD 0	Measuring transducer 1 voltage (V)
DB2.DBD 4	PT100 with 4-wire connection (°C)
DB2.DBD 8	PT100 with 3-wire connection (°C)
DB2.DBD 12	PT100 with 2-wire connection (°C)
M200.0	Acknowledge hardware interrupt
MW 100	Status hardware interrupt

STL source code

```
DATA_BLOCK DB 1
TITLE =Analog value display
VERSION : 0.1

  STRUCT
     CH_0 : INT;      //Channel 0
     CH_1 : INT;      //Channel 1
     CH_2 : INT;      //Channel 2
     CH_3 : INT;      //Channel 3
     CH_4 : INT;      //Channel 4
     CH_5 : INT;      //Channel 5
     CH_6 : INT;      //Channel 6
     CH_7 : INT;      //Channel 7
  END_STRUCT ;
BEGIN
     CH_0 := 0;
     CH_1 := 0;
     CH_2 := 0;
     CH_3 := 0;
     CH_4 := 0;
     CH_5 := 0;
     CH_6 := 0;
     CH_7 := 0;
END_DATA_BLOCK
```

```
DATA_BLOCK DB 2
TITLE =process values
VERSION : 0.1

  STRUCT
     SE_1 : REAL ;           //Voltage Transducer
     SE_2 : REAL ;           //PT100 (4)
     SE_3 : REAL ;           //PT100 (3)
     SE_4 : REAL ;           //PT100 (2)
  END_STRUCT ;
BEGIN
     SE_1 := 0.000000e+000;
     SE_2 := 0.000000e+000;
     SE_3 := 0.000000e+000;
     SE_4 := 0.000000e+000;
END_DATA_BLOCK

FUNCTION FC 1 : VOID
TITLE =Converting analog value display to process values
VERSION : 0.1

VAR_INPUT
  RawValue : INT;
  Factor : REAL ;
  Offset : REAL ;
  OverFlow : INT;
  OverRange : INT;
  UnderRange : INT;
  UnderFlow : INT;
END_VAR
VAR_OUTPUT
  MeasuredValue : REAL ;
  Status: WORD ;
END_VAR
VAR_TEMP
  TInt : INT;
  TDoubleInt : DINT ;
  TFactor : REAL ;
  TOffset : REAL ;
  TFactor1 : DINT ;
  TFactor2 : REAL ;
END_VAR
BEGIN
NETWORK
TITLE = Conversion

            L    #RawValue;
            ITD  ;
            DTR  ;
```

```
            L    #Factor;
            *R   ;
            L    #Offset;
            +R   ;
            M    #MeasuredValue;

NETWORK
TITLE =Analog value display monitoring

            L        W#16#0;
            M        #Status;

            L        #RawValue;
            L        #OverFlow;
            >=I      ;
            JC       m_of;

            L        #RawValue;
            L        #OverRange;
            >=I      ;
            JC       m_or;

            L        #Raw#
                     Value;
            L        UnderFlow;
            <=I      ;
            JC       m_uf;

            L        #RawValue;
            L        #UnderRange;
            <=I      ;
            JC       m_ur;

            JL       end;

m_of: L     W#16#800;
            M             #Status;
            JL            end;

m_or: L     W#16#40
            0;
            M        #Status;
            JL       end;

m_uf: L     W#16#200;
            M        #Status;
            JL       end;
```

```
m_ur: L     W#16#100;


            M       #Status;
            JL      end;


end:  NOP   0;
END_FUNCTION
ORGANIZATION_BLOCK OB 1
TITLE = "Main Program Sweep (Cycle)"
VERSION : 0.1

VAR_TEMP
  OB1_EV_CLASS : BYTE ;             //Bits 0-3 = 1 (Coming event), Bits 4-7
                                    = 1 (Event class 1)
  OB1_SCAN_1 : BYTE ;               //1 (Cold restart scan 1 of OB 1), 3
                                    (Scan 2-n of OB 1)
  OB1_PRIORITY : BYTE ;             //Priority of OB Execution
  OB1_OB_NUMBR : BYTE ;             //1 (Organization block 1, OB1)
  OB1_RESERVED_1 : BYTE ;           //Reserved for system
  OB1_RESERVED_2 : BYTE ;           //Reserved for system
  OB1_PREV_CYCLE : INT;             //Cycle time of previous OB1 scan
                                    (milliseconds)
  OB1_MIN_CYCLE : INT;              //Minimum cycle time of OB1
                                    (milliseconds)
  OB1_MAX_CYCLE : INT;              //Minimum cycle time of OB1
                                    (milliseconds)
  OB1_DATE_TIME : DATE_AND_TIME ;   //Date and time OB1 started
END_VAR
BEGIN
NETWORK
TITLE =Transfer of the channel values to data block DB 1

// Channel 0 -> Data block
            L    PEW 256;
            M    DB1.DBW     0;


// Channel 1 -> Data block
            L    PEW 258;
            M    DB1.DBW     2;


// Channel 2 -> Data block
            L    PEW 260;
            M    DB1.DBW     4;


// Channel 3 -> Data block
            L    PEW 262;
            M    DB1.DBW     6;


// Channel 4 -> Data block
```

```
            L    PEW 264;
            M    DB1.DBW     8;

// Channel 5 -> Data block
            L    PEW 266;
            M    DB1.DBW     10;

// Channel 6 -> Data block
            L    PEW 268;
            M    DB1.DBW     12;

// Channel 7 -> Data block
            L    PEW 270;
            M    DB1.DBW     14;

NETWORK
TITLE =Converting the analog value display -> measured value

// Channel 1 : Voltage measuring transducer 1 to 5V
            CALL FC   1 (
                      RawValue        := DB1.DBW           0,
                      Factor          := 1.447000e-004,
                      Offset          := 1.000000e+000,
                      OverFlow        := 32512,
                      OverRange       := 27649,
                      UnderRange      := -1,
                      UnderFlow       := -4865,
                      MeasuredValue   := DB2.DBD           0,
                      Status          := FW                10);

// Channel 2 : PT100
            CALL FC   1 (
                      RawValue        := DB1.DBW           4,
                      Factor          := 1.000000e-001,
                      Offset          := 0.000000e+000,
                      OverFlow        := 10001,
                      OverRange       := 8501,
                      UnderRange      := -2001,
                      UnderFlow       := -2431,
                      MeasuredValue   := DB2.DBD           4,
                      Status          := FW                20);

// Channel 3 : PT100
            CALL FC   1 (
                      RawValue        := DB1.DBW           8,
                      Factor          := 1.000000e-001,
                      Offset          := 0.000000e+000,
```

```
                    OverFlow       := 10001,
                    OverRange      := 8501,
                    UnderRange     := -2001,
                    UnderFlow      := -2431,
                    MeasuredValue  := DB2.DBD          8,
                    Status         := FW               30);

// Channel 4 : PT 100
           CALL FC   1 (
                    RawValue       := DB1.DBW          12,
                    Factor         := 1.000000e-001,
                    Offset         := 0.000000e+000,
                    OverFlow       := 10001,
                    OverRange      := 8501,
                    UnderRange     := -2001,
                    UnderFlow      := -2431,
                    MeasuredValue  := DB2.DBD          12,
                    Status         := FW               40);

NETWORK
TITLE = Acknowledge hardware interrupt
           U     M       200.0;
           EP    M       200.1;
           SPBN  m001;
           L     0;
           M     FD      100;
           M     FW      104;
           M     FW      106;
           R     M       200.0;
m001:      NOP   0;



END_ORGANIZATION_BLOCK

ORGANIZATION_BLOCK OB 40
TITLE = "Hardware Interrupt"
VERSION : 0.1

VAR_TEMP
  OB40_EV_CLASS : BYTE ;           //Bits 0-3 = 1 (Coming event), Bits
                                   4-7 = 1 (Event class 1)
  OB40_STRT_INF : BYTE ;           //16#41 (OB 40 has started)
  OB40_PRIORITY : BYTE ;           //Priority of OB Execution
  OB40_OB_NUMBR : BYTE ;           //40 (Organization block 40, OB40)
  OB40_RESERVED_1 : BYTE ;         //Reserved for system
  OB40_IO_FLAG : BYTE ;            //16#54 (input module), 16#55 (output
                                   module)
  OB40_MDL_ADDR : WORD ;           //Base address of module initiating
                                   interrupt
```

```
  OB40_POINT_ADDR : DWORD ;          //Interrupt status of the module
  OB40_DATE_TIME : DATE_AND_TIME ;   //Date and time OB40 started
END_VAR
BEGIN
NETWORK
TITLE =

            L  #OB40_IO_FLAG;        //OB40_IO_FLAG    : 16#54 = input
                                                         module
            M  MB              104;  //                : 16#55 = output
                                                         module

            L  #OB40_MDL_ADDR;       //OB40_MDL_ADDR   : Start address of
                                                         the
            M  FW              106;  //                  module to be
                                                         triggered

            L  #OB40_POINT_ADDR;     //OB40_POINT_AD   : LB8 = Exceeding
                                     DR                  the
            M  FD              100;  //                  upper limit value
         NOP   0;                    //OB40_POINT_AD   : LB9 =
                                     DR                  Undershooting the
         NOP   0;                    //                  lower limit value
END_ORGANIZATION_BLOCK

ORGANIZATION_BLOCK OB 82
TITLE = "I/O Point Fault"
VERSION : 0.1

VAR_TEMP
  OB82_EV_CLASS : BYTE ;          //16#39, Event class 3, Entering event
                                  state, Internal fault event
  OB82_FLT_ID : BYTE ;            //16#XX, Fault identification code
  OB82_PRIORITY : BYTE ;          //Priority of OB Execution
  OB82_OB_NUMBR : BYTE ;          //82 (Organization block 82, OB82)
  OB82_RESERVED_1 : BYTE ;        //Reserved for system
  OB82_IO_FLAG : BYTE ;           //Input (01010100), Output (01010101)
  OB82_MDL_ADDR : WORD ;          //Base address of module with fault
  OB82_MDL_DEFECT : BOOL;         //Module defective
  OB82_INT_FAULT : BOOL;          //Internal fault
  OB82_EXT_FAULT : BOOL;          //External fault
  OB82_PNT_INFO : BOOL;           //Point information
  OB82_EXT_VOLTAGE : BOOL;        //External voltage low
  OB82_FLD_CONNCTR : BOOL;        //Field wiring connector missing
  OB82_NO_CONFIG : BOOL;          //Module has no configuration data
  OB82_CONFIG_ERR : BOOL;         //Module has configuration error
  OB82_MDL_TYPE : BYTE ;          //Type of module
  OB82_SUB_MDL_ERR : BOOL;        //Sub-Module is missing or has error
  OB82_COMM_FAULT : BOOL;         //Communication fault
  OB82_MDL_STOP : BOOL;           //Module is stopped
  OB82_WTCH_DOG_FLT : BOOL;       //Watch dog timer stopped module
  OB82_INT_PS_FLT : BOOL;         //Internal power supply fault
  OB82_PRIM_BATT_FLT : BOOL;      //Primary battery is in fault
```

```
   OB82_BCKUP_BATT_FLT : BOOL;     //Backup battery is in fault
   OB82_RESERVED_2 : BOOL;         //Reserved for system
   OB82_RACK_FLT : BOOL;           //Rack fault, only for bus interface
                                   module
   OB82_PROC_FLT : BOOL;           //Processor fault
   OB82_EPROM_FLT : BOOL;          //EPROM fault
   OB82_RAM_FLT : BOOL;            //RAM fault
   OB82_ADU_FLT : BOOL;            //ADU fault
   OB82_FUSE_FLT : BOOL;           //Fuse fault
   OB82_HW_INTR_FLT : BOOL;         //Hardware interrupt input in fault
   OB82_RESERVED_3 : BOOL;         //Reserved for system
   OB82_DATE_TIME : DATE_AND_TIME  //Date and time OB82 started
   ;
END_VAR
BEGIN
END_ORGANIZATION_BLOCK
```

See also

General information (Page 1-1)

Index

M

O

P

R

S