SIEMENS

SIMATIC

SM331; AI 8 x 12 Bit

Getting Started
Part 1: 4-20mA

Edition 09 / 2003

A5E00253410

Safety-related Guidelines

This manual contains notices intended to ensure personal safety, as well as to protect the products and connected equipment against damage. These notices are highlighted by the symbols shown below and graded according to severity by the following texts:

Danger
indicates that death, severe personal injury or substantial property damage will result if proper precautions are not taken.

Warning
indicates that death, severe personal injury, or substantial property damage can result if proper precautions are not taken.

Caution
indicates that minor personal injury or property damage can result if proper precautions are not taken.

Note
contains important information about the product, its operation or a part of the document to which special attention is drawn.

Qualified personnel

A device/system may only be commissioned or operated by **qualified personnel**. Qualified personnel as referred to in the safety guidelines in this document are persons authorized to energize, de-energize, clear, ground, and tag circuits, equipment and systems in accordance with established safety practice.

Correct Usage

Please observe the following:

Warning
The equipment/system or the system components may only be used for the applications described in the catalog or the technical description, and only in combination with the equipment, components, and devices of other manufacturers as far as this is recommended or permitted by Siemens.
The product will function correctly and safely only if it is transported, stored, set up, and installed as intended, and operated and maintained with care.

Registered Trademarks

SIMATIC®, SIMATIC HMI® and SIMATIC NET® are registered trademarks of Siemens AG.

Third parties using for their own purposes any other names in this document which refer to trademarks might infringe upon the rights of the trademark owners.

Disclaimer of Liability

We have checked the contents of this manual for agreement with the hardware and software described. Since deviations cannot be precluded entirely, we cannot guarantee full agreement. However, the data in this manual are reviewed regularly and any necessary corrections included in subsequent editions. Suggestions for improvement are welcomed.

Siemens AG
Bereich Automation and Drives
Geschäftsgebiet Industrial Automation Systems
Postfach 4848, D- 90327 Nürnberg

Siemens Aktiengesellschaft

A5E00253410

Contents:

1 Preface

Purpose of the Getting Started

The Getting Started gives you a complete overview of the commissioning of the analog module SM331. It assists you in the installation and parameterization of the hardware of a 4-20mA sensor and the configuration with SIMATIC S7 Manager.

The target audience of the Getting Started is a beginner with only basic experience in configuration, commissioning and servicing of automation systems.

What to expect

Step by step the procedures from mounting the module to storing analog values in the STEP7 user program are explained in detail by means of an example. In the following sections you will be introduced to:

- Problem analysis
- Mechanical setup of the sample station
- Electrical connection of the sample station
- Configure hardware with SIMATIC Manager using HW Config
- Creating a small user program with STEP7 which stores the read analog values in a data block
- Triggering and interpreting diagnostic and hardware interrupts

2 Requirements

2.1 Required basic knowledge

No special knowledge in the area of automation technique is required to understand this description. As the configuration of the analog module is done with the software STEP7, proficiency in STEP7 would be advantageous.

Further information on STEP7 can be found in the electronic manuals that were delivered with STEP7.

Knowledge of PC or similar computer devices (e.g. programming devices) using the operating system Windows 95/98/2000/NT or XP is assumed.

2.2 Required Hardware and Software

The scope of delivery of the analog module consists of two parts:
The module itself and a front connector, which enables it to comfortably connect the power supply and the data connections.

Table 2-1 Components of the analog module

Quantity	Article	Order number
1	SM 331, Electrically ISOLATED 8 AI, ALARM DIAGNOSTICS	6ES7331-7KF02-0AB0
1	20-pin FRONT CONNECTOR with spring contacts	6ES7392-1BJ00-0AA0

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

Table 2-2 SIMATIC components of the sample station

Quantity	Article	Order number
1	PS 307 Power Supply AC 120/230V, DC 24V, 5A	6ES7307-1EA00-0AA0
1	CPU 315-2DP	6ES7315-2AG10-0AB0
1	MICRO MEMORY CARD, NFLASH, 4 MBYTE	6ES7953-8LM00-0AA0
1	SIMATIC S7-300, RAIL L=530MM	6ES7390-1AF30-0AA0
1	Programming device (PD) with MPI-interface and MPI cable PC with corresponding interface card	Depending on the configuration

Table 2-3 Software STEP7

Quantity	Article	Order number
1	STEP7 Software version 5.2 or later, installed on the programming device.	6ES7810-4CC06-0YX0

The following current transducers can be used for the acquisition of analog signals:

Table 2-4 Current transducers

Quantity	Article	Order number
1	2-Wire current transducer	Depending on the manufacturer
1	4-Wire current transducer	Depending on the producer

Note

This „Getting Started“ describes only the application of 4 – 20 mA current transducers in the 2-Wire or 4-Wire model. If you want to use other transducers, then you have to wire and parameterize the SM331 differently.

Furthermore, the following tools and materials are necessary:

Table 2-5 General tools and materials

Quantity	Article	Order number
Multiple	M6-bolts and nuts (Length depending on the mounting place)	standard
1	Screwdriver with blade width 3,5 mm	standard
1	Screwdriver with blade width 4,5 mm	standard
1	Wire cutting pliers and tools for stripping	standard
1	Tools to mount the cable end sleeve	standard
X m	Wire for grounding the rail, 10 mm^2 diameter. Ring terminal with 6,5 mm hole, length according to local conditions.	standard
X m	Flexible wire with 1mm^2 diameter with fitting wire end sleeves, Form A in 3 different colors – blue, red and green	standard
X m	3-wire power cord (AC 230/120V) with protective contact socket, length according to local conditions.	standard
1	Calibration device (Measuring instrument for commissioning, that can measure and supply current)	Depending on the manufacturer

3 Task

You want to connect three analog inputs to your station. One of them should have a 2-wire current transducer and the other two share a 4-wire current transducer.

You need failure diagnostic capabilities and want two sensors to be able to trigger hardware interrupts.

You have the analog input module SM331, AI8x12 Bit (order number 6ES7 331-7KF02-0AB0) available. The module is diagnostic and hardware interrupt capable and can process up to 8 analog inputs. Different measuring modes can be configured for each module (e.g. 4- 20 mA; PT 100; Thermocouple).

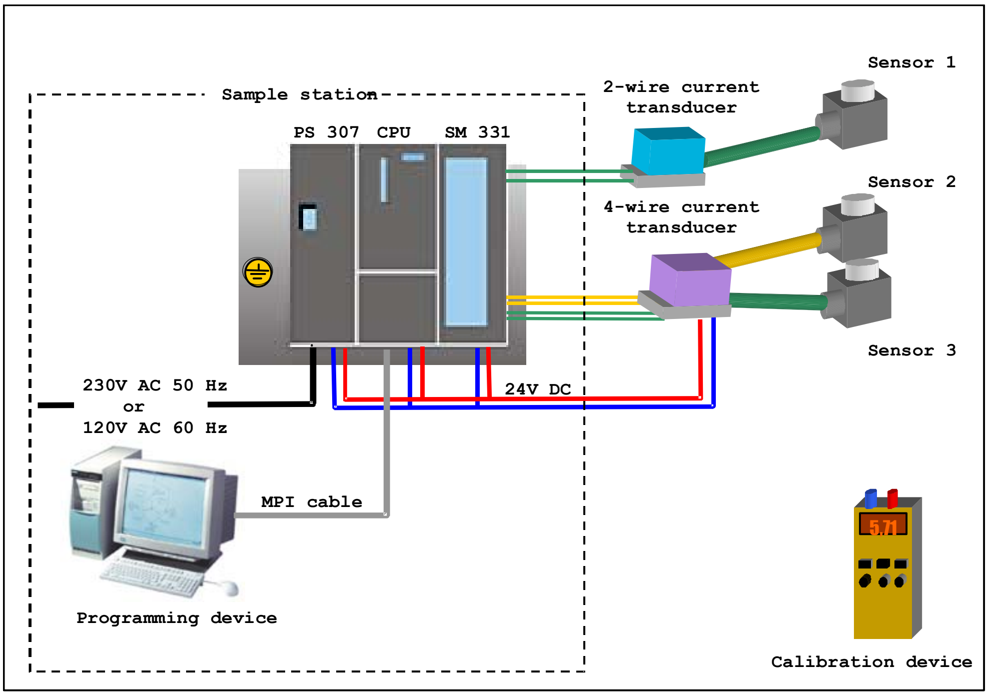


Figure 3-1 Sample station components

You will be guided through these steps

- Mechanical setup of the sample station (see chapter 4)
    - General mounting instructions for S7-300 modules
    - Configuration of the SM331 for the two selected measurement transducer types
- Electrical wiring of the sample station (see chapter 5)
    - Wiring of the power supply and the CPU
    - Wiring of the analog module
    - Standard pin layout of two measurement transducer types
    - Wiring of unused inputs
- Configuration with SIMATIC Manager (see chapter 6)
    - Use of project wizard
    - Completing the automatically generated hardware configuration
    - Integration of the supplied user program source
- User program testing (see chapter 7)
    - Interpreting the read values
    - Converting the measured values into readable analog values
- Utilizing the diagnostic capabilities of the SM331 module (see chapter 8)
    - Triggering a diagnostic interrupts
    - Analyzing the diagnostics
- Application of hardware interrupts (see chapter 8)
    - Parameterization of hardware interrupts
    - Configuration and analysis of hardware interrupts

4 Mechanical setup of the sample station

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

4.1 Mounting the sample station

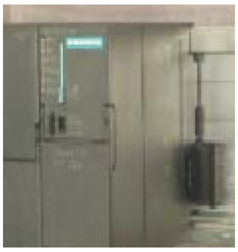
Before you can use the analog input module SM331, you need a basic setup of general SIMATIC S7-300 components.

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

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

Table 4-1 Mounting the sample station (without SM331)

Graphics	Description
	Bolt together the rail to the ground or underground (screw size: M6) so that at least 40 mm space remains above and below the rail If the base is a grounded sheet metal or a grounded mounting plate, ensure that the rail and the base are connected together with low resistance.
	Connect the rail with the protective ground wire. A M6 bolt is available for this purpose.
	Mounting the power supply • Hang the power supply to the top end of the rail
	• and tighten it to the rail underneath

Graphics	Description
	Connect the bus connector (delivered with the SM331) to the left connector on the back of the CPU
	Mounting the CPU: • Hang the CPU to the top end of the rail • Push it all the way left to the power supply • Push it down • and tighten the screw to the rail underneath

4.2 Mounting the analog module

Before the actual mounting of the SM331 the module has to be completed with a front connector and the desired measurement mode of the inputs is set.

In this section you will learn

- Which components you need
- What are the properties of the analog input module
- What a measuring range module is and how it is set up
- How you mount the already setup module

4.2.1 Components of the SM331

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 contacts (Order number 6ES7392-1BJ00-0AA0)
  - With screw contacts (Order number 6ES7392-1AJ00-0AA0)

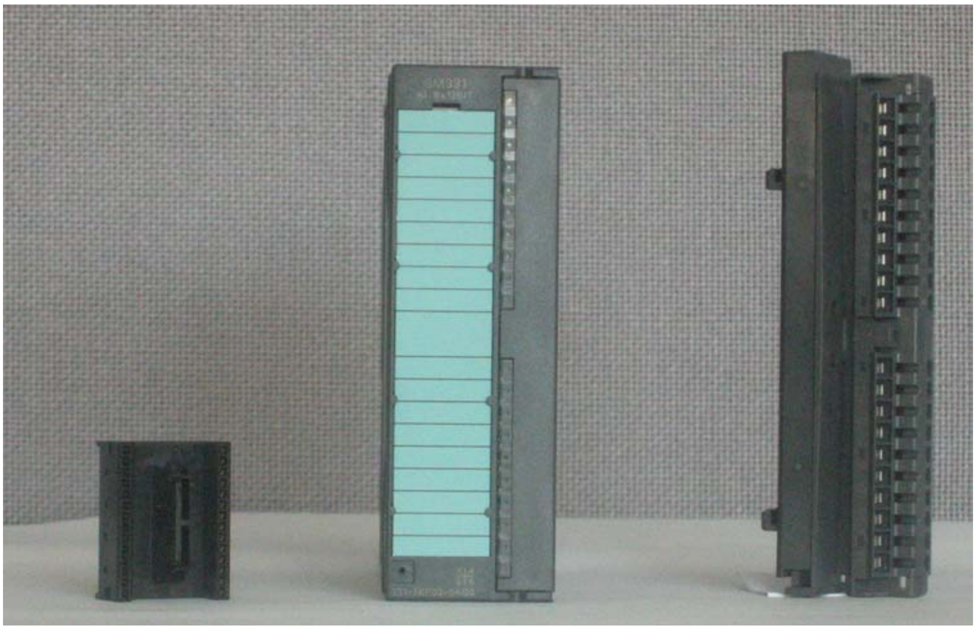


Figure 4-1 Components of the SM331

Table 4-2 The scope of delivery of SM331

Components
Module
Labeling strips
Bus connector
2 cable ties (not in the picture) to tie the external wiring

4.2.2 Properties of the analog module

- 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
  - Resistance
  - Temperature
- Configurable diagnostic interrupt
- Two channels with limit value interrupt (Only channel 0 and channel 2 are configurable)
- Electrically isolated against backplane bus
- Electrically isolated against load voltage (exception: At least one module is set to position D)

The module is a universal analog module designed for the most commonly used applications.

The desired measuring mode should be set up directly on the module with the measuring range modules (see chapter 4.2.3)

4.2.3 Measuring range modules

The module SM331 has four measuring range modules (one per channel group). The measuring range modules can be set to 4 different positions (A, B, C or D). With the set position you determine which transducer can be connected to the respective channel group.

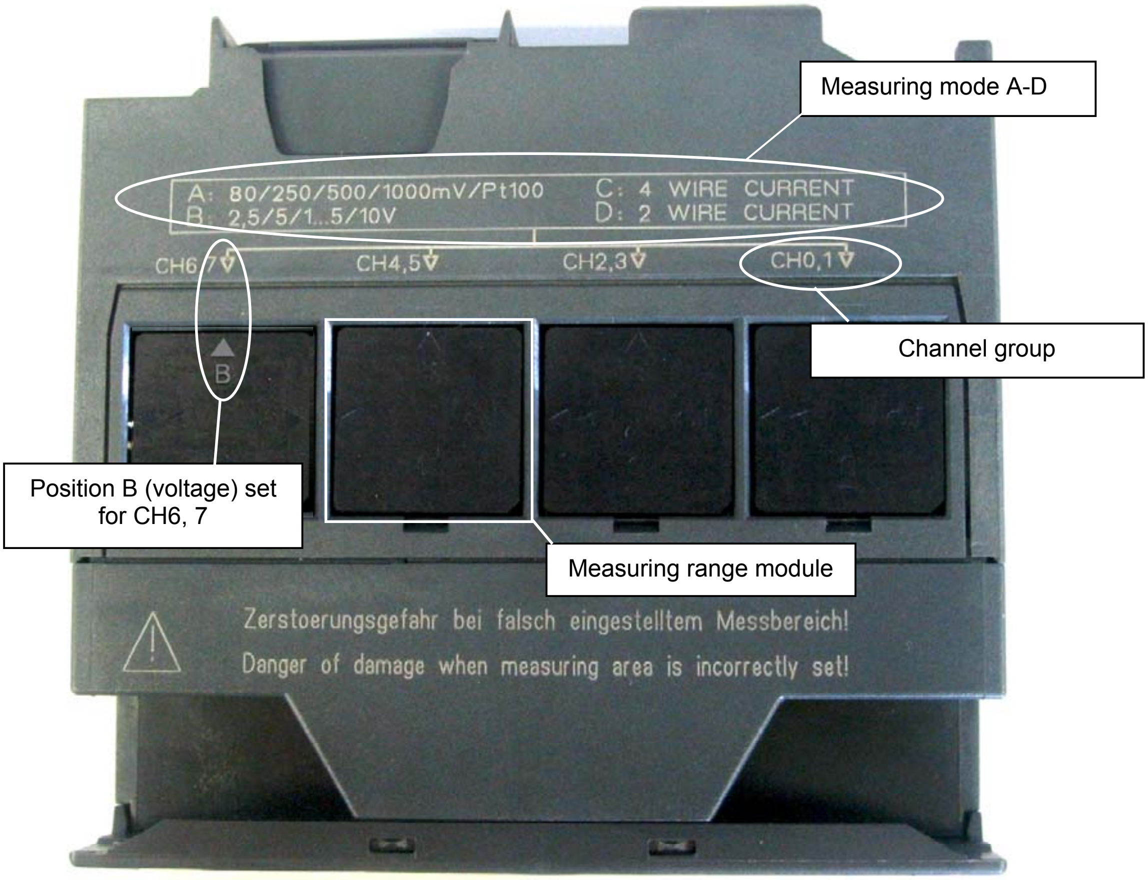


Figure 4-2 4 measuring range modules with default setting B (Voltage)

Table 4-3 Positions of the measuring range modules

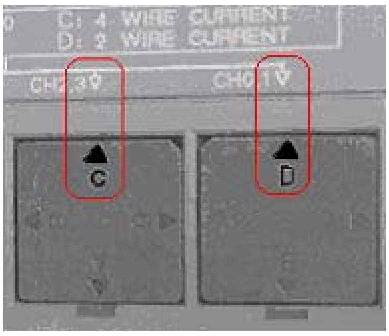
Position	Measurement type
A	Thermocouple / Resistance measurement
B	Voltage (default setting)
C	Current (2-wire transducer)
D	Current (4-wire transducer)

In our example, a sensor with a 4 to 20mA 2-wire transducer is connected to channel group 1 at input 0.

A 4-wire transducer is connected to channel group 2 at inputs 2 and 3.

Therefore, the first measuring range module should have Position D and the second should have Position C

Table 4-4 Positioning of the measuring range modules

Graphics	Description
	With a screwdriver, pull out the two measuring range modules
	Turn the measuring range module to the desired position:
	Plug the measuring range module back into the module In our example, the module should have the following positions: CH0,1: D CH2,3: C

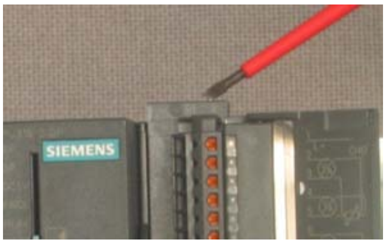
Note

When you use a 2-wire transducer, the electrical isolation against the load voltage is lost for all the channels in the module (at least one measuring range module is set to position D)

4.2.4 Mounting the SM331 module

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

Table 4-5 Mounting the SM331 module

Graphics	Description
	Mounting the SM331: • Insert the SM331 to the top part of the rail • Push it all the way left to the CPU • Push down • and tighten the screw at the bottom to the rail
	Mounting of the front connectors: • Press the upper release button of the front terminal block • Insert the front connector into the module until it snaps in

Mechanically the sample station is now completely mounted.

5 Electrical connection

This chapter shows you how the various parts of the sample station are electrically wired from the power supply to the analog module.

Warning

You might get an electrical shock if the power supply PS307 is turned on or the power cord is connected to the line.

Wire the S7-300 only in power-off state.

5.1 Wiring the power supply and the CPU

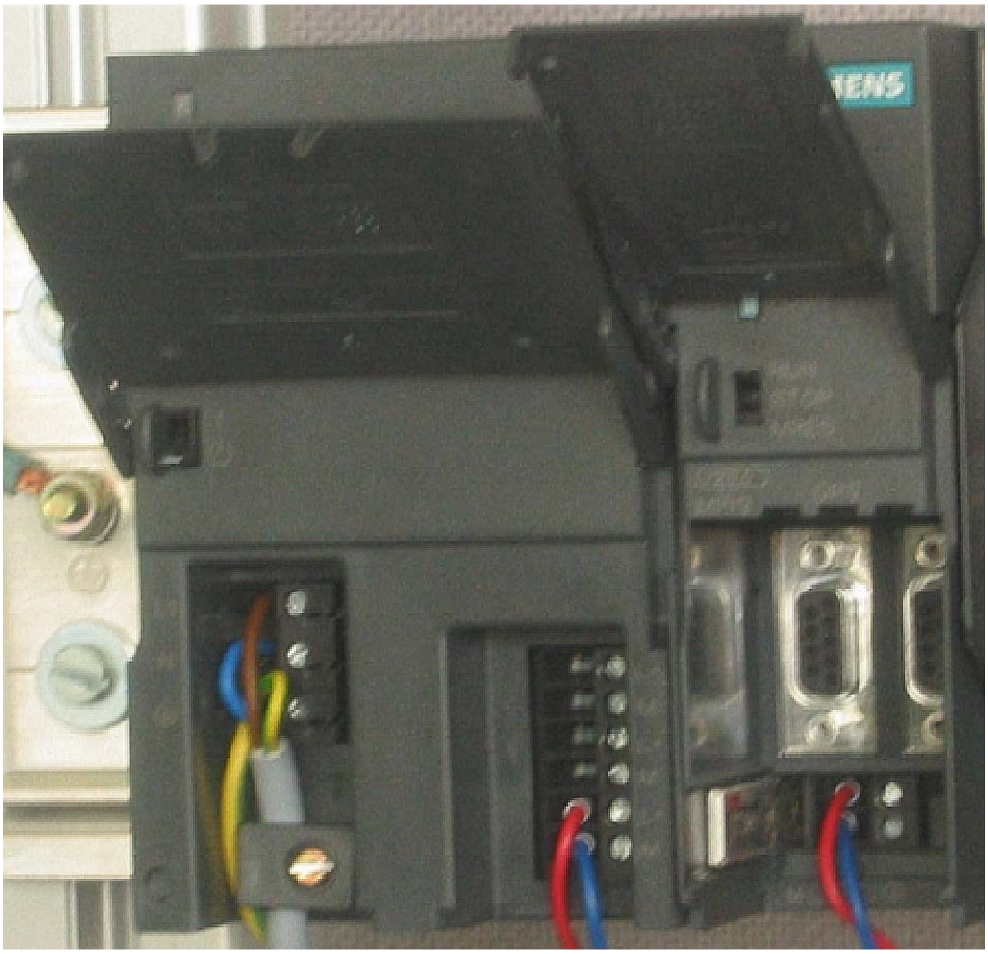
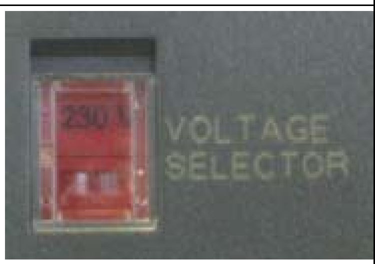


Figure 5-1 Wiring the power supply and the CPU

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

Table 5-1 Wiring the power supply and the CPU

Step	Graphics	Description
1		Open the front flaps of the power supply and the CPU
2		Unscrew the pull relief bracket on the power supply
3		Remove the insulation from the power cord, attach the cable end sleeves (for multi-wire cords) and connect it to the power supply
4		Tighten the pull relief bracket
5		Insert two connecting cables between the power supply and the CPU and tighten them
6		Confirm that the setting of the voltage selector is set to your local line voltage. The power supply's default setting is AC 230 V. If you have to change this setting, do the following: Remove the protective cap with a screwdriver, adjust the switch according to your line voltage and put the protective cap back.

5.2 Wiring the analog module

The wiring of an analog measurement transducer is dependant on its type and not on the SM331 module.

5.2.1 Current transducer wiring - Principle

Depending on the current transducer you use, you have to modify the wiring of the power supply. We differentiate between the wiring of a 2-wire current transducer and a 4-wire current transducer.

Wiring principles of a 2-wire current transducer

This transducer type is supplied with power from the analog input module.

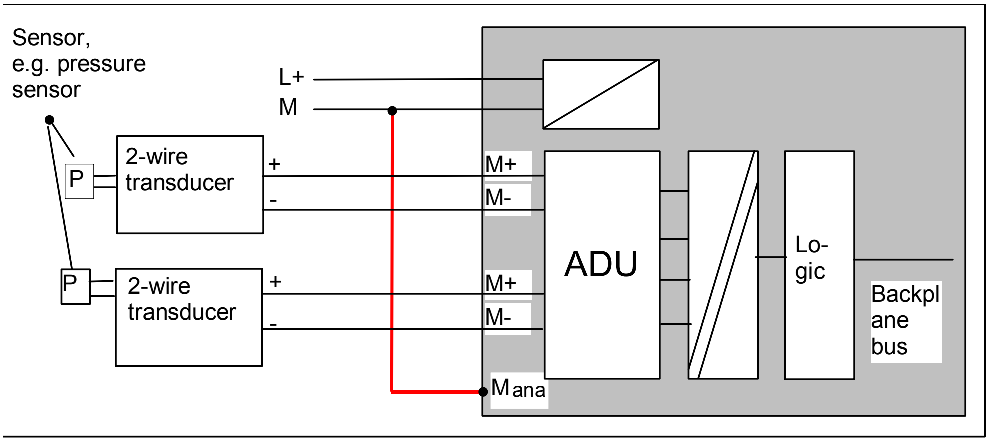


Figure 5-2 Wiring: 2-wire current transducer

Wiring principles of a 4-wire current transducer

Unlike a 2-wire transducer, this transducer has its own power supply.

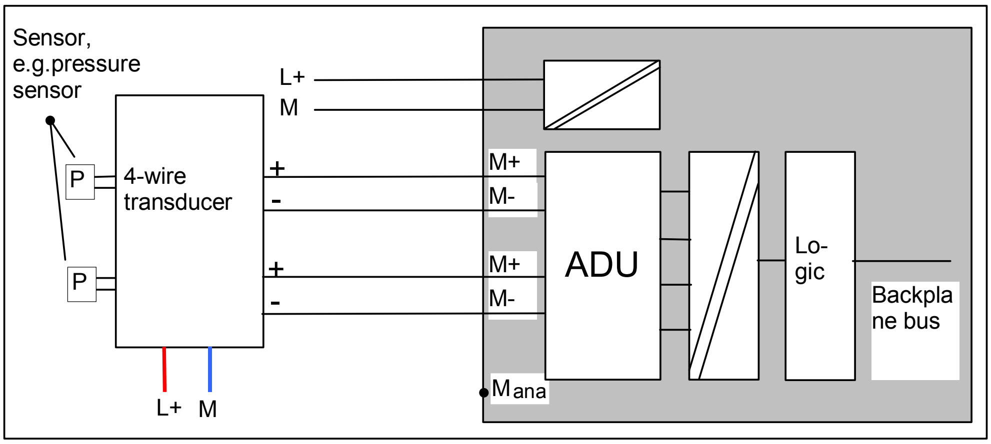


Figure 5-3 Wiring: 4-wire current transducer

5.2.2 Wiring of the analog module

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

- Connection of the power supply (Red cable)
- Connection of the 2-wire current transducer (Green cables)
- Terminate unused channels with a resistor
- Connection of the first 4-wire current transducer (Green cables)
- Connection of the second 4-wire current transducer (Green cables)
- Connection to zero potential and short-circuit the other unused channels (blue wires)

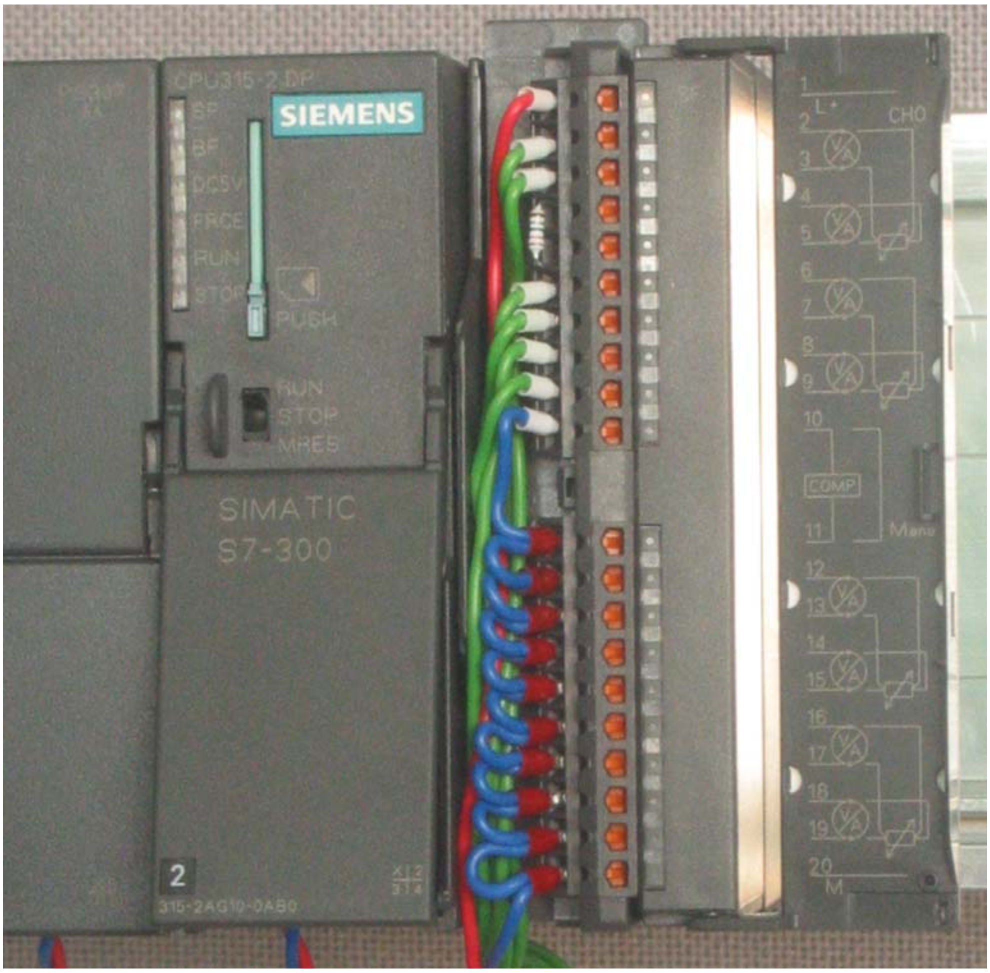


Figure 5-4 SM331 Front connector wiring

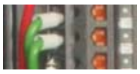
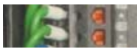
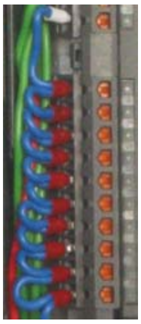
Warning

Possible destruction of the module!

If you connect a defective 4-wire current transducer to an input, which is configured for a 2-wire transducer, the module might get destroyed.

Step by step the tasks necessary for wiring are explained below:

Table 5-2 SM331 Front connector wiring

Graphics	Wiring	Comments
	Open the front flap of the SM331	The connection diagram is printed on the front flap
	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.	
	Wire the front connector as follows: Terminal 1: L+	Power supply of the module
	Terminal 2: M+ Sensor 1 Terminal 3: M- Sensor 1	Standard wiring for 2-wire current transducer
	Connect Terminal 4 and 5 with a 1.5 to 3.3 kΩ resistor	In order to maintain the diagnostic capability of channel group 0, the second unused input must be connected to a resistor
	Terminal 6: M+ Sensor 2 Terminal 7: M- Sensor 2	Standard wiring for 4-wire current transducer
	Terminal 8: M+ Sensor 3 Terminal 9: M- Sensor 3	
	Connect Terminal 10 (Comp) and terminal 11 (M_{ana}) with M Short-circuit terminals 12 to 19 and connect with M_{ana} Terminal 20: M	For measuring current Comp is not used Mandatory for 2-wire current transducers Unused channel groups should be short-circuited with M_{ana} in order to achieve a maximum interference resistance

5.2.3 Switch on now

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

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

Figure 5-5 Successful wiring, CPU in position STOP

If a red LED is lit, then there is an error in the wiring. Verify your wiring.

6 Configuration with SIMATIC Manager

In this chapter the following tasks are executed:

- Creating a new STEP7 project
- Parameterization of the hardware configuration

6.1 Create a new STEP7 Project

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

Start SIMATIC Manager by clicking the symbol „SIMATIC Manager“ on your windows desktop and create a new project with the STEP7 wizard „New Project“.

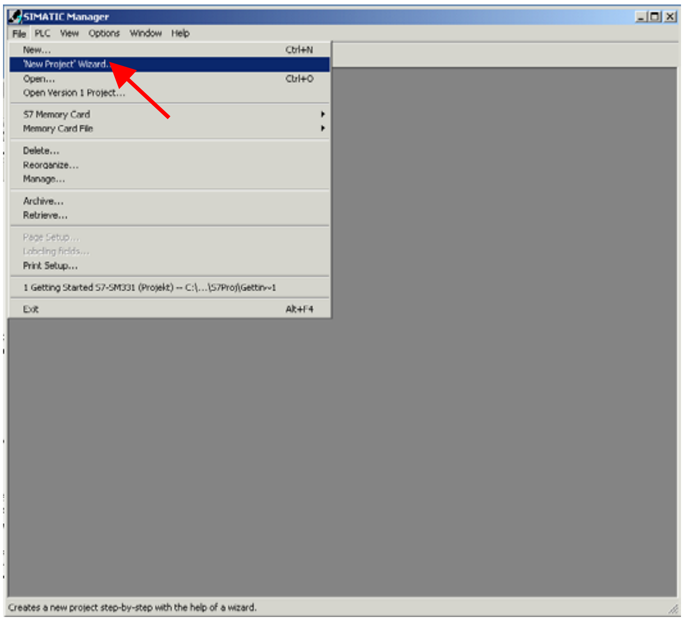


Figure 6-1 Execute STEP7 wizard „New Project“

An introduction window pops up. The wizard will guide you through the creation of a new project.

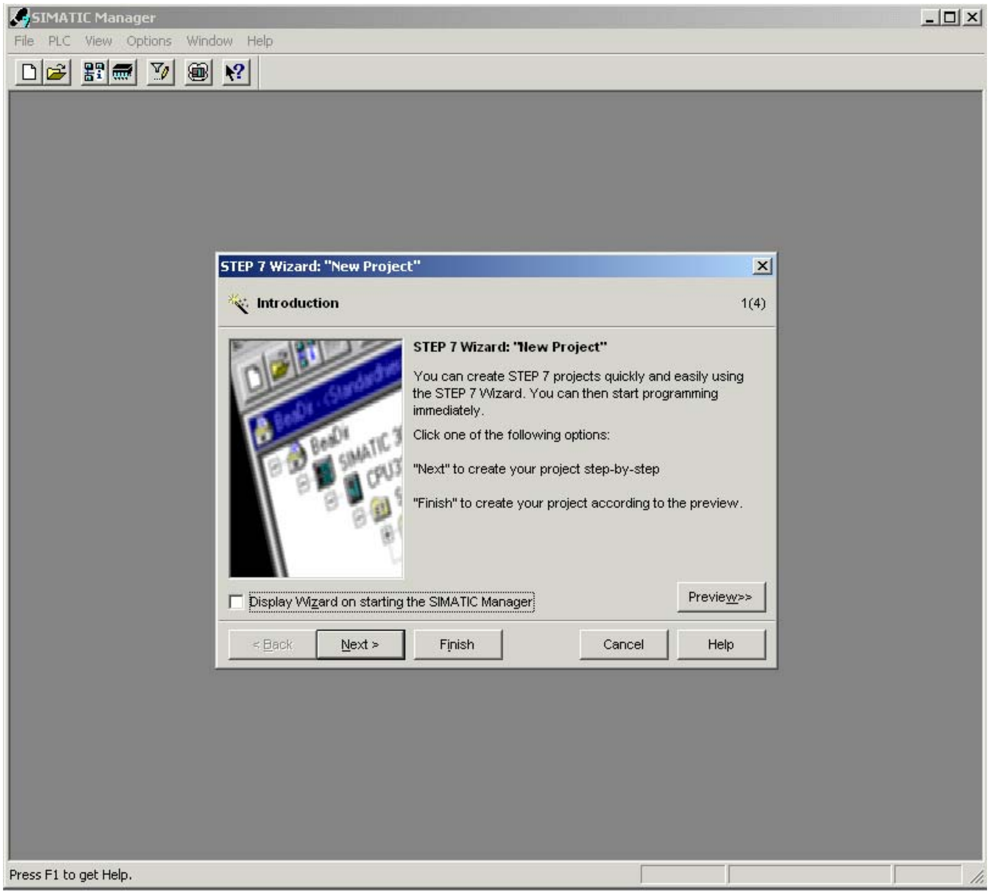


Figure 6-2 STEP7 wizard „New Project“, start

During the creation the following inputs are necessary:

- Selection of the CPU
- Define the basic user program
- Selection of organization blocks
- Project name

Click „Next“

6.1.1 CPU Selection

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

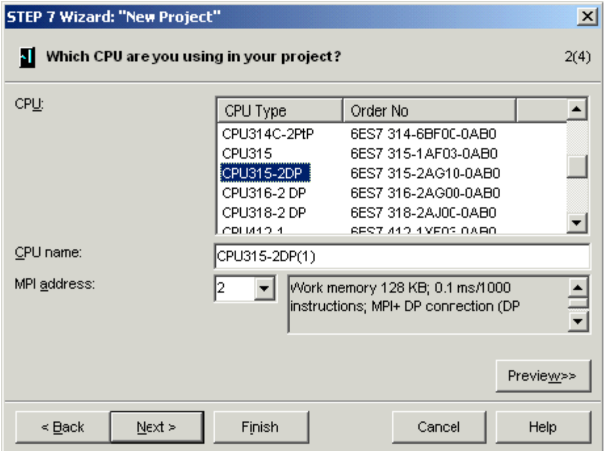


Figure 6-3 STEP7 wizard „New Project“, CPU selection

Click „Next“

6.1.2 Define the basic user program

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

- OB1 Program Cycle Organization Block
- OB40 Hardware interrupt
- OB82 Diagnostic interrupt

OB1 is required in every project and is called cyclically.
OB40 is called when a hardware interrupt occurs.
OB 82 is called when a diagnostic interrupt occurs.

In case you use a module with diagnostic capabilities and OB82 is not inserted, the CPU changes to STOP mode when a diagnostic alarm occurs.

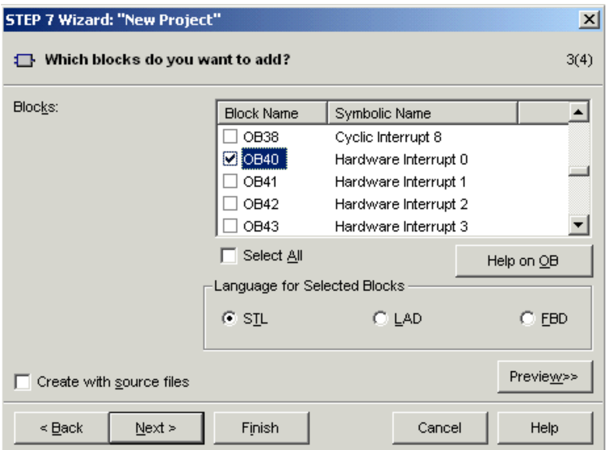


Figure 6-4 STEP7 wizard „New Project“: Insert organization blocks

Click „Next“

6.1.3 Specify the project name

Select the edit field “Project name” and overwrite the name in it with “Getting Started S7 SM331”

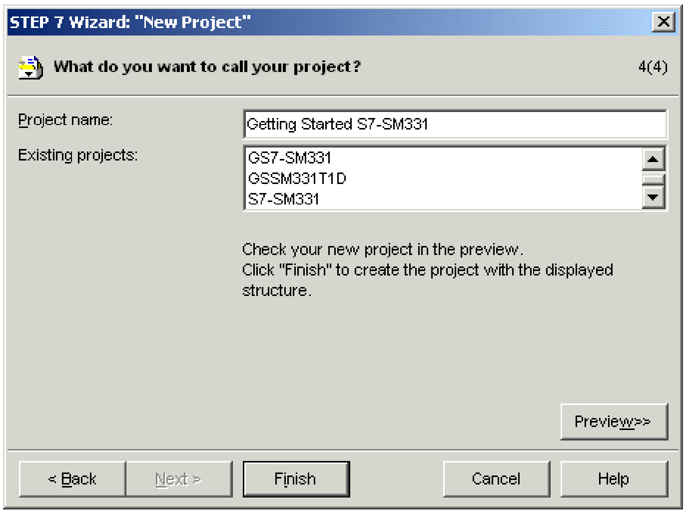


Figure 6-5 STEP7 wizard „New Project“: Specify project name

Click „Finish“. The basic STEP7 project is created automatically.

6.1.4 Resulting S7 project is created

The wizard has created the project “Getting Started S7-SM331”. In the right pane you can find the inserted organization blocks.

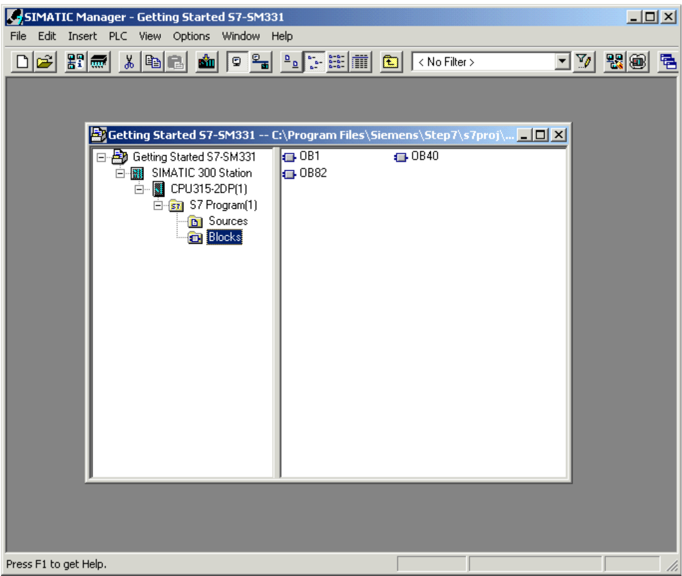


Figure 6-6 STEP7 wizard „New Project“: Result

6.2 Hardware configuration

The STEP7 wizard has created a basic S7 project. You also need a complete hardware configuration in order to create the system data for the CPU.

6.2.1 Create the hardware configuration

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

In order to do this, select the folder „SIMATIC 300 Station“ on the left hand pane. Start the hardware configuration by double clicking the folder “Hardware” on the right hand pane.

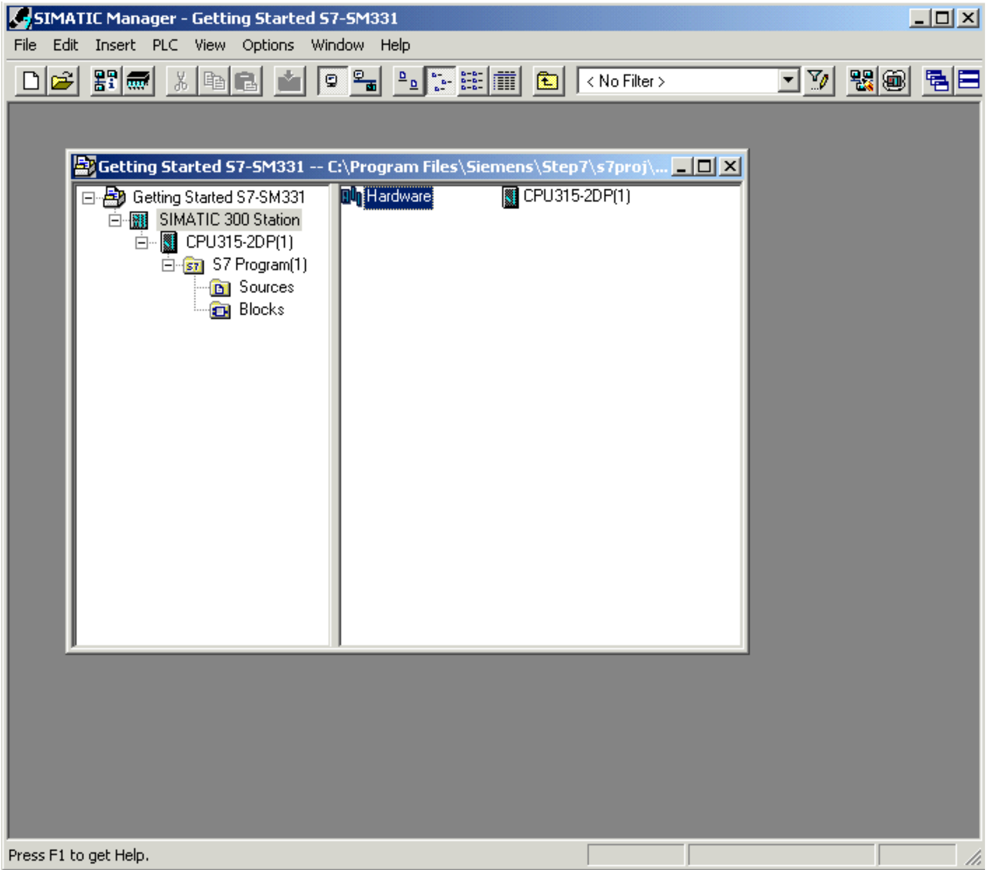


Figure 6-7 Starting the hardware configuration

6.2.2 Insert SIMATIC components

First select a power supply module from the hardware catalog.

If the hardware catalog is not visible, open it with the shortcut key Ctrl+K or by clicking the catalog symbol (blue arrow). In the hardware catalog you can browse through the folder SIMATIC 300 to the folder PS-300.

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

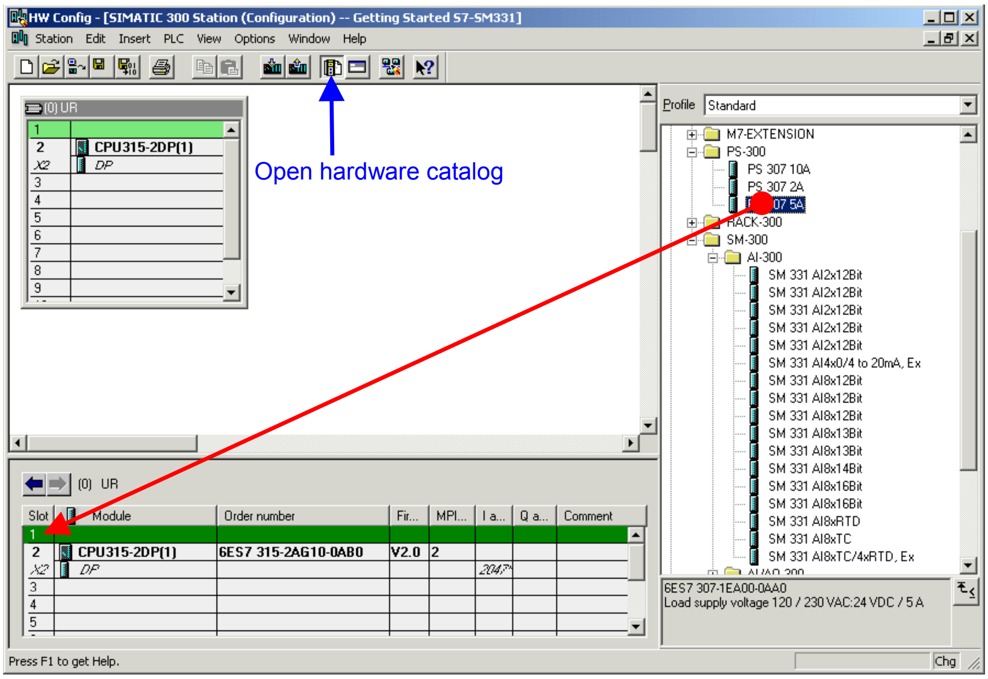


Figure 6-8 Hardware configuration: Basic configuration

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

Insert analog module

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

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

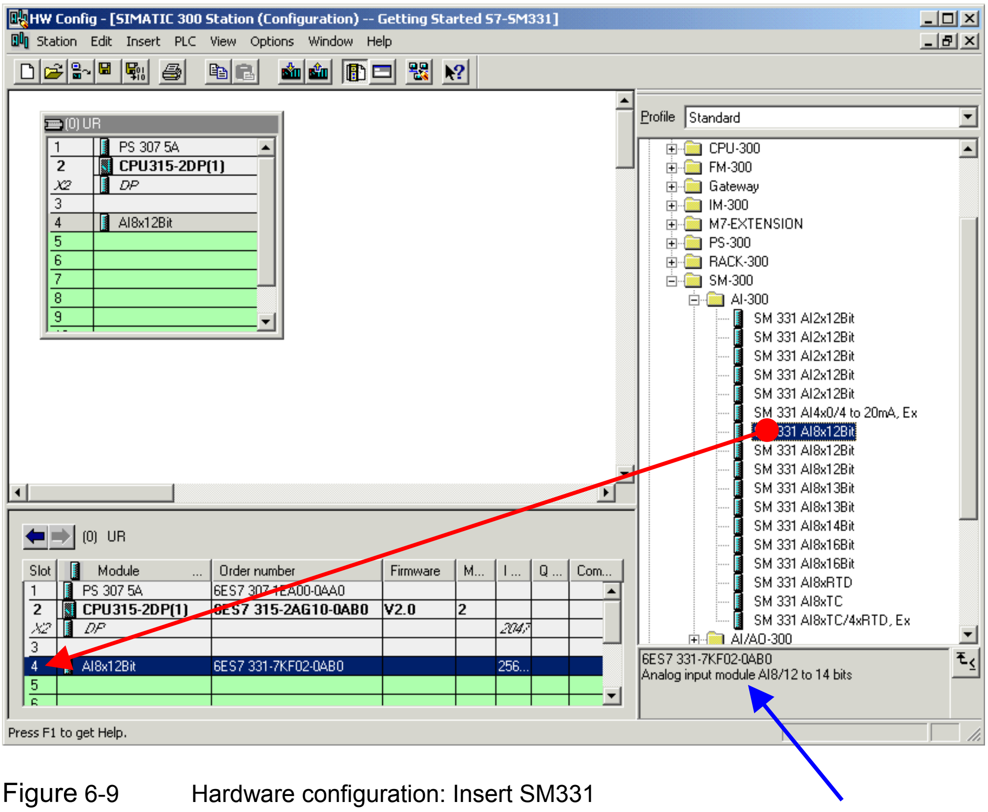


Figure 6-9 Hardware configuration: Insert SM331

Order number of the module

Drag the module into the first available field at slot 4 of your rack (see red arrow).

You have inserted all the modules into the hardware configuration. In the next step you parameterize the modules.

6.2.3 Parameterization of the analog module

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

Functionality of the sample station

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

Table 6-1 SM331 Functionality of the sample station

Functionalities	Description
Process reactions	• Diagnostic interrupt - active • Hardware interrupt when limit exceeded
Sensor 1	• 2-wire current transducer • Group diagnostics • Check for wire break • Measuring range 6 mA and 18 mA
Sensor 2 & 3	• 4-wire current transducer • Group diagnostics • Check for wire break • Measuring range 6 mA and 18 mA

Open the parameterization

Double click slot 4 that has the SM331 in it

Select the tab Inputs

Parameterize as follows:

- Diagnostic interrupt - checked
- Hardware interrupt when … - checked
- Input 0-1:
    - Measuring type: 2DMU
    - Group diagnostics - checked
    - With check for wire break - checked
- Input 2-3:
    - Measuring type: 2DMU
    - Group diagnostics - checked
    - With check for wire break – checked
    -
- Input 4-5 and 6-7
    - Measuring type: Deactivated (---)

- Interference frequency:
    - Select your power frequency (50 Hz or 60 Hz)
- Trigger for Hardware Interrupt
    - High limit 18 mA
    - Low Limit 6 mA

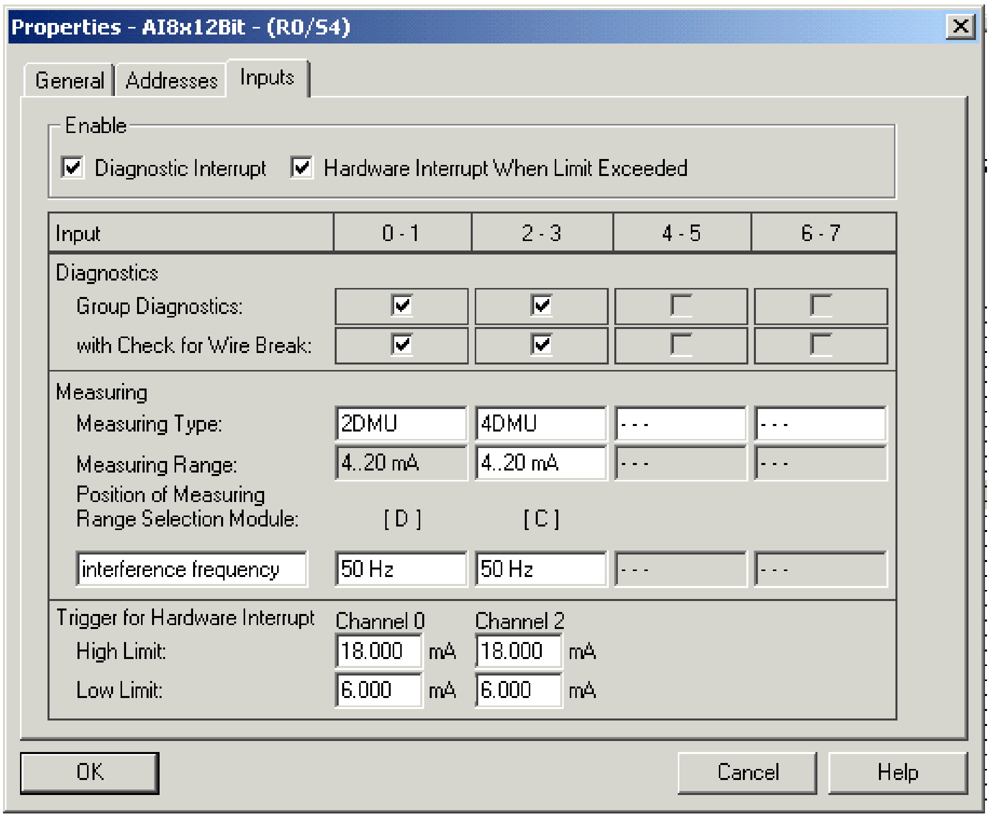


Figure 6-10 SM331: Parameterization

Explanation of the individual settings

Measuring type:

2DMU and 4DMU stand for 2-wire and 4-wire current transducers

--- means that the channels are deactivated. If you deactivate channels then the remaining channels are processed faster.

Measuring range modules

The required setting of the measuring range module (chapter 4.2.3) is displayed.

Interference frequency (Interference frequency suppression)

The frequency of the AC power supply network can interfere with the measurement values, especially in low voltage ranges and when thermocouples are used. With this parameter you specify the frequency of your power supply on site.

This parameter also influences the granularity, integration time and the basic execution period of the channel group.

Resolution (Accuracy)

The analog value is stored in a 16-bit word.

Integration time

The module requires a certain amount of time to measure the analog voltage. This time is called integration time. The higher the required accuracy is, the longer the module needs for measuring the voltage.

Basic execution period

Besides the integration time, the module also needs a certain amount of time to provide the measurement value.

Table 6-2 Relationship between accuracy, interference frequency and integration period

Resolution	Interference frequency	Integration time	Basic execution time
9 Bit	400 Hz	2,5 ms	24 ms
12 Bit	60 Hz	16,6 ms	136 ms
12 Bit	50 Hz	20 ms	176 ms
14 Bit	10 Hz	100 ms	816 ms

Hardware alarm

Only the channels 0 and 2 have hardware interrupt capabilities. You can use hardware interrupts to trigger an alarm when the analog signal exceeds its high or low limit.

Complete the hardware configuration

Close the parameter window.

Compile and save the project via Station → Save and Compile (Ctrl+S)

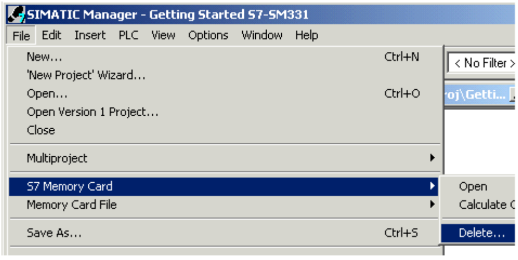
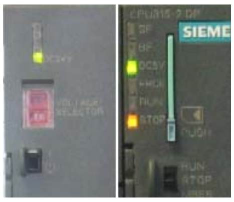
With this the hardware configuration of the project is completed.

6.2.4 Power up test

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

Power up

Table 6-3 Power up

#	Image	Description
1		Erase your Micro Memory Card with a Power PG or a PC with external programming device: In SIMATIC Manager click "File → S7 Memory Card → Delete …"
2		Turn off the CPU's power supply. Insert the MMC into the CPU. Turn on the power supply.
3		If the CPU is in RUN mode, set it to STOP mode.
4		Turn on the power supply again. If the STOP LED blinks, the CPU requests for a reset. Acknowledge this by turning the mode switch to MRES for a quick second.
5		Connect the CPU to the PG with an MPI cable. To do this, connect the MPI cable with the CPU's MPI port. Connect the other end to the PG interface of your programming device.

Download hardware configuration

Download the hardware configuration into the CPU with HW Config.

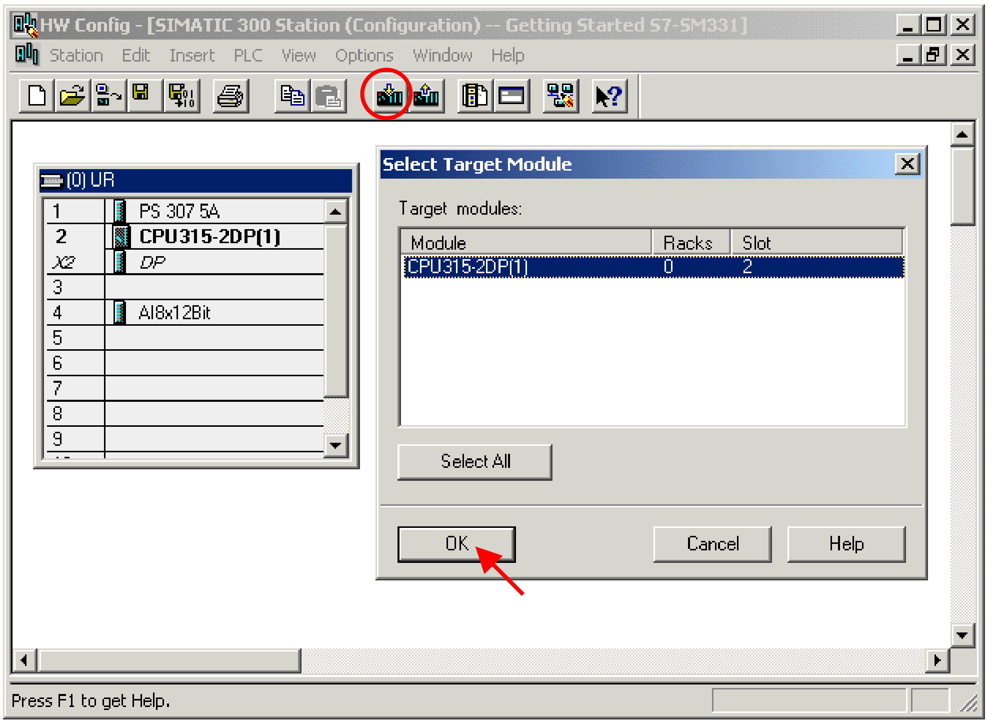


Figure 6-11 Download the CPU's hardware configuration (1)

Click the symbol „Load to module“ (shown in the red circle).

When the dialog window „Select target module“ pops up, click ok.

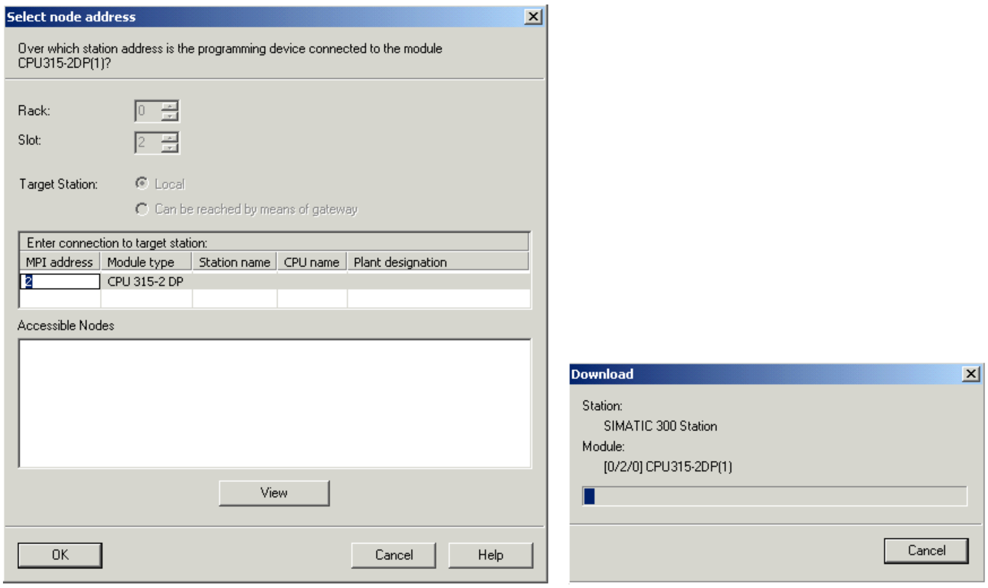


Figure 6-12 Download the CPU's hardware configuration (2)

The dialog window „Select target address” is shown. The system data are now transferred into the CPU.

Start CPU

Set the CPU to RUN mode.

If the hardware configuration was done correctly, two red LEDs (RUN and DC5V) should be lit at the CPU

Figure 6-13 CPU in error free state

6.3 STEP7 user program

6.3.1 Function of the user program

In our example the input values are stored in a data block. Also, the hardware interrupt status should be stored in a marker word. It should be possible to acknowledge the status information by means of a bit.

Furthermore the channel values (values of the input words) should be stored in another data block.

In the user program the following tasks have to be performed:

1. Cyclical storage of the analog input values in a data block (DB1)
2. Cyclical conversion of the analog input 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. Store the status in a marker word (MW100) when a hardware interrupt occurs.

Table 6-4 Structure of the user program

Execution mode	**Responsible Organisation block**	**Programming task**	**Used block or marker**
Cyclic execution	OB1	Store analog input values	DB1
		Convert and store the sensor signals	FC1, DB2
		Acknowledge hardware interrupt	M200.0
Execution triggered by hardware interrupt	OB40	Store status	MW100
Execution triggered by diagnostic interrupt	OB82	Has to be implemented because a module with diagnostic capabilities is used	---

About OB82

OB82 is used for modules with diagnostic capabilities. If the diagnostic alarm is enabled for such modules, OB82 requests for diagnoses when a failure is detected (coming and going events). As a reaction to this the operating system calls OB82.

In our example we use OB82 in order to prevent the CPU from changing to STOP mode. You can program the output on hardware interrupts.

6.3.2 Create user program

There are two ways to create a user program.

- If you know how to program STEP7 SCL, then you can create and program the necessary blocks and the function blocks in the Blocks folder of STEP7.
- You can insert the user program from an SCL source into the project. In this “Getting started” we describe this way.

Creating a user program in STEP7 requires three steps:

1. Download of the source file directly from the web page
2. Import source file
3. Compile source file

Download of the source file

You can download the source file directly from the web page from which you loaded this “Getting Started”

The German version of the source file has the name „GSSM331T1DE.AWL“.

Save the source file to your hard drive.

Import source file

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

- Right click the folder „Sources“
- Select „Insert new Object“ → External Source...

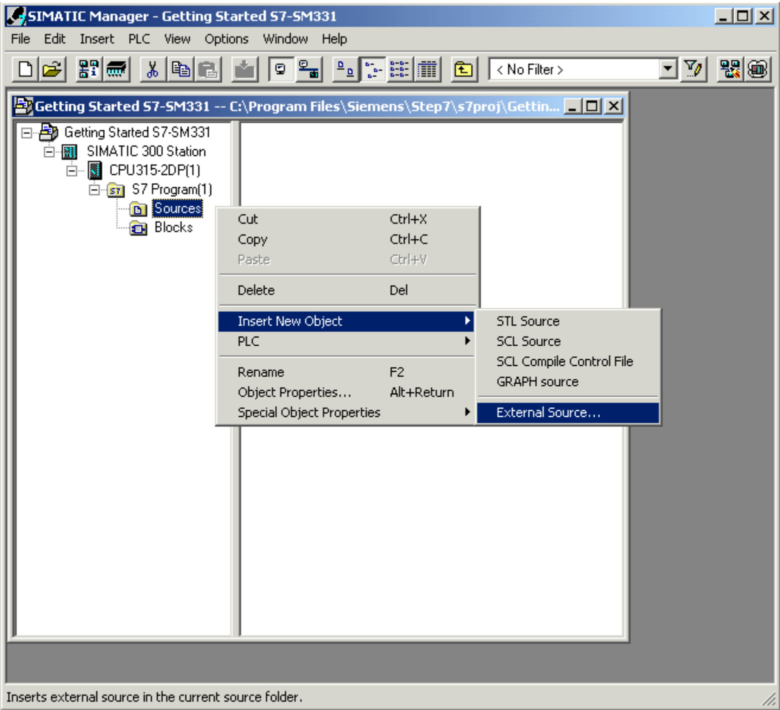


Figure 6-14 Import external source

In the dialog window „Insert external source“ browse for the source file GSSM331T1DE.AWL, which you have already downloaded and saved on your hard drive.

Select the source file GSSM331T1DE.AWL (red arrow).

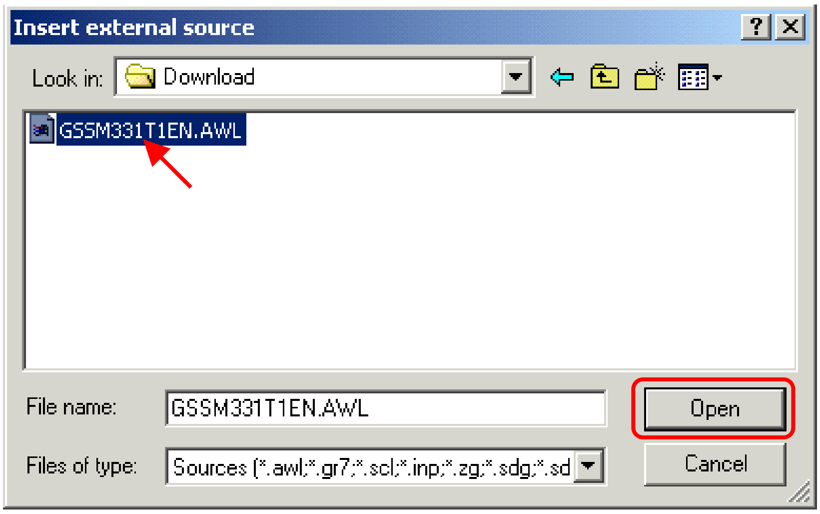


Figure 6-15 Import external source

Click „Open“.

SIMATIC Manager has opened the source file. On the right pane you can see the source file inserted.

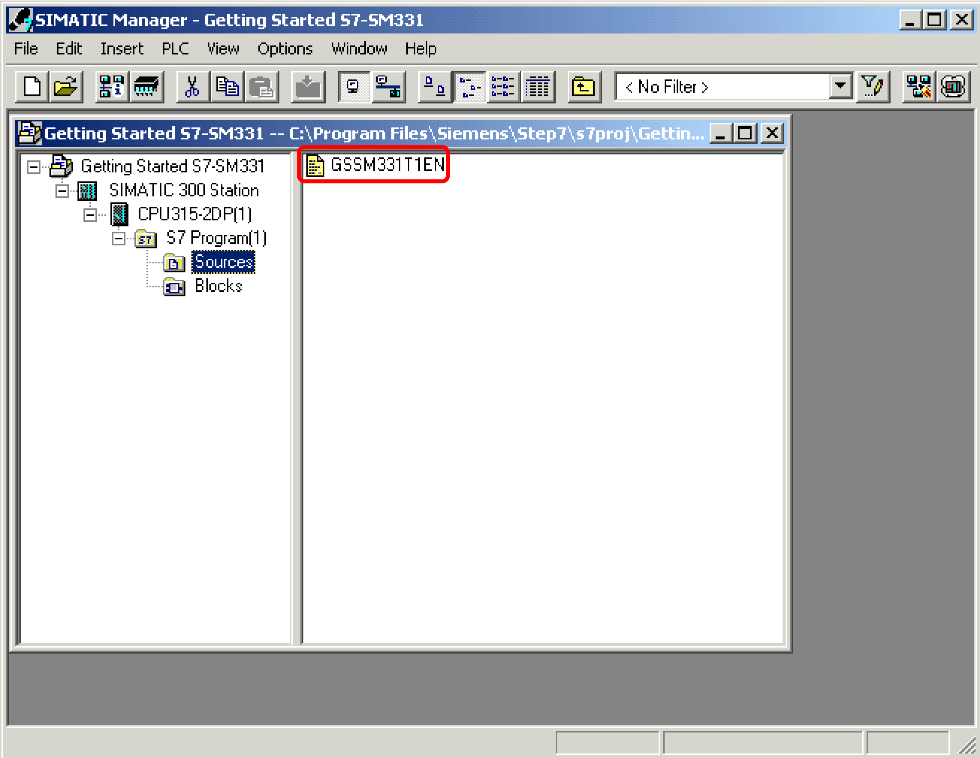


Figure 6-16 Storing the source file

Compile source code

In order to create an executable STEP7 program, the STL source has to be compiled.

Double click the source file GSSM331T1DE in the Sources folder. The source code editor is called.

In the window of the source code editor you can view the source code (code from Chapter 10).

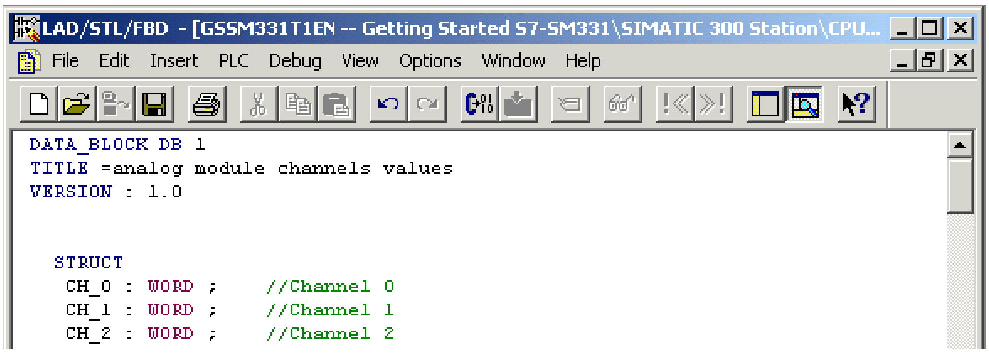


Figure 6-17 Source code editor

After the source code is loaded, start the compilation.

Press the shortcut key Ctrl+B or select File → Compile. The compilation starts immediately.

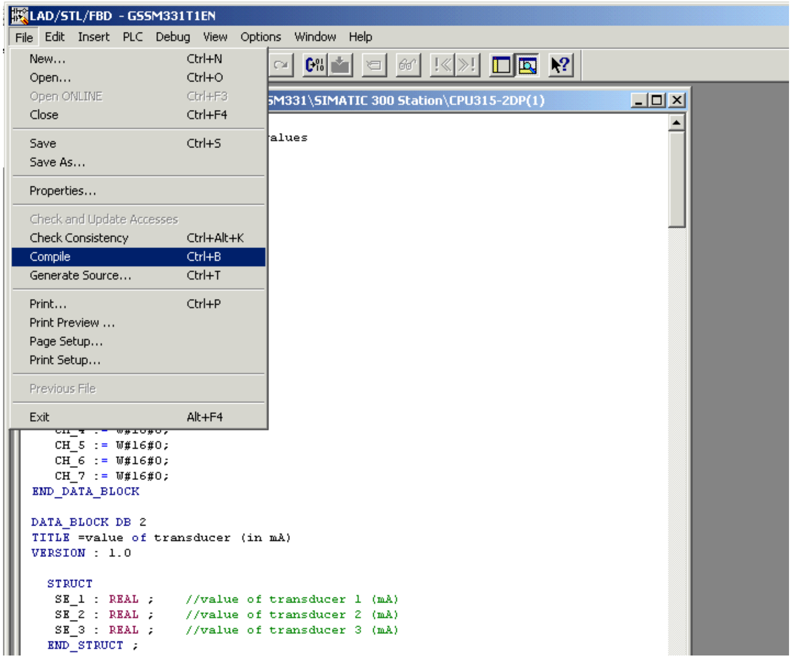


Figure 6-18 Translation of the STL source

In case of warning or error messages, check the source code.

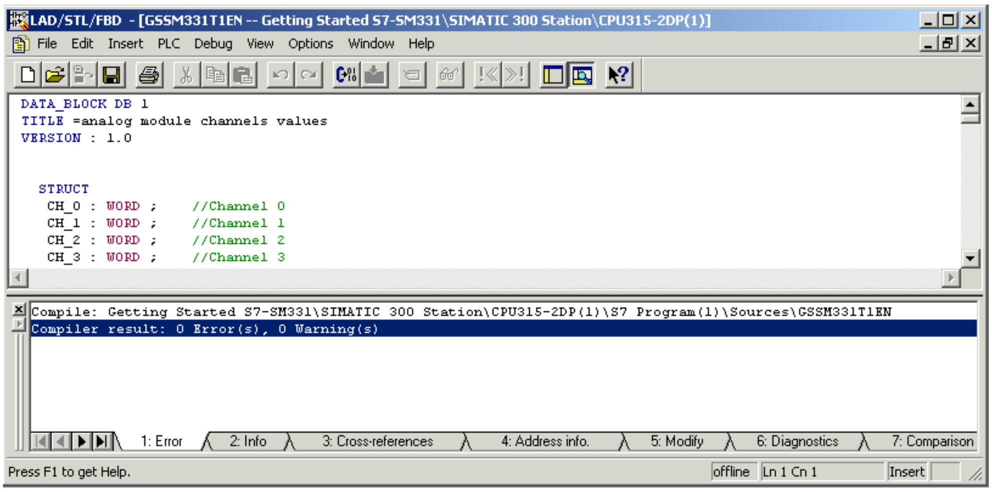


Figure 6-19 Source code editor, messages after compilation

Close the source code editor.

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

OB1, OB40, OB82, FC1, DB1 and DB2

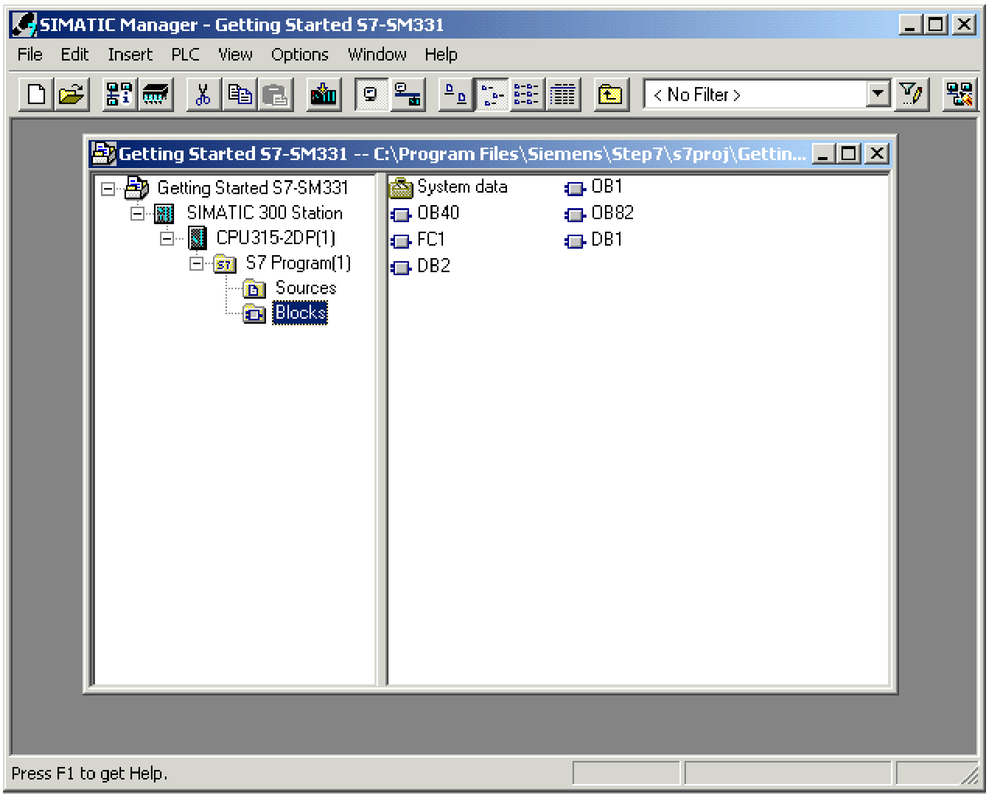


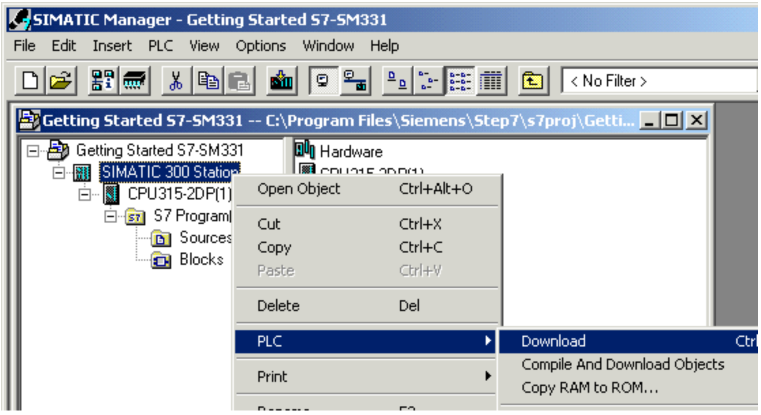
Figure 6-20 Generated blocks

7 Test the user program

7.1 Download system data and user program

Hardware and software are ready now. The next step is to download the system data and the user program into the automation system. To do this, execute the following steps:

Table 7-1 Download user program and system data

Step	Description
1	Download the user program and the system data (containing the hard-ware configuration) into the CPU. 
2	 Follow the instructions on the screen. If all sensors are properly con-nected, the CPU and the SM331 do not show any error light. The state of the CPU is dis-played by the green „RUN“ light.

S7-SmartLabel

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

The original size of the is displayed in Figure 7-1.

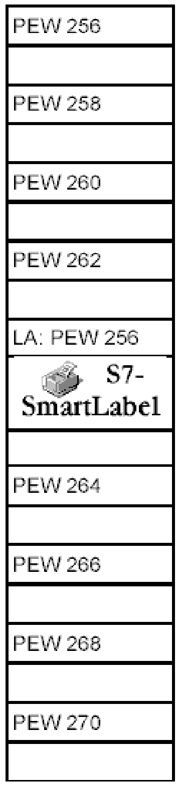


Figure 7-1 S7-SmartLabel labeling strip for the example

7.2 Visualization of the sensor signals

In order to visualize the sensor signals, 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

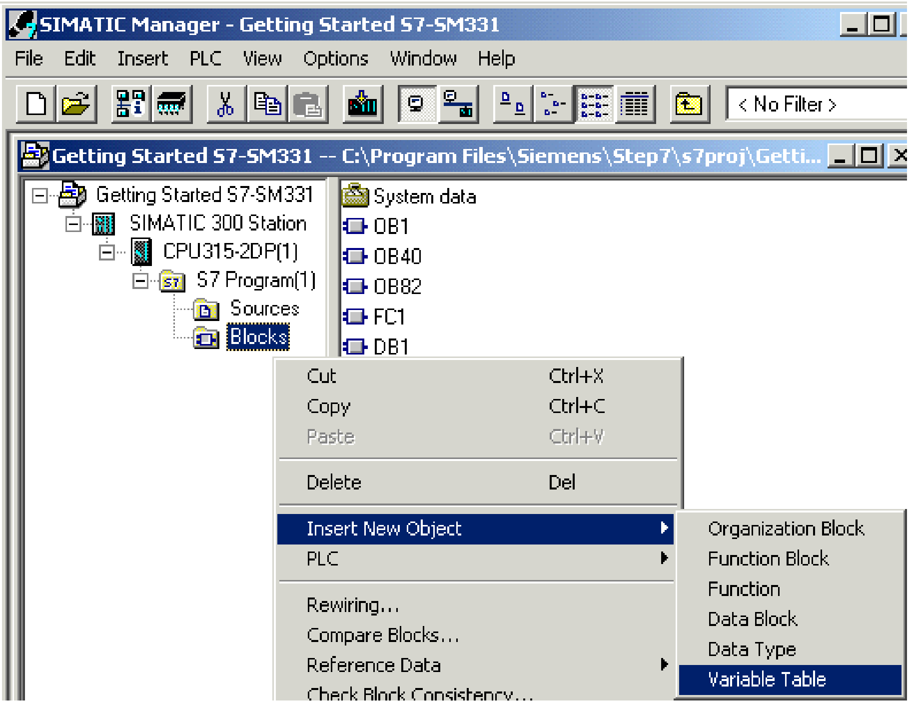


Figure 7-2 Insert Variable Table

Fill the new variable table as follows:

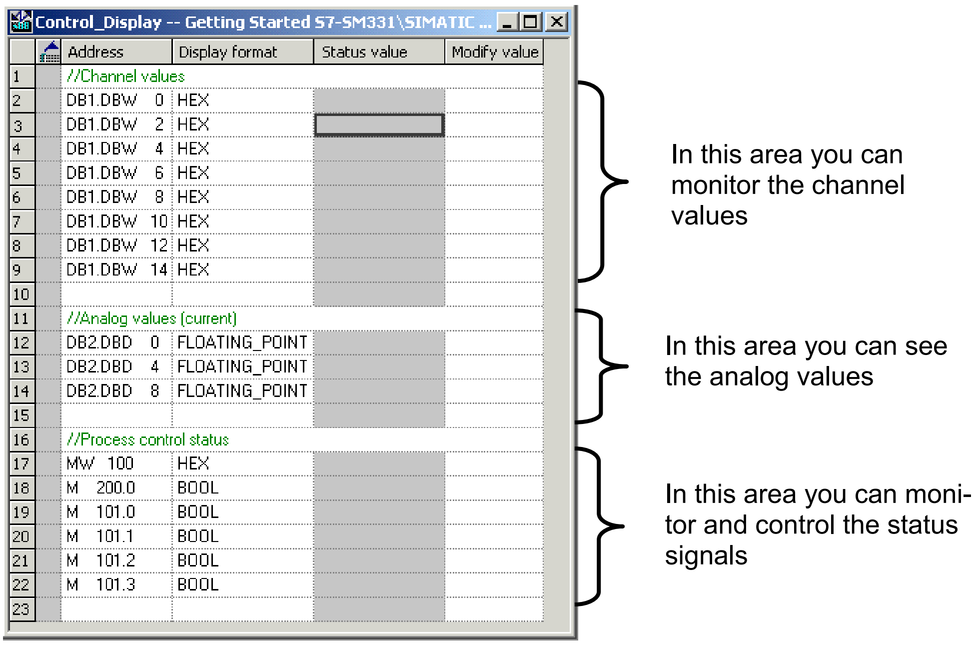


Figure 7-3 Variable table Control_Display

Table 7-2 Description of the variables

Variable	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	Transducer1 current (mA)
DB2.DBD 4	Transducer2 current (mA)
DB2.DBD 8	Transducer3 current (mA)
MW 100	Status hardware interrupt
MW 200.0	Acknowledge hardware interrupt
M101.0	Channel 0 exceeded low limit
M101.1	Channel 0 exceeded high limit
M101.2	Channel 2 exceeded low limit
M101.3	Channel 0 exceeded high limit

Monitoring of variables

In order to monitor variables, open the online view of the controller by clicking the Eye Glasses symbol. Now you can monitor the values in the data blocks and markers.

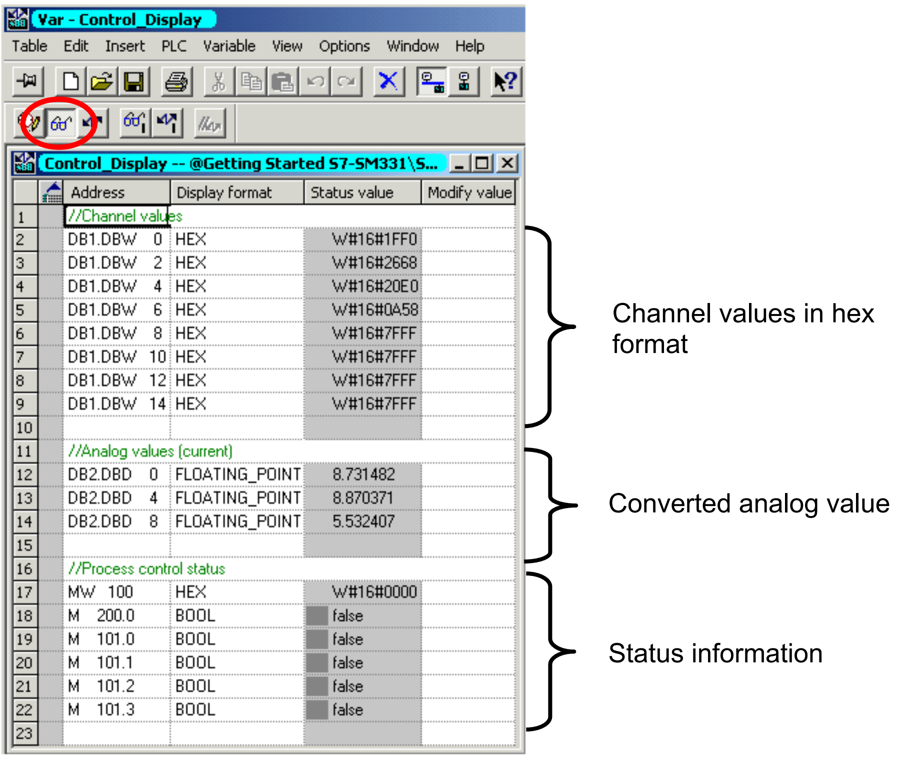


Figure 7-4 Online view of the variable table

Modification of variables

For modifying the Process Control Acknowledgement enter the desired value (TRUE or FALSE) into the column „Modify Value“. The value depends on whether you want to activate or deactivate the acknowledgement. Click the symbol with the two arrows.

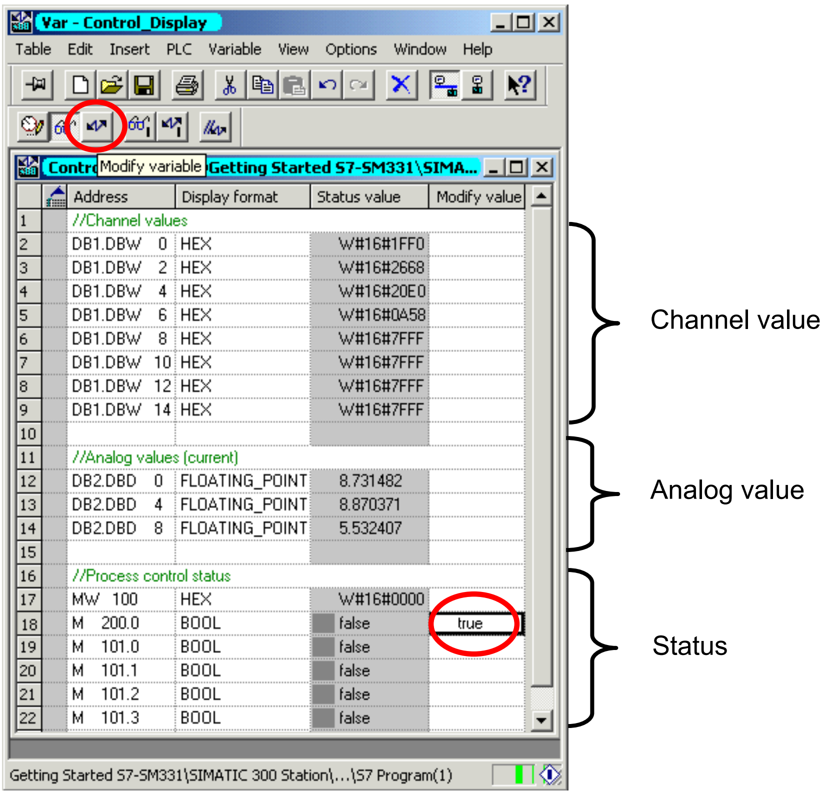


Figure 7-5 Modification of variables

Specifics for monitoring the variables

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

7.3 Display of analog values

The CPU can only process analog values in binary format. Analog input modules convert the analog process signal into a digital format (16 bit word).

Five ranges have to be taken into account when converting from digital to analog values:

Table 7-3 Display of analog value in the current range 4 to 20 mA

Hex value	Current range	Comment	Meaning
7FFF	22,96 mA	Overflow	From hex value 16#F700 on, the sensor value is above the configured measurement value range and is no more valid.
7F00			
7EFF	22,81 mA	Oversteering range	This range corresponds to a tolerance band before the overflow range is reached. Within this range the resolution is not optimal though.
6C01			
6C00	20 mA	Nominal range	The nominal range is the normal range for recording measurement values. This range guarantees optimal resolution.
5100	15 mA		
1	4 mA + 578,7 nA		
0	4 mA		
FFFF		Understeering range	Range according to the oversteering range but for low values.
ED00	1,1185 mA		
ECFF		Underflow	From hex value 16#ECFF on, the sensor signal is below the configured measurement value range and is no more valid.
8000			

It is necessary to convert the binary format of the values in order to display analog process values. In our example mA are displayed. This is done by converting the display of analog values in mA in a programmed function (FC1).

In our example we look at the values from the output of the transducer.

With the aid of a current meter you can now compare the values on the meter with the values of the analog values display. The values will be identical.

8 Diagnostic interrupt

Diagnostic interrupts enable the user program to react on hardware failures.

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

In OB82 you program the reaction on diagnostic interrupts.

8.1 Read diagnostic data from a PG

The analog input module SM331 AI8x12 has diagnostic capabilities.

Diagnostic interrupts that occur are signaled by the red „SF“ LED on the SM331 and on the CPU.

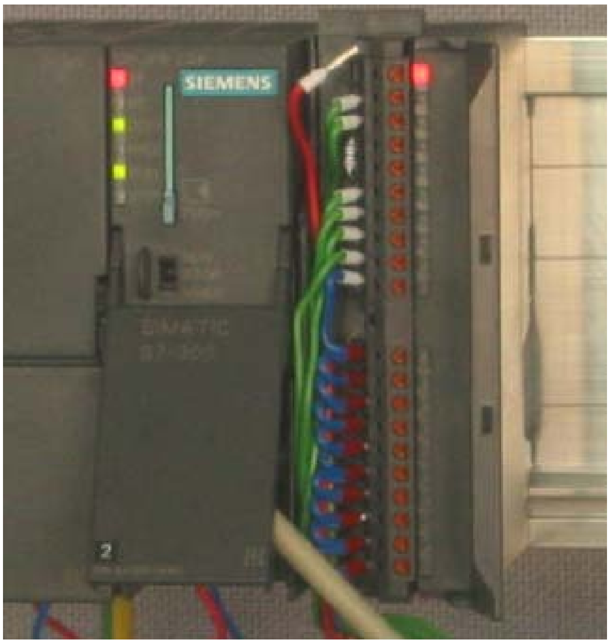


Figure 8-1 Hardware failure

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

In order to determine this, do the following:

Select the SM331 in the hardware configuration. Click the menu item CPU -→ Module Information... in order to perform a hardware diagnosis.

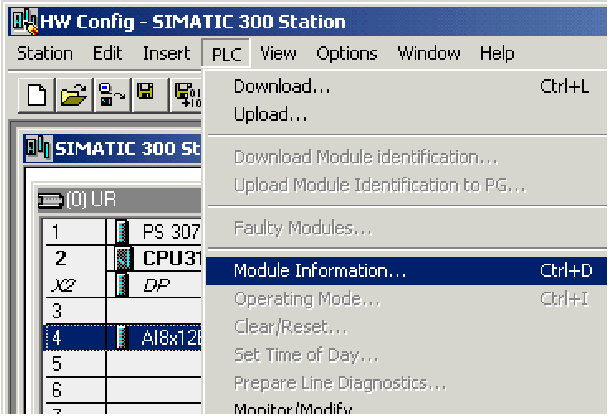


Figure 8-2 Module Information

8.2 General hardware interrupt

On the Hardware Interrupt tab you find information for the failure reported.

The interrupts are not channel dependent and apply to the entire module.

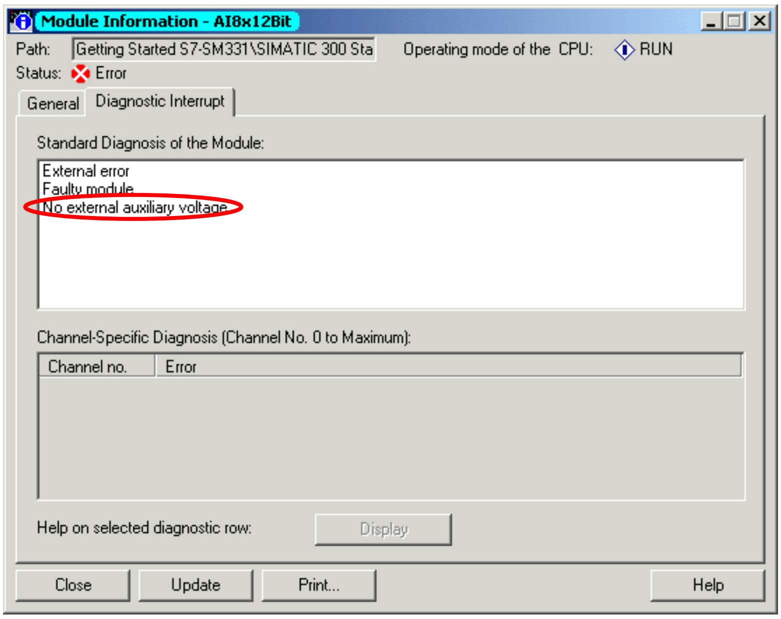


Figure 8-3 Diagnostics for SM331

8.3 Channel dependent diagnostic interrupts

There are five channel dependent diagnostic interrupts:

- Configuration or parameterization errors
- Common mode error
- Wire break
- Underflow
- Overflow

Note

Here we show you only the channel specific diagnosis for the measuring modes 2- or 4-wire current transducers. Other measuring modes are similar but not described here.

8.3.1 Configuration / parameterization errors

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

8.3.2 Common mode errors

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

In our example this failure cannot occur, because M_{ana} is connected to M for a 2-wire transducer (fixed potential).

8.3.3 Wire break

If wire break detection is activated for 2-wire transducers, there will be no direct check for a wire break. The diagnostics rather reacts on the shortfall of the low limit current value.

For a 4 to 20 mA current transducer a diagnostic message “Analog input wire break“ will be shown in the Module Information when the current goes below 3.6 mA.

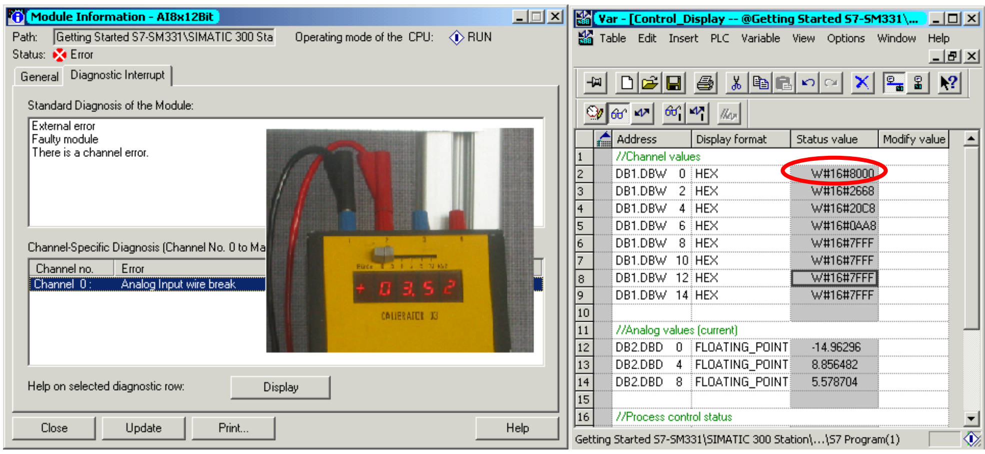


Figure 8-4 Left: Diagnostic message with wire break / Right: Variable table

The display of the analog values shows an underflow (Hex 8000) immediately even if the current measured is clearly above 1.1185 mA (see chapter 7.3).

Underflowing 3.6 mA is only possible if wire break detection was deactivated.

8.3.4 Underflow

The underflow notification is only triggered if the wire break detection is deactivated and the current is below 1.185 mA.

8.3.5 Overflow

If the current exceeds 22.81 mA, an overflow message stating „Analog input measuring range / High limit exceeded“ is displayed.

The display of the analog value (HEX 7FFF) is in the overflow range.

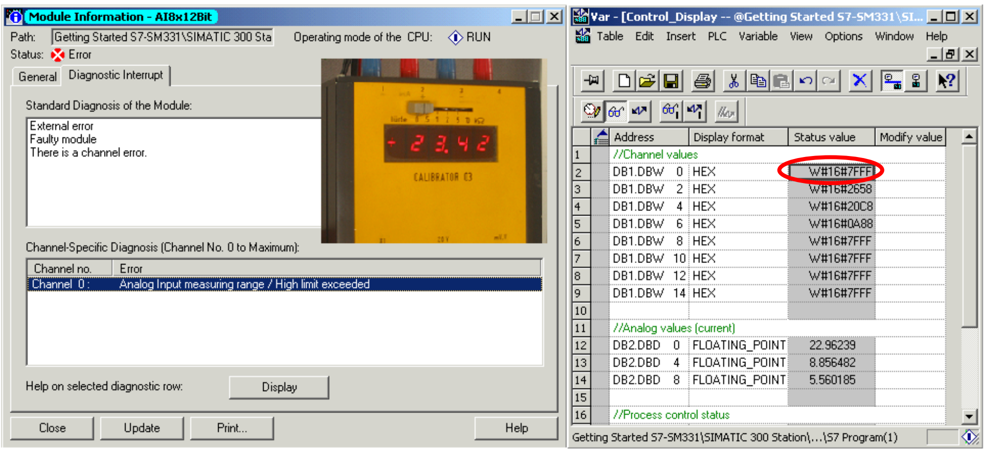


Figure 8-5 Left: Diagnostic message with overflow / Right: Variable table

---

Note

Disabled channels also have 7FFF hex as the analog display value.

---

9 Hardware interrupt

A special feature of the SM331 AI8x12bit is its capability to trigger hardware interrupts. Two channels (0 and 2) can be configured that way.

Hardware interrupts generally trigger alarm organization blocks. In our example OB40 is called.

The limit values for hardware interrupts have to be specified in mA.

Example:
You have connected a pressure sensor with a 4-20mA transducer to channel 0. Here the limit values should be specified in mA and not in Pascal (Pa).

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

Example:
If wire break detection (3.6 mA) is enabled, and you choose 3.5 mA for the low limit value, this setting is accepted by the system. A hardware interrupt will not be triggered, because the diagnostic alarm is always triggered first.

In our example, 2 channels (sensor 1 and 2) are configured with the following limits:

- Low limit value: 6 mA
- High limit value: 18 mA

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

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

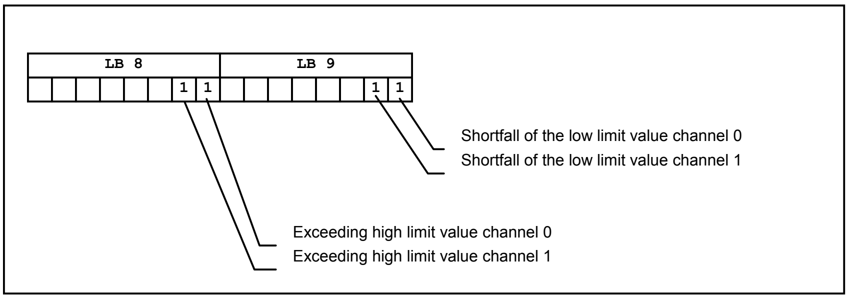


Figure 9-1 Startup information of OB40: Which event has triggered the hardware interrupt for which limit value

In the example OB40 only transfers LD8 and LD9 into a marker word (MW100). The marker word is monitored in the existing variable table. You can quit the marker word in OB1 by setting marker bit M200.0 or by setting it to TRUE in the variable table.

If you supply 5.71 mA with a calibration device to channel 0, you will get the value 0001 hex for MW100 in the variable table. That means that OB40 has been called and channel 0 exceeded its low limit value (6 mA).

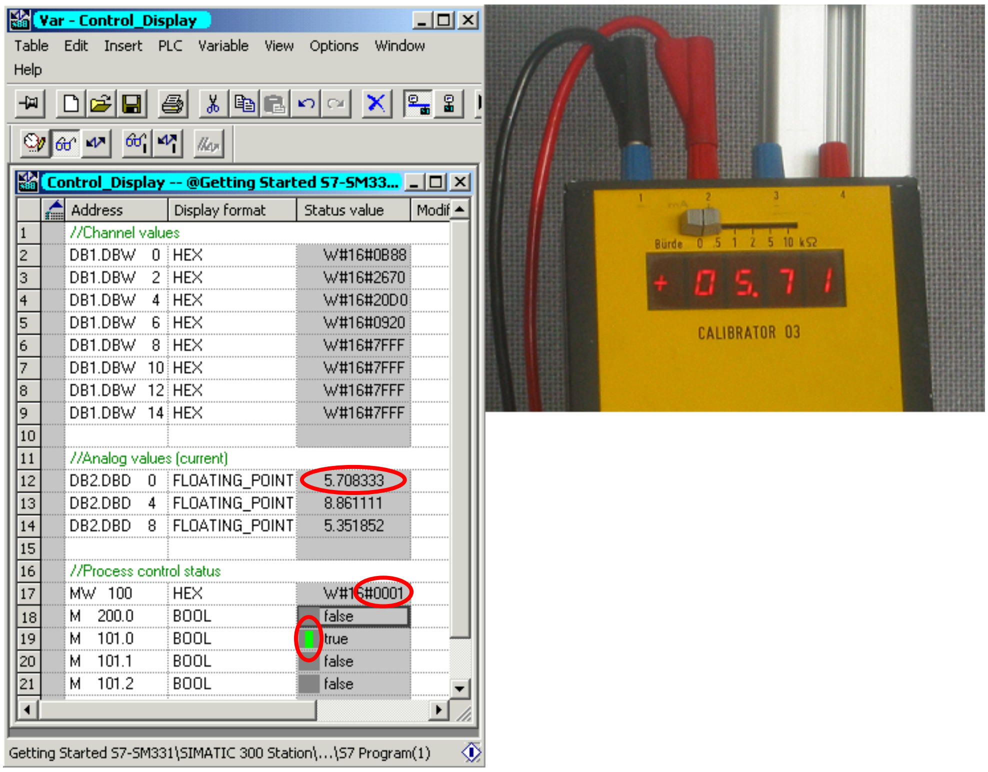


Figure 9-2 Hardware interrupt: Channel 0 exceeded low limit value

10 Source code of the user program

In this chapter you find the source code of the user program from the example.

You can also download the source as an STL file directly from the HTML page from which you have loaded this “Getting started” (see chapter 6.3.2).

STL source code

```
DATA_BLOCK DB 1
TITLE =Analog module channel values
VERSION : 1.0



  STRUCT
   CH_0 : WORD ;    // Channel 0
   CH_1 : WORD ;    // Channel 1
   CH_2 : WORD ;    // Channel 2
   CH_3 : WORD ;    // Channel 3
   CH_4 : WORD ;    // Channel 4
   CH_5 : WORD ;    // Channel 5
   CH_6 : WORD ;    // Channel 6
   CH_7 : WORD ;    // Channel 7
  END_STRUCT ;
BEGIN
   CH_0 := W#16#0;
   CH_1 := W#16#0;
   CH_2 := W#16#0;
   CH_3 := W#16#0;
   CH_4 := W#16#0;
   CH_5 := W#16#0;
   CH_6 := W#16#0;
   CH_7 := W#16#0;
END_DATA_BLOCK

DATA_BLOCK DB 2
TITLE =Transducer value (in mA)
VERSION : 1.0

  STRUCT
   SE_1 : REAL ;    // Sensor 1 current value (mA)
   SE_2 : REAL ;    // Sensor 2 current value (mA)
   SE_3 : REAL ;    // Sensor 3 current value (mA)
  END_STRUCT ;
BEGIN
   SE_1 := 0.000000e+000;
   SE_2 := 0.000000e+000;
```

```
   SE_3 := 0.000000e+000;
END_DATA_BLOCK

FUNCTION FC 1 : VOID
TITLE = Conversion of a channel’s raw values
VERSION : 1.0


VAR_INPUT
  Raw : WORD ;  // Analog value display
END_VAR
VAR_OUTPUT
  Current : REAL ;  // Current in mA
END_VAR
VAR_TEMP
  TDoubleInt : DINT ;
  TInt : INT ;
END_VAR
BEGIN
NETWORK
TITLE = Conversion of raw values in mA


      L     #Raw;
      T     #TInt;
// Only long integers can be converted into REAL format
      L     #TInt;
      ITD   ;
      T     #TDoubleInt;

      L     #TDoubleInt; //           HEX value
      DTR   ;            //  Current = --------------------
      T     #Current;    //            1728

      L     1.728000e+003; //   !        /
      /R    ;              //   !       /
      T     #Current;      //   !      /
                           //   +-----/-----------+----
                              //      4           20
      L     4.000000e+000;  // Offset adjustment
      +R    ;
      T     #Current;


END_FUNCTION
```

```
ORGANIZATION_BLOCK OB 1
TITLE = "Main Program Sweep (Cycle)"
VERSION : 1.0


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 ; //Maximum cycle time of OB1 (milliseconds)
  OB1_DATE_TIME : DATE_AND_TIME ;   //Date and time OB1 started
END_VAR
BEGIN
NETWORK
TITLE = Read channels
// Channel values 0 to 7 are loaded and stored in DB1 (channel values)
      L     PEW  256;      // Channel 0
      T     DB1.DBW    0;

      L     PEW  258;      // Channel 1
      T     DB1.DBW    2;

      L     PEW  260;      // Channel 2
      T     DB1.DBW    4;

      L     PEW  262;      // Channel 3
      T     DB1.DBW    6;

      L     PEW  264;      // Channel 4
      T     DB1.DBW    8;

      L     PEW  266;      // Channel 5
      T     DB1.DBW   10;

      L     PEW  268;      // Channel 6
      T     DB1.DBW   12;

      L     PEW  270;      // Channel 7
      T     DB1.DBW   14;
```

```
NETWORK
TITLE = Conversion
// Conversion of the channel’s raw data into current values (mA)
      CALL FC     1 (
           Raw                       := DB1.DBW    0,
           Current                   := DB2.DBD    0);

      CALL FC     1 (
           Raw                       := DB1.DBW    4,
           Current                   := DB2.DBD    4);

      CALL FC     1 (
           Raw                       := DB1.DBW    6,
           Current                   := DB2.DBD    8);



NETWORK
TITLE = Reset hardware interrupt
// Even though the hardware interrupt was reset by the hardware upon termi-
nating OB40,
// the value of the hardware interrupt must be reset manually


      U     M    200.0;
      SPBN  lbl0;
      L     MW   100;
      SSI   4;
      T     MW   100;
lbl0: NOP   0;
NETWORK
TITLE = The End

      BE    ;

END_ORGANIZATION_BLOCK

ORGANIZATION_BLOCK OB 40
TITLE = "Hardware Interrupt"
// Processing OB40_POINT_ADDR (L8 to L11)
//
//L8 High limit value exceeded
//L9 Low limit value exceeded
VERSION : 1.0

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 = Sensor 1 (Channel 0): Low limit

      U     L      9.0; // Channel 0 low limit
      SPBNB L001;
      L     W#16#1;
      L     MW   100;
      OW    ;
      T     MW   100;
L001: NOP   0;
NETWORK
TITLE = Sensor 1 (Channel 0): High limit

      U     L      8.0; // Channel 0 high limit
      SPBNB L002;
      L     W#16#2;
      L     MW   100;
      OW    ;
      T     MW   100;
L002: NOP   0;


NETWORK
TITLE = Sensor 2 (Channel 2): Low limit

      U     L      9.2; // Channel 2 low limit
      SPBNB L003;
      L     W#16#4;
      L     MW   100;
      OW    ;
      T     MW   100;
L003: NOP   0;

NETWORK
TITLE = Sensor 2 (Channel 2): High limit
```

```
      U     L      8.2; // Channel 2 high limit
      SPBNB L004;
      L     W#16#8;
      L     MW   100;
      OW    ;
      T     MW   100;
L004: NOP   0;
```

```
NETWORK
TITLE = Sensor 3 (Channel 3): Low limit
// Only for demonstration purposes; Channel 3 has now hardware interrupt ca-
pabilities.
      U     L      9.3; // Channel 3 low limit
      SPBNB L005;
      L     W#16#10;
      L     MW   100;
      OW    ;
      T     MW   100;
L005: NOP   0;
```

```
NETWORK
TITLE = Sensor 3 (Channel 3): High limit
//Only for demonstration purposes; Channel 3 has now hardware interrupt ca-
pabilities.
      U     L      8.3; // Channel 3 high limit
      SPBNB L006;
      L     W#16#20;
      L     MW   100;
      OW    ;
      T     MW   100;
L006: NOP   0;
```

```
END_ORGANIZATION_BLOCK
```