SIEMENS

SIMATIC

S7-1200

Update to the S7-1200 System Manual, edition 01/2015

Product Information

Overview to Documentation Update S7-1200

In spite of efforts to ensure the accuracy and clarity in the product documentation, some of the pages in the *S7-1200 Programmable Controller System Manual* contain information that has been identified as being incomplete, incorrect or misleading.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, solutions, machines, equipment and/or networks. They are important components in a holistic industrial security concept. With this in mind, Siemens' products and solutions undergo continuous development. Siemens recommends strongly that you regularly check for product updates.

For the secure operation of Siemens products and solutions, it is necessary to take suitable preventive action (e.g. cell protection concept) and integrate each component into a holistic, state-of-the-art industrial security concept. Third-party products that may be in use should also be considered. You can find more information about industrial security on the Internet (http://www.siemens.com/industrialsecurity).

To stay informed about product updates as they occur, sign up for a product-specific newsletter. You can find more information on the Internet (http://support.automation.siemens.com).

This document contains the following updates

Setting the maximum record length for some Profibus communication instructions

When using a CM1243-5 Profibus Master module to control an ET200SP or ET200MP Profibus device that uses a RS232, RS422, or RS485 point-to-point module, you need to explicitly set the "max_record_len" data block tag to 240 for some communication instructions as defined below:

Instruction	Limitation
Send_P2P	Set "max_record_len" in the instance DB (for example, "Send_P2P_DB".max_record_len) to 240 after running any configuration instruction such as Port_Config, Send_Config, or Receive_Config.
Modbus_Master	Set "max_record_len" in the Send_P2P section of the instance DB (for example, "Modbus_Master_DB".Send_P2P.max_record_len to 240 after running Modbus_Comm_Load.
Modbus_Slave	Set "max_record_len" in the Send_P2P section of the instance DB (for example, "Modbus_Slave_DB".Send_P2P.max_record_len to 240 after running Modbus_Comm_Load.

Explicitly assigning max_record_len is only necessary with Profibus communication; Profinet communication already uses a valid max_record_len value.

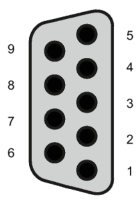
Error condition with GEO2LOG

The GEO2LOG address returns error 16#8095 when the program uses it to attempt to obtain the address of a configured signal module. You cannot use GEO2LOG to obtain addresses of central (local rack) I/O modules, but you can use it to obtain addresses of distributed I/O modules.

RS485/RS422 connector pin assignments

Table A-257 describes the pins of the RS485/RS422 connector. This update to that table provides updated information about the pin assignments:

Table 1 RS485 or RS422 connector (female)

Pin	Description	Connector (female)	Pin	Description
1	Logic or communication ground		6 PWR	+5 V with 100 ohm series resistor: Output
2 TxD+ [1]	Connected for RS422 Not used for RS485: Output		7	Not connected
3 TxD+[2]	Signal B (RxD/TxD+): Input/Output		8 TXD-[2]	Signal A (RxD/TxD-): Input/Output
4 RTS[3]	Request to send (TTL level) Output		9 TXD- [1]	Connected for RS422 Not used for RS485: Output
5 GND	Logic or communication ground		SHELL	Chassis ground

[1] Pin 2 (TxD+) and Pin 9 (TxD-) are the RS422 transmit signals.

[2] Pin 3 (RxD/Tx+) and Pin 8 (RxD/TxD-) are RS485 transmit and receive signals. For RS422, Pin 3 is RxD+ and Pin 8 is RxD-.

[3] The RTS is a TTL level signal and can be used to control another half duplex device based on this signal. It is active when you transmit and is inactive all other times.

Potential problems with performing a firmware update from the Web server

In the event of a communications disruption during a firmware update from the Web server, your Web browser could display a message asking whether you want to leave or stay on the current page. To avoid potential problems, select the option to stay on the current page.

If you close the Web browser while in the process of performing a firmware update from the Web server, you will be unable to change the operating mode of the CPU to RUN mode. If this situation happens, you must cycle power to the CPU to be able to change the CPU to RUN mode.

Correction of CPU 1211C DC/DC/Relay input current specification

The Power Supply table (Table A-15) is incorrect. The input current for the CPU only at max. load for the CPU 1211C DC/DC/Relay is 300 mA at 24 VDC (not 30 mA at 24 VDC).

Inability to write floating point values from the Web server

The Web server cannot write floating point values such as 3.5 to PLC tags or data block tags of type Real or LReal from either the Variable Status standard Web page or from a user-defined Web page. If you want to write a floating point value to a Real or LReal tag or data block tag, or to a memory address, you have the following alternatives:

- For PLC tags and standard (non-optimized) data block tags of type Real, you can use the Variable Status page to modify the value by its direct memory address rather than by the tag name. You cannot modify LReals from the Variable Status page using a memory address.
- Use integer tags for the Web server input, and use floating point math in the STEP 7 program to convert the number to the desired degree of accuracy.
- Write the Hexadecimal or binary representation of an IEEE 754 floating point number to the floating point tag or memory location. This method is only available for data entry from the Variable Status standard Web page. User-defined Web pages are not able to write binary or Hexadecimal data values to floating point tags.

Modifying Real tags through memory addresses

On the Variable Status standard Web page, you can enter the address of a Real tag from the PLC tag table or from a non-optimized data block, for example, MD0 or DB1.DBD0. Data block addresses contain the DB number and the offset within the data block, which you can find in STEP 7. You can then use the FLOATING_POINT display type to modify the memory corresponding to the Real tag. You cannot modify LReal tags by memory address.

Converting integer Web server data entry to floating point values

Because the Web server cannot write floating point values to the PLC, you could create integer tags (for example, UInt, SInt, DInt) in your STEP 7 program for the purpose of accepting data entry from the Web server. Based on your desired degree of accuracy, your integer tags could represent 10x, 100x, 1000x or some other multiple of 10 of your actual floating point value.

For example, if you are expecting floating point values to three decimal points of accuracy, you could accept data entry from operators with a multiplier of 1000 and then perform a division by 1000.0 in the STEP 7 user program.

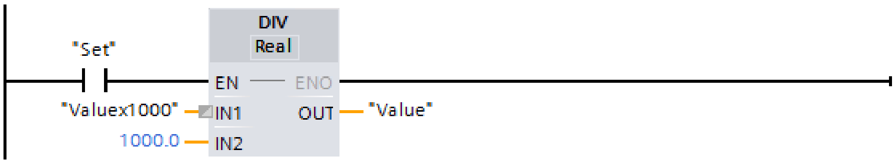
You could declare tags such as the following in a PLC tag table, or in a data block:

Tag table_1

		Name	Data type
1		Valuex1000	UInt
2		Value	Real

From the Variable Status page or from a user-defined Web page, operators would then enter an integer value for the Valuex1000 tag. For example, if the floating point value to be entered is 45.678, the operator would enter 45678 for the Valuex1000 tag.

The STEP 7 program would then include logic to produce the desired floating point value from the entered integer value, for example:



Of course, the STEP 7 program would need to perform appropriate range checking.

Converting a floating point number to Hexadecimal

The S7-1200 stores floating point numbers in IEEE 754 format. This format consists of a sign bit, a "biased" exponent for the power of 2, and the mantissa using the following formula:

<sign> 2^{exponent} * mantissa

The Real data type uses 32 bits and LReal uses 64 bits. The allocation of the bits to the sign, exponent, and mantissa for both data types are as follows:

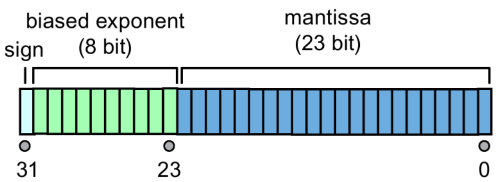


Figure 1 Real number 32-bit representation in IEEE 754 format

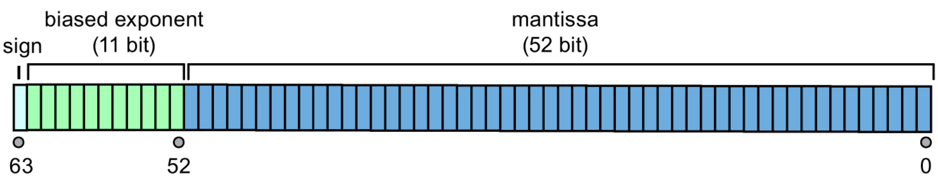


Figure 2 LReal number 64-bit representation in IEEE 754 format

To convert a floating point number to a Hexadecimal value, follow these steps:

1. Convert your Base 10 floating point number to Base 2 (binary). For example, $3.5_{10} = 11.1_2$ (2 + 1 + 1/2)
2. Normalize the resulting value to have a single digit to the left of the binary point by multiplying by the appropriate power of 2: $11.1_2 = 1.11 * 2^1$
3. Determine the sign, exponent (power of 2), and the digits to the right of the binary point of your normalized result. For this example, the sign is 0 because 3.5 is positive; the exponent is 1, and the digits to the right of the binary point are 11.
4. Determine the biased exponent by adding 127_{10} to the exponent for a 32-bit Real value, and by adding 1023_{10} to a 64-bit LReal value. For the example of 3.5, the biased exponent is 128_{10} for a Real and 1024_{10} for an LReal. Determine the binary representation of this biased exponent. The binary representation of 128_{10} is 10000000_2. (Precede with 0's as necessary to form an 8-bit biased exponent for a Real, or an 11-bit biased exponent for an LReal.)
5. Represent the mantissa in 23 bits for a Real or 52 bits for an LReal by adding 0's to the end of the digits to the right of the binary point in Step 3 for the required number of bits.

6. For example, the representation for 3.5 in both 32-bit and 64-bit IEEE 754 representation is as follows:

Sign	Exponent	Mantissa
0	10000000	11000000000000000000000

32-bit Real representation in IEEE 754 format

Sign	Exponent	Mantissa
0	10000000000	1100000000000000000000000000000000000000000000000000

64-bit LReal representation in IEEE 754 format

7. Convert the binary representation to Hexadecimal. Use a Hexadecimal digit 0 - F to correspond to each group of four binary digits.

Binary to Hexadeximal conversion			
0000 = 0	0100 = 4	1000 = 8	1100 = C
0001 = 1	0101 = 5	1001 = 9	1101 = D
0010 = 2	0110 = 6	1010 = A	1110 = E
0011 = 3	0111 = 7	1011 = B	1111 = F

Converting to Hexadecimal, the 32-bit IEEE 754 representation of 3.5 is as follows:

$0100\ 0000\ 0110\ 0000\ 0000\ 0000\ 0000\ 0000_2 = 4060\ 0000_{16}$

Correspondingly, the 64-bit is as follows:

$0100\ 0000\ 0000\ 1100\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000_2 = 400C\ 0000\ 0000\ 0000_{16}$

Note

You can find conversion tools to help with the numeric conversions, especially for numbers that do not convert easily from decimal to IEEE 754 format.

Entering binary and Hexadecimal numbers in the Web server

In Variable Status fields you precede binary numbers with 2# and Hexadecimal numbers with 16#. For the floating point value of 3.5, you could enter any of the following representations:

Real: 2#01000000011000000000000000000000

Real: 16#40600000

LReal: 2#0100000000001100000000000000000000000000000000000000000000000000

LReal: 16#400C000000000000

Effect on remote wind turbine monitor example

The remote wind turbine monitor example in the S7-1200 System Manual makes frequent use of writable floating point fields in a user-defined Web page. The S7-1200 V4.1 CPU no longer supports this example in its current form.

Start information for the hardware interrupt OB

The information in the following table supplements the information in the S7-1200 System Manual section 5.1.3.5, "Hardware interrupt OB":

Input	Data type	Description
LADDR	HW_IO	Hardware identifier of the module that triggers the hardware interrupt
USI	WORD	Identifier for future extensions (not user-relevant)
IChannel	USINT	Number of the channel that triggered the hardware interrupt
EventType	BYTE	Identifier for the event type associated with the event triggering the interrupt (for example, positive edge) This identifier can be found in the description of the respective module.

CM 1241 V2.1 firmware required for Point-to-Point, Modbus RTU, and USS library instructions

The S7-1200 CM 1241 firmware update V2.1 is necessary for the CM 1241 communication modules to support the new Point-to-Point, Modbus RTU, and USS library instructions that the *S7-1200 V4.1 Programmable Controller System Manual* describes. Without the V2.1 firmware update, the CM 1241 communication modules support only the legacy versions of these communication instructions.

Use of the DPRD_DAT and DPWR_DAT instructions to access consistent data

The DPRD_DAT and DPWR_DAT instruction descriptions are corrected and shown below:

Table 2 DPRD_DAT and DPWR_DAT instructions

LAD / FBD	SCL	Description

	`ret_val := DPRD_DAT(`
	`laddr:=_word_in_,`
	`record=>_variant_out_);`
		Use the DPRD_DAT instruction to read one or more bytes of data from one of the following locations:
		• Module or submodule in the local base
		• DP standard slave
		• PROFINET I/O device
		The CPU transfers the data read consistently. If no errors occur during the data transfer, the CPU enters the read data into the target area set up by the RECORD parameter. The target area must have the same length as you configured with STEP 7 for the selected module. When you execute the DPRD_DAT instruction, you can only access the data of one module or submodule. The transfer starts at the configured start address.

	`ret_val := DPWR_DAT(`
	`laddr:=_word_in_,`
	`record:=_variant_in_);`
		Use the DPWR_DAT instruction to transfer the data in RECORD consistently to the following locations:
		• Addressed module or submodule in the local base
		• DP standard slave
		• PROFINET I/O device
		The source area must have the same length as you configured with STEP 7 for the selected module or submodule.

- PROFIBUS supports up to 4 bytes of consistent data. Use the DPRD_DAT and DPWR_DAT instructions to access more than 4 bytes of data consistently.

CE approval - EC Directive 94/9/EC (ATEX) update

The CE approval, EC Directive 94/9/EC (ATEX) has been updated to include EN 60079-0:2012 as shown below.

The S7-1200 Automation System satisfies requirements and safety related objectives according to the EC directives listed below, and conforms to the harmonized European standards (EN) for the programmable controllers listed in the Official Journals of the European Community.

- EC Directive 2006/95/EC (Low Voltage Directive) "Electrical Equipment Designed for Use within Certain Voltage Limits"
  - EN 61131-2:2007 Programmable controllers - Equipment requirements and tests
- EC Directive 2004/108/EC (EMC Directive) "Electromagnetic Compatibility"
  - Emission standard
    EN 61000-6-4:2007+A1:2011: Industrial Environment
  - Immunity standard
    EN 61000-6-2:2005: Industrial Environment
- EC Directive 94/9/EC (ATEX) "Equipment and Protective Systems Intended for Use in Potentially Explosive Atmosphere"
  - EN 60079-0:2012 + A11:2013
  - EN 60079-15:2010: Type of Protection 'n'

The CE Declaration of Conformity is held on file available to competent authorities at:

Siemens AG
Sector Industry
DF FA AS DH AMB
Postfach 1963
D-92209 Amberg
Germany

Wiring rules

The wiring guidelines section includes the following updates:

Table 3 Wiring rules for S7-1200 CPUs, SMs, and SBs

Wiring rules for...	CPU and SM connector	SB connector
Connectible conductor cross-sections for standard wires	2 mm^2 to 0.3 mm^2 (14 AWG to 22 AWG)	1.3 mm^2 to 0.3 mm^2 (16 AWG to 22 AWG)
Number of wires per connection	1 or combination of 2 wires up to 2 mm^2 (total)	1 or combination of 2 wires up to 1.3 mm^2 (total)
Wire strip length	6.4 mm	6.3 to 7 mm
Tightening torque* (maximum)	0.56 N-m (5 inch-pounds)	0.33 N-m (3 inch-pounds)
Tool	2.5 to 3.0 mm flathead screwdriver	2.0 to 2.5 mm flathead screwdriver

* To avoid damaging the connector, be careful that you do not over-tighten the screws.

Note

Ferrules or end sleeves on stranded conductors reduce the risk of stray strands causing short circuits. Ferrules longer than the recommended strip length should include an insulating collar to prevent shorts due to side movement of conductors. Cross-sectional area limits for bare conductors also apply to ferrules.

Correction of terminal block spare kits ordering information

The S7-1200 CPU V4.0 and later - Terminal block spare kits (Table C-16) and S7-1200 SMs V4.0 and later - Terminal block spare kits (Table C-17) are corrected and shown below:

Table 4 S7-1200 CPU V4.0 and later - Terminal block spare kits - corrections

If you have S7-1200 CPU V4.0 and later (article number)	Use this terminal block spare kit (4/pk)	
	Terminal block article number	Terminal block description
CPU 1211C DC/DC/Relay (6ES7 211-1HE40-0XB0)	6ES7 292-1BC30-0XA0	3 pin, gold-plated
	6ES7 292-1AH40-0XA0	8 pin, tin-plated, keyed
	6ES7 292-1AP30-0XA0	14 pin, tin-plated
CPU 1212C DC/DC/Relay (6ES7 212-1HE40-0XB0)	6ES7 292-1BC30-0XA0	3 pin, gold-plated
	6ES7292-1AH40-0XA0	8 pin, tin-plated, keyed
	6ES7 292-1AP30-0XA0	14 pin, tin-plated
CPU 1215C DC/DC/DC (6ES7 215-1AG40-0XB0)	6ES7 292-1BF30-0XB0	6 pin, gold-plated
	6ES7 292-1AM30-0XB0	12 pin, tin-plated
	6ES7 292-1AV30-0XB0	20 pin, tin-plated

Table 5 S7-1200 SMs V4.0 and later - Terminal block spare kits

If you have S7-1200 SM V4.0 and later (article number)	Use this terminal block spare kit (4/pk)	
	Terminal block article number	Terminal block description
SM1231 AI8 x RTD (6ES7 231-5PF32-0XB0)	6ES7 292-1BL30-0XA0	11 pin, gold-plated

SM 1231, SM 1234, and SB 1231 wiring current transducers

Wiring current transducers are available as 2-wire transducers and 4-wire transducers. The SM and SB models affected are:

- SM 1231 AI 4 x 13 bit (6ES7 231-4HD32-0XB0)
- SM 1231 AI 8 x 13 bit (6ES7 231-4HF32-0XB0)
- SM 1231 AI 4 x 16 bit (6ES7 231-5ND30-0XB0)
- SM 1234 4 x Analog Input / 2 x Analog Output (6ES7 234-4HE32-0XB0)
- SB 1231 AI 1 x 12 bit (6ES7 231-4HA30-0XB0)

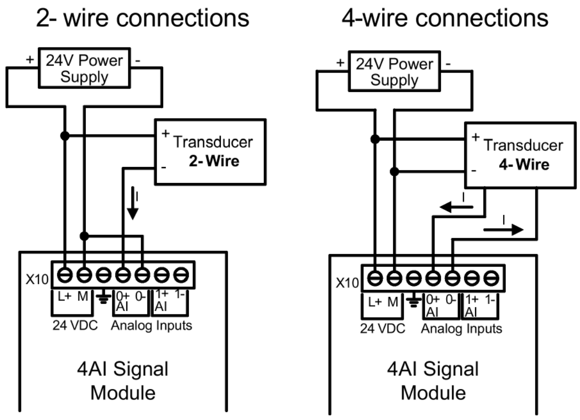


Figure 3 SM 1231 and SM 1234 wiring current transducer

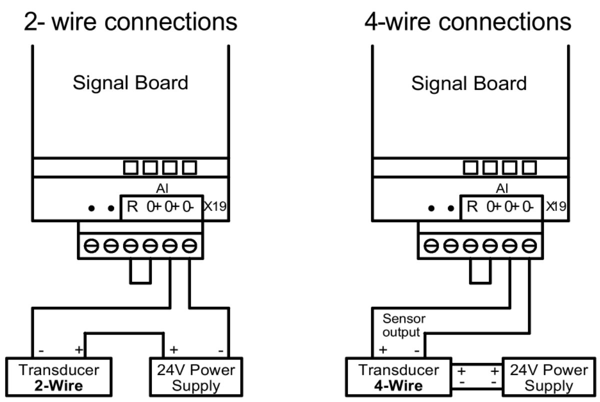


Figure 4 SB 1231 wiring current transducer

Siemens AG
Division Digital Factory
Postfach 48 48
90026 NÜRNBERG

Update to the S7-1200 System Manual, edition 01/2015
A5E03666852-AB, 03/2015