SIEMENS

SIMATIC

S7-1200
SIMATIC S7-200 to SIMATIC S7-1200 software conversion tool

Manual

1/2010

Legal information

Warning notice system

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 product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation for the specific task, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠WARNING
Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be adhered to. The information in the relevant documentation must be observed.

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
Industry Sector
Postfach 48 48
90026 NÜRNBERG
GERMANY

℗ 01/2010

Table of contents

1 Prerequisites, installation, and overview

Prerequisites

The following software must be installed on your computer or programming device:

- SIMATIC STEP 7-Micro/WIN V4.0 SP6 (or later)
- SIMATIC STEP 7 Basic V10.5 SP2
- SIMATIC S7-200 to SIMATIC S7-1200 software conversion tool

The STEP 7-Micro/WIN program that you want to convert must have these properties:

- The program must have been created using STEP 7-Micro/WIN V4.0 SP7 or an earlier version of STEP 7-Micro/WIN.
- The program must have been created using SIMATIC programming mode. IEC programming mode is not supported.
- The program must have been saved in LAD (ladder editor) format.
- The program must compile with no errors.
- All user-defined password block protection must have been removed from the STEP 7-Micro/WIN program. Remove the password protection from the main routine, subroutines, interrupt routines, and all data pages in the data block. Any program block that is password protected will not be converted.

Note

If a STEP 7-Micro/Win wizard has generated and protected program blocks, then the block protection cannot be removed and these program blocks are not converted.

Install the converter software

- STEP 7 Basic V10.5 SP2 and STEP 7-Micro/WIN V4.0 SP7 (or later) must already be installed or your computer or programming device.
- Double-click the setup.exe file that installs the SIMATIC S7-200 to SIMATIC S7-1200 software conversion tool. You can run the setup.exe file from any directory.

Table 1- 1 S7-200 program to S7-1200 program converter overview

	Task	Tool
1	Create a user program with an unspecific CPU (Page 7)	Converter
2	Configure the unspecific CPU (Page 8)	STEP 7 Basic device configuration
3	Compare the original user program (for the S7-200) with the converted program (for the S7-1200) • Program blocks (Page 10) • Symbols/tags (Page 11)	STEP 7-Micro/WIN and STEP 7 Basic
4	Complete the converted S7-1200 program and verify correct run-time operation (Page 13)	STEP 7 Basic

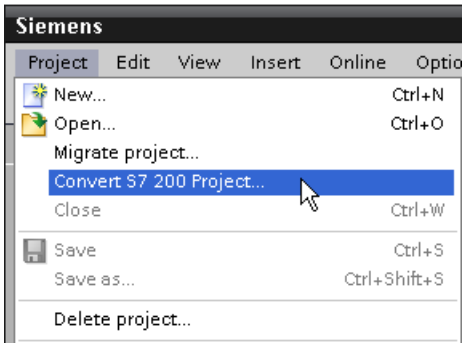
Program conversion process

2

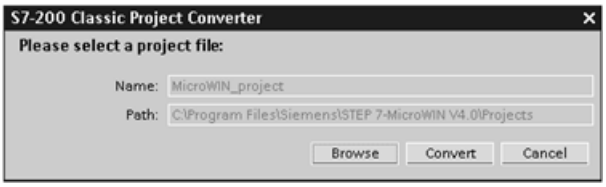
2.1 Run the converter tool

SIMATIC S7-200 to SIMATIC S7-1200 software conversion tool operation

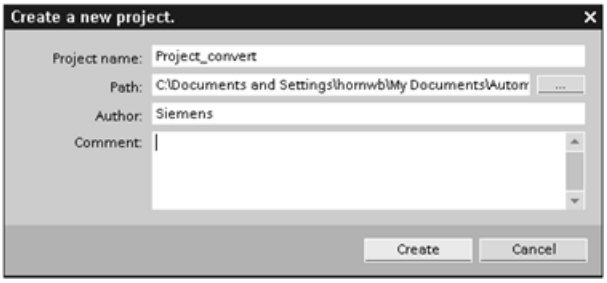
1. Start STEP 7 Basic.
2. Select the Project view.
3. From the "Project" menu, select "Convert S7 200 Project".



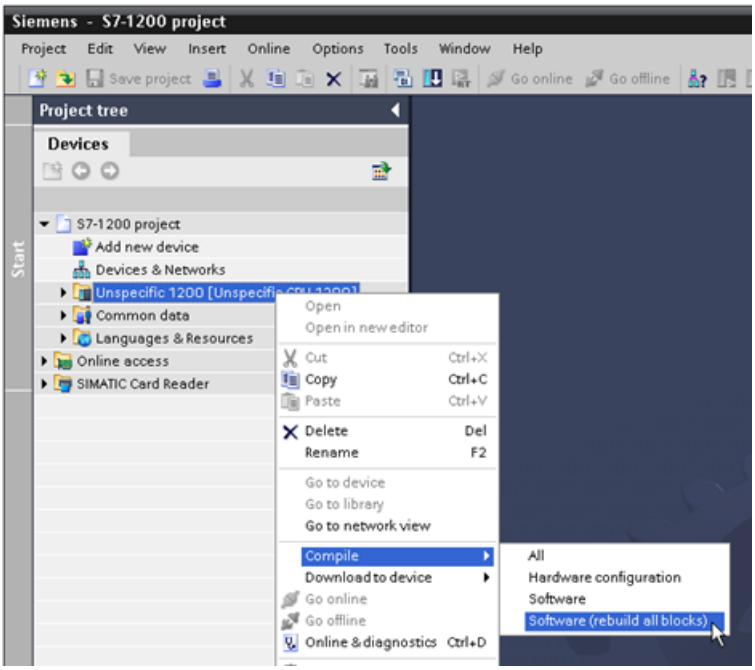
4. Click the "Browse" button on the "S7-200 Classic Project Converter" dialog and set the path to the STEP 7-Micro/WIN *.mwp project file.



5. Start the conversion process by clicking the "Convert" button.
6. Complete the conversion process by editing the "Create a new project" dialog text and clicking the "Create" button.



7. Update the blocks and links between blocks with the STEP 7 Basic program compiler. Use the mouse and right-click over the new "Unspecific CPU 1200" device in the Project tree. Select the "Compile > Software (rebuild all blocks)" item from the right-click menu.



WARNING

Due to functional differences between the S7-200 and S7-1200, your user program may not have been completely converted

An incomplete or improper conversion of your user program may cause unexpected machine or process operation which could cause death, serious injury, or property damage.

You must review and check the converted program to ensure proper and safe operation in your application.

2.2 Configure the S7-1200 CPU and I/O modules

Note

The converter tool does not configure the S7-1200 CPU and I/O modules.

1. Use the STEP 7 Basic Device configuration to either detect the configuration of the connected S7-1200 system or specify the system components from the Hardware catalog. All S7-200 programs are initially converted to the unspecified S7-1200 CPU type.
2. S7-200 system Block CPU parameters and CPU status and control functions that are programmed by SM memory addresses are not converted. You must set up the corresponding S7-1200 CPU properties using STEP 7 Basic Device configuration settings. CPU options that your S7-1200 program will use must be properly configured, before your program can access or operate those CPU options. Some of the S7-1200 CPU configuration options are listed in the following table.

Table 2- 1 S7-1200 Device configuration options compared to S7-200

S7-1200 Device configuration		S7-1200 parameter		S7-200 parameter	
Enable System and clock memory	System memory byte	Location of system memory byte (MBx)		Fixed SM (Special memory) address	
			Always 1 bit		SM0.0 Always_On
			First cycle bit		SM0.1 First_Scan_On
	Clock memory bits	Location of clock memory byte (MBx)			SM0.5 Clock-1s
Configure Digital I/O	Digital inputs	Configurable address		Fixed address	
			Input filters		Digital input filters
			Enable rising edge interrupt		Fixed assignment
			Enable falling edge interrupt		Fixed assignment
			Pulse catch		Pulse catch bits
	Digital outputs	Configurable address		Fixed address	
			RUN-to-STOP output state		Digital output table
Configure Analog I/O	Analog inputs	Configurable address		Fixed address	
			Noise reduction		No support
			Measurement type		Hardware setup
			Voltage range		Hardware setup
			Smoothing		Analog input filters
	Analog outputs	Configurable address		Fixed address	
			Output type		Hardware setup
			RUN-to-STOP output state		Analog output table
Enable High-speed counters	HSC 1, 2, 3, 4, 5, 6	Configurable address		Fixed address	
			Type of counting		Counting type only
			Operating phase		HDEF instruction Mode parameter
			Input source		Fixed assignment
			Count direction control		Mode selection and SM address
			Initial count direction		SM address
			Initial values		SM address
			Reset options		Mode selection and SM address
			Interrupt OB event configuration		ATCH instruction event parameter
Enable Pulse generators	PTO1/PWM1, PTO2/PWM2	Configurable address		Fixed address	
			Generator type		SM address
			Output source		Fixed
			Time base		SM address
			Pulse width format		milliseconds
			Cycle time		SM address
			Initial pulse width		SM address
			Hardware output		Built-in I/O

S7-1200 Device configuration		S7-1200 parameter		S7-200 parameter	
Retentive Memory	Internal flash memory	Retentive range setting		System block setting	
			The Tag table can retain a range of M memory (2048 byte maximum)		Six ranges in V, T and C actual values, or M
			A DB editor can retain a block of DB data. The 2048 byte total is shared between M and DB memory.		

2.3 Compare the program blocks in the original program and the converted program

The purpose of the comparison is to identify the subroutines, interrupt routines, and data block data that were converted.

- Open the original program in STEP 7-Micro/WIN and open the converted program in STEP 7 Basic, at the same time. Compare the POU (Program Organizational Unit) block structure of original S7-200 program with the converted STEP 7 Basic program.
- Use the STEP 7-Micro/WIN Instruction tree and open the "Program Block" and "Data Block" branches.
- Use the STEP 7 Basic Project tree and open the "Project blocks" branch.

The subroutines, interrupt routines, and V memory data groups (data block tabs) created by STEP 7-Micro/WIN wizards and library usage are protected and are not converted. Also user-protected POUs or V memory tabs are not converted.

If S7-200 code blocks are not converted, then you must create substitute program logic in the S7-1200 program. Unconverted code blocks and data block tabs (data groups) can create undefined symbolic references in POU blocks that were successfully converted. Undefined symbolic references must be resolved in the STEP 7 Basic program.

For information about specific program logic instructions, see S7-200 LAD instruction conversion (Page 23).

For more information about how the program structure is converted, see Program structure conversion (Page 15).

Table 2- 2 POU (Program Organizational Unit) conversion rules

S7-200 program	S7-1200 program
Main routine	Program cycle OB1
V memory data	DB1
Timer and counter instructions	Timer instruction with a timer DB (data block) Counter instruction with a counter DB
SBR subroutine	FC (function): • The converted FC numbering is offset by 1 compared to the S7-200 SBR number (SBR0 converts to FC1). • FCs are created with or without local parameters. • FC call parameters appear in the block interface table of the STEP 7 Basic block editor which corresponds to the local variable table of a STEP 7-Micro/WIN subroutine.
INT interrupt routine	Hardware interrupt OB or cyclic interrupt OB: • OBs are created with or without local parameters. • OB local parameters appear in the block interface table of the STEP 7 Basic block editor which corresponds to the local variable table of a STEP 7-Micro/WIN interrupt routine.

Note

STEP 7-Micro/WIN wizard and library (*.mwl file) program code is not converted

This includes subroutines, interrupt routines, V memory data, and symbol definitions. You must create substitute program logic and program symbols.

2.4 Compare the symbols/tags in the original program and the converted program

The purpose of the comparison is to identify the symbols that were completely converted, partially converted, newly created, or not converted.

- Open the original program in STEP 7-Micro/WIN and open the converted program in STEP 7 Basic, at the same time.
- Compare the STEP 7-Micro/WIN program symbols with the converted STEP 7 Basic program tags (symbolic references).

Global symbols

- STEP 7-Micro/Win global symbols are defined in the Symbol table.
- The converted STEP 7 Basic global symbols (I, Q, and M memory only) are located in the PLC Tag table or in data block DB1 (converted S7-200 V memory symbols).

Local symbols

- STEP 7-Micro/Win local symbols are defined in the Local variable tables of the main routine, subroutines, and interrupt routines.
- The converted STEP 7 Basic local symbols are defined in the Block interface table of the corresponding OBs (Organizational blocks) and FCs (Functions).

Table 2- 3 Symbol conversion rules

STEP 7-Micro/WIN program symbolic reference	Converts to this STEP 7 Basic PLC tag	STEP 7 Basic programming issues
I, Q, or M memory symbol	PLC Tag table I, Q, and M memory tags	The I/O base addresses can be modified by PLC Device configuration and can make the converted S7-200 address incorrect.
AIW and AQW (Analog I/O symbols)	PLC Tag table I and Q memory tags	
HC symbol name (High-speed counter)	PLC Tag table I memory tag	
SM and S memory symbols	Undefined program parameter The S7-1200 does not have SM or S memory. Define the parameter name in substitute program logic. In the program editor, a red squiggle underline indicates an undefined parameter name.	• Make a Tag table entry (I, Q, M, or constant): In the program editor, right-click on the parameter name and select "Define tag". • Make a data block reference: Change the undefined symbol name to a data block address. Ex. DB1.SymV0_1
V memory symbol	The V memory symbol name becomes a DB1 data block element name, or the S7-200 V memory symbol is discarded because it overlapped the address of another V memory symbol.	A discarded S7-200 V memory symbol creates an undefined parameter in converted program logic. You must assign a new symbol name and DB1 address.
T (timer memory) and C (counter memory	Timer and counter symbol names become the DB names of the data blocks associated with the converted timer and counter instructions.	• In S7-1200 LAD programs, The Q output of counter and timer boxes replaces the function of the S7-200 counter and timer bit addresses. • Timer and counter bits/current values can also be accessed as data elements within the associated DB.
L memory symbol (Local variable table)	L memory symbol (Block interface table)	

Note

STEP 7-Micro/WIN wizard and library symbols

The symbols used in subroutines, interrupt routines, and V memory data groups (data block tabs) created by STEP 7-Micro/WIN wizards or library usage are not converted. Also, symbols used in user-protected POUs or V memory tabs are not converted.

Unconverted code blocks and data block tabs (data groups) can create undefined symbols in POU blocks that were successfully converted. You must define these symbols to use symbolic addressing in substitute program logic.

2.5 Complete the STEP 7 Basic program and verify the run-time operation

1. Complete all program parameters and program logic.
2. Some S7-1200 CPU I/O channel assignments may use a different I/O channel number compared to the I/O channel numbers used in the S7-200 program. Reassign and rewire the terminal block connections as needed.
3. Compile and download the user program to the S7-1200 CPU
4. Debug and verify the run-time operation the S7-1200 program.

Note

Monitoring program variables with STEP 7 Basic

STEP-7 Micro/WIN status charts are not converted. You must create the corresponding STEP 7 Basic watch tables to monitor run-time data values.

⚠ WARNING

Due to functional differences between the S7-200 and S7-1200, your user program may not have been completely converted

An incomplete or improper conversion of your user program may cause unexpected machine or process operation which could cause death, serious injury, or property damage.

You must review and check the converted program to ensure proper and safe operation in your application.

Program structure conversion

3

3.1 Program structure conversion overview

Table 3- 1 S7-200 program components that **are** converted

STEP 7-Micro/WIN Program structure	S7-200 to S7-1200 software conversion tool action
POU code blocks	The original program block organization is preserved. All blocks that are convertible will be converted whether or not any convertible code exists in a block. STEP 7-Micro/WIN programs do not support the STEP 7 Basic block type FB (function block) with an associated DB (data block). Therefore, all STEP 7-Micro/WIN program blocks are converted to an equivalent OB (organization block) or a FC (function).
Ladder networks	STEP 7-Micro/WIN network numbering, titles, and comments are preserved. If there is an empty network or complete network that cannot be converted, then an empty or partially converted STEP 7 Basic network is created.
Box instructions	Box instruction placement is preserved. If a box instruction cannot be converted, then a corresponding generic box is created as a place holder. If the box has more than one power flow connection, then the additional power flow connections are left open for you to connect.
Main routine	The STEP 7-Micro/Win program Main POU is converted to the STEP 7 Basic program cycle OB1.
Subroutines	STEP 7-Micro/WIN program SBR subroutines are converted to FC functions. FCs are created with or without local parameters. FC local parameters appear in the block interface table of the STEP 7 Basic block editor and correspond to the local variable table parameters of a STEP 7-Micro/WIN subroutine. The converted FC numbering is offset by 1 compared to the S7-200 SBR number (SBR0 converts to FC1).
Interrupt routines	STEP 7-Micro/WIN program INT (interrupt routines) are converted to hardware interrupt OBs or cyclic interrupt OBs. The S7-200 interrupt attach (ATCH) and detach (DTCH) instructions are converted to generic box instruction placeholders. You must enable S7-1200 interrupt events with the CPU device configuration. Then, replace the generic box instructions with S7-1200 ATTACH and DETACH instructions that use a configured interrupt event.

Table 3- 2 S7-200 program components that **are not** converted

STEP 7-Micro/WIN Program structure	S7-200 to S7-1200 software conversion tool action
System Block	STEP 7-Micro/WIN System block parameters are not converted. You must use STEP 7 Basic Device configuration to set up PLC system options.
Wizard code	Program code blocks created by STEP 7-Micro/WIN wizards are not converted.
Library code	STEP 7-Micro/WIN can use Siemens supplied libraries (USS protocol and Modbus protocol) and user-defined libraries. Siemens supplied libraries are provided in STEP 7-Micro/WIN *.mwl file form and are not converted. User-defined library *.mwl files are created from user-created source program *.mwp files. The *.mwp library source programs can be converted to STEP 7 Basic programs and assigned as a global or project library with the STEP 7 Basic library functions.
Status Chart	STEP 7-Micro/WIN Status charts are not converted. You must create a STEP 7 Basic watch table to monitor the run-time values of program variables.

3.2 Memory addresses

Memory addresses are converted as defined in the following table. S7-200 I/O address ranges are fixed by the CPU operating system. However, you can modify S7-1200 I/O address ranges during CPU device configuration.
Therefore, the S7-1200 I/O addresses shown in the following table are example addresses only.

Table 3- 3 S7-200 memory addresses that **are** converted

STEP 7-Micro/WIN Program element	S7-200 address	Converted S7-1200 memory address
Inputs (for instructions that use the Process image)	I0.0	I0.0
Outputs (for instructions that use the Process image)	Q0.0	Q0.0
Inputs (for immediate instructions)	I0.0	I0.0:P
Outputs (for immediate instructions)	Q0.0	Q0.0:P
Analog Inputs	AIWx	IWy
Analog Outputs	AQWx	QWy
Variable Memory	VW0	DB1.DBW0
Marker Memory	M0.0	M0.0

STEP 7-Micro/WIN Program element	S7-200 address	Converted S7-1200 memory address
Timers	T32	S7-200 timer symbol names are converted to S7-1200 timer DB names. S7-200 timer output bit addresses are converted to a Q bit address, in the timer DB. If a S7-200 timer PT (Preset Time) input is a WORD constant, then the program value is not converted, for example "100" for PT on T32 means 100ms. You must convert the S7-200 program value to the equivalent S7-1200 TIME data format, for example "T#100ms", "T#59s999ms", or "T# 23h59s999ms".
Counters	C10	S7-200 counter symbol names are converted to S7-1200 counter DB names. S7-200 counter output bit addresses are converted to a Q bit address, in the counter DB.
High Speed Counter	HC0	ID1000 (Input Double Word 1000)

Table 3- 4 S7-200 memory addresses that **are not** converted

STEP 7-Micro/WIN Program element	S7-200 address	Converted S7-1200 memory address
Special Memory (System status and control)	SM0.0	Undefined parameter in the program logic: You must assign a new symbol name and address in a S7-1200 tag group or data block.
SCR (Sequence Control Relay)	S0.0	
Accumulators	AC0	
Indirect Addressing	*VD0, *AC0, *LD0	
	&VB0	

Analog value address conversion

S7-1200 analog addresses are set during the STEP 7 Basic device configuration. These addresses are configured as part of the analog I/O device configuration. Analog value addressing in the S7-1200 uses I and Q memory addresses. The following table lists the default address ranges that are assigned to analog values converted from a STEP 7-Micro/WIN program. You must configure the S7-1200 analog I/O to use these addresses or substitute other addresses.

Table 3- 5 S7-200 to S7-1200 analog address conversion

Analog I/O	S7-200 Analog address	S7-1200 Analog address
Analog Input 1	AIW0	IW64
Analog Input 2	AIW2	IW66
Analog Input 3 and following	AIW4...	IW500...
Analog Output 1 and following	AQW0...	QW500...

3.3 Symbol table

STEP 7-Micro/WIN program symbols must be fully defined to be converted. This means that both the symbol name and corresponding address must be valid.

STEP 7-Micro/WIN program symbols are converted as shown in the following tables.

Table 3- 6 S7-200 symbolic references that **are** converted

S7-200 symbol	S7-1200 symbol conversion result
Symbol name	Tag name
Symbol comment	Tag comment
I, Q, and M symbols	I, Q, and M tags
AIW and AQW (Analog I/O) symbols	I and Q tags
HC (High-speed counter) symbols	I tags
T (Timer) symbols	S7-200 timer symbol names are converted to S7-1200 timer DB names. S7-200 timer bit and current value symbols are converted to the corresponding data names, in the timer DB.
C (Counter) symbols	S7-200 counter symbol names are converted to S7-1200 counter DB names. S7-200 counter bit and current value symbols are converted to the corresponding data names, in the counter DB.
V memory symbols	Data block element name
L memory symbols in a local variable table	L memory symbol in a block interface table

Table 3- 7 S7-200 symbolic references that **are not** converted

S7-200 Symbol reference	S7-1200 symbol conversion result
SM (Special Memory) symbols	Undefined parameter name in program logic: You must assign a new symbol name and address in a S7-1200 tag group or data block.
S (Sequence Control Relay) symbols	
Wizard generated symbols	The generated code blocks and associated tag names are not converted. You must make new program logic, tag names, and memory address assignments.
Library generated symbols	

3.4 Data block

STEP 7-Micro/WIN data block data is converted to a single STEP 7 Basic global data block DB1. The converted DB1 has the "Symbolic access only" attribute disabled, so you can use both symbolic and absolute addressing. The STEP 7-Micro/WIN data block has tab sections which selects a data group in the data block. All tabs that are unprotected will be converted into a single STEP 7 Basic data block DB1. Data block tabs that are password protected as "read only" will not be converted. If a data block tab was generated and protected by a STEP 7-Micro/WIN wizard, then you cannot remove the protection and the tab's data group is not converted.

Symbol names for S7-200 V memory addresses as defined in the STEP 7-Micro/WIN Symbol table are converted to data names in the S7-1200 DB1.

STEP 7-Micro/WIN allows the creation of symbols with overlapping V memory data addresses. When a S7-200 symbol is assigned a four byte double word V memory address, you can also create separate symbols for bit, byte, and word addresses contained within the four bytes.

STEP 7 Basic does not allow symbolic access to overlapping data block addresses. If the STEP 7-Micro/WIN Symbol table has groups of V memory symbols with overlapping data addresses, then the symbol name for the largest sized data element is put in the converted DB1 data block. All other overlapping symbol references are discarded. You must assign discarded symbol names to new DB1 addresses.

The STEP 7 Basic data block supports an array data type (Array [lo..hi] of type).

Data arrays are created for S7-200 V memory data addresses where:

- No symbol name was assigned.
- There is a V memory address gap between symbolic declarations.

Table 3- 8 V memory to DB1 data conversion examples

Data description	S7-200 data block format	Converted S7-1200 DB1 format
Decimal value	255	255
Binary value	2#1010	2#1010
Hexadecimal value [1]	16#FFFF	w#16#FFFF
Floating Point	7.77	7.77
ASCII characters Byte size Word size Dword size	1, 2, or 4 bytes 'a' 'ab' 'abcd'	 'a' 'ab' 'abcd'
Multiple ASCII characters [2]	3, 5, or more bytes ‘abcde’	‘abcde’
String [3]	"abcde"	‘abcde’

1 Size descriptor ("w") depends upon data type

2 Converted to S7-1200 string format

3 The S7-1200 string storage format requires one byte more than the S7-200 string storage format. The extra byte is used to store the maximum string length.

3.5 Interrupt events

The S7-200 interrupt attach (ATCH) and detach (DTCH) instructions are converted to generic box instruction placeholders. You must enable S7-1200 interrupt events with the CPU device configuration. Then, replace the generic box instructions with S7-1200 ATTACH and DETACH instructions that use a configured interrupt event.

STEP 7-Micro/WIN program INT interrupt routines are converted to hardware interrupt OBs or cyclic interrupt OBs, as shown in the following tables.

Table 3- 9 Conversion of S7-200 highest interrupt priority (communication)

S7-200 event number	S7-200 Interrupt event	S7-1200 Interrupt OB conversion result
8	Port 0: Rcv character	Hardware interrupt OB
9	Port 0: Xmt complete	Hardware interrupt OB
23	Port 0: Rcv msg complete	Hardware interrupt OB
24	Port 1: Rcv msg complete	Hardware interrupt OB
25	Port 1: Rcv character	Hardware interrupt OB
26	Port 1: Xmt complete	Hardware interrupt OB
19	PTO 0 complete interrupt	Hardware interrupt OB
20	PTO 1 complete interrupt	Hardware interrupt OB

Table 3- 10 Conversion of S7-200 middle interrupt priority (discrete input)

S7-200 event number	S7-200 Interrupt event	S7-1200 Interrupt OB conversion result
0	Rising edge, I0.0	Hardware interrupt OB
2	Rising edge, I0.1	Hardware interrupt OB
4	Rising edge, I0.2	Hardware interrupt OB
6	Rising edge, I0.3	Hardware interrupt OB
1	Falling edge, I0.0	Hardware interrupt OB
3	Falling edge, I0.1	Hardware interrupt OB
5	Falling edge, I0.2	Hardware interrupt OB
7	Falling edge, I0.3	Hardware interrupt OB
12	(HSC0) CV=PV	Hardware interrupt OB
27	(HSC0) direction changed	Hardware interrupt OB
28	(HSC0) external reset	Hardware interrupt OB
13	(HSC1) CV=PV	Hardware interrupt OB
14	(HSC1) direction changed	Hardware interrupt OB
15	(HSC1) external reset	Hardware interrupt OB
16	(HSC2) CV=PV	Hardware interrupt OB
17	(HSC2) direction changed	Hardware interrupt OB
18	(HSC2) external reset	Hardware interrupt OB
32	(HSC3) CV=PV	Hardware interrupt OB
29	(HSC4) CV=PV	Hardware interrupt OB
30	(HSC4) direction changed	Hardware interrupt OB
31	(HSC4) external reset	Hardware interrupt OB
33	(HSC5) CV=PV	Hardware interrupt OB

Table 3- 11 Conversion of S7-200 lowest interrupt priority (timed)

S7-200 event number	S7-200 Interrupt event	S7-1200 Interrupt OB conversion result
10	Timed interrupt 0	OB_Cyclic interrupt (with 100 ms scan time interval)
11	Timed interrupt 1	OB_Cyclic interrupt_1 (with 100 ms scan time interval)
21	Timer T32, CT=PT interrupt	Hardware interrupt OB
22	Timer T96, CT=PT interrupt	Hardware interrupt OB

3.6 Wizards

Note

STEP 7-Micro/WIN wizard generated program code is not converted

This includes subroutines, interrupt routines, V memory data, and symbol definitions. You must create substitute program logic.

Table 3- 12 STEP 7-Micro/WIN programming wizard code conversion issues

STEP 7-Micro/WIN Wizard name	S7-1200 equivalent feature	S7-1200 programming
AS-i	Planned for the future	The current S7-1200 CPU does not support AS-i communication.
Data Log	Under development for V11 release	A data log function is not supported in the STEP 7 Basic V10.5 SP2.
EM 241 Modem/ Remote Modem	Under development for V11 release	The S7-1200 CPU does not support a modem expansion module. Also, the modem and remote modem function are not supported from the S7-1200 RS-232 module.
EM 253 Position	Different programming method available	The S7-1200 CPU does not support a position expansion module. The S7-1200 CPU position function is supported as defined by the PLCopen standard. You must rewrite this program section using the STEP 7 Basic Axis technology object.
Ethernet	Different programming method available	The S7-1200 CPU does not support an Ethernet expansion module. However, you may convert the module addressing configuration (IP, subnet, gateway). This address can be used as the default address of the S7-1200 CPU Ethernet port.
High-Speed Counter	Different programming method available	The STEP 7-Micro/WIN HSC wizard generates unprotected (visible in the editor) interrupt routines and associated code. You were required to add your own instructions within these interrupt routines. The interrupt routines contain both wizard generated and manually added code. You must replace the initialization instructions that use HSC SM memory. Replace the HSC, and HDEF instructions with the corresponding S7-1200 CPU HSC device configuration and CTRL_HSC instructions.

STEP 7-Micro/WIN Wizard name	S7-1200 equivalent feature	S7-1200 programming
Internet	Under development for V11 release	S7-200 Internet module operations such as FTP Server, FTP Client, Email, and User IDs are not supported by firmware in the initial S7-1200 CPU release.
NETR /NETW	Different programming method available	PPI network communication instructions are replaced by Ethernet T-block (TSEND_C. TRCV_C) peer-to-peer communication instructions.
PID	Different programming method available	S7-200 PID operation can be programmed directly with the PID instruction or indirectly through the PID wizard. S7-1200 PID operation uses the PID technology object and the PID_Compact instruction available in STEP 7 Basic. A S7-200 program call to a PIDx_INIT subroutine or a PID instruction must be replaced with the S7-1200 PID_Compact instruction. PID_Compact operating parameters are not converted and must be assigned and initialized by you.
PTO/PWM	Different programming method available	A S7-200 program call to a PWMx_RUN subroutine is converted to the S7-1200 CTRL_PWM instruction. You must assign new pulse control parameters and initialize the parameter values within the converted program. Pulse generator operation parameters must be set in the STEP 7 Basic PLC device configuration, before a program can use the pulse generators.
Recipe	Not supported	This feature is not supported in the STEP 7 Basic V10.5 SP2 release.
Text Display	Different programming method available	The TD panels supported by the STEP 7-Micro/WIN text display wizard are not supported by the S7-1200. You must rewrite this program section to operate a STEP 7 HMI Basic panel through an Ethernet connection.

3.7 Libraries

Note

STEP 7-Micro/WIN program code generated by *.mwl files (compiled libraries) is not converted.

This includes subroutines, interrupt routines, V memory data, and symbol definitions. You must create substitute program logic.

The STEP 7-Micro/WIN *.mwp program file is converted. The *.mwp file is the program source file for regular user programs and for user-created libraries. You must use STEP 7 Basic library commands to create a STEP 7 Basic global or project library from a converted STEP 7-Micro/WIN library source program.

S7-200 LAD instruction conversion

4

4.1 Bit logic

Table 4- 1 Bit logic conversion

S7-200 instruction	Converted?	S7-1200 instruction
-| |- Normally open contact	Yes	-| |- Normally open contact
-| / |- Normally closed contact	Yes	-| / |- Normally closed contact
-| I |- Normally open immediate contact	Yes	-| |- Normally open contact with immediate address
-| / I |- Normally closed immediate contact	Yes	-| / |- Normally closed contact with immediate address
-| NOT |- Inverter	Yes	-| NOT |-
-| P |- Positive edge detector	Yes	P_TRIG Box with M memory bit assigned as the edge history bit. The M bit assignment range starts at M4064.0
-| N |- Negative edge detector	Yes	N_TRIG Box with M memory bit assigned as the edge history bit. The M bit assignment range starts at M4064.0
-( ) Output coil	Yes	-( )- Output coil
-(I) Output immediate coil	Yes	-( )- Output coil with immediate address
-(S) Set coil	Yes	-(S)- Set coil (Set a single bit) or SET_BF (Set Bit Field for multiple bits)
-(SI) Set immediate coil	Yes	-(S)- Set coil with immediate address (Set a single bit) or SET_BF Set Bit Field with immediate addresses (Set multiple bits)
-(R) Reset coil	Yes	-(R)- Reset coil (Reset a single bit) or RESET_BF (Reset Bit Field for multiple bits)
-(RI) Reset Immediate coil	Yes	-(R)- Reset coil with immediate address (Reset a single bit) or RESET_BF Reset Bit Field with immediate addresses (Reset multiple bits)
SR Set dominant bi-stable	Yes	SR
RS Reset dominant bi-stable	No	
NOP No Operation	No	

4.2 Timers

Table 4- 2 Timer conversion

S7-200 instruction	Converted?	S7-1200 instruction
TON On-delay timer	Yes	TON
TONR On-delay retentive timer	Yes	TONR
TOF Off-delay timer	Yes	TOF
BGN_ITIME Begin Interval Time	No	
CAL_ITIME Calculate Interval Time	No	Use Clock instructions RD_SYS_T (read system time) and T_SUB (time difference) to calculate time intervals.

4.3 Counters

Table 4- 3 Counter conversion

S7-200 instruction	Converted?	S7-1200 instruction
CTU Count Up counter	Yes	CTU
CTD Count Down counter	Yes	CTD
CTUD Count Up/Down counter	Yes	CTUD
HSC High-speed counter	No	You must make the conversion which requires the following actions: - Interrupt OB creation for each HSC event and HSC steps - Interpreting and converting the S7-200 HSC SM initialization. - Combining several S7-200 HSC instructions into a single S7-1200 CTRL_HSC instruction.
HDEF High-speed counter Definition	No	There is no equivalent instruction for the S7-1200 CPU. Selection of HSC mode can only be performed by S7-1200 CPU device configuration.
PLS Pulse	No	

4.4 Compare

Table 4- 4 Compare conversion

S7-200 instruction	Converted?	S7-1200 instruction
==B, ==I, ==D, ==R, ==S Equal	Yes *	==
<>B, <>I, <>D, <>R, <>S Not equal	Yes *	<>
>=B, >=I, >=D, >=R Greater than or equal	Yes	>=
<=B, <=I, <=D, <=R Less than or equal	Yes	<=
>B, >I, >D, >R Greater than	Yes	>
<B, <I, <D, <R Less than	Yes	<

* For string comparisons (==S and <>S), only S7-200 literal string (ex. "123456") input parameters are converted. Due to differences between the S7-200 and S7-1200 string storage format, S7-200 string direct address and indirect address parameters are not converted.

4.5 Integer math

Table 4- 5 Integer math conversion

S7-200 instruction	Converted?	S7-1200 instruction
ADD_I Add Integer	Yes	ADD
ADD_DI Add Double Integer	Yes	
SUB_I Subtract Integer	Yes	SUB
SUB_DI Subtract Double Integer	Yes	
MUL_I Multiply Integer	Yes	MUL
MUL_DI Multiply Double Integer	Yes	
DIV_I Divide Integer	Yes	DIV
DIV_DI Divide Double Integer	Yes	
INC_B Increment Byte	Yes *	INC
INC_W Increment Word	Yes *	
INC_ DW Increment Double Word	Yes *	
DEC_B Decrement Byte	Yes *	DEC
DEC_W Decrement Word	Yes *	
DEC_DW Decrement Double Word	Yes *	
MUL Multiply two 16-bit integers and produce a 32-bit product	No	
DIV Divide two 16-bit integers and produce a 32-bit result consisting of a 16-bit remainder (most-significant word) and a 16-bit quotient (least-significant word).	No	

* The S7-200 increment/decrement instructions can have a separate IN and OUT address. The S7-1200 increment/decrement instructions have one IN/OUT parameter and must use a single address. If the S7-200 increment/decrement instruction uses a single IN and OUT address, then the instruction and parameters are converted. If the S7-200 increment/decrement instruction uses separate IN and OUT addresses, then the instruction is converted but not the parameters.

4.6 Floating-point math

Table 4- 6 Floating-point math conversion

S7-200 instruction	Converted?	S7-1200 instruction
ADD_R Add Real	Yes	ADD
SUB_R Subtract Real	Yes	SUB
MUL_R Multiply Real	Yes	MUL
DIV_R Divide Real	Yes	DIV
SQRT Square Root	Yes	SQRT
SIN Sine	Yes	SIN
COS Cosine	Yes	COS
TAN Tangent	Yes	TAN
LN Natural logarithm	Yes	LN
EXP Natural Exponential	Yes	EXP
PID Proportional Integral Derivative loop	No	

4.7 Move

Table 4- 7 Move conversion

S7-200 instruction	Converted?	S7-1200 instruction
MOV_B, MOV_W, MOV_DW	Yes	MOVE Data copy with source unchanged
BLKMOV_B, BLKMOV_W, BLKMOV_D	Yes	BLKMOVE
SWAP	Yes	SWAP
MOV_BIR Move Byte with Immediate Read	Yes	MOVE with immediate address
MOV_BIW Move Byte with Immediate Write	Yes	MOVE with immediate address

4.8 Convert

Table 4- 8 Convert conversion

S7-200 instruction	Converted?	S7-1200 instruction
B_I Byte to Integer	Yes	CONVERT
I_B Integer to Byte	Yes	
I_DI Integer to Double Integer	Yes	
DI_I Double Integer to Integer	Yes	
DI_R Double Integer to Real	Yes	
BCDI BCD to Integer	Yes	
I_BCD Integer to BCD	Yes	
ROUND Round to integer	Yes	ROUND

S7-200 instruction	Converted?	S7-1200 instruction
TRUNC Truncate to integer	Yes	TRUNC
I_S Integer to String	Yes *	VAL_STRG
DI_S Double Integer to String	Yes *	
R_S Real to String	Yes *	
S_I String to Integer	Yes *	STRG_VAL
S_DI String to Double Integer	Yes *	
S_R String to Real	Yes *	
DECO Decode	Yes	DECO
ENCO Encode	Yes	ENCO
ATH ASCII to Hex	No	
HTA Hex to ASCII	No	
ITA Integer to ASCII	No	
DTA Double Integer to ASCII	No	
RTA Real to ASCII	No	
SEG Segment display driver	No	

* Only S7-200 literal string (ex. "123456") input parameters are converted. Due to differences between the S7-200 and S7-1200 string storage format, S7-200 string direct address and indirect address parameters are not converted.

4.9 Program control

Table 4- 9 Program control conversion

S7-200 LAD program instruction	Converted?	S7-1200 LAD program instruction
JMP Jump to label execution control	Yes	JMP
LBL Label a program position	Yes	LBL The label for the S7_1200 JMP and LBL instructions must be alphanumeric characters. The S7-200 label identifier can only be a number. This number must be converted from a number to an equivalent set of characters, such as 1 converts to label1.
RET Return from subroutine	Yes	RET
STOP Go to STOP mode	Yes	STP
WDR Watch Dog Reset	Yes	RE_TRIGR
FOR Indexed loop execution control	No	You must recreate FOR-NEXT loop logic with JMP, LBL, ADD, and Compare instructions.
NEXT Increment loop index	No	
END End program execution	No	
DIAG_LED CPU diagnostic LED control	No	
SCR Sequence Control Relay	No	
SCRT Sequence Control Relay Transition	No	
SCRE Sequence Control Relay End	No	

4.10 Logical operations

Table 4- 10 Logical operations conversion

S7-200 instruction	Converted?	S7-1200 instruction
INV_B, INV_W, INV_DW	Yes	INV Invert
WAND_B, WAND_W, WAND_DW	Yes	AND
WOR_B, WOR_W, WOR_DW	Yes	OR
WXOR_B, WXOR_W, WXOR_DW	Yes	XOR Exclusive OR

4.11 Shift and Rotate

Table 4- 11 Shift and Rotate conversion

S7-200 instruction	Converted?	S7-1200 instruction
SHL_B, SHL_W, SHL_DW	Yes	SHL Shift Left
SHR_B, SHR_W, SHR_DW	Yes	SHR Shift Right
ROL_B, ROL_W, ROL_DW	Yes	ROL Rotate Left
ROR_B, ROR_W, ROR_DW	Yes	ROR Rotate Right
SHRB Shift Register Bit	No	

4.12 Clock and Calendar

Table 4- 12 Clock and Calendar conversion

S7-200 instruction	Converted?	S7-1200 instruction
READ_RTC Read Real-Time Clock	Yes	RD_SYS_T You must supply new data addresses.
SET_RTC Set Real-Time Clock	Yes	WR_SYS_T You must supply new data addresses.
READ_RTCX Read Real-Time Clock Extended	No	Time zone and daylight savings offsets are set by the S7-1200 CPU Device configuration. You can read the local time with the RD_LOC_T instruction.
SET_RTCX Set Real-Time Clock Extended	No	

4.13 String

Table 4- 13 String conversion

S7-200 instruction	Converted?	S7-1200 instruction
STR_LEN String Length	Yes *	LEN
STR_CPY String Copy	Yes *	CONCAT
SSTR_CPY Substring Copy	Yes *	MID
STR_CAT String Concatenate	Yes *	CONCAT
STR_FIND String Find	Yes *	FIND
CHR_FIND Character Find	No	

* Only S7-200 literal string ("abcdef") input parameters are converted. S7-200 string direct address and indirect address parameters are not converted, due to differences between the S7-200 and S7-1200 string storage format.

4.14 Communication

Table 4- 14 Communication conversion

S7-200 instruction	Converted?	S7-1200 instruction
XMT	No	Equivalent XMT/RCV message based freeport functionality is supported through the new PtP (Point to Point) instructions. However, configuration and programming are very different compared to the S7-200 SM (Special Memory) configuration. Character based freeport is not supported.
RCV	No	
NETR	No	Alternative NETR/NETW peer-to-peer PPI network operations are supported through TSEND_C/TRCV_C over Ethernet TCP/IP.
NETW	No	
GET_ADDR	No	Profibus communication is not supported.
SET_ADDR	No	

4.15 Interrupts

Table 4- 15 Interrupts conversion

S7-200 instruction	Converted?	S7-1200 instruction
-(ENI) Enable Interrupt	Yes	EN_AIRT
-(DISI) Disable Interrupt	Yes	DIS_AIRT
-(RETI) Return from Interrupt routine	Yes	RET No special return from interrupt instruction is required.
ATCH Attach Interrupt	Yes	ATTACH
DTCH Detach Interrupt	Yes	DETACH
CLR_EVNT Clear Interrupt Event	No	The S7-1200 DETACH instruction clears current and queued events.

4.16 Table

Table 4- 16 Table conversion

S7-200 instruction	Converted?	S7-1200 instruction
FILL_N Fill Table	Yes	FILL_BLK
LIFO Last In First Out	No	
FIFO First In First Out	No	
ATT Add to Table	No	
TBL_FIND Table Find	No	