SIEMENS

SIMATIC

S7-300
FM 352-5 high-speed Boolean processor

Operating Manual

05/2011
A5E00131318-04

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 relevant 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, 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 complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of 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

A5E00131318-04
℗ 07/2011

Preface

Purpose of this manual

This manual describes the purpose, features, and operation of the SIMATIC S7 FM 352-5 Boolean processor modules (order number: 6ES7352-5AH01-0AE0) and (order number: 6ES7352-5AH11-0AE0). This manual also provides support for installing, configuring, programming, and operating FM 352-5 modules.

Contents of the manual

This manual describes the FM 352-5 hardware and the software required to configure and program the modules. The manual consists of chapters with instructions and reference information (technical specifications).

This manual covers the following topics:

- Installing and wiring FM 352–5 modules
- Configuring FM 352–5 modules
- Setting parameters for FM 352–5 modules
- Programming FM 352–5 modules
- Operating the modules
- Troubleshooting and diagnostics

Related documentation

For more information on installing and programming FM 352–5 Boolean processor modules, refer to the documentation on the SIMATIC S7-300 automation system and the STEP 7 programming software.

CD–ROM

The entire electronic manuals are also available on CD-ROM as the SIMATIC Manual Collection.

Standards, certificates, and approvals

The FM 352-5 fulfills the requirements and criteria of IEC 1131, Part 2, and the requirements for the CE mark. The following approvals apply: FM Class I, Div. 2, Groups A, B, C, D and cULus Class I, Div. 2, Groups A, B, C, D.

Recycling and disposal

The FM 352-5 has low pollutant content and can therefore be recycled. For environmentally compliant recycling and disposal of your discarded device, please contact a company certified for the disposal of electronic waste.

Structure of the manual

The following tools will help you to find specific information:

- At the front of the manual, you will find a detailed table of contents and lists of the figures and tables the manual contains.
- The chapters themselves are divided into sections that are introduced by titles indicating the content of the sections following.
- At the back of the manual, you will find a detailed index with which you can find specific topics quickly.

Additional support

If you have any further questions about the use of products described in this manual and do not find the right answers here, contact your local Siemens representative (http://www.siemens.com/automation/partner):

A guide to the technical documentation for the various products and systems is available on the Internet:

- SIMATIC Guide manuals (http://www.siemens.com/simatic-tech-doku-portal)

The online catalog and online ordering systems are also available on the Internet:

- A&D Mall (http://www.siemens.com/automation/mall)

Training center

To help you get started with automation technology and systems, we offer a variety of courses. Contact your regional Training Center or the central Training Center in D-90327 Nuremberg, Germany.

- Internet: SITRAIN homepage (http://www.sitrain.com)

Technical Support

You can access technical support for all A&D projects via the following:

- Online support request form: (http://www.siemens.com/automation/support-request)

Service & Support on the Internet

In addition to our documentation, we offer a comprehensive online knowledge base on the Internet at:

Industry Automation and Drive Technologies - Homepage
(http://www.siemens.com/automation/service&support)

There you will find the following information, for example:

- The newsletter that provides up-to-date information on your products.
- The documents you need via our Search function in Service & Support.
- A forum for global information exchange by users and specialists.
- Your local partner for Automation and Drives.
- Information about on-site service, repairs, and spare parts. Much more can be found under "Services".

Table of contents

Product overview

1

1.1 Functions of the FM 352-5 module

Overview

The FM 352-5 is a Boolean processor that allows independent and extremely fast control of a process within a larger control system.

The FM 352-5 module can be configured to operate in the following ways:

- The FM 352-5 module can operate in a coprocessor configuration within an S7 programmable controller system. In this configuration, the FM 352-5 exchanges input/output data, and status and control information with the master CPU (see figure below).
- In a distributed configuration, the FM 352-5 module functions as a module of a normal ET 200M PROFIBUS DP slave connected to an S7 or non-S7 master.
- The FM 352-5 module can also operate as a stand-alone controller independently of any PLC system.

The FM 352-5 uses an integrated FPGA (Field Programmable Gate Array) for the simultaneous execution of code in contrast to the sequential execution found in normal programmable controllers. This method of execution allows an extremely fast and stable sampling time. The module processes and controls a series of integrated inputs and outputs (up to 15 inputs and 8 outputs). Apart from the normal inputs and outputs, the module also supports one of three encoder types (differential incremental encoders, single-ended incremental encoders and SSI absolute encoders). If you select SSI absolute encoders or differential incremental encoders, the 24-V encoder inputs are available as digital inputs. If you do not use any of the encoder interfaces, you can represent three digital differential inputs with the connectors for differential incremental encoders.

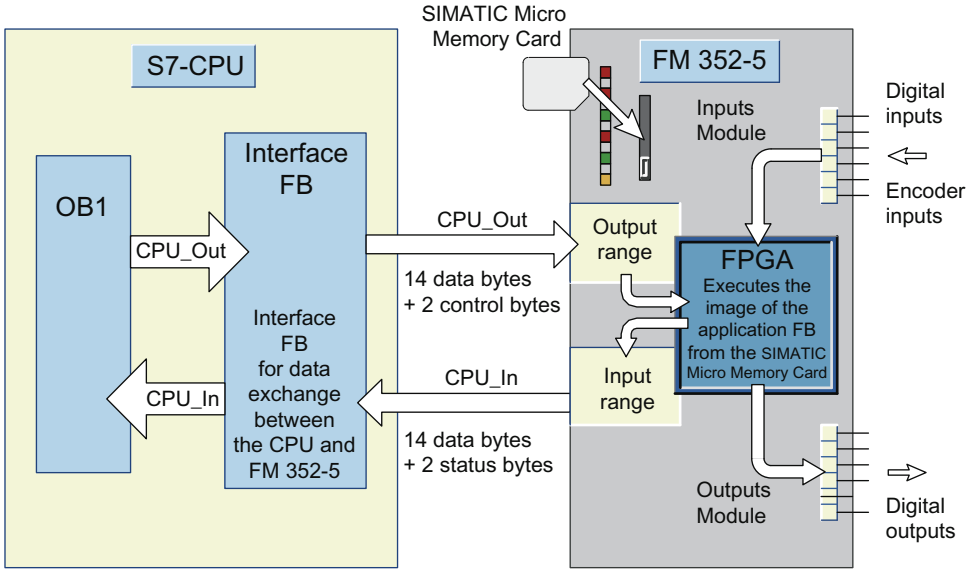


Figure 1-1 Operating the FM 352-5 module as a coprocessor

Configuring the hardware

You configure the FM 352-5 module using the FM 352-5 Configuration software with the standard Hardware Configuration application of STEP 7. The hardware configuration dialogs for the FM 352-5 module allow you to set the following properties and parameters:

- Address assignments, where you can use the S7 system default assignments, or select your own addresses (with CPUs that support address selection).
- Programming parameters, where you specify the numbers of the FBs and DBs used to save the program and select the mode.
- Operating parameters, for example interrupts, input filters, module diagnostics, output diagnostics, encoder parameters etc.

Programming the FM 352-5 module

You program the FM 352-5 module using the FM 352-5 Configuration software with the STEP 7 LAD/FBD editor (version 5.1, SP3 or greater). The FM 352-5 software provides a library of special operations for the Program Elements catalog. The library of function blocks (FBs) for the FM 352-5 includes timers, counters, shift registers, a binary scaler and a clock generator that are intended for use only with the FM 352-5 module You can also select a subset of the STEP 7 bit-logic operations, such as contacts and coils, as you create your program. Operations of the FM 352-5 software are described in chapter "Programming and operating the FM 352-5 (Page 65)".

You write your program in an application FB. Using the FM 352-5 configuration software and STEP 7, the program is compiled and then copied to a SIMATIC Micro Memory Card for non-volatile storage. The SIMATIC Micro Memory Card is inserted in the slot on the front of the module. When the FM 352-5 module is powered up, the stored program is retrieved from the SIMATIC Micro Memory Card and executed by the module.

Operation

The FM 352-5 module executes its program independently of the master CPU. The integrated inputs and outputs are local functional units and cannot be accessed directly by the master CPU. However, the user program of the CPU transfers control commands and configuration parameters to the FM 352-5 module and evaluates the status information returned by the module.

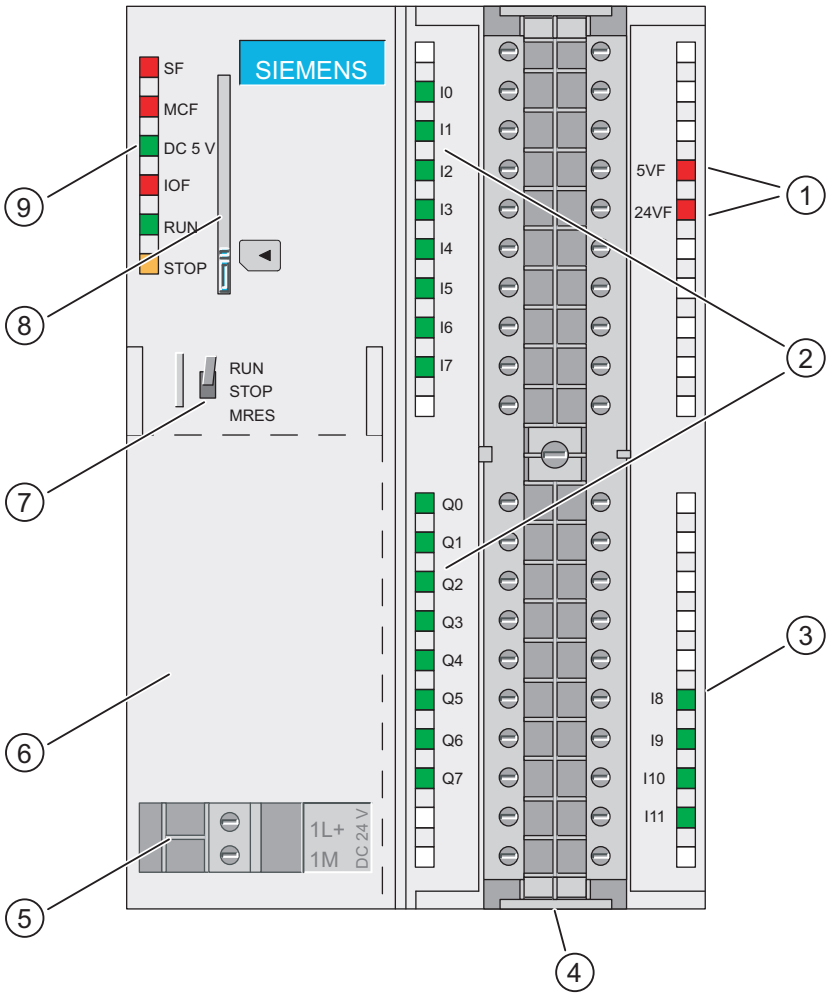
The FM 352-5 module has the following operating characteristics:

- Recording and control of fast processes (for example, high-speed inspection & rejection systems, or control of high-speed machines in the packaging, food & beverage, tobacco, and personal care product industries).
- Data exchange with the user program of the CPU (when using a coprocessor configuration). The S7 CPU has access to 16 bytes of input and 16 bytes of output data to permit transfer of control information, counted values, counter preset values, and status information using a special Interface FB (Function Block) to coordinate the data exchange (see figure above).

1.2 Physical features of the module

Elements on the front

The following figure shows the status indicators on the front of the FM 352-5 module.



(1) Status LEDs for output voltage
(2) Status LEDs for inputs/outputs
(3) Status LEDs for 24 V encoders or digital inputs
(4) Removable terminal strip
(5) Removable connector for the 24 VDC module power supply
(6) Hinged front panel
(7) Three-position switch for setting the mode
The memory-reset position (MRES) is spring-loaded with no detent.
(8) Slot for SIMATIC Micro Memory Card
The SIMATIC Micro Memory Card is inserted into this slot.
(9) Status LEDs for the module

Figure 1-2 Main features of the FM 352-5 module

Other physical features

Other features found on the module as shown in the figure include the following:

- Three-position switch to set the operating mode of the module
- Slot for the SIMATIC Micro Memory Card (non-volatile memory)
- Removable terminal connector for wiring inputs and outputs

Front connector

The removable front connector allows the following connection options:

- 24 V digital inputs: 8 inputs (up to 12 inputs if the 24 V encoder is not connected)
- 24 V digital outputs: 8 outputs
- Connectors for 24 V power supply
- Encoder signals: A differential incremental encoder (RS-422), an SSI absolute encoder, or a single-ended 24 V incremental encoder (HTL)
- 5 V and 24 V connectors to supply power to the encoders

Wiring diagram

A simplified wiring diagram is provided on the inside of the hinged front panel.

Labeling strip

A labeling strip is supplied with the module. You can use this to label the signals connected to the terminal connector. The labeling strip is inserted into the recess on the front of the hinged panel.

SIMATIC Micro Memory Card

The SIMATIC Micro Memory Card stores the program files in non-volatile memory. The SIMATIC Micro Memory Card is inserted in the slot on the front of the FM 352-5 module. A SIMATIC Micro Memory Card with 128 KB, 512 KB, or 2 MB of storage capacity is required to operate the FM 352-5 module.

The program files are downloaded from the SIMATIC Micro Memory Card to the FPGA during startup or after a memory reset.

NOTICE
Use a new or reformatted SIMATIC Micro Memory Card If the SIMATIC Micro Memory Card already contains data not associated with an FM 352-5 (for example, S7 CPU data), this can lead to the error message stating that the SIMATIC Micro Memory Card is defective when the FM 352-5 starts. Therefore, ensure that the SIMATIC Micro Memory Card is new or reformatted before using it in the FM 352-5.

1.3 System configurations

Description

The following figure shows some possible system configurations with the FM 352-5. The control program is developed in the STEP 7 environment with the FM 352-5 configuration software. The FM 352-5 module can operate in the following configurations: (1) in an S7 system, (2) in a stand-alone configuration, or (3) in a distributed system (with an S7 or non-S7 master) using PROFIBUS communication.

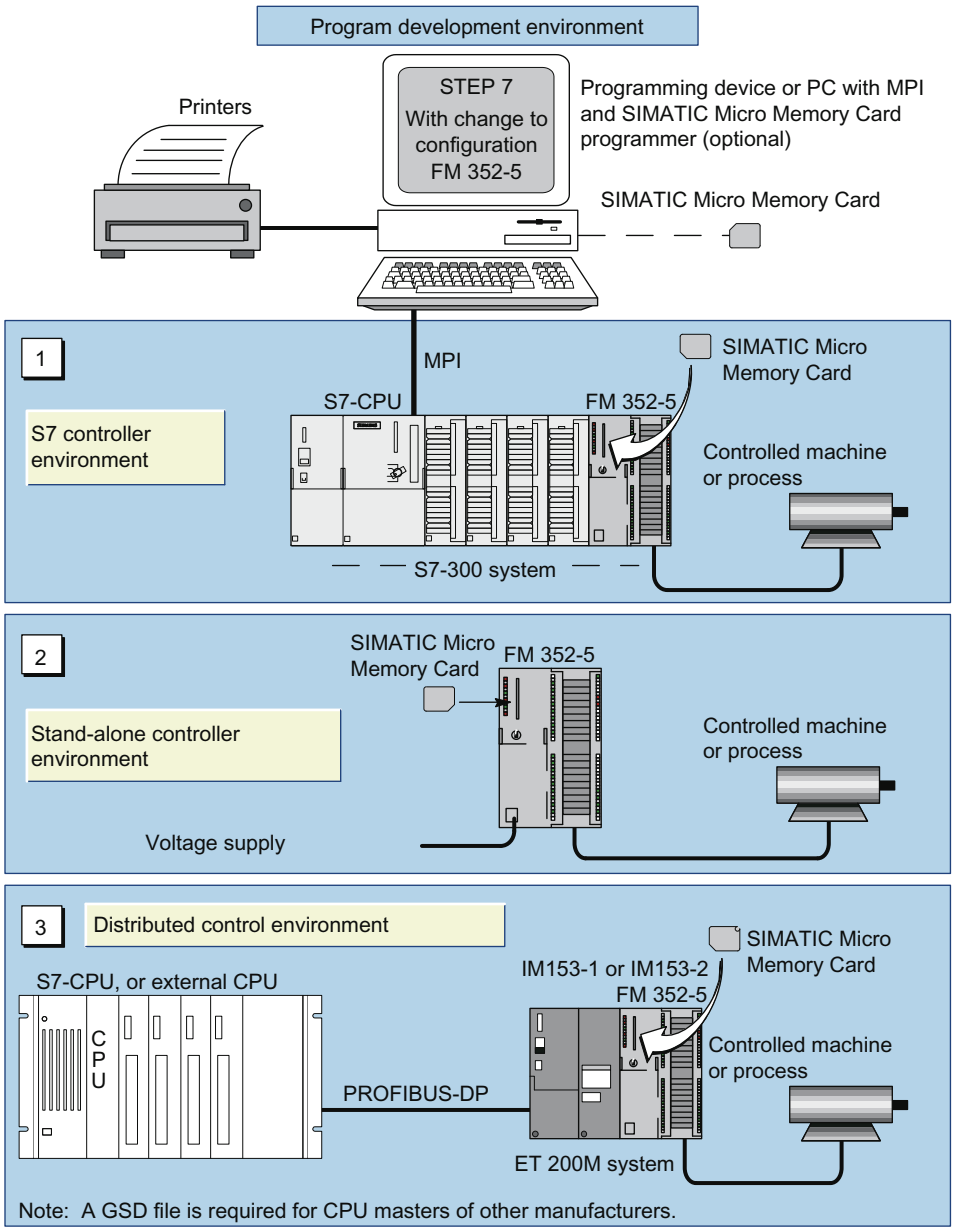


Figure 1-3 Examples of system configurations

1.4 Modes of operation

Test mode

The test mode is used to test your application program before putting the FM 352-5 module into operation. In Test mode, you can use the program monitoring and test tools available in STEP 7. This Test mode is possible only with an S7 CPU (S7-314 or higher due to memory restrictions) or the S7 PLC Simulator (S7-PLCSIM). The following figure shows the FM 352-5 in a test configuration.

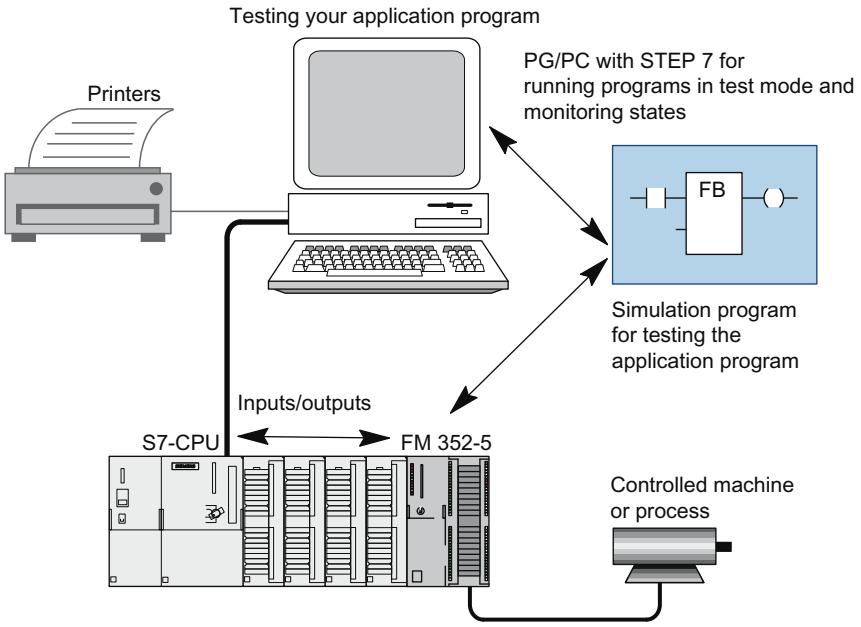


Figure 1-4 System configuration for testing your program

In Test mode, the S7 CPU executes the application FB, while the FM 352-5 module makes its inputs and outputs directly available to the S7 CPU allowing you to simulate the program at lower speed and check the wiring.

Normal mode

After fully testing the application program in Test mode, you compile the program to an FPGA image and download the program and module parameter data to the module. You can then put the FM 352-5 module into Normal mode operation.

If a master CPU is controlling the FM 352-5 module, the main control program signals the FM 352-5 to begin RUN mode or go to STOP mode via the Interface FB as long as the mode selector switch on the module is set to RUN.

In a stand-alone configuration, the module executes its program when you power up the module and set the selector switch to RUN.

Response time during program execution

As mentioned above, the response time of the FM 352-5 is extremely fast. In normal mode, the response time is measured as the elapsed time from the change at an input until the setting of an output.

The calculated response time consists of the following components:

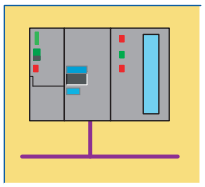
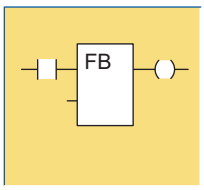
- Input delay (circuit delay + filter delay)
- Program execution time (1 µs)
- Output delay

1.5 Overview of the main tasks

Overview

The following table provides an overview of the basic tasks required to install, configure, program, and operate the FM 352-5 module when configured to operate in an S7 system.

Table 1- 1 Basic tasks to set up and operate the FM 352-5

Installing and configuring the FM 352-5 module • Install the FM 352-5 in an S7 station • Wire the FM 352-5 • Configure the properties and parameters	
Programming the FM 352-5 module • Create application FB/DB. • Test user program with STEP 7 • Compile the program and download it to FM 352-5 and SIMATIC Micro Memory Card.	
Troubleshooting • Check status LEDs. • Read diagnostics. • React to interrupts.	

Getting started

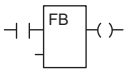
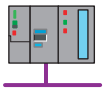
2

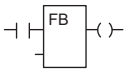
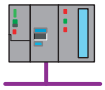
2.1 Overview of getting started

Overview

The following table provides an overview of the tasks needed to run the sample program for the FM 352-5 module.

Table 2- 1 Getting started

	**Running the sample program**	
	Install and configure the module. [1)] 1. Install the hardware components and wiring. 2. Install the configuration software. 3. Create a STEP 7 project. 4. Configure the hardware. 5. Save and compile the hardware configuration.	
	Copy the "Getting Started" sample program objects from the Sample Projects directory to your program.	
	Configure the FM 352-5 module parameters. 1. Set basic parameters as described. 2. Compile the parameters and program. 3. Compile the hardware configuration.	
	Prepare the S7 CPU to execute the sample program. 1. Download the program to the S7 CPU. 2. Set the CPU switch to RUN. 3. Set the FM 352-5 module switch to RUN.	

	**Running the sample program**	
	Run and monitor program in Test mode. 1. Initiate Test/Run mode by using the VAT table as described. 2. Monitor program execution by observing the behavior of module LEDs and the VAT table status indicators.	
	Switch from Test to Normal mode. 1. Download the program to the FM 352-5 module. 2. Initiate Normal/Run mode by using the VAT table as described. 3. Monitor program execution in Normal mode as above.	
[1] See chapter "Installing and removing the FM 352-5 (Page 27)", "Wiring the FM 352-5 (Page 31)" and "Configuring the FM 352-5 module (Page 45)".		

2.2 Running the FM 352-5 sample program

"Getting Started" application example

When you install the FM 352-5 software package, a sample project is also installed in the STEP 7 "Sample Projects" folder. The English sample project is in the following folder:

- ...\STEP7\EXAMPLES\zEn29_01

The sample program will help you become familiar with the steps needed to get a program running on the FM 352-5 module. The Blocks folder contains the components for a "Getting Started" function block that you can copy to your STEP 7 project. You can then compile and download to your system to watch a working program execute.

Note

The project contains two application FBs: FB3 is a simple "Getting Started" example and FB10 is a more complex example that uses many of the operations available for the FM 352-5 module.

Installing and configuring the hardware

Follow the steps outlined below to set up the project and configure the hardware for the "Getting Started" application example.

1. Install the FM 352-5 module in a local rack with an S7-3xx CPU.
2. Apply power to the CPU and the 1L and 2L connections on the FM 352-5 module.
3. Install the FM 352-5 configuration and programming software, as described in the section "Installing the configuration/programming software (Page 45)".
4. Create a STEP 7 project (see the section "Setting up the hardware configuration (Page 50)").
5. Create the hardware configuration (se the section "Setting up the hardware configuration (Page 50)") to match the S7-300 CPU and FM 352-5 module as installed in Step 1 above.
6. Save and compile the hardware configuration by selecting the menu command Station **>** Save and Compile.

Setting up a project

1. In the SIMATIC Manager, open the directory with the sample projects and copy the following objects from the "zEn29_01_FM352-5_Prog" Blocks folder to the Blocks folder of the S7 CPU: OB1, OB40, FB3, FB30, FB31, FB113, FB114, FB119, DB3, DB5, DB6, DB30, DB31, VAT_1 and SFC64.
2. Copy the diagnostic interrupt OB, OB 82, to your program. Use the S7 command: "**Insert > S7 Block > Organization Block > OB82"**.
3. Copy the Symbols object from the sample program to your S7 program folder of the S7 CPU.

Configuring the module parameters

1. Return to HW Config and double-click on the FM 352-5 to open the Properties dialog for the FM 352-5 module.
2. Select the Addresses tab and assign the input and output addresses.

   Note: The sample program uses address 256 in FB30 and FB31 for the inputs and outputs. If you select a different address, you will need to change the address parameters in FB30 and FB31 to match what you have selected.
3. Select the "Parameters" tab.
4. Open the Basic Parameters folder and click the check box to enable "Interrupt generation". Then select "Hardware Interrupts". Then open the Hardware Interrupts Enable folder and click the check boxes to enable all 8 hardware interrupts.

Preparing to run the sample program

If the example application FB (FB3) is open, make certain you close it first, then continue with the following steps to download the "Getting Started" application example to the S7 CPU.

1. Select the "Programming" tab and click the "Compile" button to compile the FM program (FB3). Click "OK" in the information dialog and then click "OK" to close the "FM 352-5 Properties" dialog.
2. From the HW Config window, select the menu command Station **>** Save and Compile to save and compile the entire hardware configuration.
3. From the SIMATIC Manager, download the entire Blocks folder of the S7 CPU (including the system data) to the S7 CPU.
4. Set the mode selector on the CPU to the RUN position and that of the FM 352-5 module to the RUN position. Watch the status LEDs on each module, and note that the CPU changes to RUN, but the FM 352-5 remains in STOP. (The SF status LED is also on because the module is in STOP.)

Running a program in Test mode

1. Open the VAT_1 object.
2. Select the menu command "Variable **>** Monitor" or click the "Monitor Variable" button. Then select the menu command "Variable **>** Modify" or click the "Modify Variable" button in VAT_1.

   This sets the module mode to Test/RUN by setting the variable “Run” (M0.1) to "1". (Make sure that the “Normal/Test” variable M0.0 is set to “0” requesting Test mode.)

   The LEDs on the FM 352-5 module now indicate that the module has changed to RUN.

Monitoring program execution in Test mode

With the FM 352-5 module in RUN mode, you can monitor execution of the sample program. In Test mode, STEP 7 allows you to use all of its monitoring features to monitor the execution of FB3.

- Note that the LEDs for outputs Q6 and Q7 on the module start flashing at the rate of 2 Hz and 1 Hz, respectively. Each of these outputs is driven by a CP_Gen instruction.
- Outputs Q0 through Q4 on the module flash simultaneously, along with the corresponding CPU_In.Bits[0..4] in the VAT table.
- Interrupts 0 through 4 of the module (at addresses M7.0 through M7.4 in the VAT table) also flash simultaneously. These are driven by OB40 in response to hardware interrupts of the module.

Switching program execution to Normal mode

To switch to Normal mode, you have to download the program to the FM 352-5 module and start the interface FB for normal operation, as described above.

1. Return to HW Config and double-click on the FM 352-5 to open the "Properties" dialog.
2. Select the "Programming" tab and click the "Download" button.

   During the download to the FM 352-5, the RUN LED (green) blinks rapidly while the STOP LED (yellow) is on. Once the download has successfully completed, the FM 352-5 remains in STOP mode.
3. Switch the module execution mode to Normal by writing the value "1" to the M0.0 address in the VAT_1 table. The interface FB for normal operation sends a Run command to the module.

You can monitor the same program execution in normal mode as described in "Monitoring program execution in test mode" above.

Note

In Normal mode, FB3 is executed on the FM module, not on the S7 CPU. Consequently, you will not be able to monitor the execution of FB3 using STEP 7's display of signal flow in the logic block or using other monitoring functions.

Installing and removing the FM 352-5

3

3.1 Installation rules

Planning the mechanical installation

If you want to operate the FM 352-5 module in an S7-300 system, you can find detailed information on the mechanical installation and project engineering in the Operating Instructions SIMATIC S7-300 CPU 31xC and CPU 31x: Installation (http://support.automation.siemens.com/WW/view/en/13008499). This chapter simply contains additional information.

Refer also to the design guidelines regarding lightning protection in section External Protection Circuit for FM 352-5 Boolean Processor (Page 243).

The remainder of this section and the following section deal with installation in an S7-300 system. The last section describes installation in a stand-alone system.

Installing the mounting rail

Horizontal installation of the mounting rail is preferable.

If you install the mounting rail vertically, remember the restrictions on ambient temperature, a maximum of 40° C is permitted.

Project engineering and the mechanical configuration

If you want to configure the FM 352-5 module for operation in an S7-300 system, keep to the following rules when planning the mechanical installation of your controller system:

- The maximum number of modules is restricted by the length of the mounting rail and the width of the modules.

  The FM 352-5 takes up 80 mm of space.

- The number of modules that can be installed to the right of the CPU is limited by the sum of their current consumption from the S7-300 backplane bus.

  The current consumption of the FM 352-5 from the backplane bus is 100 mA.

- The FM 352-5 can be mounted at any location for I/O modules on the rail.

Tools

To install or remove the FM 352-5 module, you need a 4.5 mm slotted screwdriver. To wire the terminal connector block, you need a 3 mm slotted screwdriver.

3.2 Installing and removing the FM 352-5 in an S7-300 system

Installing the FM 352-5

The description below explains how to mount the FM 352-5 on the rail of an S7-300 controller system. For additional information on installing modules, refer to the Operating Instructions SIMATIC S7-300 CPU 31xC and CPU 31x: Installation (http://support.automation.siemens.com/WW/view/en/13008499).

1. Plug the bus interconnector onto the bus connector of the module to the left of the FM 352-5. (The bus connector is on the back of the module, and you may need to release the module first.)
2. If you want to install additional modules to the right, first plug the bus interconnector of the next module onto the right bus connector of the FM 352-5.

   If the FM 352-5 is the last module in the row, do not fit a bus interconnector.
3. Hook the module onto the rail, slide it as far as the module on the left, and push it into place at the bottom.
4. Tighten the two screws on the bottom of the FM 352-5, applying a torque of between 0.8 and 1.1 Nm to secure the module to the rail.
5. After installing the module, you can assign a slot number to the FM 352-5. Slot labels are supplied with the CPU.

   Refer to the Operating Instructions SIMATIC S7-300 CPU 31xC und CPU 31x: Installation (http://support.automation.siemens.com/WW/view/en/13008499) for instructions on how to assign and apply slot numbers to the modules.

Removing the FM 352-5

The description below explains how to remove the FM 352-5 from the rail of an S7-300 controller system. For additional information on removing modules, refer to the Operating Instructions SIMATIC S7-300 CPU 31xC und CPU 31x: Installation (http://support.automation.siemens.com/WW/view/en/13008499).

1. Set the CPU mode selector to STOP.
2. Turn off or disconnect all power to the FM 352-5 module.
3. Open the hinged front panel on the right-hand side of the module.
4. Unscrew the securing screw of the front connector with a 3-mm screwdriver and then pull it out while holding the grips at the top and bottom. Pull firmly to release the catches.
5. Remove the group 1 power supply connector behind the panel on the left-hand side of the module. The connector block can be taken out.
6. Unscrew the two screws securing the module at the bottom using a 4.5-mm screwdriver.
7. Swing the module up and off the rail.

3.3 Installing in a stand-alone system

Mechanical installation

For a stand-alone system, it is recommended that you keep to the same basic installation guidelines and mechanical requirements that are specified for an S7-300 system. This installation system meets the safety requirements and provides the grounding, mechanical support, and resistance to vibration to help ensure proper operation of the FM 352-5 module.

For additional information on assembling DIN rails and installing modules, refer to the Operating Instructions SIMATIC S7-300 CPU 31xC und CPU 31x: Installation (http://support.automation.siemens.com/WW/view/en/13008499).

Note

If the FM 352-5 module detects that another module with an S7-300 bus connector is connected next to it on the rail, the FM 352-5 module will not change to stand-alone mode. Stand-alone operation is possible only when no bus connector is connected on either side of the FM 352-5 module.

Providing the power supplies

If you use the S7-300 rail for your stand-alone installation, you can connect an S7-300 power supply to the rail to provide power for the internal module electronics. Wire the S7-300 power supply to the 1L/1M, 2L/2M, and 3L/3M power terminals of the FM 352-5 module.

Otherwise, you will need to provide power to the module using an external 24 V DC power supply connected to the 1L/1M, 2L/2M, and 3L/3M power terminals. A removable connector is supplied with the module to simplify installation and removal of the power supply wiring.

Refer to the following chapter for more information on wiring the external power supplies.

Wiring the FM 352-5

4

4.1 General rules and regulations

Introduction

When operating the FM 352-5 as a component part of a plant or system, certain rules and regulations have to be followed depending on where the device is to be used.

This chapter provides an overview of the most important rules to remember and keep to when integrating the FM 352-5 in a plant or system.

Specific applications

Keep to the safety and accident prevention regulations applying to specific applications (for example, the directives on machines).

Emergency stop devices

Emergency stop devices complying with IEC 204 (which corresponds to DIN VDE 113) must remain effective in all the operating modes of the plant or system.

Startup of the system after specific events

The following table lists the measures to be taken when the system starts up after the occurrence of certain events.

Situation	Measure
Startup following drop or failure of the power supply Restart of the FM 352-5 following an interruption of bus communication	Dangerous operating states must not occur. If necessary, force an emergency stop.
Restart after release of the emergency off device	No uncontrolled or undefined restart must be possible.

Line voltage

The following table lists the measures to be taken regarding the line voltage.

Object	Guidelines
Permanently installed plants or systems without all-pole line disconnect switches	There must be a line disconnect switch or a fuse in the building installation system.
Load power supplies, power supply modules	The set rated voltage range must match the local power supply.
All circuits of the FM 352-5	Any fluctuations in the line voltages or deviations from the rated value must be within the permitted tolerances.

24 V DC supply

The following table lists the measures to be taken regarding the 24 V DC power supply.

Object	Measures	
Building	Outdoor lightning protection	Take lightning protection precautions (for example, lightning conductors)
24 V DC supply cables, signal lines	Indoor lightning protection	
24 V DC supply	Safe (electrical) extra-low voltage isolation	

Protection against outside electrical influences

The following table lists the measures to be taken to provide protection against electrical influences or faults.

Object	Measure
All plants or systems in which the FM 352-5 is integrated	Make sure that the plant or system is connected to a protective conductor for diverting electromagnetic interference.
Power supply, signal, and bus lines	Make sure that cabling and wiring is routed and installed correctly.
Signal and bus cables	Make sure than any wire or cable break does not result in undefined states in the plant or system.

See also

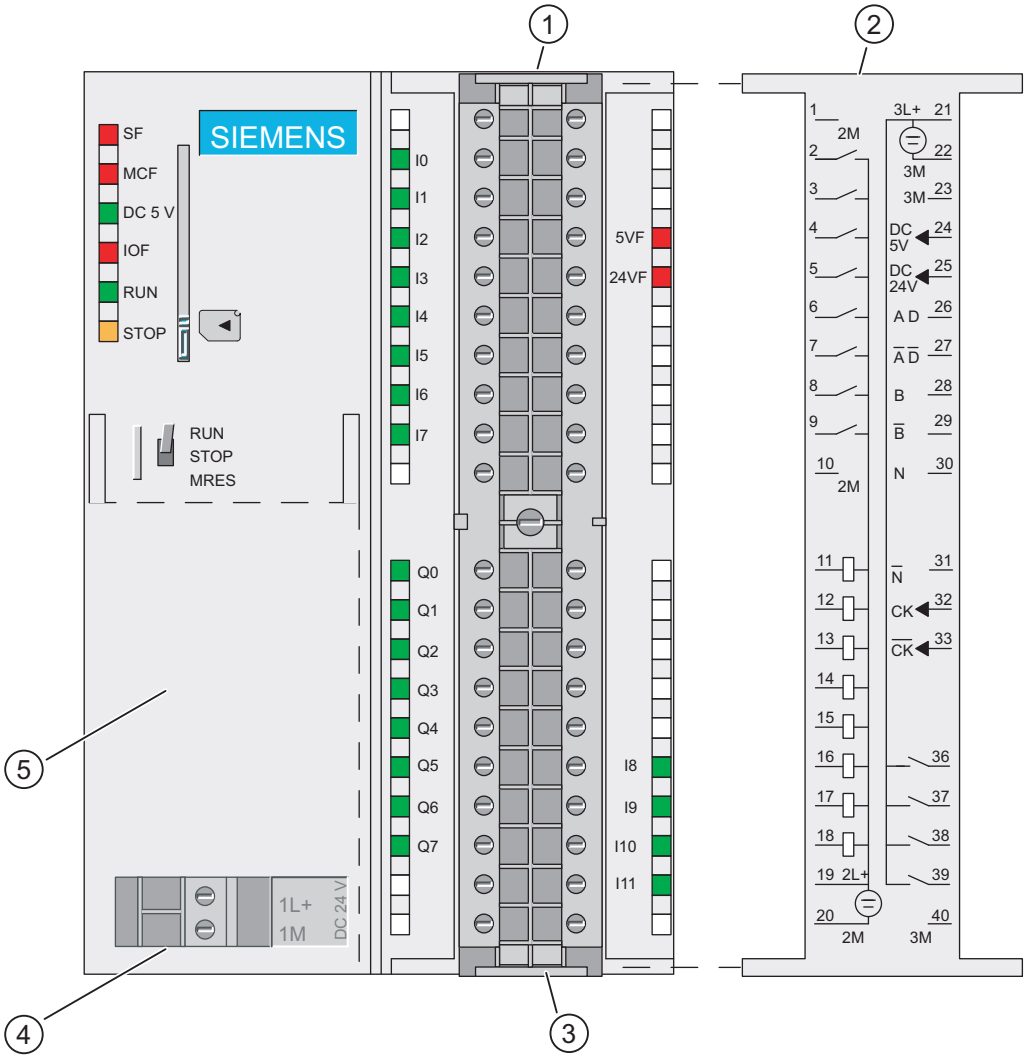
General technical specifications (Page 205)

4.2 Terminal assignments of the front connector

Terminal connector and terminal names

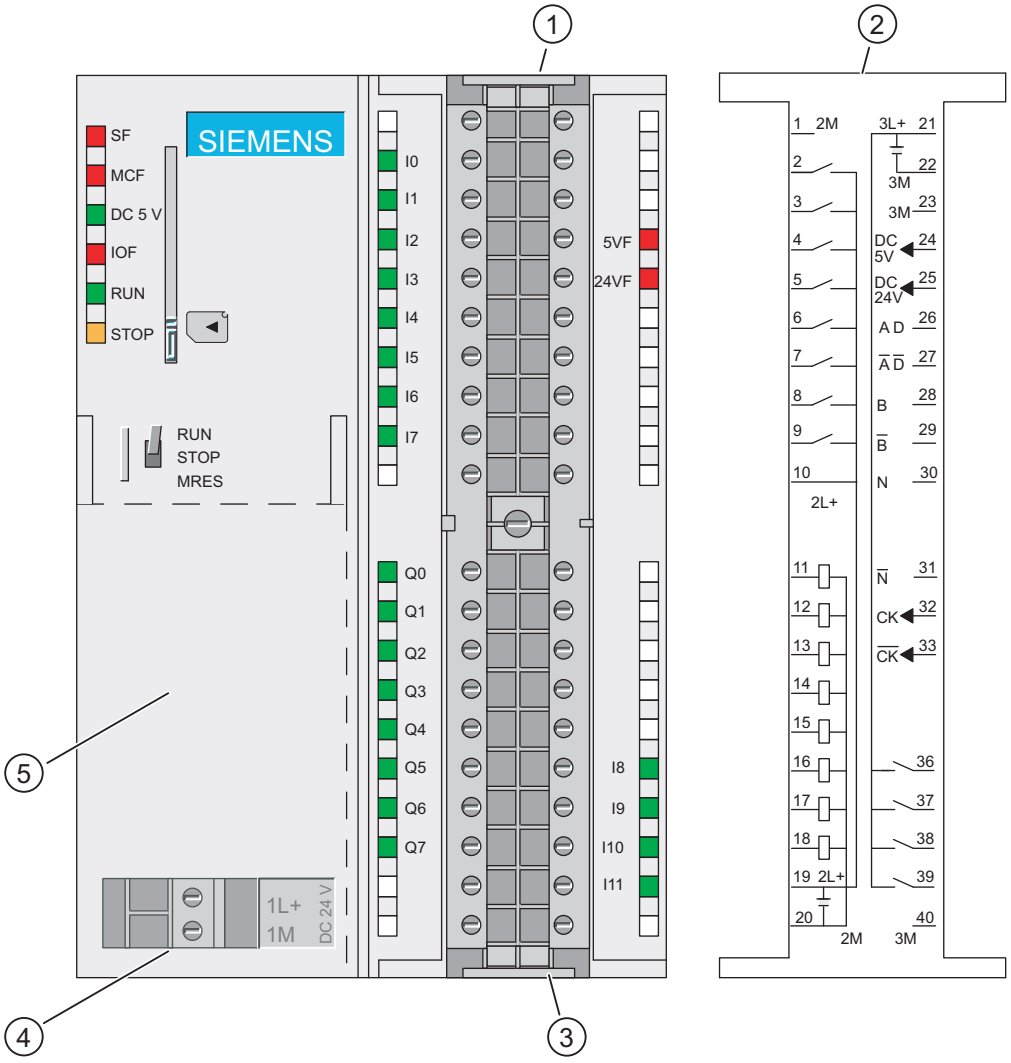
All inputs, outputs, encoder signals, and I/O power supply wiring are connected to the 40-pin terminal connector located behind the hinged panel. At the bottom left of the module, behind a hinged panel, you will find the 1L+ and 1M terminals for the 24 V DC power supply wiring for the internal electronics of the module. This connector along with the 2L+/2M terminals represent the minimum wiring required to commission the FM 352-5 module.

The following figures show the front panel of the module, the removable terminal strip and the inside of the connector cover with the terminal labels.



(1) Removable terminal strip
(2) Wiring diagram on the inside of the front panel door
(3) Strain-relief mount
(4) Removable connection for the 24 VDC module power supply
(5) Hinged front panel

Figure 4-1 Front connector of the FM 352-5AH01 module (outputs, low)



(1) Removable terminal strip

(2) Wiring diagram on the inside of the front panel door

(3) Strain-relief mount

(4) Removable connection for the 24 VDC module power supply

(5) Hinged front panel

Figure 4-2 Front connector of the FM 352-5AH11 module (outputs, high)

Terminal connector assignment

The following table lists all terminals on the left side of the terminal connector (pins 1 through 20) and the assignment for each terminal.

Table 4- 1 Terminal connector, assignments of terminals 1 through 20

Terminal no.	I/O	Name	Function	LED
1		2M	Ground for section 2 – I/O circuits	—
2	Input	I 0	Input	Green
3	Input	I 1	Input	Green
4	Input	I 2	Input	Green
5	Input	I 3	Input	Green
6	Input	I 4	Input	Green
7	Input	I 5	Input	Green
8	Input	I 6	Input	Green
9	Input	I 7	Input	Green
10		Note[2]	Section 2 – I/O circuits	—
11	Output	Q 0	Sourcing/sinking output[1]	Green
12	Output	Q 1	Sourcing/sinking output[1]	Green
13	Output	Q 2	Sourcing/sinking output[1]	Green
14	Output	Q 3	Sourcing/sinking output[1]	Green
15	Output	Q 4	Sourcing/sinking output[1]	Green
16	Output	Q 5	Sourcing/sinking output[1]	Green
17	Output	Q 6	Sourcing/sinking output[1]	Green
18	Output	Q 7	Sourcing/sinking output[1]	Green
19		2L+	Power for section 2 – I/O circuits	—
20		2M	Ground for section 2 – I/O circuits	—

[1]: FM 352-5AH01-0AE0 has sinking outputs.
FM 352-5AH11-0AE0 has sourcing outputs.

[2]: On the FM 352-5AH01-0AE0 module, terminal 10 is named 2M and serves as ground for section 2. On the FM 352-5AH11-0AE0 module, terminal 10 is named 2L+ and serves as power supply for section 2.

The following table lists all terminals on the right side of the terminal connector (pins 21 through 40) and the assignment for each terminal.

Only one encoder interface can be selected and operated at a time. If you select either SSI absolute encoders or 5-V differential incremental encoders (RS-422), then the 24-V inputs (terminals 36 through 39) are available for use as digital inputs (8 through 11). If you select no encoder interface, then terminals 26 through 31 are available for use as 5-V digital differential inputs (12, 13, and 14) in addition to the 24-V inputs (terminals 36 through 39).

Table 4- 2 Terminal connector, assignments of terminals 21 through 40

Terminal no.	I/O	Name	Encoder function				LED
			5-V encoders	SSI master	SSI listen	24-V encoders	
21		3L+	Power supply for section 3 - encoder circuits				—
22		3M	Ground for section 3 - encoder circuits				
23		3M	Ground for section 3 - encoder circuits				
24	Output	5V output	Encoder power supply 5.2 V				Red
25	Output	24V output	Encoder power supply 24 V				Red
26	Input	Encoders	Signal A	Master SSI D (data)	Listen SSI D (data)	I 12+	
27	Input	Encoders	Signal /A (inverse)	SSI /D (data inverse)	SSI /D (data inverse)	I 12-	
28	Input	Encoders	Signal B	I 13+	SSI CK (shift clock)	I 13+	
29	Input	Encoders	Signal /B (inverse)	I 13-	SSI /CK (shift clock inverse)	I 13-	
30	Input	Encoders	Signal N	I 14+	I 14+	I 14+	
31	Input	Encoders	Signal /N (inverse)	I 14-	I 14-	I 14-	
32	Output	Encoders	—	SSI CK (shift clock)	—	—	
33	Output	Encoders	—	SSI /CK (shift clock inverse)	—	—	
34	—	—	—	—	—	—	
35	—	—	—	—	—	—	
36	Input	I 8	I 8	I 8	I 8	I 8	Green
37	Input	I 9	I 9	I 9	I 9	Signal A	Green
38	Input	I 10	I 10	I 10	I 10	Signal B	Green
39	Input	I 11	I 11	I 11	I 11	Signal N	Green
40		3M	Ground for section 3 - encoder circuits				

4.3 Wiring the FM 352-5 module

Wiring Front Connectors

To attach the signal wires of your process to the terminal connector of the FM 352-5 module, follow the steps outlined below:

1. If you want to route the wires out at the bottom of the module, start at terminal 40 or 20. Connect the wires to the terminals in alternating order; in other words, terminals 39, 19, 38, 18, and so on to terminals 21 and 1 at the top of the terminal strip.

   If you want to route the wires out at the top of the module, start at terminal 1 or 21. Connect the wires to the terminals in alternating order; in other words, terminals 2, 22, 3, 23, and so on to terminals 20 and 40 at the bottom of the terminal strip.
2. Always tighten the screws of the unused terminals.
3. Attach the cable strain-relief assembly around the bundle of wires and the strain-relief anchor at the top or bottom of the front connector.
4. Tighten the pressure clamp of the strain-relief. Push the retainer on the strain-relief assembly in to the left; this will improve utilization of the available space.
5. Insert the front connector into the recessed slot in the front of the module. Rail guides are keyed to prevent the terminal block from being inserted upside down.
6. Tighten the screw from the middle of the front connector. This ensures that the front connector is properly seated and connected to the terminal pins in the module.
7. Close the front panel.
8. Use the labeling strip to identify the signal of each wire connected to the front connector.
9. Slide the labeling strip into the guides on the front door.

Wiring the power supplies

Power supply 1L provides 5 V DC power for the module's internal electronics. Connect your 24 V DC power supply to the 1L and 1M terminals on the bottom left side of the module behind the hinged panel.

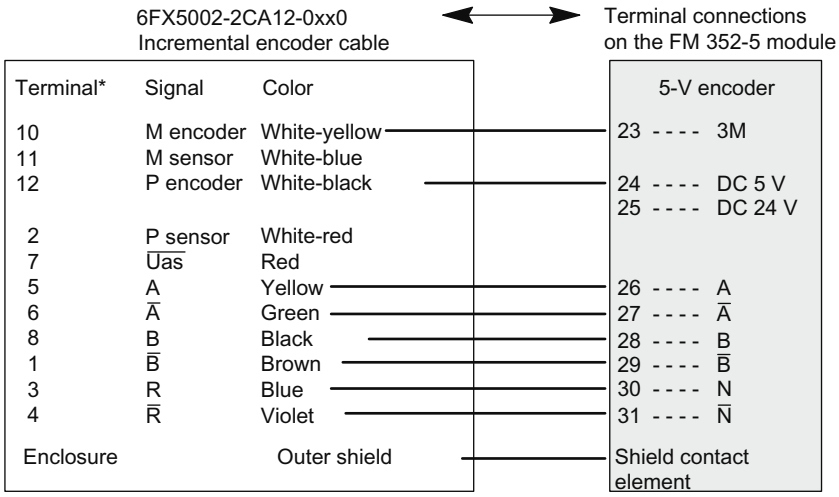
Power supply 2L powers the input and output circuits (I 0 to I 7 and Q 0 to Q 7) in the module. Connect your 24 V DC power supply to the 2L and 2M terminals to provide this power source.

Power supply 3L powers the encoder interface circuits (I 8 to I 14). It also provides a 24 V and a 5.2 V current-limited supply to power the encoders. Only one of the output supplies can be used at a time. Connect your 24 V DC power supply to the 3L and 3M terminals to provide this power source.

4.4 Connecting encoder cables

Description

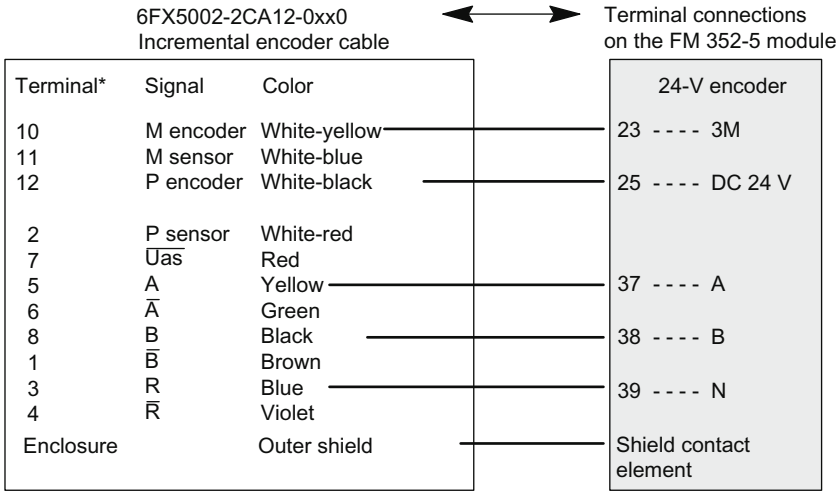
The following figure shows the pin assignment for an incremental encoder cable available from Siemens and the corresponding connections to the front connector on the FM 352-5 for the 5 V encoder interface. The last four characters of the order number specify the cable length.



* The pin assignment of the encoder connector corresponds with that of connector 6FX2003-0CE12 for encoders 6FX2001-2xxx.

Figure 4-3 Pin assignment of the incremental encoder cable for 5 V encoders (RS-422)

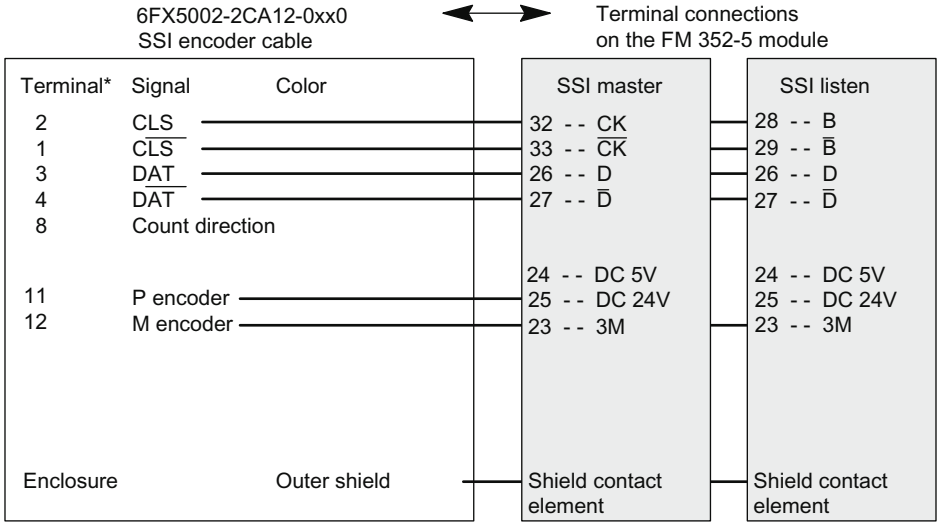
The following figure shows the pin assignment for an incremental encoder cable available from Siemens and the corresponding connections to the front connector on the FM 352-5 for the 24 V encoder interface. The last four characters of the order number specify the cable length.



* The pin assignment of the encoder connector corresponds with that of connector 6FX2003-0CE12 for encoders 6FX2001-2xxx.

Figure 4-4 Pin assignment of the incremental encoder cable for 24 V encoders (HTL)

The following figure shows the pin assignments for an SSI encoder cable available from Siemens and the corresponding connections to the front connector on the FM 352-5 for the SSI absolute encoder interface. The last four characters of the order number specify the cable length.



* The pin assignment of the encoder connector corresponds with that of connector 6FX2003-0CE12 for encoders 6FX2001-2xxx.

Figure 4-5 Pin assignment for the SSI encoder cable for SSI encoders

The SSI encoder interface can support a maximum of one master and one listen module.

Note

Supply your encoder with 5 V DC or with 24 V DC from the FM 352-5 master module depending on the power supply required by your encoder.

If the SSI Master or SSI Listen device is not an FM 352-5 module, connect the wiring to the device as recommended in the user manual of the device in question.

4.5 Connecting shielded cables via a shield contact element

Application

Using the shield contact element you can easily connect all the shielded cables of S7 modules to ground by directly connecting the shield contact element to the rail.

Design of the Shield Contact Element

The shield contact element consists of the following parts:

- A fixing bracket with two bolts for attaching the shield clamps to the rail (order no.: 6ES7390-5AA00-0AA0)
- Shield clamps

Depending on the cable cross-sections used, use one of the shield clamps listed in the following table.

Table 4- 3 Cable cross-sections and terminal elements

Cable / shield diameter	Shield terminal Order no.:
Two cables, each with a shield diameter of 2 to 6 mm	6ES7390-5AB00-0AA0
One cable with a shield diameter of 3 to 8 mm	6ES7390-5BA00-0AA0
One cable with a shield diameter of 4 to 13 mm	6ES7390-5CA00-0AA0

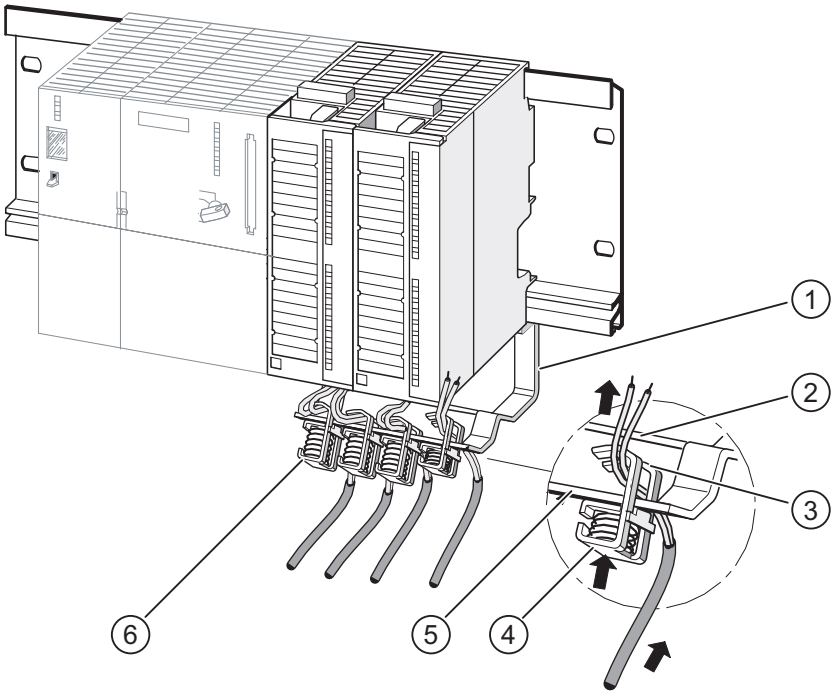
The shield contact element is 80 mm wide with space for two rows each with 4 shield clamps.

Installing the Shield Contact Element

Install the shield contact element as follows:

1. Push the two bolts of the fixing bracket into the guide on the underside of the rail. Position the fixing bracket under the modules to be wired.
2. Bolt the fixing bracket tightly to the rail.
3. The bottom of the shield terminal has a forked guide. Place the shield terminal at this position onto edge A or edge B of the fixing bracket. Press the shield terminal down and swing it into the desired position (see figure below).

   You can attach two rows each with up to four shield clamps to the shield contact element bracket.



(1) Bracket for shield contact element
(2) Edge B
(3) Forked guide
(4) The shield must be located beneath the shield terminal
(5) Edge A
(6) Shield terminal

Figure 4-6 Attaching Shielded Cables to Shield Contact Element

Attaching Cables

You can secure up to two shielded cables per shield terminal (see figure and table above). The cable is connected by its bare cable shield. There must be at least 20 mm of bare cable shield projecting. If you need more than 4 shield clamps, start wiring the rear row of clamps of the shield contact element.

Note

Use a sufficiently long cable between the shield terminal and the front connector of the module. This allows you to remove the front connector without needing to release the shield terminal.

Configuring the FM 352-5 module

5

5.1 Installing the configuration/programming software

Contents of the CD-ROM

The CD-ROM for the FM 352-5 module contains the following:

- FM 352-5 hardware configuration software (including help files and compiler)
- FM 352-5 library of function blocks (FBs) and associated help files
- User manual in PDF format
- Sample programs

Hardware requirements

Read the information on this in the readme file on the CD-ROM.

Starting the Installation Setup

The setup program installs the software components in exactly the same way as STEP 7 and other STEP 7 components. Select the language you want to use for the installation and follow the instructions as they appear on screen.

FM 352-5 Function Block Library

After installing the software, you will find an FM 352-5 library of FBs in the program elements of the STEP 7 LAD/FBD editor. The FB library contains timers, counters, shift registers, and other operations that are intended for use only with the FM 352-5 module. Some of these FBs have 16-bit and 32-bit versions of the same function. You can also use some of the standard STEP 7 bit-logic operations, such as contacts and coils as you create your program.

When you have created a project in the STEP 7 environment for your control process, you can copy any of the FBs that you intend to use from the program elements to the "Blocks" directory of your project. You can also insert them later as needed while you are creating your program.

Using STEP 7 with the FM 352-5

To configure, program, and operate the FM 352-5 module, you use STEP 7 and the FM 352-5 configuration software to perform the following functions:

1. Set up the hardware configuration for your project.
2. Set the parameters of the FM 352-5
3. Create, edit, or test your control program.
4. Download the program to the FM 352-5 module:
   - First, the program is automatically copied to the SIMATIC Micro Memory Card.
   - The FPGA is then automatically loaded.
5. Set the operating mode of the PLC and/or the module.
6. Monitor the status of the running program.

5.2 Basic tasks at a glance

Overview

The following figure shows a simplified representation of the basic tasks and tools required to generate and download an application program for the FM 352-5 module.

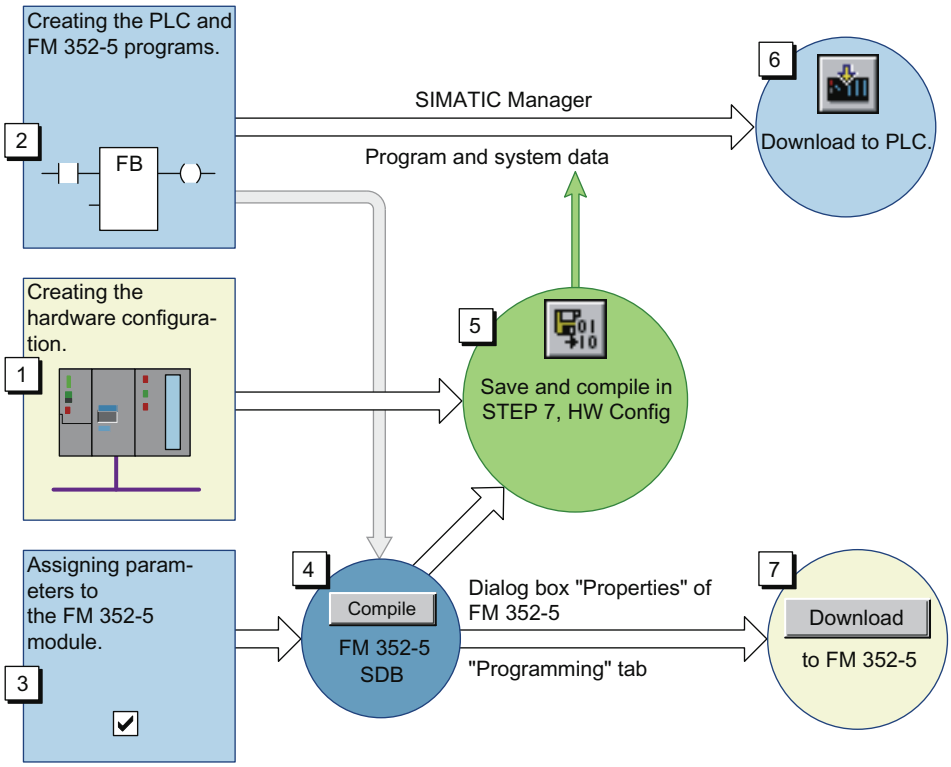


Figure 5-1 Overview of the tasks

These tasks are described in more detail below:

1. Create a hardware configuration in STEP 7 HW Config.
2. Create the application FB for the FM 352-5 module in the STEP 7 LAD/FBD editor, and create the FB call in the main program of the PLC.
3. Assign parameters to the FM 352-5 module in the "Properties" dialog.
4. Compile the application FB and hardware configuration in the FM 352-5 "Properties" dialog to generate an SDB for the FM 352-5 module.
5. Save and compile the hardware configuration in STEP 7 to generate a system data block for the CPU.
6. From STEP 7, download the program blocks and system data to the CPU.
7. From the FM 352-5 "Properties" dialog "Programming" tab, download the SDB containing the application FB and the module parameters, to the FM 352-5 module.

5.3 Checking the consistency of program and configuration

Consistency check

The "Consistency check" parameter in the "Properties" dialog box ("Parameters" > "Advanced Parameters" tab, see section "Assigning Properties and Parameters (Page 52)") prevents the wrong module program from being executed in a system that was configured for a different program. The module program and the configuration in the CPU must match to achieve a positive consistency check result. If the consistency check fails, a diagnostic error and an error in the status word of the module are reported.

The consistency parameter checks not only the program but also the hardware parameters that are known as **static** parameters. Other parameters, known as **dynamic** parameters, can be changed by the program control and do not affect the consistency check.

Ensuring Consistency

The sequence of tasks described in the previous section ensures that the consistency check will be successful. If you make any changes to the application FB or to the static parameters for the FM 352-5 module after you have followed the configuration and downloading procedures (see overview of the tasks), repeat steps 4, 5, 6, and 7 to restore consistency between the FM module and the PLC.

Maintaining Consistency

The FM 352-5 "Properties" dialog has a "Compile" button that creates a special SDB formatted for the FM 352-5 module. This special SDB is created from a combination of the application FB and the static parameters. If you make any changes to the static parameters or any changes to the application FB, you will need to recompile to generate the correct consistency. Changes made to the dynamic parameters do not make it necessary to recompile the FM 352-5 program, but the changed hardware configuration must be downloaded to the S7 CPU. If you transfer a program from a module in one system to another, you can copy the module hardware configuration from one system to the other system and then compile. After the configuration is downloaded to the CPU in the new system, you can insert the SIMATIC Micro Memory Card containing the module's program, power up the new FM 352-5 module, and execute the program. This maintains the consistency between the CPU and the module program. If the hardware configuration of one system is different from the other, the consistency check will fail.

Note

You can disable the consistency check in the "Advanced Parameters" of the "Parameters" dialog. If the consistency check for the SIMATIC Micro Memory Card or for the system data block in the CPU is disabled, the consistency is not checked and any program will be allowed to run.

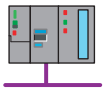
5.4 Overview of hardware configuration

Basic Steps for Installing and Configuring the FM 352-5 Module

The following table shows a summary of the basic steps required to install and configure the FM 352-5 module in an S7-300 system. (The FM 352-5 module can also be installed in a distributed system using an ET 200M station with an IM153-1 or IM153-2 module, but this chapter uses an S7-300 system as an example for the sake of simplicity.)

These steps are described in this chapter.

Table 5- 1 Installing and configuring the hardware

	**Creating the hardware configuration**
	Creating a new project (see chapter "Setting up the hardware configuration (Page 50)").
	Inserting a SIMATIC 300 station (see chapter "Setting up the hardware configuration (Page 50)"): • Inserting an S7-300 rack (mounting rail). • Inserting a power supply module. • Inserting the S7-300 CPU.
	Inserting the FM 352-5 module (see chapter "Setting up the hardware configuration (Page 50)").
	Configuring the FM 352-5 module (see chapter "Assigning properties and parameters (Page 52)" and "Selecting diagnostic parameters (Page 55)"): • Assigning the address and other basic properties. • Configuring the parameters for diagnostic interrupts. • Configuring the parameters for modes.
	Saving and compiling the hardware configuration (see chapter "Saving and compiling the hardware configuration (Page 61)").

5.5 Setting up the hardware configuration

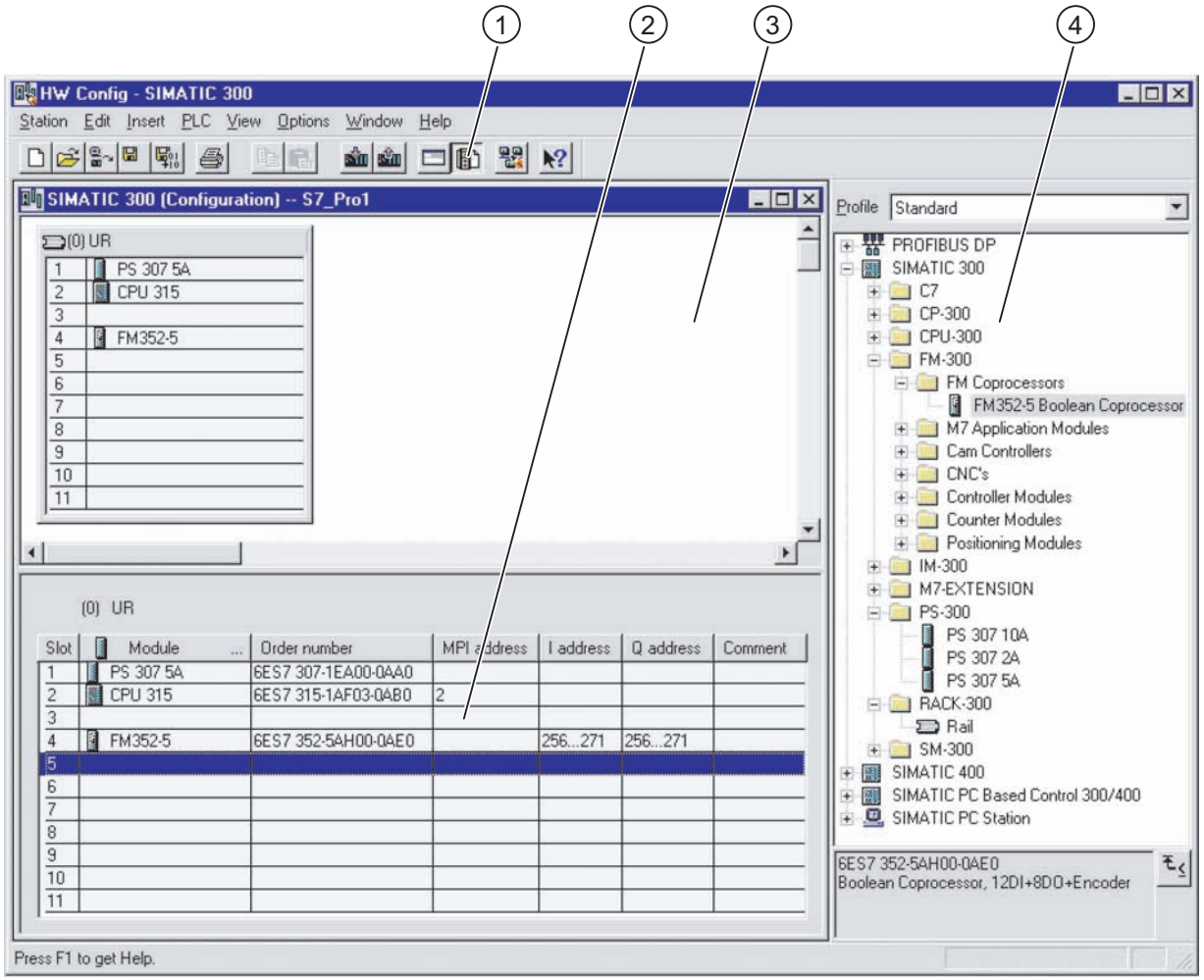
Creating a project

When you start STEP 7, the highest level in SIMATIC Manager is displayed. You can then either open an existing project or create a new one. For further information on creating a STEP 7 project, refer to the STEP 7 User Manual or the STEP 7 online help.

Accessing the Hardware Configuration

Double-click on the hardware icon in the project directory on the right-hand side to open hardware configuration.

The HW Config dialog is made up of three panes (see figure below):



(1) Click the catalog button if the hardware catalog does not appear when you first open HW Config.

(2) A table that provides details of each module placed in the selected rack, such as order numbers, network addresses, input and output addresses, etc.

(3) A blank station window in which you can place racks and insert modules into appropriate slots.

(4) A hardware catalog that contains all the S7 components needed to set up a programmable controller system.

Figure 5-2 Hardware configuration

Inserting an S7-300 station

Follow the steps outlined below to insert a SIMATIC S7-300 station:

1. Expand the SIMATIC 300 object in the hardware catalog.
2. Expand the RACK-300 folder.
3. Select an appropriate rack for your application.
4. Double-click on the rack or drag it to the station window.
5. Select and insert an appropriate power supply module from the PS-300 folder.
6. Select and insert an appropriate CPU from the CPU-300 folder.

Inserting an FM 352-5 module

Follow the steps outlined below to insert the FM 352-5 module in a SIMATIC S7-300 station:

1. Expand the FM-300 folder in the hardware catalog.
2. Expand the FM Processors folder.
3. Select the FM 352-5 Boolean processor module.
4. Select a valid slot in the rack and double-click on the module in the catalog or drag the module to a valid slot in the S7-300 station.

5.6 Assigning properties and parameters

Opening the "Properties" dialog

After the FM 352-5 module has been placed in a valid slot of the S7-300 station, you need to configure the module by assigning certain properties and parameters.

Double-click on the FM 352-5 module entry. This opens the "Properties" dialog that contains four tabs for assigning properties and parameters.

[1] The "General" tab displays the basic identification and descriptive information (see the figure below). You can also use this dialog to enter comments.

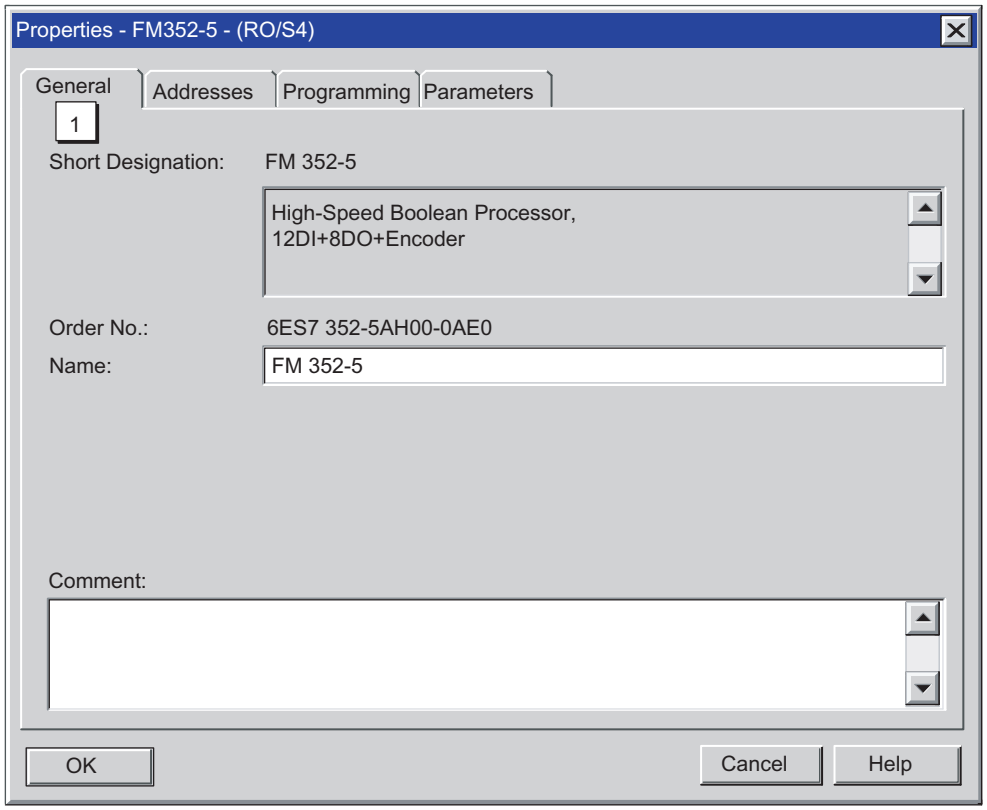
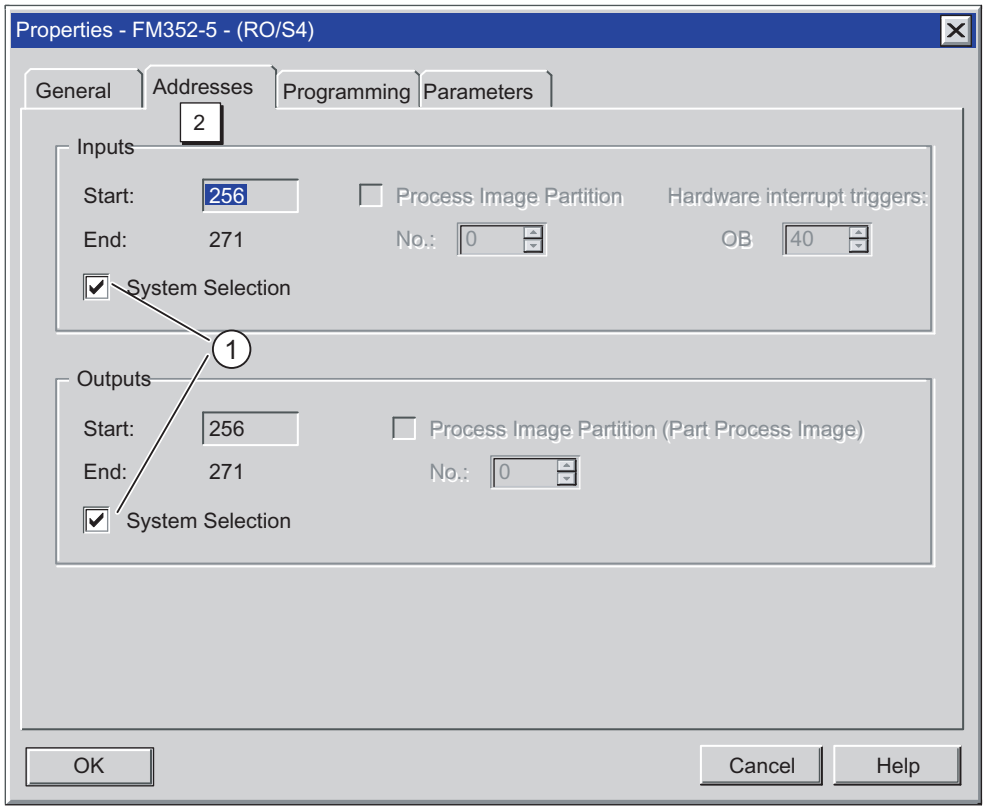


Figure 5-3 "Properties - FM 352-5" dialog, "General" tab

Setting Input and Output Addresses

[2] The "Addresses" tab displays the address assignments for the inputs and outputs as set by the system (see the figure below). You can change these addresses by unchecking the "System Selection" check box. The "Start" box can then be edited.



(1) Clear the checkbox to allow the start address to be changed (with CPUs that support address selection).

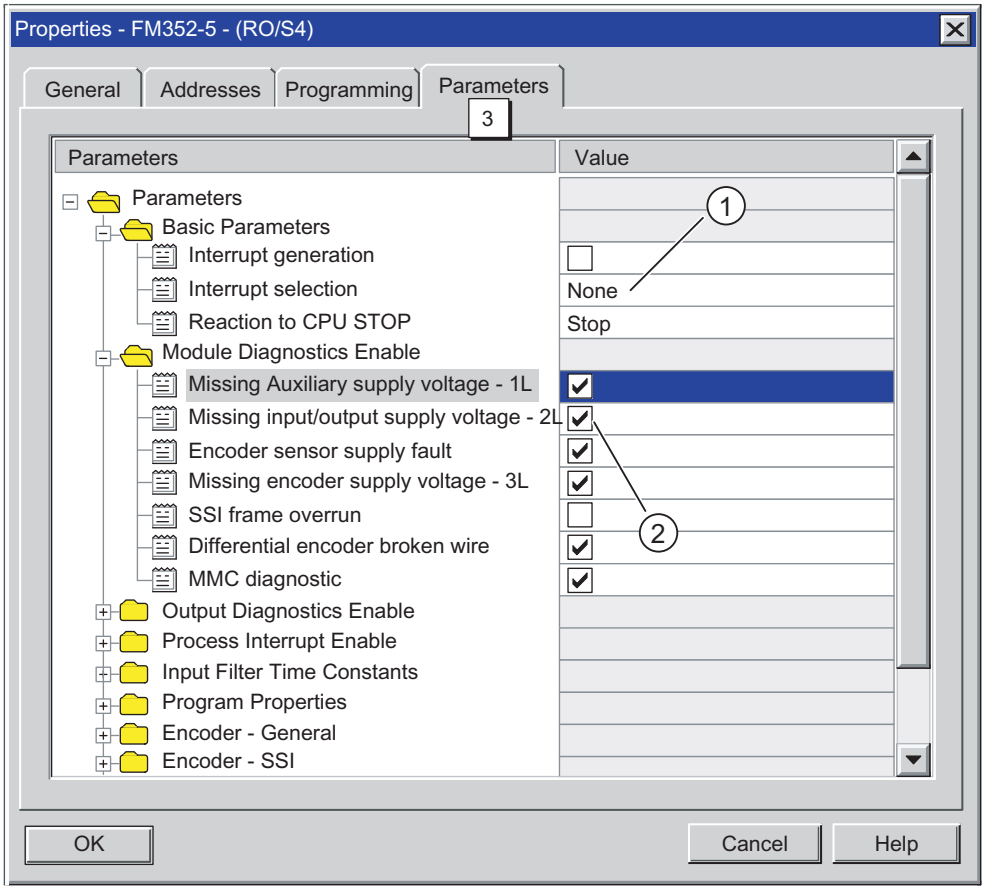
Figure 5-4 "Properties - FM 352-5" dialog, "Addresses" tab

Assigning module parameters

[3] The "Parameters" tab provides a hierarchical view of the different functions and diagnostics of the FM 352-5 module for which you can parameterize the operating states (see the figure below). The following parameters are involved and are described in the following tables:

- Enabling module diagnostics
- Enabling output diagnostics
- Enabling hardware interrupts
- Selecting input filter times
- Encoder parameters and others

Expand each folder in the left column to display the available parameter options. The column on the right changes as required to match the selected parameter. You assign parameters by selecting one of the available options. You can resize the columns in this dialog by moving the cursor to a position between the column headings. The following figure shows how to assign parameters.



(1) Click in field to display a list of parameter options.

(2) Click a check box to enable or disable each parameter or diagnostic interrupt.

Figure 5-5 "Properties - FM 352-5" dialog, "Parameters" tab

5.7 Selecting diagnostic parameters

Description

The following table provides a list of the module diagnostic and process alarms that can be set on the FM 352-5 module. These are **dynamic** parameters that can be changed by the program in RUN mode using SFC 55 to write data record 1 (see chapter "Controlling dynamic parameters (Page 106)"). These parameters are not part of the module consistency check, and can therefore be changed without generating a parameter assignment error.

Table 5- 2 Diagnostic interrupt parameters (dynamic)

Parameters	Description	Range of values	Default
Missing auxiliary supply voltage (1L)	1L power supply interrupt: reverse polarity, low voltage, internal fault, etc.	Enable, disable	Disabled
Missing input/output supply voltage (2L)	2L power supply interrupt: reverse polarity, low voltage, internal fault, etc.	Enable, disable	Disabled
Encoder sensor supply fault	Fault in the encoder power supply or wiring.	Enable, disable	Disabled
Missing encoder supply voltage (3L)	3L power supply interrupt: reverse polarity, low voltage, internal fault, etc.	Enable, disable	Disabled
SSI frame error	Incorrect frame size, power loss on the encoder, broken wire, etc.	Enable, disable	Disabled
Differential incremental encoder (RS-422) broken wire	Broken or disconnected cable, incorrect pin assignment, encoder malfunction, short-circuited encoder signals, etc.	Enable, disable	Disabled
MMC diagnostic	SIMATIC Micro Memory Card program missing or invalid, etc.	Enable, disable	Disabled
Output diagnostics*	Interrupts for outputs Q0 to Q7, individually enabled	Enable, disable	Disabled
Hardware interrupts	Hardware interrupts 0 to 7, individually enabled	Enable, disable	Disabled
* The FM 352-5 module can have an output ON time of less than 5 µs. To allow the FPGA to be able to respond to an output overload by setting the diagnostic bit, the pulse duration of the output ON time must be greater than 2 ms.			

Selecting Configuration Parameters

The following table provides a list of the configuration parameters that can be set on the FM 352-5 module. These are **static** parameters that specify how the module operates.

Note

These parameters are included in the module consistency check. The hardware configuration on the PLC and the hardware configuration in the SIMATIC Micro Memory Card of the FM 352-5 module must match to achieve a positive consistency check result. After having made any changes to the static parameters or to the application FB, you must recompile the data to generate the correct consistency (see chapter "Checking the consistency of program and configuration (Page 48)").

Table 5- 3 Configuration parameters (static)

Parameters	Range of values	Default
Interrupt generation	Enable, disable	Disabled
Interrupt selection	None, diagnostic interrupts, hardware interrupts, diagnostic and hardware interrupts	None
Reaction to PLC STOP [1)]	Stop, continue	Stop
Input filter time constants	Delays of 0, 5, 10, 15, 20, 50 microseconds, and 1.6 milliseconds delay (see following section for more information about input filtering)	0 microseconds
Stand-alone operation[2)] (under "Program properties")	Module stops with standalone, module can be operated standalone	Module stops with standalone
Encoder type selection	No encoder, SSI encoder, symm. 5-V incremental encoder (RS-422), single-sided 24 V incremental encoder (HTL)	No encoder interface
SSI encoders		
• SSI shift register length	13 bits, 25 bits	13 bits
• Clock rate	125 kHz, 250 kHz, 500 kHz, 1 MHz	125 kHz
• Delay time (monoflop)	16, 32, 48, 64 microseconds	Delay 64 µs
• Data shift direction	Left, right	Left
• Data shift	0 to 12 bits (number of bit positions by which data is shifted in the specified direction)	0 bits
• SSI mode	Master, listen	Master

Parameters	Range of values	Default
5-V and 24-V encoders		
• Signal evaluation	Pulse & direction, x1, x2, x4	Pulse/direction
• Counter type	Continuous, periodic or single	Continuous
• Counter size	16 bits, 32 bits	16 bits
• Source reset	None, HW, SW, HW and SW, HW or SW	None
• Source reset value	Constant 0, min/max value, load value	Constant 0
• Reset signal type	Edge, level	Edge
• Source load value	Constant, module application	Constant
• Source stop	None, HW, SW, HW and SW, HW or SW	None
• Load value (value loaded when load signal is active)	-2^{15} to 2^{15} -1 (16-bit counter) -2^{31} to 2^{31} -1 (32-bit counter)	0 0
• Count range Min (minimum count value)	-2^{15} to 2^{15} -1 (16-bit counter) -2^{31} to 2^{31} -1 (32-bit counter) (continuous: -32768 or -2,147,483,648)	0 0
• Count range Max (maximum count value)	-2^{15} to 2^{15} -1 (16-bit counter) -2^{31} to 2^{31} -1 (32-bit counter) (continuous: 32767 or 2,147,483,647)	32767 2.147.483.647
• Main count direction	Up count, down count	Up count
• Hardware source stop	Inputs 0 to 14	Inputs 8 (24 V)
• Hardware source reset	Inputs 0 to 14	Inputs 11 (24 V)
• Polarity of input A 3)	Active state = 0, active state = 1	Active status = 1
• Polarity of input B 3)	Active state = 0, active state = 1	Active status = 1
• Polarity of input N 3)	Active state = 0, active state = 1	Active status = 1

Parameters	Range of values	Default
Advanced parameters		
• Module diagnostics [4]	Enable, disable	Enabled
• Output diagnostics [4]	Enable, disable	Enabled
• Hardware interrupts[4]	Enable, disable	Enabled
• Consistency check [5]	Module checks for consistency, module ignores consistency	Module checks for consistency

[1] If the module is set to continue on PLC STOP and:

1. The consistency check is disabled:
   - The module continues on PLC STOP and stops if the PLC's static parameters do not match the static internal FM parameters.
   - The module continues if the parameterization is canceled by the PLC (for example, by deletion in the hardware configuration).
2. The consistency check is enabled:
   - The module continues on PLC STOP and stops if the parameters do not match, or if the module parameter assignment is canceled.

[2] When using outputs, you must also select the "Resume" option for the "Reaction to CPU STOP" parameter in the "Basic parameters" folder.

[3] The reset of the incremental encoder counter is activated by the N input, if the signal at the N input matches the polarity selected in HW Config, in other words, when the N input is in the active state. Alternatively, the reset can also be activated with any other digital input.

You can set this in HW Config by opening the "Properties" dialog box and by setting the desired digital input in the "Parameters" tab under the "Encoder - 5 V Differential Rotary Transmitter and 24-V Single Rotary Transmitter" folder for the parameter "Hardware source reset".

However, the reset is only executed when the inputs A and B have reached the active state.

[4] You release program memory resources by disabling the hardware support for any of these functions. For example, if your application program does not require hardware interrupts, you can disable the hardware support of hardware interrupts to gain more program memory. You must, however, use these advanced parameters with caution. Do not disable any of these diagnostic functions unless you are certain you will not need them in your program.

[5] Checks whether hardware configurations of FM and CPU match (see chapter "Checking the consistency of program and configuration (Page 48)").

5.8 Selecting input filters

Description of Filter Behavior

The filters of the FM 352-5 module are noise filters. Noise pulses are filtered out of the input signal if the noise pulse is shorter than the delay time. Pulses that are equal to the delay time or longer are allowed through to the program. The filters delay the input signal for the delay time.

The input delay for a given input will be determined by the input type, the voltage oscillation of the signals, the time an input is held active or inactive and the selected delay filter.

24 V input characteristics

The 24-V inputs are a slower input type and have the most variation due to the input signal characteristics. The 24-V inputs have an asymmetrical response to the input voltage; in other words, the input is turned on faster than it is turned off and there is a saturation effect (the longer an input is on, the longer it takes to turn it off).

- Turn-on time is faster than turn-off time (turn-on time is typically 1.4 µs faster than turn-off time).
- Turn-on time is faster with a higher voltage input (a 20 V input level is typically 0.25 µs slower than a 30 V input level).
- Turn-off time is faster with a lower voltage input (a 20 V input level is typically 0.6 µs faster than a 30 V input level).
- Turn-off time is slower when the input on time is longer. Inputs that are on for 0.5 µs typically turn off 1.4 µs faster than inputs that are on for 6 µs. (The turn-off time does not increase for on times greater than 6 µs.)

The following table gives the typical ON/OFF delays for each input filter.

Table 5- 4 Typical delays for 24 V digital inputs

Delay filter	ON delay	Switch-off delay	Filter variation
0	1.1 µs	2.5 µs	± 0.04 µs
5	3.4 µs	4.8 µs	± 0.09 µs
10	8.2 µs	9.7 µs	± 0.25 µs
15	13.0 µs	14.5 µs	± 0.4 µs
20	17.9 µs	19.3 µs	± 0.6 µs
50	46.9 µs	48.3 µs	± 1.6 µs
1600	1546 µs	1547 µs	± 25 µs

Filtering 24 V digital inputs

The digital 24 V inputs of the FM 352-5 are standard inputs with minimum filtering. You can configure additional filters for the inputs. The most rapid response to an input change is achieved when you select the 0 input filter for an input. You can select a different filter for each input.

24 V quadrature encoder input filtering

Quadrature encoders do use input filters. The quadrature counter also uses a 3 µs delay when the 0 delay filter is set. You should specify the same filter for each input of the quadrature encoder. If different filters are set, counting errors may result. References to the quadrature encoder inputs in the user program use the filtered inputs as specified in the parameter settings.

5 V RS-422 differential digital input characteristics

RS-422 differential inputs are the fastest input type and have the least variation due to the input signal characteristics. The RS-422 inputs are typically 0.6 µs faster turning on and 2 µs faster turning off than the 24 V inputs.

- (1.1 - 0.6) µs = 0.5 µs (on delay)
- (2.5 - 2) µs = 0.5 µs (off delay)

Input filters for SSI encoders

SSI encoders do not use any input filters. Only the minimum hardware input filter is present for the SSI encoder input signals. References to the SSI encoder inputs in the user program use the filtered inputs as specified in the parameter settings.

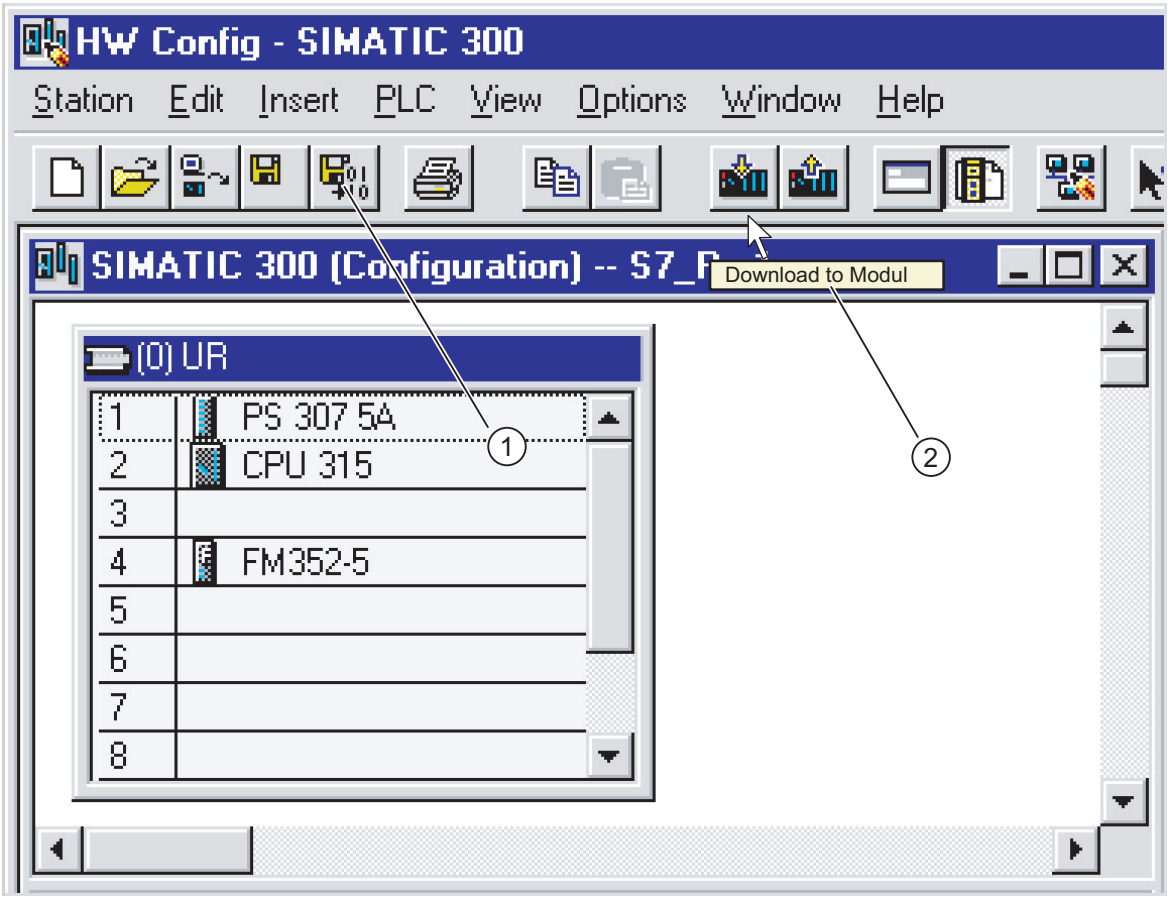
5.9 Saving and compiling the hardware configuration

Saving the Configuration

After you have selected and configured the module parameters and the diagnostic functions, you save the configuration.

To save the FM 352-5 configuration parameters, follow the steps outlined below:

1. Click "OK" in the FM 352-5 "Properties" dialog.
2. Click the "Save and Compile" button or use the menu command "Station **>** Save and Compile" in hardware configuration (see figure below).
3. Download the compiled module configuration to the S7 CPU by clicking on the "Download to Module" button or use the menu command "PLC **>** Download to Module..." in hardware configuration as shown in the following figure.



(1) Click “Save and Compile”, or select the "Station > Save and Compile" menu command.

(2) Then download the hardware configuration to the S7-CPU.

Figure 5-6 Saving and compiling the hardware configuration

5.10 Programming the controller

Description

After completing the configuration steps described in the previous sections, you are now ready to start preparing your FM 352-5 program.

[4] The "Programming" tab of the FM 352-5 "Properties" dialog provides an interface to the programming environment of FM 352-5 (see the figure below). Use the fields and buttons as described below.

1. Specify the number of the application function block that will contain the FM 352-5 program.
2. Click the "How to create new FB/DB set" button for information on how to create an FB/DB set in your project as a starting point for developing your program.
3. Click the "Edit Application FB" button to call up the STEP 7 LAD/FBD editor in which you write your application program. (For information about writing and testing a program for FM 352-5, refer to the section Programming and Operating the FM 352-5 Module (Page 65).)

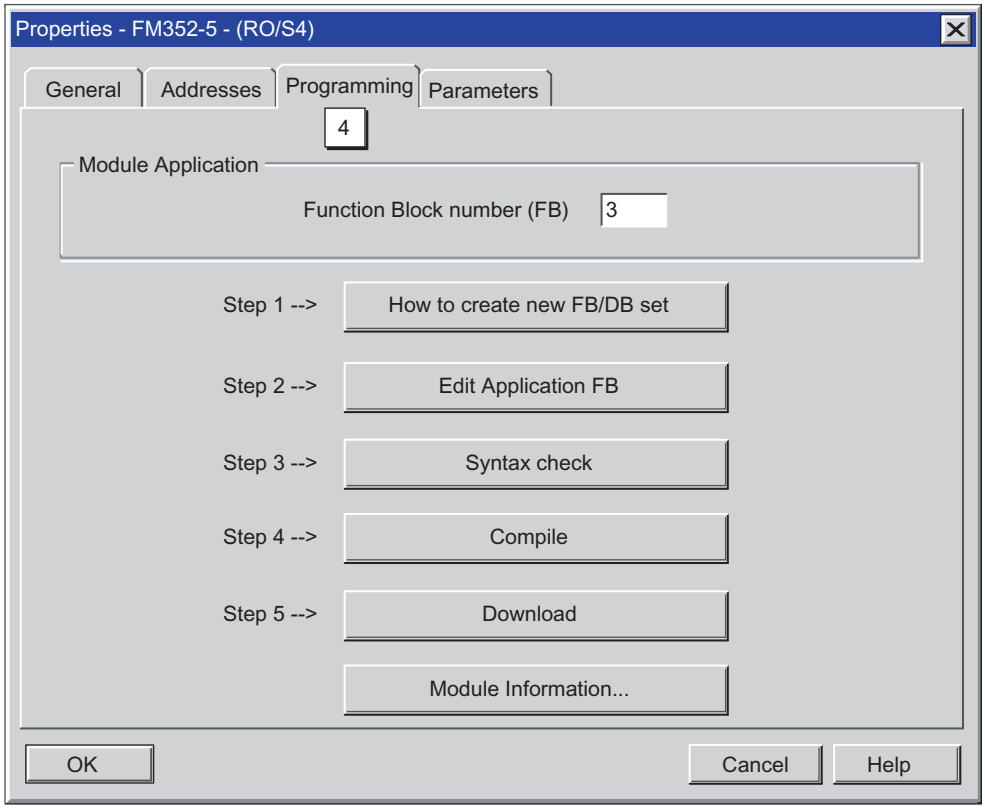


Figure 5-7 "Properties - FM 352-5" dialog, "Programming" tab

4. After writing your application FB, you can click the "Syntax check" button to check for any syntax errors that are not found by the STEP 7 LAD/FBD editor, such as the use of operations that are not supported by the FM 352-5 module.

   Any errors that are found by this syntax check must be corrected before you can successfully compile the application FB.

5. After testing the program for the FM 352-5 on the S7 CPU or in S7-PLCSIM, you are ready to compile it into an executable format for the FM 352-5 module. Click the "Compile" button to create an SDB formatted specifically for the FM 352-5 module.

   Note: This special SDB is created from a combination of the application FB and the static parameters. If you make any changes to the static parameters (those not in parameter assignment data record 1) or any changes to the application FB, you will need to recompile. Changes made to parameter assignment data record 1 (dynamic parameters) do not require the FM 352-5 program to be recompiled, but the changed hardware configuration must be downloaded to the S7 CPU.

6. Click the "Download" button to transfer the SDB from the STEP 7 programming environment to the FM 352-5 module.

7. You can use the "Module Information..." button to view diagnostic and other information about the module when STEP 7 is set to online mode after the program has been downloaded to the FM 352-5 module.

Programming and operating the FM 352-5

6

6.1 Overview

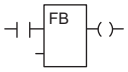
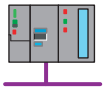
Introduction

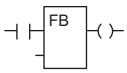
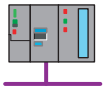
This section explains how to create and test a program for the FM 352-5 module. You will also need to refer to STEP 7 (version 5.1, SP2 or higher) documentation for the full information on creating programs. STEP 7 is the programming environment required to write, monitor, and test your program.

Overview of the tasks

The following table provides an overview of the order of the tasks necessary to create a program for the FM 352-5.

Table 6- 1 Creating the Program

	**Creating the control program**	
	Creating an application FB/DB (see section "Creating the application function block (Page 67)"): • Assign element names in the declaration section of the FB. • Use STEP 7 LAD/FBD Editor to write your program in the application FB. • Save the program in the STEP 7 editor. • Use the "Syntax error check" button in the FM 352-5 configuration tool "Programming" dialog tab to check for any syntax errors that are not found by the STEP 7 LAD/FBD editor.	
	Setting up the interface FB/DB in OB1 (see section "Setting up the interface FB/DB (Page 93)").	
	Testing the application program (see section "Debugging a program (Page 101)"). • Download program to S7 CPU (S7-314 or higher). • Use STEP 7 to monitor the FB as it executes. • Save the application FB as part of the CPU project.	

	**Creating the control program**	
	Downloading the program to the FM 352-5 module (see section "Download program to FM 352-5 module (Page 102)"): • Compile the application FB in the "Programming" tab. • Download program to FM 352-5 module.	
	Use STEP 7 to copy the program to the SIMATIC Micro Memory Card using the SIMATIC Micro Memory Card programming device (see section "Download program to FM 352-5 module (Page 102)").	

6.2 Creating the application function block

Editing the Application FB/DB Set

The application FB is the function block in your main control program that will contain the program instructions for the FM 352-5 module.

To create a new application FB/DB for your FM 352-5 module program, follow the steps outlined below:

1. In the SIMATIC Manager, open the FM 352-5 library and copy the following objects from the Blocks folder to the Blocks folder of the S7 CPU: The application FB (FB3), the interface FB for the test mode (FB30) with DB30, and the interface FB for normal mode (FB31) with DB31. (Be sure to enter the same FB number in the "Application FB" box in the FM 352-5 "Configuration" dialog.)
2. From the Library folder, copy the instruction FBs that you want to use in your FM 352-5 application program to the Blocks folder of the S7 CPU.
3. You can also copy the symbol table from the FM 352-5 library to the Blocks folder of the S7 CPU to use as a starting point. You can then change symbol names as required.
4. Click the "Edit the Application FB" button on the "Programming" tab to open the application FB for editing. The STEP 7 LAD/FBD editor displays the function block with its predefined declaration section. Adapt the declaration table to suit your application. (Names have already been assigned to the elements in the declaration table in the sample FB, but you can change these names as necessary where allowed.)
5. Enter your program logic.
6. Create a DB in STEP 7 by selecting the "Insert > S7 Block > Data Block" menu command. In the "Properties" dialog that appears, enter the DB number you want to use.
7. In the next field, select "Instance DB"
8. In the third field, select the application FB number that corresponds to the modified application FB for the FM 352-5 module. Then confirm your entries with OK.

   A new DB is created in the Blocks directory of your project.

As you enter the operations for the FM 352-5 program, you use the declared variables as addresses. Because the program in the application FB is intended to function in the FM 352-5 module, the addresses cannot access any of the S7 CPU memory areas. The following tables show how you declare the address names for use in your FM 352-5 program.

Interface to the FM 352-5 module

Programming the FM 352-5 is the same as programming a function block using the STEP 7 LAD/FBD Editor. The application FB (FB_APP) represents the application of the FM and the variable declaration table of the FB represents the FM's resources.

The input section of the declaration table is used to represent the FM's external inputs. The output section is used to represent the FM's external outputs and the static section is used to represent the FM's internal resources.

External resources of the FM 352-5 module: The external resources that are available to the FM 352-5 module's application program consist of the following objects:

- Interface to the process side:
  - 12 digital inputs (inputs for the FM application) — 24 Volt
  - 3 digital inputs (inputs to the FM application) — 5 V differential
  - 8 digital outputs (outputs from the FM application)
- Interface to the S7-300/400 CPU:
  - 14 bytes of the CPU output area assigned to the module (inputs to the FM application)
  - 14 bytes of the CPU input area assigned to the module (outputs from the FM application)

Internal resources of the FM 352-5 module: The internal resources that are available to the FM 352-5 module's application program consist of the following objects:

- Module interrupts
- Flip-flops
- Positive and negative edge detectors
- Elements represented by the FBs in the FM 352-5 library (timers, counters, etc.)
- Connectors
- Encoder interface
- Status information

Input section: The input section has two entries.

The first entry consists of the 15 bits representing the digital inputs of the FM's process interface. You can declare either 15 individual declarations of the type BOOL each with a unique name which you assign, or you can declare an array of BOOL with 15 elements and you name the array.

The second entry consists of the 14 bytes from the CPU output area. This must be declared as a structure with the name CPU_Out, its length must be a total of 14 bytes, and its position in the declaration table must always be at address 2. However, it can be composed of elements of the data types, BOOL, BYTE, WORD, INT, or DINT with element names that you yourself assign.

Output section: The output section has two entries.

The first entry consists of the 8 bits representing the digital outputs of the FM's process interface. You can declare either 8 individual declarations of the type BOOL each with a unique name which you assign, or you can declare an array of BOOL with 8 elements and you name the array.

The second entry consists of the 14 bytes to the CPU input area. This must be declared as a structure with the name CPU_In, its length must be a total of 14 bytes, and its position in the declaration table must always be at address 18. However, it can be composed of elements of the data types, BOOL, BYTE, WORD, INT, or DINT with element names that you yourself assign.

Static section: The static section has a variable number of entries depending upon the amount of internal resources required by your application. The first two are required but the remaining are optional and only required if needed in the application program.

The first entry consists of between 1 and 8 bits representing the module interrupts (hardware interrupts). You can declare either 1 to 8 individual declarations of type BOOL each with a unique name which you assign, or you can declare an array of BOOL with up to 8 elements and you name the array. The address of the first declared interrupt must be 32.

The second entry in the static section must be the structure named “ST” with the elements named exactly as shown in the table "Example of a declaration table, static section" at the fixed address 34. This represents the diagnostic status bits generated by the module for use by the application if specific action is required.

If an encoder is used in the application, the third entry in the static section must be the structure named “Encoder” with the elements named exactly as shown in the table "Example of a declaration table", encoder structure" at the fixed address 38. This represents the encoder resources for access by the application.

The FM 352-5 specific operations represented as FBs in the FM 352-5 library are declared as named static variables of multiple instances. These declarations can appear anywhere in the static elements section after the encoder structure as individual declarations. These declarations are shown in the table "Example of a declaration table, FBs of the FB library".

Flip-flops as well as positive and negative edge detectors are represented as static Boolean variables and are declared as a structure named "FF" and a structure named "Edge" respectively. Both structures can contain any combination of elements of the types BOOL or array of BOOL as required by your application. These declarations are shown in the table "Example of a declaration table, other operations".

Connections between the elements and intermediate result storage are represented as elements of the structure named "Conn" that can consist of any combination of elements of data type BOOL, INT, DINT, WORD, DWORD with names assigned by you. These declarations are shown in the table "Example of a declaration table, connectors".

For more information on creating FBs and multiple instances, see section 9 — Creating logic blocks in the SIMATIC Programming with STEP 7 Manual (http://support.automation.siemens.com/WW/view/en/45531107).

Assigning Input Elements

Use the input section of the declaration table to assign the input elements to be used in the program, as shown in the table below. These include the physical inputs of the module and the 14 byte structure used by the CPU user program for the inputs of the FM 352-5 module.

Table 6- 2 Example of a declaration table for the application FB, input section (as in STEP 7 V5.1)

Address	Decl.	Name	Type	Comment
Input section: This input is position-specific. The first 15 bits are digital inputs of the FM 352-5. You can specify a list of type BOOL or an array of BOOL (but not both). You can also assign names to the inputs.				
0.0 (cannot be changed)	in	DIn (can be changed)	ARRAY [0..14] (can be changed)	Digital inputs - (0..11 = 24 V) (12..14 = RS-422 differential)
*0.1	in		BOOL (can be changed)	
Input section: Bytes 2 through 15 are position-specific data from the CPU for the FM 352-5 module. Any combination of BOOL, array of BOOL, BYTE, WORD, INT, or DINT that totals 14 bytes is allowed. You can assign names to the inputs.				
2.0 (cannot be changed)	in	CPU_Out (cannot be changed)	STRUCT	14 bytes from the CPU as inputs for the FM.
+0.0	in	Bits (can be changed)	ARRAY [0..15] (can be changed)	...Some can be Boolean.
*0.1	in		BOOL (can be changed)	
+2.0	in	T1_PV (can be changed)	DINT (can be changed)	...Some can be DINT. (DINT must start at +2, +6, or +10)
+6.0	in	T2_PV (can be changed)	BYTE (can be changed)	...Some can be BYTE (must be mapped to INT by the MOVE operation)
+7.0	in	CmpByte (can be changed)	BYTE (can be changed)	
+8.0	in	C1_PV (can be changed)	INT (can be changed)	...Some can be INT (INT must start at an even byte boundary).
+10.0	in	CP_Period (can be changed)	WORD (can be changed)	...Some can be WORD.
+12.0	in	CMPInt (can be changed)	INT (can be changed)	Total structure length must be 14 bytes.
=14.0 (cannot be changed)	in		END_STRUCT	

Note

Data is consistent only over long word boundaries (4 bytes). To ensure data consistency, a 32-bit double integer (DINT) element must start at +2, +6, or +10.

Assigning Output Elements

Use the output section of the declaration table to assign the output elements of the module to be used in the program, as shown in the table below. These involve the physical outputs of the module and the 14 byte structure that is used by the CPU user program for the outputs of the FM 352-5 module.

Table 6- 3 Example declaration table for the application FB, output section (as in STEP 7 V5.1)

Address	Declaration	Name	Type	Comment
Output section: This output is position-specific. The first 8 bits are digital outputs of the FM 352-5. You can specify a list of the type BOOL or an array of BOOL (but not both). You can also assign names to the outputs.				
16.0 (cannot be changed)	out	DOut (can be changed)	ARRAY [0..7] (can be changed)	24 V digital outputs of this cycle.
*0.1	out		BOOL (can be changed)	
Output Section: The CPU inputs are outputs of the FM 352-5 module. This output is position-specific. Any combination of BOOL, array of BOOL, BYTE, WORD, INT, or DINT that totals 14 bytes is allowed. You can assign names to the outputs.				
18.0 (cannot be changed)	out	CPU_In (cannot be changed)	STRUCT	14 bytes assigned as inputs and returned to the CPU.
+0.0	out	Bits (can be changed)	ARRAY [0..15] (can be changed)	...Some can be Boolean.
*0.1	out		BOOL (can be changed)	
+2.0	out	T2_CVasByte (can be changed)	BYTE (can be changed)	...Some can be BYTE.
+3.0	out	C1_CVasByte (can be changed)	BYTE (can be changed)	
+4.0	out	T2_CV (can be changed)	INT (can be changed)	...Some can be INT.
+6.0	out	T1_CV (can be changed)	DINT (can be changed)	...Some can be DINT. (DINT must start at +2, +6, or +10)
+10.0	out	Enc_CV1 (can be changed)	DINT (can be changed)	Total structure length must be 14 bytes.
=14.0 (cannot be changed)	out		END_STRUCT	
	in_out			

Assigning Static Elements

The static section of the declaration table contains the internal resources of the FM 352-5 module to be used in the program.

The first two sections consist of 8 hardware interrupt bits and module status bits of the FM 352-5 module, as shown in the table below. The module status bits cannot be changed.

Table 6- 4 Example declaration table for the application FB, static section (as in STEP 7 V5.1)

Address	Declaration	Name	Type	Comment
Static section: This definition is position-specific. The first 8 bits are interpreted as hardware interrupts (hardware interrupts that trigger OB40). You can specify a list of the type BOOL or an array of BOOL (but not both). You can also assign names to the elements.				
32.0 (cannot be changed)	stat	Intr (can be changed)	ARRAY [0..7] (can be changed)	Resources for module interrupts. High limit fixed. Do not change.
*0.1	stat		BOOL (can be changed)	
Static section: This definition is position-specific. These are module status bits. Do not change.				
34.0 (cannot be changed)	stat	ST (cannot be changed)	STRUCT	Resources for module status bits. High limit fixed. Do not change.
+0.0 (cannot be changed)	stat	FIRSTSCAN (cannot be changed)	BOOL (cannot be changed)	First cycle after a STOP to RUN transition.
+0.1 (cannot be changed)	stat	M3L (cannot be changed)	BOOL (cannot be changed)	Power supply for 3L is missing.
+0.2 (cannot be changed)	stat	ESSF (cannot be changed)	BOOL (cannot be changed)	Encoder power supply is overloaded.
+0.3 (cannot be changed)	stat	M2L (cannot be changed)	BOOL (cannot be changed)	Power supply for 2L is missing.
+0.4 (cannot be changed)	stat	M1L (cannot be changed)	BOOL (cannot be changed)	Power supply for 1L is missing.
+2.0 (cannot be changed)	stat	OVERLOAD (cannot be changed)	ARRAY [0..7] (cannot be changed)	Output [x] is overloaded.
*0.1 (cannot be changed)	stat		BOOL (cannot be changed)	
=4.0 (cannot be changed)	stat		END_STRUCT	

This part of the static section contains the encoder structure, as shown in the table below. These elements cannot be changed. The entire structure, however, can be deleted if the encoder is not used.

Table 6- 5 Example of a declaration table for the application FB, encoder structure (as in STEP 7 V5.1)

Address	Declaration	Name	Type	Comment
Static section: This definition is position-specific. The encoder is a structure that has a fixed number of elements. The names cannot be changed, but the size of Cur_Val and Load_Val must be set to INT or DINT according to which size of encoder is configured.				
38.0 *	stat	Encoder*	STRUCT	Encoder structure. Do not change.
+0.0 *	stat	Direction *	BOOL *	Status: Direction 0 = up count, 1 = down count
+0.1 *	stat	Home *	BOOL *	Status: 1 = encoder is at home position.
+0.2 *	stat	Homed *	BOOL *	Status: 1 = Home was adopted since power up
+0.3 *	stat *	Overflow *	BOOL *	Status: 1= overflow (displayed for the duration of one cycle)
+0.4 *	stat	Underflow *	BOOL *	Status: 1= Underflow (displayed for 1 cycle)
+0.5 *	stat	SSIFrame *	BOOL *	Status: SSI frame error or power loss
+0.6 *	stat	SSIDataReady *	BOOL *	Status: 0 = SSI encoder has not yet shifted valid data, 1 = data available
+0.7 *	stat	Open_Wire *	BOOL *	Status: 1 = Encoder has open wire
+1.0 *	stat	Hold *	BOOL *	Hold software input for incremental encoder
+1.1 *	stat	Reset *	BOOL *	Reset software input for incremental encoder
+1.2 *	stat	Load *	BOOL *	Load software input for incremental encoder
+2.0 *	stat	Cur_Val *	DINT (can be changed)	Current value for incremental encoder: DINT for 32-bit encoder, INT for 16-bit encoder
+6.0 *	stat	Load_Val *	DINT (can be changed)	Load value for the encoder: DINT or INT
=10.0 *	stat		END_STRUCT	
* If an encoder structure is used, it cannot be changed. If it is not used, it can be deleted.				

This part of the static section contains multiple-instance declarations of each FB from the FM 352-5 library, as shown in the table below. These names can be changed.

Table 6- 6 Example of a declaration table for the application FB, FBs of the FB library (as in STEP 7 V5.1)

Address	Declaration	Name	Type	Comment
Static section: These definitions are not position-specific. The FM 352-5 module recognizes the multiple-instance FB based on the type ("CTU16", "TP32", etc.). The FBs are from the library of the FM 352-5. You can assign names to the FBs. The types of the FB pin names (IN, OUT, etc.) must be specified. This is required for the connectors.				
48.0 (can be changed)	stat	UCtr1 (can be changed)	"CTU16" (can be changed)	The 16-bit up counter is a multiple instance of FB121 from the FM 352-5 library.
60.0 (can be changed)	stat	DCtr1 (can be changed)	"CTD16" (can be changed)	16-bit down counter (FB122)
72.0 (can be changed)	stat	UDCtr1 (can be changed)	"CTUD16" (can be changed)	16-bit up/down counter (FB123)
84.0 (can be changed)	stat	UDCtr2 (can be changed)	"CTUD32" (can be changed)	32-bit up/down counter (FB120)
102.0 (can be changed)	stat	TmrP1 (can be changed)	"TP32" (can be changed)	32-bit timer (FB113)
120.0 (can be changed)	stat	TmrOn1 (can be changed)	"TON32" (can be changed)	32-bit timer (FB114)
138.0 (can be changed)	stat	TmrOf1 (can be changed)	"TOF32" (can be changed)	32-bit timer (FB115)
156.0 (can be changed)	stat	TmrP2 (can be changed)	"TP16" (can be changed)	16-bit timer (FB116)
170.0 (can be changed)	stat	TmrOn2 (can be changed)	"TON16" (can be changed)	16-bit timer (FB117)
184.0 (can be changed)	stat	TmrOf2 (can be changed)	"TOF16" (can be changed)	16-bit timer (FB118)
198.0 (can be changed)	stat	SReg1 (can be changed)	"SHIFT" (can be changed)	Shift registers (FB124 to FB127)
718.0 (can be changed)	stat	SReg2 (can be changed)	"SHIFT2" (can be changed)	
1238.0 (can be changed)	stat	BiS (can be changed)	"BiScale" (can be changed)	2:1 binary scaler (FB112)
1244.0 (can be changed)	stat	Clk50 (can be changed)	"CP_Gen" (can be changed)	Pulse generator (FB119)

Note

Your project must contain all FBs that are listed in the declaration section of the application FB in order to be accessible for execution. Any declared FBs that have no corresponding FB in the project will appear in red.

This part of the static section contains declarations for flip-flop operations and positive and negative edge operations, as shown in the table below. These names can be changed.

Table 6- 7 Example of a declaration table for the application FB, other operations (as in STEP 7 V5.1)

Address	Declaration	Name	Type	Comment
Static section: This definition is not position-specific. You can change the names inside the structure except for "FF". You can use any combination of BOOL and array of the type BOOL.				
1254.0 (can be changed)	stat	FF (cannot be changed)	STRUCT	Resources for R/S and S/R. Each element must be BOOL or an array of BOOL.
+0.0 (can be changed)	stat	FirstFF (can be changed)	BOOL (can be changed)	The number of elements can be increased as required.
+0.1 (can be changed)	stat	SecondFF (can be changed)	BOOL (can be changed)	The names of elements can be freely assigned.
+0.2 (can be changed)	stat	ThirdFF (can be changed)	BOOL (can be changed)	
+2.0 (can be changed)	stat	MoreFFs (can be changed)	ARRAY [0..15] (can be changed)	
*0.1	stat		BOOL (can be changed)	
=4.0 (can be changed)	stat		END_STRUCT	
Static section: This definition is not position-specific. You can change the names inside the structure except for "Edge". You can use any combination of BOOL and array of the type BOOL.				
1258.0 (can be changed)	stat	Edge (cannot be changed)	STRUCT	Resources for edge detection. Each element must be BOOL or an array of BOOL.
+0.0 (can be changed)	stat	FirstEdge (can be changed)	BOOL (can be changed)	The number of elements can be increased as required.
+0.1 (can be changed)	stat	SecondEdge (can be changed)	BOOL (can be changed)	The names of elements can be freely assigned.
+0.2 (can be changed)	stat	ThirdEdge (can be changed)	BOOL (can be changed)	
+2.0 (can be changed)	stat	Edge4to10 (can be changed)	ARRAY [4..10] (can be changed)	
*0.1	stat		BOOL (can be changed)	
+4.0 (can be changed)	stat	LastEdge (can be changed)	BOOL (can be changed)	
=6.0 (can be changed)	stat		END_STRUCT	

This part of the static section contains declarations for connectors, as shown in the table below. These names can be changed.

Table 6- 8 Example of declaration table for the application FB, connectors (as in STEP 7 V5.1)

Address	Declaration	Name	Type	Comment
Static section: This definition is not position-specific. You can change the names inside the structure except for "Conn". You can use any combination of BOOL, INT, DINT or Array of BOOL, INT, or DINT.				
1264.0 (can be changed)	stat	Conn (cannot be changed)	STRUCT	Resources for connectors.
+0.0 (can be changed)	stat	XCon (can be changed)	BOOL (can be changed)	Elements can be BOOL.
+2.0 (can be changed)	stat	arrXCon (can be changed)	ARRAY [0..31] (can be changed)	Elements can be an array of BOOL.
*0.1	stat		BOOL (can be changed)	
+6.0 (can be changed)	stat	ICon (can be changed)	INT (can be changed)	Elements can be INT.
+8.0 (can be changed)	stat	arrICon (can be changed)	ARRAY [0..3] (can be changed)	Elements can be an array of INT.
*2.0	stat		INT (can be changed)	
+16.0 (can be changed)	stat	DICon (can be changed)	DINT (can be changed)	Elements can be DINT.
+20.0 (can be changed)	stat	arrDICon (can be changed)	ARRAY [0..3] (can be changed)	Elements can be an array of DINT.
*4.0	stat		DINT (can be changed)	
=36.0 (can be changed)	stat		END_STRUCT	
Temp section: This definition is position-specific. The name cannot be changed.				
0.0 (cannot be changed)	temp	Dummy (cannot be changed)	BOOL (cannot be changed)	Is used where an output coil is required by STEP 7 to execute the operation but is not needed by your program.

Ensuring Data Consistency

When transferring data to the FM 352-5 using the 14 bytes, you need to consider the following points to ensure data consistency:

For consistency of data type DINT (or less):

- For data type DINT, the address must be 2, 6, or 10 in the structure.
- For data type INT, the address must be on an even number boundary.
- No precautions need to be taken if the data type is BYTE or smaller.

For consistency of data type greater than DINT:

A control bit must be used to store the data that must be consistent. The data must be transferred to the module, then the control bit must be set to store the data. The control bit can be edge detected (POS) to reduce the number of cycles needed for the transfer. You can use such a handshake as follows:

1. Set the control bit to 0.
2. Write the data.
3. Read the reflected control bit (which must be looped back in the user program) and wait for 0.
4. Set the control bit to 1 (the FM application program must store the data on this edge).
5. Read the reflected control bit and wait for 1.

The interface is now ready for the sequence to repeat.

Updating the Instance Data Block

The instance data block (DB) of the application FB contains the data elements required by the FB to execute the program in test mode. If you make certain changes to the FB declaration section, such as adding or deleting multiple instances of an operation, then the DB no longer matches the FB. When the CPU executes the FB in test mode, the CPU may go to STOP mode if access errors occur as a result of the mismatch.

To update the DB so that it will match the changes made to the FB, follow the steps outlined below:

1. Delete the existing instance DB belonging to the modified FB.
2. Select the menu command "Insert **>** S7 Block **>** Data Block" right-click and select the command "Insert New Object **>** Data Block" in the context menu.
3. In the "Properties" dialog that appears, enter the same number as the deleted DB.
4. In the next field, select "Instance DB"
5. In the third field, select the number of the modified application FB for the FM 352-5 module.
6. Confirm with "OK." The new instance DB is created in the Blocks folder of your project and is updated to contain data that matches the FB.

Selecting standard STEP 7 operations for the application FB

To create your application FB, you use bit-logic operations (for example, contacts and coils) and comparison operations which come from the standard list of STEP 7 operations, as shown in the figure below.

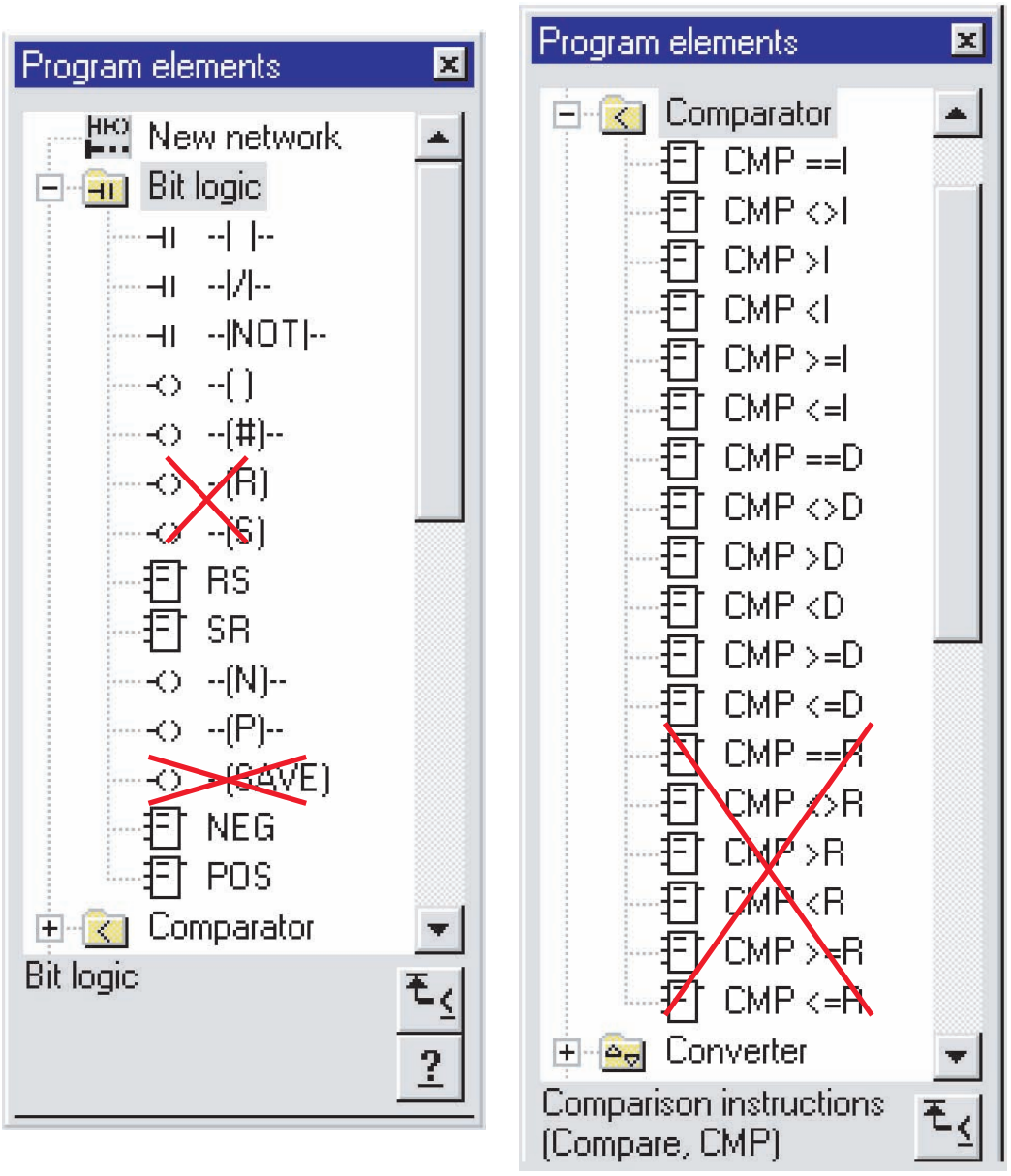
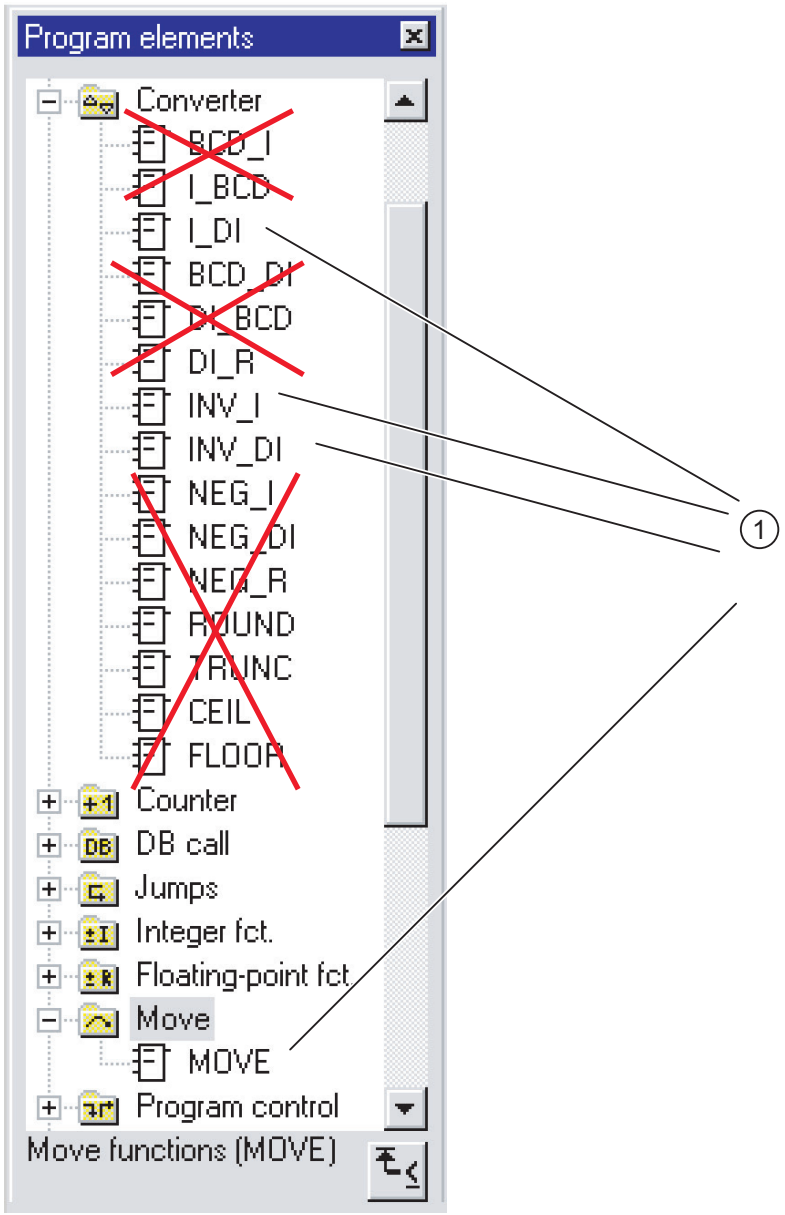


Figure 6-1 Valid bit logic and comparison operations from STEP 7 for the FM 352-5

Additional STEP 7 operations for the application FB

The figure below shows four additional operations from the STEP 7 catalog that can be used for the FM 352-5. The conversion operations I_DI, INV_I, INV_DI and the MOVE operation.



(1) You can use the I_DI, INV_I, INV_DI and the MOVE operations from the STEP 7 catalog.

Figure 6-2 Valid conversion and move operations from STEP 7 for the FM 352-5

The following figure shows the shift/rotate operations from the STEP 7 catalog that are valid for the FM 352-5.

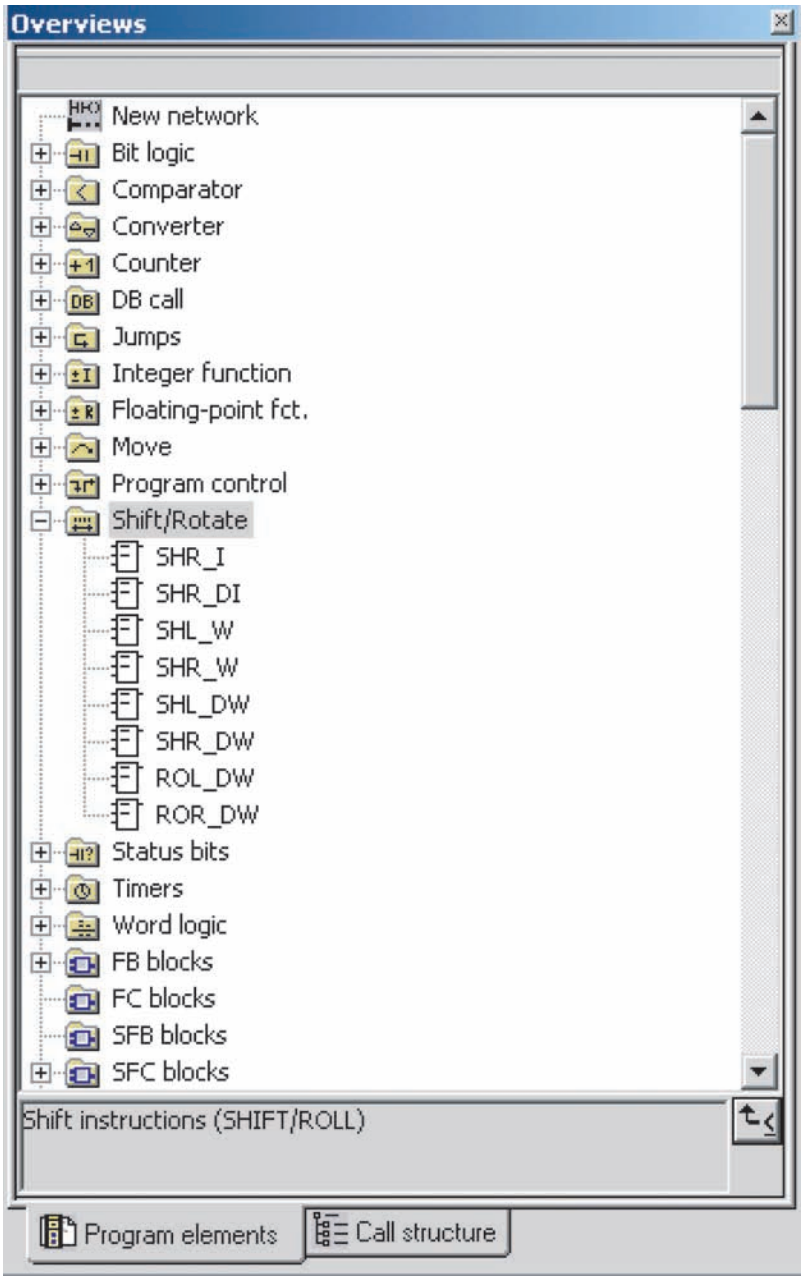


Figure 6-3 Valid shift/rotate operations from STEP 7 for the FM 352-5

The following figure shows the word logic operations from the STEP 7 catalog that are valid for the FM 352-5.

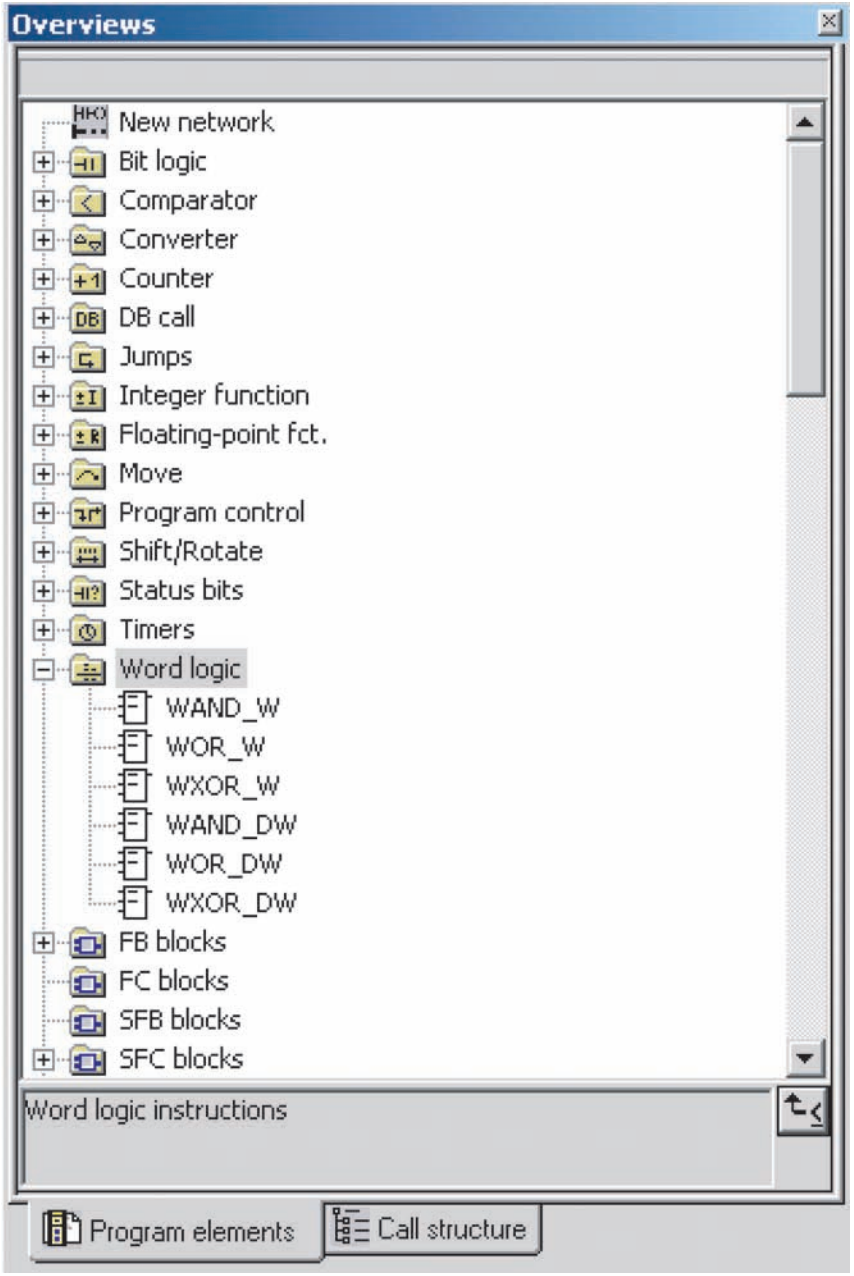


Figure 6-4 Valid word logic operations from STEP 7 for the FM 352-5

Using the FM 352-5 library operations

You can also use function blocks that were specially developed for the FM 352-5 module. These FBs are located in the FM 352-5 library (see figure below).

To select the FBs that you need for your application program, follow the steps outlined below:

1. In the operation catalog, expand the Library folder, then select the FM 352-5 object and expand it.
2. Expand the FM 352-5 Library folder. The full list of FBs is displayed, along with their symbolic names.

3. Select the FBs you require for your program and double-click or drag them to your application program.
4. Change each FB to a multiple-instance call. Right-click on the FB and open the context menu. Then select the "Change to Multiple Instance Call..." menu command. Enter the name of the multiple-instance block as defined in the application FB declaration section.

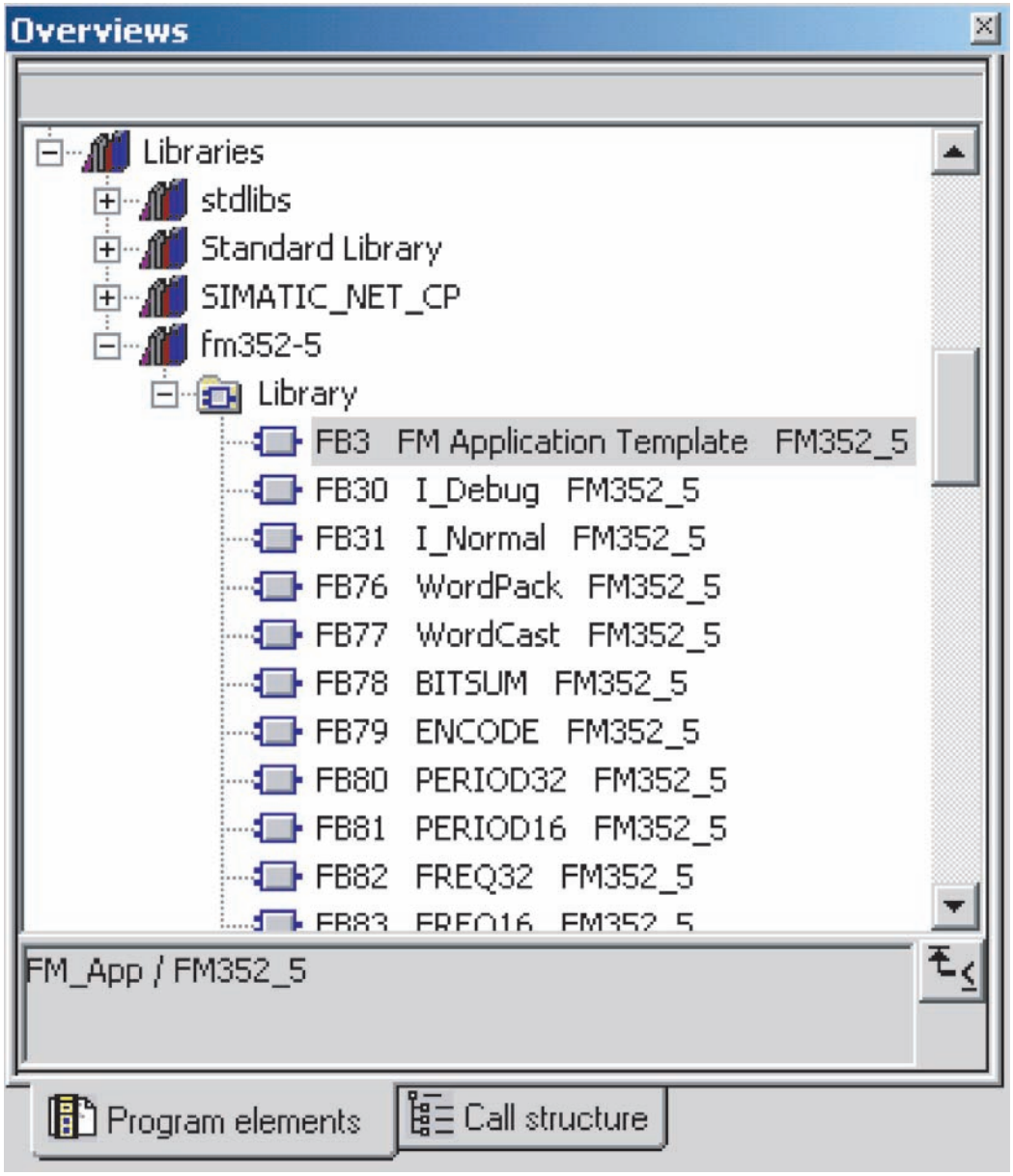


Figure 6-5 FM 352-5 library of FBs

Addresses of the operations

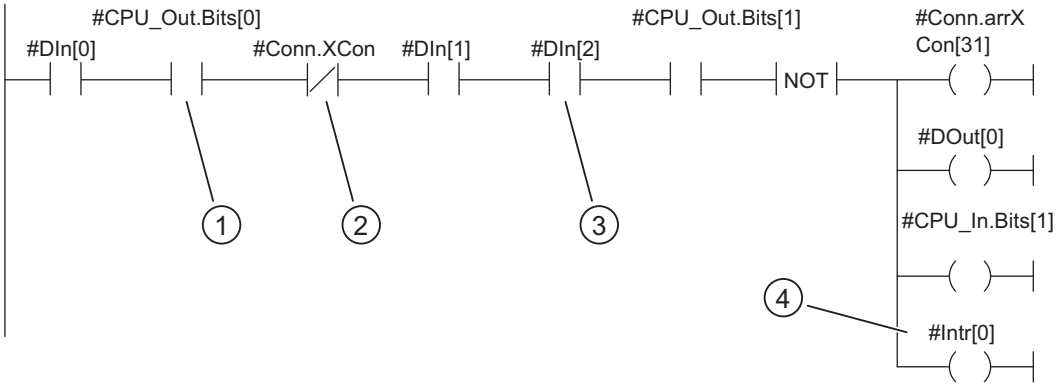
Because the program in the application FB is intended to function in the FM 352-5 module, the addresses cannot access any of the S7 CPU memory areas. The following table shows the addresses of the operations that can be used in your program.

Table 6- 9 Addresses of the operations

Addresses of the operations	Declaration section	Description
Input addresses		
FM 352-5 inputs	Input	Digital inputs of the FM 352-5
CPU outputs	Input	14 bytes from the CPU as inputs for the FM.
Connectors	Static	Similar to bit memories in S7 programs.
Constants (non-boolean)	—	
Module status bits	Static	Diagnostic interrupts.
Encoder status bits and current value	Static	Encoder structure. Set Cur_Val to INT or DINT according to size of the configured encoder.
Output addresses*		
FM 352-5 outputs	Output	Digital outputs of the FM 352-5
CPU inputs	Output	14 bytes from the FM returned as inputs to the CPU.
Connectors	Static	Similar to bit memories in S7 programs.
Hardware interrupts	Static	Bits that are interpreted as hardware interrupts (hardware interrupts that trigger OB40).
Encoder control bits and load value	Static	Encoder structure. Load Val to INT, or set DINT, depending on the size of the configured encoder.
Midline outputs*		
Connectors	Static	Similar to bit memories in S7 programs.
* Output operands and midline outputs can be written to only once in the application FB.		

Examples of Input and Output Operands

The network in the following figure shows the types of addresses that can be used to label contacts when displayed in LAD. Any declared boolean input can be used as a contact. Output coils, as shown in the figure below, can be labeled with any declared boolean output or interrupt (Intr[x]).



(1) Output of the CPU as an input

(2) Boolean connector

(3) Digital input bit from the module

(4) One of eight module interrupts

Figure 6-6 Input and output addresses of the FM 352-5

Examples of FBs from the library

The following figure shows an example of a 32-bit pulse timer (FB113 from the FM 352-5 Library). This timer is declared as a multiple-instance call in the Stat area.

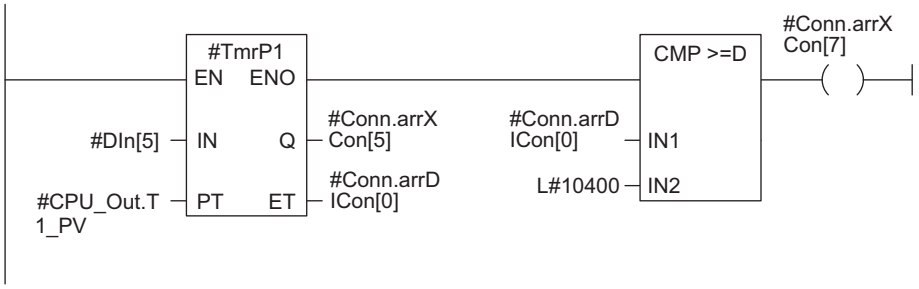


Figure 6-7 Example of a 32-bit pulse timer from the FBs of the library

The following figure shows examples of two shift registers (FB124 and FB125 from the FM 352-5 library). Each shift register is declared as a separate instance. Internal stages cannot be accessed. Only the output stage can be accessed inside the program.

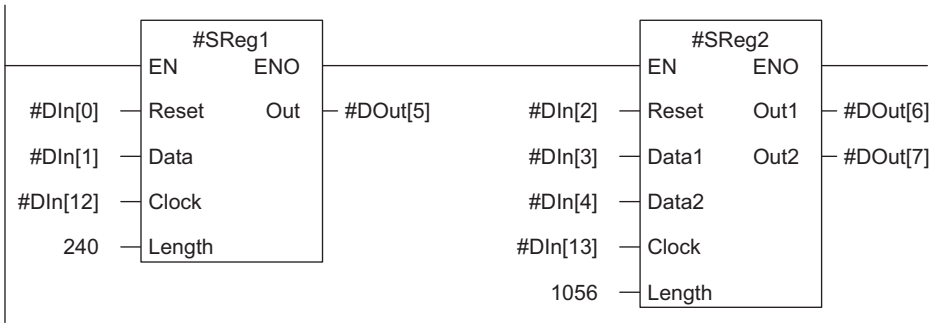
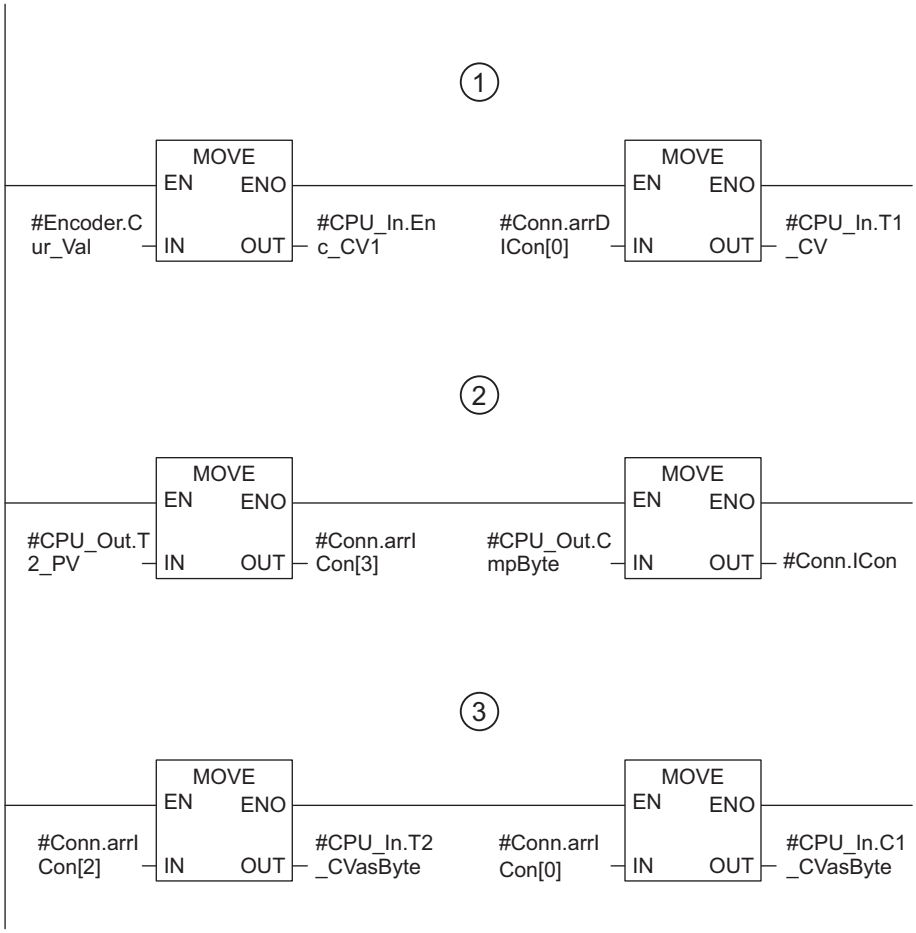


Figure 6-8 Examples of shift registers from the FBs of the library

The following figure shows examples of how the MOVE operation can be used to connect values to the CPU inputs. When necessary, the MOVE operation can also be used to convert values from one data type to another.



(1) The MOVE operation can be used to connect values to the CPU inputs. With no logic for EN, the MOVE operation is interpreted as a connector. With the logic for EN, the value of MOVE is retentive, requiring storage.

(2) The MOVE operation can be used to convert a byte from the CPU output area to the data type INT to be used for compares or defaults. This works for positive numbers only, since the MOVE operation works without a sign.

(3) The MOVE operation can be used to convert a current value of the INT data type to the BYTE data type in the CPU input area.

Figure 6-9 Examples of conversions with the MOVE operation

The following figure shows how the MOVE operation can be used to convert data type DINT to INT. You can do this only if the DINT value is within the limits for the INT data type. You can also convert data type INT to DINT, but in order to preserve the sign, you need to use the I_DI operation.

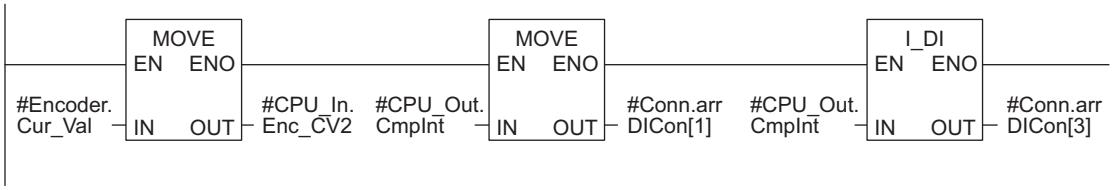
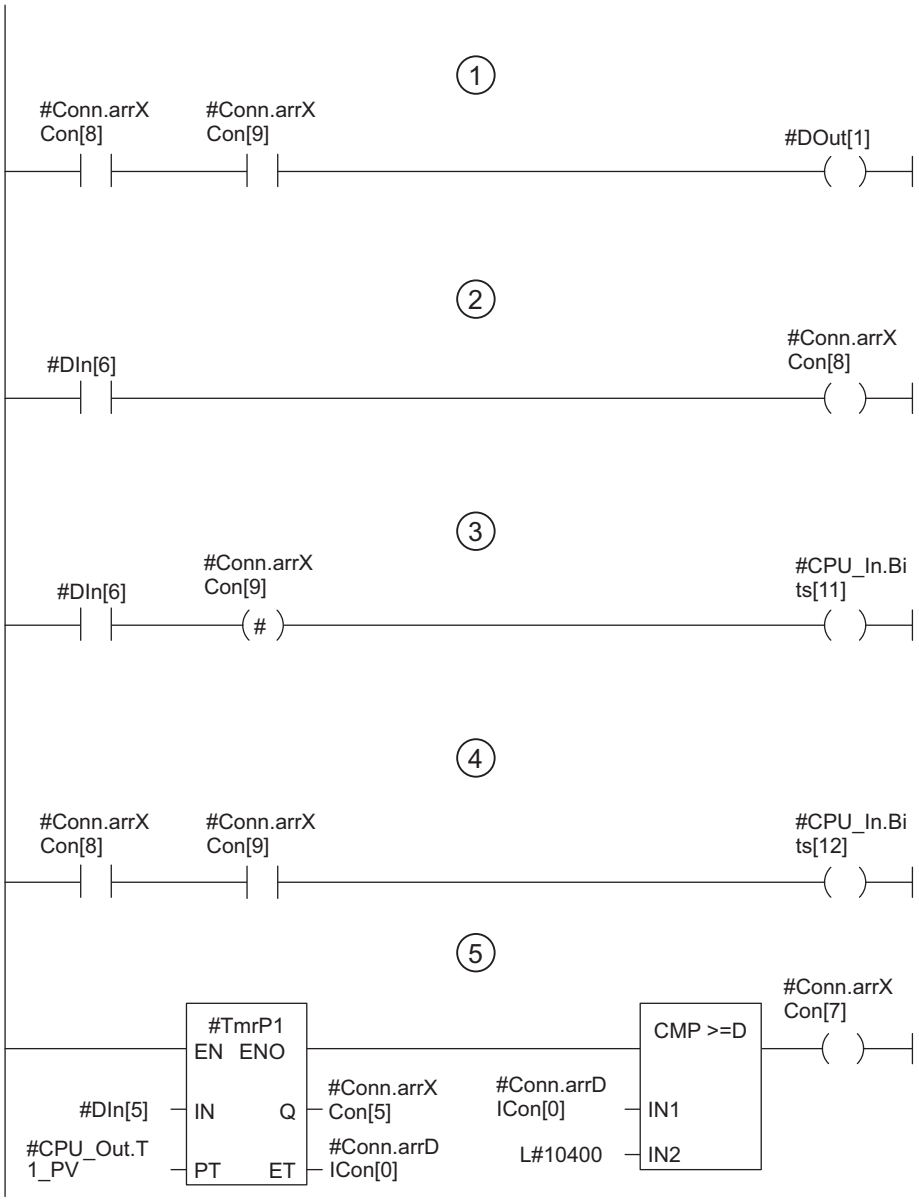


Figure 6-10 Example of MOVE and I_DI operations for conversion

Connectors

Connectors are a special type of address required by the FM 352-5 to provide control functionality similar to bit memories in standard S7 programs.

The figure below shows how connectors are used with preceding or following elements.



(1) In this network, the connectors are referenced before they are output, so they are from the previous scan cycle.

(2) In this network, the connector output, Conn.arrXCon[8], connects to following references.

(3) The midline output Conn.arrXCon[9] connects to following references. Midline outputs are allowed for connectors only.

(4) Since the connectors in this network are referenced after they are output, they are from the same scan cycle, and thus they represent a direct connection.

(5) Connectors can be BOOL, DINT, BYTE, or WORD data types.

Figure 6-11 Examples of connectors

Multi-phase clocking

The FM 352-5 module uses an onboard processor, the FPGA, to execute code in parallel rather than sequentially as standard programmable controllers do. This method of execution allows an extremely fast and stable sampling time. To eliminate runtime differences in the programmed networks, multi-phase clocking was implemented.

Multi-phase clocking is a technique included in the FM 352-5 translator software to manage the correct time sequencing of retentive elements relative to connectors in the different networks of the application program. Twelve clock pulses are available, eleven to clock elements with storage (flip-flops, counters, etc.), and the twelfth to clock the outputs.

The module's 12-phase clock uses the connectors to synchronize the execution of preceding or following elements in the networks.

The following two rules apply to the FM 352-5 software:

- If a connector is referenced as an input of an element **before** it is output, this element obtains the connector's value from the previous scan cycle.
- If a connector is referenced as an input of an element **after** it is output, this element obtains the connector's value from the current scan cycle.

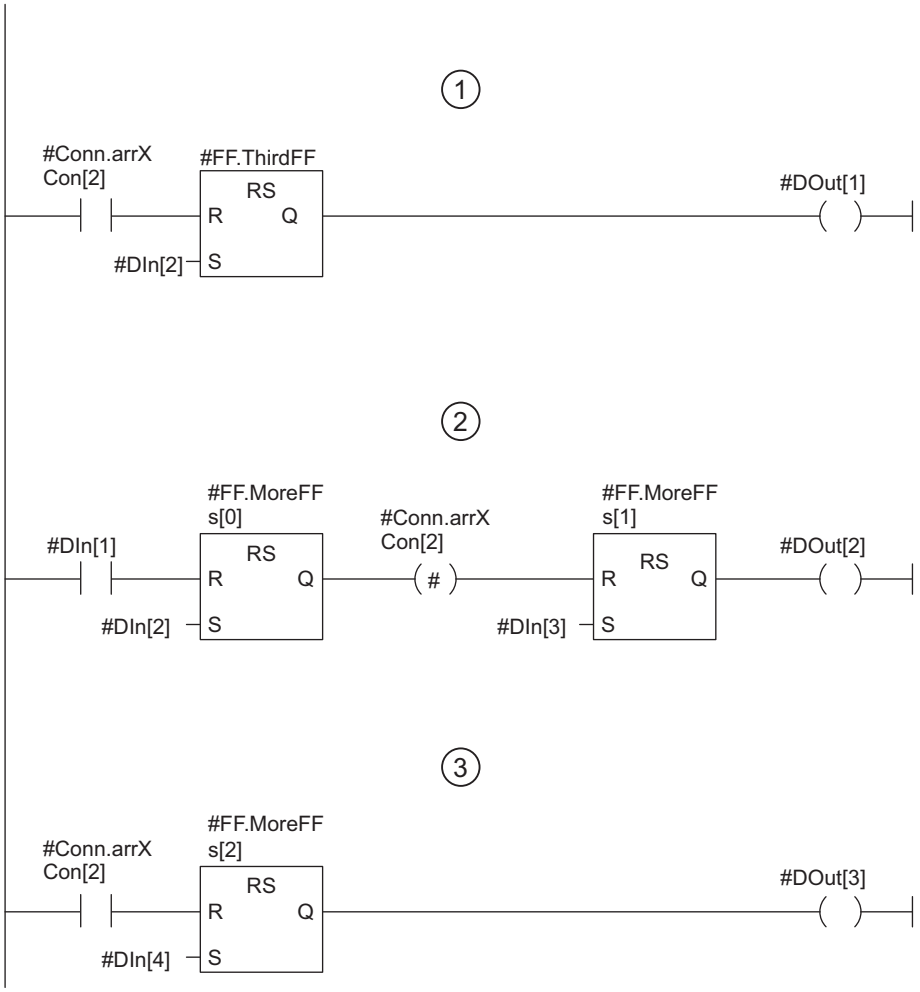
The use of 12-phase clocking means you can connect up to 11 storage elements in series without worrying about extending the scan cycle time. If you insert too many elements in series, the software displays an error message that helps you take the necessary action to meet the multi-phase clock rules.

Another advantage of multi-phase clocking is that it generates the same logical sequence of the program in the FPGA as when the S7 CPU executes the program in Test mode.

The retentive elements are the following:

- Timers
- Counter
- Flip-flops
- Edge detection
- Shift register
- Binary scaler

The following figure shows examples of multi-phase clocking of retentive elements with connectors.



(1) In this network, the Conn.arrXCon[2] connector is from the previous scan cycle because it is referenced before it is output. ThirdFF is clocked with phase 1.

(2) In this network, MoreFFs[0] is clocked with phase 1, and MoreFFs[1] is clocked with phase 2. The output DOut[2] is clocked with the last phase. The midline output connector Conn.arrXCon[2] is valid after the phase 1 clock.

(3) Since Conn.arrXCon[2] was set with a midline output between the phase 1 and phase 2 clocks in the network above, MoreFFs[2] in this network is assigned to the phase 2 clock.

Figure 6-12 Examples of Multi-phase Clocking of Retentive Elements

The following figure shows a graphic representation of how inputs and outputs are handled by the multi-phase clocking of the FM 352-5 module. The total response time is calculated by adding the input delays, scan cycle time, and output delays, as shown in the figure. Inputs from the CPU are delayed by the CPU cycle, the I/O scan, and the module's microprocessor scan cycle. Outputs to the CPU are delayed by the module's microprocessor scan cycle, the I/O scan cycle, and the CPU scan cycle.

The previous figure explains the example program logic that determines when the "FF.MoreFFs[x]" elements are clocked.

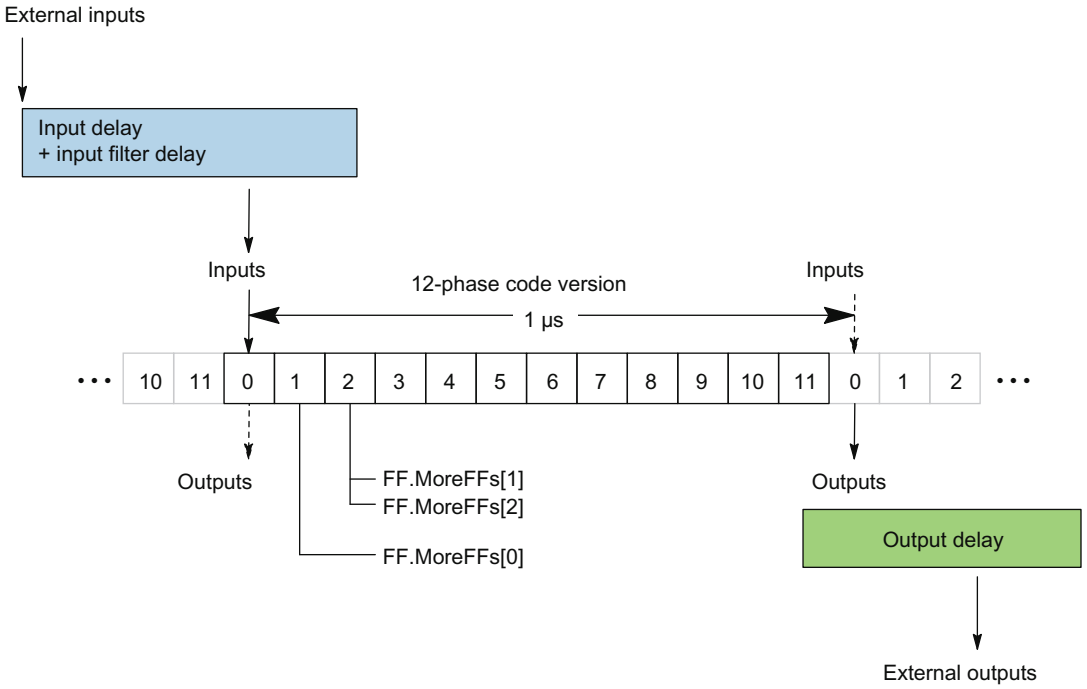


Figure 6-13 Multi-Phase Clocking and I/O Timeline

6.3 Setting up the interface FB/DB

Overview

The FM 352-5 library contains two Interface FBs that allow the S7 CPU user program (OB1, for example) to control the mode and operating states of the FM 352-5 module. You need to insert an appropriate interface FB call in OB1 to handle the exchange of data between the CPU and the FM 352-5 module.

If a programmed SIMATIC Micro Memory Card is inserted in the module at startup, the FM 352-5 copies the program from the SIMATIC Micro Memory Card to the FPGA, sets normal mode, and changes to STOP. If a programmed SIMATIC Micro Memory Card is not inserted in the module, FM 352-5 copies its internal program to the FPGA, sets normal mode, and changes to STOP.

If configured to operate in an S7 environment, the mode and operating state are decided by the Interface FB and the RUN/STOP selector located on the FM 352-5's front panel.

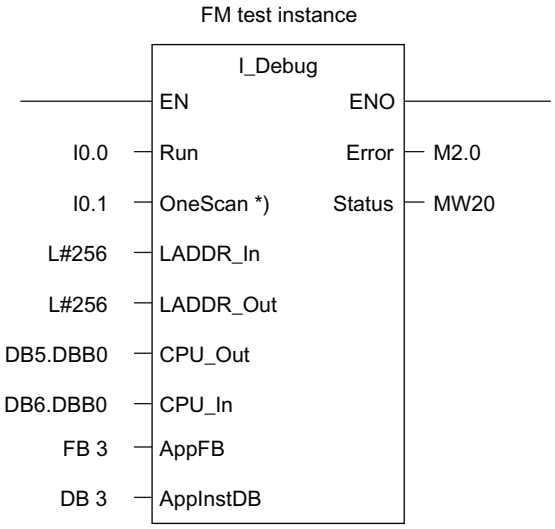
Calling the Interface FB for Test Mode

The transition from Normal to Test mode is initiated by the CPU user program calling the interface FB for Test mode (FB30 in the FM 352-5 library). As a result of this mode change command, the FM 352-5 replaces the program in the FPGA with its internal test program.

To test your application FB using the S7 CPU with the FM 352-5 module in test mode, download the following elements to the CPU in addition to the blocks in your regular CPU program:

- Application FB containing the FM 352-5 program with the up-to-date instance DB.
- Interface FB for test mode of the FM with instance DB (FB 30/DB 30 in the FM 352-5 library)

The following figure shows the structure of the FB labeled "I_Debug" that is used to call the application FB in test mode.



*) The "OneScan" input only works in Normal mode.

Figure 6-14 Interface FB to execute the Test mode

data flow in test mode

In test mode, the S7 CPU executes all programs so that you can use the various program monitoring and testing capabilities of STEP 7 to test your application program. The FM 352-5 module operates in a pass-through mode, making its inputs and outputs directly available to the S7 CPU.

The following figure shows the flow of input and output data between OB1, the application FB with its instance DB, and the FM 352-5 module inputs and outputs over the Test interface FB when the Test interface FB is called by OB1.

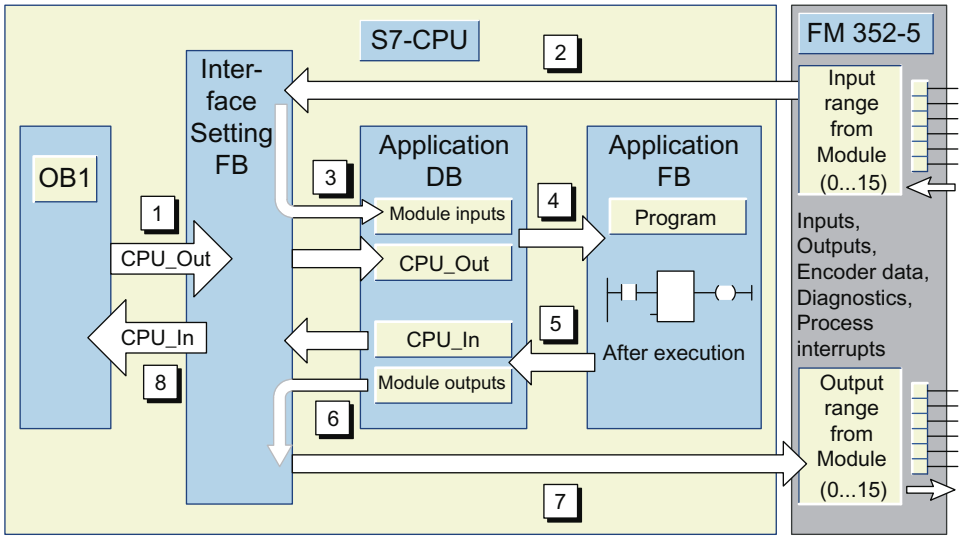


Figure 6-15 Data Exchange in Test Mode

The data flows in the following sequence:

- [1] OB1 calls the Test interface FB that communicates with the FM 352-5 module and associated application FB.
- [2] The Test interface FB reads the inputs of the FM 352-5 module, and (3) transfers the data, along with the CPU_Out interface data, to the instance DB associated with the application FB. The Test Interface FB then calls the application FB.
- [4] The application FB reads the input data from its instance DB and uses this data to execute its program.
- [5] While the program is executed, the application FB writes the output data back to its instance DB and returns to the Test interface FB.
- [6] The Test interface FB reads the results of program execution from the application FB's instance DB, and (7) writes the output results to the module, which then sets the outputs.
- [8] The Test interface FB also copies the program execution results back to the CPU_In area of OB1.

Calling the Normal Interface FB

The change from Test to Normal can be initiated by clicking the "Download" button on the FM 352-5 configuration software "Programming" tab. When the download to the FM 352-5 begins, the module changes to STOP and copies the downloaded file to the FPGA.

The SIMATIC Micro Memory Card is not changed by the download. The FM 352-5 module remains in Normal mode when the download completes and remains in STOP until the CPU user program calls the interface FB for normal operation (FB31 in the FM 352-5 library) with a 1 signal at the Run input and the RUN/STOP selector in the RUN position. With this call, the FM 352-5 module begins executing the program that was downloaded to the FPGA.

The following figure shows the structure of the "I_Normal" FB that is used to call the application FB in normal operation.

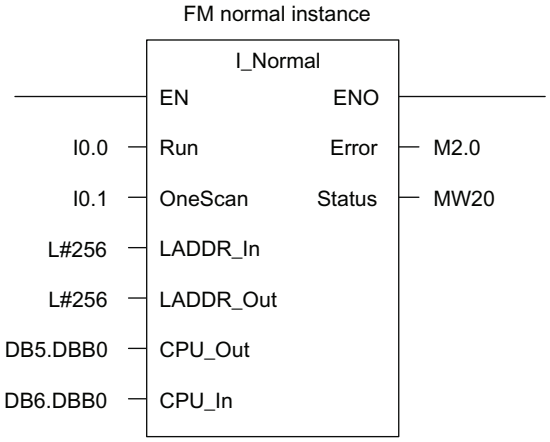


Figure 6-16 Interface FB for Normal Mode Execution

data flow in normal mode

In normal operation, the application FB executed in the FPGA (Field Programmable Gate Array) of the FM 352-5 module. The application FB was compiled and copied to the SIMATIC Micro Memory Card card that is installed in the FM 352-5 module.

At startup, the FPGA reads the image of the FB that was stored in the SIMATIC Micro Memory Card. Any time power to the system is lost or interrupted, the FPGA program is lost. When power is restored, the FPGA again reads the program from the SIMATIC Micro Memory Card.

The following figure shows the flow of input and output data between OB1 and the FM 352-5 module inputs and outputs over the interface FB. The interface FB transfers CPU_Out data from the CPU to the module, and CPU_In data from the module to the CPU.

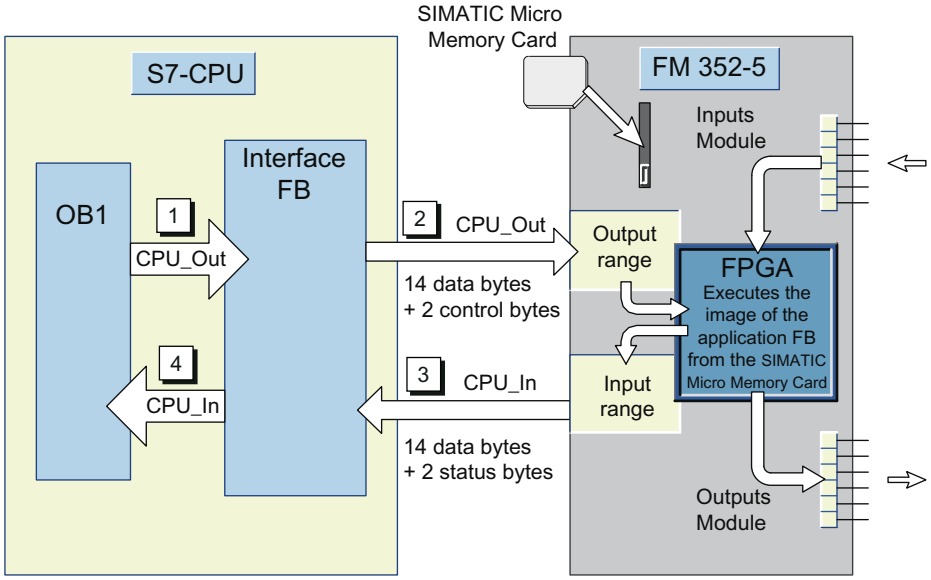


Figure 6-17 Data Exchange in Normal Mode

Interface FB parameters

The following table lists the parameters of the interface FB and describes their functions. Enter the addresses for the module inputs and outputs and the pointers to the data structures that are exchanged between the CPU and the module.

Table 6- 10 Interface FB Parameter Definitions

Parameters	Data type	Definition
Run	BOOL	When set to 1, this bit requests the module to change to RUN mode. If the mode selector on the module is also in the RUN position and the OneScan input is set to 0, then the module changes to RUN. When set to zero, the module changes to STOP mode even if the selector on the module is in the RUN position.
OneScan	BOOL	When set to 1, this bit enables the single-scan mode. As long as this input is 1, the module will execute one scan cycle each time the Run input changes from zero to one. When set to zero, the module follows the Run input.
LADDR_In	DINT	Logical address of the FM 352-5 inputs. It must match the address assigned to the inputs in the hardware configuration.
LADDR_Out	DINT	Logical address of the FM 352-5 outputs and must match the address assigned to the outputs in hardware configuration.
CPU_Out	POINTER	Points to the 14-byte structure which is the source for the data to be transferred to the module as CPU outputs. The structure should match the structure defined in the application FB.
CPU_In	POINTER	Points to the 14-byte structure which is the destination for the data to be transferred from the module as CPU inputs. The structure should match the structure defined in the application FB.
Error	BOOL	This bit is set if the module is configured for testing and called in normal mode or vice versa. Detailed information can be found in the "Status" parameter.
Status	INT	This parameter contains the status word output by the module (see section "User data interface (Page 201)", heading "Definitions of the Control Bytes and Status Bytes").
AppFB*	Block_FB	The number of the application FB for the FM 352-5 module that are used in test mode.
AppInstDB*	Block_DB	The number of the instance DB of the FB for the FM 352-5 module, used in Test mode.
* This parameter is used only in the "FM Interface Debug" FB in test mode.		

CPU_Out structure

The following table shows an example of the 14-byte structure that transfers data from the CPU to the FM 352-5 module. In the sample interface FB, this structure is called by the pointer DB5.DBB0 that calls data block 5 (see table below).

Table 6- 11 Example Declaration Table for the Application FB, Input Section (as displayed in STEP 7 V5.1)

Address	Declaration	Name	Type
Input section: Bytes 2 through 15 are data from the CPU for the FM 352-5 module.			
2.0	in	CPU_Out	STRUCT
+0.0	in	Bits	ARRAY [0..15]
*0.1	in		BOOL
+2.0	in	T1_PV	DINT
+6.0	in	T2_PV	BYTE
+7.0	in	CmpByte	BYTE
+8.0	in	C1_PV	INT
+10.0	in	CP_Period	WORD
+12.0	in	CMPInt	INT
=14.0	in		END_STRUCT

Table 6- 12 Example of a data block - DB5.DBB0 (as in STEP 7 V5.1)

Address	Name	Type	Output value
0.0		STRUCT	
+0.0	Bits	ARRAY [0..15]	
*0.1		BOOL	
+2.0	T1_PV	DINT	L#0
+6.0	T2_PV	BYTE	B#16#0
+7.0	CmpByte	BYTE	B#16#0
+8.0	C1_PV	INT	0
+10.0	CP_Period	WORD	W#16#0
+12.0	CMPInt	INT	0
=14.0		END_STRUCT	

CPU_In structure

The following table shows an example of the 14-byte structure that returns data from the FM 352-5 module to the CPU. In the sample interface FB, this structure is called by the pointer DB6.DBB0 that calls data block 6 (see table below).

Table 6- 13 Example declaration table for the application FB, output section (as in STEP 7 V5.1)

Address	Declaration	Name	Type
Section of the outputs: The CPU inputs are the outputs from the FM 352-5 module to the CPU.			
18.0	out	CPU_In	STRUCT
+0.0	out	Bits	ARRAY [0..15]
*0.1	out		BOOL
+2.0	out	T2_CVasByte	BYTE
+3.0	out	C1_CVasByte	BYTE
+4.0	out	T2_CV	INT
+6.0	out	T1_CV	DINT
+10.0	out	Enc_CV1	DINT
=14.0	out		END_STRUCT

Table 6- 14 Example of a data block - DB6.DBB0 (as in STEP 7 V5.1)

Address	Name	Type	Output value
0.0		STRUCT	
+0.0	Bits	ARRAY [0..15]	
*0.1		BOOL	
+2.0	T2_CVasByte	BYTE	B#16#0
+3.0	C1_CVasByte	BYTE	B#16#0
+4.0	T2_CV	INT	0
+6.0	T1_CV	DINT	L#0
+10.0	Enc_CV1	DINT	L#0
=14.0		END_STRUCT	

6.4 Debugging a program

Downloading the program to the S7 CPU

Before you test your application FB, you should check the syntax using the "Syntax check" button in the "Configuration" dialog of the FM 352-5 on the "Programming" tab. Correct any syntax errors that may have been found during the check.

You must test your program in the STEP 7 environment so that you can monitor the execution of the program instructions.

To test your application FB using the S7 CPU with the FM 352-5 module in test mode, download the following elements to the CPU in addition to the blocks in your regular CPU program:

- Application FB containing the FM 352-5 program with the up-to-date instance DB.
- Interface FB for test mode of the FM with instance DB (FB 30/DB 30 in the FM 352-5 library)

To download the program to the S7 CPU, follow the steps outlined below:

1. In HW Config, select the menu command "Station > Save and Compile" to save and compile the hardware configuration.
2. In the SIMATIC Manager, download the S7 program Blocks folder (including the system data) to the S7 CPU.

Monitoring the Program Execution

STEP 7 provides several options for monitoring the execution of your program. Refer to STEP 7 documentation for information on how to use the program monitoring functions.

By using an iterative procedure when editing the application FB and downloading it again each time to check the execution results, you can check that the program meets your needs before downloading it to the FM 352-5 module.

Saving the Program to the CPU Project

After you are satisfied that the application FB executes correctly, save any changes you made to the application FB in the CPU project.

In the LAD/FBD editor, click the Save button or select the menu command "File > Save".

6.5 Download program to FM 352-5 module

Compiling the Application FB

To create the special SDB that contains the hardware configuration and the application FB in a form that can be read by the FPGA, you must compile the application FB for the FM 352-5 module. After creating and testing your application program, follow the steps below to compile the program and the hardware configuration in the SDB for the FM 352-5 module:

1. Open the FM 352-5 "Configuration" dialog, and select the "Programming" tab.
2. Click the "Compile" button.

Downloading the Program to the FM 352-5

After compiling the application FB for the FM 352-5 module, you can download the SDB to the FM 352-5 module. The FPGA derives its code from the image that is transferred during the download.

Requirement

- Use a new or reformatted SIMATIC Micro Memory Card for the FM 352-5 module (if this SIMATIC Micro Memory Card has been previously used outside an FM 352-5).
- A SIMATIC Micro Memory Card with 128 KB, 512 KB, or 2 MB of storage capacity is required to operate the FM 352-5 module.

Procedure

To download the SDB to the FM 352-5 module, follow these steps:

1. Open the FM 352-5 "Configuration" dialog, and select the "Programming" tab.
2. Click the "Download" button.

Downloading changes the FM 352-5 module to normal operation. When the download to the FM 352-5 begins, the module changes to STOP and copies the downloaded file to the FPGA and SIMATIC Micro Memory Card. The FM 352-5 module remains in normal operation when the download operation completes and remains in STOP even if the CPU user program continues to attempt to call the interface FB for Test mode (possible only in RUN).

Running the FM 352-5 module in normal operation

To switch FM 352-5 to RUN in normal operation, you must set the RUN/STOP selector to the RUN position, the calls to the interface FB for Test mode must be stopped, and the interface FB for normal operation (FB31 in the FM 352-5 library) must be called with a 1 signal at the Run input by the CPU user program. With this call, the FM 352-5 module begins executing the program that was downloaded to the FPGA. As long as the OneScan input has the 0 signal, the FM 352-5 continues to execute the program until one of the following events occur:

- The interface FB for Test mode is called and this changes the FM 352-5 module back to Test mode and restores the internal test program in the FPGA.
- The power is turned on again after an interruption, which restores the program contained in the SIMATIC Micro Memory Card in the FPGA, provided the program is valid; otherwise, the internal test program is restored.
- You perform a memory reset as described in section "Memory functions (Page 108)", which restores the program contained in the SIMATIC Micro Memory Card in the FPGA, provided the program is valid.

Cyclic execution on the FM 352-5 module in normal operation

You can set the single scan cycle for the FM 352-5 in normal operation by calling the interface FB for normal operation with a 1 signal at the OneScan input and changing the signal at the Run input from 0 to 1. Each time the Run input changes signal 1, the FM 352-5 executes one scan cycle.

Save the application FB of the FM 352-5 to a SIMATIC Micro Memory Card

You can make additional copies of the FM 352-5 program on SIMATIC Micro Memory Cards by using an EPROM programming device, such as the one built into the SIMATIC PG.

Requirement

- Use a new or reformatted SIMATIC Micro Memory Card for the FM 352-5 module (if this SIMATIC Micro Memory Card has been previously used outside an FM 352-5).
- A SIMATIC Micro Memory Card with 128 KB, 512 KB, or 2 MB of storage capacity is required to operate the FM 352-5 module.

Procedure

To copy the program of the FM 352-5 to the SIMATIC Micro Memory Card, follow these steps:

1. Insert the required SIMATIC Micro Memory Card into the EPROM programming device.
2. Select "S7-Memory Card"  in SIMATIC Manager, or the File **>** S7-Memory Card **>** Open command to open the "S7-Memory Card" window.
3. Copy the FM 352-5 system data folder containing SDB 32512 from the Blocks folder of the FM 352-5 program to the SIMATIC Micro Memory Card.

After copying the program to the SIMATIC Micro Memory Card, you can insert it in the slot of an FM 352-5 module. When the module starts up, it loads the FPGA program from the SIMATIC Micro Memory Card and changes to normal operation.

6.6 Stand-alone operation

Requirements

Stand-alone operation with the FM 352-5 module is possible only after you have completed your program development within the STEP 7 environment and copied a valid program and hardware configuration to the SIMATIC Micro Memory Card using the memory card programmer built into a Siemens PG or an EPROM programming device connected to a PC.

If a programmed SIMATIC Micro Memory Card is inserted in the FM 352-5 module, the module can become a stand-alone CPU, as long as stand-alone operation is enabled in the configuration software and no I/O backplane bus is detected. During stand-alone operation, the following functions are not supported:

- Diagnostic and process interrupts (SF LED is illuminated for diagnostic faults if this function is enabled in the hardware configuration on the SIMATIC Micro Memory Card).
- CPU_In data (including status).
- CPU_Out data (including control); all access to CPU_Out data is interpreted as 0.

Executing the Program

At startup, the FPGA reads the image of the FB stored on the SIMATIC Micro Memory Card card and can execute the program if the mode selector on the module is set to RUN mode (see figure below).

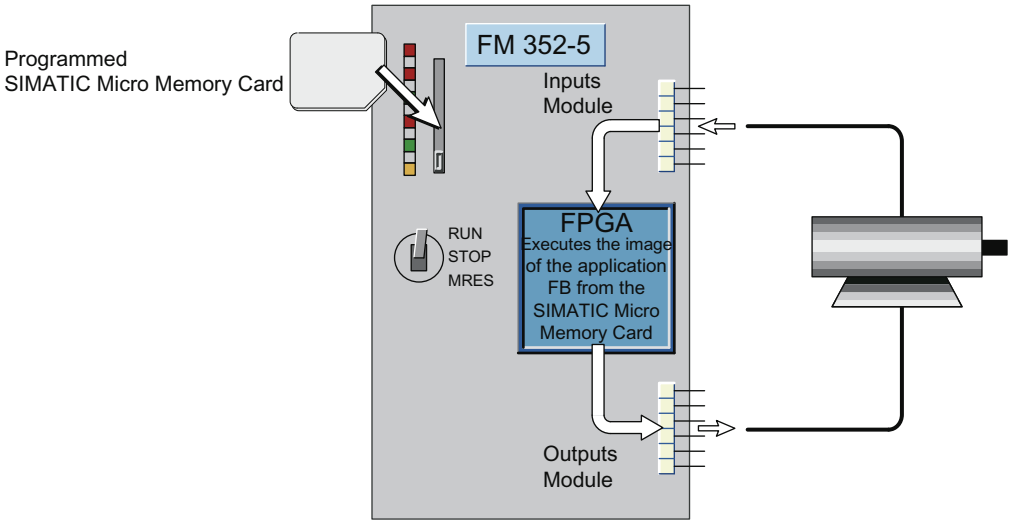


Figure 6-18 Stand-alone operation

6.7 Controlling dynamic parameters

Using System Function 55 to Write Dynamic Parameters

With SFC55 "WR_PARM" (write parameters), you can modify the dynamic parameters in data record 1 and transfer them to the FM 352-5 module. These parameters take effect when SFC 55 is called. The parameters transferred to the module do not, however, overwrite the parameters of the module in the corresponding SDB if they exist there. After the CPU changes from RUN to STOP and from STOP to RUN again or after the CPU is turned off and on again, the original parameters are effective again.

Parameterization Data Record 1 Dynamic Parameters

The dynamic parameters of data record 1 include the enabled diagnostics interrupts and enabled hardware interrupts. The following table defines the dynamic parameters in data record 1 that you can modify with SFC 55.

Table 6- 15 Parameterization Data Record 1

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	M1L	M2L	ESSF	M3L				
1	SSIF	DBW						
2	O7	O6	O5	O4	O3	O2	O1	O0
3	MMC							
4	PAE7	PAE6	PAE5	PAE4	PAE3	PAE2	PAE1	PAE0
5								
6								
7								

Table legend:

Name	Description of enabling interrupts	Value
M1L:	Missing auxiliary supply voltage (1L)	0 = disable 1 = enable
M2L:	Missing input/output supply voltage (2L)	0 = disable 1 = enable
ESSF:	Encoder sensor supply fault (overload)	0 = disable 1 = enable
M3L:	Missing encoder supply voltage (3L)	0 = disable 1 = enable
SSIF:	SSI frame error	0 = disable 1 = enable
DBW:	Wire break symm. RS-422 incremental encoder	0 = disable 1 = enable
O7-O0:	Output overload (can be enabled individually)	0 = disable 1 = enable

Name	Description of enabling interrupts	Value
MMC:	Diagnostics for SIMATIC Micro Memory Card	0 = disable 1 = enable
PII:	Hardware interrupt (can be enabled individually)	0 = disable 1 = enable
Note: Unused bits are reserved and must be set to 0.		

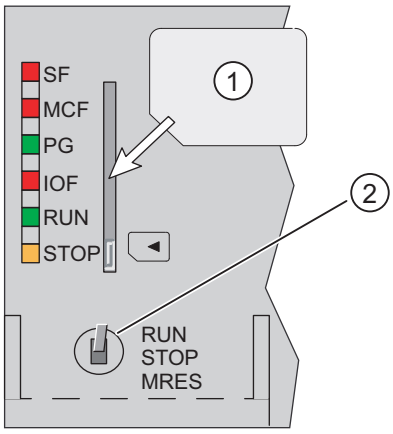
6.8 Memory functions

Resetting the Memory

Resetting the memory of the FM 352-5 causes the FPGA to read the image from the SIMATIC Micro Memory Card. No program memory contents are retained. All outputs are turned off, and counters and timers are reset.

To reset the memory of the FM 352-5 module, follow the steps outlined below:

1. Set the mode switch on the module to the STOP position.
2. Set the mode selector to the MRES position (see figure below) and hold it until the STOP status LED goes off and back on (about 3 seconds).
3. Release the mode selector allowing it to return to the STOP position.
4. Set the mode selector to the MRES position and hold it until the STOP status LED stops flashing.



(1) Programmed SIMATIC Micro Memory Card

(2) Mode selector switch

Figure 6-19 Resetting the Memory

Note

The memory reset position (MRES) is spring-loaded with no detent.

To reset memory:

1. Set the mode selector to STOP.
2. Set the selector to MRES and hold it there for 3 seconds.
3. Release the selector.
4. Set the selector to MRES and hold it there until the LED stops flashing.

Removing the SIMATIC Micro Memory Card during operation

You can remove the SIMATIC Micro Memory Card while the module is in RUN mode without having an impact on the operation of the module as long as the power is not interrupted. You can also switch between the module modes RUN and STOP when the SIMATIC Micro Memory Card is not inserted as long as the power is not interrupted. If there is a power loss, the FM 352-5 module changes to STOP and cannot return to RUN mode until a valid SIMATIC Micro Memory Card is inserted.

The SF LED and MCF LED are lit when the SIMATIC Micro Memory Card is removed from the module. The MCF fault only clears after the module has verified that a new SIMATIC Micro Memory Card is valid. Verification occurs when: The SIMATIC Micro Memory Card is loaded from STEP 7 or when the module is started up or reset.

6.9 Instruction Set for LAD Programming

Introduction

The following operations are supported by the Ladder Logic editor and instruction browser of STEP 7. The bit-logic instructions (contacts and coils) and some additional operations come from the standard operations of STEP 7. The FM 352-5-specific function blocks are available in the FM 352-5 Library.

STEP 7 operations for the FM 352-5

The following table lists the symbolic names and descriptions of the STEP 7 operations available for the FM 352-5.

Note

The status word is not available and is not updated by the FM 352-5.

Table 6- 16 STEP 7 operations for the FM 352-5

Symbolic name	Description
MOVE	Move a specified value (Page 114)
I_DI	Convert integer (16 bit) to double integer (32 bit) (Page 114)
SR	Set/reset flip-flop (Page 115)
RS	Reset/set flip-flop (Page 115)
-(P)-	Detect positive RLO edge (Page 116)
-(N)-	Detect negative RLO edge (Page 116)
POS	Positive edge detection (Page 117)
NEG	Negative edge detection (Page 117)
CMP	Comparison function (Page 118)
INV_I	Generate one's compliment for 16-bit integer (Page 119)
INV_DI	Generate one's compliment for 32-bit double integer (Page 120)
WAND_W	AND word operation (Page 121)
WOR_W	OR word operation (Page 122)
WXOR_W	Exclusive OR word operation (Page 123)
WAND_DW	AND double word operation (Page 124)
WOR_DW	OR double word operation (Page 125)
WXOR_DW	Exclusive OR double word operation (Page 126)
SHR_I	Shift right 16-bit integer operation (Page 127)
SHR_DI	Shift right 32-bit integer operation (Page 128)
SHL_W	Shift left word operation (Page 129)
SHR_W	Shift right word operation (Page 130)

Symbolic name	Description
SHL_DW	Shift left double word operation (Page 131)
SHR_DW	Shift right double word operation (Page 132)
ROL_DW	Rotate left double word operation (Page 133)
ROR_DW	Rotate right double word operation (Page 134)

6.9.1 normally open input

Description

This operation is in the standard list of STEP 7 operations.

Table 6- 17 NO contact input

LAD representation	Parameters	Data type	Addresses	Description
<Address> 	<Address>	BOOL	Input	The address identifies the bit whose signal state is queried.

6.9.2 normally closed input

Description

This operation is in the standard list of STEP 7 operations.

Table 6- 18 Normally closed input

LAD representation	Parameter	Data type	Addresses	Description
<Address>	<Address>	BOOL	Input	The address identifies the bit whose signal state is queried.

6.9.3 Output coil

Description

This operation is in the standard list of STEP 7 operations.

Table 6- 19 Output Coil

LAD representation	Parameter	Data type	Addresses	Description
<Address>	<Address>	BOOL	Output	The address identifies the bit whose signal state is set.

6.9.4 NOT

Description

This operation is in the standard list of STEP 7 operations.

Table 6- 20 NOT

LAD representation	Parameter	Data type	Addresses	Description
<Address> ——┤NOT├——	—	—	—	Inverts signal flow (negates the RLO bit).

6.9.5 Midline output connector

Description

This operation is in the standard list of STEP 7 operations. You must label each connector with a unique element that is declared in the structure Conn.

Table 6- 21 Midline output connector

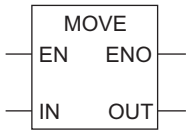
LAD representation	Parameter	Data type	Addresses	Description
<Conn.*label*> ——(#)——	Conn.*label*	BOOL	Conn.*label*	An intermediate assigning element which saves the RLO bit (power flow status) to a specified element in the Conn structure The midline output element saves the logical result of the preceding branch elements.

6.9.6 MOVE

Description

This operation is in the standard list of STEP 7 operations. The value specified at the IN input is copied to the address specified at the OUT output. With logic for EN, the MOVE value is retentive, requiring storage and a phase clock.

Table 6- 22 MOVE

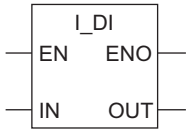
LAD representation	Parameter	Data type	Addresses	Description
	IN	All data types with a length of 8, 16, or 32 bits.	Input	Source value
	OUT	All data types with a length of 8, 16, or 32 bits.	Output	Destination address of the value specified at the IN input

6.9.7 Convert integer (16 bits) to double (32 bits) integer (I_DI)

Description

This operation is in the standard list of STEP 7 operations. I_DI reads the content of the IN parameter as an integer (16 bits) and converts it to a double integer (32 bits). The result is output by the OUT parameter.

Table 6- 23 Convert integer (16 bits) to double (32 bits) integer (I_DI)

LAD representation	Parameter	Data type	Addresses	Description
	IN	INT	Input	Integer value (16 bits) to convert
	OUT	DINT	Output	Result: Double integer (32 bits)

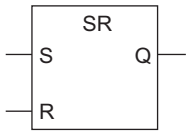
6.9.8 Set/reset flip-flop (SR)

Description

This operation is in the standard list of STEP 7 operations. You must label each SR operation with a unique element that is declared in the FF structure.

SR (set/reset flip-flop) is set if the signal state is 1 at the S input and 0 at the R input. SR is reset if the signal state is 0 at the S input and 1 at the R input. If the RLO is 1 at both inputs, SR is reset.

Table 6- 24 Set/reset flip-flop (SR)

LAD representation	Parameter	Data type	Addresses	Description
<FF.label>  	S	BOOL	Input	Enables set operation
	R	BOOL	Input	Enables reset operation
	Q	BOOL	Output	Signal state of output
	FF.label	BOOL	—	FF identifier

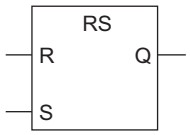
6.9.9 Reset/set flip-flop (RS)

Description

This operation is in the standard list of STEP 7 operations. You must label each RS operation with a unique element that is declared in the FF structure.

RS (reset/set flip-flop) is reset if the signal state is 1 at the R input and 0 at the S input. It is set if the signal state is 0 at the R input and 1 at the S input. If the RLO is 1 at both inputs, RS is set.

Table 6- 25 Reset/set flip-flop (RS)

LAD representation	Parameter	Data type	Addresses	Description
<FF.label>  	R	BOOL	Input	Enables reset operation
	S	BOOL	Input	Enables set operation
	Q	BOOL	Output	Signal state of output
	FF.label	BOOL	—	FF identifier

6.9.10 Detect positive edge —(P)—

Description

This operation is in the standard list of STEP 7 operations.

—(P)— (detect positive edge) detects a signal change in the **<address>** from 0 to 1 and displays it as RLO = 1 after the operation. The current signal state in the RLO is compared with the signal state of the address, the edge memory bit. If the signal state of the address is 0 and the RLO was 1 before the operation, the RLO will be 1 (pulse) after the operation, and 0 in all other cases. The RLO prior to the operation is stored in the address.

Table 6- 26 Detect positive RLO edge

LAD representation	Parameter	Data type	Addresses	Description
<Address> ———(P)———	<Address>	BOOL	Edge.*label*	Edge memory bit that stores the previous signal state of RLO

6.9.11 Detect negative edge —(N)—

Description

This operation is in the standard list of STEP 7 operations.

—(N)— (detect negative edge) detects a signal change in the **<address>** from 1 to 0 and displays it as RLO = 1 after the operation. The current signal state in the RLO is compared with the signal state of the address, the edge memory bit. If the signal state of the address is 1 and the RLO was 0 before the operation, the RLO will be 1 (pulse) after the operation, and 0 in all other cases. The RLO prior to the operation is stored in the address.

Table 6- 27 Detect negative RLO edge

LAD representation	Parameter	Data type	Addresses	Description
<Address> ———(N)———	<Address>	BOOL	Edge.*label*	Edge memory bit that stores the previous signal state of RLO

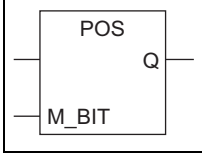
6.9.12 Detect signal positive edge (POS)

Description

This operation is in the standard list of STEP 7 operations. You must label the M_BIT input with a unique element that is declared in the Edge structure.

POS (detect positive signal edge) compares the signal state of <address> with the signal state from the previous scan cycle that is stored in M_BIT. If the current RLO state before the operation is 1 and the state of the <address> bit is 1, and the previous state of the bit was 0 (detection of rising edge), the RLO bit will be 1 after this operation.

Table 6- 28 Detect signal positive edge (POS)

LAD representation	Parameters	Data type	Addresses	Description
<Address>  	Q	BOOL	Output	One-shot output
	<Address>	BOOL	Input	Scanned signal
	M_BIT	BOOL	Edge.*label*	Edge memory bit that stores the previous signal state of <address>

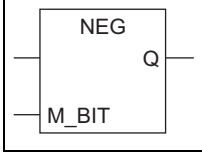
6.9.13 Detect negative signal edge (NEG)

Description

This operation is in the standard list of STEP 7 operations. You must label the M_BIT input with a unique element that is declared in the Edge structure.

NEG (detect negative signal edge) compares the signal state of <address> with the signal state from the previous scan cycle that is stored in M_BIT. If the current RLO state before the operation is 1 and the state of the <address> bit is 0 and the previous state of that bit was 1 (detect negative edge), the RLO bit will be 1 after this operation.

Table 6- 29 Detect negative signal edge (NEG)

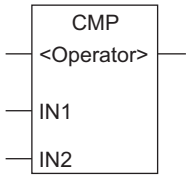
LAD representation	Parameter	Data type	Addresses	Description
<Address>  	Q	BOOL	Output	One-shot output
	<Address>	BOOL	Input	Scanned signal
	M_BIT	BOOL	Edge.*label*	Edge memory bit that stores the previous signal state of <address>

6.9.14 Comparison function (CMP)

Description

This operation is in the standard list of STEP 7 operations. The operation can be programmed with 16-bit or 32-bit values. The comparison function can be used like a normal contact. It can be located at any position where a normal contact could be placed. IN1 and IN2 are compared according to the type of comparison you choose. If the comparison is true, the RLO of the function is 1.

Table 6- 30 Comparison function (CMP)

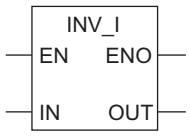
LAD representation	Parameter	Data type	Addresses	Description
	IN1	INT, DINT	Input, constant	First comparison value
	IN2	INT, DINT	Input, constant	Second comparison value
	Operator			Relational operator
	IN1 is equal to IN2			= =
	IN1 is not equal to IN2			< >
	IN1 is greater than IN2			>
	IN1 is less than IN2			<
	IN1 is greater than or equal to IN2			> =
	IN1 is less than or equal to IN2			< =

6.9.15 Generate one's complement for 16-bit integer (INV_I)

Description

The INV_I operation reads the content of the IN parameter and performs a EXCLUSIVE OR function with the hexadecimal mask W#16#FFFF. This operation changes every bit to its opposite state. ENO always has the same signal state as EN. With logic for EN, the INV_I value is retentive, requiring storage and a phase clock.

Table 6- 31 Generate one's complement for 16-bit integer (INV_I)

LAD representation	Parameter	Data type	Addresses	Description
	EN	BOOL	Input	Enable input
	ENO	BOOL	Output	Enable output
	IN	INT	Input	Integer input value (16 bits)
	OUT	INT	Output	Ones complement of the 16-bit integer IN

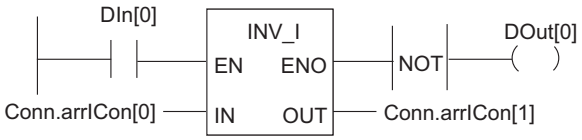


Figure 6-20 Example of the INV_I Operation

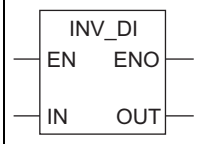
If DIn[0] = "1", each bit of Conn.arrICon[0] is inverted, for example:
Conn.arrICon[0] = 01000001 10000001 becomes Conn.arrICon[1] = 10111110 01111110.
Output DOut[0] is "1" if the inversion is not performed (ENO = EN = 0).

6.9.16 Generate one's compliment 32-bit double integer (INV_DI)

Description

The INV_DI operation reads the content of the IN parameter and performs a EXCLUSIVE OR function with the hexadecimal mask W#16#FFFF FFFF. This operation changes every bit to its opposite state. ENO always has the same signal state as EN. With logic for EN, the INV_DI value is retentive, requiring storage and a phase clock.

Table 6- 32 Generate one's compliment 16-bit double integer (INV_DI)

LAD representation	Parameter	Data type	Addresses	Description
	EN	BOOL	Input	Enable input
	ENO	BOOL	Output	Enable output
	IN	DINT	Input	Double integer input value, 32 bits
	OUT	DINT	Output	Ones complement of the 32-bit integer IN

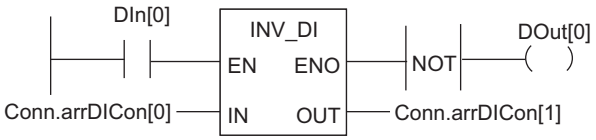


Figure 6-21 Example of the INV_DI Instruction

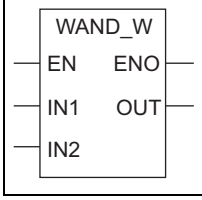
If DIn[0] = "1", each bit of Conn.arrDICon[0] is inverted, for example:
Conn.arrDICon[0] = F0FF FFF0 becomes Conn.arrDICon[1] = 0F00 000F.
Output DOut[0] is "1" if the inversion is not performed (ENO = EN = 0).

6.9.17 WAND_W (word) AND words

Description

The WAND_W (AND words) operation is activated by signal state "1" at the enable (EN) input and ANDs the two word values at IN1 and IN2 bit by bit. The values are interpreted as pure bit patterns. The result can be scanned at the OUT output. ENO has the same signal state as EN. With logic for EN, the WAND_W value is retentive, requiring storage and a phase clock.

Table 6- 33 WAND_W (WORD) AND words

LAD representation	Parameter	Data type	Addresses	Description
	EN	BOOL	Input	Enable input
	ENO	BOOL	Output	Enable output
	IN1	WORD	Input	First value of the logic operation
	IN2	WORD	Input	Second value of the logic operation
	OUT	WORD	Output	Result word of the logic operation

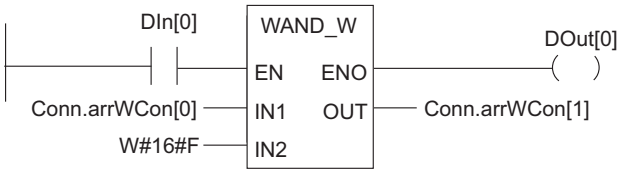


Figure 6-22 Example of the WAND_W (AND Words) Instruction

The operation is executed if DIn[0] = "1". Only bits 0 to 3 of Conn.arrWCon[0] are relevant, the remaining bits of Conn.arrWCon[0] are masked by the IN2 word bit pattern:

Example	
Conn.arrWCon[0]	= 01010101 01010101
IN2	= 00000000 00001111
Conn.arrWCon[0] AND IN2 = Conn.arrWCon[1]	= 00000000 00000101
DOut[0] is "1" if the operation is executed.	

6.9.18 WOR_W (word) OR words

Description

The WOR_W (word) OR word operation is activated by signal state "1" at the enable (EN) input and ORs the two word values at IN1 and IN2 bit by bit. The values are interpreted as pure bit patterns. The result can be scanned at the OUT output. ENO has the same signal state as EN. With logic for EN, the WOR_W value is retentive, requiring storage and a phase clock.

Table 6- 34 WOR_W (word) OR words

LAD representation	Parameter	Data type	Addresses	Description
WOR_W EN ENO IN1 OUT IN2	EN	BOOL	Input	Enable input
	ENO	BOOL	Output	Enable output
	IN1	WORD	Input	First value of the logic operation
	IN2	WORD	Input	Second value of the logic operation
	OUT	WORD	Output	Result word of the logic operation

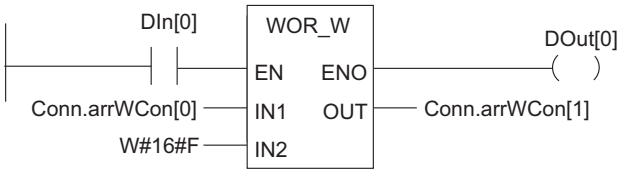


Figure 6-23 Example of the WOR_W (Word) OR Word Instruction

The operation is executed if DIn[0] is "1". Bits 0 to 3 are set to "1", all other Conn.arrWCon[0] bits are not changed.

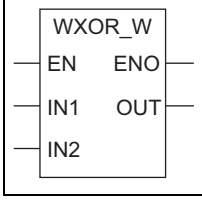
Example	
Conn.arrWCon[0]	= 01010101 01010101
IN2	= 00000000 00001111
Conn.arrWCon[0] OR IN2 = Conn.arrWCon[1]	= 01010101 01011111
DOut[0] is "1" if the operation is executed.	

6.9.19 WXOR_W (word) Exclusive OR words

Description

The WXOR_W (word) Exclusive OR word operation is activated by signal state "1" at the enable (EN) input and XORs the two word values at IN1 and IN2 bit by bit. The values are interpreted as pure bit patterns. The result can be scanned at the OUT output. ENO has the same signal state as EN. With logic for EN, the WXOR_W value is retentive, requiring storage and a phase clock.

Table 6- 35 WXOR_W (word) Exclusive OR words

LAD representation	Parameter	Data type	Addresses	Description
	EN	BOOL	Input	Enable input
	ENO	BOOL	Output	Enable output
	IN1	WORD	Input	First value of the logic operation
	IN2	WORD	Input	Second value of the logic operation
	OUT	WORD	Output	Result word of the logic operation

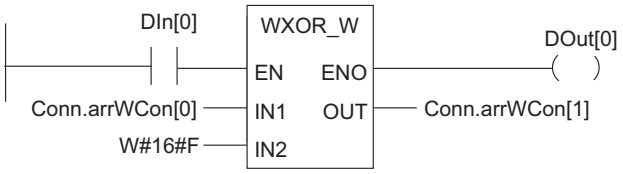


Figure 6-24 Example of the WXOR_W (Word) Exclusive OR Word Instruction

The operation is executed if DIn[0] is "1".

Example	
Conn.arrWCon[0]	= 01010101 01010101
IN2	= 00000000 00001111
Conn.arrWCon[0] XOR IN2 = Conn.arrWCon[1]	= 01010101 01011010
DOut[0] is "1" if the operation is executed.	

6.9.20 WAND_DW (word) AND double words

Description

The WAND_DW (word) AND double word operation is activated by signal state "1" at the enable (EN) input and ANDs the two word values at IN1 and IN2 bit by bit. The values are interpreted as pure bit patterns. The result can be scanned at the OUT output. ENO has the same signal state as EN. With logic for EN, the WAND_DW value is retentive, requiring storage and a phase clock.

Table 6- 36 WAND_DW (word) AND double words

LAD representation	Parameter	Data type	Addresses	Description
WAND_DW EN ENO IN1 OUT IN2	EN	BOOL	Input	Enable input
	ENO	BOOL	Output	Enable output
	IN1	DWORD	Input	First value of the logic operation
	IN2	DWORD	Input	Second value of the logic operation
	OUT	DWORD	Output	Result double word of logic operation

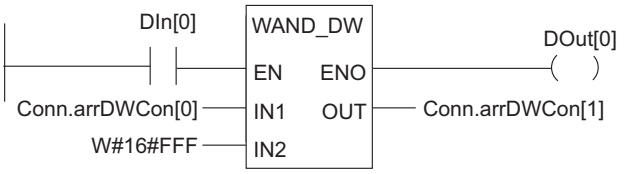


Figure 6-25 Example of the WAND_DW (Word) AND Double Word Instruction

The operation is executed if DIn[0] is "1". Only bits 0 through 11 of Conn.arrDWCon[0] are relevant, the remaining bits of Conn.arrDWCon[0] are masked by the IN2 bit pattern:

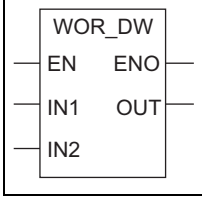
Example	
Conn.arrDWCon[0]	= 01010101 01010101 01010101 01010101
IN2	= 00000000 00000000 00001111 11111111
Conn.arrDWCon[0] AND IN2 = Conn.arrDWCon[1]	= 00000000 00000000 00000101 01010101
DOut[0] is "1" if the operation is executed.	

6.9.21 WOR_DW (word) OR double words

Description

The WOR_DW (word) OR double word operation is activated by signal state "1" at the enable (EN) input and ORs the two word values at IN1 and IN2 bit by bit. The values are interpreted as pure bit patterns. The result can be scanned at the OUT output. ENO has the same signal state as EN. With logic for EN, the WOR_DW value is retentive, requiring storage and a phase clock.

Table 6- 37 WOR_DW (word) OR double words

LAD representation	Parameter	Data type	Addresses	Description
	EN	BOOL	Input	Enable input
	ENO	BOOL	Output	Enable output
	IN1	DWORD	Input	First value of the logic operation
	IN2	DWORD	Input	Second value of the logic operation
	OUT	DWORD	Output	Result double word of logic operation

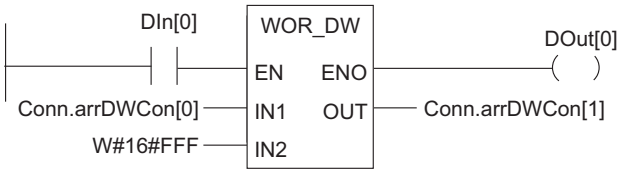


Figure 6-26 Example of the WOR_DW (Word) OR Double Word Instruction

The operation is executed if DIn[0] is "1". Bits 0 to 11 are set to "1", the remaining Conn.arrDWCon[0] bits are not changed.

Example	
Conn.arrDWCon[0]	= 01010101 01010101 01010101 01010101
IN2	= 00000000 00000000 00001111 11111111
Conn.arrDWCon[0] AND IN2 = Conn.arrDWCon[0]	= 01010101 01010101 01011111 11111111
DOut[0] is "1" if the operation is executed.	

6.9.22 WXOR_DW (word) Exclusive OR double words

Description

The WXOR_DW (word) Exclusive OR double words operation is activated by signal state "1" at the enable (EN) input and XORs the two word values at IN1 and IN2 bit by bit. The values are interpreted as pure bit patterns. The result can be scanned at the OUT output. ENO has the same signal state as EN. With logic for EN, the WXOR_DW value is retentive, requiring storage and a phase clock.

Table 6- 38 WXOR_DW (word) Exclusive OR double words

LAD representation	Parameter	Data type	Addresses	Description
WXOR_DW EN ENO IN1 OUT IN2	EN	BOOL	Input	Enable input
	ENO	BOOL	Output	Enable output
	IN1	DWORD	Input	First value of the logic operation
	IN2	DWORD	Input	Second value of the logic operation
	OUT	DWORD	Output	Result double word of logic operation

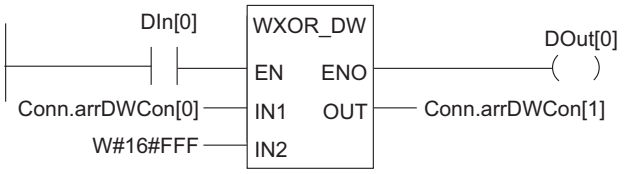


Figure 6-27 Example of the WXOR_DW (Word) Exclusive OR Double Word Instruction

The operation is executed if DIn[0] is = "1":

Example	
Conn.arrDWCon[0]	= 01010101 01010101 01010101 01010101
IN2	= 00000000 00000000 00001111 11111111
Conn.arrDWCon[1] = Conn.arrDWCon[0] XOR IN2	= 01010101 01010101 01010101 01010101
DOut[0] is "1" if the operation is executed.	

6.9.23 SHR_I shift right 16-bit integer

Description

The SHR_I shift right 16-bit integer operation is activated by signal state "1" at the Enable (EN) input. The SHR_I operation is used to shift bits 0 to 15 of input IN bit by bit to the right. Bits 16 to 31 are not affected. Input N specifies the number of bit positions to be shifted. If N is greater than 16, the command operates as if N = 16 was set. The bit positions shifted in from the left to fill vacated bit positions are assigned the signal state of bit 15 (sign bit of the integer). This means these bit positions are assigned "0" if the integer is positive and "1" if the integer is negative. The result of the shift operation can be queried at the OUT output. ENO has the same signal state as EN. With logic for EN, the SHR_I value is retentive, requiring storage and a phase clock.

Table 6- 39 SHR_I shift right 16-bit integer

LAD representation	Parameter	Data type	Addresses	Description
SHR_I EN ENO IN OUT N	EN	BOOL	Input	Enable input
	ENO	BOOL	Output	Enable output
	IN	INT	Input	Value to be shifted
	N	WORD	Input	Number of bit positions to be shifted
	OUT	INT	Output	Result of shift operation

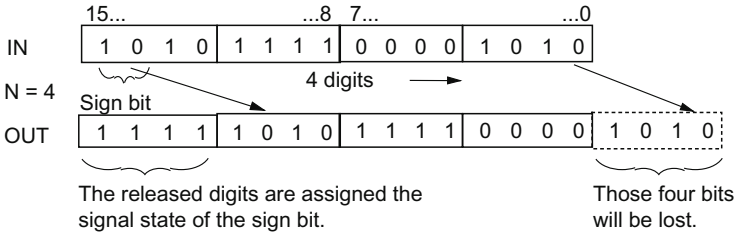


Figure 6-28 Example of Bit Shifts for the SHR_I Instruction

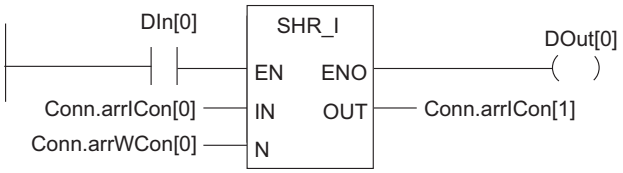


Figure 6-29 Example of the SHR_I Shift Right Integer Instruction

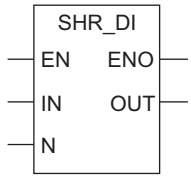
The SHR_I box is activated by "1" at DIn[0]. Conn.arrICon[0] is loaded and shifted right by the number of bits specified with Conn.arrWCon[0]. The result is written to Conn.arrICon[1]. DOut[0] is "1" if the operation is executed.

6.9.24 SHR_DI shift right 32-bit double integer

Description

The SHR_DI Shift right double integer operation is enabled by signal state "1" at the Enable (EN) input. The SHR_DI operation is used to shift bits 0 to 31 of input IN bit by bit to the right. Input N specifies the number of bit positions to be shifted. If N is greater than 32, the command operates as if N = 32 was set. The bit positions shifted in from the left to fill vacated bit positions are assigned the signal state of bit 31 (sign bit of the 32-bit integer). This means these bit positions are assigned "0" if the integer is positive and "1" if the integer is negative. The result of the shift operation can be queried at the OUT output. ENO has the same signal state as EN. With logic for EN, the SHR_DI value is retentive, requiring storage and a phase clock.

Table 6- 40 SHR_DI shift right 32-bit double integer

LAD representation	Parameter	Data type	Addresses	Description
	EN	BOOL	Input	Enable input
	ENO	BOOL	Output	Enable output
	IN	DINT	Input	Value to be shifted
	N	WORD	Input	Number of bit positions to be shifted
	OUT	DINT	Output	Result of shift operation

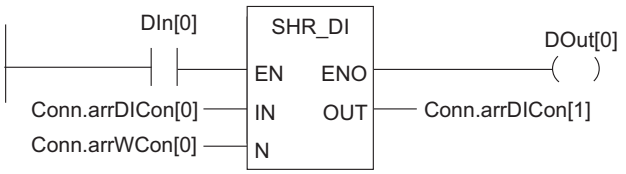


Figure 6-30 Example of the SHR_DI Shift Right Double Integer Instruction

The SHR_I box is activated by "1" at DIn[0]. Conn.arrDICon[0] is loaded and shifted right by the number of bits specified with Conn.arrWCon[0]. The result is written to Conn.arrDICon[1].

DOut[0] is "1" if the operation is executed.

6.9.25 SHL_W shift left word

Description

The SHL_W shift left word operation is activated by signal state "1" at the Enable (EN) input. The SHL_W operation is used to shift bits 0 to 15 of input IN bit by bit to the left. Bits 16 to 31 are not affected. Input N specifies the number of bit positions to be shifted. If N is higher than 16, the command writes a "0" at the OUT output. The same number (N) of zeros is shifted from the right in order to occupy the positions which have become free. The result of the shift operation can be queried at the OUT output. ENO has the same signal state as EN. With logic for EN, the SHL_W value is retentive, requiring storage and a phase clock.

Table 6- 41 SHL_W shift left word

LAD representation	Parameter	Data type	Addresses	Description
SHL_W EN ENO IN OUT N	EN	BOOL	Input	Enable input
	ENO	BOOL	Output	Enable output
	IN	WORD	Input	Value to be shifted
	N	WORD	Input	Number of bit positions to be shifted
	OUT	WORD	Output	Result of shift operation

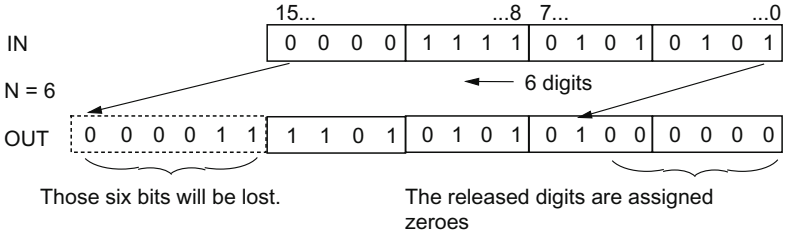


Figure 6-31 Example of Bit Shifts for the SHL_W Instruction

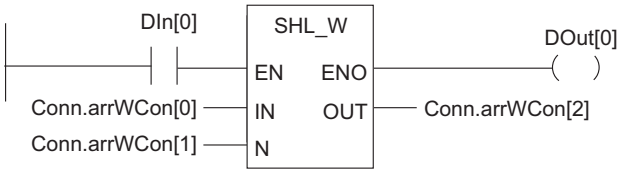


Figure 6-32 Example of the SHL_W Shift Left Word Instruction

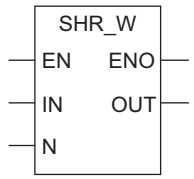
The SHL_W box is activated when "1" is set at DIn[0]. Conn.arrWCon[0] is loaded and shifted left by the number of bits specified with Conn.arrWCon[1]. The result is written to Conn.arrWCon[2]. DOut[0] is "1" if the operation is executed.

6.9.26 SHR_W shift right word

Description

The SHR_W shift right word operation is enabled by signal state "1" at the Enable (EN) input. The SHR_W operation is used to shift bits 0 to 15 of input IN bit by bit to the right. Bits 16 to 31 are not affected. Input N specifies the number of bit positions to be shifted. If N is higher than 16, the command writes a "0" at the OUT output. The same number (N) of zeros is shifted from the left in order to occupy the positions which have become free. The result of the shift operation can be queried at the OUT output. ENO has the same signal state as EN. With logic for EN, the SHR_W value is retentive, requiring storage and a phase clock.

Table 6- 42 SHR_W shift right word

LAD representation	Parameter	Data type	Addresses	Description
	EN	BOOL	Input	Enable input
	ENO	BOOL	Output	Enable output
	IN	WORD	Input	Value to be shifted
	N	WORD	Input	Number of bit positions to be shifted
	OUT	WORD	Output	Result of shift operation

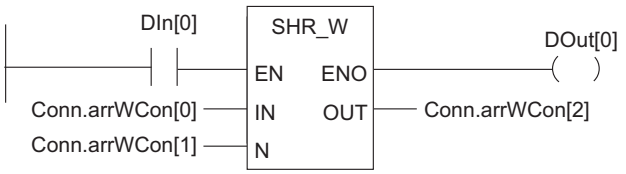


Figure 6-33 Example of the SHR_W Shift Right Word Instruction

The SHR_W box is activated when "1" is set at DIn[0]. Conn.arrWCon[0] is loaded and shifted right by the number of bits specified with Conn.arrWCon[1]. The result is written to Conn.arrWCon[2].

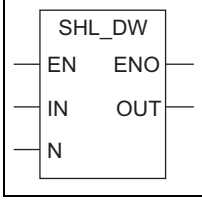
DOut[0] is "1" if the operation is executed.

6.9.27 SHL_DW shift left double word

Description

The SHL_DW shift left double word operation is enabled by signal state "1" at the Enable (EN) input. The SHL_DW operation is used to shift bits 0 to 31 of input IN bit by bit to the left. Input N specifies the number of bit positions to be shifted. If N is higher than 32, the command writes a "0" at the OUT output. The same number (N) of zeros is shifted from the right in order to occupy the positions which have become free. The result of the shift operation can be queried at the OUT output. ENO has the same signal state as EN. With logic for EN, the SHL_DW value is retentive, requiring storage and a phase clock.

Table 6- 43 SHL_DW shift left double word

LAD representation	Parameter	Data type	Addresses	Description
	EN	BOOL	Input	Enable input
	ENO	BOOL	Output	Enable output
	IN	DWORD	Input	Value to be shifted
	N	WORD	Input	Number of bit positions to be shifted
	OUT	DWORD	Output	Result of shift operation

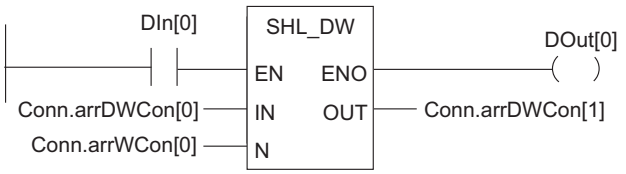


Figure 6-34 Example of the SHL_DW Shift Left Double Word Instruction

The SHL_DW box is enabled when "1" is set at DIn[0]. Conn.arrDWCon[0] is loaded and shifted left by the number of bits specified with Conn.arrWCon[0]. The result is written to Conn.arrDWCon[1].

DOut[0] is "1" if the operation is executed.

6.9.28 SHR_DW shift right double word

Description

The SHR_DW shift right double word operation is enabled by signal state "1" at the Enable (EN) input. The SHR_DW operation is used to shift bits 0 to 31 of input IN bit by bit to the right. Input N specifies the number of bit positions to be shifted. If N is larger than 32, the command writes a "0" at the OUT output and sets the bits CC 0 and OV in the status word to "0". The same number (N) of zeros is shifted from the left in order to occupy the positions which have become free. The result of the shift operation can be queried at the OUT output. ENO has the same signal state as EN. With logic for EN, the SHR_DW value is retentive, requiring storage and a phase clock.

Table 6- 44 SHR_DW shift right double word

LAD representation	Parameters	Data type	Address	Description
SHR_DW EN ENO IN OUT N	EN	BOOL	Input	Enable input
	ENO	BOOL	Output	Enable output
	IN	DWORD	Input	Value to be shifted
	N	WORD	Input	Number of bit positions to be shifted
	OUT	DWORD	Output	Result of shift operation

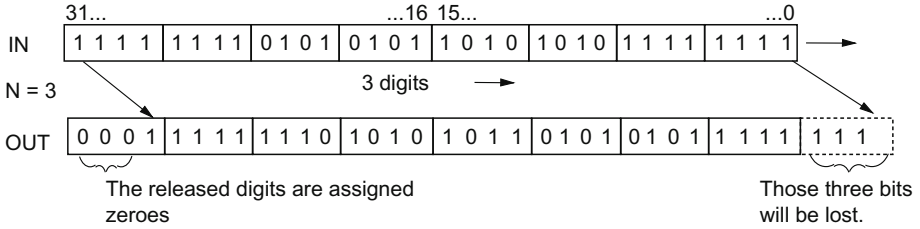


Figure 6-35 Example of Bit Shifts for the SHR_DW Shift Right Double Word Instruction

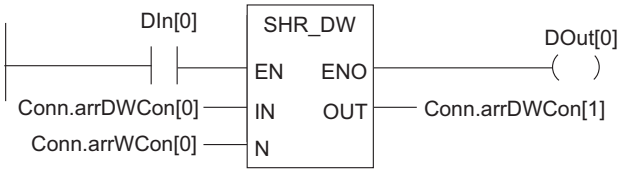


Figure 6-36 Example of the SHR_DW Shift Right Double Word Instruction

The SHR_DW box is enabled when "1" is set at DIn[0]. Conn.arrDWCon[0] is loaded and shifted right by the number of bits specified with Conn.arrWCon[0]. The result is written to Conn.arrDWCon[1]. DOut[0] is "1" if the operation is executed.

6.9.29 ROL_DW rotate left double word

Description

The ROL_DW rotate left double word operation is enabled by signal state "1" at the Enable (EN) input. The ROL_DW operation is used to rotate the entire contents of input IN bit by bit to the left. Input N specifies the number of bit positions for the rotation. If N is greater than 32, the double word IN is rotated by ((N-1) modulo 32)+1 positions. The bit positions coming from the right are occupied with the signal state of the bits which have been rotated to the left (left rotation). The result of the rotation operation can be queried at the OUT output. ENO has the same signal state as EN. With logic for EN, the ROL_DW value is retentive, requiring storage and a phase clock.

Table 6- 45 ROL_DW rotate left double word

LAD representation	Parameter	Data type	Addresses	Description
ROL_DW EN ENO IN OUT N	EN	BOOL	Input	Enable input
	ENO	BOOL	Output	Enable output
	IN	DWORD	Input	Value to be rotated
	N	WORD	Input	Number of bit positions to be rotated
	OUT	DWORD	Output	Result of rotation operation

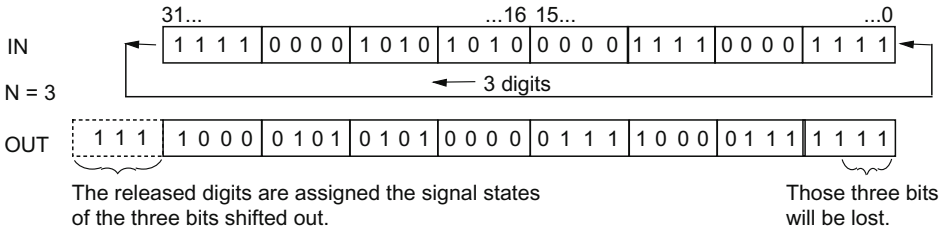


Figure 6-37 Example of Bit Shifts for the ROL_DW Rotate Left Double Word Instruction

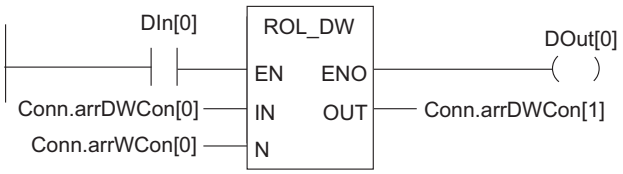


Figure 6-38 Example of the ROL_DW Rotate Left Double Word Instruction

The ROL_DW box is enabled when "1" is set at DIn[0]. Conn.arrDWCon[0] is loaded and rotated to the left by the number of bits specified with Conn.arrWCon[0]. The result is written to Conn.arrDWCon[1].

DOut[0] is "1" if the operation is executed.

6.9.30 ROR_DW rotate right double word

Description

The ROR_DW rotate right double word operation is enabled by signal state "1" at the Enable (EN) input. The ROR_DW operation is used to rotate the entire contents of input IN bit by bit to the right. Input N specifies the number of bit positions for the rotation. If N is greater than 32, the double word IN is rotated by ((N-1) modulo 32)+1 positions. The bit positions coming from the left are occupied by the signal state of the bits which have been rotated to the right (right rotation). The result of the rotation operation can be queried at the OUT output. ENO has the same signal state as EN. With logic for EN, the ROR_DW value is retentive, requiring storage and a phase clock.

Table 6- 46 ROR_DW rotate right double word

LAD representation	Parameter	Data type	Addresses	Description
ROR_DW EN ENO IN OUT N	EN	BOOL	Input	Enable input
	ENO	BOOL	Output	Enable output
	IN	DWORD	Input	Value to be rotated
	N	WORD	Input	Number of bit positions to be rotated
	OUT	DWORD	Output	Result of rotation operation

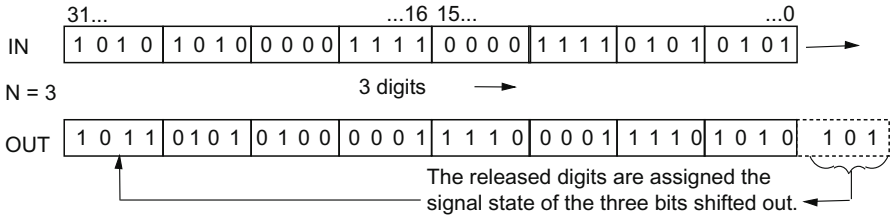


Figure 6-39 Example of Bit Shifts for the ROR_DW Rotate Right Double Word Instruction

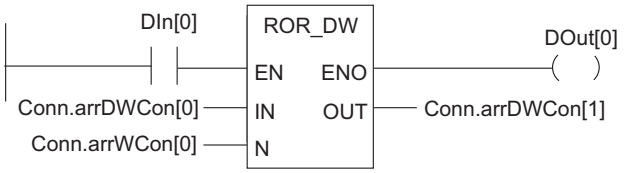


Figure 6-40 Example of the ROR_DW Rotate Right Double Word Instruction

The ROR_DW box is enabled when "1" is set at DIn[0]. Conn.arrDWCon[0] is loaded and rotated to the right by the number of bits specified with Conn.arrWCon[0]. The result is written to Conn.arrDWCon[1].

DOut[0] is "1" if the operation is executed.

6.10 Operations in the FM 352-5 Library

Overview

The following table lists the FBs from the FM 352-5 Library, their symbolic names, and includes a functional description of each. You can change the numbers of the FBs after you have copied them or as you copy them to the Blocks folder of your program.

Table 6- 47 FBs in the FM 352-5 library

FB number	Symbolic name	Description
FB 112	BiScale	Binary scaler (Page 137)
FB 116	TP16	16-bit pulse (Page 138)
FB 113	TP32	32-bit pulse (Page 138)
FB 117	TON16	16-bit on delay timer (Page 139)
FB 114	TON32	32-bit on delay timer (Page 139)
FB 118	TOF16	16-bit off delay timer (Page 140)
FB 115	TOF32	32-bit off delay timer (Page 140)
FB 119	CP_Gen	Clock pulse generator (Page 141)
FB 121	CTU16	16-bit up counter (Page 142)
FB 122	CTD16	16-bit down counter (Page 143)
FB 123	CTUD16	16-bit up/down counter (Page 144)
FB 120	CTUD32	32-bit up/down counter (Page 144)
FB 124	SHIFT	Bit shift register, 1 bit; maximum length = 4096 (Page 145)
FB 125	SHIFT2	Bit shift register, 2 bits; maximum length = 2048 (Page 145)
FB 126	SHIFT4	Bit shift register, 4 bits; maximum length = 1024 (Page 145)
FB 127	SHIFT8	Bit shift register, 8 bits; maximum length = 512 (Page 145)
FB 85	SHIFT16	INT shift register; maximum length = 256 (Page 145)
FB 84	SHIFT32	DINT shift register; maximum length = 256 (Page 145)
FB 104	FMABS32	Absolute value, 32 bits (Page 147)
FB 105	FMABS16	Absolute value, 16 bits (Page 147)
FB 110	DatSel32	Data selector, 32 bits (Page 147)
FB 111	DatSel16	Data selector, 16 bits (Page 147)
FB 106	FMAdd32	Add, 32 bits (Page 148)
FB 107	FMAdd16	Add, 16 bits (Page 148)
FB 108	FMSub32	Subtract, 32 bits (Page 148)
FB 109	FMSub16	Subtract, 16 bits (Page 148)
FB 100	FMMul32	Multiply, 32 bits (Page 149)
FB 101	FMMul16	Multiply, 16 bits (Page 150)
FB 102	FMDiv32	Divide, 32 bits (Page 151)
FB 103	FMDiv16	Divide, 16 bits (Page 152)

FB number	Symbolic name	Description
FB 79	ENCODE	Locates most significant bit set in a DWORD (Page 153)
FB 78	BITSUM	Counts set bits in a DWORD (Page 154)
FB 93	BitPack_W	Packs 16 digital bits into a WORD (Page 155)
FB 92	BitPack_DW	Packs 32 digital bits into a DWORD (Page 155)
FB 91	BitCast_W	Converts a WORD to 16 digital bits (Page 156)
FB 90	BitCast_DW	Converts a DWORD to 32 digital bits (Page 156)
FB 87	BitPick_W	Selects a bit from a WORD (Page 157)
FB 86	BitPick_DW	Selects a bit from a DWORD (Page 157)
FB 95	BitInsert16	Inserts a bit into an INT (16 bits) (Page 158)
FB 94	BitInsert32	Inserts a bit into a DINT (32 bits) (Page 158)
FB 89	BitShift_W	Bit shift register, length: 16 bits (Page 159)
FB 88	BitShift_DW	Bit shift register, length: 32 bits (Page 159)
FB 76	WordPack	Concatenates 2 WORDs into 1 DWORD (Page 160)
FB 77	WordCast	Converts 1 DWORD into 2 WORDs (Page 161)
FB 81	PERIOD16	Period measurement, 16 bits (Page 162)
FB 80	PERIOD32	Period measurement, 32 bits (Page 162)
FB 83	FREQ16	Frequency measurement, 16 bits (Page 163)
FB 82	FREQ32	Frequency measurement, 32 bits (Page 163)
FB 97	FIFO16	Delete first value, 16 bits (Page 164)
FB 96	FIFO32	Delete first value, 32 bits (Page 164)
FB 99	LIFO16	Delete last value, 16 bits (Page 166)
FB 98	LIFO32	Delete last value, 32 bits (Page 166)

6.10.1 Binary scaler (BiScale)

Description

The binary scaler (FB112) provides a way of producing a series of output pulses at half the rate of the input pulses.

Each rising edge at input C inverts the output Q effectively dividing the frequency of the input by two, as shown in the figure below.

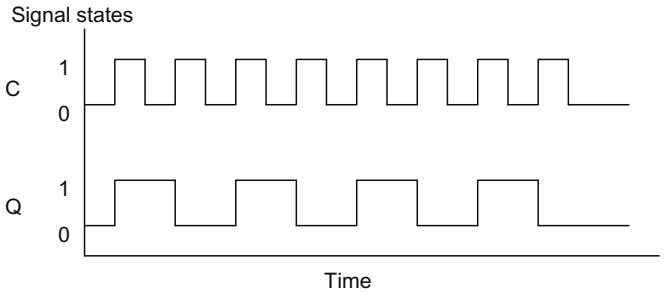
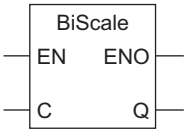


Figure 6-41 Timing Diagram for Binary Scaler (BiScale)

Table 6- 48 Binary scaler (BiScale)

LAD representation	Parameter	Data type	Addresses	Description
	C	BOOL	Input	Input to be converted.
	Q	BOOL	Output	Output of the function.
Note: No logic is allowed at the EN input.				

6.10.2 Pulse timers (TP16 and TP32)

Description

This timer is available in two versions: As a 16-bit (FB116) and a 32-bit (FB113) timer.

Pulse timers "TP16" and "TP32" generate a pulse with the length PT.

A rising signal edge at input IN starts the pulse. Output Q remains set for the time PT regardless of changes in the input signal (in other words even when the IN input changes back from 0 to 1 before the time PT has expired). The ET output provides the time for which output Q has already been set. The maximum value of the ET output is the value of the PT input. Output ET is reset when input IN changes to 0; however, not before the time PT has expired.

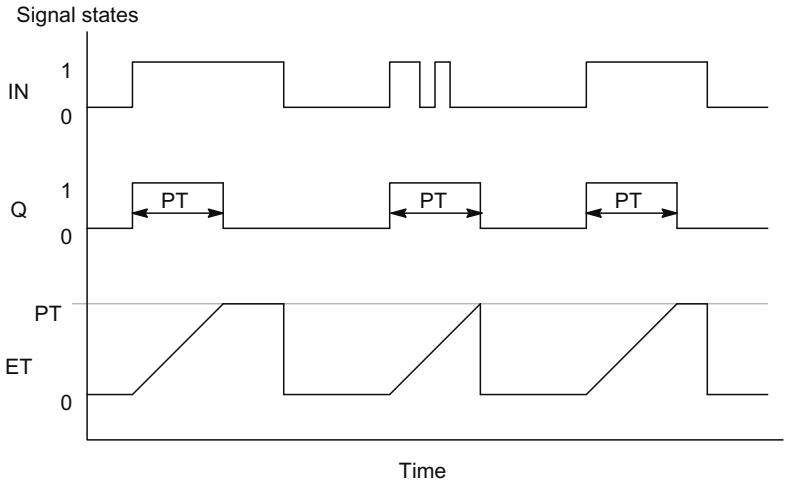


Figure 6-42 Timing Diagram for Pulse Timer (TP)

Table 6- 49 Pulse Timer (TP)

LAD representation	Parameter	Data type	Addresses	Description
TP EN ENO IN Q PT ET	IN	BOOL	Input	Start input.
	PT	INT, DINT	Input, constant	Duration of the pulse in 10 µs units. PT must be a positive constant.
	Q	BOOL	Output	Status of the time.
	ET	INT, DINT	Output	Elapsed time.
Note: No logic is allowed at the EN input.				

6.10.3 On-delay timers (TON16 and TON32)

Description

This timer is available in two versions: As a 16-bit (FB117) and a 32-bit (FB114) timer.

"TON16" and "TON32" delay a rising signal edge by the time PT.

A rising edge at the IN input causes a rising edge at output Q after the time PT has expired. Q then remains set until the IN input changes to 0 again. If the IN input changes to 0 before the time PT has expired, output Q remains set to 0.

The ET output provides the time that has passed since the last rising edge at the IN input. Its maximum value is the value of the PT input. ET is reset when the IN input changes to 0.

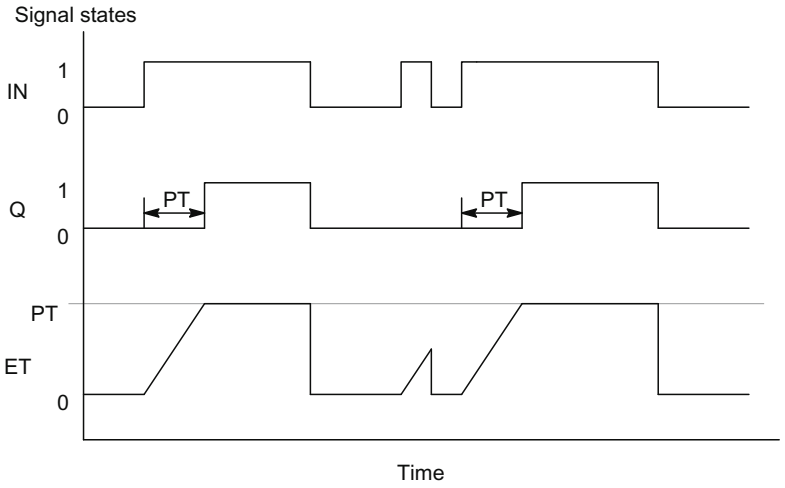


Figure 6-43 Timing Diagram for On-Delay Timer (TON)

Table 6- 50 On-Delay Timer (TON)

LAD representation	Parameter	Data type	Addresses	Description
TON EN ENO IN Q PT ET	IN	BOOL	Input	Start input.
	PT	INT, DINT	Input, constant	Length of the on delay in 10 µs units. PT must be a positive constant.
	Q	BOOL	Output	Status of the time.
	ET	INT, DINT	Output	Elapsed time.
Note: No logic is allowed at the EN input.				

6.10.4 Off-delay timers (TOF16 and TOF32)

Description

This timer is available in two versions: As a 16-bit (FB118) and a 32-bit (FB115) timer.

"TOF16" and "TOF32" delay a falling edge by the time PT.

A rising edge at the IN input causes a rising edge at output Q. A falling edge at the IN input causes a falling edge at output Q delayed by the time PT. If the IN input changes back to 1 before the time PT has expired, output Q remains set to 1. The ET output provides the time that has elapsed since the last falling edge at the IN input. Its maximum value is, however the value of the PT input. ET is reset when the IN input changes to 1.

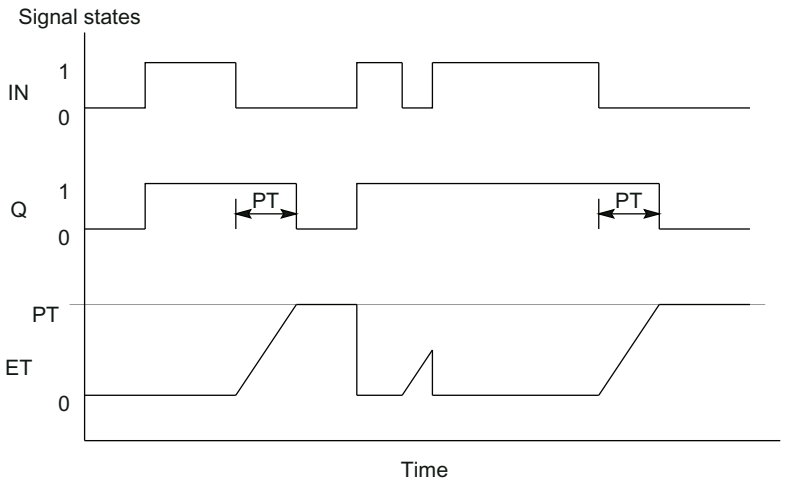


Figure 6-44 Timing Diagram for Off-Delay Timer (TOF)

Table 6- 51 Off-Delay Timer (TOF)

LAD representation	Parameter	Data type	Addresses	Description
TOF EN ENO IN Q PT ET	IN	BOOL	Input	Start input.
	PT	INT, DINT	Input, constant	Length of the off delay in 10 µs units. PT must be a positive constant.
	Q	BOOL	Output	Status of the time.
	ET	INT, DINT	Output	Elapsed time.
Note: No logic is allowed at the EN input.				

6.10.5 Clock pulse generator (CP_Gen)

Description

The clock pulse generator (FB119) allows you to output a pulse at a specified frequency from less than 1 Hz to a maximum of 50 kHz.

When the signal state at the ENABLE input is 1, a clock pulse is generated at the Q output, as shown in the figure below. The output frequency is specified by inverting the value of the word input (WORD) that is an unsigned integer represented as a hexadecimal value multiplied by 20 µs.

The frequency is equal to 50,000 ÷ PERIOD.

PERIOD is equal to 50,000 divided by the desired frequency. Example:

- When PERIOD = W#16#C350, a frequency of 1 Hz is output.
- When PERIOD = W#16#1, a frequency of 50 kHz is output.

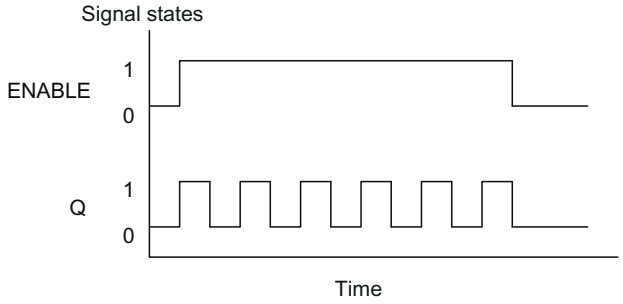


Figure 6-45 Timing Diagram for Clock Pulse Generator (CP_Gen)

Table 6- 52 Clock pulse generator (CP_Gen)

LAD representation	Parameter	Data type	Addresses	Description
CP_Gen EN ENO ENABLE Q PERIOD	ENABLE	BOOL	Input	Start input.
	Q	BOOL	Output	Status of the time.
	PERIOD	WORD	Constant or variable (connector or CPU_Out)	The number of 20 µs steps in the period.
Note: No logic is allowed at the EN input.				

6.10.6 Up counter (CTU16)

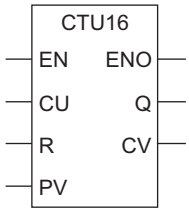
Description

You can count up with "CTU16" (FB121). The counter is incremented by 1 on a rising edge at the CU input. If the count value reaches the upper limit of 32767, it is no longer incremented. Each subsequent rising edge at the CU input no longer has an effect.

Signal state 1 at the R input resets the counter to the value 0 regardless of the value currently at the CU input.

The Q output indicates whether the current counted value is greater than or equal to the preset value PV.

Table 6- 53 Up counter (CTU16)

LAD representation	Parameter	Data type	Addresses	Description
	CU	BOOL	Input	Counter input.
	R	BOOL	Input	Reset input. R is dominant over CU.
	PV	INT	Input, constant	Preset value. Refer to parameter Q for the effect of PV.
	Q	BOOL	Output	Status of the counter: Q has the following value: • 1 if CV ≥ PV • 0 in all other situations
	CV	INT	Output	Current count value (possible value: 0 to 32767).

6.10.7 Down counter (CTD16)

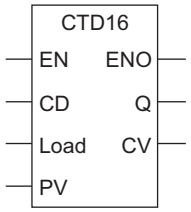
Description

You can count down with "CTD16" (FB122). The counter is decremented by 1 on a rising edge at the CD input. If the count value reaches the lower limit of -32768, it is no longer decremented. Any subsequent rising edge at the CD input no longer has an effect.

Signal state 1 at the LOAD input sets the counter to the preset value PV regardless of the value currently at the CD input.

The Q output indicates whether the current counted value is less than or equal to 0.

Table 6- 54 Down counter (CTD16)

LAD representation	Parameter	Data type	Addresses	Description
	CD	BOOL	Input	Counter input.
	Load	BOOL	Input	Load input. LOAD input is dominant over CD.
	PV	INT	Input, constant	Preset value. The counter is preset to PV when the signal level at the LOAD input is 1.
	Q	BOOL	Output	Status of the counter: Q has the following value: • 1 if CV ≤ 0 • 0 in all other situations
	CV	INT	Output	Current count value (possible value: -32768 to +32767).

6.10.8 Up/down counters (CTUD16 and CTUD32)

Description

The "CTUD" counter is available in two versions: As a 16-bit (FB123) and a 32-bit (FB120) up/down counter.

The count value is changed by a rising edge as follows:

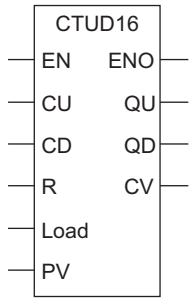
- The counted value is incremented by 1 on a rising edge at the CU input. If the count value reaches the upper limit, it is no longer incremented.
- The counted value is decremented by 1 on a rising edge at the CD input. If the count value reaches the lower limit, it is no longer decremented.

If there is a rising edge at both the CU and CD input in one cycle, the counter retains its current value.

A signal level 1 at the LOAD input presets the counter to the value PV regardless of the values at the CU and CD inputs.

The signal level 1 at the R input resets the counter to the value 0 regardless of the values at the CU, CD and LOAD inputs. The QU output indicates whether the current counted value is greater than or equal to the preset value PV. The QD output indicates whether the value is less than or equal to zero.

Table 6- 55 Up/Down Counter (CTUD)

LAD representation	Parameter	Data type	Addresses	Description
 (or CTUD32)	CU	BOOL	Input	Count up input.
	CD	BOOL	Input	Count down input.
	R	BOOL	Input	Reset input. R is dominant over CU.
	Load	BOOL	Input	Load input. LOAD input is dominant over CD.
	PV	INT, DINT	Input, constant	Preset value. The counter is preset to PV when the signal level at the LOAD input is 1.
	QU	BOOL	Output	Status of the counter: QU has the following value: • 1 if CV ≥ PV • 0 in all other situations
	QD	BOOL	Output	Status of the counter: QD has the following value: • 1 if CV ≤ 0 • 0 in all other situations
	CV	INT, DINT	Output	Current count value. Possible values: -32768 to +32767 for the 16-bit counter -2,147,483,648 to +2,147,483,647 for the 32-bit counter

6.10.9 Bit shift Registers (SHIFT, SHIFT2, SHIFT4, SHIFT8, SHIFT16 and SHIFT32)

Description

The "SHIFT" operation is available in six versions (FB124 through FB127, FB84, and FB85), defined by the number of simultaneously shifted bits.

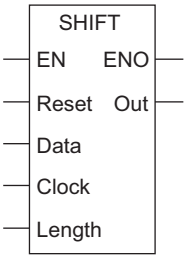
When the Clock input changes from 0 to 1, the value at the Data input is shifted into the first stage of the shift register and is shifted again on each subsequent Clock edge. The output is set by the last stage in the shift register. When the EN and Reset are both on, all of the stages of the shift register are reset to 0.

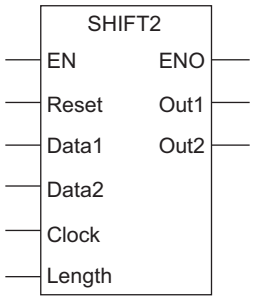
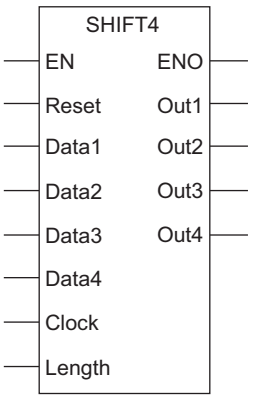
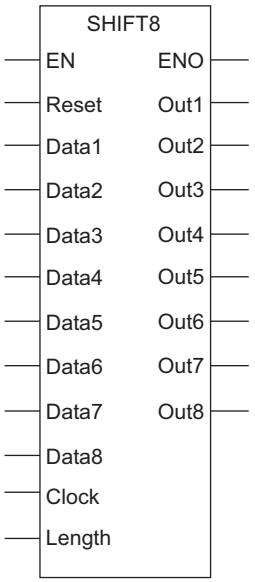
Note

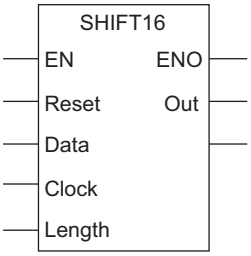
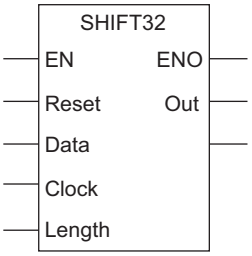
The SHIFT32 operation requires 2 RAM blocks. The SHIFT, SHIFT2, SHIFT4, SHIFT8 and SHIFT16 operations each require one RAM block.

All bit shift registers, the LIFO, and FIFO operations require RAM blocks. The maximum number of RAM blocks supported by the FM 352-5 module is 10.

Table 6- 56 Bit Shift Registers (SHIFT)

LAD representation	Para-meter	Data type	Addresses	Description
 (or SHIFT2, SHIFT4, SHIFT8, SHIFT16, SHIFT32)	Reset	BOOL	Input	A 1 at this input and a 1 at the EN resets all the stages of the shift register to 0.
	Data	BOOL, INT, DINT	Input	Data input for the shift register.
	Clock	BOOL	Input	Edge pulse input that moves the data through the shift register.
	Length	INT	Constant	Length of the shift register. Range: 2 to 4096 SHIFT 2 to 2048 SHIFT2 2 to 1024 SHIFT4 2 to 512 SHIFT8 2 to 256 SHIFT16 2 to 256 SHIFT32
	Out	BOOL, INT, DINT	Output	Output of the shift register

LAD representation	LAD representation	LAD representation
		

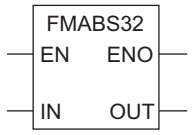
LAD representation	LAD representation
	

6.10.10 Absolute value (FMABS32 and FMABS16)

Description

The ABS operation writes the absolute value of the number supplied at the IN input to the OUT output. The absolute value of a number is the number without its sign.

Table 6- 57 Absolute value (FMABS32 and FMABS16)

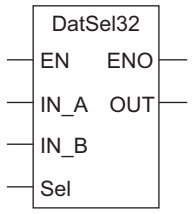
LAD representation	Parameter	Data type	Addresses	Description
	IN	INT, DINT	Input	Input value: Floating point
	OUT	INT, DINT	Output	Output value: Absolute value of the floating-point number
Note: No logic is allowed at the EN input.				

6.10.11 Data selector (DatSel32 and DatSel16)

Description

The DatSel operation provides the function of a 2-to-1 multiplexer by copying the value at the IN_A input to output OUT if input Sel is logic "0", or copying the value at the IN_B input to OUT if Sel is logic "1". An N-to-1 multiplexer can be created by cascading multiple DatSel operations.

Table 6- 58 Data selector (DatSel32 and DatSel16)

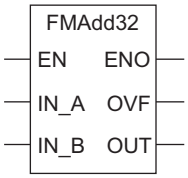
LAD representation	Parameter	Data type	Addresses	Description
	IN_A	INT, DINT	Input	Input value A
	IN_B	INT, DINT	Input	Input value B
	Sel	BOOL	Input	If 0, the value of IN_A is copied to the output. If 1, the value of IN_B is copied to the output.
	OUT	INT, DINT	Output	Output value: • IN_A if Sel = 0 • IN_B if Sel = 1
Note: No logic is allowed at the EN input.				

6.10.12 Add (FMAdd32 and FMAdd16)

Description

FMAdd adds the value at the IN_A input to the value at the IN_B input and writes the result to the OUT output. The OVF output is set to logic "1" if an overflow occurs; otherwise, it is logic "0".

Table 6- 59 Add (FMAdd32 and FMAdd16)

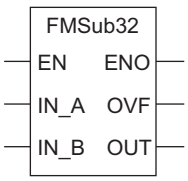
LAD representation	Parameter	Data type	Addresses	Description
	IN_A	INT, DINT	Input	Input value A
	IN_B	INT, DINT	Input	Input value B
	OVF	BOOL	Output	1 if add results in overflow
	OUT	INT, DINT	Output	Output value: = IN_A + IN_B
Note: No logic is allowed at the EN input.				

6.10.13 Subtract (FMSub32 and FMSub16)

Description

FMSub subtracts the value at the IN_B input from the value at the IN_A input and writes the result to the OUT output. The OVF output is set to logic "1" if an overflow occurs; otherwise, it is logic "0".

Table 6- 60 Subtract (FMSub32 and FMSub16)

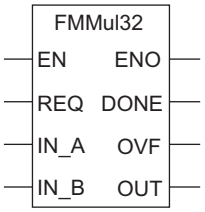
LAD representation	Parameter	Data type	Addresses	Description
	IN_A	INT, DINT	Input	Input value A
	IN_B	INT, DINT	Input	Input value B
	OVF	BOOL	Output	if subtract results in overflow
	OUT	INT, DINT	Output	Output value: = IN_A - IN_B
Note: No logic is allowed at the EN input.				

6.10.14 Multiply double (32-bit) integer (FMMul32)

Description

FMMul32 multiplies the integer value (32 bits) at the IN_A input by the integer value at the IN_B input and writes the result to the OUT output. The DONE output signals that the result is available. The valid range for inputs IN_A, IN_B and for the OUT output is -2,147,483,648 to +2,147,483,647. The OVF output is set to logic "1" if an overflow occurs; otherwise, it is logic "0".

Table 6- 61 Multiply double (32-bit) integer (FMMul32)

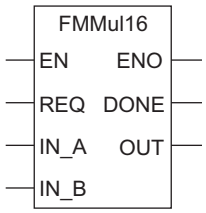
LAD representation	Parameters	Data type	Addresses	Description
	REQ	BOOL	Input	Enables the multiply operation on a 0 to 1 change. It must remain 1 until DONE = 1; otherwise, multiply terminates.
	IN_A	DINT	Input	Input value A
	IN_B	DINT	Input	Input value B
	DONE	BOOL	Output	1 = result is available
	OVF	BOOL	Output	1, if multiplication results in overflow
	OUT	DINT	Output	Output value: = IN_A × IN_B
Note: No logic is allowed at the EN input.				

6.10.15 Multiply 16-bit integer (FMMul16)

Description

FMMul16 multiplies the 16-bit integer value at the IN_A input by the integer value at the IN_B input and writes the double integer result to the OUT output. The DONE output signals that the result is available. The valid range for inputs IN_A and IN_B is -32768 to +32767.

Table 6- 62 Multiply 16-bit integer (FMMul16)

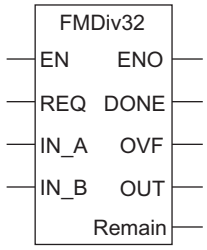
LAD representation	Parameter	Data type	Addresses	Description
	REQ	BOOL	Input	Enables the multiply operation on a 0 to 1 change. It must remain 1 until DONE = 1; otherwise, multiply terminates.
	IN_A	INT	Input	Input value A
	IN_B	INT	Input	Input value B
	DONE	BOOL	Output	1 = result is available
	OUT	DINT	Output	Output value: = IN_A × IN_B
Note: No logic is allowed at the EN input.				

6.10.16 Divide double (32-bit) integer (FMDiv32)

Description

FMDiv32 divides the 32-bit double integer value at the IN_A input by the double integer value at the IN_B input and writes the result to the OUT output and the remainder to the Remain output. The DONE output signals that the result is available. The valid range for inputs IN_A, IN_B and for the division remainder is -2,147,483,648 to +2,147,483,647. The OVF output is set to logic "1" if an overflow occurs; otherwise, it is logic "0". When OVF is "1", the OUT and Remain outputs are set to "0".

Table 6- 63 Divide double (32-bit) integer (FMDiv32)

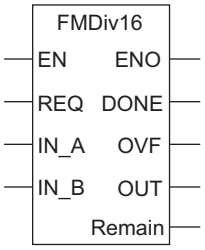
LAD representation	Parameters	Data type	Address	Description
	REQ	BOOL	Input	Enables the "Divide" operation on a 0 to 1 change. It must remain 1 until DONE = 1; otherwise, divide terminates.
	IN_A	DINT	Input	Dividend
	IN_B	DINT	Input	Divisor
	DONE	BOOL	Output	1 = result is available
	OVF	BOOL	Output	1, if divide results in overflow
	OUT	DINT	Output	Output value: = IN_A ÷ IN_B
	Remain	DINT	Output	Remainder of the division.
Note: No logic is allowed at the EN input.				

6.10.17 Divide 16-bit integer (FMDiv16)

Description

FMDiv16 divides the 16-bit integer value at the IN_A input by the integer value at the IN_B input and writes the result to the OUT output and the remainder to the Remain output. The DONE output signals that the result is available. The valid range for the IN_A input is -2,147,483,648 to +2,147,483,647. The valid range for input IN_B and outputs OUT and Remain is -32768 to +32767. The output OVF is set to logic "1" if an overflow occurs; otherwise, it is "0". When OVF is "1", the OUT and Remain outputs are set to "0".

Table 6- 64 Divide 16-bit integer (FMDiv16)

LAD representation	Parameter	Data type	Addresses	Description
	REQ	BOOL	Input	Enables the "Divide" operation on a 0 to 1 change. It must remain 1 until DONE = 1; otherwise, divide terminates.
	IN_A	DINT	Input	Dividend
	IN_B	INT	Input	Divisor
	DONE	BOOL	Output	1 = result is available
	OVF	BOOL	Output	1, if divide results in overflow
	OUT	INT	Output	Output value: = IN_A ÷ IN_B
	Remain	INT	Output	Remainder of the division.
Note: No logic is allowed at the EN input.				

6.10.18 Encode binary position (ENCODE)

Description

The ENCODE function converts the contents of IN to a binary number corresponding to the bit position of the leftmost set bit in IN, and returns the result as the function's value. If IN is either DW#16#00000001 or DW#16#00000000, a value of 0 is returned. If there is logic for EN, the output is latched. The output changes only when EN is active. With logic for EN, the ENCODE value is retentive, requiring storage and a phase clock.

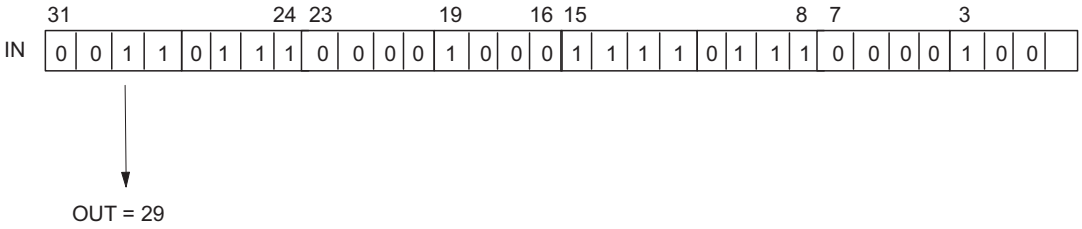


Figure 6-46 Example of ENCODE

Most significant bit set is in bit position 29

LAD representation	Parameter	Data type	Addresses	Description
ENCODE EN ENO IN OUT	EN	BOOL	Input, constant	Signal state 1 at the enable input activates the box.
	IN	DWORD	Input, constant	Variable to be encoded.
	ENO	BOOL	Output	Enable output follows the signal state of EN.
	OUT	INT	Output	Value output.

Error Information

This function does not detect any error states.

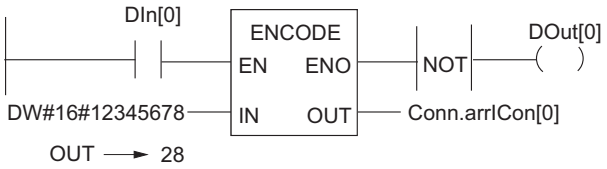


Figure 6-47 Example of the Encode Binary Position Function

If the signal state of input DIn[0] is 1 (activated), the ENCODE function is executed.

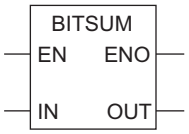
DOut[0] is "1" if the operation is executed.

6.10.19 Sum number of bits (BITSUM)

Description

The BITSUM function counts the number of bits that are set to a value of 1 in the IN input and returns this as the function's value. With logic for EN, the BITSUM value is retentive, requiring storage and a phase clock.

Table 6- 65 Sum Number of Bits Function

LAD representation	Parameter	Data type	Addresses	Description
	EN	BOOL	Input	Signal state 1 at the enable input activates the box.
	ENO	BOOL	Output	Enable output has the signal state 1 if the function is executed without error.
	IN	DWORD	Input	Variable in which bits are counted.
	OUT	INT	Output	Value output.

Error information

This function does not detect any error states.

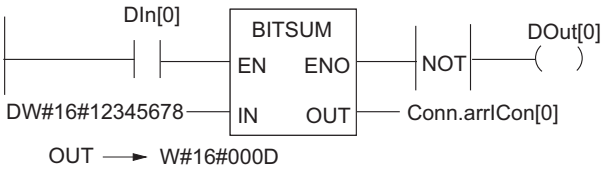


Figure 6-48 Example of the Sum Number of Bits Function

If the signal state of input DIn[0] is 1 (activated), the BITSUM function is executed. In this example, the value output in Conn.arrICon[0] is 13 ("D" in hexadecimal notation). This is the number of bits set to 1 in the hexadecimal double word input DW#16#12345678.

DOut[0] is "1" if the operation is executed.

6.10.20 BitPack_W and BitPack_DW

Description

The BitPack operation is available in two versions, as a 16-bit (FB93) and a 32-bit (FB92) version defined by the destination WORD or DWORD. When the FB is enabled the BOOL inputs (IN0-IN15 or IN0-IN31) are packed to form a WORD or a DWORD. IN0 is the LSB and IN15 or IN31 is the MSB of OUT. With logic for EN, the BitPack_W or BitPack_DW value is retentive, requiring storage and a phase clock.

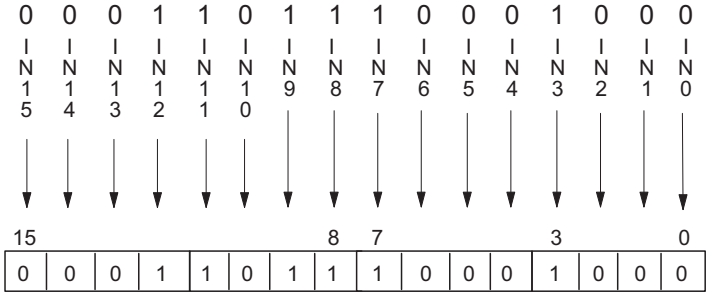


Figure 6-49 Example of BitPack_W and BitPack_DW

LAD representation	LAD representation	Param.	Data type	Address	Description
BitPack_W EN ENO IN0 OUT IN1 IN3 IN4 IN5 IN6 IN7 IN8 IN9 IN10 IN11 IN12 IN13 IN14 IN15	BitPack_DW EN ENO IN0 OUT IN1 IN3 IN4 IN5 • • • IN26 IN27 IN28 IN29 IN30 IN31	INn	BOOL	Input, constant	Inputs to be packed
		OUT	WORD, DWORD	Output	Output of function

6.10.21 BitCast_W and BitCast_DW

Description

The BitCast operation is available in two versions; a 16-bit (FB91) and a 32-bit (FB90) version defined by the input WORD or DWORD. When the FB is enabled the WORD or DWORD is converted into individual bits, BOOL outputs (OUT0–OUT15 or OUT0–OUT31). OUT0 is the LSB and OUT15 or OUT31 is the MSB of IN. With logic for EN, the BitCast_W or BitCast_DW value is retentive, requiring storage and a phase clock.

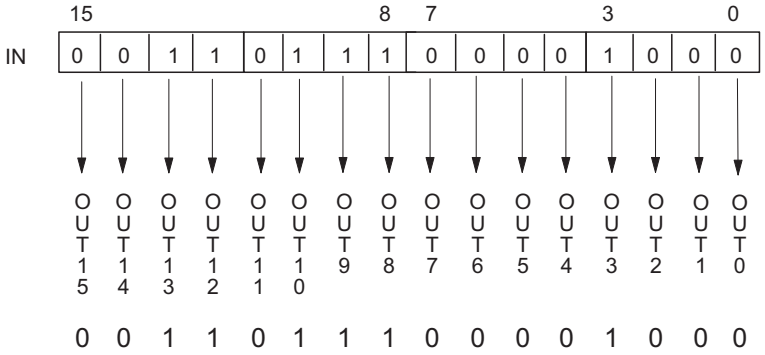


Figure 6-50 Example of BitCast _W and BitCast_DW

LAD representation	LAD representation	Param.	Data type	Address	Description
BitCast_W EN ENO IN OUT0 OUT1 OUT2 OUT3 OUT4 OUT5 OUT6 OUT7 OUT8 OUT9 OUT10 OUT11 OUT12 OUT13 OUT14 OUT15	BitCast_DW EN ENO IN OUT0 OUT1 OUT2 OUT3 OUT4 OUT5 • • • OUT26 OUT27 OUT28 OUT29 OUT30 OUT31	IN	WORD, DWORD	Input, constant	Input to be converted
		OUTn	BOOL	Output	Output of function

6.10.22 BitPick_W and BitPick_DW

Description

The BitPick operation is available in two versions; a 16-bit (FB87) and a 32-bit (FB86) version defined by the input WORD or DWORD.

When the FB is enabled the selected bit within the input WORD or DWORD is transferred to OUT. If SELECT is 0 then the LSB of the input WORD or DWORD is transferred to OUT. If SELECT is 15 (or 31) then the MSB of the input WORD (DWORD) is transferred to OUT. If there is logic for EN, the output is latched. The output changes only when EN is active. With logic for EN, the BitPick_W or BitPick_DW value is retentive, requiring storage and a phase clock.

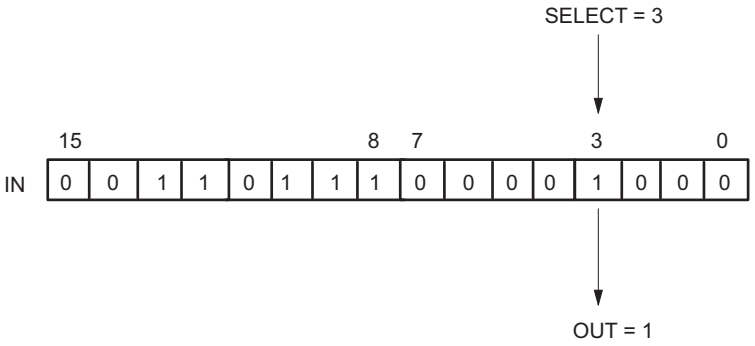


Figure 6-51 Example of BitPick_W and BitPick_DW

LAD representation	LAD representation	Param.	Data type	Addresses	Description
BitPick_W EN ENO IN OUT SELECT	BitPick_DW EN ENO IN OUT SELECT	IN	WORD, DWORD	Input, constant	Input from which the bit is selected
		SELECT	INT	Input, constant	Bit position to be selected within IN
		OUT	BOOL	Output	Output of function

6.10.23 Bitinsert

Description

The BitInsert operation is available in two versions; a 16-bit (FB95) and a 32-bit (FB94) version defined by the input WORD or DWORD.

When the FB is enabled the selected bit within the input WORD or DWORD is replaced, all other bits are transferred with no change. If SELECT is 0 then the LSB of the input WORD or DWORD is replaced by BIT. If SELECT is 15 (or 31) then the MSB of the input WORD (DWORD) is replaced by BIT. If there is logic for EN, the output is latched. The output changes only when EN is active. With logic for EN, the BitInsert value is retentive, requiring storage and a phase clock.

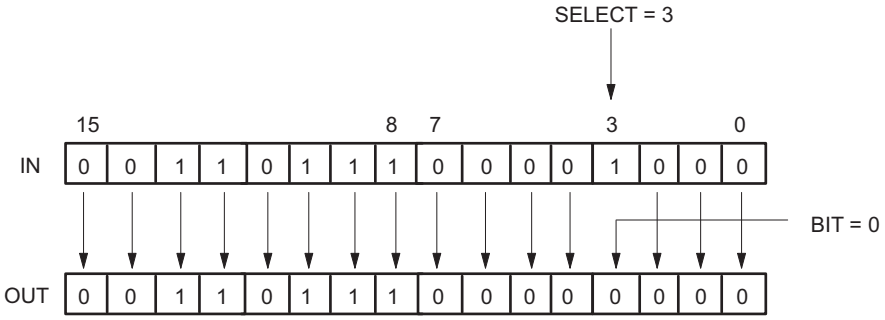


Figure 6-52 Example of BitInsert

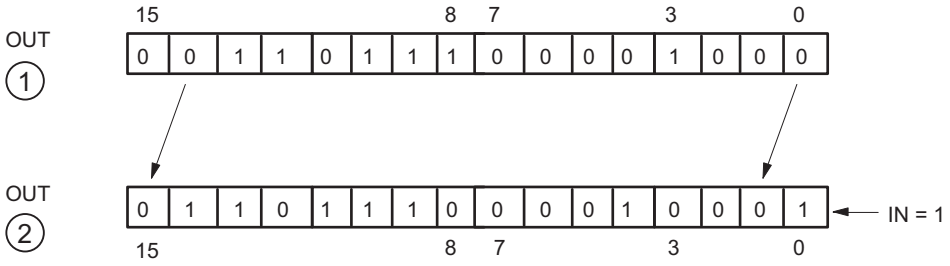
LAD representation	LAD representation	Param.	Data type	Addresses	Description
BitInsert16 EN ENO IN OUT SELECT Bit	BitInsert32 EN ENO IN OUT SELECT Bit	IN	INT, DINT	Input, constant	Input from which the bit is selected
		SELECT	INT	Input, constant	Bit position to be replaced within OUT
		Bit	BOOL	Input, constant	Bit to be inserted in OUT
		OUT	INT, DINT	Output	Output of function

6.10.24 BitShift_W and BitShift_DW

Description

The BitShift operation is available in two versions; a 16-bit (FB89) and a 32-bit (FB88) version defined by the output WORD or DWORD.

When the FB is enabled and SHIFT is active the input BOOL is left-shifted into the output WORD (OUT). The MSB of OUT is discarded. The LSB is replaced with BOOL IN. If EN and RESET are active simultaneously then OUT is reset to 0000 or 00000000. There is a shift in each scan cycle where EN and SHIFT are both active. This operation is retentive and requires one phase.



(1) OUT prior to execution

(2) OUT after execution

Figure 6-53 Example of BitShift_W and BitShift_DW

LAD representation	LAD representation	Param.	Data type	Addresses	Description
BitShift_W EN ENO Reset OUT IN SHIFT	BitShift_DW EN ENO Reset OUT IN SHIFT	IN	BOOL	Input, constant	Input bit to be shifted into LSB of OUT
		SHIFT	BOOL	Input, constant	If 1 and EN is active, shift is enabled
		Reset	BOOL	Input, constant	If 1 and EN is active, OUT is reset to 0000 (00000000)
		OUT	WORD	Output	Output of function

6.10.25 WordPack

Description

When the FB is enabled the input WORDs are concatenated into one DWORD. IN_A is the most significant word and IN_B is the least significant word. If there is logic for EN, the output is latched. The output changes only when EN is active. This operation requires one phase if there is logic for EN. With logic for EN, the WordPack value is retentive, requiring storage and a phase clock.

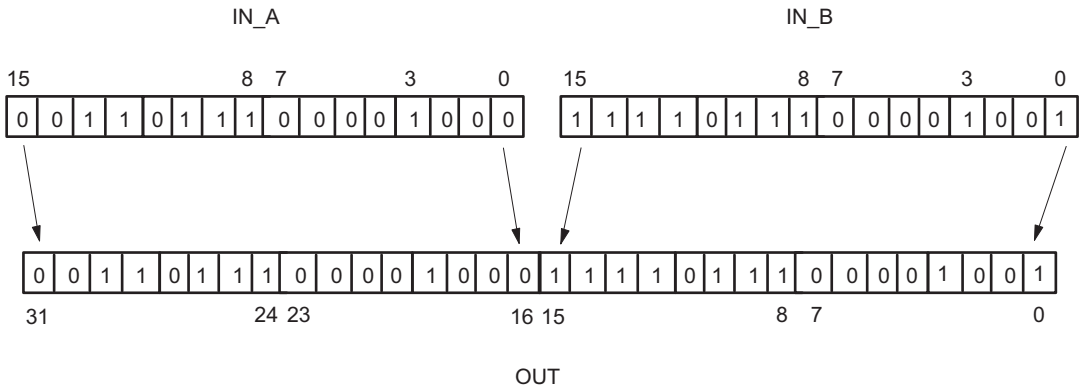


Figure 6-54 Example of WordPack

LAD representation	Parameter	Data type	Addresses	Description
WordPack EN ENO IN_A OUT IN_B	IN_A	WORD	Input, constant	Input with the most significant word
	IN_B	WORD	Input, constant	Input with the least significant word
	OUT	DWORD	Output	Output of function

6.10.26 WordCast

Description

When the FB is enabled, the DWORD input word is converted into two WORDs. OUT_A is the most significant word and OUT_B is the least significant word. If there is logic for EN, the output is latched. The output changes only when EN is active. This operation requires one phase if there is logic for EN. With logic for EN, the WordCast value is retentive, requiring storage and a phase clock.

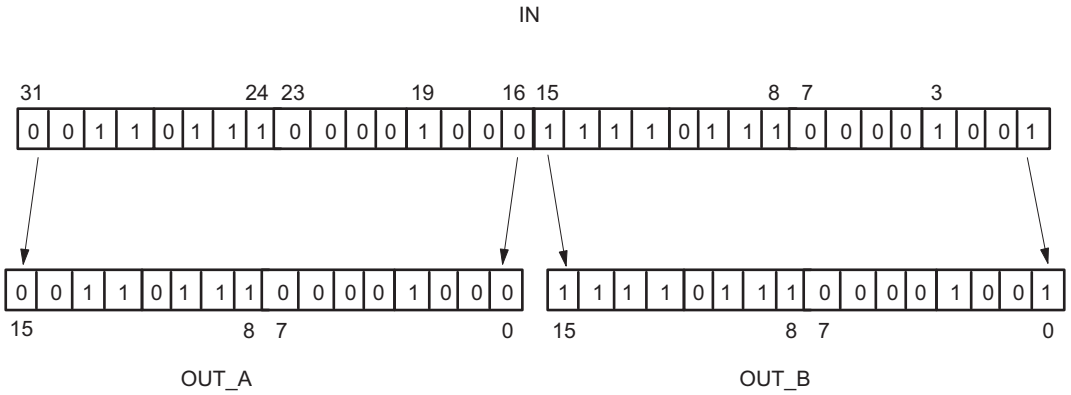


Figure 6-55 Example of Wordcast

LAD representation	Parameter	Data type	Addresses	Description
WordCast EN ENO IN OUT_A OUT_B	IN	DWORD	Input, constant	Input with the most significant word
	OUT_A	WORD	Output	Output of the function, most significant word of IN
	OUT_B	WORD	Output	Output of the function, least significant word of IN

6.10.27 Period measurement (PERIOD16, PERIOD32)

Description

This PERIOD operation is available in two versions: As a 16-bit bit version (FB81) and a 32-bit version (FB80) defined by the output WORD or DWORD. While EN is active, OUT is updated on every rising edge at IN. VALID is true when OUT has valid data. VALID is false if OUT cannot represent the count (rollover occurs) and it is false until the initial period has not been measured. OUT is useful for measuring low frequencies where FREQ would require a lengthy period to determine the frequency. This operation requires one clock pulse. If the module changes to STOP or if EN is inactive, the OUT operation is reset. Two rising edges must be preset at IN before OUT can be represented.

PERIOD16 is used to measure periods of 2 to 65535 (2^{16}-1) microseconds. Periods greater than 32767 (2^{15}-1) microseconds will appear negative. VALID will be 0 if the period exceeds 65535 microseconds.

PERIOD32 is used to measure periods of 2 to 4,294,967,295 (2^{32}-1) microseconds. Periods greater than 2.147.483.647 (2^{31}-1) microseconds will appear negative. VALID will be 0 if the period exceeds 4,294,967,295 microseconds.

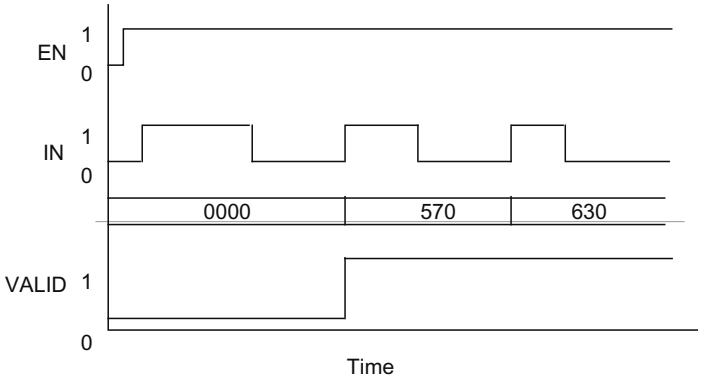


Figure 6-56 Example of PERIOD16, PERIOD32

LAD representation	LAD representation	Para-meter	Data type	Addresses	Description
PERIOD16 EN ENO IN VALID OUT	PERIOD32 EN ENO IN VALID OUT	IN	BOOL	Input	Input signal whose frequency is measured
		VALID	BOOL	Output	Indicates PERIOD is valid
		OUT	INT, DINT	Output	Output of function

6.10.28 Frequency measurement (FREQ16, FREQ32)

Description

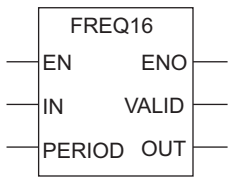
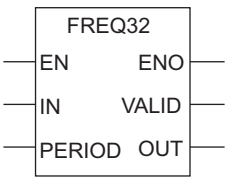
This FREQ operation is available in two versions: As a 16-bit bit version (FB83) and a 32-bit version (FB82) defined by the output WORD or DWORD.

While EN is active FREQ counts the number of rising edges at IN during the number of microseconds defined in PERIOD. OUT is updated at an interval of PERIOD microseconds. VALID is true when OUT has valid data. VALID is false if OUT cannot represent the count (rollover occurs) and will also be false if the initial period has not elapsed. This operation requires one clock pulse. If the module changes to STOP or if EN is inactive, the FREQ operation is reset. The number of microseconds defined in period must elapse before the OUT can be represented.

FREQ16 is used to measure frequencies of 0 to 65535 (2^{16}-1). Frequences greater than 32767 (2^{15}-1) microseconds will appear negative. VALID is 0 if the frequency exceeds 65535.

FREQ32 is used to measure frequencies of 0 to 4,294,967,295 (2^{32}-1). Frequencies greater than 2,147,483,647 (2^{31}-1) will appear negative. VALID is 0 if the frequency exceeds 4,294,967,295.

The FREQ operation outputs OUT in Hz if the period is set to 1000000 (1 second). If period is set to 10,000,000 (10 seconds) then OUT is output in units of 0.1 Hz (in other words, if OUT = 600, then the frequency is 60.0 Hz). The output value is retentive and uses one clock phase.

LAD representation	LAD representation	Para-meters	Data type	Address	Description
		IN	BOOL	Input	Input signal whose frequency is measured
		PERIOD	DINT	Input, constant	Time period for frequency measurement (in microseconds)
		VALID	BOOL	Output	Indicates that frequency data is valid
		OUT	INT, DINT	Output	Output of function

6.10.29 Delete first value (FIFO16, FIFO32)

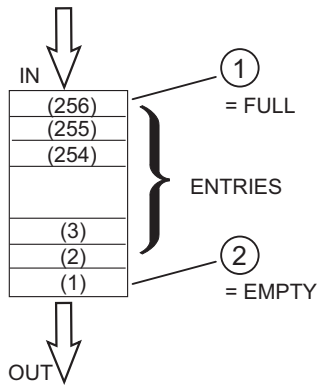
Description

The FIFO operation is available in two versions:a 16-bit version (FB97) and a 32-bit version (FB96) defined by the data width. The FIFO shift register stores entries that are written into the FIFO box and represents the stored data upon request. When WRITE and EN are active, the data present at IN is written into the FIFO box. The oldest entry in the FIFO box is available at OUT until it is discarded by activating READ_NEXT. The next to oldest entry then becomes the oldest entry. If the FIFO box is full (256 entries) then FULL becomes active. Any write operation that occurs while FULL is active will be discarded. EMPTY signals that the FIFO box is empty (0 entries). OUT is indeterminate while EMPTY is active. ENTRIES indicates the number of entries contained in the FIFO box. If EN and RESET are active simultaneously then the FIFO box is cleared. All entries are reset to 0 and EMPTY is activated. The output value is retentive and uses one clock phase.

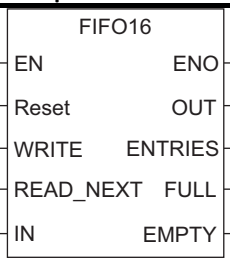
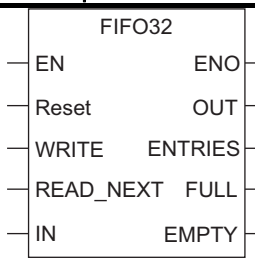
Note

The FIFO16 instruction consumes 1 RAM block. The FIFO32 operation requires 2 RAM blocks.

All bit shift registers, the LIFO, and FIFO operations require RAM blocks. The maximum number of RAM blocks supported by the FM 352-5 module is 10.

	Scan cycle n	Scan cycle n+1	Scan cycle n+2
 1) Entry 2) No entry	Output conditions Entry 1 = 5 Entry 2 = 100 Entry 3 = 125 Entry 4 = -1	Entry 1 = 1 Entry 2 = 100 Entry 3= 125 Entry 4 = -1 Entry 5 = 654	Entry 1 = 100 Entry 2 = 125 Entry 3= -1 Entry 4 = 654
	ENTRIES = 4 FULL = 0 EMPTY = 0 OUT = 5 IN = 654 WRITE = 1 READ_NEXT = 0	ENTRIES = 5 FULL = 0 EMPTY = 0 OUT = 5 IN = 0 WRITE = 0 READ_NEXT = 1	ENTRIES = 4 FULL = 0 EMPTY = 0 OUT = 100 IN = 0 WRITE = 0 READ_NEXT = 0

LAD representation	LAD representation	Parameters	Data type	Addresses	Description
		Reset	BOOL	Input, constant	If 1 and EN is active, the FIFO entries are reset to 0000 (00000000)

LAD representation	LAD representation	Parameters	Data type	Addresses	Description
		WRITE	BOOL	Input, constant	If 1, FULL = 0 and EN is active, IN is written into the FIFO
		READ_NEXT	BOOL	Input, constant	If 1, EMPTY = 0 and EN is active, next entry is placed in OUT
		IN	INT, DINT	Input, constant	Data input to FIFO
		OUT	INT, DINT	Output	Data output from FIFO
		ENTRIES	INT	Output	Indicates number of entries stored in the FIFO
		FULL	BOOL	Output	1 indicates that FIFO is full and cannot be written to (256 entries)
		EMPTY	BOOL	Output	1 indicates that FIFO is empty (0 entries)

6.10.30 Delete last value (LIFO16, LIFO32)

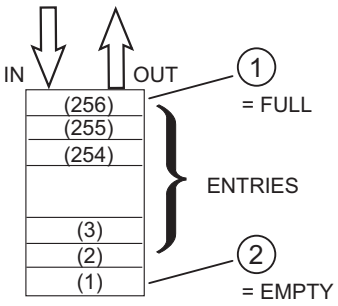
Description

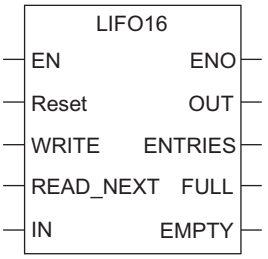
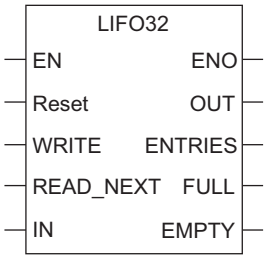
The LIFO operation is available in two versions:as a 16-bit version (FB99) and as a 32-bit version (FB98) defined by the data width. The LIFO shift register stores entries that are written into the LIFO box and represents the stored data upon request. When the WRITE and EN inputs are active, the data present at IN is written into the LIFO box. The newest entry in the LIFO box is represented at OUT until it is discarded by activating READ_NEXT. The next to newest entry then becomes the newest entry. If the LIFO box is full (256 entries), then FULL becomes active. Any write operation that occurs while FULL is active will be discarded. EMPTY signals that the LIFO box is empty (0 entries). OUT is indeterminate while EMPTY is active. ENTRIES indicates the number of entries contained in the LIFO box. If EN and RESET are active simultaneously then the LIFO box is cleared. All entries are reset to 0 and EMPTY is activated. The output value is retentive and uses one clock phase.

Note

The LIFO16 operation requires 1 RAM block. The LIFO32 operation requires 2 RAM blocks.

All bit shift registers, the LIFO, and FIFO operations require RAM blocks. The maximum number of RAM blocks supported by the FM 352-5 module is 10.

	Scan cycle n	Scan cycle n+1	Scan cycle n+2
 1) Entry 2) No entry	Output conditions Entry 1 = 5 Entry 2 = 100 Entry 3 = 125 Entry 4 = -1	Entry 1 = 5 Entry 2 = 100 Entry 3 = 125 Entry 4 = -1 Entry 5 = 654	Entry 1 = 5 Entry 2 = 100 Entry 3 = 125 Entry 4 = -1
	ENTRIES = 4 FULL = 0 EMPTY = 0 OUT = -1 IN = 654 WRITE = 1 READ_NEXT = 0	ENTRIES = 5 FULL = 0 EMPTY = 0 OUT = 654 IN = 0 WRITE = 0 READ_NEXT = 1	ENTRIES = 4 FULL = 0 EMPTY = 0 OUT = -1 IN = 654 WRITE = 0 READ_NEXT = 0

LAD representation	LAD representation	Parameters	Data type	Addresses	Description
		Reset	BOOL	Input, constant	If 1 and EN is active, the LIFO entries are reset to 0000 (00000000)
		WRITE	BOOL	Input, constant	If 1, FULL = 0 and EN is active, IN is written to the LIFO
		READ_NEXT	BOOL	Input, constant	If 1, EMPTY = 0 and EN is active, next entry is placed in OUT
		IN	INT, DINT	Input, constant	Data input to LIFO
		OUT	INT, DINT	Output	Data output from LIFO
		ENTRIES	INT	Output	Indicates number of entries stored in the LIFO
		FULL	BOOL	Output	Indicates that the LIFO is full and cannot be written to (256 entries)
		EMPTY	BOOL	Output	Indicates that the LIFO is empty (0 entries)

Encoder signals and their evaluation

7

7.1 Types of encoders

Encoder types

The FM 352-5 module allows you to connect one of the following encoder types:

- Differential incremental encoder/RS-422 (16-bit or 32-bit counter)
- 24 V single-ended incremental encoder (16-bit or 32-bit counter)
- SSI absolute encoder (13-bit or 25-bit resolution)

Any inputs that are not required by the encoder type selected are available as general purpose inputs.

Encoder Interface Signals

The following table lists the signals that are used by each encoder and the corresponding position for each signal on the terminal strip.

Table 7- 1 Encoder signals

Encoders	Signal	Terminal number
RS-422 differential incremental encoder	Signal A Signal /A (inverse) Signal B Signal /B (inverse) Signal N Signal /N (inverse)	26 27 28 29 30 31
24 V differential incremental encoder (HTL)	Signal A Signal B Signal N	37 38 39
SSI encoder (master mode)	SSI D (data) SSI /D (data inverse) SSI CK (output shift clock pulse) SSI /CK (output shift clock pulse inverse)	26 27 32 33
SSI Encoder (Listen mode)	SSI D (data) SSI /D (data inverse) SSI CK (input shift clock pulse) SSI /CK (input shift clock pulse inverse)	26 27 28 29

Encoder Control

The following table lists the control signals, set with hardware and software, that determine how the incremental encoders operate.

- Select these operating controls in the "Parameters" tab of the FM 352-5 hardware configuration "Properties" dialog (refer to section "Assigning properties and parameters (Page 52)").
- You assign the software controls in your application FB by selecting the appropriate element from the declaration table (see table below) to use in your program.

Table 7- 2 Operating Controls for Incremental Encoders

Encoder parameters	Range of values	Default
Encoder signal evaluation	Pulse & direction, x1, x2, x4	Pulse and direction
Source reset	None, HW, SW, HW and SW, HW or SW	None
Source reset value	Constant 0, Min/Max value, Load value	Constant 0
Reset signal type	Edge, level	Edge
Source load value	Constant, module application	Constant
Source stop	None, HW, SW, HW and SW, HW or SW	None
Load value	*Input field**	0
Count range minimum	*Input field**	0
Count range maximum	*Input field**	32767 (16 bits) or 2147483647 (32 bits)
Main count direction	Up count, down count	Up count
Hardware source stop	Inputs 0 to 14	Input 8
Hardware source reset	Inputs 0 to 14	Input 11
* Enter a value within the range of -32768 to 32767 (for a 16-bit counter) or -2147483648 to 2147483647 for a 32-bit counter.		

The following table shows the encoder structure as it appears in the declaration table of the application FB. This provides the status information and software controls of the encoder.

Table 7- 3 Example Declaration Table for the Application FB, Encoder Structure

Address	Declaration	Name	Type	Comment
Static section: This definition is position-specific. The encoder is a structure that has a fixed number of elements. The names cannot be changed, but the size of Cur_Val and Load_Val must be set to INT or DINT according to which size of encoder is configured.				
38.0	stat	Encoders	STRUCT	Encoder structure. Do not change.
+0.0	stat	Direction	BOOL	Status: Direction 0 = up count, 1 = down count
+0.1	stat	Home	BOOL	Status: 1 = encoder is at home position.
+0.2	stat	Homed	BOOL	Status: 1 = Home was adopted since power up
+0.3	stat	Overflow	BOOL	Status: 1= overflow (displayed for the duration of one cycle)
+0.4	stat	Underflow	BOOL	Status: 1= Underflow (displayed for 1 cycle)
+0.5	stat	SSIFrame	BOOL	Status: SSI frame error or power loss
+0.6	stat	SSIDataReady	BOOL	Status: 0 = SSI encoder has not yet shifted valid data, 1 = data available
+0.7	stat	Open_Wire	BOOL	Status: 1 = Encoder has open wire
+1.0	stat	Hold	BOOL	Hold software input for incremental encoder
+1.1	stat	Reset	BOOL	Reset software input for incremental encoder
+1.2	stat	Load	BOOL	Load software input for incremental encoder
+2.0	stat	Cur_Val	DINT	Current value for incremental encoder: DINT for 32-bit encoder, INT for 16-bit encoder
+6.0	stat	Load_Val	DINT	Load value for the encoder: DINT or INT
=10.0	stat		END_STRUCT	

7.2 Counting modes of the incremental encoder

Counting Modes

The FM 352-5 module supports a 16-bit or a 32-bit incremental encoder counter. The counter can function in one of three modes:

- Continuous
- Single
- Periodic

These modes are described in this section.

Selecting Edge or Level Reset

The reset function for each of the three counting modes can be set for edge or level. This functions as follows:

- Edge: Reset is dominant. If Hold and Reset are activated simultaneously, the count is reset, and then held.
- Level: Hold is dominant. If Hold and Reset are activated simultaneously, no reset occurs. If Hold is removed first, the count is reset. If both Hold and Reset are removed simultaneously, the count is reset. If Reset is removed before Hold, no reset will occur.

Encoder Status Bits

As described in this section, the module returns status bits to indicate the following conditions:

- Counting direction: Indicates the direction of the last count.
- Overflow: Indicates that the counter has reached the maximum value and exceeded it (incremented by 1). The overflow bit is on for one scan cycle.
- Underflow: Indicates that the counter has reached the minimum value and exceeded it (decremented by 1). The underflow bit is on for one scan cycle.
- Homed: Indicates that the encoder has reached its home position since the last power up, and that position data is accurate (the encoder is synchronized).
- Home: Indicates that the encoder is currently at the home position, which is defined as a reset of the counter.

The encoder status bits, except for "Homed", are reset when the module changes to STOP.

Counter Behavior Common to the Three Counting Modes

If the counter is loaded with a value outside the count range, then the counter counts in the requested direction, and rolls over at the upper limit. (This rollover is not reported in the overflow or underflow status bits.) Once the counter value is within the specified range, it remains within the range until a Load or Reset loads it outside the range.

The counting process can be started or stopped using the software Hold or Reset signals, but the counter is neither held nor reset when the module goes to STOP mode. Software controls (Reset, Hold, and Load) are cleared by module STOP. The counter continues to count based on hardware inputs. The counter is not affected when the PLC changes to STOP. The current count value can be loaded using the load signal.

Continuous Counting Mode

In the continuous counting mode, the count ranges are variable and can be changed.

- Count range (16-bit counter): -32768 to 32767
- Count range (32-bit counter): -2,147,483,648 to 2,147,483,647

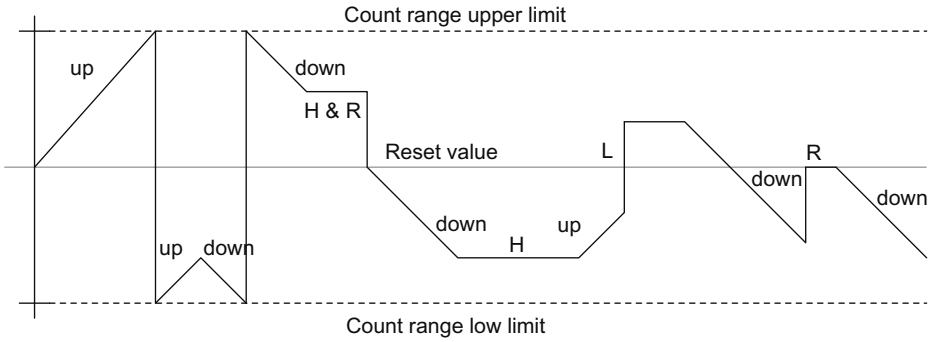
At power-up, the counter has a start value of 0, until either the hardware configuration or the software program give it a different starting value. You must initialize the counter to a known value with a reset or load before you begin counting. You can program the reset signal to load the counter with 0, the minimum value, or the load value.

The "main count direction" parameter has no effect on this counter mode.

When counting up, the module increments to the maximum value, then rolls over to the minimum value and continues counting. (This rollover is reported in the overflow status bit.)

When counting down, the module decrements to the minimum value, then rolls over to the maximum value and continues counting. (This rollover is reported in the underflow status bit.)

The figure below illustrates how continuous counting functions.



up:	Up count
down:	Down count
H:	Hold is active
R:	Reset is active
L:	Loading is active

Figure 7-1 Continuous Counting Mode

Single Counting

In the single counting mode, you can specify the count range as listed below, depending on whether you select the 16-bit counter or the 32-bit counter:

- Counting range (16-bit counter): -32768 to 32767
- Counting range (32-bit counter): -2,147,483,648 to 2,147,483,647

You must initialize the counter to a known value with a reset or load before you begin counting. You can program the reset signal to load the counter with 0, the minimum or maximum value, or the load value.

When the "main count direction" is set to Count Up, the counter behaves in the following ways:

- It increments to the maximum value, then rolls over to the minimum value and holds this value until reset or loaded. (This rollover is reported in the overflow status bit.)
- It decrements to the lower limit of the counter, rolls over to the upper limit, and continues counting. (This rollover is not reported in the overflow or underflow status bits.)

When the "main count direction" is set to Count Down, the counter behaves in the following ways:

- It decrements to the minimum value, then rolls over to the maximum value and holds this value until reset or loaded. (This rollover is reported in the underflow status bit.)
- It increments to the upper limit of the counter, rolls over to the lower limit, and continues counting. (This rollover is not reported in the overflow or underflow status bits.)

The figure below illustrates how single counting functions.

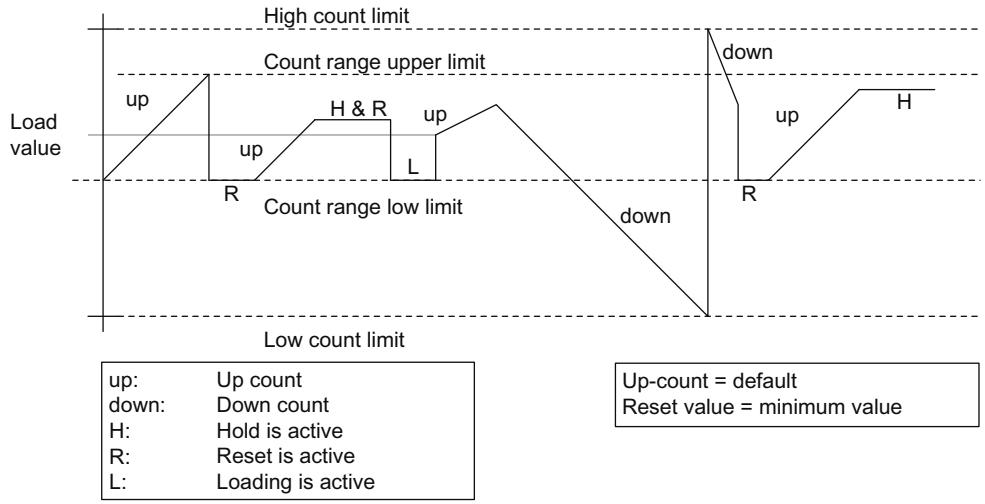


Figure 7-2 Single counting

Periodic counting

In the periodic counting mode, you can specify the count range.

- Counting range (16-bit counter): -32768 to 32767
- Counting range (32-bit counter): -2,147,483,648 to 2,147,483,647

You must initialize the counter to a known value with a reset or load before you begin counting. You can program the reset signal to load the counter with 0, the minimum or maximum value, or the load value.

When the "main count direction" is set to Count Up, the counter behaves in the following ways:

- It increments to the maximum value, then rolls over to the minimum value and continues counting. (This rollover is reported in the overflow status bit.)
- It decrements to the lower limit of the counter, rolls over to the upper limit, and continues counting. (This rollover is not reported in the overflow or underflow status bits.)

When the "main count direction" is set to Count Down, the counter behaves in one of the following ways:

- It decrements to the minimum value, then rolls over to the maximum value and continues counting. (This rollover is reported in the underflow status bit.)
- It increments to the upper limit of the counter, rolls over to the lower limit, and continues counting. (This rollover is not reported in the overflow or underflow status bits.)

The figure below illustrates how periodic counting functions.

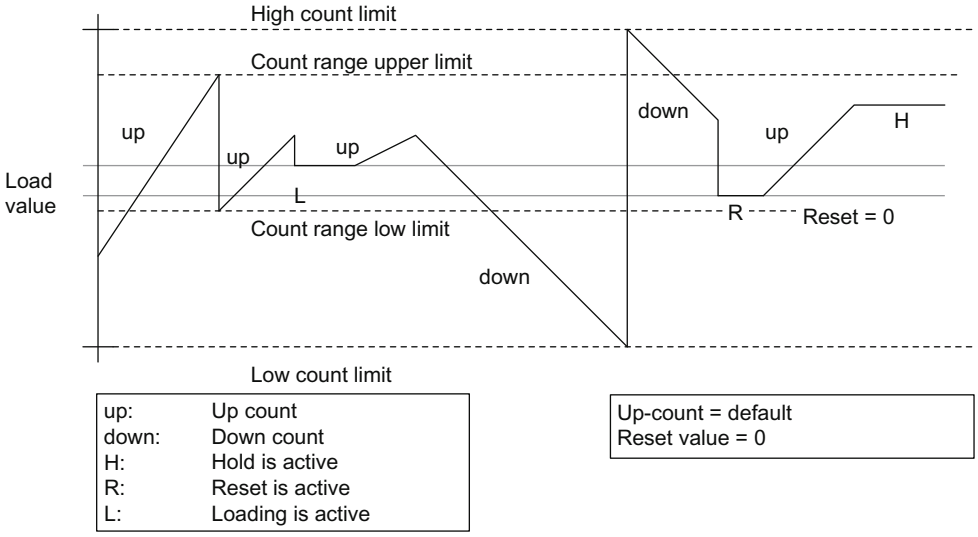


Figure 7-3 Periodic counting

7.3 RS-422 differential encoder signals

RS-422 differential encoder signals

The differential encoder supplies the differential signals A, /A, B, /B and N, /N to the module. The signals /A, /B, and /N are the inverted signals of A, B, and N. The signals A and B are phase-shifted by 90°. Encoders with these six signals are known as differential or symmetric encoders.

Signals A and B are used for counting. Signal N is used for setting the counter to the reset value if the parameters are set accordingly.

The following figure shows the time sequence of these signals.

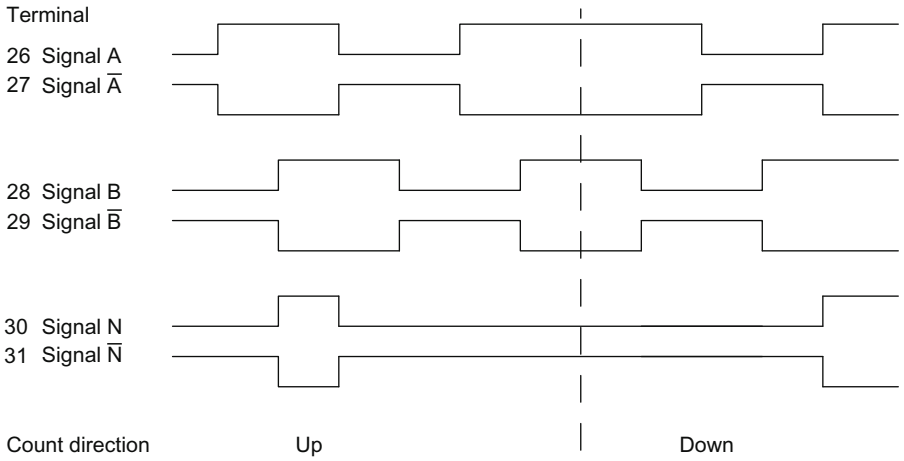


Figure 7-4 RS-422 differential encoder signals

The module recognizes the count direction from the phase relationship of signals A and B.

Note

When a quadrature encoder is selected, the broken-wire diagnostic function checks the signal status of A, /A (inverse), B, /B (inverse) and N, /N (inverse). If one of the inputs is not used, you must strap it in order to provide a non-zero differential voltage. Otherwise, the unused input will cause a broken-wire indication. To avoid a broken-wire diagnostic message, tie the unused input signals X to +5V and /X (inverse) to GND.

7.4 24 V single-ended encoder signals (HTL)

24 V single-ended encoder signals (HTL)

The 24 V single-ended incremental encoder supplies the signals A, B, and N in the same phase relationship as the signals A, B, and N in the case of the differential incremental encoder. The signals A and B are phase-shifted by 90°.

Encoders that do not supply inverse signals are known as single-sided or asymmetric encoders.

There are also encoders with a direction level. The following figure shows the sequence over time of the 24 V pulse encoder signals with direction level and the resulting count pulses.

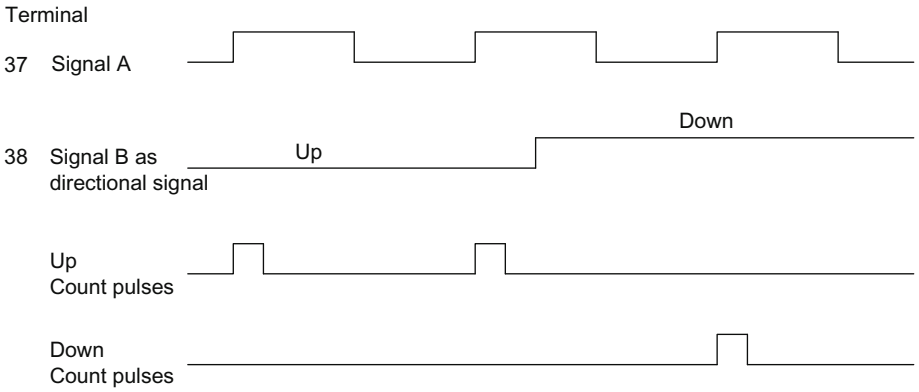


Figure 7-5 Signals of a 24 V Pulse Encoder with Direction Level

7.5 Pulse Evaluation

Introduction

The counters of the FM 352-5 count the edges of the signals. Normally, the edge at A is evaluated for a single evaluation (x1). To achieve a higher resolution, you can assign the parameter for the encoder signal evaluation to use double or quadruple (x2 or x4) evaluation of the signals. Use the "Parameters" tab in the FM 352-5 Configuration dialog to select the type of encoder signal evaluation.

The A and B signals must be displaced by 90° to select single, double, or quadruple evaluation.

Pulse and direction

When you select Pulse & Direction for the encoder signal evaluation type, the module counts on the rising edge of each signal A pulse. If signal B is 0 (low), the counter is **incremented**. If signal B is 1 (high), the counter **decrements**.

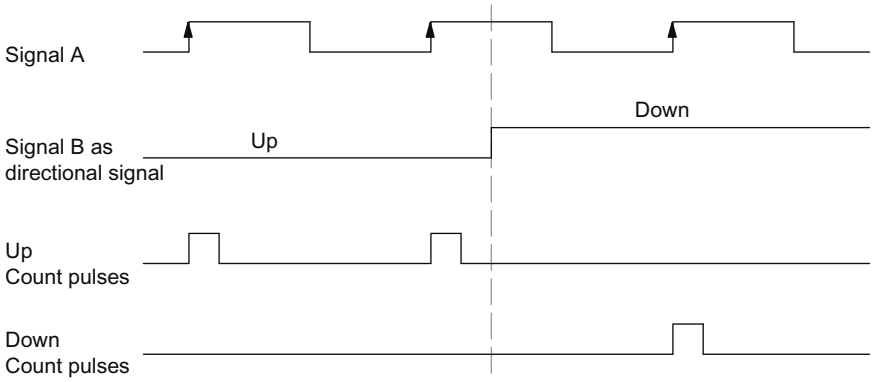


Figure 7-6 Pulse & Direction Counting

Single Evaluation

Single evaluation (x1) means that only one edge of A is evaluated.

- The counter **increments** on a rising edge of A when B is low.
- The counter **decrements** on a falling edge of A when B is low.

The figure below illustrates single evaluation of the signals.

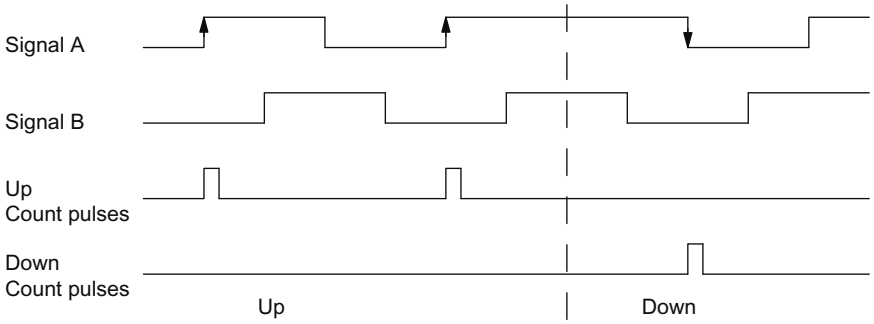


Figure 7-7 Single Evaluation

Double Evaluation

Double evaluation (x2) means that the rising and falling edges of signal A are evaluated. The level of signal B determines the direction of counting.

- The counter **increments** on the rising edge of A when B is low, and on the falling edge of A when B is high.
- The counter **decrements** on the rising edge of A when B is high, and on the falling edge of A when B is low.

The figure below illustrates double evaluation of the signals.

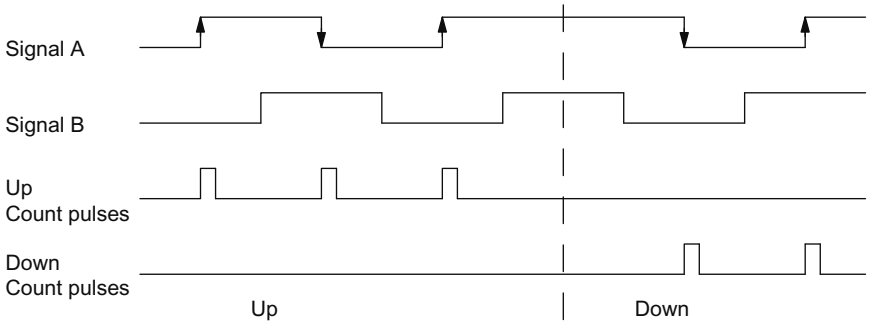


Figure 7-8 Double Evaluation

Quadruple Evaluation

Quadruple evaluation (x4) means that the rising and falling edges of A and B are evaluated. The levels of signals A and B determine the direction of counting.

- The counter **increments**: on the rising edge of A when B is low, on the falling edge of A when B is high, on the rising edge of B when A is high, and on the falling edge of B when A is low.
- The counter **decrements**: on the falling edge of A when B is low, on the rising edge of A when B is high, on the falling edge of B when A is high, and on the rising edge of B when A is low.

The figure below illustrates quadruple evaluation of the signals.

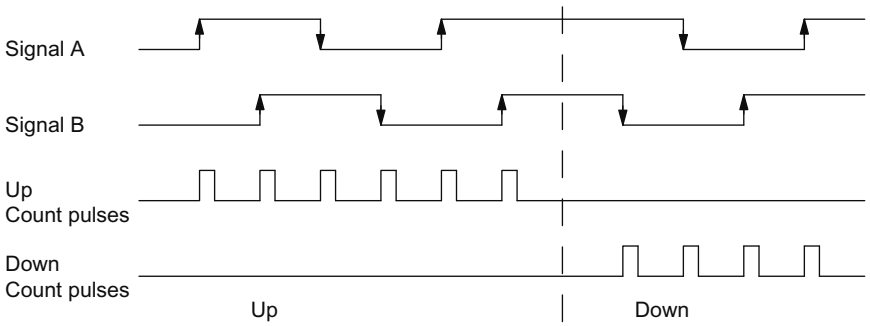


Figure 7-9 Quadruple Evaluation

7.6 SSI absolute encoders

SSI encoder overview

Absolute encoders with synchronous-serial interface (SSI) assign a fixed numeric value to each position. This value is permanently available and can be read out serially. The FM 352-5 module processes Gray code only.

Multi-turn SSI encoders have a frame length of 25 bits. The FM 352-5 module can process 24 bits.

Single-turn SSI encoders have a frame length of 13 bits (12 bits of data).

Delay time

Use the "Parameters" configuration tab to set the delay time for the SSI encoder to 16, 32, 48, or 64 µs.

For an SSI Master, you must select a delay time equal to or greater than the encoder's specified minimum time. If you do not know the specification for your encoder, select 64 µs. For an SSI Listen application, you must select a delay time equal to or less than the master’s delay time.

Shift Register Frame Length

You can select a shift register frame length of 13 bits or 25 bits in the "Parameters" tab, depending on the frame length of your SSI encoder.

Clock rate

You can select a clock rate of 125 kHz, 250 kHz, 500 kHz, or 1 MHz in the Parameters tab dialog, based on the capabilities of the encoder, the update time required, and the length of the cable. The maximum clock rate you can select is limited by the length of shielded encoder cable you use.

- At 125 kHz, the maximum cable length is 320 meters.
- At 250 kHz, the maximum cable length is 160 meters.
- At 500 kHz, the maximum cable length is 60 meters.
- At 1 MHz, the maximum cable length is 20 meters.

For an SSI slave (Listen mode), clock rate selection is not possible.

Data shift direction

You can select the direction of data to shift left or right in the "Parameters" tab.

Normalization Data Shift Length

You can specify the number of bit positions to be shifted within the range of 0 to 12 bits in the "Parameters" tab. Normalization allows the SSI encoder data to be scaled to more convenient units used in the module program.

SSI mode

You can select Master or Listen for the SSI mode. Only one module can be a master. The Listen mode allows other modules to connect to the same encoder for synchronized control.

Note

In SSI mode, broken wire diagnostics checks the signal status of D or /D (inverse) only.

Diagnostics and troubleshooting

8

8.1 Reading the status LEDs

Status LEDs

The status LEDs on the front of the module indicate the following conditions, as described in the table below:

Table 8- 1 Status LED definitions

LED label	LED	Color	Description
SF		Red	Indicates a fault condition in the module.
MCF		Red	Indicates an error condition in the SIMATIC Micro Memory Card of the module.
DC5V		Green	Indicates the power status of the module.
IOF		Red	Indicates an I/O fault condition: Output overload, missing 2L or 3L, broken wire, SSI fault.
RUN		Green	Indicates the module is in RUN mode.
STOP		Yellow	Indicates that the module is in STOP mode.
I0 to I11		Green	Indicates which inputs are on.
Q0 to Q7		Green	Indicates which outputs are on.
5VF		Red	Indicates an overload in the 5 V power supply output.
24VF		Red	Indicates an overload in the 24 V power supply output.

How the LEDs operate

The status LEDs behave as described in the following table depending on the operation being executed:

Table 8- 2 Behavior of status LEDs

Active LEDs	LED	Behavior	Operation
All LEDs		On for 1 second	LED test at startup.
RUN STOP		Fast flashing (2 Hz) On	Downloading to the module from the SIMATIC Micro Memory Card or the PC.
RUN STOP		Slow flashing (0.5 Hz) Off	When module is in Test/RUN mode.
RUN STOP		Slow flashing (0.5 Hz) On	When module is in Test/STOP mode.

8.2 Diagnostic messages

Responding to Diagnostic Interrupts

If you want your program to respond to an internal or external module fault, you can set a diagnostics interrupt that stops the cyclic program of the CPU and calls the diagnostics interrupt OB (OB82).

Events that can Initiate Diagnostics Interrupts

The following events or conditions trigger diagnostics interrupts:

- Module parameter assignment missing
- Error in module parameter assignment
- Watchdog time-out
- Processor failure
- Flash memory error
- RAM test error during startup

You can set the following conditions to trigger diagnostics interrupts:

- Output overload
- External auxiliary voltage missing (1L)
- Missing input/output supply voltage (2L)
- Missing encoder supply voltage (3L)
- SSI frame error
- Overloaded encoder supply (24 V or 5 V)
- Wire break (RS-422/symmetrical incremental encoders only)
- SIMATIC Micro Memory Card error
- Consistency error

Enabling the Diagnostics Interrupts

The Hardware Configuration dialog provides a "Parameters" tab where you can select which diagnostics you want to enable. You also select whether the module will trigger diagnostics interrupts and/or hardware interrupts.

Responses to a Diagnostics Interrupt

If an event occurs that can trigger a diagnostic interrupt, the following happens:

- The diagnostic information is stored in data records 0, 1, and 128.
- The SF error LED lights up.
- The diagnostics interrupt OB is called (OB82).
- The diagnostic data record 0 is entered in the start information of OB82.

If OB82 has not been programmed, the CPU changes to STOP.

Reading the Data Record from the Module

The diagnostic data record 0 is automatically transferred to the start information when the diagnostics OB is called. These four bytes are stored in bytes 8 to 11 of OB82. Data record 0 reports module-level diagnostics.

Assignments of diagnostic data record 0

The following table shows the assignments of diagnostic data record 0 in the start information. All unlisted bits are insignificant and take the value zero.

Table 8- 3 Assignments of diagnostic data record 0

Byte	Bit	Meaning	Remarks	Event no.
0	0	Error on module	Is set at each diagnostic event	8:x:00
	1	Internal error.	Set for all internal faults	8:x:01
	2	External fault	Set for all external faults	8:x:02
	3	Channel fault		8:x:03
	4	Fault in external auxiliary voltage	1L supply missing[1]	8:x:04
	6	Module parameters not assigned[2]	Parameter data record 0 not received	8:x:06
	7	Error in parameter assignment[2]	Incorrect parameter, mismatch, or consistency check failure (if enabled)	8:x:07
1	0..3	Type class	Always assigned with 8	
	4	Channel information available		
2	0	Incorrect or missing module	Set when SIMATIC Micro Memory Card is missing.	8:x:31
	2	Operating state STOP	Set when not in RUN mode	8:x:32
	3	Watchdog tripped[2]	Module fault	8:x:33

Byte	Bit	Meaning	Remarks	Event no.
3	1	Processor failure[2]	Processor self-test failed	8:x:41
	2	EPROM error[2]	Flash memory checksum error	8:x:42
	3	RAM error[2]	RAM test error during startup	8:x:43
	6	Hardware interrupt lost	Hardware interrupt has been detected and cannot be signaled since the same event has not yet been acknowledged by the user program in the CPU.	8:x:46
1 I/O and encoder diagnostics, inputs, and outputs are invalid or off. The module goes to STOP. 2 The module goes to STOP.				

Assignments of diagnostic data record 1

The first four bytes of diagnostics data record 1 are identical with diagnostics data record 0. Data Record 1 reports channel-specific diagnostics. The additional bytes are used by data record 1 to report input, output, and encoder interface diagnostics, according to channel types. You can use SFC 59 to read this diagnostic data record.

The following table shows the assignments of diagnostic data record 1. All unlisted bits are insignificant and take the value zero. (**Note:** Diagnostic information is not updated while the "Busy" bit of the module status byte is "1".)

Table 8- 4 Assignments of diagnostic data record 1

Byte	Bit	Meaning	Remarks
0..3	—	Same as data record 0	
Input diagnostics — channel type $F0_H$			
4		Channel type $F0_H$.	Channel type diagnostics
5		8 (length of channel in bits)	Lists the number of diagnostic bits per channel
6		1 (channel count)	Number of successive channels of the same type
7		Channel vector	
8	5	Missing I/O supply voltage (2L)	
Note: When missing I/O supply voltage diagnostics is active, inputs I0 to I7, outputs Q0 to Q7, and I/O diagnostics are invalid.			
Encoder interface diagnostics — channel type $F4_H$			
9		Channel type $F4_H$.	Channel type diagnostics
10		16 (length of channel in bits)	Lists the number of diagnostics bits per channel
11		1 (channel count)	Number of successive channels of the same type
12		Channel vector	

Byte	Bit	Meaning	Remarks
13	0	Wire-break at symmetrical incremental encoder (RS422)	SSI or 5 V encoder
	1	SSI frame error	SSI encoder selected
	3	Encoder sensor supply overload	Encoder selected or inputs used
	4	Missing encoder supply voltage (3L)	Encoder selected or inputs used
14	—	—	Encoder diagnostics, byte 2
Note: When missing encoder supply voltage diagnostics is active, inputs I8 to I14, encoder outputs, and encoder diagnostics are invalid.			
Output diagnostics — channel type 72_H			
15		Channel type 72_H.	Channel type diagnostics
16		8 (length of channel in bits)	Lists the number of diagnostic bits per channel
17		8 (channel count)	Number of successive channels of the same type
18		Channel vector	
19	2	Output 0 overload	Output diagnostics, byte 1
20	2	Output 1 overload	Output diagnostics, byte 2
21	2	Output 2 overload	Output diagnostics, byte 3
22	2	Output 3 overload	Output diagnostics, byte 4
23	2	Output 4 overload	Output diagnostics, byte 5
24	2	Output 5 overload	Output diagnostics, byte 6
25	2	Output 6 overload	Output diagnostics, byte 7
26	2	Output 7 overload	Output diagnostics, byte 8
27	—	00	Even byte length filler
Note: Because it is not possible to sense an overload when an output is off, the overload report will be removed three (3) seconds after the overload condition is corrected or the output is turned off.			

Assignments of diagnostic data record 128

The following table shows the assignments of diagnostic data record 128. You can use SFC 59 (RD_REC) to read data record 128 for diagnostic information, product order number, firmware version, and module status information.

Table 8- 5 Assignments of diagnostic data record 128

Byte	Meaning	Remarks
0 - 27	Diagnostics	Same as diagnostic data record 1
28 - 47	Order number (6ES7 352-5AHXX-0AE0)	Product order number for FM 352-5
48 - 49	Type ID	>08C1
50 - 51	Hardware version	
52 - 53	Reserved	
54 - 65	Reserved	
66 - 69	Firmware version number	
70 - 74	FPGA size	Number of bytes for FPGA download
75 - 76	Current loaded FPGA program	See note 1
77 - 78	Module status information	See note 2
79	Even byte filler	00
1 This number is the consistency check word as it appears after an FM 352-5 compile and download. In Test mode, this is the FPGA test program version. 2 See status bytes 1 and 2 in "Definitions of the Control Bytes and Status Bytes", section "User data interface (Page 201)".		

Wire-Break Diagnostics

The following table lists some of the possible causes of an encoder wire-break and some possible actions you can take to remedy the problem. The diagnostic function cannot identify the exact cause of the fault. It is also not possible for wire-break diagnostics to detect all connection and hardware faults.

Table 8- 6 Encoder broken wire diagnostics

Possible causes	Possible corrective actions
Encoder cable broken or not plugged in.	Check the encoder cable to ensure that wires are properly connected. Ensure that your installation conforms to the encoder specifications and to the FM 352-5 module requirements. Check the parameters that you assigned in the "Hardware Configuration" dialog, "Parameters" tab to ensure correct setup.
Encoder has no quadrature signals.	
Incorrect pin assignment.	
Encoder signals short-circuited.	
The encoder is not operating.	

Note

When broken wire diagnostics is enabled and the SSI absolute encoder is not selected, signals A, /A (inverse), B, /B (inverse), and N, /N (inverse) signals are checked.

When broken wire diagnostics is enabled for an SSI absolute encoder, only signals D and /D (inverse) are checked.

8.3 Interrupts

Alarm processing

The FM 352-5 can trigger hardware (process) interrupts and diagnostic interrupts. You service these interrupts in an interrupt OB. If an interrupt is triggered and the corresponding OB is not loaded, the CPU changes to STOP (refer to the *Programming with STEP 7* manual).

You can enable interrupt servicing as follows:

1. Enabling general interrupts for the entire module:
   - Select the module in HW Config.
   - Using the menu command, "Edit > Object Properties > Parameters tab > Basic Parameters":
     - Enable interrupt generation and choose the appropriate interrupt.
     - Select the folder for enabling process interrupts and enable (check mark) those process interrupts events that are appropriate.
   - Save and compile the hardware configuration.
   - Download the hardware configuration to the CPU.
2. Click on the "Program" tab, compile the FM application, and then download to the FM 352-5.

Lost Hardware Interrupts

If the processing of a hardware interrupt is not yet completed in the hardware interrupt OB, the module registers all subsequent hardware interrupt events. If an event occurs again before the hardware interrupt can be triggered, the module triggers the "hardware interrupt lost" diagnostic interrupt.

Evaluation of a Hardware Interrupt

If a hardware interrupt is triggered by the FM 352-5, the following information is available in the double-word variable OB40_POINT_ADDR.

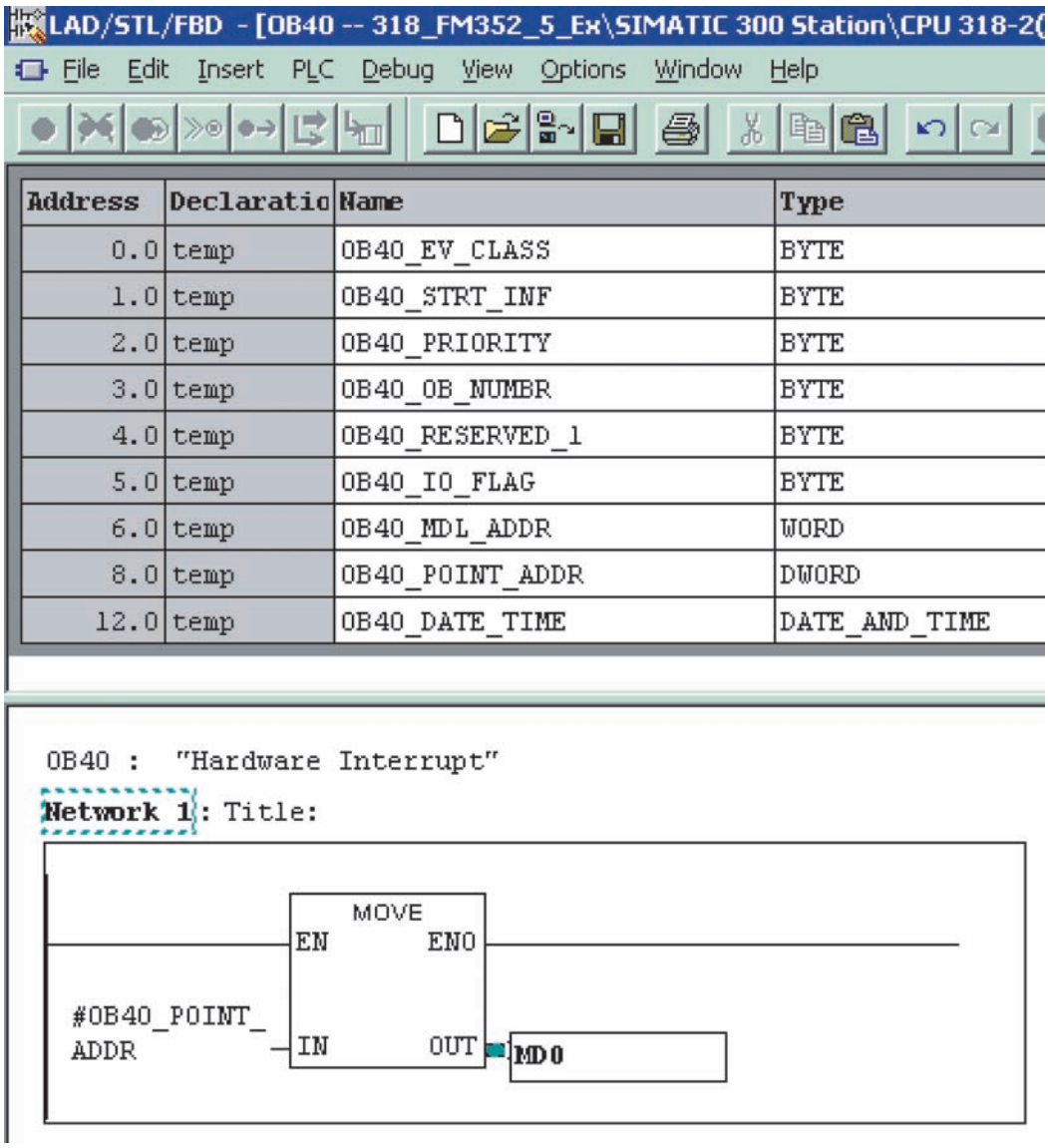


Figure 8-1 Accessing the OB40 interrupts through Ladder Logic

Table 8- 7 Content of the double word 0B40_POINT_ADDR

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Int. 7	Int. 6	Int. 5	Int. 4	Int. 3	Int. 2	Int. 1	Int. 0
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0

8.4 Error correction

Overview

The following table lists the diagnostic faults/errors reported by the FM 352-5 module in data record 0, data record 1, or data record 128 according to the byte and bit numbers. Each fault is reported by STEP 7 in online mode as shown in the table. The description of each error and its possible causes are also given in the table.

Table 8- 8 Errors Reported by the Module and Possible Causes

Byte	Bit	STEP 7 online message	FM 352-5 fault/error description	What the diagnostic fault/error means	Possible causes of error
0	0	Faulty module	Set for all errors The red SF LED is on for **all** errors.	Check DR0, byte 0, bit 1:3 for error entries. FM 352-5 is in STOP mode. **Note:** Diagnostic interrupts must be enabled before they can be reported.	Use the STEP 7 or FM 352-5 diagnostics tools to analyze the problem.
0	1	Internal error	Set for any internal error	The error is caused by a program or parameter assignment error. FM 352-5 is in STOP mode.	Use the STEP 7 or FM 352-5 diagnostics tools to analyze the problem.
0	2	External error	Set for any external error not reported by channel	The error is external to the FM 352-5, and there is no channel data.	Use the STEP 7 or FM 352-5 diagnostics tools to analyze the problem.
0	3	Channel error	Set for any channel error	The error is external and confined to a channel of FM 352-5.	Use the STEP 7 or FM 352-5 diagnostics tools to analyze the problem.
0	4	No external auxiliary voltage	1L supply missing The green DC5V LED is off.	The 24V input to the FM 352-5 1L terminal is not present, or is below specified minimum voltage. The FM 352-5 has detected that there is no power on the S7-300 backplane. **Note:** This diagnostics interrupt must be enabled before it can be reported.	The 24V supply or the wiring that connects to the FM 352-5 1L terminal is bad. The voltage is not 20.4 to 28.8 V at the 1L terminal. The terminals are not screwed tight. The terminal strip is not seated correctly. The S7-300 backplane is faulty.
0	6	Parameters have not been assigned to the module	Parameter data record 0 not received	The FM 352-5 has not received parameter assignment data from the PLC, or the module has lost parameter assignment data. A communications error has occurred in the system.	The PLC hardware configuration has errors. The system communications network is faulty. The system must be restarted and parameters assigned again.

Byte	Bit	STEP 7 online message	FM 352-5 fault/error description	What the diagnostic fault/error means	Possible causes of error
0	7	Incorrect parameters on the module	Parameter assignment error	The FM 352-5 program consistency check has failed. This means that the program or parameters that were loaded to the FM 352-5 module from the SIMATIC Micro Memory Card or the PG do not match the parameters that were downloaded from the PLC. **Note:** The consistency check may be disabled in the FM 352-5 module "Advanced Parameters" folder. Parameter data from the PLC is not allowed for FM 352-5.	The FM 352-5 program on the SIMATIC Micro Memory Card does not match the hardware configuration stored on the PLC and loaded to the FM 352-5 module on startup or a transition of the PLC from STOP to RUN. The FM 352-5 program has not been compiled and downloaded by (1) FM 352-5 and (2) S7 Hardware Configuration since being changed. The FM 352-5 hardware configuration has not been compiled and downloaded by (1) FM 352-5 and (2) S7 Hardware Configuration since being changed. The runtime parameter assignment data (by SFC) for the FM 352-5 contains an error.
1	4		Set when 0.3 is set	The error is external and confined to a channel of FM 352-5.	Use the STEP 7 or FM 352-5 diagnostics tools to analyze the problem.
2	0	User module incorrect, missing.	Set when SIMATIC Micro Memory Card is missing. The red MCF LED is on.	A Micro Memory Card has not been detected. **Note:** This diagnostics interrupt must be enabled before it can be reported.	SIMATIC Micro Memory Card is missing. The SIMATIC Micro Memory Card is not correctly inserted. The SIMATIC Micro Memory Card connectors are dirty.

Byte	Bit	STEP 7 online message	FM 352-5 fault/error description	What the diagnostic fault/error means	Possible causes of error
2	2	Faulty module, internal error **Note:** STEP 7 does not provide a message for "FM module in STOP mode".	Set when not in RUN mode The yellow STOP LED is on.	The FM 352-5 has been set to STOP with the RUN/STOP selector. The FM 352-5 has not received the RUN command, or has received a STOP command from the PLC. The FM 352-5 has not received a change to the "RUN/Test" command at startup. The parameter setting is for FM 352-5 to "RUN when PLC Stops" but it is in the Test mode. The FM 352-5 has entered or will not leave the STOP mode because of a bad parameter or program error.	The FM 352-5 RUN/STOP selector is in the STOP position. The PLC RUN/STOP selector is in the STOP position and the FM 352-5 has not been enabled to "RUN when PLC stops" (normal operation only). The FM 352-5 receives the "Normal/RUN" command, and does not have a valid program loaded via PG or SIMATIC Micro Memory Card. The PLC program does not have all the FM 352-5 interface FBs and DBs installed and enabled. (See the section "Getting started (Page 21)" in this manual). The initial FM 352-5 "RUN/Test" command was not preceded by another command. If the parameter assignment error bit (DR 0, byte 0, bit 7) is also set, take the corrective action for that error code.
2	3	Watchdog time-out	Watchdog error	The FM 352-5 processor has performed an illegal operation and has been stopped.	An internal fault or external EMI has caused a fatal error. Restart the FM 352-5 and check whether the error persists. If it does, the FM 352-5 is either defective or exposed to heavy electrical interference.
3	1	Processor failure	Processor self-test failed	The FM 352-5 processor did not successfully complete internal self-test on power up.	An internal fault or external EMI has caused a fatal error. Restart the FM 352-5 and check whether the error persists. If it does, the FM 352-5 is either defective or exposed to heavy electrical interference.
3	2	EPROM error	Flash memory checksum error	The FM 352-5 program memory has failed power on test.	An internal fault or external EMI has caused a fatal error. Restart the FM 352-5 and check whether the error persists. If it does, the FM 352-5 is either defective or exposed to heavy electrical interference.

Byte	Bit	STEP 7 online message	FM 352-5 fault/error description	What the diagnostic fault/error means	Possible causes of error
3	3	RAM error	RAM test error during startup	FM 352-5 work memory has failed the power on self-test.	An internal fault or external EMI has caused a fatal error. Restart the FM 352-5 and check whether the error persists. If it does, the FM 352-5 is either defective or exposed to heavy electrical interference.
3	6	Process alarm lost	Set if there is a hardware interrupt queue overflow.	Hardware interrupts from the FM 352-5 are occurring faster than the PLC can service them. Hardware interrupts to the FM 352-5 are occurring more often than the FM 352-5 processor can service them. **Note:** This diagnostics interrupt must be enabled before it can be reported.	The frequency of the hardware interrupt is too high. The program of the interrupt OB is too long. The PLC is not fast enough.
8	5	Digital input sensor or load voltage missing	Missing input/output supply voltage (2L) The red IOF LED is on.	The 24V input to the FM 352-5 2L is not present, or is below specified minimum voltage. Other I/O diagnostics are not valid when this error occurs. **Note:** This diagnostics interrupt must be enabled before it can be reported.	The 24V supply or the wiring that connects to the FM 352-5 2L terminal is faulty/incorrect. The voltage is not 20.4 to 28.8V on the 2L terminal. The terminals are not screwed tight. The terminal strip is not seated correctly.
13	0	FM positioning, broken wire in incremental encoder	Wire-break at symmetrical incremental encoder (RS422) The red IOF LED is on.	The FM 352-5 differential inputs AD, /AD, B, /B, N, /N (only AD, /AD, only, if SSI encoder is enabled) are not wired correctly, not connected, or they have incorrect signals applied. **Note:** This diagnostics interrupt must be enabled before it can be reported.	Faulty wiring from the FM 352-5 encoder interface to the encoder. The terminals are not screwed tight. The terminal strip is not seated correctly. When no encoder is selected, or if a differential encoder (RS-422) is selected, all 6 inputs must be connected to RS-422 compatible output drivers. The encoder connecting cables are too long. The encoder is faulty.

Byte	Bit	STEP 7 online message	FM 352-5 fault/error description	What the diagnostic fault/error means	Possible causes of error
13	1	FM positioning, error in absolute value encoder	SSI frame error The red IOF LED is on.	The SSI encoder data does not match the expected format for the type encoder that was set. The SSI encoder data is not being received by the FM 352-5. **Note:** This diagnostics interrupt must be enabled before it can be reported.	Faulty wiring from the FM 352-5 encoder interface to the encoder. The terminals are not screwed tight. The terminal strip is not seated correctly. The wrong encoder parameters have been selected for the encoder used. The encoder connecting cables are too long. The encoder is faulty.
13	3	FM positioning, voltage monitor sensing	Encoder sensor supply fault (overload) The red IOF LED is on. *and* The red 24VF LED is on. *or* The red 5VF LED is on.	The 24 V DC or 5 V DC encoder supply is shorted or overloaded. Other FM positioning diagnostics are invalid if this error occurs. **Note:** This diagnostics interrupt must be enabled before it can be reported.	Faulty wiring from the FM 352-5 encoder interface to the encoder. The encoder is overloading or shorting the 24 V DC or 5 V DC supply.
13	4	FM positioning, voltage monitoring +/- 15V	Missing encoder supply voltage (3L) The red IOF LED is on.	The 24 V input to the FM 352-5 3L is not present or is below specified minimum voltage. Short-circuit or overload at the 5 VDC encoder supply. Other FM positioning diagnostics are invalid if this error occurs. **Note:** This diagnostics interrupt must be enabled before it can be reported.	Incorrect wiring of the 24 V supply to the FM 352-5 3L terminal. The voltage is not 20.4 to 28.8 V at the 3L terminal. The terminals are not screwed tight. The terminal strip is not seated correctly. The 5 V DC supply wiring is incorrect. Short-circuit or overload at the 5 VDC encoder supply.
19	2	Channel 0 digital output short circuit	Channel x is overloaded. The red IOF LED is on.	The FM 352-5 output Qx is shorted or has been overloaded. This diagnostics function is only activated if the channel is enabled and a fault has occurred. **Note:** This diagnostics interrupt must be enabled before it can be reported.	The connecting wires or the load have intermittent or continuous faults. The load is above the maximum current rating. The output is switching beyond the maximum specified operating frequency.
20	2	Channel 1 . . .			
21	2	Channel 2 . . .			
22	2	Channel 3 . . .			
23	2	Channel 4 . . .			
24	2	Channel 5 . . .			
25	2	Channel 6 . . .			
26	2	Channel 7 . . .			

Using the FM 352-5 with non-S7 masters

9

9.1 Prerequisites for non-S7 applications

Overview

The FM 352-5 module can be used in a non-S7 automation system over a PROFIBUS-DP I/O channel. The module is designed to operate as a 16-byte input/16-byte output module when installed in an ET 200M rack. The PROFIBUS-DP interface is provided by an IM153-1 or IM153-2 module.

Tools and Prerequisites

The non-S7 automation system must have DP master capability and its configuration tool must be capable of importing the GSD file for the ET 200M.

The FM 352-5 module must have a SIMATIC Micro Memory Card programmed with STEP 7. The SIMATIC Micro Memory Card must contain SDB 32512 that was created in the STEP 7 environment.

The user program of the non-S7 automation system must manage the data transfer between itself and the module according to the declared interface of the application FB as programmed in STEP 7. The automation system must also perform mode control via the control bytes.

The following sections provide further details on how to use the FM 352-5 in a third-party automation system.

See also

Installing the configuration/programming software (Page 45)

Overview (Page 65)

9.2 Non-S7 CPU system requirements

Importing GSD File Data

In systems with third-party CPUs, you must import the GSD file with a configuration software package that can incorporate the GSD file data to create your hardware configuration. Consult the documentation for your system for information on how to import the GSD file. You can find GSD files on the Internet at Siemens - Industry Automation and Drive Technologies - Service&Support (http://support.automation.siemens.com/WW/view/en/10805317/133100). The path can also be found at http://www.profibus.com under the "Libraries" tab, Siemens.

SIMATIC Micro Memory Card Programming

For third-party CPU systems, you must program the SIMATIC Micro Memory Card independent of the FM 352-5 module. To do this, you need either a Siemens PG with SIMATIC Micro Memory Card programming capability or a PROM writer that can program an SIMATIC Micro Memory Card. After programming the SIMATIC Micro Memory Card, insert the SIMATIC Micro Memory Card in the FM 352-5 module.

Developing an Interface Function

As a non-S7 CPU system user, you must develop a function in your program to control the module's interface that meets your specific system's requirements.

Your program interface must be able to command the FM 352-5 module to enter normal operation and RUN/STOP modes. It must also manage the transfer of data between the module and the master CPU.

In addition, if you have not yet commissioned the FM 352-5 module in your STEP 7 environment when you created and tested your program, you may want to incorporate controls to be able to switch to Test mode to check that the module is correctly connected to the inputs and outputs and that the module counter configuration is correct. Single-scan program execution is another tool that is useful in testing a program.

9.3 User data interface

User data

The master CPU has access to a total of 16 bytes of input data and 16 bytes of output data during the FM 352-5 module operation. The first two output bytes are used to transmit **control** information, and the first two input bytes output **status** information to the CPU.

In normal operation, the remaining 14 bytes are free-from inputs and outputs exchanged between the module and the CPU, as shown in the table below.

Table 9- 1 User Data Input and Output Bytes in Normal Mode

Byte address	Output data (to module)	Input data (from module)
0	Control byte 1	Status byte 1
1	Control byte 2	Status byte 2
2 . . 15	Free outputs . . Free outputs	Free inputs . . Free inputs

In Test mode, the remaining 14 bytes are pre-defined, as shown in the table below. This mode allows the module to transmit specific internal information to and from the Test FB to check program operation and wiring.

Table 9- 2 User Data Input and Output Bytes in Test Mode

Byte address	Output data (to module)	Input data (from module)
0	Control byte 1	Status byte 1
1	Control byte 2	Status byte 2
2	Digital outputs (0 - 7)	Digital inputs (0 - 7)
3		Digital inputs (8 - 14)
4		
5		Power supply status
6		SSI status
7		Output overload
8		SIMATIC Micro Memory Card status
9		
10		Encoder status 1
11	Encoder control	Encoder status 2
12	Encoder load value MSB	Encoder data MSB (32-bit)
13	Encoder load value	Encoder data
14	Encoder load value	Encoder data MSB (16-bit)
15	Encoder load value LSB	Encoder data LSB

Definitions of the Control Bytes and Status Bytes

The control and status bytes are defined in the following table. The control bytes allow your program to control the operation of the module (RUN, STOP, or single scan). The status bytes allow your program to determine the status of the module as well as the status of the SIMATIC Micro Memory Card inserted in the module. The following table defines the bit patterns for each of the operating modes, the operating status conditions, and the SIMATIC Micro Memory Card status.

Table 9- 3 Control Bytes and Status Bytes for the FM 352-5

Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Control byte 1	Reserved	Reserved	Reserved	Reserved	**Operating mode**			
Control byte 2	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Status byte 1	Reserved	**BUSY***	Reserved	Reserved	**Operating state**			
Status byte 2	Reserved	Reserved	Reserved	Reserved	Reserved	**SIMATIC Micro Memory Card status**		
* This bit indicates that the module is not ready for data transfer or other operations, and will not update I/O or diagnostics.								

Table 9- 4 Bit Definitions of the Control and Status Bytes

Bits	Command to module	Bits	Module reaction
	Operating mode		**Operating state**
0000	Continue current normal operation	0001	Normal mode — STOP
0001	Normal mode — STOP	0010	Normal mode — RUN
0010	Normal mode — RUN	0101	Test mode — STOP (outputs off)
0101	Test mode — STOP	0110	Test mode — RUN
0110	Test mode — RUN	1010	Single scan mode
1010	Single scan mode — SCAN once*		
1000	Single scan mode — no change (idle)		**SIMATIC Micro Memory Card status**
		000	SIMATIC Micro Memory Card OK
		001	SIMATIC Micro Memory Card not found
		010	SIMATIC Micro Memory Car d program defective or invalid
		011	SIMATIC Micro Memory Card program missing
		100	SIMATIC Micro Memory Card program corrupt
		111	SIMATIC Micro Memory Card and data record 0/128 do not match (applies to S7 masters only)
* If the single scan bit is set to 1, the module executes one cycle scan when the RUN bit changes from 0 to 1.			

Bit Definitions of the Encoder Status Bytes

The bits of the status bytes defined in the following tables allow your program to determine the status of the encoder.

Table 9- 5 Encoder Status Byte 1

Bit no.	Definition	Module reaction
7 to 1	Reserved	0
0	Encoder selected	1 = encoder has been selected

Table 9- 6 Encoder Status Byte 2

Bit no.	Definition	Module reaction
7	SSI data available	1 = SSI data is available
6	SSI frame	1 = SSI data error
5	Underflow*	1 = underflow of the encoder count
4	Overflow*	1 = overflow of the encoder count
3	Homed	1 = encoder has been homed (synchronized)
2	Home*	1 = encoder is at home (reset) position
1	Last count direction	1 = last count input direction was down
0	Size	1 = encoder counter or SSI encoder is 32-bit
* These bits may change faster than the automation system scan and are not visible most of the time.		

Bit Definitions of the Encoder Control Byte

The bits of the control byte defined in the following table allow your program to control operation of the encoder.

Table 9- 7 Encoder Control Byte

Bit no.	Definition	Command to module
7	Reserved	0
6	Reserved	0
5	Reserved	0
4	Reserved	0
3	Reserved	0
2	Load	1 = load encoder counter
1	Software reset	1 = reset the encoder counter
0	Software hold	1 = hold the encoder counter value

Bit Definitions of the Power Supply Status Byte

The bits of the power supply status byte defined in the following table allow your program to determine the status of each of the power supplies to the module.

Table 9- 8 Power Supply Status Byte

Bit no.	Definition	Module reaction
7	1L missing	1 = missing auxiliary supply voltage (1L)
6	2L missing	1 = missing input/output supply voltage (2L)
5	Encoder sensor supply fault	1 = encoder power supply or wiring fault
4	3L missing	1 = missing encoder supply voltage (3L)
3	Reserved	0
2	Reserved	0
1	Reserved	0
0	Reserved	0

Bit Definitions of the SSI Encoder Status Byte

The bits of the SSI encoder status byte defined in the following table allow your program to determine the status of the SSI encoder.

Table 9- 9 SSI Encoder Status Byte

Bit no.	Definition	Module reaction
7	SSI frame error	1 = SSI data frame fault
6	Differential incremental encoder (RS-422) broken wire	1 = broken wire or encoder malfunction detected
5 - 0	Reserved	0

Bit definitions of the SIMATIC Micro Memory Card status byte

The bits of the SIMATIC Micro Memory Card status byte defined in the following table allow your program to determine the status of the SIMATIC Micro Memory Card.

Table 9- 10 SIMATIC Micro Memory Card status byte

Bit no.	Definition	Module reaction
7	SIMATIC Micro Memory Card error	1 = SIMATIC Micro Memory Card found
6 - 0	Reserved	0

Technical specifications

A

A.1 General technical specifications

The following technical specifications are described in the Operating Instructions SIMATIC S7-300 CPU 31xC and CPU 31x: Installation (http://support.automation.siemens.com/WW/view/en/13008499):

- Standards and certifications
- Electromagnetic compatibility
- Shipping and storage conditions
- Mechanical and climatic environment conditions
- Specifications for insulation tests, protection class, degree of protection, and rated voltage
- Rated voltages

Observing the Design Guidelines

SIMATIC products meet the requirements if you observe the design guidelines described in the manuals when installing and operating the equipment.

Refer also to the installation guidelines regarding lightning safety in section External Protection Circuit for FM 352-5 Boolean Processor (Page 243).

A.2 Technical specifications

Overview

Dimensions and weights	
Dimensions (W x H x D)	80 x 125 x 130 mm
Weight	Approx. 434 g (with 1L connection, without I/O connection or SIMATIC Micro Memory Card)

Data for specific modules	
Number of inputs	12 (24 V DC) 3 (RS-422)
Number of outputs	8

Voltage, Currents, Potentials	
Rated voltage supply to the electronic system (1L+, 2L+, 3L+)	24 V DC, class 2 power supply
• Reverse polarity protection	Yes
• Power failure bypass	5 ms
Electrical isolation	
• Between the field side I/O card (2L) and the encoder card (3L)	75 VDC, 60 VAC
• Between the field side I/O card (2L) and logic	75 VDC, 60 VAC
• Between auxiliary supply (1L) and logic	75 VDC, 60 VAC
• Between auxiliary supply (1L) and field side of encoder or I/O card (2L or 3L)	75 VDC, 60 VAC
• Potential differences between M terminals and central ground	75 VDC, 60 VAC
Insulation tested with	500 VDC
Current consumption	
• From input voltage 1L+ at 20.4 - 28.8 V	Max. 150 mA
• From input voltage 2L+ at 20.4 - 28.8 V	Max. 200 mA
• From input voltage 3L+ with 5.2-V or 24-V encoder	600 mA max., with encoder supply fully loaded
• From input voltage 3L+ at 20.4 - 28.8 V	200 mA max., with no encoder supply load
• From backplane bus	Typically 130 mA

Voltage, Currents, Potentials	
Power loss of the module	Typically 6.5 W

Data for selecting sensors	
Input voltage	
• Rated value	24 VDC
• For signal "1"	11 V to 30 V
• For signal "0"	-30 V to 5 V
Input current	
• For signal "1"	Typically 3,8 mA
• For signal "0"	≤ 1.5 mA
Input frequency	Max. 200 kHz
Hardware input delay	Max. 3 µs
Selectable input delay times	None, 5 µs, 10 µs, 15 µs, 20 µs, 50 µs, 1.6 ms
Minimum pulse duration for program response[1]	1 µs, 5 µs, 10 µs, 15 µs, 20 µs, 50 µs, 1.6 ms
Cable length, sensors	100 meters unshielded, 600 meters shielded. Shielded cable is recommended when less than 1.6 ms filtering is selected.
Minimum pulse duration (max. SW counter frequency)	1 µs (200 kHz)
Connection of two-wire BEROs	Possible
• Permitted bias current	Off (idle): Max. 1.5 mA On: Min. 3,2 mA
[1] The input delay filter is a noise (pulse) filter. It may not reject a continuous wave of 1/delay.	

Data for selecting an actuator (5AH01: current sinking output)	
Output voltage • For signal "1"	Max. (M +0.5 V)
Output current	
• At signal "1" - Nominal value Permitted range	0.5 A 5 mA to 0.6 A
• For signal "0" (discharge current)	Max. 1.0 mA
Total current of the outputs	Max. 4 A
Output delay, (for resistive load)	
• For "1" to "0"	Max. 3.2 µs Typically 1.7 µs

Data for selecting an actuator (5AH01: current sinking output)	
• For "0" to "1"	Max. 2 µs Typically 1,0 µs
Output dv/dt, (for resistive load)	
• For "1" to "0"	Max. 15 V/ µs Typically > 50 V/ µs
• For "0" to "1"	Max. 12 V/ µs Typically > 39 V/ µs
Lamp load	Max. 5 W
Connecting two outputs in parallel	
• For redundant triggering of a load	Possible
• To increase performance	Possible max. 1 A (resistive only)
Triggering a digital input	Not possible
Switching frequency	
• For resistive load	Max. 20 kHz at 0.5 A max. 100 kHz at 0.25 A
• With inductive loads[1]	See "Switching frequency for inductive loads without commutating diodes (Page 220)".
• For lamp load	Max. 10 Hz
Limit (internal) of the inductive circuit interruption voltage	Max. M (+55 V) Typically M (+45 V)
Short-circuit protection for the output[2]	Electronic
• Threshold on	Typically 1.7 A to 3.5 A
Cable length	
• Unshielded	100 m
• Shielded	600 m
[1]: Not protected by inductive kickback >55 mJ [2]: The outputs are not protected against reverse voltage if the current is not limited to < 3 A.	

Data for selecting an actuator (5AH11: current sourcing output)	
Output voltage • For signal "1"	 Min. 2L+ (-0.5 V)
Output current	
• At signal "1" - Nominal value Permitted range	 0.5 A 5 mA to 0.6 A
• For signal "0" (discharge current)	Max. 1.0 mA
Total current of the outputs	Max. 4 A
Output delay, (for resistive load)	
• For "1" to "0"	Max. 6 µs Typically 2,5 µs
• For "0" to "1"	Max. 4 µs Typically 2.5 µs
Output dv/dt, (for resistive load)	
• For "1" to "0"	Max. 15 V/ µs Typically > 50 V/ µs
• For "0" to "1"	Max. 12 V/ µs Typically > 39 V/ µs
Lamp load	Max. 5 W
Connecting two outputs in parallel	
• For redundant triggering of a load	Possible
• To increase performance	Possible max. 1 A (resistive only)
Triggering a digital input	Possible
Switching frequency	
• For resistive load	Max. 20 kHz at 0.5 A max. 100 kHz at 0.25 A
• With inductive loads[1]	See "Switching frequency for inductive loads without commutating diodes (Page 220)".
• For lamp load	Max. 10 Hz
Limit (internal) of the inductive circuit interruption voltage	Max. L+ (-55 V) Typically L+ (-45 V)
Short-circuit protection for the output[2,3]	Electronic
• Threshold on	Typically 1.7 A to 3.5 A

Data for selecting an actuator (5AH11: current sourcing output)	
Cable length	
• Unshielded	100 m
• Shielded	600 m
[1]: Not protected from inductive kickback > 55mJ [2]: The outputs are not protected against reverse voltage if the current is not limited to < 3 A. [3]: L2 interruption sufficient to cause the outputs to become invalid, (but not long enough to signal missing diagnostic) will cause "output overload" diagnostic on any outputs that are on.	

Encoder section	
Input frequency	
RS-422 input	1 MHz max.
24 V DC input	Max. 200 kHz
Encoder signal evaluation	Pulse & direction, x1, x2, x4
Source reset	None, HW, SW, HW and SW, HW or SW
Source reset value	Constant 0, min/max value, load value
Reset signal type	Edge, level
Source load value	Constant, module application
Source stop	None, HW, SW, HW and SW, HW or SW
Load value	User input or module application
Count range minimum	User input
Count range maximum	User input
Main count direction	Up count, down count
Hardware source stop	Inputs 0 to 14
Hardware source reset	Inputs 0 to 14
Counting modes	Continuous, single, periodic
Count range, 16-bit Count range, 32-bit	-32768 to 32767 -2147483648 to 2147483647
Encoder signals	
• 5 V (RS-422)	A, /A, B, /B and N, /N
• 24 V (HTL)	A, B, and N
SSI encoders	
• SSI signals	D, /D, CK and /CK
• Message frame length	25 bits or 13 bits, Gray code
• Resolution	Max. 16,777,216
• Delay times (monoflop)	16, 32, 48 or 64 µs
• SSI shift register length	13 bits or 25 bits
• Clock rate	125 kHz, 250 kHz, 500 kHz, or 1 MHz

Encoder section	
• Data shift direction	Left or right
• Data shift length	0 to 12 bits
• SSI modes	Master, listen (up to two stations)
Cable length, HTL incremental encoders, Siemens type 6FX2001-4	25 m shielded, max. at 50 kHz 50 m shielded, max. at 25 kHz
Cable length, RS-422 (5V) incremental encoders, Siemens type 6FX2001-2, 5 V supply	32 m shielded, max. at 500 kHz
Cable length, RS-422 (5V) incremental encoders, Siemens type 6FX2001-2, 24-V supply	100 m shielded, max. at 500 kHz
Cable length, SSI absolute encoders, Siemens type 6FX2001-5, 24-V supply	Max. 320 m shielded, at 125 kHz Max. 160 m shielded, at 250 kHz Max. 60 m shielded, at 500 kHz Max. 20 m shielded, at 1 MHz

Frame times of the encoders		
Encoder frame times	13-bit	25-bit
• 125 kHz	108 µs	204 µs
• 250 kHz	54 µs	102 µs
• 500 kHz	27 µs	51 µs
• 1 MHz	14 µs	26 µs

Sensor power supply outputs	
5.2 V output power for sensors and encoders[1]	
• Supply output	5.2 V ± 5%
• Output current	Max. 250 mA
• Time-of-day interrupts	Yes, electronic. (Not protected from application of normal or counter voltage.)
• Diagnostics	Yes
24 V output power for sensors and encoders[1]	
• Supply output	3L+ -1 V (max.)
• Output current	Max. 400 mA
• Time-of-day interrupts	Yes, electronic. (Not protected from application of normal or counter voltage.)
• Diagnostics	Yes
[1] Only one of the output power supplies for encoders can be used at a time, not both together.	

Status, interrupts, diagnostics	
Interrupts	Yes
• Hardware interrupts – 1L missing[1] – 2L missing[1] – 3L missing[1] – Encoder overload[1] – Encoder broken wire[1] – SSI frame error[1] – Output [1, 2] overload – SIMATIC Micro Memory Card error	• Parameters can be assigned – Diagnostics data record – Diagnostics data record – Diagnostics data record – Diagnostics data record – Diagnostics data record – Diagnostics data record – Diagnostics data record – Diagnostics data record
• Hardware interrupts	Yes, 8 Hardware interrupts **Note:** Hardware interrupts; "Alarm-N is set on the PROFIBUS after a 24 V input sets Intr [x]" • typ. 63 µs • Max. 200 µs Maximum sustained Hardware interrupt rate (without Hardware interrupt lost) • 400 Hz (2.5 ms)
Diagnostic functions	Yes
• Group error display	SF, red LED
• SIMATIC Micro Memory Card error [3]	MCF, red LED
• Monitoring of the power supply voltage of the electronics	DC5V, green LED
• I/O fault status	IOF, red LED
• RUN mode	RUN, green LED
• STOP mode	STOP, yellow LED
• Power supply fault (encoder)	5VF, red LED 24VF, red LED
• Input status	Green LED (I 0 to I 11)
• Output status	Green LED (Q 0 to Q 7)

Boolean processor operation	
Execution time	1 µs
PLC update cycle time	≈2.6 ms (max. 5 ms)
Program and hardware response time	2 to 6 µs, input to output

[1] Diagnostic indications for these conditions are available only when enabled in the "Parameters" tab of the FM 352-5 "Properties" dialog.

[2] "Output overload" diagnostics may not be reported if the output pulse duration is less than 2 ms (5AH01), or less than 20 µs (5AH11).

[3] MCF LED status is only updated when the SIMATIC Micro Memory Card is removed or when the module is to read or write the SIMATIC Micro Memory Card.

A.3 Block diagram

Description

The following figure shows a block diagram with the essential hardware components of the FM 352-5 module.

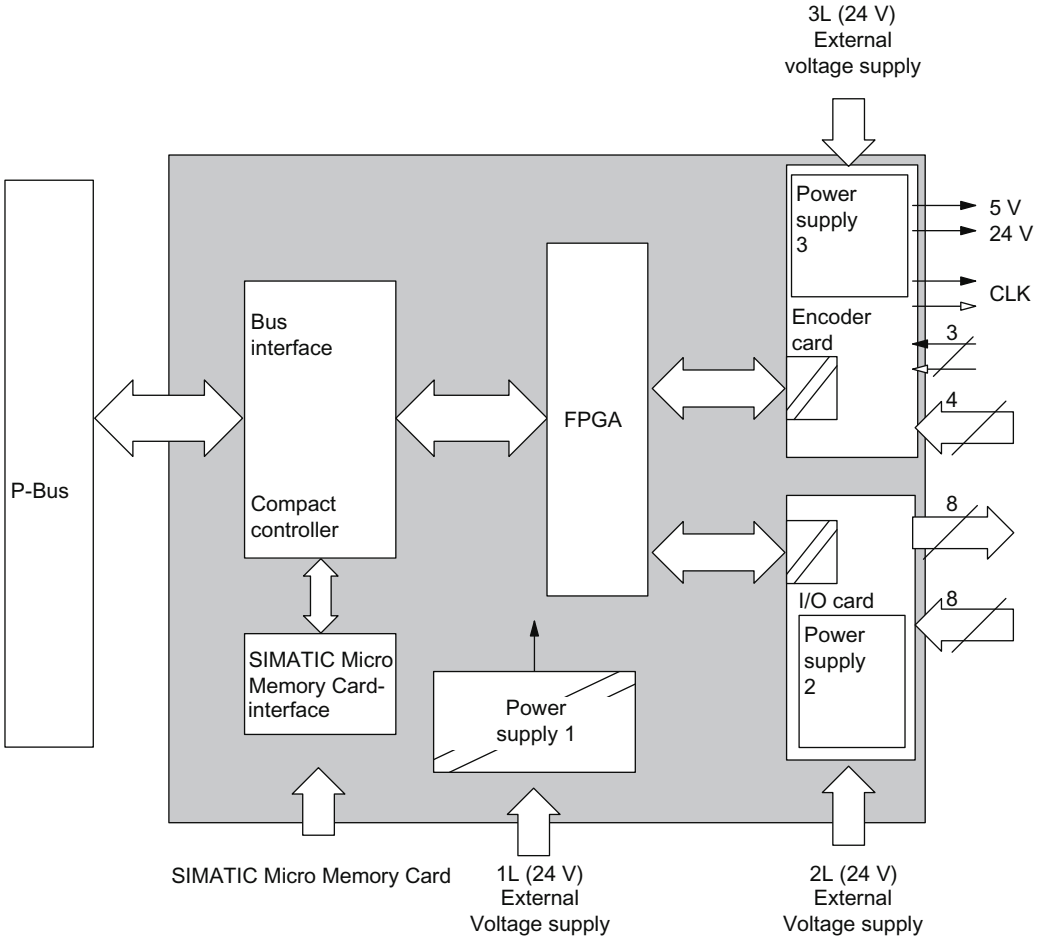


Figure A-1 Functional Block Diagram of the FM 352-5 Module

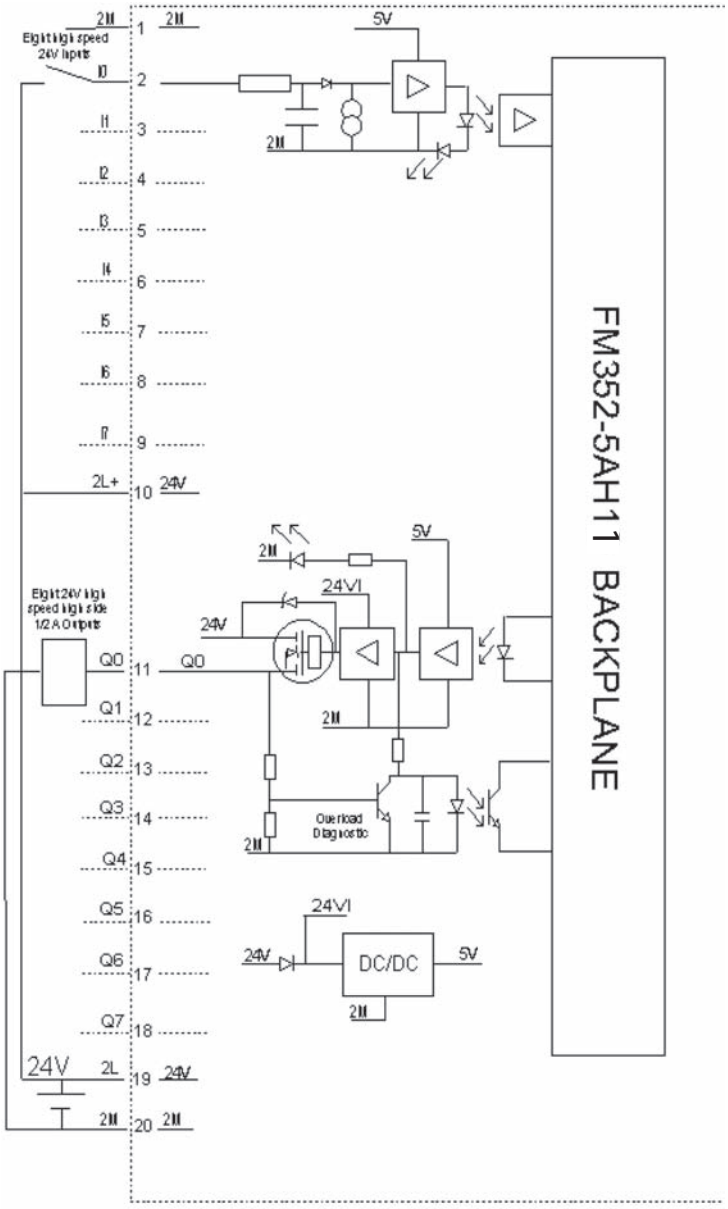


Figure A-2 Function Block Diagram of the I/O Card for module FM 352-5AH11-0AE0

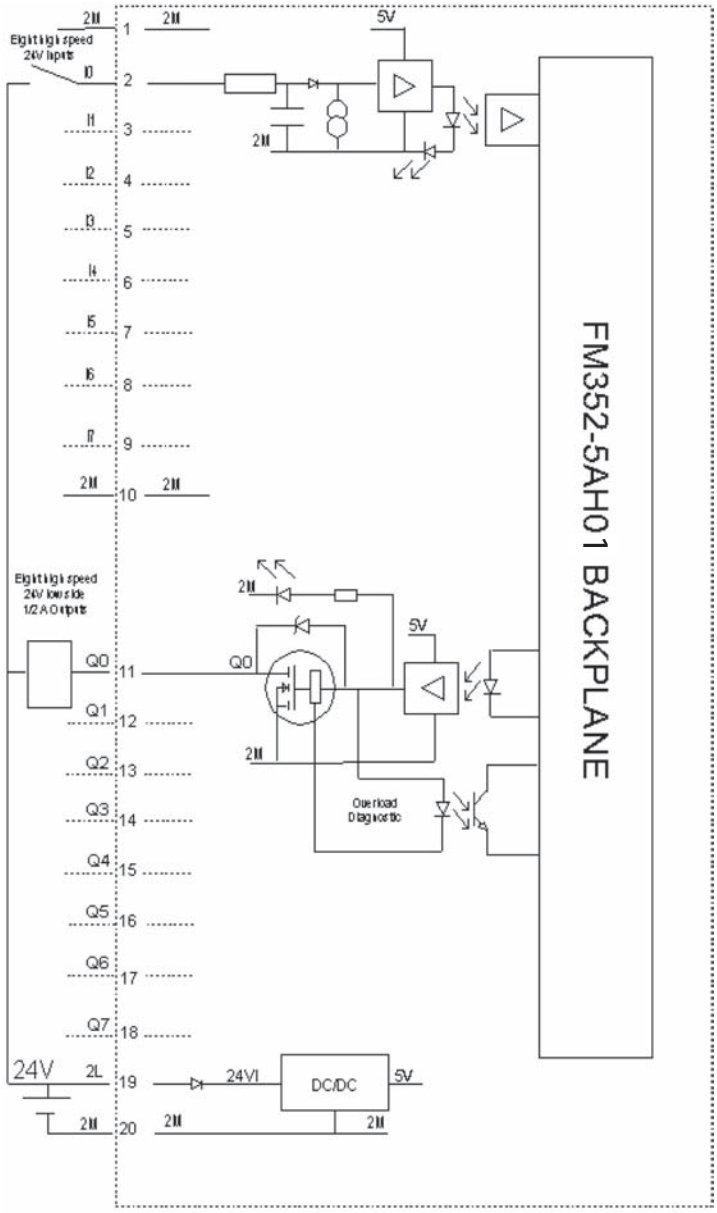


Figure A-3 Function Block Diagram of the I/O Card for module FM 352-5AH01-0AE0

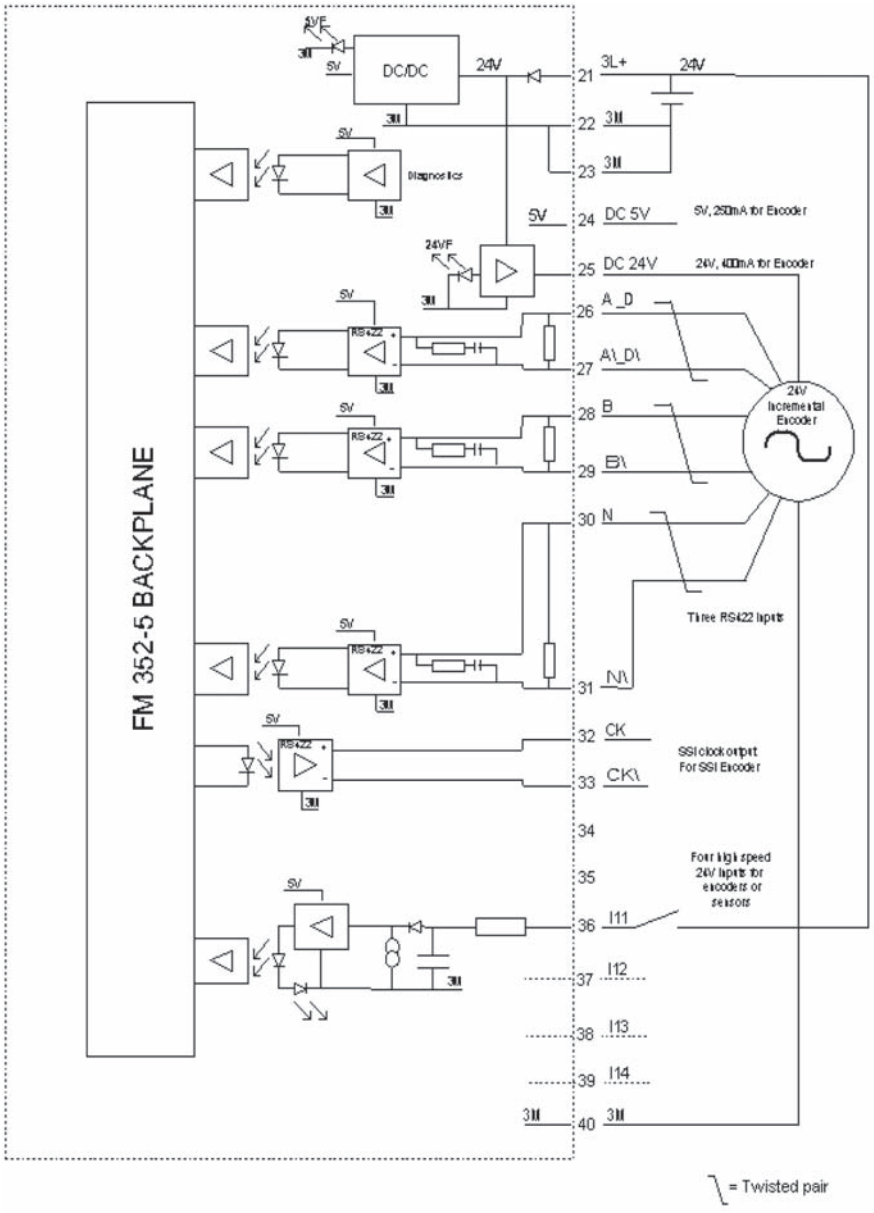


Figure A-4 Function Block Diagram of the Encoder Card for module FM 352-5AHx1

A.4 Operating data

Switching Frequency Derating Charts

The following figure shows how the output channels are derated by the operating temperature as the switching frequency increases up to 100 kHz at an output load of 500 mA.

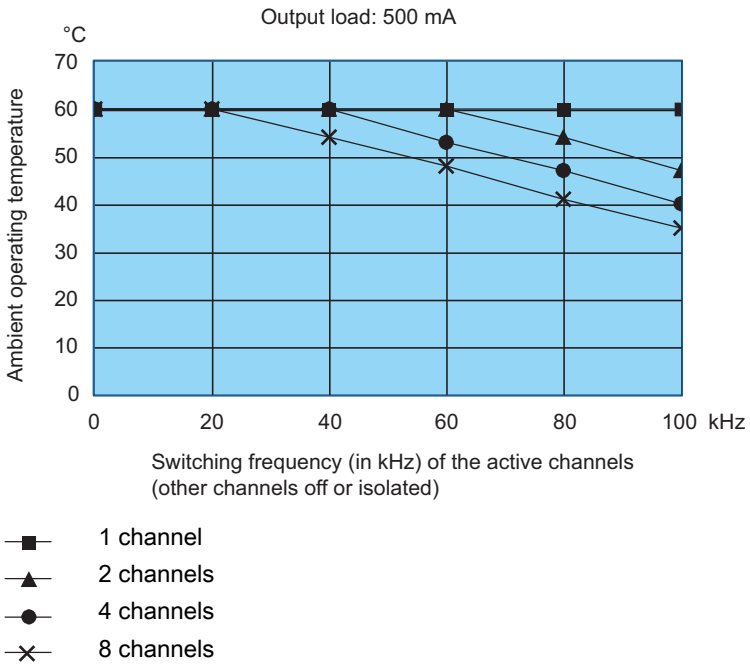


Figure A-5 Switching Frequency and Ambient Temperature at 500 mA Output Load

The following figure shows how the output channels are derated for maximum load current as the switching frequency increases up to 100 kHz at 60° C operating temperature.

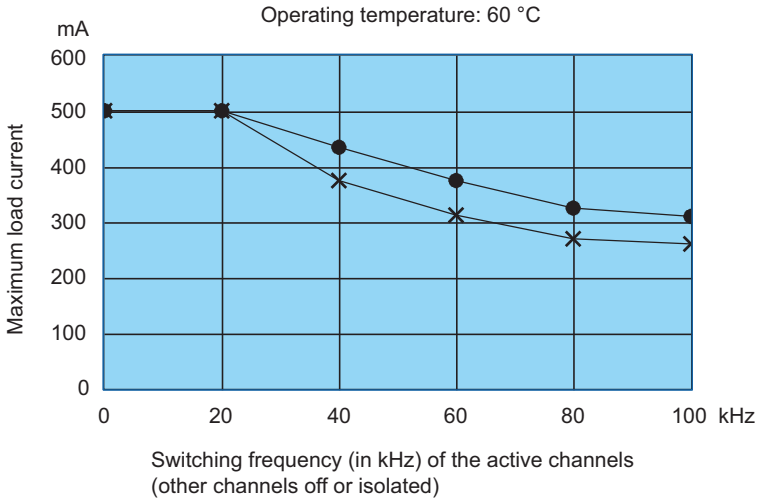


Figure A-6 Switching Frequency and Maximum Output Current at 60° C

A.5 Switching frequency for inductive loads without commutating diodes

Maximum Inductor Energy Rating

The energy contained in the inductance of the relay will damage the FM 352-5 output if the destruction limit is exceeded. The energy is proportional to the inductance of the relay and the current through the relay.

Determining the Inductive Load Characteristics

If you do not know the characteristics of your inductive load, use this procedure to estimate them.

If you know *R* and *L*, you can resolve to *T* with the equation *T=L/R*. To determine the characteristics of an unknown load, measure the relay steady state "On" current '*I*' at 24 V. Measure '*T*' the time that the current requires when the relay is on to reach 63.2% of the "On" value. '*R*'= 24 V/*I,* and *'L' = T*R.* Example: First, assume on "on" current of 100 mA. Then assume relay current rise time ('*T*') from 0 to 63% of 100 mA (63 mA) is 2ms. 2 ms = *L/R*. Resolved to *R*, 24/0.1 = 240 ohms. Resolved to *L*, 0.002*240 = 480 mH.

Reading Graph 1

To determine if the energy stored in the inductor can be handled by the FM 352-5 module without commutation diodes, refer to the following figure. Example: With the values determined from the inductive load characteristics (relay current = 100 mA and relay inductance = 480 mH), follow the vertical line from 100 mA up to the 0.5 H line. This is well below the switching limit line. Note that an inductor of up to 2 H is acceptable at 100 mA. If the inductor had been larger than 2 H or the current had been greater than 200 mA, then commutation diodes are required parallel to the relay. There is no special inductive switching limitation if commutation diodes are used. The following diagram shows the maximum relay inductance dependent of the inductor current.

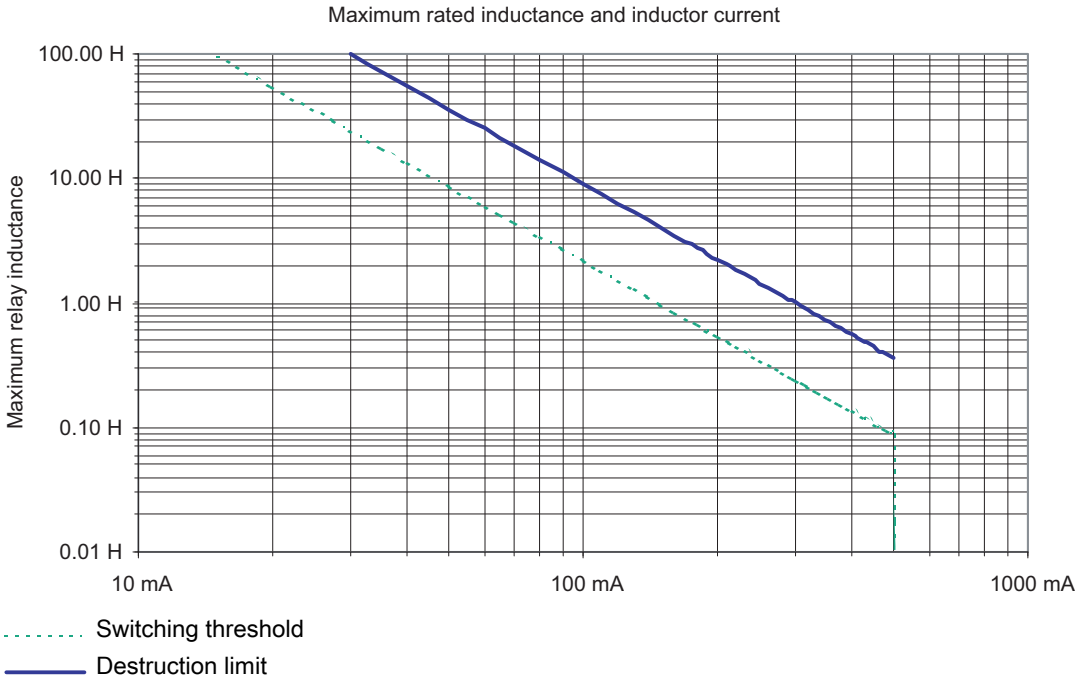


Figure A-7 Graph 1 Maximum Rated Inductance and Inductor Current

Maximum Inductive Switching Rate

Once you have determined that your inductive load can be switched by the FM 352-5, you must also verify that you can switch it at the maximum rate that you require. Energy must be absorbed by the FM 352-5 output each time that the inductor is switched off. For this reason, there is a maximum thermal limit for the rate that an inductive load can be switched. Refer to Graph 2 for this limit.

Reading Graph 2

To determine the maximum rate that the FM 352-5 will switch the load, refer to the following figure. Follow the *L/R* = 2 ms line horizontally to the 100 mA limit line. The thermal maximum switching rate of 50 Hz is the cross point for *L/R* = 2 ms and *I* = 100 mA. If a higher switching rate is required, then commutation diodes will be required. There is no limitation of the FM 352-5 as to switching rate if commutation diodes are used.

Graph 2 is valid for the FM 352-5 switching inductive loads without commutating diodes, all I/O loaded to the rated maximum at 60 °C. The diagram shows L/R in milliseconds depending on the maximum switching rate.

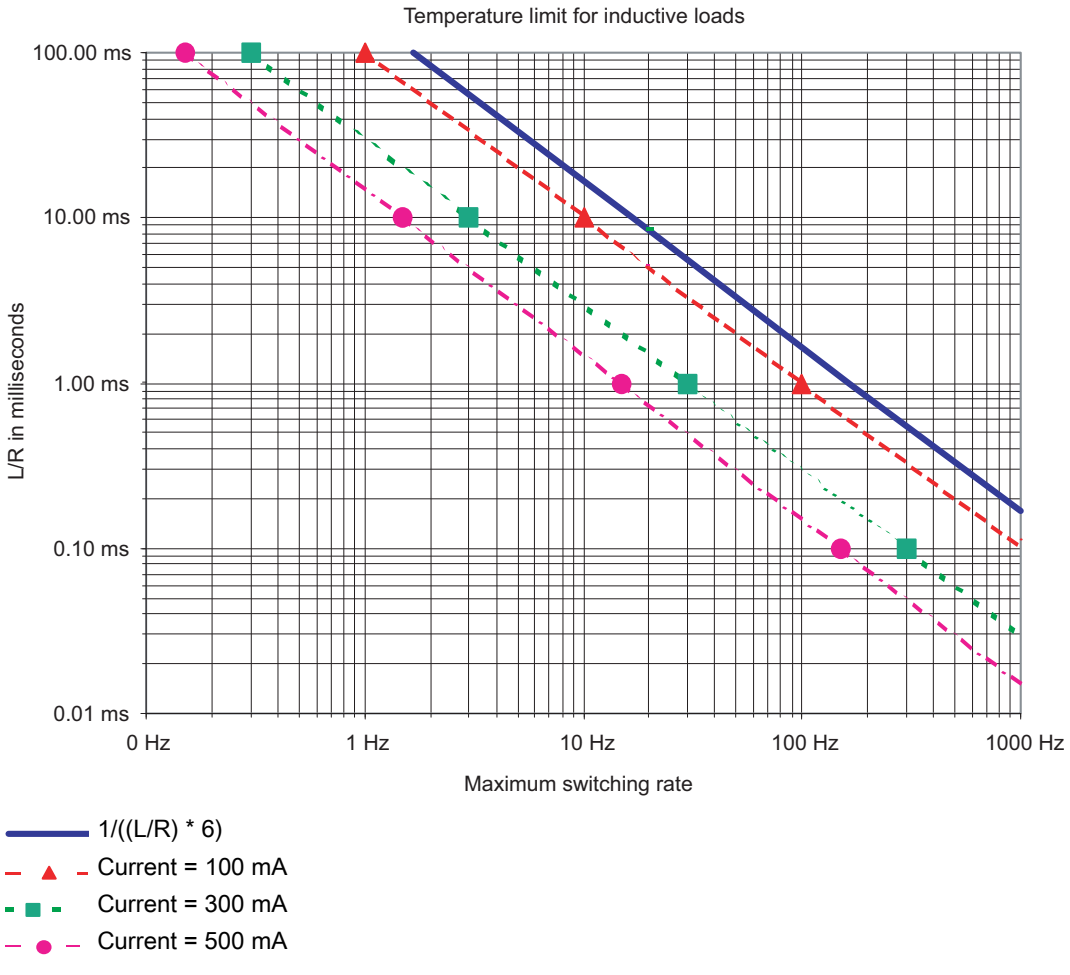


Figure A-8 Graph 2 Thermal for Inductive Load

Application Notes and Assumptions

The following information is a list of application notes and assumptions that pertain to the FM 352-5 module.

- *3*L/R* is the time required to charge the inductance to 95% by V_{in}. It is assumed to be the minimum on or off time for the relay to open or close.
- *1/((L/R)*6)* is assumed the theoretical maximum switching frequency for the relay. (It will probably be lower).
- The relay duty cycle must not be greater than 50% at the maximum switching frequency.
- If the thermal switching limit of an output on the FM 352-5 is exceeded, then reliability may be reduced unless the maximum ambient temperature is below 60° C or the I/O loading is less than maximum.
- The FM 352-5 will not be damaged by brief current or thermal overloads, but will be damaged if an inductive load exceeds the destruction limit. The single pulse avalanche energy rating of the FM 352-5 output is 55 mJ maximum.
- The FM 352-5 provides clamping for inductive reset at 45 V typical, 40 V minimum, 55 V maximum. The turn off time of the inductor is affected by the reset voltage. When the turn off time is an appreciable part of the cycle time the effects of this variability should be checked.
- The FM 352-5's inductive switching limits are the same as the resistive limits if commutation diodes are used.

Commutation Diodes

If the relay inductance and current is beyond the power handling capability of the FM 352-5, a silicon or Schottky diode may be placed across it to absorb the inductive kick. The current capability of the diode must be at least as great as the operating current of the relay, and the reverse voltage must be greater than the maximum relay supply voltage. The diode must be capable of dissipating the energy in the inductor at the maximum programmed cycle rate of the FM 352-5 output.

Diode commutation of a relay is relatively slow. If faster commutation is required, a zener diode may be placed in opposition to the silicon or schottky commutation diode. Higher commutation voltage will reduce reset time, but the commutation voltage must always be less than the minimum FM 352-5 commutation voltage of 40V. The diode pair must be capable of dissipating the energy in the inductor at the maximum programmed cycle rate of the FM 352-5 output.

The following figure shows the use of commutation diodes.

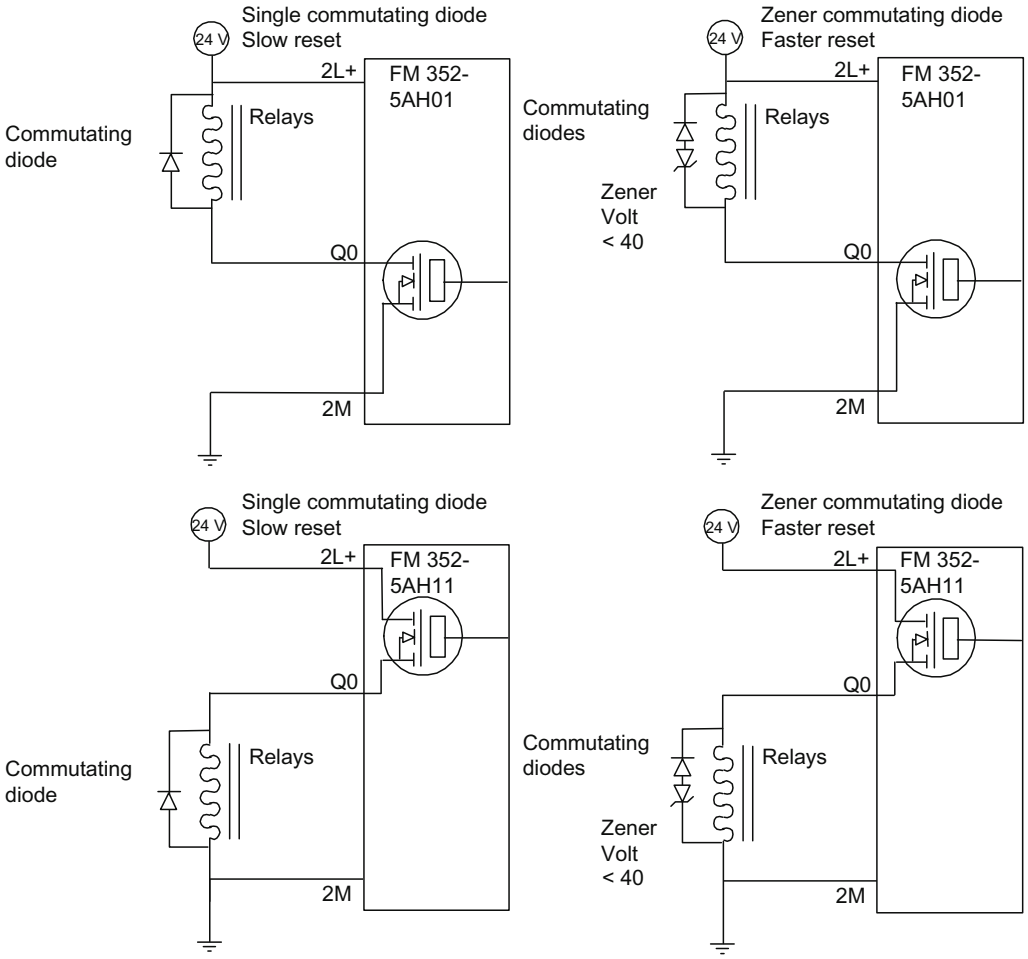


Figure A-9 Application of Commutation Diodes

FPGA Resources Used by Instructions

A total of 1200 logic modules are available on the FPGA processor as resources. Of this total, 436 logic modules are the fixed resources used, or overhead. The following list shows the maximum number of logic modules each operation requires. The actual total may be less after the program has been compiled. To estimate the size of your program, add the fixed resources (436), the encoder selected, and the logic modules for each operation in your program. The compiler provides an exact utilization percentage at compile time

Table A- 1 Resources of FPGA Used by Instructions

Operation	Logic modules
Flip flops, etc.	
BISCALE*	2
CP_GEN*	29
POS*	2
NEG*	2
SR*	1
RS*	1
Counter	
CTD16*	36
CTU16*	31
CTUD16*	47
CTUD32*	99
Timers	
TOF16*	26
TOF32*	55
TON16*	25
TON32*	53
TP16*	26
TP32*	54
Shift register	
SHIFT*	18
SHIFT2*	18
SHIFT4*	18
SHIFT8*	19
SHIFT16*	21
SHIFT32*	29
SHR_I*	36
SHR_I_U	36
SHR_DI*	88
SHR_DI_U	87
ROL_DW*	81
ROL_DW_U	80
SHL_DW*	81

Operation	Logic modules
SHL_DW_U	80
SHL_W*	35
SHL_W_U	34
SHR_DW*	81
SHR_DW_U	81
SHR_W*	34
SHR_W_U	34
FIFO32*	19
FIFO16*	19
LIFO32*	21
LIFO16*	21
BitShift_DW*	17
BitShift_W*	19
Arithmetic operations	
FMABS16	18
FMABS32	37
FMAdd16	9
FMAdd32	17
FMDIV16*	86
FMDIV32*	153
FMMUL16*	62
FMMUL32*	118
BITSUM*	21
BITSUM_U	21
ENCODE*	19
ENCODE_U	19
Data transmission	
MOVE (latched)	17
MOVE_U (unlatched)	0
DatSel16	8
DatSel32	16
WordPack*	17
WordPack_U	0
WordCast*	17
WordCast_U	0
BitPick_DW*	10
BitPick_DW_U	10
BitPick_W*	5
BitPick_W_U	5
BitCast_DW*	17
BitCast_DW_U	0
BitCast_W*	9

Operation	Logic modules
BitCast_W_U	0
BitPack_DW*	17
BitPack_DW_U	0
BitPack_W*	9
BitPack_W_U	0
BitInsert32*	33
BitInsert32_U	32
BitInsert16*	17
BitInsert16_U	16
Encoder selected	
Encoder 16 bit	64
Encoder 32 bit	117
SSI master 13 bit	61
SSI master 25 bit	100
SSI listen 16 bit	77
SSI listen 32 bit	122
None	0
Comparator	
CMP16_EQ	6
CMP16_GE	8
CMP16_GT	8
CMP16_LE	8
CMP16_LT	8
CMP16_NE	6
CMP32_EQ	11
CMP32_GE	25
CMP32_GT	25
CMP32_LE	25
CMP32_LT	25
CMP32_NE	11
Type conversion	
I_DI*	9
I_DI_U	0
INV_DI*	17
INV_DI_U	0
INV_I*	9
INV_I_U	0
Logical operations	
AND	1
OR	1
XOR	1
Word logic operations	

Operation	Logic modules
WAND_W*	9
WAND_W_U	8
WAND_DW*	17
WOR_DW_U	16
WOR_W*	9
WOR_W_U	8
WOR_DW*	17
WOR_DW_U	16
WXOR_DW*	17
WXOR_DW_U	16
WXOR_W*	9
WXOR_W_U	8
Miscellaneous	
FREQ32*	71
FREQ16*	51
PERIOD32*	43
PERIOD16*	23
== (INT)	6
>= (INT)	8
>= (INT)	8
<= (INT)	8
< (INT)	8
<> (INT)	6
== (DINT)	11
>= (DINT)	25
>= (DINT)	25
<= (DINT)	25
< (DINT)	25
<> (DINT)	11
* Operation has memory and uses one clock phase. -U unlatched, not retentive	

FPGA Resources Used by Hardware Support of Diagnostics

The parameters listed under Advanced parameters determine whether the FM 352-5 compiler will include the associated diagnostic hardware elements in the compiled FPGA image. If the associated diagnostic hardware element is enabled, then the parameters listed under Module diagnostics enable, Output diagnostics enable, and Hardware interrupts enable can be used to individually enable or disable the corresponding event to interrupt the S7 CPU. If the associated diagnostic hardware element is not enabled, then the dynamic parameters have no effect.

The default for the hardware support of each of the advanced parameters is "enabled" (box checked). If your application does not require a particular diagnostic or Hardware interrupt, then you may disable the corresponding advanced parameter, which generally makes more logic modules available for the application program. Since the FM 352-5 compiler optimizes the logic modules used in the FPGA image by packing unrelated functions into logic modules, removing the diagnostic function may not lower the logic module count but it does make space available for packing additional program logic into your application FB.

It is recommended that you keep the advanced parameters enabled even if you do not use a particular diagnostic, as long as your application fits in the FPGA. This allows field service personnel to enable diagnostics with an SFC to troubleshoot a problem without requiring the FM 352-5 configuration software to be installed on the target system.

The following table shows the number of logic modules associated with each advanced parameter:

Table A- 2 Resources of FPGA Used By Advanced Parameters

Parameters	Logic modules
Module diagnostics hardware support	
Missing auxiliary supply voltage (1L)	3
Missing input/output supply voltage (2L)	11
Encoder sensor supply fault	12
Missing encoder supply voltage (3L)	11
SSI frame error	34
Differential incremental encoder (RS-422) broken wire	10
Output diagnostics hardware support	
Output overload, Q0. . A7	12 each
Hardware interrupts hardware support	
Hardware interrupt 0 . 7	4 each

A.6 Function block declaration table

Overview

The following table shows an example of a declaration table with descriptions of each of the input, output, and static sections.

Table A- 3 Example Declaration Table for the Application FB (as displayed in STEP 7 V5.1)

Address	Declaration	Name	Type	Comment
Input section: This input is position-specific. The first 15 bits are digital inputs of the FM 352-5. You can specify a list of type BOOL or an array of BOOL (but not both). You can also assign names to the inputs.				
0.0 (cannot be changed)	in	DIn (can be changed)	ARRAY [0..14] (can be changed)	Digital inputs - (0..11 = 24 V) (12..14 = RS-422 differential)
*0.1	in		BOOL (can be changed)	
Input section: Bytes 2 through 15 are position-specific data from the CPU to the FM 352-5 module. Any combination of BOOL, Array of BOOL, BYTE, WORD, INT, or DINT which total 14 bytes, is allowed. You can assign names to the inputs.				
2.0 (cannot be changed)	in	CPU_Out (cannot be changed)	STRUCT	14 bytes from the CPU as inputs to the FM.
+0.0	in	Bits (can be changed)	ARRAY [0..15] (can be changed)	...Some can be Boolean.
*0.1	in		BOOL (can be changed)	
+2.0	in	T1_PV (can be changed)	DINT (can be changed)	...Some can be DINT. (DINT must start at +2, +6, or +10)
+6.0	in	T2_PV (can be changed)	BYTE (can be changed)	...Some can be BYTE (must be mapped to INT by the MOVE operation)
+7.0	in	CmpByte (can be changed)	BYTE (can be changed)	
+8.0	in	C1_PV (can be changed)	INT (can be changed)	...Some can be INT (INT must start at an even byte boundary).
+10.0	in	CP_Period (can be changed)	WORD (can be changed)	...Some can be WORD.
+12.0	in	CMPInt (can be changed)	INT (can be changed)	Total structure length must be 14 bytes.
=14.0 (cannot be changed)	in		END_STRUCT	

Address	Declaration	Name	Type	Comment
Output section: This output is position-specific. The first 8 bits are digital outputs of the FM 352-5. You can specify a list of the type BOOL or an array of BOOL (but not both). You can also assign names to the outputs.				
16.0 (cannot be changed)	out	DOut (can be changed)	ARRAY [0..7] (can be changed)	24 V digital outputs of this cycle.
*0.1	out		BOOL (can be changed)	
Output section: The CPU Inputs are outputs from the FM 352-5 module. This output is position-specific. Any combination of BOOL, array of BOOL, BYTE, WORD, INT, or DINT that totals 14 bytes is allowed. You can assign names to the outputs.				
18.0 (cannot be changed)	out	CPU_In (cannot be changed)	STRUCT	14 bytes assigned as inputs and returned to the CPU.
+0.0	out	Bits (can be changed)	ARRAY [0..15] (can be changed)	...Some can be Boolean.
*0.1	out		BOOL (can be changed)	
+2.0	out	T2_CVasByte (can be changed)	BYTE (can be changed)	...Some can be BYTE.
+3.0	out	C1_CVasByte (can be changed)	BYTE (can be changed)	
+4.0	out	T2_CV (can be changed)	INT (can be changed)	...Some can be INT.
+6.0	out	T1_CV (can be changed)	DINT (can be changed)	...Some can be DINT. (DINT must start at +2, +6, or +10)
+10.0	out	Enc_CV1 (can be changed)	DINT (can be changed)	Total structure length must be 14 bytes.
=14.0 (cannot be changed)	out		END_STRUCT	
	in_out			
Static section: This definition is position-specific. The first 8 bits are interpreted as hardware interrupts (process interrupts that trigger OB40). You can specify a list of the type BOOL or an array of BOOL (but not both). You can also assign names to the elements.				
32.0 (cannot be changed)	stat	Intr (can be changed)	ARRAY [0..7] (can be changed)	Resources for module interrupts. Upper limit fixed. Do not change.
*0.1	stat		BOOL (can be changed)	

Address	Declaration	Name	Type	Comment
Static section: This definition is position-specific. These are module status bits. Do not change.				
34.0 (cannot be changed)	stat	ST (cannot be changed)	STRUCT	Resources for module status bits. Upper limit fixed. Do not change.
+0.0 (cannot be changed)	stat	FIRSTSCAN (cannot be changed)	BOOL (cannot be changed)	First scan after a change from STOP to RUN.
+0.1 (cannot be changed)	stat	M3L (cannot be changed)	BOOL (cannot be changed)	Power supply for 3L is missing.
+0.2 (cannot be changed)	stat	ESSF (cannot be changed)	BOOL (cannot be changed)	Encoder power supply is overloaded.
+0.3 (cannot be changed)	stat	M2L (cannot be changed)	BOOL (cannot be changed)	Power supply for 2L is missing.
+0.4 (cannot be changed)	stat	M1L (cannot be changed)	BOOL (cannot be changed)	Power supply for 1L is missing.
+2.0 (cannot be changed)	stat	OVERLOAD (cannot be changed)	ARRAY [0..7] (cannot be changed)	Output [x] is overloaded.
*0.1 (cannot be changed)	stat		BOOL (cannot be changed)	
=4.0 (cannot be changed)	stat		END_STRUCT	
Static section: This definition is position-specific. The Encoder is a structure that has a fixed number of elements. The names cannot be changed, but the size of Cur_Val and Load_Val must be set to INT or DINT according to which size of encoder is configured.				
38.0 *	stat	Encoder*	STRUCT	Encoder structure. Do not change.
+0.0 *	stat	Direction *	BOOL *	Status: Direction 0 = counting up, 1 = counting down
+0.1 *	stat *	Home *	BOOL *	Status: 1= encoder is at home position.
+0.2 *	stat	Homed *	BOOL *	Status: 1 = Home was adopted since power up
+0.3 *	stat	Overflow *	BOOL *	Status: 1= overflow (displayed for 1 scan cycle)
+0.4 *	stat	Underflow *	BOOL *	Status: 1= Underflow (displayed for 1 cycle)
+0.5 *	stat *	SSIFrame *	BOOL *	Status: SSI frame error or power loss
+0.6 *	stat	SSIDataReady *	BOOL *	Status: 0 = SSI encoder has not yet shifted valid data, 1 = data available
+0.7 *	stat	Open_Wire *	BOOL *	Status: 1 = Encoder has open wire
+1.0 *	stat	Hold *	BOOL *	Hold software input for incremental encoder

Address	Declaration	Name	Type	Comment
+1.1 *	stat	Reset *	BOOL *	Software inputs Incremental encoder reset
+1.2 *	stat	Load *	BOOL *	Load software input for incremental encoder
+2.0 *	stat	Cur_Val *	DINT (can be changed)	Current value for incremental encoder: DINT for 32-bit encoder, INT for 16-bit encoder
+6.0 *	stat	Load_Val *	DINT (can be changed)	Load value for the encoder: DINT or INT
=10.0 *	stat		END_STRUCT	
* If an encoder structure is used, it cannot be changed. If it is not used, it can be deleted.				
Static section: These definitions are not position-specific. The FM 352-5 module recognizes the multiple-instance FB from the type ("CTU16", "TP32", etc.). The FBs are from the library of the FM 352-5. You can assign names to the FBs. The types of the FB pin names (IN, OUT, etc.) must be specified. This is required for the connectors.				
48.0 (can be changed)	stat	UCtr1 (can be changed)	"CTU16" (can be changed)	The 16-bit up counter is a multiple instance of FB121 from the library FM 352-5.
60.0 (can be changed)	stat	DCtr1 (can be changed)	"CTD16" (can be changed)	16-bit down counter (FB122)
72.0 (can be changed)	stat	UDCtr1 (can be changed)	"CTUD16" (can be changed)	16-bit up/down counter (FB123)
84.0 (can be changed)	stat	UDCtr2 (can be changed)	"CTUD32" (can be changed)	32-bit up/down counter (FB120)
102.0 (can be changed)	stat	TmrP1 (can be changed)	"TP32" (can be changed)	32-bit timer (FB113)
120.0 (can be changed)	stat	TmrOn1 (can be changed)	"TON32" (can be changed)	32-bit timer (FB114)
138.0 (can be changed)	stat	TmrOf1 (can be changed)	"TOF32" (can be changed)	32-bit timer (FB115)
156.0 (can be changed)	stat	TmrP2 (can be changed)	"TP16" (can be changed)	16-bit timer (FB116)
170.0 (can be changed)	stat	TmrOn2 (can be changed)	"TON16" (can be changed)	16-bit timer (FB117)
184.0 (can be changed)	stat	TmrOf2 (can be changed)	"TOF16" (can be changed)	16-bit timer (FB118)
198.0 (can be changed)	stat	SReg1 (can be changed)	"SHIFT" (can be changed)	Shift registers (FB124 to FB127)
718.0 (can be changed)	stat	SReg2 (can be changed)	"SHIFT2" (can be changed)	
1238.0 (can be changed)	stat	BiS (can be changed)	"BiScale" (can be changed)	2:1 binary scaler (FB112)
1244.0 (can be changed)	stat	Clk50 (can be changed)	"CP_Gen" (can be changed)	Pulse generator (FB119)

Address	Declaration	Name	Type	Comment
Static section: This definition is not position-specific. You can change the names inside the structure except for „FF„. You can use any combination of BOOL or Array of BOOL.				
1254.0 (can be changed)	stat	FF (cannot be changed)	STRUCT	Resources for R/S and S/R. Each element must be BOOL or an array of BOOL.
+0.0 (can be changed)	stat	FirstFF (can be changed)	BOOL (can be changed)	The number of elements can be increased as needed.
+0.1 (can be changed)	stat	SecondFF (can be changed)	BOOL (can be changed)	The names of elements can be freely assigned.
+0.2 (can be changed)	stat	ThirdFF (can be changed)	BOOL (can be changed)	
+2.0 (can be changed)	stat	MoreFFs (can be changed)	ARRAY [0..15] (can be changed)	
*0.1	stat		BOOL (can be changed)	
=4.0 (can be changed)	stat		END_STRUCT	
Static section: This definition is not position-specific. You can change the names inside the structure except for "Edge". You can use any combination of BOOL or Array of BOOL.				
1258.0 (can be changed)	stat	Edge (cannot be changed)	STRUCT	Resources for edge detection. Each element must be BOOL or an array of BOOL.
+0.0 (can be changed)	stat	FirstEdge (can be changed)	BOOL (can be changed)	The number of elements can be increased as needed.
+0.1 (can be changed)	stat	SecondEdge (can be changed)	BOOL (can be changed)	The names of elements can be freely assigned.
+0.2 (can be changed)	stat	ThirdEdge (can be changed)	BOOL (can be changed)	
+2.0 (can be changed)	stat	Edge4to10 (can be changed)	ARRAY [4..10] (can be changed)	
*0.1	stat		BOOL (can be changed)	
+4.0 (can be changed)	stat	LastEdge (can be changed)	BOOL (can be changed)	
=6.0 (can be changed)	stat		END_STRUCT	

Address	Declaration	Name	Type	Comment
Static section: This definition is not position-specific. You can change the names inside the structure except for "Conn". You can use any combination of BOOL, INT, DINT or Array of BOOL, INT, or DINT.				
1264.0 (can be changed)	stat	Conn (cannot be changed)	STRUCT	Resources for connectors.
+0.0 (can be changed)	stat	XCon (can be changed)	BOOL (can be changed)	Elements can be BOOL.
+2.0 (can be changed)	stat	arrXCon (can be changed)	ARRAY [0..31] (can be changed)	Elements can be an array of BOOL.
*0.1	stat		BOOL (can be changed)	
+6.0 (can be changed)	stat	ICon (can be changed)	INT (can be changed)	Elements can be INT.
+8.0 (can be changed)	stat	arrICon (can be changed)	ARRAY [0..3] (can be changed)	Elements can be an array of INT.
*2.0	stat		INT (can be changed)	
+16.0 (can be changed)	stat	DICon (can be changed)	DINT (can be changed)	Elements can be DINT.
+20.0 (can be changed)	stat	arrDICon (can be changed)	ARRAY [0..3] (can be changed)	Elements can be an array of DINT.
*4.0	stat		DINT (can be changed)	
=36.0 (can be changed)	stat		END_STRUCT	
Temp section: This definition is position-specific. The name cannot be changed.				
0.0 (cannot be changed)	temp	Dummy (cannot be changed)	BOOL (cannot be changed)	Is used where an output coil is required by STEP 7 to execute the operation but is not needed by your program.

A.7 Valid operations for FM 352-5 module

LAD operations in STEP 7 program elements

The following table lists the LAD operations that are valid for the FM 352-5 module. Operations in italics are function blocks that are available in the FM 352-5 library after you install the FM 352-5 configuration software. These FBs are found in the STEP 7 Program Elements catalog in the "Libraries" container.

Table A- 4 Valid operations for FM 352-5

Operation	Tanks/containers	Description
--| |--	Bit logic	NO contact
--|/|--	Bit logic	NC contact
--|NOT|--	Bit logic	Invert power flow
--( )	Bit logic	Coil
--(#)--	Bit logic	Midline output
RS	Bit logic	Reset/set flip-flop
SR	Bit logic	Set/reset flip-flop
--(N)--	Bit logic	Detect negative RLO edge
--(P)--	Bit logic	Detect positive RLO edge
NEG	Bit logic	Negative edge detection
POS	Bit logic	Positive edge detection
CMP	Comparator	Comparison operations, integer and double integer values (16 bits and 32 bits) only; real values are not supported.
I_DI	Converter	Convert integer (16 bit) to double integer (32 bit)
MOVE	MOVE	Assign a value
INV_I	Converter	Generate one's compliment for 16-bit double integer
INV_DI	Converter	Generate one's compliment for 32-bit double integer
WAND_W	Word logic operation	AND word operation
WOR_W	Word logic operation	OR word operation
WXOR_W	Word logic operation	Exclusive OR word operation
WAND_DW	Word logic operation	AND double word operation
WOR_DW	Word logic operation	OR double word operation
WXOR_DW	Word logic operation	Exclusive OR double word operation
SHR_I	Shift/rotate operation	Shift right 16-bit integer operation
SHR_DI	Shift/rotate operation	Shift right 32-bit integer operation
SHL_W	Shift/rotate operation	Shift left word operation
SHR_W	Shift/rotate operation	Shift right word operation
SHL_DW	Shift/rotate operation	Shift left double word operation
SHR_DW	Shift/rotate operation	Shift right double word operation
ROL_DW	Shift/rotate operation	Rotate left double word operation
ROR_DW	Shift/rotate operation	Rotate right double word operation

Operation	Tanks/containers	Description
BiScale	FM 352-5 library	Binary scaler
TP32	FM 352-5 library	32-bit pulse
TON32	FM 352-5 library	32-bit on delay timer
TOF32	FM 352-5 library	32-bit off delay timer
TP16	FM 352-5 library	16-bit pulse
TON16	FM 352-5 library	16-bit on delay timer
TOF16	FM 352-5 library	16-bit off delay timer
CP_Gen	FM 352-5 library	Clock pulse generator
CTUD32	FM 352-5 library	32-bit up/down counter
CTU16	FM 352-5 library	16-bit up counter
CTD16	FM 352-5 library	16-bit down counter
CTUD16	FM 352-5 library	16-bit up/down counter
SHIFT	FM 352-5 library	Bit shift register, 1 bit; maximum length = 4096
SHIFT2	FM 352-5 library	Bit shift register, 2 bits; maximum length = 2048
SHIFT4	FM 352-5 library	Bit shift register, 4 bits; maximum length = 1024
SHIFT8	FM 352-5 library	Bit shift register, 8 bits; maximum length = 512
SHIFT16	FM 352-5 library	INT shift register; maximum length = 256
SHIFT32	FM 352-5 library	DINT shift register; maximum length = 256
FMABS32	FM 352-5 library	Absolute value, 32 bits
FMABS16	FM 352-5 library	Absolute value, 16 bits
DatSel32	FM 352-5 library	Data selector, 32 bits
DatSet16	FM 352-5 library	Data selector, 16 bits
FMAdd32	FM 352-5 library	Add, 32 bits
FMAdd16	FM 352-5 library	Add, 16 bits
FMSub32	FM 352-5 library	Subtract, 32 bits
FMSub16	FM 352-5 library	Subtract, 16 bits
FMMul32	FM 352-5 library	Multiply, 32 bits
FMMul16	FM 352-5 library	Multiply, 16 bits
FMDiv32	FM 352-5 library	Divide, 32 bits
FMDiv16	FM 352-5 library	Divide, 16 bits
ENCODE	FM 352-5 library	Locates most significant bit set in a DWORD
BITSUM	FM 352-5 library	Counts set bits in a DWORD
BitPack_W	FM 352-5 library	Packs 16 digital bits into a WORD
BitPack_DW	FM 352-5 library	Packs 32 digital bits into a DWORD
BitCast_W	FM 352-5 library	Converts a WORD to 16 digital bits
BitCast_DW	FM 352-5 library	Converts a DWORD to 32 digital bits
BitPick_W	FM 352-5 library	Selects a bit from a WORD
BitPick_DW	FM 352-5 library	Selects a bit from a DWORD
BitInsert16	FM 352-5 library	Inserts a bit into an INT (16 bits)
BitInsert32	FM 352-5 library	Inserts a bit into a DINT (32 bits)
BitShift_W	FM 352-5 library	Bit shift register, length 16 bits
BitShift_DW	FM 352-5 library	Bit shift register, length 32 bits

Operation	Tanks/containers	Description
WordPack	FM 352-5 library	Concatenates 2 WORDs into 1 DWORD
WordCast	FM 352-5 library	Converts 1 DWORD into 2 WORDs
PERIOD16	FM 352-5 library	Period measurement, 16 bits
PERIOD32	FM 352-5 library	Period measurement, 32 bits
FREQ16	FM 352-5 library	Frequency measurement, 16 bits
FREQ32	FM 352-5 library	Frequency measurement, 32 bits
FIFO16	FM 352-5 library	Delete first value, 16 bits
FIFO32	FM 352-5 library	Delete first value, 32 bits
LIFO16	FM 352-5 library	Delete last value, 16 bits
LIFO32	FM 352-5 library	Delete last value, 32 bits

FBD Instructions from STEP 7 Program Elements

The following table lists the FBD operations that are valid for the FM 352-5 module. Operations in italics are function blocks that are available in the FM 352-5 library after you install the FM 352-5 configuration software. These FBs are found in the STEP 7 Program Elements catalog in the "Libraries" container.

Table A- 5 FBD Instructions for FM 352-5

Operation	Tanks/containers	Description
>= 1	Bit logic	Or gate
&	Bit logic	AND operation
XOR	Bit logic	Exclusive OR
--|	Bit logic	Binary input
-o|	Bit logic	Negation
--(=)	Bit logic	Assign
--(#)--	Bit logic	Midline output
RS	Bit logic	Reset/set flip-flop
SR	Bit logic	Set/reset flip-flop
--(N)--	Bit logic	Detect negative RLO edge
--(P)--	Bit logic	Detect positive RLO edge
NEG	Bit logic	Negative edge detection
POS	Bit logic	Positive edge detection
CMP	Comparator	Comparison operations, integer and double integer values only; real values are not supported.
I_DI	Converter	Convert integer (16 bit) to double integer (32 bit)
MOVE	MOVE	Assign a value
INV_I	Converter	Generate one's compliment for 16-bit double integer
INV_DI	Converter	Generate one's compliment for 32-bit double integer
WAND_W	Word logic operations	AND word operation
WOR_W	Word logic operations	OR word operation
WXOR_W	Word logic operations	Exclusive OR word operation
WAND_DW	Word logic operations	AND double word operation
WOR_DW	Word logic operations	OR double word operation
WXOR_DW	Word logic operations	Exclusive OR double word operation
SHR_I	Shift/rotate operations	Shift right 16-bit integer operation
SHR_DI	Shift/rotate operations	Shift right 32-bit integer operation
SHL_W	Shift/rotate operations	Shift left word operation
SHR_W	Shift/rotate operations	Shift right word operation
SHL_DW	Shift/rotate operations	Shift left double word operation
SHR_DW	Shift/rotate operations	Shift right double word operation
ROL_DW	Shift/rotate operations	Rotate left double word operation
ROR_DW	Shift/rotate operations	Rotate right double word operation
BiScale	FM 352-5 library	Binary scaler
TP32	FM 352-5 library	32-bit pulse

Operation	Tanks/containers	Description
TON32	FM 352-5 library	32-bit on delay timer
TOF32	FM 352-5 library	32-bit off delay timer
TP16	FM 352-5 library	16-bit pulse
TON16	FM 352-5 library	16-bit on delay timer
TOF16	FM 352-5 library	16-bit off delay timer
CP_Gen	FM 352-5 library	Clock pulse generator
CTUD32	FM 352-5 library	32-bit up/down counter
CTU16	FM 352-5 library	16-bit up counter
CTD16	FM 352-5 library	16-bit down counter
CTUD16	FM 352-5 library	16-bit up/down counter
SHIFT	FM 352-5 library	Bit shift register, 1 bit; maximum length = 4096
SHIFT2	FM 352-5 library	Bit shift register, 2 bits; maximum length = 2048
SHIFT4	FM 352-5 library	Bit shift register, 4 bits; maximum length = 1024
SHIFT8	FM 352-5 library	Bit shift register, 8 bits; maximum length = 512
SHIFT16	FM 352-5 library	INT shift register; maximum length = 256
SHIFT32	FM 352-5 library	DINT shift register; maximum length = 256
FMABS32	FM 352-5 library	Absolute value, 32 bits
FMABS16	FM 352-5 library	Absolute value, 16 bits
DatSel32	FM 352-5 library	Data selector, 32 bits
DatSet16	FM 352-5 library	Data selector, 16 bits
FMAdd32	FM 352-5 library	Add, 32 bits
FMAdd16	FM 352-5 library	Add, 16 bits
FMSub32	FM 352-5 library	Subtract, 32 bits
FMSub16	FM 352-5 library	Subtract, 16 bits
FMMul32	FM 352-5 library	Multiply, 32 bits
FMMul16	FM 352-5 library	Multiply, 16 bits
FMDiv32	FM 352-5 library	Divide, 32 bits
FMDiv16	FM 352-5 library	Divide, 16 bits
ENCODE	FM 352-5 library	Locates most significant bit set in a DWORD
BITSUM	FM 352-5 library	Counts set bits in a DWORD
BitPack_W	FM 352-5 library	Packs 16 digital bits into a WORD
BitPack_DW	FM 352-5 library	Packs 32 digital bits into a DWORD
BitCast_W	FM 352-5 library	Converts a WORD to 16 digital bits
BitCast_DW	FM 352-5 library	Converts a DWORD to 32 digital bits
BitPick_W	FM 352-5 library	Selects a bit from a WORD
BitPick_DW	FM 352-5 library	Selects a bit from a DWORD
BitInsert16	FM 352-5 library	Inserts a bit into an INT (16 bits)
BitInsert32	FM 352-5 library	Inserts a bit into a DINT (32 bits)
BitShift_W	FM 352-5 library	Bit shift register, length 16 bits
BitShift_DW	FM 352-5 library	Bit shift register, length 32 bits
WordPack	FM 352-5 library	Concatenates 2 WORDs into 1 DWORD
WordCast	FM 352-5 library	Converts 1 DWORD into 2 WORDs

Operation	Tanks/containers	Description
PERIOD16	FM 352-5 library	Period measurement, 16 bits
PERIOD32	FM 352-5 library	Period measurement, 32 bits
FREQ16	FM 352-5 library	Frequency measurement, 16 bits
FREQ32	FM 352-5 library	Frequency measurement, 32 bits
FIFO16	FM 352-5 library	Delete first value, 16 bits
FIFO32	FM 352-5 library	Delete first value, 32 bits
LIFO16	FM 352-5 library	Delete last value, 16 bits
LIFO32	FM 352-5 library	Delete last value, 32 bits

External Protection Circuit for FM 352-5 Boolean Processor

B

The SIMATIC S7 FM 352-5 module is available in two different versions:

- FM 352-5AH1x-0AE0 has sourcing outputs
- FM 352-5AH0x-0AE0 has sinking outputs

The information contain in SIMATIC S7-300 CPU 31xC and CPU 31x Operating Instructions: Installation (http://support.automation.siemens.com/WW/view/en/13008499), "Lightning and Overvoltage Protection" appendix applies to both modules. However, special wiring is required for the outputs.

In order to subject the modules to surges in conformity with IEC 61000-4-5, external protective circuitry is required.

Ordering data

The components required for wiring the 24 V power supply and 24 V outputs can be obtained from the following sources:

- **Surge arrester BLITZDUCTOR VT**

  BTV AD 24

  Item No. 918402

  DEHN + SÖHNE GmbH + Co. KG.

  P.O. Box 16 40

  D-92306 Neumarkt, Germany

- **Transil Diode**

  P6KE36A 600W

  (e.g. STMicroelectronics, ON Semiconductor, Motorola)

Wiring

The following connection diagrams show how the components are to be connected in accordance with the specifications.

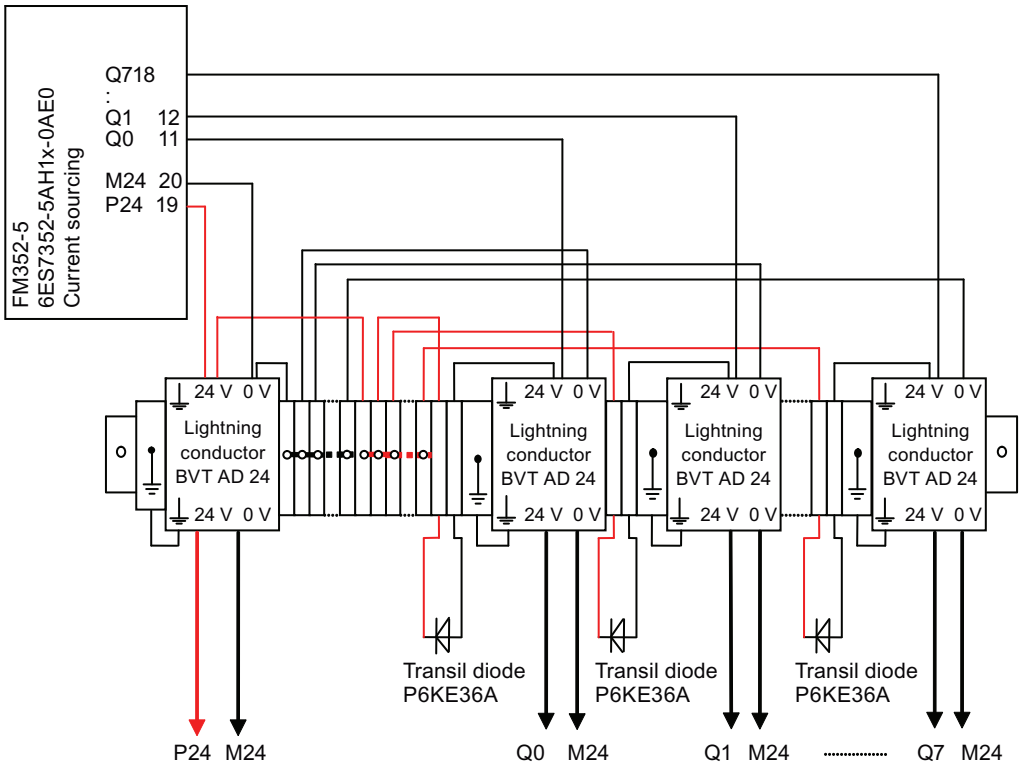


Figure B-1 Connection diagram for the FM 352-5AH1x-0AE0 (sourcing outputs)

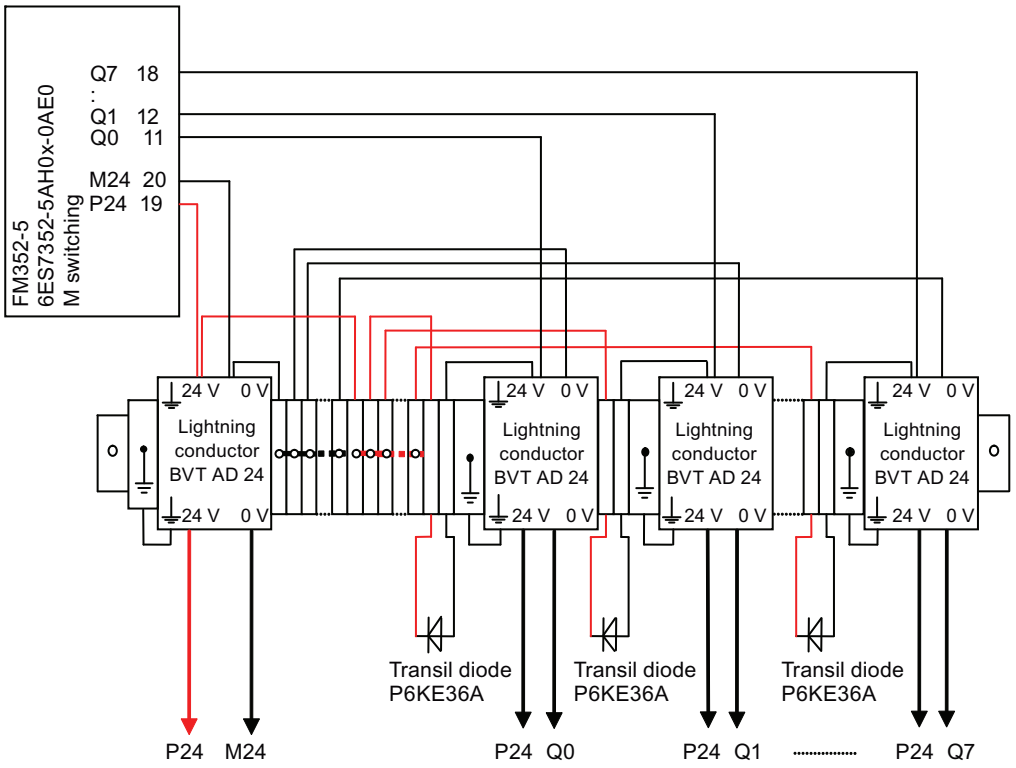


Figure B-2 Connection diagram for the FM 352-5AH0x-0AE0 (sinking outputs)

Parts lists

C

Parts included with the FM 352-5

The following parts are included with the FM 352-5 module:

Table C- 1 Parts for the FM 352-5 module

Part	Description	Order number
P-bus connection expansion bus	To connect FM module on S7 rail to adjacent module	6ES7390-0AA00-0AA0
2-pin connector	For 24 VDC module power supply	—
Label, for 40-pin connector	To identify input and output signals	6ES7392-2XX10-0AA0
Front panel, I/O terminal connector	To cover wiring terminals	—
Front panel, 24 V power supply	To cover external power connector	—

Accessory Components for the FM 352-5

The following accessories are required to operate the FM 352-5 module:

Table C- 2 Accessory Components for the FM 352-5

Part	Description	Order number
40-pin front connector	For input and output signals to the module	Screw-in contacts: 6ES7392-1AM00-0AA0 Spring-loaded contacts: 6ES7392-1BM01-0AA0
SIMATIC Micro Memory Card [1)]	For non-volatile program and configuration data storage; required by the module for program execution.	You can use SIMATIC Micro Memory Card with 128 KB, 512 KB, and 2 MB. The specified order numbers represent the state as of January 2011. 128 KB: 6ES7953-8LG20-0AA0 512 KB: 6ES7953-8LJ20-0AA0 2 MB: 6ES7953-8LL20-0AA0
[1)] For FM 352-5 modules delivered prior to 2008, there are restrictions as to the use of the most recent SIMATIC Micro Memory Cards. You can find more information about this on the Internet at Siemens - Industry Automation and Drive Technologies - Service&Support (http://support.automation.siemens.com/WW/view/en/25393901).		

The following table lists some of the recommended parts that can be used with the FM 352-5 module. The "XXXX" digits at the end of a part number indicate that the catalog offers several different versions of the part, which are identified by different part numbers.

Table C- 3 Recommended Parts for the FM 352-5 Module

Part	Description	Order number
SSI encoders	RS422, TTL	6FX2001-5XXXX
Single-ended encoder	RS422, TTL	6FX2001-2XXXX
Single-ended encoder	Optical HTL incremental encoder	6FX2001-4XXXX
Encoder connector	To connect to encoder: 12-wire connection, pack of 1	6FX2003-0SU12 You can find more information on the Internet at Siemens - Industry Automation and Drive Technologies - Service&Support (http://support.automation.siemens.com/WW/view/en/22103021).
Cable	Suitable for all encoders: 12-wire, 200 m (other lengths are available; refer to your catalog for other part numbers).	6FX8008-1BD21-3AA0
Shield contact element	Fixing bracket with two bolts for attaching shield terminals to the rail	6ES7390-5AA00-0AA0
Terminal element	For one cable with a shield diameter of 3 to 8 mm	6ES7390-5BA00-0AA0
Terminal element	For one cable with a shield diameter of 4 to 13 mm	6ES7390-5CA00-0AA0