SIEMENS

SIMATIC

S7-300 Programmable Controller
Quick Start

Primer

Preface, Contents

Requirements for Working with the Primer	1
Defining and Structuring the Control Task	2
Installing and Wiring Your S7-300	3
Working with the STEP 7 Programming Software	4
Configuring and Assigning Parameters to Your S7-300	5
Programming Logic Blocks with STEP 7	6
Downloading and Testing Your User Program	7

Glossary, Index

C79000-G7076-C500-01

Safety Guidelines

This manual contains notices which you should observe to ensure your own personal safety, as well as to protect the product and connected equipment. These notices are highlighted in the manual by a warning triangle and are marked as follows according to the level of danger:

Danger

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

Warning

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

Caution

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

Note

draws your attention to particularly important information on the product, handling the product, or to a particular part of the documentation.

Qualified Personnel

The device/system may only be set up and operated in conjunction with this manual.

Only **qualified personnel** should be allowed to install and work on this equipment. Qualified persons are defined as persons who are authorized to commission, to ground, and to tag circuits, equipment, and systems in accordance with established safety practices and standards.

Correct Usage

Note the following:

Warning

This device and its components may only be used for the applications described in the catalog or the technical description, and only in connection with devices or components from other manufacturers which have been approved or recommended by Siemens.

This product can only function correctly and safely if it is transported, stored, set up, and installed correctly, and operated and maintained as recommended.

Trademarks

SIMATIC® and SINEC® are registered trademarks of SIEMENS AG.

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

Siemens AG
Automation Group
Industrial Automation Systems
Postfach 4848, D-90327 Nürnberg

Disclaimer of Liability

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

Technical data subject to change.

Preface to the Primer

The Preface gives you an overview of the information contained in the Primer.

What Does This Primer Describe?

Aims of the Primer

With this Primer, we aim to show you how easily you can install the S7-300 programmable logic controller and program it with the relevant STEP 7 programming software.

The Primer supports you with:

- Simple step-by-step instructions and
- An easily understandable sample program that leads you through the entire programming procedure and which can be used as a basis for your own program.

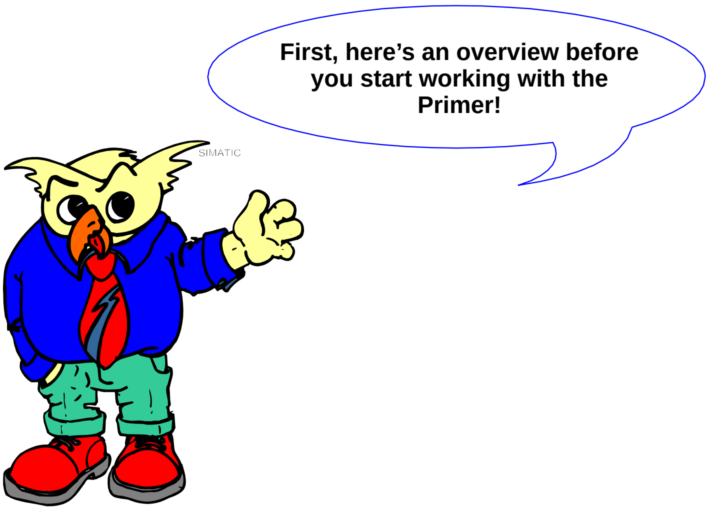
If you have not yet had any experience with our SIMATIC programmable controllers, this Primer will provide you with the information you need as a first-time user.

Structure of the Primer

The form and content of the Primer are such that you can work through it chapter by chapter.

The individual chapters describe what you have to do in chronological order. Our intention here is to provide you with a 'connecting thread' that you can follow when installing and programming your S7-300 programmable controller.

The following page gives you an overview of the contents of the individual chapters to help you find your way around the Primer.



What Does Each Chapter Contain?

The individual chapters of the Primer provide you with the following information and procedures:

Chap. 1

- Chapter 1: **Requirements for Working with the Primer**

  This chapter lists the hardware and software requirements for working with the Primer.

Chap. 2

- Chapter 2: **Defining and Structuring the Control Task**

  Using the TLIGHT sample program, we show how you can design a program for a specific control task and how the program is then executed by your programmable controller.

Chap. 3

- Chapter 3: **Installing and Wiring Your S7-300**

  In this chapter, we explain the minimum components required to make up an S7-300 and how you can install and wire your S7-300 and connect it to a programming device or PC.

Chap. 4

- Chapter 4: **Working with the STEP 7 Programming Software**

  Here, you learn how to use the STEP 7 programming software. In addition, you are also introduced to the STEP 7 objects with which you will work.

Chap. 5

- Chapter 5: **Configuring and Assigning Parameters to Your S7-300**

  This chapter tells you how you can configure and assign parameters to your S7-300 in order to adapt it to your requirements.

Chap. 6

- Chapter 6: **Programming Logic Blocks with STEP 7**

  This chapter contains examples for programming in the STEP 7 languages STL (statement list), LAD (ladder logic), and FBD (function block diagram).

Chap. 7

- Chapter 7: **Downloading and Testing Your User Program**

  Finally, we show you how to proceed when you want to download your user program to the CPU and test it.

Contents

	Preface to the Primer	**iii**
1	**Requirements for Working with the Primer**	**1-1**
	Hardware Requirements	1-2
	Software Requirements	1-3
	Other References for This Chapter	1-4
2	**Defining and Structuring the Control Task**	**2-1**
	How Do You Design a Control Task?	2-2
	How to Define and Structure Your Control Task	2-3
	Other References for This Chapter	2-7
3	**Installing and Wiring Your S7-300**	**3-1**
	What Do You Need to Know Before you Start?	3-2
	How Do You Install Your S7-300?	3-3
	How Do You Wire Your S7-300?	3-5
	How Do You Connect Your Programming Device/PC?	3-6
	Other References for This Chapter	3-7
4	**Working with the STEP 7 Programming Software**	**4-1**
	What is STEP 7?	4-2
	How Do You Work with STEP 7?	4-3
	Which STEP 7 Objects Should You Know?	4-4
	This is How You Create a Project Structure!	4-6
	Other References for This Chapter	4-7
5	**Configuring and Assigning Parameters to Your S7-300**	**5-1**
	What is Configuring and Assigning Parameters?	5-2
	How to Configure and Assign	5-3
	First Perform a Memory Reset of Your CPU!	5-4
	How Do You Configure and Assign Parameters to Your S7-300?	5-5
	What Does a Configuration Table Look Like?	5-6
	Other References for This Chapter	5-7

6 **Programming Logic Blocks with STEP 7** **6-1**

Procedure for Programming Blocks 6-2

How Do You Create the Required Blocks? 6-3

What are STL, LAD, and FBD? 6-4

How Do You Program Blocks in STL? 6-7

How Do You Program Blocks in LAD? 6-10

How Do You Program Blocks in FBD? 6-14

Other References for This Chapter 6-18

7 **Downloading and Testing Your User Program** **7-1**

Basic Procedure for Downloading and Testing 7-2

How Do You Download the User Program to Your S7-300? 7-3

What Do You Have to Know about Testing? 7-4

How Do You Test Your STL Program? 7-6

How Do You Test Your LAD Program? 7-8

How Do You Test Your FBD Program? 7-10

Other References for This Chapter 7-12

Glossary

Index

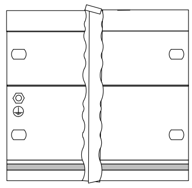
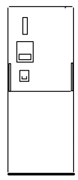
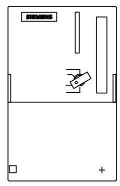
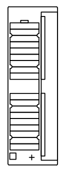
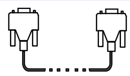
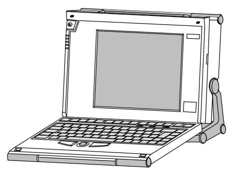
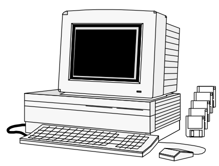
1 Requirements for Working with the Primer

We have made every effort to keep the hardware and software requirements for working with the Primer to a minimum.

Hardware Requirements

What Hardware Do You Need?

For working with the Primer and executing the TLIGHT sample program, you require the following hardware components:

Component	Function	Illustration
Rail	... functions as a mounting rack for an S7-300.	
Power supply module (PS)	... converts the supply voltage (120/230 VAC) to 24 VDC operating voltage for supplying the S7-300.	
CPU (central processing unit) Backup battery (optional)	... executes the user program; supplies the S7-300 backplane bus with 5 V; communicates via the MPI (multipoint interface) with other CPUs, or with a programming device/PC.	
Simulator module (6ES7 374...) with 8 digital inputs and 8 digital outputs	... gives you the option of testing the user program on startup and during operation by simulating sensor signals via switches and displaying signal states at the outputs via LEDs.	
MPI cable	... connects a programming device/PC with a CPU.	
Programming device (PG) with an MPI module and pre-installed STEP 7 software **or**	... configures, assigns parameters, programs, and tests the S7-300 programmable controller.	
Personal computer (PC) with an MPI module or PC/MPI cable with pre-installed STEP 7 software	... configures, assigns parameters, programs, and tests the S7-300 programmable controller.	

Software Requirements

What Software Do You Need?

For working with the Primer and executing the TLIGHT sample program, you require:

- WINDOWS 95 and
- The STEP 7 software package, version 3.

STEP 7 Software Package

STEP 7 contains a SETUP program that carries out the installation automatically.

On-screen prompts lead you step-by-step through the entire installation process.

Please note

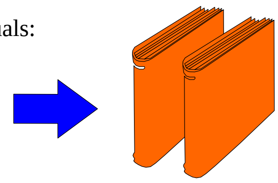
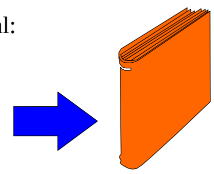
To work with the Primer, we assume you have already installed the STEP 7 software package.

If you have an older version of STEP 7, you must install the new version (see above) of our STEP 7 software.

Other References for This Chapter

Where Can You Find Other References?

If you still have questions and you need additional information regarding hardware and software requirements, our manuals may be of assistance.

You Will Find Information on...	In...
the hardware requirements for working with STEP 7	the following manuals:  S7-300 Programmable Controller, Hardware and Installation and S7-300, M7-300 Programmable Controller, Module Specifications
installing the STEP 7 software	the following manual:  Standard Software for S7 and M7 STEP 7

Defining and Structuring the Control Task 2

We will show you exactly how to implement a simple control system using the TLIGHT sample program.

How Do You Design a Control Task?

Basic Procedure

Using the TLIGHT sample program, we intend to show you step-by-step how to proceed in designing a control task. You can then transfer this procedure to your own applications.

Figure 2-1 shows you an overview of the procedure with the help of a few questions:

Defining and structuring your control task

What is to be controlled?

↓

What subtasks are to be performed?

↓

What safety requirements apply?

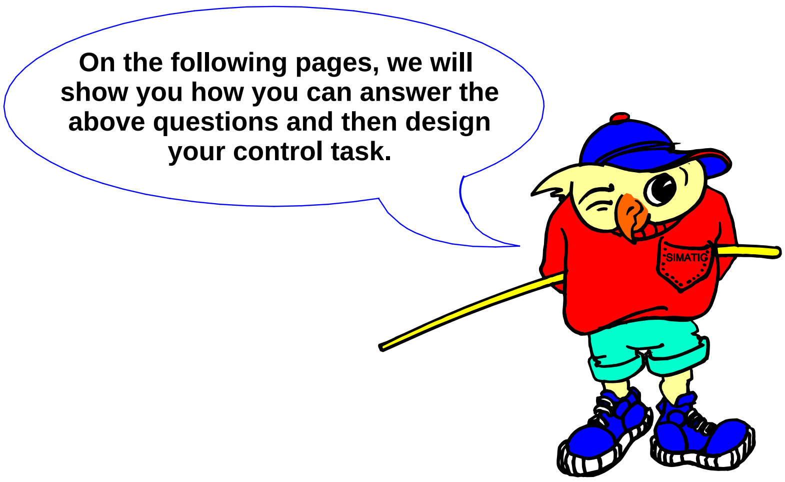
↓

What other requirements apply?

↓

Which addresses do you need?

Figure 2-1 Defining and Structuring a Control Task



How to Define and Structure Your Control Task

What is to be Controlled?

The TLIGHT sample program to be created will control vehicle and pedestrian traffic at a pedestrian crossing, as shown in Figure 2-2.

What Subtasks are to be Performed?

The TLIGHT program is to control both

- The traffic lights for vehicle traffic (traffic lights) and
- The traffic lights for pedestrian traffic (pedestrian lights).

The traffic lights are equipped with the usual red, yellow, and green signals.

The pedestrian lights each have a green and red signal as well as a pushbutton for pedestrians to request green.

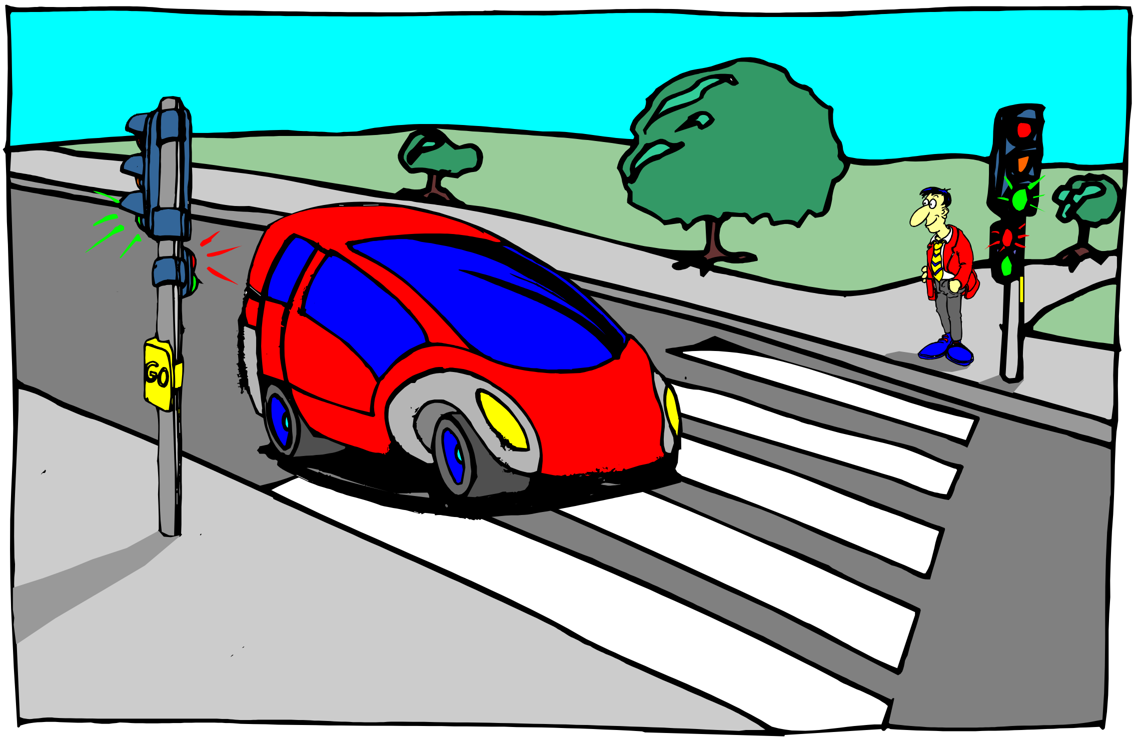


Figure 2-2 The TLIGHT Sample Program is to Control Traffic at a Pedestrian Crossing as Shown

What Safety Requirements Apply?

You must take into account the following safety requirements in order to prevent any hazard to pedestrians or drivers:

- The default setting should be green for the traffic light and red for the pedestrian light, in order to define a safe starting point.
- If the program receives a request to change the pedestrian light to green as a result of the pushbutton being pressed, the traffic light changes from green to red via yellow, as shown in Figure 2-3.

What Other Requirements Apply?

Apart from the safety requirements, you must now define how long the individual signal phases are to last and when they are to start:

- The yellow phase for vehicle traffic is to last 3 seconds.
- The red phase for vehicle traffic is to last 16 seconds and it is to start simultaneously with the green phase for pedestrians.
- The green phase for pedestrians is to last 10 seconds.
- As soon as the green phase for pedestrians has finished, the pedestrian signal is to change to red.
- The red/yellow phase for vehicle traffic is to last 3 seconds.
- The delay for the next green request for pedestrians is to last 1 second.

Figure 2-3 gives you a schematic overview of how the TLIGHT sample program should work:

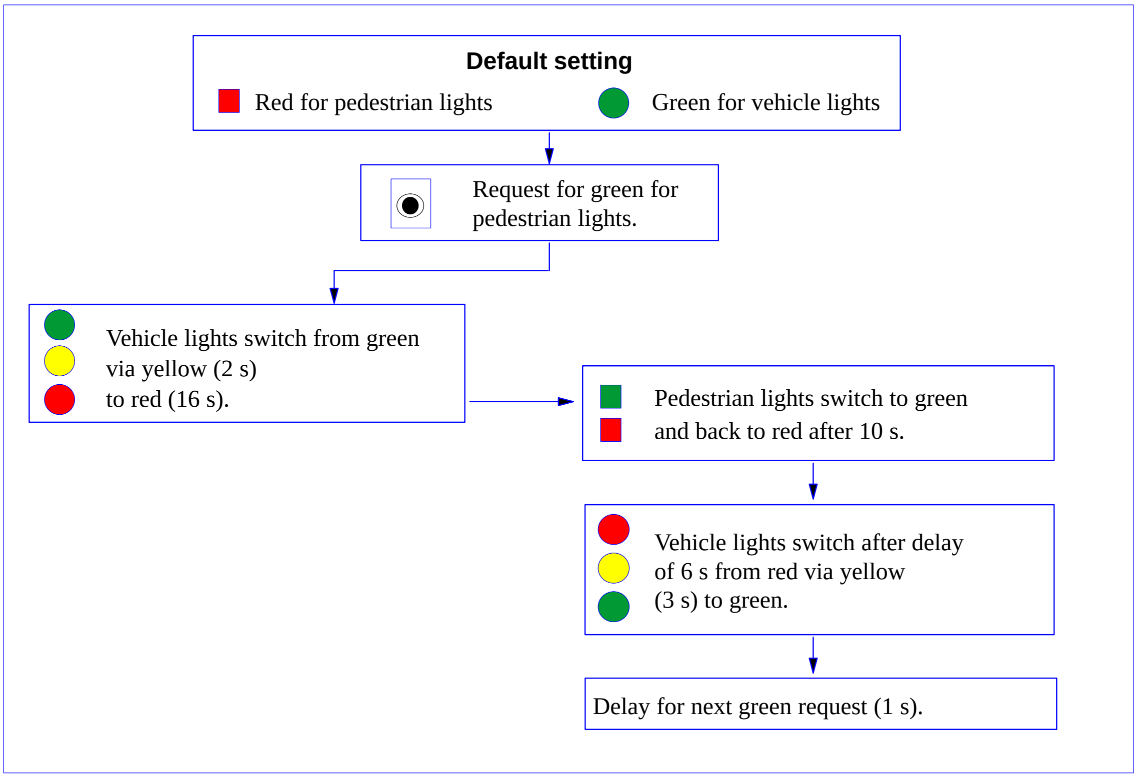


Figure 2-3 Schematic Overview of the Traffic Light Control Sequence

Which Addresses Do You Need?

In order for the TLIGHT sample program to simulate a traffic light control system on the simulator module of your S7-300, you must define the following addresses to which you can also assign a symbolic name:

- 2 inputs (I) for requesting green for the pedestrians on both sides of the street.
- 5 outputs (Q) for controlling the signal indicators of both sets of lights.
- 1 memory bit (M) for switching the signal after a green request from a pedestrian.
- 5 timers (T) to determine the duration of each signal phase. The timers each have the S5Time format.

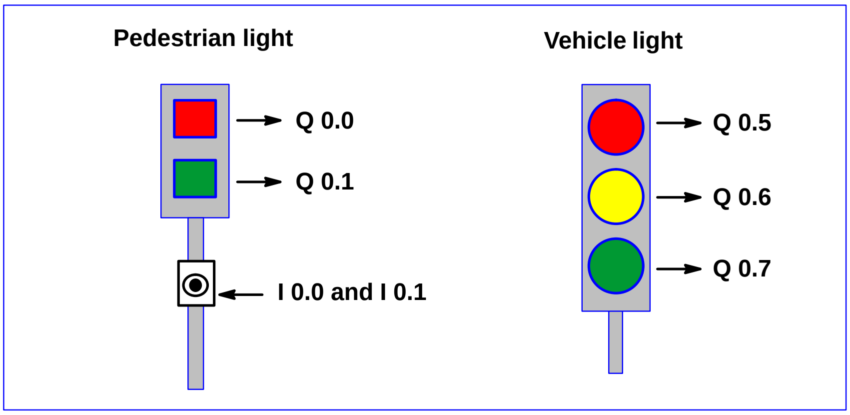


Figure 2-4 Traffic Light Indicators and the Required Inputs and Outputs

Address	Description	Example of Symbolic Names
Q 0.0 Q 0.1 Q 0.5 Q 0.6 Q 0.7	Red for pedestrians Green for pedestrians Red for vehicles Yellow for vehicles Green for vehicles	Ped_Red Ped_Green Veh_Red Veh_Yel Veh_Green
I 0.0 I 0.1	Pushbutton on right-hand side of street Pushbutton on left-hand side of street	Pushb_Right Pushb_Left
M 0.0	Memory bit for switching the signal after a green request from a pedestrian	Signal_Mem
T 2 T 3 T 4 T 5 T 6	Duration of yellow phase for vehicles Duration of green phase for pedestrians Delay red phase for vehicles Duration of red/yellow phase for vehicles Delay next green request for pedestrians	Veh_Yel_Phase Ped_Green_Phase Veh_Del_Red Veh_Red_Yel_Phase Ped_Del_Green

Sample Program Sequence

If input I 0.0 or I 0.1 is set by the pedestrian green request, the following occurs:

- The traffic light switches from Q 0.7 (green) via Q 0.6 (yellow) to Q 0.5 (red) and
- The pedestrian light switches from Q 0.0 (red) to Q 0.1 (green)

The outputs are set alternately by the timers defined in the program.

Figure 2-5 shows the signal states of the inputs and outputs during the TLIGHT sample program sequence:

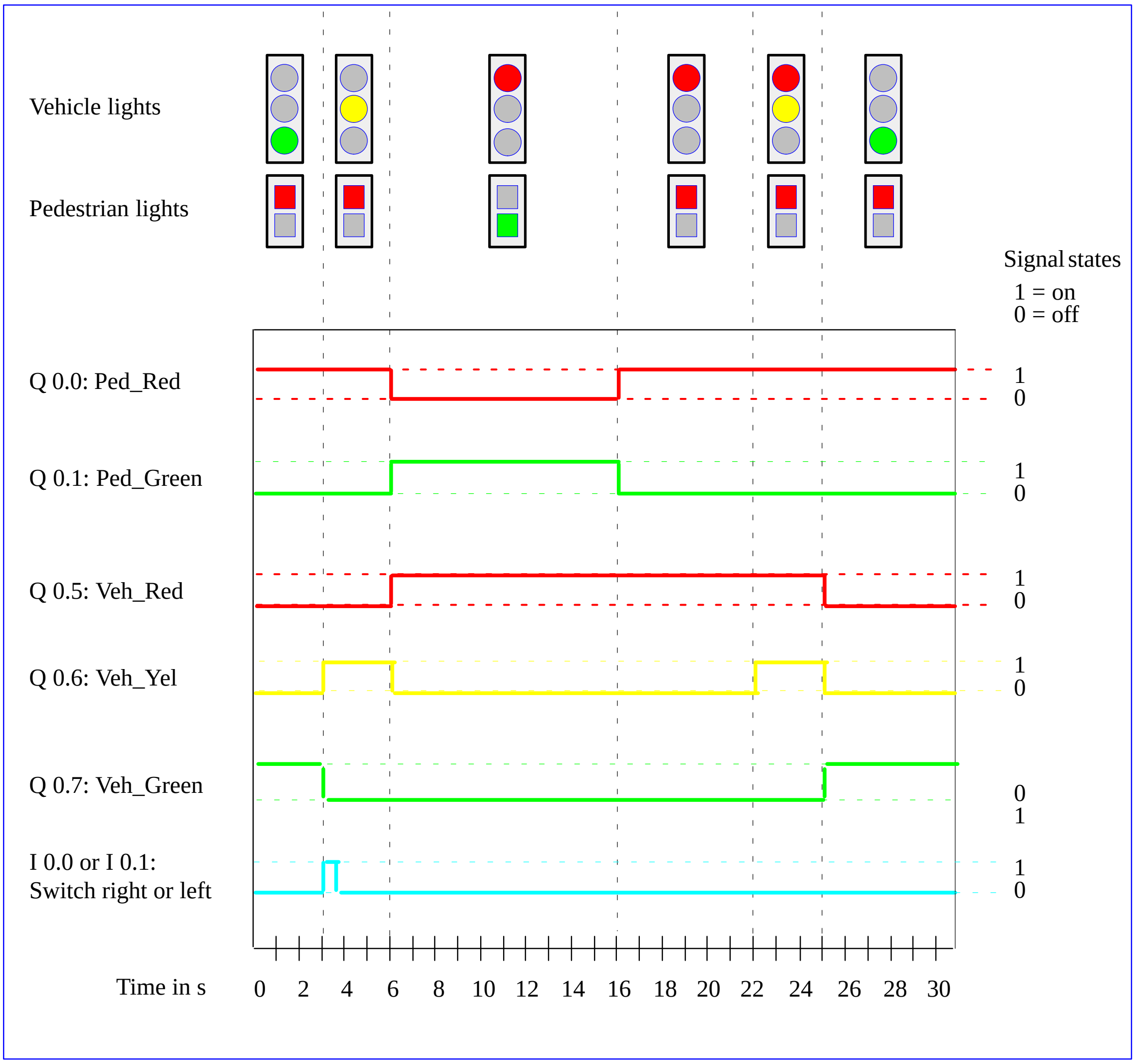
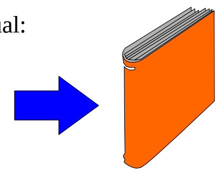


Figure 2-5 Signal States of the Inputs and Outputs During the TLIGHT Sample Program Sequence

Other References for This Chapter

Where Can You Find Other References?

If you still have questions and you need additional information with regard to defining and structuring your own control task, our manuals may be of assistance.

You Will Find Information on...	In...
defining and structuring a control task and on overall program design for your S7-300	the following manual:  System Software for S7-300 and S7-400 Program Design

3

Installing and Wiring Your S7-300

This chapter shows you step-by-step how to proceed.

What Do You Need to Know Before You Start?

Are There Any Slot Rules?

There are a few simple slot rules that you need to observe when installing your S7-300 (see Figure 3-1):

- The power supply module (PS) must always be plugged in as the first module on the left of the DIN rail.
- The CPU is always the second module to be plugged in to the right of the power supply module.
- Up to 8 signal modules can be plugged in to the right of the CPU.

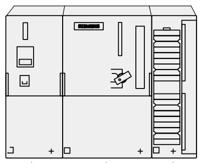


Figure 3-1 Simple Slot Rules for Arranging the Modules

We require only one simulator module for working with the primer and for executing our sample program.

Please Note

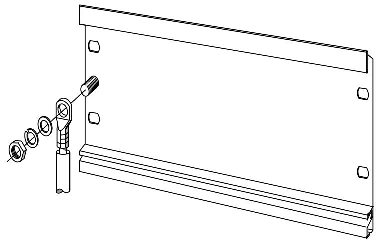
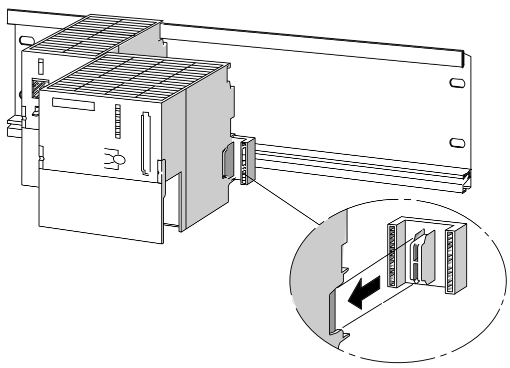
You can install your S7-300 either in a flat or an upright position.

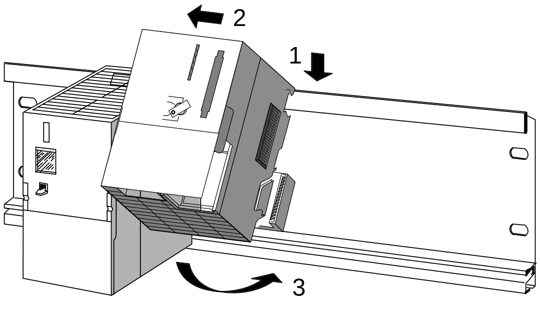
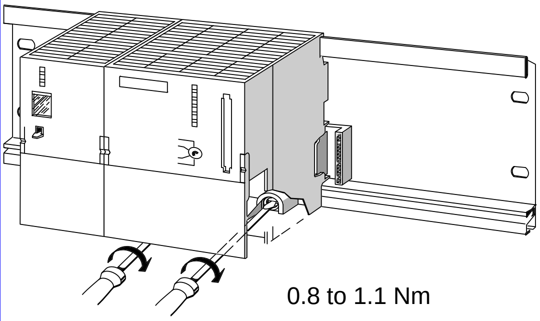
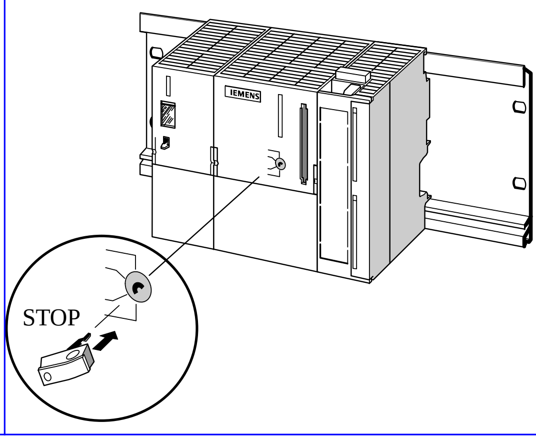
We have decided on horizontal installation, and the following explains how.

How Do You Install Your S7-300?

Basic Procedure

You can install your S7-300 with just a few adjustments as shown below:

Step	Procedure	Illustration
1.	Install the DIN rail and ground the connection.	
2.	Plug the bus connector into the module to be installed (the illustration shows the CPU).	

Step	Procedure	Illustration
3.	Hang each module (the illustration shows the CPU) onto the DIN rail and swing it down.	
4.	Screw the module tight.	
5.	Insert the key into the CPU.	

How Do You Wire Your S7-300?

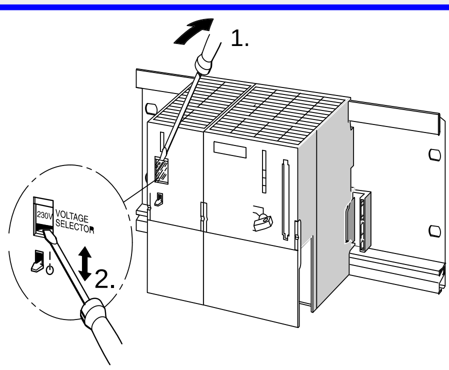
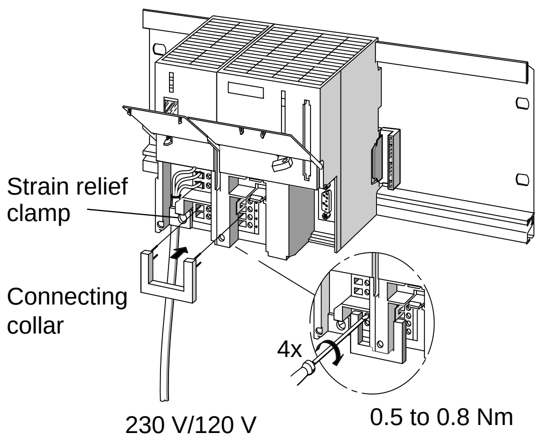
Overview

The table below shows you the basic procedure for wiring your S7-300. You should observe the assembly guidelines which are described in the manuals listed at the end of this chapter.

Warning

Only wire the S7-300 when the power is switched off!

You can wire the PS 307 power supply module to the CPU 313/314/315 via the connecting collar enclosed. The PS 307 power supply module with the CPU 312 IFM is wired via the front connector of the integral inputs/outputs on the CPU 312 IFM.

<table>
<tr><th>Step</th><th>Procedure</th><th>Illustration</th></tr>
<tr><td>1.</td><td>Set the voltage (120/230 VAC) on the power supply module.</td><td>
</td></tr>
<tr><td>2.</td><td>Wire the power supply module (PS 307) to the CPU (311/314/315) with the connecting collar.</td><td>
</td></tr>
</table>

How Do You Connect Your Programming Device/PC?

Connecting a Programming Device/PC to an S7-300

Connect your programming device/PC to your S7-300 via an MPI cable. The MPI cable is included with your programming device.

Figure 3-2 shows how the multipoint interfaces on the programming device and on your S7-300 are connected by an MPI cable:

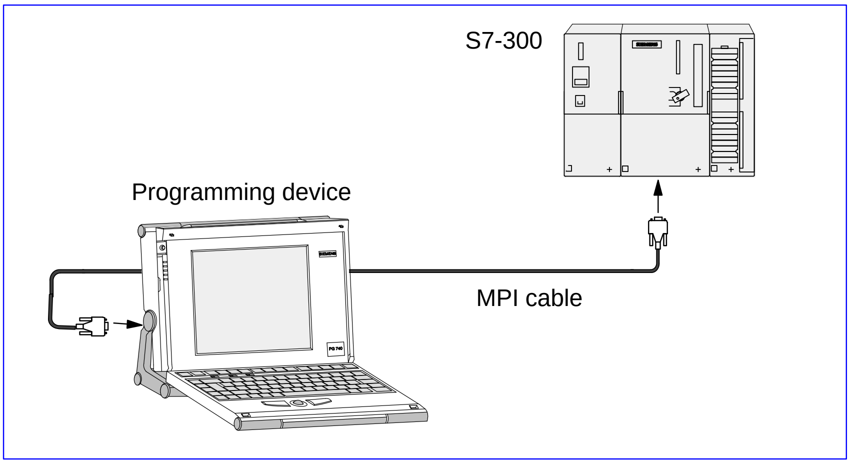


Figure 3-2 Connecting a Programming Device to an S7-300 via a Multipoint Interface

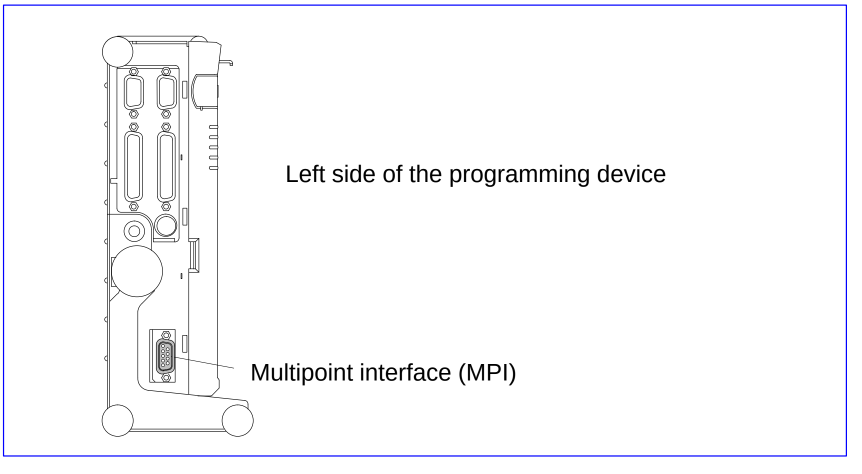


Figure 3-3 Left-Hand Side of the Programming Device with the Multipoint Interface

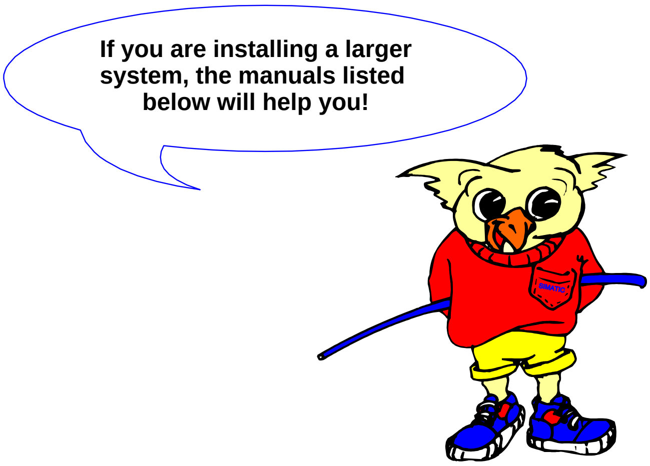
Switching On the CPU

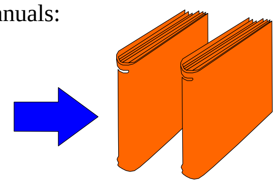
Now switch on the CPU of your S7-300 and turn the key to RUN-P.

Other References for This Chapter

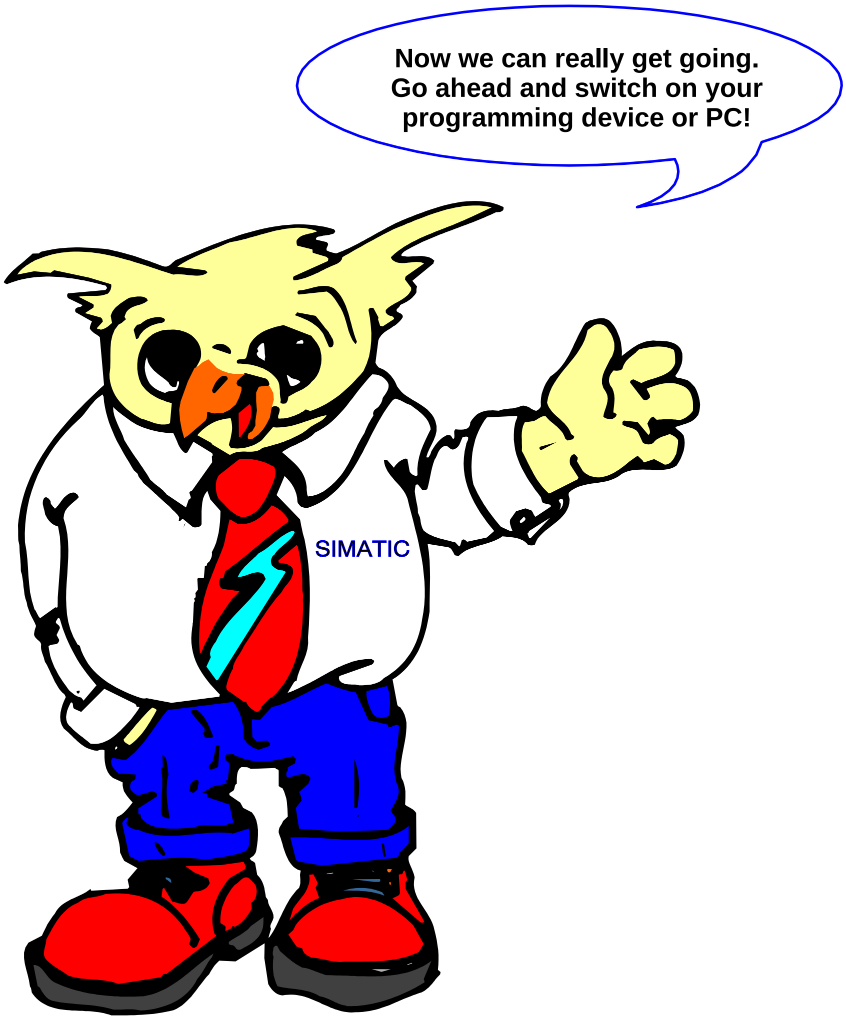
Where Can You Find Other References?

You will find additional information regarding the installation of complex systems in our manuals.



You Will Find Information on...	In...
• planning the mechanical and electrical installation of an S7-300 • addressing the S7-300 modules • installing an S7-300 • wiring an S7-300 • installing an MPI network for communications • memory cards for the S7-300 • spare parts for the S7-300 • the S7-300 modules	the following manuals:  S7-300 Programmable Controller, Hardware and Installation and S7-300 and M7-300 Programmable Controllers Module Specifications

Working with the STEP 7 Programming Software 4



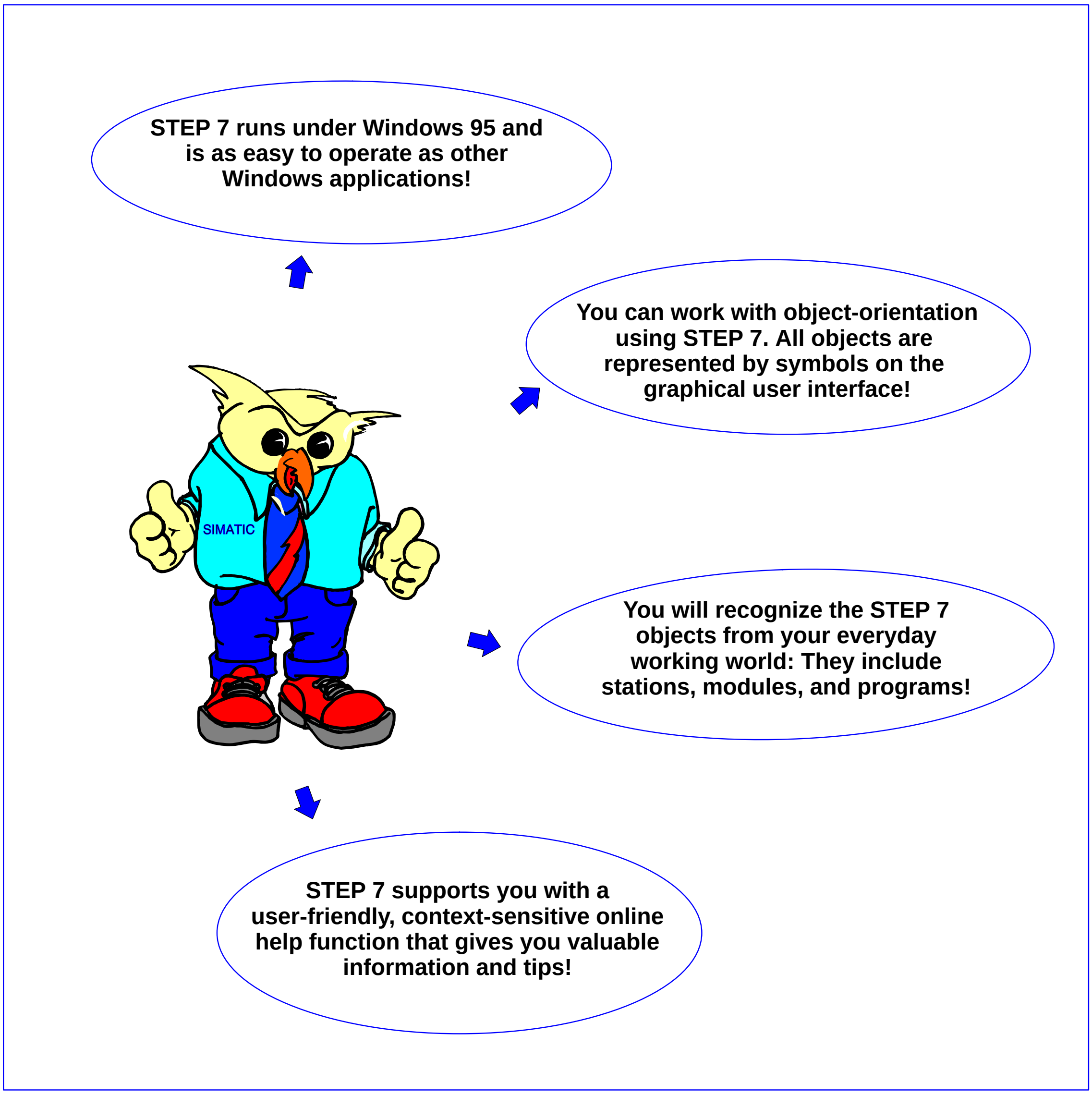
On the following pages, you will be introduced to the STEP 7 programming software.

What is STEP 7?

Introduction

STEP 7 is the programming software for SIMATIC S7/M7 and therefore for your S7-300. STEP 7 provides you with the entire functionality required for configuring, programming, and assigning parameters to your S7-300. Our programming software gives you effective support when solving your automation task.

The figure below shows you the most important features of STEP 7 which will make your task considerably easier:



How Do You Work with STEP 7?

Starting Up the Software

You can start up the software quite simply by double-clicking on the icon for the SIMATIC Manager.

Result: The project window for the SIMATIC Manager opens.

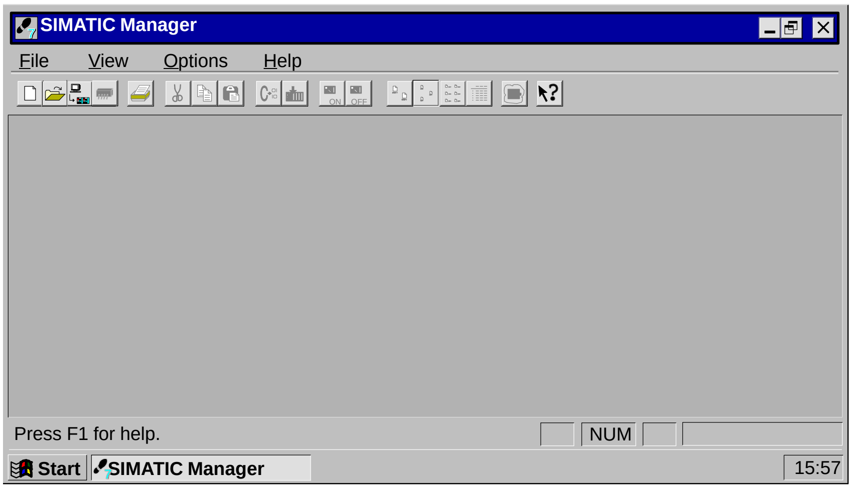


Figure 4-1 The SIMATIC Manager Window

SIMATIC Manager

The SIMATIC Manager is the entry interface for programming with STEP 7. It shows a hierarchical representation of all the objects in a project, which enable you to access all the functions you require to solve your automation task.

Starting from the SIMATIC Manager, you can:

- Configure and assign parameters to your S7-300
- Program your S7-300

Further Procedures

Your task is to create an automation solution for a traffic light control system. You will make it easier for yourself if you acquire and digest the following basic information:

- What objects you require for working with STEP 7 and
- How you can arrange and handle these objects.

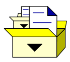
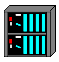
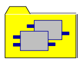
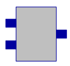
You will get to know these objects on the next page.

Which STEP 7 Objects Should You Know?

Introduction

The table below shows you the STEP 7 objects that you should know for the TLIGHT programming example.

You can see from the table which symbols are assigned to the individual objects and what they mean:

Symbol	Object	Description	In Container
	Project	Represents the entirety of all data and programs of an automation solution	Lies at the top of an object hierarchy
	SIMATIC 300 station	Represents a hardware installation with one or more programmable modules	Project
	Programmable module	Represents a programmable module (CPU)	Station
	S7 program	Consists of the “Blocks” and “Source Files” containers and the “Symbols” object	Programmable module or project
	Blocks	Contains the blocks in your user program	S7 program
	Block (offline) Block (online)	Can be, for example: • Logic blocks (OBs and FCs)	Blocks

Creating Objects

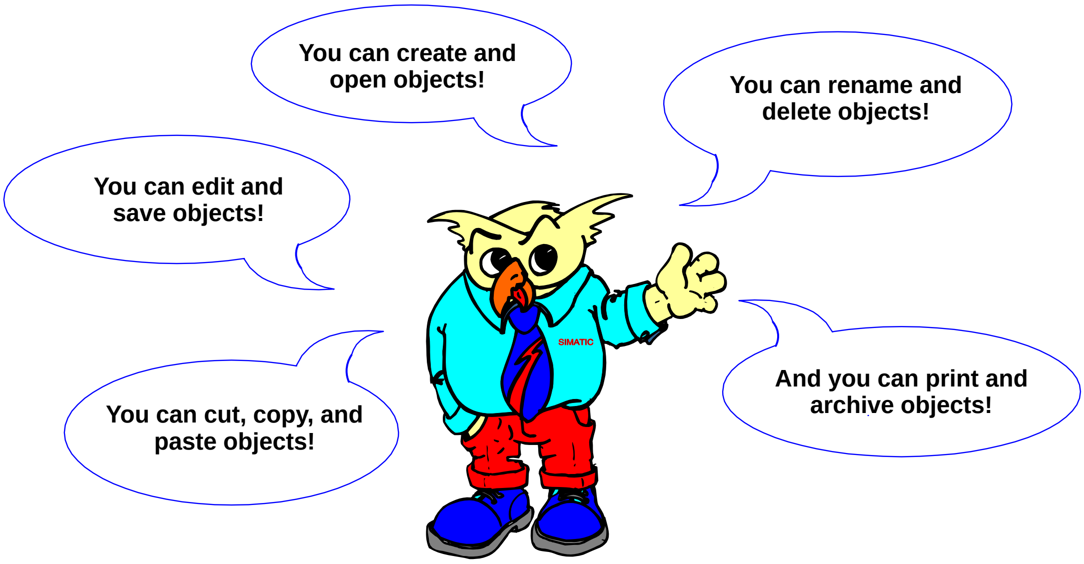
You create objects such as projects via the menu command **File** → **New...**. These objects can in turn contain other objects, such as programs and blocks. You can assign these via commands in the **Insert** menu. Blocks contain no further objects. Opening a block starts the assigned editor with which you can then edit the contents of the block.

Representing Objects

You can change the representation of objects and set the object properties in various dialog boxes.

Actions with Objects

You have access to all the processing functions via STEP 7 objects:



The Most Important Object is a ‘Project’

With STEP 7, you can structure a system into projects. A project contains the entire database for an automation solution. Creating a project or a project structure is therefore an essential prerequisite for working with STEP 7. Figure 4-2 shows you how a project is structured:

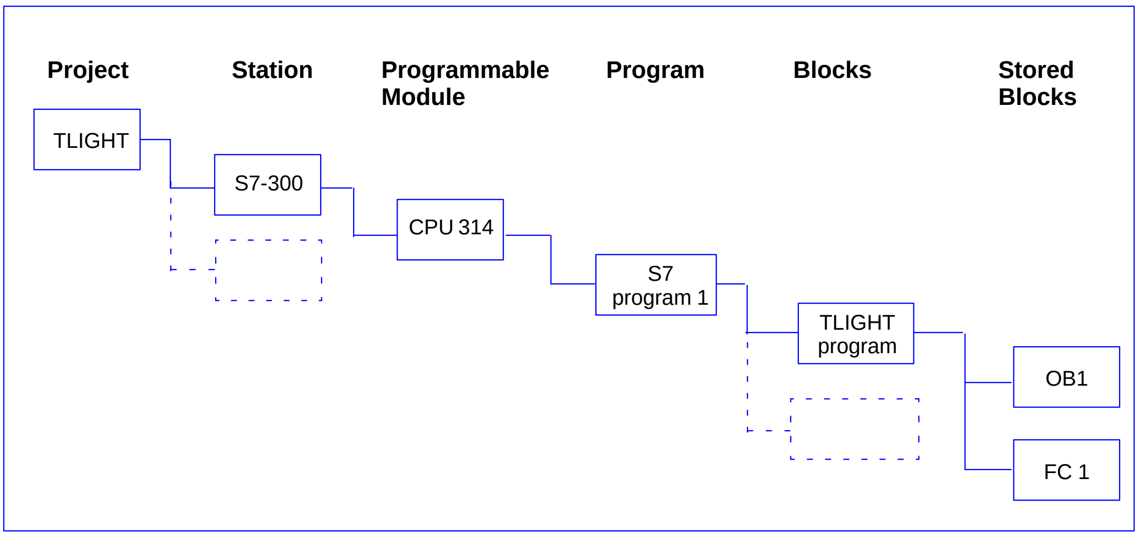


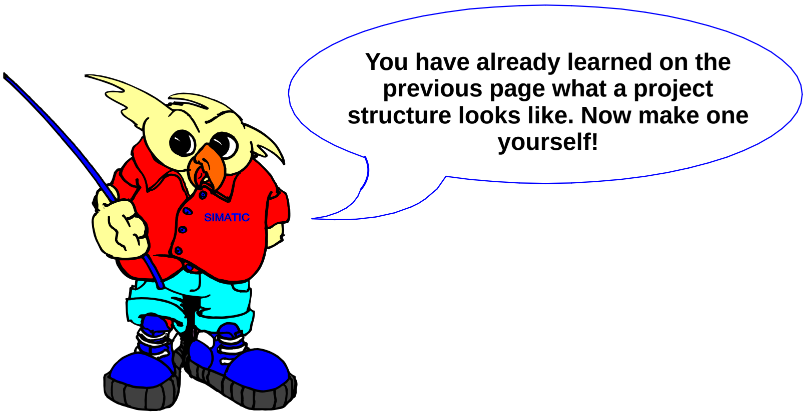
Figure 4-2 This is What a Project Structure Looks Like

On the next page, you will find out how to create a project and a station for the TLIGHT sample program.

This is How You Create a Project Structure!

Introduction

For the TLIGHT sample program, you need a project structure in which you can store the relevant data for this control task.



How Do You Proceed?

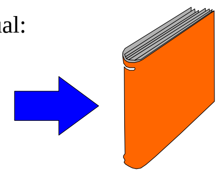
The table below shows you step-by-step how to create a project and a station:

Step	Procedure	Result
1.	Click on the menu command **File** → **New** → **Project...** in the SIMATIC Manager or click on the relevant icon in the toolbar. In the next dialog box, enter the name ‘TLIGHT’ and click on “OK” to create the project.	The new project with the name ‘TLIGHT’ is displayed. You can now insert further objects.
2.	Add a station to the open project with the menu command **Insert** → **Hardware** → **SIMATIC 300 Station**. In the project window, click on the “+” in front of the project symbol if the station is not displayed.	The station and its name are displayed in the project window. You have now inserted a hardware station in the project “TLIGHT”.

Other References for This Chapter

Where Can You Find Other References?

If you still have questions and you need additional information on creating projects, stations, and programs, our manuals may be of assistance.

You Will Find Information on...	In...
creating projects, stations, and programs	the following manual:  System Software for S7 and M7 STEP 7

Configuring and Assigning Parameters to Your S7-300

5

You can make all the necessary settings via the STEP 7 software.

What is Configuring and Assigning Parameters?

What Does Configuring Mean?

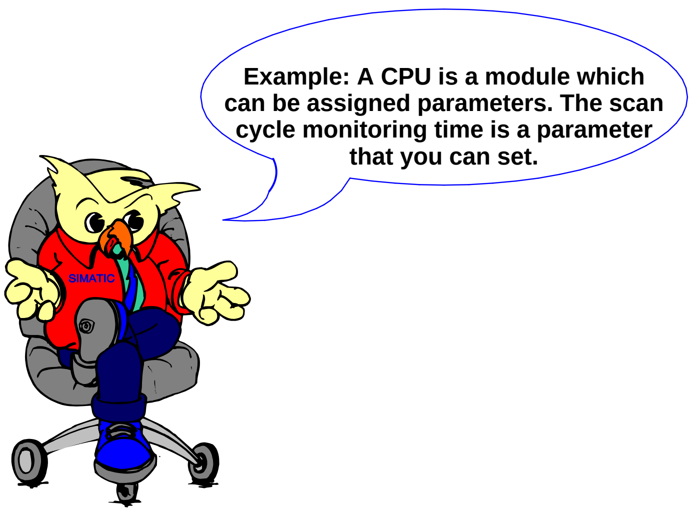
'Configuring' is arranging modules in a configuration table.

Just as in your actual system, you arrange the modules on a mounting rack using STEP 7. You can select the modules from an electronic catalog and enter them in the relevant slot in the configuration table. The slot in the configuration table should correspond to the real slot on the mounting rack.

STEP 7 then assigns an address to each module in the configuration table.

What Does Assigning Parameters Mean?

'Assigning parameters' means setting the characteristics and behavior of modules.



How to Configure and Assign Parameters

Prerequisites

Before you can enter a new configuration and assign parameters to the CPU of your S7-300, you must have created a project and selected the object that you want to configure (here, your SIMATIC 300 station).

To make sure that there are no ‘old’ blocks in your CPU, you should perform a memory reset before downloading the new configuration to the CPU.

Basic Procedure

We aim to show you the basic procedure for configuring and assigning parameters using the TLIGHT sample program. The figure below gives you an overview of this procedure:

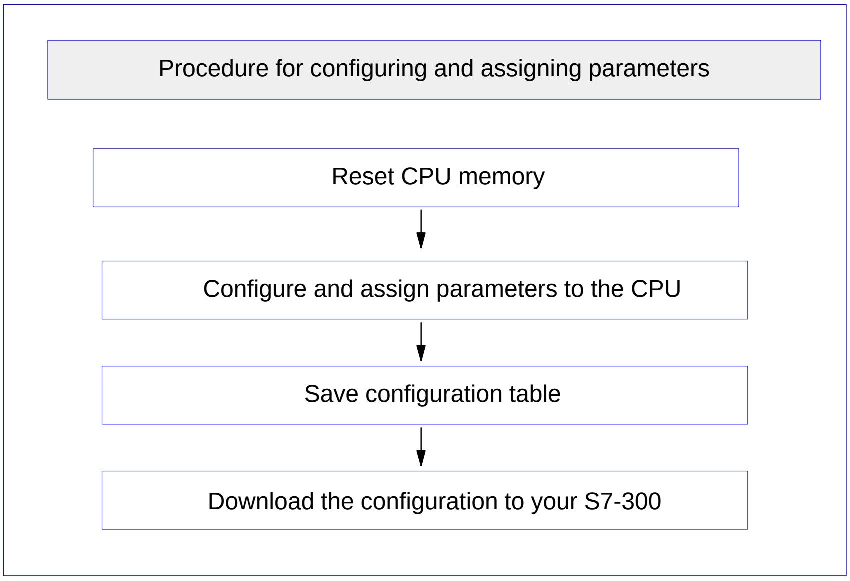
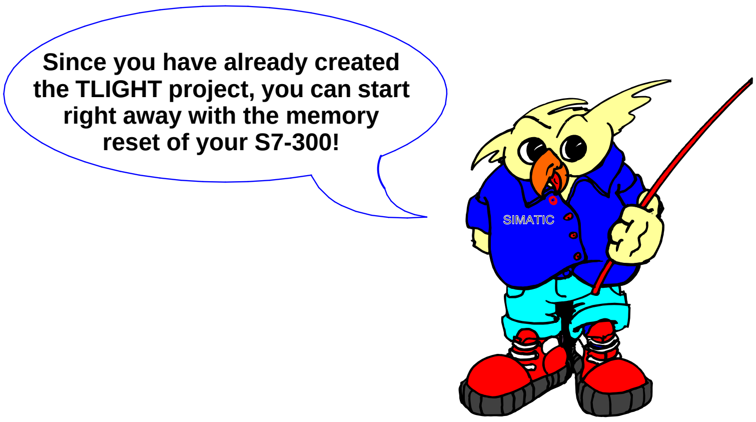


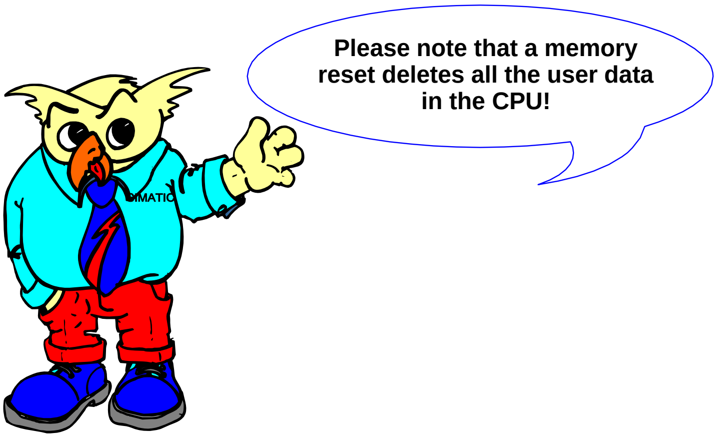
Figure 5-1 Procedure for Configuring and Assigning Parameters



First Perform a Memory Reset of Your CPU!

Introduction

Before downloading your configuration data to the CPU of your S7-300, perform a memory reset of the CPU to make sure that it does not contain any ‘old’ blocks.



How Do You Proceed?

The table below shows you step-by-step how to perform a memory reset on the CPU of your S7-300:

Step	Procedure	Result
1.	Click on the menu command **File** → **Open** → **Accessible Nodes** in the SIMATIC Manager and select the MPI address of your CPU in the subsequent dialog box to establish an online connection.	An online connection to your CPU now exists.
2.	Display the current operating mode of your CPU with the menu command **PLC** → **Operating Mode...**	The current mode of your CPU is displayed.
3.	Set the CPU to STOP by clicking on the ‘Stop’ button and confirm with “OK”. Exit the dialog box by clicking on “Close”.	The CPU goes to STOP.
4.	Call the ‘Memory Reset’ function with the menu command **PLC** → **Clear/Reset...** and confirm. The keyswitch on your CPU must be in the position “RUN-P” or “STOP”.	The following process now takes place in the CPU: • The CPU is reset and the entire user program is deleted. • The system parameters and the CPU and module parameters are reset to the default settings. • The CPU breaks off all existing connections.

How Do You Configure and Assign Parameters to Your S7-300?

How Do You Proceed?

The table below shows you step-by-step how to configure and assign parameters to your S7-300:

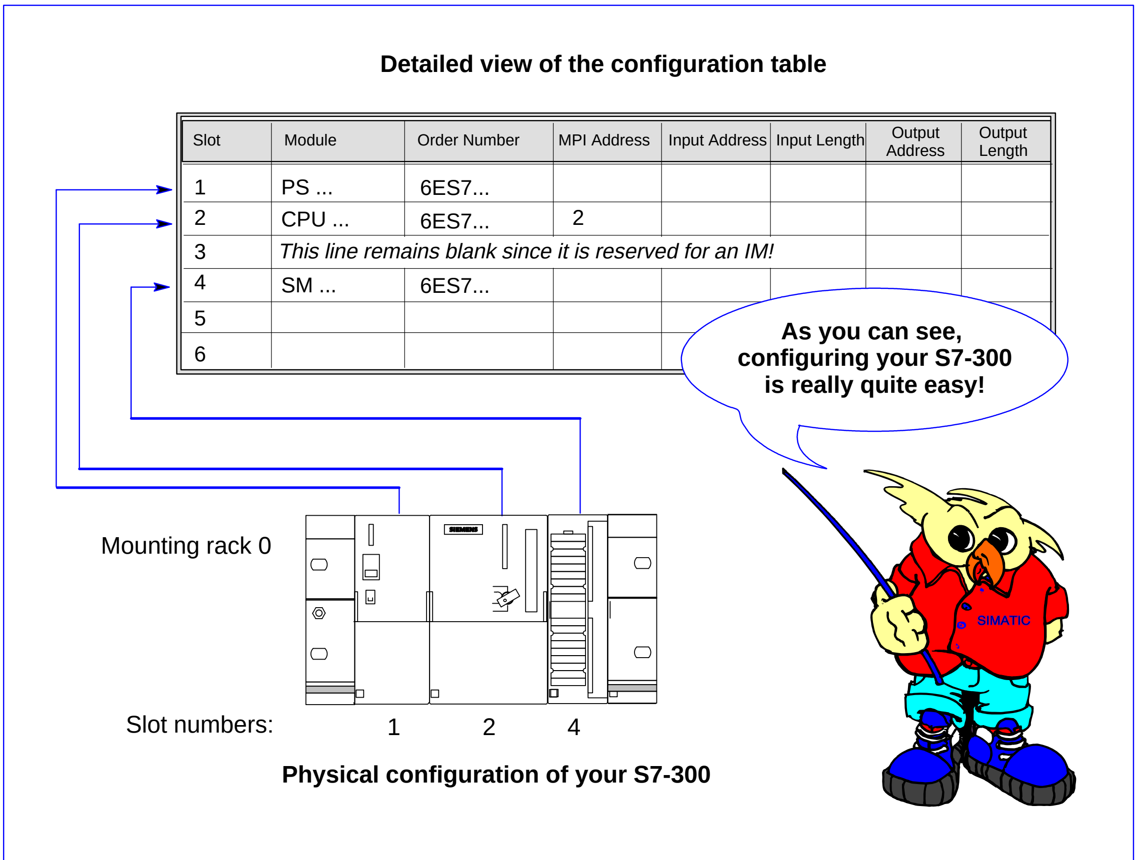
Step	Procedure	Result
1.	Select the SIMATIC 300 station (1) in your project and call up the configuration table with the menu command **Edit** → **Open Object.**	The configuration table appears on the screen with the working window and the “Hardware Catalog” window containing all the available modules. If the hardware catalog is not displayed, you can open it with “Ctrl + K”.
2.	In the hardware catalog, select the mounting rack via **SIMATIC 300** → **RACK 300** → **Rail** and drag and drop it into the working window.	A table is displayed showing the rail and its slots.
3.	Click on the table and select the detailed view of the configuration table with the menu command **View** → **Details.**	The detailed view of the configuration table with an overview of the order numbers and the module addresses appears on the screen, as shown in Figure 5-2.
4.	In the hardware catalog, select the module you want to use via **SIMATIC 300**: • Power supply module (PS 307 2A) • CPU (CPU 314) and • Simulator module (SM 323 DI8/DO8) and drag and drop these into lines 1, 2, and 4 of the configuration table.	The selected modules are displayed in the relevant lines of the configuration table. **Note:** Slot 3 in the configuration table remains free since this slot is reserved for a module that is not required here (IM).
The configuration table is now complete. You can now assign parameters to your CPU as described in steps 5 and 6. You must then save the configuration and download it as described in steps 7 and 8.		
5.	In the configuration table, double-click on the line of the module you want to configure – the CPU in this case.	A dialog box appears with information and the assignable parameters of the CPU.
6.	Click in the dialog box on the ‘Cycle / Clock Memory’ tab and change the set scan cycle monitoring time to 100 ms. Exit the dialog box with ‘OK’.	The default scan cycle monitoring time of 150 ms has been changed to 100 ms.
7.	Save your overall configuration to the TLIGHT project with the menu command **File** → **Save**.	Your overall configuration is saved on the hard disk of your programming device/PC in the TLIGHT project.
8.	Download the overall configuration to your S7-300 with the menu command **PLC** → **Download** → **To Module....** Click on ‘OK’ in the dialog box “Select Target Module” and confirm the preset node address in the dialog box “Define Node Address” with “OK”.	Your overall configuration is downloaded to the CPU with the node address 2. The modified parameters become effective the next time you execute a complete restart.

What Does a Configuration Table Look Like?

Configuration Table Example

You must fill in a configuration table according to the hardware configuration of your S7-300. The relevant position in the configuration table is assigned to the actual slot in your mounting rack.

Figure 5-2 shows you an example of converting the actual configuration of your S7-300 to the configuration table:



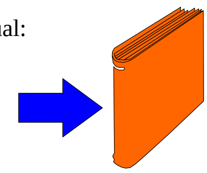
Slot	Module	Order Number	MPI Address	Input Address	Input Length	Output Address	Output Length
1	PS ...	6ES7...					
2	CPU ...	6ES7...	2				
3	*This line remains blank since it is reserved for an IM!*						
4	SM ...	6ES7...					
5							
6							

Figure 5-2 From Your S7-300 Configuration to the Configuration Table

Other References for This Chapter

Where Can You Find Other References?

If you still have questions and you need additional information on configuring and assigning parameters to your S7-300, our manuals may be of assistance.

You Will Find Information on...	In...
configuring and assigning parameters to your S7-300	the following manual:  Standard Software for S7 and M7 STEP 7

Programming Logic Blocks with STEP 7 6

Programming your blocks is easy and user-friendly with STEP 7.

Procedure for Programming Blocks

Introduction

You can program your programmable controller quite easily by creating a user program that you download to the CPU of your S7-300. The user program you have to create consists of different blocks which enable you to structure your program. For the TLIGHT sample program, you require only two blocks:

- An organization block (OB1) for cyclic program execution and
- A function (FC1) into which you enter the actual program.

What is an Organization Block (OB)?

An organization block (OB) is the interface between the operating system of the CPU and your user program. The order in which the user program is executed is defined in an OB.

What is a Function (FC)?

A function (FC) is a logic block without ‘memory’ which can nevertheless transfer parameters. This block is especially suitable for programming frequently recurring functions.

Basic Procedure

Using the TLIGHT sample program, we aim to show you the basic procedure for programming blocks. Figure 6-1 gives you a task-oriented overview:

Procedure for programming blocks

Create blocks:
- Create FC1 and assign language editor (STL, LAD, or FBD)

↓

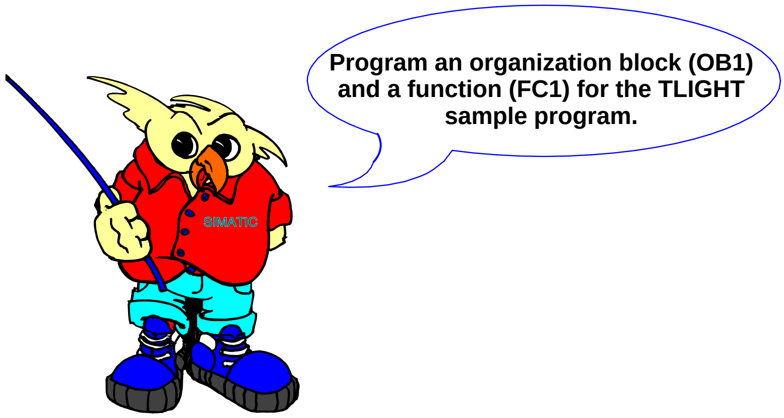
Program blocks (in STL, LAD, or FBD):
- Start language editor, declare variables and enter program, divided into networks.

↓

Save and download blocks:
- Save the blocks in your programming device/PC and download them to the CPU

Figure 6-1 Basic Procedure for Programming

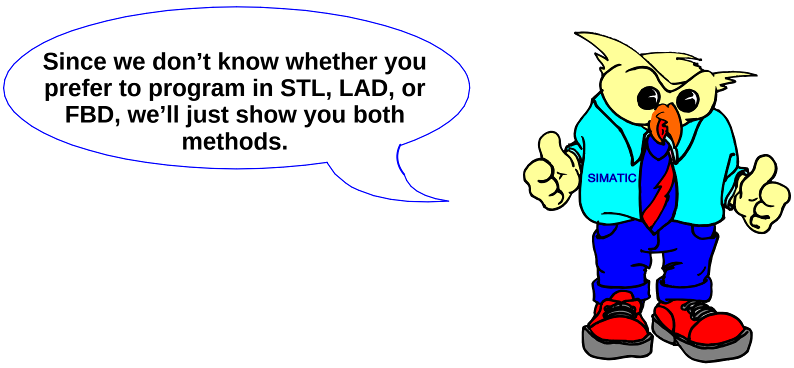
How Do You Create the Required Blocks?



How Do You Proceed?

The table below shows you how to create the required function:

Step	Procedure	Result
1.	Open the TLIGHT project in the SIMATIC Manager with the menu command **File** → **Open** → **Project ....**	The project window for the TLIGHT sample program is opened offline.
2.	Open the containers of the TLIGHT project down to the lowest level by clicking on "+" and select the "Blocks" container.	The OB1 stored in the "Blocks" container is displayed offline.
3.	Insert a function with the name FC1 using the menu command **Insert** → **S7 Block** → **FC.** In the dialog box which then appears, select the programming language you require (STL, LAD, or FBD) and confirm with "OK".	FC1 and OB1 are displayed offline in the project window of the SIMATIC Manager.



What are STL, LAD, and FBD?

STL = Statement List

STL stands for Statement List and is a textual programming language in STEP 7. The syntax of the statements is close to machine code: Instructions or operations are followed by addresses. You can use this to program your control task in STL for optimal memory capacity and runtime.

LAD = Ladder Logic

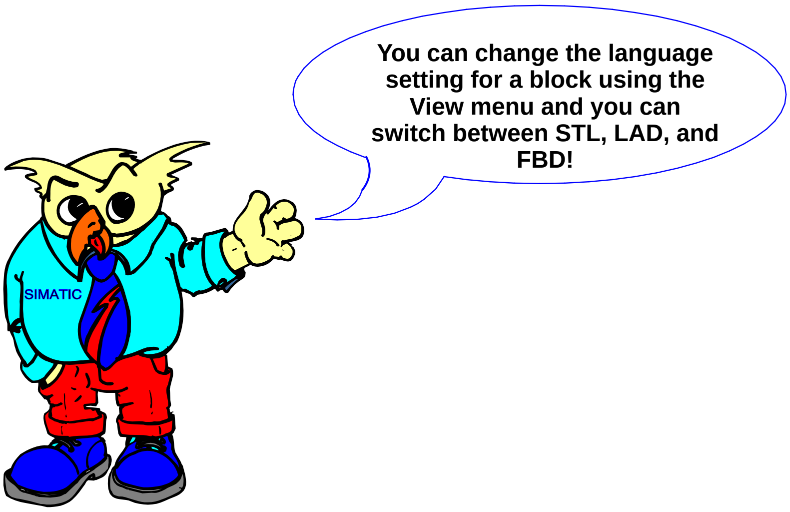
LAD stands for Ladder Logic and is a graphical programming language in STEP 7. The syntax of the statements is similar to a circuit diagram and enables you to easily follow the signal flow between conductor bars via contacts, complex elements and coils.

FBD = Function Block Diagram

FBD stands for Function Block Diagram and is also a graphic programming language in STEP 7. The statements are represented as logic boxes, similar to those in boolean algebra. As with the wiring diagrams in digital technology, you can follow the signal flow by means of the boxes.

STL, LAD, and FBD are Integrated into STEP 7

STL, LAD, and FBD are integrated into the STEP 7 Standard software. So after you have installed STEP 7, all the editor, compiler, and test functions of STL, LAD, and FBD are available to you.

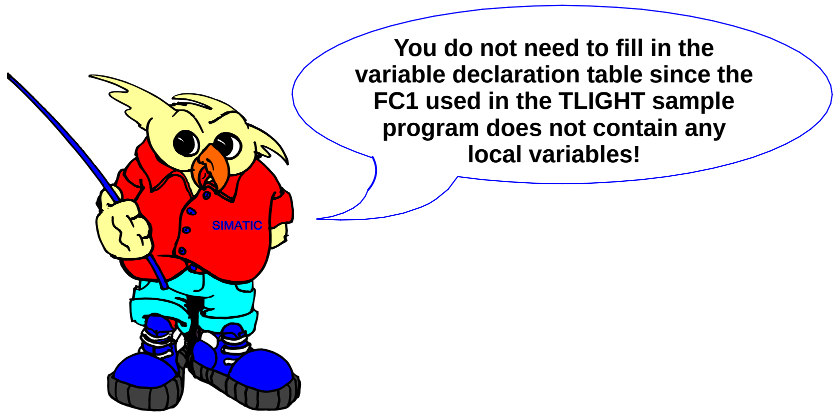


Starting STL, LAD, and FBD

If you double-click on the FC1 which has already been programmed, the language editor you have assigned (STL, LAD, or FBD) will be started and a split window appears with:

- The **variable declaration table** of the block in the upper half and
- The **code section** of the block in the lower half of the window. This is where you write your program.

Variable Declaration Table



Code Section

In the code section, you enter the program for your block as STL statements or as LAD and FBD elements in networks. The incremental STL, LAD, and FBD editor executes a syntax check immediately after each statement or element is entered and displays any errors in red italics. Any such syntax errors must be corrected before saving the block.

In the TLIGHT sample program, the code section consists of several networks, which in turn contain a list of statements or a circuit.

Components of the Code Section

In the code section of a block, you can edit the block title, block comments, network title, network comments, and STL statements or LAD/FBD elements within individual networks, as shown in Figure 6-2:

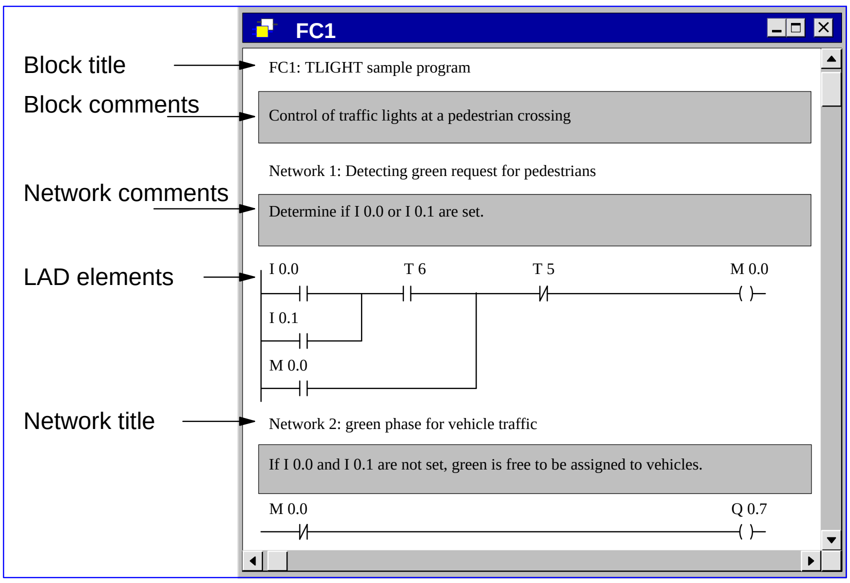


Figure 6-2 Structure of the Code Section for a LAD Block

How Do You Program Blocks in STL?

Prerequisite

You must have created FC1 and OB1 as blocks and assigned the incremental STL editor as the language editor.

How Do You Proceed?

The table below shows you step-by-step how to enter STL statements:

Step	Procedure	Result
1.	Start the assigned STL editor by double-clicking on FC1 in the SIMATIC Manager.	FC1 is opened and appears in a split window with: • The variable declaration table and • The code section of the block.
2.	Select the menu command **Insert** → **Network** to create a new network or click on the relevant icon in the toolbar.	The first network is displayed in the code section of the open FC1.
3.	Select the area below the network comment and enter the STL statements listed below line for line via the keyboard.	After each line is entered, it is checked for syntax errors. Lines containing errors are displayed in red. These must be corrected before the block is saved.
4.	Create a total of 11 networks as described above and enter the statements listed below in the relevant networks.	If no syntax errors are displayed (red background), programming of FC1 is complete and you can now save it.
5.	Save the complete and correctly created FC1 with the menu command **File** → **Save**	The block is saved on your programming device/PC.
6.	Then program OB1 in the same way.	OB1 is programmed and saved on your programming device/PC.

STL Statements for FC1

Enter the STL statements listed below for the TLIGHT sample program network by network in your FC1. Proceed as described in the table. You have already defined the required addresses in Chapter 2 of the Primer.

Network 1:		**Detecting green request for pedestrians**
	A(	
	A(	
	O	I 0.0
	O	I 0.1
	)	
	A	T 6
	O	M 0.0
	)	
	AN	T 5
	=	M 0.0
Network 2:		**Green phase for vehicles**
	AN	M 0.0
	=	Q 0.7
Network 3:		**Timer start for yellow phase for vehicles**
	A	M 0.0
	L	S5T#3S
	SI	T 2
Network 4:		**Yellow phase for vehicles**
	A	M 0.0
	A(	
	ON	T 2
	O	T 4
	)	
	=	Q 0.6
Network 5:		**Red phase for vehicles**
	A	M 0.0
	A	T 2
	=	Q 0.5
Network 6:		**Timer start for green phase for pedestrians**
	A	Q 0.5
	L	S5T#10S
	SI	T 3
Network 7:		**Green phase for pedestrians**
	A	Q 0.5
	AN	T 3
	=	Q 0.1

Network 8:	**Timer start for yellow phase delay for vehicles**
A	M 0.0
A	T 3
L	S5T#6S
SI	T 4

Network 9:	**Red phase for pedestrians**
A	M 0.0
A(	
ON	T 2
O	T 3
)	
ON	M 0.0
=	Q 0.0

Network 10:	**Timer start for red/yellow phase for vehicles**
A	M 0.0
A	T 4
L	S5T#3S
SI	T 5

Network 11:	**Timer start for green request delay**
A	Q 0.7
L	S5T#1S
SI	T 6

STL Statements for OB1

Enter the STL statement below for the TLIGHT sample program in your OB1. Proceed as described in the table on the previous page.

Network 1:	**FC1 call for traffic light control**
CALL	FC1

How Do You Program Blocks in LAD?

Prerequisite

You have created OB1 and FC1 as blocks and you have assigned the incremental LAD editor as the language editor.

Starting the LAD Language Editor

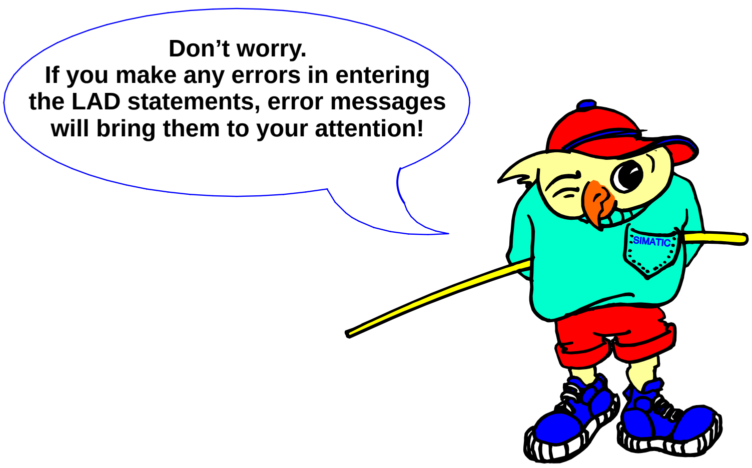
If you double-click on the FC1 which has already been programmed in the SIMATIC Manager, the assigned LAD editor is started and you can program the block.

Rules for Entering LAD Elements

A LAD network or circuit can consist of several elements in several branches. All elements and branches of such a network must be linked together. The left conductor bar does not count as a connection.

There are some rules to observe when programming in LAD:

- Each LAD network must be terminated with a coil or box.
- Branches that could result in current flow in the opposite direction cannot be edited.
- Branches that would cause a short-circuit cannot be edited.



Methods of Entering LAD Elements

There are the following methods for inserting LAD elements:

- You can enter NO contacts, NC contacts, or coils via function keys F2, F3, or F7 or you can select the icons for NO contacts, NC contacts, or coils from the toolbar.
- You can select and insert LAD elements with the menu command **Insert → LAD Element**.
- In addition, you can select and insert program elements, for example, timers, from a catalog. Open the “Program Elements” catalog with the keys “Ctrl + K” or using the corresponding icon in the toolbar.

How Do You Proceed?

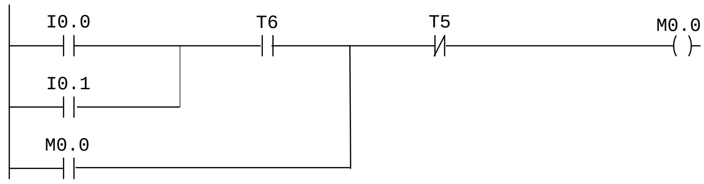
The table below shows you step-by-step how to enter LAD elements:

Step	Procedure	Result
1.	Start the assigned LAD editor by double-clicking on FC1 in the SIMATIC Manager.	FC1 is opened and a split window appears with: • The variable declaration table and • The code section of the block.
2.	Select the menu command **Insert→ Network** to create a new network or click on the relevant icon in the toolbar.	The first network is displayed in the code section of the open FC1. It contains a branch with a coil.
3.	Select the area below the network comment and enter the LAD elements listed below via the menu command **Insert → LAD Element** or using the corresponding function keys.	After each element is entered, it is checked for syntax errors. Elements containing errors are displayed in red. These must be corrected before the block is saved.
4.	Create a total of 11 networks as described above and enter the elements in the relevant networks.	If no syntax errors are displayed (red background), programming of FC1 is complete and you can now save it.
5.	Save the complete and correctly created FC1 with the menu command **File → Save**	The block is saved.
6.	Then program OB1 in the same way.	OB1 is programmed and saved on your programming device/PC.

LAD Elements for FC1

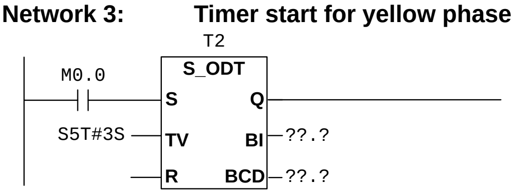
Enter the LAD elements listed below for the TLIGHT sample program network by network in your FC1. Proceed as described in the table. You have already defined the required addresses in Chapter 2 of the Primer.

Network 1: **Detecting green request for pedestrians**

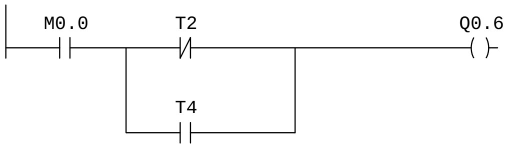


Network 2: **Green phase for vehicles**

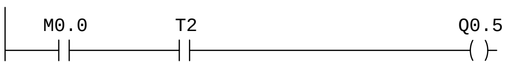
Network 3: **Timer start for yellow phase for vehicles**



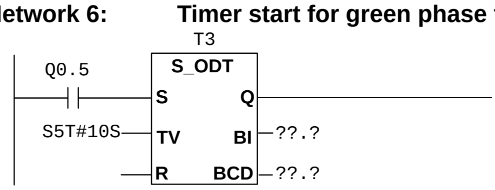
Network 4: **Yellow phase for vehicles**



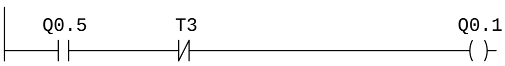
Network 5: **Red phase for vehicles**



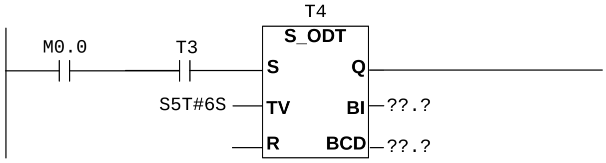
Network 6: **Timer start for green phase for pedestrians**



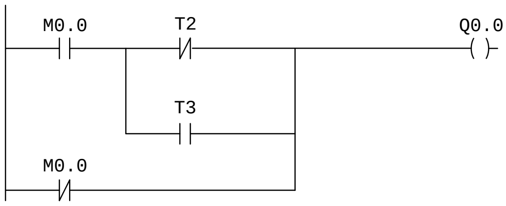
Network 7: **Green phase for pedestrians**



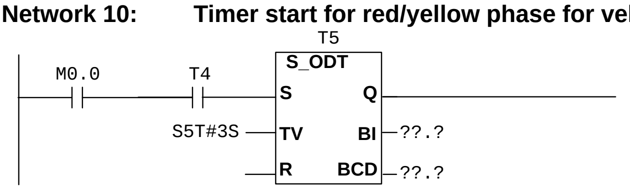
Network 8: **Timer start for yellow phase delay for vehicles**



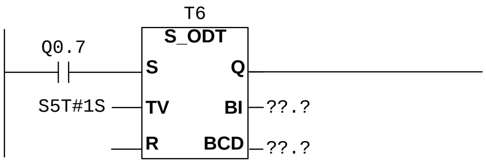
Network 9: **Red phase for pedestrians**



Network 10: **Timer start for red/yellow phase for vehicles**



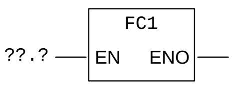
Network 11: **Timer start for green request delay**



LAD Elements for OB1

Enter the LAD elements below for the TLIGHT sample program in your OB1. Proceed as described in the table on the previous page.

Network 1: **FC1 call for traffic light control**



How Do You Program Blocks in FBD?

Prerequisite

You have created OB1 and FC1 as blocks and you have assigned the incremental FBD editor as the language editor.

Starting the FBD Language Editor

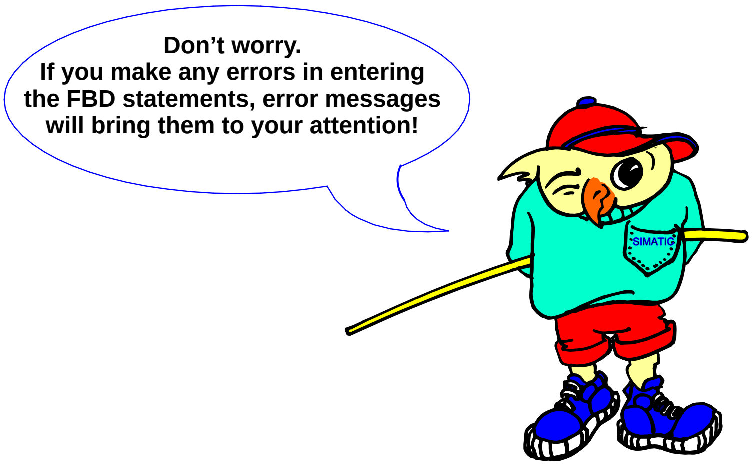
If you double-click on the FC1 which has already been programmed in the SIMATIC Manager, the assigned FBD editor is started and you can program the block.

Rules for Entering FBD Elements

An FBD network can consist of several elements in several branches. All elements and branches of such a network must be linked together.

There are some rules to observe when programming in FBD:

- Each FBD network must end with either an assignment or a box.
- The following FBD elements cannot be used to end a network: compare boxes (-=-), midline outputs (-#-), and positive (-P-) or negative (-N-) edge evaluation.
- You can attach standard boxes (flip-flops, counters, math instructions, etc.) to boxes with binary logic.
- You cannot program unlinked logic operations with separate outputs in a network.



Methods of Entering FBD Elements

The following methods are available for inserting FBD elements:

- You can insert AND boxes, OR boxes, OUTPUT boxes, and insert and negate inputs using the function keys F2, F3, F7, F8, or F9. Alternatively, you can select the required elements from the toolbar.
- You can select and insert FBD elements with the menu command **Insert → FBD Element**.
- In addition, you can select and insert program elements, for example, timers, from a catalog. Open the “Program Elements” catalog with the keys “Ctrl + K”, or with the corresponding icon in the toolbar.

How Do You Proceed?

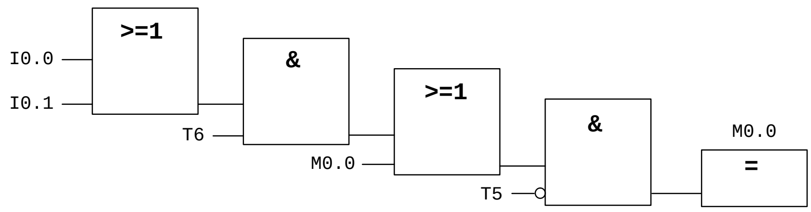
The table below shows you step-by-step how to enter FBD elements:

Step	Procedure	Result
1.	Start the assigned FBD editor by double-clicking on FC1 in the SIMATIC Manager.	FC1 is opened and a split window appears with: • The variable declaration table and • The code section of the block.
2.	Select the menu command **Insert→ Network** to create a new network or click on the relevant icon in the toolbar.	The first network is displayed in the code section of the open FC1. It contains a branch with a coil.
3.	Select the area below the network comment and enter the FBD elements listed below via the menu command **Insert → FBD Element** or using the corresponding function keys.	After each element is entered, it is checked for syntax errors. Elements containing errors are displayed in red. These must be corrected before the block is saved.
4.	Create a total of 11 networks as described above and enter the elements in the relevant networks.	If no syntax errors are displayed (red background), programming of FC1 is complete and you can now save it.
5.	Save the complete and correctly created FC1 with the menu command **File → Save**	The block is saved.
6.	Then program OB1 in the same way.	OB1 is programmed and saved on your programming device/PC.

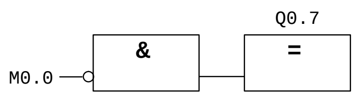
FBD Elements for FC1

Enter the FBD elements listed below for the TLIGHT sample program network by network in your FC1. Proceed as described in the table. You have already defined the required addresses in Chapter 2 of the Primer.

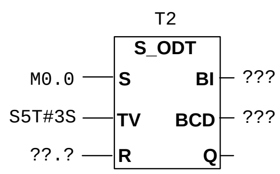
Network 1: **Detecting green request for pedestrians**



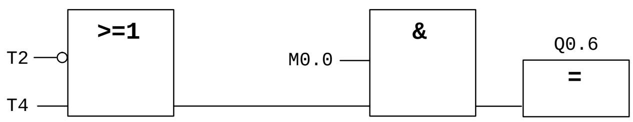
Network 2: **Green phase for vehicles**



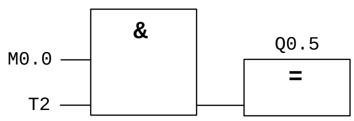
Network 3: **Timer start for yellow phase for vehicles**



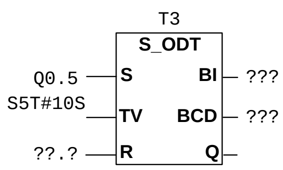
Network 4: **Yellow phase for vehicles**



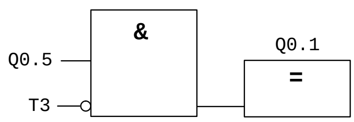
Network 5: **Red phase for vehicles**



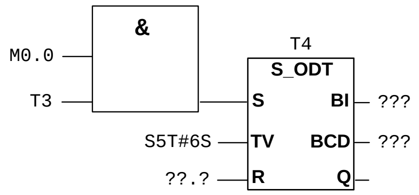
Network 6: **Timer start for green phase for pedestrians**



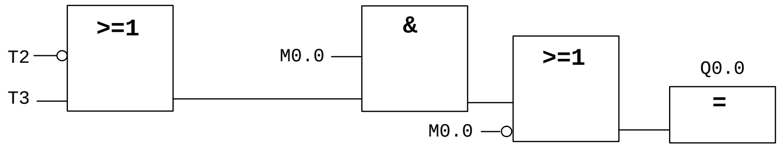
Network 7: **Green phase for pedestrians**



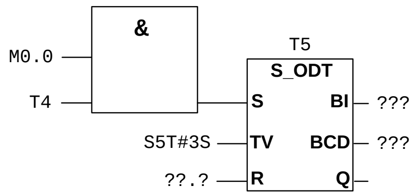
Network 8: **Timer start for yellow phase delay for vehicles**



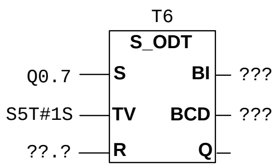
Network 9: **Red phase for pedestrians**



Network 10: **Timer start for red/yellow phase for vehicles**



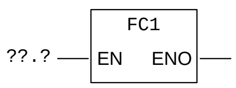
Network 11: **Timer start for green request delay**



FBD Elements for OB1

Enter the FBD elements below for the TLIGHT sample program in your OB1. Proceed as described in the table on the previous page.

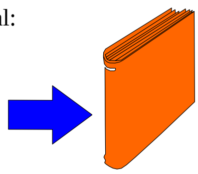
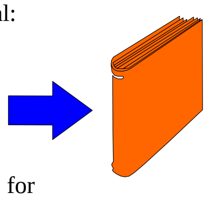
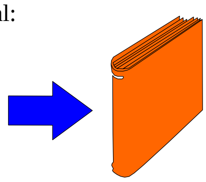
Network 1: **FC1 call for traffic light control**



Other References for This Chapter

Where Can You Find Other References?

If you still have questions and you need additional information regarding programming your blocks in STL, LAD, and FBD, our manuals may be of assistance.

You Will Find Information on...	In...
programming blocks in STL	the following manual:  Statement List (STL) for S7-300/S7-400 Programming
programming blocks in LAD	the following manual:  Ladder Logic (LAD) for S7-300/S7-400 Programming
programming blocks in FBD	the following manual:  Function Block Diagram (FBD) for S7-300/S7-400 Programming

Downloading and Testing Your User Program

7

When your program has been downloaded and tested, it can be executed.

Basic Procedure for Downloading and Testing

Introduction

You cannot test your program until you have downloaded it to the CPU of your S7-300.

You can download either individual blocks or the complete user program to the CPU of your programmable controller. Only individual blocks can be tested.

Basic Procedure

We aim to show you the basic procedure for downloading and testing using the TLIGHT sample program. The figure below gives a task-oriented overview:

Procedure for downloading and testing your program

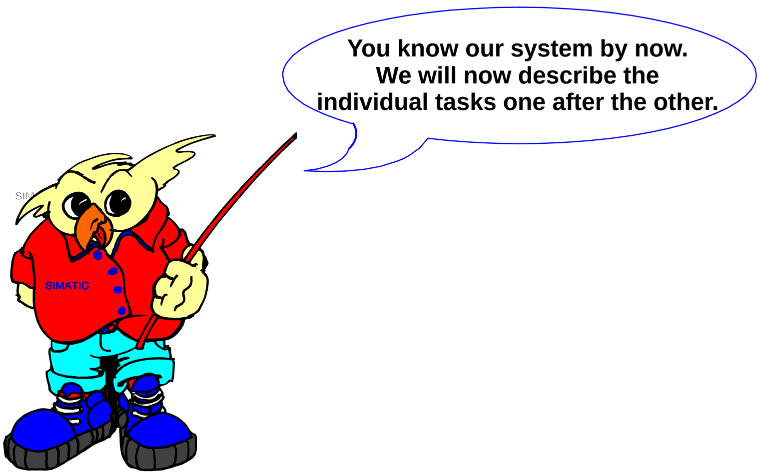
Download the user program to the CPU

↓

Testing an STL, LAD, or FBD block:

- Open the block online
- Define the settings for the test display
- Define the trigger conditions
- Select the test environment
- Start and stop the test

Figure 7-1 Procedure for Downloading and Testing the User Program



How Do You Download the User Program to Your S7-300?

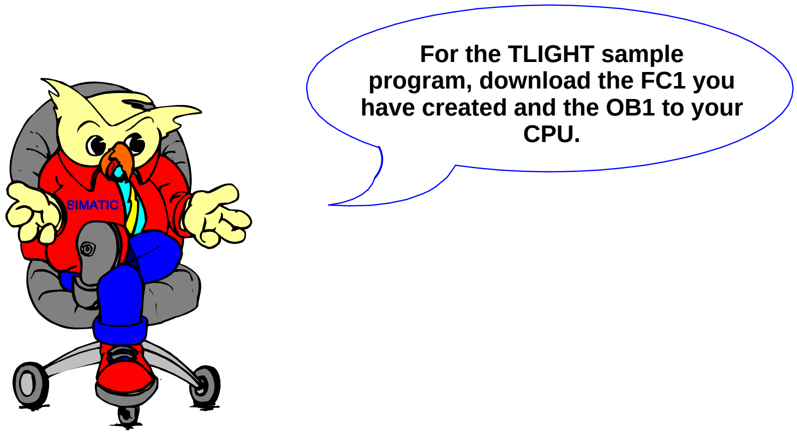
Prerequisites

The following prerequisites must be fulfilled before you can download the user program to your S7-300:

- There must be an online connection between your programming device and your S7-300.
- The program to be downloaded must have been compiled without errors.
- The CPU of your S7-300 must be in the STOP state.

What to Load, and Order of Calls

To test individual blocks, you must always load OB1 with all the blocks called in it. These blocks must be in reverse call order.



How Do You Proceed?

The table below shows you the procedure for downloading your user program:

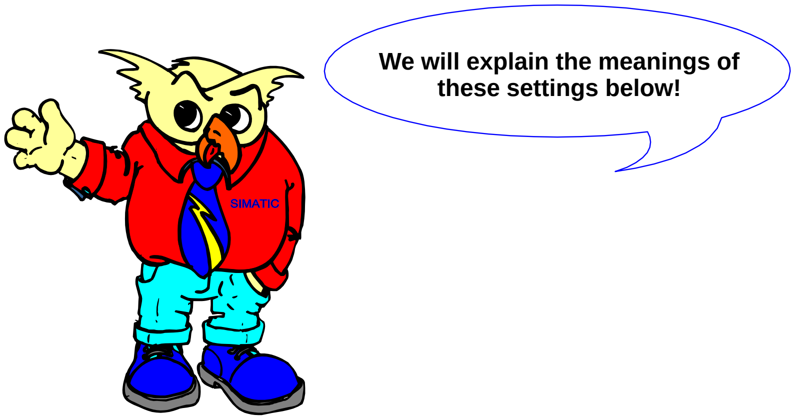
Step	Procedure	Result
1.	Select FC1 and OB1 in the project window of the SIMATIC Manager (offline view) in the “Blocks” container while holding the SHIFT key down, and select the menu command **PLC → Download.**	Both blocks are downloaded to the CPU of your S7-300.
2.	Select the menu command **View → Online** and open the containers of the TLIGHT project up to “S7 Program” (online). Change the operating mode of the CPU with the menu command **PLC → Operating Mode...** Click on “Complete Restart” in the dialog box which then appears, and then on “OK” and “Close”.	The CPU goes to RUN via complete restart. The TLIGHT sample program you have created will now be executed cyclically.

What Do You Have to Know about Testing?

Introduction

Regardless of whether you want to test a block created in STL, LAD, or FBD, you can always:

- Define the trigger condition,
- Select the test environment and
- Define the settings for the test display.



Meaning of the Trigger Condition

When you set the trigger condition, you set the call environment for the block to be tested. The test function is only executed if the trigger condition set is also fulfilled. You can choose between three different settings, as shown in Figure 7-2:

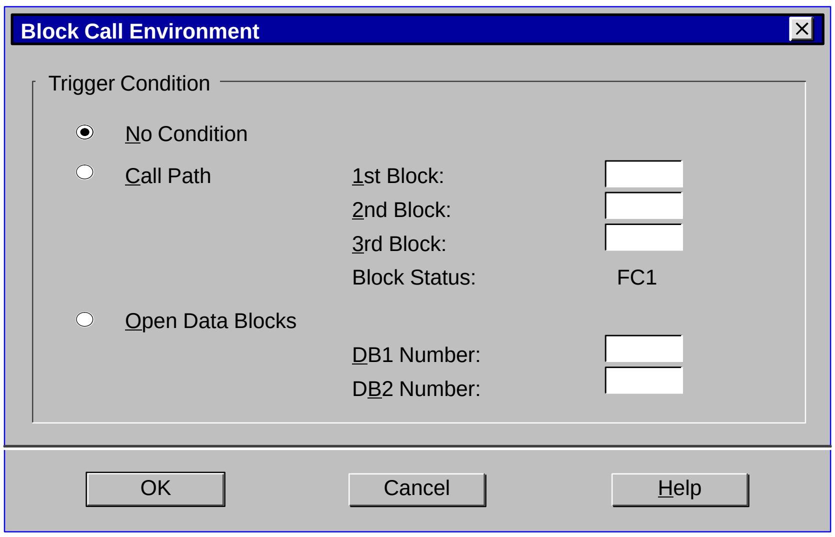


Figure 7-2 Defining the Trigger Condition

For the sample program, the default setting ‘No Condition’ applies since the call environment of the block to be tested is not important.

Meaning of the Test Environment

You can choose between the two online test situations 'Process' and 'Laboratory' in which you can test your program;

- With the 'Process' test environment, the status of statements during a loop run is only detected on the first loop run.
- With the 'Laboratory' test environment, the status of statements during a loop run is detected on every loop run.

For our TLIGHT sample program, we will use the 'Process' test environment that has already been set as the default setting.

Setting the Test Display for STL

For testing your STL program, you can select the status fields you want to have displayed for the STL program status.

For the TLIGHT sample program, select status bit, result of logic operation (RLO) and standard status for display. These are the default settings. You can make this selection by clicking on the 'Default' button.

Setting the Test Display for LAD and FBD

For testing your LAD and FBD program, you can define how you want to display the signal flow within the networks of a block. You can select color and line thickness for two possible cases:

- 'Status not fulfilled': In this case, the conditions along the current path have not been fulfilled; no current is flowing (broken line).
- 'Status fulfilled': In this case the conditions along the current path have been met; current is flowing (unbroken line).

How Do You Test Your STL Program?

Introduction

You test your STL program by displaying the program status for each STL statement in the status fields you have selected. Program status display is updated cyclically and only displayed for the area visible to the STL editor.

Prerequisites

In order to be able to display the program status, the following prerequisites must be fulfilled:

- You must have saved the block without errors and downloaded it to the CPU.
- The CPU is in RUN and the user program is running.
- You must open the block to be tested online.

How Do You Proceed?

The table below shows you how to test an STL program:

Step	Procedure	Result
1.	Select the “Blocks” container in your online project “TLIGHT”. In the right-hand window, select the block you want to test (here FC1), and open the block with the menu command **Edit** → **Open Object**.	FC1 is opened online.
2.	Select the menu command **Debug** → **Call Environment...** and click ‘No Condition’ as the trigger condition in the dialog box which follows. Close the dialog box with “OK”.	This means that you have selected **no** trigger condition for the call environment of the block.
3.	Select the ‘Process’ test environment via the menu command **Debug** → **Test Environment** → **Process**.	This means that the status of the statements for your program test will only be detected on the first loop run.
4.	In the ‘LAD/STL/FBD’ dialog box, select the ‘STL’ tab with the menu command **Options** → **Customize...** and click here on ‘Default’. Exit the dialog box with “OK”.	This defines the options for the STL program test. Status bit, RLO, and standard status will be displayed.
5.	Start recording the program status with the menu command **Debug** → **Monitor**	The STL program status is displayed in the code section of FC1 in the form of a table.
6.	Stop recording the program status with the same menu command **Debug** → **Monitor**.	Display of the STL program status is switched off again.

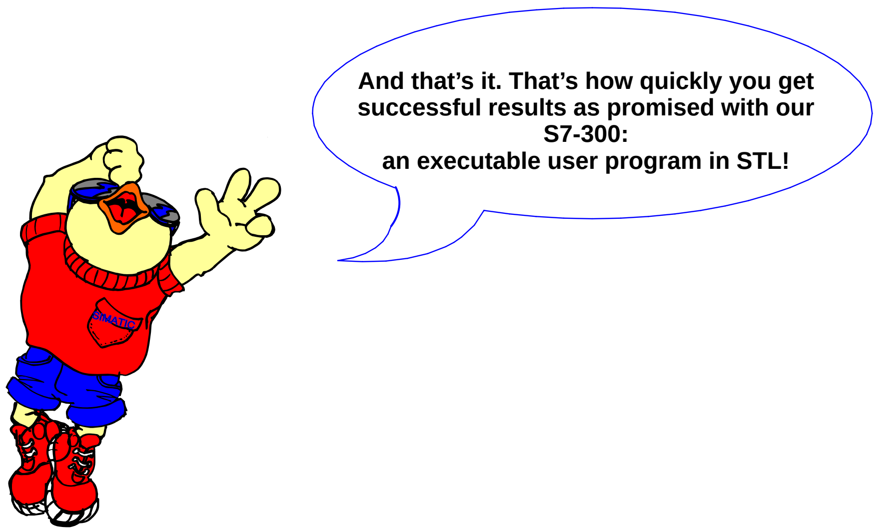
Displaying the Program Status in STL

The program status is only displayed for the area visible to the editor.

```
Network 1:  Detecting green request for pedestrians
                                RLO  STA  STANDARD
    A(                           0    1        0
    A(                           0    1        0
    O      I 0.0                 0    0        0
    O      I 0.1                 0    0        0
    )                            0    1        0
    A      T 6                   0    1        0
    O      M 0.0                 0    0        0
    )                            0    1        0
    AN     T 5                   0    0        0
    =      M 0.0                 0    0        0

Network 2 : Green phase for vehicles
                                RLO  STA  STANDARD
    AN     M 0.0                 1    0        0
    =      Q 0.7                 1    1        0
```

Figure 7-3 Example of Displaying the Program Status in STL



How Do You Test Your LAD Program?

Introduction

You can test your LAD program by defining how you want to display the signal flow within the networks of a block.

Prerequisites

To display the signal flow, the following prerequisites must be fulfilled:

- You must have saved the block without errors and downloaded it to the CPU.
- The CPU is in RUN and the user program is running.
- You must open the block to be tested online.

How Do You Proceed?

The table below shows you how to test a LAD program:

Step	Procedure	Result
1.	Select the “Blocks” container in your online project “TLIGHT”. In the right-hand window, select the block you want to test (here FC1), and open it with the menu command **File** → **Open Object**.	FC1 is opened online.
2.	Select the menu command **Debug** → **Call Environment...** and click ‘No Condition’ as the trigger condition in the dialog box which follows. Close the dialog box with “OK”.	This means that you have selected **no** trigger condition for the call environment of the block.
3.	Select the ‘Process’ test environment via the menu command **Debug** → **Test Environment** → **Process**.	This means that the status of the statements for your program test will only be detected on the first loop run.
4.	In the ‘STL/LAD/FBD’ dialog box, select the ‘LAD/FBD’ tab with the menu command **Options** → **Customize...** and click here on the settings you want for color and line thickness. Exit the dialog box with “OK”.	This defines the options for the LAD program test. Current flow is displayed in the colors and line thicknesses selected.
5.	Start recording the program status with the menu command **Debug** → **Monitor**	The LAD program status is displayed in the code section of FC1 in the form of a signal flow.
6.	Stop recording the program status with the same menu command **Debug** → **Monitor**.	Display of the LAD program status is switched off again.

Displaying the Program Status in LAD

The program status is only displayed for the area visible to the editor.

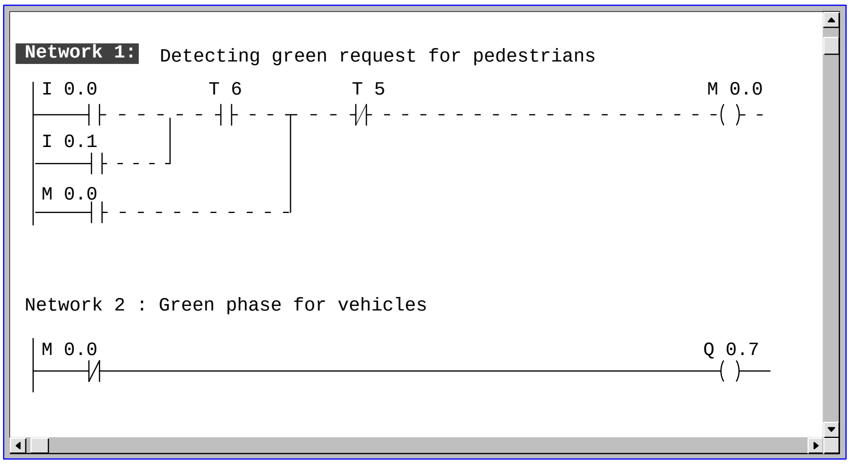
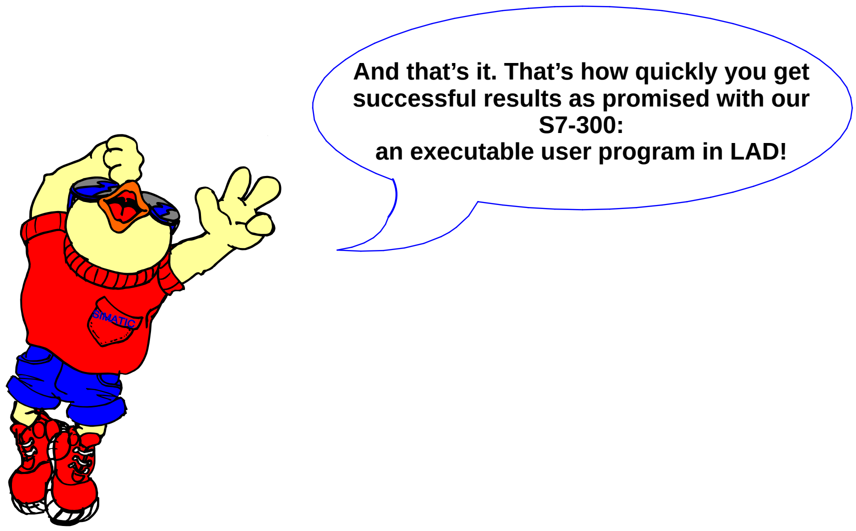


Figure 7-4 Example of Displaying the Program Status in LAD



How Do You Test Your FBD Program?

Introduction

You can test your FBD program by defining how you want to display the signal flow within the networks of a block.

Prerequisites

To display the signal flow, the following prerequisites must be fulfilled:

- You must have saved the block without errors and downloaded it to the CPU.
- The CPU is in RUN and the user program is running.
- You must open the block to be tested online.

How Do You Proceed?

The table below shows you how to test an FBD program:

Step	Procedure	Result
1.	Select the “Blocks” container in your online project “TLIGHT”. In the right-hand window, select the block you want to test (here FC1), and open it with the menu command **File** → **Open Object**.	FC1 is opened online.
2.	Select the menu command **Debug** → **Call Environment...** and click ‘No Condition’ as the trigger condition in the dialog box which follows. Close the dialog box with OK.	This means that you have selected **no** trigger condition for the call environment of the block.
3.	Select the ‘Process’ test environment via the menu command **Debug** → **Test Environment** → **Process**.	This means that the status of the statements for your program test will only be detected on the first loop run.
4.	In the ‘STL/LAD/FBD’ dialog box, select the ‘LAD/FBD’ tab with the menu command **Options** → **Customize...** and click here on the settings you want for color and line thickness. Exit the dialog box with OK.	This defines the options for the FBD program test. Current flow is displayed in the colors and line thicknesses selected.
5.	Start recording the program status with the menu command **Debug** → **Monitor**	The FBD program status is displayed in the code section of FC1 in the form of a signal flow.
6.	Stop recording the program status with the same menu command **Debug** → **Monitor**.	Display of the FBD program status is switched off again.

Displaying the Program Status in FBD

The program status is only displayed for the area visible to the editor.

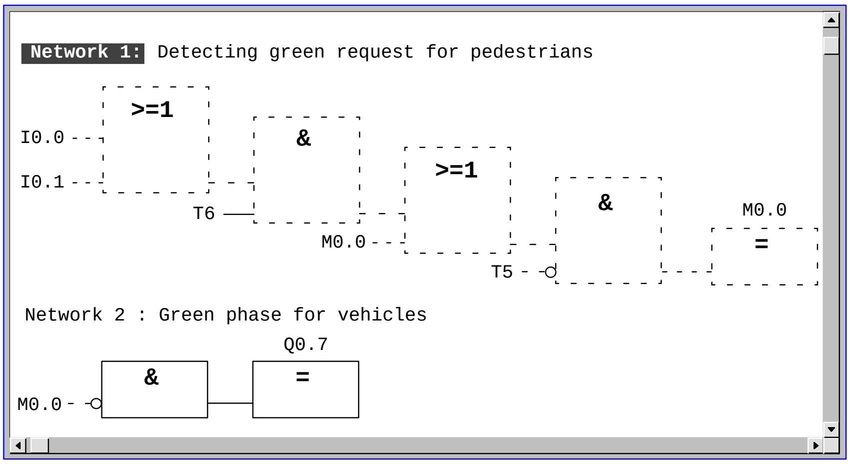
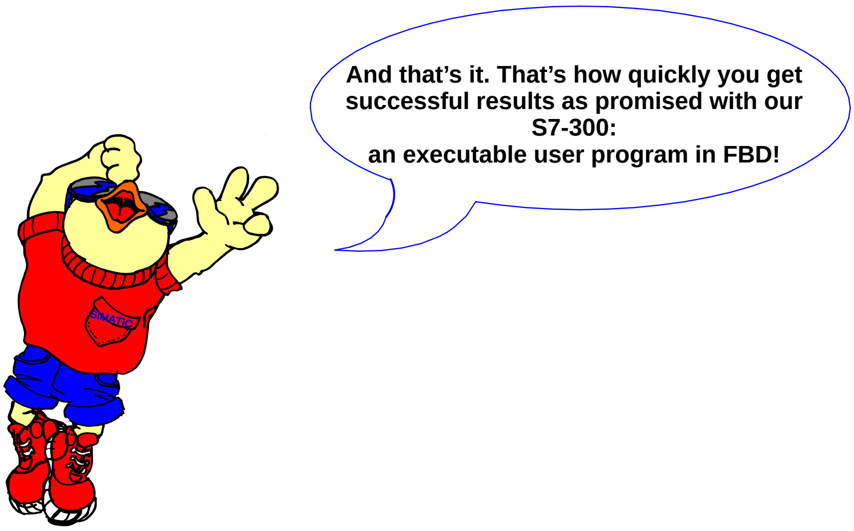


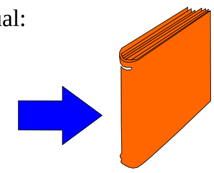
Figure 7-5 Example of Displaying the Program Status in FBD



Other References for This Chapter

Where Can You Find Other References?

If you still have questions and you need additional information on downloading and testing your user program, our manuals may be of assistance.

You Will Find Information on...	In...
how to download and test your user program	the following manual:  Standard Software for S7 and M7 STEP 7

Glossary

A

Address

An address is part of a STEP 7 statement and specifies what the processor should execute the instruction on. Addresses can be absolute or symbolic.

Automation System

An automation system is a programmable logic controller (PLC) in the case of SIMATIC S7, a SIMATIC C7 control system (PLC with integrated operator panel), or a SIMATIC M7 automation computer.

B

Backplane Bus

The backplane bus of a SIMATIC S7 programmable logic controller supplies the plugged-in modules with the internal operating voltage and enables data exchange between modules. In the S7-400, the backplane bus is divided into the I/O bus (P bus) and the communication bus (C bus). In the S7-300, the backplane bus has a modular design in the form of U rails that each connect two modules with each other.

Block

Blocks are part of the user program and can be distinguished by their function, their structure, or their purpose. STEP 7 provides the following types of blocks:

- Logic blocks (FB, FC, OB, SFB, SFC)
- Data blocks (DB, SDB)
- User-defined data types (UDT)

C

Central Processing Unit (CPU)

The CPU is the central module in the programmable controller consisting of a control unit, arithmetic unit, memory, operating system, and interfaces to the signal modules and function modules.

Complete Restart

When a CPU starts up (for example, when the mode selector is moved from STOP to RUN or when power is turned on), before cyclic program processing starts (OB1), either the organization block OB101 (restart; only in the S7-400) or OB100 (complete restart) is processed first. In a complete restart the process-image input table is read in and the STEP 7 user program processed starting with the first statement in OB1.

F

Function (FC)

According to the International Electrotechnical Commission's IEC 1131–3 standard, functions are logic blocks without static data. A function allows you to pass parameters in the user program, which means they are suitable for programming complex functions that are required frequently, for example, calculations.

I

Instruction

An instruction is part of a STEP 7 statement and specifies what the processor should do.

L

Logic Block

In SIMATIC S7, a logic block is a block that contains part of the STEP 7 user program. The other type of block is a data block which contains only data. The following list shows the types of logic blocks:

- Organization block (OB)
- Function block (FB)
- Function (FC)
- System function block (SFB)
- System function (SFC)

M

Memory Reset (MRES)

The memory reset function deletes the following memories in the CPU:

- Work memory
- Read/write area of the load memory
- System memory with the exception of the MPI parameters and the diagnostic buffer

Multipoint Interface (MPI)

The multipoint interface is the programming device interface in SIMATIC S7. It allows a number of programming devices, text display operator interfaces, and operator panels to be accessed from a central point. The nodes on the MPI can communicate with each other.

N

Network

A network consists of one or more connected subnets (for example, Industrial Ethernet, PROFIBUS, MPI) with any number of nodes. Several networks can exist side by side.

O

Organization Block (OB)

Organization blocks form the interface between the CPU operating system and the user program. The sequence in which the user program should be processed is laid down in the organization blocks.

P

Process Image

The signal states of the digital input and output modules are stored in the CPU in a process image. There is a process-image input table (PII) and a process-image output table (PIQ).

Programming Device (PG)

A personal computer with a special compact design, suitable for industrial conditions. A programming device is completely equipped for programming the SIMATIC programmable logic controllers.

Project

A project is a container for all objects in an automation task, independent of the number of stations, modules, and how they are connected in a network.

S

S7 Program

An S7 program is a container for blocks, source files, and charts for S7 programmable modules.

Startup

The CPU goes through the STARTUP mode during the transition from the STOP mode to the RUN mode. It can be set using the mode selector, or following power-on, or by an operation on the programming device.

Symbol

A symbol is a name defined by the user, taking syntax rules into consideration. This name can be used in programming and in operating and monitoring once you have defined it (for example, as a variable, a data type, a jump label, or a block).
Example: Address: I 5.0, Data Type: BOOL, Symbol: Emer_Off_Switch

T

Timer (T)

Timers are an area in the system memory of the CPU. The contents of these timers is updated by the operating system asynchronously to the user program. You can use STEP 7 instructions to define the exact function of the timer (for example, on-delay timer) and start processing it (Start).

U

User Program

The user program contains all the statements and declarations and the data required for signal processing to control a plant or a process. The program is linked to a programmable module (for example, CPU, FM) and can be structured in the form of smaller units (blocks).

Index

A

Actions with objects, 4-5
Assigning parameters, 5-2
Assigning parameters to the S7-300, 5-5

B

Blocks
  creating, 6-3
  downloading, 7-3
  programming, 6-2
  testing, 7-4

C

Code section of blocks, definition, 6-6
Configuration table, example, 5-6
Configuring, 5-2
Configuring the S7-300, 5-5
Connecting a programming device/PC, 3-6
Connecting a programming device/PC to an S7-300, 3-6
Creating a project structure, 4-6
Creating objects, 4-4

D

Downloading and testing the user program, 7-2

E

Entering
  FBD elements, 6-15
  LAD elements, 6-11
  STL statements, 6-7

F

Function Block Diagram (FBD), definition, 6-4

H

Hardware components, 1-2
Hardware requirements, 1-2

I

Installing the S7-300, 3-3, 3-4
  slot rules, 3-2

L

LAD programs testing, 7-8
Ladder Logic (LAD), definition, 6-4

M

Memory reset of the CPU, 5-4
Multipoint interface (MPI), 3-6

O

Other references
  addressing the S7-300 modules, 3-7
  configuring and assigning parameters to your S7-300, 5-7
  creating programs, 4-7
  creating projects, 4-7
  creating stations, 4-7
  defining and structuring a control task, 2-7
  downloading and testing the user program, 7-12
  hardware requirements, 1-4
  installing an S7-300, 3-7
  installing MPI network for communications, 3-7
  installing the STEP 7 software, 1-4
  memory cards for the S7-300, 3-7
  planning the installation of an S7-300, 3-7
  programming blocks, 6-18

spare parts for the S7-300, 3-7
the S7-300 modules, 3-7
wiring the S7-300, 3-7

P

Procedure
configuring and assigning parameters to the S7-300, 5-5
creating a project structure, 4-6
creating blocks, 6-3
defining and structuring your control task, 2-2
downloading a user program, 7-3
entering FBD elements, 6-15
entering LAD elements, 6-11
entering STL statements, 6-7
installing the S7-300, 3-3
memory reset of the CPU, 5-4
testing a LAD program, 7-8
testing an FBD program, 7-10
testing an STL program, 7-6
Programming blocks, 6-2
Projects with STEP 7, 4-5

R

Representing objects, 4-4
Rules
for entering FBD elements, 6-14
for entering LAD elements, 6-10
for entering STL statements, 6-7

S

Sample program, sequence, 2-6
Sample program, TLIGHT, 2-6
LAD elements for FC1, 6-12
LAD elements for OB1, 6-13
other requirements, 2-4
required addresses, 2-5
inputs, 2-5
memory, 2-5
outputs, 2-5
timers, 2-5
safety requirements, 2-4
schematic overview, 2-4
STL statements for FC1, 6-8
STL statements for OB1, 6-9
subtasks, 2-3
symbolic names of addresses, 2-5
timers defined, 2-6
Setting the test display for LAD, 7-5
Setting the test display for STL, 7-5
SIMATIC Manager, 4-3
Slot rules, 3-2
Software requirements, 1-3
Statement list (STL), definition, 6-4
STEP 7
actions with objects, 4-5
basic information, 4-3
creating a project structure, 4-6
creating objects, 4-4
objects, 4-4
projects, 4-5
representing objects, 4-4
significant features, 4-2
software package, 1-3
structuring a project, 4-5
STL program testing, 7-6
Structuring a project, 4-5

T

Testing
FBD programs, 7-10
LAD programs, 7-8
STL programs, 7-6
Testing FBD programs, 7-10
Testing the user program
defining the settings for the test display, 7-4
defining the trigger condition, 7-4
selecting the test environment, 7-4
TLIGHT sample program
FBD elements for FC1, 6-16
FBD elements for OB1, 6-17

U

User program, downloading and testing, 7-2

W

Wiring your S7-300, 3-5

Siemens AG
AUT E 146

Östliche Rheinbrückenstr. 50
D–76181 Karlsruhe
Federal Republic of Germany

From:
Your Name: _
Your Title: _
Company Name: _
Street: _
City, Zip Code _
Country: _
Phone: _

Please check any industry that applies to you:

- ❒ Automotive
- ❒ Chemical
- ❒ Electrical Machinery
- ❒ Food
- ❒ Instrument and Control
- ❒ Nonelectrical Machinery
- ❒ Petrochemical
- ❒ Pharmaceutical
- ❒ Plastic
- ❒ Pulp and Paper
- ❒ Textiles
- ❒ Transportation
- ❒ Other _ _ _ _ _ _ _ _ _ _ _

Remarks Form

Your comments and recommendations will help us to improve the quality and usefulness of our publications. Please take the first available opportunity to fill out this questionnaire and return it to Siemens.

Please give each of the following questions your own personal mark within the range from 1 (very good) to 5 (poor).

1. Do the contents meet your requirements? ☐
2. Is the information you need easy to find? ☐
3. Is the text easy to understand? ☐
4. Does the level of technical detail meet your requirements? ☐
5. Please rate the quality of the graphics/tables: ☐

Additional comments:

_ _

_ _

_ _

_ _

_ _

_ _

_ _

_ _

_ _

_ _

_ _

_ _