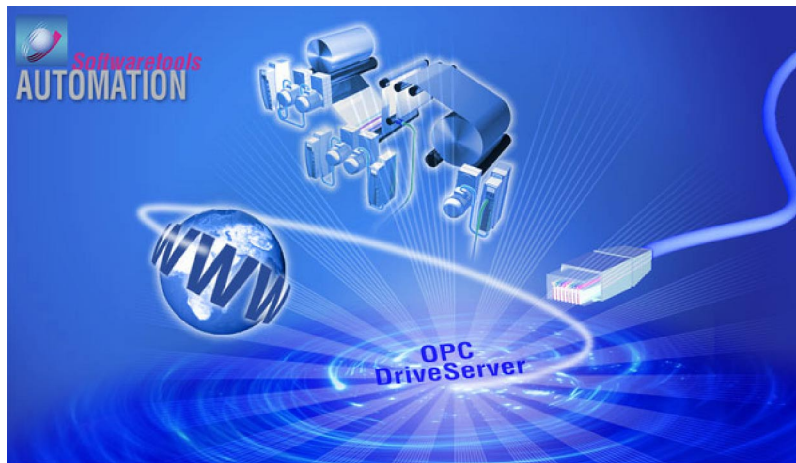


EDBDRS3  
00472714



# Lenze

## *Handbuch/Manual*



## ***DriveServer***



### ***Global Drive***

### ***Busserver S7***

*Erste Schritte/Getting Started*



**CAUTION:**

The software is supplied to the user as described in this document. Any risks resulting from its quality or use remain the responsibility of the user. The user must provide all safety measures protecting against possible maloperation.

We do not take any liability for direct or indirect damage, e.g. profit loss, order loss or any loss regarding business.

Version	ID no.	Changes
1.0 08/2003 TD11	00472714	First edition

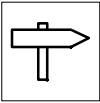
© 2003 Lenze Drive Systems GmbH

No part of this documentation may be copied or made available to third parties without the explicit written approval of Lenze Drive Systems GmbH.

All information given in this documentation has been carefully selected and tested for compliance with the described hardware and software. Nevertheless, discrepancies cannot be ruled out. We do not accept any responsibility or liability for any damage that may occur. Required corrections will be included in the updates of this documentation.

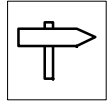
Microsoft, Windows and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the U.S.A and/or other countries.

All other product names mentioned in this documentation are trademarks of the corresponding owners.



## Contents

<b>1</b>	<b>Preface and general information</b>	<b>66</b>
1.1	Conventions used	66
1.2	Layout of the safety information	67
1.3	Scope of delivery	67
<b>2</b>	<b>System requirements</b>	<b>68</b>
2.1	Connection to the PLC	68
2.2	Supported communication channels	69
2.3	Supported PLC hardware	71
<b>3</b>	<b>Software installation</b>	<b>72</b>
3.1	Component selection	73
<b>4</b>	<b>Software configuration</b>	<b>74</b>
4.1	Interface module configuration	74
4.2	Bus server configuration	75
4.2.1	User interface of the S7 configuration tool	75
4.2.2	Automatic PLC scanning	76
4.2.3	Manual addition of PLCs	77
4.2.4	PLC parameters	78
4.2.5	Automatic drive scanning	79
4.2.6	Manual addition of drives	80
4.2.7	Drive parameters	80
4.2.8	Configuration of sub-systems	81
4.2.9	Manual deletion of configuration entries	83
4.2.10	Save configuration & exit bus server	83
<b>5</b>	<b>Creating a STEP7 project</b>	<b>84</b>
5.1	Overview of the STEP7 blocks	84
5.2	Typical program flow	85
5.3	Files provided	86
5.3.1	STEP7 library	86
5.3.2	STEP7 example programs	87
5.3.3	Device description files for the Lenze drives	87
5.4	Creating new projects	88
5.5	Hardware configuration	89
5.5.1	Configuring the MPI interface of the CPU	89
5.5.2	Configuring the PROFIBUS interface of the CPU	89
5.5.3	Configuring additional hardware components	90
5.6	Creating a STEP7 program	91
5.6.1	Accessing a drive via OPC	91
5.6.2	Accessing several drives via OPC	93
5.6.3	Accessing drives connected to PROFIBUS CPs	93
5.6.4	Accessing process and parameter data from the PLC	94



5.7	Testing the STEP7 program	95
5.7.1	Structure of the DriveServer name area	95
5.8	Optimisation of the cycle time	96
5.9	Accessing PLC data	97
<b>6</b>	<b>Appendix</b>	<b>98</b>
6.1	Time load of the PLC cycle	98
6.2	Transfer times	98
6.3	Communication function parameters	100
6.3.1	FB90 (FB DRIVE)	100
6.3.2	FB91 (FB DRIVE CP)	101
6.3.3	FC90 (FC Parameter)	102
6.3.4	FC91 (FC Process Data, 3 words)	104
6.3.5	FC92 (FC Process Data, 2 words)	105
6.3.6	FC93 (FC Process Data, 3 words, CP)	106
6.3.7	FC94 (FC Process Data, 2 words, CP)	107
6.4	Remote maintenance via modem	108
6.5	Example project S7-400	109
6.5.1	Hardware configuration	109
6.5.2	Program blocks used	110
6.5.3	Symbolic variable names	111
6.5.4	OB1/FC4	112
6.5.5	Process data transfer	112
6.5.6	Parameter data access	113
6.5.7	Reading a parameter	114
6.5.8	Writing a parameter	115
6.6	Example project S7-300 with PROFIBUS-CP	116
6.6.1	Hardware configuration	116
6.6.2	Program blocks used	117
6.6.3	Symbolic variable names	117
6.6.4	OB1/FC3	118
6.6.5	Process data transfer	119
6.6.6	Parameter data access	120
6.6.7	Reading/writing a parameter	121
6.7	FAQ - Frequently asked questions and answers	122



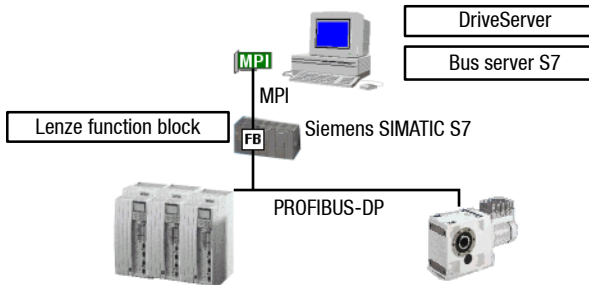
## Busserver S7

# 1 Preface and general information

The **DriveServer** is a software which provides easy integration of drives into open automation structures based on OPC (OLE for Process Control).

When used together with the **bus server S7**, you can access drives which are connected via PROFIBUS-DP and controlled via a Siemens SIMATIC S7 PLC.

The following figure shows the architecture of such a system:

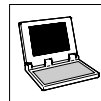


- A bus system (MPI, Ethernet or PROFIBUS) is used to connect the PC with the **DriveServer** to the PLC. The DriveServer uses the **bus server S7** as OPC bus server.
- The S7 program uses special Lenze function blocks via which the DriveServer can access the drives through the PLC as if the PLC was not connected.
- The drives are connected to the PROFIBUS master of the PLC via PROFIBUS-DP.

## 1.1 Conventions used

This Manual uses the following conventions to distinguish between different types of information:

Type of information	Marking	Example
Names of dialog boxes	<i>italics</i>	The Options dialog box...
Parameter identifiers		The parameter PB_Address...
Control elements (buttons, input fields, checkboxes, etc.)	<b>bold</b>	Click the OK button to...
STEP7 block identifier		The data block DB80...
Menu commands		Use the command <b>Copy</b> to ...
		If the execution of a function requires several commands, the individual commands are separated by an arrow: Select <b>File→Open</b> to...
Keyboard commands	<b>&lt;bold&gt;</b>	Use <b>&lt;F1&gt;</b> to open the Online Help. If a command requires a combination of keys, a "+" is placed between the key symbols: Use <b>&lt;Shift&gt;+&lt;ESC&gt;</b> to ...
Program listings	Courier	TimerNr := T1
Keywords	<b>Courier bold</b>	



## 1.2 Layout of the safety information





All safety information have a uniform layout:

- The icon characterises the type of danger.
- The signal word characterises the severity of danger.
- The note describes the danger and suggests how to avoid the danger.



### Signal word

Note

Icon	Signal word	Meaning	Consequences if disregarded
 hazardous electrical voltage	<b>Danger!</b>	<b>Impending danger to persons</b>	Death or severe injuries
 general danger	<b>Warning!</b>	<b>Potential, very hazardous situation for persons</b>	Death or severe injuries
	<b>Caution!</b>	<b>Potential, hazardous situation for persons</b>	Light injuries
	<b>Stop!</b>	<b>Potential damage to material</b>	Damage to the drive system or its environment
	<b>Note!</b> <b>Tip!</b>	<b>Note or tip</b>	

## 1.3 Scope of delivery

Packing list <sup>1</sup>	Important
<ul style="list-style-type: none"> <li>• 1 CD ROM "DriveServer with bus server S7"</li> <li>• This Manual</li> <li>• Manual "DriveServer - Getting Started"</li> <li>• Supplement "Licence and contract terms"</li> </ul>	<p>After receipt of the delivery, check immediately whether the items delivered match the accompanying papers. Lenze does not accept any liability for deficiencies claimed subsequently.</p> <p>Claim</p> <ul style="list-style-type: none"> <li>• visible transport damage immediately to the forwarder.</li> <li>• visible deficiencies/incompleteness immediately to your Lenze representative.</li> </ul>
<p><sup>1</sup> The following notes about the scope of supply replace the notes about the scope of supply provided in the manual "DriveServer - Getting Started".</p>	



## Busserver S7

## 2 System requirements

The following minimum requirements on hardware and software must be met in order to use the bus server S7:

- Microsoft® Windows® 98/Me, Windows NT® oder Windows 2000/XP
- IBM compatible PC with Intel® Pentium®-90 processor
- 64 MB Random Access Memory (RAM)  
(128 MB RAM with Windows 2000/XP)
- 80 MB free hard disk capacity
- CD-ROM drive

Please also observe the system requirements of the interface module!  
(See the following sub-section.)



### Tip!

We recommend to use a Windows NT, Windows 2000 or Windows XP operating system!

- With the same hardware, the program execution times are much shorter under Windows NT/2000/XP than under Windows 98/Me.
- When several computers are interconnected, it is not possible to start OPC servers automatically through a DCOM connection with Windows 98/Me.

## 2.1 Connection to the PLC

The PC must be equipped with a suitable interface module to communicate with the PLC.

- The following interface modules are available for MPI/PROFIBUS:

Manufacturer	Designation	Article No.	Version/description
Siemens	CP5611	6GK1561-1AA00	PCI card (32-bit) to connect a PU or PC with PCI bus to PROFIBUS or MPI.
	CP5511	6GK1551-1AA00	PCMCIA card (16-bit) to connect a PU or notebook to PROFIBUS or MPI.
	PC adapter	for connection to S7-300/-400 for:	
	RS-232/485	6ES7972-0CA23-0XA0	RS-232 → MPI or PROFIBUS
	USB	6ES7972-0CB20-0XA0	USB → MPI or PROFIBUS
Helmholz	SSW7	700-751-1VK11	RS-232 → MPI or PROFIBUS
IBH Softec	IBHLink	2026	Adaptor cable Ethernet → MPI

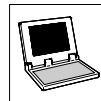
- It is possible to use standard network cards for Ethernet connections.



### Note!

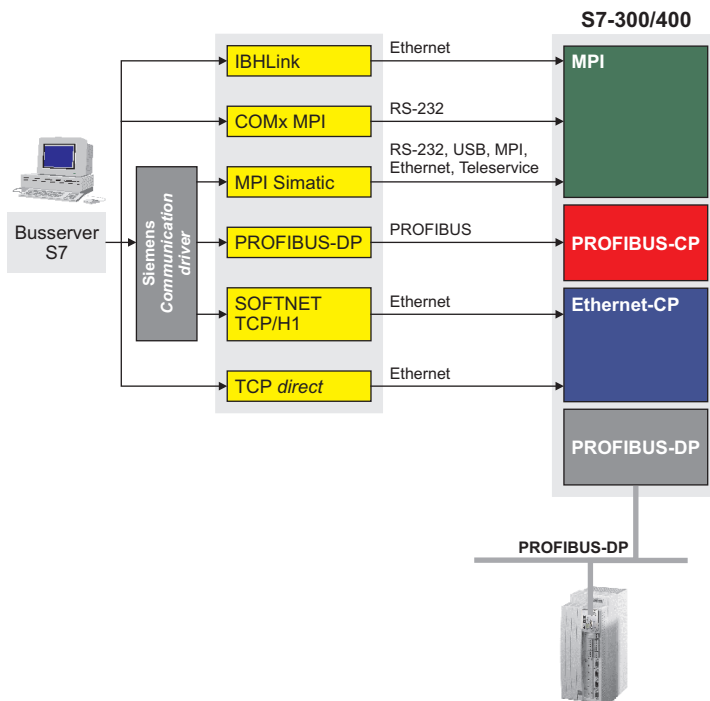
Please observe the Installation/Commissioning Instructions for the interface module given by the manufacturer.

More information on the configuration of the interface module can be found in chapter 4.1. (□ 74)



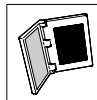
## 2.2 Supported communication channels

The bus server S7 supports the following communication channels:



### Note!

Detailed information on the supported communication channels and additional hardware/software requirements can be found in the table on the next page.



Communication channel	PC interface	PC interface module	Transmission medium	PLC interface module	PLC interface	Additional software required
IBHLink	Ethernet	-	Ethernet	IBHLink adaptor cable Ethernet/MPI	MPI	IBHNet (V1.1 or higher)
COMxMPI	COMx	-	RS-232	PC adapter RS-232/485		-
MPI Simatic	USB	-	USB	PC adapter USB		Siemens communication driver (V5.0 or higher)
	COMx	-	RS-232	PC adapter RS-232/485		
	PCI	Siemens CP5611	MPI	-		
	PCMCIA	Siemens CP5511	-	-		
Modem			Teleservice	Siemens Teleservice adapter	+ Siemens teleservice (V5.0 or higher)	
Ethernet	-	Ethernet	IBHLink adaptor cable Ethernet/MPI	+ IBHNet (V1.1 or higher)		
PROFIBUS DP	COMx	-	RS-232	PC adapter RS-232/485	PROFIBUS-CP	Siemens communication driver (V5.0 or higher)
	USB	-	USB	PC adapter USB		
	PCI	Siemens CP5611	PROFIBUS	-		
	PCMCIA	Siemens CP5511	-	-		
SOFTNET TCP/H1	Ethernet	-	Ethernet TCP/H1	-	Ethernet-CP	Siemens SOFTNET/SIMATICNET (V3.0 or higher)
			Ethernet TCP	-		
TCP direct						-

Notes	
COMxMPI	"x" stands for the number of the COM port. Use "MPI Simatic" as communication channel if the Siemens communication driver (V5.0 or higher) is installed on your PC or simultaneous access with Lenze and Siemens software is required.
SOFTNET TCP/H1	Under normal circumstances, the TCP protocol is used. When using older CPs it may happen that only the Siemens H1 protocol is supported. This is why the bus server S7 also supports the H1 protocol.
TCP direct	If the corresponding Siemens software is installed on your PC, we recommend to use "SOFTNET TCP/H1" as communication channel. Also use "SOFTNET TCP/H1" as communication channel if simultaneous access with Lenze and Siemens software is required.



## 2.3 Supported PLC hardware

The following PLC hardware is supported: Siemens S7-300 and S7-400

- For S7-400, the controllers must be connected to the PROFIBUS of the CPU.
- For S7-300, the controllers can also be connected to PROFIBUS CP.
- If an Ethernet connection is to be established between PC and PLC, the PLC must be equipped with an Ethernet CP (e. g. CP343-1, CP443-1).
- If a PROFIBUS connection is to be established between PC and PLC, the PLC must be equipped with a PROFIBUS CP (e. g. CP343-5, CP443-5).

### Required PLC data blocks/RAM requirements

Two data blocks are required for data exchange with the drives. In addition, one function block instance is to be activated for each drive to be addressed. This means that one data block per drive (instance DB) is required for data management.

The RAM requirements of the PLC can be obtained from the following table:

Function block type	RAM requirements	Number of function blocks required
Data exchange DB	508 bytes	2
Instance DB	150 bytes	Number of drives



### Note!

The number of PLCs which can be simultaneously addressed depends on the interface module of the PC or PLC.

- The Siemens PC adapter supports only 4 communication links. Therefore, only 4 PLCs can be addressed simultaneously.

For more information, please read the documentation on the interface module provided by the manufacturer.



## Busserv S7

### 3 Software installation

Proceed as follows to install the bus server software on your PC:

1. Start Windows.
2. Insert the "DriveServer with bus server S7" CD-ROM into your CD-ROM drive.  
If the auto-start function of your CD-ROM drive is activated, the installation program will be started automatically. If so, proceed with step 5.
3. Select **Run...** from the start menu.
4. Enter the letter for your CD-ROM drive followed by "\setup.exe" (e.g. "e:\setup.exe") and confirm the entry by clicking **OK**.
5. Follow the instructions of the installation program.



#### Note!

##### Windows NT/2000/XP

Installation under Windows NT/2000/XP requires administrators rights!

##### Windows 98

DCOM is not installed as standard in Windows 98 and must be installed separately, if necessary.

- The bus server installation program checks if DCOM has been installed and suggests to install DCOM if DCOM is not available.
- The bus server software cannot be installed without DCOM.

#### Tip!

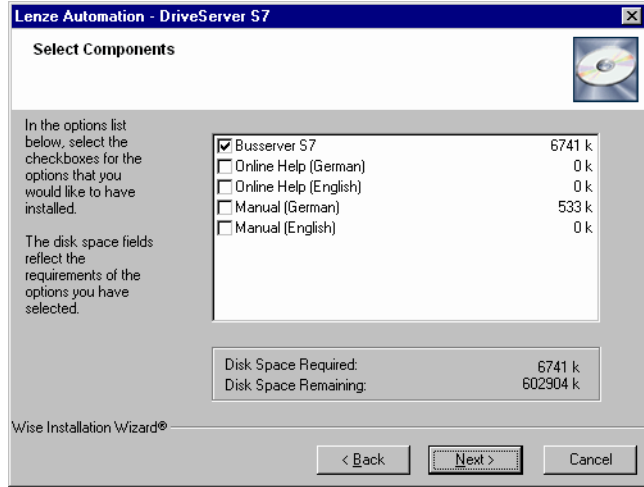
For notes about the DCOM configuration, please refer to the manual "DriveServer - Getting Started."



### 3.1 Component selection

From the installation program of the DriveServer with bus server S7 you can select, which components you want to install on your PC. Selection is divided into two steps:

1. Selection of the DriveServer components. (See manual "DriveServer - Getting Started").
2. Selection of the bus server S7 components:



Components	Info
Bus server S7	Driver for communication with Siemens SIMATIC S7 PLC
Online Help (German/English)	Online Help for the program
Manuals (German/English)	Manuals for the program in Adobe® Acrobat® format (PDF)



## Busserver S7

# 4 Software configuration

## 4.1 Interface module configuration

If you want to use "MPI Simatic", "PROFIBUS-DP" or "SOFTNET TCP/H1" as communication channel, the corresponding interface module must be configured in order to enable communication between the bus server and the PLC(s).

The communication parameters for all other communication channels are directly selected in the bus server configuration tool!

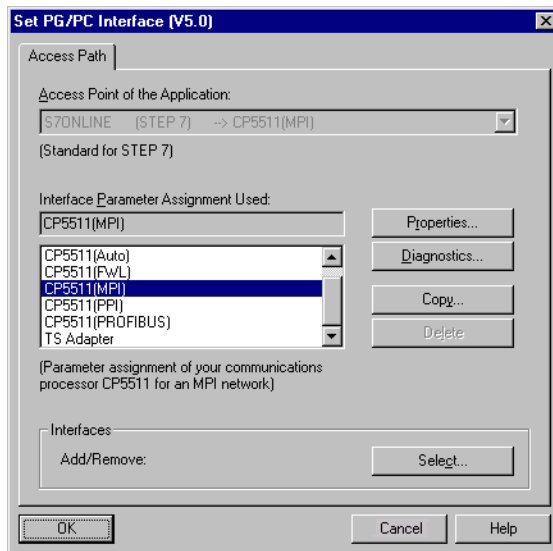


### Note!

The configuration program for the interface module is included in the Siemens communication drivers which are automatically installed when installing the STEP 7 configuration software or a number of other Siemens software products.

The configuration program is available in the Windows control panel:

1. Select **Settings**→**Control panel**.
2. Double-click the symbol **Set PG/PC interface** to open the *Set PG/PC interface* dialog box:



3. Configure the interface according to your system environment and confirm the settings by clicking **OK**.



## 4.2 Bus server configuration

The "S7 configuration tool" program is used for the configuration of the bus server. Upon configuration, the PLC and the drives configured in the respective PLC program are selected:

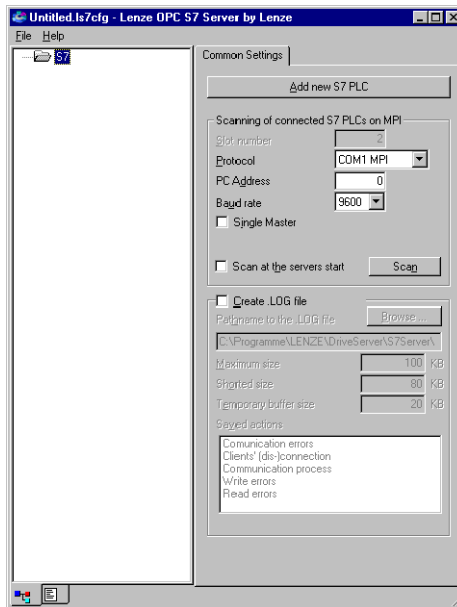
1. Select **Programs→Lenze→DriveServer→S7 configuration tool**.
2. Confirm the *Lenze OPC S7 Server* info dialog box with **OK**.
  - The configuration saved last will be loaded automatically.
  - If you want to create a new configuration, select **File→New**.

### 4.2.1 User interface of the S7 configuration tool

The S7 configuration tool is subdivided into two sections:

- The left side indicates the **configuration entries** in form of a tree, similar to the directories of the Microsoft Explorer.
- The right side shows all **Settings** for the configuration entry selected from the tree.

If you create a new configuration, the first visible entry will be "S7":





## Busserv S7



### Tip!

If a plus sign appears in front of the element, this element has sub-elements:

	<p>A click on the plus sign opens the list of sub-elements. A minus sign appears instead of the plus sign.</p>
	<p>Clicking the minus sign closes the list of sub-elements.</p>

### 4.2.2 Automatic PLC scanning

First, the connected PLCs have to be indicated. If "MPI Simatic", "PROFIBUS-DP" and "COMxMPI" are used for communication, you can use the scan function available in the S7 configuration to do so.

- For all other communication channels and as an alternative to using the scan function you can configure the PLC manually. (77)



### Tip!

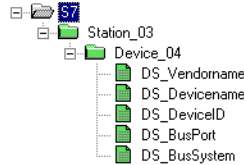
The scan function scans the MPI network for connected PLCs and for the drives configured in the respective PLC program.

- Configuration of the bus server is only possible, when the PLC program has been completed.
- To ensure that the drives configured in the PLC program can be detected by the scan function, the PLC must have been at least once operated in RUN mode with the completed PLC program. Only in this case, the instance data – i. e. also the PROFIBUS address – has been saved in the online data blocks!
- It might be useful to perform the scanning already earlier to check the connection.

1. Use the text field **CPU Slot Number** to enter the PLC slot into which the CPU card is plugged.
  - The slot position of the CPU card depends on the PLC type and the power supply used for the PLC.
2. Use the list field **Communication** to select the communication channel.
  - If your interface module supports several communication channels and the Siemens communication drivers have been installed on your PC, select the communication channel which uses the Siemens communication drivers, if possible.
3. Click **Scan** to start scanning.
  - The scanning progress is indicated in a dialog box.
  - If several CPUs are connected to the bus and the CPU cards are plugged into different slots, the "complete scanning" must be repeated several times with the corresponding setting in the text field **CPU Slot Number**.



The PLCs found and the drives configured in the PLC program are listed as configuration entries in the tree structure:



### Tip!

The S7 configuration tool automatically creates several OPC items for each drive, which allow the DriveServer to identify the drive automatically.

- For special applications, more OPC items can be added manually. In general, however, the DriveServer creates the OPC items.

If the **Scan at the Server's Start of Comm.** checkbox is activated, the PLC and the drives configured in the PLC program will be detected every time the bus server is started.

- This option is useful, if the bus server is connected to a portable PC which is operated at different systems.
- If the bus server is operated at only one system, this option should be de-activated to speed up starting the bus server.
- If several CPUs are connected to the bus:
  - Only one CPU slot number is taken into account for scanning when the bus server is started.
  - If the CPU cards use different slot positions, this option is not useful since it is not possible to find all PLCs.

## 4.2.3 Manual addition of PLCs

For the "SOFTNET TCP/H1" and "TCP communication channel and as an alternative to using the scan function you can add the connected PLCs manually to the configuration.

1. Click **Add New S7 PLC** to add a PLC to the configuration.
2. Use the **Name** input field to enter a name for the PLC and confirm with **OK**.
  - The name of the new PLC may only contain letters. It must not contain any special characters.

The PLC will be added with the entered name as a configuration entry to the tree structure, and you can now configure its parameters (communication channel, address, CPU slot number, etc.) in the text fields on the right.



## Busserver S7

### 4.2.4 PLC parameters

The below table contains all parameters which are relevant for the PLC configuration. The parameters which can be set depend on the selected communication channel.

Parameter name	Meaning
Communication channel	Selection of the connection between PC and PLC. <ul style="list-style-type: none"> <li>See chapter 2.1, "Connection to the PLC".  68</li> </ul>
Response time	Maximum time in [s] within which the PLC is expected to respond. <ul style="list-style-type: none"> <li>Unless the PLC responds within the selected time, it is likely that a communication error has occurred and an error message will be indicated.</li> <li>Normally, the default settings need not be changed.</li> </ul>
PLC address <sup>1)</sup>	Address of the PLC connected to the PC. <ul style="list-style-type: none"> <li>When using an MPI/PROFIBUS connection, you only have to select the MPI/PROFIBUS address.</li> <li>When using an Ethernet connection, you have to select the MPI address and the IP or H1 address.</li> </ul>
CPU slot number <sup>2)</sup>	PLC slot into which the CPU card is plugged. <ul style="list-style-type: none"> <li>Default: 2</li> </ul>
Mounting rack number <sup>2)</sup>	Mounting rack which includes the CPU card of the PLC. <ul style="list-style-type: none"> <li>Default: 0</li> </ul>
PC-MPI address <sup>1) 3)</sup>	MPI or PROFIBUS address of the PC
Baud rate <sup>3)</sup>	MPI bus transmission rate between PC and PLC
Max. MPI address <sup>3)</sup>	Highest device address on MPI bus
Only master <sup>3)</sup>	Select this checkbox, if the PC is the only master on the bus. Otherwise, the system might stop if a token is lost in case of a communication error.
Routing in MPI/TCP/H1 sub-networks	Selection of the sub-network type if the selected communication channel allows routing in sub-systems. <ul style="list-style-type: none"> <li>For communication channels and PLCs which support the routing function.  81</li> </ul>
Sub-network ID	Identification of the sub-network.  81
Target MPI/TCP/H1 address	Address of the PLC in the sub-system.  81
DB number of the OPC list <sup>1)</sup>	Number of the data block which is used for data exchange between PC and PLC. <ul style="list-style-type: none"> <li>This number is detected by the scan function.</li> <li>Unless the numbers of the enclosed Lenze blocks are changed, enter "80" for DB80.</li> </ul>
Maximum number of devices to be found.	To save time, in particular, when scanning the PLC program for connected devices at the server's start, it is possible to enter the maximum drive address to be checked. <ul style="list-style-type: none"> <li>If "0" is selected, all drive addresses will be scanned (default).</li> </ul>
Scan at Server Start of Comm.	The PROFIBUS addresses are automatically detected when the bus server is started and need not be configured manually.
<sup>1)</sup> This parameter is automatically set during automatic PLC scanning. <sup>2)</sup> When using sub-systems, this parameter refers to the PLC in the sub-system and not to the master PLC.  81 <sup>3)</sup> This parameter is only valid for the "COMx MPI" communication channel.	



## 4.2.5 Automatic drive scanning

If the PLC is connected to the PC and the PLC program has already been transmitted to the PLC, you can automatically scan the drives configured in the PLC program, unless they have already been identified during automatic PLC scanning.

1. Select the corresponding PLC from the configuration tree.
2. Ensure that the PLC has been at least once operated in RUN mode with the completed PLC program. Only in this case, the instance data – i. e. also the PROFIBUS address – has been saved in the online data blocks!
3. Click **Scan** to start scanning for drives.
  - The scanning progress is indicated in a dialog box.
  - The detected drives will be listed as configuration entries in the tree structure below the selected PLC.
  - The scan function automatically detects the number of the data block **DB OPC** which is used to exchange data between the PC and the PLC.



### Tip!

The S7 configuration tool automatically creates several OPC items for each drive, which allow the DriveServer to identify the drive automatically.

- For special applications, more OPC items can be added manually. In general, however, the DriveServer creates the OPC items.

If the **Scan at the Server's Start of Comm.** checkbox is activated, the drives configured in the PLC program will be detected every time the bus server is started.

- This option is useful, if the bus server is connected to a portable PC which is operated at different systems.
- If the bus server is operated at only one system, this option should be de-activated to speed up starting the bus server.



## Busserver S7

### 4.2.6 Manual addition of drives

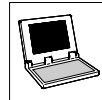
As an alternative to using the scan function, you can add the drives manually to the configuration.

1. Select the PLC to which you want to add a device from the tree structure.
2. Click **Add device**.
3. Use the **Name** input field to enter a name for the device and confirm with **OK**.
  - The name of the new device may only contain letters. It must not contain any special characters.

The device will be added with the entered name as a configuration entry to the tree structure, and you can now configure its parameters in the text fields on the right.

### 4.2.7 Drive parameters

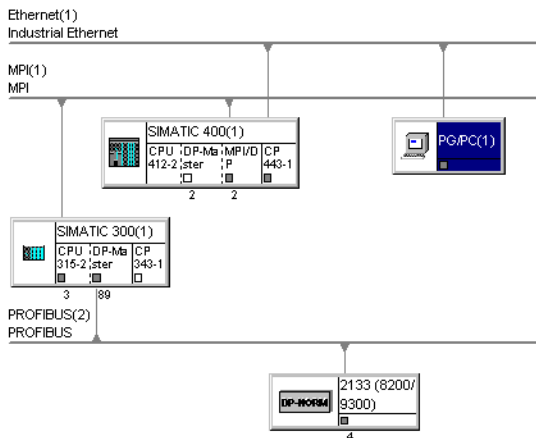
Parameter name	Meaning
DP Address of the Device	PROFIBUS address
Timeout	<p>Maximum time in [s] the bus server will wait for the processing of a parameter command.</p> <ul style="list-style-type: none"> <li>• When the parameter sets are transferred, many commands are simultaneously transmitted. Depending on the cycle time of the CPU, waiting times between 20 - 30 s may therefore occur.</li> <li>• If problems occur during parameter set transfer, try a higher timeout setting first.</li> </ul>



## 4.2.8 Configuration of sub-systems

If the PLC connected to the PC includes a "routing function", it is possible to access controllers which are not connected to the PROFIBUS of the (master) PLC, but to the PROFIBUS of a lower-level PLC.

The below figure shows an example of such an architecture:



- In this example, the SIMATIC 400 is the master PLC and directly connected to the PC via Ethernet.
- The SIMATIC 300 is used as sub-system to which a controller is connected via PROFIBUS.
- The master PLC and the sub-system are connected via an MPI connection.
- Please consider that both the PLCs and the PC have to be added to the STEP 7 project.
  - Use the "NetPro" STEP 7 service program to do so.

### Configuration steps required

The following configuration steps have to be carried out to access the controller connected to the sub-system (SIMATIC 300):

1. Add the master PLC manually to the configuration. (□ 77)
2. Select the communication channel and enter the address of the master PLC.
3. Enter the CPU slot number and the mounting rack number of the sub-system.
4. Select the sub-network type (here: MPI).
5. Enter the sub-network ID (see tip on the next page).
6. Enter the address of the sub-system (MPI, TCP or H1 address).

Now you can access the sub-system. Using the scan function you can find out under which addresses controllers have been configured in the PLC program of the sub-system.



## Busserver S7

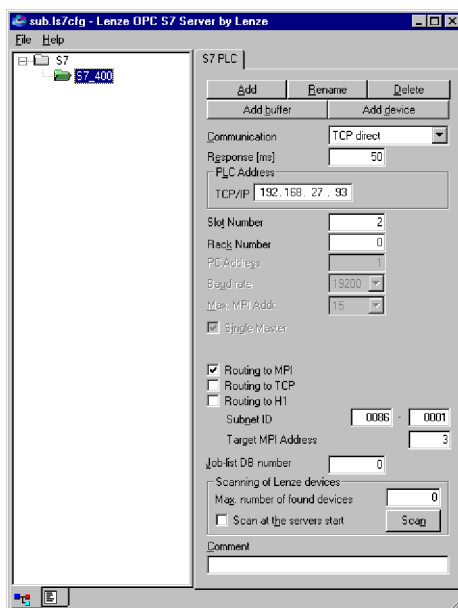


### Tip!

Proceed as follows to identify the sub-network ID in the STEP 7 program:

1. Start the "NetPro" service program, e. g. by double-clicking a network in the SIMATIC manager or clicking "Connections" when the CPU is focussed.
2. Right-click the network connecting the controls with each other and select the **Object Properties** command from the context menu.
3. Enter the sub-network ID indicated in the **Object Properties** dialog box in the Lenze S7 configuration tool.

The below figure shows an example of a possible configuration:



### Tip!

If controllers are connected to the PROFIBUS of the master PLC and the PROFIBUS of the sub-system, two PLCs must be configured in the Lenze S7 configuration tool.

- In this case, the master PLC data are identical for both configuration entries.
- For the sub-system PLC, you also have to enter the sub-system data as described above.



## 4.2.9 Manual deletion of configuration entries

You can always delete configuration entries (PLCs, devices, OPC items, etc.) from the tree structure.

1. Select the configuration entry which you want to delete including all sub-configuration entries from the tree structure.
2. Click **Delete**.
  - Before deletion, you will be prompted if you really want to delete the configuration entry.
3. Confirm with **Yes** to delete the configuration entry from the tree structure or click **No** to cancel deletion.

## 4.2.10 Save configuration & exit bus server

When the configuration is complete, you can save it and exit the bus server:

- Select **File→Save** to save the current configuration.
- Select **File→Exit** to end the S7 configuration tool.

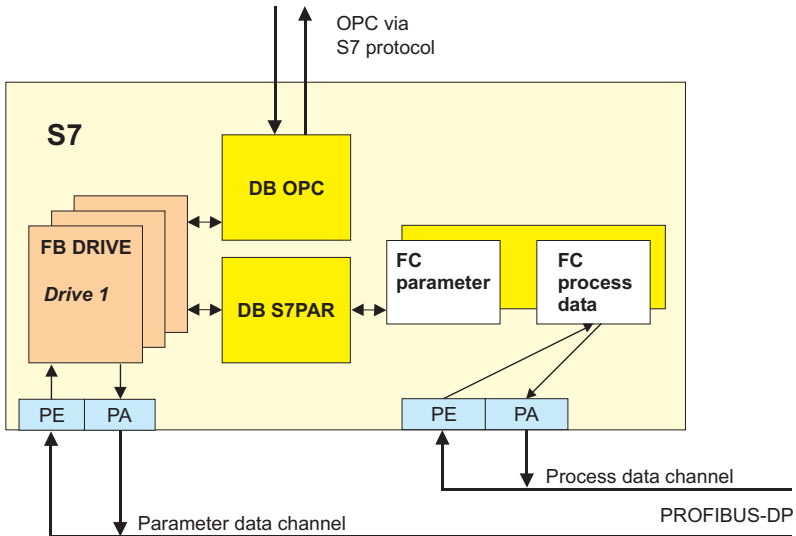


## Busserver S7

# 5 Creating a STEP7 project

## 5.1 Overview of the STEP7 blocks

The following figure gives you an overview of the different STEP7 blocks:



Symbolic block name	Block number	Function
FB DRIVE	FB 90	Call one instance of <b>FB DRIVE</b> for each drive. Allocate one instance data block to each instance. Each instance checks cyclically whether a command for the corresponding drive is available in the data block <b>DB OPC</b> or <b>DB S7PAR</b> and executes this command, if necessary.
DB OPC	DB 80	The data block <b>DB OPC</b> is used for data exchange between the bus server and the drives. A list of the commands for the individual drive controllers is saved in <b>DB OPC</b> .
DB S7PAR	DB 81	The data block <b>DB S7PAR</b> is used to write parameter data from the PLC. The function <b>FC Parameter</b> is used to write the parameter data jobs to <b>DB S7PAR</b> .

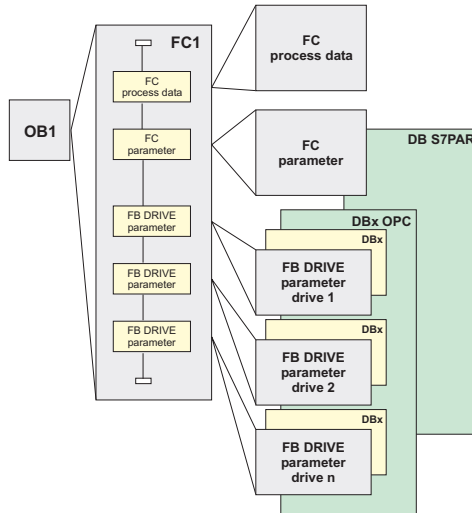
In this description, symbolic names are used for the blocks. If you want to use these names instead of the individual block numbers in the STEP7 project, you first have to create them in the symbol table.

In the following sub-chapter, this program structure will be explained in detail.



## 5.2 Typical program flow

The following figure shows the typical sequence of a STEP7 program with OPC access:



- The standard block **OB1** forms the basis. This block is included in every STEP7 program and automatically activated in every cycle. The **OB1** calls all other function blocks required for the particular application.
- In the above figure, the standard block **OB1** calls the function **FC1** which initiates the real access to the drives.
- In every cycle, the function **FC Process Data** is called in **FC1** to process the process data. Since it is not possible to access process data via OPC, you need not change between PLC and OPC commands in this case.
- The function block **FB DRIVE** is called several times for OPC access. For each drive, you have to create an instance with a separate data block.
- Each instance of **FB DRIVE** has access to two data blocks, **DB OPC** and **DB S7PAR**.
- Every time **FB DRIVE** is called (multiple use), a special instance data block has to be transmitted, which contains the data for the appropriate drive, e. g. the PROFIBUS address. The **FB DRIVE** does not use any global data.
- **DB OPC** and **DB S7PAR** must not be changed by the user. The user can, however, write parameter data jobs from the PLC to the corresponding fields of **DB S7PAR** using the function **FC Parameter**.
- The function **FC Parameter** can be called at any point in **FC1**.
- The change between the parameter data jobs from the OPC and the PLC is made by the **FB DRIVE**.
- The function **FC Process Data** is not affected by this and continues to be called in every cycle.



## Busserver S7

### 5.3 Files provided

#### 5.3.1 STEP7 library

The individual blocks are available in form of a library and can be used in every STEP7 program. The following table gives you an overview of the blocks available in the library:

Block number	Symbolic block name	Function	Version
DB30		Data block for data exchange with an external PROFIBUS communication processor (CP) <ul style="list-style-type: none"> <li>When using UDT93 (send).</li> </ul>	
DB31		Data block for data exchange with external PROFIBUS communication processor (CP) <ul style="list-style-type: none"> <li>When using UDT93 (receive).</li> </ul>	
DB80	DB OPC	Data block for parameter data jobs via OPC.	
DB81	DB S7PAR	Data block for parameter data jobs from the PLC.	
FB90	FB DRIVE	Function block <ul style="list-style-type: none"> <li>For every drive one instance is to be created.</li> </ul>	2.3
FB91	FB DRIVE CP	Function block <ul style="list-style-type: none"> <li>For every drive one instance is to be created.</li> </ul>	1.0
FC90	FC Parameter	Function for parameter data jobs from the PLC.	1.2
FC91	FC Process Data	Function for process data access (3 process data words)	1.2
FC92	FC Process Data	Function for process data access (2 process data words)	1.2
FC93	FC Process Data CP	Function for process data access (3 process data words) <ul style="list-style-type: none"> <li>When using an external PROFIBUS communication processor (CP)</li> </ul>	1.3
FC94	FC Process Data CP	Function for process data access (2 process data words) <ul style="list-style-type: none"> <li>When using an external PROFIBUS communication processor (CP)</li> </ul>	1.3
UDT90 ... UDT93		Universal data type for parameter data access to Lenze drive controllers.	

When the "DriveServer with bus server S7" software is installed, the library is saved by default in the directory:

**C:\Programs\Lenze\DriveServer\S7**

#### Integration of the library in the STEP7 environment:

1. Select **File→Retrieve** in the S7 manager.
2. Select the file **Ldslib.arj** in the sub-directory **\S7** of the DriveServer installation.
3. Select the target directory.  
(Normally, the sub-directory "**\s7libs**" of the STEP7 installation.)



### 5.3.2 STEP7 example programs

The following example programs are part of the bus server S7 delivery package. They only differ in their hardware configuration:

File name	Hardware configuration
Ldsexmpl.arj/zip	PLC S7-400 with CPU412-2DP
Ldsexmp2.arj/zip	PLC S7-300 with CPU312-2DP and CP342-5

When the "DriveServer with bus server S7" software is installed, the example programs are saved by default in the directory:

**C:\Programs\Lenze\DriveServer\S7**



#### Tip!

If you want to use one of the example programs, adapt the hardware configuration to your system environment.

Detailed information on the creation of a STEP7 program can be found in chapter 5.6. (□ 91)

#### Integration of an example program in the STEP7 environment:

1. Select **File→Retrieve** in the S7 manager.
2. Select the corresponding file (**Ldsexmpl.arj/zip** and / or **Ldsexmp2.arj/zip**) in the sub-directory **\S7** of the DriveServer installation.
3. Select the target directory.  
(Normally, the sub-directory "**\s7proj**" of the STEP7 installation.)



#### Tip!

After retrieving the example project, you will be asked if the project is to be opened. If you answer this question with no, the project might not be added to the project list and must be opened later using the **Scan** command.

### 5.3.3 Device description files for the Lenze drives

When the "DriveServer with bus server S7" software is installed, the device description files of the Lenze drives for PROFIBUS are saved by default in the directory:

**C:\Programs\Lenze\DriveServer\GSD**

Copy these data into the sub-directory "**\s7data\GSD**" of the STEP7 installation.



#### Tip!

The latest versions of the GSD files are available on our Internet homepage: <http://www.Lenze.de> under **Service→Downloads→PROFIBUS**.



## Busserver S7

### 5.4 Creating new projects



#### Tip!

The following description is based on Version 5.0 of STEP7. The dialog boxes used in other STEP7 versions may be slightly different.

Select **File→Assistant 'New Project'** in the SIMATIC Manager to be guided through project creation. Please note:

- Select your hardware.
  - In our example, a **SIMATIC 300** with **CPU 315-2 DP** is used.
- Blocks need not be added yet.
- We recommend to use the **IL** language for all blocks.

Once the required information has been entered, the assistant creates a project structure similar to the one shown below:



The next step is to configure the MPI and the PROFIBUS interface for the project. More details about this can be found in the next chapter.

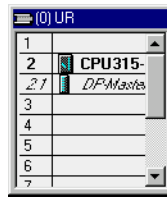


## 5.5 Hardware configuration

The Hardware Manager is used to configure the hardware used:

- Select the entry **CPU 315-2 DP** in the STEP7 project structure.
- Double-click the symbol **Hardware** on the right side of the window to open the Hardware Manager.

At this point, the Hardware Manager only lists the CPU-315-2 with integrated DP Master:



### 5.5.1 Configuring the MPI interface of the CPU

Proceed as follows to configure the MPI interface of the CPU:

1. Right-click the entry **CPU315-2 DP(1)** to open the context menu.
2. Select the entry **Object Properties** from the context menu to open the *Object Properties* dialog box.
3. Click in the **General** tab in the **Interface** group box the **Properties** button to open the *Properties - MPI interface* dialog box.
4. Configure the MPI address and the MPI sub-network in the **Parameters** tab and confirm the settings with **OK**.

### 5.5.2 Configuring the PROFIBUS interface of the CPU

Proceed as follows to configure the PROFIBUS interface of the CPU:

1. Select the **DP Master** line in the slot table.
2. Select **Insert→DP Master System** unless a DP master system has already been automatically created.



#### Tip!

If a DP master system has already been automatically created:

- Right-click the entry **DP Master System** to open the context menu.
  - Select the entry **Object Properties** from the context menu to open the *DP Master System* dialog box.
- 
3. Enter the DP device address of the DP master in the *DP Master System* dialog box.
  4. Create a new PROFIBUS sub-network with the profile **DP** and the desired baud rate.
  5. Select **View→Catalog** to open the hardware catalog to add PROFIBUS devices.



## Busserver S7

6. Select the corresponding device from the list and use the mouse to drag it onto the graphically illustrated PROFIBUS string.
  - If the hardware catalog does not include any Lenze modules (e.g. Lenze module 2131 under **PROFIBUS DP→Additional Field Devices→Drives→2131** check if the Lenze GSD files have been saved in the right directory on the hard disk. If necessary, update the catalog by clicking **Options→Update Catalog**.
7. Select a configuration for the corresponding device.
  - Important: Select a configuration which consistently transmits 8-byte parameters. Use, for example, "PAR(8ByteKons.)+PZD(2WorteKons)" or "PAR(4WorteKons.)+PZD(3WorteKons)".
8. Enter the PROFIBUS device address of the drive controller onto which the module is plugged.
  - In STEP7 version 5.1 and higher, the user is no longer automatically asked to select the configuration. Select the desired configuration in the catalog window and use the mouse to drag the object into the module list.
9. Double-click the first line in the module list to open the *DP Slave Properties* dialog box.
  - The Hardware Manager automatically suggests a suitable I/O address, in this example I/O address 1000.
  - Each device may have different I/O addresses. It is important that the input and the respective output use the same I/O address.
  - The I/O addresses allocated must be indicated later, when calling **FB DRIVE** and should therefore be written down.
  - We recommend that I/O address 1000 is used for initial testing.
10. Enter the settings for the process data channel into the second line of the module list.
  - The settings depend on the respective application.
  - The selected settings must be indicated later, when calling **FB DRIVE** and should therefore be written down.

The hardware configuration for the drive is now complete.

- Configure the other drives accordingly.

### 5.5.3 Configuring additional hardware components

Depending on the hardware equipment of your PLC, it might be necessary to configure additional hardware components using the Hardware Manager, e. g. communication processors for PROFIBUS or Ethernet.

- Detailed information can be found in the documentation for the corresponding hardware components.
- After the hardware configuration has been completed, you can close the Hardware Manager and integrate the Lenze function blocks into the project. For additional information, please read the following chapter 5.6.



## 5.6 Creating a STEP7 program

The Lenze function blocks can be integrated in different ways. For a simple test, the blocks supplied in the SIMATIC Manager can be copied from the library to a new project.



### Note!

If you have configured drive controllers in your project which are temporarily disconnected from the mains, please observe the following:

- Drive controllers which are completely disconnected from the mains and are fully de-energised, are no longer recognised as PROFIBUS devices by the PLC. This can be prevented by supplying the function block/fieldbus module with an external voltage.
- Instances of the **FB DRIVE** for disconnected drive controllers have to be skipped in the PLC program.
  - A query can be performed via the status words of the blocks **FC91**, **FC92**, **FC93** or **FC94**. If the status word of these blocks indicates "0", the drive controller is not switched on (or not available) and the corresponding **FB DRIVE** instance must be skipped. (□ 113)

If an S7-300 is used, you can address drive controllers which are directly connected to the PROFIBUS master of the CPU card and drive controllers which are connected with an external PROFIBUS communication processor (CP).

- For this, please use the block **FB DRIVE CP** instead of the block **FB DRIVE**.

In the first step, communication is to be performed via OPC with only one drive connected to the PROFIBUS master of the CPU.

### 5.6.1 Accessing a drive via OPC



### Tip!

You can use the provided example programs for initial commissioning. In this case, you only have to adapt the hardware configuration to your system environment. (□ 87)

If you want to create a program on your own, you simply have to integrate the supplied blocks into the SIMATIC Manager.

### Proceed as follows to create your own program:

1. Create a new project with the block **OB1**.
2. Select **File→Open**, click **Libraries** and select the **LenzeDriveServer**.
  - In the new window, you will find all blocks contained in the "LenzeDriveServer" library.
3. Simply drag and drop blocks **FB90**, **DB80** and **DB81** into your new project .
4. Call an instance of **FB90** in **OB1**.
  - When programming in IL enter the following program code:  
`CALL FB90, DBxxx`  
 (xxx represents a free data block number)  
 This call automatically creates the data block.
5. Edit the list of transfer parameters.



## Busserv S7

### Example of a transfer parameter list:

```
CALL "Lenze-DriveServer_FB", DB100           FB90

Tag                :=
PB_Address         :=B#16#9
EA_Address        :=1000
DriveServer_DB    :="Lenze-DriveServer_FB"   DB80
DrivePar_DB       :="S7-Parameter"          DB81
TimerNr           :=T1
TimeOut           :=
Busy              :=MB100
```

**BE**

Transfer parameters used in the above list:

- The parameter *Tag* must not be specified.
- In this example, the PROFIBUS address is address 9.
- I/O address 1000 was set as a device parameter in the Hardware Manager.
- **T1** is to be used as timer.
- As the parameter *Timeout* was not specified, the factory setting (1s) is used.
- The flag byte **MB100** is used to ascertain, if an instance is accessing an I/O.



### Tip!

The PROFIBUS and the I/O addresses have to be configured twice, once in the Hardware Manager and once in the calling program. It is important that the two configurations correspond.

Provided that a drive is available at PROFIBUS address 9, the system can be operated with this program.

- Load the program into the PLC and try to detect the drive automatically using the S7 configuration tool. (☞ 76)
- Once the completed configuration has been registered and saved with the S7 configuration tool, the DriveServer can access all parameters of the drive.



### Tip!

The PLC programs can be written in any S7 language, the function block **FB DRIVE** (FB90) is, however, protected and cannot be processed in any of these languages.



## 5.6.2 Accessing several drives via OPC

The project is to be expanded to enable communication with several drives.

- Add the corresponding devices to the Hardware Manager, as described for the drive already configured.
- Select one instance of the function block **FB DRIVE** (FB90) for every drive connected in the program.
  - One instance data block, one timer and one flag bit each is allocated to every instance.
  - The remaining parameters are to be determined in the same way as described in the example of communication with one drive only.

## 5.6.3 Accessing drives connected to PROFIBUS CPs

If an S7-300 is used, you can address drive controllers which are directly connected to the PROFIBUS master of the CPU card and drive controllers which are connected with an external PROFIBUS communication processor (CP).

For this, please use the block **FB DRIVE CP** instead of the block **FB DRIVE** in the STEP7 program.

**Example of a transfer parameter list:**

```
CALL "Lenze-DriveServer_FB", DB100          FB91

Tag                :=
PB_Address         :=B#16#9
First_DriveDW     :=0
DriveSend_DB      :="Lenze-DriveServer_Send"  DB30
DriveRecv_DB      :="Lenze-DriveServer_Recv"   DB31
DriveServer_DB    :="Lenze-DriveServer_FB"     DB80
DrivePar_DB       :="S7-Parameter"            DB81
TimerNr           :=T1
TimeOut           :=
Busy              :=MB100
```

**BE**

The following transfer parameters are used in addition to the parameters listed in chapter 5.6.1 on page 91.

- The parameter *FirstDriveDW* defines the byte position of the first data word of the connected drive controller.
  - For each drive controller, 8 data words are reserved in the data blocks **DriveSend\_DB** and / or **DriveRecv\_DB**. The first 4 data words are needed for parameter access and the last 4 data words for process data access.
  - The drive controllers must have successive I/O addresses.
- **DriveSend\_DB** is the send block for communication with the CP.
  - The block consists of up to 15 variables of data type **UDT91 ... UDT93**. The data type to be used depends on the number of process data words used.
  - The delivery package includes **DB30** with **UDT93** (4 process data words).
- **DriveRecv\_DB** is the receive block for communication with the CP.
  - The block consists of up to 15 variables of data type **UDT91 ... UDT93**. The data type to be used depends on the number of process data words used.
  - The delivery package includes **DB31** with **UDT93** (4 process data words).



## Busserv S7



### Note!

If you want to address drive controllers connected to the PROFIBUS master of the CPU and drive controllers connected to an external PROFIBUS communication processor (CP), it must be ensured that each PROFIBUS address is only assigned once.

It is not possible to address two drive controllers with the same PROFIBUS address, even if the drive controllers are operated at different PROFIBUS strings.



### Tip!

- You can use the provided example program "Ldsexmp2.arj" for testing.
- If possible, the drive controllers should always be operated at the PROFIBUS master of the CPU, since communication will thus be slightly faster.

## 5.6.4 Accessing process and parameter data from the PLC

The following functions from the "LenzeDriveServer" library can be used to access process and parameter data from the PLC:

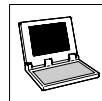
Block number	Symbolic block name	Function
FC90	FC Parameter	Function used to write parameter data jobs in <b>DB S7PAR</b> .
FC91	FC Process Data	Function for process data access (3 process data words)
FC92	FC Process Data	Function for process data access (2 process data words)
FC93	FC Process Data	Function for process data access (3 process data words) <ul style="list-style-type: none"> <li>• When using an external PROFIBUS communication processor (CP)</li> </ul>
FC94	FC Process Data	Function for process data access (2 process data words) <ul style="list-style-type: none"> <li>• When using an external PROFIBUS communication processor (CP)</li> </ul>

For information about the basic structure of such a project, please see chapter 5.2. (□ 85)



### Tip!

- The aforementioned example program "Ldsexmpl.arj" contains both OPC and PLC accesses, and can be used as basis for your own program developments.
- An explanation of the function parameters for **FC90 ... FC94** can be found in the Appendix. (□ 100)



## 5.7 Testing the STEP7 program

When the program has been completed, it can be loaded into the PLC and tested. The S7 configuration tool can be used for program testing.

You can use the S7 configuration tool to scan the PLC program for the drives used.

- The configuration tool first tries to detect which data blocks are allocated as job lists.
- The instance data blocks for the drives are then scanned.
  - For every drive found, a corresponding configuration entry is created in the tree structure in the left part of the S7-MPI configuration tool. The block number of the job list is entered into the input field provided.

If these entries correspond to the expected result, further tests can be made with the DriveServer:

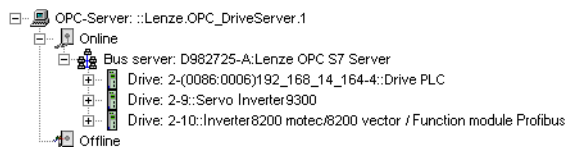
- Save the configuration file of the S7 configuration tool and start the configuration of the DriveServer. For more information about configuration, please refer to the documentation “DriveServer - Getting Started”.

### 5.7.1 Structure of the DriveServer name area

Each drive controller has its own name within the DriveServers. The name consists of the following elements separated by a hyphen:

- Address of the master PLC
- Sub-network ID (in round brackets; only if a slave PLC exists)
- Address of the slave PLC (only if a slave PLC exists)
- PROFIBUS address of the drive controller

**Example:**



- Two drive controllers with PROFIBUS address 9 and 10 are connected to a master PLC with MPI address 2.
- In addition, a slave PLC is connected to the master PLC via TCP.
  - The corresponding TCP sub-network has the sub-network ID 0086:0006
  - The slave PLC has the IP address 192.168.14.164
- A drive controller with PROFIBUS address 4 is connected to the slave PLC.



## Busserv S7

### 5.8 Optimisation of the cycle time

When **FB DRIVE** is activated, the PLC cycle is always under load, even if no parameter communication takes place. Since several parameters are checked when **FB DRIVE** is called, programming errors can be detected and the corresponding error messages can be indicated. However, the checking costs computing time.

To save computing time, you can operate **FB DRIVE** in "expert mode". This reduces the required computing time by approx. 50 %.

#### Proceed as follows to activate the expert mode for FB DRIVE

Open the instance data block and set the variable *XpertMode* to TRUE:

0.0	in	Tag	STRING [ 18 ]	'LENZE_DRIVE'	'LENZE_DRIVE'
20.0	in	PB_Address	BYTE	B#16#0	B#16#0
22.0	in	EA_Address	INT	0	0
24.0	in	DriveServer_DB	BLOCK_DB	DB 1	DB 1
26.0	in	DrivePar_DB	BLOCK_DB	DB 1	DB 1
28.0	in	TimerNr	TIMER	T 0	T 0
30.0	in	TimeOut	S5TIME	S5T#1S	S5T#1S
32.0	out	Busy	BYTE	B#16#0	B#16#0
34.0	stat	XpertMode	BOOL	FALSE	TRUE



#### Note!

When the expert mode is activated, **FB DRIVE** no longer checks the transferred parameters!

Therefore, the expert mode should only be activated if you are sure that the transferred parameters are correct.



#### Tip!

If you have created a program which uses an older version of **FB 90**, STEP7 will indicate different time stamps for **FB 90** and the corresponding instance data block.

If so, change to the IL editor and create a new STEP7 instance data block. Then you can use the expert mode.



## 5.9 Accessing PLC data

With bus server S7, you can access both the parameters of the drive controllers and the PLC data. You can use any OPC client to access the PLC data and, in some cases, to change them.

The following PLC data can be accessed via OPC:

- Flags
- Data blocks
- Inputs
- Outputs
- Timers
- Numerators

For every data type to be read, a block must be added to the configuration within which the individual OPC items must be configured.



---

### Tip!

More information on this topic can be found in the Online Help for the Lenze S7 configuration tool.

---



## Busserver S7

# 6 Appendix

## 6.1 Time load of the PLC cycle

The function blocks are cyclically called. The PLC cycle is permanently under load.

- The exact processing time depends on the computing power of the CPU.
- If real communication takes place, processing takes longer than if a block is activated without a command to be processed.
- When the PLC program is complete, the cycle load can be reduced by activating the expert mode. (📖 96)

The below table lists typical processing times for two different CPU types:

Block number Operation	Processing time	
	CPU 412-2 DP (S7-400)	CPU 315-2 DP (S7-300)
FC90	< 0.1 ms	0.6 ms
FC91	< 0.1 ms	0.7 ms
FC92	< 0.1 ms	0.6 ms
FB90		1 ... 7 ms
	Expert mode	< 1 ms
	Idle running	< 1 ms
FB91		1 ... 7 ms
	Expert mode	1 ... 3 ms
	Idle running	1 ... 2 ms

## 6.2 Transfer times

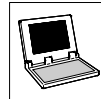
The following characteristics are important for the transfer times.



### Tip!

Please note that the times indicated are minimum values.

- If PROFIBUS is under high load or if a complex program is to be processed, times will rise accordingly and may be considerably longer.
- Communication disturbances and access to PROFIBUS devices that are not available may substantially affect transfer times.
- Transfer times also depend on the operating system used. For optimum transfer times, we recommend to use Windows NT, Windows 2000 or Windows XP.

**Output parameters:**

- Global Drive Control (GDC) and DriveServer were used to load a complete parameter set.
- An S7-315 (cycle time: 15 ms) and an S7-412 (cycle time: 2 ms) were used as PLC.
- PROFIBUS was set to 1.5 Mbit/s.
- Different interface modules were used.
- MPI baud rate: 187.5 Kbaud
- Baud rate of the serial interface for PC adapter connection: 38.4 Kbaud

**Transfer times detected for one parameter set:**

PLC		Controller	Interface module	
Type	Connection		PC-PLC connection	Transfer time
S7-315	PROFIBUS on the CPU	82xx vector	CP5511 (187.5 Kbaud)	44 s
	CP342-5	82xx vector	CP5511 (187.5 Kbaud)	54 s
	PROFIBUS on the CPU	93xx	CP5511 (187.5 Kbaud)	107 s
	CP342-5	82xx vector	60 s	64 s
S7-412	PROFIBUS on the CPU	82xx vector (FIF)	CP5511	12 s
		82xx vector (AIF)	CP5511	13 s
		82xx vector (FIF)	PC adapter	24 s
		82xx vector (FIF)	Ethernet	11 s
		93xx	CP5511	30 s
		93xx	PC adapter	56 s
		93xx	Ethernet	28 s

**Summary:**

- The transfer time mainly depends on the cycle time of the PLC.  
The longer the cycle time, the lower the influence of the bus rates.
- A fast connection between PC and PLC accelerates data transfer. Yet, the cycle time remains the limiting factor.
- Some CPU cards permit the baud rate of the MPI bus to be set to up to 12 Mbit/s. If technology allows, this increase in speed should be taken advantage of.
- However, the PROFIBUS transfer rate only has a minor influence on the transfer time of a parameter set.



## Busserver S7

### 6.3 Communication function parameters

#### 6.3.1 FB90 (FB DRIVE)

##### Function block

- For every drive one instance is to be created.
- The function block is based on the functions SFC14/15.

Parameters	Data type	Possible values/meaning
Tag	IN: STRING	Special identifier; do not change
PB_Address	IN: BYTE	0...31 PROFIBUS address of the drive (hexadecimal) • Address 1 = 1, address 15 = F
EA_Address	IN: INT	I/O address of the drive (decimal)
DriveServer_DB	IN: BLOCK_DB	Data block with parameter data job list via OPC • e.g. <b>DB80</b>
DrivePar_DB	IN: BLOCK_DB	Data block with parameter data job list of the PLC • e.g. <b>DB81</b>
TimerNo	IN: TIMER	Timer for monitoring the drive • e.g. <b>T1</b>
TimeOut	IN: S5TIME	Timeout for monitoring. • If not indicated, pre-setting "S5T#1S" (= 1 s) is used.
Busy	OUT: BYTE	Status of the function:
		0x00 Not active (jobs have been processed)
		0x01 Active (jobs are still being processed)
		0x10 Incorrect identifier in the data block, incorrect data block indicated or overwritten: Reload data block
		0x11 <i>PB_Address</i> incorrect (permissible values: 1 ... 123)
		0x40 <i>Count</i> outside the range 0 ... 31
		0xF0 No module configured under <i>PB_Address</i>
		0xF1 Length of source range <> Length of configured user data
0xFF General I/O access error		



### 6.3.2 FB91 (FB DRIVE CP)

#### Function block

- For every drive one instance is to be created.
- The function block is based on the functions SFC58/59.

Parameters	Data type	Possible values/meaning
Tag	IN: STRING	Special identifier; do not change
PB_Address	IN: BYTE	0...31 PROFIBUS address of the drive (hexadecimal) • Address 1 = 1, address 15 = F
FirstDriveDW	IN: INT	Byte offset of the first drive data word in the data exchange DB (decimal)
DriveSend_DB	IN: BLOCK_DB	Data exchange DB for send jobs to the external PROFIBUS communication processor (CP)
DriveRecv_DB	IN: BLOCK_DB	Data exchange DB for read jobs from the external PROFIBUS communication processor (CP)
DriveServer_DB	IN: BLOCK_DB	Data block with parameter data job list via OPC • e.g. <b>DB80</b>
DrivePar_DB	IN: BLOCK_DB	Data block with parameter data job list of the PLC • e.g. <b>DBB1</b>
TimerNo	IN: TIMER	Timer for monitoring the drive • e.g. <b>T1</b>
TimeOut	IN: S5TIME	Timeout for monitoring. • If not indicated, pre-setting "S5T#1S" (= 1 s) is used.
Busy	OUT: BYTE	Status of the function:
		0x00 Not active (jobs have been processed)
		0x01 Active (jobs are still being processed)
		0x10 Incorrect identifier in the data block, incorrect data block indicated or overwritten: Reload data block
		0x11 <i>PB_Address</i> incorrect (permissible values: 1 ... 123)
		0x40 <i>Count</i> outside the range 0 ... 31
		0xF0 No module configured under <i>PB_Address</i>
		0xF1 Length of source range <> Length of configured user data
0xFF General I/O access error		



## Busserver S7

### 6.3.3 FC90 (FC Parameter)

Function for parameter data jobs from the PLC.

Parameters	Data type	Possible values/meaning
DrivePar_DB	IN: BLOCK_DB	Data block containing the parameter data <ul style="list-style-type: none"> <li>e.g. <b>DB80</b></li> </ul>
Count	IN: INT	0...31 Memory location within <i>DrivePar_DB</i>
PB_Address	IN: BYTE	PROFIBUS address of the drive (hexadecimal) <ul style="list-style-type: none"> <li>Address 1 = 1, address 15 = F</li> </ul>
R_W	IN: BOOL	TRUE Read drive
		FALSE Write drive
Code	IN: WORD	Lenze code (hexadecimal): w#16#XX
SubCode	IN: BYTE	Lenze sub-code (hexadecimal): jB#16#XX
W_Value	IN: DINT	Value to be transmitted (FIX32)
Status	OUT: BYTE	Status of the function:
		0x00 No errors
		0x40 <i>Count</i> outside the range 0 ... 31
		0x80 Controller error <ul style="list-style-type: none"> <li>For details see <i>R_Value</i></li> </ul>
		0x81 When calling <i>FB90</i> , data for <i>DriveServer_DB</i> or <b>DrivePar_DB</b> are invalid (identifier missing or incorrect). <ul style="list-style-type: none"> <li>Use the blocks supplied with the DriveServer library.</li> </ul>
		0x82 <i>PB_Address</i> incorrect (permissible values: 1 ... 123)
		0x83 I/O error <ul style="list-style-type: none"> <li>For details see <i>R_Value</i></li> </ul>
0x84 Job could not be processed within the time set under <i>TimeOut</i> .		



Parameters	Data type	Possible values/meaning
R_Value	OUT: DINT	Read value (FIX32)
		If <i>Status</i> = 0x83, an I/O error occurred:
		F0 F0 F0 F0 Module does not exist or SFC is not available (decimal: -252 645 136)
		F1 F1 F1 F1 Data length <> 8 bytes (decimal: -235 802 127)
		FF FF FF FF Other I/O error (decimal: -1)
		If <i>Status</i> = 0x80, the drive controller indicated an error. <ul style="list-style-type: none"> <li>Additional error codes and further explanations can be found in the descriptions for the individual PROFIBUS interface modules.</li> </ul>
		06 03 00 00 No access permission
		06 05 00 10 Unpermissible job parameter
		06 05 00 11 Invalid sub-index
		06 05 00 12 Data length too long
		06 07 00 00 Object does not exist
		06 08 00 00 Data types do not correspond
		08 00 00 20 Job cannot be executed at present
		08 00 00 21 Cannot be executed because of local control
		08 00 00 22 Cannot be executed because of operating state of device
08 00 00 30 Value range has been exceeded		
08 00 00 40 Collision with other values		
Trigger	INOUT: BOOL	TRUE Start job processing. <ul style="list-style-type: none"> <li>Automatically reset to FALSE, when the job has been completely processed and the result is available.</li> </ul>



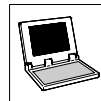
## Busserver S7

### 6.3.4 FC91 (FC Process Data, 3 words)

#### Function for process data access

Consistent process data transfer is a prerequisite for using this function!

Parameters	Data type	Possible values/meaning	
EA_Address	IN: INT		I/O address of the process data channel <ul style="list-style-type: none"> <li>e. g. 1008</li> </ul>
Enable	IN: BOOL	TRUE	Drive enabled
		FALSE	Drive inhibited
QSP	IN: BOOL	TRUE	Quick stop (QSP) active
		FALSE	Quick stop (QSP) not active
Reset	IN: BOOL	FALSE $\neq$ TRUE	TRIP reset executed
Setpoint1	IN: INT		Setpoint process value (AIF-IN:W1)
Setpoint2	IN: INT		Setpoint process value (AIF-IN:W2)
ControlWord	IN: WORD		Control word according to DRIVECOM specification <ul style="list-style-type: none"> <li>Bit0 ... Bit4 are hidden and cannot be written.</li> </ul>
RDY	OUT: BOOL	TRUE	Drive is ready for operation
		FALSE	Drive is not ready for operation
TRIP	OUT: BOOL	TRUE	Drive signals TRIP
		FALSE	Drive does not signal TRIP
ActualValue1	OUT: INT		Actual process value (AIF-OUT:W1)
ActualValue2	OUT: INT		Actual process value (AIF-OUT:W2)
StatusWord	OUT: WORD		Status word according to DRIVECOM specification



### 6.3.5 FC92 (FC Process Data, 2 words)

#### Function for process data access

The function **FC92** can be used for both consistent and inconsistent process data transfer.

Parameters	Data type	Possible values/meaning	
EA_Address	IN: INT		I/O address of the process data channel <ul style="list-style-type: none"> <li>e. g. 1008</li> </ul>
Enable	IN: BOOL	TRUE	Drive enabled
		FALSE	Drive inhibited
QSP	IN: BOOL	TRUE	Quick stop (QSP) active
		FALSE	Quick stop (QSP) not active
Reset	IN: BOOL	FALSE $\nrightarrow$ TRUE	TRIP reset executed
Setpoint1	IN: INT		Setpoint process value (AIF-IN:W1)
ControlWord	IN: WORD		Control word according to DRIVECOM specification <ul style="list-style-type: none"> <li>Bit0 ... Bit4 are hidden and cannot be written.</li> </ul>
RDY	OUT: BOOL	TRUE	Drive is ready for operation
		FALSE	Drive is not ready for operation
TRIP	OUT: BOOL	TRUE	Drive signals TRIP
		FALSE	Drive does not signal TRIP
ActualValue1	OUT: INT		Actual process value (AIF-OUT:W1)
StatusWord	OUT: WORD		Status word according to DRIVECOM specification



## Busserver S7

### 6.3.6 FC93 (FC Process Data, 3 words, CP)

#### Function for process data access

Consistent process data transfer is a prerequisite for using this function!

Parameters	Data type	Possible values/meaning	
FirstDriveDW	IN: INT		Byte offset of the first drive data word in the data exchange DB (decimal)
DriveSend_DB	IN: BLOCK_DB		Data exchange DB for send jobs to the external PROFIBUS communication processor (CP)
DriveRecv_DB	IN: BLOCK_DB		Data exchange DB for read jobs from the external PROFIBUS communication processor (CP)
Enable	IN: BOOL	TRUE	Drive enabled
		FALSE	Drive inhibited
QSP	IN: BOOL	TRUE	Quick stop (QSP) active
		FALSE	Quick stop (QSP) not active
Reset	IN: BOOL	FALSE → TRUE	TRIP reset executed
Setpoint1	IN: INT		Setpoint process value (AIF-IN:W1)
Setpoint2	IN: INT		Setpoint process value (AIF-IN:W2)
ControlWord	IN: WORD		Control word according to DRIVECOM specification • Bit0 ... Bit4 are hidden and cannot be written.
RDY	OUT: BOOL	TRUE	Drive is ready for operation
		FALSE	Drive is not ready for operation
TRIP	OUT: BOOL	TRUE	Drive signals TRIP
		FALSE	Drive does not signal TRIP
ActualValue1	OUT: INT		Actual process value (AIF-OUT:W1)
ActualValue2	OUT: INT		Actual process value (AIF-OUT:W2)
StatusWord	OUT: WORD		Status word according to DRIVECOM specification



### 6.3.7 FC94 (FC Process Data, 2 words, CP)

#### Function for process data access

The function **FC94** can be used for both consistent and inconsistent process data transfer.

Parameters	Data type	Possible values/meaning	
FirstDriveDW	IN: INT		Byte offset of the first drive data word in the data exchange DB (decimal)
DriveSend_DB	IN: BLOCK_DB		Data exchange DB for send jobs to the external PROFIBUS communication processor (CP)
DriveRecv_DB	IN: BLOCK_DB		Data exchange DB for read jobs from the external PROFIBUS communication processor (CP)
Enable	IN: BOOL	TRUE	Drive enabled
		FALSE	Drive inhibited
QSP	IN: BOOL	TRUE	Quick stop (QSP) active
		FALSE	Quick stop (QSP) not active
Reset	IN: BOOL	FALSE → TRUE	TRIP reset executed
Setpoint1	IN: INT		Setpoint process value (AIF-IN:W1)
ControlWord	IN: WORD		Control word according to DRIVECOM specification <ul style="list-style-type: none"> <li>• Bit0 ... Bit4 are hidden and cannot be written.</li> </ul>
RDY	OUT: BOOL	TRUE	Drive is ready for operation
		FALSE	Drive is not ready for operation
TRIP	OUT: BOOL	TRUE	Drive signals TRIP
		FALSE	Drive does not signal TRIP
ActualValue1	OUT: INT		Actual process value (AIF-OUT:W1)
StatusWord	OUT: WORD		Status word according to DRIVECOM specification

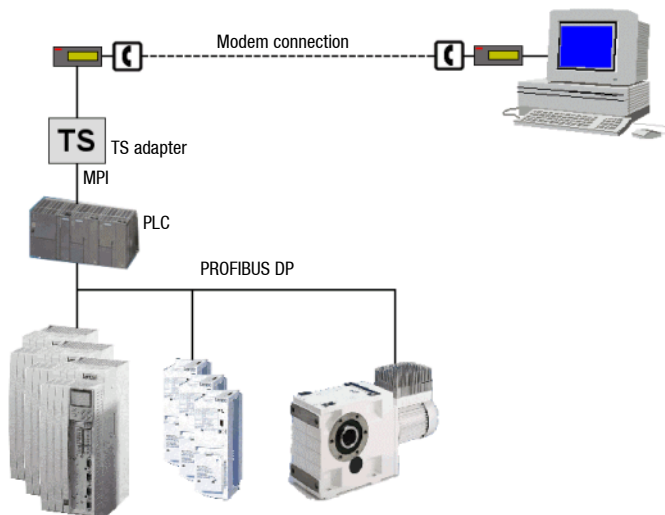


## Busserver S7

### 6.4 Remote maintenance via modem

Siemens offers a TS adapter which can be used for remote maintenance. This adapter is connected between the MPI connection of the PLC and a commercial modem. The programming device which is also equipped with a modem can now communicate with the PLC via a standard telephone connection.

Instead of using the programming device to carry out the remote maintenance, you can also use the bus server installed at a PC equipped with a modem.



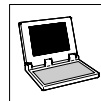
In addition to the TS adapter for the PLC, the following software components are required for the PC to carry out remote maintenance via a modem:

- Siemens TeleService software
- Siemens STEP7



#### Tip!

- The connection with the remote maintenance system always has to be established manually.
- The bus server may only be started after the connection has been established, otherwise communication is not possible.



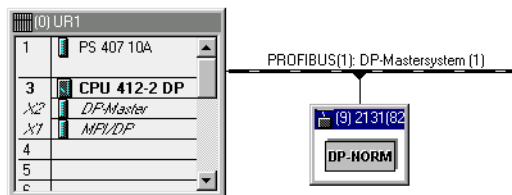
## 6.5 Example project S7-400

The example project "LenzeDriveServer" described in this chapter can be found in the sub-directory **VS7** of the DriveServer installation.

### 6.5.1 Hardware configuration

- An S7-400 station is used as PLC.
- CPU 412-2 DP and power supply PS 407 10A are configured in a RACK 400.
- A PROFIBUS-DP sub-network with master system No. 1 and a baud rate of 1.5 Mbit/s is available.
- A Lenze fieldbus module of type 2131 is used as DP slave. This fieldbus module can be used for 8200 and 9300 drive controllers.
  - PROFIBUS address: 9 (hexadecimal)
  - Start address of parameter channel: 1000
  - Start address of process data channel: 1008
  - Consistent communication with 8-byte parameter data and 2 process data words (Configuration "PAR(8ByteKons.)+PZD(2Worte)")

The following figure shows the hardware configuration in the Hardware Manager:



#### Tip!

The slot table in the Hardware Manager lists the settings of the parameter channel (slot 0) and the process data channel (slot 1).

The data provided in the columns **I-address** and **O-address** are required to call the function **FC92** in the PLC program:

Slot			I-address	O-address
0	183	PAR(8ByteKons.)+PZD(2Worte)	1000..1007	1000..1007
1	113	-> PAR(8ByteKons.)+PZD(2Worte)	1008..1011	1008..1011

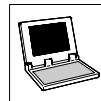


## Busserver S7

### 6.5.2 Program blocks used

The project folder **Blocks** contains the following program blocks:

Block	Function
OB1	Organisation block (program cycle); used to call <b>FC4</b>
FC4	Function 4; used to call all other blocks
FB90	Function block <b>FB DRIVE</b> enables parameter access via OPC and from the PLC
FC90	Function <b>FC Parameter</b> for parameter data jobs from the PLC
FC91/FC92	Function for DP Process Data (3 words/2 words)
DB80	Data block <b>DB OPC</b> for data exchange between OPC and <b>FB90</b>
DB81	Data block <b>DB S7PAR</b> for data exchange between <b>FC90</b> and <b>FB90</b>
DB100	Instance data block for <b>FB90</b> (DP Slave1)
SFC14/SFC15	System blocks for communication via PROFIBUS-DP
UDT90	Universal data type for parameter data access to Lenze drive controllers
VAT10	Variables table for changing the example parameters



### 6.5.3 Symbolic variable names

Symbolic variable names are used in the “LenzeDriveServer” example project. These names are saved in the symbol table.

#### Input parameters:

Symbol	Address	Data type	Comment
Enable	M10.0	BOOL	TRUE = Enable drive
QSP	M10.1	BOOL	TRUE = Set quick stop
Reset	M10.2	BOOL	TRUE = Reset TRIP in the drive
TriggerRead	M10.6	BOOL	TRUE = Start reading a parameter
TriggerWrite	M10.7	BOOL	TRUE = Start writing a parameter
Code	MW202	WORD	Lenze code number of the parameter
Subcode	MB201	BYTE	Lenze sub-code number of the parameter
WriteValue	MD204	DINT	Value to be transmitted (FIX32)
ControlWord	MW24	WORD	Control word according to DRIVECOM specification <ul style="list-style-type: none"> <li>• Bit0 ... Bit4 are hidden and cannot be written.</li> </ul>
PB-Address	MB200	BYTE	PROFIBUS address of the drive
TimeOutTimer_OPC	T1	TIMER	Timer for monitoring the drive
DB-OPC-Server	DB80	DB80	Data block for parameter data jobs via OPC
S7-Parameter	DB81	DB81	Data block for parameter data jobs from the PLC

#### Output parameters:

Symbol	Address	Data type	Comment
RDY	M30.0	BOOL	TRUE = Drive is ready for operation
TRIP	M30.1	BOOL	TRUE = TRIP set in the drive
BUSY	MB150	BYTE	TRUE = Parameter data jobs are still being processed
ActualValue1	MW32	INT.	Actual process value
Setpoint1	MW20	INT.	Setpoint process value
ReadValue	MD224	DINT	Read value (FIX32)
StatusWord	MW36	WORD	Status word according to DRIVECOM specification
WriteStatus	MB210	BYTE	Status of the function <b>FC90 (FC Parameter)</b> when writing a parameter
ReadStatus	MB220	BYTE	Status of the function <b>FC90 (FC Parameter)</b> when reading a parameter
WriteResult	MD214	DINT	Error code displayed when writing a parameter was not successful
Value	MD228	DINT	Contains parameter <i>W_Value</i> within the read routine; can be ignored when reading a parameter

We recommend the use of symbolic variable names for clarity. However, they are not compulsory.



## Busserver S7

### 6.5.4 OB1/FC4

With **OB1**, only the function **FC4** is called, in which all other blocks are called.

### 6.5.5 Process data transfer

#### FC4, network 1

The following call is used to transmit two process data words (2PZD).

- Process data word 1 is always the DRIVECOM control word.
- Process data word 2 can be freely selected.

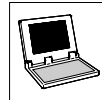
```
CALL FC 92                //FC92, 2 PZD
EA_Address                :=1008          //I-address, decimal
Enable                    :="Enable"      //BOOL
QSP                       :="QSP"        //BOOL
Reset                     :="Reset"      //BOOL
Setpoint1                 :="Setpoint1"  //INT, decimal
ControlWord               :="ControlWord" //WORD, control word
RDY                       :="RDY"        //OUT, BOOL
TRIP                      :="TRIP"       //OUT, BOOL
ActualValue1              :="ActualValue1" //OUT, INT
StatusWord                :="StatusWord" //OUT, status word
```

- The start address for the process data channel can be obtained from the hardware configuration. (here: 1008)
- The inputs *Enable*, *QSP* and *Reset* are set to enable the controller:
  - *Enable* = TRUE
  - *QSP* = FALSE
  - *Reset* = FALSE
- The second process data word is assigned to the setpoint input.
- Control word and status word conform to the DRIVECOM specification.
- The DRIVECOM status machine has already been implemented in **FC92**.
- The outputs *RDY* and *TRIP* provide information about the controller status.
- *ActualValue1* is the value read out.



#### Tip!

Detailed information about the parameters of **FC92** can be found in chapter 6.3.5. (105)  
Use function block **FC91** to transmit three process data words. (104)



## 6.5.6 Parameter data access

The **FB90 (FB DRIVE)** enables accessing parameter data both via OPC and from the PLC. The changeover between these two communication channels takes place automatically.

- One instance each of **FB90** is to be called for each drive (DP Slave). In addition, one instance data block for each drive is required for variables storage.
- Parameter data jobs via OPC are saved in **DB OPC**.
- Parameter data jobs from the PLC are saved in **DB S7PAR**.

### FC4, network 2

```

L      "StatusWord"                //disconnected
L      0                          //drive controllers
==I                               //are
SPB   NEXT                        //skipped.

      CALL FB 90 , DB100

      Tag                          :=                //STRING
PB_Address      :=B#16#9          //BYTE, B#16#x
EA_Address      :=1000            //I-address, decimal
DriveServer_DB  := "DB-OPC-Server" //DBx; e.g. DB80
DrivePar_DB     := "S7-Parameter"  //DBx; e.g. DB81
TimerNr        := "TimeOutTimer_OPC" //TIMER
TimeOut        :=S5T#1S          //TIME
Busy           := "BUSY"          //OUT, BYTE

NEXT: NOP 0

```

- Use the *StatusWord* to check if the drive controller is switched on ( *StatusWord* <> 0). If it is not, the call of **FB90** is skipped.
- In the above example, **FB90** uses the instance data block **DB100** for variables storage.
- The input *Tag* is not assigned.
- The PROFIBUS device address and the start address of the parameter channel are indicated as shown above.
- In the above example, both parameter data blocks (**DB80**, **DB81**) are used.
- The output byte *Busy* provides information about the communication status.



### Tip!

Detailed information about the parameters of **FB90** can be found in chapter 6.3.1. (100)



## Busserver S7

### 6.5.7 Reading a parameter

#### FC4, network 3

```

UN "TriggerRead"
= "TriggerRead"
CALL FC 90 //data types
DrivePar_DB := "S7-Parameter" //DB81
Count := 0 //decimal: 0-31
PB_Address := "PB-Address" //B#16#9
R_W := TRUE //FALSE or TRUE
Code := "Code" //W#16#C, code 12
SubCode := "Subcode" //B#16#0, Sub-code 0
W_Value := "Value" //DINT oder L#xxx
Status := "ReadStatus" //OUT, BYTE, status byte
R_Value := "ReadValue" //OUT, DINT, value
Trigger := "TriggerRead" //INOUT, BOOL

```

- The variable *TriggerRead* is always set to TRUE. Once a parameter read request has been processed correctly, *TriggerRead* is automatically reset to FALSE.
- **FC90** and parameter *R\_W* = TRUE are called to transmit a read request for a drive parameter.
- Data exchange with **FB DRIVE (FB90)** takes place via the data block **S7-Parameter (DB81)**.
- Input parameters are the request number, PROFIBUS address, write/read request and the code and sub-code to be read.
- Output parameters are the read value and the status byte.
- Set parameter *R\_W* to FALSE to transmit a write request for a drive parameter.



#### Tip!

Detailed information about the parameters of **FC90** can be found in chapter 6.3.3. (📖 102)



## 6.5.8 Writing a parameter

### FC4, network 4

```

UN "TriggerWrite"
= "TriggerWrite"
CALL FC 90 //data types
DrivePar_DB := "S7-Parameter" //DB81
Count := 1 //decimal: 0-31
PB_Address := "PB-Address" //B#16#9
R_W := FALSE //FALSE or TRUE
Code := "Code" //W#16#C, code 12
SubCode := "Subcode" //B#16#0, Sub-code 0
W_Value := "WriteValue" //DINT oder L#xxx
Status := "WriteStatus" //OUT, BYTE, status byte
R_Value := "WriteResult" //OUT, DINT, value
Trigger := "TriggerWrite" //INOUT, BOOL

```

- The variable *TriggerWrite* is always set to TRUE. Once a parameter write request has been processed correctly, *TriggerWrite* is automatically reset to FALSE.
- **FC90** and parameter *R\_W* = FALSE are called to transmit a write request for a drive parameter.
- Data exchange with **FB DRIVE (FB90)** takes place via the data block **S7-Parameter (DB81)**.
- Input parameters are the request number, PROFIBUS address, write/read request, code and sub-code and the actual value.
- Output parameters are the read value and the status byte.
- Set parameter *R\_W* to TRUE to transmit a read request for a drive parameter.



### Tip!

Detailed information about the parameters of **FC90** can be found in chapter 6.3.3. (📖 102)



## Busserver S7

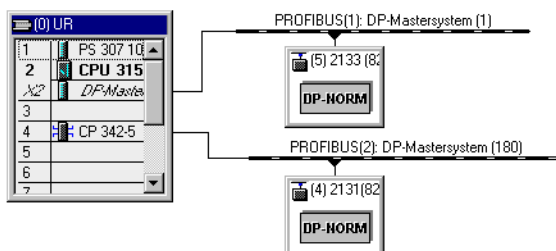
### 6.6 Example project S7-300 with PROFIBUS-CP

The example project "LenzeDriveServer" described in this chapter can be found in the sub-directory **VS7** of the DriveServer installation.

#### 6.6.1 Hardware configuration

- An S7-300 station is used as PLC.
- CPU 315-2 DP, CP 342-5 and power supply PS 307 10A are configured in a RACK 300.
- A PROFIBUS-DP sub-network with master system No. 1 (connected to the CPU) and another sub-network with master system No. 180 (connected to the CP) are available. The drive controllers are connected to the second master system.
- Lenze fieldbus modules of type 2133 and 2131 are used as DP slave. These fieldbus modules can be used for 8200 and 9300 drive controllers.
  - PROFIBUS addresses: 4 and 5 (hexadecimal)
  - Consistent communication with 8-byte parameter data and 4 process data words (Configuration "PAR(Kons.)+PZD(4Worte)")

The following figure shows the hardware configuration in the Hardware Manager:





## 6.6.2 Program blocks used

The project folder **Blocks** contains the following program blocks:

Block	Function
OB1	Organisation block (program cycle); used to call <b>FC1</b>
FC1	Siemens FC block <b>DP_SEND</b>
FC2	Siemens FC block <b>DP_RECV</b>
FC3	Function 3; used to call all other blocks
FB91	Function block <b>FB DRIVE CP</b> enables parameter access via OPC and from the PLC
FC90	Function <b>FC Parameter</b> for parameter data jobs from the PLC
FC93/FC94	Function for DP Process Data (3 words/2 words)
DB80	Data block <b>DB OPC</b> for data exchange between OPC and <b>FB90</b>
DB81	Data block <b>DB S7PAR</b> for data exchange between <b>FC90</b> and <b>FB90</b>
DB30/31	Data blocks for data exchange between CPU and external communication processor (based on <b>UDT93</b> )
DB90/100	Instance data blocks for <b>FB91</b>
SFC57/SFC58	System blocks for communication via PROFIBUS-DP
UDT90 - UDT93	Universal data types for parameter data access to Lenze drive controllers
VAT4	Variables table for changing the example parameters

## 6.6.3 Symbolic variable names

The same symbolic variable names are used as in the previously described example project "S7-400".  
(□ 111)

We recommend the use of symbolic variable names for clarity. However, they are not compulsory.



## Busserver S7

### 6.6.4 OB1/FC3

Call the **F3** function in **OB1**. All other Lenze blocks are called from this function. Then call the Siemens FC blocks **DP\_SEND** and / or **DP\_RECV**, which are required for data exchange with the communication processor:

```

CALL FC 1
CPLADDR      :=W#16#100
SEND         :=P#DB30.DBX 0.0 WORD 15
DONE         :=M1.0
ERROR        :=M1.1
STATUS       :=MW2

CALL FC 2
CPLADDR      :=W#16#100
RECV         :=P#DB31.DBX 0.0 WORD 15
NDR          :=M1.2
ERROR        :=M1.3
STATUS       :=MW3
DPSTATUS     :=MB12
  
```

The parameter *CPLADDR* needs the I/O address of the PROFIBUS-CP. The address is listed in the slot table of the Hardware Manager:

Slot				I-address	O-address
1	PS 307 10A	6ES7 307-1KA00-0AA0			
2	<b>CPU 315-2 DP</b>	<b>6ES7 315-2AF00-0AB0</b>	3		
2	<i>DP-Master</i>			<i>1023*</i>	
3					
4	CP 342-5	6GK7 342-5DA01-0XE0	4	256...271	256...271

- Please observe that the address data in the Hardware Manager are decimal values (here: 256), functions **FC1** and **FC2**, however, expect hexadecimal entries (here: 100).

Specify the data block used for data exchange between communication processor and CPU.

- In this example, **DB30** is used to send data and **DB31** to receive data. When **UDT93** (4 process data words) is used, 16 bytes are reserved for each drive controller in the data blocks.
- Use the parameter *RECV* to indicate the number of drive controllers (here: 15).



### Tip!

Each block can comprise a maximum of 240 bytes. Therefore a maximum of 15 drive controllers can be addressed by one block. The maximum number depends on the number of process data words used.

If more than 15 devices are to be supported, additional data blocks must be created for data exchange and the functions **FC1** and **FC2** must be cyclically activated for these blocks.

More information about **FC1** and **FC2** can be found in the STEP7 documentation.



## 6.6.5 Process data transfer

### FC3, network 1

The following call is used to transmit two process data words (2PZD).

- Process data word 1 is always the DRIVECOM control word.
- Process data word 2 can be freely selected.

```
CALL FC 94
FirstDriveDW      := 8
DriveSend_DB     := "DriveSend_DB"
DriveRecv_DB     := "DriveRecv_DB"
Enable           := "Enable"
QSP              := "QSP"
Reset            := "Reset"
Setpoint1       := "Setpoint1"
ControlWord      := "ControlWord"
RDY              := "RDY"
TRIP             := "TRIP"
ActualValue1    := "ActualValue1"
StatusWord      := "StatusWord"
```

- The parameter *FirstDriveDW* defines the byte position of the first data word of the connected drive controller.
  - For each drive controller, 8 data words are reserved in the data blocks **DriveSend\_DB** and / or **DriveRecv\_DB**. The first 4 data words are needed for parameter access and the last 4 data words for process data access.
  - The drive controllers must have successive I/O addresses.
- The inputs *Enable*, *QSP* and *Reset* are set to enable the controller:
  - *Enable* = TRUE
  - *QSP* = FALSE
  - *Reset* = FALSE
- The second process data word is assigned to the setpoint input.
- Control word and status word conform to the DRIVECOM specification.
- The DRIVECOM status machine has already been implemented in **FC94**.
- The outputs *RDY* and *TRIP* provide information about the controller status.
- *ActualValue1* is the value read out.



### Tip!

Detailed information about the parameters of **FC94** can be found in chapter 6.3.7. (☞ 107)  
Use function block **FC93** to transmit three process data words. (☞ 106)



## Busserver S7

### 6.6.6 Parameter data access

The **FB91 (FB DRIVE CP1)** enables access parameter data both via OPC and from the PLC. The changeover between these two communication channels takes place automatically. The block works in the same way as **FB90**.

- One instance each of **FB91** is to be called for each drive (DP Slave). In addition, one instance data block for each drive is required for variables storage.
- Parameter data jobs via OPC are saved in **DB OPC**.
- Parameter data jobs from the PLC are saved in **DB S7PAR**.

#### FC3, network 2

```

L      "StatusWord"
L      0
==I
SPB   NEXT

      CALL FB 91 , DB100
      Tag          :=
      PB_Address   :=B#16#9
      FirstDriveDW :=0
      DriveSend_DB :="DriveSend_DB"
      DriveRecv_DB :="DriveRecv_DB"
      DriveServer_DB :="DB-OPC-Server"
      DrivePar_DB  :="S7-Parameter"
      TimerNr      :="TimeOutTimer_OPC"
      TimeOut      :=S5T#1S
      Busy         :="BUSY"

NEXT: NOP 0

```

- Use the *StatusWord* to check if the drive controller is switched on (*StatusWord* <> 0). If it is not, the call of **FB91** is skipped.
- In the above example, **FB91** uses the instance data block **DB100** for variables storage.
- The input *Tag* is not assigned.
- The PROFIBUS device address and the data blocks for data exchange with the communication processor and the byte offset are indicated as shown above.
- In the above example, the data blocks **DB80** ("**DB-OPC-Server**") and **DB81** ("**S7-Parameter**") are used for **DB OPC** and **DB S7PAR**.
- The output byte *Busy* provides information about the communication status.



#### Tip!

Detailed information about the parameters of **FB91** can be found in chapter 6.3.2. (📖 101)



## 6.6.7 Reading/writing a parameter

For reading and writing of parameters it is not important whether the device is connected to the PROFIBUS of the CPU or to the PROFIBUS of an external communication processor. In both cases, the function **FC90** is used.



### Tip!

Detailed information

- about reading a parameter can be found in chapter 6.5.7. (📖 114)
- about writing a parameter can be found in chapter 6.5.8. (📖 115)
- about the parameters of **FC90** can be found in chapter 6.3.3. (📖 102)



## Busserver S7

### 6.7 FAQ - Frequently asked questions and answers

Question	Answer
Which baud rate is to be set for the MPI interface module in the PG/PC control?	Depends on the interface module and the PLC. (Standard: 187.5 kbit/s)
Is it possible to change the block numbers of the Lenze blocks if they have already been assigned in the current STEP7 project?	Yes, it is.
Can the bus server S7 only work with the block numbers known to it?	No, the bus server S7 also works with modified block numbers.
Can the bus server S7 also find drives, if the PLC is in STOP mode?	Yes, but the PLC must have been in RUN mode at least once before.
Can the DriveServer also find drives and parameters if the PLC is in STOP mode?	No, to find drives and parameters, the PLC must be in RUN mode.
Which Siemens software is required for the bus server S7?	Depends on the application. For many applications, there is no particular Siemens software required. For further information, please read chapter 2.2. (□ 69)
How can I see, if the Siemens communication drivers have been installed on the PC?	If the Siemens communication drivers have been installed, the "Set PG/PC interface" configuration program is available in the control panel.
How can I see, which version of the Siemens communication drivers has been installed on the PC?	Open the "Set PG/PC interface" configuration program in the control panel. The version is indicated in the title bar of the program. If the version is not indicated, go to the Windows system directory, find the file "S7EPATDX.CPL" and list the version properties.
How do I install the Siemens communication drivers?	The Siemens communication drivers are part of the basic Siemens STEP7 package and are automatically installed when STEP7 is installed.
Which operating system should be used?	We recommend to use a Windows NT, Windows 2000 or Windows XP operating system. <ul style="list-style-type: none"> <li>• With the same hardware, the program execution times are much shorter under Windows NT/2000/XP than under Windows 98/Me.</li> <li>• When several computers are interconnected, it is not possible to start OPC servers automatically through a DCOM connection with Windows 98/Me.</li> </ul>
Which communication channel/driver should be used?	With some PC modules (e. g. PC adapter), you can choose which driver you want to use. If available, the Siemens communication driver should be used.
Is it possible to work simultaneously with STEP7 and the DriveServer/Global Drive Control?	Yes, it is, but the Siemens driver must be used for simultaneous operation. <ul style="list-style-type: none"> <li>• Since communication with S7 is, however, very often comparatively slow, the baud rate of the two applications might be reduced depending on the amount of data to be transmitted.</li> </ul>



Question	Answer
Can you see in the PLC program if the drive controller is switched on?	<p>Yes, you can see this by means of a status word. (□ 113)</p> <ul style="list-style-type: none"> <li>• A prerequisite for this kind of monitoring is the additional use of the process data channel.</li> <li>• In addition, the drive controllers should use an external 24V supply.</li> </ul>
Should the drive controllers use an external instead of an internal 24V supply?	<p>Yes, absolutely, otherwise PROFIBUS will slow down if devices are missing. In addition, this allows you to detect easily in the PLC program if drive controllers are switched on.</p>
Why is the error message "Can't make directory..." displayed in the DOS window when the "Ldslib.arj" library or the "Ldsexmpl.arj" example project is retrieved with the default STEP7 program "arj.exe"?	<p>The DOS program "arj.exe" can only retrieve archives up to a defined directory depth which might be exceeded if STEP7 is not installed in the root directory of the hard disk (e. g. "c:\...") but in a sub-directory (e. g. "c:\Programs\...").</p> <ul style="list-style-type: none"> <li>• Use the program "pkzip25.exe" part of the STEP7 delivery package and retrieve the ZIP archives "Ldslib.zip"/"Ldsexmpl.zip" instead of retrieving the ARJ archives "Ldslib.arj"/"Ldsexmpl.arj".</li> <li>• The program to be used for archiving is configured in the SIMATIC Manager via <b>Options→Settings</b>.</li> </ul>



## ***Busserver S7***