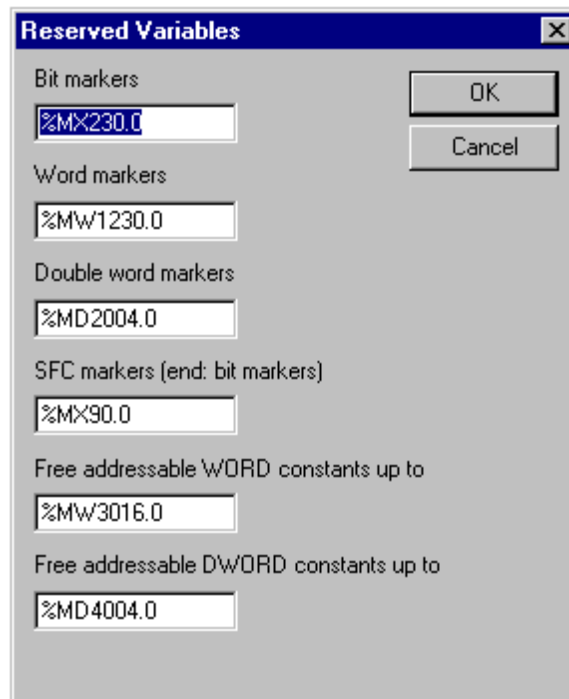


Peculiarities and Restrictions  
for the 40/50 Series





# Contents

<b>1</b>	<b>907 AC 1131 - peculiarities and restrictions for S40..50</b>	<b>3</b>
<b>1.1</b>	<b>Introduction</b>	<b>3</b>
1.1.1	Considerable differences	3
1.1.2	A small example program for S40..50	3
<b>1.2</b>	<b>Program structure for S40..50</b>	<b>14</b>
1.2.1	Program blocks S40..50	14
1.2.2	The main program PLC_PRG for S40..50	14
1.2.3	Subroutines for S40..50	14
1.2.4	Interrupt routines for S40..50	14
<b>1.3</b>	<b>Programming languages for S40..50</b>	<b>16</b>
1.3.1	Instruction List (IL)	16
1.3.2	Structured Text (ST)	16
1.3.3	Sequential Function Chart (SFC)	16
1.3.4	Ladder Diagram (LD)	17
1.3.5	Function Block Diagram (FBD)	17
<b>1.4</b>	<b>Peculiarities of the menus in 907 AC 1131 for S40..50</b>	<b>18</b>
1.4.1	Editors for S40..50	18
1.4.2	Creating and deleting objects	18
1.4.3	Project management	19
<b>1.5</b>	<b>Commands in the Online menu</b>	<b>20</b>
1.5.1	In offline mode available commands of the Online menu for S40..50	21
1.5.2	In online mode available commands of the Online menu for S40..50	28
<b>1.6</b>	<b>Peculiarities in the Resources tab</b>	<b>34</b>
1.6.1	Object libraries	34
1.6.2	Global variables	34
1.6.3	Controller configuration	35
1.6.4	Task configuration	35
1.6.5	Watch and recipe manager	36
<b>2</b>	<b>Data types and variables for S40..50</b>	<b>36</b>
<b>2.1</b>	<b>Data types in S40..50</b>	<b>36</b>
<b>2.2</b>	<b>Variables of S40..50</b>	<b>38</b>
<b>3</b>	<b>Operators and blocks for S40..50</b>	<b>40</b>

<b>3.1</b>	<b>Operators for S40..50 in 907 AC 1131 .....</b>	<b>40</b>
3.1.1	Overview: Operators for S40..50 .....	40
3.1.2	Blocks of the standard IEC library for S40..50 .....	42
<b>3.2</b>	<b>Block library for S40..50.....</b>	<b>43</b>
<b>3.3</b>	<b>Conversion of 907 PC 331 blocks to 907 AC 1131 S40..50.....</b>	<b>46</b>
<b>4</b>	<b>Index .....</b>	<b>50</b>

## 1.1 Introduction

### 1.1.1 Considerable differences

907 AC 1131 is a complete development environment for ABB Advant Controllers of the series 90 (S90) and 40..50 (S40..50).

The most considerable difference between PLCs of series 90 and series 40..50 is the operating system. For the series 90 the programming system 907 AC 1131 and the multitasking runtime system are inseparable and belong together. The processor code for series 90 is generated when translating the project. For series 40..50 the project **project.pro** is translated into IL code of the EBS operating system during translation. The translated code can be viewed in the ASCII file **project.cod** or **project.adr**. The code contained in this file is sent to the PLC.

Compared to series 90, the PLCs of series 40..50 have a smaller functional scope. As a result not all features offered by the programming system 907 AC 1131 can be used for series 40..50.

This description shows the considerable differences and restrictions.

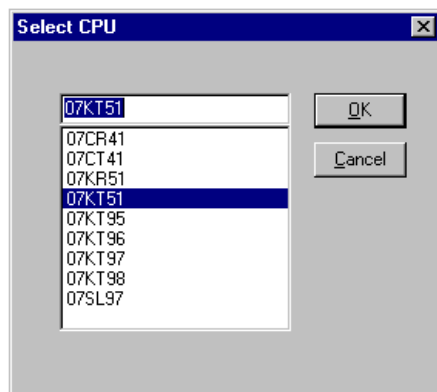
### 1.1.2 A small example program for S40..50

In order to give you an easy approach to the programming software 907 AC 1131 for series 40..50 the following descriptions show you how to create a small example program, load it into the controller and test it.

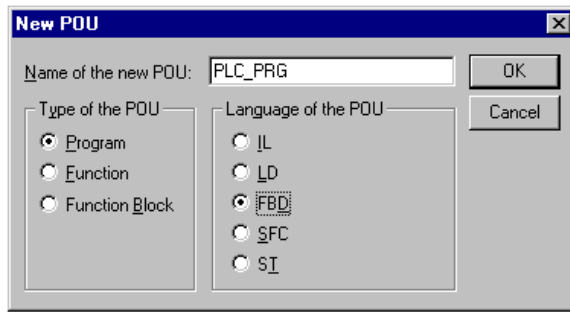
**Task:** Input E62\_00 shall be routed to output A62\_00 of the controller 07KT51.

Perform the following steps:

1. Connect the controller 07 KT 51 to 24 V DC (refer to volume 1 – Hardware S40..50).
2. Connect COM1 of the controller 07 KT 51 to the serial interface of your PC using the system cable 07 SK 50.
3. Start the 907 AC 1131 software. The project you edited last is opened. If necessary, close this project by selecting "File/Close".
4. Select "File/New".
5. A window appears where you can select the PLC. Using the mouse, select the controller "07KT51" and click on "OK".

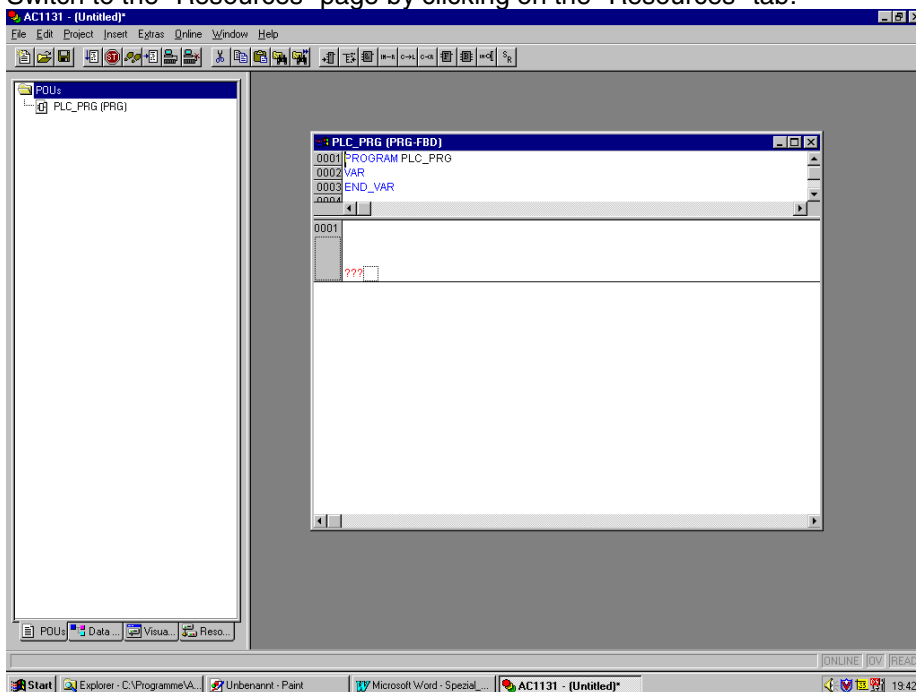


- Then a window appears where you can select the block name, the block type and the PLC language. For the name and the type keep the settings PLC\_PRG and PROGRAM. Select "FBD" for the language and confirm with "OK".

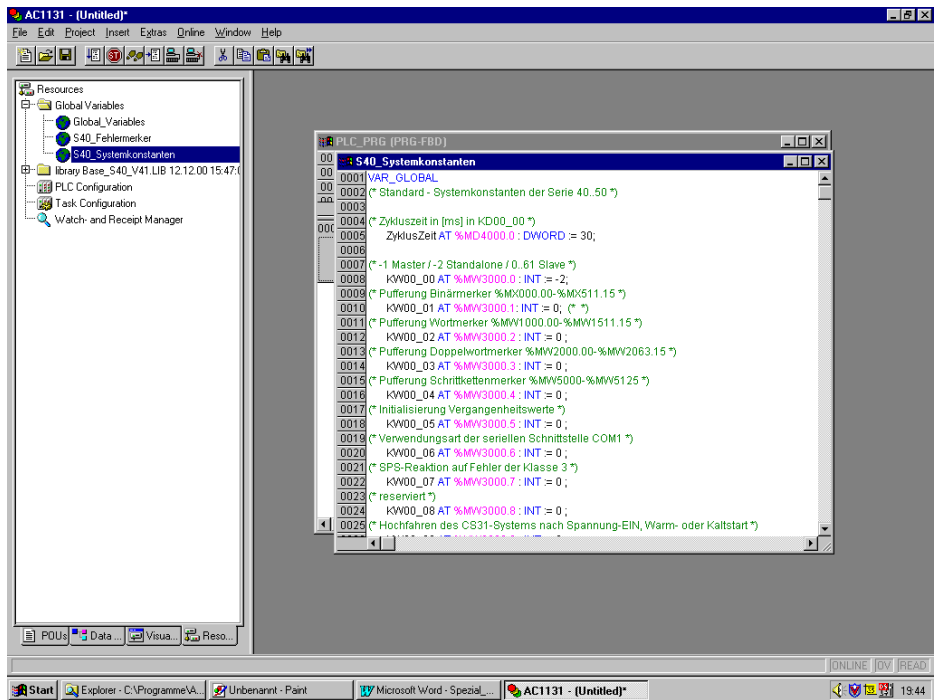


- You have now opened a new project. When inserting the block PLC\_PRG the following global variables were already created in the folder "Global Variables" of the "Resources" page.
  - Object "S40\_Fehlermerker" with all system and error flags of S40..50
  - Object "S40\_Systemkonstanten" with the default values of the system constants

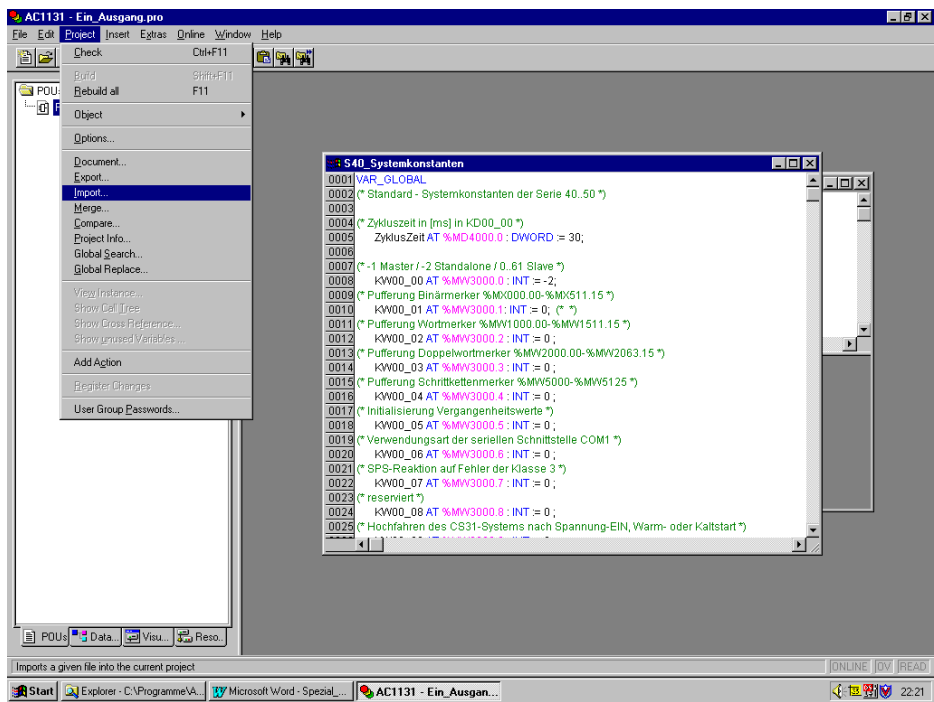
Switch to the "Resources" page by clicking on the "Resources" tab.



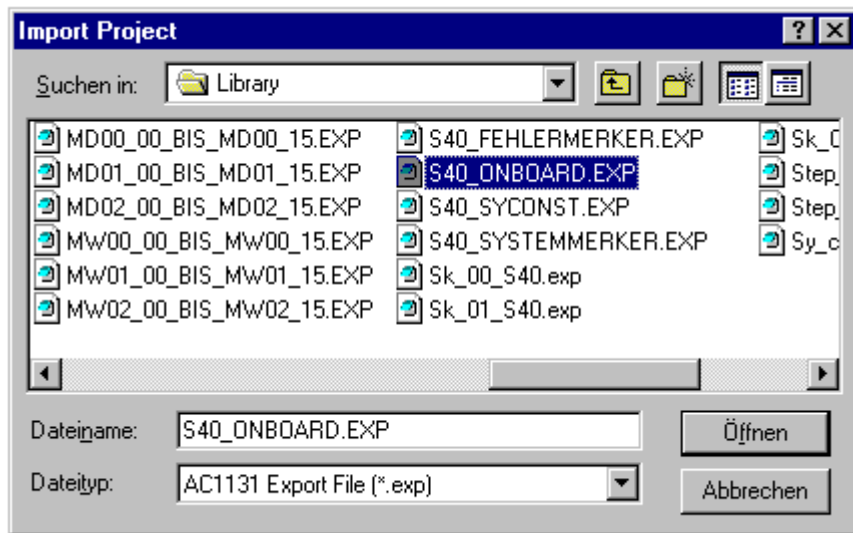
In this page, click on "Global Variables" or open the respective folder. You can see the respective variables if you open the individual objects with a double click. (In the example the object "S40\_Systemkonstanten" was opened.)



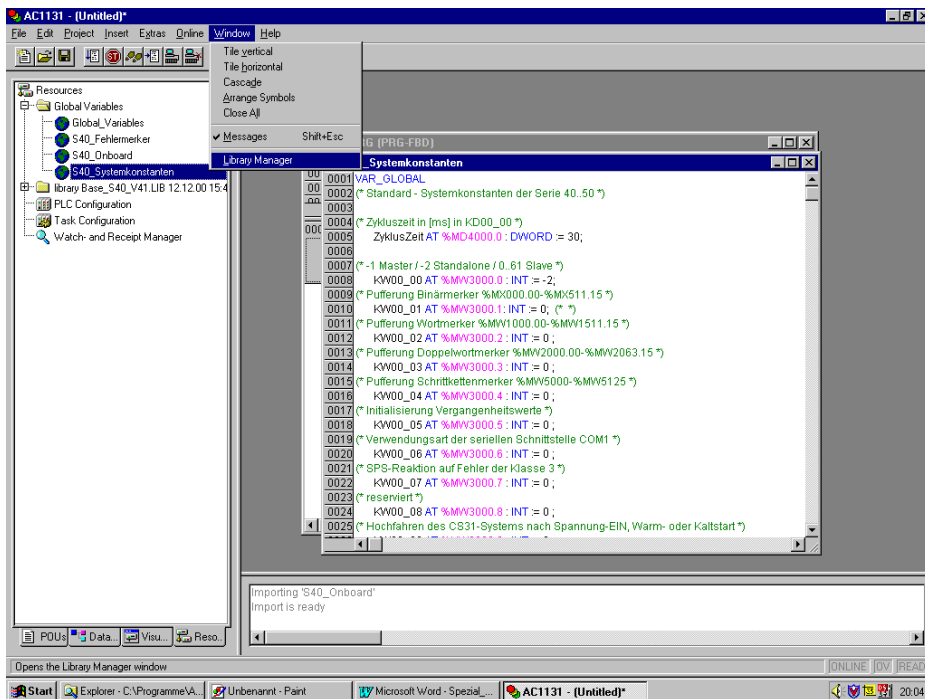
In the "Resources" page, select the folder "Global Variables".  
Now select "Project/Import".



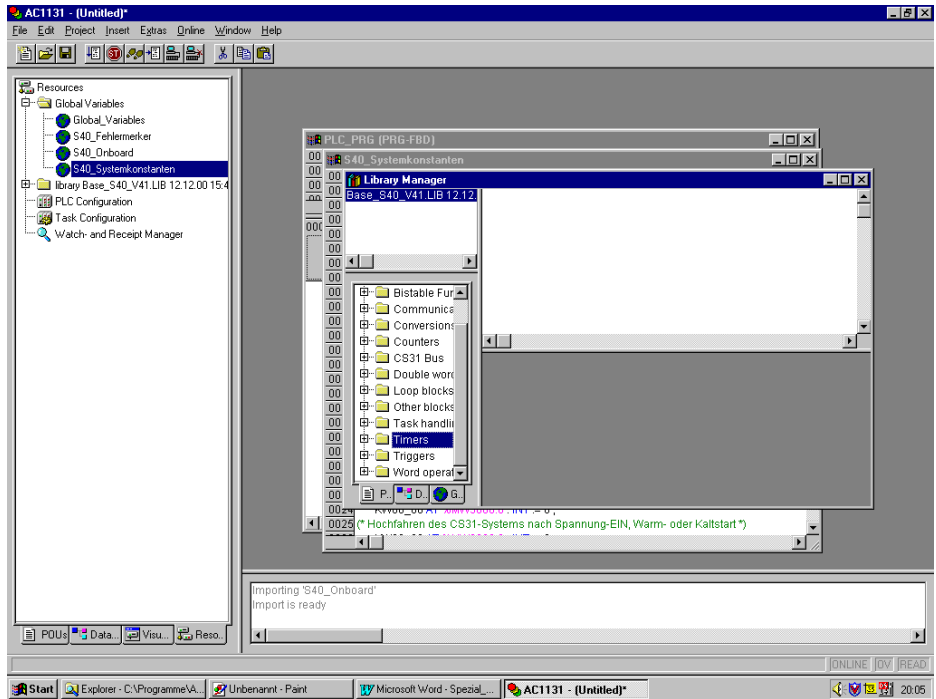
All \*.exp files are displayed now (in the Library directory). Select "S40\_ONBOARD" and confirm it by clicking on "Open". Now the import file "S40\_ONBOARD.exp" is loaded. This file contains the declaration of all onboard in- and outputs of the PLCs S40..50.



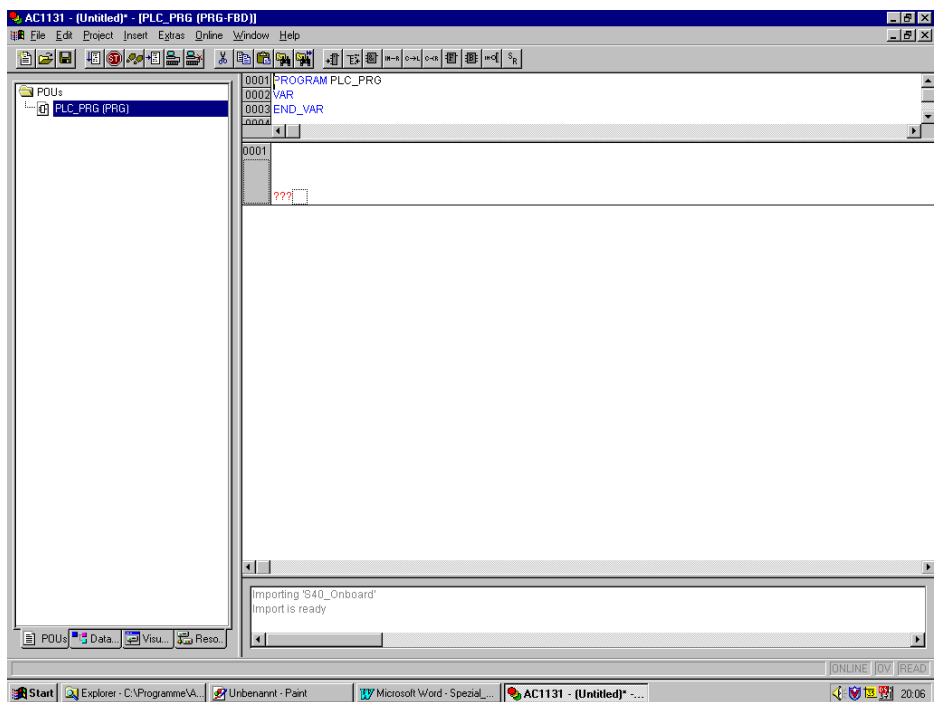
When you create a new project the library Base\_S40\_V41.LIB is also loaded into the library management. Select "Window/Library Manager".



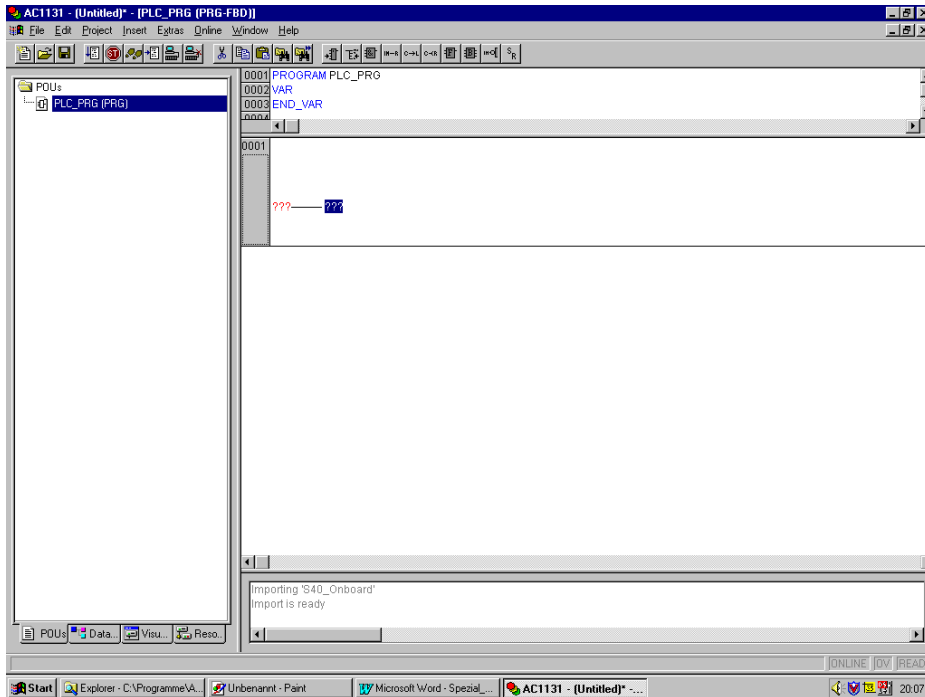
In the upper left window of the library management you can see "Base\_S40\_V41.LIB".



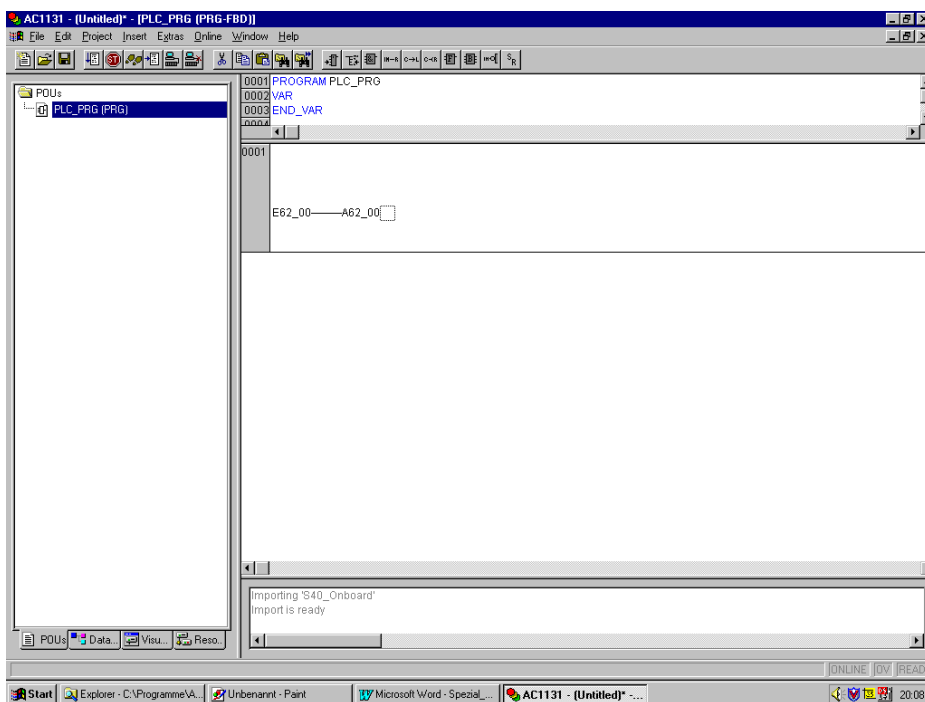
- Switch back to the POU's by clicking on the "POUs" tab and select the only existing entry "PLC\_PRG".



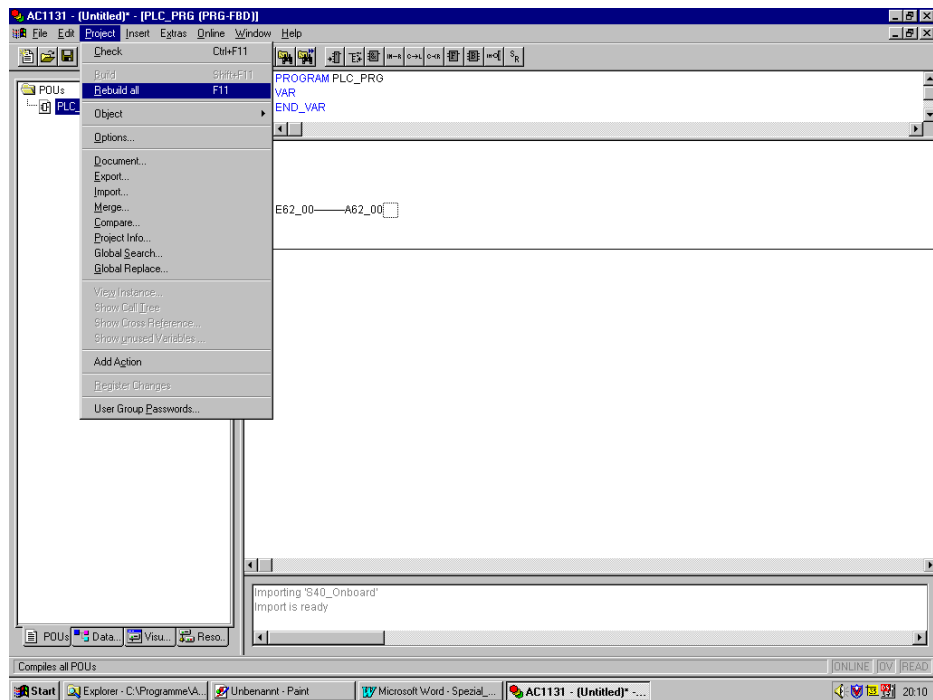
- In the network "0001", position the mouse cursor on the project cursor (which is represented by a "quadrilateral") and press <CTRL><A> (press the keys <CTRL> or <STRG> and <A> simultaneously) or click with the right mouse button and then select "Assign" from the appearing context menu. The display is as following:



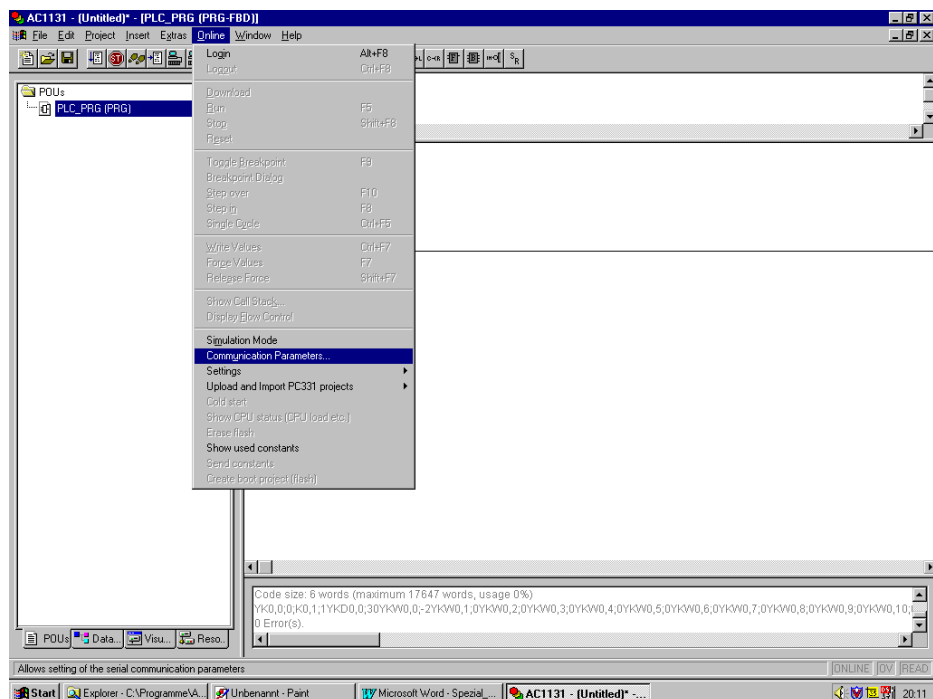
- Instead of the question marks, enter the symbolic name for input E62\_00 on the left side and the name for output A62\_00. To do this, left-click on the question marks, type the name and confirm with <Enter>. These names are predefined in the import file S40\_ONBOARD.exp. If you want to use other symbolic names, you have to change the names in the object "S40\_Onboard" located in the folder "Global Variables" of the "Resources" tab. After changing the names you must translate the project by pressing the <F11> key or by selecting "Project/Rebuild all" from the menu in order to announce the global variables to the block(s).



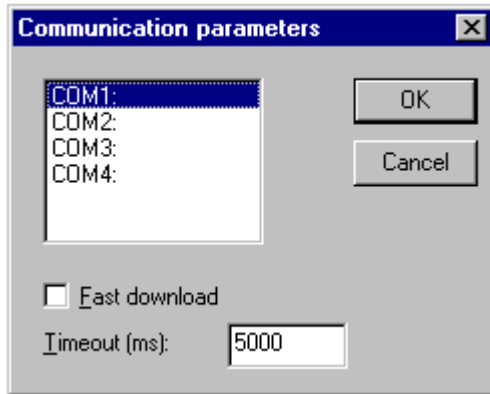
- The definition of our project is now completed and the project has to be translated. To do this, press the <F11> key or select "Project/Rebuild all" from the menu.



- If no errors were detected the message "0 errors" is displayed in the last line of the message box. The used constants, the program size in words, the maximum program size and the memory utilization in [%] are also displayed. Now the communication parameters for the connection to the controller have to be set. To perform this, please click on "Online/Communication Parameters".

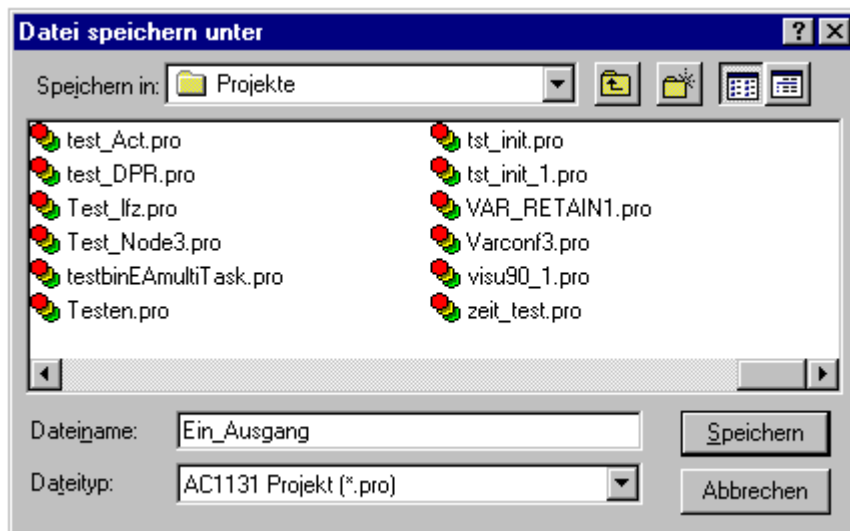


A window for setting the communication parameters appears.



Select the serial PC interface (usually COM1). Timeout must be set to  $\geq 5000$  ms. **Do not** select "Fast download". Click on "OK" to apply the communication parameters.

13. In the next step we will save the project. You can also save the program already during creation. For this, select "File/Save as" and then specify the desired path and project name:



In our example the project is saved under the name "Ein\_Ausgang.pro" to the folder "Projekte". The file extension "pro" is automatically created by the 907 AC 1131 software.




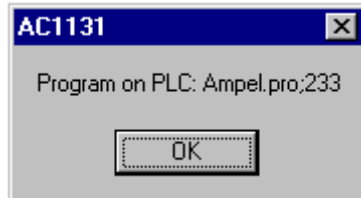
**Note:**

Do not use any **umlaut (ä, ö, ü) or special characters for the project name**. This may possibly cause a loss of the project!

14. In the next step we will load the program into the controller.

Because in the 907 AC 1131 software the variables status is read from all open windows (even if they are in the background) you should close all windows before logging into the controller and open just the window where you want to have the status indication. To close the windows select "**Window/Close all**". The PLC\_PRG is opened again by double clicking on the name.

To login please select "Online/Login" or the icon . After the connection is established a window appears displaying the project name in the controller and its checksum:

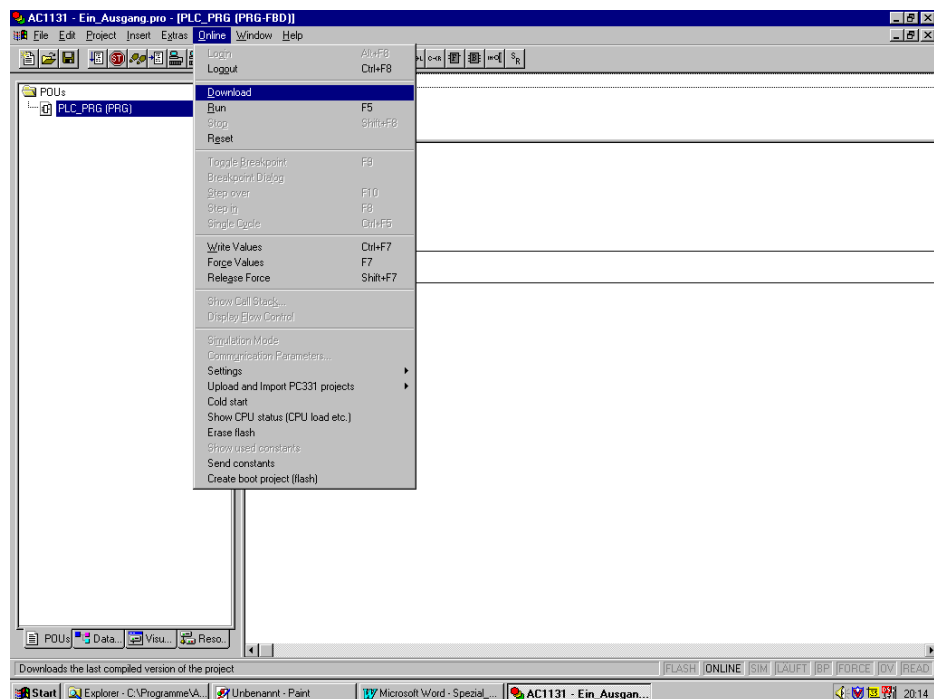


Confirm with "OK". The variables status is displayed on the screen.

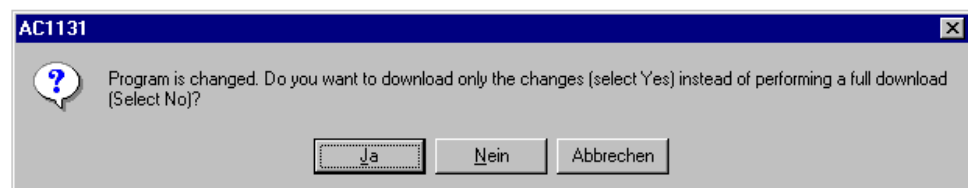
If the message "Communication error / Logout" is displayed, the connection to the controller could not be established. If a second attempt fails as well, proceed as described under "Offline mode commands in the Online menu for S40..50 / Communication Parameters" in this documentation.

The status bar of 907 AC 1131 displays "**ONLINE**" in black. If the field "**LÄUFT**" is displayed in black the controller is in the RUN mode.

Select "Online/Download" from the menu to load the program.




Wait for the next message:




Moving the mouse too fast after selecting "**Online/Download**" can cause the message to be minimized immediately and displayed in the Windows status bar as a "AC1131" switch. In this case click on this switch in the Windows status bar to display the message again.

Now you can select between "Online change"→"Ja" ("Yes") and "Download"→"Nein" ("No"). Clicking on "Abbrechen" ("Cancel") causes an immediate logout.

For S40..50 controllers of the category R02xx no "Online change" can be performed. Furthermore this category does not allow flashing during RUN mode. In this case you must stop the controller by selecting "**Online/Stop**" or clicking on the icon  before the download is performed.

To perform a download please select "No". After expiration of the Timeout time the program is downloaded. You can follow the download process by viewing the progress indication. However, the progress indication can hardly be seen for such a small example program.

After the download the program is flashed. This process lasts as long as the 907 AC 1131 status indication displays the field "**FLASH**" in black.

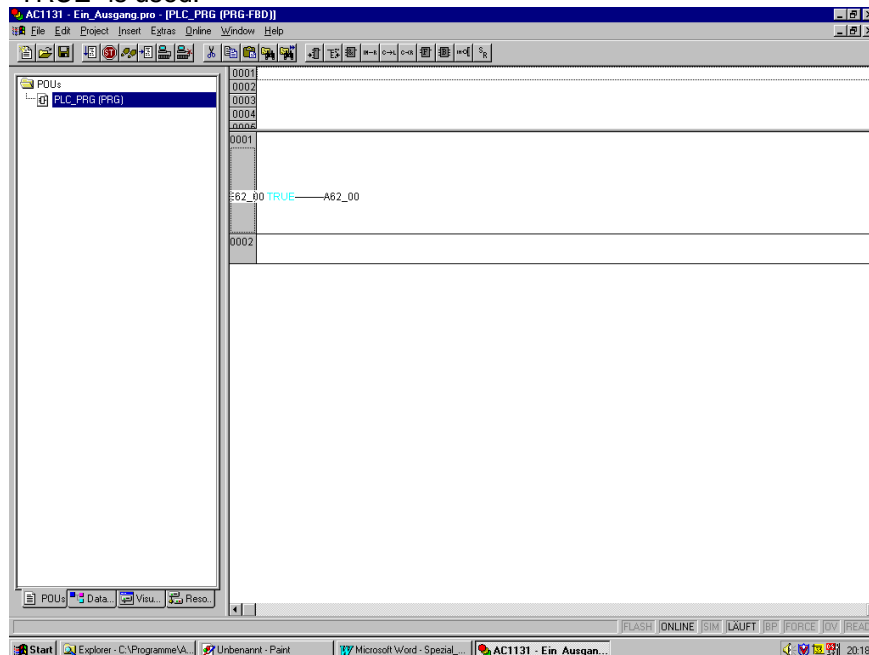
15. Now the program has to be started. To do this select "Online/Start" or click on the icon . The 907 AC 1131 status bar displays "LÄUFT" in black color and the LED "RUN" on the controller lights up.

If the following message is displayed the RUN-STOP switch is set to position "1".



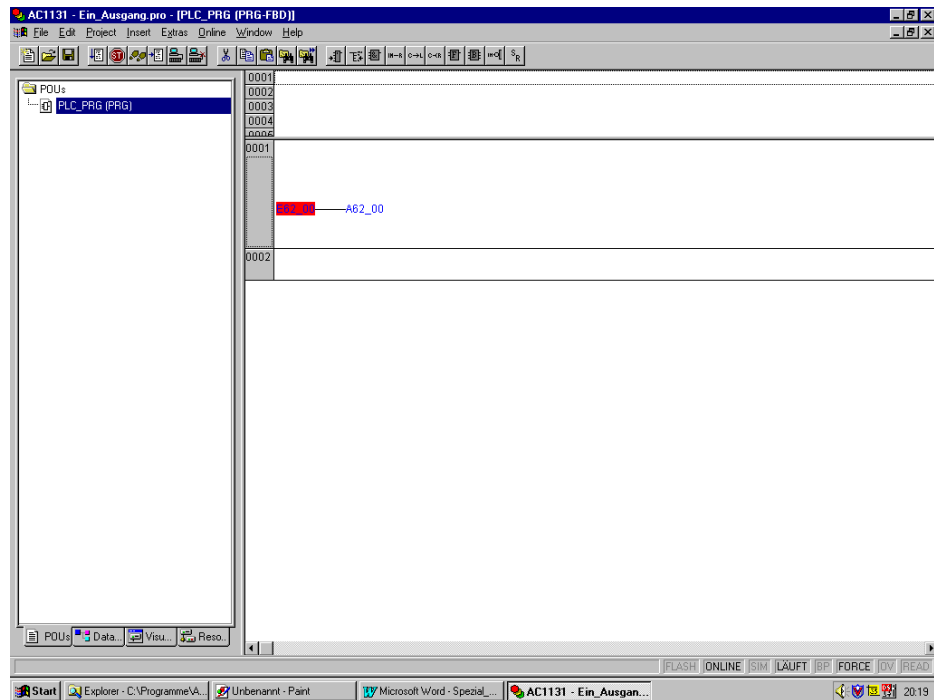
Set the switch to the "ON" position.


16. In order to test the program we are going to force input E62\_00 to TRUE. To do this, please double click on the icon "E62\_00". Next to "E62\_00" the entry "TRUE" is displayed. After performing another double click, "FALSE" is displayed. Double clicking once more releases forcing. This way you can choose the desired value for forcing the variable. In our example "TRUE" is used.



Now the variable E62\_00 is prepared for forcing.

Press the <F7> key or select "Online/Force values" from the menu. This will force the variable. On the PC, A62\_00 becomes blue and the output OUT 0 of the PLC is connected through. The LED OUT 0 lights up. To display the forcing process the 907 AC 1131 status bar displays the field "FORCE" in black.



Now you can force the input to FALSE again or "Release forcing". Using "Online/Logout" or the icon  the connection to the PLC is disconnected again. If nothing is selected under "Online/Settings for Force", forcing is released again when logging out.

17. To exit the software 907 AC 1131 use "File/Exit".

## 1.2 Program structure for S40..50

### 1.2.1 Program blocks S40..50

907 AC 1131 projects for series 40..50 may only contain the following program blocks:

- The main program - **PLC\_PRG**
- A maximum of **12 subroutines**
- A maximum of **3 interrupt routines**

All blocks are of the type Program. **Functions** and **function blocks** cannot be programmed.

The blocks can be programmed in different IEC languages. However, inside of one block the languages cannot be mixed.

**Cycle time setting** is not performed in the task configuration but using the system constant KD00,00 / %MD4000.0.

### 1.2.2 The main program PLC\_PRG for S40..50

In projects for series 40..50 the main program is always the program block **PLC\_PRG**. It must not be renamed for S40..50.

The main program can contain logical connections, library block calls, subroutine calls and the release of interrupt tasks.

### 1.2.3 Subroutines for S40..50

Subroutines are objects in the "Blocks" tab. They must be of the **type Program** and must not have any transfer parameters. Functions and function blocks are not allowed.

In the main program these subroutines must be called by their name. Subroutines may only be called from the main program. A subroutine never can call another subroutine.

One subroutine can be called from the main program at least one time and maximally 255 times.

### 1.2.4 Interrupt routines for S40..50

In the S40..50 PLCs two hardware interrupts and one cyclic (time controlled) interrupt are possible.

Interrupt routines are objects in the "Blocks" tab. They must be of the type Program and must not have any transfer parameters. Functions and function blocks are not allowed.

For the block names the following rules apply:

- **Hardware interrupt 1:** rising edge of input E62\_02 / %IX62.2  
Block: **INT1**
- **Hardware interrupt 2:** rising edge of input E62\_03 / %IX62.3  
Block: **INT2**

- Interrupt 3..9: reserved  
Blocks: INT3 .. INT9 (reserved)
- **Interrupt 10:** cyclic call  
Block: **INT10\_100**  
The value 100 means that the call is performed every 100 ms.  
The minimum call interval is 1 ms.

If the blocks are renamed they are translated like normal subroutines.

Enabling or disabling of interrupt routines is performed in the main program PLC\_PRG using the block **VTASK**. Only one VTASK block is allowed per interrupt. There is only one priority for the interrupts.

Inside of an interrupt routine **no time functions** such as ESV, TON, ASV may be called.

## 1.3 Programming languages for S40..50

### 1.3.1 Instruction List (IL)

The following restrictions for series 40..50 apply to this language:

- No brackets
- Limited range of operators (refer to section 3 - Operators for series 40..50)
- Limited range of data types and variables (refer to section 2 - Data types and variables for series 40..50)
- No back jumps
- A maximum of 42 forward jumps via other jumps / labels

### 1.3.2 Structured Text (ST)

The following restrictions for series 40..50 apply to this language:

- Limited range of operators (refer to section 3 - Operators for series 40..50)
- Limited range of data types and variables (refer to section 2 - Data types and variables for series 40..50)
- No CASE, FOR, WHILE, REPEAT and EXIT instructions

### 1.3.3 Sequential Function Chart (SFC)

The following restrictions for series 40..50 apply to this language:

- Limited range of operators (refer to section 3 - Operators for series 40..50)
- Limited range of data types and variables (refer to section 2 - Data types and variables for series 40..50)
- No IEC steps

### 1.3.4 Ladder Diagram (LD)

The following restrictions for series 40..50 apply to this language:

- Limited range of operators (refer to section 3 - Operators for series 40..50)
- Limited range of data types and variables (refer to section 2 - Data types and variables for series 40..50)
- No back jumps
- A maximum of 42 forward jumps via other jumps / labels

### 1.3.5 Function Block Diagram (FBD)

The following restrictions for series 40..50 apply to this language:

- Limited range of operators (refer to section 3 - Operators for series 40..50)
- Limited range of data types and variables (refer to section 2 - Data types and variables for series 40..50)
- No back jumps
- A maximum of 42 forward jumps via other jumps / labels

## 1.4 Peculiarities of the menus in 907 AC 1131 for S40..50

### 1.4.1 Editors for S40..50

For series 40..50 all commands of the "Edit" menu are available in the editors as well as in the Organizer object.

#### *Declaration editor*

Declaration editors are used for the variables declaration of blocks and global variables as well as in the watch and recipe manager.

For series 40..50 only the variable declaration keywords **VAR** and **END\_VAR** or **VAR CONSTANT** and **END\_VAR** exist.

Declaration	Available in S40..50	Note
VAR_INPUT	no	
VAR_OUTPUT	no	
VAR_IN_OUT	no	
VAR	yes	
VAR RETAIN	no	Buffered data areas are determined by system constants
VAR CONSTANT	yes	

Refer to section "Data types and variables for S40..50" for additional notes about variables for S40..50.

#### *Text editor and graphical editors*

All functions described in the programming manual also apply to S40..50.

### 1.4.2 Creating and deleting objects

The following restrictions apply to the functions in the "Project" / "Object" menu (or right mouse button in the Organizer object):

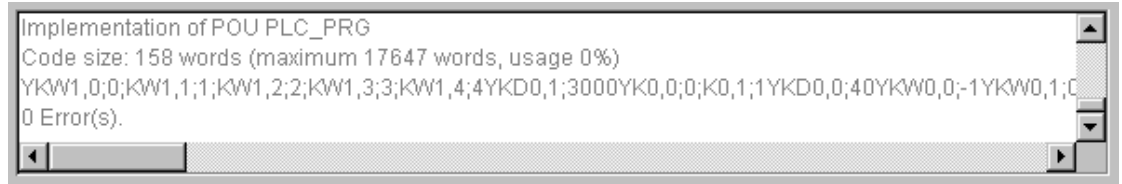
- "Add action" – there are no actions for S40..50.
- "View Instance" – without function because function blocks do not exist.
- "Convert Object" – possible, but limited.

### 1.4.3 Project management

The project management commands are contained in the "File" and "Project" menus. For S40..50 the same functions are available than for S90.

In contrast to series 90 the project information (menu "Project" / "Info") is not transferred to the controller. Only the project name and a checksum are kept on the controller.

When translating a project for S40..50 the following information are displayed in the message window.



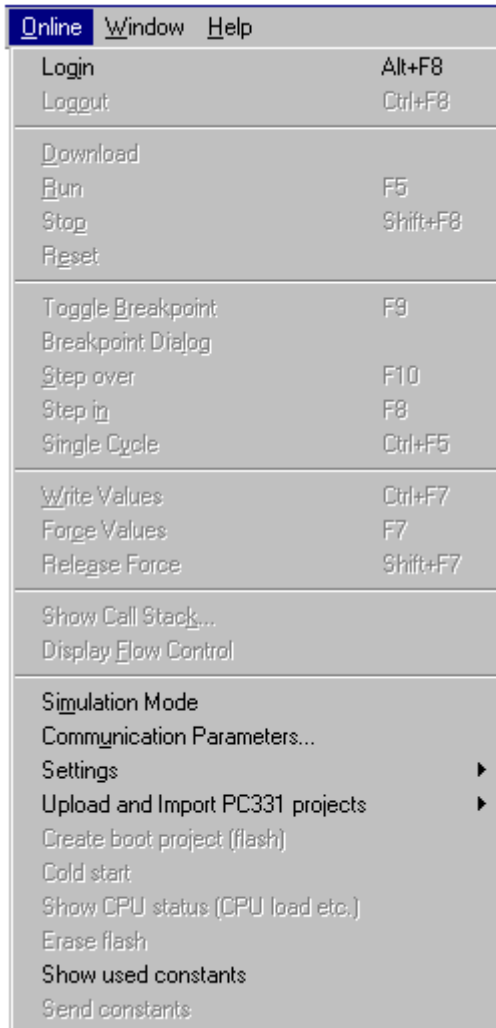
```
Implementation of POU PLC_PRG
Code size: 158 words (maximum 17647 words, usage 0%)
YKW1,0;0;KW1,1;1;KW1,2;2;KW1,3;3;KW1,4;4YKD0,1;3000YK0,0;0;K0,1;1YKD0,0;40YKW0,0;-1YKW0,1;C
0 Error(s).
```

For the code size the size of the ASCII IL is displayed in words and as a percentage value. Additionally also the used constants are displayed.

## 1.5 Commands in the Online menu

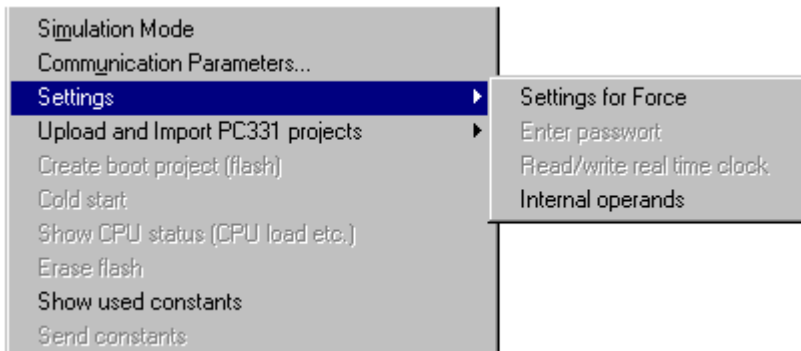
If you have entered a series 40..50 PLC in the controller configuration, the standard Online menu is expanded by the special functions for series 40..50.

The **Online** menu is as following:

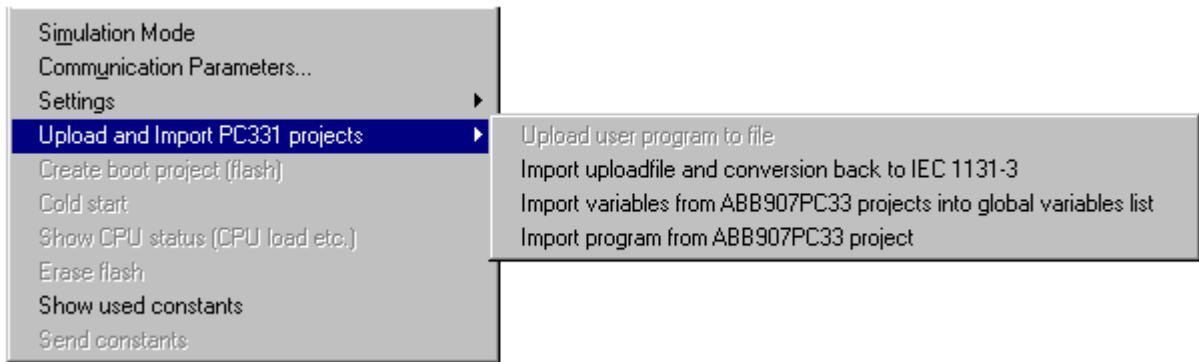


Most of these Online commands are available on series 40..50.

In the menu **"Settings"** are the following commands:



In the menue "**Upload and Import PC331 projects**" are the following commands:



### 1.5.1 In offline mode available commands of the Online menu for S40..50

The following commands are available in the logged-off state (offline mode):

- Login
- Simulation Mode
- Communication Parameters
- Settings for Force
- Internal operands
- Show used constants
- Import upload file and conversion back to IEC 1131-3

#### ***Login:***

After selecting the menu item "Login" the connection to the PLC is established. A window appears containing the following message: „Program on PLC: project\_name.pro;checksum“. This message displays the name and checksum of the project on the PLC.

After successful completion of the login the values of the visible variables are displayed. Of course, the variables must be declared using the valid data types for S40..50 (BOOL, INT, DINT, TIME).

In contrast to series 90 it is not automatically checked if there are differences in the project on the PC and on the PLC. As a result the dialog for sending the program is not automatically switched on.

For series 40..50 the communication between the PLC and the PC is not performed via the GATEWAY. As a result the Windows status bar does not display the GATEWAY icon.

In 907 AC 1131 the **variables of all opened windows** are transferred even if they are not visible. In order to obtain an effective status indication you should open only one window, if possible. For this, select "**Windows/Close all**". Then open the desired window.

If the message "**Communication error, logout**" is displayed when logging in, please check the connection between PLC and PC and the communication parameters in the menu "Online" / "Communication Parameters".



**Note:**

If the cables and the communication parameters are OK you should check the communication between PLC and PC using a **terminal program (e.g. Hyperterminal)**. Set the communication parameters to 9600 baud, 8 data bits, 1 stop bit, no parity. After pressing the <ENTER> key the response "Falsche Eingabe" should be displayed. If it is not possible to establish a connection to the controller, please check again the connecting cable (07 SK 50) and the availability of the serial interface of the PC.

**Settings for COM1 e.g. for Windows NT V4.0:**

- Select "Start" => "Programs" => "Administrative Tools (Common)" => "Windows NT Diagnostics" and from the tab "Services" select "Devices". The "Device" "Serial" must have the "State" "Running".

- Your serial interfaces (e.g. COM1 and COM2) should be displayed under "Start" => "Programs" => "Control Panel" => "Ports". Select the port (e.g. COM1) and then the "Settings": 9600 Baud, 8 data bits, no parity, 1 stop bit, no flow control. After selecting "Advanced" the "Advanced Settings for COM1" are displayed:

COM Port Number: 1

Base I/O Port Address: Default

Interrupt Request Line (IRQ): Default

**FIFO Enabled:** Checked for „yes“

- Under "Start" => "Settings" => "Control Panel" => "Devices" search the "Device" „Serial“. It should have the "State" "Started" and the "Startup" "Automatic".

## Simulation:

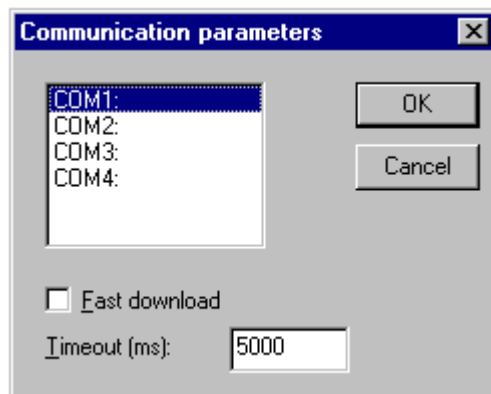
In order to simulate an user program the simulation mode can be selected. However this is accompanied by considerable restrictions:

1. The blocks defined in the S40..50 library Base\_S40\_Vxx.LIB cannot be used in simulation mode. The same applies to the external libraries of series 90.
2. For the simulation all data types of 907 AC 1131 can be handled. However, in the controllers of series 40..50 only a considerably limited number of data types exists (see also section 2 of this documentation).
3. The simulation enables all types of blocks (program, function, function block). On the controllers only programs and some standard function blocks (IEC or ABB function blocks) can be executed. Users cannot create their own functions or function blocks. The program blocks must not contain parameters (see also section "Program structure for S40..50").
4. Interrupt routines cannot be simulated.
5. All operators and standard IEC functions and function blocks listed in section 3 can be used for simulation. Only a limited number of them is available on the controllers.
6. The restrictions for the IEC languages ST, IL, FBD, LD and SFC are mentioned in section 1.

## Communication Parameters:

The communication parameters are set to the fix values 9600 baud, no parity, 8 data bits, 1 stop bit, no handshake.

Because the communication is not performed via GATEWAY and programming is only possible via the serial interface the following dialog appears after selecting the menu item "Communication Parameters".



Select the serial interface and confirm with OK. To obtain faster transfer times for long programs you can select (✓) the option "Fast download". Timeout should be set to  $\geq 5000$  ms.

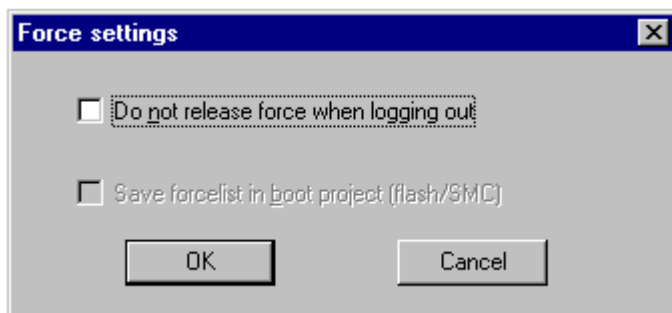


### Note:

The "Fast download" option does **not yet** work in the present version.

### ***Settings / Settings for Force:***

When selecting this function the same dialog appears than for series 90.

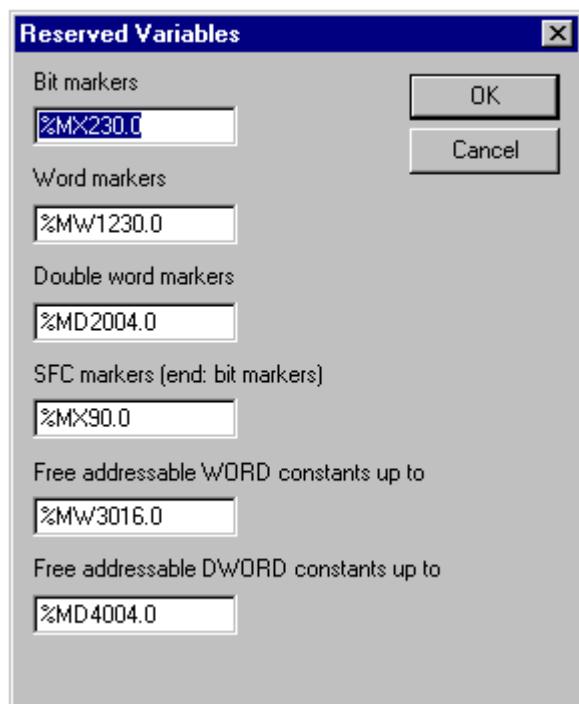


Usually the forced values are deleted when logging out. If you want to keep the forced values after logging out, select the option "Do not release force when logging out" in this dialog. The setting "Serve force list in boot project (Flash/SMC)" is only possible for series 90 and therefore cannot be selected.

### ***Settings / Internal operands:***

In the "Internal operands" menu the flag ranges reserved for compilation are entered. These are the intermediate flags from the M, MW and MD range as well as ranges from the KW and KD range for the conversion of the direct constants.

After selecting the menu item the following dialog appears:



The values entered here are default values and should match most of the applications.

The following settings can be performed:

1. **Bit markers from (up to %MX254.15):**

Default setting: %MX230.0

The flags M230,00 / %MX230.0 .. M254,15 / %X254.15 are reserved for intermediate flags. The range from M00,00 / %MX0.0 up to the beginning of the SFC flags can be used by the user's choice.

The address of the first flag of the reserved range is entered.

2. **Word markers from (up to %MW1253.15):**

Default setting: %MW1230.0

The flag words MW230,00 / %MW1230.0 .. MW253,15 / %MW1253.15 are reserved for intermediate flags.

The range MW00,00 / %MW1000.0 .. MW229,15 / %MW1229.15 can be used by the user's choice.

The address of the first reserved word is entered.

3. **Double word markers from (up to %MD2007.15):**

Default setting: %MD2004.0

The flag double words MD04,00 / %MD2004.0 .. MD07,15 / %MD2007.15 are reserved for intermediate flags.

The range MD00,00 / %MD2000.0 .. MD03,15 / %MD2003.15 can be used by the user's choice.

The address of the first reserved double word flag is entered.

4. **SFC markers (end: bit markers):**

Default setting: %MX90.0

The flags M90,00 / %MX90.0 up to the beginning of the bit flags are reserved as special flags for sequential function chart.

The range M00,00 / %MX0.0 .. M89,15 / %MX89.15 can be used by the user's choice.

The address of the first flag of the reserved range is entered.

5. **Free addressable WORD constants up to:**

Default setting: %MW3016.0

The word constants KW00,00 / %MW3000.0 .. KW00,15 / %MW3000.15 are reserved for system constants.

The range KW01,00 / %MW3001.0 .. KW15,15 / %MW3015.15 can be used by the user's choice.

The range KW16,00 / %MW3016 .. KW31,15 / %MW3031.15 is reserved for converting the direct constants.

6. **Free addressable DWORD constants up to:**

Default setting: %MD4004.0

The double word constant KD00,00 / %MD4000.0 is reserved as a system constant.

The range KD00,01 / %MD4000.1 .. KD03,15 / %MD4003.15 can be used by the user's choice.

The range KD04,00 / %MD4004 .. KD07,15 / %MD4007.15 is reserved for converting the direct constants.

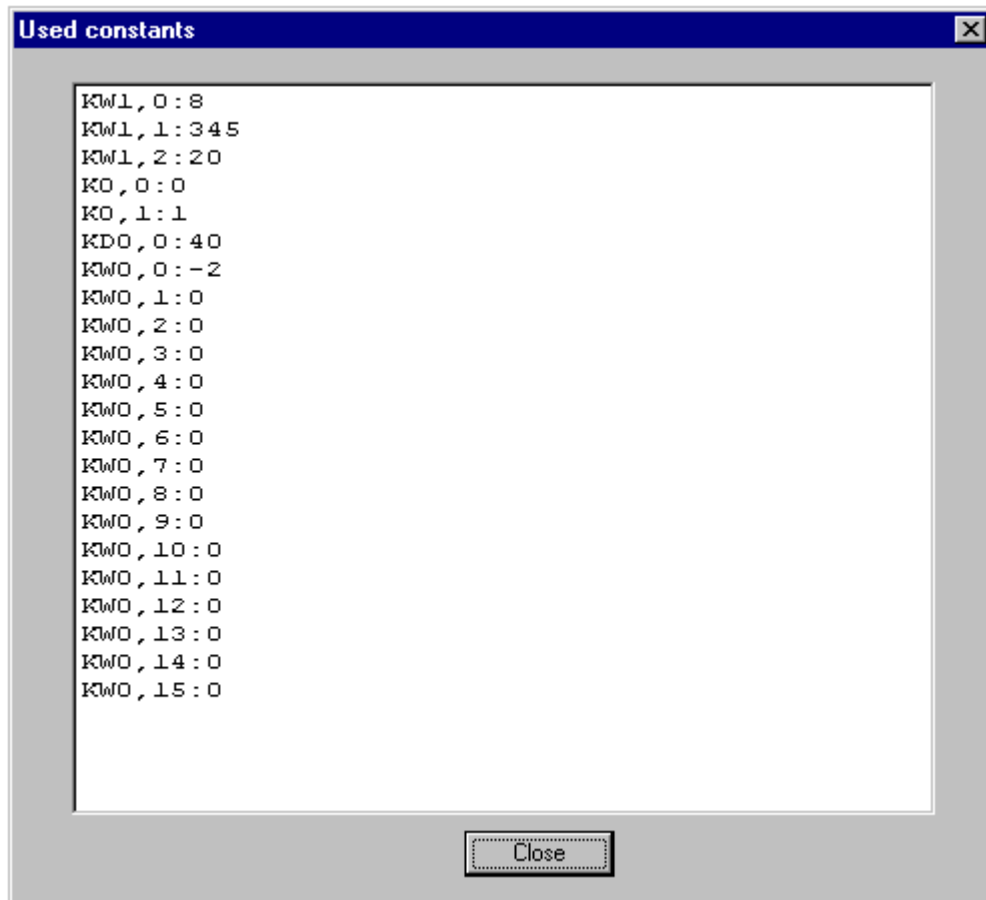


**Note:**

In the present version the ranges of the "Free addressable WORD constants up to" and the "Free addressable DWORD constants up to" are not reserved. For converting the direct constants non-declared word and double word constants are used.

### *Show used constants:*

When selecting this menu item after an error-free translation of the program all used constants (including the system constants) are displayed together with their initialization values. The display is as following:



If the translation of the program is faulty the output is not updated.

### ***Upload and Import PC331 projects / Import upload file and conversion back to IEC 1131-3:***

Using this function a program previously uploaded from the controller using the menu item "Upload user program to file" can be re-imported into 907 AC 1131.

After selecting the menu item the following inquiry appears:



Import is only possible into the IEC language IL.



**Note:**

This functionality is not yet realized in the present version.

### ***Upload and Import PC331 projects / Import variables from ABB907PC33 projects into global variables list:***

### ***Upload and Import PC331 projects / Import program from ABB907PC33 project:***



**Note:**

This functionality is not yet realized in the present version.

## 1.5.2 In online mode available commands of the Online menu for S40..50

The following commands are **available** in the logged-in state (online mode):

- Logout
- Download
- Run
- Stop
- Reset
- Write Values
- Force Values
- Release Force
- Create boot project (flash user program)
- Settings for Force (refer to the previous section "Commands in offline mode")
- Cold start
- Enter password
- Read/write real time clock
- Show CPU status (CPU load etc.)
- Erase flash
- Send constants
- Upload user program to file

The following functions are **not available** for S40..50 and therefore greyed out:

- Breakpoints
- Step
- Single Cycle
- Display Flow Control
- Show Call Stack

### *Logout:*

Using the command "Logout" the connection to the PLC is disconnected.

If the parameter "Do not release force when logging out" is not selected under "Settings for Force", forcing is released. If this parameter is selected and values are forced, an inquiry appears whether forcing should be released.

### *Download:*

Use the command "Download" if you want to transfer the project to the controller. The following messages can appear:

1. "The program was changed. Do you want to load only the changes (Yes) or perform a full download (No)?  
Select "**No**" if you want to send the **complete program (download)**.  
Select "**Yes**" for "**Online change**".



#### **Note:**

If you have a **PLC of the category R2002** (firmware version 22) no "Online change" can be performed. Therefore always select "**No**". In addition, for these controllers the program must be stopped prior to loading the program ("Online" => "Stop") because flashing is not possible while in Run mode.

2. "Program checksum has not changed. Do you want to perform a download regardless of this fact?  
Select "Yes" if you want to send.  
Select "No" to display only the current values.

When downloading the program, the program and the constants are transferred to the controller. Subsequently the program and the constants are stored in the Flash memory. During the flash process the indication "FLASH" in the status bar is displayed in black (otherwise the indication is displayed in grey).

If the controller is in the Run mode and flashing is not possible the message "Änderung/Flashen im RUN von angeschlossener CPU nicht unterstützt! Gehen Sie in STOP, um Programm zu laden." appears.

### ***Start:***

Using this command the program in the PLC is started.

If the RUN/STOP switch is set to position "1" the message "RUN nicht möglich" is displayed. Set the switch to the "ON" position.

### ***Stop:***

Using this command the program in the controller is stopped.

### ***Reset:***

After selecting this command the following inquiry appears: "Do you really want to reset your PLC?" Selecting "Yes" initiates a "**warm start**" of the controller.

After a successful completion of the warm start the controller signalizes for example the following:

„ABB Advant AC31 : OPERATION PROGRAM V2.2 03.10.2000 / SERIE 50“

### ***Write Values:***

Using the option "Write Values" you can overwrite all valid variables of the S40..50. Overwriting means here that the variables are overwritten by a predetermined value once. They can be changed at any time in the program.

Proceed as follows to overwrite variables:

1. Login to the controller.
2. Double click on the variable to be overwritten.  
For binary variables first the value TRUE and after another double click FALSE is displayed next to the variable. Select the desired value by double clicking.  
For INT and DINT variables a window appears displaying the old value and an input field for entering the new value. Enter the desired value and confirm with "OK". After this the variable is prepared for overwriting.
3. Select "Online" => "Write Values" (or key combination <CTRL> + <F7>). The selected variable(s) are transferred to the PLC and overwritten one time.

### ***Force Values:***

In the S40..50 only inputs and outputs can be forced. Forcing of inputs means that the force value is assigned to the inputs at the beginning of the program regardless of the actual value of the input. Forcing of outputs means that the force values are assigned to the outputs at the end of the program regardless of the value which was assigned to the outputs during the program execution. Forcing is continued until it is released again.

Proceed as follows to force variables:

1. Login to the controller.
2. Double click on the variable to be forced.  
For binary variables first the value TRUE and after another double click FALSE is displayed next to the variable. Select the desired value by double clicking.  
For INT and DINT variables a window appears displaying the old value and an input field for entering the new value. Enter the desired value and confirm with "OK". After this the variable is prepared for forcing.
3. Select "Online" => "Force Values" (or press the <F7> key). The selected variable(s) are transferred to the PLC and forced.

### ***Release Force:***

Forcing is released by one of the following:

- "Online" => "Release Force" (forcing is released for all variables).
- Key combination <Shift> + <F7>.
- Double clicking on the variable and selecting "Release Force".
- Logging out (see Settings for Force).
- Switching the supply voltage OFF/ON or performing a warm start will release forcing but the force values are kept (see Settings for Force).

### ***Create boot project (flash):***

Using this command the user program is written to the Flash memory.

When downloading the program using "Online" => "Download" the program is automatically flashed. So you do not need to perform this command additionally.

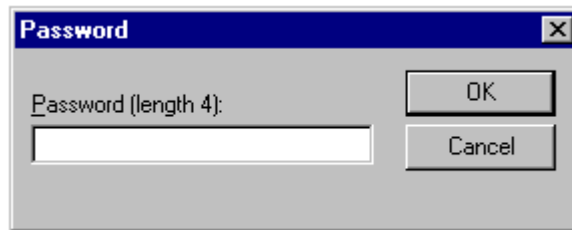
### ***Cold start:***

Using this command a "cold start" is initiated on the PLC.

After a successful completion of the cold start the controller signalizes for example the following:  
"ABB Advant AC31 : OPERATION PROGRAM V2.2 03.10.2000 / SERIES 50"

### ***Settings / Enter password:***

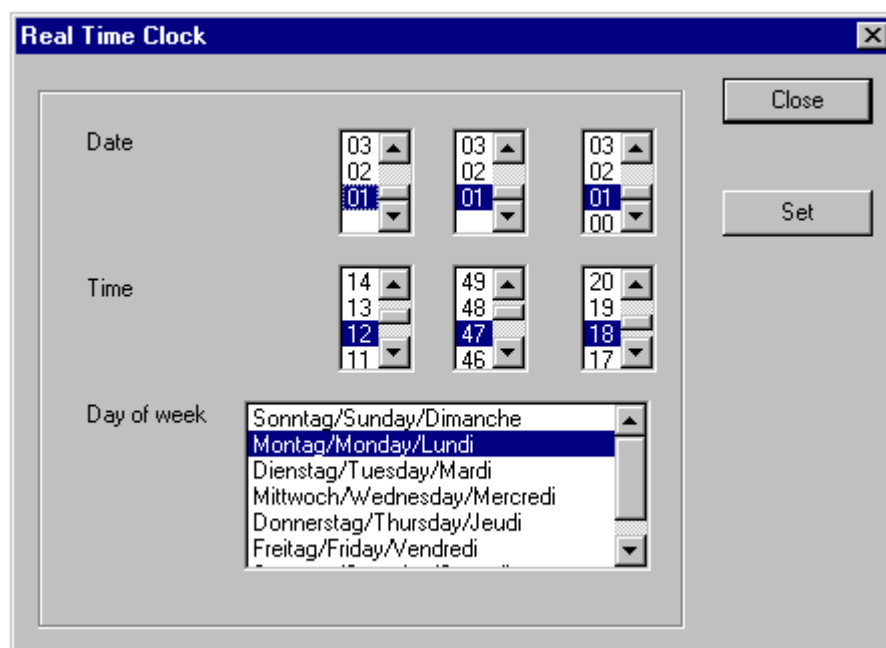
After selecting this menu item the following dialog appears:



Enter the desired password. It must have a length of 4 characters, hexadecimal (0 .. F). After clicking on "OK" it is transferred to the PLC.

### ***Settings / Read/write real time clock:***

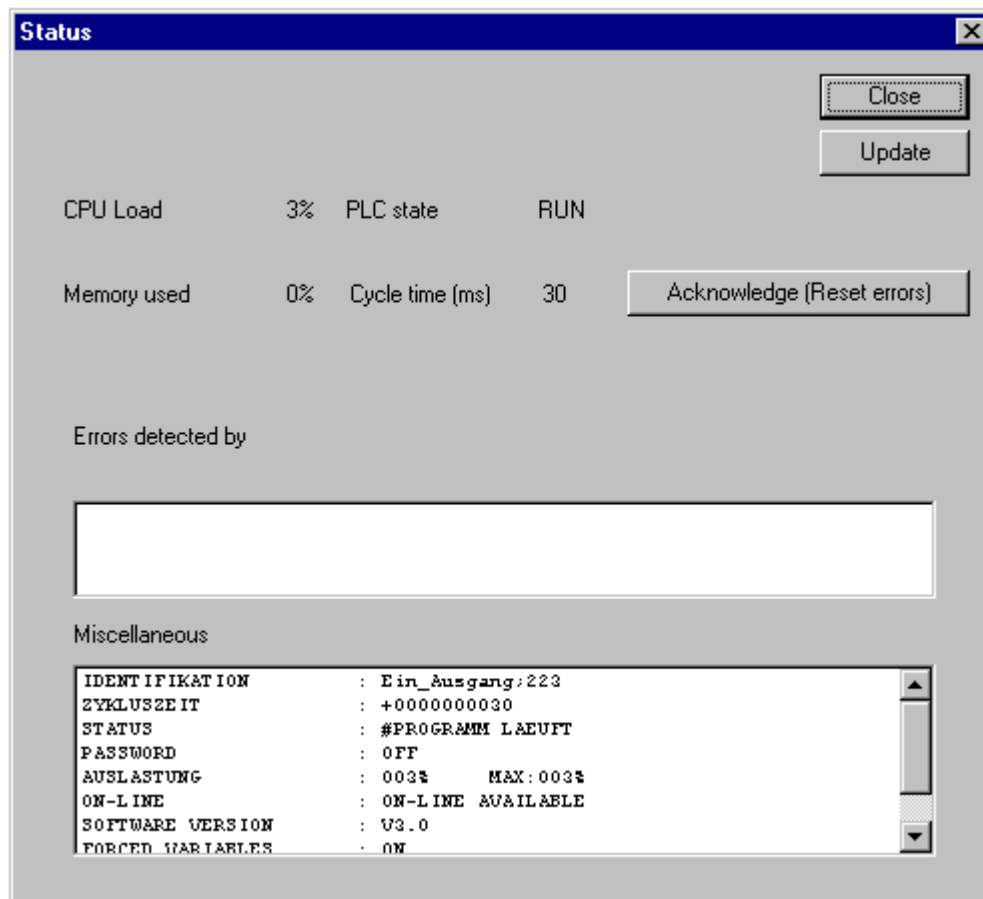
After selecting this menu item the following dialog appears:



The current values are displayed. The date is displayed in the order day, month, year, the time is displayed in the order hour, minute, second and the weekday is displayed in clear text. In order to set the clock select the values for date, time and weekday and then click on "Set". After setting/applying the values are read once again and displayed. The dialog is closed by clicking on "Close".

### Show CPU status (CPU load etc.):

Use this menu option to obtain information about the current controller status.



The following information is displayed:

CPU utilization (CPU load) in %

PLC mode (RUN or STOP)

Memory utilization in %

Cycle time in ms

Error list - Indication of errors in clear text

Miscellaneous - Response of the PLC to the terminal command "ST"

### Erase flash:

Using this command you can erase the user flash memory.

### Send constants:

Using this command only the constants are sent to the controller and subsequently loaded into the flash memory.



#### Note:

If you have a PLC of the category R2002 flashing in Run mode is not possible. Consequently you must stop the program before transferring.

### ***Upload and Import PC331 projects / Upload user program to file:***

Using this command the ASCII IL of a program is uploaded back from the controller and written to a file.

Using the command "Import upload file and conversion back to IEC 1131-3" this file can be imported to 907 AC 1131 in offline mode.



**Note:**

This functionality is not yet realized in the present version.

## 1.6 Peculiarities in the Resources tab

For series 40..50 the following objects are available in the Resources page of the Object Organizer:

- **Libraries**  
Display of the libraries used in the project.
- **Global variables**  
**For S40..50 all variables are global.**
- **Controller configuration**  
Here, the selection of series 40..50 or series 90 is performed by selecting the PLC.
- **Task configuration**  
Not of importance for S40..50.
- **Watch and recipe manager**  
For displaying and pre-assignment of variable values.

### 1.6.1 Object libraries

In the object "Libraries" all libraries of the project are displayed. For series 40..50 this is always **Base\_S40\_Vxx.LIB** (where Vxx represents the version, e.g. V41).

### 1.6.2 Global variables

Under Resources / Global variables the following three objects are created when creating a new project:

- **Globale Variablen**  
Here, VAR\_GLOBAL and VAR\_GLOBAL CONSTANT can be declared from the valid variables of S40..50.

VAR\_GLOBAL RETAIN cannot be used for S40..50. For buffering of variables corresponding ranges with system constants must be determined:

KW00\_01 AT %MW3000.1 for buffering of flags %MX000..

KW00\_02 AT %MW3000.2 for buffering of flag words %MW1000..

KW00\_03 AT %MW3000.3 for buffering of step chains %MX5000..

KW00\_04 AT %MW3000.4 for buffering of double words %MD2000..

- **S40\_Fehlermerker**  
In this object the system and error flags are pre-declared.  
M255\_00 AT %MX255.0 .. M255\_15 AT %MX255.15  
MW254\_00 AT %MW1254.0 .. MW255\_15 AT %MW1255.15
- **S40\_Systemkonstanten**  
In this object the system constants of S40..50 are pre-defined  
KD00\_00 AT %MD4000.0 : DINT := 30; (\* cycle time \*)  
KW00\_00 AT %MW3000.0 .. KW00\_15 AT %MW3000.15

As for series 90 you can also create further objects for series 40..50.

In addition it is possible to insert already pre-assigned import files. For this, select "Project" => "Import" and then select from the pre-assigned \*.exp files:

– S40_ONBOARD.EXP	Onboard inputs and outputs of the S40..50 PLCs
– M00_00_to_M00_15.EXP	M00,00 .. M00,15
– M01_00_to_M01_15.EXP	M01,00 .. M01,15
– M02_00_to_M02_15.EXP	M02,00 .. M02,15
– M03_00_to_M03_15.EXP	M03,00 .. M03,15
– M04_00_to_M04_15.EXP	M04,00 .. M04,15
– M05_00_to_M05_15.EXP	M05,00 .. M05,15
– M06_00_to_M06_15.EXP	M06,00 .. M06,15
– M07_00_to_M07_15.EXP	M07,00 .. M07,15
– M08_00_to_M08_15.EXP	M08,00 .. M08,15
– M09_00_to_M09_15.EXP	M09,00 .. M09,15
– M10_00_to_M10_15.EXP	M10,00 .. M10,15
– MW00_00_to_MW00_15.EXP	MW00,00 .. MW00,15
– MW01_00_to_MW01_15.EXP	MW01,00 .. MW01,15
– MW02_00_to_MW02_15.EXP	MW02,00 .. MW02,15
– MD00_00_to_MD00_15.EXP	MD00_00_to_MD00_15.EXP
– MD01_00_to_MD01_15.EXP	MD01_00_to_MD01_15.EXP
– MD02_00_to_MD02_15.EXP	MD02_00_to_MD02_15.EXP
– SK_00.EXP	S00,00 .. S00,15
– SK_01.EXP	S01,00 .. S01,15
– SK_02.EXP	S02,00 .. S02,15
– S40_SYCONST.EXP	automatically inserted when selecting "File" / "New"
– S40_FEHLERMERKER.EXP	automatically inserted when selecting "File" / "New"

### 1.6.3 Controller configuration

Using the controller configuration the selection of the PLC is performed.

The selected PLC is entered under "File" => "New" into the controller configuration.

Select from the following controllers for S40..50:

Series 40 → 07CR41 or 07CT41

Series 50 → 07KR51 or 07KT51

**If you enter a PLC of series 90 here, your project is treated as if it was created for series 90!!!** In this case the error message "Error: Cannot open a \*.obj" appears during the translation process because BASE\_S40\_Vxx.LIB does not contain an object but IEC\_S90\_Vxx.LIB does contain an object.

### 1.6.4 Task configuration

For series 40..50 the task configuration is not of importance.

The cycle time is set using the system constant KD00,00 / %MD4000.0.

The initialization value of this constant represents the cycle time in [ms].

**Example:** KD00\_00 AT %MD4000.0 : INT := 30; (\* cycle time 30 ms \*)

The system constant is predefined in the object **S40\_Systemkonstanten**.

## 1.6.5 Watch and recipe manager

Only variables which are valid for S40..50 can be read and written.

## 2 Data types and variables for S40..50

### 2.1 Data types in S40..50

In 907 AC 1131 the following data types can be used. Not all of them are available for S40..50.

Data type	Available in S40..50	Restrictions
BOOL	yes	no
Integer data types	yes	yes, see below
REAL / LREAL	no	
STRING	no	
Time data types	yes	yes, see below
Array	no	
Pointer	no	
Enumeration type	no	
Structures	no	
References	no	

#### *BOOL*

Variables of the type **BOOL** can have the values TRUE and FALSE. The code generated and sent for series 40..50 corresponds to K0,0 and K0,1.

#### *Integer data types*

Integer data types are: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT and UDINT. As shown in the table only the types **INT** and **DINT** are available for series 40..50. The following limits apply to the ranges:

Type	Lower limit	Upper limit	Memory space	Available in S40..50
BYTE	0	255	8 bits	no
WORD	0	65535	16 bits	no
DWORD	0	4294967295	32 bits	no
SINT	-128	127	8 bits	no
USINT	0	255	8 bits	no
INT	-32768	32767	16 bits	yes
UINT	0	65535	16 bits	no
DINT	-2147483648	-2147483647	32 bits	yes
UDINT	0	4294967295	32 bits	no

For the time functions an automatic conversion is performed between DINT and TIME.

#### *REAL and LREAL*

REAL and LREAL are floating-point data types. These data types are **not available for S40..50**.

## ***STRING***

Variables of the type **STRING** can include any character string.  
This data type is **not available for S40..50**.

## ***Time data types***

For S40..50 this data type is available in the time functions. For the time functions the data type **TIME** is automatically converted into **DINT**.

**TIME\_OF\_DAY (TOD)** and **DATA\_AND\_TIME (DT)** are **not available for the S40..50**.

## ***Array***

This data type is **not available for S40..50**.

## ***Pointer***

This data type is **not available for S40..50**.

## ***Enumeration type***

An enumeration type is a self-defined data type consisting of a collection of string constants. These constants are called enumeration values.  
This data type is **not available for S40..50**.

## ***Structures***

This data type is **not available for S40..50**.

## ***References***

The self-defined data type **Reference** serves for the creation of an alternative name for a variable, constant, function block, type, etc.  
This data type is **not available for S40..50**.

## 2.2 Variables of S40..50

The table below lists **all variables** of series 40..50:

Variable in PLC	Type	Address in 907 PC 1131
E00,00 .. E79,15	Inputs BOOL	%IX000.00 .. %IX079.15
A00,00 .. A79,15	Outputs BOOL	%QX000.00 .. %QX079.15
M000,00 .. M099,15 M230,00 .. M255,15	Flags BOOL	%MX000.00 .. %MX099.15 %MX230.00 .. %MX255.15
S000,00 .. S125,15	Steps BOOL	%MX5000.00 .. %MX5125.15
K0,0 .. K0,1	Constants BOOL	FALSE and TRUE
EW00,00 .. EW68,15	Analog inputs INT	%IW1000.00 .. %IW1068.15
AW00,00 .. AW68,15	Analog outputs INT	%QW1000.00 .. %QW1068.15
MW000,00 .. MW099,15 MW230,00 .. MW255,15	Flag words INT	%MW1000.00 .. %MW1099.15 %MW1230.00 .. %MW1255.15
KW00,00 .. KW31,15	Word constants INT	%MW3000.00 .. %MW3031.15
MD00,00 .. MD07,15	Double word flags DINT / TIME	%MD2000.00 .. %MD2007.15
KD00,00 .. KD07,15	Double word constants DINT / TIME	%MD4000.00 .. %MD4007.15

### *Examples for the variables declaration:*

1. Input E62,00 shall be designated the symbolic name Switch1:  
Switch1 AT %IX62.0 : BOOL; (\* Switch1 assigned to input E62,00 \*)
2. Pump\_ON shall be allocated to output A62,03:  
Pump\_ON AT %QX62.3 : BOOL; (\* pump ON if TRUE \*)
3. The actual value of a counter shall be allocated to MW10,10:  
Actual\_value\_counter AT %MW1010.10 : INT; (\* actual value of the counter to MW10\_10 \*)
4. The set point shall be allocated to constant KW05,00 and shall be 500:  
Set\_point AT %MW3005.0 : INT := 500; (\* set point to KW05\_00 \*)

### ***Restrictions for variables:***

For series 40..50 the following restrictions apply for the variables:

1. All variables are in the **addressable range** %M, %I, %Q (see table above).
2. There are **no internal variables**. The instances which are to be created for the function blocks of the block library BASE\_S40\_Vxx.LIB are just dummies for the translation. The blocks must always be seized with real variables. Therefore it is not possible to work with instance variables.
3. It is **not possible** to access flags **byte or word wise** (as for series 90). Words cannot be accessed bit or byte wise. Double words cannot be accessed bit wise, byte wise or word wise.
4. An **initial value** can only be allocated to the word and double word constants.
5. All variables are **global**.
6. For the translation process variable ranges are reserved for usage as intermediate flags. The beginning of these ranges can be set. (Menu: Online / Internal operands). The meaning of the individual ranges is described later.

## 3 Operators and blocks for S40..50

### 3.1 Operators for S40..50 in 907 AC 1131



**Note:**

The operators are directly integrated in the 907 AC 1131 software.

#### 3.1.1 Overview: Operators for S40..50

Operator	Name in 907 AC 1131	Available in S40..50	Notes
<b>Arithmetical operators</b>			
Addition of variables	ADD	yes	Type INT
Multiplication of variables	MUL	yes	Type INT
Subtraction of variables	SUB	yes	Type INT
Division of variables	DIV	yes	Type INT
Modulo division of a variable	MOD	no	
Block index	INDEXOF	no	
Number of bytes of one data type	SIZEOF	no	
<b>Bitstring operators</b>			
Bit wise AND	AND	yes	Type BOOL
Bit wise OR	OR	yes	Type BOOL
Bit wise exclusive OR	XOR	yes	Type BOOL
Bit wise NOT	NOT	yes	Type BOOL
<b>Bit shift operators</b>			
Shift to left, bit wise	SHL	no	
Shift to right, bit wise	SHR	no	
Rotation to left, bit wise	ROL	no	
Rotation to right, bit wise	ROR	no	
<b>Selection operators</b>			
Binary selection	SEL	yes	Types INT, BOOL In 907 PC 331: AWT
Maximum function	MAX	yes	Type INT
Minimum function	MIN	yes	Type INT
Limitation	LIMIT	no	
Multiplexer	MUX	no	

Operator	Name in 907 AC 1131	Available in S40..50	Notes
<b>Comparison operators</b>			
Greater than	GT	yes	Types INT, DINT
Lower than	LT	yes	Types INT, DINT
Lower than or equal to	LE	yes	Type INT
Greater than or equal to	GE	yes	Type INT
Equal	EQ	yes	Types INT, DINT
Not equal	NE	yes	Types INT, DINT
<b>Address operators</b>			
Address function	ADR	no	
Contents operator	^	no	
<b>Calling operator</b>			
Calling a function block	CAL	yes	
<b>Numerical operators</b>			
Absolute value of a number	ABS	yes	Type INT
Square root of a number	SQRT	yes	Types INT, DINT
Natural logarithm	LN	no	
Common logarithm	LOG	no	
Exponential function	EXP	no	
Sine of a number	SIN	no	
Cosine of a number	COS	no	
Tangent of a number	TAN	no	
Arcus sine of a number	ASIN	no	
Arcus cosine of a number	ACOS	no	
Arcus tangent of a number	ATAN	no	
Raising two numbers to a higher power	EXPT	no	
<b>Type conversions</b>			
Conversions BOOL_TO	BOOL_TO_xxx	no	
Conversions TO_BOOL	xxx_TO_BOOL	no	
Conversion between integer numerical types	xxx_TO_yyy	no	
Conversions REAL_TO	REAL_TO_xxx	no	
Conversions LREAL_TO	LREAL_TO_xxx	no	
Conversions TIME_TO	TIME_TO_xxx	no	
Conversions TIME_OF_DAY	TOD_TO_xxx	no	
Conversions DATE_TO	DATE_TO_xxx	no	
Conversions DATE_AND_TIME	DT_TO_xxx	no	
Conversions STRING_TO	STRING_TO_xxx	no	
Conversion REAL → INT, integer part	TRUNC	no	

### 3.1.2 Blocks of the standard IEC library for S40..50



**Note:**

Blocks of the standard IEC library which are available for S40..50 are directly integrated in 907 AC 1131 or in the library Base\_S40\_V41.LIB / Base\_S40\_V41.BIB. There is no existing library IEC\_S40\_xx.LIB!

Block	Name in 907 AC 1131	Available in S40..50	Notes
<b>String functions</b>			
Length of a string	LEN	no	
Left initial string of a string	LEFT	no	
Right initial string of a string	RIGHT	no	
String part of a string	MID	no	
Linking two strings together	CONCAT	no	
Inserting a string, beginning from a particular position	INSERT	no	
Deleting a part of a string	DELETE	no	
Replacing a part of a string	REPLACE	no	
Finding a part of a string	FIND	no	
<b>Bistable function blocks</b>			
Set, dominating	SR	yes	Type BOOL In 907 PC 331: RS
Reset, dominating	RS	yes	Type BOOL In 907 PC 331: SR
Software semaphore	SEMA	no	
<b>Triggers</b>			
Rising edge	F_TRIG	yes	Type BOOL, name: I_PLUS
Falling edge	F_TRIG	yes	Type BOOL, name: I_MINUS
<b>Counters</b>			
Up-counter	CTU	yes	Type INT
Down-counter	CTD	no	
Up- and down-counter	CTUD	no	
<b>Timers</b>			
Pulse generator	TP	yes	
On delay	TON	yes	
Off delay	TOF	yes	

## 3.2 Block library for S40..50

Name: Base\_S40\_V41.LIB and  
Base\_S40\_V41.BIB

Group/Block	Description	Type	Possible usage			
			Series 40	Series 50	Version	
<b>Binary operations (bit operations)</b>						
AWTB	Selection gate Bit	FB	X	X	2.2	
BMELD127	Binary value change annunciator	FB	X	X	2.2	
BMELD16						127 values max.
BMELD32						16 values max.
BMELD64						32 values max.
BMELD8						64 values max.
BMELD8	8 values max.					
IDLB	Read indexed, Bit	FB	X	X	2.2	
IDSB	Write indexed, Bit	FB	X	X	2.2	
<b>Bistable function</b>						
RS	Reset, dominating	FB	X	X	2.2	
SR	Set, dominating	FB	X	X	2.2	
<b>Communication</b>						
MODBUSB	MODBUS master COM1, BOOL data	FB	X	X	2.2	
MODBUSW	MODBUS master COM1, INT data	FB	X	X	2.2	
MODMASTB	MODBUS master COM1/2, BOOL data	FB	X	X	2.2	
MODMASTW	MODBUS master COM1/2, INT data	FB	X	X	2.2	
SINIT	Initialization of the serial interface	FB	X	X	2.2	
<b>Conversions</b>						
BCDDUAL	BCD - DUAL conversion, Word	F	X	X	2.2	
DUALBCD	DUAL - BCD conversion, Word	F	X	X	2.2	
DWW	Double word - Word conversion	FB	X	X	2.2	
TIME_W	Time → Words	FB	X	X	2.2	
W_TIME	Words → Time	FB	X	X	2.2	
WDW	Word - Double word conversion	F	X	X	2.2	
<b>Counters</b>						
CTU	Up-counter	FB	X	X	2.2	
CTUH	Fast counter	FB	X	X	2.2	
VRZ	Up-down-counter	FB	X	X	2.2	
<b>CS31 bus</b>						
CS31CO	Configure AC31 modules	FB	X	X	2.2	
CS31QU	Acknowledge AC31 errors	FB	X	X	2.2	
<b>Double word operations</b>						
ADDD	Addition double word	FB	X	X	2.2	
DIVD	Division double word	FB	X	X	2.2	
DWAND	Double word AND	F	X	X	2.2	
DWOR	Double word OR	F	X	X	2.2	
DWXOR	Double word XOR	F	X	X	2.2	
MULD	Multiplication double word	FB	X	X	2.2	
SQRTD	Square root double word	FB	X	X	2.2	
SUBD	Subtraction double word	FB	X	X	2.2	
<b>Loop control</b>						
PI	Proportional integral controller	FB	X	X	2.2	
PIDT1	PIDT1 controller	FB	X	X	2.2	

Group/Block	Description	Type	Possible usage			
			Series 40	Series 50	Version	
<b>Other blocks</b>						
CONFIO1	Configuration of analog channels	1 channel max.	FB	X	X	2.2
CONFIO4		4 channels max.				
CONFIO8		8 channels max.				
DIN	Direct reading of an input	FB	X	X	2.2	
DOUT	Direct writing of an output	FB	X	X	2.2	
UHR	Display / set real-time clock	FB	X	X	2.2	
<b>Task handling</b>						
VTASK		FB	X	X	2.2	
<b>Timers</b>						
ASV	OFF Delay	FB	X	X	2.2	
ESV	ON Delay	FB	X	X	2.2	
MOA	Monostable element "abort"	FB	X	X	2.2	
MOAT	Monostable element "abort" + actual value					
MOK	Monostable element "constant"	FB	X	X	2.2	
NPULSE	Pulse generator	FB	X	X	2.2	
PDM	Pulse-duration modulator	FB	X	X	2.2	
TOF	OFF delay with actual value	FB	X	X	2.2	
TON	ON delay with actual value	FB	X	X	2.2	
TP	Pulse generator with actual value	FB	X	X	2.2	
<b>Triggers</b>						
I_MINUS	Falling edge	FB	X	X	2.2	
I_PLUS	Rising edge	FB	X	X	2.2	
<b>Word operations</b>						
AWT	Selection gate Word	FB	X	X	2.2	
BEG	Limiter word	F	X	X	2.2	
COPY	Copying memory areas	F	X	X	2.2	
FKG16	Function generator	16 sample points max.	FB	X	X	2.2
FKG2		2 sample points max.				
FKG256		256 sample points max.				
FKG32		32 sample points max.				
FKG4		4 sample points max.				
FKG64		64 sample points max.				
IDLm	Read word variable, indexed	FB	X	X	2.2	
IDSm	Write word variable, indexed	FB	X	X	2.2	
LIZU16	List allocator	16 values max.	FB	X	X	2.2
LIZU256		256 values max.				
LIZU32		32 values max.				
LIZU64		64 values max.				
LIZU8		8 values max.				
MUL2N	Multiplication by 2 to the power of n	F	X	X	2.2	
MULDI	Multiplication with division	F	X	X	2.2	
NEGW	Negation word	F	X	X	2.2	
PACK4	Pack bits in one word	4 bits max.	FB	X	X	2.2
PACK8		8 bits max.				
PACK16		16 bits max.				

Group/Block	Description		Type	Possible usage		
				Series 40	Series 50	Version
SQRTW	Square root word		FB	X	X	2.2
UNPACK4	Unpacking a word into bits	4 bits max.	FB	X	X	2.2
UNPACK8		8 bits max.				
UNPACK16		16 bits max.				
WOL	Read word with enabling		F	X	X	2.2
WAND	Word AND		F	X	X	2.2
WOR	Word OR		F	X	X	2.2
WXOR	Word XOR		F	X	X	2.2

### 3.3 Conversion of 907 PC 331 blocks to 907 AC 1131 S40..50



**Note:**

If in the controller the function call in IL differs from the CE name in 907 PC 331, this call is additionally noted in the IL column.

1) This function is realized in IL using a sequence of commands and/or blocks.

Designation	Name in 907 PC 331		907 AC 1131	
	FBD	IL	Designation	Realized by
<b>Binary functions</b>				
AND	&	1)	AND	Operator
OR	/	1)	OR	Operator
Exclusive OR	=1	1)	XOR	Operator
Allocation	=	1)	ST	Operator
Negation	=N	1)	NOT	Operator
<b>Memory functions</b>				
Allocation Set Memory	=S	1)	S	Operator
Allocation Reset Memory	=R	1)	R	Operator
Set Memory, dominating	RS	1)	SR	BASE_S40_V41
Reset Memory, dominating	SR	1)	RS	BASE_S40_V41
<b>Arithmetical functions, Word</b>				
Addition / Word	+	1)	ADD	Operator
Subtraction / Word	-	1)	SUB	Operator
Multiplication / Word	*	1)	MUL	Operator
Division / Word	:	1)	DIV	Operator
Multiplication with Division / Word	*:	MULDI	MULDI	BASE_S40_V41
Multiplication by 2 to the power of N / Word	MUL2N		MUL2N	BASE_S40_V41
Absolute value generator	BETR		ABS	Operator
Allocation Word	=W	1)	ST	Operator
Allocation of direct constant to word variable	ZUDKW		-	907 AC 1131 internal
Square root / Word	SQRT	SQRT	SQRTW	Base_S40_V41
<b>Arithmetical functions, Double word</b>				
Addition / Double word	+D	ADDD	ADD	Base_S40_V41
Subtraction / Double word	-D	SUBD	SUB	Base_S40_V41
Multiplication / Double word	*D	MULD	MUL	Base_S40_V41
Division / Double word	:D	DIVD	DIV	Base_S40_V41
Square root / Double word	SQRT	SQRT	SQRTD	Base_S40_V41
<b>Comparison functions</b>				
Greater than	>	1)	GT	Operator
Greater than or equal to	>=	1)	GE	Operator
Equal	=?	1)	EQ	Operator
Not equal	<>	1)	NE	Operator
Less than	<	1)	LT	Operator
Less than or equal to	<=	1)	LE	Operator

Designation	Name in 907 PC 331		907 AC 1131	
	FBD	IL	Designation	Realized by
<b>Comparison functions, Double word</b>				
Equal	=?D	VGLD	EQ	Operator
Greater than	>D	VGRD	GT	Operator
Less than	<D	VKLD	LT	Operator
<b>Time functions</b>				
ON Delay	ESV		ESV	BASE_S40_V41
OFF Delay	ASV		ASV	BASE_S40_V41
Monostable element "abort"	MOA		MOA	BASE_S40_V41
Monostable element "constant"	MOK		MOK	BASE_S40_V41
<b>Time functions with actual value indication</b>				
OFF Delay	TOF		TOF	BASE_S40_V41
ON Delay	TON		TON	BASE_S40_V41
Monostable element "constant"	TP		TP	BASE_S40_V41
Monostable element "abort"	MOAT		MOAT	BASE_S40_V41
<b>Counter functions</b>				
Up-down counter / Word	VRZ		VRZ	BASE_S40_V41
Up-counter	CTU		CTU	BASE_S40_V41
Counter incremental transmitter input	CTUH		CTUH	BASE_S40_V41
<b>Program control functions</b>				
Conditional jump to label	SPBM	SPRUNG	SPRUNG	907 AC 1131 internal
Target label	MRK	MR	LABEL	907 AC 1131 internal
Program end	PE		-	Automatically generated at program end
Conditional program end	=PE	1)	-	Can be realized with jump
Subroutine call for assembler program	CALLUP		-	Internally used for MODBUS functions
Subroutine call	CAL_FB		-	Generated for subroutine call in 907 AC 1131
Subroutine definition	DEF_FB		-	Generated at the beginning of a subroutine in 907 AC 1131
Return	RET		-	Generated at the end of a subroutine in 907 AC 1131
Enable interrupt program	VTASK		VTASK	BASE_S40_V41
Interrupt routine definition	TASK		-	Generated at the beginning of an interrupt routine in 907 AC 1131
Direct reading of an input	DIN	DI	DIN	BASE_S40_V41
Direct writing of an output	DOUT	DO	DOUT	BASE_S40_V41

Designation	Name in 907 PC 331		907 AC 1131	
	FBD	IL	Designation	Realized by
<b>Format conversion</b>				
BCD -> DUAL conversion / Word	BCDDUAL	BCDBIN	BCDDUAL	BASE_S40_V41
Dual -> BCD conversion / Word	DUALBCD	DUALBIN	DUALBCD	BASE_S40_V41
Pack Bits -> Word	PACK		PACK4(8, 16)	BASE_S40_V41
Unpacking Word -> Bits	UNPACK		UNPACK4 (8,16)	BASE_S40_V41
Word to double word conversion	WDW		WDW	BASE_S40_V41
Double word to word conversion	DWW		DWW	BASE_S40_V41
<b>Pulse</b>				
Pulse (positive edge)	I+	1)	I_PLUS	BASE_S40_V41
Pulse (negative edge)	I-	1)	I_MINUS	BASE_S40_V41
<b>Logical connection with word size</b>				
AND connection, Word	WAND		WAND	BASE_S40_V41
OR connection, Word	WOR		WOR	BASE_S40_V41
Exclusive OR connection, Word	WXOR		WXOR	BASE_S40_V41
Negation, Word	NEG	==	NEGW	BASE_S40_V41
<b>Logical connection with double word size</b>				
AND connection, Double word	DWAND		DWAND	BASE_S40_V41
OR connection, Double word	DWOR		DWOR	BASE_S40_V41
Exclusive OR connection, Double word	DWXOR		DWXOR	BASE_S40_V41
<b>Accessing physical addresses</b>				
Copying memory areas	COPY		COPY	BASE_S40_V41
Read word with enabling	WOL		WOL	BASE_S40_V41
<b>Functions of a higher order</b>				
Selection gate Bit	AWTB		AWTB / SEL	BASE_S40_V41 / Operator
Selection gate Word	AWT		AWT / SEL	BASE_S40_V41 / Operator
Maximum value generator, Word	MAX		MAX	Operator
Minimum value generator, Word	MIN		MIN	Operator
Limiter, Word	BEG		BEG	BASE_S40_V41
Binary value change annunciator	BMELD		BMELD8 (16,32,64,127)	BASE_S40_V41
Configuration of analog channels	CONFIO		CONFIO1 (4, 8)	BASE_S40_V41
Function generator	FKG		FKG2 (4, 16, 32, 64, 256)	BASE_S40_V41
List allocator	LIZU		LIZU	BASE_S40_V41
Read word variables, indexed	IDLm	IDL	IDLm	BASE_S40_V41
Write word variable, indexed	IDSm	IDS	IDSm	BASE_S40_V41
Read binary variable, indexed	IDLB		IDLB	BASE_S40_V41
Write binary variable, indexed	IDSB		IDSB	BASE_S40_V41
Display / set clock	UHR		UHR	BASE_S40_V41
Pulse generator for driving stepper motors	NPULSE		NPULSE	BASE_S40_V41

Designation	Name in 907 PC 331		907 AC 1131	
	FBD	IL	Designation	Realized by
<b>Control engineering functions</b>				
Proportional integral controller	PI		PI	BASE_S40_V41
PIDT1 controller	PIDT1		PIDT1	BASE_S40_V41
Pulse duration modulator	PDM		PDM	BASE_S40_V41
<b>Communication via serial interface</b>				
Initializing and configuring a serial interface	SINIT		SINIT	BASE_S40_V41
Output of ASCII characters and hex values via the serial interface	DRUCK		Inline Pragma	S40 Inline Pragma
Reception of characters	EMAS		Inline Pragma	S40 Inline Pragma
MODBUS master	MODBUS	CALLUP	MODBUSB, MODBUSW, MODMASTB, MODMASTW	BASE_S40_V41
<b>CS31 functions</b>				
Configure AC31 modules	CS31CO		CS31CO	BASE_S40_V41
Acknowledge AC31 errors	CS31QU		CS31QU	BASE_S40_V41

---

**B**

Base\_S90\_V41.LIB 43  
block library S40..50 43

---

**C**

code size S40..50 19  
cold start S40..50 30  
commands Online menu S40..50 20  
    available in offline mode 21  
    available in online mode 28  
communication parameters S40..50 10, 23  
constants (displaying) S40..50 26  
controller configuration S40..50 35  
conversion back to IEC 1131-3 S40..50 27  
conversion of blocks from 907PC331 series 40..50 46  
CPU status, display S40..50 32

---

**D**

data types S40..50 36  
declaration editor S40..50 18  
downloading a program S40..50 28

---

**E**

example program S40..50 3

---

**F**

forcing S40..50 12, 24, 30  
Function Block Diagram S40..50 17  
functional scope S40..50 46

---

**G**

**global variables S40..50 34**

---

**I**

IEC standard library for S40..50 42  
import files S40..50 5, 34  
Import program from 907PC331 S40..50 27  
import upload file S40..50 27  
Import variables from 907PC331 S40..50 27  
Instruction List (IL) S40..50 16  
internal operands S40..50 24  
interrupt routines S40..50 14

---

**L**

Ladder Diagram S40..50 17  
loading a program S40..50 11

login S40..50 21  
logout S40..50 28

---

**M**

main program S40..50 14

---

**O**

operators for S40..50 40

---

**P**

password S40..50 31  
PLC selection S40..50 35  
program blocks S40..50 14  
program upload from PLC S40..50 33  
project files S40..50 3  
project name S40..50 / S90 10

---

**R**

real time clock S40..50 31  
reset S40..50 29

---

**S**

Sequential Function Chart S40..50 16  
simulation S40..50 23  
start program S40..50 29  
stop program S40..50 29  
Structured Text S40..50 16  
subroutines S40..50 14  
system constants S40..50 4

---

**V**

variables S40..50 38

---

**W**

warm start S40..50 29  
write values S40..50 29





---

**ABB STOTZ-KONTAKT GmbH**

Eppelheimer Straße 82 Postfach 101680  
D-69123 Heidelberg D-69006 Heidelberg

Telephone +49 6221 701-0  
Telefax +49 6221 701-1111  
E-Mail [desst.help@de.abb.com](mailto:desst.help@de.abb.com)  
Internet <http://www.abb.de/sst>

**ABB Control**

10, rue Ampère Z.I. - B.P. 114  
F-69685 Chassieu cedex France

Telephone +33 (0) 4 7222 1722  
Telefax +33 (0) 4 7222 1935  
E-mail  
Internet <http://www.abb.com/lowvoltage>