

Table of Contents

4	Programming the Advant Controller 31-S	4- 1
4.1	Survey over the 907 PC 339	4- 1
4.2	Procedure	4- 1
4.3	Path structure	4- 1
4.4	Data structure	4- 2
4.5	Program structure	4- 3
4.6	Procedure in detail	4- 7
4.7	Copying projects in 907 PC 339	4-12
4.7.1	Using the 907 PC 339 copy function	4-12
4.8	Transferring 907 PC 33 projects to 907 PC 339	4-13
4.9	Program COMPARE	4-13
4.10	Example for a configuration in a safety-related application	4-14

4 Programming the Advant Controller 31-S

4.1 Survey over the 907 PC 339

For programming the Advant Controller 31 S the programming software 907 PC 331 and additionally the safety package 907 PC 339 is required.

The software has been prepared for the generation of a safety-related project. It comprises:

- a special *library with safety-related connection elements* for the AC31-S (see volume 5 in this file)
- one *example project* which demonstrates the structure of a safety-related project
- one *default project* which the user must take as reference for the generation of his own safety-related project. When opening a new project, the default project is automatically loaded under the new project name.

4.2 Procedure

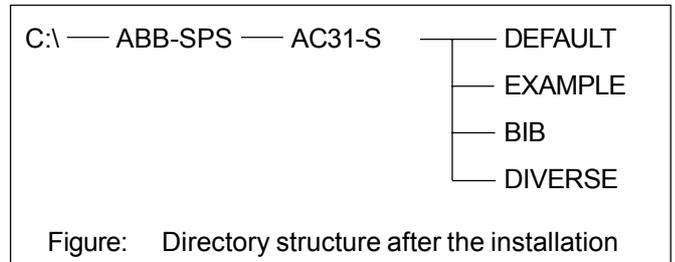
The procedure when programming/changing an Advant Controller 31-S can be roughly split up into the following steps:

- The user defines a specification for his safety functions
 - Analysis of the functions
 - Description of the dependencies.
- Open the project.
- Define the data structure in
 - safety-related part and
 - non-safety-related part.Here, predefined values exist (see paragraph 4.5).
- Programming the application
 - in safety-related part observing certain predefined data in 4.5
 - in non-safety-related part in function block diagram.
- Program in Advant Controller 31-S:
 - load, save program in Flash-EEPROM
 - test
 - modify (taking the check lists described in volume 7 of this manual into account).
- Compare program in controller with program in programming system (which is a menu item).
- Checking whether both programs are identical.
- Archive the project.
- Document the project.
- After TÜV acceptance:
 - Verify of the Flash-EEPROM.

A complete listing of all individual steps is included in the safety-related checklist (volume 7).

4.3 Path structure

Each project in the 907 PC 339 is stored in his own directory in DOS. After the installation of 907 PC 339, the directories are organized in accordance with the following figure.



The EXAMPLE directory holds a copy of the default project. Here, you can make yourself familiar with the structure of a safety-related project and also load the example to the control system, if necessary.

In the DEFAULT directory, the default project is stored. It must not be changed by the user as internal control functions for the Advant Controller 31-S have already been pre-programmed here.

In the DIVERSE directory you can store everything which is not a safety-related project. For instance, this directory holds the online and simulation lists already known from the 907 PC 331.

The BIB directory contains the library with the safety-related and non-safety-related connection elements.

4.4 Data structure

User flag ranges (work flags)

All word flags MW xx, yy, which are used in the safety-related program part, must be initialized at the program start with the CEs S_INIT and S_INSK (already included in the DEFAULT project). This definition was evaluated according to typical applications and will be sufficient for the most applications.

The work flags from the safety-related program part (see also under 'Program structure') may not be used without a special format conversion within the non-safety-related program part. This program conversion is executed in the program module S_APP-N.

Data ranges

For programming in FBD and for the translation into instruction list, flag ranges and labels are specified for jumps in the data range editor of the 907 PC 339. These flag ranges and labels are pre-defined for all program modules

of the default project. The definition can be found in the data range editor on program module level.

- ! If further program modules are to be added in the application, the data range of the newly added module must be checked for overlapping and, if necessary, corrected.

The data ranges of various program modules must not overlap with reference to the global variables as M, MW, MD and MRK. Any overlapping can be checked by pressing <F3> in the data range editor on the overall project level. The overall data range editor differs from the data range editor on program module level. If you want to page forward and backward between the global local bit and word flags, press <F1> or <F2> in the overall data range editor.

All M, MW, MD and MRK, defined in the data range editor, are used by the programming system 907 PC 339 during the translation. They must not be used by the user.

4.5 Program structure

Program modules (PMs)

A new project must be called as follows:

C:\ABB-SPS\AC31-S\AC31-S<projectname>

A new sub-directory is generated:

C:\ABB-SPS\AC31-S\<projectname>

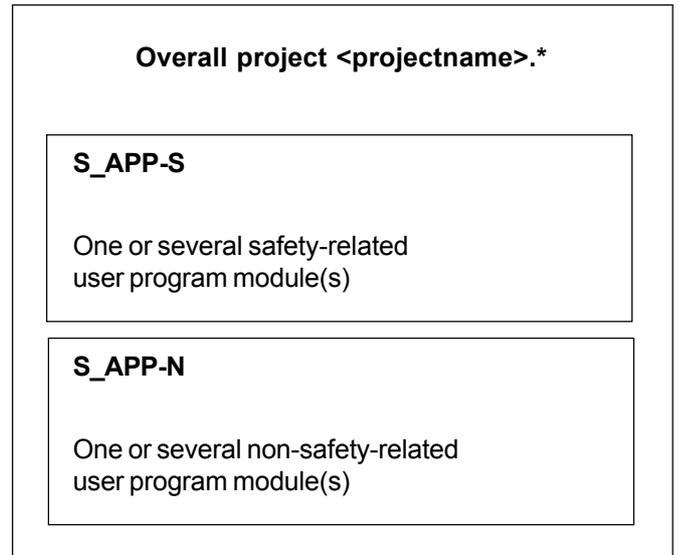
The default project is automatically copied to this directory, it consists of two parts:

- The first part is the safety-relevant user program module (S_APP-S) which may no longer be changed after the acceptance by an acceptance institute. If necessary, further modules can be added. In this module, all important parameters and functions are pre-defined, some of them can be modified by the user.
- The second part is the changeable non-safety-relevant user program module (S_APP-N). If necessary, further modules can be added.

These program parts are stored in various files so that any unintended modification can be excluded with high probability. The linkage of these program modules (PMs) independent of each other is made in the modularization editor in the 907 PC 339. The order of the PMs specified in the table of this editor must not be changed.

These program modules are comprised under the overall project name. The overall project name is identical to the sub-directory name in which these project files are stored. It is also the overall project name, with which the 907 PC 339 always must be started.

The following figure gives a rough overview over the program modules in a safety-related program.



It is possible that one project comprises several safety-related plus several non-safety-related user program modules.

If the safety-related application is to be spread to several PMs, the overall structure as described for the PM "S_APP-S" in the following must remain unchanged. For instance, the module "S_APP-S" could be split at segment plan 14; the lateral could be assigned to a PM "S_APP-S1" while the remaining segment plans would be stored in a new PM under the name "S_APP-S2".

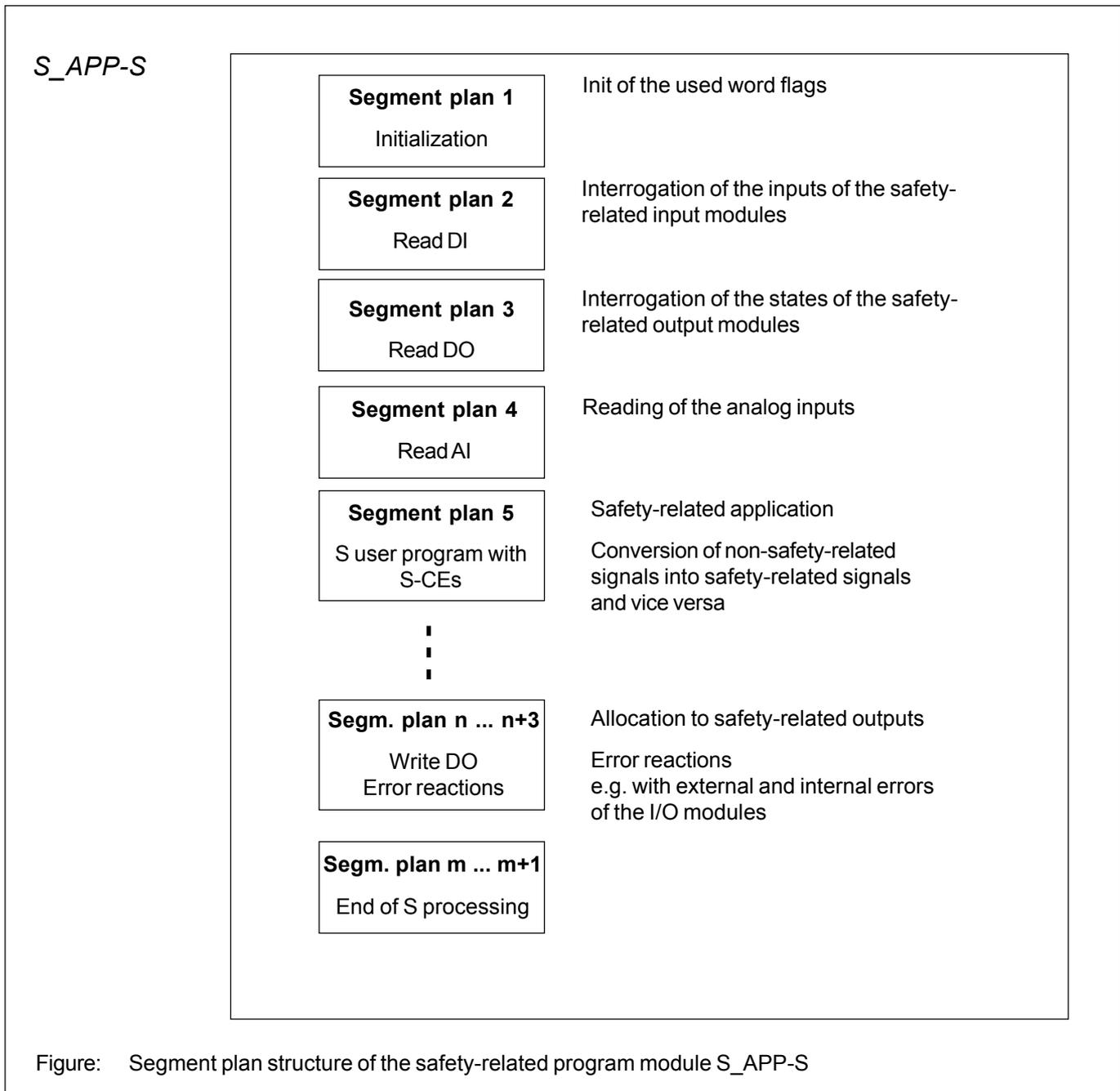
Segment plan structure

Each program module comprises several segment plans which result in a contents-related and logical structure. The following figures show the structure of each individual program module.

Program module S_APP-S

Here, the safety-related user program is stored. As the following figure shows, this user program is included in a frame of parameterization, head and foot. In this program module, programming must only be made in FBDs and also only with safety-related connection elements.

The program must no longer be changed after the acceptance by the TÜV. For the verification, a program comparison "Compare" must be executed. This is described in chapter 4.9.



Segment plan 1: Initialization

Here, the flags used in the S module are initialized (S_INIT, S_INSK).

Segment plan 2: Read DI

Here, the safety-related binary data are read in from the safety-related input modules 07 DI 90-S.

For reading the S input modules, the connection element S_LEB is used.

Segment plan 3: Read DO

Here, the states of the safety-related output modules 07 DO 90-S are read in.

For reading, the connection elements S_LAB are used.

Segment plan 4: Read AI

Here, the safety-related analog data from the input modules 07 AI 90-S are read in.

For reading this input modules, the connection elements S_LEA are used.

Segment plan 6: Safety-related application (S processing)

The programming of the application is done exclusively using safety-related CEs, here.

Here, also the conversion of safety-related signals into non-safety-related signals is carried out and vice versa (e.g. the input of parameters via a control unit for safety-related further processing).

Segment plan n ... n+3: Write DO / Error reactions

Here, the allocation is made to safety-related outputs of the S module 07 DO 90-S. For writing of S output modules, the S-CEs S_SAB are used.

Here, also the reaction to external and internal errors on I/O modules is planned. Using the S-CE ABORT it can be precisely defined under which conditions the control program is to be aborted.

Segment plan m ... m+1:**End of the safety-related application**

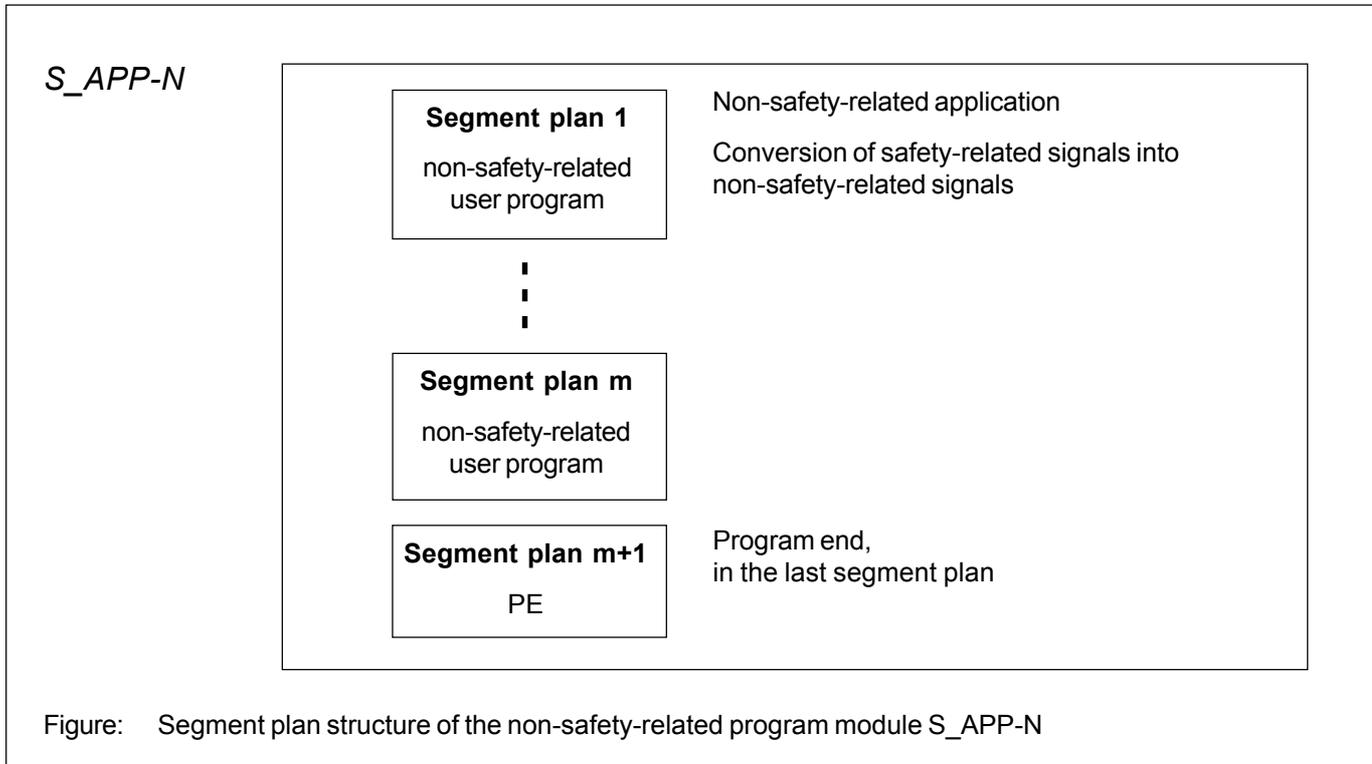
Here, the jump destination of the jump defined in the second segment plan is specified.

Program module S_APP-N

Non-safety-related application

In the non-safety-related program part, all non-safety-related functions are executed. Safety-related work flags may be used here if they are converted into the non-safety-related format. This must be done at the beginning of the first segment plan within the program module. Above all, this makes sense if these variables are used for sig

nalling or for an operating and monitoring device. The conversion from the S format into the conventional binary format is done using the CE S/B. For word flags in the S word format, the CE S/W is used. The placement of this conversion (normal binary/word format) in the non-safety-related program module bears the advantage that the user can access safety-related signals subsequently without a change in the safety-related program module being necessary.



4.6 Procedure in detail

Planning

The programming in 907 PC 339 is preceded by a detailed planning. The programming is nothing else than the conversion of a carefully planned and sufficiently documented solution into a description understood by the automation system.

Using the safety-related check list to be found in volume 7, the operator must first plan the required signals and split up the tasks into a safety-related and a non safety-related part. This is done in close context with the planning of the hardware as described in volume 3 of this manual.

Programming environment

Make sure that the latest 907 PC 339 software version is installed on your personal computer and that the PC specifications meet the hardware requirements as described in volume 1 of this file.

Opening a new project

The program modules with the pre-defined segment plans explained under the program structure are available for each new project.

For opening a new project or for editing an existing project, type the following:

```
AC31-S <projectname> <Enter>
```

After the input of a new project name, a sub-directory under the same name is generated. The default project is copied to the new project. This is done using the "Copy project" function of the 907 PC 339 and runs fully automatically. After copying has been completed (this can take some time), the project can be edited under the new name <projectname>.

The default project only sets the frame. Some of the CEs must still be assigned variables, e.g. which inputs of the S input modules are to be read.

The contents of the constants KW 0,0 ... KW 0,15 and KD 0,0 are already pre-assigned.

Signal assignment

Before starting the programming of the safety-related application, the signals must be split up into safety-related and non-safety-related signals as indicated in the check list. This definition should be made in the variable editor of the 907 PC 339. The safety-related signals are all defined in the variable editor of the program module S_APP-S. The non-safety-related signals are located in the program module S_APP-N.

Programming

After all settings have been made, you can start programming the application.

Module S_APP-S

The safety-related part must be programmed in the function block diagram (FBD).

In order to edit the safety-related program module S_APP-S, select "Project man." in the main menu and call "Program module". The program module level of S_APP-S appears. For programming, now call the FBD/LD editor.

What you see now are the pre-programmed segment plans as represented under "program structure".

In segment plan 1, various safety MWs are initialized. If the number is not sufficient, the user can increase the number of MWs.

In the segment plans "Read DI, Read AI, Read DO" change the module addresses indicated there to the values corresponding to your system. Depending on the type and number of safety-related I/O devices to be used you must insert and configure further CEs of the types S_LEB, S_LAB or S_LEA. Concerning the configuration of these CEs please refer to the detailed description in volume 5 of this manual.

In the **segment plans 6 and following** for the safety-related application, the function sequence of the safety-related application is programmed exclusively using CEs. Here, also the conversion of signals from the non-safe-area into the safe-area is carried out and vice versa (CEs S_B/S, S_S/B, S_W/S and S_S/W).

In the **segment plans "Write DO"**, the assignment to the safety-related outputs of the S module 07 DO 90-S is made. For this purpose, the CEs S_SAB are used. Again, you must carry out the expansions necessary for your system configuration and match the module addresses to be indicated at the connection elements. Please note also the section "Program modifications" following further down. Concerning the configuration of this CE please refer to the detailed description in volume 5 of this manual.

Here, also the reaction to external and internal errors on I/O modules is planned in accordance with the specifications of the TÜV or the local inspector. With the CE S_ABORT, the operator can define precisely under which conditions the entire system is to be switched off and the control program is to be aborted.

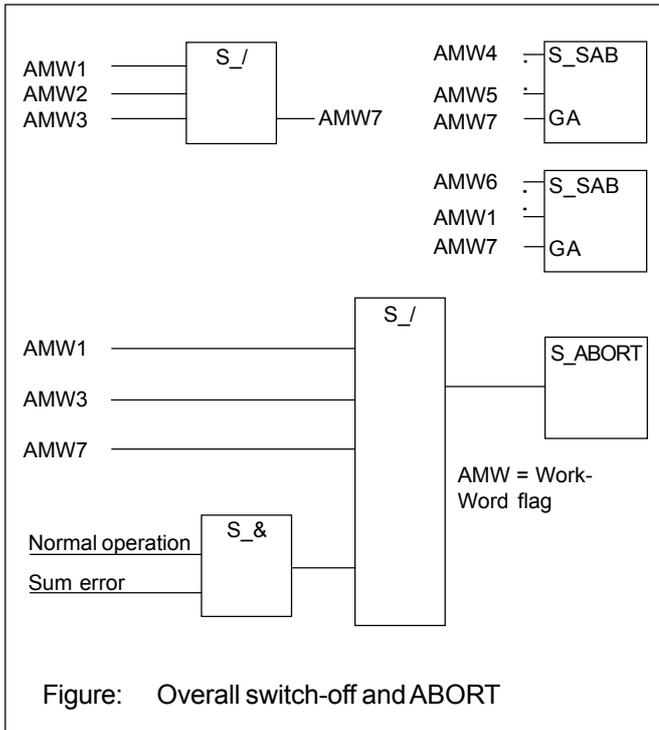
The following procedure is useful:

In the READ segment plans, the single and sum error outputs of the CEs S_LEB, S_LEA and S_LAB are assigned to the work flags AMWxx. If now a system shut down must be executed in accordance with the specifications for single channel or sum errors, the non-affected output modules will first be switched to the safe state

(zero state) via the overall switch-off (input GA of the CEs S_SAB). Only after this, the ABORT function will be initiated causing the shut down of the overall system.

If, in accordance with the TÜV specifications, a restricted safety operation after the occurrence of an external error is permissible, certain functions may still be released in accordance with the planning. E.g. in case of a crane application: Only load-reducing measures are still permissible after a defect of a limit switch.

For the behaviour in the service operation, it can also exactly be specified here when the ABORT function is to be carried out.



The figure above shows also that for a restricted safety operation the relevant signals must be distributed to several output modules.

Concerning the configuration of S_ABORT please refer to the detailed description in volume 5 of this manual.

Before you quit the FBD/LD editor, the generated program must still be translated. For this purpose, select the function "Translate all".

Please note the rules as indicated in the section "Guidelines for the program generation".

Module S_APP-N

Here, you will program the application which is not safety-related. We recommend to use the FBD/LD editor and have already pre-defined a module for you.

If you require binary variables from the safety-related program module, these must be converted into the non-safety-related binary format with the help of the CE S_S/B. For word variables, you will use the CE S_S/W.

Please note that, when adding further segment plans, the program end "PE" must be located in the last segment plan. In this last segment plan, no further instruction should be placed.

If you wish to add another program module, please read the section "Program changes".

Guidelines for the program generation

- Split-up into safety-related and non-safety-related application. The pre-defined module structure and segment plan structure must be kept under all circumstances.
- The data ranges of the individual modules, in particular of the global flags and labels, must not overlap. How to check this is described under "Data ranges" earlier in this manual.
- Global flag setting for safety modules.
- The safety-related part must be programmed in function block diagram (FBD).
- No files of other program software versions must be copied into the 907 PC 339. For instance, no function blocks must be installed to 907 PC 339. 907 PC 339 includes a complete list of all function blocks for AC31-S.
- No project files or segments may be copied under DOS. To copy, always use the block functions in 907 PC 339 or the "Copy project" function in the "Project man." menu at the overall project level.
- In the safety-related part, programming must be done exclusively using safety-related CEs. In volume 5 of this manual, you will find the description concerning the S_CEs.
- Carry out data backup at the end of each day where modifications have been made. Use the "Data backup" function in the main menu to do so. Further instructions can be found in the file 907 PC 33.
- If the message "Insufficient disk space" is displayed, quit the programming software 907 PC 339 immediately and provide more capacity on your hard disk.

Global flag setting for safety modules

Setting must be realized for each safety module

Realization

Customer action

Reaktion of 907 PC 339

Start main menu

907PC339 shows main menu

Call for "**program module**"



907PC339 shows window with
S-APP-S ;safety module
:
S-APP-N

Select "**S-APP-S**"



907PC339 shows window module menu S-APP-S

In block "config PC33" select "**module configuration**"



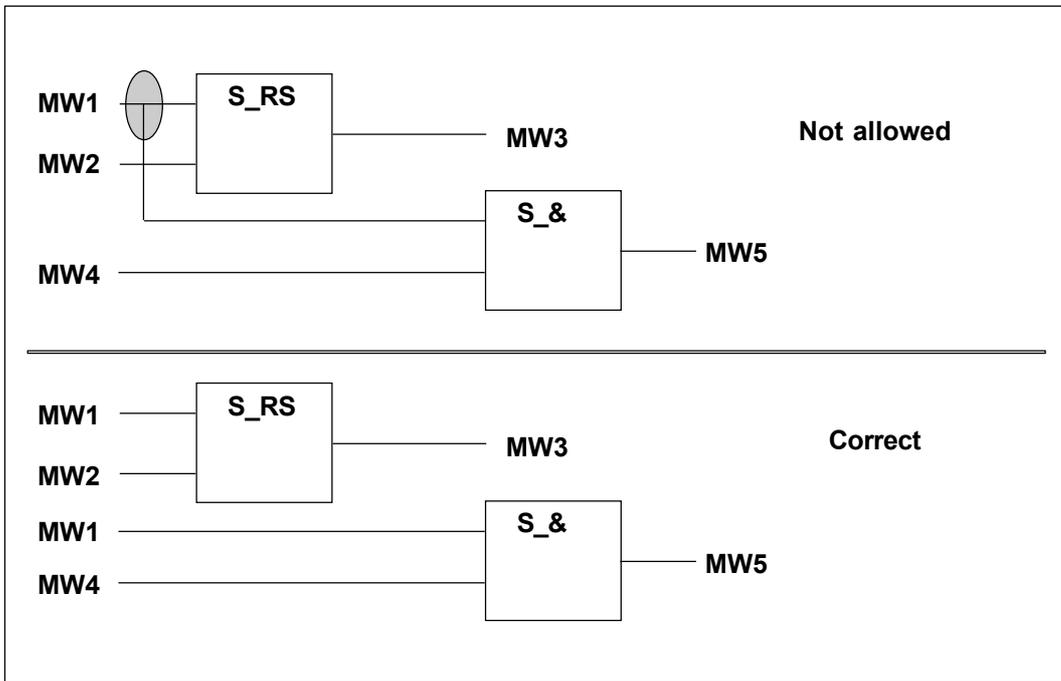
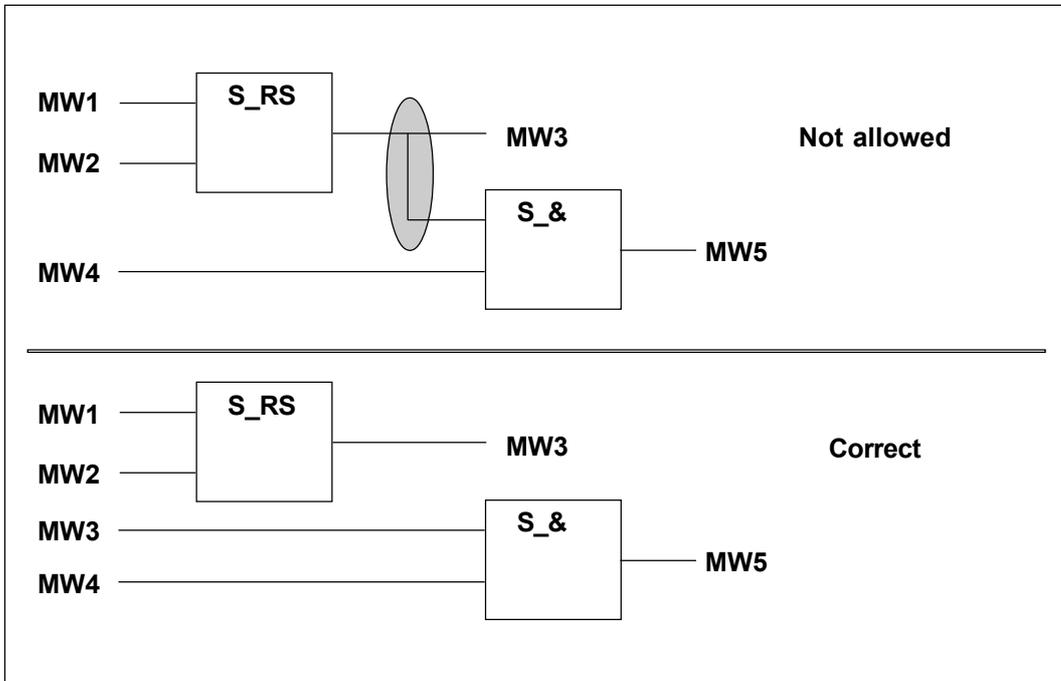
907PC339 shows window "**module configuration**"

Select: datarange "module"
datafault flag "global"

Confirm with "ESC"



907PC339 accepts the settings



Load and test program

For testing the generated program, it must first be loaded into the central unit 07 KT 94-S. During this phase, the program operation from a RAM memory is possible and also recommended due to the modifications which frequently have to be carried out.

Online functions and online modifications

The following functions familiar from 907 PC 331 are not available in 907 PC 339, because they cause a safety shut down when the control system is running (Central error FK2): Set breakpoint, Single cycle, Stopping, Jogging and Continue.

The handling and operation of the further functions available under 907 PC 339 for testing the program is described in the file 907 PC 331.

The functions

- forcing and
- online modifications

are permissible for S applications. If, however, the user program includes time elements such as S-ESV and S-ASV, the PLC cycle stops after an online modification and displays a FK2 error. After this, a reset must occur, e.g. supply voltage on/off.

If during commissioning or in normal operation a shut down of the error class 1 or 2 (LED FK1 or FK2) occurs, the error number and address where the error occurred can be displayed in terminal operation using the command ST<CR> (status).

e.g.

ST <CR>

```
Program identification: .....
cycle time:           .....
Program status:       .....
Active test function: .....
TRACE register:       .....
error message:        .....
```

Error class FK2: 510

yyyy zzzz 0000 0000

FK2	Error class 2
510	Error number (see volume 7 under "Diagnosis and Trouble Shooting")
yyyy	Program address in Hex related to the absolute program start
zzzz	Additional information to individual error numbers
0000	No significance

How to determine the program address where an error occurred?

- Start up the terminal operation, enter the command ST<CR>, note the yyyy address (see definition of the error numbers) and translate it to decimal notation.
- If the project includes only one safety-related module S_APP-S, the above mentioned address must be located in this module. The defective program function can be displayed with the menu item "Display translated IL" (Menu: FBD, space key, Translate and send, Display translated IL).
- If the project includes two or more S modules, it is necessary to determine first in which module the error occurred. For this purpose, select the first module under Config PC33, Data ranges. The module address is displayed in the head of the representation.
S: xxxx Start address of the module
E: yyyy End address of the module
L: zzzz Length of the module
With this information the absolute error address within the module can be determined.

Example:

Decimal address from terminal operation = 6388

1. safety-related module (S_APP-S1)

S: 0000 E: 3145 L: 3144

2. safety-related module (S_APP-S2)

S: 3146 E: 4358 L: 4357

As the error address is higher than 3145, it is located in the second module.

$6388 - 3146$ (start address 2. Module) = 3242

The error function occurred at address 3242 in the second safety-related module.

In the main menu "Call Progr. Module", select the 2. module S_APP-S2, Edit, FBDLD, Space key, Translate/Send, Display translated IL. In the column "WN" (word number) search the address 3242. Here, the error occurred.

Determining the system reaction time

This item is important for the further process of the project and should therefore be carried out already during the program development. The system reaction time is subject to a limit. The calculation of the entire system reaction time is described in detail in volume 3, chapter 3.1 "Pre-conditions" in this manual. The cycle time of the PLC program is included in this calculation.

The cycle time can be determined in the "terminal emulation".

Enter:

```
> KD 00,00 <CR> displays  
> KD 00,00 00010 i.e. 10 ms
```

It can be changed in steps of 5 ms, however the maximum of 70 ms should not be exceeded.

In this connection also the capacity utilization of the central unit is important:

Enter:

```
> AL <CR>  
> AL 80 i.e. 80%
```

That means with a cycle time of 10 ms the central unit needs 8 ms for processing the program and 2 ms runs in "no-load operation". However, during this time it must e.g. also handle the interfaces COM1 and COM2.

The value of AL should never become higher than 80, otherwise the cycle time must be increased in flag KD 00,00 (see Calculating the max. cycle time, volume 3, chapter 3.1.2).

Programming EPROMs

After each modification, the programmer should also save the program in the integrated Flash-EPROM (Command in main menu, PLC Commun. 2, Progr. EPROM, or "SP" in the terminal emulation).

Verify EPROMs

For this purpose, the function "Compare program" is carried out. The project generated on the programming system 907 PC 339 is compared to the program stored in the EPROM. If the PLC program in the EPROM matches the project, the message "No differences" is displayed.

Course:

- Translate and send program completely
- Save program in Flash-EPROM
- Carry out cold start (menu item) or supply voltage off/on. This will copy the content of the Flash-EPROM into the RAM.
- "Verify program". This comparison must be executed at the end of the commissioning in any case. This ensures that the correct program is available in the central unit.

Archiving the program

All files stored in the sub-directory of the safety-related project must be archived after successful starting if no further changes to the program are to be carried out.

Archive the entire sub-directory <projectname>. Use the data backup function of the 907 PC 339 to perform this. Further instructions can be found in the file 907 PC 33.

Do not forget to archive also the library in \BIB if you have generated your own CEs for the non-safety-related project part.

If the programming has been completed, also the instruction list of all program modules, the function block diagrams converted into IL must be archived. For this purpose, enter the program module levels of the individual modules and select the function "Print 1", "IL". As print target, the expression "PRN" must be changed into a file name e.g. "<name>.DRU". The print-out will then be made into this file. This file must then be archived and used as a basis for a comparison in case of later program changes.

Note:

Comments must not be placed within sets. (This is excluded when programming in FBD/LD).

The left margin of the print file must be set to 10 (main menu item *Print param.* -> *sub-menu Print parameters* -> *Left margin: 10*)!

Program documentation

The project, i.e. the program modules and the settings in the various editors of the 907 PC 339, must be documented completely. For this purpose, the 907 PC 339 provides appropriate utilities.

The program modules must be documented individually by printing out the following from the 907 PC 339:

- FBD/LD (if necessary IL, if used for non-safety-related modules)
- Data ranges
- System configuration
- Variable lists

From the overall project level, the following must be printed:

- Overall reference list has already been printed during the check for using reserved variables.
- Total data range
- Modularization data

Program changes

- Modifications during program development:
 - Adding of a further non-safety-related program module: Matching the entries in the data range editor of the new module so that any overlapping with the ranges of other modules are excluded. Check as described under "Data ranges" in chapter 4.4 "Data structure". If this program module is placed after the pre-defined program module "S_APP-N", the program end must be removed from the last segment plan of "S_APP-N" and be placed in the last segment plan of the newly added program module.

- Modifications after the program acceptance by an acceptance institute:

- Changes in the non-safety-related part

Proof must be made that the changes cause no reactions in the safety-related part. This is done by comparing the new project to the old project using the archived files <modulename>.DRU. The program COMPARE is used (see chapter 4.9).

- Changes to the safety-related part

After that, a new acceptance by the acceptance institute is necessary. The same procedure as described above must be used again.

4.7 Copying projects in 907 PC 339

If you have already created a safety-related project with 907 PC 339 and would like to re-use and modify it, you have the following possibility:

4.7.1 Using the 907 PC 339 copy function

You call the function by selecting the menu items *Project man.* -> *Copy project*. The following menu appears:

ABB 07 KT 94-S	Copy project	EXAMPLE
Source project__: (path and name of the project to be copied)		
Target project__: (path and name of the new project)		
Copy project_____ : Y		
Copy library_____ : N		
Copy all files to project__: Y		
Enter the name of the source project.		
<F1> Start copying		
<ESC> Abort		

Enter the path and name of the project to be copied in the "Source project" field. You can open an overview of the existing projects by pressing <Ctrl>-U.

Enter the path and name of the new project in the "Target project" field. You can specify any directory you want. If the directory to which you want to copy the project does not yet exist, it is automatically created during the copy procedure. After this you have to specify whether the project and library or only one of the two are to be copied. We recommend not to copy the library together with the project. You should always work with the same library.

The elements selected are copied after pressing <F1>.

For modularized projects, i.e. for all safety-related projects, a different target directory must be specified.

Copy all files to project:

(Y): All program and variable modules are copied to the same directory as the target project. However, if the source project has a path structure, this is lost. This is not the case, however, with safety-related projects. All files of a project are always stored in the same directory, with the exception of the library which is stored in the BIB directory.

4.8 Transferring 907 PC 32 projects to 907 PC 339

Non-safety related projects generated in 907 PC 32 can be transferred to 907 PC 339 with the data transfer program supplied with 907 PC 339. This enables you to re-use older projects. Frequently used program parts then do not have to be re-written. A detailed description of how to proceed can be found in the file "907 PC 33 General Part".

4.9 Program COMPARE

After program modifications, it must be proved that the PLC program for the safety-related part has not been changed after the latest acceptance inspection by the TÜV or by the local inspector. For this purpose, the printed instruction lists of the individual program modules before and after a program change must be compared to each other. This comparison is carried out using the auxiliary program COMPARE.

COMPARE is stored in the directory ABB-SPS\AC31-S.

Function of the COMPARE program

COMPARE is an auxiliary program for the comparison of two IL files. It compares two print files set-by-set starting from a certain address up to the end of the shorter file. It only reads the instructions of the instruction lists printed into the files. Head and foot notes, comments and symbols are ignored. As a result, all those sentences which do not match are displayed. If both sentences compared vary in length, both are displayed. The comparison will then be continued within the next sentence. If the only difference between both files is that sentences were deleted or added, the COMPARE program will show all sentences following a deleted or added sentence as mismatches as well. In this case, COMPARE offers the option of starting the comparison with a different sentence number in each file. Different comparison types are selected via parameter.

Notes:

For comparing the versions of the safety-related program module, the parameters should be selected in a way that differences are not ignored. These parameters are marked with *.

Adapting the data ranges to the respective application program can cause differences in the scratch flags of the individual modules when comparing.

Before using the COMPARE program

If you wish to compare two AC31-S program modules to each other you must print out the IL list of each program.

This is done via the main menu item *Print 1* -> submenu *IL*. As print target, you simply enter a file name instead of the pre-defined value PRN.

Note:

Comments must not be placed within the sentences of the ILs to be printed.

The left margin of the print file must be set to 10 (main menu item *Print param.* -> submenu *Print param.* -> *Left margin: 10*)!

Using the COMPARE program

In order to use COMPARE, you must use the following syntax:

```
COMPARE file1 file2 output /parameter
```

File1 and *file2* are the names of both files to be compared.

The comparison result can be displayed or stored in different ways by specifying the *output* parameter:

- If you leave out *output* the comparison result is displayed on the screen.
- If you specify *PRN* the comparison result will be output on your printer.
- If you specify a filename, the comparison result will be stored in this file.

By the specification of various parameters, you can control the type of comparison and the output. Several parameters are simply specified together, e.g. *'/DN'*.

The following parameters are available:

<i>/L</i>	Ignore sentence length Usually, i.e. without this parameter, sentences of various lengths will not be checked any further for the reason of the mismatch. This parameter allows to compare sentences of various lengths from the beginning of the sentence up to the end of the shorter set or up to the first mismatch.
<i>/S</i>	Ignore sentence number This parameter allows to compare sentences if these have different sentence numbers.
<i>/W</i>	Ignore word number Using this parameter, the word numbers are ignored during comparison.
<i>/C</i>	Ignore commands Using this parameter, the commands within the words are ignored during comparison. This will for instance ignore the difference between <i>'&'</i> and <i>'&N'</i> .
<i>/T</i>	Ignore operand type Using this parameter, the operand types are ignored during comparison.
<i>/A</i>	Ignore operand address Using this parameter, the operand addresses are ignored during comparison.

<i>/D</i>	Indicates the number of mismatched sentences only. Using this parameter, only the number of different sentences is indicated.
<i>/N</i>	Screen display without interruption Usually, the display of the comparison results is stopped after the screen is full and can be continued by pressing a key. This interruption is made ineffective by this parameter.
<i>/Qn</i>	Abort after n differences Using this parameter, the display is aborted after n differences have been found.
<i>/1#xxxxx</i>	Comparison in file1 starting with sentence number <i>xxxxxx</i> . This parameter specifies that the comparison will be started in file1 from sentence number <i>xxxxxx</i> . The sentence number specified must always have 5 digits (e.g. <i>/1#01013</i>). If this sentence number does not exist an error message is output. If different start addresses are specified for both files, this will automatically switch on the parameters <i>/S</i> (ignore sentence number) and <i>/W</i> (ignore word number).
<i>/2#xxxxx</i>	Comparison in file2 starting with sentence number <i>xxxxxx</i> . This parameter specifies that the comparison will be started in file2 from sentence number <i>xxxxxx</i> . The sentence number specified must always have 5 digits (e.g. <i>/2#01013</i>). If this sentence number does not exist an error message is output. If different start addresses are specified for both files, this will automatically switch on the parameters <i>/S</i> (ignore sentence number) and <i>/W</i> (ignore word number).

After the comparison has been completed, the number of found mismatches is indicated. If no differences were encountered, the files are identical (except for differences of ignored elements). The comparison can be aborted by pressing <Esc>.

Examples:

```
COMPARE AWL1.DRU AWL2.DRU
```

Both files will be compared and the comparison result will be displayed on the screen.

```
COMPARE AWL1.DRU AWL2.DRU PRN /1#00002T
```

Both files will be compared. In file AWL1.DRU, the comparison will be started with sentence number 2. Furthermore the operand types are ignored. The comparison result is output on the printer.

```
COMPARE AWL1.DRU AWL2.DRU compare.dat
```

Both files will be compared. The comparison result will be written into the file compare.dat.

4.10 Example for a configuration in a safety-related application

The following figure shows the example configuration of a safety-related application. It also shows that non-safety-related I/O modules can be used together with safety-related I/O modules on one line of the CS31 system bus. Of course, only non-safety-related tasks may be carried out with the non-safety-related I/O modules. The corresponding programming is then made in the program module S_APP-N.

Configuration

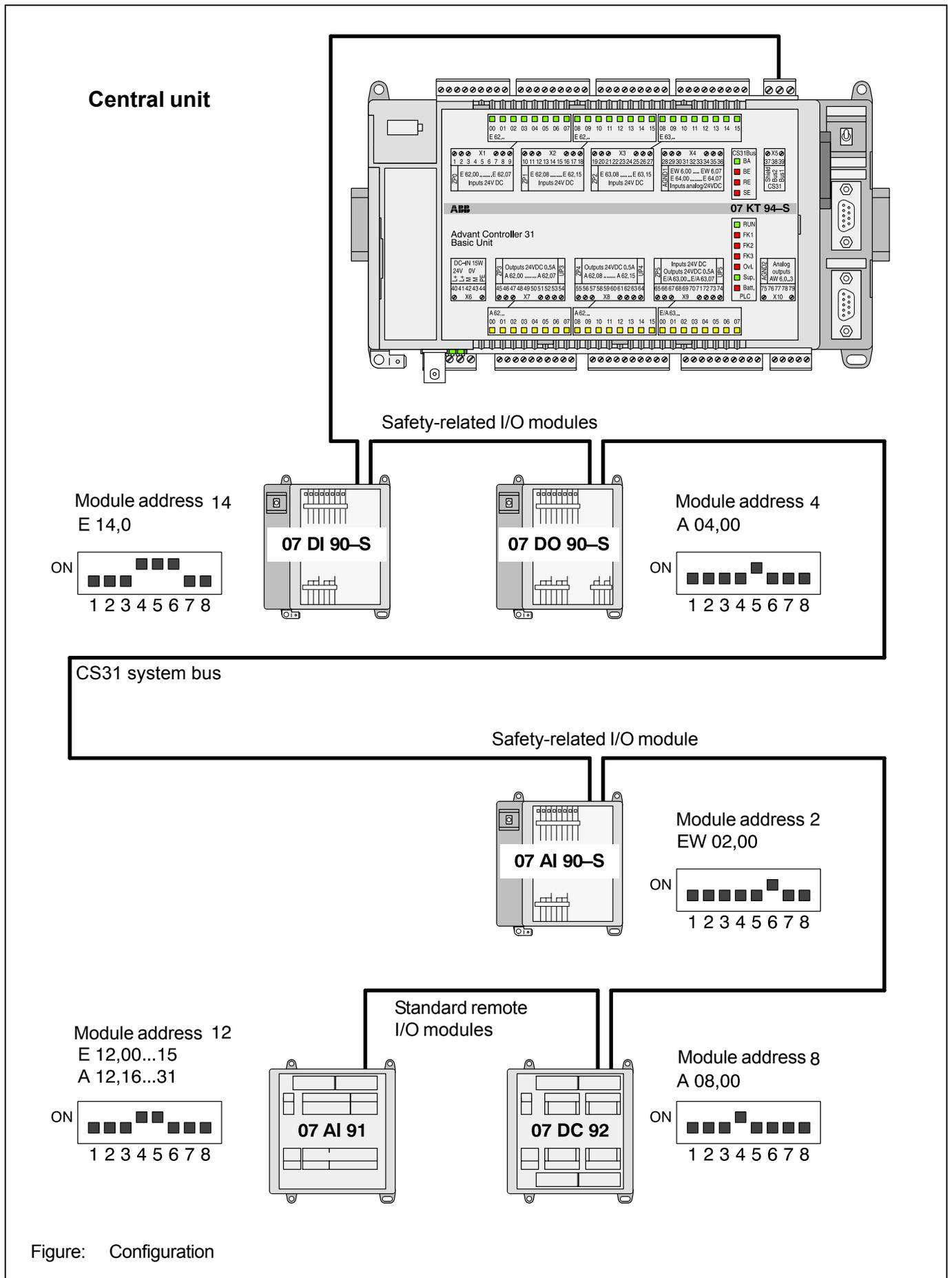


Figure: Configuration



ABB STOTZ-KONTAKT GmbH

Eppelheimer Straße 82 Postfach 101680
69123 Heidelberg 69006 Heidelberg
Germany Germany

Telephone +49 6221 701-0
Telefax +49 6221 701-240
E-Mail desst.help@de.abb.com
Internet <http://www.abb.de/stotz-kontakt>