

SAIA® PCD
Process Control Devices

**Programming tools
for WINDOWS - PG 4**



SAIA-Burgess Companies

Switzerland	SAIA-Burgess Electronics AG Freiburgstrasse 33 CH-3280 Murten ☎ 026 672 77 77, Fax 026 670 19 83	France	SAIA-Burgess Electronics Sàrl. 10, Bld. Louise Michel F-92230 Gennevilliers ☎ 01 46 88 07 70, Fax 01 46 88 07 99
Germany	SAIA-Burgess Electronics GmbH Daimlerstrasse 1k D-63303 Dreieich ☎ 06103 89 060, Fax 06103 89 06 66	Nederlands	SAIA-Burgess Electronics B.V. Hanzeweg 12c NL-2803 MC Gouda ☎ 0182 54 31 54, Fax 0182 54 31 51
Austria	SAIA-Burgess Electronics Ges.m.b.H. Schallmooser Hauptstrasse 38 A-5020 Salzburg ☎ 0662 88 49 10, Fax 0662 88 49 10 11	Belgium	SAIA-Burgess Electronics Belgium Avenue Roi Albert 1er, 50 B-1780 Wemmel ☎ 02 456 06 20, Fax 02 460 50 44
Italy	SAIA-Burgess Electronics S.r.l. Via Cadamosto 3 I-20094 Corsico MI ☎ 02 48 69 21, Fax 02 48 60 06 92	Hungary	SAIA-Burgess Electronics Automation Kft. Liget utca 1. H-2040 Budaörs ☎ 23 501 170, Fax 23 501 180

Representatives

Great Britain	Canham Controls Ltd. 25 Fenlake Business Centre, Fengate Peterborough PE1 5BQ UK ☎ 01733 89 44 89, Fax 01733 89 44 88	Portugal	INFOCONTROL Electronica e Automatismo LDA. Praceta Cesário Verde, No 10 s/cv, Massamá P-2745 Queluz ☎ 21 430 08 24, Fax 21 430 08 04
Denmark	Malthe Winje Automation AS Håndværkerbyen 57 B DK-2670 Greve ☎ 70 20 52 01, Fax 70 20 52 02	Spain	Tecnosistemas Medioambientales, S.L. Poligono Industrial El Cabil, 9 E-28864 Ajalvir, Madrid ☎ 91 884 47 93, Fax 91 884 40 72
Norway	Malthe Winje Automasjon AS Haukelivn 48 N-1415 Oppegård ☎ 66 99 61 00, Fax 66 99 61 01	Czech Republic	ICS Industrie Control Service, s.r.o. Modranská 43 CZ-14700 Praha 4 ☎ 2 44 06 22 79, Fax 2 44 46 08 57
Sweden	Malthe Winje Automation AB Truckvägen 14A S-194 52 Upplands Väsby ☎ 08 795 59 10, Fax 08 795 59 20	Poland	SABUR Ltd. ul. Druzynowa 3A PL-02-590 Warszawa ☎ 22 844 63 70, Fax 22 844 75 20
Suomi/ Finland	ENERGEL OY Atomitie 1 FIN-00370 Helsinki ☎ 09 586 2066, Fax 09 586 2046		
Australia	Siemens Building Technologies Pty. Ltd. Landis & Staefa Division 411 Ferntree Gully Road AUS-Mount Waverley, 3149 Victoria ☎ 3 9544 2322, Fax 3 9543 8106	Argentina	MURTEN S.r.l. Av. del Libertador 184, 4° "A" RA-1001 Buenos Aires ☎ 054 11 4312 0172, Fax 054 11 4312 0172

After sales service

USA	SAIA-Burgess Electronics Inc. 1335 Barclay Boulevard Buffalo Grove, IL 60089, USA ☎ 847 215 96 00, Fax 847 215 96 06
------------	---

SAIA® Process Control Devices

Programming tools for MS WINDOWS

PG4 - Version 1.3

SAIA-Burgess Electronics Ltd. 1996. All rights reserved
Edition 26/748 E1 - 05.1996

Subject to technical changes

Updates

Manual : Programming tools for MS WINDOWS - PG4 Version 1.3 - Edition E1

Date	Chapter	Page	Description
27.10.2000	---	---	Small updates for the "Support Homepage"

Contents

	Page
1. Introduction, overview	1-1
1.1 Brief description of the PG4	1-1
1.2 Who works with the PG4 ?	1-3
1.3 Individual tools of the PG4	1-4
1.3.1 FUPLA (Function Block Diagram : FBD)	1-4
1.3.2 KOPLA (Ladder Diagram : LD)	1-5
1.3.3 SAIA® GRAFTEC (Sequential Flow Chart : SFC)	1-6
1.3.4 IL Editor (IL = Instruction List)	1-8
1.3.5 Configurator	1-9
1.3.6 Project Library	1-10
1.3.7 Project Manager	1-11
2. Installation of the PG4	2-1
2.1 Personal computer requirements	2-1
2.2 Installation	2-2
2.3 Contents of the PG4 directory	2-6
2.4 Retrospective inst. of individual function families	2-7
2.5 Delete installation	2-8
3. Project Library, Project Manager and Configurator	3-1
3.1 Running the Project Library	3-1
3.1.1 'File' submenu	3-2
3.1.2 'View' submenu	3-5
3.1.3 'Help' submenu	3-6
3.1.4 Toolbar	3-7
3.1.5 Status bar	3-7

	Page	
3.2	Running the Project Manager	3-9
3.2.1	Main menu of Project Manager	3-9
3.2.2	'File' submenu	3-10
3.2.3	'View' submenu	3-14
3.2.4	'Resource' submenu	3-15
3.2.5	'Make' submenu	3-17
3.2.6	'Online' submenu	3-20
3.2.7	'Tools' submenu	3-25
3.2.8	'Help' submenu	3-26
3.2.9	Toolbar	3-27
3.2.10	Status bar	3-28
3.3	Running the Configurator	3-29
3.3.1	Calling the main window	3-29
3.3.2	"Connection" function	3-32
3.3.3	"Memory" function	3-34
3.3.4	"S-BUS" function	3-38
3.3.5	"Clock" function	3-44
3.3.6	"History" function	3-45
3.3.7	"Password" function	3-46
4.	FUPLA (Function Block Diagram : FBD)	4-1
4.1	Introduction	4-1
4.2	Description of menus	4-2
4.2.1	Main menu (main window)	4-2
4.2.2	'File' submenu	4-4
4.2.3	'Edit' submenu	4-7
4.2.4	'Page' submenu	4-13
4.2.5	'Resource' submenu	4-18
4.2.6	'Compile' submenu	4-22
4.2.7	'Online' submenu	4-24
4.2.8	'Help' submenu	4-30
4.3	Using FUPLA	4-31
4.3.1	Creating a new function plan	4-31
4.3.2	Viewing an existing function plan	4-39
4.3.3	Changing an existing function plan	4-40

	Page	
4.4	The FUPLA function families	4-41
4.5	KOPLA functions (Ladder Diagram: LD)	4-49
	These two very extensive chapters are contained in a separate manual entitled	
	"The FUPLA and KOPLA function families"	
	Order code: "26/749 E"	
4.6	Introductory example	4-51
4.6.1	Standard function plan (FUPLA/FBD)	4-53
4.6.2	Function plan with ladder diagram symbols (KOPLA/LD)	4-70
4.6.3	Function plan with multiple pages	4-74
4.7	Further examples	4-79
4.7.1	FUPLA file with FB/PB/XOB edited in FBD	4-80
4.7.2	FUPLA file with FB/PB edited in IL	4-83
4.7.3	Adding a FUPLA file to an existing IL-Program	4-89
4.7.4	Program consisting of several FUPLA files	4-91
5.	GRAFTEC (Sequential Flow Chart : SFC)	5-1
5.1	Introduction	5-1
5.2	Description of functions and menus	5-2
5.2.1	Screen elements	5-2
5.2.2	'File' submenu	5-4
5.2.3	'Edit' submenu	5-9
5.2.4	'View' submenu	5-16
5.2.5	'Search' submenu	5-20
5.2.6	'Mode' submenu	5-22
5.3.7	'Online' submenu	5-25
5.2.8	'Page' submenu	5-30
5.2.9	'Help' submenu	5-32
5.2.10	Toolbar	5-35

	Page	
5.3	Using GRAFCET, introductory example	5-37
5.4	Further examples	5-61
5.4.1	GRAFTEC program edited in FUPLA/KOPLA	5-61
5.4.2	GRAFTEC program edited in IL	5-61
5.4.3	GRAFTEC with multiple sequential blocks (SB)	5-66
5.4.4	Combination of GRAFCET and IL program	5-70
5.4.5	Combination of GRAFCET and FBD program	5-73
6.	The IL editor (SEDTWIN)	6-1
	List of all SAIA [®] PCD instructions	from 6-2
Appendix A:	Error and warning messages	A-1
	Messages for the Projects Library	A-1
	Messages for the Project Manager	A-5
	Messages for the FUPLA	A-11
	Messages for the Resource Manager	A-29
	Messages for the GRAFCET	A-33
Appendix B:	Reservations in the PG4 - V 1.3	B-1



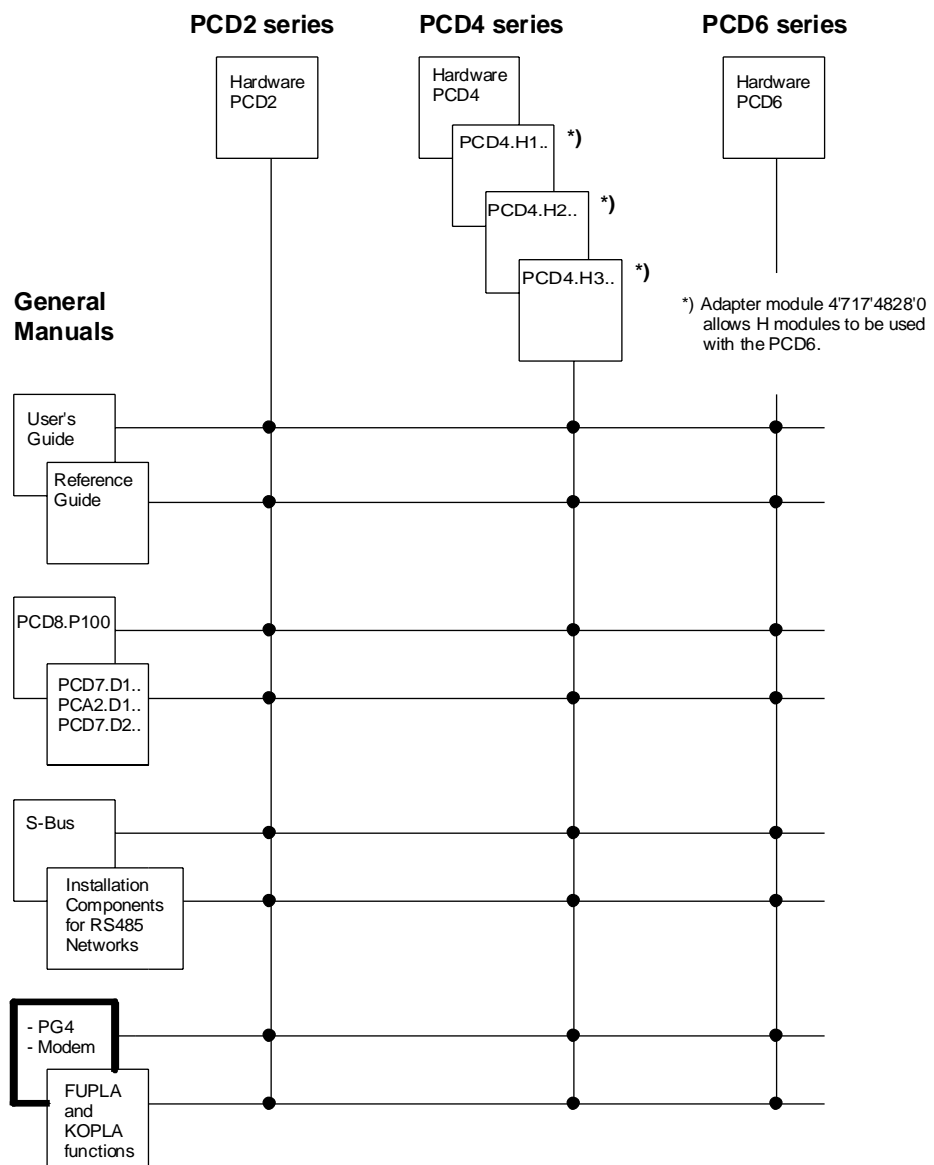
Please note:

A number of detailed manuals are available to aid installation and operation of the SAIA PCD. These are for use by technically qualified staff, who may also have successfully completed one of our "workshops"

To obtain the best performance from your SAIA PCD, closely follow the guidelines for assembly, wiring, programming and commissioning given in these manuals. In this way, you will also become one of the many enthusiastic SAIA PCD users.

If you have any technical suggestions or recommendations for improvements to the manuals, please let us know. A form is provided on the last page of this manual for your comments.

Summary



Reliability and safety of electronic controllers

SAIA AG is a company which devotes the greatest care to the design, development and manufacture of its products:

- state-of-the-art technology
- compliance with standards
- ISO 9001 certification
- international approvals: e.g. Germanischer Lloyd,
- Det Norske Veritas, CE mark ...
- choice of high-quality componentry
- quality control checks at various stages of production
- in-circuit tests
- run-in (burn-in at 85°C for 48h)

Despite every care, the excellent quality which results from this does have its limits. It is therefore necessary, for example, to reckon with the natural failure of components. For this reason SAIA AG provides a guarantee according to the "General terms and conditions of supply".

The plant engineer must in turn also contribute his share to the reliable operation of an installation. He is therefore responsible for ensuring that controller use conforms to the technical data and that no excessive stresses are placed on it, e.g. with regard to temperature ranges, overvoltages and noise fields or mechanical stresses.

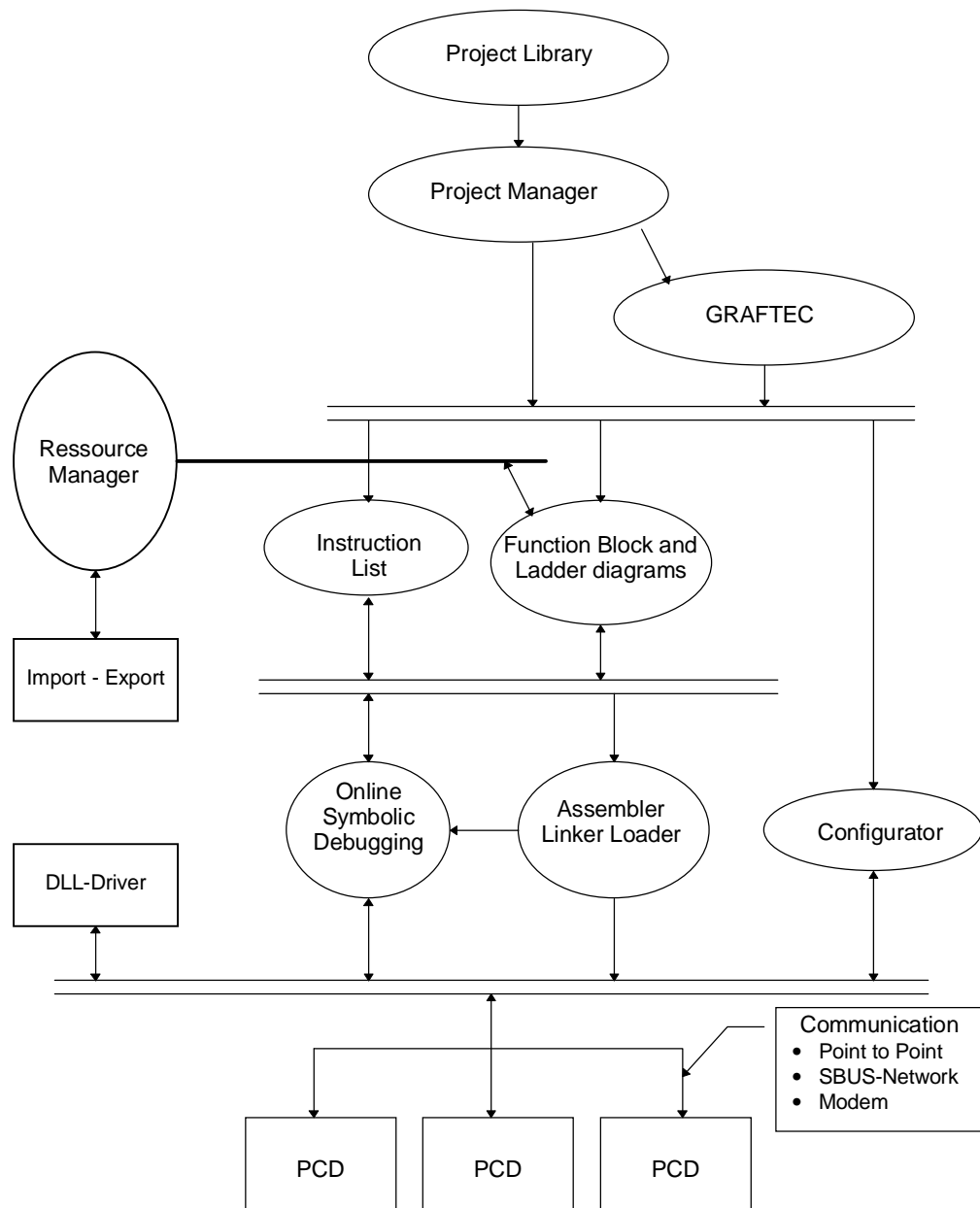
In addition, the plant engineer is also responsible for ensuring that a faulty product in no case leads to personal injury or even death, nor to the damage or destruction of property. The relevant safety regulations should always be observed. Dangerous faults must be recognized by additional measures and any consequences prevented. For example, outputs which are important for safety should lead back to inputs and be monitored from software. Consistent use should be made of the diagnostic elements of the PCD, such as the watchdog, exception organization blocks (XOB) and test or diagnostic instructions.

If all these points are taken into consideration, the SAIA PCD will provide you with a modern, safe programmable controller to control, regulate and monitor your installation with reliability for many years.

1. Introduction, overview

1.1 Brief description of the PG4

Under the working title "PG4", SAIA® is creating a new, modern programming package for the PCD family of process control devices. The package complies with IEC 1131-3 and is based on the MS-WINDOWS graphical user interface of an IBM PC. The PG4 includes the following elements:



The PG4 should make it possible to program a PCD without detailed knowledge of the comprehensive instruction set of the PCD family. The wide range of online functions should also simplify the commissioning of an installation and the detection of errors in a process. The user-friendly "Make" tool lets functions such as compile, assemble, link, load and switch-to-run operate automatically in the background, barely noticed by the user. Resource management (listing the elements used such as inputs, outputs, flags, registers, etc.) is also largely automatic. Comprehensive documentation tools are planned for project documentation.

The PG4's high level of programming comfort does have a drawback: apart from the heavy demands on the PC, it should be noted that the program code is more extensive leading to longer processing or response times for the program than for programs written in IL (instruction list). This should be taken into consideration for large-scale programs and those where timing is critical. In such cases it is logical to use conventional programming technology for the software and multi-processor technology for the hardware.

Explanation of the syntax used in this manual for names and input:

- "xxx": Terms and general titles are written between double inverted commas, e.g. "Project Library", "Call SB" instruction, "PCD2.F510" and have no further significance.
- 'xxx': Executable functions, which are executed with the mouse, are generally written between single inverted commas, e.g.: 'File' - 'New...', 'OK', 'Cancel', etc.
- <xxx>: Input for keyboard entry is generally written between angle brackets, e.g. <10>, <ALT> ("ALT" key), <CR> ("Return" or "Enter" key), <EXAMP1> etc.

1.2 Who works with the PG4 ?

As already mentioned in the introduction, one of the aims is to program a PCD without knowledge of the comprehensive PCD instruction set.

One targetted group of users comprises professionals who occasionally want to program their own small-to-medium system, i.e. without drawing on an experienced programmer, but also without acquiring detailed knowledge of such subjects as the instruction set, program structures and the use of programming tools such as the IL editor, assembler, linker, etc.

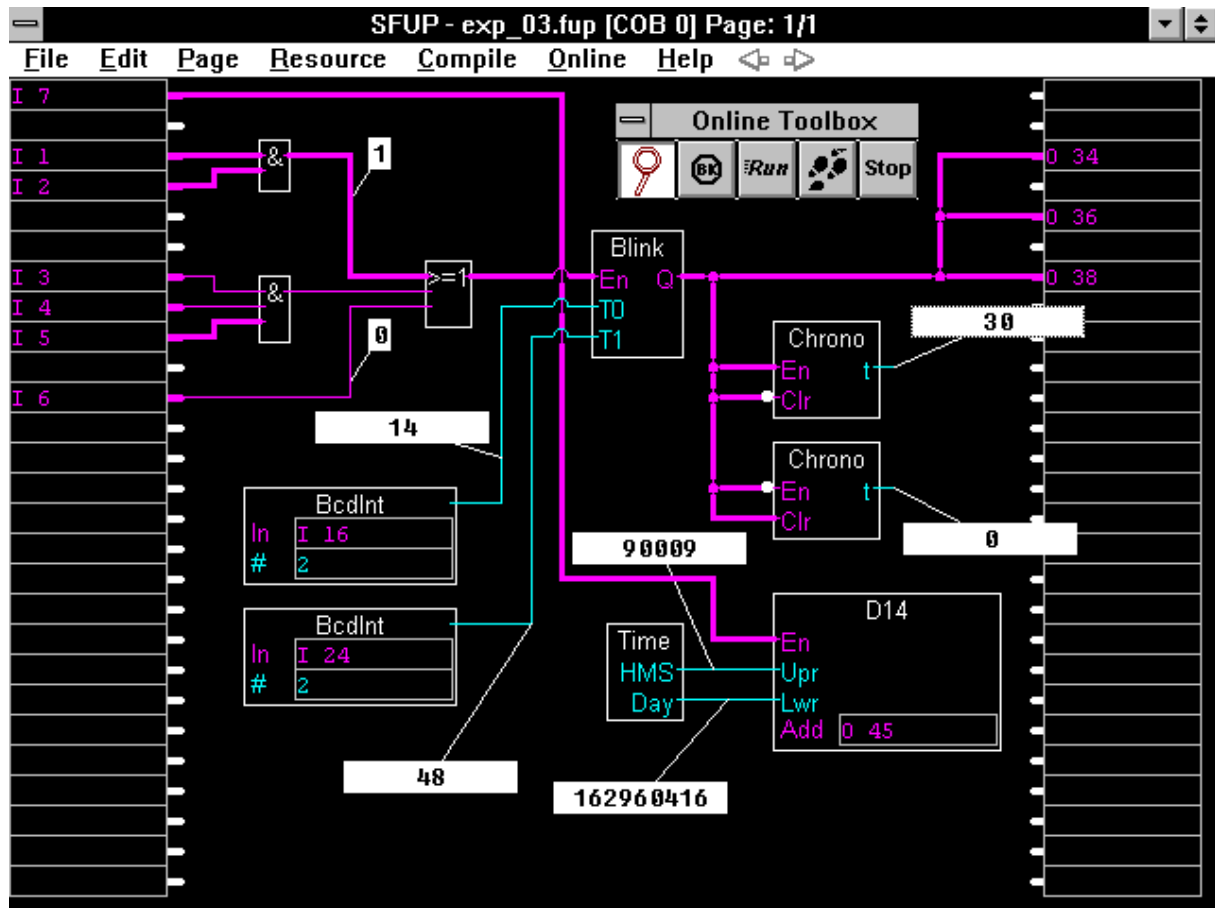
It is therefore possible to draw up a small installation in the widely used ladder diagram (KOPLA), a moderate machine controller in GRAFTEC, combined with the ladder diagram or an air conditioning system in function plan.

Even PCD programming specialists, who have already successfully realized quite large projects with the standard programming tools, can benefit from the convenient online functions and graphical programming interface. It is also possible to add PG4 elements to existing routines. This can be useful when ever important parts of the program must be viewed on-line.

1.3 Individual tools of the PG4 (brief descriptions)

1.3.1 FUPLA (function plan)

FUPLA is an editor for graphical programming with function plan symbols such as AND and OR gates, flip-flops, time delays, counters, mathematical functions, etc.



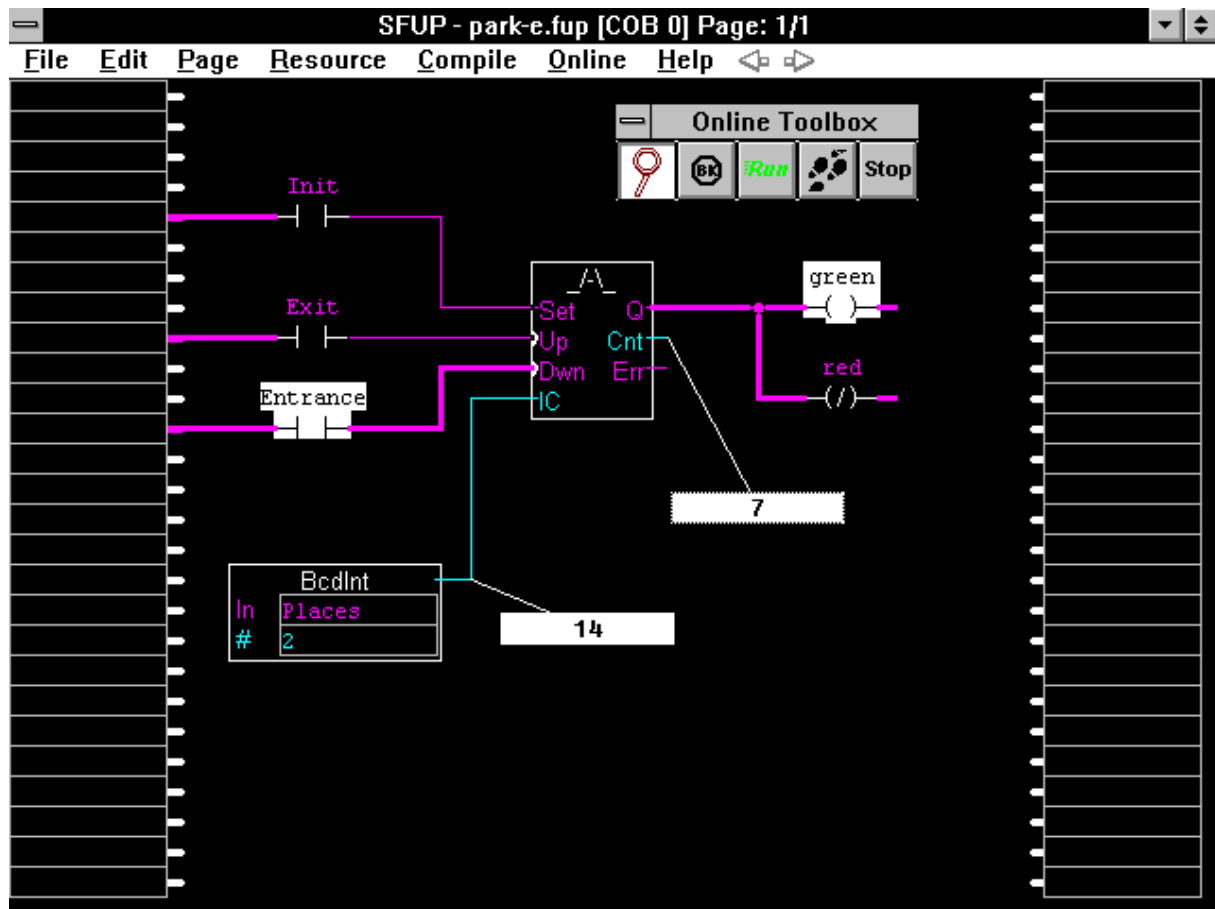
A program can consist of a single FUPLA (on 1 screen page or up to 200), but can also be combined with other, possibly pre-existing routines written in IL (instruction list).

FUPLA includes the tools for writing and loading the program into the PCD ('Make'). The Debugger and the EPROM Programmer can also be called from FUPLA, if required.

See chapter 4 for the detailed description of FUPLA.

1.3.2 KOPLA (contact plan = ladder diagram)

KOPLA is a subsidiary function of FUPLA. KOPLA is called from FUPLA. KOPLA is mainly used to write simple programs with inputs, outputs, counting and timing functions. (KOPLA specialists can also conjure up very complex programs). The individual functions of KOPLA have been standardized. (IEC 1131-3)

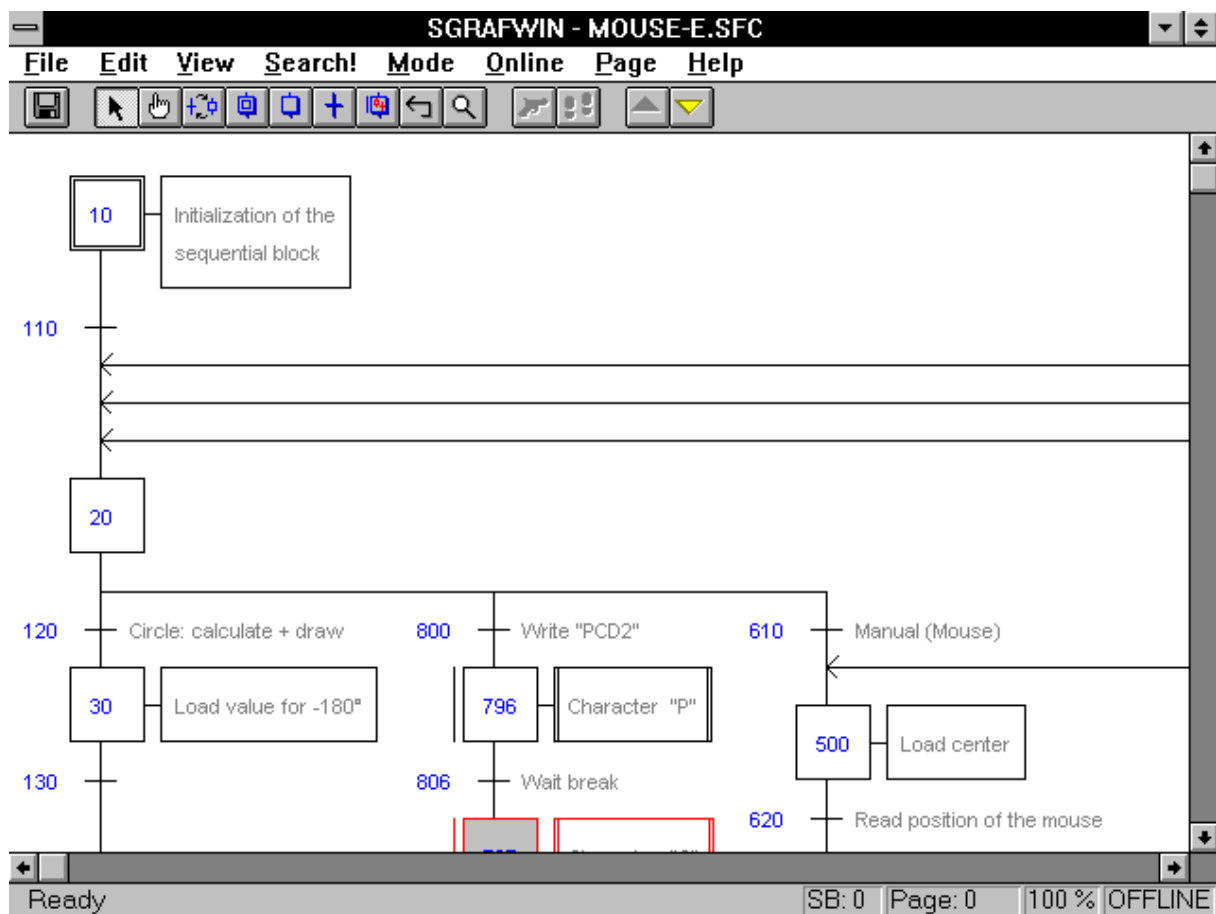


KOPLA also combines very well with GRAFTEC (see next section).

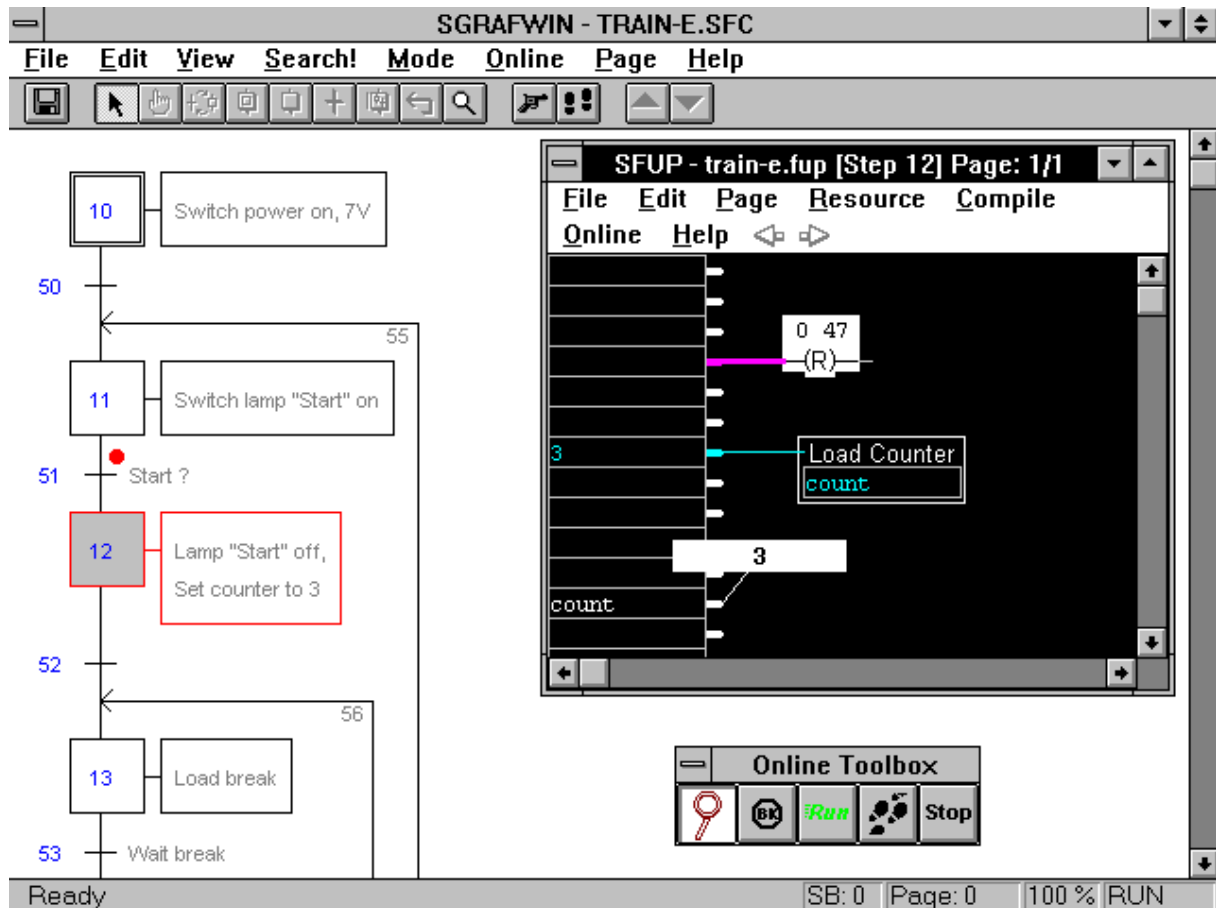
1.3.3 SAIA® GRAFTEC

GRAFTEC is a structural tool for processes which run progressively, such as machine controllers. GRAFTEC is described under the French standard NF-C-03-190.

SAIA® GRAFTEC has proved itself over many years in PCD programming. The previous system program forms the basis of the new PG4 GRAFTEC (SGRAFWIN). It is therefore compatible with older programs edited in GRAFTEC. A new feature is the graphical interface with zoom capabilities and space for longer comments.



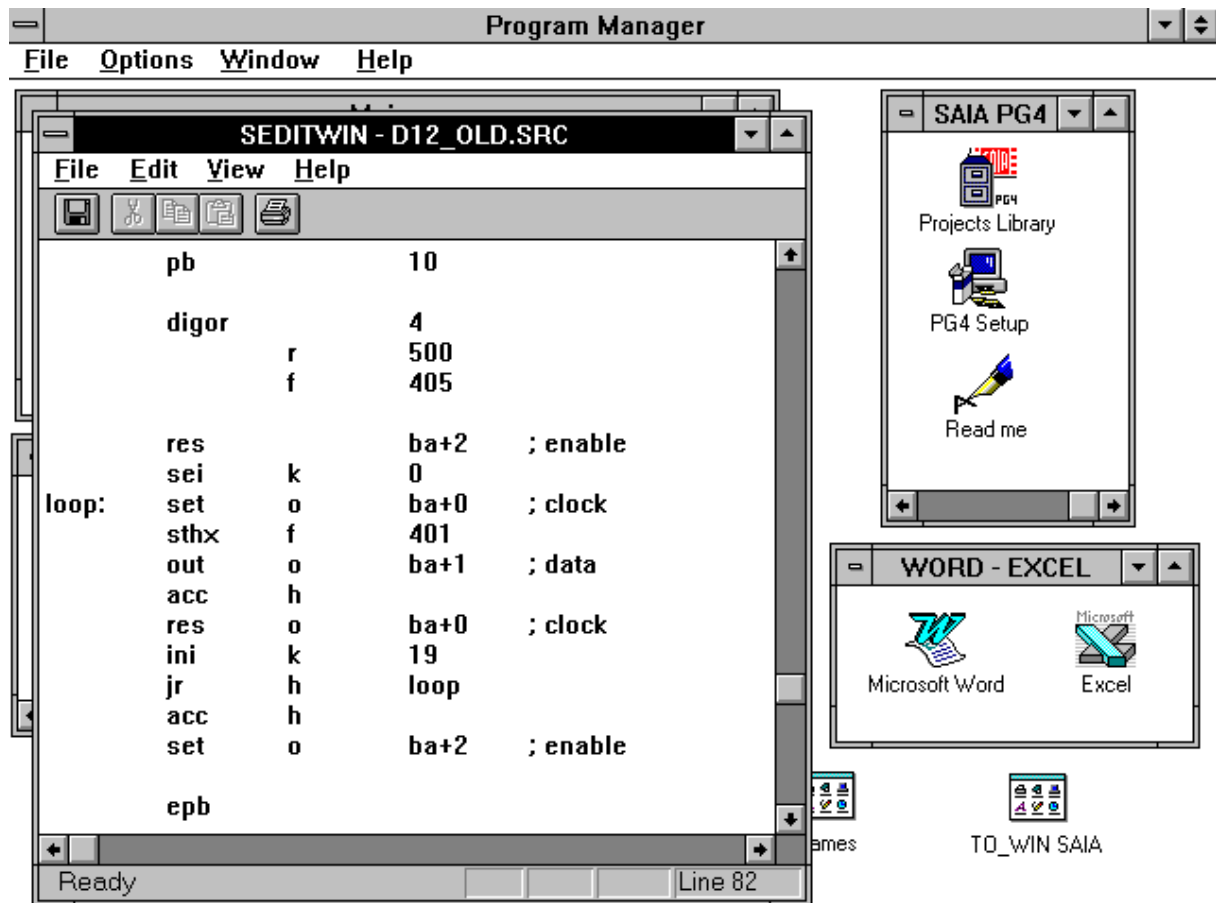
In the PG4 it is preferable to edit the individual steps (ST) and transitions (TR) in KOPLA (or FUPLA). This enables online viewing not only of the GRAFTEC sequence (and therefore of the controller's running sequence) but also of ST and TR contents.



The individual STs and TRs can also be edited in IL (instruction list) using the PG4 editor (SEDTWIN). However, ST and TR contents cannot then be viewed online. For each ST or TR it is possible to select an editor individually.

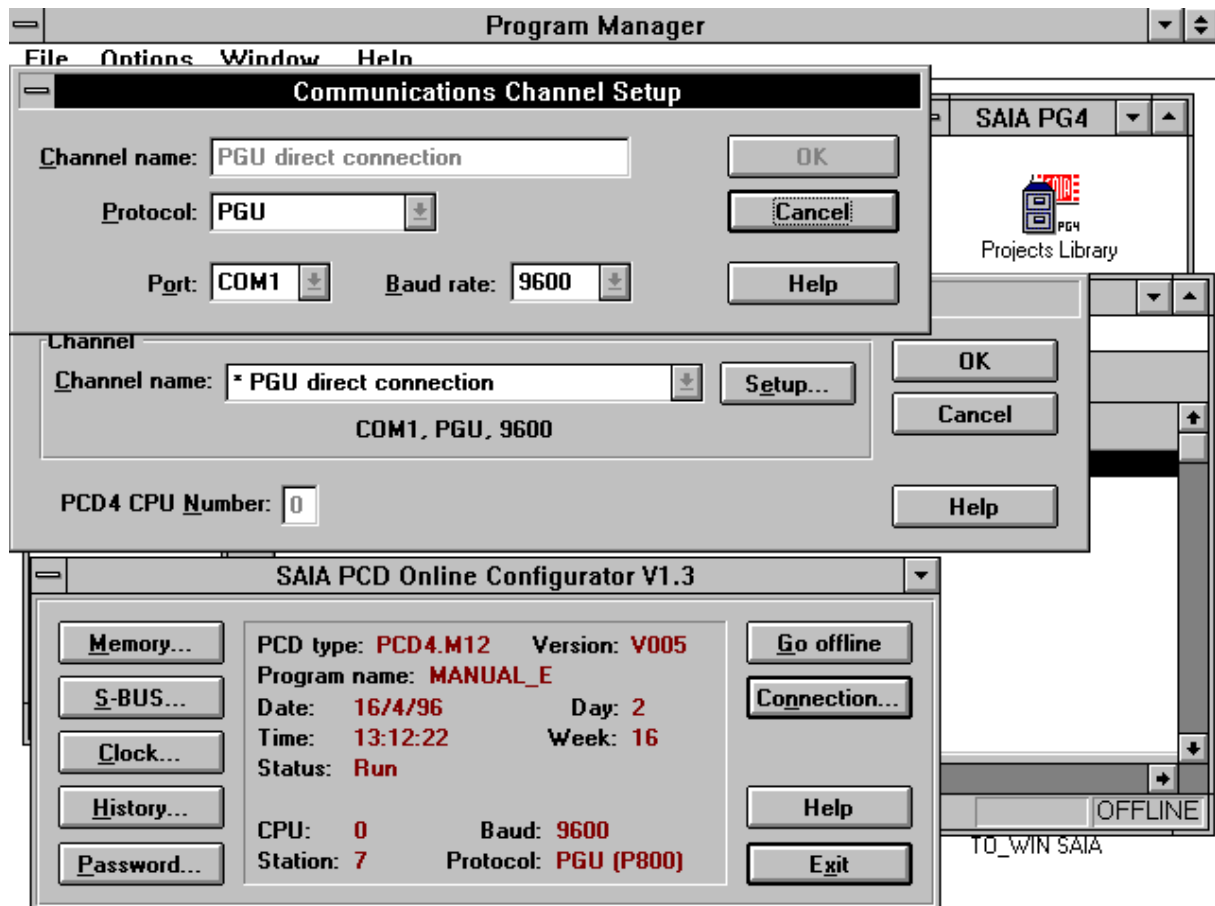
1.3.4 IL editor (IL = Instruction List)

With this editor it is possible to write routines in Instruction List (IL) form without having to exit the PG4. The editor corresponds roughly to the standard WINDOWS editor. At present, the editor still does not have syntax checking for the PCD and also has no instructions for programming the PCD. An IL editor, comparable to the popular "SEDIT" from the PG3, is planned for a later version of the PG4.



1.3.5 Configurator

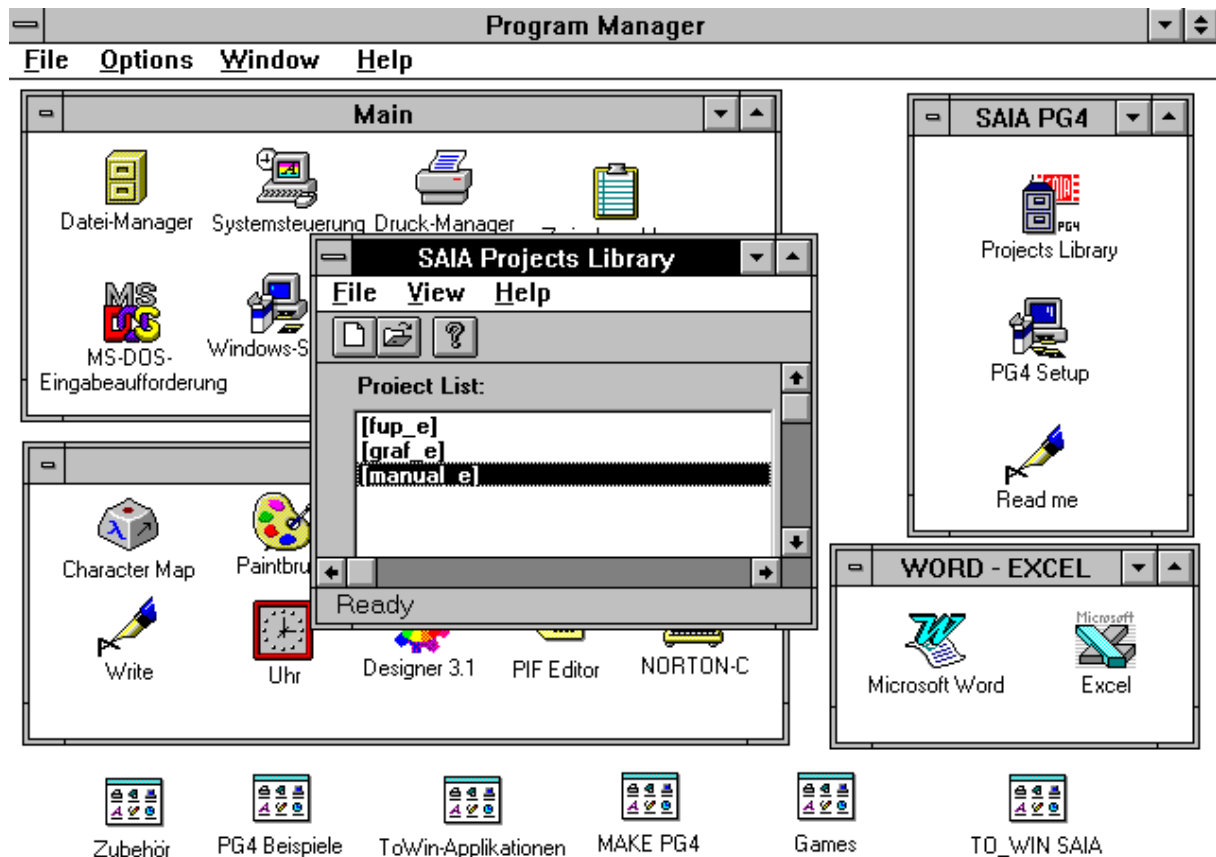
With the configurator it is possible, from the PG4, to see how the connected PCD has been configured. The configuration can also be modified online.



Apart from the configuration of the programming unit (IBM PC), it is also possible to specify or adjust (online) the memory allocation, PCD clock and serial transmission mode between the PCD and the programming unit. The PCD history file can also be inspected and, if necessary, deleted.

1.3.6 Project Library

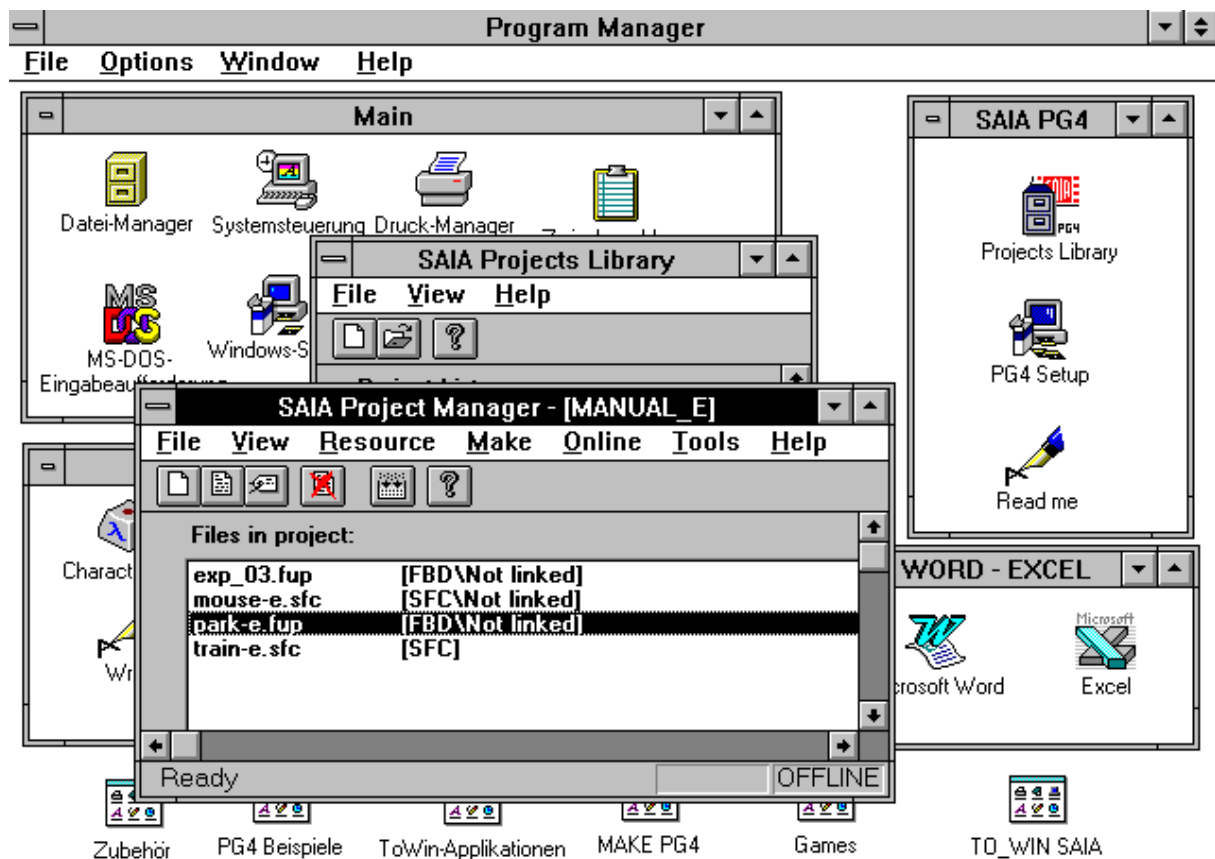
The largest structural unit in the PG4 is the project. In its simplest form, a project consists of a single file. However, normally there are several individual files which must finally be linked to form a loadable file and then loaded into the PCD. Administration of the various projects takes place in the Project Library.



1.3.7 Project Manager

The Project Manager is the central tool in the PG4. The individual files of a project are administered here. New files can be opened, deleted, or imported from other directories or data carriers. The appropriate editor (FBD - SFC - IL) can be selected for each new file. All the resources, i.e. not only the user defined resources, but also the dynamic ones (elements which the compiler uses to store interim results) are administered here for the whole project.

It is from the Project Manager that the individual routines of a project are linked, i.e. it is possible to choose which files to link together and which not to. Loading the program into the PCD and the online options are also selected and enabled here.



From here it is also possible to call the online configurator, the debugger, the PG3 (if available) and to access DOS directly.

Notes :

2. Installation of the PG4

2.1 Personal computer requirements

- Processor: 386 or 486 (100% IBM compatible)
(WINDOWS must run in 386 enhanced mode)
- RAM 1): min. 2 MByte (4 MByte preferred)
(640 KB conventional memory,
1024 KB extended memory)
- Hard disk: min. 10 MByte free space
- Mouse: supported by WINDOWS
- Screen: VGA
- Disk drive: 1, high density, 3½"
- DOS: 3.3 or higher
- WINDOWS: 3.1 or higher
- PG3 Utilities: not necessary for entry

- 1) The maximum size for executable programs under DOS should be around 550 kByte for assembling large files (can be checked with the DOS command "mem"). If there is not enough room in memory, the assembler advises: "Fatal Error 8: Out of memory". In this case, all DOS applications which are not absolutely necessary should be closed.

2.2 Installation

The order number for the PG4 package is PCD8.P4E1. It consists of 4 x 3½" HD diskettes labelled:

- DISTRIBUTION DISK 1/4
- DISTRIBUTION DISK 2/4
- DISTRIBUTION DISK 3/4
- DISTRIBUTION DISK 4/4

Diskette 1/4 contains the installation program "SETUP.EXE" and approx. 20 other install and driver files, some of which are compressed.

Diskettes 2/4 and 3/4 contain more compressed install files.

Diskette 4/4 contains a number of compressed install files plus 3 subdirectories entitled:

- APPSDIR
- EXAMPLES
- FBOX

"APPSDIR" is the application directory and can be used to store files for simultaneous installation during setup, e.g. more recent or corrected .EXE and .DLL files (programs).

Subdirectory "EXAMPLES" contains two typical examples for demonstration purposes: "FUP" and "GRAF" in source format. The comments and symbolic names have been adapted for each language version of the package. To run these examples on a PCD, compile them and execute "Make".

Subdirectory "FBOX" contains the FUPLA functions in the appropriate language: English, German or French.

The "Readme" file is also on this diskette. This file also includes details for installing the package. It is possible to view the file before carrying out installation (Write format).

PG4 actual installation:

Installation on the local hard disk.

It is assumed that your PC meets the requirements mentioned under 2.1 and that WINDOWS has already been installed. If a previous FUPLA or PG4 version has already been installed, deletion of that version is recommended

- Call WINDOWS
- Insert diskette 1/4 (DISTRIBUTION DISK 1/4) in drive A:
- Now either select drive A: from the file manager and double-click on file SETUP.EXE or, from the program manager, select "File" - "Execute...", type in

<A:SETUP.EXE>,

click on "OK" or type <CR>.

- The following message is displayed for half a minute

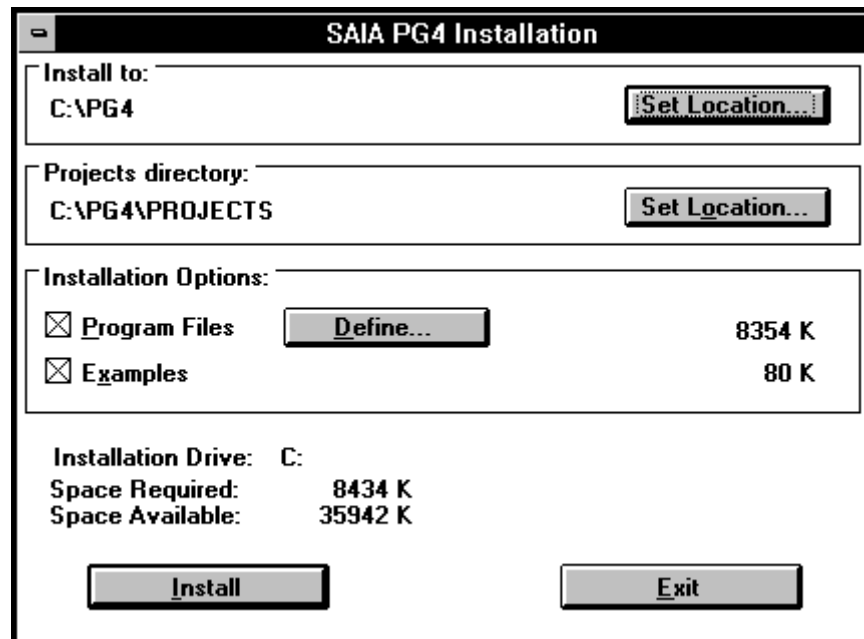
Initializing setup...

- After which the following window appears:



- Click on "Continue"

- The next window appears:



An assumption is made that the PG4 is stored in a "\PG4" directory on drive C:. If you would prefer to install it differently, you can specify a different drive and directory name here.

The amount of memory required and the remaining space available can also be seen.

If everything is ready, click on 'Install'.

The PG4 is being installed. At intervals of a few minutes, you will be asked to insert diskettes 2, 3 and 4. On each occasion, this prompt must be acknowledged by clicking on 'OK' or with <CR>.

After a few more minutes the following question appears:

"Are there extra PG4 disks to install?"

Normally you have no extra disks to install. The answer to this question is therefore "NO".

If, however, you do want to install extra disks, answer 'YES'. The extra disk should no be inserted and acknowledged with 'OK' or <CR>. The extra PG4 disk is loaded. After a while the same question appears as before. This time the answer to give is 'NO', as the extra disk has now been loaded.

- A new question appears:

"Do you want to read the release notes ?"(i.e. the description of the package)

"YES" or "NO". This is left up to you.

- The PG4 has now been installed and the "SAIA PG4" program group has been generated.



- The order of individual icons, changing names, removing and adding programs are all carried out according to the rules of WINDOWS and no further explanation is given here.

Installation in a network

Installation should be carried out in stages. First the whole package is installed on the server. Then the "shared" installation is possible of every IBM PC connected to the server.

Installation of the server station is initiated with

```
A:SETUP.EXE /A
```

The rest is similar to installation on the local hard disk.

So that the package is capable of running on one or more private (client) stations, the "shared" installation takes place in the client stations, e.g. with

```
P:\PG4\SETUP.EXE /N
```

ensuring that the drive and path match your installation. The "/N" is essential. Only the "PROJECTS" directory and the three icons are installed on the local PC. The PG4 runs on the server, which means that the latter must be switched on to working with the PG4.

It is also possible to carry out local installation from the server. In this case the whole package is installed from the server onto the individual PC, and it is possible to work independently of the server.

2.3 Content of the "PG4" directory

What the individual programs are:

- Sfup: FUPLA/KOPLA
- SgrafWin: SAIA[®]-GRAFTEC under WINDOWS
- SeditWin: Simple IL editor under WINDOWS
- SbugWin: Debugger as in PG3 (DOS). The debugger is included in the PG4 package. It is not part of the PG3.
- SconfWin: Configurator
- Smak: Make utility
- Sasm: Assembler
- Slink: Linker
- SPG4: Project library, output program of PG4
- Sprojman: Project manager
- ReadMe: Brief description of the present version
- SRM: Resource manager

Organization of the PG4 directory:

Executable programs (.EXE, etc.) are located in the top-level directory. It can be seen to contain not only the new WINDOWS programs, but also the assembler (SASM), linker (SLINK) and debugger (SBUG).

A further subdirectory "FBOX" has been installed. This subdirectory is important for the user, since it contains all the FUPLA functions.

- SFUPxxx.LIB: The macros for each individual function
- SFUPxxx.DEF: Definitions for the construction of individual function boxes
- SFUPxxx.IDX: Names of the function families and of individual functions plus all the labels and texts for the adjust windows. This file is also available in German.
- SFUPxxx.HLP: All help texts. This file is also available in German and French.

All .IDX and .HLP files are available in English, German or French.

2.4 Retrospective installation of individual function families

To obtain another language version (e.g German, French, English, Italian, ..) the .IDX and .HLP files must be changed. There are various ways of doing this:

- with the "PG4 Setup" WINDOWS program, which was installed during setup, by having the new files ready for loading on extra diskette no. 3, or
- directly from DOS, by loading the diskette with the desired language version of the .IDX and .HLP files into the "FBOX" subdirectory. Before doing this, the original (English) files should be saved first, for possible reinstallation at a later date.

It is also possible to store various language versions in various subdirectories, allowing the use of batch files for loading one or other language version into the "FBOX" subdirectory and then calling them.

2.5 Delete installation

If it is necessary to delete the PG4 from a PC, the whole of directory

C:\PG4

should be removed, together with all subdirectories.

In addition, the following two files should be deleted from the WINDOWS directory:

C:\WINDOWS\SPG4.INI and
C:\WINDOWS\SPG4MODM.INI

3. Project Library, Project Manager and Configurator

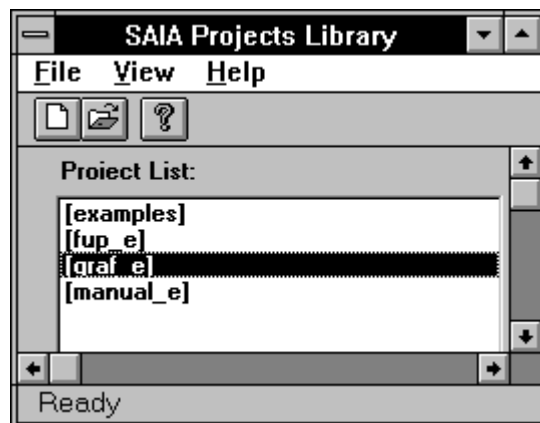
3.1 Running the Project Library



In the PG4, loadable PCD files are called projects. In its simplest form, a project consists of a single file, but normally several individual files are ultimately linked to create a loadable file and loaded into the PCD. The various projects are administered in the Project Library.

A project consists of the program name and other information relevant for an application. 1 project supports 1 PCD system.

After calling the "Project Library" from WINDOWS Program Manager the following window appears:

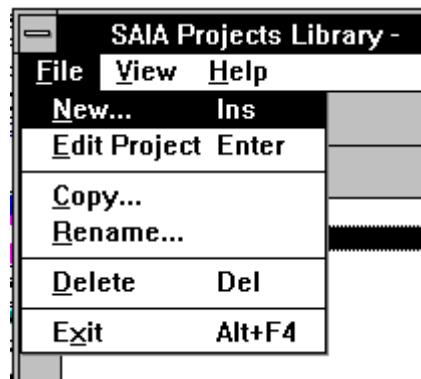


All existing projects are listed and can be opened with a double click, i.e. the 'Project Manager' is called, from which all the separate files of a project can be viewed and worked on (see next section).

The menu bar contains 3 submenus: 'File', 'View' and 'Help'.

3.1.1 'File' submenu

The 'File' submenu appears as follows:



'New...': After clicking on 'New...' the following window appears:



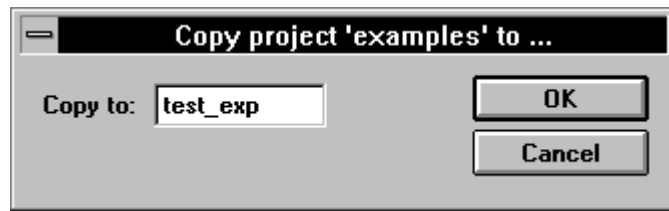
A new project can be defined. The file name can have a max. length of 8 digits. The following characters are accepted:

- 'a' - 'z'
- 'A' - 'Z'
- '0' - '9'
- '_', '-'
- 'ä', 'Ä', 'å', 'Å', 'ö', 'Ö', 'ç', 'ü', 'Ü', 'é', 'É', 'ñ', 'Ñ'.

After 'OK' a new project directory is created. 'Cancel' can be used to abort.

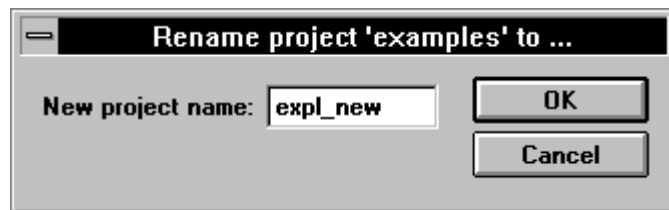
'Edit Project': The Project Manager is called (see next section).

'Copy...': Copies or duplicates the current project indicated in the header bar to a newly defined project. Afterwards, the original project still remains.



After 'OK' there would be two identical projects named "examples" and "Beispiel".

'Rename...': Any project can be renamed.



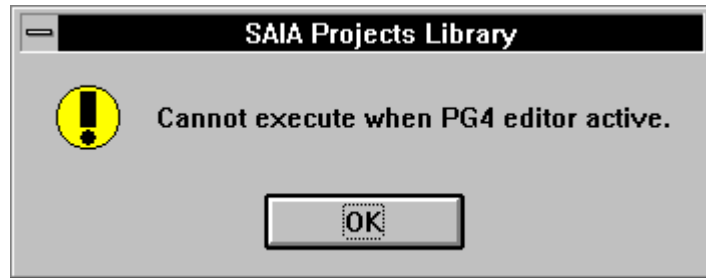
After 'OK' the original project "Beispiel" becomes "bsp_neu".

'Delete': The current project, i.e. the one selected in the project list, can be deleted. This deletes all the files belonging to the project together with the file names.

After clicking on 'Delete...' the following window appears:



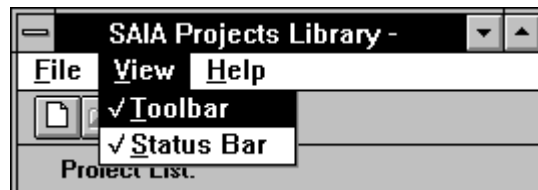
If 'Delete' is executed when an editor or the project manager is still open, the following text is displayed:



All editors and the project manager should be closed before 'Delete' can be executed.

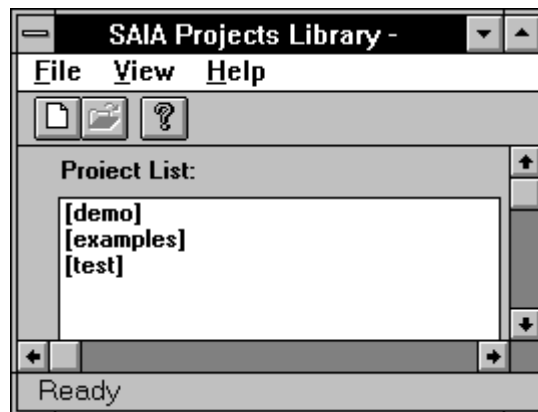
'Exit': The project library, or PG4, is exited.

3.1.2 'View' submenu

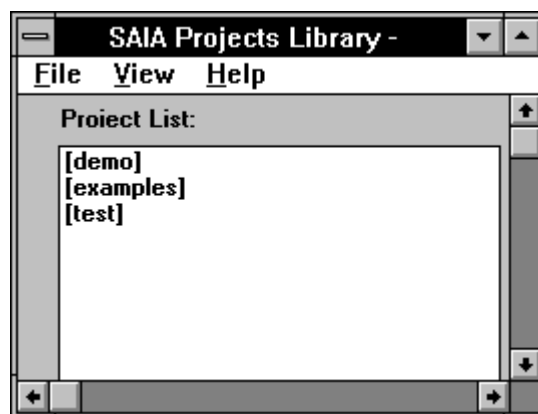


'Toolbar': Enables the toolbar to be displayed or cleared.

'Status Bar': Enables the status bar to be displayed or cleared.

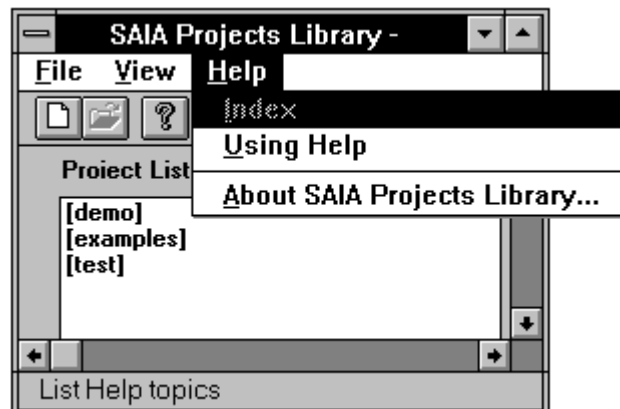


Appearance with toolbar and status bar.



Appearance without toolbar and status bar.

3.1.3 'Help' submenu



'Index': Calls the help file. <F1> has same effect.

At present, not yet available.

'Using Help': Calls instructions for the WINDOWS help files.

'About SAIA Project Library...':

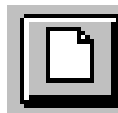
Calls the following window with information on the installed version of the PG4.



3.1.4 Toolbar



Clicking the mouse on one of the tool buttons is a convenient way of selecting the most frequently used functions, without opening a menu every time.



Opens a new file.
Identical to menu 'File' - 'New...!'

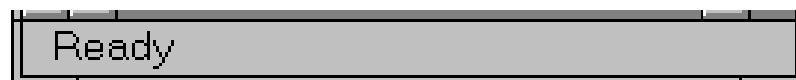


Opens the current project.
Identical to menu 'File' - 'Edit Project'.



Calls the window with information on the installed version of the PG4.
Identical to menu 'Help' - 'About SAIA...'

3.1.5 Status bar



The status bar on the bottom margin of the window displays the current operating status or readiness of the project library.

If the mouse button is held down on a submenu, the operating status of the submenu is displayed. The same applies for toolbar symbols. In this case, if the function is not to be executed, the tool button must be exited while holding the mouse button down.

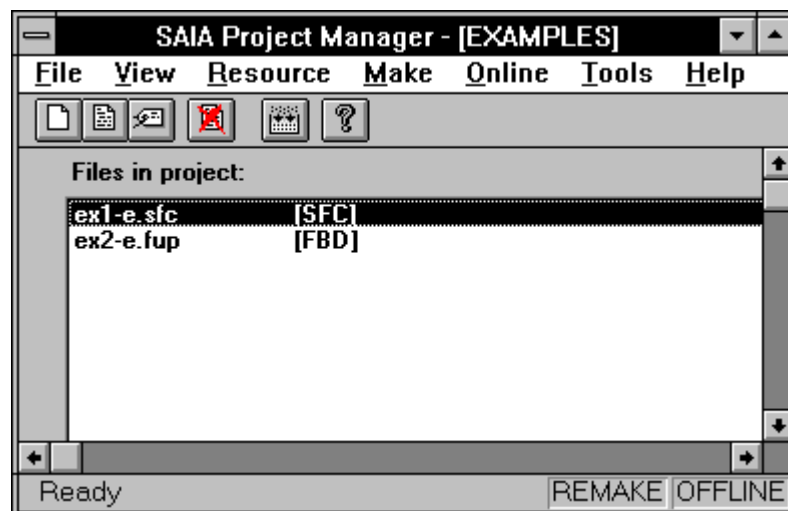
Notes :

3.2 Running the Project Manager

3.2.1 Main menu of Project Manager



The Project Manager is the central tool in the PG4. The individual files of a project are administered here. New files can be opened, deleted, or imported from other directories or data carriers. The appropriate editor (FBD - SFC - IL) can be selected for each new file. All the resources, i.e. not only the user defined resources, but also the dynamic ones (elements which the compiler uses to store interim results) are administered here for the whole project.



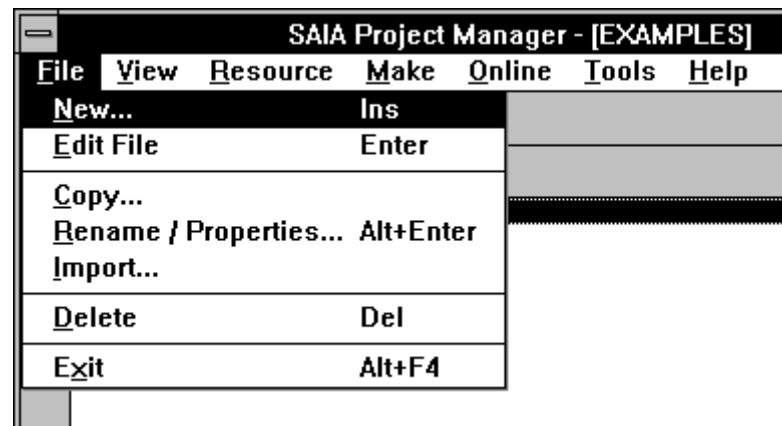
The project manager is called from the project library by double clicking on a new or existing project, or by clicking on the toolbar button to open the current project.

All existing files are listed and can be opened by double clicking. The file extension indicates the editor used to create an existing file:

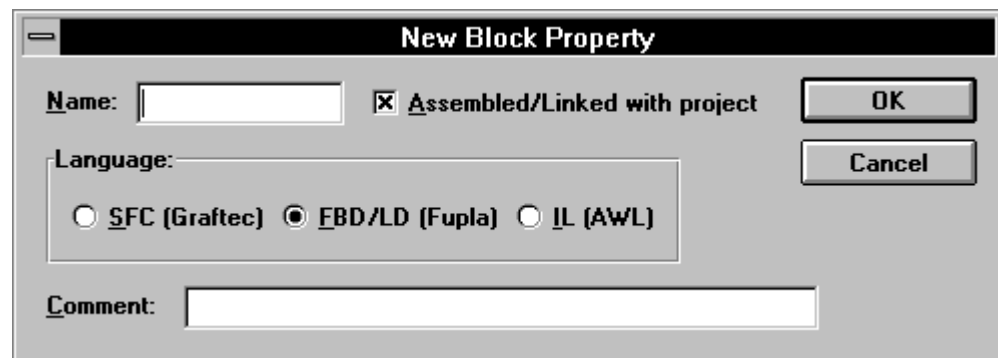
- filename.FUP: FUPLA - Function Block Diagram (FBD)
- filename.SFC: GRAFTEC - Sequential Flow Chart (SFC)
- filename.SRC: Instruction List (IL) (SRC = source)

3.2.2 'File' submenu

The 'File' submenu appears as follows:



'New...': To open a new file.
After clicking on 'New...' the following window is displayed:



The name of the new file to be created is indicated at "Name". The file name can have a maximum length of 8 digits. The following characters are accepted:

- 'a' - 'z'
- 'A' - 'Z'
- '0' - '9'
- '_, '-'
- 'ä', 'Ä', 'à', 'À', 'ö', 'Ö', 'ç', 'ü', 'Ü', 'é', 'É', 'ñ', 'Ñ'.

An editor is selected by clicking the mouse on the appropriate switching spot:

SFC - FBD/LD - IL

At "Assembled/Linked with project" it is possible to choose whether, after 'Make', this new file will be linked with the other files in the same project or not (x = will be linked).

At "Comment" it is possible to enter a comment for the file up to a maximum of 40 characters.

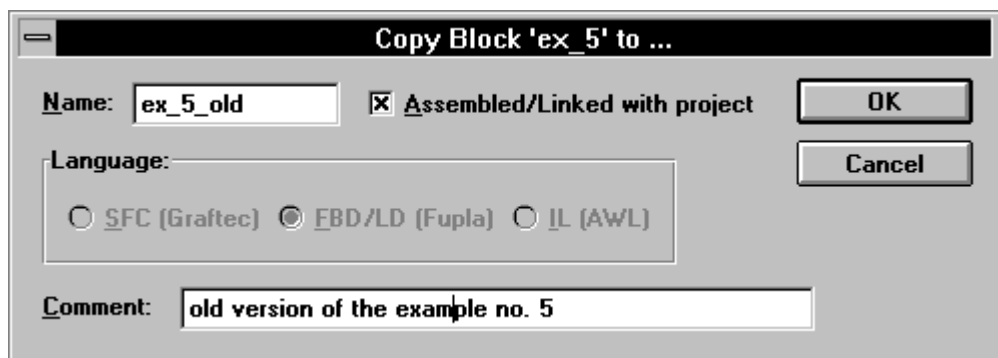
After 'OK' the new file is opened. This file is entered in the project manager's file list, together with any comment.

'Cancel' can be used to abort.

'Edit File': Calls the current file editor. If the file concerned is a new one, it can be edited. If it is an existing file, it can be viewed and modified.

Actual editing is explained fully in the chapters on individual editors.

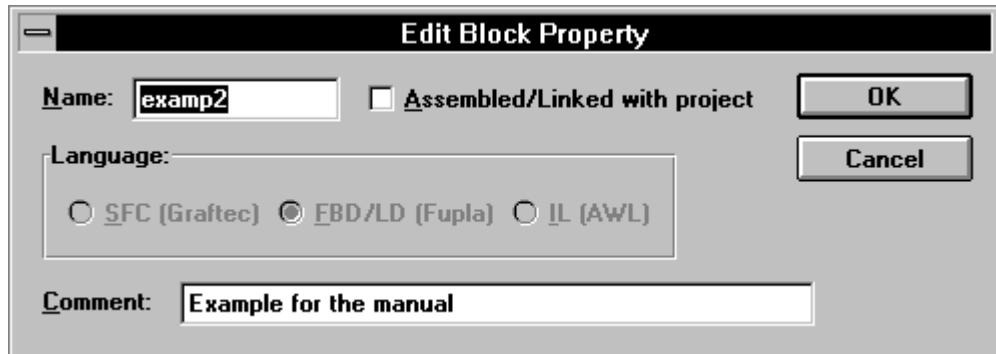
'Copy...': Copies or duplicates the current file, as specified in the header bar, to a new file, as named in the following window. Afterwards, the original file still remains.



In this case, file "example2" is copied to the new, additional file "beispiel". Afterwards, the editor, any comment and the "Assembled/Linked" option still remain.

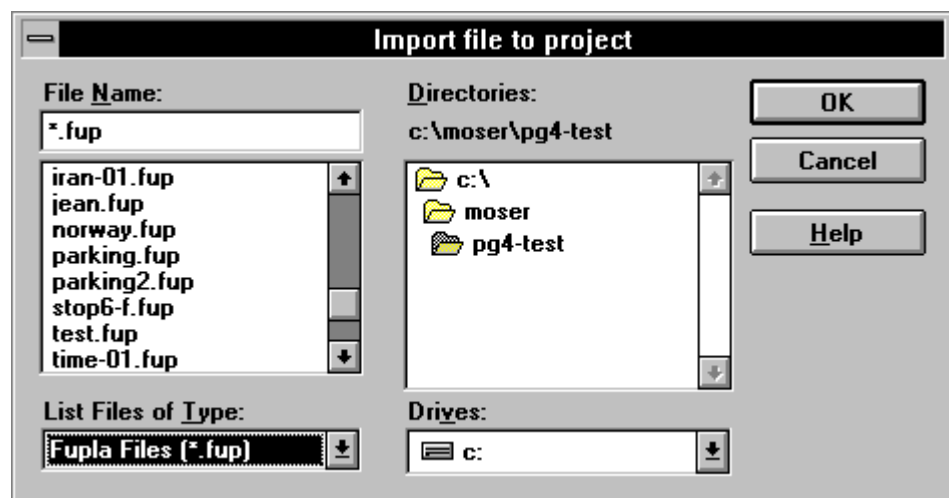
'Rename/Properties...':

The current file can be renamed and its properties, e.g. the comment, the "Assembled/Linked" option, can be modified.

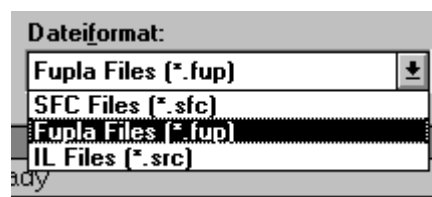


'Import...': With this submenu, files can be imported from other projects, directories or data carriers (diskettes, hard disk ...) into the current project.

This is particularly useful when old files are to be imported into new projects.



Files for importing must have the format ".fup", ".sfc" or ".src". The format to search for in directories can be specified in the "File format" window:

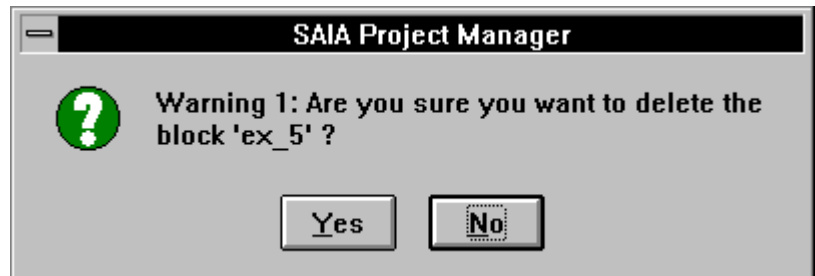


If "*" is specified, all files are displayed and can also be imported. Files with other formats than those mentioned will still be copied, but will not appear in the project manager.

For example, if old GRAFTEC files which still have the format ".SRC" are to be inserted, they must be renamed from ".SRC" to ".SFC" **before** their inclusion with Import. Otherwise an ".SRC" file is automatically interpreted as an "IL" file and further change is impossible.

'Delete':

The file marked in the project manager window can be deleted. The following window appears immediately afterwards:



After "Yes", all files belonging to 'examp2' are removed from the project manager's directory.

'Exit':

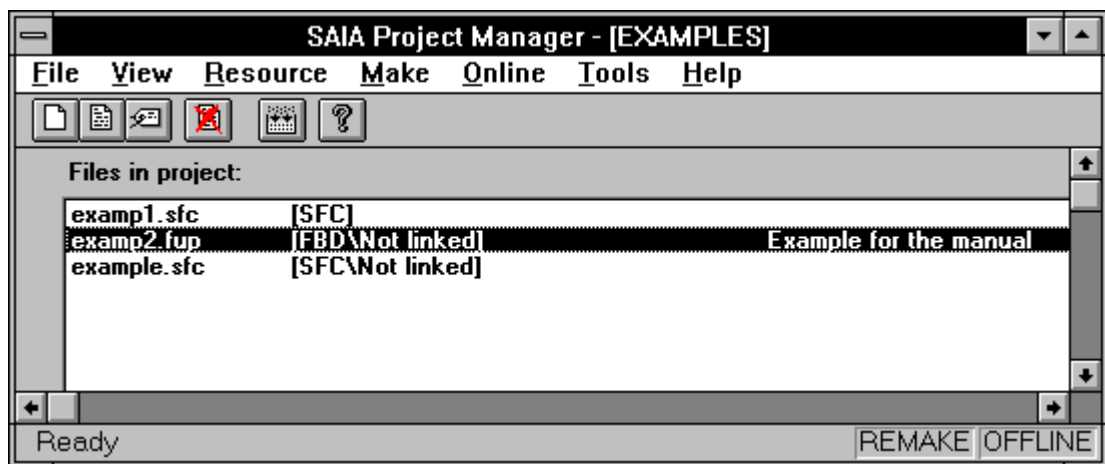
Exits the project manager into the project library.

3.2.3 'View' submenu

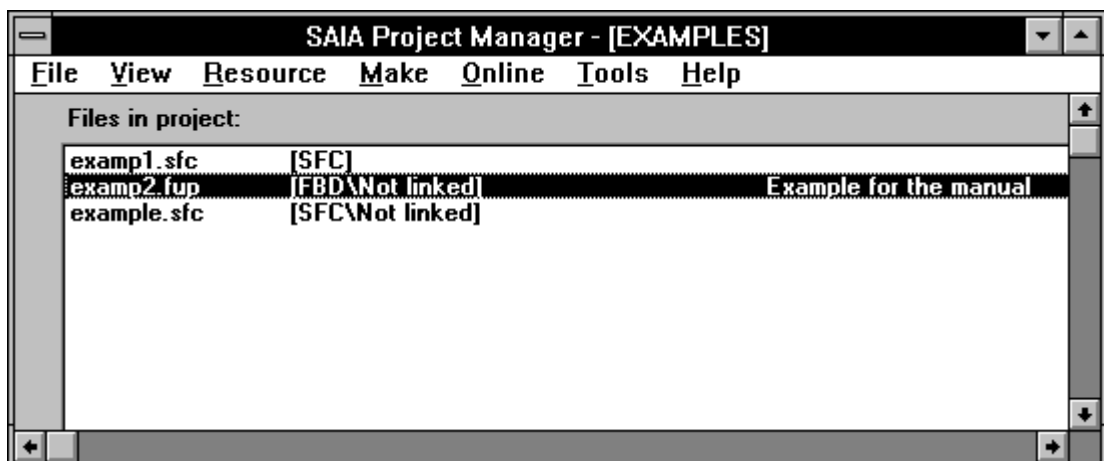


'Toolbar': Enables the toolbar to be displayed or cleared.

'Status Bar': Enables the status bar to be displayed or cleared.

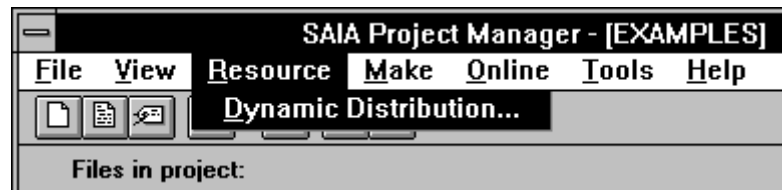


Appearance with toolbar and status bar



Appearance without toolbar and status bar

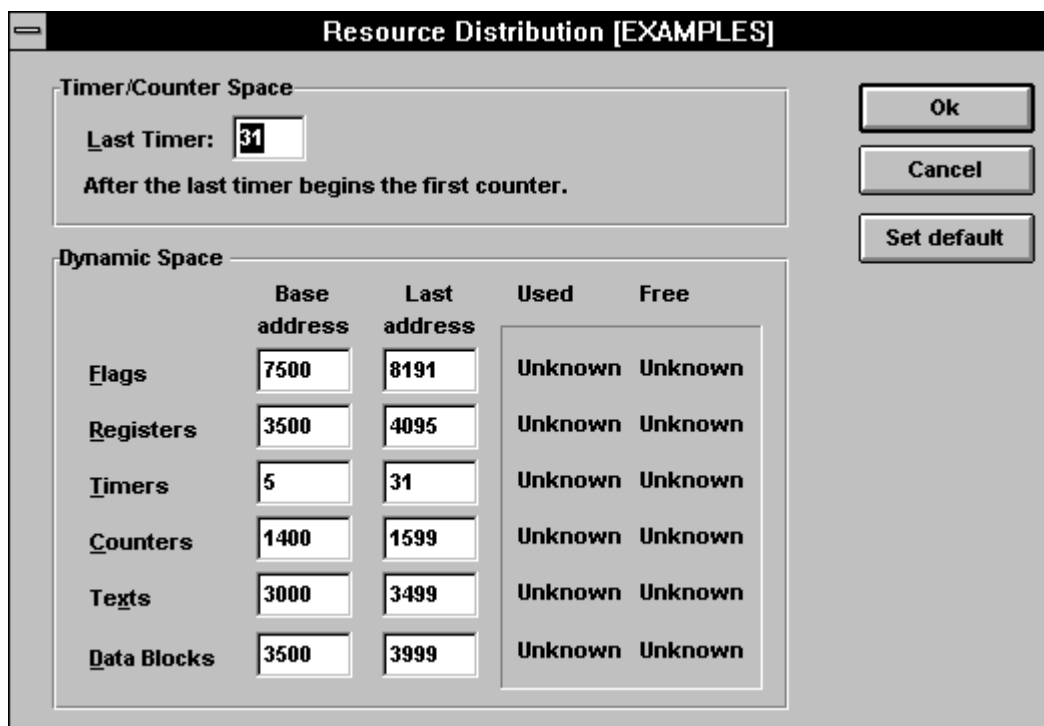
3.2.4 'Resource' submenu



'Dynamic Distribution': (Distribution of dynamic resources)

On the one hand, this submenu assigns the distribution of timers / counters in the PCD. The total available T/Cs are 1600. Addresses from 0 up to the defined address are timer locations. The remainder up to address 1599 are counters. The standard value is 31, i.e. there are 32 timers located at addresses 0 to 31. If a project requires the provision of, for example, 100 timers, <99> should be specified at "Last Timer". Do not reserve many more timers than are actually required. (Too many timers place an unnecessary load on the CPU).

On the other hand, this menu defines the dynamic resources for the entire project, which the compiler can use to store provisional results. With the "Set default" option, a standard distribution is made is suitable for medium sized projects. This distribution can be adjusted at any time.



In a newly defined project, Unknown appears under "Used" and "Free". This is because the distribution is only carried out after the first "Make" (i.e. after assembly and linking).

After the successful execution of Make, the window appears roughly as follows:

Resource Distribution [EXAMPLES]

Timer/Counter Space

Last Timer:

After the last timer begins the first counter.

Dynamic Space

	Base address	Last address	Used	Free
Flags	<input type="text" value="7500"/>	<input type="text" value="8191"/>	28	664
Registers	<input type="text" value="3500"/>	<input type="text" value="4095"/>	46	550
Timers	<input type="text" value="5"/>	<input type="text" value="31"/>	2	25
Counters	<input type="text" value="1400"/>	<input type="text" value="1599"/>	1	199
Texts	<input type="text" value="3000"/>	<input type="text" value="3499"/>	0	500
Data Blocks	<input type="text" value="3500"/>	<input type="text" value="3999"/>	0	500

Buttons: Ok, Cancel, Set default

A detailed description of the use of absolute and dynamic resources, and of program compiling, follows in Chapters 4 and 5.

3.2.5 'Make' submenu

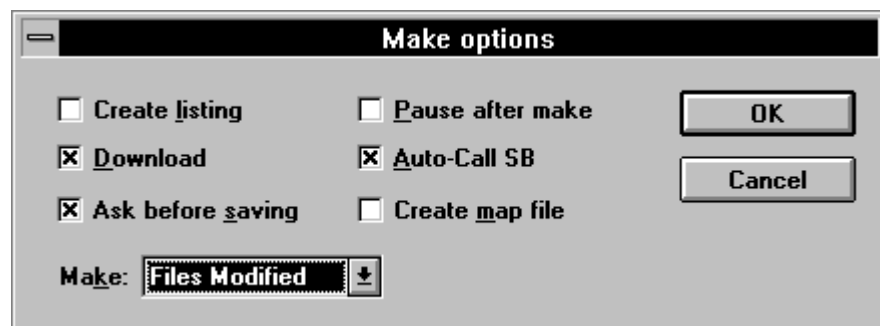


This important tool is explained in detail with examples in Chapters 4 and 5. The individual menu items are only dealt with in general here.

'Make': This tool is used to link all files listed in the project manager which have not already been marked as "Not linked", i.e. the loadable PCD file is generated (filename.pcd). Normally the file created is also loaded, the CPU is put into Run and the PG4 is switched online.

'Make Options...':

The various options can be set in advance and stored.



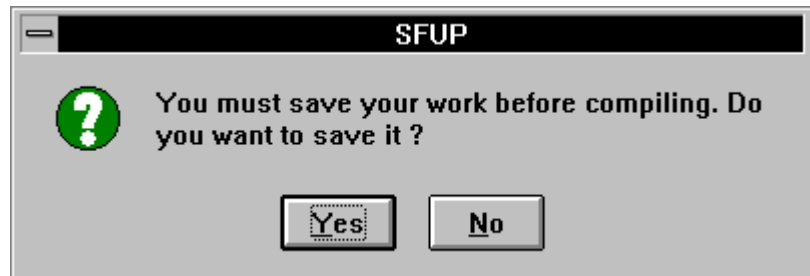
Options enabled: x (with mouse click)

Create listing: During assembly a .LST file is generated.

Download: The finished .PCD file is downloaded automatically to the PCD without further question.

See also 'Download Options...' in the 'Online' submenu.

Ask before saving: If this option is enabled, a question is displayed before any 'Make' asking whether the modified file should be saved.



If the option is not enabled, saving is automatic.

Pause after make: A pause is enabled after assembly and linking. The user is prompted to press any key to continue. In this way, all linker messages can be read.

If this option is not selected, successful linking is directly followed by switching to the "Downloader" and, depending on the choice of "Download Options", the program is downloaded and put into Run.

Auto-Call SB: If all or part of the program is edited in GRAFTEC, enabling this option automatically inserts a "Call SB" instruction from a COB.

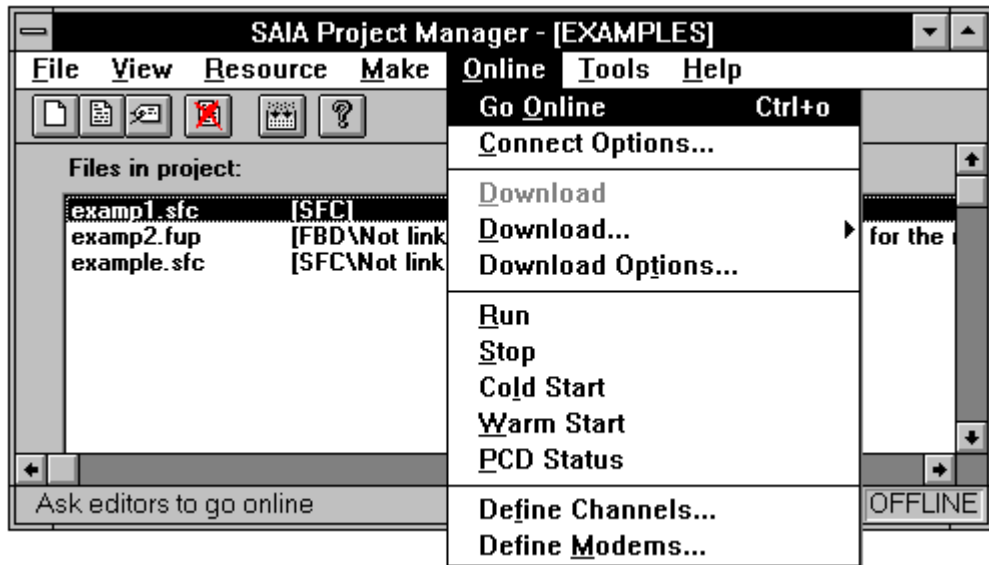
Create map file: If this option is enabled, a .MAP file is created during linking. (ASCII file with information on program length, etc.)

Make:

It is possible to select whether 'Make' applies to all files or only to modified files.



3.2.6 'Online' submenu



'Go Online/Offline':

A program running in the PCD can be switched online with this option. The status of all binary elements and all values in integer or floating point format can be viewed online in the function block diagram (FUPLA).

If the program in the PCD and the file in the PG4 no longer match each other, i.e. if there has been modification of the FUPLA, one of the following messages appears:

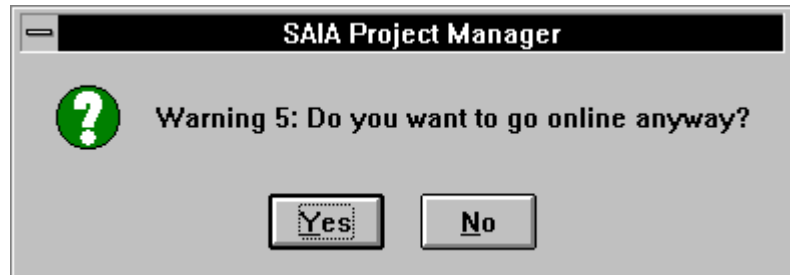


... you must recompile your program.



... you must reload the file.

If parameters in an adjust window have been modified, it is possible to go online after recompiling, without repeating 'Make'. However, correct operation is not then guaranteed. The following window is displayed for such cases:



With large programs this can be useful for trying out a parameter modification in the adjust window. It should however be used with caution.

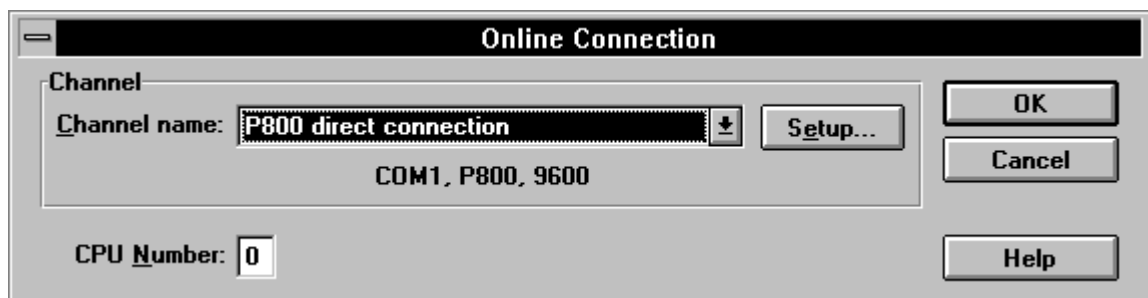
'Connect Options...':

This submenu enables the online connection between the PG4 (IBM PC) and the PCD to be selected.

Three standard modes are available:

- P800 direct connection
- S-BUS direct connection
- S-BUS dial-up modem connection

Other modes are freely definable.



Example: window for "P800 direct connection"

A more thorough description is given in section 3.3.2.

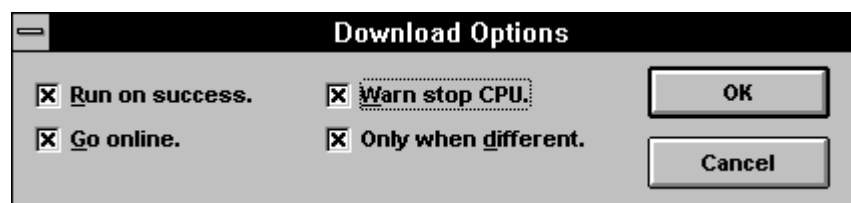
'Download' Used for downloading a PCD program which has been successfully prepared with 'Make'.

'Download...': The following selection box appears:



It is possible to choose whether program code, user texts or extension memory data are to be downloaded.

'Download Options...':



Options enabled: x (with mouse click)

The options are:

- Run on success: The CPU goes into Run after a successful download.
- Go online: The PG4 automatically goes online in Run
- Warn stop CPU: Before downloading, a warning is given that all CPUs will stop.
- Only when different: Downloading only occurs if the project to be downloaded differs from the existing one.

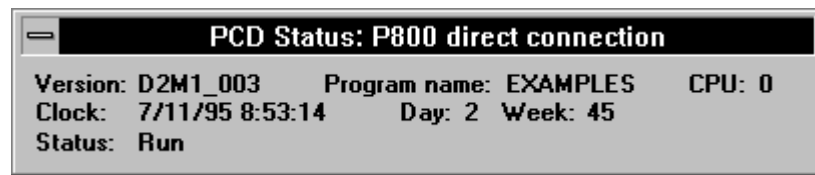
'Run': Switches the connected PCD into Run.

'Stop': Switches the connected PCD into Stop.

'Cold Start': Executes a cold start (with initialization routine)

'Warm Start': Executes a warm start (without initialization routine).

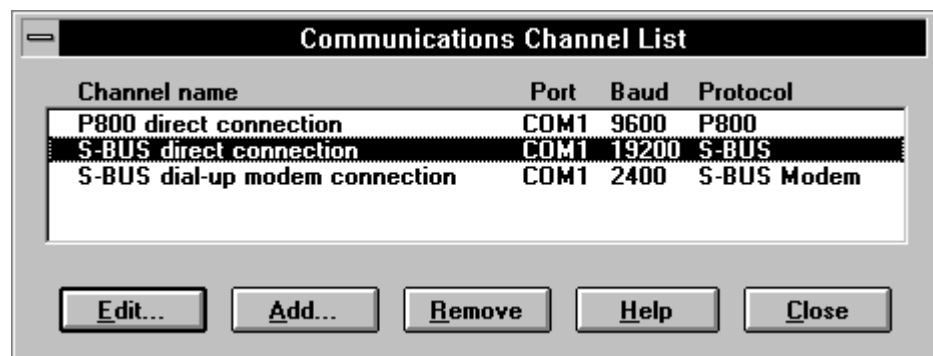
'PCD Status': The following online window appears:



It shows: CPU version; firmware version; name of program loaded; CPU number (PCD4/6); clock date and time; day of week (1 = Monday); week number; and CPU status (Run, Stop, Halt, Conditional Run).

'Define Channels...':

The following window appears:



The communications channel can be selected.

More can be found on this subject in section 3.3 "Running the configurator".

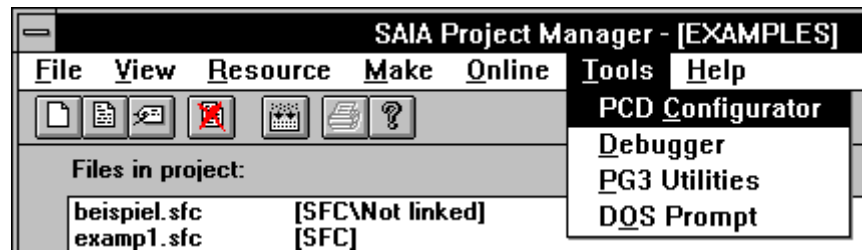
'Define Modems...':

The following window appears:



This submenu allows the user to describe the specifications of his modem, which are then stored in the communications library and used for any connection with that modem. This tool is not normally utilized, since specifications are already listed for the most commonly used modems.

3.2.7 'Tools' submenu



Various helpful tools can be called from this window.

'PCD Configurator':

The online configurator can be used not only to configure the programming unit (IBM PC) but also to define or adjust online the distribution of memory, the PCD clock, or the serial communications mode between the PCD and the programming unit. The history file of the PCD can also be viewed and, if necessary, deleted.

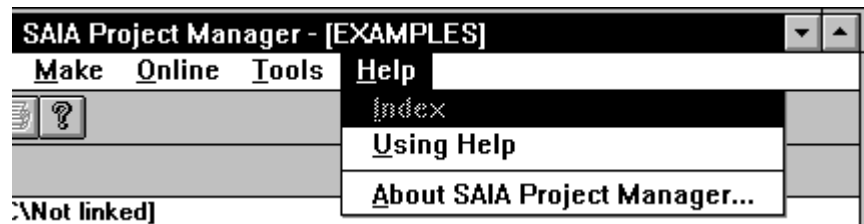
A detailed description of the online configurator follows in section 3.3.

'Debugger': The PG3 debugger can be called. This proven tool is a DOS application. The same tool is contained in the PG3 DOS programming package.

'PG3 Utilities': If the PG3 programming package is installed, it can be run directly from here. The PG3 is not part of the PG4 and must be purchased and installed separately.

'DOS Prompt': It is possible to switch directly to the DOS prompt. To exit DOS again, type <Exit>.

3.2.8 'Help' submenu



'Index': Calls the Help file. <F1> gives the same result.

Currently not yet available.

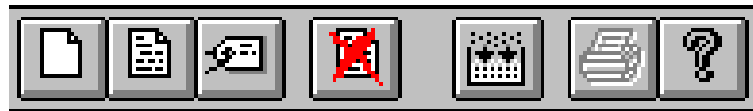
'Using Help': Calls the instructions for the WINDOWS Help files.

'About SAIA Project Manager...':








Displays the following window with information on the installed PG4 version.



3.2.9 Toolbar and "Short Keys"



Clicking the mouse on one of the tool buttons is a convenient way of selecting the most frequently used functions, without opening a menu every time.

	<Ins>	Opens a new file in the current project. Identical to menu 'File' - 'New...'
	<CR>	Opens the marked file. Identical to menu 'File' - 'Edit File'.
	<Alt> + <CR>	Renames a file or changes the properties of a file. Identical to menu 'File' - 'Rename/Properties...'
		Deletes a file. Identical to menu 'File' - 'Delete'.
	<Ctrl> + <M>	Assembles and links the user program. Identical to menu 'Make' - 'Make'.
		Provided for print function. Not yet executable in the present version.
	<Shift> + <F1>	Calls the window with information on the installed PG4 version. Identical to menu 'Help' - 'About SAIA Project Man.'
	<Ctrl> + <O>	Calls the online function. Identical to menu 'Online' - 'Go Online/Offline'

3.2.10 Status bar

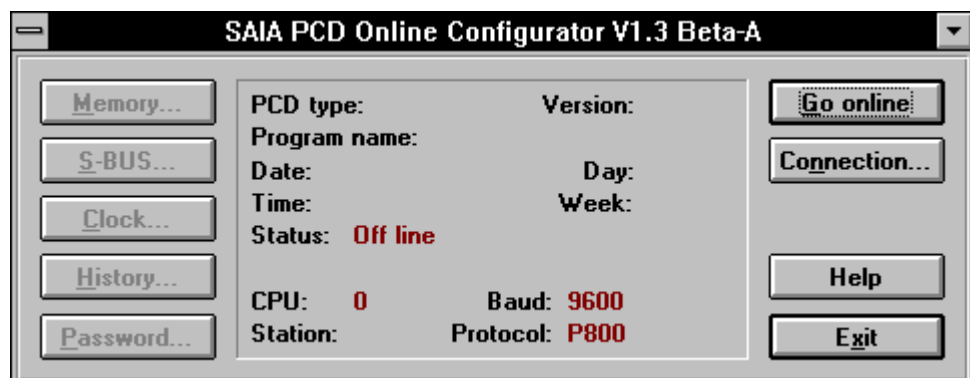


The status bar at the lower window margin shows current function and the ready status of the project library.

If the mouse button is held down on a submenu, the function of that submenu is displayed. The same also applies for the buttons on the toolbar. If the function is not to be executed, move off the tool with the mouse button held down.

- **Date:** Current date by the PCD hardware clock (if provided). If there is no integral hardware clock, the date which appears is 1/1/92.
- **Time:** Current time by the hardware clock
- **Day:** Current day of week: 1 = Monday, ... 7 = Sunday
- **Week:** Current week number of current year
- **Status:** CPU status: RUN, STOP, HALT or Conditional Run
- **CPU:** for PCD2 always 0,
for PCD4 0 or 1
for PCD6 0 ... 6
- **Protocol:** P800 or S-BUS
- **Baud:** Transmission speed:
for P800: always 9600 Baud
for S-BUS: 110 ... 38'400 Baud
- **Station:** Station number:
for P800: not relevant
for S-BUS: 0 ... 254

If the PCD has not been correctly connected or is not switched on, the following is displayed:



After a few seconds, the following message also appears:



or:

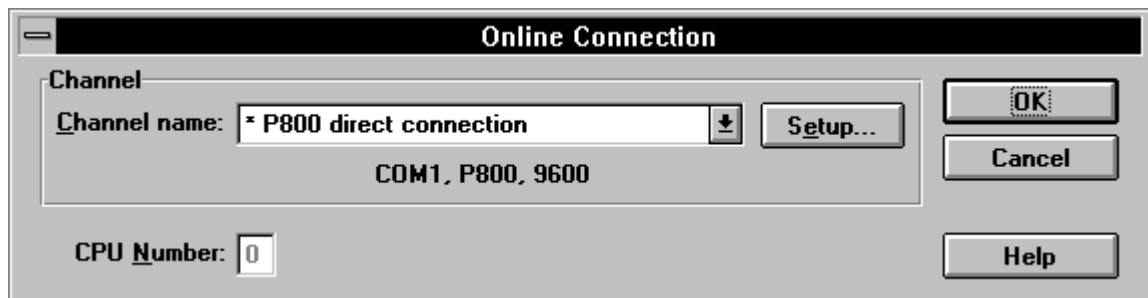


- Help: Detailed description of the configurator in English.
- Exit: Exit the configurator
- Go online/offline:

Switches the configurator online or offline.

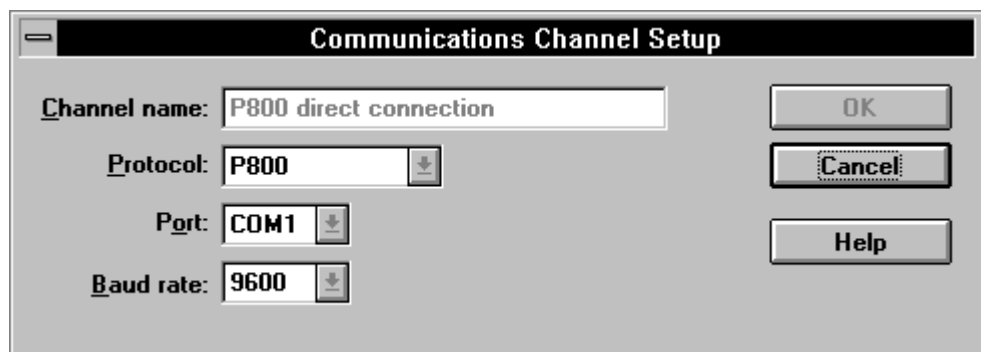
3.3.2 'Connection...' function

After clicking on 'Connection...' in the main window, the following screen appears:



The current communications mode is displayed.

After clicking on 'Setup...' the following window appears:



The current settings are shown.

Changes can only be made when the configurator is switched offline. S-BUS must already be configured if the S-BUS protocol is to be used.

- Channel name: Name of the communications channel. In the "Online Connection" window, one of the three standard installed protocols can be selected:
 - P800 direct connection
 - S-BUS direct connection
 - S-BUS dial-up modem connection

Additional protocols can be defined.

- Protocol: Communications protocol between the PCD and the programming unit: P800 or S-BUS. (Selectable when box has been opened).
- Port: Interface of the IBM PC: COM 1, COM 2 (COM3 ...). When the box has been opened, the available interfaces are listed for selection and confirmation with "OK".

- **Baud rate:** When the box has been opened, all standard transmission speeds are displayed (110 ... 38'400 Baud) and can be selected. However, this only applies when the S-BUS protocol has been set. If P800 is chosen, the baud rate is always 9600.
- **S-BUS Station:** If S-BUS protocol has been selected, the station number is displayed. It is possible here to select between 0 ... 254.
- **CPU Number:** For a PCD4 fitted with an M240 or M440 a choice is possible between CPU 0 or 1. For a PCD6 with a number of CPUs the display shows the CPU to which the programming unit is connected. The CPU can only be changed by unplugging and reconnecting.
- **Cancel:** Aborts Setup and returns to the main window.
- **Help:** Detailed description of the configurator in English.

An example showing how to switch from P800 to S-BUS can be found at the end of this section.

3.3.3 'Memory...' function

After clicking on "Memory..." in the main window, and depending on the connected PCD type, one of the following dialog box appears:

PCD2 Memory Configuration

Type: PCD2.M1_V003 Code/Text memory: RAM

Code/text memory size: 128K Bytes

Extension memory size: 24K Bytes

	Code K Lines	Text K Bytes	Extension K Bytes
CPU 0	28	16	24
Totals	128K Bytes		24KB

Buttons: OK, Cancel, Set Defaults, Help

PCD4 Memory Configuration

Type: PCD4.M14_V005 Code/Text memory: RAM

Code/text memory size: 64K Bytes, PCD4/PCD7.Rxxx

Extension memory size: None

	Code K Lines	Text K Bytes	Extension K Bytes	CPU Present
CPU 0	14	8	0	<input checked="" type="checkbox"/>
CPU 1	0	0	0	<input type="checkbox"/>
Totals	64K Bytes		0KB	

Buttons: OK, Cancel, Set Defaults, Help

PCD6 Memory Configuration

Type: PCD6.M2_V009 Code/Text memory: RAM

Code/text memory size: 256K Bytes, PCD6.R510/R600

Extension memory size: None

	Code K Lines	Text K Bytes	Extension K Bytes	CPU Present
CPU 0	19	10	0	<input checked="" type="checkbox"/>
CPU 1	0	0	0	<input type="checkbox"/>
CPU 2	18	10	0	<input checked="" type="checkbox"/>
CPU 3	0	0	0	<input type="checkbox"/>
CPU 4	18	10	0	<input checked="" type="checkbox"/>
CPU 5	0	0	0	<input type="checkbox"/>
CPU 6	0	0	0	<input type="checkbox"/>
Totals	256K Bytes		0KB	

Buttons: OK, Cancel, Set Defaults, Help

The individual functions are explained below:

- Type: PCD type (see 3.1) and current firmware version

- Code/Text memory:

Memory medium: RAM or EPROM.

All the memory configuration described below can only be carried out if the memory type is RAM.

- Code/text memory size:

After the memory size box has been opened, a list is displayed of the possible options, e.g. for the PCD4:

64K Bytes, PCD4/PCD7.Rxxx
128K Bytes, PCD7.Rxxx
256K Bytes, PCD7.Rxxx

From this it is possible to select the option which corresponds to the hardware.

- Extension memory size:

After the extension memory size box has been opened, a list is displayed of the possible options, e.g. for the PCD4:

"None" or "172K Bytes"

If no extension memory is to be used, select "None".

If the task requires use of the extension memory, it should be selected here while ensuring that the appropriate memory module is fitted. For details see the relevant hardware manual.

- CPU 0: The two fields "Code, K Lines" and "Text, K Bytes" define how user memory is divided up. On the next line the combined "Total" "xxK Bytes" must be equal to the value displayed in the "Code/text memory size" box, otherwise the memory allocation will not be accepted.
Error message:

Error message:



The total is given in K bytes. It should be remembered that 1 line of code equals 4 bytes. This value should therefore be multiplied by 4 and the number of text bytes added.

Example:

14K lines of code * 4	=	56K bytes
+ 16K text bytes	=	16K bytes

Total		64K bytes

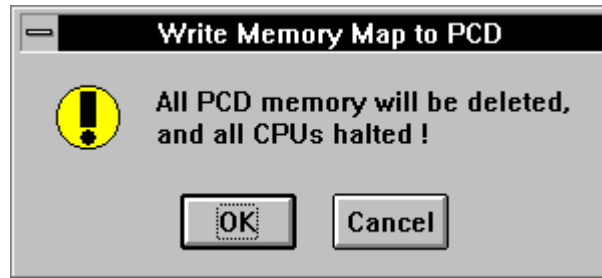
The next field indicates whether there is extension memory and how large it is.

- CPU 1: 2 CPUs can be used with the PCD4, while the PCD6
CPU x: can have up to 7. The "CPU Present" check boxes indicate which CPUs are actually fitted.

Memory is allocated for CPUs 1..6 in the same way as for CPU 0.

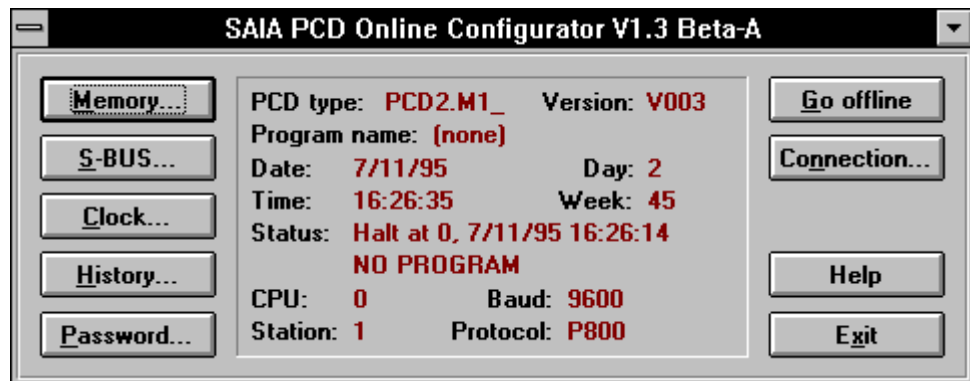
- Set Defaults: Clicking on this field sets the standard configuration of available memory to whichever CPUs are present and allocates a code segment and text segment to each CPU. This can be left as it is or manually adjusted to suit the task.

- **OK:** Clicking on "OK" makes the following IMPORTANT message appear:



If OK is pressed, the memory is reformatted and remains empty. The halt LED on the PCD is turned on.

The main window displays the following:



- **Cancel:** Aborts memory configuration and returns to the main window.

3.3.4 'S-BUS...' function

Consult the S-BUS manual for detailed information on the SAIA® S-BUS.

After clicking on "S-BUS..." in the main window, the following dialog box appears:

The screenshot shows the 'PCD2 S-BUS Configuration' dialog box. It features the following elements:

- S-BUS Station number:** A text box containing '1'. To its right, the text '255 = Not an S-BUS station.' is displayed.
- PGU port number:** A dropdown menu showing '0'.
- PGU port Baud rate:** A dropdown menu showing '19200'.
- Port 0 type:** A dropdown menu showing 'RS-232'.
- S-BUS Timing:** A section containing three rows:
 - 'Training sequence Delay (TS):' with a text box '0' and 'mS'.
 - 'Turnaround delay (TN):' with a text box '0' and 'mS'.
 - 'Response timeout:' with a text box '0' and 'mS'.
- Public Line Modem:** A section with a checkbox 'PGU port uses public line modem' (unchecked) and a 'Modem Name:' label above a text box.
- Buttons:** 'OK', 'Cancel', 'Help', and 'Gateway...' are located on the right side of the dialog.

This function is used to configure S-BUS for communications between the PCD and the programming unit (IBM PC). The first online contact between the IBM PC and the PCD always takes place using P800 protocol.

From the "SETUP..." menu, communications can only be switched to "S-BUS" if the PCD has been configured for S-BUS. This is done from this dialog box, which transfers the configuration to the PCD using P800 protocol.

The individual functions are explained below:

- S-BUS Station number:

The station number of the connected station must be specified: 0 ... 254.

255 = no S-BUS station

- PGU port Number, CPU 0:

The PCD serial port through which communication is to take place with the programming unit (IBM PC) must be specified. A choice is possible between ports 0 ... 3. (This will usually be port 0, since this is always present).

For the PCD6, the S-BUS protocol can only be used through ports 0 ... 3 and not through the PGU interface via the interface processor.

- PGU port Baud rate:

After this box has been opened all standard transmission speeds are displayed (110 ... 38'400) and can be selected.

- Port 0 type: This item only appears if a PCD2 is connected. When opened, a choice is offered between "RS232" and "RS485". Both these interface types are available as standard on the PCD2. On the PCD4, port 0 is always type RS232.

For the other interfaces (1 - 3) interface type depends on the hardware. This also applies for the PCD4 and PCD6.

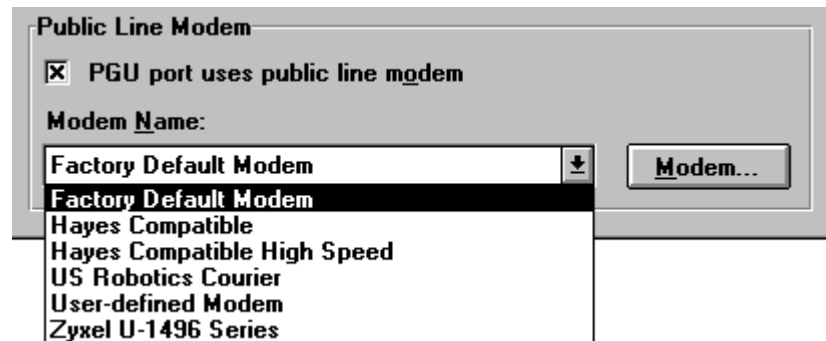
S-BUS Timing:

- Training sequence Delay (TS):
- Turnaround delay (TN):
- Response timeout:

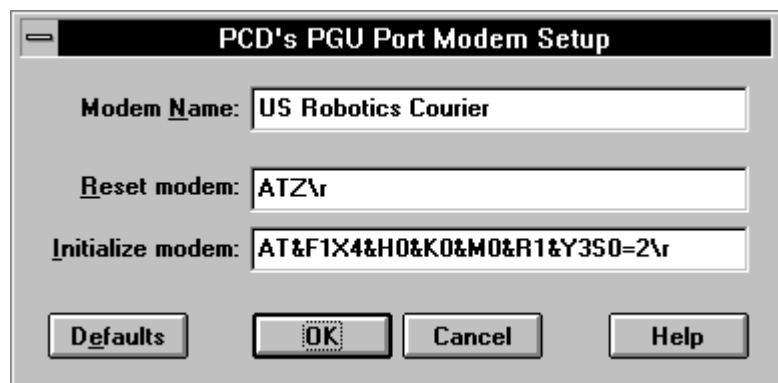
These 3 parameters are only relevant for connection via a modem. The appropriate manuals should be consulted.

Public line modem:

This option can be switched on or off by clicking on the check box. When it is switched on an "x" is displayed. By clicking the mouse, a list of supported modem types can be displayed and one of them can be selected.

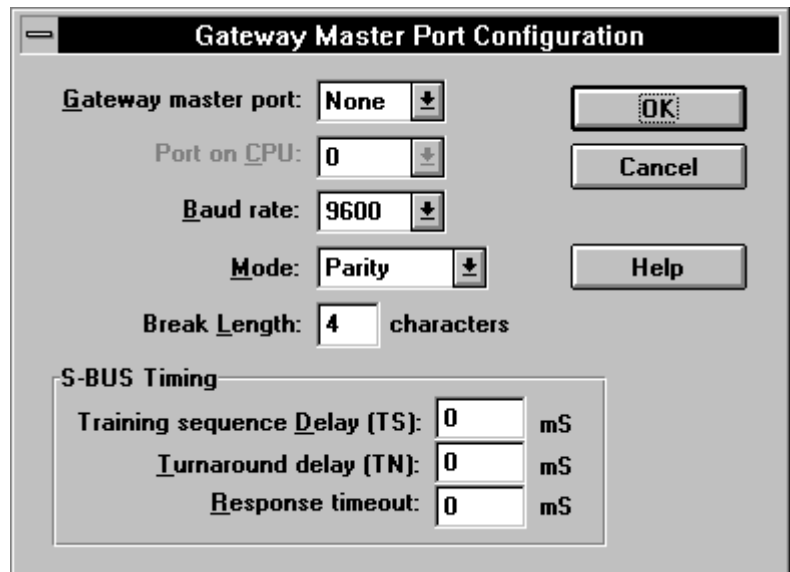


A single click on the 'Modem...' button displays a window with detailed information on the chosen modem, which can be accepted or adapted.



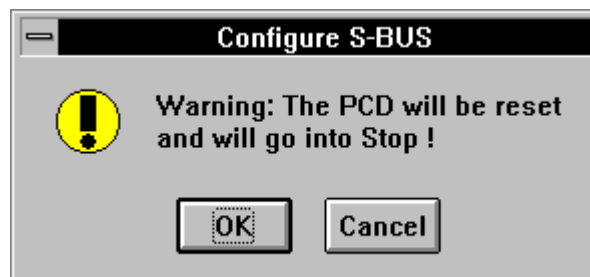
Please refer to the S-BUS manual for more extensive information on modem communications.

'Gateway...': After clicking on 'Gateway...' the following window appears:



Please refer to the S-BUS manual for more extensive information on the gateway functions.

'OK': "OK" does not immediately validate the settings. The following IMPORTANT message appears first:



If it's OK to reset and stop the CPU, press the "OK" button. The CPU stops and the S-BUS configuration window is closed.

Using "SETUP...", the same parameters should now be entered as in the "S-BUS..." menu, i.e.

Protocol: S-BUS, the station number is entered automatically

Baud rate: e.g. 2400 *) → "OK"

*) must be same as for S-BUS configuration

The main window appears. The "PCD - IBM PC" connection is now running, as configured, using the S-BUS protocol.

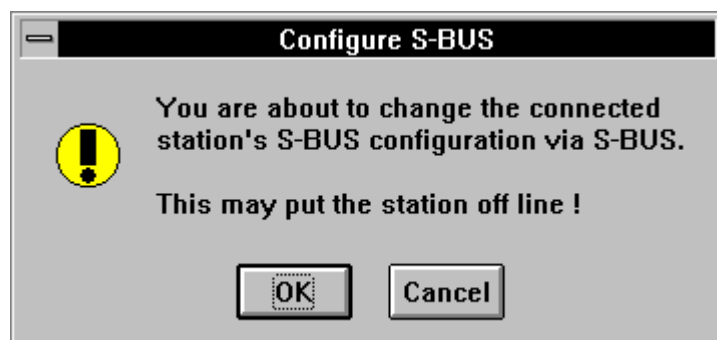
The CPU is in "Stop" and must be returned to "Run" using the debugger or FUPLA. To achieve this the configurator should be exited by clicking on "Exit" and the debugger or FUPLA should be selected from the "SAIA PG4" group in the WINDOWS program manager.

If, for example, the baud rate set deviates from the one configured, "OK" results in the loss of the connection. This message appears:



The CPU remains in "Run". The incorrect entry can be rectified in "Setup..." and confirmed with "OK", or it is possible to switch to the P800 protocol. The P800 protocol should always operate without problem.

If the baud rate or station number are to be changed in S-BUS mode, this should first be done in the "S-BUS..." menu and confirmed with "OK". The familiar message appears advising that all CPUs will be reset and put into stop. If this is authorized with "OK", a new message appears:



With "Cancel" the whole process can be reversed.

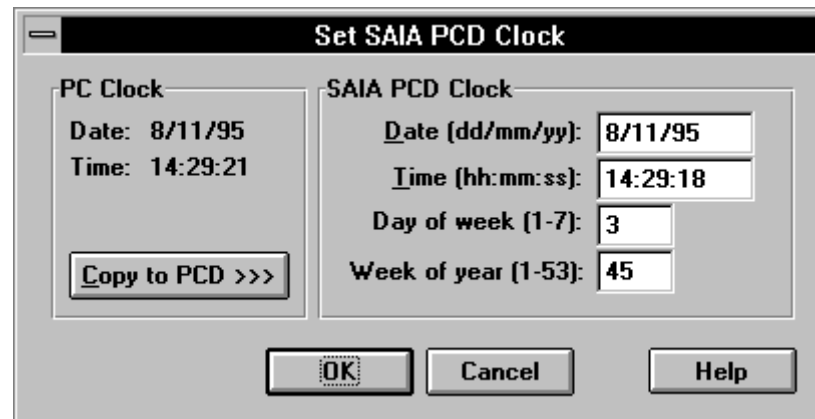
After 'OK' this message appears



The appropriate baud rate should be set in "Setup..." and confirmed with 'OK'. Via the debugger or FUPLA it is then possible to switch the station back into "Run".

3.3.5 'Clock...' function

After clicking on "Clock..." in the main window, the following dialog box appears:



The "PC Clock" on the left contains the current date and time in the IBM PC, The "SAIA PCD Clock" on the right shows the date, time, day of week and week of year for the connected PCD. This naturally assumes that the PCD is equipped with a hardware clock.

Clicking on the "Copy to PCD >>>" button copies the date and time from the IBM PC to the PCD.

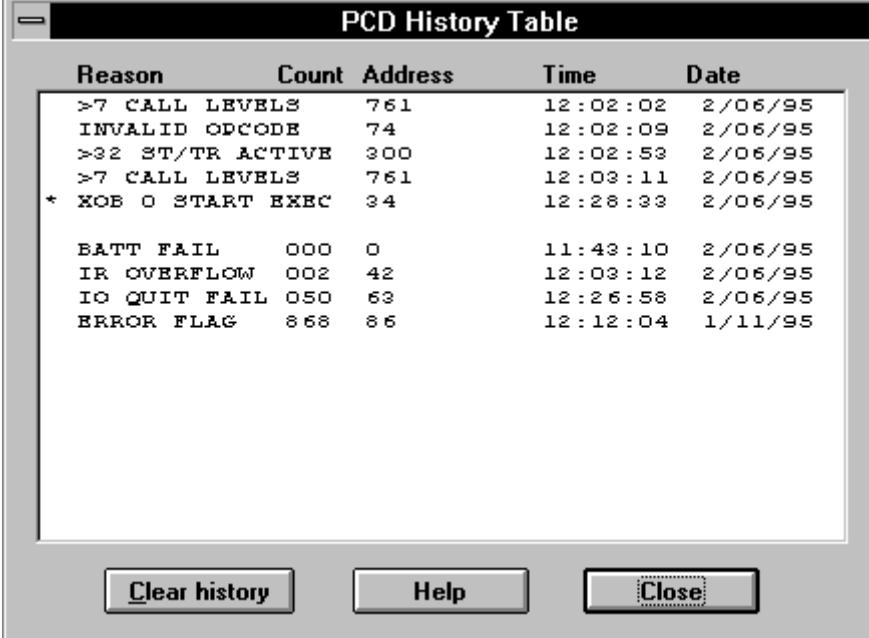
The values in the "SAIA PCD Clock" window can also be entered individually. The entered values are checked for validity.

A reminder:

The easiest way to adjust the clock in an IBM PC is from the "System controller" in the WINDOWS "Main group" with "Date/time", or in DOS using the commands "Date" and "Time".

3.3.6 'History' function

After clicking on "History..." in the main window, the following dialog box appears:



Reason	Count	Address	Time	Date
>7 CALL LEVELS	761		12:02:02	2/06/95
INVALID OPCODE	74		12:02:09	2/06/95
>32 ST/TR ACTIVE	300		12:02:53	2/06/95
>7 CALL LEVELS	761		12:03:11	2/06/95
* XOB 0 START EXEC	34		12:28:33	2/06/95
BATT FAIL	000	0	11:43:10	2/06/95
IR OVERFLOW	002	42	12:03:12	2/06/95
IO QUIT FAIL	050	63	12:26:58	2/06/95
ERROR FLAG	868	86	12:12:04	1/11/95

The history table is displayed for the chosen CPU in the connected PCD.

With one click on the "Clear history" button, this memory can be deleted.

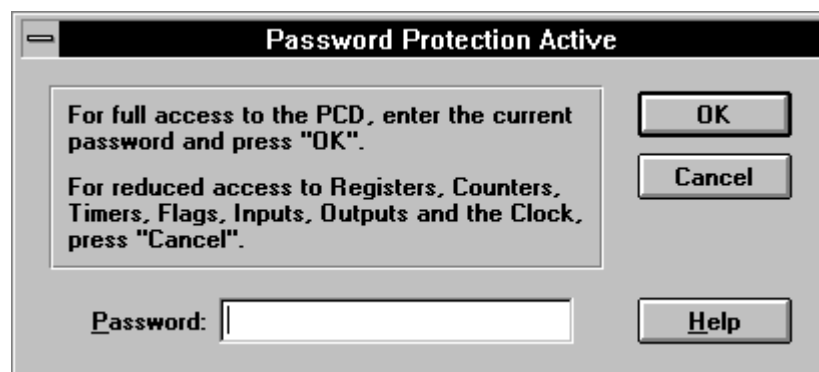
3.3.7 'Password' function

More recent PCDs with the following firmware

PCD2	from firmware version	V003
PCD4	from firmware version	V005
PCD4.M4.	from firmware version	V001
PCD6.M5	from firmware version	V004
PCD6	from firmware version	V007

have a password protection mechanism which allows limited access only to PCD resources for unauthorized persons. If password protection is active, only the reduced communications protocol can be used. This allows access to registers, timers, counters, flags, inputs, outputs and the real time clock. Other resources, such as the memory map (memory distribution and user program), S-BUS configuration and history file, cannot be reached.

If the PCD is protected with a password, the following window appears when the configurator is called:



If the correct password is typed in and followed by 'OK' or <CR>, everything proceeds normally.

If an incorrect password is entered, the following message appears:



After 'OK', the password entry window returns. The incorrect password, which remains visible as "****", must be deleted before any renewed attempt. Otherwise the last password is joined onto the end of the previous one.

If 'Cancel' is selected, work proceeds with reduced access. If one of the functions 'Memory', 'S-BUS' or 'History' is selected, the following message appears:

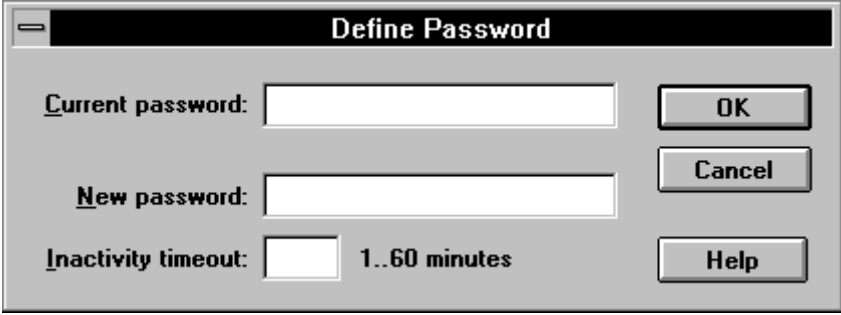


If it is necessary to go from reduced mode to full access, activate the 'Go online/offline' switch twice. The password entry window returns.

If the password is to be entered for the first time, changed, or deleted, the 'Password' switch in the configurator should be operated.

A password can only be changed or deleted if that password is known.

After clicking on 'Password', the following (empty) window appears:



The screenshot shows a dialog box titled "Define Password". It has three input fields: "Current password:", "New password:", and "Inactivity timeout:". The "Inactivity timeout:" field is currently set to "1..60 minutes". To the right of the input fields are three buttons: "OK", "Cancel", and "Help".

If there is not yet any password, the "Current password" field should be left empty. Enter the password in the "New password" field. A password can have a maximum length of 24 characters. For reasons of compatibility with the PCD8.P100 servicing unit, it is advisable not to use more than 8 characters for the password.

A timeout of between 1 and 60 minutes must also be defined. This timeout determines how much time must elapse after the last activity between the PG4 and the PCD before switching back from full access mode to reduced mode.

The password is confirmed with 'OK' and definition is aborted with 'Cancel'.

To change a password, enter the old one in the line above and the new one in the line below.

To delete a password, enter the existing password above and leave the lower field empty.

N. B.: If reduced mode is set after a timeout, the password procedure must be repeated. The 'Go online/offline' button must be operated twice. The password entry window is displayed again.

The password procedure must also be executed if the PCD has been switched off and then on again.

4. FUPLA (Function Block Diagram: FBD)

4.1 Introduction

'FUPLA' is an abbreviation of FUNCTION PLAN. A function plan is the pictorial representation of a logical task using standardized functional elements such as gates, flip-flops, counters, timers, displays, etc..

As one of the SAIA[®] programming tools, FUPLA provides the graphical programming environment. One or more screen pages are edited using graphical symbols and corresponding connections. When the graphical program is complete, i.e. when the task has been formulated graphically, the FUPLA compiler generates a standard ".SRC" format source file. This file must then be assembled, linked and loaded into the PCD. Depending on the settings selected in the "Make Options..." menu, the program switches into Run and can be viewed online on the screen.

The diagram on the screen is converted directly into source code. This easiest case, therefore, requires no knowledge of the instruction set and program structures of the PCD family.

Commissioning a FUPLA program is done from the same graphical screen from which the program was edited. In online operation, all binary connection lines show their status directly:

fine line	= L	(Low)
heavy line	= H	(High)

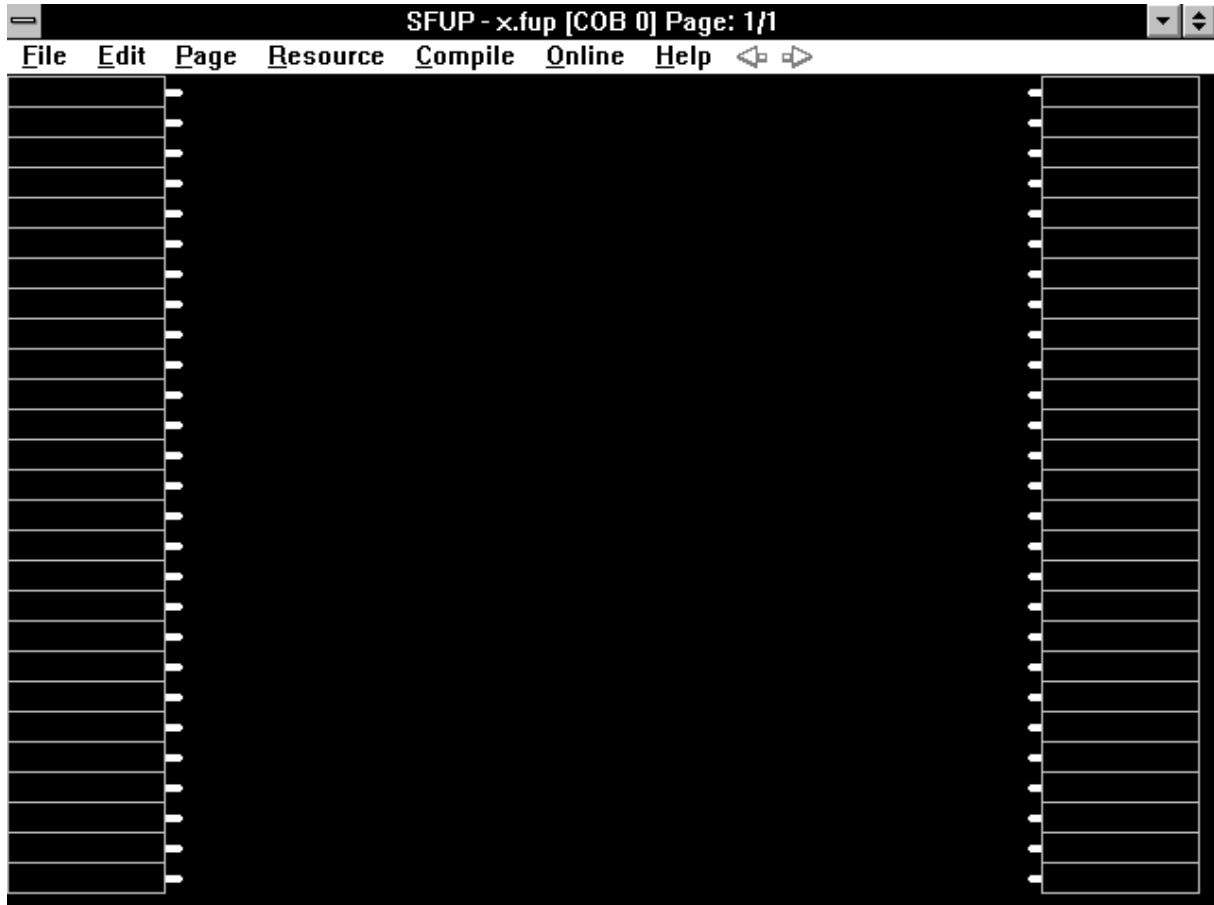
Almost anywhere on the diagram it is possible to insert a small window or "probe". These probes display the actual value in integer format, floating-point, or as 1 or 0 for binary states.

A program can consist of a single FUPLA (on 1 screen page or up to 200), but can also be linked with other FUPLA files, or possibly with pre-existing AWL files, where such "other" files may also contain PBs (program blocks), FBs (function blocks), SBs (sequential blocks) or XOBs (exception blocks), which can in turn again be edited in FUPLA.

KOPLA (contact plan) is a subset of FUPLA and is explained later.

4.2 Description of FUPLA menus

4.2.1 Main menu (main window)



The title line contains the program name: "SFUP", followed by the name of the file being edited. The file name only appears after the first save with 'Save As...' from the 'File' menu. A new file is shown as "(Undefined)".

To the right of SFUP and the file name is the name of the current page, the page number and total number of pages in the current FUPLA. The page title is defined from the 'Page/Info...' dialog box. If no page name is defined, only the page number and page total are shown. A new FUPLA starts as "Page: 1/1".

The main menu bar shows the FUPLA commands:

File Edit Page Resource Compile Online Help ◀ ▶

Clicking on a command opens one of the sub-menus, which appears as a window. Clicking next to the window makes it disappear.

Below the menu bar is the actual FUPLA entry window. It consists of a 55 x 55 grid. On the left 27 fields are provided for inputs, while there are 27 output fields on the right.

Inputs are PCD elements. In simple programs these are usually PCD inputs. However, they may also be flags, outputs, registers, timers, counters or constants in integer or floating point format.

Outputs are also PCD elements. In simple programs these are usually PCD outputs. However, registers or flags can also be used, e.g. for further processing in other parts of the program or for the transfer of intermediate results to other FUPLA pages in the same or another file.

If the FUPLA does not fill the whole screen, scroll bars appear on the lower and right-hand margins, as is usual with WINDOWS. Individual FUPLA pages can be leafed through, but not scrolled.

In the following sections individual submenus are described in the order in which they appear on the menu bar. The order in which the individual functions are applied in practice is described in section 4.3: "Using FUPLA".

Note: This document only describes operating FUPLA with a mouse. "Clicking" means pressing the mouse button on the left (of a right-handed mouse). Any clicking of the button on the right is expressly indicated when necessary.

Keyboard operation is laborious and is not fully supported for is usual in WINDOWS.

As usual with WINDOWS, the keys for selecting menus or submenus from the keyboard are indicated by an underlined letter or key combination in the individual menu window.

4.2.2 'File' submenu

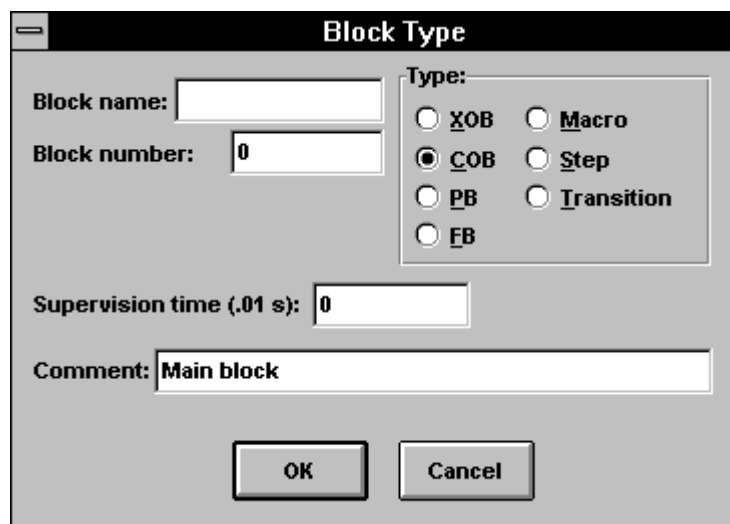
Clicking on 'File' displays the following menu window:



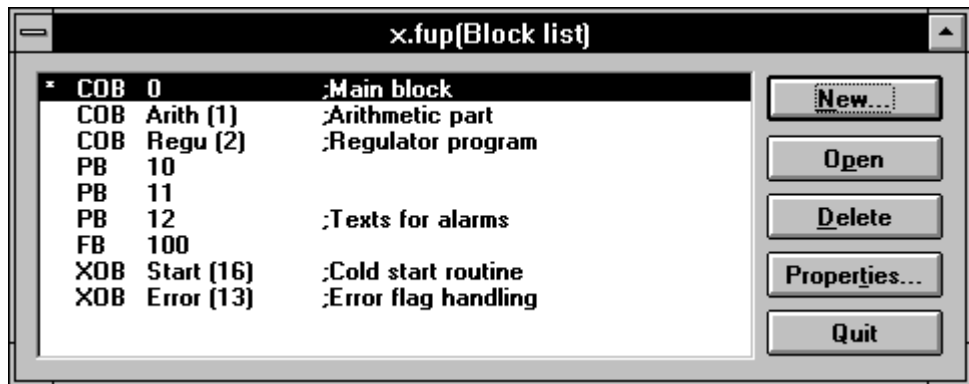
Important: If a menu item is followed by ... an additional submenu is displayed when it is selected.

'Save': The file is saved.

'Block Properties...': This specifies the kind of PCD block, XOB, COB, PB, FB Macro, Step or Transition which will result after compiling the FUPLA. It is also possible to name the block, define the monitoring time for COBs and edit a comment. If no block is specified, COB 0 is selected automatically (default).



Block List...: A list of all blocks contained in the current file is displayed. The block marked with "*" is the one which can be seen on the screen.



'New...': Calls the "Block Type" window, from which a new block can be selected and added.

'Open...': Calls the block which has been clicked on in the list. Double clicking in the list produces the same result.

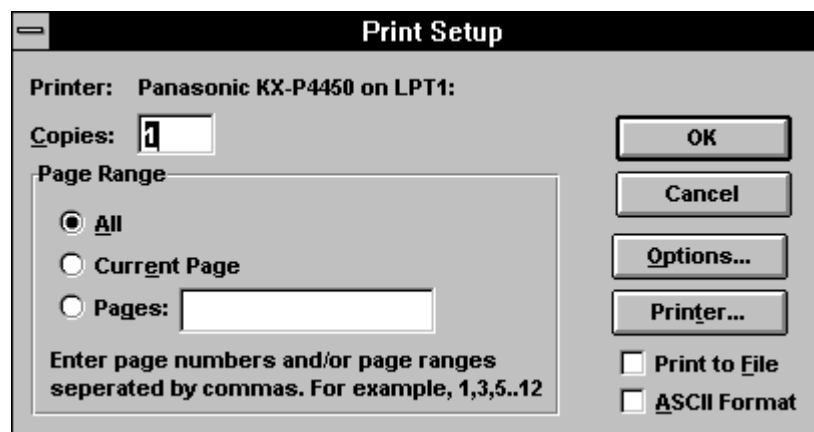
'Delete': The block selected from the list is deleted.

'Properties...':Calls the "Block Type" window.

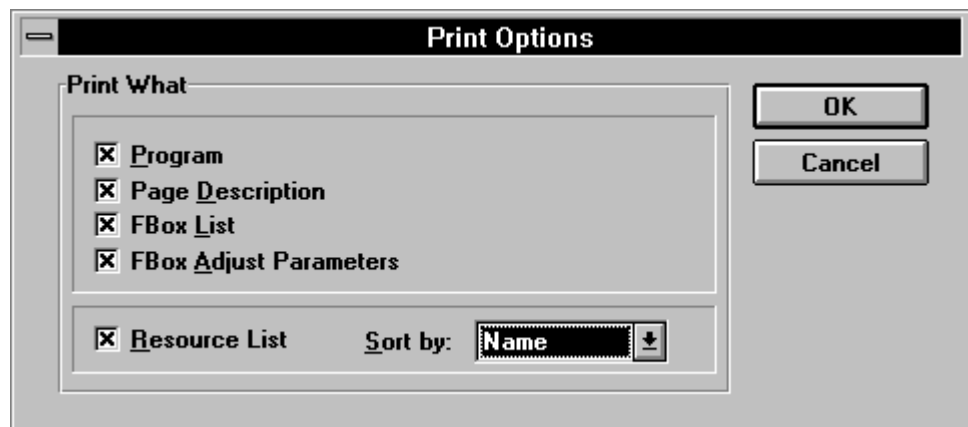
'Quit': Exits the window

Clear Errors: If errors have been discovered during compilation, some of them will be marked on the diagram on the screen. This command removes the error messages sideways from the screen.

'Print...': The following window appears:



Options...: The following window is displayed:



The various options can be switched on or off with a mouse click. "Sort by:" can be used to sort the resource list by "Name" (symbolic name) or "Media" (inputs, outputs, flags, etc.).

"Print to File" and "ASCII Format"

If both are switched off (no x), the resulting printout is in graphical format according to the settings in the "Print Options" window.

Print to File: A file is generated with the extension .FPR. This file can be printed later with the DOS command "Print".

ASCII Format: When this option is switched on, the "Print to File" option is also switched on automatically. An ASCII file with the extension ".TXT" is generated.

Use: For printing with line printers (no graphics) or for incorporation in text files using an ASCII editor.

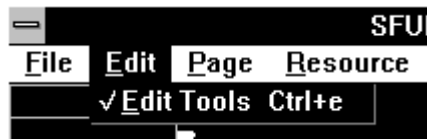
Examples with printouts in the formats described are shown in section 4.6, "Introductory example".

'Printer Setup...': One of the installed printers can be selected.

'Exit': Exits FUPLA.

4.2.3 'Edit' submenu

Clicking on 'Edit' displays the following menu window:



'Edit tools': Clicking on this function displays the 'Toolbox'.



The toolbox can be moved around with the mouse in the usual way.

For standard use, the toolbox contains 6 common functions for drawing the function plan diagram.

The buttons are "radio buttons", i.e. only one button can be depressed at a time. When another button is selected, the previous one is released (pops up).



Arrow

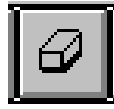
General pointer symbol, e.g. for positioning the cursor when entering labels or addresses, for displaying help on a particular function symbol, or for calling the adjust window (double click on the previously positioned symbol).



Lines

Lines are displayed in various colours depending on their data type:

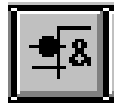
- red: 1-bit connections, e.g. between logical linkages
- blue: integer values, e.g. the values sent to a blinker or the connections between arithmetical integer functions.
- yellow: floating point values, e.g. connections between arithmetic floating point functions.



Eraser

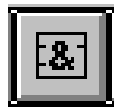
Clicking on this button changes the cursor to a fine red cross appear on the screen. It can be moved around with the mouse as necessary. If this cross is positioned onto a function symbol or connection line and clicked, the symbol or line is erased.

If a symbol is deleted by mistake, it can be drawn again directly by switching over to "Functions".



Inverter

Clicking on the inverter makes the cursor appear on the screen as an arrow, combined with the inverter. An inverter can be placed on the left margin(inputs), right margin (outputs) or on any binary function symbol. The binary function is inverted with a single click.

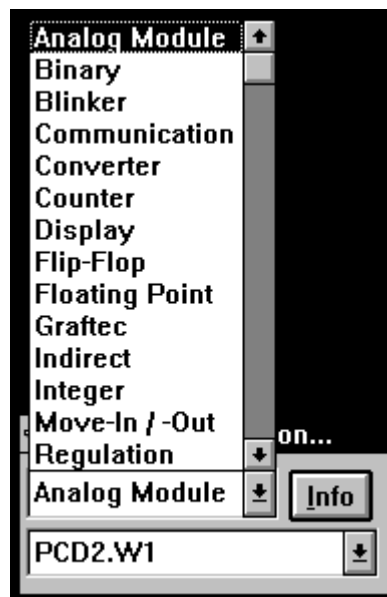


Functions

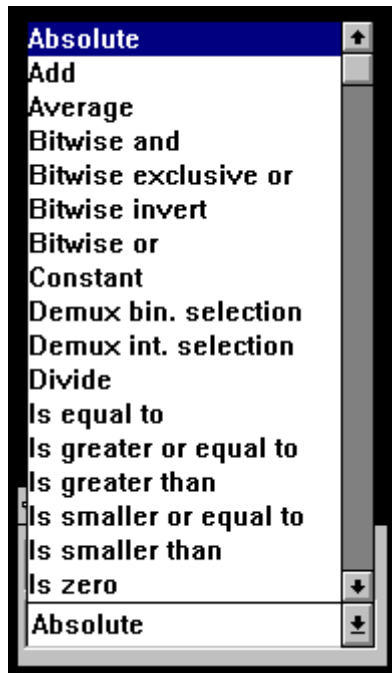
Displays sub-menu 'FBos Selection...'



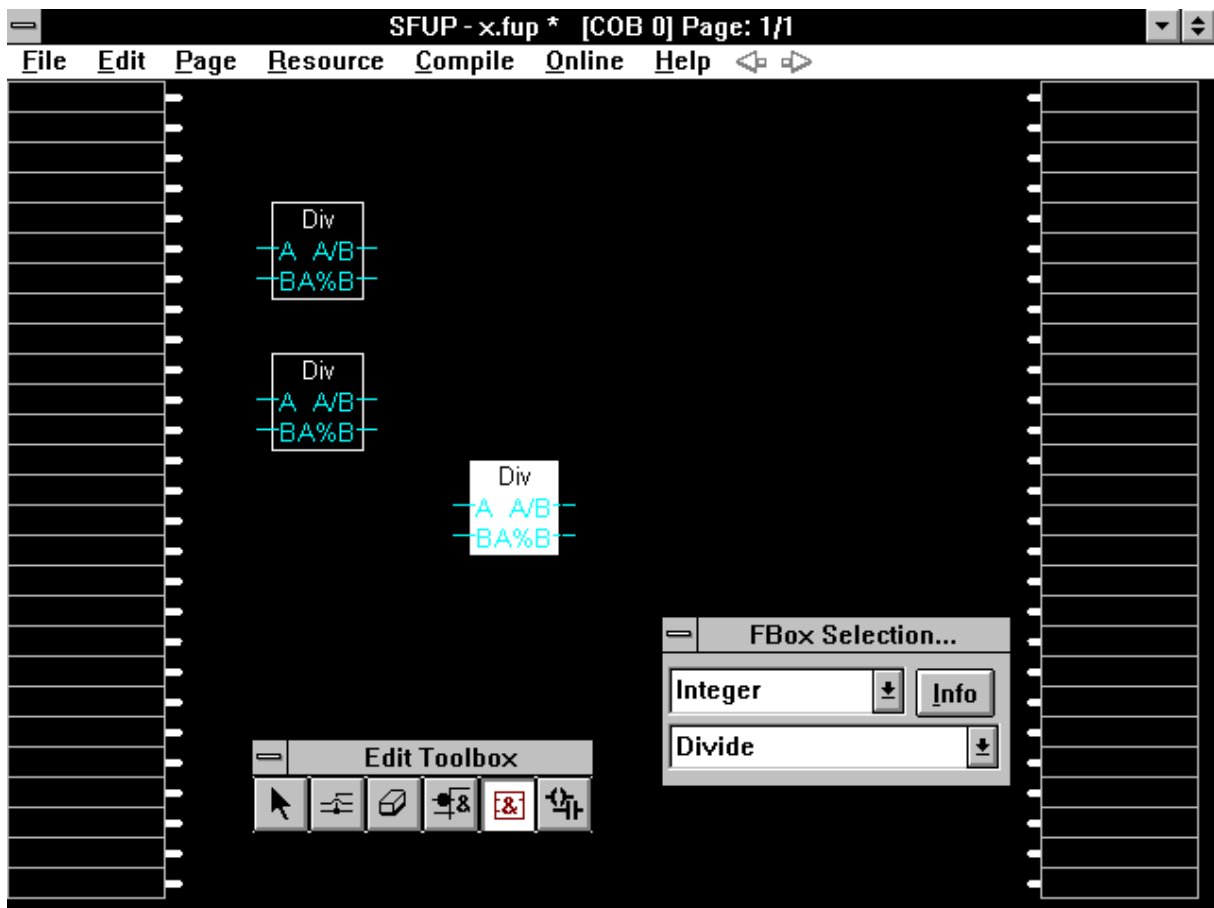
Clicking on the upper list box displays the list of function families in alphabetical order. Clicking on the desired family, e.g. 'Integer' selects the integer functions.



Clicking on the lower list box (showing 'Add') displays a list of all functions in the "Integer" family, and a particular function can be chosen.



For example, if 'Divide' is clicked on, the "Division" function is selected and the mouse can be used to position it on the screen with a further click.

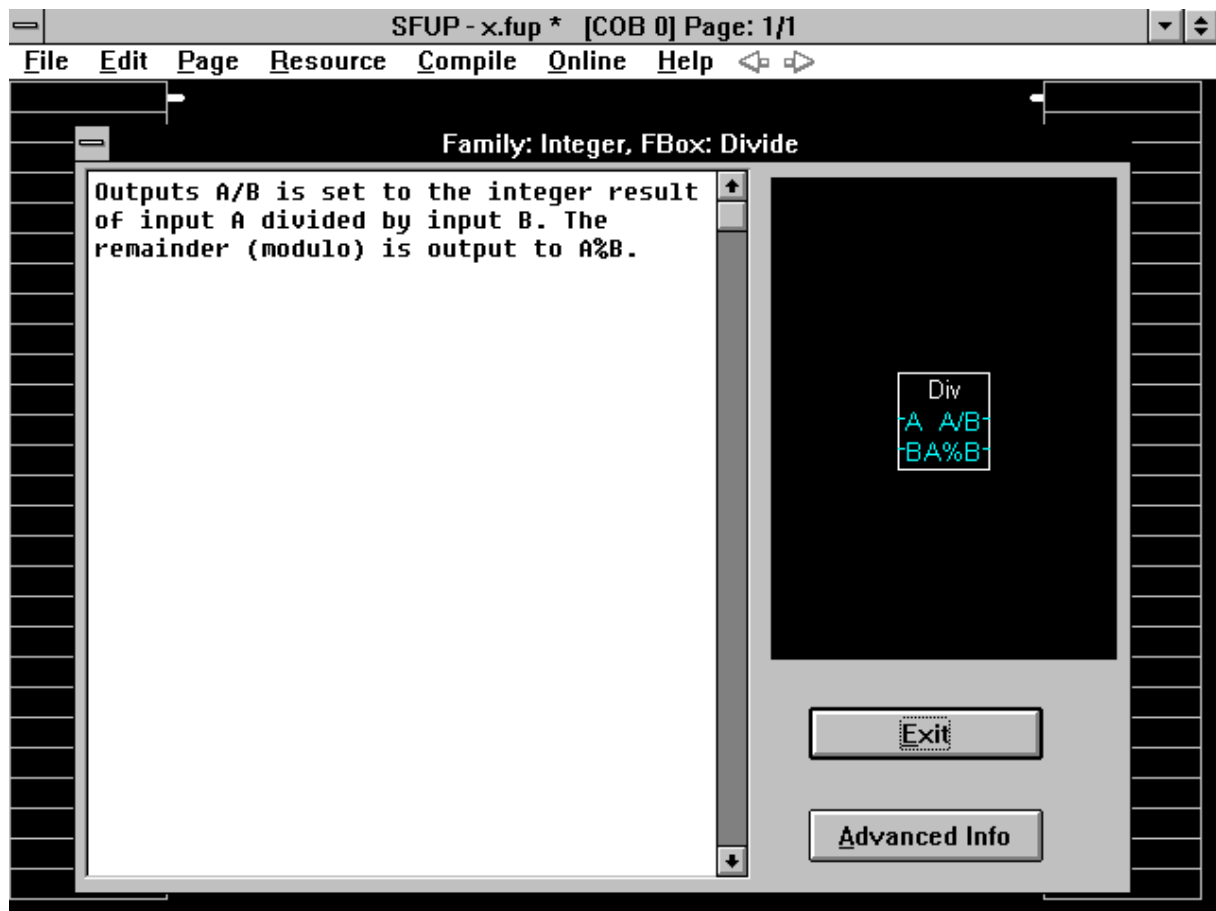


Further, similar functions can be positioned on the screen.

Clicking on the right-hand mouse button displays the arrow and the FBox can be repositioned horizontally.

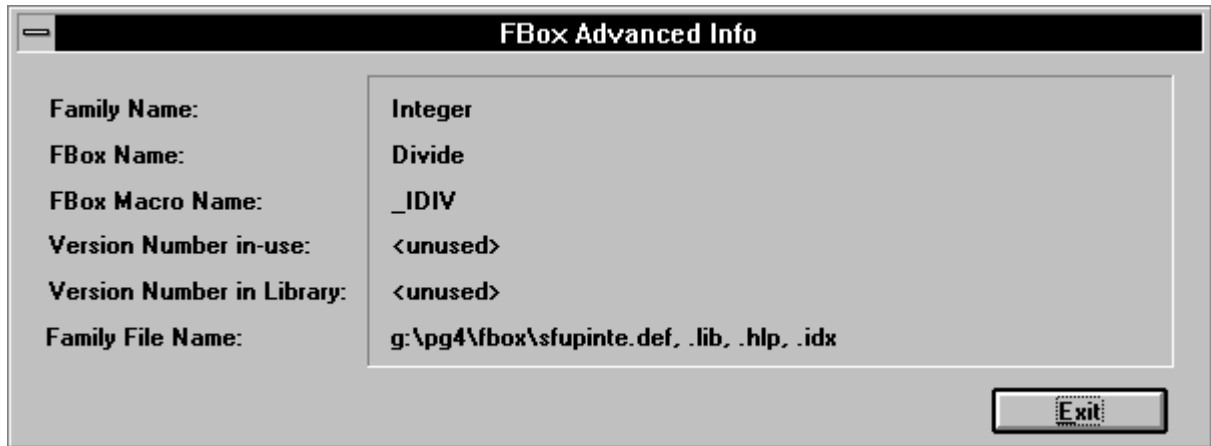
To move an FBox vertically, delete it with the eraser and then immediately call the "function" again. By selecting the deleted FBox, it can be repositioned.

Info: Clicking on 'Info' displays an explanation of the selected function.



The manual entitled "The functions of FUPLA and KOPLA" (order number 26/749 E) contains detailed information on the individual functions.

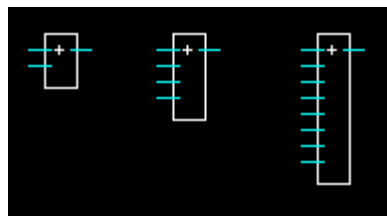
Advanced Info: Displays a window with detailed information on the current FBox. This information is not usually important for the user, but can be very useful when there are updates, new versions, or discrepancies between different language versions, etc.



There are 3 different types of FBox:

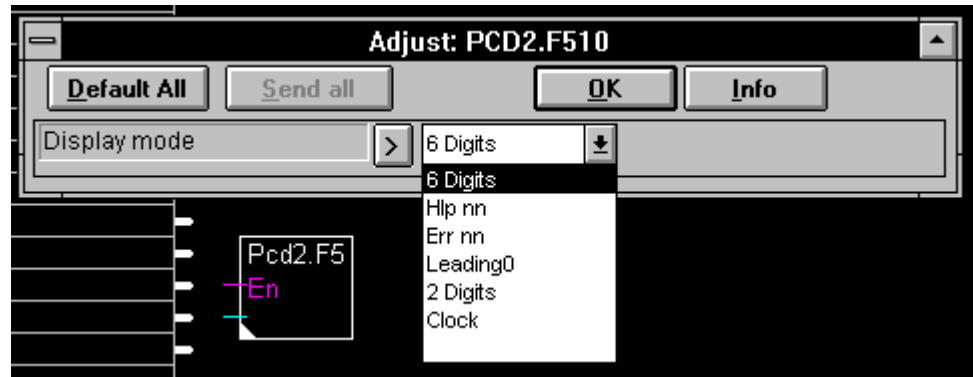
- simple FBoxes (as just described)
- stretchable FBoxes
- more complex Fboxes with an adjust window

With the "stretchable" Fboxes, the number of inputs or outputs can be selected by dragging the mouse when it is drawn. Let us take the 'Addition' function from the "Integer" family as an example. The FBox is positioned with an initial click. By dragging the mouse towards the bottom of the screen, the number of inputs can now be selected between 2 and 8. Termination is achieved by clicking again. If the right-hand mouse button is pressed after positioning, the Fbox can be repositioned.



Various more complex functions have an adjust window, in which it is possible not only to enter function editing data, but also to view online information about the process. This window is displayed by double-clicking on the function box with the arrow cursor. From the adjust window it is also possible to call the "Info" for the current FBox.

Function "PCD2.F510" from the "Display" function family is shown as an example.



For Fboxes without an adjust window, the relevant "Info" can always be displayed by double-clicking on the individual FBox. The arrow symbol must be selected for this.



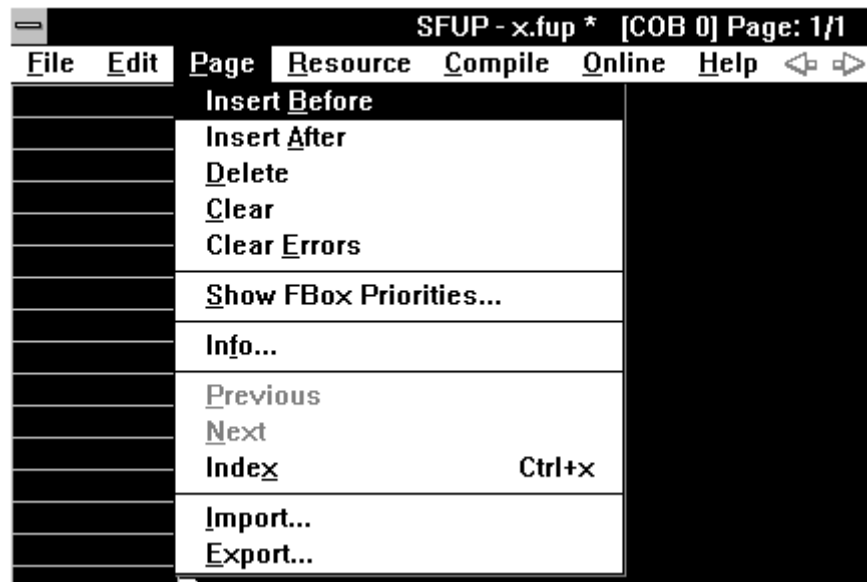
(Ladder diagram) Clicking on this button displays the KOPLA toolbox. Any one of the 10 symbols can be picked up with a mouse click, moved across the screen and positioned on the diagram with a click.



Clicking on the 'Info...' (Help) button displays a description of the selected function.

4.2.4 'Page' menu

Clicking on 'Page' displays the following sub-menu:



This sub-menu is used for referencing one or more FUPLA pages.

As already mentioned in section 4.1, a function plan consists of one or more pages which can be leafed through, but which cannot be scrolled. During editing, debugging or online viewing it must be possible to select or create individual pages, this is done from this sub-menu.

Insert before: This command inserts a new blank page before the current one.

Insert after: This command inserts a new blank page after the current one.

Delete: Deletes and removes the current page.

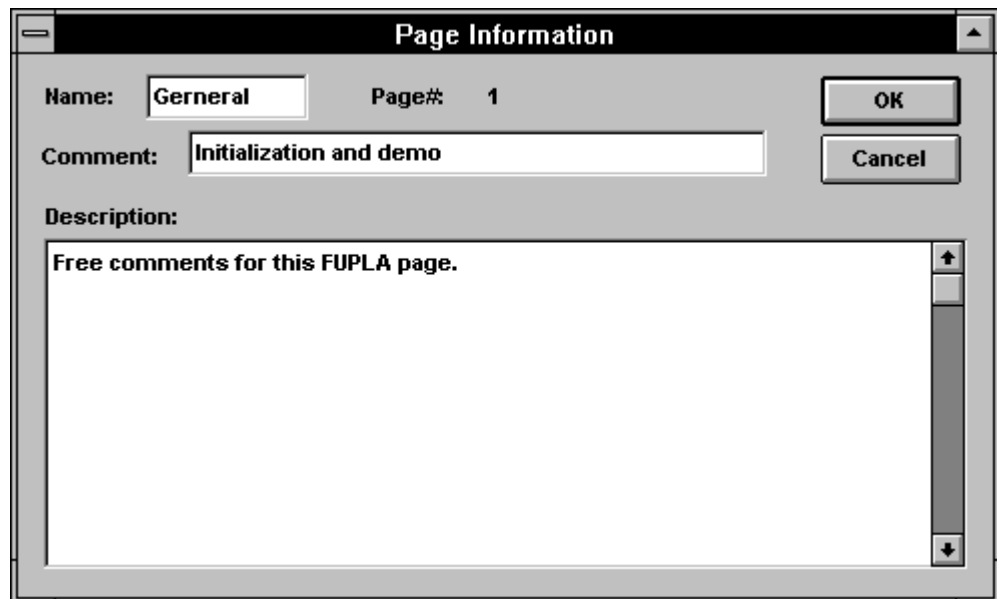
Clear: Clears (but does not remove) the current page.

Clear Errors: If errors were discovered during compiling, some of them are shown on the diagram on the screen. This command is used to remove the error messages.

Show FBox priorities...:

This function displays the processing sequence of each function on the current page. Attached to each function is a little white box containing a number, starting from 1. A small dialog box also appears, containing a button labelled 'Click Me'. Clicking on this button removes the priority boxes from the screen.

Info...: After clicking on 'Info...' a window with 3 fields appears:



- **Name:** Name of current page (max 10 characters). The page name is shown in the title line on the screen.
- **Comment:** Comment on the current page (max. 40 characters). This comment is shown in the 'Index' window (a list of all pages in the FUPLA file).
- **Description:** In the current version 1.2, this is limited to 400 characters.

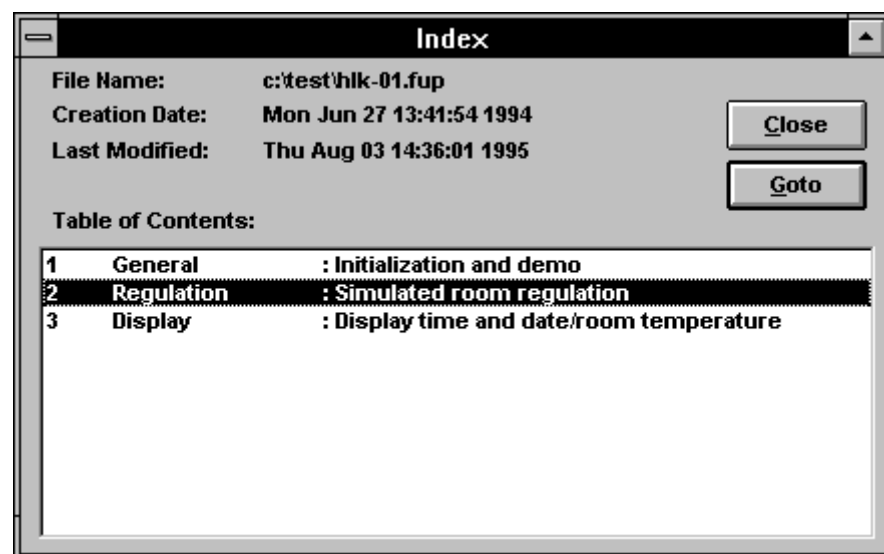
Previous: Displays the previous page. The same function can also be executed with the ← arrow on the menu bar.

Next: Displays the next page. The same function can also be executed with the → arrow on the menu bar.

Index: Clicking on 'Index' displays a window showing an index of all pages in a FUPLA file. Up to 200 pages can be created.

The window contains the file name with path, the date and time of creation (1st save) and the date and time of the last change.

The index itself contains the page numbers, titles and comments for each page.



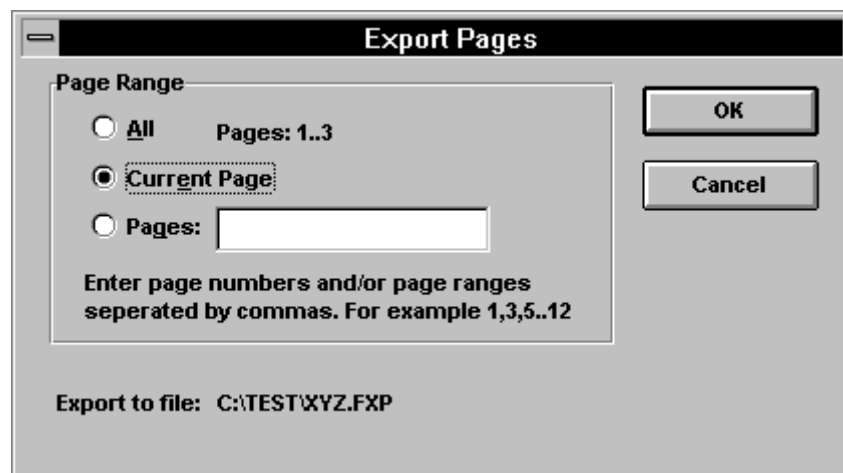
Import/Export:

The import and export functions are available for side-ways copying.

Export: Creates an export file "xxx.fxp". This file can be inserted (imported) into a new or pre-existing FUPLA file. Exporting is possible for a complete file (with a number of FUPLA pages), the current page, or a selection of different pages from a FUPLA program.



After 'OK' the following dialog box appears:

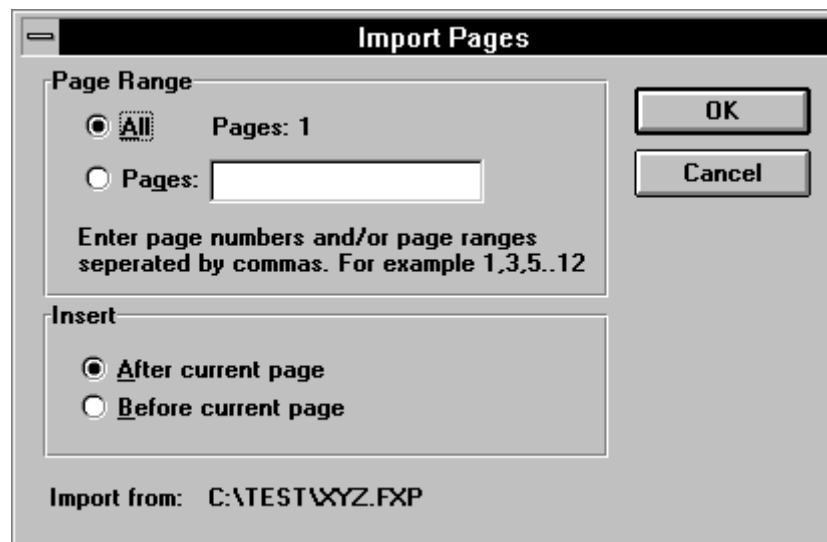


The export file contains the FBoxes and all connections, but none of the addressing.

Import: An "xxx.fxp" file generated with the export function can be inserted (imported) into a new or pre-existing FUPLA file with "Import". Either the whole file can be inserted, or only individual pages. The imported part of the program can be inserted before or after the current FUPLA page.



After 'OK' the following dialog box appears:



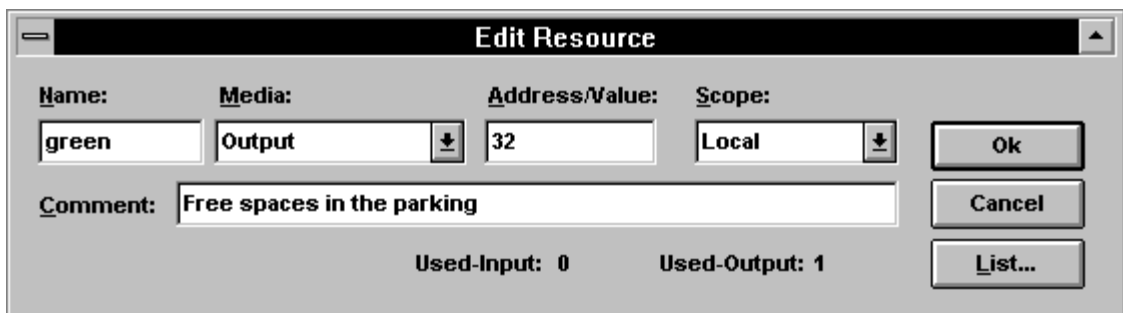
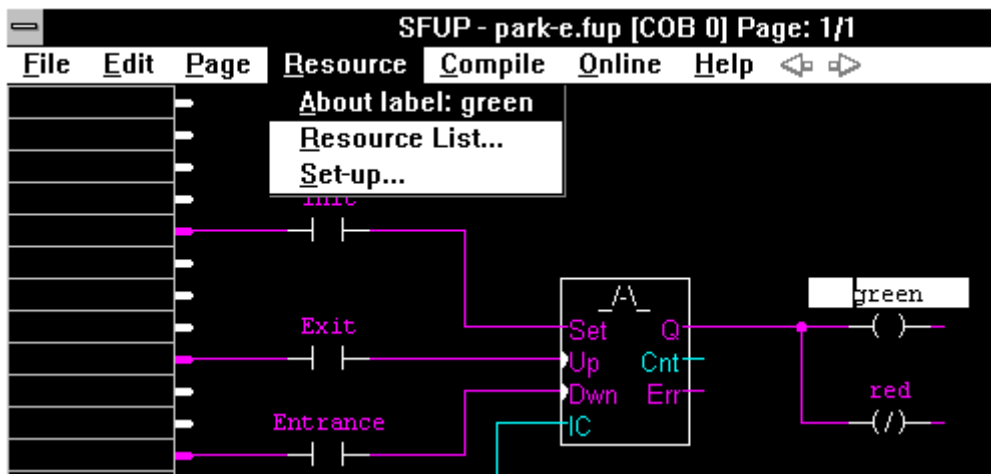
The import file contains the FBoxes and all connections, but none of the addressing.

4.2.5 'Resource' menu

Clicking on 'Resource' displays the following sub-menu:



About: If a resource address is selected on the screen, this address appears at 'About' and can be handled immediately, i.e. the dialog box for that resource is displayed.



The fields 'Name:', 'Address/Value:' and 'Comment' are for keyboard entry. The fields 'Media' and 'Scope' offer a selection which can be displayed and marked by clicking the mouse.

The procedure for editing resources is described in detail in section 4.6, "Introductory example".

'Resource List...':

List of resources used.

Clicking on 'Resource List...' displays all the resources used in this file.

Resource list				
Quit! Options! <u>S</u> ymbols				
Entrance	I	1	Local	
Exit	I	2	Local	
Init	I	8	Local	
Places	I	16	Local	
green	O	32	Local	Free spaces in the parking
red	O	34	Local	Parking full
	K	2	Local	

Double-clicking on any individual element in this list displays the detailed window for that element, as already seen.

Syntax for symbols:

- Symbols must start with a letter (a-z, A-Z).
- A symbol must consist of at least 2 characters.
- Underlining "_" is permitted, but not as first character.
- ,,,:;\ etc. are not allowed.
- The "ß" (German double s) cannot be used.
- Assembler declarations (PUBL, AUTO, EQU...) and mnemonics (STH, COB, BITL...) cannot be used as symbolic names. See also the list of commands in Chapter 6.

Use of the "Resource" menu is dealt with in detail in sections 4.3 "Using FUPLA" and 4.5 "Introductory example"

Description of individual menu items:

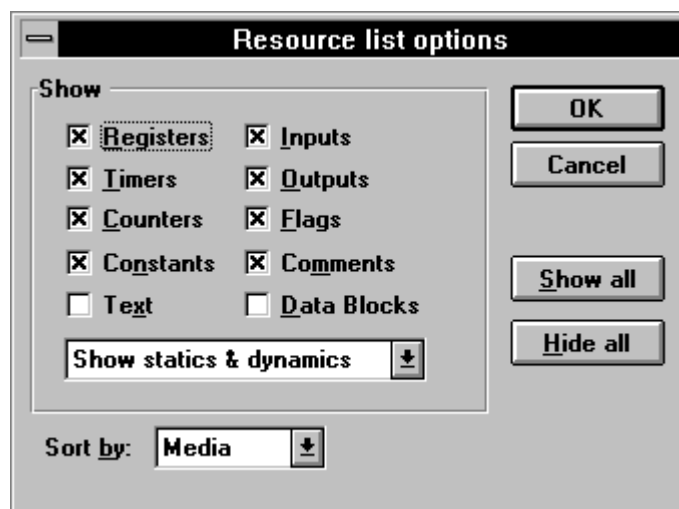
'Symbols': A small menu window is displayed with "New", "Edit", "Delete".

'New': Not yet supported, i.e. an element can only be defined if it has already been edited in FUPLA.

'Edit': The detailed window on the element indicated in the list is displayed on the screen. (Same function as double-clicking on the element line in the list).

'Delete': An element can only be deleted if it has previously been removed from the FUPLA diagram.

'Options!': The following window is displayed:



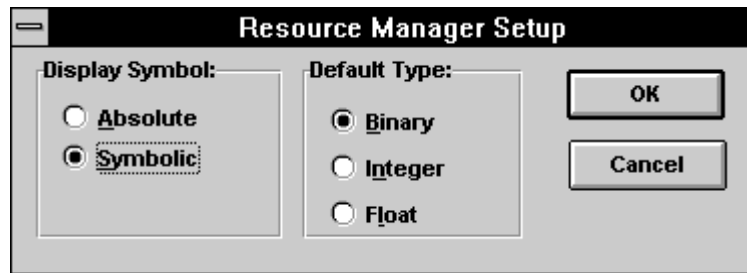
It is possible to select which elements (resources) are to be listed.

Regarding static and dynamic variables, please refer to the "Introductory example" in section 4.5.

'Sort by': The resources can be sorted by element type (Media) or in alphabetical order by element name (Name).

'Quit': Exits the resource menu.

'Set-up...': When clicked on, the following window is displayed:



"Display Symbol:"

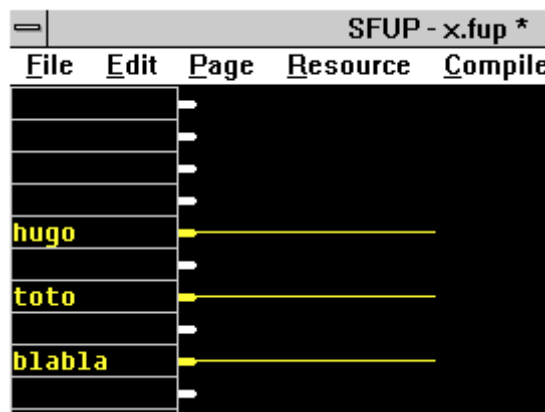
'Absolute': Shows all elements in the diagram with absolute addresses, i.e. I 5, O 35, K 35, etc.

'Symbolic': Shows all elements with symbolic names, e.g. "warm", "valve", "load_value", providing a symbolic name has been assigned to the element.

Click on the 'Absolute' or 'Symbolic' radio button and confirm with 'OK' to change the representation.

"Default Type":

A format can be predefined for resources which are not connected. For example, if "Binary" is selected and a symbol is written into the input or output level, it will be defined as binary. If "Integer" or "Float" is preset, the symbol is defined as an integer or floating point value.



4.2.6 'Compile' menu

'Compile' displays the following sub-menu:



This menu defines and starts the compilation of the function plan. When 'Make' is activated, the program is automatically prepared, i.e. compiled, assembled and linked, then loaded into the PCD and finally put into Run and displayed online, depending on the definitions set under 'Make Options...'

'Compile': If function plan editing is finished, the block type has been selected, resources are fully defined, all connections drawn and dynamic variables defined, it is possible to initiate the generation of the source program simply by selecting 'Compile'.

During compilation the following message is shown:



For a small FUPLA of e.g. 1 page only, compilation is over so quickly that this message is barely perceptible.

If no error messages appear, the source file has been generated correctly.

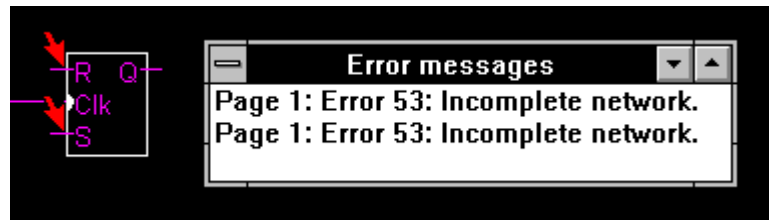
Error messages:

If it was not possible to execute compilation successfully, a message box containing the following text appears:

STOP Compilation error detected.

Another window lists all errors detected, grouped according to page and cause.

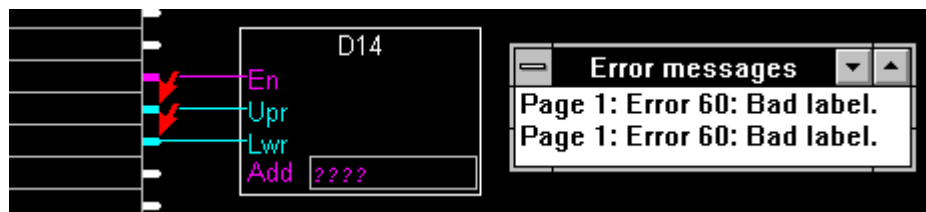
Missing connections are the most common source of errors. These are identified by the message "Incomplete network". Missing connections are marked on the display with a red arrow.



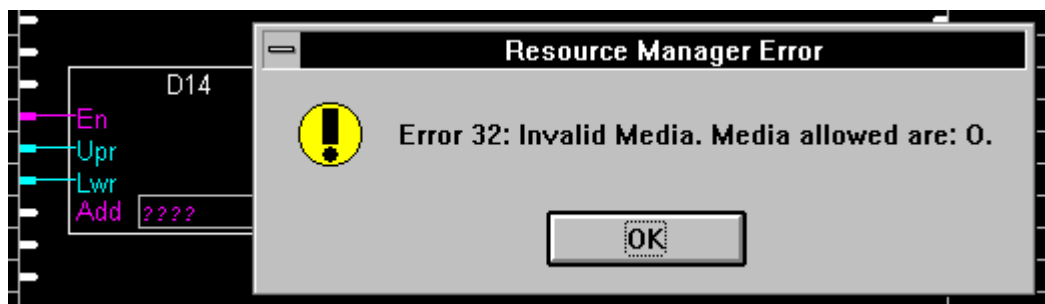
Another source of errors are the loop-back connections. These are identified by the message "Loop-back detected" and are also marked on the display with red arrows.



Missing addresses or incorrect media types are also indicated with red arrows.



All other errors are recognized during editing and reported (in this example, I 25 has been specified at "Add", whereas outputs only are allowed)...



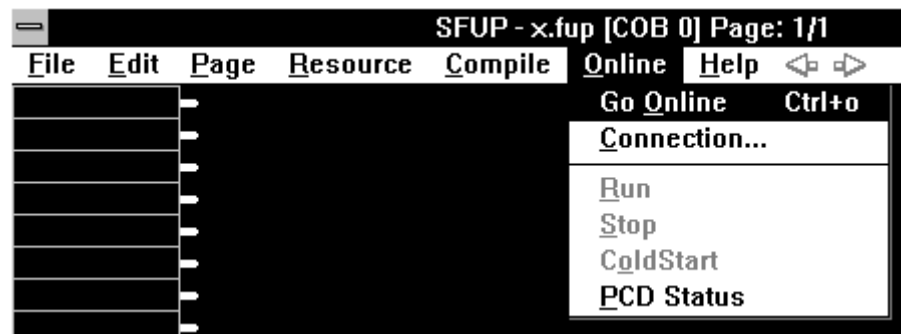
... or else they cannot be edited in the first place.

'Make': Clicking on 'Make' compiles the function plan program (if this has not already been done), then assembles and links it according to entries in the "Make Options..." dialog box described below.

Downloading and online operation are also executed independently according to the "Online" settings described later.

4.2.7 'Online' menu

Clicking on 'Online' displays the following sub-menu:



If the FUPLA program has been properly compiled, assembled, linked and loaded, the online feature can be enabled.

'Go Online': The FUPLA will now run online. All binary connections are automatically adjusted to their actual status, i.e.:

fine red connections	→	status L
heavy red connections	→	status H

The KOPLA symbols, i.e. the inputs, outputs and flags, show their status automatically:

Symbol normal	→	off
Symbol white	→	on

The 'Online Toolbox' is also displayed.



The two keys on the left are "radio buttons", only one can ever be depressed. When one is pressed, the other is always released.

The remaining 3 keys work as normal buttons.

The online toolbox can be moved around and reposition with the mouse.

Online operation can be cancelled by selecting 'Online' from the main menu bar. The top line of the online menu now shows 'Go Offline'. Clicking on this line selects offline operation.

All online functions are reversed. However, the current CPU status is retained. (If the CPU was put into 'Stop' during online operation, it will remain stopped).

CPU status is shown by flashing of either the 'Run' or 'Stop' buttons.

Button 1:



The symbol represents a magnifying glass. The cursor becomes a magnifying glass pointer. If the pointer is moved to a binary (red) line and clicked, a small "online probe" appears, containing "0", "1" or "*".

"0" = binary status = L

"1" = binary status = H

"*" = state not defined

If the pointer is clicked on an integer line (blue) or floating point line (yellow), an online probe appears with a width of 10 digits. According to the number format, either an integer or floating-point number is displayed and is refreshed. If the program is not in RUN, this field also shows "*****".

Online probes can be moved around with the mouse while the mouse button is held depressed.

To delete a probe, click on the probe again. To transfer a probe, click on its connecting line again. The probe will disappear. Only one probe can be connected to each line.

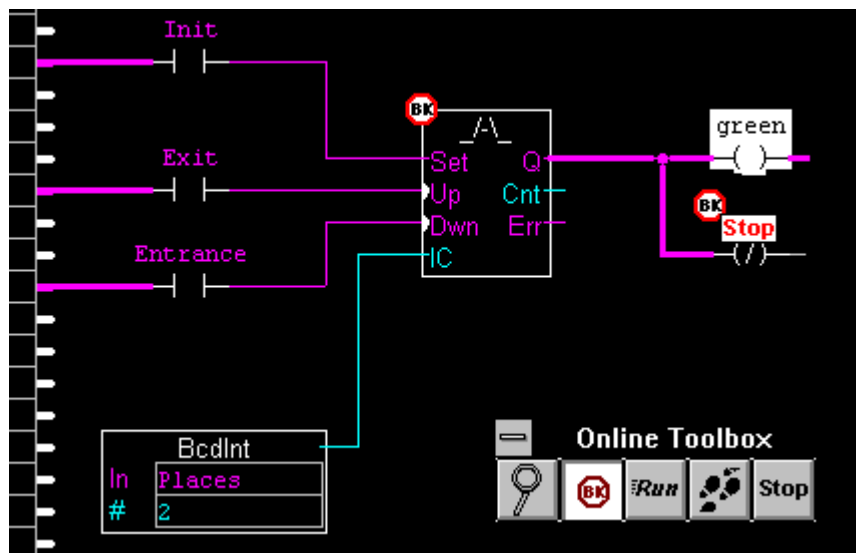
Button 2:



Activates the breakpoint feature. Clicking on a FUPLA function defines a breakpoint, and "BK" is displayed in the left-hand corner of the function. The program will stop when the first function with a breakpoint is reached. (The 'Stop' button flashes to show the status. Pressing the 'Run' button continues).

If two functions have been selected with a breakpoint: after clicking on the 'Run' key, the program runs from the first breakpoint function to the second. Clicking on 'Run' again makes the program run on to the first breakpoint position, and so on. This procedure also works with only one breakpoint, although it is not so easy to see the continuation as it is with two or more breakpoints.

The current breakpoint (function at which the program is paused) is shown on the screen as a white field labelled "Stop".



Online probes are refreshed only after the function has been executed. If the program is paused at another position, *'s are shown instead of the logical state or value. In this case binary lines are shown in white.

Breakpoints are cancelled by clicking on the breakpoint function again with the "BK" button still active. If all breakpoints are cleared, the program can be operated again by pressing the 'Run' button.

Button 3: 'Run' push-button.



Flashes green when the program is in "Run".

Flashes yellow when the program is in "Conditional Run" (put into "Conditional Run" by the Debugger).

Button 4:



"Step by Step" push-button. Each click makes the program step to the next function symbol, retaining the same sequence as shown by the 'Page/Show FBox priorities...' command (Section 4.2.4). This is a kind of "Trace" operation, in which entire functions are executed rather than individual instructions. The 'Stop' button flashes.

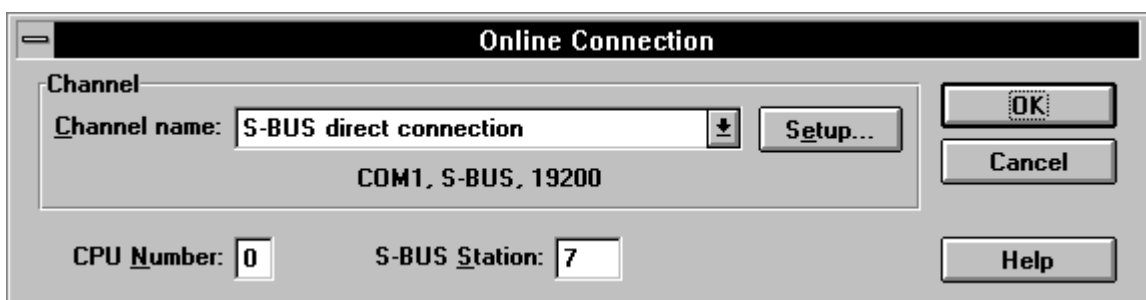
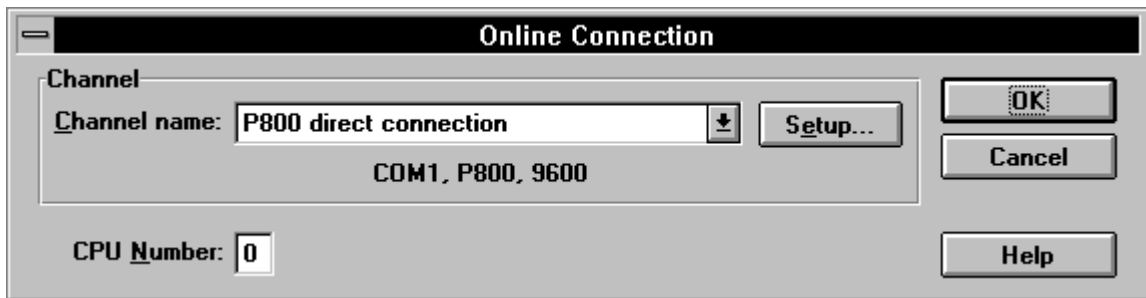
Online probes are only refreshed after a function has been executed, and remain until the end of the FUPLA page. When the page is re-executed, *'s are shown instead of the logical states or values, until the corresponding function has been executed again.

"Step by Step" operation is switched off again by selecting 'Run'.

Button 5: 'Stop' push-button. Flashes when the program is in "Stop" or when "Step by Step" mode is selected.



'Connection...': Clicking on 'Connection' displays one of the following two windows, depending on the communications protocol:



The communications protocol is indicated. If the PG4 is switched offline, it is also possible to change the communications mode.

The procedure is the same as in the project manager, submenu 'Online' - 'Connect Option'. (See Chapter 3.6.2).

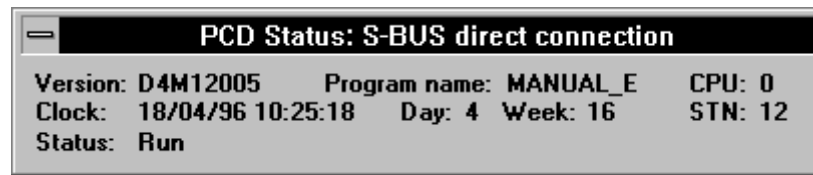
'Run': A click here switches the CPU into "Run".

'Stop': A click here switches the CPU into "Stop".

'Cold Start': A click here executes a "Restart Cold". The CPU remains in "Stop".

'Run', 'Stop' and 'Cold Start' are only active when "Online" is selected.

'PCD Status': Displays the following window:

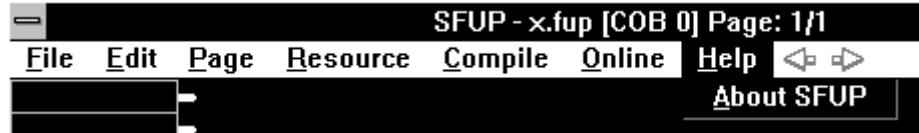


The window shows:

- Protocol: S-Bus direct connection
- CPU type: PCD4.M12
- Firmware version: 005
- CPU number: 0
- Program name: MANUAL_E
- Date/time: 18/04/96 10:25:18
- Day of week: 4 (Thursday)
- Week: 16
- Status: Run

4.2.8 'Help' menu

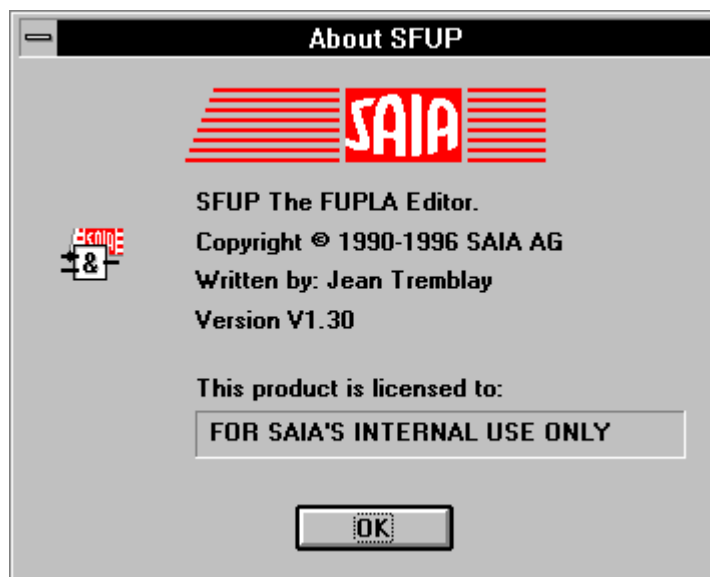
Clicking on 'Help' displays the following sub-menu:



'Fupla Documentation': | These help texts are not yet
 'PG3 Utilities': | available.

'About SFUP': Displays a brief description of the package.

It is possible to check the PG4 or FUPLA version number.



4.3 Using FUPLA

This chapter explains the individual steps for creation, modification, commissioning and online operation of a function plan program.

For descriptions of individual function families and individual functions refer to section 4.4 "Function families", or to the help text (Info) attached to each function.

The details to be noted during editing are shown in section 4.5 "Introductory example".

4.3.1 Creating a new function plan

It is assumed that the PG4 has been correctly installed according to the instructions in Chapter 2.

An initial FUPLA program "EXP_01" is to be edited in a project called

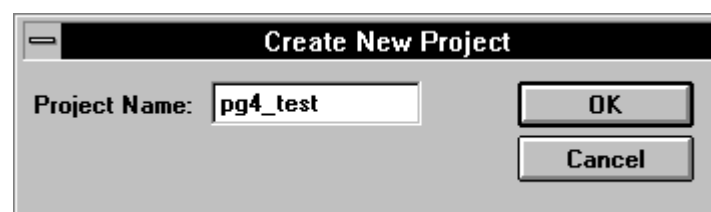
PG4_TEST

and commissioned.

- 1.) The project library is called from the WINDOWS program manager by double-clicking on the "SAIA PG4" program group:

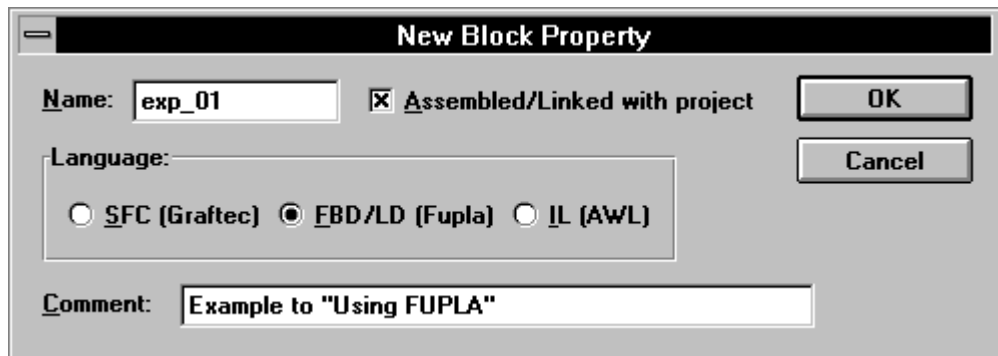


- 2.) The project library is on the screen. With 'File' - 'New...' the following dialog box appears:



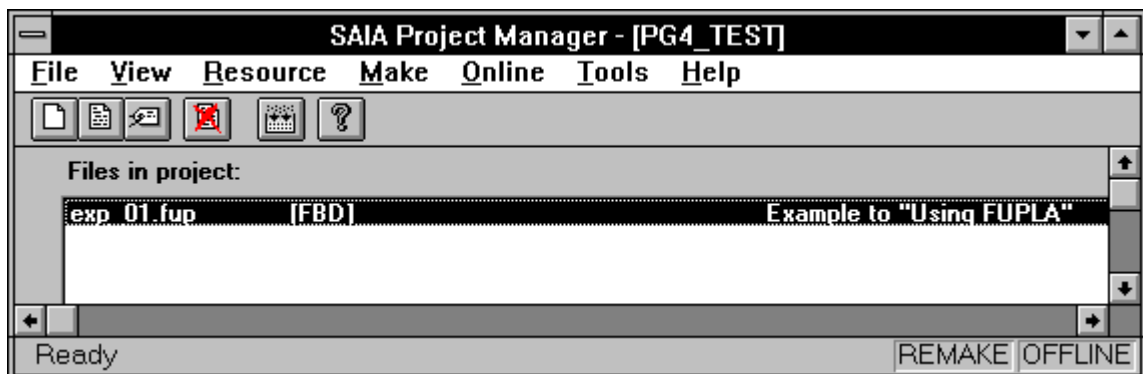
At "Project Name" we enter the name of our project <pg4_test> and click on 'OK'. Our new project is now included in the project list.

- 3.) Double-clicking on the line for our project displays the project manager window. After 'File' - 'New...' the following dialog box appears, which we complete as shown:



As we want to edit our example in FUPLA, we select "FBD/LD (Fupla)". Since we want to be able to run the resulting program, we leave "Assembled/Linked with project" switched on (x) and confirm with 'OK'.

Our file is entered in the project manager window.



- 4.) Double-clicking on the line with the file name displays an empty FUPLA page. Expand to full screen if the FUPLA is displayed as a partial screen.

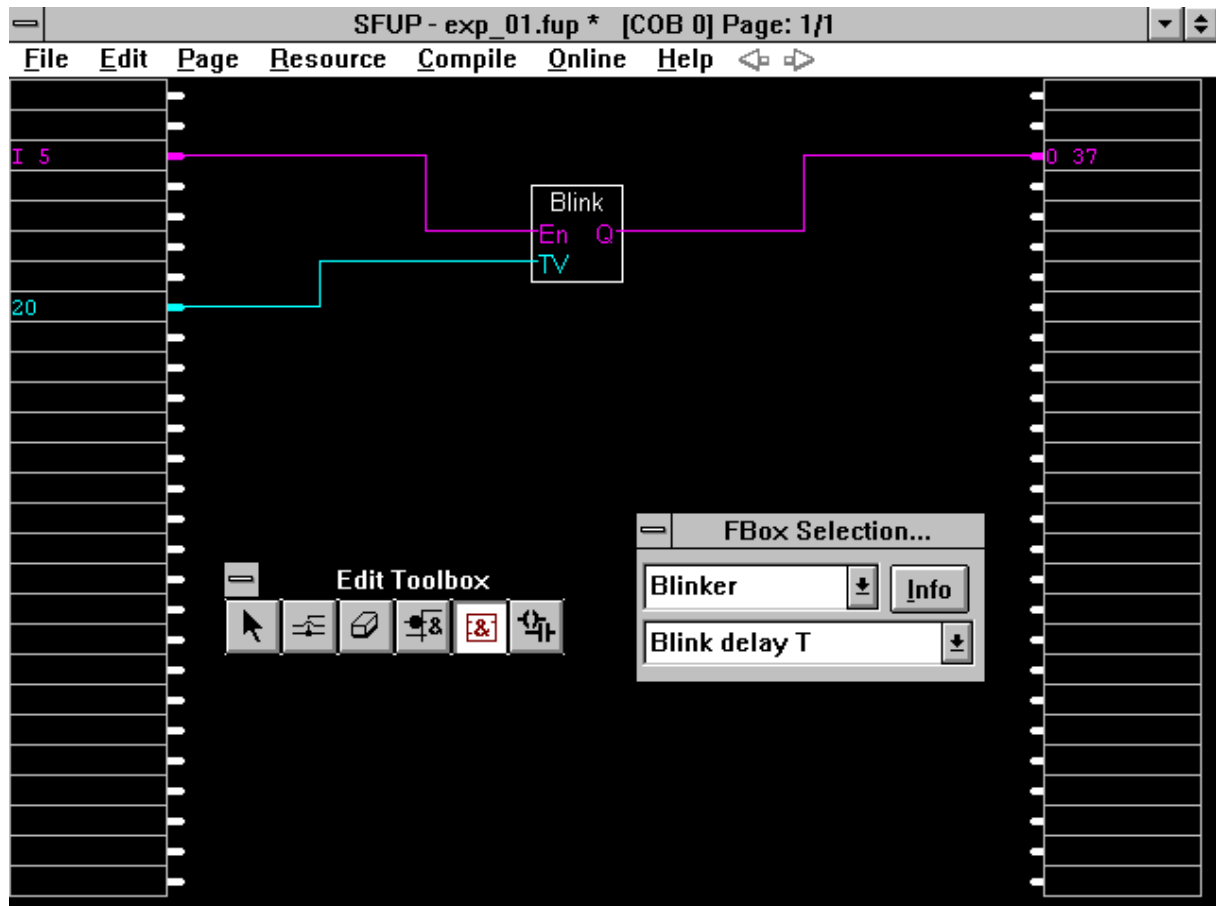
The file name is already entered on the header bar:



- 5.) We are now ready to edit a function plan user program.

We will start with a trivial example for clarity:

If input 5 is switched on, output 37 is to blink at the rate of 1 sec on, 1 sec off.



- 6.) When a new FUPLA is called, the toolbox is already running. If not, it can be selected with 'Edit' - 'Edit Tools'.
- 7.) Click on the "Functions" symbol in the edit toolbox. The 'FBox Selection...' dialog box appears at the bottom right. This window can also be moved using the mouse.

Clicking on the down-arrow at 'Analog modules' shows the list of available function families. (Another click on the same arrow makes the list disappear). We select the "Blinker" family.

The lower "Functions" box probably now contains 'Sample'. Click on the arrow to the right of it: all functions in the "Blinker" family appear. Here we select 'Blink delay T' (symmetrical blinker).

The mouse can now be used to position the function box as desired. By clicking the left-hand mouse key, the box is positioned approximately in the middle of the upper half of the screen. Other similar boxes can be positioned anywhere on the screen. Try it.

For our example we only need one 'Blink delay T' function box. Unwanted functions can be deleted with the "Eraser".

- 8.) The connections to inputs (left) and outputs (right) must now be drawn. This can be started from the function box or from the inputs and outputs.

Let's start at the function box.

Press the "Connection lines" button in the edit toolbox.

Move the small square cursor over the red connecting wire on the left side of the 'Blink' box at input "En", and click the left-hand mouse button. Move the mouse to the left and draw a line. With luck the line will go directly to an input connection. If so, click on the input connection with the left-hand mouse button to terminate the line, then click the right-hand mouse button to end the line. If you arrive between two input connections, you must put one or two angles in the line. This is done by clicking the left-hand mouse button. Try it.

Proceed in the same way for the blue input "TV" (time value).

The same procedure also applies on the output side (on the right of the box), blinker output "Q".

Notice that, when started from the function box, the line appears immediately as either red or blue (red for binary, blue for integer values). If the line starts outside the box at the left or right margin, it remains white until it connects to the function box, because the data type is unknown until then.

- 9.) That concludes graphical editing of the function plan for this simple example. The inputs and output must now be identified. At this point we could assign them from the Resource Manager, but an easier method is also available by assigning the input, the value and the output directly.

From the edit toolbox, select the arrow button and move the mouse into the text box on the left of the screen to the point where our previously drawn (red) input line begins, and click the left-hand mouse button. The keyboard can now be used to enter a type and value:

i 5 or I 5

It is important to leave a space between "i" and "5", otherwise "i5" is interpreted as a symbol (common error!).

We move the cursor with the mouse or the <↓> key onto the entry field and type from the keyboard

10

for 1 sec.

Using the mouse, we go to the output field and type

o 37 or O 37

It is important to leave a space between "o" and "37", otherwise "o37" is interpreted as a symbol (common error!).

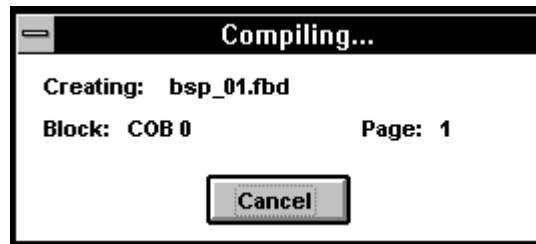
- 10.) Our function plan is now ready and can be compiled into a source program.

With 'File' - 'Save' we can store what has been edited so far, or we can attempt to compile the function plan directly.

(Saving is automatic if "Ask before saving" is switched off in the 'Make' - 'Make Option...' submenu).

We select 'Compile' from the menu bar and click on 'Compile' in the sub-menu.

A message box appears briefly:



If no other message follows, compilation was successful. The source file

EXP_01.FBD

has been generated, together with auxiliary files "exp_01.inc" and "exp_01.sym".

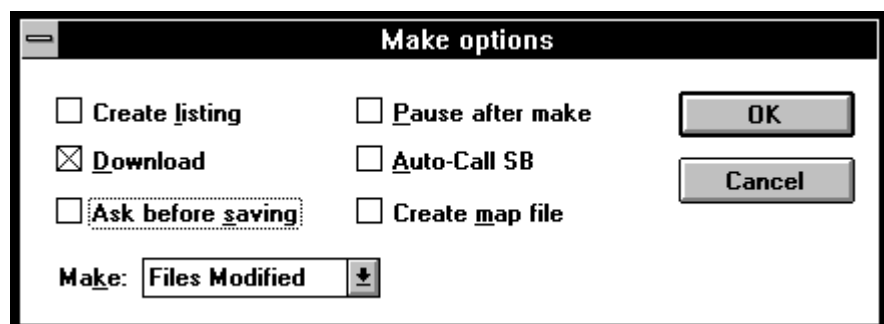
Any compilation errors, such as missing connections, inadmissible loops or missing addresses, would be listed and displayed here.

- 11.) Further processing of the program (i.e. assembling, linking, downloading, switching into run mode and selecting the online option) requires some definitions to be set when used for the first time.

For our first example we want to reach our goal as quickly as possible, i.e. to go with a single click from the function plan to the program running in the PCD.

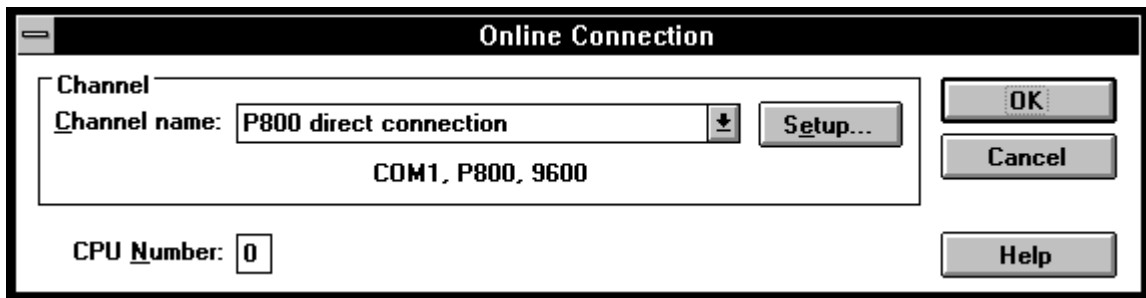
We go to the project manager.

Under 'Make' - 'Make Options...' we select the following



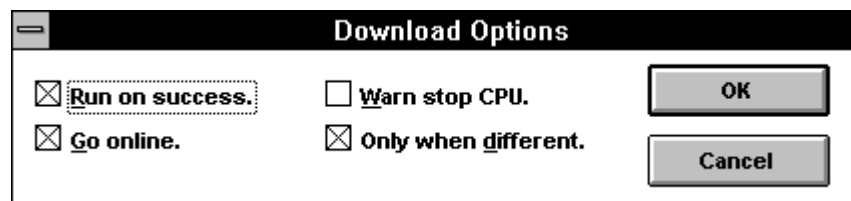
and confirm with 'OK'

At 'Online' - 'Connect Options...' we select the P800 protocol



and confirm with 'OK'.

From the 'Online' menu again, we select the following 'Download options...'



and confirm with 'OK'.

Each time these definitions are confirmed with 'OK' they are stored and remain valid for all subsequent operations.

After a few successful trials with the proposed settings, try changing them individually to gain a practical understanding of the meaning of individual definitions.

- 12.) If all settings have been made, go back into the FUPLA and click on 'Make' in the 'Compile' menu.

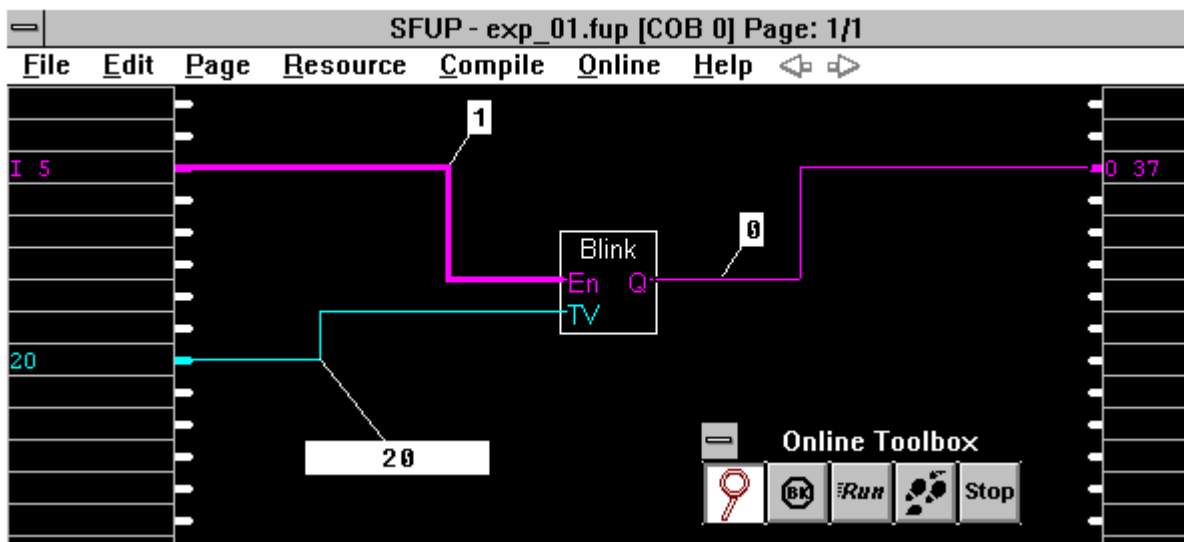
The function plan "exp-01.fup" is now compiled, if this has not already happened. Assembly then takes place, creating file "exp-01.obj". After linking we have the loadable file with the name of the project

"PG4_TEST.PCD".

This file is downloaded, the CPU is switched into Run and the online option is enabled.

- 13.) Program function can be tested with the PCD: If input 5 is switched on, output 37 will blink.
- 14.) The red connection line from FUPLA input (I 5) to the blinker input is thickened when input 5 turns on, and restored when input 5 is switched off. We can also see the output blinking just from the thickness of the red output connection.

The online toolbox has already been displayed. An online probe can be attached to the blue connection for time value entry and the value 10 will be displayed. The input and the output connections can also each be fitted with an online probe, so that the actual binary states can be viewed online.



Note:

As first sight, this whole procedure may seem rather confusing, especially the linking and the file names. A "proper" project normally consists of several individual files written or drawn with various editors and provided with different names. These files are then linked to form a loadable file with the name of the project and downloaded into the PCD. Our minimal program follows the same rules and is therefore rather overshadowed by the whole structure.

4.3.2 Viewing an existing function plan

It is assumed that the preceding example has been worked through exactly as suggested.

The example "EXP-01.FUP" is in sub-directory "c:\test\pg4-test". The program has been downloaded to the PCD and works in "Run".

The individual steps to view the program online are as follows:

- 1.) Run WINDOWS from DOS.
- 2.) Call the project library by double-clicking on the PG4 icon.
- 3.) Click on project "PG4_TEST" and open it.
- 4.) The project manager is displayed. Select file "EXP_01.FUP" and open the file by double-clicking or with the toolbar button.
- 5.) Our FUPLA file is displayed on the screen (assuming it actually is a FUPLA file).
- 6.) Click on 'Online' - 'Go online'. The 'Online Toolbox' appears. Reposition if necessary.

Attach online probes at the desired points.

If the following message appears:



this means that the function plan diagram on the screen no longer matches the downloaded .PCD file, because of a modification either in the function plan or in the .PCD file (planned or unintentional).

In this case it must be recompiled, and "Make" must be done again.

4.3.3 Changing an existing function plan

Even the smallest modification of a function plan requires recompiling and therefore also reassembly, linking and reloading.

In our trivial example "EXP-01.FUP" located in sub-directory "c:\test\pg4-test", the function box "Blink delay T" is to be repositioned horizontally.

The following steps must be done:

- 1.) Run WINDOWS.
- 2.) Run FUPLA by double clicking on the FUPLA icon and maximize it.
- 3.) Click on and open the project "PG4_TEST".
- 4.) The project manager is displayed. Select file "EXP_01.FUP" and open it by double-clicking or with the toolbar button.
- 5.) Our FUPLA file appears on the screen.
- 6.) For horizontal repositioning (1) of the "Blink delay T" function box, place the mouse pointer on the FBox and press the left-hand mouse button. The box is displayed in white and can be repositioned horizontally by dragging the mouse. When the mouse button is released, the box remains in its new position.

Since the box is now located at new coordinates on the work surface, the function plan must be recompiled.
- 7.) Recompile, assemble, download, put into "Run" mode and prepare the online options with 'Compile' 'Make'.

-
- (1) If the box is to be repositioned vertically, it must first be deleted, a new box should be placed in the new location and the connections layed out accordingly. Unfortunately, dragging currently functions only in the horizontal plane.

4.4 The function families of the FUPLA

Overview over the function families:

4.4.1	Binary	Binary functions
4.4.2	Flip-Flop	Flip-Flops
4.4.3	Counter	Counters
4.4.4	Time related	Time related functions
4.4.5	Blinker	Blinker
4.4.6	Integer	Integer arithmetic
4.4.7	Floating Point	Floating point arithmetic
4.4.8	Converter	Converters (binary-integer-floating point)
4.4.9	Indirect	Indirect addressing
4.4.10	Move-In/-Out	Move Data
4.4.11	Display	Displays
4.4.12	Graftec	GRAFTEC functions
4.4.13	Special	Special functions
4.4.14	Analog module	Read in and output analogue values
4.4.15	Regulation	Regulation (PID control).
4.4.16	User definable	User-defined functions
4.4.17	Communication	Serial communications (mode "D")
4.5	The functions of the KOPLA (Ladder diagram)	

A detailed description of each function can be found in a separate manual entitledl

"The FUPLA and KOPLA function families"

Order reference: 26/749 E

The following is a list of all functions, which also forms the contents list of the above manual: "The FUPLA and KOPLA function families".

Overview of functions in the individual function families

(arranged according to function and usage *)

4.4.1	Binary functions
4.4.1.1	And 2-10 inputs
4.4.1.2	Or 2-10 inputs
4.4.1.3	Xor 2-10 inputs
4.4.1.4	Move
4.4.1.5	Dynamize
4.4.1.6	High
4.4.1.7	Low
4.4.1.8	Not connected
4.4.1.9	Mux binary selection
4.4.1.10	Mux integer selection
4.4.1.11	Demux binary selection
4.4.1.12	Demux integer selection
4.4.1.13	I/O indirect
4.4.1.14	Flag indirect
4.4.1.15	Even 2-10 inputs
4.4.1.16	Odd 2-10 inputs
4.4.2	Flip-Flops
4.4.2.1	Toggle
4.4.2.2	Type D
4.4.2.3	Type RS dynamized
4.4.2.4	Type SR dynamized
4.4.2.5	Type JK
4.4.2.6	Type RS clocked
4.4.2.7	Type SR clocked
4.4.2.8	Type RS
4.4.2.9	Type SR

*) In FUPLA all functions are automatically arranged in alphabetical order.

4.4.3 Counters

- 4.4.3.1 Up with preset
- 4.4.3.2 Down with preset
- 4.4.3.3 Up
- 4.4.3.4 Up/down with preset
- 4.4.3.5 Up/down with preset and clear

4.4.4 Time related - Time function elements

- 4.4.4.1 On delay
- 4.4.4.2 Store delay
- 4.4.4.3 Exclusive pulse
- 4.4.4.4 Off delay
- 4.4.4.5 On/off delay
- 4.4.4.6 Off delay with reset
- 4.4.4.7 Pulse
- 4.4.4.8 Chronometer
- 4.4.4.9 Time (Hardware clock)
- 4.4.4.10 Start delay

4.4.5 Blinker

- 4.4.5.1 Blink delay T
- 4.4.5.2 Blink delay T0/T1
- 4.4.5.3 Sample

4.4.6	Integer arithmetic
4.4.6.1	Add
4.4.6.2	Subtract
4.4.6.3	Multiply
4.4.6.4	Divide
4.4.6.5	Square root
4.4.6.6	Average
4.4.6.7	Constant
4.4.6.8	Absolute
4.4.6.9	Bitwise and
4.4.6.10	Bitwise or
4.4.6.11	Bitwise exclusive or
4.4.6.12	Bitwise invert
4.4.6.13	Is equal to
4.4.6.14	Is greater or equal to
4.4.6.15	Is greater than
4.4.6.16	Is smaller or equal to
4.4.6.17	Is smaller than
4.4.6.18	Is zero
4.4.6.19	Limit
4.4.6.20	Maximum
4.4.6.21	Minimum
4.4.6.22	Move
4.4.6.23	Move when enabled
4.4.6.24	Move and store
4.4.6.25	Switch
4.4.6.26	Multiplexer with binary selection
4.4.6.27	Multiplexer with integer selection
4.4.6.28	Demultiplexer with binary selection
4.4.6.29	Demultiplexer with integer selection
4.4.6.30	Shift left
4.4.6.31	Shift right
4.4.6.32	Rotate left
4.4.6.33	Rotate right
4.4.6.34	Register indirect
4.4.6.35	T/C indirect
4.4.6.36	Not connected

4.4.7	Floating point arithmetic
4.4.7.1	Add
4.4.7.2	Subtract
4.4.7.3	Multiply
4.4.7.4	Divide
4.4.7.5	Square root
4.4.7.6	Average
4.4.7.7	Constant
4.4.7.8	Absolute
4.4.7.9	Sine
4.4.7.10	Cosine
4.4.7.11	ARC tangent
4.4.7.12	Natural exponent
4.4.7.13	Natural log
4.4.7.14	Is equal to
4.4.7.15	Is greater or equal to
4.4.7.16	Is greater than
4.4.7.17	Is smaller or equal to
4.4.7.18	Is smaller than
4.4.7.19	Is zero
4.4.7.20	Limit
4.4.7.21	Maximum
4.4.7.22	Minimum
4.4.7.23	Move
4.4.7.24	Move when enabled
4.4.7.25	Move and store
4.4.7.26	Switch
4.4.7.27	Multiplexer with binary selection
4.4.7.28	Multiplexer with integer selection
4.4.7.29	Demultiplexer with binary selection
4.4.7.30	Demultiplexer with integer selection
4.4.7.31	Not connected

4.4.8 Converters (binary-integer-floating point)

- 4.4.8.1 Bin to int 1-8
- 4.4.8.2 Bin to int 1-24
- 4.4.8.3 Bin to int quick (PCD format)
- 4.4.8.4 Bin to int reverse quick (PCA format)

- 4.4.8.5 Int to bin 1-8
- 4.4.8.6 Int to bin 1-24
- 4.4.8.7 Int to bin quick (PCD format)
- 4.4.8.8 int to bin reverse quick (PCA format)

- 4.4.8.9 BCD to int
- 4.4.8.10 BCD to int quick (PCD format)
- 4.4.8.11 BCD to int reverse quick (PCA format)

- 4.4.8.12 Int to BCD
- 4.4.8.13 Int to BCD quick (PCD format)
- 4.4.8.14 Int to BCD reverse quick (PCA format)

- 4.4.8.15 1-bit to int with shift
- 4.4.8.16 1-bit to int LSB

- 4.4.8.17 Int to 1-bit with shift
- 4.4.8.18 Int LSB to 1-bit

- 4.4.8.19 Float to int
- 4.4.8.20 Int to float

4.4.9 Indirect addressing

- 4.4.9.1 Copy to outputs
- 4.4.9.2 Read from inputs
- 4.4.9.3 Copy to flags
- 4.4.9.4 Read from flags
- 4.4.9.5 Copy to registers integer
- 4.4.9.6 Read from registers integer
- 4.4.9.7 Copy to registers float
- 4.4.9.8 Read from registers float
- 4.4.9.9 Copy to Timer/Counter
- 4.4.9.10 Read from Timer/Counter
- 4.4.9.11 Timer with indirect addressing
- 4.4.9.12 Counter with indirect addressing
- 4.4.9.13 Read logic state from Timer/Counter

4.4.10 Move Data

- 4.4.10.1 Move-In Bit
- 4.4.10.2 Move-Out Bit
- 4.4.10.3 Move-In Nibble
- 4.4.10.4 Move-Out Nibble
- 4.4.10.5 Move-In Digit
- 4.4.10.6 Move-Out Digit
- 4.4.10.7 Move-In Byte
- 4.4.10.8 Move-Out Byte
- 4.4.10.9 Move-In Word
- 4.4.10.10 Move-Out Word

4.4.11 Displays

- 4.4.11.1 Display module PCA2.D12
- 4.4.11.2 Display module PCA2.D14
- 4.4.11.3 Display module PCD2.F510 for numeric displays
- 4.4.11.4 Display module PCD2.F510 for text displays

4.4.12 GRAFTEC functions

- 4.4.12.1 Load timer
- 4.4.12.2 Load timer conditional
- 4.4.12.3 Load counter
- 4.4.12.4 Load counter conditional
- 4.4.12.5 Increment counter
- 4.4.12.6 Decrement counter
- 4.4.12.7 Timer is zero
- 4.4.12.8 Counter is zero
- 4.4.12.9 End of transition
- 4.4.12.10 Wait time
- 4.4.12.11 Wait pulse

4.4.13 Special functions

- 4.4.13.1 Watch dog
- 4.4.13.2 Watch dog enable

4.4.14 Analog modules

4.4.14.1	PCD2.W1	Analog Input module PCD2.W1 (12 Bit)
4.4.14.2	PCD2.W2	Analog Input module PCD2.W2 (10 Bit)
4.4.14.3	PCD2.W4	Analog Output module PCD2.W4 (8 Bit)
4.4.14.4	PCD2.W5	Analog Input/Output module PCD2.W5 (12 Bit)
4.4.14.5	PCD4.W1	Analog Input/Output module PCD4.W1 (12 Bit)
4.4.14.6	PCD4.W3	Analog Input module PCD4.W3 (12 Bit + Sign)
4.4.14.7	PCD4.W4	Analog Output module PCD4.W4 (8 Bit)
4.4.14.8	PCD6.W1	Analog Input/Output module PCD6.W1 (12 Bit)
4.4.14.9	PCD6.W3	Analog Input module PCD6.W3 (12 Bit + Sign)
4.4.14.10	PCD6.W4	Analog Output module PCD6.W4 (8 Bit)

4.4.15 Regulation (PID control)

4.4.15.1 PID FBox

4.4.16 User-defined functions

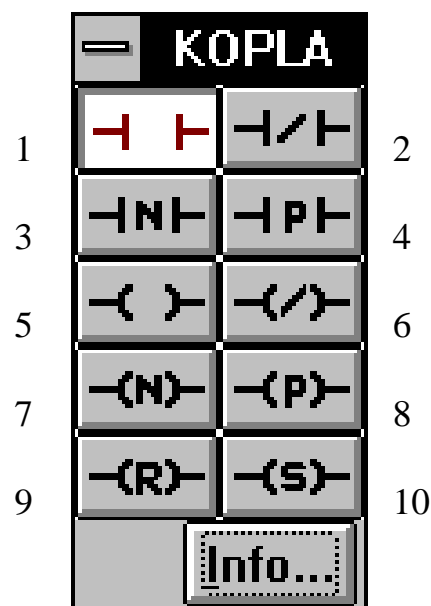
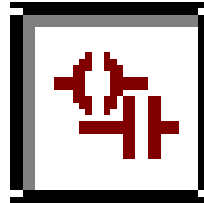
4.4.16.1	Call PB
4.4.16.2	Call FB
4.4.16.3	Call SB
4.4.16.4	User block 1
4.4.16.5	User block 2
4.4.16.6	User block 3
4.4.16.7	User block 4
4.4.16.8	User block 5

4.4.17 Serial communications (mode "D")

4.4.17.1	Interface parameters
4.4.17.2	Interface parameters external
4.4.17.3	Receive 1-20 I/O/F
4.4.17.4	Receive 1-20 R/T/C/Clock
4.4.17.5	Transmit 1-20 I/O/F
4.4.17.6	Transmit 1-20 R/T/C/Clock
4.4.17.7	Receive I/O/F multiple
4.4.17.8	Receive R/T/C multiple

4.5 The function families of the KOPLA (Ladder diagram)

The call of the KOPLA is done directly from the FUPLA toolbox and not via the "FBox Selection" menu.

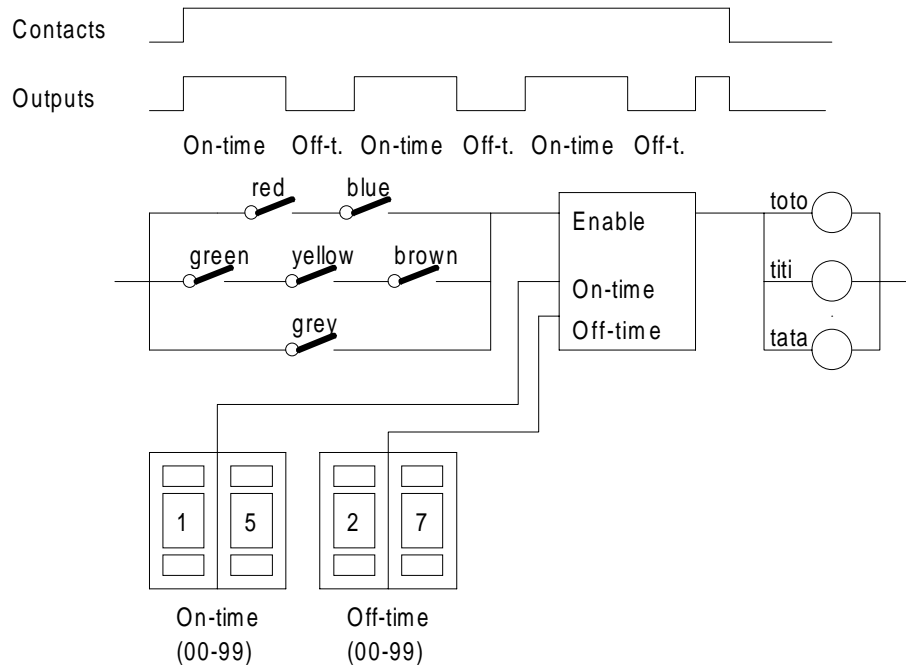


- | | | |
|--------|------------------|---------------|
| 4.5.1 | Contact | |
| 4.5.2 | Contact closed | |
| 4.5.3 | Contact negative | Negative edge |
| 4.5.4 | Contact positive | Positive edge |
| 4.5.5 | Coil | |
| 4.5.6 | Coil closed | |
| 4.5.7 | Coil negative | Negative edge |
| 4.5.8 | Coil positive | Positive edge |
| 4.5.9 | Coil reset | |
| 4.5.10 | Coil set | |

Notes :

4.6 Introductory example (FUPLA)

As a first step, a blinker is to programmed according to the following diagram:



The example is designed for the PCD4 demonstration model V-PCX 20.

The blinker becomes active when either the switches "red" and "blue" or "green" and "yellow" and "brown" or "grey" are closed. The three outputs "toto", "titi" and "tata" blink together, for as long as at least one of the three inputs conditions is satisfied.

Input and output element addresses are as follows:

switch "red"	→	PCD input 1
switch "blue"	→	PCD input 2
switch "green"	→	PCD input 3
switch "yellow"	→	PCD input 4
switch "brown"	→	PCD input 5
switch "grey"	→	PCD input 6
output "toto"	→	PCD output 34
output "titi"	→	PCD output 36
output "tata"	→	PCD output 38

The on and off times are entered via the two 2-digit BCD switches. The timebase in FUPLA is always 100ms. The on and off times can therefore each be selected between 0.1 and 9.9 sec.

The BCD switch "on time" is wired to PCD inputs 16 - 23.
The BCD switch "off time" is wired to PCD inputs 24 - 31.

The main steps in the task are:

- edit the function plan
- identify the elements with symbolic names
- define the elements in the "Resource" menu
- compile
- assemble, link, download
- commissioning
- online viewing of the task in the function plan

Two programming methods are offered for this example:

- standard function plan
- KOPLA (ladder diagram)

We will discuss both methods and begin with the standard function plan with AND and OR gates for the linkage.

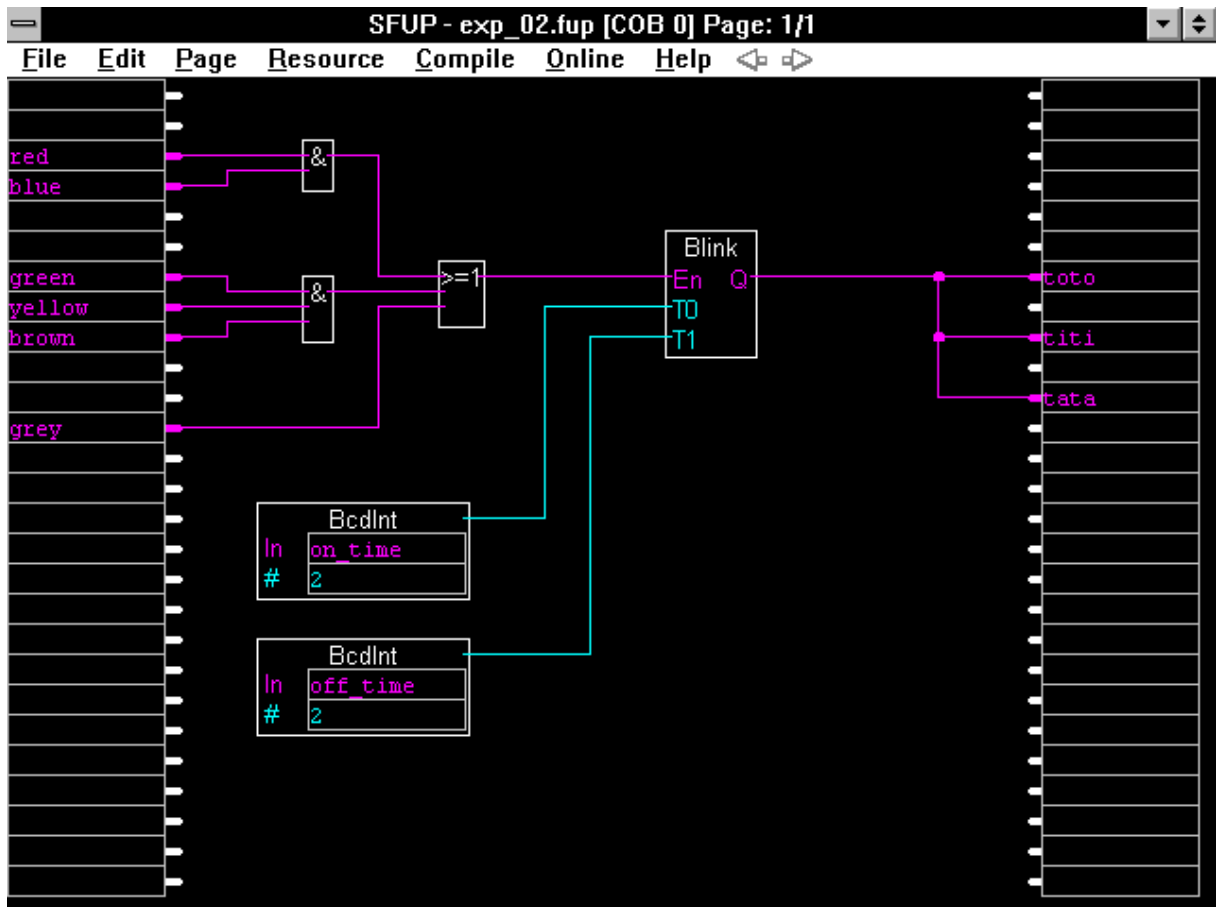
4.6.1 Standard function plan (FUPLA)

The new project is given the name "INTRO". Our FUPLA program which is located in this project is given the name "EXP_02".

The example is to be structured as an stand-alone executable program in COB 10.

The dynamic variables are to be left at their default values.

The function plan diagram shown below should be created in approximately this form.



Procedure for graphical editing of the function plan:

- Execute WINDOWS and then FUPLA from the Program Manager. FUPLA is displayed full screen.
- The project "INTRO" is opened with 'File' - 'New...'.
• Switch to the project manager and use 'File' - 'New...' to define file "EXP_02" as an "FBD/LD" file. A trial comment is also entered for this file.
- Double-click on the file name "EXP-02" in the "Files in Project" list (or use the toolbar button) to select the FUPLA and display it full screen.
- Display the edit toolbox with 'Edit' 'Edit tools' and drag it to the lower margin.
- Display the function box list. From the window 'FBox Selection...' select the 'Blinker' family, and select function box 'Blink delay T0/T1' "Blink delay T0/T1" has separate value inputs for "T0" and "T1".
- Position the "Blink delay T0/T1" box approximately as shown with a click on the left-hand mouse key.
- Go back with the mouse to the 'FBox Selection...' window, select the 'Binary' family function 'And 2-10 inputs'.
- With the mouse, position the AND gate approximately as shown. The first click positions the box. Now the number of inputs between 2 and 10 can be selected by moving the mouse downwards. Two inputs are required. The second click fixes the box with the displayed number of inputs.
- If the right-hand mouse button is pressed after the first click, the AND gate can be repositioned.
- The same procedure applies for the next AND gate, but this time it needs three inputs.
- Go back with the mouse to the 'FBox Selection...' window and select the function 'Or 2-10 Inputs' from the same 'Binary' family.
- Position this OR gate as shown. The procedure is the same as for the AND gates.
- It is now a question of converting the inputs from both BCD switches to an integer value. Select the 'Converter' family.
- From the functions in this family select 'BCD to int quick'.
- Position the two FBoxes approximately as shown.
- All the function boxes necessary for the example are on the screen. It is now a question of creating the connections. Select the "Connection" function from the toolbox.
- Start with the inputs. Move the small square cursor to the input connection (left-hand margin). Now click the left-hand mouse button. Drag the mouse to the upper AND gate: the line follows it. On reaching the AND gate, first click the left button to signal the end of this movement and then click RIGHT button to terminate the connection.

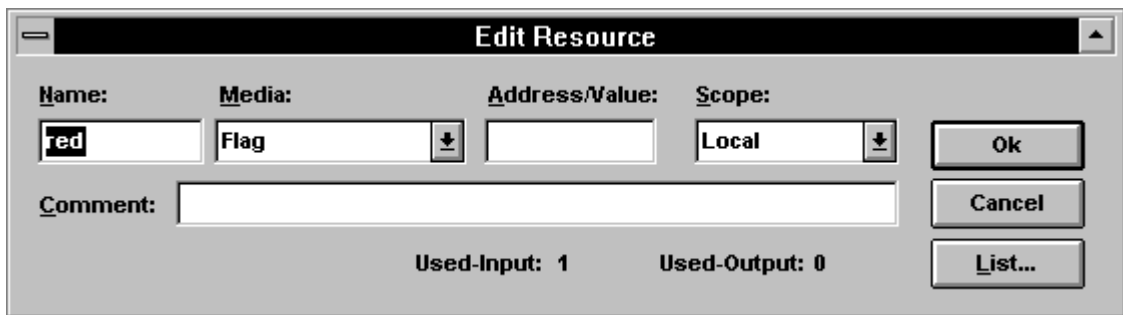
- If the direction of the line is to be changed, click on the desired position with the left-hand mouse button. It is now possible to draw the connection further in another direction.
- The same procedure applies for all other connections. The colour of the lines is set automatically, depending on whether they are binary (red) or integer (blue) values.
- With the outputs, a line can first be dragged from the blinker function box output to a FUPLA output. A branch can now be attached to any point on this line by clicking on the left-hand mouse button and then dragging.
- The drawing of the function plan diagram is now complete. Now the I/Os and the two converter FBoxes must be labelled.
- To exit connecting line mode, either click on the right-hand mouse button or select another function with the left-hand mouse button.
- Label the diagram as shown in the example. Start with "red". Move the mouse pointer into the text field and click on the left-hand button, "red" must be typed from the keyboard.
- The easiest way to reach the next field, in this case the "blue" field, is to use the cursor arrow keys (it is also possible with the mouse).
- To reach the outputs, use the mouse to move the cursor.
- The converter FBox fields are labelled by moving the mouse indicator inside the top or bottom box and clicking with the left-hand button. The easiest way to reach the other boxes is again to use the cursor arrow keys.
- The graphical part of programming has now been concluded. We save what has been edited so far with 'File' - 'Save'.

Procedure for defining individual resources:

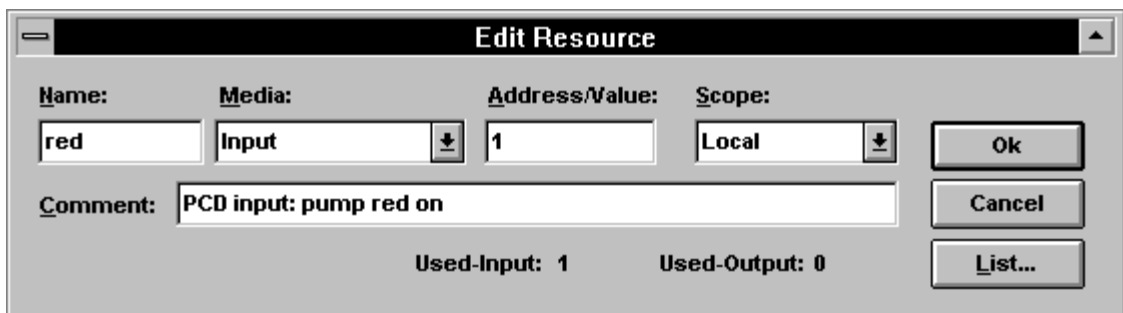
Select the arrow from the toolbox. We will start with the "red" input. The cursor is moved onto this input field and clicked on. Now select "Resource" from the menu bar.



The 1st line of the menu window reads "About label: red". Click on it to display the detailed window on the resource "red". This detailed window can also be called by double-clicking on the label "red".



Complete the window and confirm with "OK" or <CR>.



The same procedure should be carried out for all resources. When all of them have been defined, execute 'Save'.

'Resource' - 'Resource list' now enables the whole list to be displayed.

Quit!	Options!	Symbols			
red	I	1	Local	PCD input: pump red	
blue	I	2	Local	PCD input: pump blue	
green	I	3	Local	PCD input: valve green	
yellow	I	4	Local	PCD input: valve yellow	
brown	I	5	Local	PCD input: valve brown	
grey	I	6	Local	PCD input: manual	
on_time	I	16	Local	Switch-on time of the blinker	
off_time	I	24	Local	Switch-off time of the blinker	
toto	O	34	Local	PCD output: lamp "run"	
titi	O	36	Local	PCD output: display status	
tata	O	38	Local	PCD output: supervision	
	K	2	Local		
	K	2	Local		

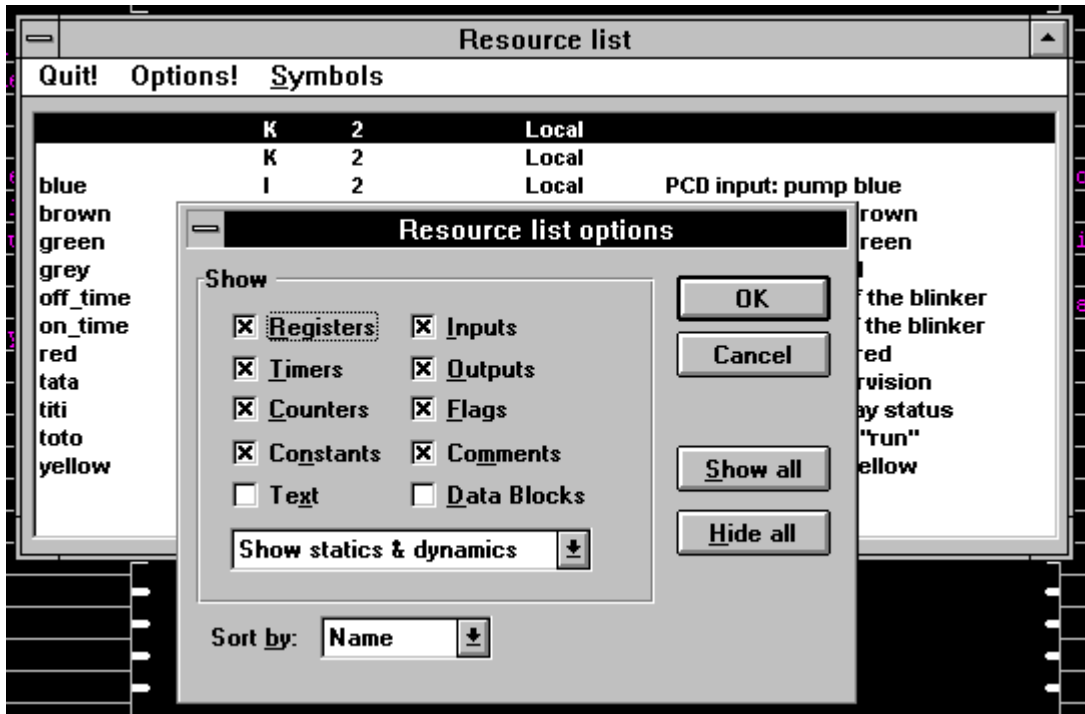
Details of each item can be displayed by displaying the resource list, and double-clicking on the desired resource.

The 'Edit Resource' dialog box is open, showing the following fields:

- Name:** blue
- Media:** Input
- Address/Value:** 2
- Scope:** Local
- Comment:** PCD input: pump blue
- Used-Input:** 1
- Used-Output:** 0

Buttons: Ok, Cancel, List...

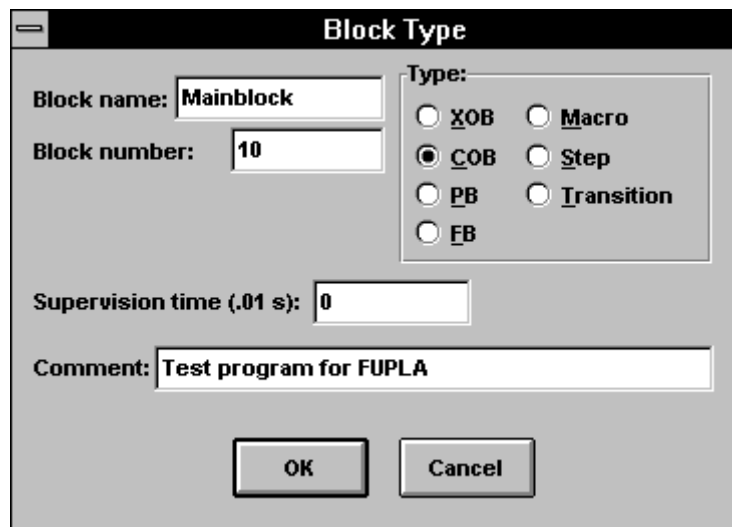
'Options' enables the resource list to be structured according to personal wishes.



'Cancel' and 'Quit' are used to exit the resource manager.

In this example we will put the executable code in COB 10.

If no COB is specified, COB 0 is used by default. This can be changed from the "Block Type" dialog box, displayed by the 'File' 'Block Properties...' command. This contains a 'Block number' field, in which <10> is entered from the keyboard. Comments can also be edited. Confirm and exit with "OK".



Editing is now complete. Save the updated file "exp-02.fup" once again with 'File' - 'Save'.

Compiling, assembling, linking, downloading and commissioning:

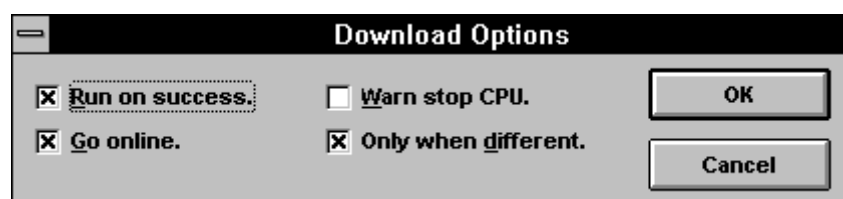
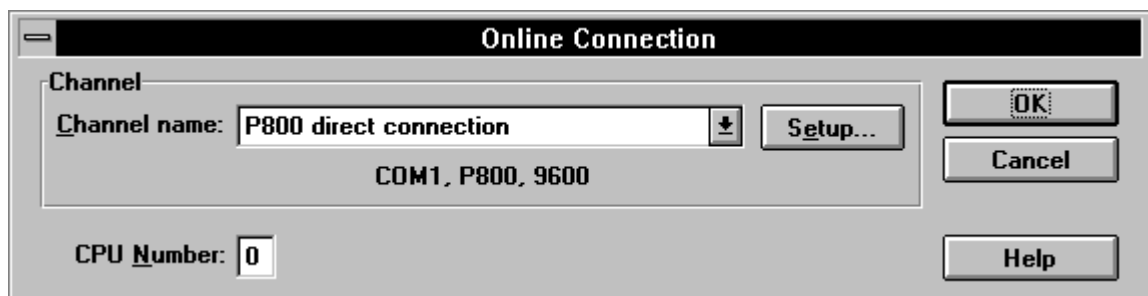
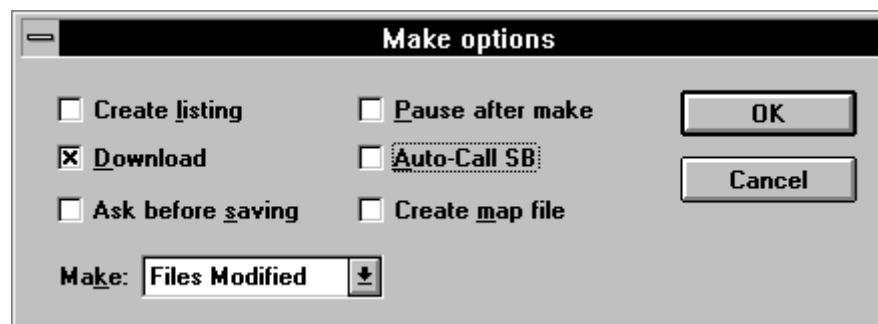
By simply clicking on 'Compile' (from the menu of the same name 'Compile') the source file "exp-02.src" is generated.

While it is compiling, a message box appears briefly:



If no other message appears after the compilation, the source file has been successfully created.

With 'Make', a single button can now be pressed to assemble, link and download the program, place it in RUN and switch it online. In order to use 'Make', the settings in the sub-menus 'Compile' - 'Make...', 'Online' - 'Online options...' and 'Online' - 'Download options...' should be established according to requirements; otherwise the settings already given in section 4.2.6 and 4.2.7 should remain valid.

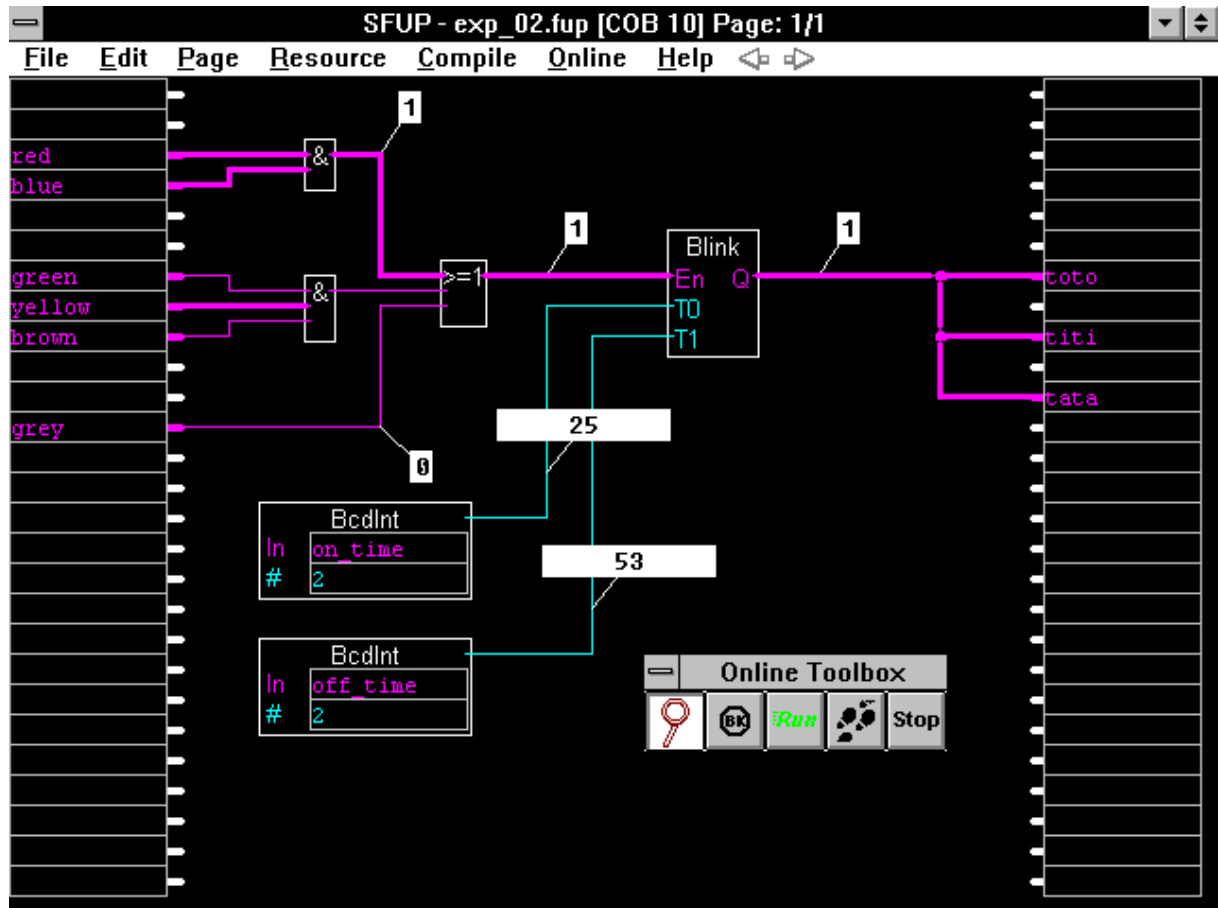


If the settings are correct, it is possible, as described above to click on the 'Make' submenu from the 'Compile' menu. The previously compiled program is now assembled, linked, downloaded, switched into RUN and the online option is enabled.

The program can now be tested in the PCD itself and on the screen.

The test in the PCD hardly needs explanation. During testing it may be useful to change the addressing of elements to "absolute" from the 'Resource' - 'Set up...' dialog box This shows the actual values of "red" and "blue".

On the screen, all the binary (red) connections appear as heavy lines if switched on, or fine lines if switched off. Online probes can now be placed on the diagram at points of interest, so that the program's function can be viewed.



Modifying or adding to a function plan

As an exercise, a few functions will be added to the example "EXP_02". You may want to view online the reading of the on and off times, and the display of the current date/time on the PCA2.D14.

Based on the example "exp_02", we will create a new program: "exp_03". (We want to leave the old program in its original form.)

Exit online mode for "exp_02". Go back to the project manager and type 'File' - 'Copy...'. Then change "exp_02" to "exp_03", adapt the comment and enter 'OK'.

To make more space available for additional functions, the function plan diagram which has already been created may need to be contracted and moved to the left. All function boxes and all vertical connections can be moved (dragged) horizontally with the mouse. Move the mouse pointer onto the line or box to be repositioned, press and hold down the left-hand mouse button and drag it to the new position. For FBoxes containing label fields, e.g. in the two converters (BCD to int quick), the arrow should be inside the box and also outside the label field. Please try this.

Now we bring out the edit toolbox again with 'Edit' 'Edit tools'. To view the reading of the blinker times, select the function 'Chronometer' (stopwatch) from the 'Time related' family and position two of these function boxes roughly as in the following diagram. The "Chronometer" functions are fully explained in the help texts, but a brief description is given here.

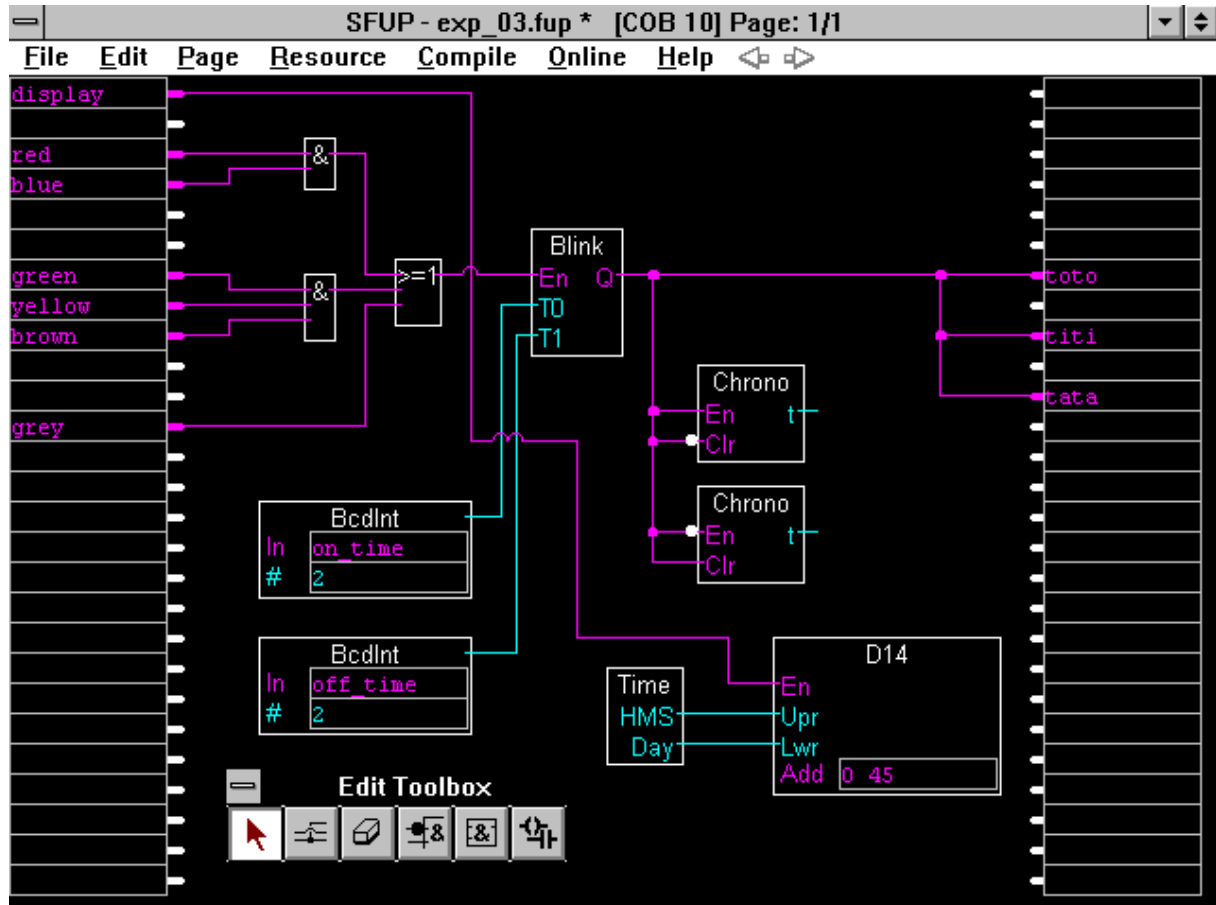
If input "En" (Enable) = H, the "Chronometer" runs. If input "En" = L, the clock stops. If input "Clr" (Clear) = H, the clock is reset to 0. The clock can be read at output "t".

In our example the two "Chronometer" functions are coupled at the blinker output so that at any time one "Chronometer" is running and the other is being reset. This is achieved by the inverter function, which can be selected from the toolbox. In the function plan, the inverter function is clicked directly onto the function box or an input/output. The inverter function is identified by a large white dot on the screen.

Leave the outputs of both "Chronometer" functions open. In online operation it is possible to attach probes to them, which show the duration in seconds since the last "Clear". It would also be possible to connect both outputs to PCD registers, so that these values can be used elsewhere in the program.

With the 'Time' function from the 'Time related' family, the internal hardware clock of the PCD can be incorporated into the program. The upper output "HMS" shows the hours, minutes and seconds, and the lower output "Day" shows the day of week and week number.

The efficient 'D14 Module' function from the 'Display' family can be used to incorporate the PCD2.D14 module directly into the user program, i.e. without a lengthy selection routine.



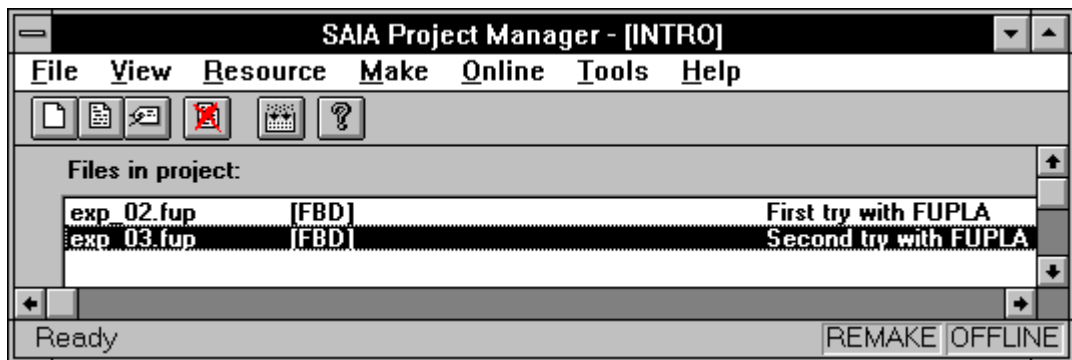
The value to be displayed in the upper 6 digits of the display is placed at input "Upr"; the value to be displayed in the lower 6 digits is placed at input "Lwr". The display is activated by the binary input "En" (Enable), H = active, L = inactive. The "En" input can, for example, be coupled to a PCD input. If the "D14 Module" is to be active continuously, the function 'High' from the 'Binary' family can be connected to input "En". Alternatively, it is also possible to make the connection via an undesignated input field. (An undesignated input field is always high.)

In address field "Add", the lowest of 3 consecutive PCD outputs is specified.

- O 45 means: O 45 → "Clock" input of D14
- O 46 → "Data" input of D14
- O 47 → "Enable" input of D14

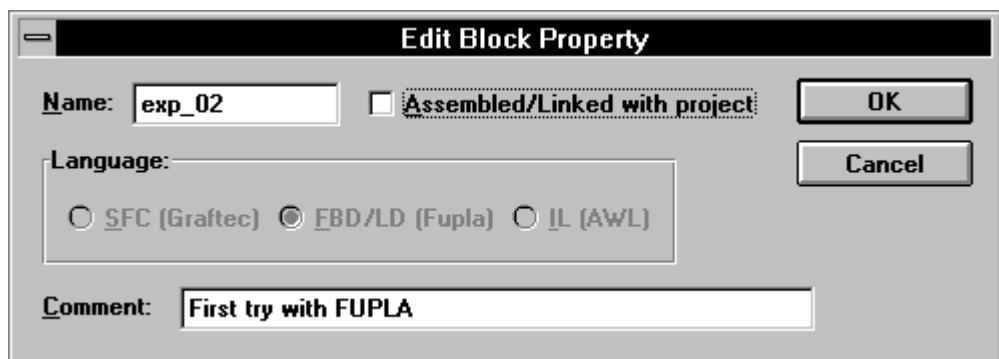
In our example the "D14 Module" with input "Display", defined in the "Resource Manager" as input 7, is activated. In the following it is also shown that connections can cross over each other if required.

The modified and expanded function plan is saved. It is then recompiled, assembled, linked, downloaded and switched into "Run" mode with 'Make' and function plan is switched online.

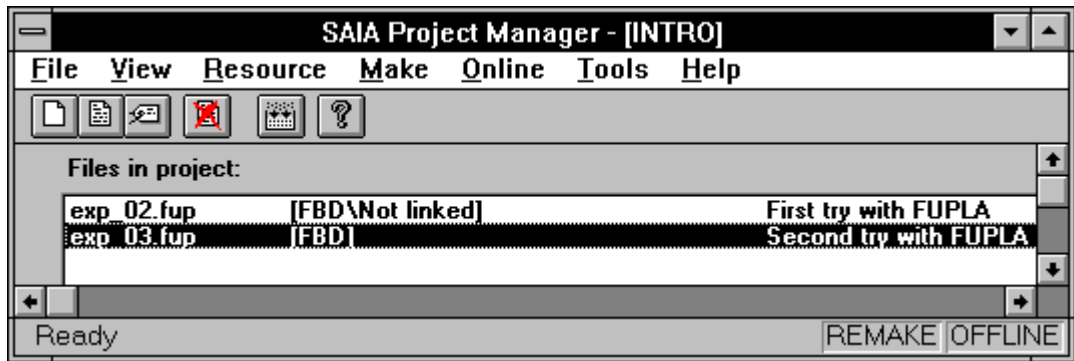


Our project "INTRO" now has both FUPLA programs "exp_02" and "exp_03". If a "Make" is executed after compiling both files, both files would be linked and our project "INTRO" would contain both of them. This would, of course, be illogical.

We will only be working with the new, expanded example "exp_03". The old file "exp_02" must be "switched off". We do this with 'File' - 'Rename/Properties...', during which the "Assembled/Linked with project" option is switched off.

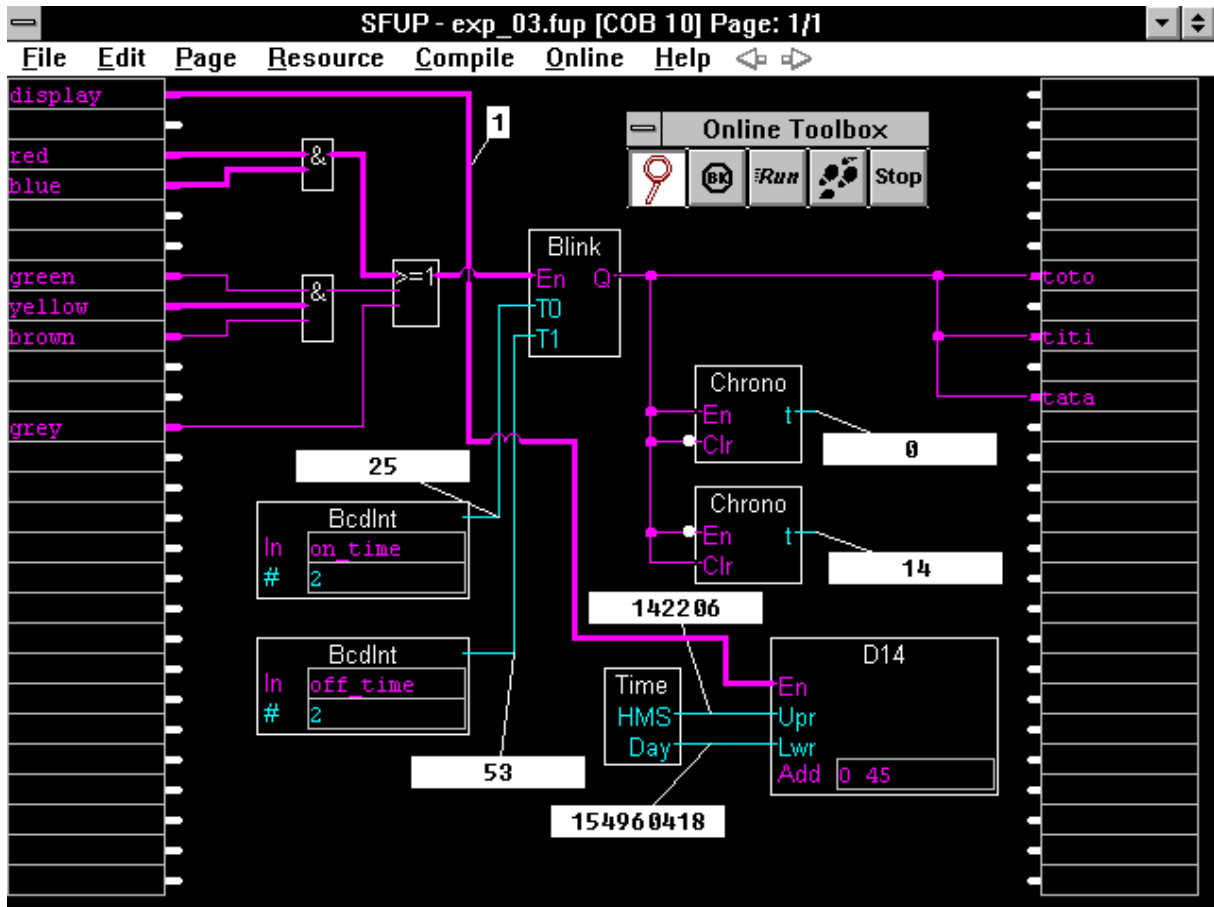


In the "Files in project" window of the project manager, the old program "exp_02_d" is now listed as [FBD/Not linked]. This means that this part of the program will not be included in any "Make".



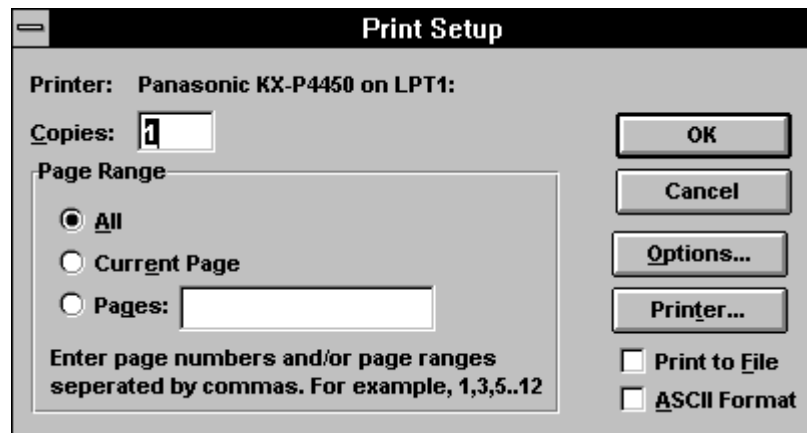
The new program can now be compiled in the usual way and 'Make' executed afterwards.

The following illustration shows the expanded function plan in online operation with online probes.

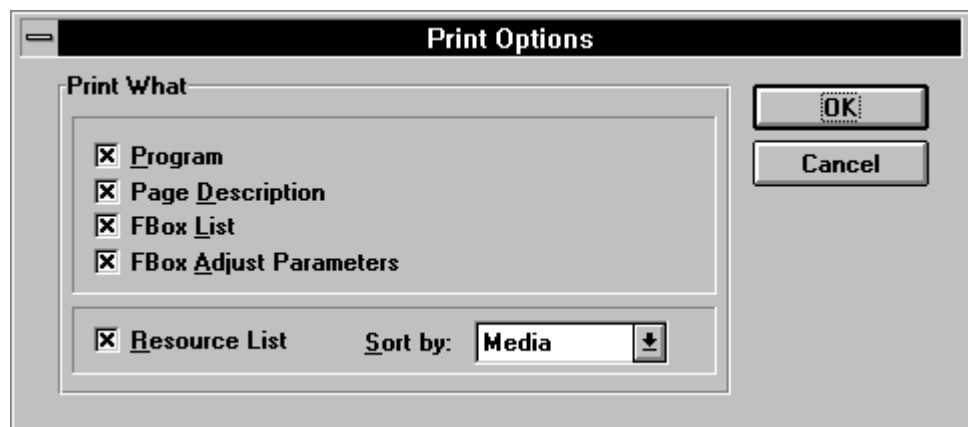


Printing the FUPLA diagram

Selecting 'File' - 'Print' displays the following menu window.



The 'Options...' switch opens the following window. We want to print everything.



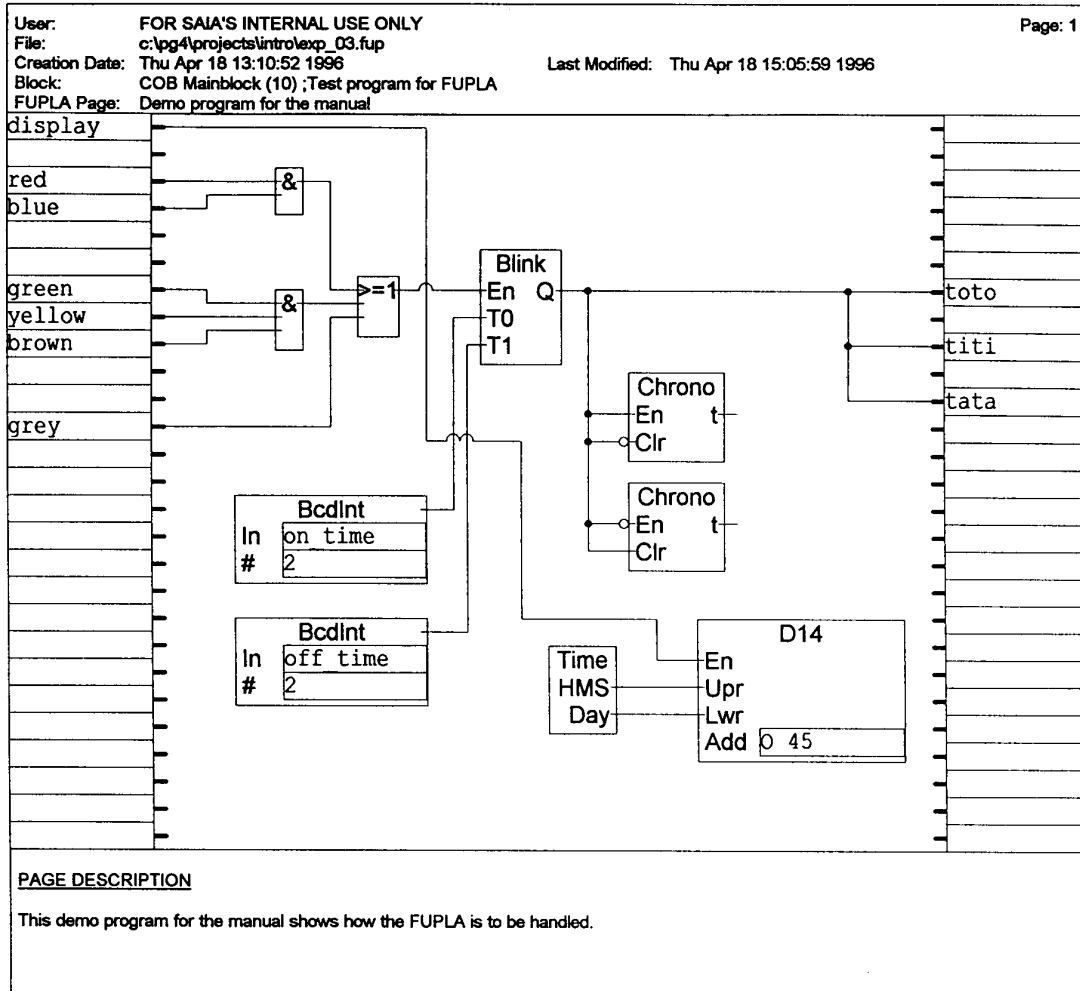
If neither 'Print to File' nor 'ASCII Format' is switched on in the "Print Setup" window, printing takes place in proper pixel format, in portrait or landscape style, as the following 4 pages show.

If 'ASCII Format' is selected, 'Print to File' is also switched on. After 'OK' you will be asked to enter a file name of the text file. After entering

EXP_03.TXT and 'OK'

an ASCII file will be created, which any printer can print easily. A print-out can be seen on the two last pages of this section.

For both formats, the individual FUPLA pages (1 page only in our example) appear first, followed by a list of all resources used, including symbolic names and any comments.

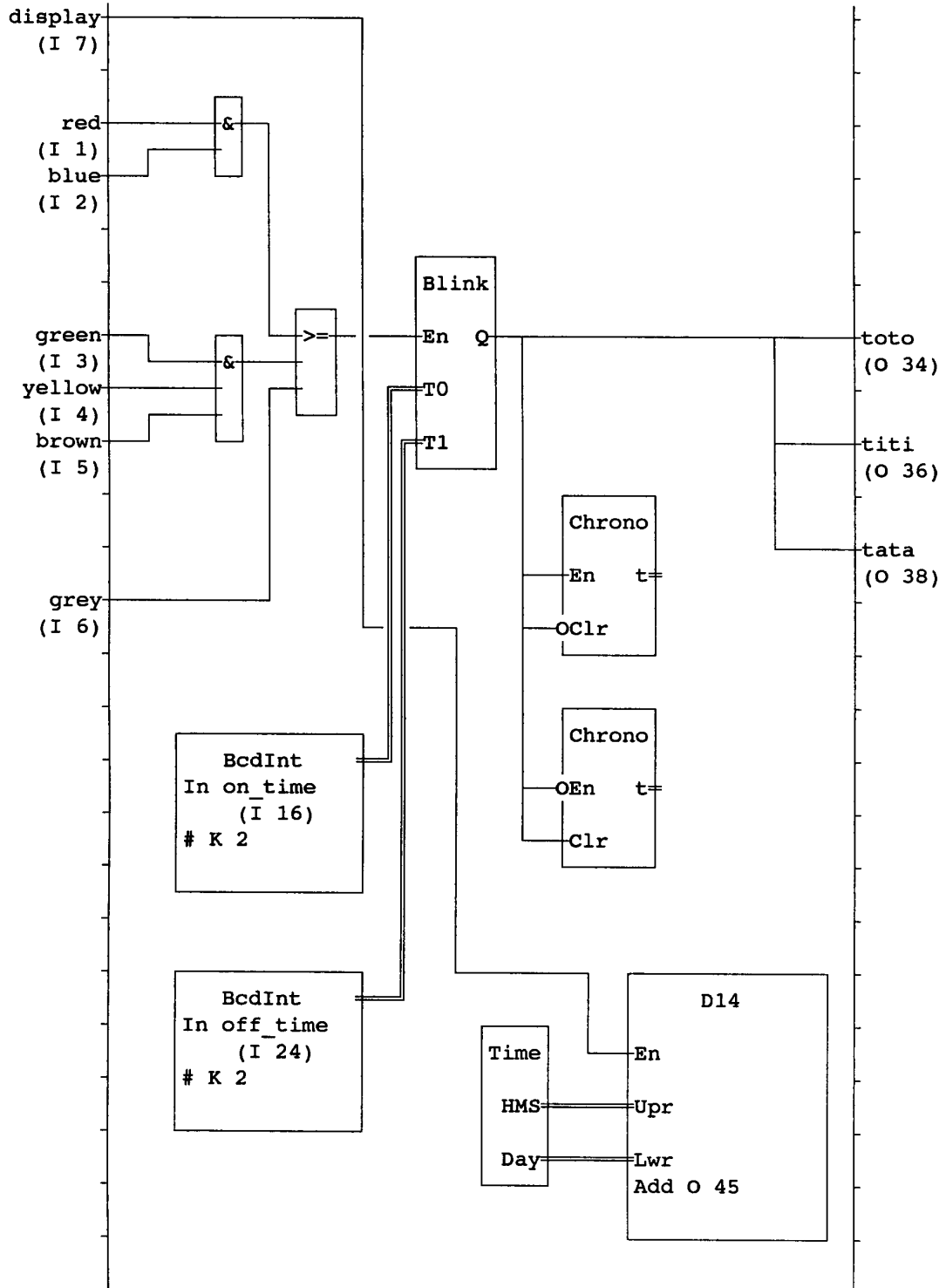


<u>SYMBOL NAME</u>	<u>RESOURCE</u>	<u>COMMENT</u>
red	I 1	PCD input: pump red
blue	I 2	PCD input: pump blue
green	I 3	PCD input: valve green
yellow	I 4	PCD input: valve yellow
brown	I 5	PCD input: valve brown
grey	I 6	PCD input: manual
display	I 7	Enable input for the D14 display module
on_time	I 16	Switch-on time of the blinker
off_time	I 24	Switch-off time of the blinker
toto	O 34	PCD output: lamp "run"
titu	O 36	PCD output: display status
tata	O 38	PCD output: supervision
	O 45	
	K 2	
	K 2	
Mainblock	COB 10	Test program for FUPLA
		*** END OF RESOURCE LIST ***

File name: c:\pg4\projects\intro\exp_03.fup

Page: Page#: 1 Comment: Demo program for the manual

Description: This demo program for the manual shows how the FUPLA is to be handled.



SYMBOL NAME	RESOURCE	COMMENT
=====	=====	=====
red	I 1	PCD input: pump red
blue	I 2	PCD input: pump blue
green	I 3	PCD input: valve green
yellow	I 4	PCD input: valve yellow
brown	I 5	PCD input: valve brown
grey	I 6	PCD input: manual
display	I 7	Enable input for the D14 display module
on_time	I 16	Switch-on time of the blinker
off_time	I 24	Switch-off time of the blinker
toto	O 34	PCD output: lamp "run"
titi	O 36	PCD output: display status
tata	O 38	PCD output: supervision
	O 45	
	K 2	
	K 2	
Mainblock	COB 10	Test program for FUPLA

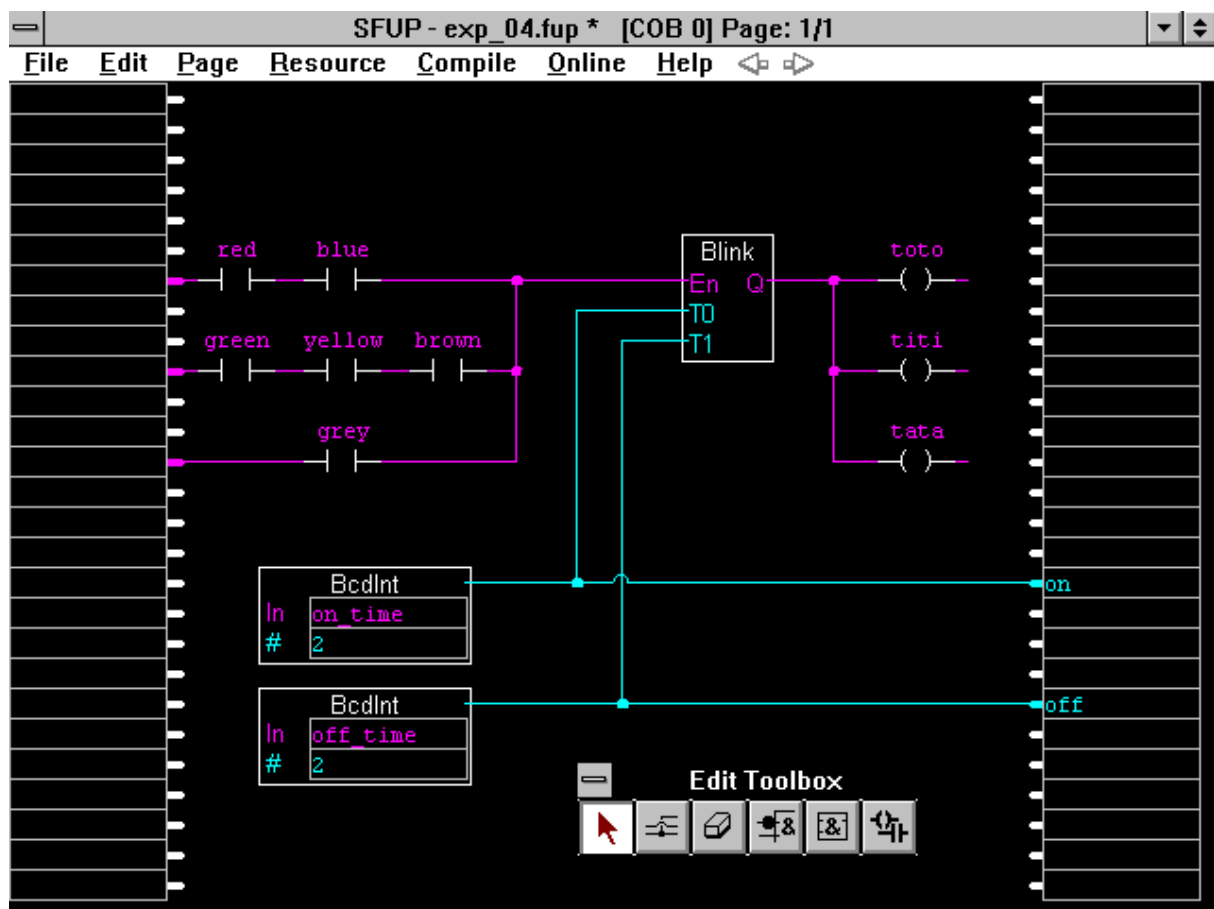
4.6.2 Function plan with ladder diagram symbols (KOPLA)

The program is called "EXP_04" and also belongs in the project "INTRO".

The basic task and points of reference are the same as for the preceding examples "EXP_02" and "EXP_03".

It differs from the previous examples in its use of KOPLA ladder diagram symbols for the logical linkages instead of AND and OR gates.

The function plan diagram now looks as follows:

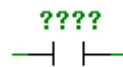


The two connections between the converters and the outputs, here labelled "on" and "off", are not relevant to the present case. They will be used and explained in the next example.

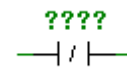
Select the 'Ladder diagram' symbols from the edit toolbox. Position the inputs (contacts), outputs (coils), blinker and converters.

In this example, the arrangement of contacts corresponds precisely to the task. It should be noted that the contacts represent physical contacts, i.e. an AND branch is only fulfilled when all contacts are closed, regardless of whether these contacts are open or closed in the idle state. The single decisive factor is the physical effect.

The same applies with "Ladder diagram" if only those elements are used which are "Normally Open" (as in the example).



Normally Open



Normally Closed

The terms "make" and "break" contacts or "operating" and "home" contacts have been deliberately avoided, as they conceal various philosophies, depending on professional background.

A "Normally Open" element means that the element is conductive in the on state (H).

A "Normally Closed" element means that the element is conductive in the off state (L).

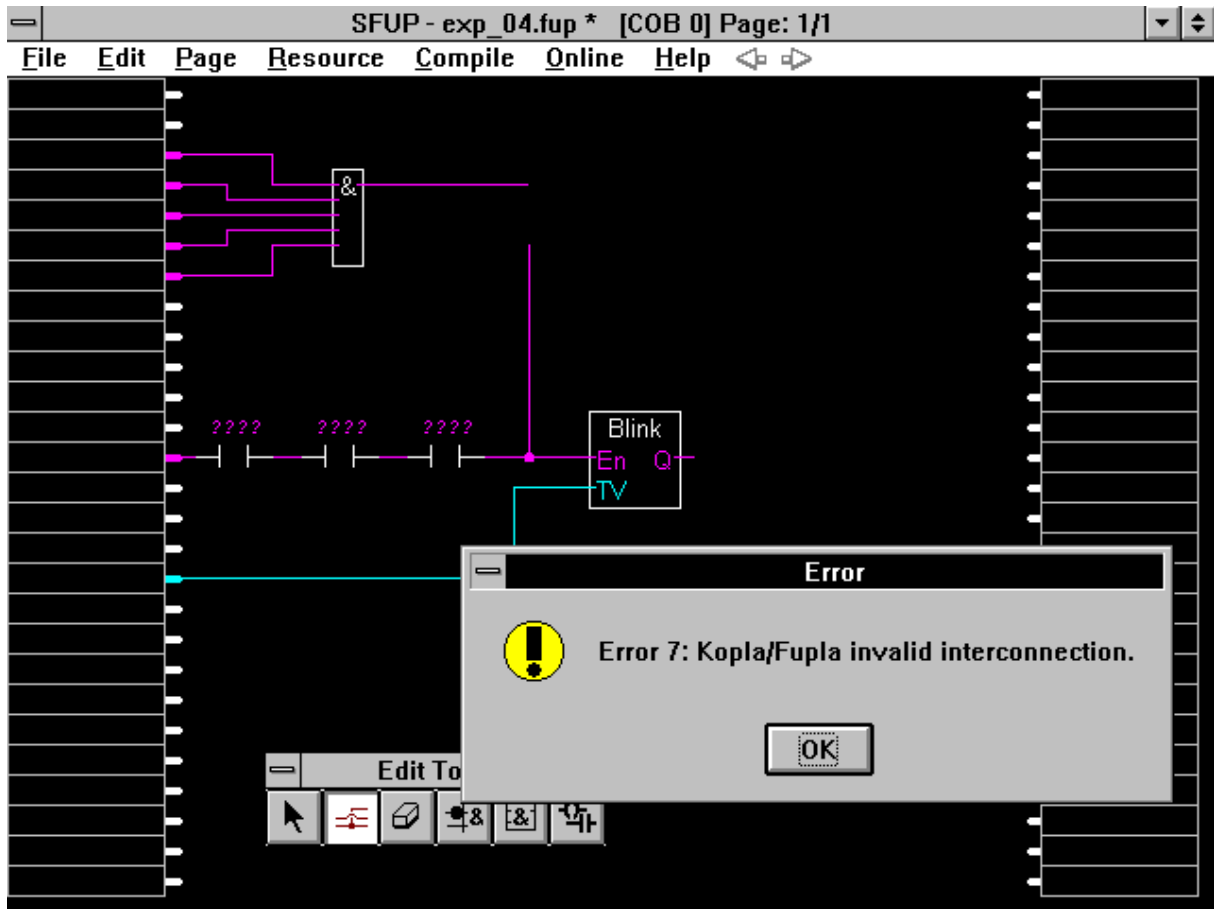
Visually, this is not a physical contact, but the coil of a relay which either has a make contact: "Normally Open", or a break contact: "Normally Closed". The effect in terms of on/off refers to the coil and not to the contact.

When creating a "Ladder diagram" care should be taken that all input elements (contacts) are connected to the input power rail, i.e. to a high signal. Connecting the output elements (outputs) to the output rail (right-hand side) is optional.

Individual ladder diagram elements can be given absolute addresses or symbolic names. Handling by the Resource Manager is exactly the same as for the standard function plan.

Ladder diagram elements can be mixed with the logical FUPLA elements on the same screen page and in the same file. However, a self-contained linkage can only be created using elements of the same type.

The following combination is not allowed. If, despite this, an attempt is made to edit a combination like this, an appropriate warning is issued while drawing and the connection cannot even be made.



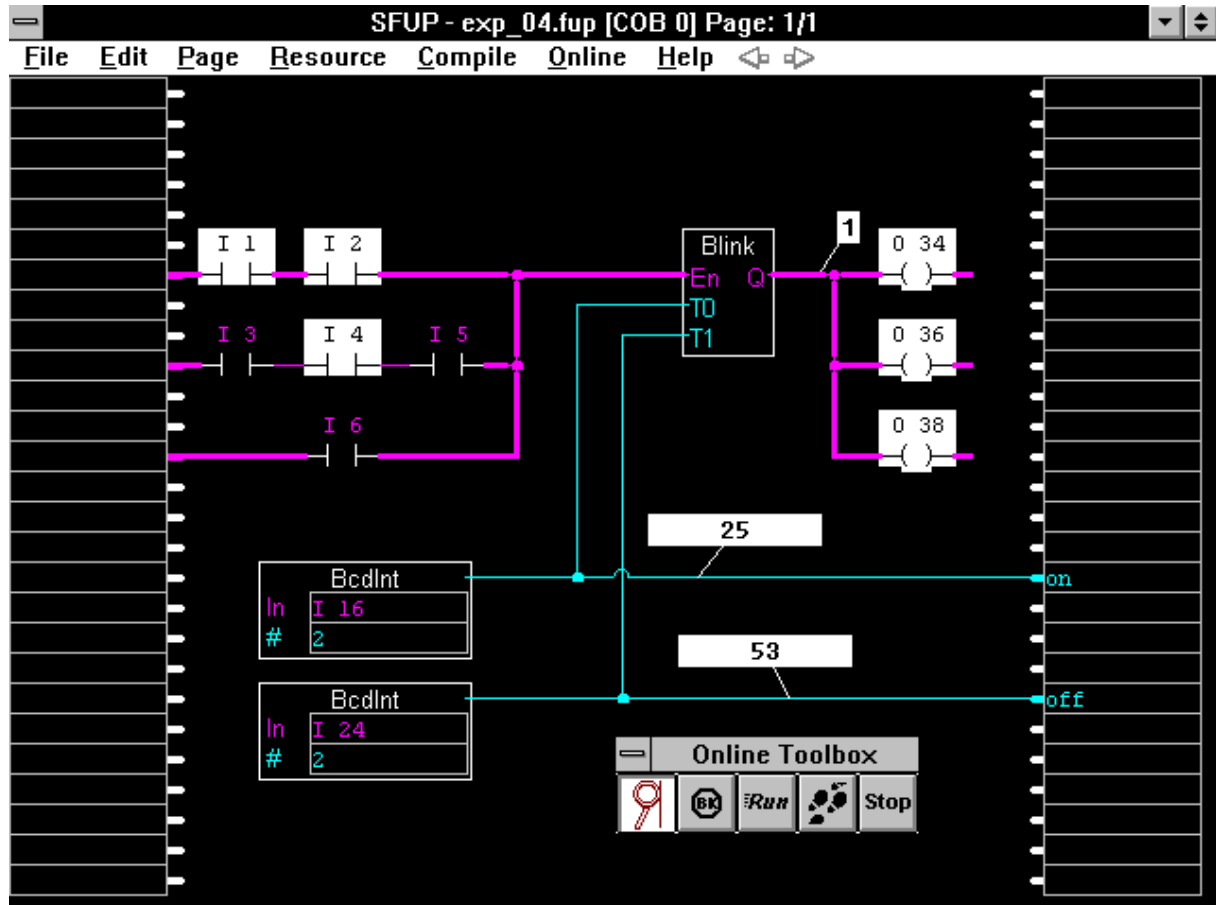
Invalid KOPLA/FUPLA interconnection

Compilation and subsequent "Make" are exactly the same as in the previous example, i.e. in the project manager's "Files in Project" list, the two earlier programs "exp_02_d" and "exp_03_d" should be switched off [FBD\Not linked], so that only the current example "exp_04_d" is linked and downloaded to the PCD.

Files in project:	
exp_02.fup	[FBD\Not linked]
exp_03.fup	[FBD\Not linked]
exp_04.fup	[FBD]

In online operation, KOPLA displays the states of inputs, outputs and flags very effectively. If an element remains in its "normal" appearance, it has not been activated. If the element is shown in white, it is active. KOPLA also displays the (red) connection lines as fine or heavy, depending on the state. Binary probes can also be connected.

Probes can be connected to the blue connections, in the usual way, for the display of integer values.

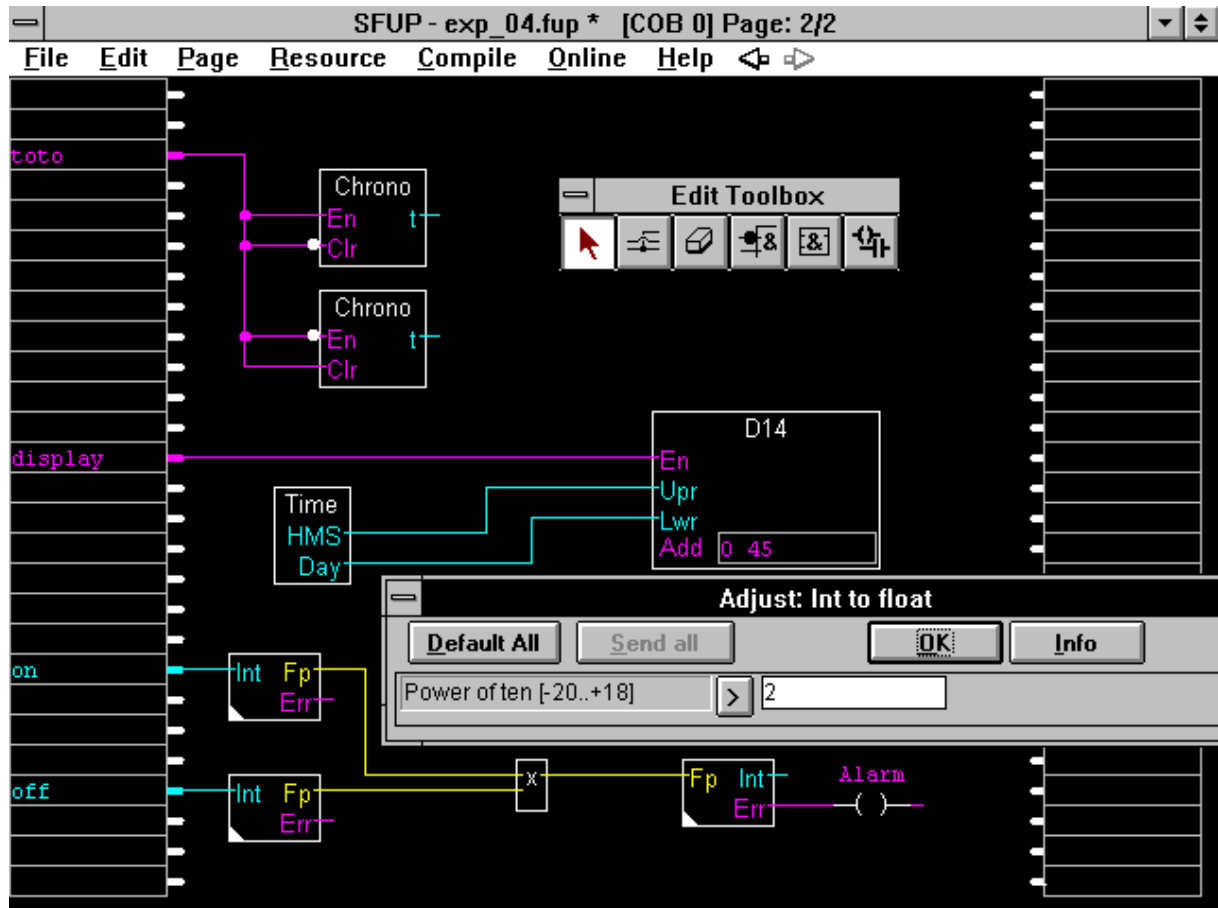


If the same additions are to be made as in the preceding example, there will be a shortage of space. We will extend the function plan to a second page.

4.6.3 Function plan with multiple pages

The same additions as for the preceding example are to be implemented for this one too: the "stopwatch" for displaying blinker on and off times, and the display of the date/time using a PCA2.D14 display module. It would be possible to compact the program "EXP-04.FUP" sufficiently to provide space for these additions. However, as an exercise we will open a new FUPLA page for the additions.

Leave the existing page 1 exactly as it is, and open a second page with 'Page' 'Insert after'. The second page cannot be seen yet, but the right arrow (next page) is now enabled on the menu bar, the left arrow (previous page) is still disabled. Click with the mouse pointer on the right arrow (next page). The new empty page is displayed, and can be edited in the usual way. The left arrow (previous page) is now enabled, which can be used to switch back to the first page.



After displaying the toolbox, the same functions can be edited as in the preceding example, as shown in the illustration. The "Chronometer" functions are coupled to one of the outputs, e.g. "toto". No direct connections can be made from one page to another. Output "toto" from page 1 is defined on page 2 as an input and continued from there.

The clock and "D14 Module" functions are identical to those in the preceding example. The "display" input is assigned to input 7 from the Resource Manager.

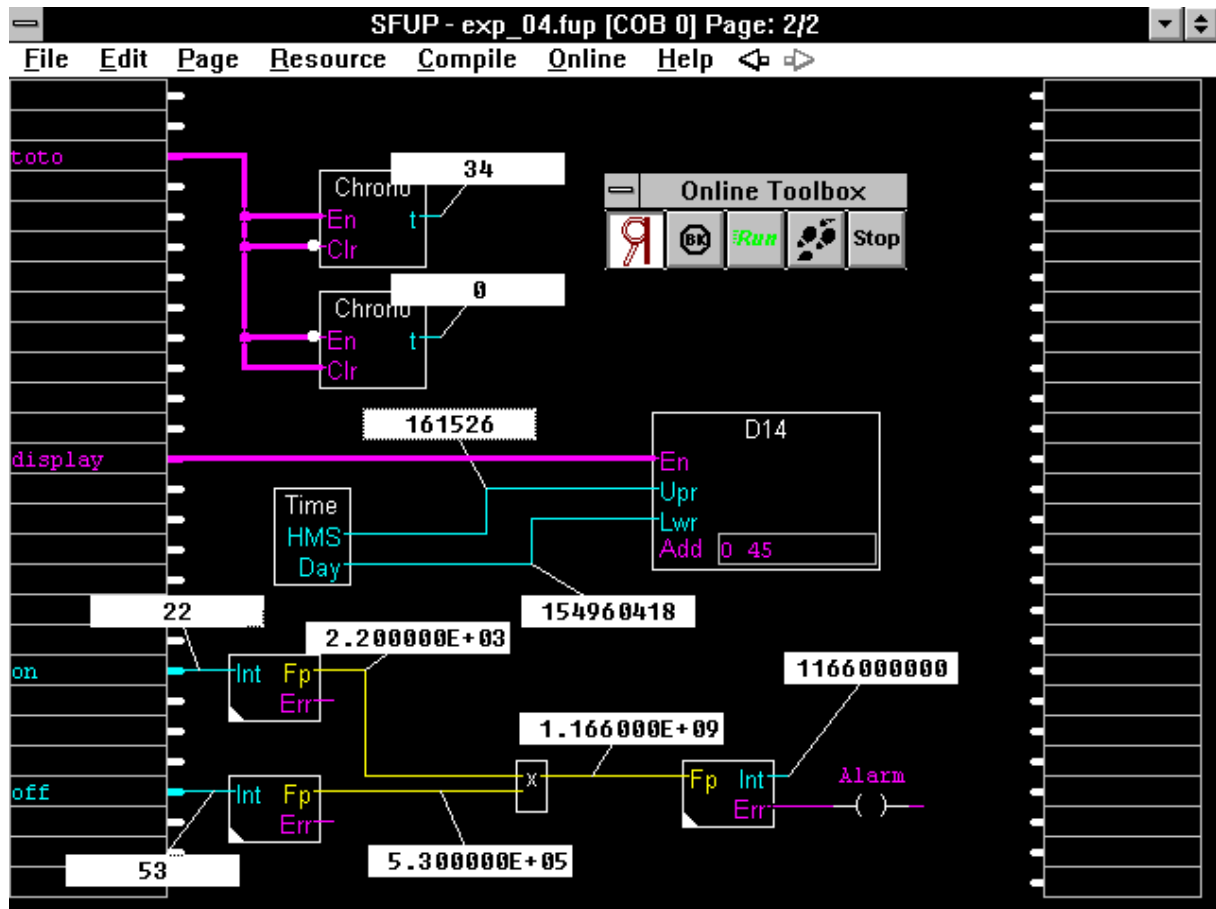
As an example, we can insert a small floating-point program. The two BCD values on page 1 are to be taken and converted into floating point format. On page 1 the two values have been connected to the output rail as "on" and "off". However, neither of the variables are handled in the resource manager. These are dynamic (or internal) variables, which are assigned independently by the system. These variables can be read again into the same or another FUPLA page, as shown here.

Both values are increased by a factor (power) of ten: 10^2 and 10^4 respectively, and then multiplied together (yellow connections). The power of ten is specified in the adjust window. This is invoked by double-clicking on the relevant converter FBox. The value obtained is subsequently converted back to integer format. If multiplication results in an overflow of the 31-bit (2,147,483,647) range of a PCD register, the "Err" (error) output of the "Float to int" converter is activated. So that this error can be easily visible online, an "alarm" flag is edited into the ladder diagram.

After saving with 'File' 'Save', compiling, assembling, linking, downloading etc. ('Make'), both pages of the function plan can be provided with online probes. Pages can be changed while online by using the arrows on the menu bar or the menu 'Page' 'Next'/'Previous' commands, even if the program consists of more than two pages.

For example, if a program consisted of 20 pages, it would be easier to select the appropriate page directly from the 'Page' 'Index' dialog box, instead of leafing through all the pages. If there are many pages it is also sensible to give each page a concise name and a comment in addition to a description.

The next illustration shows the various online probes. Notice the floating point format, shown here for the first time.



Both pages can be given comments. Open the "Page Information" dialog box with the 'Page' 'Info' command. This window always relates to the FUPLA page which is currently displayed.

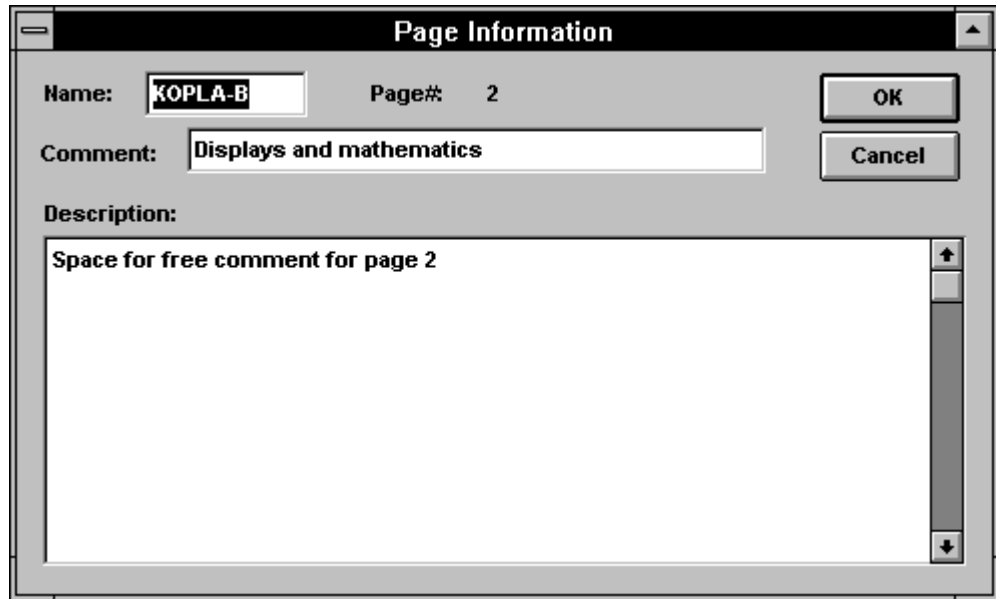
If on FUPLA page 1, enter the name "KOPLA-A" in the "Name" field, and "KOPLA-B" for FUPLA page 2.

To describe the open FUPLA page, enter an optional text in the "Description" field. (With the current version V1.3, this description is limited to 400 characters).

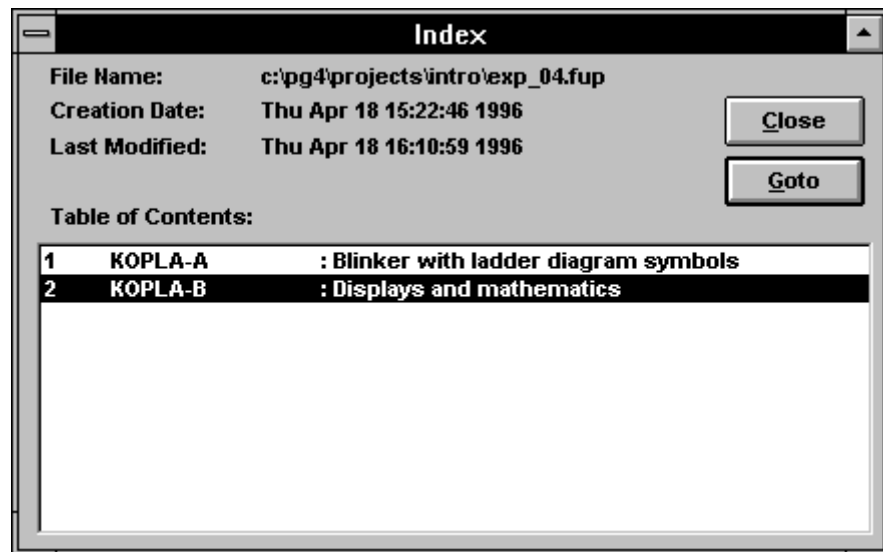
Enter an abbreviated description in the "Comment" field. The comment is limited to 40 characters.

The next illustration shows the "Page Information" dialog box for FUPLA page 1.

The illustration after that shows the "Index" dialog box for our example "EXP_04", which is displayed with the 'Page' 'Index' command.

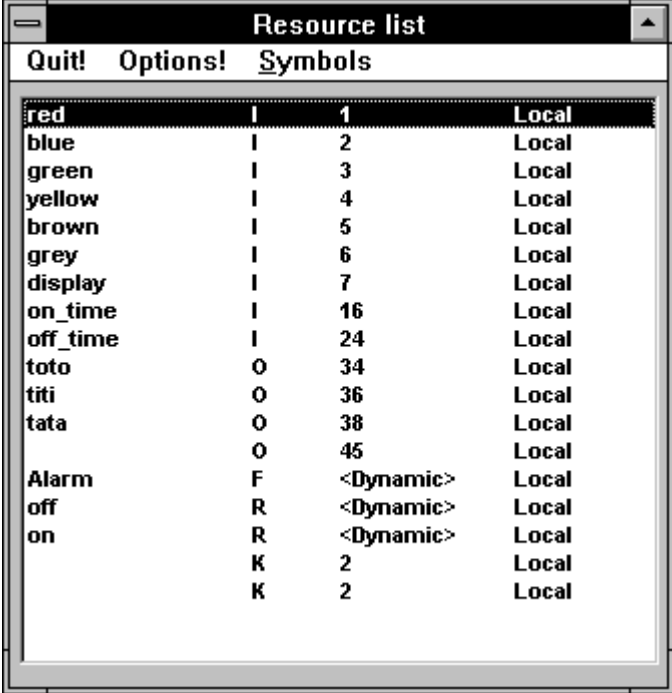


"Page Information" window for page 2 of the FUPLA diagram. This window is called with 'Page' - 'Info...'.



"Index" window for our example "EXP_05_D", called with 'Page' - 'Index'.

Display the resource list again. The two variables "off" and "on", which are used to transfer values from the first to the second page and which had not been assigned an absolute address, appear as "(Dynamic)". For this purpose the compiler takes 2 registers from the elements assigned in the 'Resource' - 'Dynamic Variables...!' menu.



Resource list			
Quit!	Options!	Symbols	
red	I	1	Local
blue	I	2	Local
green	I	3	Local
yellow	I	4	Local
brown	I	5	Local
grey	I	6	Local
display	I	7	Local
on_time	I	16	Local
off_time	I	24	Local
toto	O	34	Local
titi	O	36	Local
tata	O	38	Local
	O	45	Local
Alarm	F	<Dynamic>	Local
off	R	<Dynamic>	Local
on	R	<Dynamic>	Local
	K	2	Local
	K	2	Local

4.7 Further examples

The examples shown and discussed in the previous section made exclusive use of FUPLA or KOPLA, and were edited in a single file. In practice, tasks may exist which go beyond these limits.

For example, it is not possible with the current FUPLA to transfer text via a serial interface to a printer or terminal. Another task might be to combine a FUPLA program with an existing program edited in IL. Cases may also arise where a program becomes so large that it must be divided into several separate modules, and then linked. The following typical cases can be identified, although combinations of them are also possible: (Other combinations using GRAFTEC are discussed later in the GRAFTEC section).

- FUPLA file containing FBs/PBs and XOBs, all of which are edited in FUPLA/KOPLA.
- FUPLA file containing FBs/PBs and XOBs, some or all of which are edited in IL.
- Program consisting of several COBs, some of which are programmed in FUPLA and some in IL.
- Program consisting of 2 or more FUPLA files.

This section provides a brief discussion of such situations.

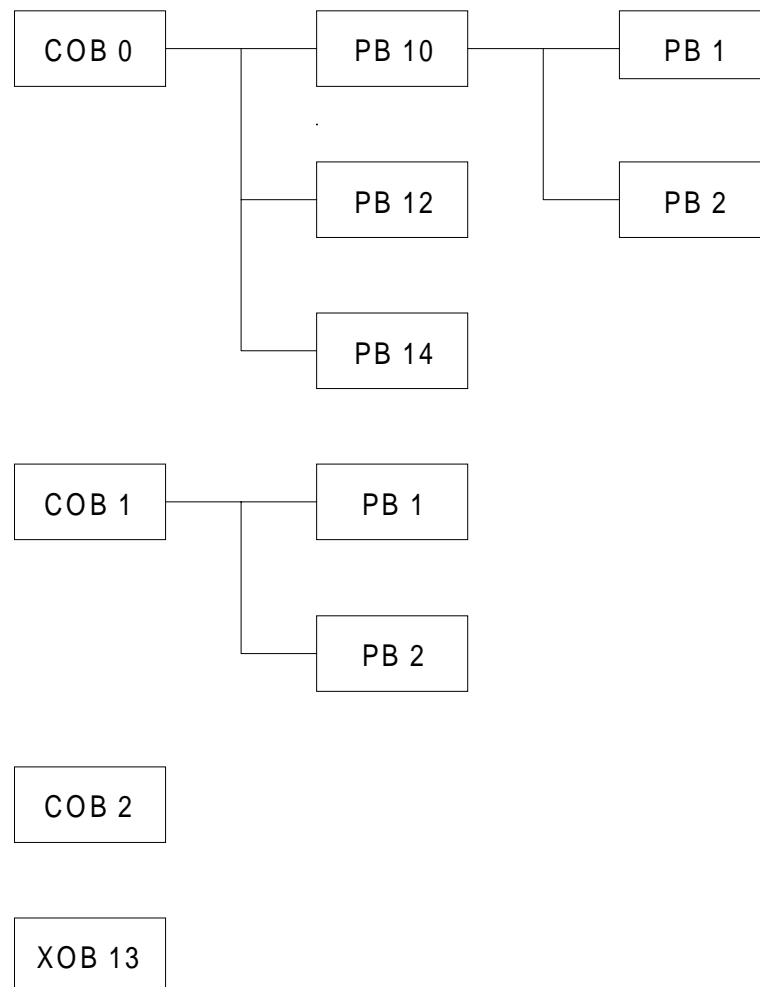
It should first be stressed that, compared with earlier versions, the present version V 1.3 (or later) handles the above combinations without problem. This is because the resources for the whole project are managed in common by the project manager, which also takes over linking the separate parts of the program.

4.7.1 FUPLA file containing FB/PB/XOBs, all edited in FUPLA

This is an ideal case in proven BLOC TEC structure for the PCD family. The FBs/PBs contain frequently used routines which are called conditionally or unconditionally from the main block (COB), or from a PB, FB or XOB. Note that FBs and PBs are treated in the same way by the present version: FB s cannot be called with parameter lists.

It would be excessive to demonstrate an operational, practical example here, as it would be too varied in extent. However, the procedure for creating such a program is outlined below.

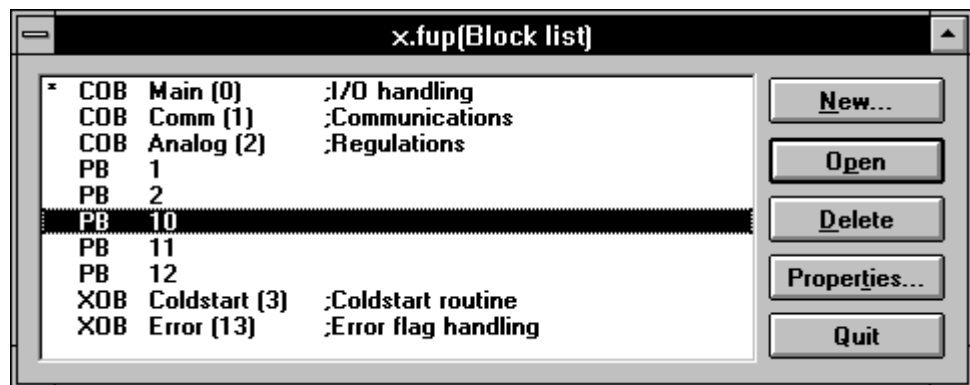
Program structure of a project:



After selecting FUPLA, the submenu 'Block List...' is called from 'File' on the menu bar. COB 0 has already been entered and could be programmed straight away. However, we first want to define all the other blocks in the program.

Select 'New'. COB 1 is already selected, press 'OK'. Enter COB 2, the PBs and XOBs one after the other, the order of entry is not significant. With 'Properties...' it is possible to assign names and comments to the blocks, or to change block numbers.

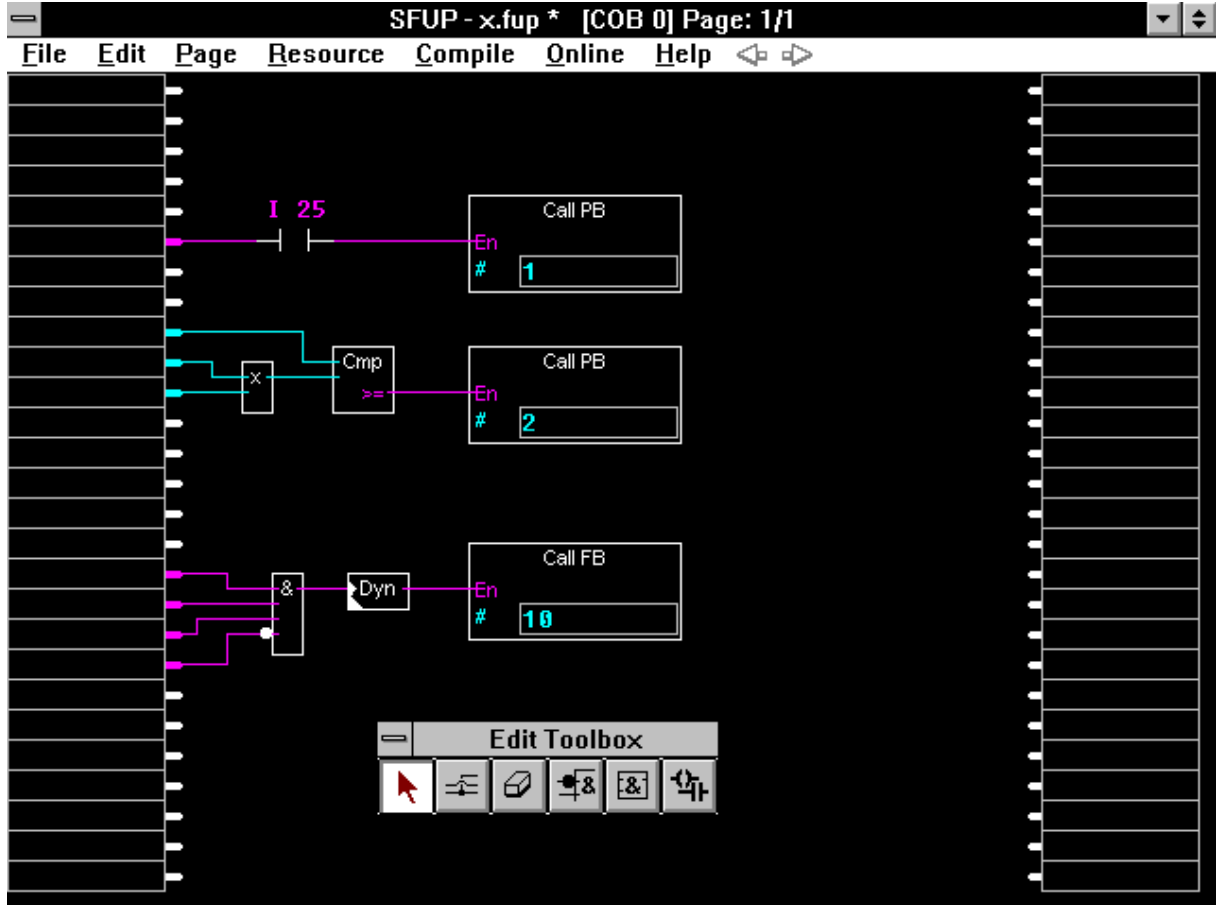
The window now has the following appearance.



An asterisk at the start of the block's line shows which block is currently selected. Double-clicking on a line or clicking on 'Open' selects the block, initially for editing and later for online or offline viewing of the block. This menu is the primary way of selecting a block. Other blocks can be defined later, or deleted.

PB/FBs are called from a COB or XOB with the 'Call PB' or 'Call FB' functions from the 'User definable' function family.

The following are conditional calls for PB s from a COB.



4.7.2 FUPLA file containing FB/PBs edited in IL (Instruction List)

The example will transmit texts from a FUPLA program.

The FUPLA program consists of a blinker connected to several outputs and a time display on F510 of a PCD2.

A number of texts should also be output to a PCD7.D100 terminal:

Switch on input 1 → Text 101
Switch on input 2 → Text 102
Switch on input 3 → Text 103

The relevant text should only be sent when there is a rising edge on the associated input (dynamized). There will be no polling of the "Text Busy" flag.

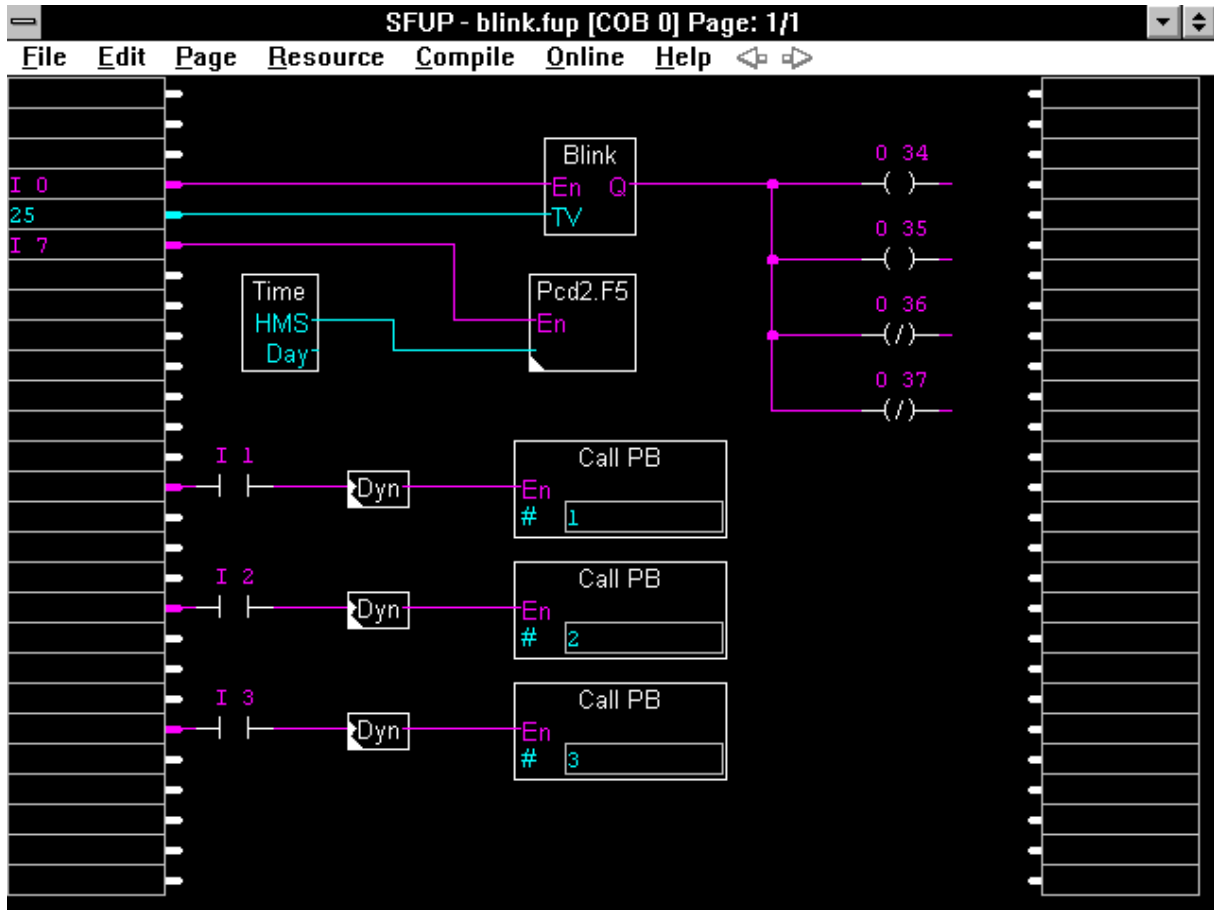
The interface is to assigned once only., i.e. when the PCD is powered up (in XOB 16). (The "interface parameters" FBox from the "Communication" function family is not suitable for our example).

This problem can be neatly solved by calling some PBs (Program Blocks) from FUPLA. The PBs in turn contain the instructions for sending texts and assigning the interface. The texts themselves are located either in the relevant PB or anywhere in the IL program. The IL file is edited preferably using the SEDITWIN editor supplied with the PG4. However, the IL file can also be created with any ASCII editor, or with SAIA's familiar SAIA editor supplied with the PG3 package. An IL file is therefore also edited next to the FUPLA file. This IL file is ultimately linked to the compiled FUPLA file to produce a loadable .PCD file.

It should be stressed here that

- the resources should be divided up clearly and distinctly,
- a different name should be given to each module in the project and
- the programs should be linked by the PG4, not by the PG3.

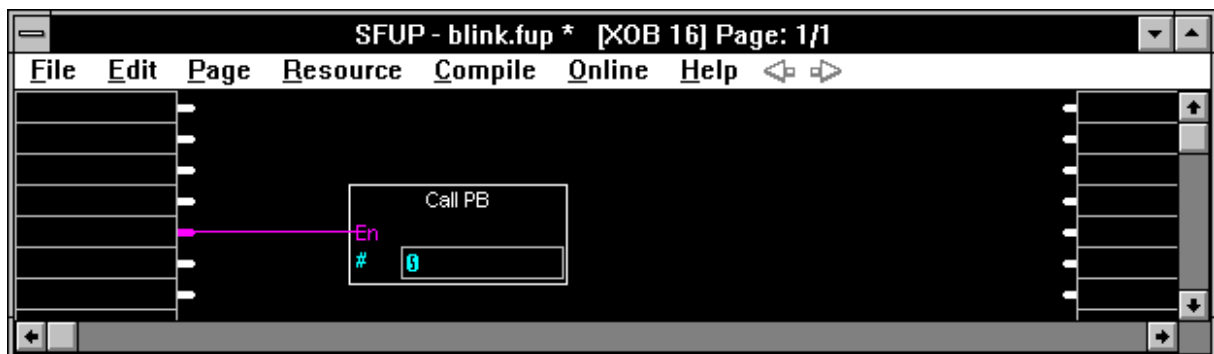
The name given to the project is "TEXT". We will first edit the FUPLA file in the normal way with the blinker and the display. We will then call the 3 text output PB s. This file, called "BLINK.FUP", appears roughly as follows:



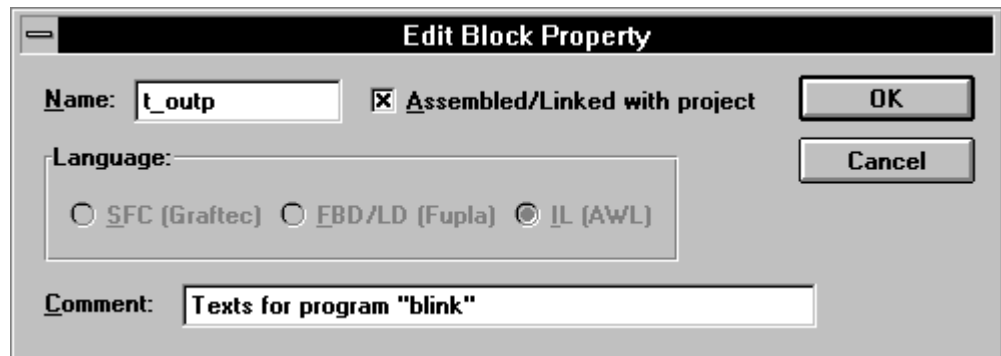
Next we must program interface assignment in XOB 16.

From 'File' on the menu bar we select sub-menu 'Block List...'. This window already contains "* COB 0". The asterisk indicates that COB 0 is the part of the program which is currently shown on the screen. We continue by selecting 'New...'. From the choices shown we click on 'XOB', enter the number <16> and press 'OK'. XOB 16 is now shown in the list. Now select XOB 16 by double-clicking on it. A new, empty FUPLA page appears.

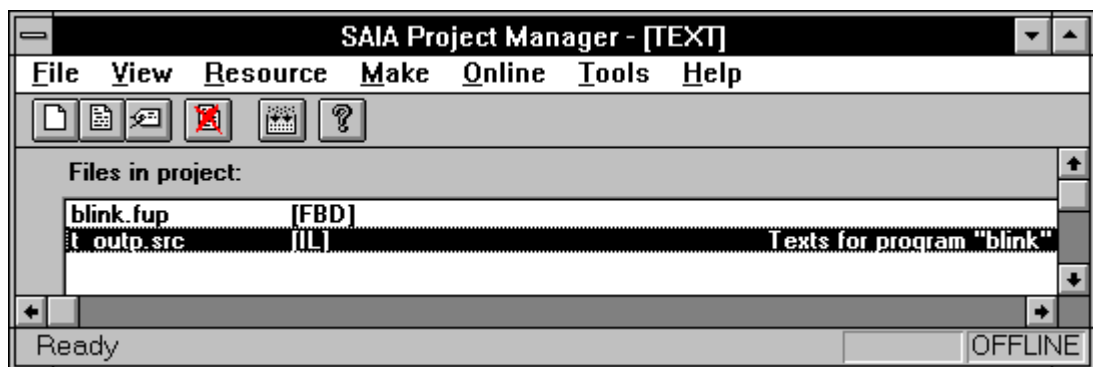
We edit a call for PB 0. We drag the "En" input to an empty input field, so that the PB is called unconditionally. (The status of an empty input field is always "H"). Since XOB 16 will be executed once only at PCD power-up, the call is only made once.



After saving, we return to the project manager. It is now necessary to edit the IL file. We give it the name "T_OUTP" (text output).



Since this concerns an IL program, select "IL (AWL)". Click on 'OK' to include this IL file in the project list.



Double-clicking on this file name calls the IL editor (SEEDITWIN). Here we program the 4 PB s in IL.

The IL programming technique cannot be dealt with here. Please consult the following manuals:

- "User guide for the PCD family" 26/732
- "Instruction set for the PCD family" 26/733

The IL program then resembles the following:

```

pb      0

sasi    1      ; for terminal PCD7.D100
        0
$ssasi
text    0 "uart:9600,8,e,1;mode:mc0;diag:f0,r0"
$endsasi

        epb
; -----

pb      1

stxt    1
        101

text    101    "<12>SAIA AG <10><13>"
        "CH-3280 Murten/Switzerland"

        epb
; -----

pb      2

stxt    1
        102

text    102    "<12>SAIA GmbH<10><13>"
        "D-63267 Dreieich"

        epb
; -----

pb      3

stxt    1
        103

text    103    "<12>SAIA-Burgess G.m.b.H<10><13>"
        "A-5020 Salzburg"

        epb
; -----

```

Note for programmers who have the PG3 package:

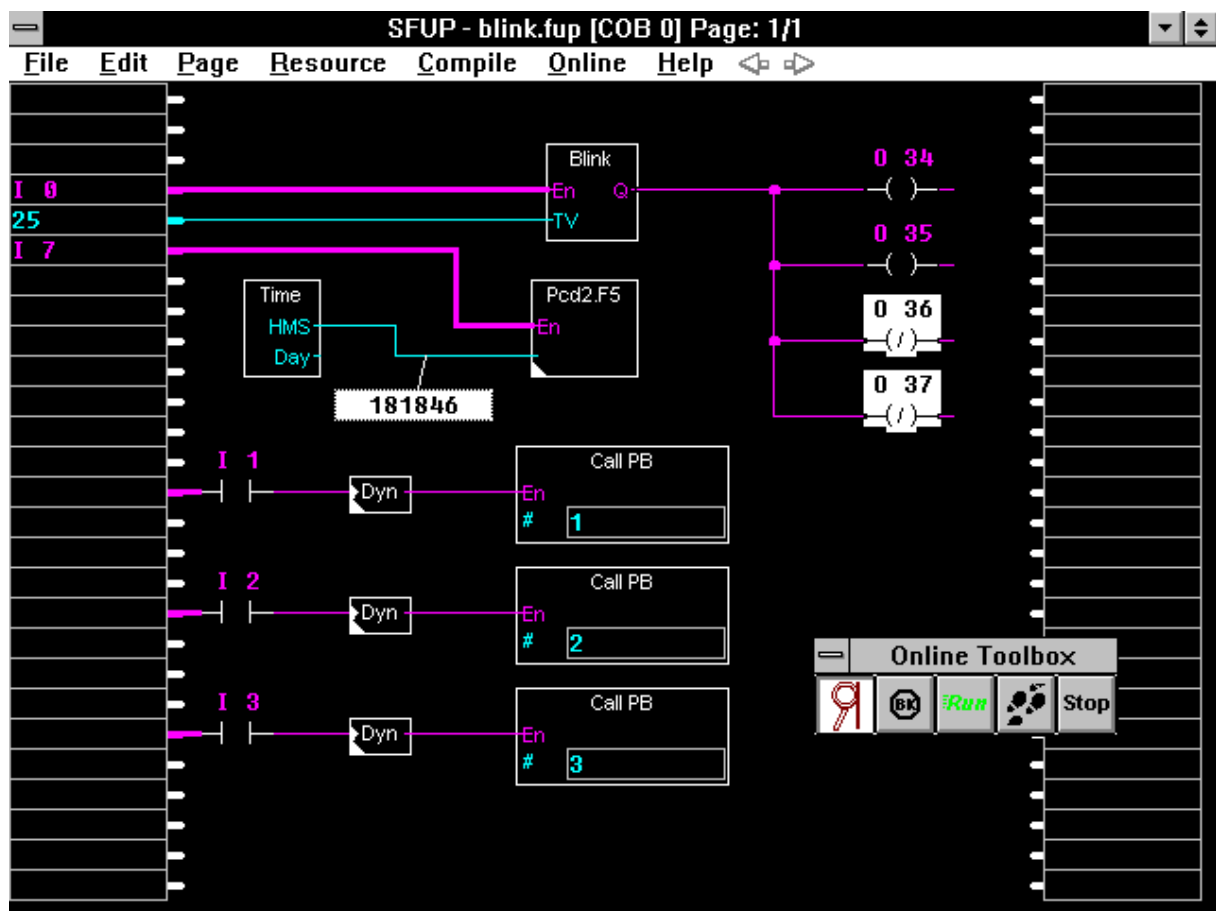
If the IL part has been programmed with the PG3, it is best to assemble it before leaving the PG3. This makes any errors much easier to find.

Once the IL file has been saved, it is now a question of processing both files further and finally loading the program and testing it.

We return to the project manager. Both newly created files appear in the list. Call the FUPLA program "blink.fup" again and compile it (if this has not already been done). Then go back to the project manager and start 'Make'. In accordance with the settings, both files are assembled, linked, downloaded and switched into Run.

If both files are really to be linked, the option [FBD\Not linked] must be switched off in both of them!

If the FUPLA diagram is called again, it can be switched online.



I	O	Local
I	1	Local
I	2	Local
I	3	Local
I	7	Local
O	34	Local
O	35	Local
O	36	Local
O	37	Local
K	0	Local
K	1	Local
K	2	Local
K	3	Local
K	25	Local

Additional notes concerning the above example:

The resource list of the FUPLA diagram only contains the resources used in the FUPLA file. These are the 5 inputs and the 4 outputs. The numbers of the 4 PBs are only included as constants in the list. The same applies for the blinker TV value. None of the resources used in the IL file, in this example the 4 texts, are shown in the resource list of the FUPLA diagram.

If the IL file was created with SEDIT from the PG3, the resources of that file will be listed in SEDIT's resource manager.

If more complex programs are edited partly in FUPLA and partly in IL, care should be taken to ensure that the resources, including block numbers, of the program's two parts are unique, otherwise conflicts may arise when the programs are linked and executed.

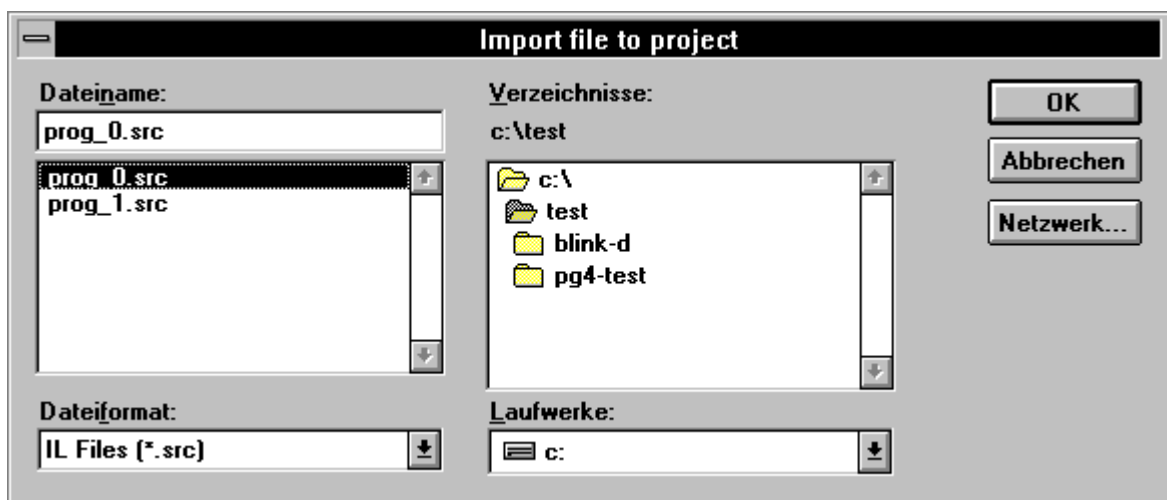
Another important point concerns the dynamic resources. It should be strictly ensured that no IL program uses any resources from the range of defined dynamic resources. If the predefined standard ranges are accepted, it is often forgotten that these ranges cannot be used in another application. Although the assembler reports collisions of this type, these messages are not always easily understood by the less experienced programmer.

4.7.3 Adding a FUPLA file to an existing IL program

An existing IL program is to be composed of the 2 files "PROG_0.SRC" with a COB 0 and "PROG_1.SRC" with a COB 1. A new module is to be edited in FUPLA and combined with the existing IL program. The FUPLA file is to be put into COB 5 and given the name "PROG_5.FUP". The 3 files are to be located in project "AAA".

From WINDOWS we call the PG4. The project library is displayed. With 'File' - 'New...' we open our project "AAA" and go to the project manager. Our two existing IL files are in directory C:\TEST. Both of these files are to be integrated into our project "AAA". We select

'File' - 'Import...'

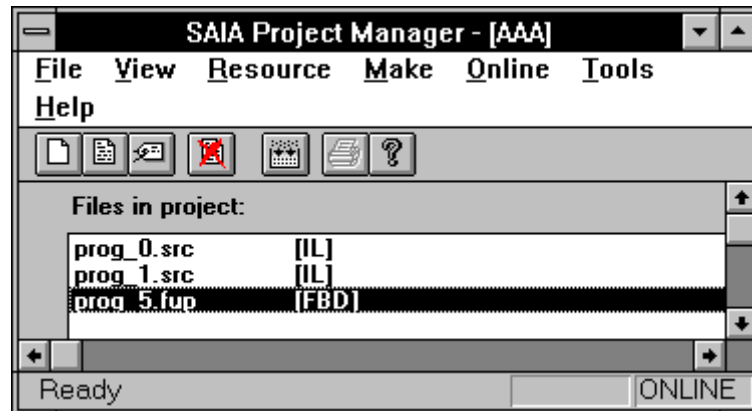


and complete the dialogue box according to the rules of WINDOWS, so that the desired files can be imported to our project with 'OK'.



The new FUPLA file to be edited is now opened with 'File' - 'New...', <PROG_5> as an "FBD/LD" file. We select the FUPLA diagram, use 'File' - 'Block Properties...' to adjust the COB number (0 → 5), then edit and compile our FUPLA file.

Finally, we return to the project manager and execute 'Make'.



According to the settings, the 3 files are assembled, linked, downloaded and switched into Run. The FUPLA diagram can be viewed online.

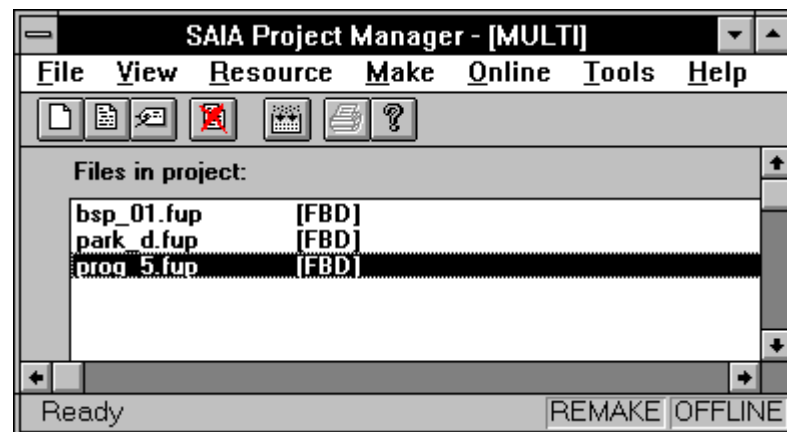
The IL programs can be viewed in the debugger, selectable from the project manager with 'Tools' - 'Debugger'. Before calling the debugger, the FUPLA diagram should be switched offline, to avoid any conflict on the serial interface of the programming unit. This would be announced with the following message.



4.7.4 Program consisting of several FUPLA files

Whereas with earlier versions of the PG4 it was a brave venture to combine several FUPLA files so that all modules could be viewed online, with the present version it has now become child's play. Since all the resources of a project are managed in common by the project manager, several FUPLA files can be listed in the project manager and then assembled, linked and downloaded with 'Make'. The assignment of dynamic resources also takes place automatically for the whole project.

A few points should be noted:



Each of the 3 files must be in a different COB. If not, an appropriate message is displayed during linking and the loadable file "PROJECTNAME.PCD" is not created.

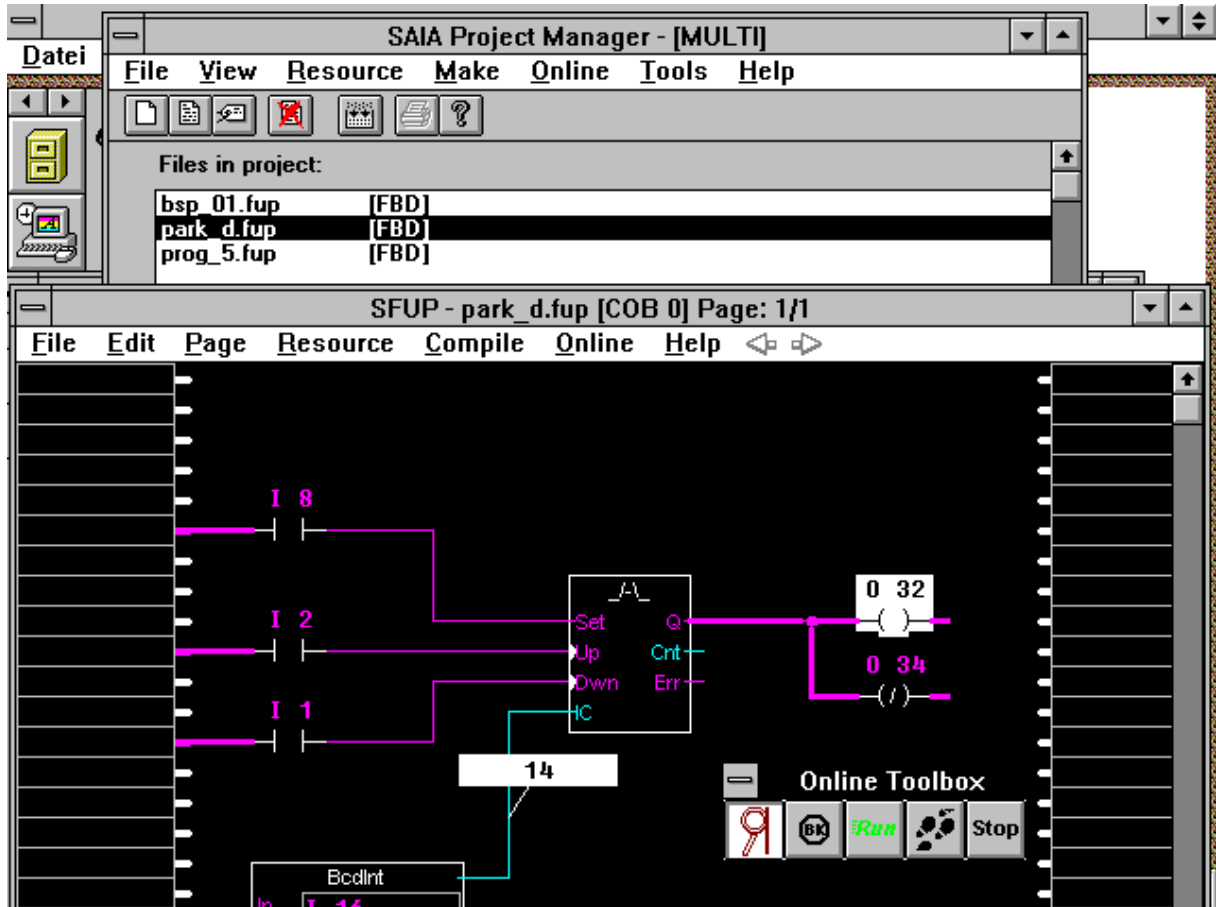
When, for example, 3 FUPLA files are provided in one project, all 3 files must be selected in turn and compiled individually, even if they have been compiled once already as an individual file. This is necessary for the correct assignment of dynamic resources.

When the program has been successfully created and downloaded, each file can be selected and viewed online. Care should be taken to ensure that no other projects remain open in the background, as this can slow down online operation considerably.

Although dividing a project into different FUPLA files can be handled today without problem, excessive use of this technique should be avoided. Multiple-page FUPLA programs should be used in preference to the method with multiple files. The division of a FUPLA program into several FUPLA files is only really necessary when a single file exceeds 8k of PCD code (as the assembler cannot process more code at once) or when a file requires more than 2000 resources.

Practice:

To simplify switching between the separate files and the project manager when working with multiple FUPLA files, WINDOWS allows the screens to be moved about so that both are visible and accessible.



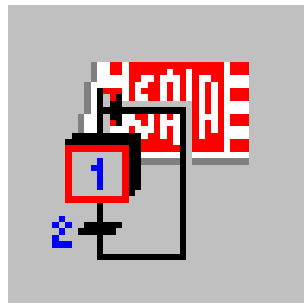
5. GRAFTEC (SGRAFWIN)

5.1 Introduction

SAIA® GRAFTEC is described in detail in chapter 3.3 of the "User manual for the PCD family" (publication 26/732 E). This PG4 manual is only intended to explain the structure and operation of the GRAFTEC editor, also with the use of examples.

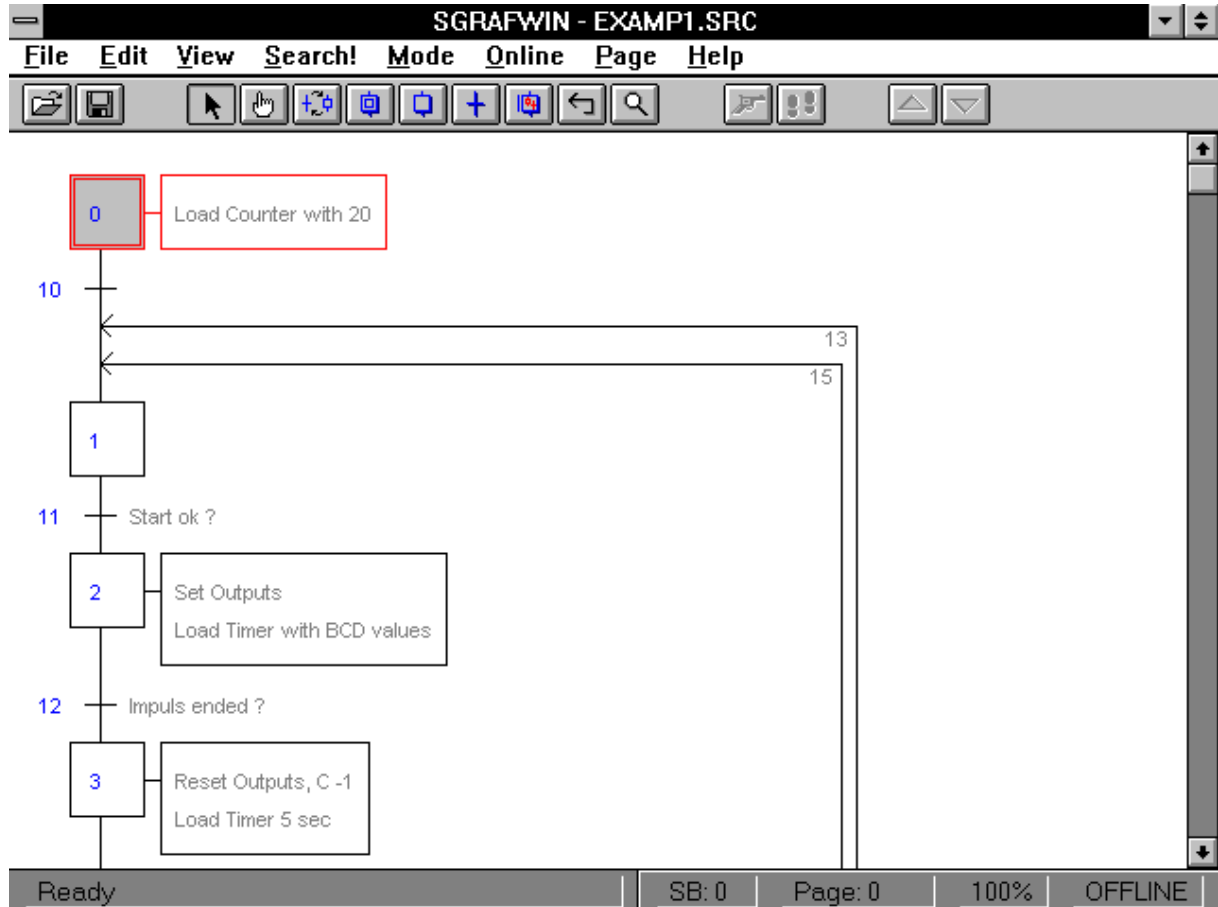
The GRAFTEC editor has a very detailed, interactive "Help" file in English.

Knowledge of SAIA® GRAFTEC and how to work with MS-WINDOWS is assumed.



5.2 Description of functions and menus

5.2.1 Screen elements



The title line contains the name of the programming tool: "SGRAFWIN" (SAIA® GRAFTEC under MS-WINDOWS), followed by the name of the file being edited. The file name only appears after the first save with 'Save As...' from the 'File' menu.

The main menu bar shows the GRAFTEC commands:

File Edit View Search! Mode Online Page Help

Clicking on a command displays the sub-menu.

Below the menu bar is the toolbar (symbol bar).



The toolbar contains the functions most frequently used when working on a GRAFTEC file. Clicking on one of the function symbols executes the function directly or opens it. The toolbar is described in section 5.2.10.

Below the toolbar is the actual GRAFTEC entry window. At the left-hand and lower margins there are scroll bars for scrolling the entry window.

At the very bottom is the status line. This contains information about the current GRAFTEC file.

In the following sections individual submenus are described in the order in which they appear on the menu bar. The order in which the individual functions are applied in practice is described in section x.x: "Using GRAFTEC".

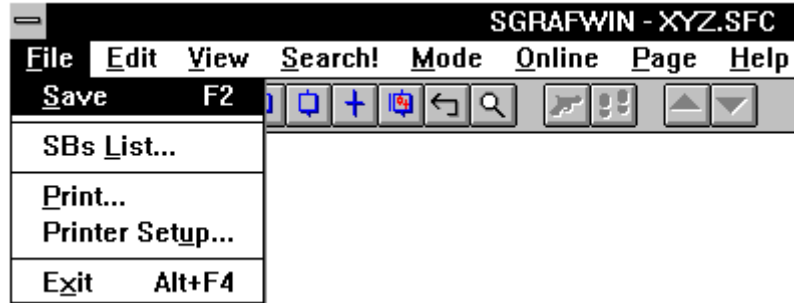
Note: This document only describes operating GRAFTEC with a mouse. "Clicking" means pressing the left-hand mouse button (on a right-handed mouse). Use of the right-hand mouse button is described where necessary.

Using the menus with the keyboard requires the the "ALT" key to be held down while pressing the main menu's underlined command character. When a drop-down menu is displayed, the short-cut keys can be seen.

Consult 'Keyboard' in the 'Help' menu for details of keyboard usage.

5.2.2 'File' menu

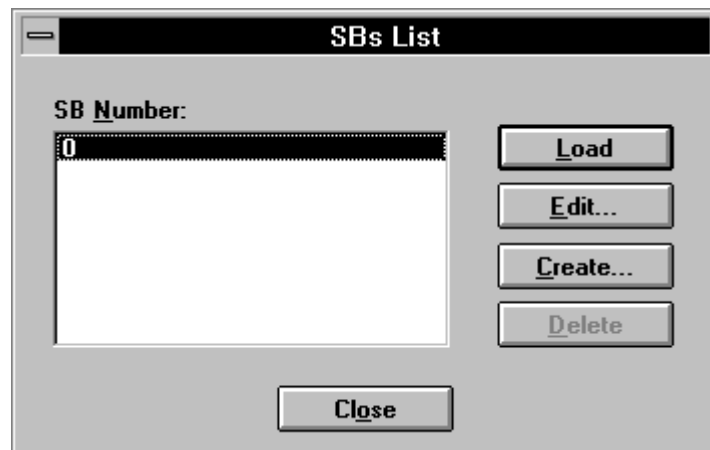
Clicking on 'File' displays the following sub-menu:



Important: If a menu item is followed by ..., an additional dialog box is displayed when it is selected.

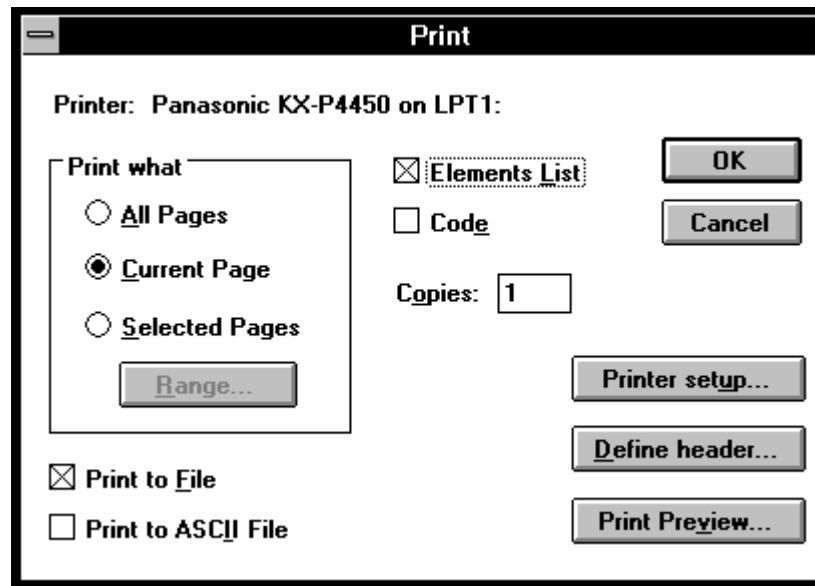
'Save': The GRAFTEC file will be saved under the name defined in the project manager.

'SBs List...': Clicking on 'SBs List...' displays the following window:

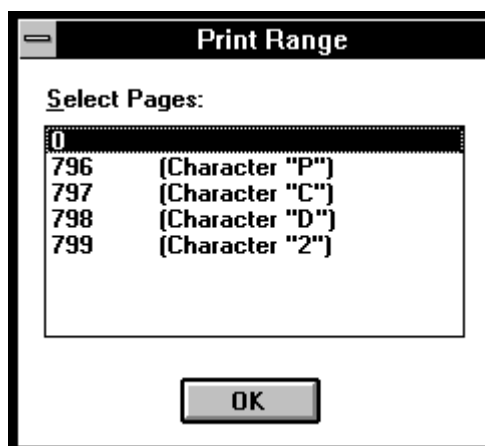


This menu is used for handling sequential blocks (SBs). If a small program of one SB only is to be created, SB0 is selected automatically. In this case the above menu is not used.

'Print...': This menu is used to arrange the print-out of a GRAFTEC file.



- 'All Pages': All pages are printed, i.e. the basic structure and all other pages (macro steps).
- 'Current Page': Only the currently edited page (macro step) or the basic structure is printed.
- 'Selected Pages'. The pages marked after pressing 'Range...' are printed.

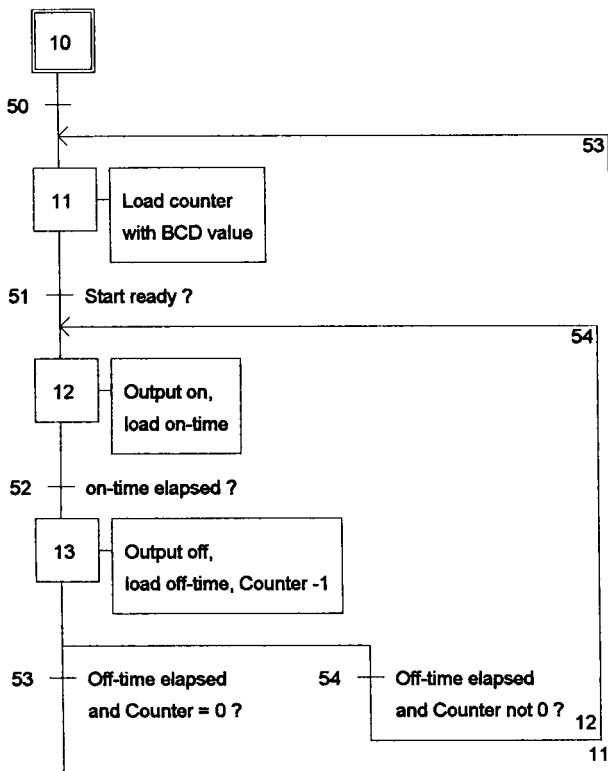


- 'Print to File': When switched off *): prints in fine resolution.

When switched on: file "name.prn" is created. This file can be printed later in fine resolution with the DOS "print" command.

USER : FOR SAIA'S INTERNAL USE ONLY
 EDITOR : SGRAFWIN V1.3
 FILENAME : C:\PG4\PROJECTS\MAN_E\BLINK_1.SFC
 DATE : 23.04.96

TITLE : GRAFTEC example
 SUBTITLE : Blinker with preselection
 SB : 0
 PAGE : 0



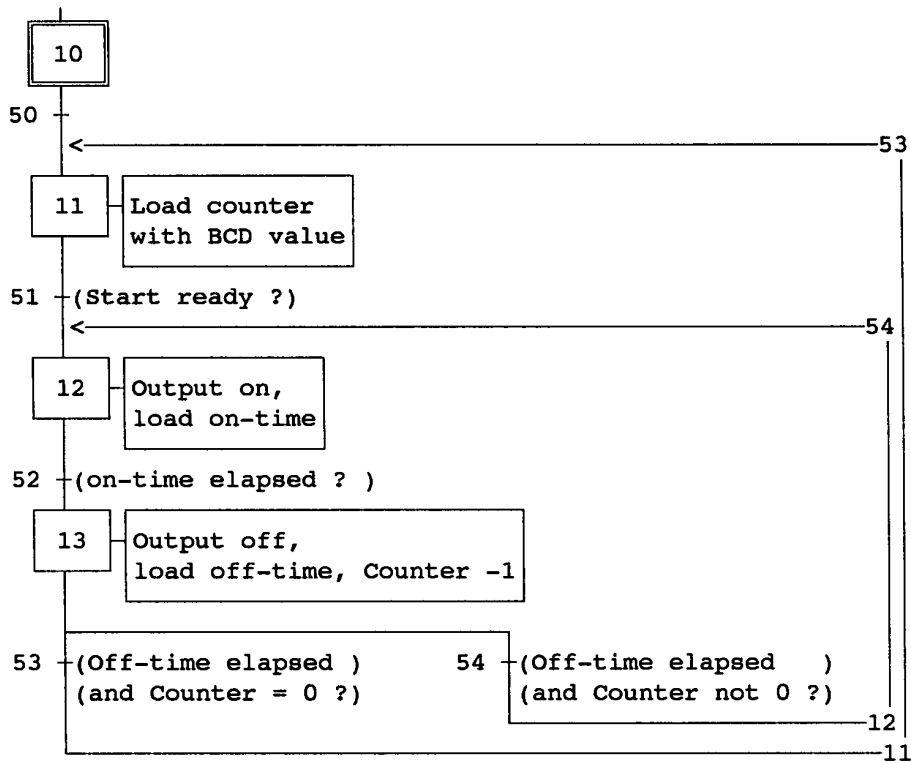
- 'Print to ASCII File': When switched off *): prints in fine resolution.

When switched on: ASCII file "name.gls" is created.
 This can be printed out on any printer with the ASCII character set.

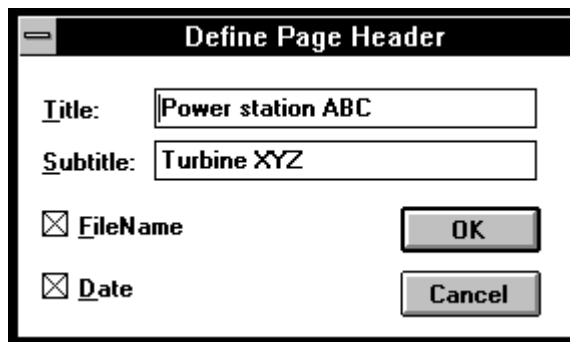
*): both print options switched off

```

*****
USER:          FOR SAIA'S INTERNAL USE ONLY
EDITOR:        SGRAFWIN V1.3
TITLE:         GRAFTEC example
SUBTITLE:      Blinker with preselection
FILENAME:      BLINK_1.SFC
DATE:          23.04.96
SB:            0
PAGE:          0
*****
    
```



- 'Elements List': When switched on, a list of all GRAFTEC elements (IST, ST, TR), their symbolic names and comments is printed or written to the file.
- 'Code': When switched on, the AWL code for each GRAFTEC element is printed. (Not FUPLA code)
- 'Copies': The number of print copies can be specified (1 - 999)
- 'Printer setup...': The printer can be selected and installed (see WINDOWS)
- 'Define Header...':



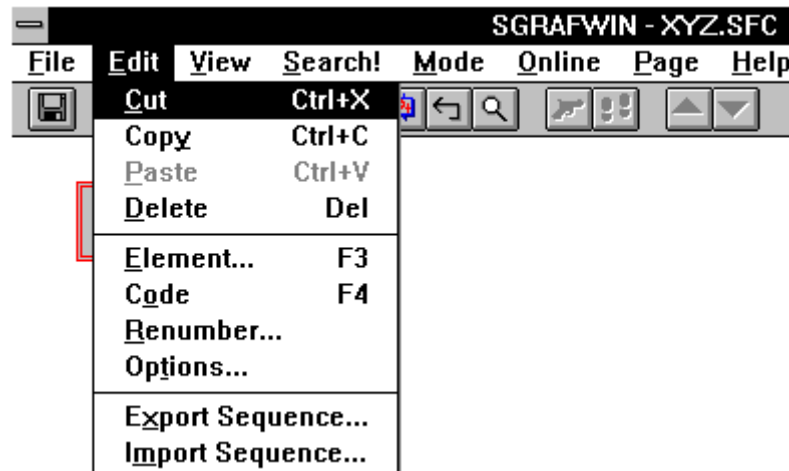
- 'Title', 'Subtitle': A title and subtitle can be entered, which appear at the top of each sheet when the file is printed. The maximum length each text is 25 characters.
- 'FileName': Prints the file name (x)
- 'Date': Prints the date (x)
- 'OK': Confirms, executes
- 'Cancel': Cancels and exits the menu

'Printer setup': The printer can be selected and installed (see WINDOWS)

'Exit': Cancels the 'File' menu and exits

5.2.3 'Edit' menu

Clicking on 'Edit' displays the following sub-menu:



Cut: | With these 3 standard WINDOWS functions, it is possible to select individual elements or connected sequences from a GRAFTEC structure and cut them out or copy them by inserting (pasting) into the same or another GRAFTEC file.

Copy: |

Paste: |

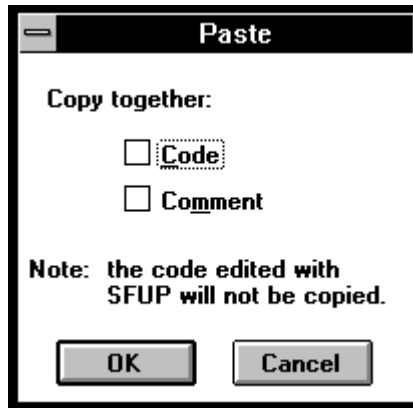
'Cut': An individual GRAFTEC element (ST, TR) can only be cut if it is located at the end of a sequence which has not been closed.

A sequence from a closed structure must start with a TR and end with a ST, or vice versa. The sequence to be cut must be marked in "Sequence mode". (See 'Mode' menu, 5.2.6 or 'Toolbar' 5.2.10).

The cut section is stored temporarily on the WINDOWS clipboard.

'Copy': The same applies as for "Cut". The difference is that the element or sequence to be copied from the structure remains where it is and is not cut out.

'Paste': The section stored on the clipboard with "Cut" or "Copy" is positioned at the active point in the GRAFTEC structure. The following window appears:



- 'Code': When clicked on (x), the AWL code of the pasted element or sequence is inserted with it.
- 'Comment': When clicked on (x), the comment of the pasted element or sequence is also accepted.

'Delete':

Any section marked in "Sequence mode", or the last individual element of a structure which has not been closed, can be deleted with 'Delete'. The same rules apply as for 'Cut' and 'Copy'.

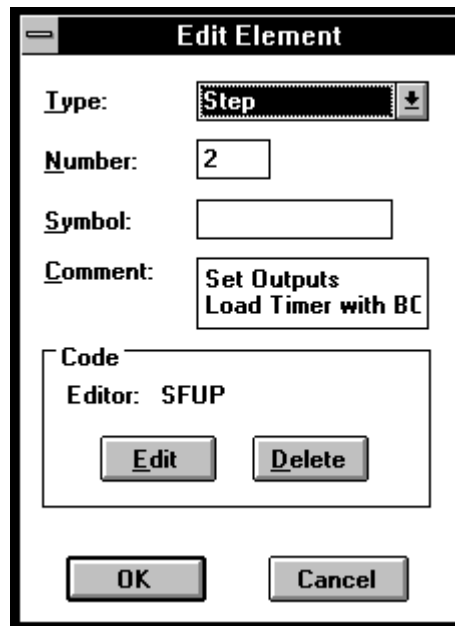
Before it is deleted, the following window appears:



If there is an error in marking the section to be deleted, the following message appears:



'Element...': Clicking here displays the following window:



This window refers to the current GRAFTEC element, i.e. to the element on which the cursor is positioned (red element).

- 'Type': "Initial step", "Step", "Transition" or "Macro step".

Here a step (ST) can be converted into an initial step (IST) and vice versa.

- 'Number': The ST or TR number is displayed. The number can be changed here. If a number is already in use, the following message appears:



'Symbol': The GRAFTEC element can be assigned a symbol name of max. 10 characters.

'Comment': A user definable comment of max. 40 characters can be edited on 2 lines.

'Code': Displays which editor has been selected for the current GRAFTEC element:

- "None": None
- "SEEDITWIN": AWL editor of the PG4
- "SFUP": FUPLA

'Edit': If "None" or "SEEDITWIN" is specified, "SEEDITWIN" is invoked and the GRAFTEC element can be edited in AWL.

If "FUPLA" is selected, FUPLA is invoked and the GRAFTEC element can be edited in FUPLA.

See the examples in section 5.5 for details.

'Delete': The code for this element is deleted. However, the following question appears:

"Are you sure?"

which should be answered with 'yes' or 'no'.

'OK': Accepts all the data in the whole window and exits the window.

'Cancel': Cancel. The window is exited without accepting the changes made.

'Code': Clicking on this submenu switches directly to the code editor defined for the current GRAFTEC element:

"SEEDITWIN" or "SFUP"

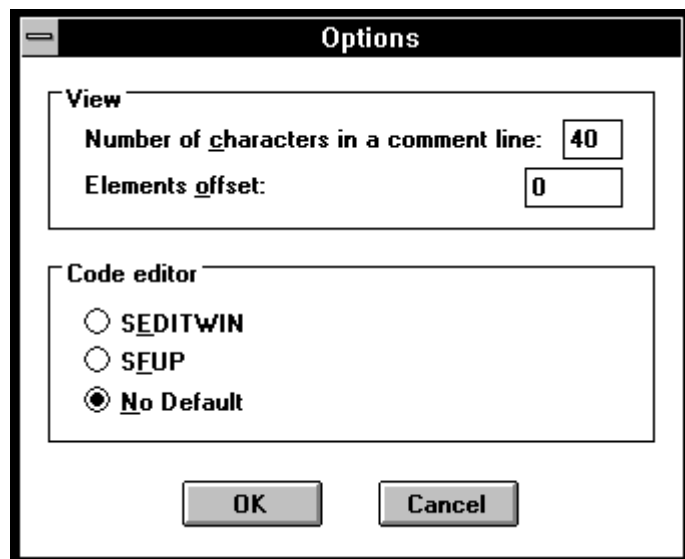
Has the same effect as 'Edit' in the 'Element...' menu.

'Renumber...': Clicking here displays the following window:



The whole GRAFTEC file can be renumbered, starting from the specified offset address. Renumbering starts from the top left of the diagram, continuing to the right and then down.

'Options...': Clicking here displays the following window:



View:

- 'Number of characters in a comment line:'

max: = 60

This value is stored for all subsequent work.

- '(I)Steps offset' (Offset der IStep- und Step-Nummern)

A base address can be specified here, from which the automatic numbering of (I)Steps subsequently edited will begin. (This specification is not stored).

- 'Transitions offset' (Offset der Transitionen-Nummern)

A base address can be specified here, from which the automatic numbering of transitions subsequently edited will begin. (This specification is not stored).

Code editor: Is used to define an editor for the whole file.

- "SEEDITWIN": AWL editor of the PG4
- "SFUP": FUPLA
- 'No Default': (no standard preset)

If a file is programmed partly in AWL and partly in FUPLA, 'No Default' should be selected. The following box then appears after clicking on 'Code':



The editor is selectable at each element.

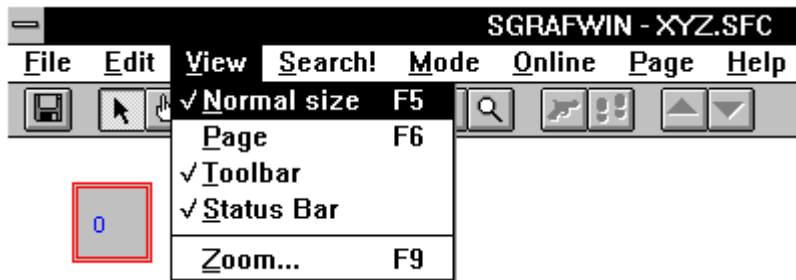
- 'OK': Accepts all the data in the whole window and exits the window.
- 'Cancel': Cancel. The window is exited without accepting the changes made.

'Export Sequence...': Similar to 'Copy'. However, the marked sequence is stored in a file "name.gbk" on the hard disk or other memory medium and is not put on the clipboard.

'Import Sequence': Similar to 'Paste'. However, the sequence to be inserted is read from a file "name.gbk" on the hard disk or other memory medium and is not obtained from the clipboard.

5.2.4 'View' menu

Clicking on 'View' displays the following sub-menu:

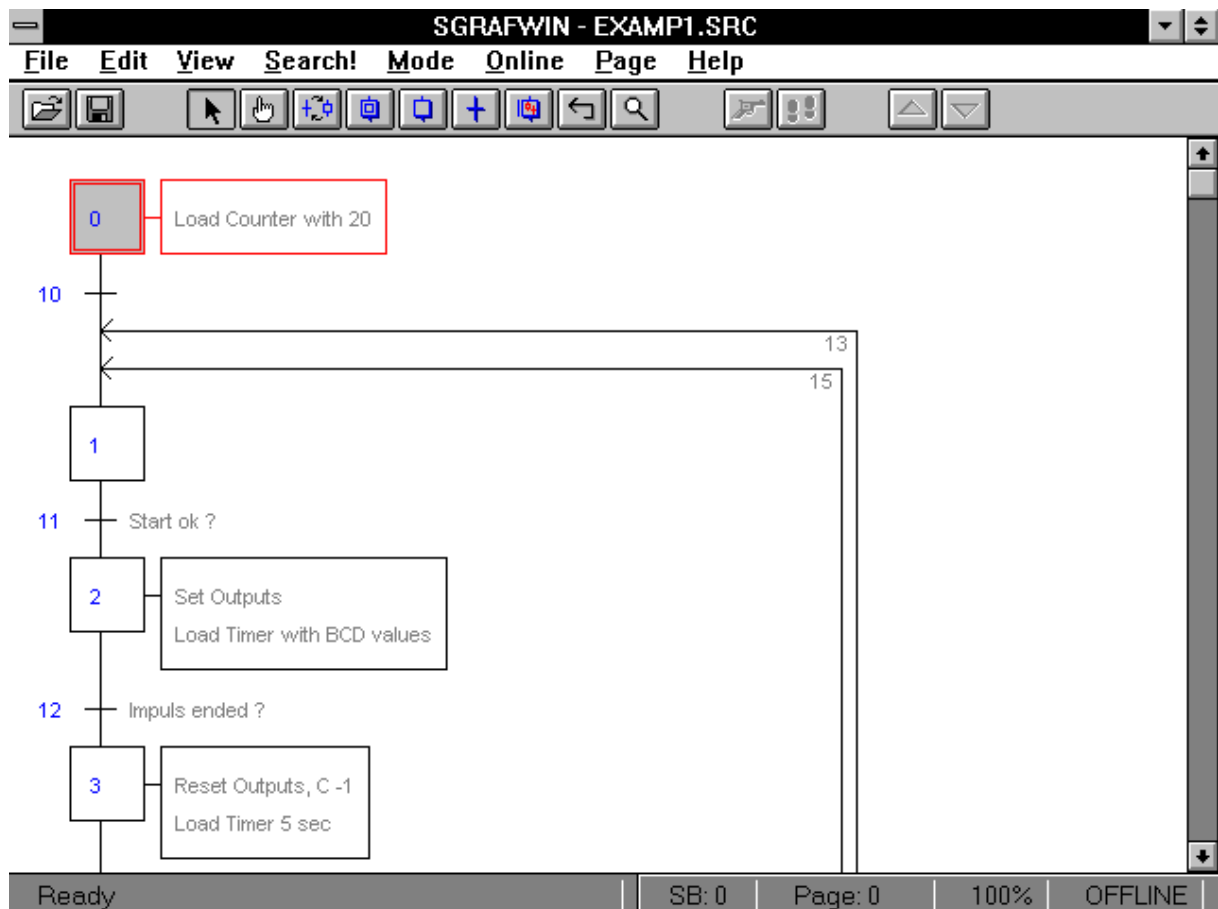


This menu optimizes the viewing of GRAFTEC files on the screen.

'Normal size': This format is set when GRAFTEC is invoked for the first time from the program manager.

The zoom factor is = 100.

Example of normal size view:

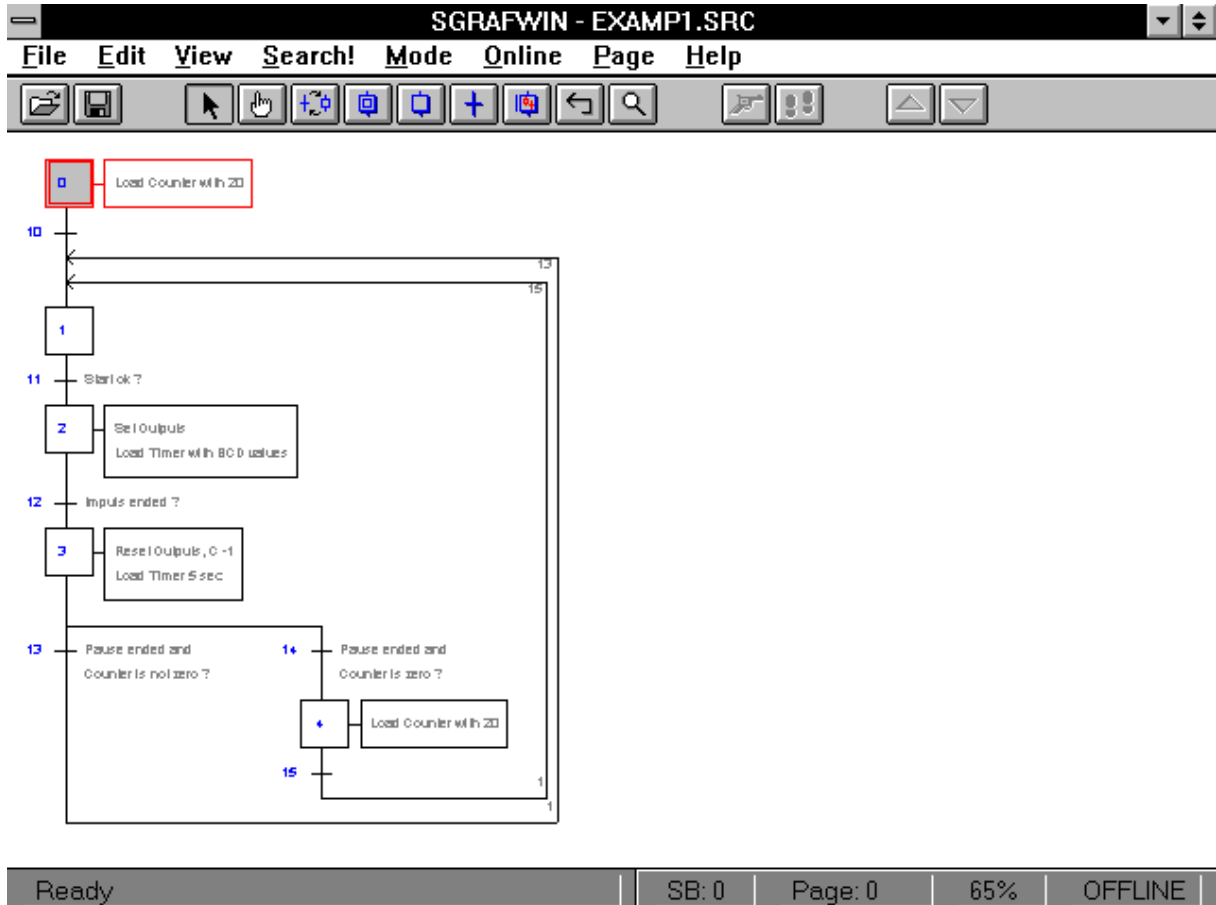


'Page':

This extremely convenient option represents the selected GRAFTEC file on a single screen page. The zoom factor is adjusted automatically.

Example viewing the same file as on the previous page, but in 'View Page' mode.

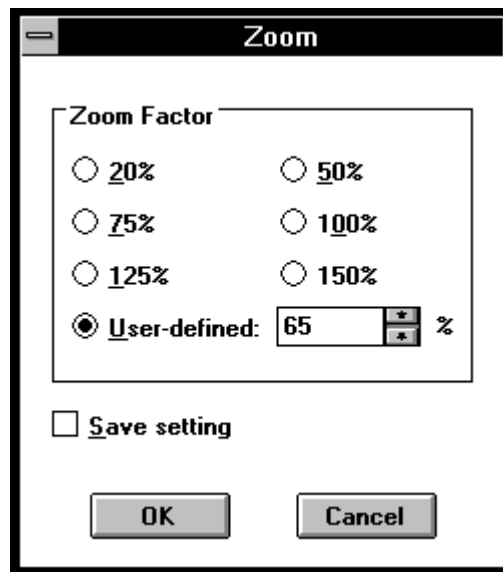
The zoom factor is now 65%. (Displayed on the status line)



'Toolbar': The Toolbar can be switched on and off.

'Status Bar': The Status Bar can be switched on and off.

'Zoom...': After clicking here the following window appears:

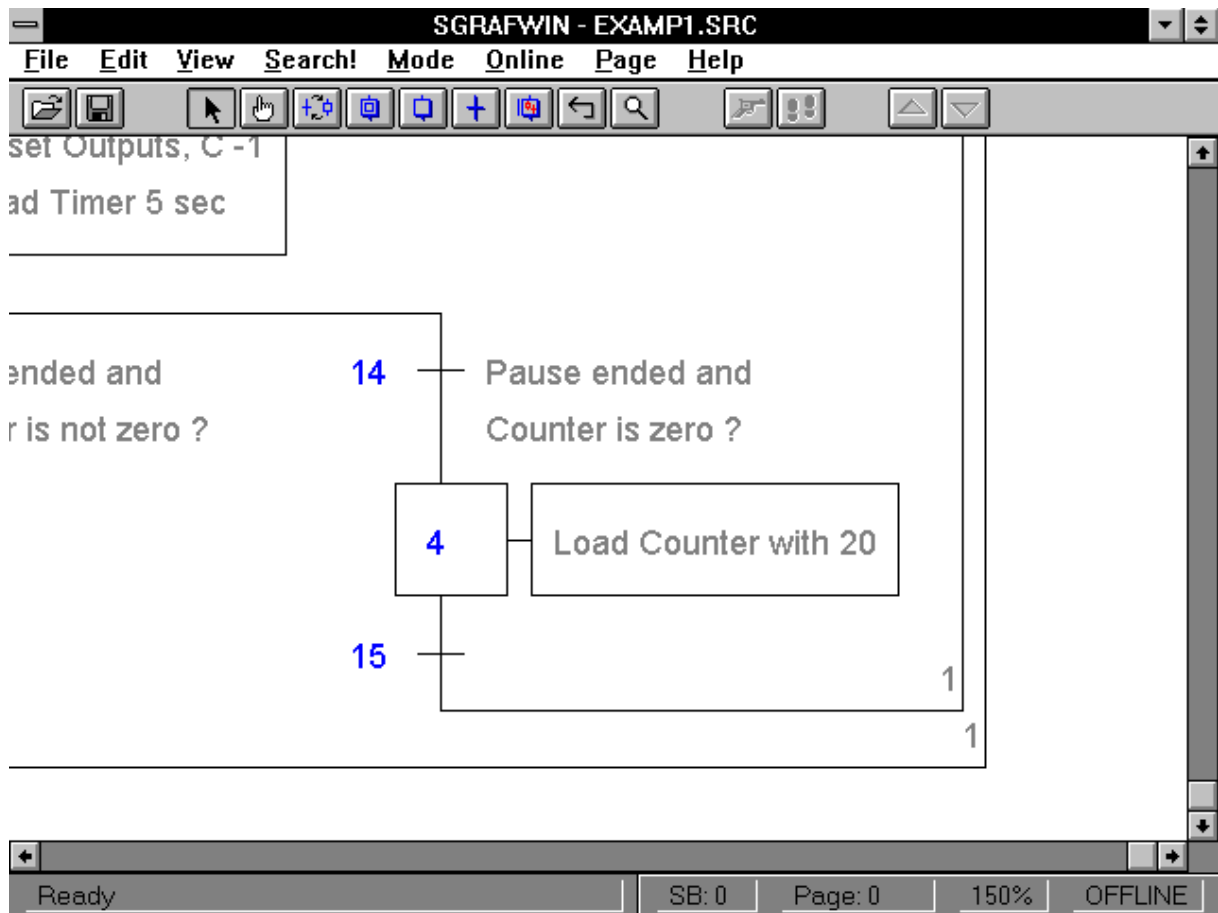


- 'Zoom Factor': A fixed zoom factor can be selected, or one can be typed next to "User defined" as a value between 20 and 150.
- 'Save setting': The zoom factor is saved. (x) When any file is subsequently selected, this zoom factor applies again, even if 'View Page' mode has been switched to in the meantime.

'OK': Accepts all the data in the whole window and exits the window.

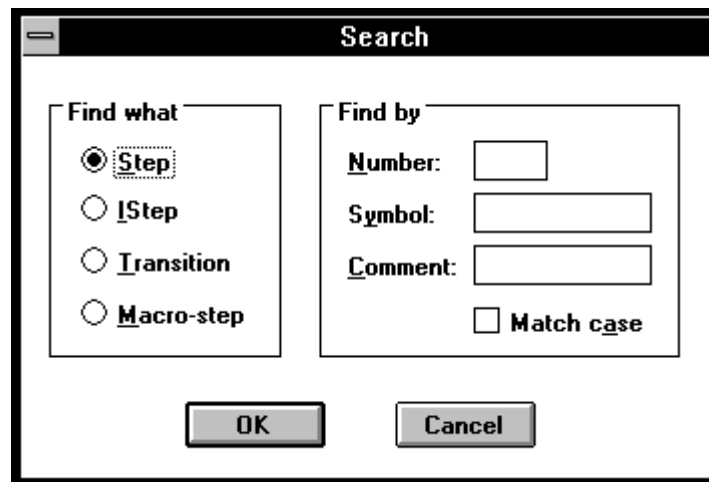
'Cancel': Cancel. The window is exited without accepting the changes made.

Example viewing the same file as on the previous page, but with a zoom factor of 150. The scroll bars can be used to scroll the displayed area.



5.2.5 'Search!' menu

Clicking on 'Search!' displays the following window:



This function is used to find GRAFTEC elements in an existing program.

The individual functions are explained below:

- | | |
|---|---|
| Find what: | What element type must be found? |
| <ul style="list-style-type: none"> • 'Step': • 'IStep': • 'Transition': • 'Macro-step': | <ul style="list-style-type: none"> Step (ST) Initial step (IST) Transition (TR) Macro step (Page) |
| Find by | What are the criteria by which the search should be made? |
| <ul style="list-style-type: none"> • 'Number': • 'Symbol': • 'Comment': | <ul style="list-style-type: none"> By GRAFTEC element number By GRAFTEC element symbolic name By GRAFTEC element comment, or part of that comment. |

- 'Match case': Upper/lower case in comments must match.

If this function is not enabled (no x), differences of case are disregarded in the search.

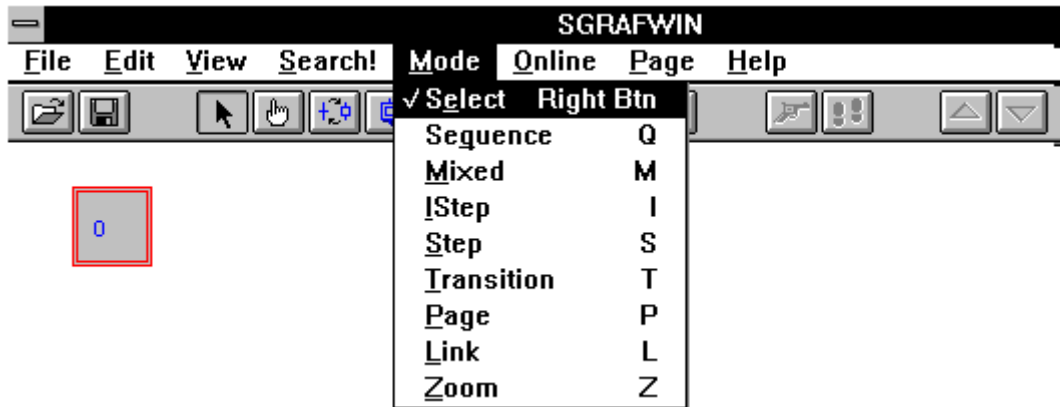
If the function is enabled (x), differences of case are significant.

For a successful search, only one of the 3 search criteria must be specified. If no search criteria are specified, the next element of the type selected (ST, TR, Macro step) is found.

If several search criteria are specified, the number has 1st priority, symbol 2nd and comment 3rd.

5.2.6 'Mode' menu

Clicking on 'Mode' displays the following sub-menu:



This menu selects the mode for editing a GRAFTEC structure. The form of the cursor can change depending on which mode is selected.

All the modes listed below can also be selected via the toolbar. The appropriate toolbar symbol is also shown here. The toolbar is described in section 5.2.10.

Brief description of the individual modes:

'Select':



If this mode is enabled, it is possible to select an element or connection from a GRAFTEC structure.

The cursor takes the form of an arrow for this mode.

'Sequence':



If this mode is enabled, it is possible to mark a sequence, i.e. a coherent part starting from the last selected element and ending at the point now clicked on in 'Sequence' mode.

The cursor takes the form of a hand in this mode.

'Mixed':

This mode is for editing chains of steps and transitions. Elements are added at the cursor position, alternating with each click: either a ST or, at the next click, a TR, then a ST again.

The cursor takes the form of a pencil in this mode. The symbol for the next element (ST or TR) is shown beside it.

'IStep':

The cursor setting is fixed at "IStep" (Initial step).

The cursor takes the form of a pencil in this mode. The initial step symbol is shown beside it.

'Step':

The cursor setting is fixed at "Step" (ST).

A ST is added at the cursor position. If a ST is inserted after a TR in a closed structure, another TR will also be added.

The cursor takes the form of a pencil in this mode. The step symbol is shown beside it.

'Transition':

The cursor setting is fixed at "Transition" (TR).

A TR is added at the cursor position. If a TR is inserted after a ST in a closed structure, another ST will also be added.

The cursor takes the form of a pencil in this mode. The transition symbol is shown beside it.

'Page':

The cursor setting is fixed at "Page" (Macro step).

A new (empty) page is inserted or added at the cursor position.

The cursor takes the form of a pencil in this mode. The page symbol is shown beside it.

'Link':

Mode for linking parts of structures or for closing loops.

The cursor takes the form of an arrow in this mode.

'Zoom':

The cursor takes the form of a hand with magnifying glass in this mode.

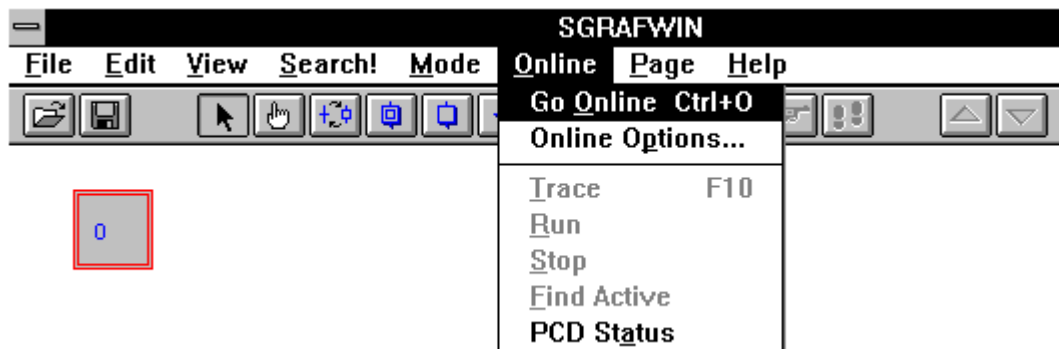
A part of the GRAFTEC structure can be marked and then maximized to full-screen size.

The cursor is positioned in the top right corner of the selected section. The left-hand mouse button is then pressed and held down. The mouse is dragged down and to the right until the section reaches the desired size. Releasing the mouse button displays the marked area in maximized form.

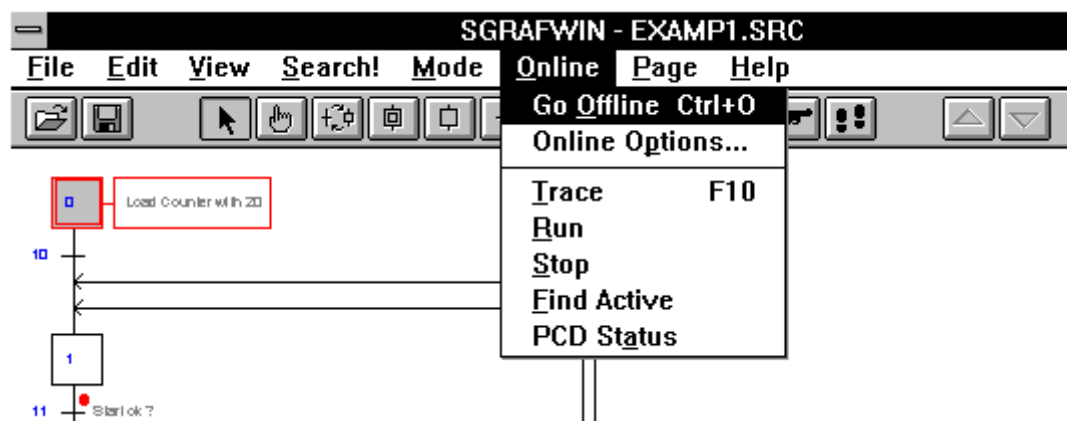
The zoom factor is shown on the status line. (max: 999)

5.2.7 'Online' menu

If the PCD is offline, clicking on 'Online' displays the following submenu:



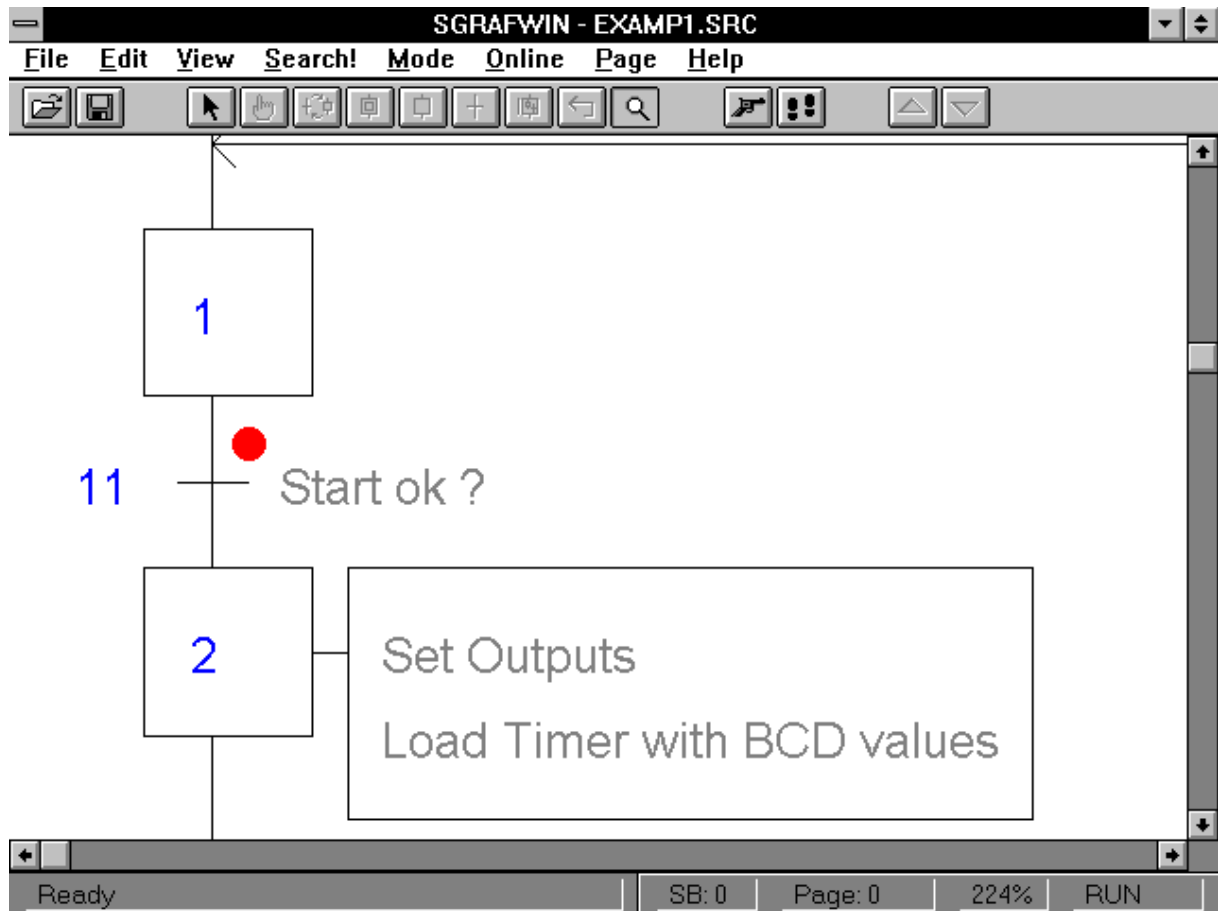
If the PCD is connected online with the programming unit, clicking on 'Online' displays the following submenu:



It is possible to communicate with the PCD via this menu. This depends on the correct connection of the programming unit to the PCD. The downloaded user program and the program on the screen must be the same. The PCD must also be switched on.

'Go Online':

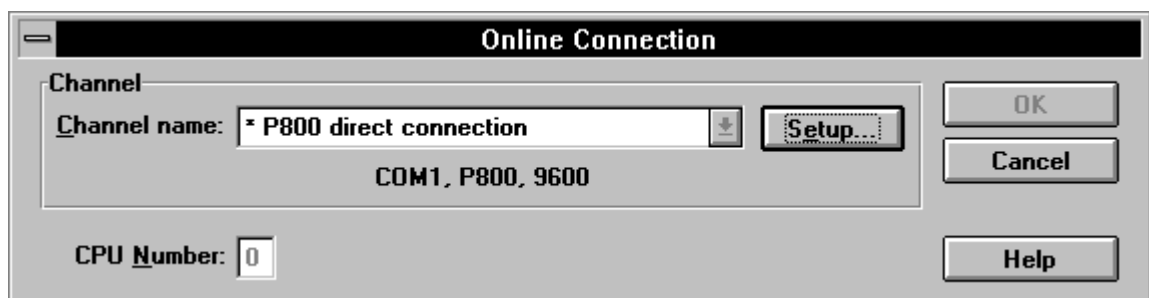
If all the above conditions are met, clicking on 'Go Online' establishes the connection. It is now possible in the GRAFTEC structure to view the running of the program or process. The active transition(s) are marked with a red spot (colour screen required).

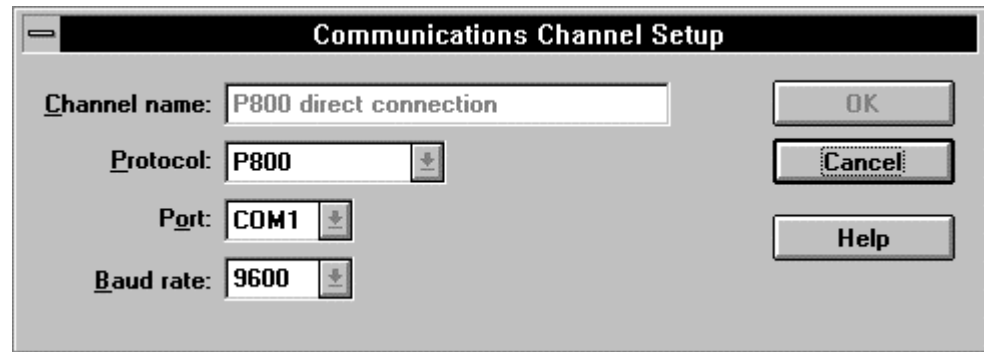


On this screen, TR 11 is active. If the TR is fulfilled, i.e. at the end of the TR the ACCU = H, the next ST is executed and the next TR (12) becomes active. The red spot goes directly to the next TR.

'Go Offline': It is possible to switch back from "Online" to "Offline" mode.

'Connection...': The type of the connection between the programming unit and the PCD is shown.





A connection with the P800 protocol always works.

If the connection is to be made in the S-Bus protocol, the PCD must first be configured. When this has been done correctly, it is possible to switch between both protocols online.

'Trace':



Each time 'Trace' is activated, the process runs to the next TR and stops there until either 'Trace' or 'Run' is selected again. In this way the GRAFTEC structure is worked through step-by-step, not just the individual instruction lines.

Please note:

'Trace' only switches forward when the TR has been fulfilled. However, this means that the next ST has also been executed. The ST after the active TR has therefore already been executed!

The 'Trace' and 'Stop' functions can also be executed by clicking on the toolbar key shown here on the left.

'Run':

The CPU is switched into 'Run'. The program can be viewed running online.



'Run' can also be executed by clicking on the toolbar key shown here on the left.

'Stop':

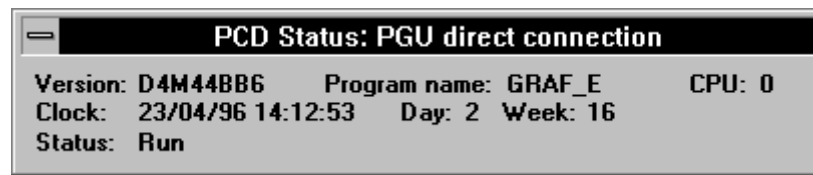
The CPU is stopped. The program can be run again with 'Run' or 'Trace'.

'Find Active':

The red spot indicates the active TR. If the GRAFTEC structure exceeds one screen page, the active TR may not be visible. By clicking on 'Find Active', the currently active TR is moved into the picture and marked in red. If the active TR is on another page, that page is opened and the actual active TR is shown, not just the page specified in which the active TR can be found.

It is, of course, also possible to zoom the GRAFTEC structure ('View' - 'Page'). The active TR or page is visible. In the case of a very large structure, however, the red spot can become so small that the present function 'Find Active' again becomes extremely useful.

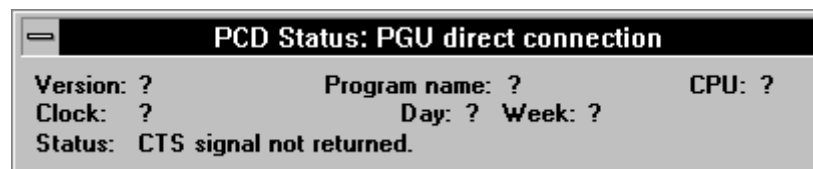
'PCD Status': The following window appears:



The window shows:

- Protocol: PGU direct connection (P800)
- CPU type: PCD4.M44
- Firmware version: BB6
- CPU number: 0
- Programm Name GRAF_E
- Date/time 23/04/96 14:12:53
- Day of week 2 (Tuesday)
- Week 16
- Status Run

If the connection between the programming unit and the PCD is not OK, or if the PCD is not switched on, the following dialog box is shown:

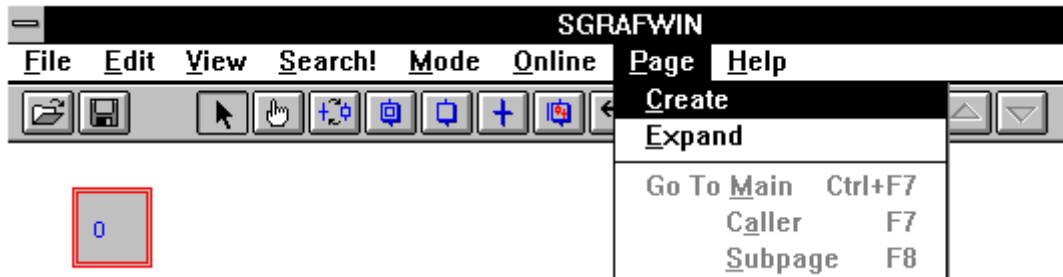


If we try to go online and the connection can not be established the following message is displayed:



5.2.8 'Page' menu

Clicking on 'Page' displays the following submenu:



This menu allows pages (macro steps) to be created and worked with generally. Refer to the "User manual for the PCD family" (Publication 26/732), section 6.2 for the theory of these pages.

It is possible either to define pages early, when the GRAFTEC structure is being edited, or later, by combining coherent structural parts of the finished structure to form pages. In an existing structure, a page must always start with a ST and end with a ST. The page can only have 1 input and 1 output. Pages can be nested to practically any depth.

If a page is to be created in a new structure, the simplest way is to click on the page symbol in the toolbar to insert a page anywhere in the structure.

The present 'Page' menu is mainly used when pages are to be created from an existing structure.

- **'Create':** The sequence to be declared as a page is marked. Clicking on 'Create' produces a page from the marked sequence.

If the 'Create' command is used on a ST, it is converted into a macro step.

- **'Subpage':** The 'Subpage' menu item is now visible. Clicking on this item quits the structure from which the page was created and the page itself is displayed.

- **'Go to Caller':** If we are on a page or subpage, the 'Go to the point at which the page was called.
- **'Go to Main':** If we are on a page or subpage, the 'Go to Main' menu item is visible. Clicking on this item results in a return to the main structure from which the route to the page originated.
- **'Expand':** 'Expand' undoes the created page, i.e. the page structure is reintegrated into the main structure, or into the next level closer to the main structure.

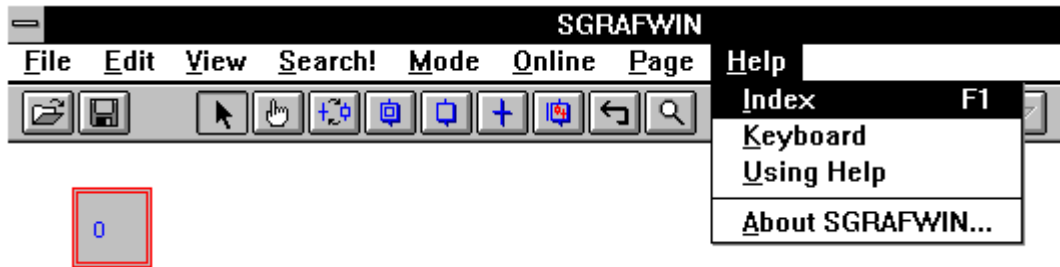
This 'Page' menu is useful above all for the retrospective creation of pages, or for undoing (expanding) them. The two arrows on the far right of the toolbar provide a much more user friendly way of viewing a GRAFTEC program.



As soon as the cursor lands on a macro-step (page) in the main structure, the down-arrow becomes active (yellow). Clicking on it opens the page. We are now in the page structure and the up-arrow becomes active. Clicking on the up-arrow returns us to the main structure. The same applies, of course, if a subpage is called from a page. Both arrows are then active and it is possible to switch in both directions (nested pages).

5.2.9 'Help' menu

Clicking on 'Help' displays the following sub-menu:



The 'Help' files are in English. They will be available in German at a later date. In the meantime, help in other languages can be obtained from the different language versions of this manual, the user manual for the PCD family, and from the standard WINDOWS 'Using Help' file.

- 'Index':

Commands

File Menu	:	
Edit Menu	:	
View Menu	:	Section 5.2
Search Menu	:	
Mode Menu	:	of this
Online Menu	:	
Page Menu	:	manual
Help Menu	:	
Toolbar	:	

Managing the GRAFTEC structure

Editor Functionalities	see user manual for the
Illegal Structures	PCD family

Messages

- Warning Messages
- Error Messages

- **'Keyboard': Keys**

- Cursor Movement Keys
 - Deleting Keys
 - Dialog Box Keys
 - Editing Keys
 - Help Keys
 - Menu Keys
 - System Keys
 - Text Selection Keys
 - Window Keys
 - Shortcuts

- **'Using Help':**

- Standard WINDOWS file

- The language of this file depends on which WINDOWS language version has been installed.

- **'About SGRAFWIN':**

- The following window appears:



The programming tool version is displayed.

There is also a list of the GRAFTEC elements used in the program invoked.

The space left in memory on the current drive is shown.

- 'More Info...': The following window appears:


















In addition, the used GRAFTEC elements are listed.

5.2.10 Toolbar

The toolbar contains the functions most frequently used when working on a GRAFTEC file. Clicking on one of the function symbols executes the function directly or introduces it.

Most functions have already been described in sections 5.2.6, 5.2.7 and 5.2.8. In this section, therefore, only the keyword for each symbol is given.

		File open button
	<F2>	File save button
	right mouse button	Select button
	<q>	Sequence button
	<m>	Mixed button
	<i>	IStep button
	<s>	Step button

	<t>	Transition button
	<p>	Page button
	<l>	Link button
	<z>	Zoom button
		Run button
	<F10>	Trace button
	<F7>	Caller Page button (1 level nearer to the main structure)
	<F8>	Subpage button

5.3 Using GRAFTEC, introductory example

With the help of some examples, this section demonstrates how to create a GRAFTEC program, edit the structure, and view the process online on the screen. Knowledge of GRAFTEC structures and FUPLA is assumed. Familiarity with the menu items in the preceding section (SGRAFWIN) is also assumed.

Creating a new GRAFTEC program

The separate steps are explained using a blinker and entering some external values.

The project name is

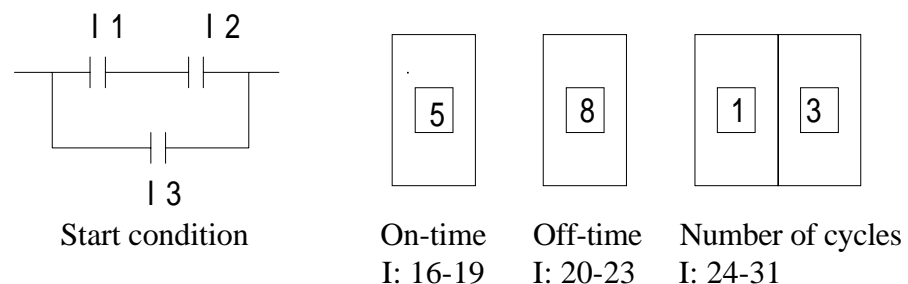
"GRAF_NEW"

The file name of our 1st example is

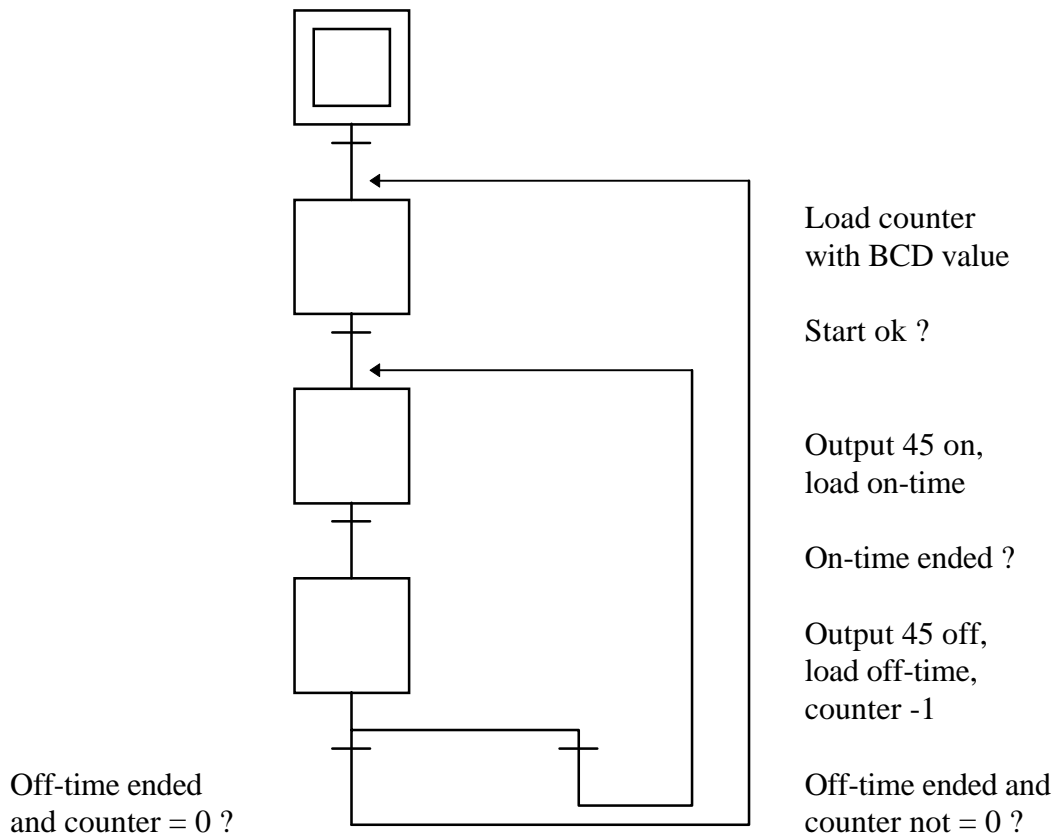
"BLINK_1"

Task for the 1st example:

After switching on inputs 1 AND 2 OR input 3, output 45 should blink a certain number of times. The off and on times should be preset individually between 1 and 9 sec by a 1-digit BCD switch. The number of blink cycles should be preset by a 2-digit BCD switch (01-99 blink cycles).



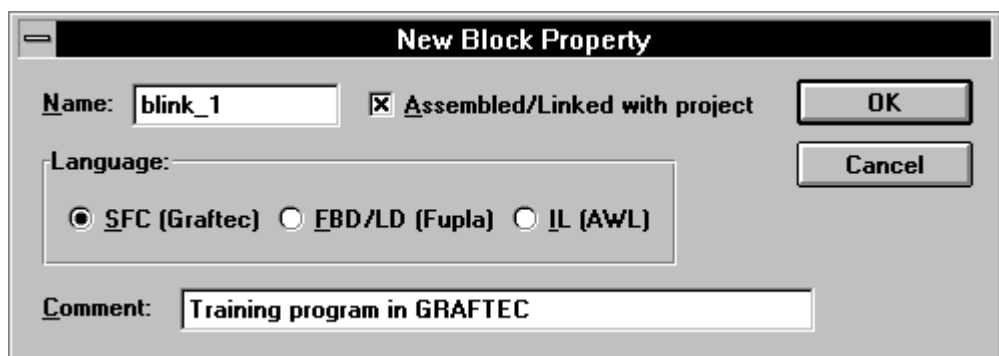
The GRAFTEC structure is as follows:



This structure must now be reproduced on the screen to create the program.

- 1.) In the project library, we select 'File' - 'New...' to open project "GRAF_NEW"
- 2.) Enter the project manager by double-clicking on our project "GRAF_NEW" in the projectlist.

We specify as follows:

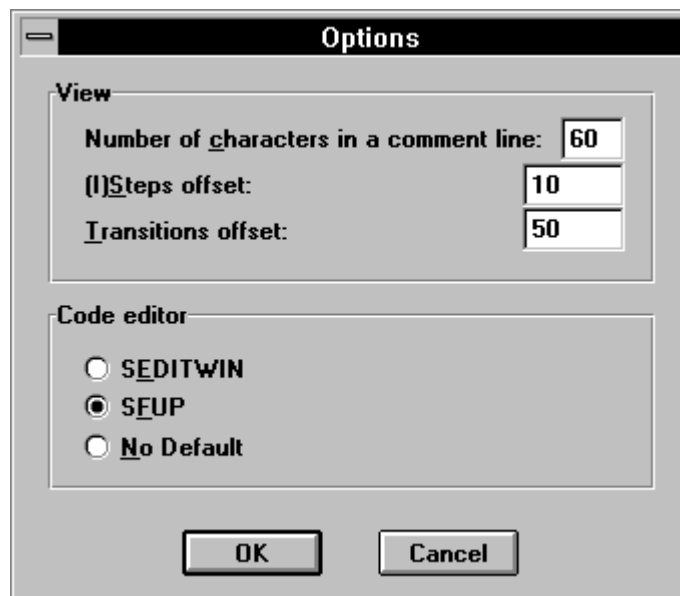


At "Language" it is important for us to select 'SFC (Graftec)'.

- 3.) Call the GRAFTEC editor by double-clicking on our file name "blink_1.sfc" in the project list of the project manager.
- 4) Before we start actual editing of the GRAFTEC structure, we select a few options:

'Edit' - 'Options...'

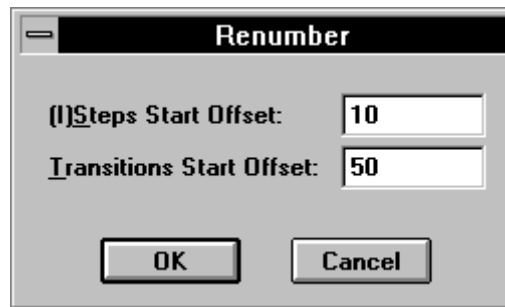
- Select the maximum of 60 characters for each comment.
- Define the (I)Step offset as 10
- Define the transition offset as 50
- Select "SFUP" as the code editor



and confirm with 'OK'.

- 5) We now edit the GRAFTEC structure.

Since IST 0 is drawn automatically when the GRAFTEC editor is called, it will already have been designated as "0" before we executed the "(I)Step" offset. Before editing starts, the IST should be renumbered as "10". This is achieved by 'Edit' - 'Renumber...'. The following window appears:



After 'OK' the IST will have the number 10.



Click on the "Mixed" button from the toolbar.

The cursor appears as a pencil with the TR symbol. Bring the tip of the pencil (cursor) roughly into the middle of the lower edge of the already (automatically) drawn IST and click.

TR 50 is drawn in. The cursor now shows the ST symbol, which indicates the next element. Click on the lower end of TR 50. ST 11 is drawn, etc.

When editing a GRAFTEC structure, it is worth drawing the longest chain of steps first, and only drawing in the branches, cross connections and return connections later.

Continue until TR 53.



To draw a return connection to ST 11, select "Link". TR 53 is still marked (last element edited).

Bring the tip of the cursor to the middle of the upper edge of ST 11 and click. The connection is drawn.

If the message

"Link not possible here"

appears, try again. The cursor must be positioned very accurately here.

The alternative branch to TR 53 must now be drawn.



Select the "Transition" function. The tip of the cursor must be positioned in the middle of TR 53 and clicked on. The new TR 14 is now located to the right of TR 53.

Use the "Link" function again to draw the reverse connection to ST 12.

Our GRAFTEC structure is now complete. If the zoom factor has been left at 100%, the structure drawn will be a little larger than the screen.

By selecting

'View' - 'View Page'

the size of the structure is adjusted to fit the screen in the best way. By selecting

'View' - 'Zoom...'

you can choose the old factor (100%) or another one.

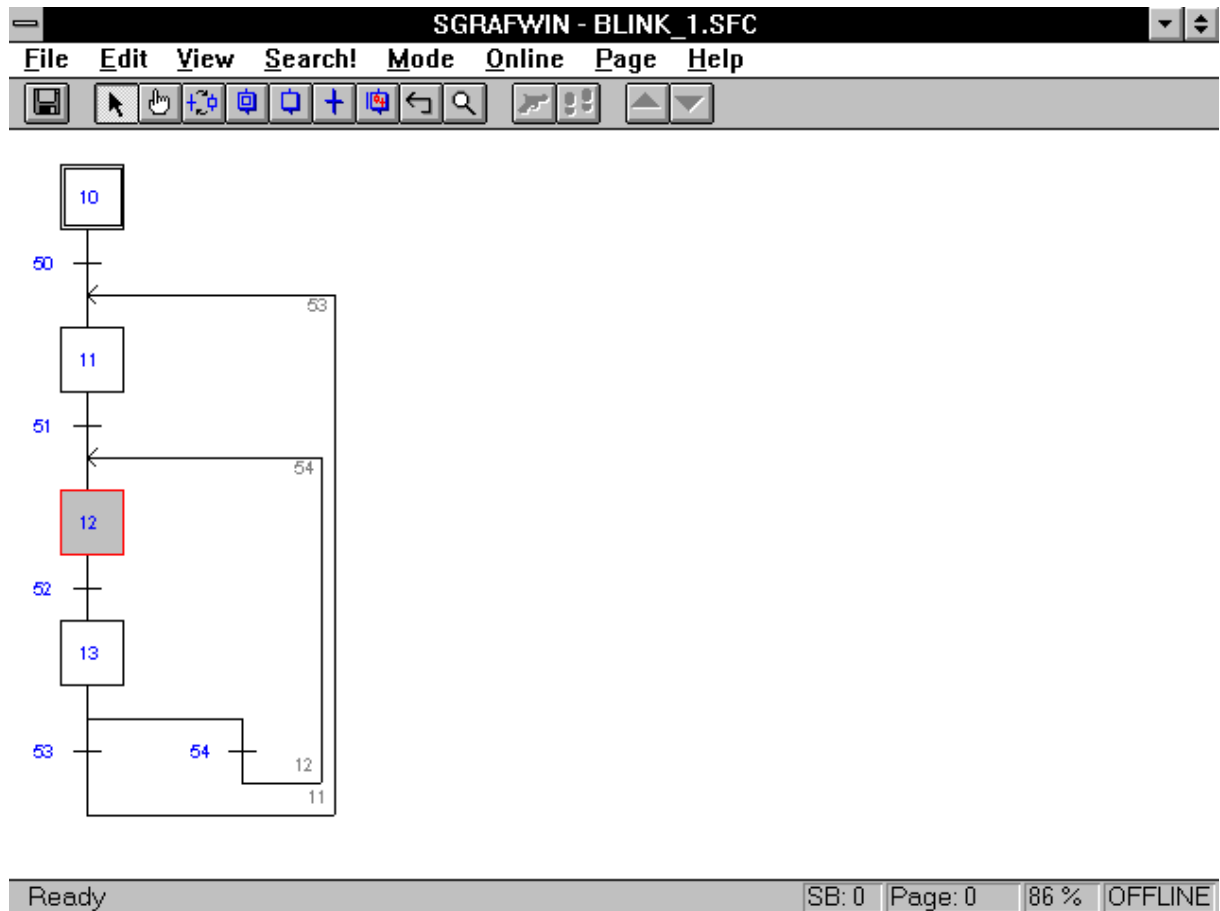


By clicking on the "Zoom" button, you can mark an area (drag the mouse from top left to bottom right while pressing the left-hand mouse button). The marked area is maximized to full-screen format. This function is particularly useful for the more complex structures.

Try it.



As a precaution, save the structure by clicking on "File save".



6) Editing comments

Start editing by clicking, for example, on ST 11. Select

'Edit' - 'Element',

then click into the "Comment" field, write the comment (max. 60 characters on 2 lines) and close with 'OK'. The result is shown immediately on the diagram. Edit the other STs and TRs in the same way and save everything again by clicking on "File save".

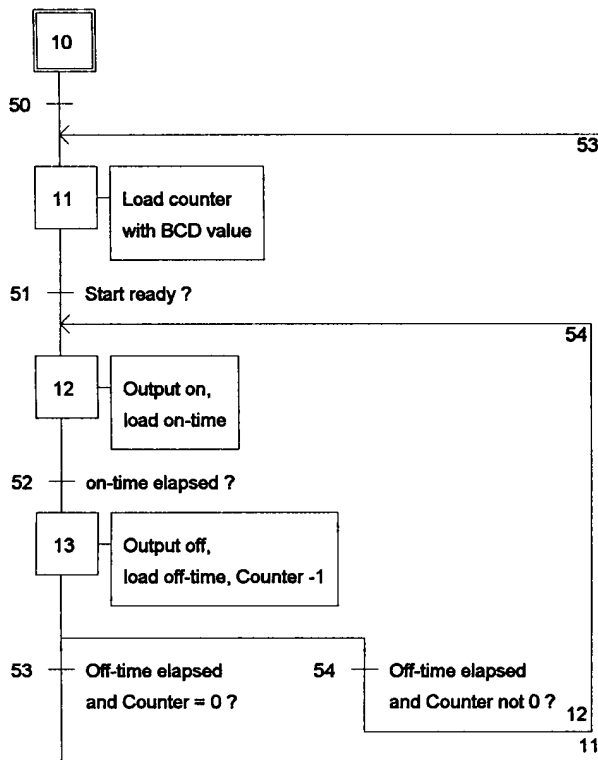
- 7) Before coding the individual STs and TRs, print out the structure with comments.

'File' - 'Print...'

At 'Define Header' it is possible to give this example a title and subtitle. Do this.

The printout then appears as follows:

USER : FOR SAIA'S INTERNAL USE ONLY EDITOR : SGRAFWIN V1.3 FILENAME : C:\PG4\PROJECTS\GRAF_NEW\BLINK_1.SFC DATE : 23.04.96	TITLE : GRAFTEC example SUBTITLE : Blinker with preselection SB : 0 PAGE : 0
---	---



This printout is the core of the GRAFTEC program. With clear comments and a little knowledge of GRAFTEC rules, it should be possible to understand how the installation or partial installation works.

If one also manages to show the structure running online, i.e. to see which point is active and possibly even to look inside one or other important element (ST, TR) while it is online, then we have a tool which provides even a novice with a clear insight into the operation of a process.

However, before we reach that stage, the program must be provided with program code. This can be done in different ways.

8) **Editing the program code**

Different possibilities are available for this:

- Code in FUPLA
- Code in AWL (instruction list)

As already mentioned in the PG4 introduction (1.4.3), in the PG4 it is preferable to edit individual STs and TRs in KOPLA or FUPLA.

Leave IST 10 and TR 50 empty. These two elements currently only serve the purposes of the GRAFTEC structure.

Go to ST 11 and klicken once. ST 11 is selected.

'Edit' - 'Code'

or a double click on ST 11 will call an empty FUPLA page.

9) The actual programming can now be started.

Double click on ST 11. An empty FUPLA page appears. The format can be resized appropriately. The position and size of the FUPLA window will also be saved.

In ST 11 a counter is to be loaded with a 2-digit BCD value. ST 11 is edited as follows:

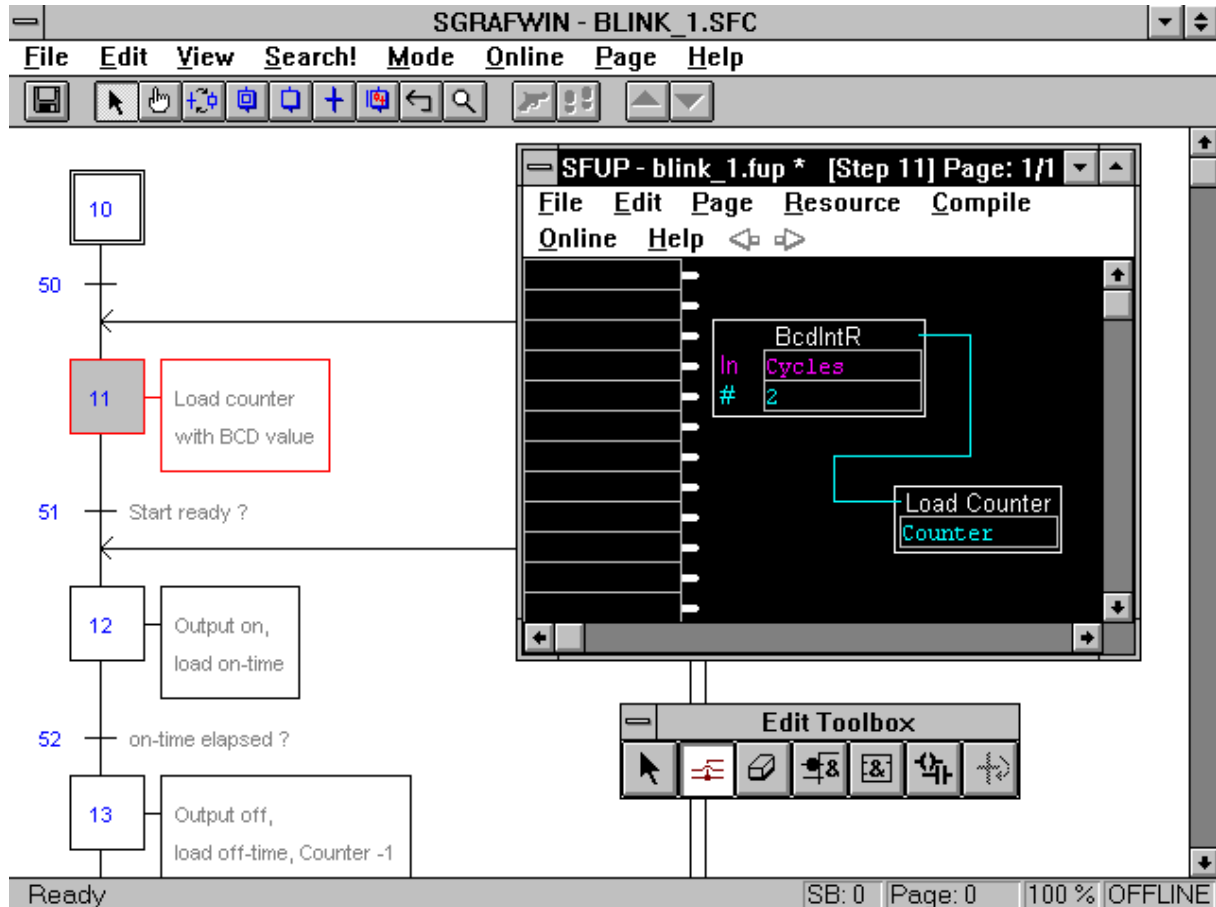
After calling the toolbox with 'Edit' - 'Tools', the 'Load counter' function is selected from the 'Graftec' family and the FBox is positioned. The counter is to be loaded with a BCD value. From the 'Converter' family, select the 'BCD to int quick' (or 'BCD to int reverse quick') function and position it also.

The converter output must be connected to the counter input.

The address fields must be completed. It is possible to choose absolute or symbolic addresses. At the converter, type against "In": <Cycles> (as a symbolic name) and against "#": <2> for 2 digits.

Give the counter the symbolic address <Counter>.

The screen now appears as follows:



Leave the symbol name "Counter" as it is, i.e. do not define any counters with a fixed address. When the file is compiled, a counter will be assigned automatically from the dynamic resources.

At "Cycles" a definition must be made, since the compiler cannot know where, i.e. from which hardware inputs, the BCD information is obtained. While the "In: Cycles" field is still active, select from the FUPLA menu bar

'Resource' - 'About label: Cycles'

and define:



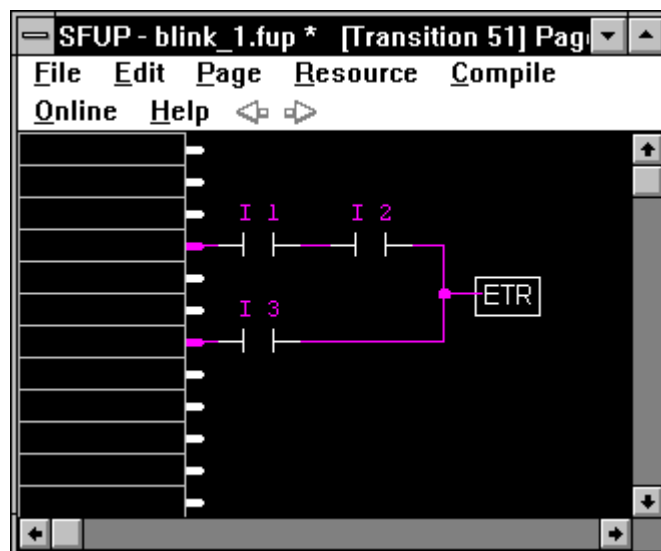
Go now to TR 51. Click here to display a new, empty FUPLA window. It is at this point that the start condition must be programmed. Select KOPLA.

Three inputs are position according to the function demanded.



A TR must be terminated with the 'End of transition' (ETR) function from the 'Graftec' family. This function can also be taken from the toolbox (far right). In order for a program to run from a TR towards the next ST, the input at FBox 'ETR' must be high.

The 3 inputs and the 'End of transition' FBox are connected and, as required by the job specification, the inputs are provided with absolute addresses.

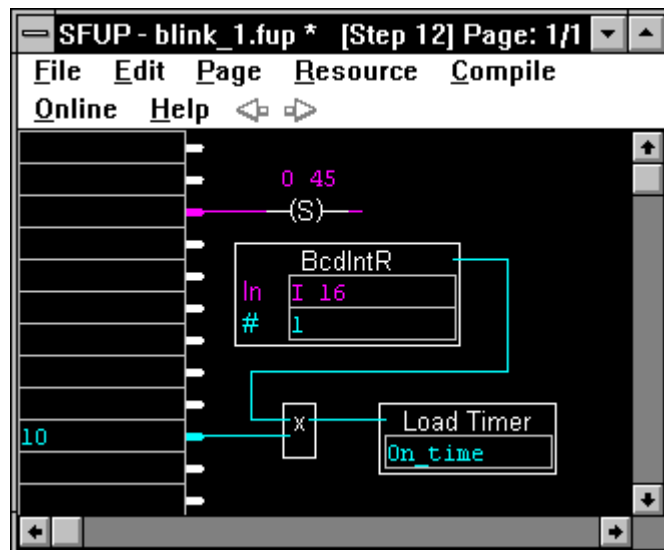


This FUPLA fragment can now be saved, or we can edit the next GRAFTEC element directly. FUPLA remains in the background, i.e. even while editing the next element, we stay in the same FUPLA file. To save, select

'File' - 'Save'

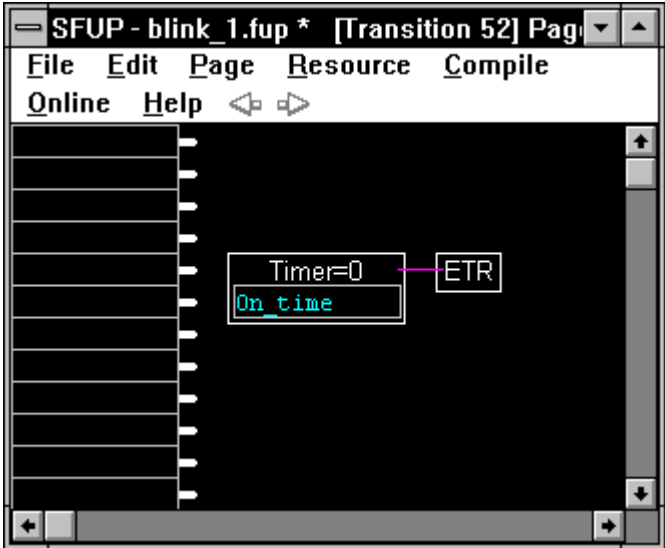
At ST 12, set the output <O 45> with the appropriate KOPLA function.

We read the BCD information from input 16 (to 19) as 1 digit. Since the timer is to be loaded in seconds, while in FUPLA the time interval is always 1/10 sec., we multiply the BCD value by 10 and supply this value to the 'Load timer' FBox from the 'Graftec' function family. The multiplication constant <10> is typed directly into the appropriate input field.



At TR 52 we wait until the on-time is reached with the 'Timer is zero' function from the 'Graftec' function family.

Each TR must be terminated with the 'End of transition (ETR)' function.

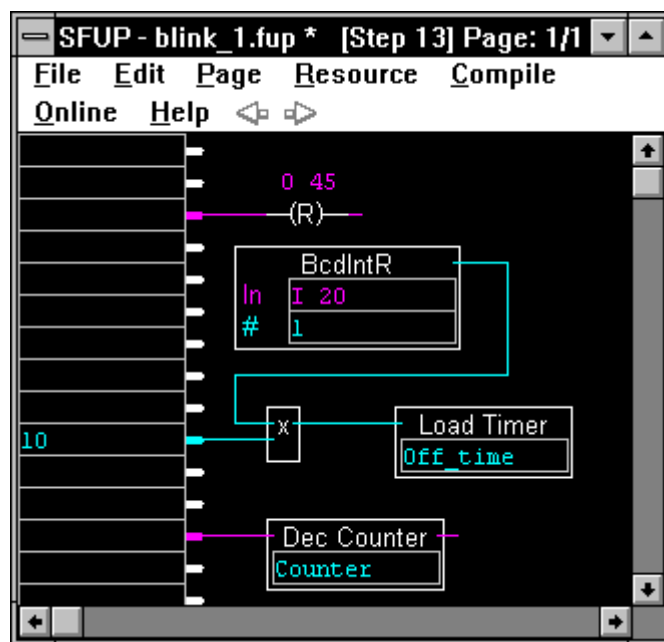


ST 13 is very similar to ST 12.

Output <O 45> is reset with the KOPLA function.

The off-time is supplied to the timer using the available BCD value multiplied by 10.

In addition, the counter for the number of cycles (which has the symbol name <Counter>) is decremented using the 'Decrement counter' function from the 'Graftec' function family.

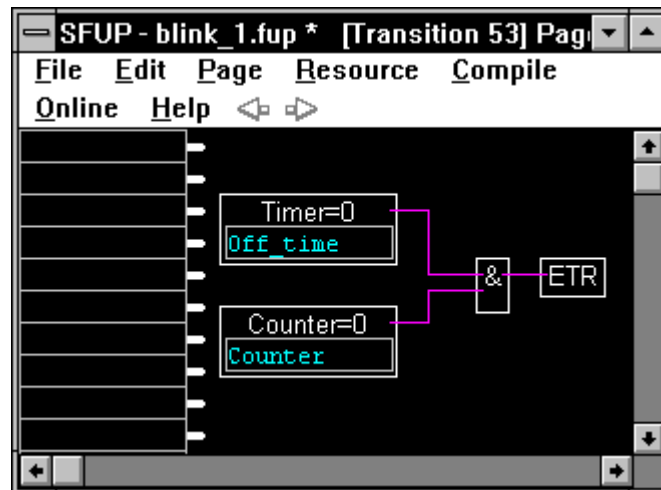


TR 53 has the task of registering the end of the off-time period and, in addition, determining whether the cycle counter has zeroed.

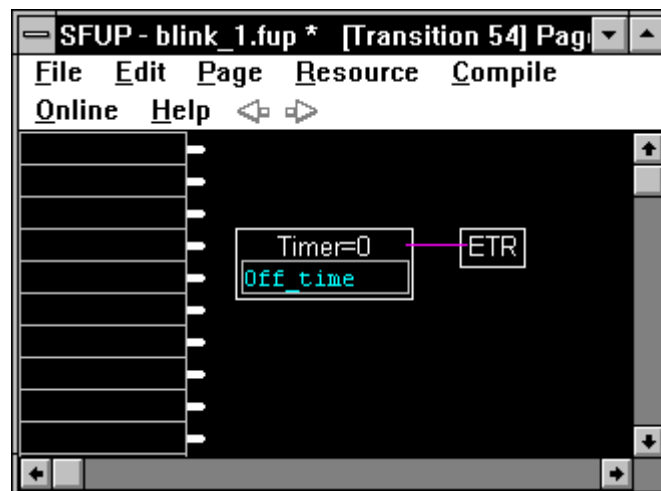
The end of the off-time period is again registered with the 'Timer is zero' function while the 'Counter is zero' function does the same for the counter. Both functions belong to the 'Graftec' function family.

The AND operation on both items of information is achieved with the 'And 2-10 inputs' function from the 'Binary' function family.

Each TR must conclude with the 'End of transition' (ETR) function. This TR is therefore only fulfilled when both the timer 'Off-time' is finished AND the counter 'Counter' has zeroed. Only then will the 'ETR' function input become high.



TR 54 only registers the end of the off-time period.



All GRAFTEC elements necessary for the program to run have now been programmed.

Looking at the resource list reveals the following:

	Symbol	Value	Type
	I	1	Local
	I	2	Local
	I	3	Local
	I	16	Local
	I	20	Local
Cycles	I	24	Local
	O	45	Local
Counter	C	<Dynamic>	Local
Off_time	T	<Dynamic>	Local
On_time	T	<Dynamic>	Local
	K	1	Local
	K	1	Local
	K	2	Local
	K	10	Local
	K	10	Local

The "On-time" and "Off-time" timers, and the "Counter" are listed as "dynamic", i.e. their allocation is automatic from the "Dynamic variables" available to the compiler. All other resources have a fixed address or fixed value.

- 10) Now that the program has been created, it must be compiled.

From a GRAFTEC resource (ST or TR) edited in FUPLA, click on the FUPLA menu bar

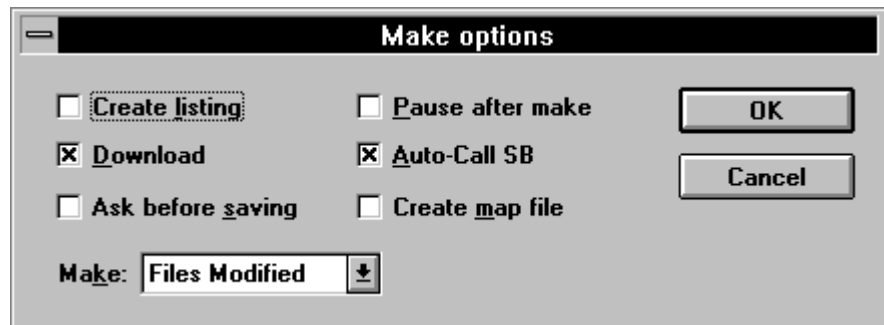
'Compile' - 'Compile'

If no error message is displayed, compiling has been successful and the file "blink_1.fbd", plus some auxiliary files, have been generated in the background.

- 11) The program must now be assembled, its parts linked, downloaded to the PCD, put into Run and switched online.

This takes place with the "Make" tool.

We return to the project manager and select 'Make' - 'Make Options...' auf.



The 'Auto-Call SB' option must be enabled, so that the GRAFTEC program, which is located in an SB (sequential block), will also be called and incorporated in the course of the program.

- 12) If all this has been done and the PCD is connected via the correct interface to the IBM PC (and switched on), it is possible to go to a GRAFTEC resource edited in FUPLA and select:

'Compile' - 'Make'

Do this from TR 51 (Start condition OK?)

Assembling and linking are viewed in a window. As long as there are no errors, this window has no significance. If an error occurs (for example, if any symbols have not been correctly declared or if "Auto-Call SB" has not been enabled) important information can be seen here.

```

PG4 MAKE UTILITY
SAIA PCD MAKE UTILITY V1.3 Beta
An auto-call for SB 0 was generated.

SAIA PCD MACRO ASSEMBLER $197
Assembling: SB041023.SRC
To:        SB041023.OBJ

Free memory: 379600

Assembly complete, 0 warnings, 0 errors

SAIA PCD LINKER $197
Linking: BLINK_1.OBJ+SFUPDEBUG.OBJ+SB041023.OBJ
To:      GRAF_NEU.PCD

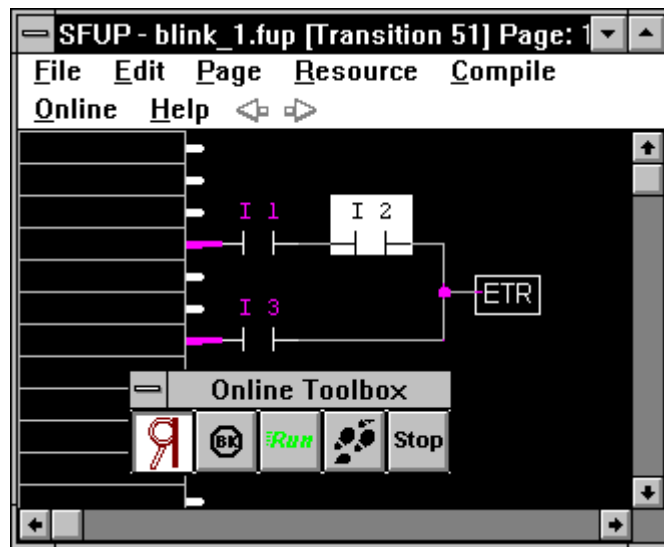
Free memory: 343264
Code size: 201 lines (804 bytes)
Text size: 0 bytes
Exten mem size: 0 bytes (Init size: 0 bytes)
Global symbols: 5

Linkage complete, 0 warnings, 0 errors

```

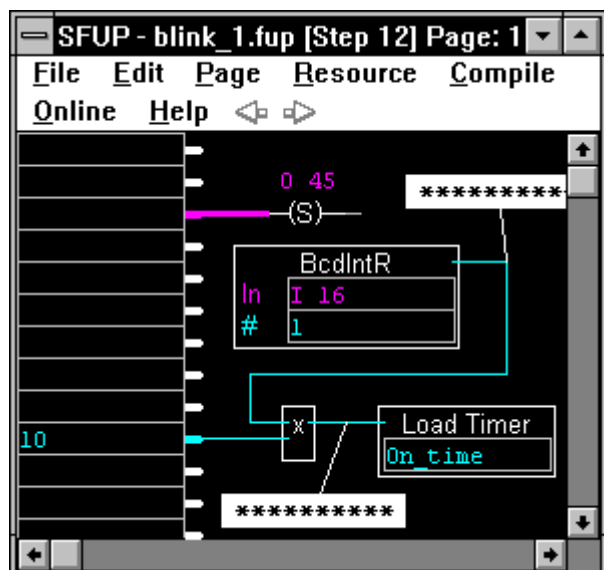
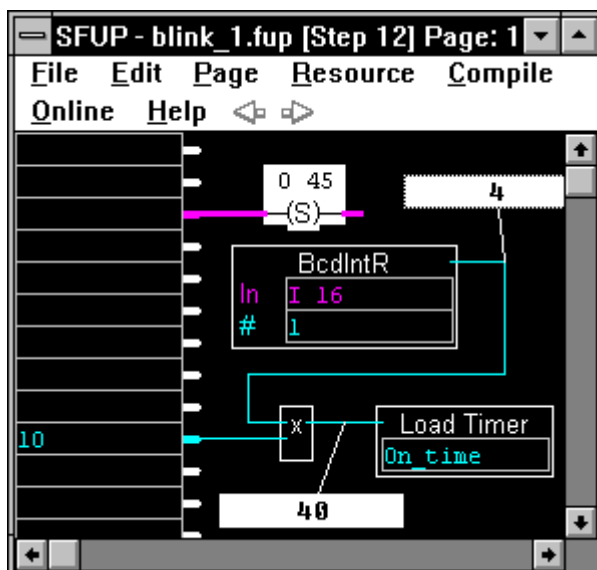
If everything has gone smoothly and if the options have been selected as suggested in the previous screens, the program is assembled, linked, loaded into the PCD and switched online.

- 13) TR 51 is displayed online. It is possible to view the actuation of inputs 1, 2 or 3.



With a mouse click it is also possible to call and view online all other STs and TRs edited in FUPLA, e.g. ST 12.

The logical state of output O 45 is always displayed online. However, the two values "BCD value" and "BCD value * 10" are only displayed briefly while the ST in the program is actually being processed. Afterwards the online window shows "*****". The mechanism for displaying these values continuously is explained later. (see item 13)



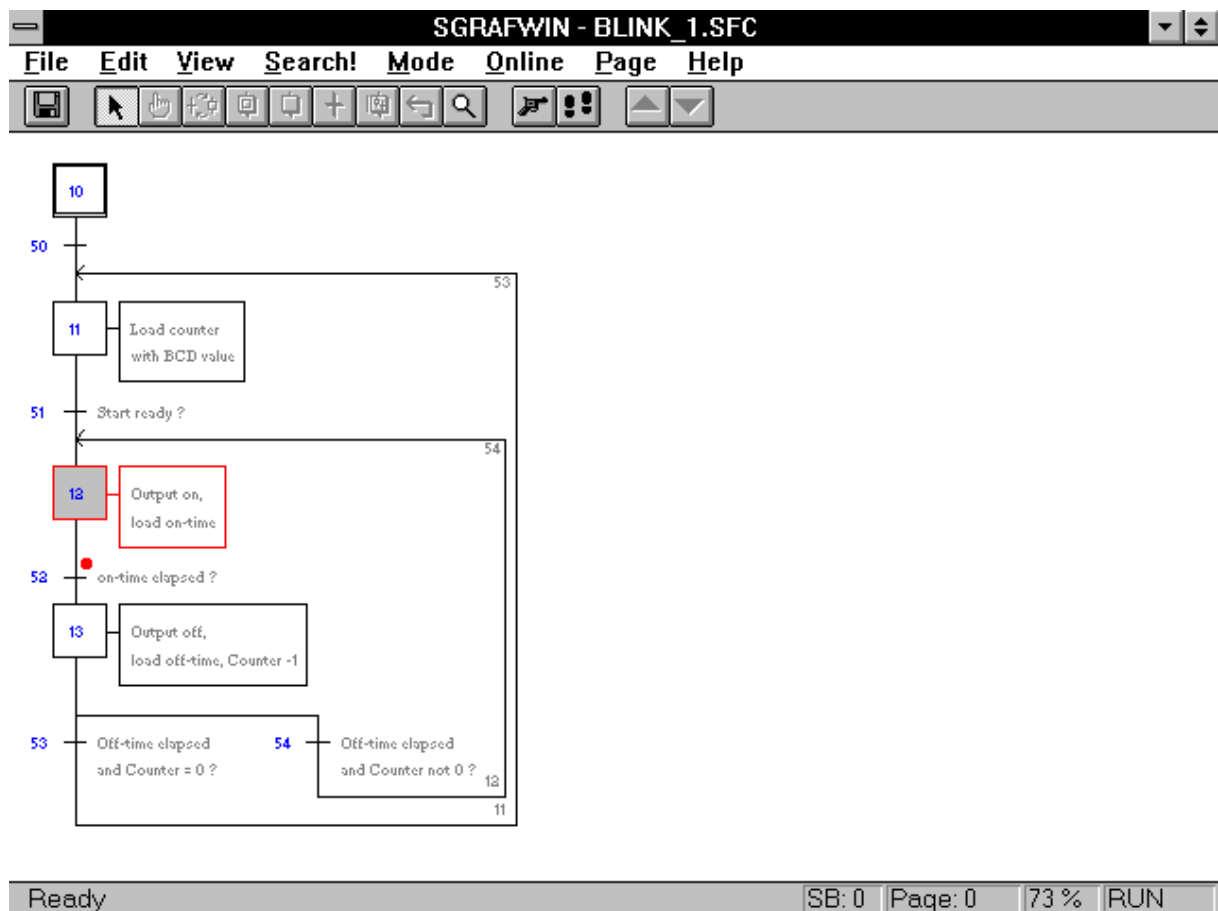
Return now to the GRAFTEC structure.

- 14) To view GRAFTEC running online, select

'Online' - Go Online'

from the GRAFTEC menu bar. If a FUPLA window is still open (on or offline) it is cleared by calling GRAFTEC online.

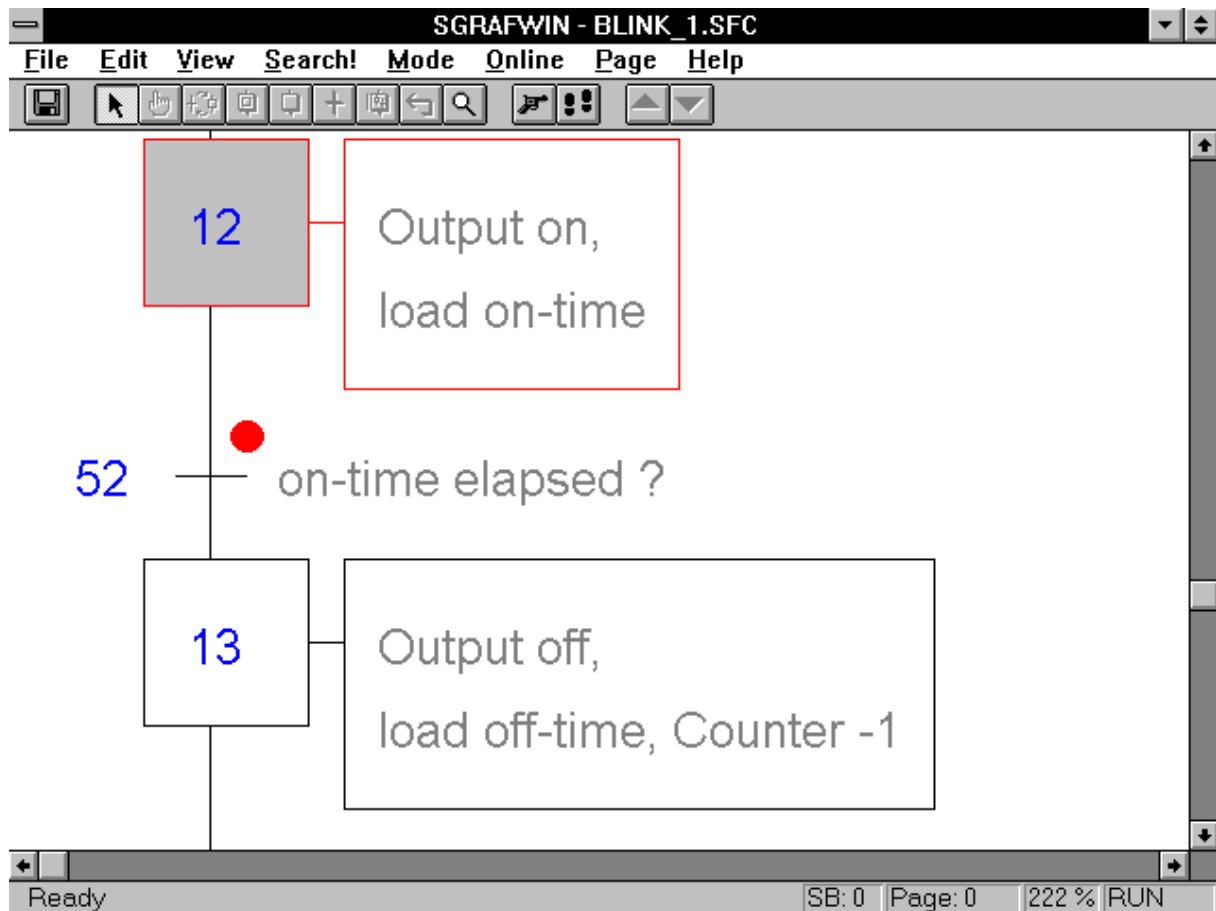
The screen presentation is roughly as follows:



If our blinker is switched on, i.e. if an input condition is met, it can be viewed running. The red spot shows the current position of the program.

If the start condition is not met, the red spot remains at TR 51 (Start condition OK?).

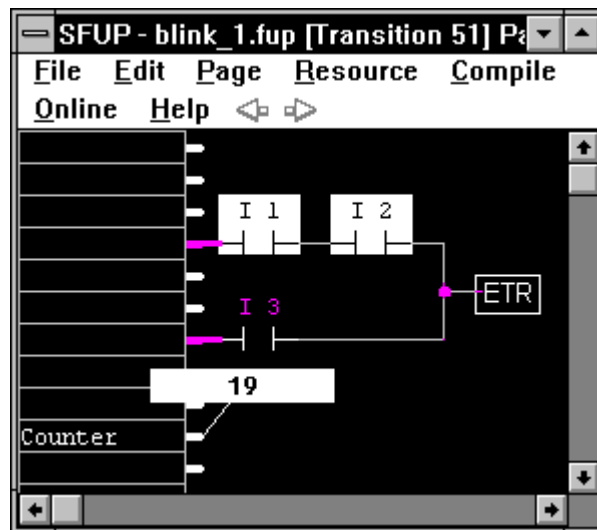
It is also possible to zoom and scroll the picture while GRAFTEC is switched online. Try it.



Try out the "Trace" and "Run" functions also.

Even when GRAFTEC is running online, it is possible to click open and view online the separate STs and TRs edited in FUPLA. The screen now shows 2 programs running online at the same time.

- 15) Item 13) touched on the problem of continuously refreshing the display of values. This is further clarified here with the addition of some practical tips.
- 15a) From TR 51, the "Counter" status is to be displayed online in an adjust window. For this purpose, the symbol name of the resource to be displayed must be typed into an input field. If the resource has been given an address it is also possible to specify the absolute address.



Switch FUPLA offline. After the entry, 'Make' should be executed again, since the FUPLA has been modified.

It is now possible to attach an online probe to this resource and to view its current value.

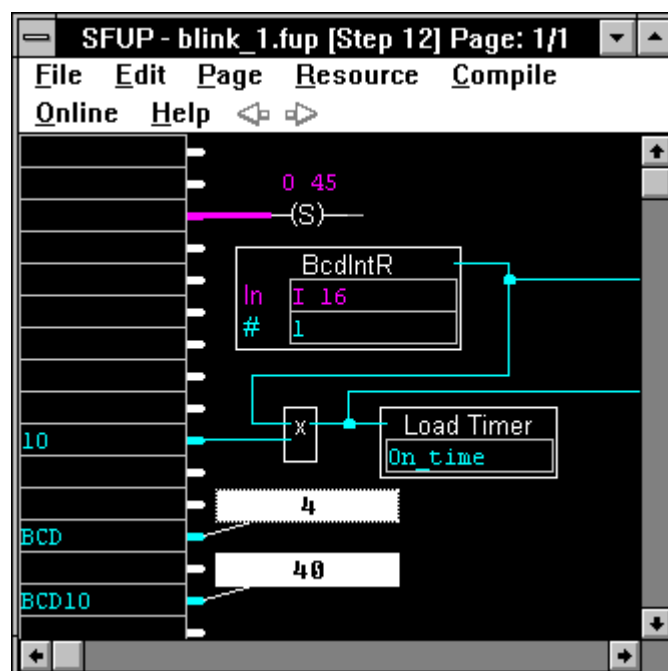
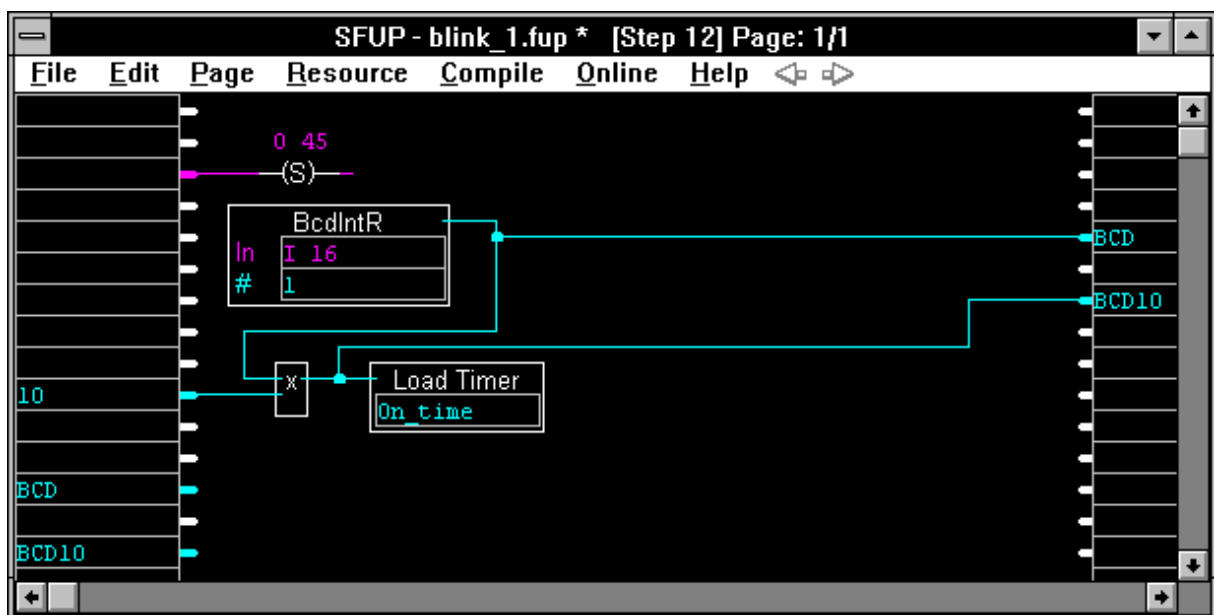
The FUPLA window shown for TR 51 demonstrates another characteristic of GRAFTEC programming with FUPLA. The connection lines, which in logic operations show their connection type by colour (red = binary) and status by line thickness (fine = L, heavy = H), only appear in colour and fine/heavy when the TR is active. If the program pointer (red spot) is located elsewhere in the GRAFTEC structure, the lines are neutral (fine and white).

- 15b) Return to ST 12. As a practical exercise, the BCD value and that value multiplied by 10 are to be displayed.

The trick used in item 13a) does not work here, as these are temporary results and not values in an addressed resource.

One solution would be to drag each of the values to an output field and to allocate a symbolic name or absolute address (for a register) to these fields.

In a similar way to 15b), these names or addresses can be accessed again at 2 input fields and displayed online.



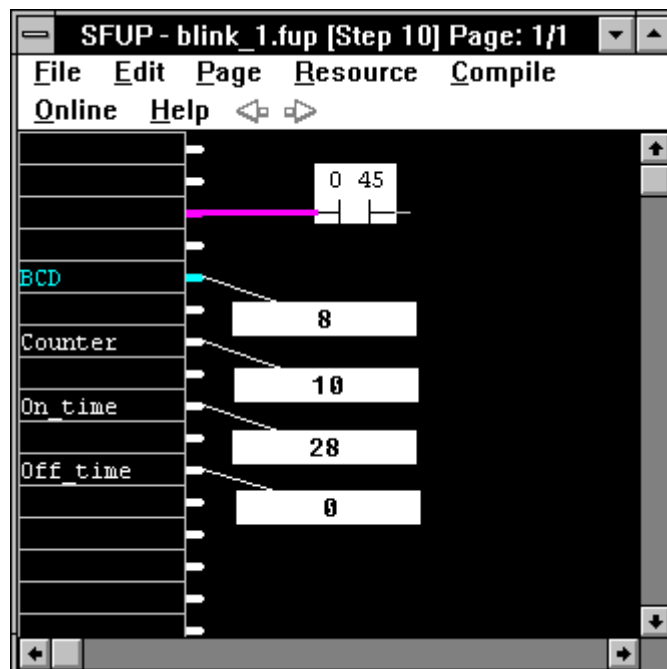
It is, of course, immediately obvious that these values are only refreshed when ST 2 is processed. Changing the entry while the program pointer is located elsewhere in the GRAFTEC structure does not change the display. The values displayed are therefore those available when ST 2 was last processed, instead of just "*****".

The values produced in a GRAFTEC program are certainly not all significant all the time. In order not to put an unnecessary load on the program, the storage of all values has been dispensed with. A practicable solution for the most important values displayed is demonstrated here.

- 15c) Another idea for displaying the values is to summarize the most important values for the whole program or for part of it.

Our example shows the status of output O 45, the BCD value set, the status of the cycle counter and the current status of the two timers "On-time" and "Off-time".

For this we can edit the empty IST 10.



The screenshot displays the SGRAFWIN software interface for a ladder logic program named 'BLINK_1.SFC'. The main window shows a ladder logic diagram with several steps:

- Step 10:** The starting step, highlighted with a red box.
- Step 11:** 'Load counter with BCD value'.
- Step 12:** 'Output on, load on-time'.
- Step 13:** 'Output off, load off-time, Counter -1'.
- Step 53:** 'Off-time elapsed and Counter = 0 ?'.
- Step 54:** 'Off-time elapsed and Counter not 0 ?'.

Transitions are labeled with numbers: 50, 51, 52, 53, 54, and 11.

An 'Online Toolbox' is visible at the bottom of the main window, containing icons for search, back, run, and stop.

Overlaid on the right is a smaller window titled 'SFUP - blink_1.fup [Step 10] Page: 1/1'. This window displays real-time data for the current step:

- BCD:** 0 45
- Counter:** 8
- On time:** 4
- Off time:** 63

The status bar at the bottom of the main window shows 'Ready', 'SB: 0', 'Page: 0', '73 %', and 'RUN'.

It is now possible both to display the important values online and to view the position of the program pointer in the GRAFTEC structure.

Notes :

5.4 Further examples

5.4.1 GRAFTEC program edited in FUPLA/KOPLA

This has already been described in detail in section 5.3.

5.4.2 GRAFTEC program edited in IL

The editing of a GRAFTEC structure is described in detail in section 5.3 and familiarity with it is assumed here. This section concerns the coding of an existing, commented GRAFTEC structure in IL (Instruction List).

Let us first look at the history of SAIA[®] GRAFTEC editing.

SAIA[®] GRAFTEC in the previous "PG3" programming package, for which all program code is in IL, has proved itself in many thousands of applications since the arrival of the PCD in 1986.

At first code was entered with any ASCII editor, e.g. IBM's PE. For experienced programmers this is still a very popular method today.

Later, SAIA's convenient "SEdit" editor was developed, with syntax control during code entry, automatic resource management and many other features. This editor was integrated into the PG3. SEDIT was initially available only in English, and later in German tool SEDIT is also a straight DOS editor.

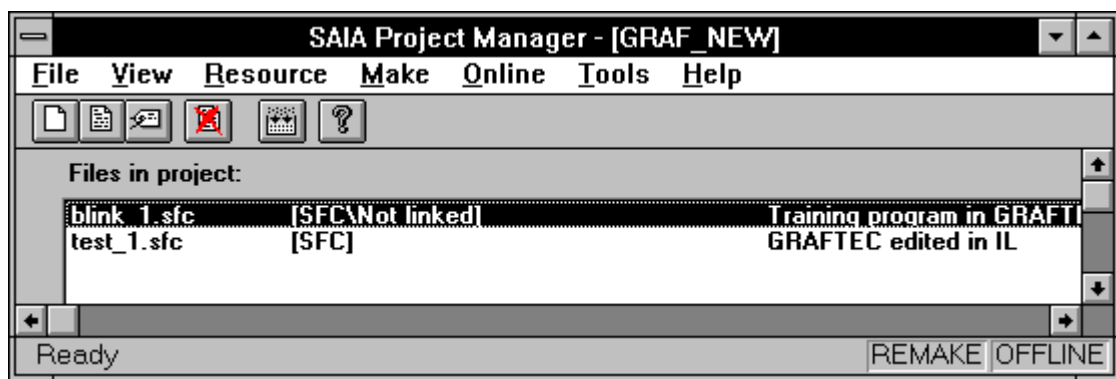
For the PG4 it was necessary to find an editor which would enable IL programming under WINDOWS. Using the "Notepad" (MS-WINDOWS standard editor) as its basis, SAIA developed "SEditWIN" and incorporated it into the PG4. The editor uses WINDOWS features - mouse control, cutting, copying and pasting via a clipboard - but as an IL editor for the PCD its capabilities are the same as, for example, IBM's PE, i.e. no syntax control and no resource management.

A simple, operational example is used to show GRAFTEC editing in IL.

Task: When input 5 is switched on, output 33 should switch on, switching off again when input 5 is switched off.

The program name is "TEST_1". It is to be stored in the existing project "GRAF_NEW".

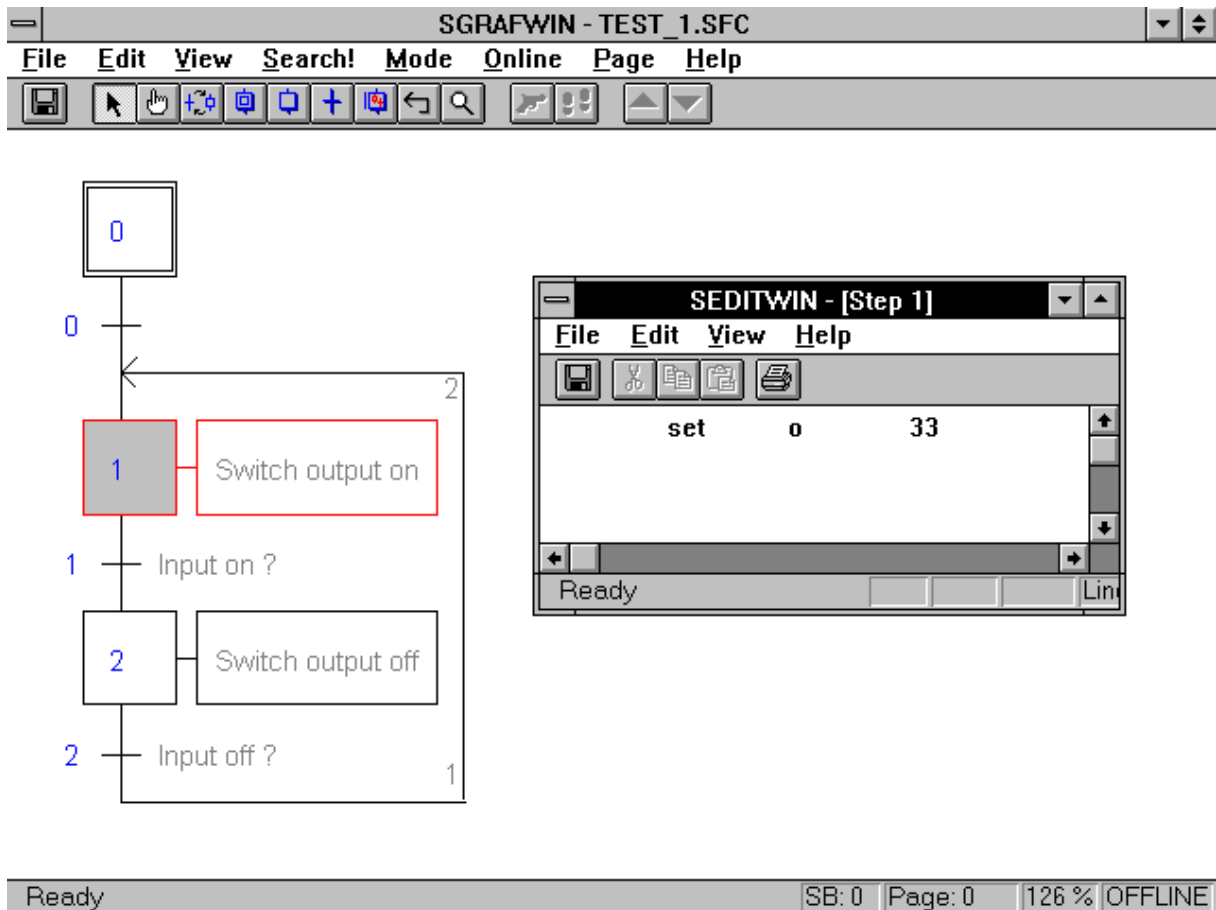
Go to the project library and select "graf_new". Enter the project manager and open our new file "test_1". At 'Language' select 'SFC (Graftec)'. Disable the existing file "blink_1.sfc" by selecting 'File'- 'Rename/Properties' and switching off "Assembled/Linked with project".



The GRAFTEC editor is called by double-clicking on "test_1.sfc". In the 'Edit' - 'Options...' menu, select "SEDTWIN" or "No Default" and confirm with 'OK'. Leave the (I)Step and transition offsets at 0.

The GRAFTEC structure can now be edited and commented as usual.

From any ST or TR in the GRAFTEC structure, it is now possible by double-clicking or single-clicking and 'Edit' - 'Code' to invoke SEDTWIN directly or indirectly. The window can be brought down to a manageable size in the usual way, so that the GRAFTEC structure remains visible next to the editor window. The code can now be entered according to the rules of IL programming.

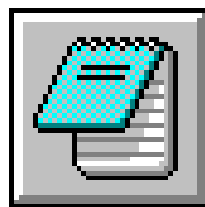


Before moving to another ST or TR, the previously entered code does not have to be saved. Simply click on the next resource to be coded. If coding is finished, it can be saved both in GRAFTEC and the IL editor with 'File' - 'Save' or by clicking on the memory toolbar symbol.

The file created is called

"TEST_1.SFC"

The entire IL file can be viewed (and edited) in the standard WINDOWS editor.



The file appears roughly as follows:

```

SB          0
;-----
IST          0
           0  0
EST          ;0
;-----
ST   1          ;Switch on output
      I   0
      I   2      ;Input off ?
      O   1      ;Input on ?
set   o   33
EST          ;1
;-----
ST   2          ;Switch off output
      I   1      ;Input on ?
      O   2      ;Input off ?
res   o   33
EST          ;2
;-----
TR   0
      I   0
      O   1      ;Switch on output
ETR          ;0
;-----
TR   1          ;Input on ?
      I   1      ;Switch on output
      O   2      ;Switch off output
sth   i   5
ETR          ;1
;-----
TR   2          ;Input off ?
      I   2      ;Switch off output
      O   1      ;Switch on output
stl   i   5
ETR          ;2

ESB          ;0

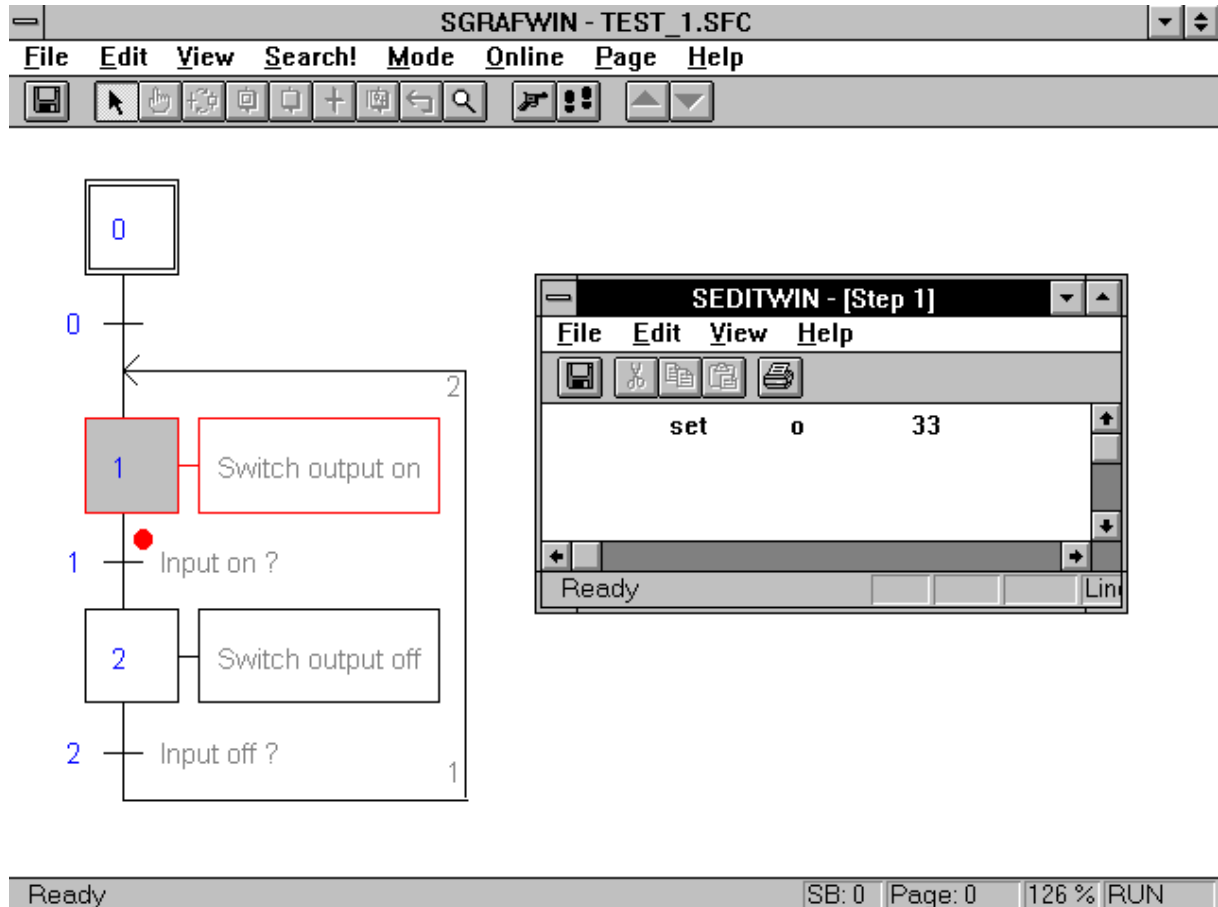
```

(The text in lower-case letters was entered as code inSEEDITWIN. The text in upper-case letters was created by the GRAFTEC editor).

To process the program further, i.e. to execute "Make", we return to the project manager. It is important that in 'Make' - 'Make Options...' the 'Auto Call SB' option is switched on, so that the resultant SB (sequential block) will be called and incorporated in the running program. Failure to do this produces a "No COB present" message during linking. 'Auto Call SB' opens COB 15. In this example, our SB 0 will be called from COB 15.

When everything has been correctly defined, 'Make' can be started. The program will be assembled, linked, downloaded to the PCD and switched into Run, according to our definitions.

The GRAFTEC editor can be selected again to view the process running online. The IL code of each ST and TR can also be displayed by double-clicking on the resource. In contrast to coding with FUPLA, in IL it is not possible to view the individual resources online.



By selecting 'Tools' - 'Debugger' from the project manager, the debugger can be called. The actual downloaded program can then be viewed with <Display> - <Programm> <0>.

5.4.3 GRAFTEC with multiple sequential blocks (SB)

If a user program consists of several SBs, it is necessary to ensure that each SB has a different SB number and that any ST or TR can only appear once in the same program. A total of 2000 ST s and 2000 TR s are available. The easiest way is to select a different offset for each SB, leaving enough free numbers between the individual SB s to avoid any conflicts with neighbouring SB numbers as that SB expands. Since the 3 files concerned are mutually independent, automatic numbering in each of them starts at zero if no offset has been specified. It is also possible to edit all the SB s without an offset, renumbering them later with the function 'Edit' - 'Renumber...'. SB numbers can also be adjusted later.

A small example is intended to demonstrate all this:

A project entitled

"MULTI_SB"

is to be programmed with 3 SBs called

"TEST_A", "TEST_B" and "TEST_C"

The SB numbers and offsets are defined as follows:

TEST_A:	in SB 1 with offset 100	programmed in FUPLA
TEST_B:	in SB 2 with offset 200	programmed in FUPLA/IL
TEST_C:	in SB 3 with offset 300	programmed in IL

The preceding example can be taken as the function, with a different input and output selected in each SB. It is important here that the real sequential process of the program should be made visible, to demonstrate the correct online function.

The 3 program modules are defined in the project manager.



The files are then programmed one after another. As mentioned earlier, the SB numbers and the ST/TR offset can be defined before or after.

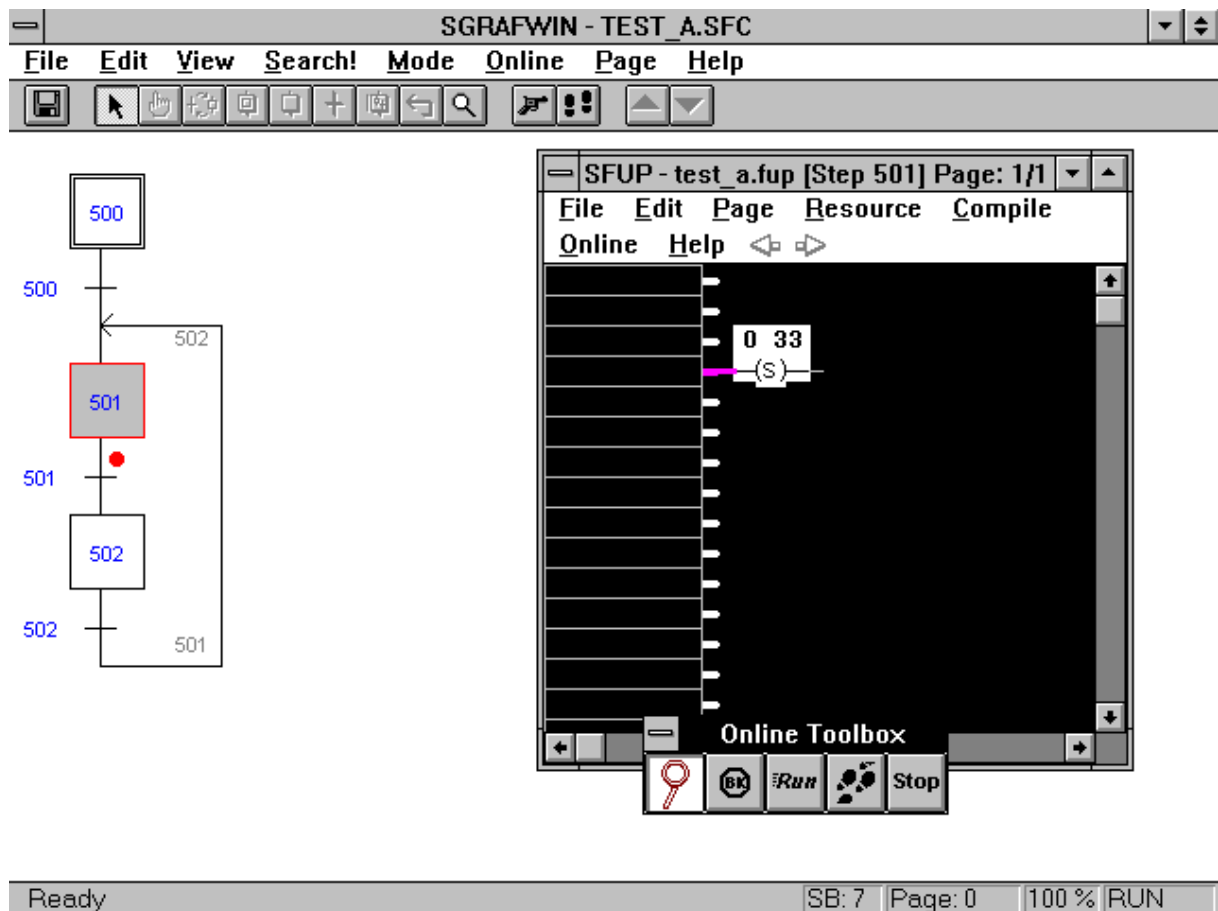
All files are to be saved. Files edited wholly or partially in FUPLA should be compiled. We then return to the project manager and execute 'Make', for which all 3 files must be active, i.e. in 'File' - 'Rename/Properties' "Assembled/Linked with project" must be switched on.

The 3 files are assembled to form the loadable file

"MULTI_SB.PCD"

linked and, according to the definitions, downloaded and switched into Run.

Correct function can be viewed from the PCD. One after another, files can be called from the project manager and their sequential process viewed online. With SB s edited wholly or partially in FUPLA, it is also possible to view online any ST or TR edited in FUPLA.

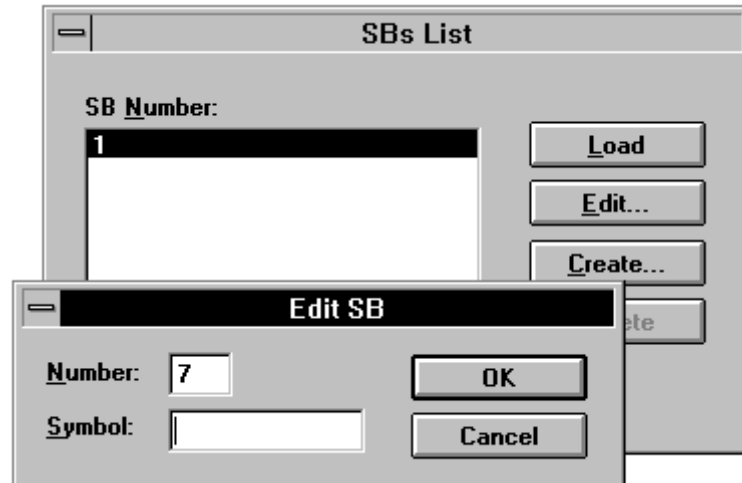


If, for example, the file "TEST_A" is to be renumbered:

SB 1 → SB 7

Offset 100 → Offset 500

we select this file, switch it offline, go to 'File' - 'SBs List...'



and use 'Edit...!' to change SB number 1 to <7>, confirming with 'OK'.

To renumber the ST/TRs, we go to 'File' - 'Renumber...'



and enter the new offset, in this case <500>, confirming with 'OK'.

Within this file, an ST or TR which was edited in FUPLA must now be called and recompiled. Afterwards, a new 'Make' can be executed directly from here or from the project manager.

Direct coding of a GRAFTEC file in IL

An efficient method of coding a GRAFTEC file is to save and exit the GRAFTEC structure edited in PG4 (or PG3), so that the whole file can then be loaded directly into the code editor. In this case it is possible to use either "SEDITWIN" from the PG4 or the PG3's popular SAIA editor "SEDIT" (if available).

If "SEDITWIN" is used, strict care should be taken not to change anything in the GRAFTEC structure (incomings and outgoings), otherwise it would no longer be possible to construct it in the GRAFTEC editor. If "SEDIT" is used, the structure is write protected and cannot be modified by the user, neither by accident nor intentionally.

Any other ASCII editor can, of course, also be used, such as PE or Q-Edit. The versatile copying capabilities of these editors can be utilized to the full and several files can be kept simultaneously open, constantly switching among them. This can save time and be very useful for copying the same, or similar, frequently used routines. The same technique can be used in "SEDITWIN", since this editor also allows the opening of several files and switching among them. Unfortunately, "SEDIT" does not allow this.

The PG4 manages only those resources which are programmed in FUPLA. The IL code is not managed in SEDITWIN. If the IL code is written in the PG3's SEDIT, the resources used here are managed separately in SEDIT.

5.4.4 Combination of GRAFTEC and IL programs

The technique is the same as in the previous section. One or more GRAFTEC files are edited in FUPLA, IL or a mixture and, if written wholly or partially in FUPLA, also compiled.

Further files edited in IL can be included in the project list. A 'Make' will assemble these files and linked with the GRAFTEC programs to form a loadable file. It is necessary to ensure that different COB numbers are allocated.

Example:

- Project: "GRAF_IL" with program modules
- TEST_A GRAFTEC file wholly edited in FUPLA
- TEST_B GRAFTEC file partially edited in FUPLA
- TEST_C GRAFTEC file edited in IL
- TEST_D straight IL file with COB 2
- TEST_E straight IL file with COB 3
- TEST_F file with PB and FB edited in IL
- TEST_G file with user texts

The programming steps are:

Define project in the project library, then open the individual program modules in the project manager:



For GRAFTEC files, the steps discussed in the preceding section apply:

- Adjust offset
- Adjust SB numbers
- Edit GRAFTEC structure
- Enter FUPLA or. IL code
- Compile files with FUPLA code

Then edit the other files in IL.

- Separate block numbers (COB, PB, FB) clearly
- Control text numbers

Go to the project manager and execute 'Make'. For this, all 7 files must be active, i.e. in 'File' - 'Rename/Properties' "Assembled/Linked with project" must be switched on.

The 7 files are assembled, linked to form the loadable file

"GRAF_IL.PCD"

and, according to the definitions, downloaded and switched into Run.

Correct function can be viewed from the PCD. One after another, files can be called from the project manager and their sequential process viewed online. With SB s edited wholly or partially in FUPLA, it is also possible to view online any ST or TR edited in FUPLA.

If "Pause after make" is enabled (x), it is possible to view correct assembly and linkage.

```

PG4 MAKE UTILITY
Assembling: SB029938.SRC
To:         SB029938.OBJ

Free memory: 379600

Assembly complete, 0 warnings, 0 errors

SAIA PCD LINKER $197
Linking: TEST_G.OBJ+TEST_F.OBJ+TEST_E.OBJ+TEST_D.OBJ+TEST_C.OBJ+TEST_B.OBJ+
TEST_A.OBJ+SFUPDBUG.OBJ+SB029938.OBJ
To:       GRAF_IL.PCD

Free memory: 332448
Code size: 219 lines (876 bytes)
Text size: 60 bytes
Exten mem size: 0 bytes (Init size: 0 bytes)
Global symbols: 5

Linkage complete, 0 warnings, 0 errors

<Press a key>

```

The arrangement of the various blocks can be viewed in the debugger:

XOB 16	Created automatically by FUPLA. In "TEST_E" an XOB 16 was also programmed for assigning the serial interface. The linker itself moves the content of this XOB 16 into the XOB 16 created by FUPLA.
SB 1	Program module TEST_A
SB 2	Program module TEST_B
SB 3	Program module TEST_C
COB 2	Program module TEST_D
COB 3	Program module TEST_E
PB 8/9	Program module TEST_F
FB 999	Created by FUPLA (for ONLINE). Called once from every block (ST, TR, PB) programmed in FUPLA.
COB 15	Created by PG4 with CSB 1, CSB 2, CSB 3.

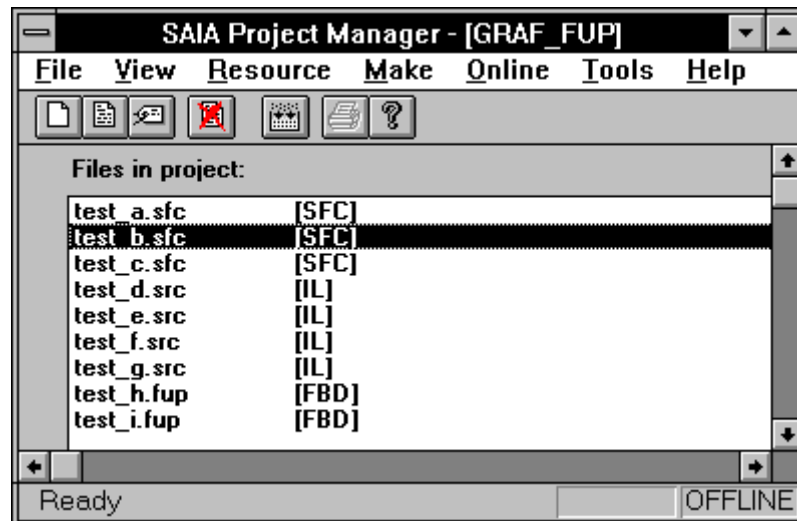
5.4.5 Combination of GRAFTEC, IL and FBD programs

A combination of all 3 types of program follows the same pattern as the preceding examples. To show this, let us take the previous program and combine it with two more, mutually independent FBD files.

The project in the earlier example, "GRAF_IL", is to be retained. The 'File' - 'Copy...' function from the project library is used to copy the existing "GRAF_IL" project and give it a new name:

"GRAF_FUP"

In this way the old program is retained while building on previously edited material.



In the project manager, the two new files "TEST_H" and "TEST_I" are added to the project list as "FBD" files and edited one after the other. The COB numbers of both files must be adjusted:

TEST_H	→	COB 5
TEST_I	→	COB 6

After saving and compiling the new FBD files, we return as usual to the project manager and execute 'Make', for which all 9 files must be active, i.e. in 'File' - 'Rename/Properties' "Assembled/Linked with project" must be switched on.

The 9 files are assembled, linked to form the loadable file

"GRAF_FUP.PCD"

and, according to the definitions, downloaded and switched into Run.

Correct function can be viewed from the PCD. One after another, files can be called from the project manager and the sequential process of GRAFTEC files viewed online. With SB s edited wholly or partially in FUPLA, it is also possible to view online any ST or TR edited in FUPLA. The newly added FBD files can also, of course, be called and viewed on-line.

Management of the dynamic resources for all files edited wholly or partially in FUPLA takes place in the project manager. With 'Resource' - 'Dynamic Distribution...' it is possible to view the distribution and number of dynamic resources used.

Resource Distribution [GRAF_FUP]

Timer/Counter Space

Last Timer:

After the last timer begins the first counter.

Dynamic Space

	Base address	Last address	Used	Free
Flags	7500	8191	71	621
Registers	3500	4095	45	551
Timers	5	31	4	23
Counters	1400	1599	0	200
Texts	3000	3499	0	500
Data Blocks	3500	3999	0	500

Buttons: Ok, Cancel, Set default

6. The IL editor (SEDITWIN)

With this editor, program modules can be written in instruction list form without exiting the PG4. The editor is similar to the standard WINDOWS editor. It utilizes the WINDOWS features: mouse control, cutting, copying, pasting via the clipboard. The editor does not yet have any syntax control or resource management, and there is no "online help" for PCD programming.

It is planned to include an IL editor, similar to the PG3's popular "SEDIT", in a later version of the PG4.

Since knowledge of WINDOWS is assumed for the use of the PG4, familiarity with the operation of "SEDITWIN" can also be assumed.

"SEDITWIN" has a detailed, interactive "Help" file in English. This help file will be translated into German at a later date.

The IL programming technique cannot be discussed here. The following manuals should be consulted:

- "User guide for the PCD family" 26/732
- "Instruction set for the PCD family" 26/733

The following pages provide a list of all SAIA® PCD instructions.

BIT Instructions

STH	STart High
STL	STart Low
ANH	ANd High
ANL	ANd Low
ORH	OR High
ORL	OR Low
XOR	eXclusive OR
ACC	ACCu operations
DYN	DYNamic (edge detection)
OUT	OUTput the accu status to an element
SET	SET element
RES	RESet element
COM	COMplement element
SETD	SET element Delayed
RESD	RESet element Delayed

WORD Instructions

LD	LoaD (32 bit value)
LDL	LoaD Low word (lower 16 bits)
LDH	LoaD High word (higher 16 bits)
DSP	load DiSPlay register
INC	INCRement register or counter
DEC	DECrement register or counter
SEI	SEt Index register
INI	INcrement Index register (+1)
DEI	DEcrement Index register (-1)
STI	STore Index register
RSI	ReStore Index register
MOV	MOVE data
COPY	COPY data
GET	GET data
PUT	PUT data
TFR	TransFeR data
BITI	single BIT In register, PCD format
BITIR	single BIT In register Reversed, PCA format
BITO	single BIT Out from register, PCD format
BITOR	single BIT Out from register Reversed, PCA format
DIGI	DIGIt in register, PCD format
DIGIR	DIGIt in register Reversed, PCA format
DIGO	DIGit Out from register, PCD format
DIGOR	DIGit Out from register Reversed, PCA format

AND	AND registers (32 bits)
OR	OR registers (32 bits)
EXOR	EXOR registers (32 bits)
NOT	complement register (32 bits)
SHIU	SHIfT registers Up
SHID	SHIfT registers Down
ROTU	ROTate registers Up
ROTD	ROTate registers Down
SHIL	SHIfT register contents Left
SHIR	SHIfT register contents Right
ROTL	ROTate register contents Left
ROTR	ROTate register contents Right

INTEGER arithmetic

ADD	ADD registers
SUB	SUBtract registers
MUL	MULTiply registers
DIV	DIVide registers
SQR	SQuare Root
CMP	CoMPare registers

FLOATING POINT arithmetic

IFP	Integer to Floating Point
FPI	Floating Point to Integer
FADD	Floating point ADDition
FSUB	Floating point SUBtraction
FMUL	Floating point MULtiplication
FDIV	Floating point DIVision
FSQR	Floating point SQure Root
FCMP	Floating point CoMPare
FSIN	Floating point SINE function
FCOS	Floating point COSine function
FATAN	Floating point Arc TANgent function
FEXP	Floating point EXPonential function
FLN	Floating point Natural Logarithm function
FABS	Floating point ABSolute value

BLOCTEC Instructions

COB	Cyclic Organsation Block
ECOB	End of COB
XOB	eXception Organisation Block
EXOB	End of XOB
PB	Program Block
EPB	End of Program Block
CPB	Call Program Block
CPBI	Call Program Block Indirect
FB	Function Block
EFB	End of Function Block
CFB	Call Function Block
NCOB	change to Next COB
SCOB	Stop COB
CCOB	Continue COB
RCOB	Restart COB

GRAFTEC Instructions

SB	Sequential Block
ESB	End Sequential Block
CSB	Call Sequential Block
RSB	Restart Sequential Block
IST	Initial STep
ST	STep
EST	End of STeps
TR	TRansition
ETR	End of TRansition

SERIAL COMMUNICATIONS Instructions

SASI	Serial communication ASIgn interface
SASII	Serial communication ASIgn interface Indirect
SRXD	Serial communication Receive Character (Mode C)
STXD	Serial communication Transmit Character (Mode C)
STXT	Serial communication Transmit Text (Mode C)
SRXM	Serial communication Receive Media
SRXMI	Serial communication Receive Media Indirect
STXM	Serial communication Transmit Media
STXMI	Serial communication Transmit Media Indirect
SICL	Serial communication Input Control Line
SOCL	Serial communication Output Control Line
SCON	Serial communication CONnect
SCONI	Serial communication CONnect Indirect

LAN2 Instructions

LRXD	Lan2 Receive Data
LTXD	Lan2 Transmit Data
LRXS	Lan2 Receive Status
LTXS	Lan2 Transmit Status

CONTROL Instructions

JR	Jump Relative
JRD	Jump Relative Direct
JPI	JumP Indirect
HALT	HALTs the cpu
LOCK	LOCK semaphore
UNLOCK	UNLOCK semaphore

DEFINITION Instructions

DEFVM	DEFine Volatile Memory (Flags)
DEFTC	DEFine Timer/Counter
DEFTB	DEFine Time Base
DEFTR	DEFine Timer Resolution
DEFWPR	DEFine Write Protected area in Run
DEFWPH	DEFine Write Protected area in Halt

SPECIAL Instructions

NOP	No OPeration
RTIME	Read TIME (hardware clock)
WTIME	Write TIME (hardware clock)
PID	PID control algorithm
TEST	TEST hardware
DIAG	XOB detail-DIAGnostic
SYSRD	SYStem ReaD
SYSWR	SYStem WRite
SYSCMP	SYStem CoMPare
ALGI	AnaLoGue Input (PCA modules)
ALGO	AnaLoGue Output (PCA modules)
STHS	STart High Slow
OUTS	set element from accu (OUT) Slow

Appendix A: Error and warning messages

Messages for the Projects Library

Error 1:	Cannot create project. The given name is already in use.
WHY:	A project already uses this name. Cannot create two projects with the same name.
EFFECT:	The Projects Library asks for a new name.
Error 2:	Cannot create project. The project name is invalid.
WHY:	The project name chosen contains characters which are not valid.
EFFECT:	The Projects Library asks for a new name.
Error 3:	Invalid project name.
WHY:	The project name chosen contains characters which are not valid.
EFFECT:	The Projects Library asks for a new name.
Error 4:	Cannot go to project directory '<i>project name</i>'.
WHY:	The Projects Library could not enter into the project directory. The project directory is probably invalid.
EFFECT:	Cannot open the project selected.
Error 5:	Cannot execute when PG4 editor active.
WHY:	Many commands which copy project's contents cannot execute when one of the PG4 editor is active.
EFFECT:	The command is aborted.
ACTION:	Close all PG4 editors running before using this command.
Error 6:	Cannot delete the file '<i>file name</i>'.
WHY:	A file within the project could not be deleted.
EFFECT:	The project cannot be completely be removed.
Error 7:	Cannot delete the directory '<i>directory name</i>'.
WHY:	The project directory could not be removed. Network problem ?
EFFECT:	The project directory will remain in the project list.
ACTION:	The user should try to remove it manually.

- Error 8: Project already exists. Cannot rename project.**
WHY: A project already uses this name. Cannot create two projects with the same name.
EFFECT: The operation is cancelled.
- Error 9: File already exists. Cannot rename project.**
WHY: A file in the project root directory already uses this name. Cannot create the directory project when a file used same name.
EFFECT: The operation is cancelled.
- Error 10: An error occurred during the project renaming. Operation cancelled.**
WHY: An error occurred when tried to rename the project.
EFFECT: The operation is cancelled.
- Error 11: Project already exists. Cannot copy project.**
WHY: A project already uses this name. Cannot create two projects with the same name.
EFFECT: The operation is cancelled.
- Error 12: File already exists. Cannot copy project.**
WHY: A file in the project root directory already uses this name. Cannot create the directory project when a file uses same name.
EFFECT: The operation is cancelled.
- Error 14: Cannot create directory '*directory name*'. Command not executed.**
WHY: Cannot create the directory for the new project.
EFFECT: Operation cancelled.
- Error 15: Could not create the project root directory '*project root directory*' !**
WHY: The *project root directory* could not be created. The operating system cancelled the operation.
EFFECT: The program aborts. This will only happen at start-up.
ACTION: Control that the variable [Global] ProjectsDir= in the file SPG4.INI contains a valid root directory for the projects.

- Error 16:** **The project root directory '*project root directory*' is not a directory !**
WHY: The *project root directory* is not a directory. The PG4 was not installed properly !
EFFECT: The program aborts. This will only happen at start-up.
ACTION: Control that the variable [Global] ProjectsDir= in the file SPG4.INI contains a valid root directory for the projects.
- Error 17:** **Cannot copy file '*file name*' to file '*file name*'. Operation cancelled.**
WHY: A file could not be copied.
EFFECT: The project was partially copied !
ACTION: WARNING: Make sure all the components which needed to be copied were copied.
- Error 18:** **Cannot go to project directory '*directory name*'.**
WHY: Cannot set the working directory in the project directory *directory name*.
EFFECT: The program aborts. This will only happen at start-up.
ACTION: Control that the variable [Global] ProjectsDir= in the file SPG4.INI contains a valid root directory for the projects.

WARNINGS

Warning 1: This command will delete all the files allocated for the project '*projectname*'.

Are you sure you want to delete the project '*project-name*' ?

WHY: This warning is there to instruct the user of the consequence of his action.

EFFECT: If the user's answer is yes the entire project will be removed from disk.

Messages for the Project Manager

Error 1:	You must re-make the project!
WHY:	Cannot go online, the project needs to be re-make.
Error 2:	Cannot open port!
WHY:	The communication port cannot be opened. Maybe an other application is using it. Maybe the port does not exist. The communication library gives a message explaining why this is.
EFFECT:	The application cannot go online.
ACTION:	From the communication library's message, the user should know what to do.
Error 3:	File does not match PCD contents!
WHY:	The content of the PCD memory is not the same as the file .PCD contained in the project directory.
EFFECT:	Going online will not debug the same program as the one viewed by the user.
ACTION:	The user should re-download the program.
Error 4:	Cannot go online!
WHY:	The project manager could not come online. Probably a bad connection configuration.
EFFECT:	Cannot go online.
ACTION:	Verify the communication setup and make sure the cable is set properly.
Error 5:	Cannot find file <i>file</i>
WHY:	The file <i>file</i> cannot be found.
EFFECT:	The operation cannot take place.
Error 6:	Cannot copy file <i>fileSrc</i> to <i>fileDst</i>.
WHY:	Cannot copy the <i>fileSrc</i> to <i>fileDst</i> , maybe out of disk space, maybe no write access on file <i>fileDst</i> , maybe no read access on <i>fileSrc</i> .
EFFECT:	Operation is canceled.
ACTION:	Make sure can read <i>fileSrc</i> , make sure can write to file <i>fileDst</i> , make sure the disk is not full.
Error 8:	Cannot rename block when PG4 editor active.
WHY:	A block cannot be renamed while on of the PG4 editor is active. This could result in confusion in right-access between the project manager and the different editors.
EFFECT:	Command cannot be executed.
ACTION:	Exit all PG4 editors before renaming a block.

- Error 9: Cannot execute when PG4 editor active.**
WHY: Many instructions are not available when one of the PG4 editors is active. There could occur conflicts in access rights.
EFFECT: Some commands are not available when one editor is active.
ACTION: Close all editors from the SAIA programming tool if you want to perform this command.
- Error 10: Project needs to be remake !**
WHY: There is a part of the project which is not build. the user program is not uptodate.
EFFECT: The user must re-make his application.
ACTION: The user should choose make before he downloads the program.
- Error 11: Cannot restart CPU *Number* !**
WHY: The command restart CPU *Number* was not accepted by the PCD. A bad communication configuration or the PCD has old FW version or FW crashed (which is most unlikely).
ACTION: The communication library should return information which will indicate to the user what he should do.
- Error 12: This is a reserved word! Use an other name.**
WHY: The block name used is not permitted.
EFFECT: Cannot use this name.
ACTION: Choose an other name.
- Error 13: Cannot download program *program_name* in CPU *number* !**
WHY: The program could not be downloaded in the PCD. The problem could be: bad connection, bad online configuration, bad PCD memory configuration.
EFFECT: The operation is canceled.
ACTION: From the communication library error message the user should take action.
- Error 14: Cannot run CPU *number* !**
WHY: The command Run was not accepted by the PCD. An invalid program was downloaded, a bad connection, or a bad memory configuration could be the cause of the problems.
ACTION: The communication library should return information which will indicate to the user what he should do.

- Error 15: Cannot stop CPU *number* !**
WHY: The command Stop was not accepted by the PCD. An invalid program was downloaded, a bad connection, or a bad memory configuration could be the caused of the problems.
ACTION: The communication library should return information which will indicate to the user what he should do.
- Error 16: Could not remove entry from project file: PROJECT.INF !**
WHY: The file PROJECT.INF cannot be updated because we cannot write to the project file PROJECT.INF.
EFFECT: Cannot modify the project, cannot remove project files from the Project file descriptor.
ACTION: Make sure that the disk is not full. Make sure that we have write access to the file PROJECT.INF.
- Error 17: Could not add entry in project file: PROJECT.INF !**
WHY: The file PROJECT.INF cannot be updated because, the disk is full, or we cannot write to the project file PROJECT.INF.
EFFECT: Cannot add new block to project.
ACTION: Make sure that the disk is not full. Make sure that we have write access to the file PROJECT.INF.
- Error 18: Disk Full !**
WHY: The disk is probably full.
EFFECT: The command cannot be executed.
ACTION: Free some disk space.
- Error 20: The range for dynamic Texts and dynamic Data Blocks must not overlap.**
WHY: The range chosen for dynamic Text and dynamic Data Blocks overlap.
EFFECT: This is forbidden.
ACTION: The user must select a range where they don't overlap.
- Error 22: Cannot create file PROJECT.DRA.**
WHY: The file PROJECT.DRA could not be created. Maybe it is in used, maybe it has no write access.
EFFECT: The make cannot take place.
ACTION: Exit SPROJMAN and remove manually the file project.dra.
- Error 23: Cannot create file PROJECT.MAK.**
WHY: The file PROJECT.MAK could not be created. Maybe it is in used, maybe it has no write access.
EFFECT: The make cannot take place.
ACTION: Exit SPROJMAN and remove manually the file project.mak.

- Error 24:** **File '*file*' does not exist.**
WHY: The file *file* does not exist.
EFFECT: The operation cannot be completed.
- Error 25:** **File *file* must be recompiled.**
WHY: The file *file* must be recompiled. It was created by the FUPLA editor SFUP.EXE. In order to build the application the FUPLA needs to compile the FUP file it created to generate a FBD file. The FBD file has a format which SASM and SLINK understand. SPROJMAN can only build the application with SFC, SRC, and FBD files.
EFFECT: Cannot build the application.
ACTION: Call FUPLA to edit *file*. Recompile the file *file*.
- Error 26:** **Cannot create file '*file*'. Operation aborted.**
WHY: A file *file* could not be opened. Because not enough disk space, or we did not have the access to write to file *file*.
EFFECT: The operation is canceled.
ACTION: Make sure enough disk space, and make sure we have the access rights for this file.
- Error 27:** **Project file list too long !!!**
WHY: There are too many files in the project.
EFFECT: Cannot add a project name to the project list. This error should in theory never occur.
- Error 28:** **There is an invalid file entry '*entry_string*' in the project file PROJECT.INF.**
WHY: There is an invalid entry in the file list.
EFFECT: The entry is ignored. Error should never occur unless user go manually modify the PROJECT.INF file !
ACTION: Don't edit the project.inf file.
- Error 29:** **Invalid project name !**
WHY: The Project Manager was started in an invalid directory.
EFFECT: The program cannot run in some invalid directories.
ACTION: This will never happens if you always use the Projects Library program to call the Project Manager.
- Error 30:** **This block is already in the list! Choose an other name.**
WHY: Two blocks cannot have the same name.
EFFECT: The command cannot be executed
ACTION: Use an other name.
- Error 31:** **Cannot import files from the current project directory!**
WHY: The import cannot import in the current directory, files which are already in the current directory.
EFFECT: Operation cannot be completed.
ACTION: Choose an other file.

- Error 32: Invalid file extension.**
WHY: The user tried to import a file with an invalid extension.
EFFECT: The operation is canceled.
- Error 33: Invalid file name.**
WHY: The name specified is not allowed.
EFFECT: The name is refused.
ACTION: Choose an other name. Hint many accents are not accepted.
- Error 34: Invalid file name. No extension must be supplied.**
WHY: You must not specify any extensions in the file name.
EFFECT: Invalid file name is refused.
ACTION: Choose an other name.
- Error 35: Invalid file name. No path must be specified.**
WHY: You must not specify any path in the file name.
EFFECT: Invalid file name is refused.
ACTION: Choose an other name.

WARNINGS

- Warning 1: Are you sure you want to delete the block '*block*' ?**
WHY: Deleting a block also deletes all the files associated with the block! The user needs to be warned.
EFFECT: User must decide. The files will be removed if the user answers Yes. The command will be canceled if the user answers No.
- Warning 2: File '*file*' already in project! Do you want to overwrite it?**
WHY: The file *file* is already in the project.
EFFECT: If you answer Yes, you will lose the content of the file already in the project. If you answer No the operation will be canceled.
- Warning 3: File '*file*' already in directory! Do you want to overwrite it?**
WHY: The file *file* is already in the project directory.
EFFECT: If you answer Yes, you will lose the content of the file already in the project directory. If you answer No the operation will be canceled.
- Warning 4: The build was cancelled !**
WHY: The user or SPROJMAN canceled the build.
- Warning 5: Do you want to go online anyway?**
WHY: This is the warning which gives the possibility to the user to go online even if the project is not up to date.
EFFECT: If users answer Yes, the project manager will go online, else it will not.
ACTION: The user should decide if he really can go online or not.
- Warning 6: Restarting CPU 0 will restart all CPUs.**
WHY: Restarting CPU 0 will also restart all the other CPU. This message will happen only on a PCD4/5/6.
ACTION: The user must confirm if he wants it or not.
- Warning 7: The modem is still active. Do you want to hangup ?**
WHY: When the user goes offline and the modem is active, the Project Manager asks if the user wants to hangup the modem.
EFFECT: The user can decide. If he does not want to hangup it will remain active, else it will be hangup.

Messages for the FUPLA

Error 2:	NOT IMPLEMENTED: <i>IDMXXX (nnn)</i>.
WHY:	Should never happen. This indicates that a menu command is not implemented.
Error 3:	AddToGrid: Can't add grid element xxx, attribute aaa, at location (x, y).
WHY:	Should never happen. This is an internal message indicating the corruption of the grid.
Error 4:	Short circuit of output connectors.
WHY:	The compiler indicates that there are two FBD box output connector which are linked together.
Error 5:	Cyclic connection not allowed.
WHY:	The compiler is indicating that the output of FBOX's network is fed back to the input.
Error 6:	Input and output labels cannot be directly connected.
WHY:	The user cannot directly connect the input connector and the output connector.
Error 7:	Kopla/Fupla invalid interconnection.
WHY:	The user has drawn a network where there is an output of a kopla element connected with the output of a fbox or a label.
Error 8:	Compilation errors detected.
WHY:	There was a compilation error detected.
EFFECT:	The compilation cannot be completed.
ACTION:	Correct the errors.
Error 9:	Invalid range.
WHY:	An invalid range was specified.
ACTION:	To specify range 1 to 5 and range 9 to 10, the user can use the following syntax: 1..5,9..10
Error 10:	Nothing to import !
WHY:	The file which was selected has nothing to be imported.
Error 14:	The macro 'End of Transition' must be used only once per transition !
WHY:	The fbox ETR was used more than once in the TR.
EFFECT:	Compilation stop.
ACTION:	Remove all unnecessary ETR FBox.
Error 15:	Cannot renumber a block while online !
WHY:	This action cannot be done when fupla is online.
ACTION:	First go offline.

Error 16:	Cannot renumber from step x to transition y.
WHY:	Step x cannot be found. Should never happens.
EFFECT:	Renumber cannot take place
Error 17:	Cannot renumber from step x to step y.
WHY:	Step x cannot be found. Should never happens.
EFFECT:	Renumber cannot take place
Error 18:	Cannot renumber from transition x to transition y.
WHY:	Transition x cannot be found. Should never happens.
EFFECT:	Renumber cannot take place
Error 19:	Cannot renumber from transition x to transition y.
WHY:	Transition x cannot be found. Should never happens.
EFFECT:	Renumber cannot take place
Error 20:	Cannot delete block while online !
WHY:	Block cannot be deleted when editor online.
ACTION:	Go offline.
Error 21:	Cannot delete block: <i>block</i>.
WHY:	Block <i>block</i> cannot be found. Should never happens.
EFFECT:	Action cannot take place
Error 22:	Too many elements, can't verify cyclic connections.
WHY:	The network is too big and the consistency test cannot be executed. This error should never occur.
Error 23:	CanGoto: Fatal error.
WHY:	Error in network scan algorithm. Should never occur.
Error 24:	Fatal Error in Module %s, line %d.
WHY:	A fatal error occurred in module xxx, at line ddd. This error should never occur.
ACTION:	Contact your SAIA representation.
Error 25:	Out of memory in Module %s, line %d.
WHY:	Not enough memory to execute action.
ACTION:	Free some memory by exiting other Windows application.
Error 26:	Can't show compilation errors. Out of memory.
WHY:	Not enough memory to execute action.
ACTION:	Free some memory by exiting other Windows application.
Error 27:	Not enough memory to read file.
WHY:	Not enough memory to execute action.
ACTION:	Free some memory by exiting other Windows application.

Error 28:	Cannot save document. Probably out of disk space !
WHY:	Not enough memory on disk to save the file.
EFFECT:	Fupla file cannot be saved.
ACTION:	Free disk space by removing unnecessary files.
Error 29:	Cannot save document. Error opening the temp file %Fs.
WHY:	A temp file which was used to store in memory the fupla file edited was corrupted. Probably the user destroyed one of them or he has a bad disk.
Error 30:	Constant type mismatched.
WHY:	The format used for a constant is not adequate for the type of network.
Error 31:	Not enough memory to create a new block !
WHY:	Not enough memory to execute action.
ACTION:	Free some memory by exiting other Windows application.
Error 32:	Invalid media for this field.
WHY:	The media type is not adequate for the type of network. E.g.: cannot use a flag in a floating point network.
Error 33:	There is an internal static variable defined by the user which is not defined properly !
WHY:	The user has specified an absolute address of a media for an internal adjust variable static variable which does not match the media type of the internal static variable specified in the macro definition.
EFFECT:	Cannot compile.
ACTION:	Change the media of the internal fbox static variable.
Error 34:	FBox: %Fs, empty definition.
WHY:	The fbox definition has no width defined, or has no entry defined. This should never occur unless someone modified the fbox definition file.
EFFECT:	The fbox cannot be used.
ACTION:	Do not use this fbox.
Error 35:	FBox: %Fs, invalid line definition. '%s'.
WHY:	The designer of the FBOX has included an invalid line definition in the fbox. This should never occur unless the fbox definition was modified.
ACTION:	Contact your FBOX supplier.
Error 36:	Too many static variables declared in FBox: %s.
WHY:	The designer of the FBOX has included too many static variables. This should never occur unless the fbox definition was modified.
ACTION:	Contact your FBOX supplier.

- Error 37:** **Too many dynamic variables declared in FBox: %s.**
WHY: The designer of the FBOX has included too many dynamic variables. This should never occur unless the fbox definition was modified.
ACTION: Contact your FBOX supplier.
- Error 39:** **Can't put in grid the function box: '%Fs'.**
WHY: The fup file being read is probably corrupted.
- Error 41:** **Can't put line between (%d, %d) - (%d, %d).**
WHY: The fup file being read is probably corrupted.
- Error 42:** **FUP file: '%Fs', invalid.**
WHY: The fup file being read is probably corrupted.
- Error 43:** **Can't find file '%Fs'.**
WHY: The file specified does not exist.
ACTION: Verify the name entered or the path given.
- Error 44:** **Can't create file '%s'.**
WHY: The disk is full, the name is invalid, or the directory or file specified is right protected.
ACTION: Make sure the directory and the file where you want to right is not right protected.
- Error 45:** **MSDOS system error.**
WHY: An error was sent by the lower level of the disk operating system.
- Error 46:** **Disk full.**
WHY: The hard disk where the information saved is full.
EFFECT: The action cannot be completed.
ACTION: Remove unnecessary files from your hard disk.
- Error 47:** **Cannot read from file *File*.**
WHY: The file cannot be read. It is probably corrupted. Or the file does not exist.
- Error 48:** **Invalid file name.**
WHY: The file name specified is not valid. Fup file must end with the extension .fup.
- Error 49:** **Cannot write to file *File*.**
WHY: The file *File* is maybe set in write protected mode.
ACTION: Try to change the setting of the file *File*.
- Error 50:** **Invalid network.**
WHY: The FUP file contains an invalid network definition. This should never happen, unless the FUP file is corrupted.
EFFECT: The FUP file is invalid and cannot be used.

Error 51:	No memory to allocate macro name.
WHY:	Not enough memory to execute action.
ACTION:	Free some memory by exiting other Windows application.
Error 52:	No memory to allocate the macro family name.
WHY:	Not enough memory to execute action.
ACTION:	Free some memory by exiting other Windows application.
Error 53:	Incomplete network.
WHY:	An input connector of a fbox or an not empty output label receives no input information.
ACTION:	Connect something to it or remove it.
Error 54:	Too many connectors in Page.
WHY:	There are too many connectors in a page and the compilation cannot take place.
ACTION:	Remove some fbox from the page and put them on the next page.
Error 55:	Loop back detected.
WHY:	There is a network which has its output fed back to its input.
EFFECT:	The compilation cannot take place.
ACTION:	Remove the loop back.
Error 56:	Not enough memory to compile.
WHY:	Not enough memory to execute action.
ACTION:	Free some memory by exiting other Windows application.
Error 57:	Not enough memory to show priorities.
WHY:	Not enough memory to execute action.
ACTION:	Free some memory by exiting other Windows application.
Error 58:	Too many pages.
WHY:	There are too many pages specified in this fup file.
ACTION:	Start an other module file.
Error 59:	Bad type.
WHY:	The label indicated has no media defined.
EFFECT:	The compilation cannot take place.
ACTION:	Define the label media.
Error 60:	Bad label.
WHY:	The label indicated is not defined.
EFFECT:	The compilation cannot take place.
ACTION:	Define the label.

- Error 61: The macro ETR is allowed only in a transition.**
WHY: The macro ETR was used in a block which is not a transition.
EFFECT: The compilation cannot take place.
ACTION: Remove the fbox ETR.
- Error 62: Can't compile. Block was not typed.**
WHY: The block was not typed.
EFFECT: The compilation cannot take place.
ACTION: Define a block type.
- Error 63: The End Of Transition (ETR) macro must be used in Transition %d.**
WHY: The fbox ETR was not used in the transition.
EFFECT: The compilation cannot take place.
ACTION: Place the ETR fbox.
- Error 64: Can't compile. Macro must be named.**
WHY: A block typed macro was defined but not named.
EFFECT: The compilation cannot take place.
ACTION: Name the macro.
- Error 65: FBox not in library.**
WHY: The fbox family is not in the library. Maybe the library is not well installed.
EFFECT: The compilation cannot take place, else there would be an error at the make time.
ACTION: Do not use the fbox unless you have its library files.
- Error 66: You must recompile %s.**
WHY: The current FUP file was not compiled since its last modification.
EFFECT: Cannot go online.
ACTION: Re-compile the FUP file.
- Error 67: Supervision time too large.**
WHY: The supervision time out specified by the user was too big.
- Error 68: Block number too large. Maximum is %d.**
WHY: The block number selected is too large.
ACTION: The maximum which can be used is indicated in the error message.
- Error 69: Invalid block number.**
WHY: The block number specified is invalid.
ACTION: Choose a valid block number.
- Error 70: No block type defined.**
WHY: The block type was not defined.
ACTION: Select a block type.

Error 71:	Cannot select block '%s'.
WHY:	An application is asking fupla to select a block which is not present in its current file.
Error 72:	Cannot create new block '%s'.
WHY:	Maybe not enough memory to create a new block.
ACTION:	Leave a unused Windows applications.
Error 73:	Invalid page range.
WHY:	The page range specified is not valid.
ACTION:	Make sure the page exist. Make sure you are using the proper page range syntax. E.g: 2..7, 15, 18 specify the following pages 2 3 4 5 6 7 15 and 18.
Error 74:	Can't open file '%Fs'.
WHY:	The file probably does not exist.
ACTION:	Choose an other file, or look in an other directory.
Error 75:	Field copies out of range !
WHY:	Too many copies were asked.
EFFECT:	The print cannot take place.
ACTION:	Lower the number of copies.
Error 76:	Cannot print page!
WHY:	An error occurred during printing.
EFFECT:	The print cannot take place.
Error 77:	No printer driver installed !
WHY:	There is no printer driver installed.
ACTION:	Install a printer driver. See Windows documentation
Error 78:	Can't create virtual cursor.
WHY:	Not enough memory to build a cursor.
EFFECT:	The application cannot continue.
Error 79:	Cannot execute. Probably out of memory.
WHY:	Not enough memory to execute action.
ACTION:	Free some memory by exiting other Windows application.
Error 80:	Invalid file extension.
WHY:	The file extension you have selected is not valid.
Error 81:	Can not run CPU #%%d.
WHY:	The CPU does not respond.
ACTION:	Check cable. Check Online Connection... configuration.
Error 82:	Can not stop CPU #%%d.
WHY:	The CPU does not respond.
ACTION:	Check cable. Check Online Connection... configuration.

- Error 83:** **Can not restart CPU #%d.**
WHY: The CPU does not respond.
ACTION: Check cable. Check Online Connection... configuration.
- Error 84:** **Can not run conditionally CPU #%d.**
WHY: The CPU does not respond.
ACTION: Check cable. Check Online Connection... configuration.
- Error 85:** **Can not get instruction pointer from CPU %d.**
WHY: The CPU does not respond.
ACTION: Check cable. Check Online Connection... configuration.
- Error 86:** **Can't find break point address.**
WHY: Cannot find the label which will allow the online utilities to set a breakpoint.
EFFECT: Cannot set break point.
ACTION: Re-make.
- Error 87:** **The debugger is out of timer. It will not behave properly.**
WHY: There are no more timers available in the Windows system.
EFFECT: The online will not behave properly.
ACTION: Restart Windows.
- Error 88:** **No DBSTRUCT found in pcd file. Cannot debug.**
WHY: The DBSTRUCT cannot be found in the PCD file. Probably not recompiled with the project manager.
EFFECT: Cannot get the debug information.
ACTION: Re-make the user application.
- Error 89:** **Debug structure __DBSTRC__ invalid.**
WHY: The DBSTRUCT is invalid. Probably a wrong version of the SAIASYM.DLL driver or a wrong version of the assembler.
EFFECT: Cannot get the debug information.
ACTION: Re-make the user application. Make sure you have the good version of software components.
- Error 90:** **This command requires SAIA Project manager to be active !**
WHY: The project manager is not active. The project manager builds the project.
EFFECT: The make cannot take place.
ACTION: Start the project manager.

Error 91:	The project manager is not present. The project is maybe out of date ! Do you want to continue ?
WHY:	The project manager is the one which inspects if the user program is up to date.
EFFECT:	The project manager will not verify if the project is up to date if the user answers yes.
ACTION:	Answer yes if you want to continue. Else answer no.
Error 92:	Cannot find all debug info.
WHY:	Some debugging information cannot be found in the PCD file to go online.
EFFECT:	The refresh might not operate properly.
ACTION:	Re-make the project.
Error 93:	Invalid value.
WHY:	An invalid value was entered in an adjust window field.
ACTION:	Enter a valid value or cancel the modifications done to the adjust window.
Error 94:	Out of range value. The Range is: %s.
WHY:	An value out of the allowed range was entered in an adjust window field.
ACTION:	Enter a value within the allowed range or cancel the modifications done to the adjust window.
Error 95:	Bad beginning of page record.
WHY:	A beginning of page record is invalid. The file is most probably corrupted.
EFFECT:	Cannot read the file.
Error 96:	Bad FBOX read.
WHY:	A FBOX record is invalid. The file is most probably corrupted.
EFFECT:	Cannot read the file.
Error 97:	Bad input/output label read.
WHY:	An input/output label record is invalid. The file is most probably corrupted.
EFFECT:	Cannot read the file.
Error 98:	Bad network read.
WHY:	A network record is invalid. The file is most probably corrupted.
EFFECT:	Cannot read the file.
Error 99:	There is not enough memory to copy FBOX definitions.
WHY:	Not enough memory to execute action.
ACTION:	Free some memory by exiting other Windows application.

- Error 100: Too many FBox per family.**
WHY: There were too many fbox defined in a fbox family. This should never occur unless the fbox definition was modified or you have receive an invalid FBOX family.
ACTION: Contact your FBOX supplier.
- Error 101: Not enough memory to allocate macro descriptor.**
WHY: Not enough memory to execute action.
ACTION: Free some memory by exiting other Windows application.
- Error 102: You have placed too many fbox in the page. Maximum is %d.**
WHY: There are too many fbox placed in a page.
EFFECT: The editor forbid that.
ACTION: Put the extra fboxes on an other page.
- Error 104: Macro name too long: '%s'. Will be truncated.**
WHY: The name of a macro is too long. This should never occur unless the fbox definition was modified or you have receive an invalid FBOX.
ACTION: Contact your FBOX supplier.
- Error 105: Macro: '%Fs', bad adjust variable: '%s'.**
WHY: The adjust variable defined is invalid. This should never occur unless the fbox definition was modified or you have receive an invalid FBOX.
ACTION: Contact your FBOX supplier.
- Error 106: Macro: '%Fs', no memory to store adjust variables.**
WHY: Not enough memory to execute action.
ACTION: Free some memory by exiting other Windows application.
- Error 107: Too many lines to read.**
WHY: A line record was to big in the file read. The file is most probably corrupted.
EFFECT: Cannot read the file.
- Error 108: Read line. Invalid record.**
WHY: A line record was invalid. The file is most probably corrupted.
EFFECT: Cannot read the file.
- Error 109: Too many labels to record.**
WHY: There are too many labels in the current page to record.
EFFECT: The page cannot be saved.
ACTION: Remove some fbox.

- Error 110: Cannot read label record. Too big.**
 WHY: A I/O label record is to big in the file read. The file is most probably corrupted.
 EFFECT: Cannot read the file.
- Error 111: Invalid variable '%s', in '%s' section.**
 WHY: An input/output label entry is invalid. The file is most probably corrupted.
 EFFECT: Cannot read the file.
- Error 112: Macro: '%Fs', entry Face= too long.**
 WHY: The Face name given to the fbox is too long. This should never occur unless the fbox definition was modified or you have received an invalid FBOX.
 ACTION: Contact your FBOX supplier.
- Error 113: Entry too big. Cannot delete symbol in this entry.**
 WHY: An entry was to big in the file read. The file is most probably corrupted.
 EFFECT: Cannot read the file.
- Error 114: Not enough memory to read FBox definition.**
 WHY: Not enough memory to execute action.
 ACTION: Free some memory by exiting other Windows application.
- Error 115: FBOX line too long !**
 WHY: There is a fbox which has a line too long. This should never occur unless the fbox definition was modified.
 ACTION: Contact your FBOX supplier.
- Error 116: Too many labels in page ! Maximum is %d.**
 WHY: There are too many labels in a page. This will happens if there are too many Kopla elements in a page. Each kopla element has a label on them.
 EFFECT: If there are too many labels in a page the compilation cannot take place.
 ACTION: Remove some fbox which have a label and put them on an other page.
- Error 118: Not enough memory to load page function box.**
 WHY: Not enough memory to execute action.
 ACTION: Free some memory by exiting other Windows application.
- Error 119: The macro '%s' was defined more than once. Last macro definition will be discarded.**
 WHY: The macro specified was defined more than once. This should never occur unless the fbox library was modified.
 EFFECT: The last macro definition will be discarded.
 ACTION: Contact your FBOX supplier.

- Error 121: Cannot compile bad function box coordinates.
FBox: %Fs.**
WHY: The temp files used to store the FUP file in memory is corrupted, and stops the compiler from working properly.
EFFECT: The compilation cannot take place.
ACTION: Leave FUPLA, and re-edit your file.
- Error 122: Cannot compile. Too many family used.**
WHY: There were too many family used in the same module.
EFFECT: The compilation cannot take place.
ACTION: Use less family.
- Error 123: Cannot compile. Bad network read in variable: '%s'.**
WHY: The temp files associated to the file being edited was probably corrupted.
EFFECT: The compilation cannot take place.
ACTION: Quit fupla. Re-edit the last file. Re-compile.
- Error 125: Cannot compile!**
WHY: The compilation cannot take place for previously displayed reasons.
- Error 126: Invalid file '%Fs'.**
WHY: The file read is invalid.
ACTION: Make sure it is not corrupted.
- Error 127: Cannot find informations concerning FBOX #%%ld.**
WHY: The file read has an old format, and fupla cannot find all information concerning the specified fbox.
EFFECT: The file cannot be read.
- Error 128: Bad horizontal line read during conversion.**
WHY: The file read has an old format, and seems to be corrupted.
- Error 129: Bad vertical line read during conversion.**
WHY: The file read has an old format, and seems to be corrupted.
- Error 130: Macro: '%Fs', bad adjust variable: '%s'. Default value out of range.**
WHY: The adjust variable defined has a default value which is out of range. This should never occur unless the fbox definition was modified or you have received an invalid FBOX.
ACTION: Contact your FBOX supplier.

- Error 131: Macro: '%Fs', bad adjust variable: '%s'. Set value out of range.**
WHY: The adjust variable defined has a set value which is out of range. This should never occur unless the fbox definition was modified or you have received an invalid FBOX.
ACTION: Contact your FBOX supplier.
- Error 132: Macro: '%Fs', bad adjust variable: '%s'. No range defined.**
WHY: The adjust variable defined in the has no range defined. This should never occur unless the fbox definition was modified.
ACTION: Contact your FBOX supplier.
- Error 133: Macro: '%Fs', bad adjust variable: '%s'. Button must be an online variable.**
WHY: The adjust variable defined in the macro is assigning a button to an offline variable. This should never occur unless the fbox definition was modified.
ACTION: Contact your FBOX supplier.
- Error 134: Not enough memory to select this item.**
WHY: Not enough memory to execute action.
ACTION: Free some memory by exiting other Windows application.
- Error 136: Cannot create backup file '%s'.**
WHY: Maybe the backup file has read only attribute. Maybe the disk is corrupted.
ACTION: Leave FUPLA, remove the file specified in the message, and return to FUPLA.
- Error 137: File '%Fs' too long. Maximum 64K.**
WHY: This file is too long and cannot be entirely read. This is relevant to fbox definition files.
ACTION: The file should segmented in more files.
- Error 138: Macro: '%Fs', bad adjust variable: '%s'. Too many ranges defined.**
WHY: The adjust variable defined in the macro has too many ranges defined. This should never occur unless the fbox definition was modified.
ACTION: Contact your FBOX supplier.
- Error 139: Macro: '%Fs', variable: '%s' too long. It will be truncated.**
WHY: There was a variable definition in the Macro which was too long.
EFFECT: The variable will be truncated.
ACTION: Use smaller line definition.

- Error 140:** **File '%Fs', format (%d) too old.**
WHY: This file was created with an older version of FUPLA which cannot be converted to the present version.
ACTION: An older version of FUPLA maybe could convert this file too a newer version.
- Error 141:** **File: '*file*' was created with FUPLA version x.**
WHY: The file was create with a newer version of fupla which is not compatible with the current version of fupla used.
EFFECT: The file cannot be read.
ACTION: Use the newer version of fupla to read this newer file format.
- Error 142:** **No FBox at gPoint:(x, y).**
WHY: This error should never occur.
- Error 143:** **(Cannot find block properties!)**
WHY: This should never occur. A block lost his properties.
- Error 144:** **A name must be specified for a macro.**
WHY: When the user creates a macro block he must specify a name.
- Error 145:** **Compilation cancelled by user !**
WHY: The user cancelled the compilation with the cancel button.
- Error 146:** **Cannot open the specified label window !**
WHY: There is probably not enough memory to open the specified Window.
EFFECT: Cannot edit the label.
ACTION: Free some memory by closing unused applications, or by re-booting Windows.
- Error 147:** **Cannot open file %s.**
WHY: The file specified cannot be opened.
ACTION: Probably the file does not exist.
- Error 148:** **Invalid PCD file: %s.**
WHY: The PCD file is corrupted.
EFFECT: Cannot go online.
ACTION: Re-make your application.
- Error 149:** **Cannot read PCD file: %s.**
WHY: The PCD file is corrupted or does not exist.
EFFECT: Cannot go online.
ACTION: Re-make your application.

- Error 150:** **No symbol information in the PCD file: %s.**
WHY: This PCD file was not build with symbolic information.
EFFECT: Cannot read its symbolic information.
ACTION: Re-make your application using the SPROJMAN application.
- Error 151:** **Bad symbol table header checksum in PCD file: %s.**
WHY: The PCD file is corrupted.
EFFECT: Cannot go online.
ACTION: Re-make your application.
- Error 152:** **Bad module list record checksum in PCD file: %s.**
WHY: The PCD file is corrupted.
EFFECT: Cannot go online.
ACTION: Re-make your application.
- Error 153:** **Bad local symbol table checksum in PCD file: %s.**
WHY: The PCD file is corrupted.
EFFECT: Cannot go online.
ACTION: Re-make your application.
- Error 154:** **The file *FILE* was not linked in the program.**
WHY: The module we want to debug was not linked in the PCD file.
EFFECT: Cannot debug the current module.
- Error 155:** **Out of memory !**
WHY: Not enough memory to execute action.
ACTION: Free some memory by exiting other Windows application.
- Error 156:** **Unknown SAIASYM error !**
WHY: The module SAIASYM.DLL has returned an unknown error status. This error should never occur.
- Error 157:** **Cannot import file !**
WHY: Could not read the imported file or could not create the temporary files to import the specified file. Maybe out of disk space.
ACTION: Make sure there is enough disk space.

WARNINGS

- Warning 1: The Graftec FBOX family is not present !**
WHY: The graftec FBOX family is not present. It should always be there unless your package is not well installed.
EFFECT: You will not be able to create Graftec logic.
ACTION: Make sure your PG4 is well installed.
- Warning 2: The Kopla FBOX family is not present !**
WHY: The KOPLA FBOX family is not present. It should always be there unless your package is not well installed.
EFFECT: You will not be able to create KOPLA logic.
ACTION: Make sure your PG4 is well installed.
- Warning 4: Function box '%Fs', is not available.**
WHY: The user tried to select a FBOX which is not available in the library. Maybe the FBOX library is not well installed.
EFFECT: Cannot select the specified FBOX.
ACTION: Reinstall the fbox library.
- Warning 5: Family: %s already in use.**
WHY: There are two fbox definition file which are linked to the same family.
EFFECT: The second file will be discarded.
- Warning 6: No FUPLA Box defined.**
WHY: The fbox library is empty. Maybe the PG4 package is not well installed.
EFFECT: You will not be able to add new fbox in the fupla file.
- Warning 7: The variable (%Fs) description will be truncated.**
WHY: This should never occur.
- Warning 8: The variable 'SFUPLibDir' in the SPG4.INI file (the directory for FUPLA's macro libraries) is undefined.**
WHY: The variable SFUPLibDir is not defined. Probably the PG4 package is not installed properly, or the file SPG4.INI could not be found.
EFFECT: The FBOX library will be empty.
ACTION: Re-installed the PG4.
- Warning 9: File: '%Fs' does not exist. A new file will be created.**
WHY: The user asked to edit a file which does not exist. FUPLA is warning the user that we are about to edit a new file.

Warning 10: **Your PC is now connected to a PCD containing a different user program! Going online with this PCD could crash the user program! Do you want to go online ?**

WHY: The user has disconnected the PGU and connected it to an other PCD.

ACTION: The user should go offline.

Warning 11: **Restarting CPU 0 will restart all CPUs.**

WHY: The PCD FW is done such that when restarting the CPU 0 it restarts all the other CPUs.

Notes :

Messages for the Resource Manager

Error 1:	Bug: Invalid symbol handle.
WHY:	The file containing the symbol was corrupted. This should never occur.
ACTION:	Use back up file.
Error 3:	Cannot create the file '<i>file</i>'.
WHY:	The file could not be created, because out of disk space or file is write protected, or file name is not valid.
EFFECT:	Command cannot be executed.
ACTION:	Make sure enough disk space, make sure file is not write protect, and file name is valid.
Error 4:	Out of memory.
WHY:	Not enough memory to execute action.
ACTION:	Free some memory by exiting other Windows application.
Error 5:	Value out of range. Range is: <i>x</i> to <i>y</i>.
WHY:	A value specified was out of the range allowed.
EFFECT:	The command cannot be accepted.
ACTION:	Put value within range allowed.
Error 6:	DB invalid data !
WHY:	Some data entered in the DB is invalid.
EFFECT:	Command cannot be accepted.
ACTION:	Replace invalid data.
Error 7:	DB contains too big value data.
WHY:	The data contained in the DB contains value which are too big.
EFFECT:	Command cannot be accepted.
ACTION:	Replace the values which are too big.
Error 8:	Quotation marks not allowed in text definition.
WHY:	When entering the text content a quotation mark was entered.
EFFECT:	Command cannot be accepted.
ACTION:	Remove quotation mark.
Error 9:	Disk full !
WHY:	The hard disk, where the information is saved, is full.
EFFECT:	The action cannot be completed.
ACTION:	Remove unnecessary files from your hard disk.

Error 10:	Cannot open file: 'file'.
WHY:	The file could not be opened because it probably did not exist.
EFFECT:	The action cannot be completed.
ACTION:	Make sure the file name and the path given is correct.
Error 11:	Invalid scope !
WHY:	An invalid scope was specified. This should never occur.
Error 12:	Empty symbol definition !
WHY:	A symbol line had no definition. This occurs during an importation.
EFFECT:	Line is ignored.
ACTION:	Remove line, correct line or ignore line.
Error 13:	Invalid count !
WHY:	The count specified during the importation is invalid.
EFFECT:	Line is ignored.
ACTION:	Remove line, correct line or ignore line.
Error 14:	Out of memory.
WHY:	Not enough memory to execute action.
ACTION:	Free some memory by exiting other Windows application.
Error 15:	Invalid name.
WHY:	The name specified is invalid.
ACTION:	Change name.
Error 16:	Name is a reserved word.
WHY:	The name specified is used by the PCD system.
ACTION:	Choose an other name.
Error 17:	Invalid value.
WHY:	The value specified is invalid.
ACTION:	Specify a valid value.
Error 18:	Read error on resource file.
WHY:	An error occurred during the reading of the resource file. Disk error and file corrupted could be the cause.
ACTION:	Use backup files or fixed disk.
Error 19:	Try to read an old resource file format.
WHY:	The resource file read cannot be updated. This should never occur.
Error 20:	Write error on resource file.
WHY:	Cannot write the resource file because the disk is full or is corrupted.
ACTION:	Free some disk space.

Error 21:	Resource database is full.
WHY:	The maximum of entry in the database was reached.
ACTION:	Segment your block in more module.
Error 22:	Name already used.
WHY:	A symbol is already using this name.
ACTION:	Use an other name.
Error 23:	Media and value already used.
WHY:	The specified Media address is already defined.
ACTION:	Use an other address.
Error 24:	Cannot delete, symbol already used.
WHY:	Cannot remove the symbol because it is in use.
Error 25:	Media incompatible to connected type.
WHY:	The media specified is incompatible with the connect type.
ACTION:	Use an other media.
Error 26:	This media cannot be used as an output.
WHY:	This media cannot be used as an output
ACTION:	Use an other media
Error 27:	This media cannot be used as an input.
WHY:	This media cannot be used as an input
ACTION:	Use an other media
Error 28:	A name must be specified, when no value specified.
WHY:	A name must be specified if no value specified.
EFFECT:	Command cannot be accepted.
ACTION:	Specify a name or a value.
Error 29:	An address/value must be specified for medias which are Input/Output or Constants.
WHY:	An address or a value must be specified for medias which are I or O or K.
EFFECT:	Command cannot be accepted.
ACTION:	You must specify an address.
Error 30:	A name must be specified when public or extern scope is used.
WHY:	A name must be specified when public or extern scope is used.
EFFECT:	Command cannot be accepted.
ACTION:	You must specify a name or specify the resource local.

- Error 31:** **No value must be specified for extern symbols.**
WHY: A value was specified for a symbol which is specified external.
EFFECT: Command cannot be accepted.
ACTION: Remove value or specify it to be not external.
- Error 32:** **Invalid Media. Media allowed are: *media allowed list*.**
WHY: The media specified is not allowed for the specific context.
EFFECT: Command cannot be accepted.
ACTION: Use a valid media type which is listed in the list.
- Error 33:** **Invalid media.**
WHY: The media specified is not allowed for the specific context.
EFFECT: Command cannot be accepted.
ACTION: Use a valid media type.

WARNINGS

- Warning 1:** **The block definition: '*BlockSymbol*' was not updated by the import command !**
WHY: This will occur if the user tried to import a symbol associated to a media which is not a block type, to replace the symbol name of a defined block.
EFFECT: The importation of that block is ignored.

Messages for the GRAFTEC

Error 100:	BUG - <location>
WHY:	GRAFTEC detected an internal error.
EFFECT:	The program is aborted.
ACTION:	Contact SAIA, reporting the error message displayed and the last operation performed.
Error 101:	Out of memory.
WHY:	The last operation requires more memory than is available.
EFFECT:	If fatal, the program is aborted, else only the last command is aborted.
ACTION:	Try to work with smaller source modules. Close any unnecessary open applications.
Error 102:	Cannot find DLL <file>.
WHY:	GRAFTEC couldn't load the library module <file>. Possible causes: - The file was not found or invalid. - There is not enough memory. - The Windows version is incorrect.
EFFECT:	The program is aborted.
ACTION:	Check that the file is located in the current directory, in the Windows directory or in the directories specified by the PATH environment variable.
Error 103:	File <filename> not found or invalid.
WHY:	The file <filename> could not be found or read.
EFFECT:	The file <filename> is not opened, and the current operation is aborted.
ACTION:	Check that the file <filename> really exists and that it has a correct format.
Error 104:	Read error on file: <filename>.
WHY:	The file currently being read is corrupted (soft or hard). For example, if a floppy disk is removed during read operation.
EFFECT:	The current operation is aborted.
Error 105:	Write error on file: <filename>.
WHY:	When writing to the disk, GRAFTEC detected a malfunction.
EFFECT:	The current operation is aborted.
ACTION:	Check if the floppy disk was removed during the operation or if there is sufficient disk free space on it.

- Error 106: Cannot open file: <filename>.**
WHY: When reading or writing a file, GRAFTEC could not access the disk properly.
EFFECT: The current operation is aborted.
ACTION: Check if the floppy disk was removed during the operation.
- Error 107: Error reading file pointer.**
WHY: When reading a file pointer, GRAFTEC detected a malfunction.
EFFECT: The code of an element is not available.
ACTION: Check if the floppy disk was removed during the operation.
- Error 108: Error moving file pointer.**
WHY: When setting a file pointer, GRAFTEC detected a malfunction.
EFFECT: The code of an element cannot be saved.
ACTION: Check if the floppy disk was removed during the operation.
- Error 109: Unable to load the code editor: <program>.**
WHY: When trying to edit the code of an element HIDM_CODE, the editor program was missing or access protected, or the environment segment was invalid.
EFFECT: The code of the element cannot be edited.
ACTION: Check that the specified program is located in the directory pointed by the APPSDIR entry in SPG4.INI, in the current directory, in the Windows directory or in the directories specified by the PATH environment variable.
- Error 110: Cannot create window.**
WHY: GRAFTEC could not create a new window (for example, the status barterm status bar, the toolbarterm toolbar or the Print Previewprint preview box).
EFFECT: If fatal, the program is aborted, otherwise only the last command is aborted.
ACTION: Add more memory to your system.
- Error 111: Too many timers.**
WHY: When executing the Go Online HIDM_ONOFFLINE command, GRAFTEC could not load a Windows timer because the maximum number has been reached.
EFFECT: The command is aborted.
ACTION: Close unnecessary open applications to decrease the number of Windows timers used.

- Error 112: Cannot create dialog box.**
WHY: GRAFTEC could not create a new dialog box.
EFFECT: The command is aborted.
ACTION: Close any unnecessary open applications or add more memory to your system.
- Error 113: Bad macro name <macro name>.**
WHY: When calling FUPLA to edit an element, GRAFTEC detected that the FUPLA macro was invalid.
EFFECT: The edit is aborted.
- Error 114: Invalid file extension in <filename>.**
WHY: When opening or saving a file, GRAFTEC detected that the filename extension was not .SFC.
ACTION: You must specify a .SFC filename extension.
- Error 115: Cannot remove SFUP block corresponding to <element>.**
WHY: When deleting an element edited with FUPLA or its code, GRAFTEC could not load SFUP in order to delete the corresponding block.
EFFECT: The element or code is not deleted.
ACTION: Check that the specified program is located in the directory pointed by the APPSDIR entry in SPG4.INI, in the current directory, in the Windows directory or in the directories specified by the PATH environment variable.
- Error 116: Invalid DIB file <filename>.**
WHY: When displaying a Device Independent Bitmap (DIB), GRAFTEC detected that the file is not a valid DIB file.
EFFECT: The bitmap is not displayed.
ACTION: Use a valid bitmap file.
- Error 117: Cannot find SFUP block corresponding to <element>.**
WHY: When editing the code of an element edited with FUPLA, GRAFTEC could not find the corresponding block in FUPLA.
EFFECT: The element is not associated with any FUPLA code anymore.
- Error 118: Cannot renumber SFUP block corresponding to <element>.**
WHY: When renumbering an element edited with FUPLA, GRAFTEC could not load SFUP in order to renumber the corresponding block.
EFFECT: The element is not renumbered.
ACTION: Check that the specified program is located in the directory pointed by the APPSDIR entry in SPG4.INI, in the current directory, in the Windows directory or in the directories specified by the PATH environment variable.

- Error 119: Invalid GRAFTEC file <filename>.**
WHY: When opening a file, GRAFTEC could not read it because it does not contain a GRAFTEC structure.
EFFECT: File read aborted.
ACTION: Open a file edited by GRAFTEC.
- Error 120: Cannot convert <filename> to SFC format.**
WHY: When converting a .SRC to .SFC file, GRAFTEC could not complete the operation because of an error.
EFFECT: Conversion aborted.
ACTION: Open a file edited by GRAFTEC.
- Error 121: Cannot copy <source> to <destination>.**
WHY: When converting a .SRC to .SFC file, GRAFTEC could not copy the file source to destination.
EFFECT: Conversion aborted.
ACTION: Check if you have enough disk space.
- Error 122: Cannot delete <filename>.**
WHY: When converting a .SRC to .SFC file, GRAFTEC could not delete the file <filename>.
EFFECT: Conversion aborted.
ACTION: Check if <filename> is a read-only file and clear the attribute.
- Error 123: Cannot rename <filename>.**
WHY: When converting a .SRC to .SFC file, GRAFTEC could not rename the file <filename>.
EFFECT: Conversion aborted.
- Error 200: Unexpected start of block.**
WHY: When reading the source file, GRAFTEC detected a start of block in the body of another block.
EFFECT: File read is aborted.
ACTION: Assemble your file before invoking GRAFTEC.
- Error 201: Unexpected end of block.**
WHY: Reading the source file, GRAFTEC encountered an EXOB, ECOB, EPB, EFB, ESB, EST or ETR instruction outside of its context.
EFFECT: File read is aborted.
ACTION: Assemble your source file before invoking GRAFTEC.
- Error 202: Unexpected IST/ST/TR.**
WHY: When reading the source file, GRAFTEC detected a start of Step or Transition outside a sequential block.
EFFECT: File read is aborted.
ACTION: Assemble your file before invoking GRAFTEC.

Error 203:	Cannot expand symbol table.
WHY:	There is not enough memory to expand the symbol table.
EFFECT:	The command is aborted.
ACTION:	Split this file into separate modules or reduce the number of symbols.
Error 204:	Duplicate symbol.
WHY:	A symbol with the same name was already defined.
EFFECT:	The command is rejected.
ACTION:	Choose an other name for that symbol. Warning: symbol names are not case sensitive.
Error 205:	SB's symbols must be declared before use.
WHY:	This is the unique restriction where symbols cannot be declared after use.
EFFECT:	The command is rejected.
ACTION:	Put the EQU line of that symbol before the SB definition.
Error 206:	Unresolved reference.
WHY:	When reading the source file, GRAFTEC found a start of block instruction, but no valid block number or symbol after it. NB: for SB's the symbol MUST be declared before use.
EFFECT:	File read is aborted.
ACTION:	Check the start of block instruction.
Error 207:	Duplicate element.
WHY:	When reading the source file, GRAFTEC detected two Steps or two Transitions or two SBs with the same number.
EFFECT:	File read is aborted.
ACTION:	Choose an other number for that element.
Error 208:	Bad output list.
WHY:	When reading the source file, GRAFTEC detected a reference to a non-existent element in the output list of a Step or a Transition.
EFFECT:	File read is aborted.
ACTION:	Assemble your file before invoking GRAFTEC.
Error 209:	Bad input list.
WHY:	When reading the source file, GRAFTEC detected a reference to a non-existent element in the input list of a Step or a Transition.
EFFECT:	File read is aborted.
ACTION:	Assemble your file before invoking GRAFTEC.

- Error 210: Asymmetrical link.**
WHY: When reading the source file, GRAFTEC detected that two elements were not referencing each other symmetrically.
EFFECT: File read is aborted. The faulty elements are listed in the error box.
ACTION: Do not create links by hand in the source file, GRAFTEC does it for you properly.
- Error 211: Constant out of range in <file> : <entry_name> (default value used).**
WHY: When reading the configuration file <file>, GRAFTEC detected an out of range value associated with the entry <entry_name>.
EFFECT: The default value is used.
- Error 212: Out of range number <number>.**
WHY: When reading the source file, GRAFTEC detected an out of range value.
EFFECT: File read is aborted.
ACTION: Assemble your file before invoking GRAFTEC.
- Error 213: SB <SBnumber> already exists.**
WHY: When renumbering a SB, GRAFTEC detected that the new SB number was already in use.
ACTION: The number of the new SB must be re-entered.
- Error 214: Unexpected end of file.**
WHY: The ESB instruction of the last SB is missing.
EFFECT: File read is aborted.
ACTION: Assemble your source file before invoking GRAFTEC.
- Error 300: Too many input links.**
WHY: When reading the source file or inserting a new element, GRAFTEC detected that a Step or a Transition has more than 32 input links.
EFFECT: If reading the source file, aborts the reading, otherwise cancels the insertion of the element in process.
- Error 301: Too many output links.**
WHY: When reading the source file or inserting a new element, GRAFTEC detected that a Step or a Transition has more than 32 output links.
EFFECT: If reading the source file, aborts the reading, otherwise cancels the insertion of the element in process.

- Error 302: Unable to build graph.**
WHY: When inserting an element, GRAFTEC detected that there was not enough memory to build the new structure.
EFFECT: The command is aborted.
ACTION: Exit the program without saving the structure.
- Error 303: Macro-Step not found.**
WHY: While executing the command, GRAFTEC did not find the specified Macro-step in the current SB.
- Error 304: (I)Step not found.**
WHY: While executing the command, GRAFTEC did not find the specified (I)Step in the current SB.
- Error 305: Transition not found.**
WHY: While executing the command, GRAFTEC did not find the specified Transition in the current SB.
- Error 306: Maximum number of elements per page reached.**
WHY: When inserting a new element, GRAFTEC detected that the maximum number of elements (512) in a Page has been reached.
EFFECT: The new element is not inserted in the structure.
ACTION: Decrease the number of elements per Page by creating new Page.
- Error 307: (I)Step cannot be inserted here.**
WHY: While inserting an IStep or a Step, GRAFTEC detected that the selected element was the first or last element of a subpage, an element with no input or a double horizontal link.
EFFECT: The insertion is aborted.
- Error 308: Transition cannot be inserted here.**
WHY: While inserting a Transition, GRAFTEC detected that the selected element was the first or last element of a subpage, an element with no input or a simple horizontal link.
EFFECT: The insertion is aborted.
- Error 309: Link not possible here.**
WHY: While inserting a link, GRAFTEC detected that the source or destination element was an illegal element or link.
EFFECT: - If the source element is illegal, the command is aborted and the Select mode is activated.
- If the destination is illegal, you must specify another destination.

- Error 310: Cannot remove a link here.**
WHY: While removing a link, GRAFTEC detected that the selected part of the structure was not a link.
EFFECT: The operation is aborted.
- Error 311: Bad start of sequence.**
WHY: - While selecting a sequence, GRAFTEC detected that the first element was not a Step, an IStep or a Transition.
- While executing the Page Create command, GRAFTEC detected that the first element of the selected sequence was a Transition, a link, the first or last Step of a subpage or a Step with no input.
EFFECT: The operation is aborted, and the Select mode is activated.
- Error 312: Bad end of sequence.**
WHY: - While selecting a sequence, GRAFTEC detected that the last element was not a Step, an IStep or a Transition.
- While executing the Page Create-command, GRAFTEC detected that the last element of the selected sequence was a Transition, a link, the first or the last Step of a subpage, a Step with no input or the first Step of the sequence.
EFFECT: The operation is aborted, and the Select mode is activated.
- Error 313: Bad sequence.**
WHY: While executing the Copy or Cut Delete-command, GRAFTEC detected that the specified sequence could not be removed or re-inserted properly.
EFFECT: You must specify another sequence.
- Error 314: Sequence cannot be reinserted here.**
WHY: While executing the Cut, the Paste or the Import-command, GRAFTEC detected that the element where to insert the sequence was a bad element or that the sequence could not be re-inserted in the middle of a sequence.
ACTION: Check the first and last element of the sequence.
- Error 315: Removing sequence produces too many output links.**
WHY: While executing the Cut or Delete-command, GRAFTEC detected that too many output links were appended.
EFFECT: The command is aborted.
- Error 316: Missing number.**
WHY: When editing an element, GRAFTEC detected that the element had no number.
EFFECT: You must give a number to the element.

Error 317:	Out of range value.
WHY:	While reading the source file or a number from the keyboard, GRAFTEC detected a range error for a value.
EFFECT:	You must re-enter the value.
Error 318:	Attempt to link parallel processes.
WHY:	While inserting a link, GRAFTEC detected an invalid end of link. For example, a loop back from a Step to a Transition is not allowed.
EFFECT:	You must specify another destination.
Error 319:	Element cannot be a Macro-Step.
WHY:	While executing the Page Create command, GRAFTEC detected that the selected element was a Transition, a link or the first or last Step of a subpage.
EFFECT:	The command is aborted.
Error 320:	Element is already a Macro-Step.
WHY:	While executing the Page Create command, GRAFTEC detected that the selected element was a Macro-step.
EFFECT:	The command is aborted.
Error 321:	Element is not a Macro-Step.
WHY:	While executing the Page Expand command, GRAFTEC detected that the selected element was not a Macro-step.
EFFECT:	The operation is aborted.
ACTION:	To expand a Page, you must select a Macro-step before executing the Page Expand command.
Error 322:	Element cannot be edited.
WHY:	- While executing the Edit Element command, GRAFTEC detected that the selected element was not a Step, IStep, Transition or Macro-step. - While executing the Edit Code command, GRAFTEC detected that the selected element was not a Step, IStep or Transition, or that the structure is not saved when editing a element with FUPLA.
EFFECT:	The command is aborted.
Error 324:	Maximum number of elements reached.
WHY:	When inserting a new element, GRAFTEC detects that the maximum number of elements (4000) has been reached.
EFFECT:	The new element is not inserted in the structure.

- Error 325: No Page(s) selected: Use 'Range...'**
WHY: When printing or using the Print preview command, there were no Pages selected when using the Selected Pages button in the Print dialog box.
EFFECT: The command is aborted.
ACTION: Press the Range-button to define the Pages to be printed.
- Error 326: Cannot find printer driver.**
WHY: When printing, GRAFTEC could not find the current printer driver, because there is no printer installed.
EFFECT: The command is aborted.
ACTION: Install a new printer using the Windows Control Panel (see your Windows documentation).
- Error 327: Cannot print page.**
WHY: When printing, an error occurred.
Possible causes:
- Not enough disk space or memory available for spooling.
- You terminated the job through Print Manager.
EFFECT: The printing is aborted.
ACTION: Check that there is enough free space left on your disk.
- Error 328: Element already exists.**
WHY: When editing an element, GRAFTEC detected that the specified element number was already used by another element.
EFFECT: You must re-enter the value.
- Error 329: Incomplete sequence.**
WHY: When executing the Cut, Copy, Delete or Page Create command in Sequence mode, you have not defined the last element of the sequence.
EFFECT: Define the last element of the sequence. See Select a sequence.
- Error 330: Bad symbol <symbolName>.**
WHY: The symbol's name you have edited is not valid. Symbols cannot begin with a digit (0-9). Characters allowed in a symbol are:
_ (underscore), A..Z, a..z, 0..9, and foreign characters:
Æ Å Ä å ä à É é è ë ì î ï Ö ö ò ò Û ü û ú Ç ç ñ Ñ ÿ.
EFFECT: You must re-enter the symbol's name.
- Error 331: Cannot renumber elements.**
WHY: When using the Renumber-command, GRAFTEC could not renumber elements.

- Error 400: PCD not connected or powered off.**
WHY: - When executing the Trace, Run, Stop or Find Active command, GRAFTEC was in Offline mode.
- When executing the Go Online command or during On-line mode, GRAFTEC could not communicate properly with the PCD.
EFFECT: The command is aborted and GRAFTEC goes Offline.
ACTION: - Go Online.
- Check that the PCD is powered on and connected properly to the correct serial port on your Personal Computer.
- Error 403: PCD halted, command not accepted.**
WHY: While executing the Run, Stop or Trace command, the PCD has detected a command not accepted by the CPU.
EFFECT: The command aborted, GRAFTEC goes Offline.
ACTION: Check the current version of your firmware and update it if necessary
If the PCD is halted, run SBUG and use the "Restart Cold" command.
- Error 404: Communication error.**
WHY: Executing Go Online command, or during Online, a communication error has occurred (parity, framing, bad CRC...).
EFFECT: GRAFTEC goes Offline.
ACTION: Retry. Check the communication cable between your Personal Computer and the PCD.
- Error 405: PCD/File program mismatch.**
WHY: Executing the Go Online-command, or during Online, GRAFTEC detected that the program within the PCD and the program read from the disk are not the same.
EFFECT: GRAFTEC goes Offline.
ACTION: Load the currently edited file into the PCD using SBUG's "File Load" command.
- Error 406: No active Transitions.**
WHY: While executing the Find Active command, no active Transitions were found.
EFFECT: The command is aborted.
- Error 407: Cannot read active Transitions.**
WHY: In Online mode, GRAFTEC couldn't read the active Transitions from the PCD.
EFFECT: The active Transitions are not correctly displayed.
ACTION: Check the message in the title of the error box.

- Error 408: Cannot run CPU <number>.**
WHY: While executing the Run command, GRAFTEC couldn't put the PCD's CPU into run mode.
EFFECT: The command is aborted.
ACTION: Check the message in the title of the error box.
- Error 409: Cannot stop CPU <number>.**
WHY: While executing the Stop command, GRAFTEC couldn't stop the PCD's CPU.
EFFECT: The command is aborted.
ACTION: Check the message in the title of the error box.
- Error 410: Cannot run conditionally CPU <number>.**
WHY: While executing the Trace command, GRAFTEC couldn't put the CPU into "Conditional Run" mode.
EFFECT: The command is aborted.
ACTION: Check the message in the title of the error box.
- Error 411: SB <number> never called in the PCD program.**
WHY: When executing the Trace command, GRAFTEC detected that you are trying to trace in an SB that is never called in the program.
EFFECT: The PCD returns to the last status before tracing.
ACTION: In your program, use the CSB instruction to call the specified SB.
- Error 412: Cannot go online.**
WHY: When executing the Go Online-command, a communication error has occurred (PCD not connected, cable fault, ...).
EFFECT: The command is aborted.
ACTION: - Check that the PCD is powered on and connected properly to the correct serial port on your Personal Computer.
- Check the communication cable between your Personal Computer and the PCD.
- Error 413: You must save the structure before going online.**
WHY: When executing the Go Online command, GRAFTEC detected that a new structure had not been saved, or that there has been modifications since last save.
EFFECT: The command is aborted.
ACTION: Save your structure.

WARNINGS

Warning 11: Missing include filename.

WHY: When reading the source file, GRAFTEC detected a \$INCLUDE directive with no filename.

EFFECT: May cause errors during file checking.

Warning 12: Conditional assembly not supported.

WHY: When reading the source file, GRAFTEC detected one of the \$IF, \$IFN, \$IFDEF, \$IFNDEF assembler directives.

EFFECT: The entire source file will be processed.

May cause errors during file checking.

ACTION: Never use conditional assembly directives in the body of an SB.

Warning 13: End of number table is reached, offset is set to 0.

WHY: When creating an element, the end of the number table is reached.

No number is available between the elements offset and the maximum number of elements.

EFFECT: The offset is set to zero because the upper limit of the table has been reached.

Warning 14: Cannot find printer driver. (default values used)

WHY: When printing to an ASCII file, GRAFTEC could not find the current printer driver to determine the length and height of a printer sheet.

EFFECT: The default values are used (80 characters per line and 60 lines per sheet).

Warning 15: Cannot change the current drive to <drive>.

WHY: While opening a source file, GRAFTEC could not change the current drive to the drive where the opened file is located.

EFFECT: The current drive and directory is not the same as the one containing the opened file.

Warning 16: Cannot change the current directory to <directory>.

WHY: While opening a source file, GRAFTEC could not change the current directory to the directory where the opened file is located.

EFFECT: The current drive and directory is not the same as the one containing the opened file.

Notes :

Appendix B: Reservations in the PG4 - V 1.3

The following are further, important functions of the PG4 concept, which require major additional development expenditure to realize. The functionalities listed below are not yet included in the present version 1.3:

IL (Instruction List):	Powerful editor Resource manager Debugger
Communication:	Text and character output PROFIBUS applications
FUPLA:	Printing multiple blocks (especially in GRAFTEC) Fbox editor Inclusion of FB s with parameters set Data block transfer Online help
GRAFTEC:	Printing FUPLA code Copying FUPLA code when sequences are copied.
Resource manager:	ONLINE display and modification capabilities Cross-references
Program manager:	EPROM creation Data transfer
PG4 general:	Run-time package Variable time basis for timers. Online downloading of program modules

Unfortunately, it is not yet possible to give firm details of performance, priorities or deadlines for the items mentioned.

Notes :

SAIA® Process Control Devices

Programming tools for MS WINDOWS

PG4 - Version 1.4

This document contains the main differences between versions 1.3 and 1.4, plus the new capabilities of version 1.4.

This document is a supplement to

Manual 26/748 E1, dated 05.96

SAIA-Burgess Electronics Ltd. 1998. All rights reserved
Edition 26/748A E1 - 01.1998

Subject to technical changes

Updates

Manual : Programming tools for MS WINDOWS - PG4 Version 1.4 - Edition E1

Date	Chapter	Page	Description
27.10.2000	---	---	Small updates for the "Support Homepage"

Major additions and enhancements

The main change has been to enhance the PG4 with the following applications:

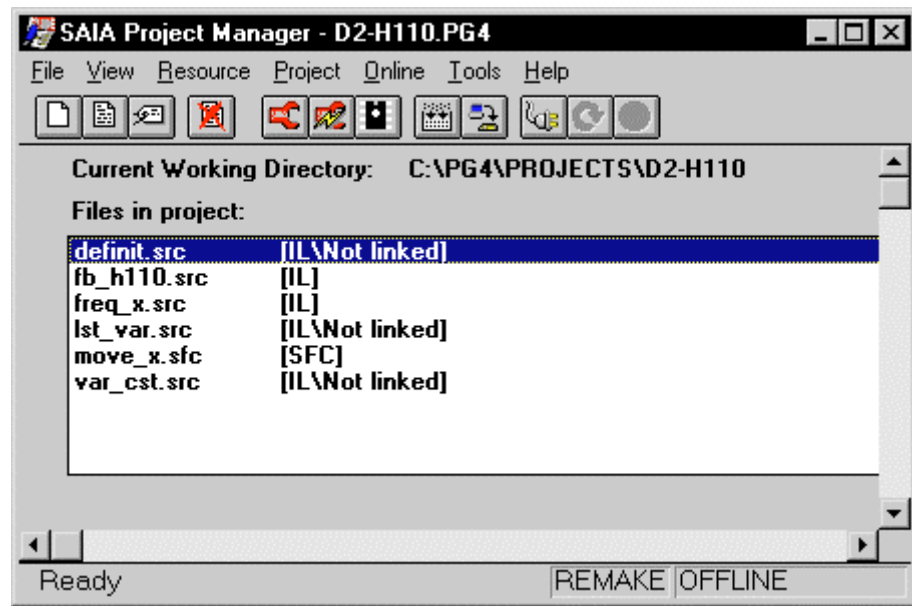
- Additions and enhancements pages 02 - 14
- Configuration file editor (offline configurator) pages 15 - 26
- EPROM programmer pages 27 - 41
- New designed FUPLA
- New FBox library
- RIO editor is now integrated into the PG4 software
- PROFIBUS FMS editor too, is now integrated into the PG4 environment
- PG4 supports the PCD2.M220 (PC/104)
- New functions in the S-Bus communication

All applications are called from the Project Manager, resulting in a few changes to the latter.

FUPLA has also received a facelift with the addition of a toolbar. The previous toolbox (editing tools) could be moved freely around the screen, the new version 1.4 has the toolbox and online functions on the toolbar, and also in a new menu item 'Mode'.

The package now has comprehensive help files in English.

The new Project Manager

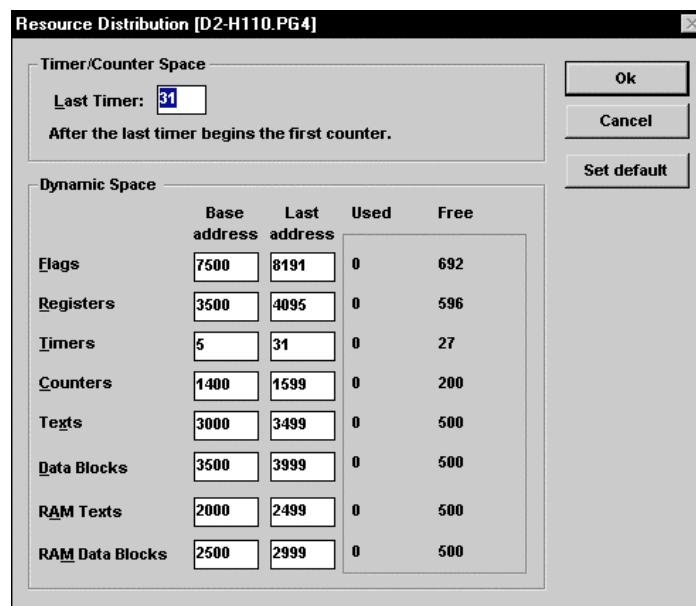


Note the expanded toolbar. There is a description of each button at the end of this section.

Menu items up to the new item 'Project' are the same as in the previous version. Other differences are indicated below.

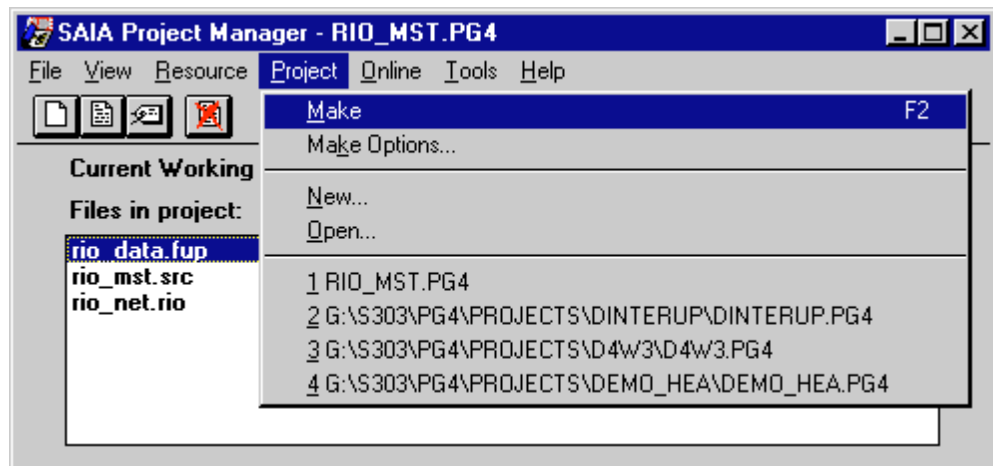
'Resource' submenu

After clicking on this item, 'Dynamic distribution...' is shown. Another click displays the following window:



The items 'RAM Texts' and 'RAM Data Blocks' have been added.

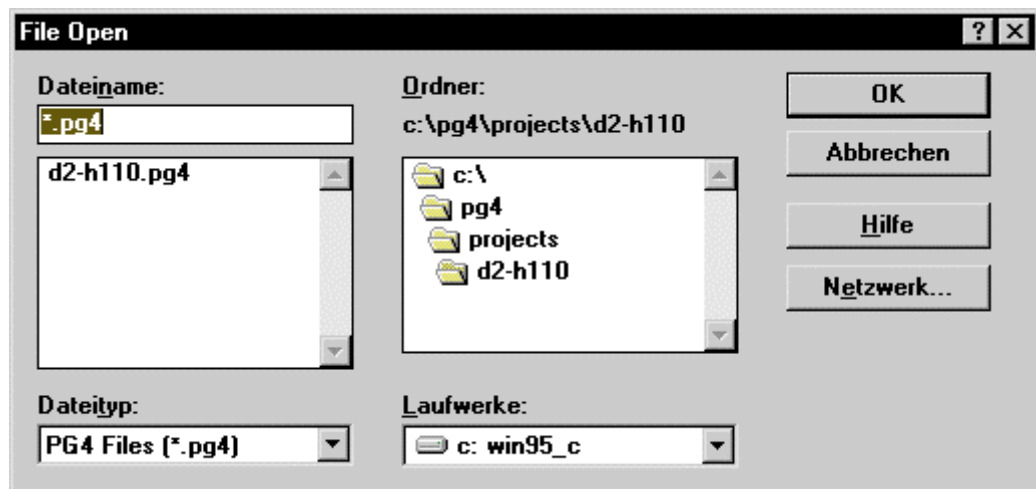
New 'Projects' submenu



The 'Make' and 'Make Options...' commands have been transferred to this submenu. Files are now automatically opened and compiled when needed. You don't have to compile each file separately before starting the MAKE.

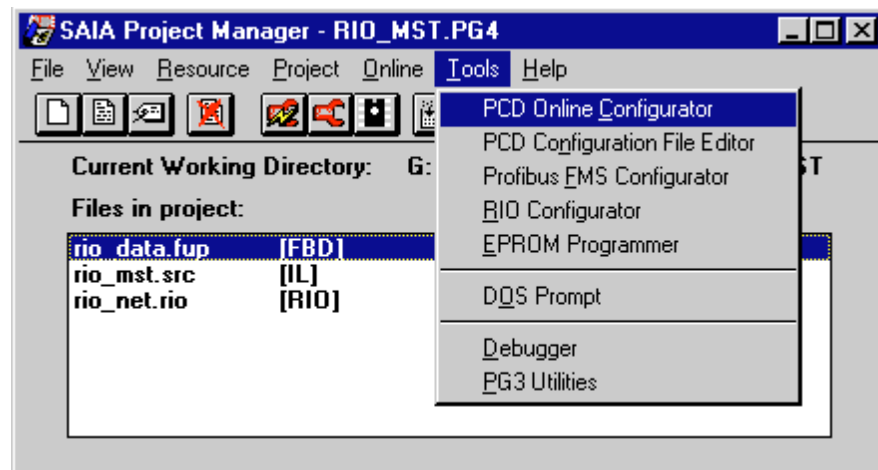
'NEW' creates a new project directly from the Project Manager.

'Open...' can be used to open a project file.



'1, 2, 3, 4' enables the last 4 projects used to be reopened.

Expanded 'Tools' submenu

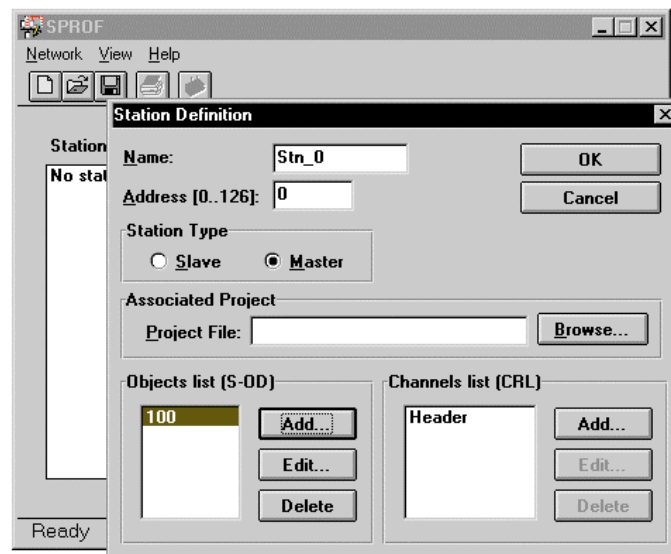


New tools:

- PCD Online Configurator
- PCD Configuration File Editor
- Profibus FMS configurator
- RIO configurator
- EPROM programmer

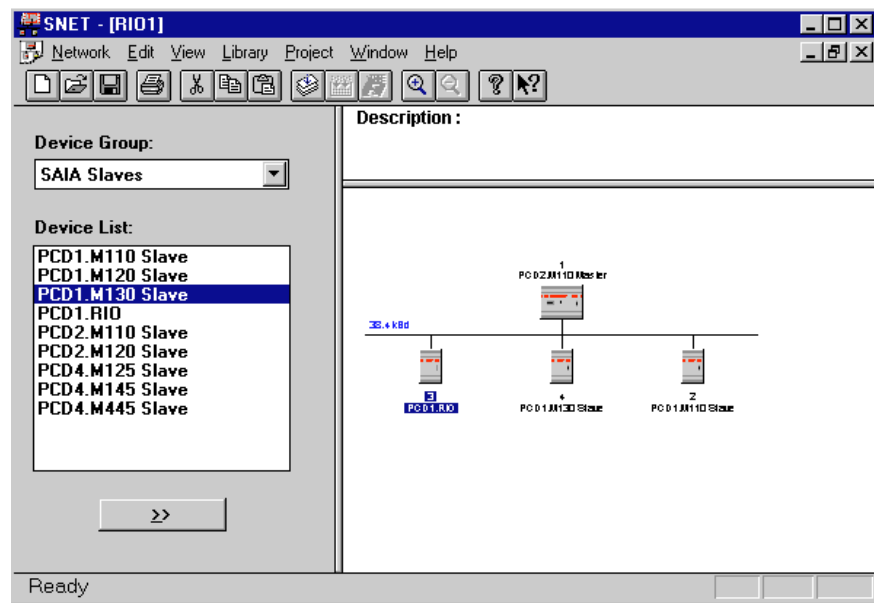
The 'PCD Online Configurator', the 'PCD Configuration File Editor' and the 'EPROM Programmer' are described in the next sections.

The PROFIBUS-FMS configurator looks like this:



For instructions, see PROFIBUS manual (26/742).

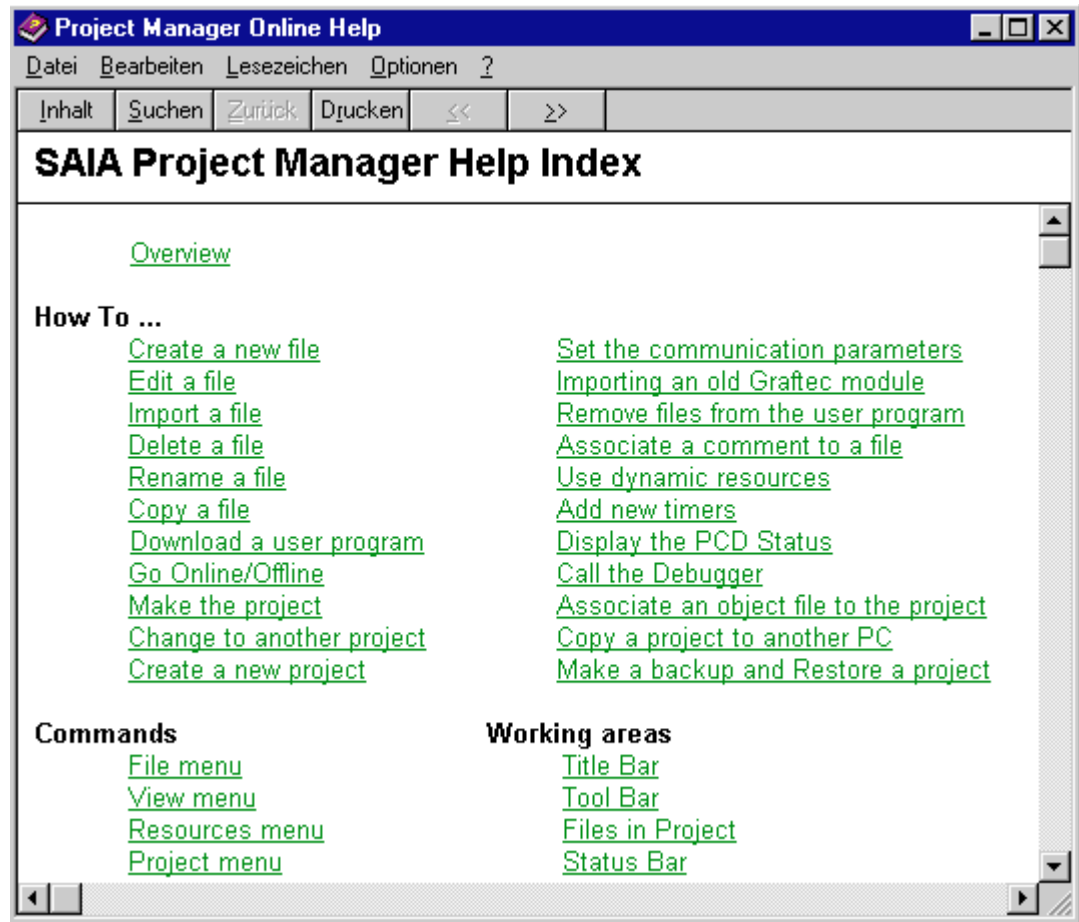
The RIO configurator is like this:



For instructions, see RIO manual (26/751).

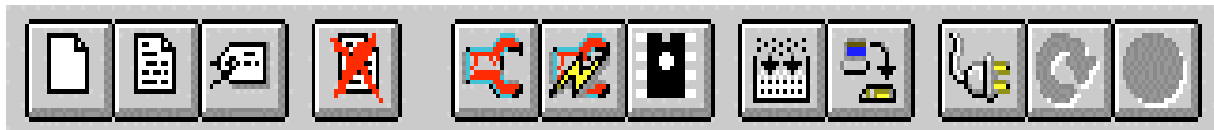
'Help' submenu

The overview of help topics is as follows:



The new list of all "Errors" and "Warnings" is very useful.

Project Manager's new toolbar



<Ins> Opens a new file in the current project
same as menu 'File' - 'New...'



<CR> Opens the marked file.
same as menu 'File' - 'Edit File'



<Alt>+ <CR> Renames a file or changes the properties of a file
same as menu 'File' - 'Rename/properties...'



 Deletes a file
same as menu 'File' - 'Delete'



Online configuration of PCD
same as menu 'Tools' - 'PCD Online Configurator'



Offline configuration of PCD.
same as menu 'Tools' - 'PCD Configuration File Editor'



EPROM programmer
same as menu 'Tools' - 'EPROM Programmer'



<Ctrl> + <M> Assembles and links the user program
same as menu 'File' - 'Make'



Downloads the program to the PCD
same as menu 'Online' - 'Download'



Switches the editor to online mode

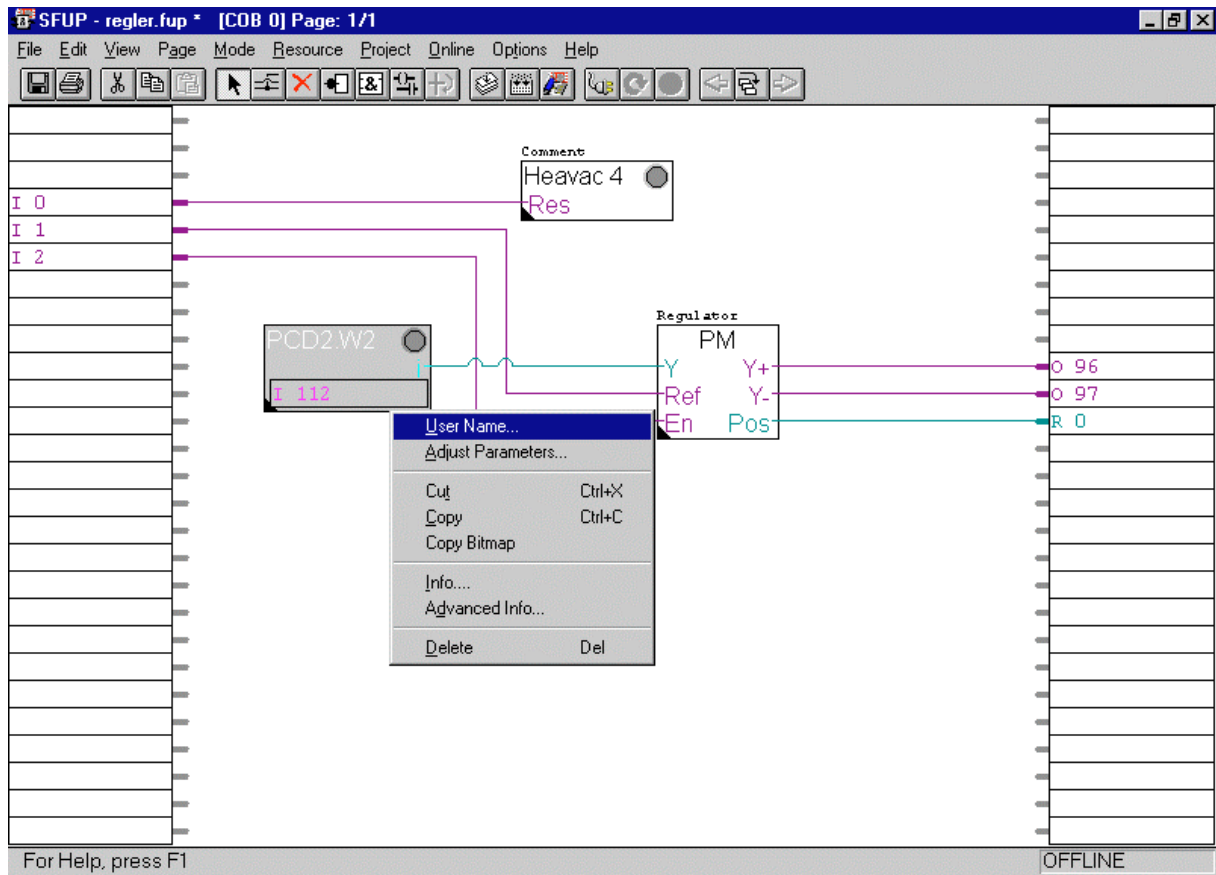


Puts the PCD into Run, when online



Stops the PCD, when online

Redesigned FUPLA user interface



FUPLA's new toolbar



Save
same as 'File' - 'Save'



Print
same as 'File' - 'Print...' (New DIN Format !)



















Cut out → clipboard
same as 'Edit' - 'Cut'



Copy → clipboard
same as 'Edit' - 'Copy'



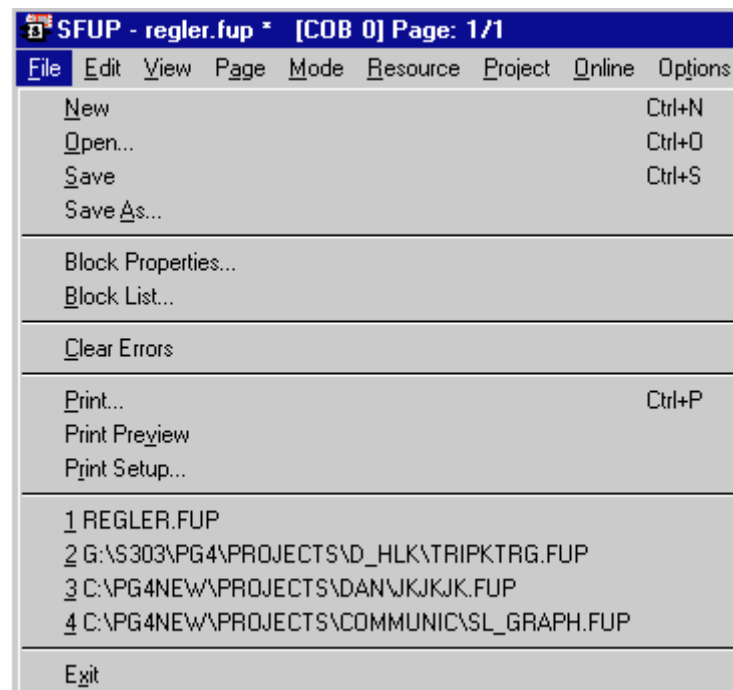
Insert contents of clipboard
same as 'Edit' - 'Paste'

	Cursor positioning, gen. character symbol same as 'Mode' - 'Select'
	Connection lines same as 'Mode' - 'Lines'
	Delete (eraser) same as 'Mode' - 'Erase'
	Inverter for binary signals same as 'Mode' - 'Invert Binary Inputs'
	Select new FUPLA functions same as 'Mode' - 'Add FBox'
	Select new KOPLA functions same as 'Mode' - 'Add KOPLA'
	Terminate a GRAFTEC transition same as 'Mode' - 'Add End of Transition'
	Compile current FUPLA file same as 'Compile' - 'Compile'
	Executes 'Make' same as 'Compile' - 'Make'
	Calls the Project Manager same as 'Compile' - 'Project Manager'
	Switches FUPLA online/offline same as 'Online' - 'Go Online' / 'Go Offline'
	Switches CPU into Run (when online) same as 'Online' - 'Run'
	Switches CPU to Stop (when online) same as 'Online' - 'Stop'
	Go back 1 FUPLA page same as 'Page' - 'Previous'
	Show index of all FUPLA pages same as 'Page' - 'Index'
	Go to next FUPLA page same as 'Page' - 'Next'

Other functions in menu without switching symbol.

- Expanded 'File' - Menu
- 'Mode' - 'Set Probes': sets the online probes
- 'Mode' - 'Set Breakpoints'
- 'Options' - 'Colors': Changes the colors of the Fupla-elements and the background.
- 'Project' replaces the point 'Compile'.
- Improved use of the right-hand mouse button.

Extended File menu

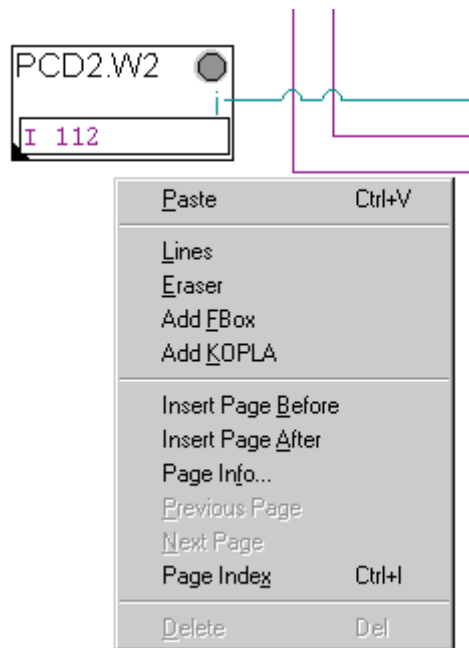


Fuplafiles may now be managed and opened from the pulldown menu 'File':






- 'New' creates a fuplafile. The existing file will be closed and saved.
- 'Open' opens a existing file.
- 'Print' prints the fupla schematic in the DIN format.
- 'Print Preview' opens a view on the schematic and the surrounding documentation.
- The list allows the direct access to the last four files you worked with.

Use off the right-hand button

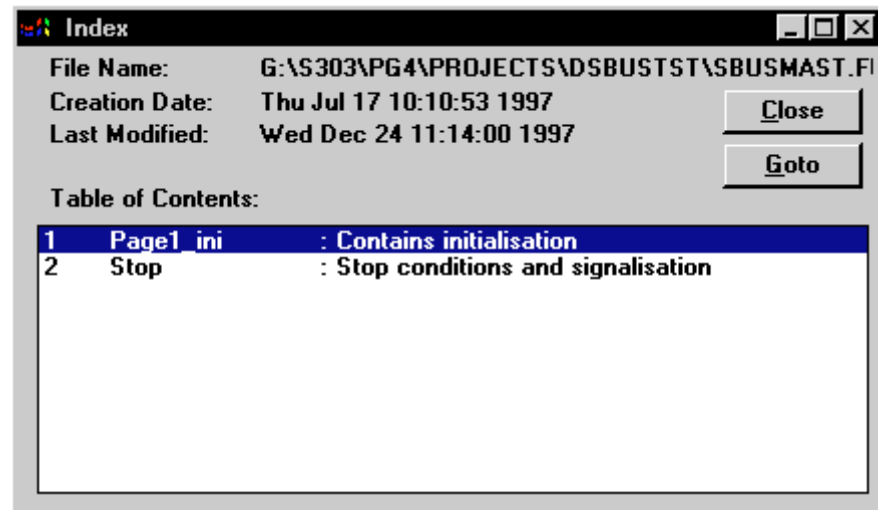
The activation off the right-hand mouse button on a fupla schematic opens the following window:



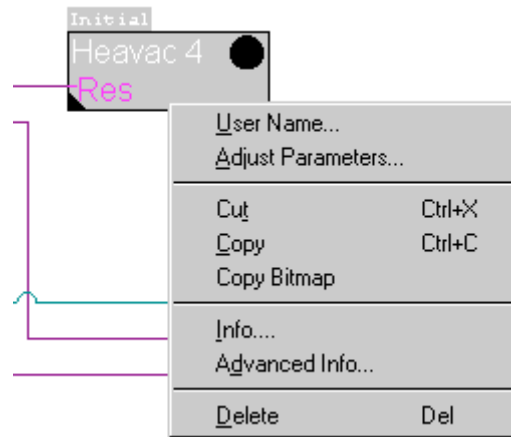
There after you have a list off functionality's:

- 'Paste' inserts a previously copied FBox. The parameters off the FBox are also copied.! corresponds to the  button.
- 'Lines' corresponds to the  button.
- 'Eraser' corresponds to the  button.
- 'Add FBox' corresponds to the  button.
- 'Add KOPLA' corresponds to the  button.
- 'Page Info....' opens a textwindow. In the textwindow you may document you schematics. Once you print the schematic, this text will be included in the documentation.


- 'Page Index' opens a view on all the pages off your fuplfile. Each page has his own little comment. The 'GOTO' button opens the corresponding page. The date of creation as well as the date when the last modification occurred are listed.



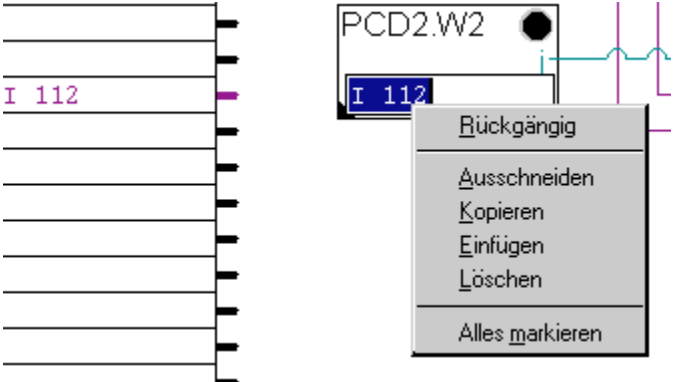
The activation off the right-hand mouse button on a FBox opens the following window:



This window has some new features:

- With 'User Name...' you can add a nametag to each FBox. This is particularly useful in order to increase the readability of the schematics. The username off the example above is 'Initial'. There is the possibility to add some lines of explanation to each FBox.
- 'Adjust Parameters...' opens the window with all the parameters. You can have the same result by clicking twice on the FBox.
- 'Cut' and 'Copy' are convenient too and they work from one page to the other. They do not only move the FBox they also move the attached parameters.
- 'Copy Bitmap' copies the BitMap of the FBox. This is mainly for documentation purposes.
- 'Info...' opens the helpfile of the respective FBox.
- 'Advanced Info...' contains additional information like the version an so on.
- 'Delet' corresponds to the  button.

The activation off the right-hand mouse button on a Resource opens the following window:



This is a way to copy resources rapidly. Double-click let's you edit the activated resource.

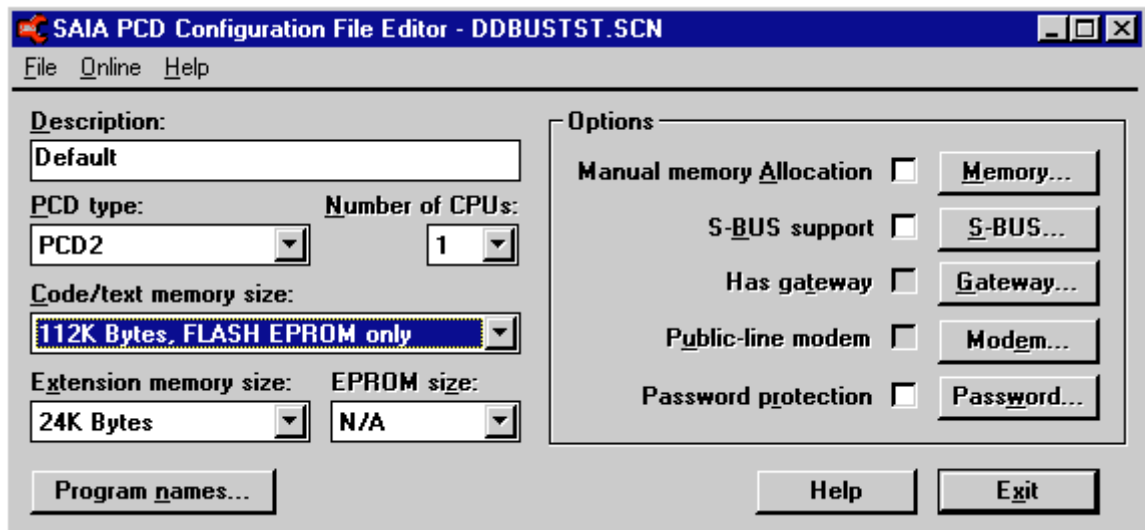
Configuration file editor

The "SAIA PCD Configuration File Editor" (SCONFWIN.EXE) configures memory allocation, serial communications between the programming unit and PCD (S-Bus), and password protection.

The configuration is stored in the PCD's battery-backed RAM.

If the user program in the PCD is stored in EPROM then the configuration is also stored in the EPROM and it cannot be modified. In this case, the configuration file editor must be used to create a configuration file before an EPROM is produced with the EPROM programmer.

This configuration file can also be downloaded into the connected PCD, or uploaded from the PCD and displayed.



With the "SAIA PCD **Online** Configurator" (SCONFWIN.EXE), the configuration is altered directly in the PCD's memory. However, it can be useful to have the configuration stored in a file, so that it can be viewed at any time and downloaded again. Also, when configuring different PCDs with similar configurations, the configuration file can be used to simplify the work.

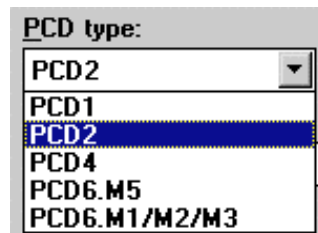
The file extension for the configuration file is **.SCN** (Saia ConfigurationN)

Main window

Description:

A general-purpose text which can be used as a title. This text appears in the configuration file under 'Comment'. It is not stored in the PCD.

PCD type:

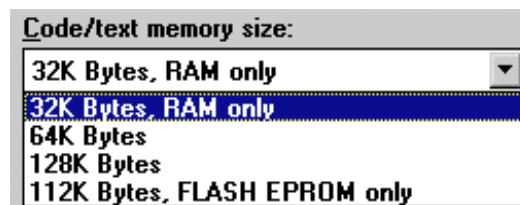


The PCD type must be selected first because it affects the values of other configuration items, such as the memory size, memory type and number of CPUs.

Number of CPUs:

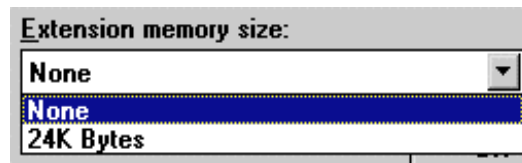
Defines the number of CPUs fitted, for PCD4 (max. 2 CPUs) and PCD6 (max. 6 CPUs).

Code/text memory size:



Memory selection, shown for the PCD2.

This must be set to the type and size of memory actually fitted in the PCD. The list depends on the chosen PCD type. Custom memory allocation can be done using 'Manual memory Allocation' in the 'Options' section. If custom memory allocation is not used, the default allocation applies. The PG4 supports now FLASH EPROM!!

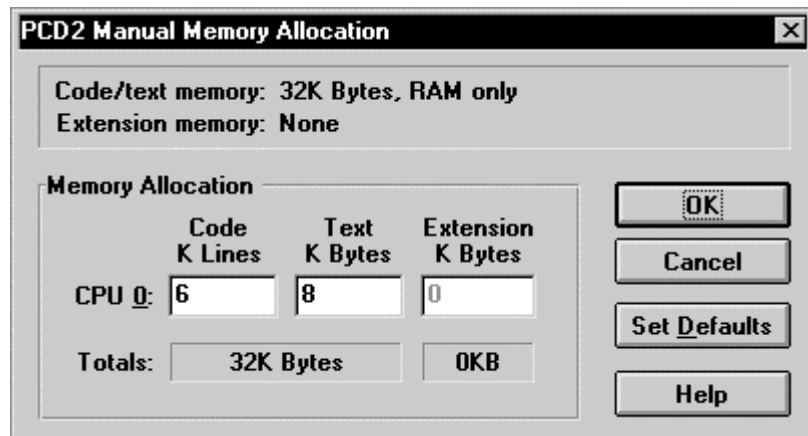
Extension memory size:

Shown. for the PCD2. 'None' signifies no extension memory.

Refer to the relevant hardware manuals for the possibilities offered by extension memory.

EPROM size:

Shows the size of EPROM needed for the selected PCD and memory. N/A is shown if EPROMs cannot be used with the selected memory card or size. In some cases the EPROM size is selectable.

Options:**Manual memory Allocation.**

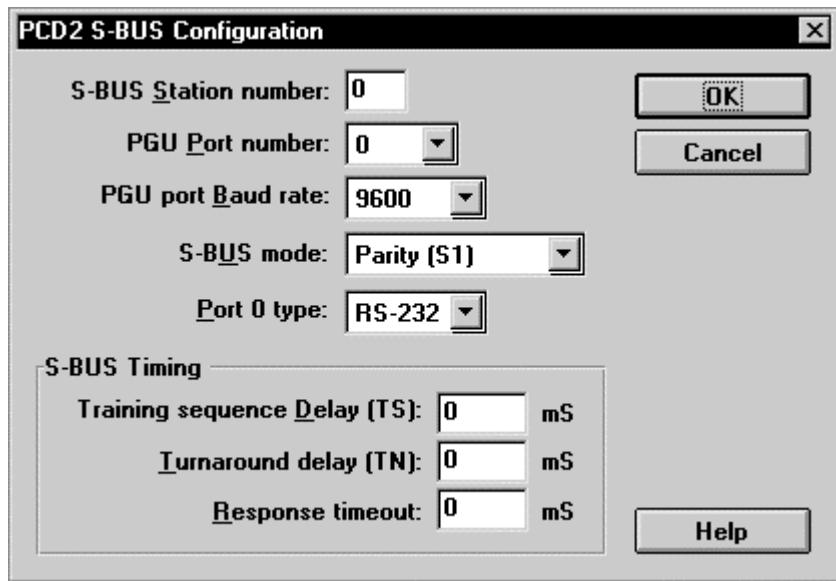
Example for a PCD2. The 'Manual memory allocation' option must be checked to display this window by pressing the "Memory..." button.

The sum of code and text sizes must match the total memory size. The number of K code lines is multiplied by 4 to give the number of K bytes, because each program line uses 4 bytes (32 bits).

For this example:

$$\begin{array}{rcl}
 6 \text{ K code lines} \times 4 & = & 24 \text{ K Bytes} \\
 8 \text{ K text} & = & 8 \text{ K Bytes} \\
 & & \text{--} \\
 \text{Total} & = & 32 \text{ K Bytes}
 \end{array}$$

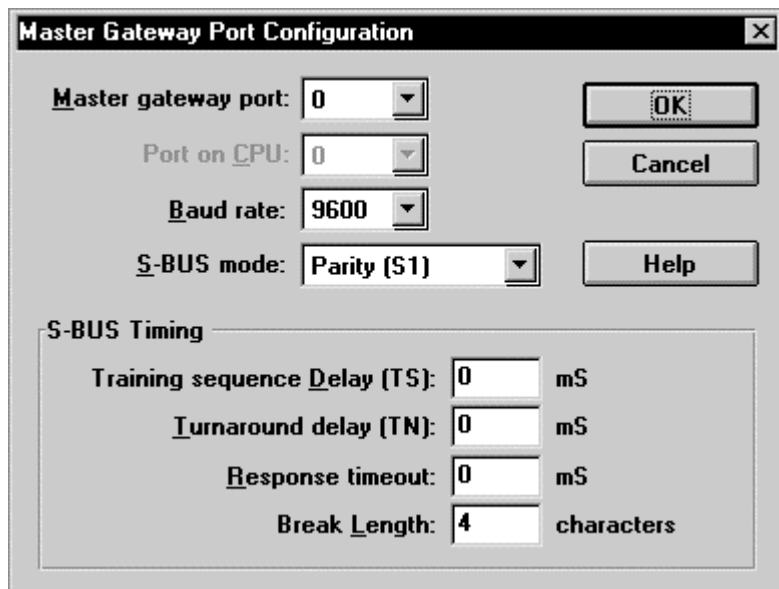
S-Bus support:



Window for setting S-Bus parameters. The 'S-BUS support' option must be checked to display this window by pressing the 'S-BUS...' button.

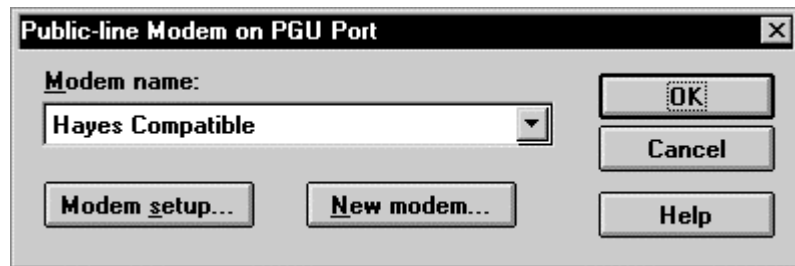
See the appropriate manuals for further information.

Has gateway:



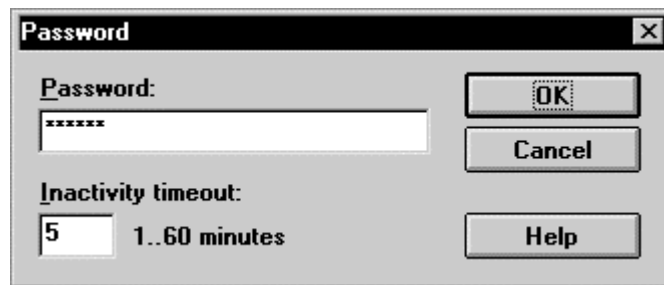
Window for setting the master gateway port configuration. The 'Has gateway' option must be checked to display this window with the 'Gateway...' button.

See the appropriate manuals for further information.

Public-line modem:

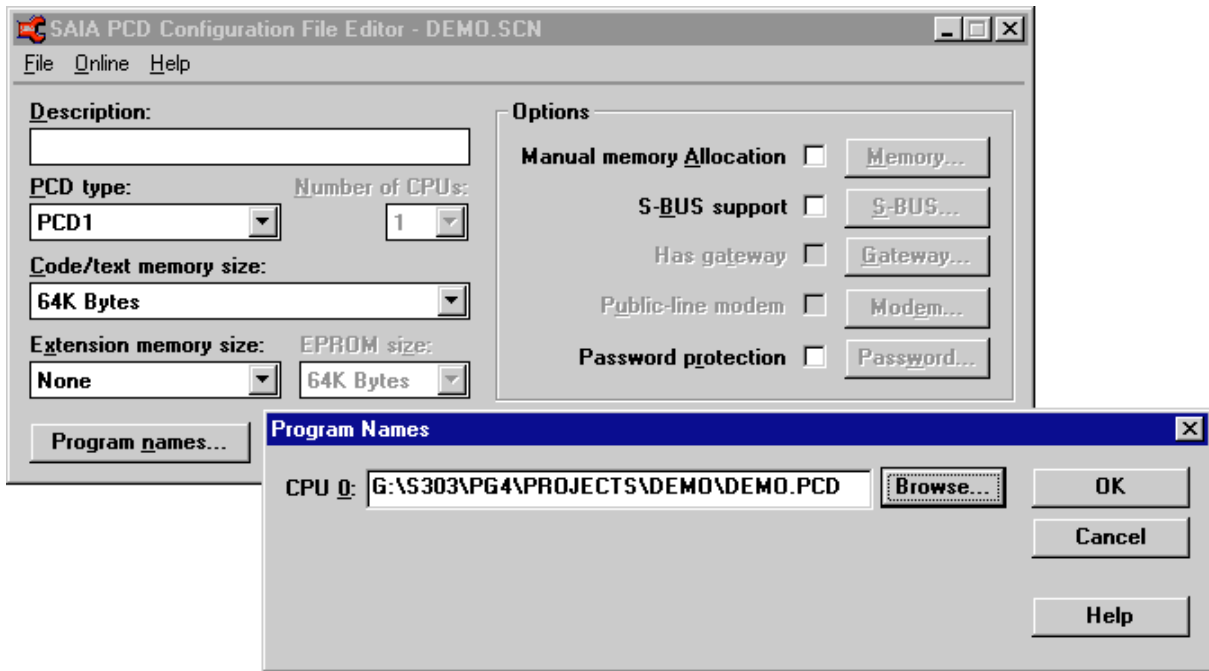
Window for selecting the PGU port's modem. The 'S-Bus support' **and** the 'Public-line modem' options must be checked to display this window with the 'Modem...' button.

See the appropriate manuals for further information.

Password protection:

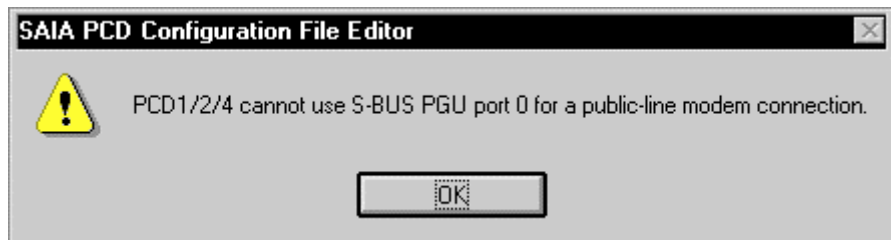
Window for setting password protection for access to the PCD control and data. The 'Password protection' option must be checked to display this window with the 'Password...' button.

See the appropriate manuals for further information.



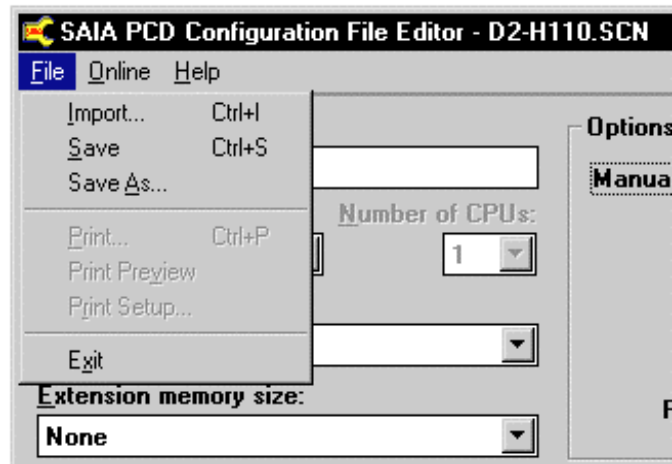
OK

The configuration file editor is exited with 'OK'. If data has been modified, 'File' - 'Save' or 'File' - 'Save as...' should be done before exiting. Before saving, an error message may be displayed if the configuration is incorrect, for example:



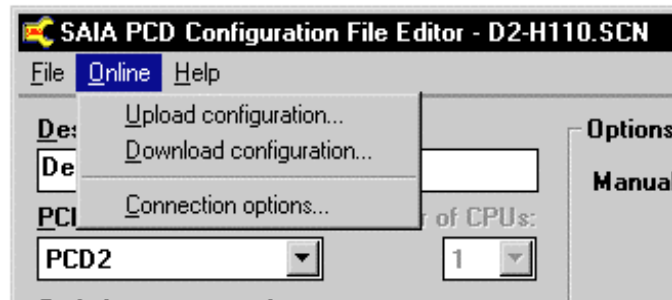
Menus

File submenu



The printing commands are not available in this version.

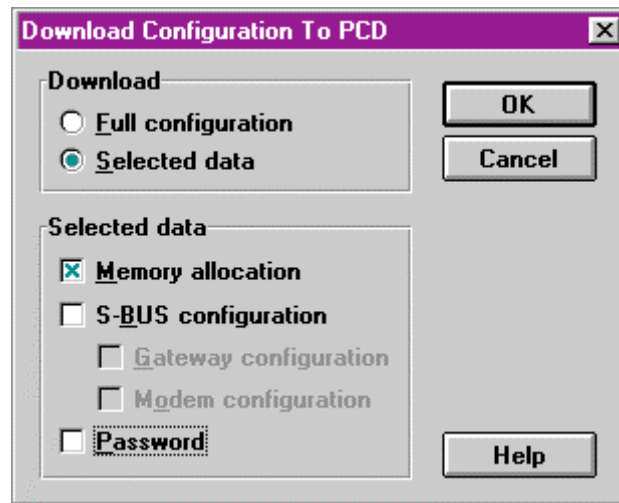
Online submenu



Upload configuration:

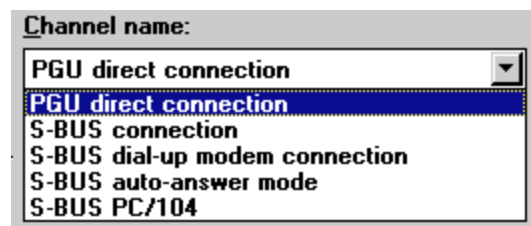
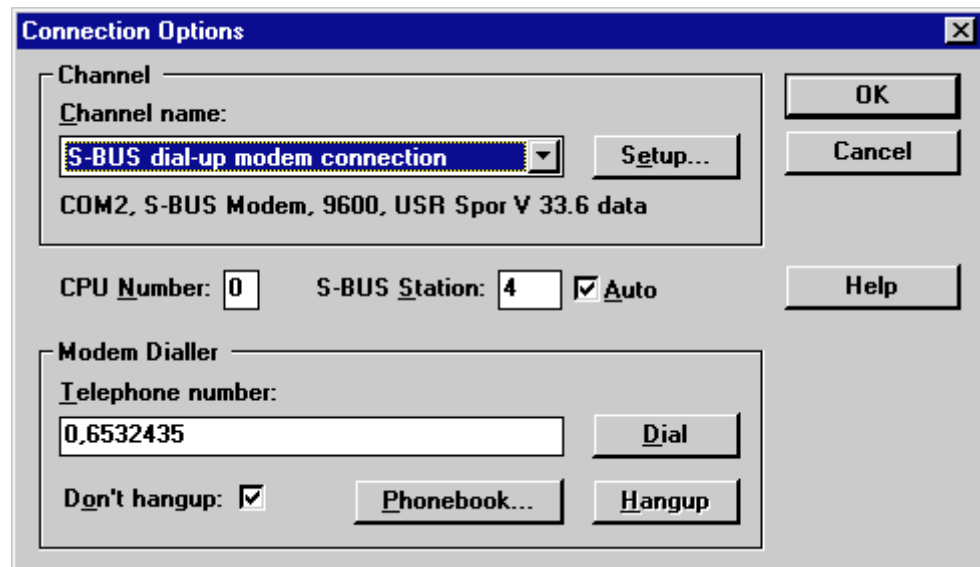
Uploads and displays the configuration of the connected PCD. This can then be saved in a file if required.

Download configuration:



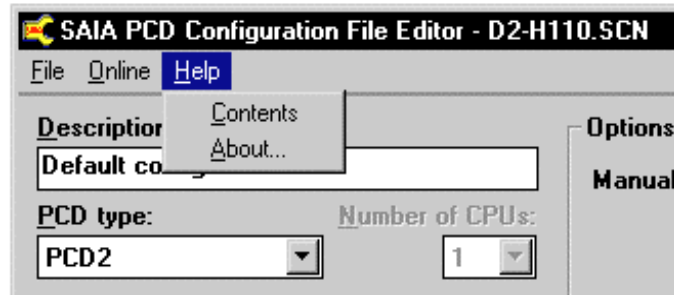
If 'Full configuration' is selected, the complete configuration is downloaded. If 'Selected data' is chosen (as shown), individual data can be selected and downloaded.

Connection options:

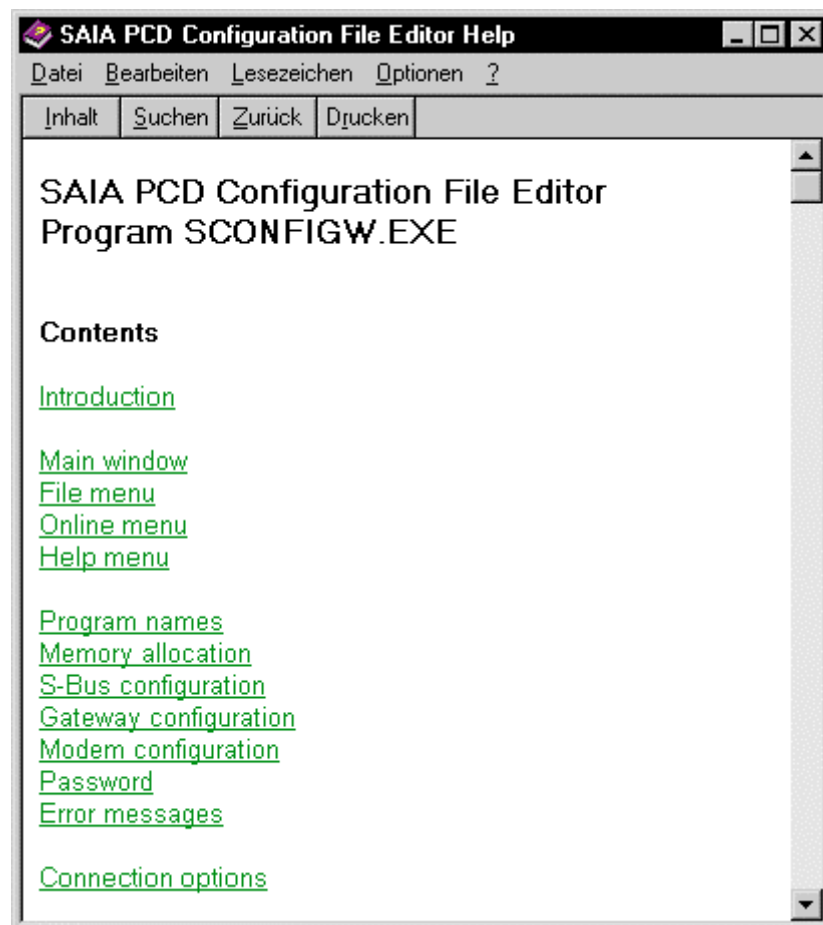


The online connection between programming unit (PC) and PCD is selected here, and can be configured with 'Setup...'

Help submenu



Contents: This displays the index of the interactive English language help. The help texts are not available in other languages, but you can consult the manual in the relevant language.



'About...:' Shows the current version and the licensed user's name.

The configuration file "D2-H110.SCN", used in the examples, looks like this:

```
[Pcd]
PcdType=2
Comment=Default configuration
NumberOfCpus=1
ManualMemAlloc=1
CodeTextMemSize=32
ExtenMemSize=0
CodeSizes=6,0,0,0,0,0,0
TextSizes=8,0,0,0,0,0,0
ExtenSizes=0,0,0,0,0,0,0
Password=0FAF369C
PasswordTimeout=5
ExtensionInit=0
EpromType=5
EpromSize=64

[SbusPgu]
SbusUsed=Yes
Station=0
PguPorts=0,-1,-1,-1,-1,-1,-1
PgulsRS485=No
BaudRate=7
Mode=0
TsDelay=0
TnDelay=0
Timeout=0
ModemUsed=No
ModemName=

[SbusGateway]
GatewayUsed=No
MasterPort=0
PortOnCpu=0
BaudRate=7
Mode=0
TsDelay=0
TnDelay=0
Timeout=0
BreakLength=4

[ProgramNames]
Cpu0=
Cpu1=
Cpu2=
Cpu3=
Cpu4=
Cpu5=
Cpu6=
```

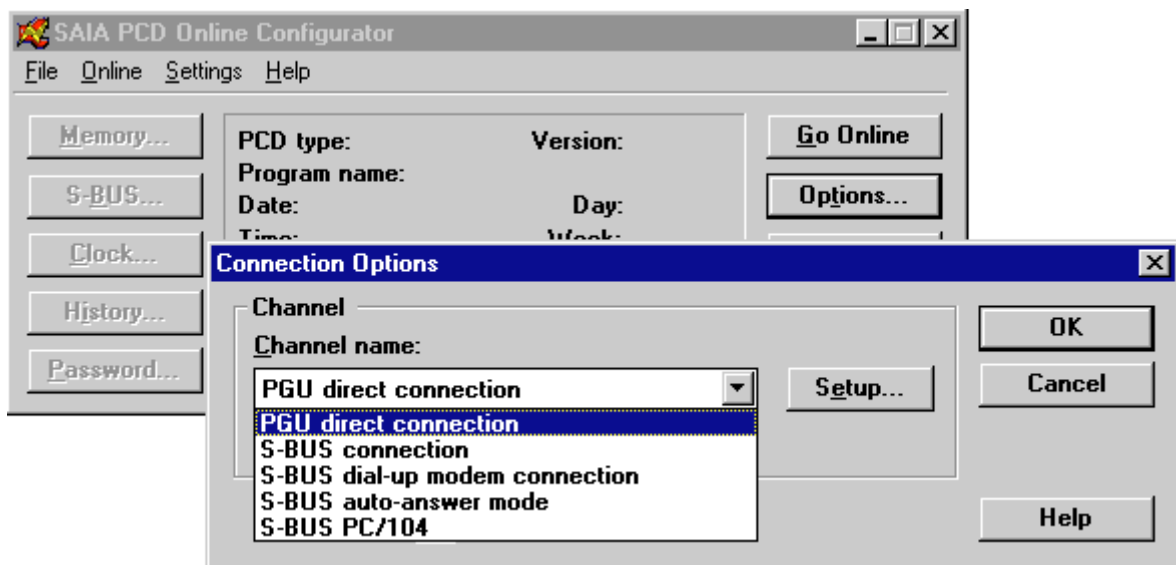
Online Configurator

The Onlineconfigurator has been changed in some points:

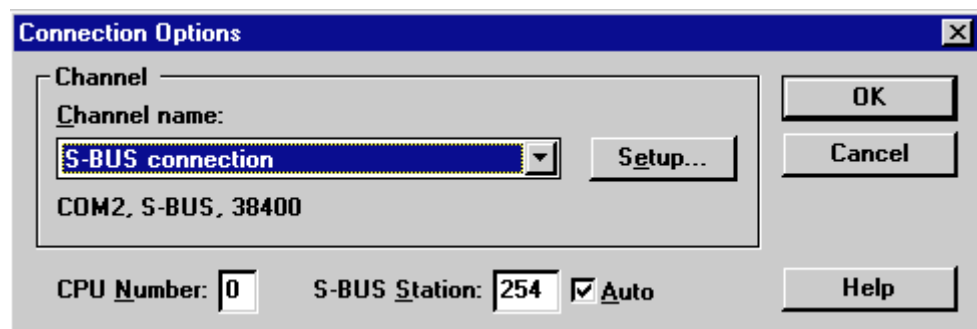
Connect Options

The menu 'Connect Options' contains two new possibilities to establish a communication between a PC and a PCD.

1. Now PG4 supports the PCD2.M220 (PC104). Using a high speed dual-port RAM.
2. The 'Auto Answer'-mode enables the PCD to telephone to the PC. The 'Auto Answer'-mode doesn't support the download off programs. Read the helpfiles for more information and the example code.

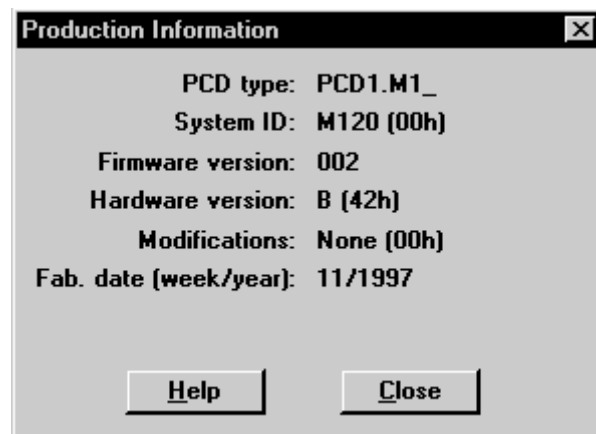
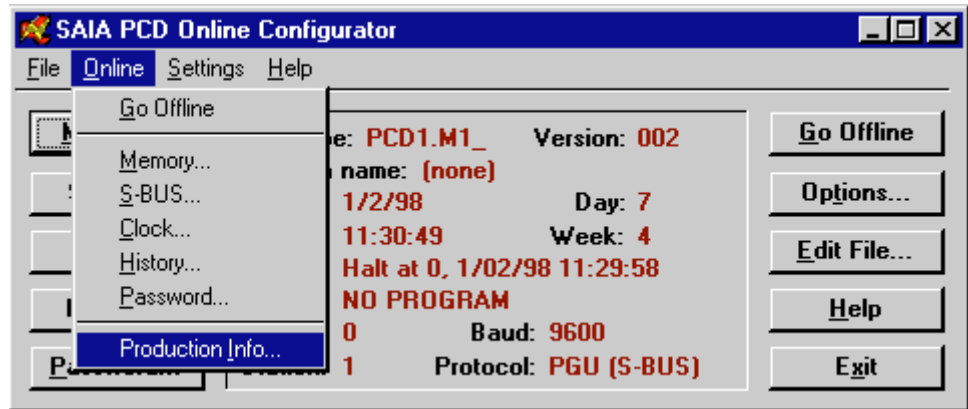


You can establish a S-Bus connection without knowing the station- number off the partnerstation. When the Checkbox 'Auto' is activated, the PG4 automatically looks out for the right address. PG 4 establishes the connection by himself. PG4 makes out by himself whether to use 'Parity', 'Break' or 'Data-Mode'.



Pruduction Info

The menupoint 'Production Info' opens a window containing important informations about the hardware-type and version:

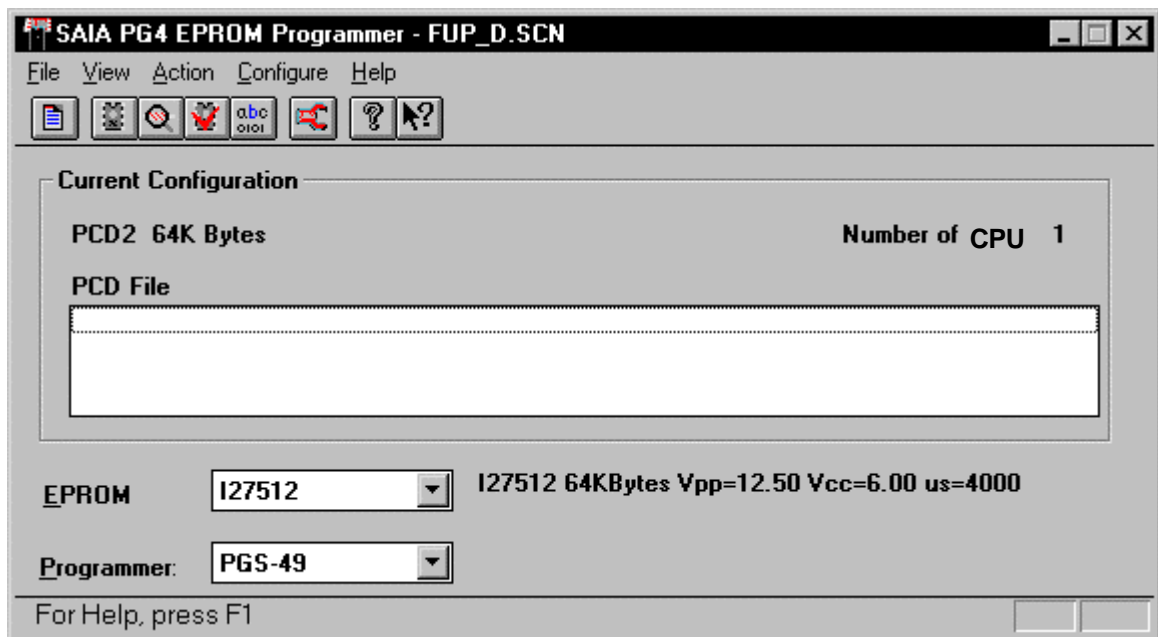


These informations are important for the SAIA Support-team. By telling these informations to the support agent, you can avoid a lot off misunderstandings.

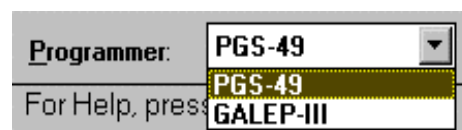
EPROM programmer

The PG4's EPROM programmer writes PCD user programs into EPROMs, or creates hexadecimal files from which EPROMs can be programmed using any EPROM programming device. The EPROM programmer is connected to the PC by a serial RS-232 port, or by a parallel port, depending on the model.

Main window



The following EPROM programmers are supported:



PGS 49: EPROM programmer used for many years with the SAIA[®] PCD. It is no longer available, but, since many of these devices are still in use it continues to be supported. It is connected to the serial port.

GALEP-III: New programmer, recommended by SAIA-Burgess Electronics and supported by the PG4. Connected to a parallel port.

Menus

File submenu



View submenu

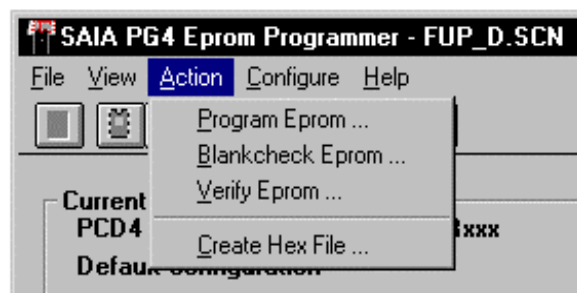


Toolbar: Shows or hides the toolbar

Status Bar: Shows or hides the status bar

Report: Active only when using the PGS 49 and when creating hexadecimal files. The GALEP-III EPROM programmer does not create a report.

Action submenu



Program EPROM... : Initiates programming of the EPROM or EPROMs when using the PGS-49 or GALEP III programmer.

Blankcheck EPROM... :

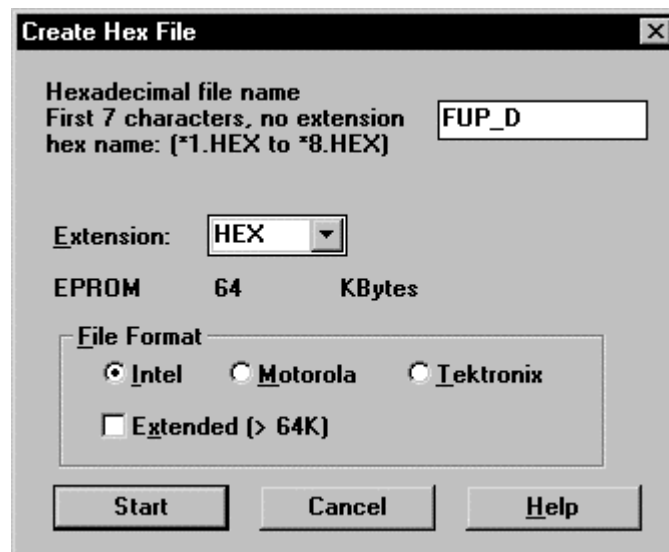
Initiates the blankcheck (to check that the EPROM is blank) when using the PGS-49 or GALEP III programmer.

Verify EPROM... :

Compares the data in EPROMs with a user program when using the PGS-49 or GALEP III programmer.

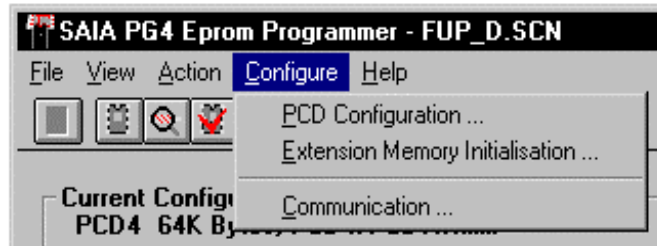
Create Hex File...:

Creation of hex files. The following window is shown:

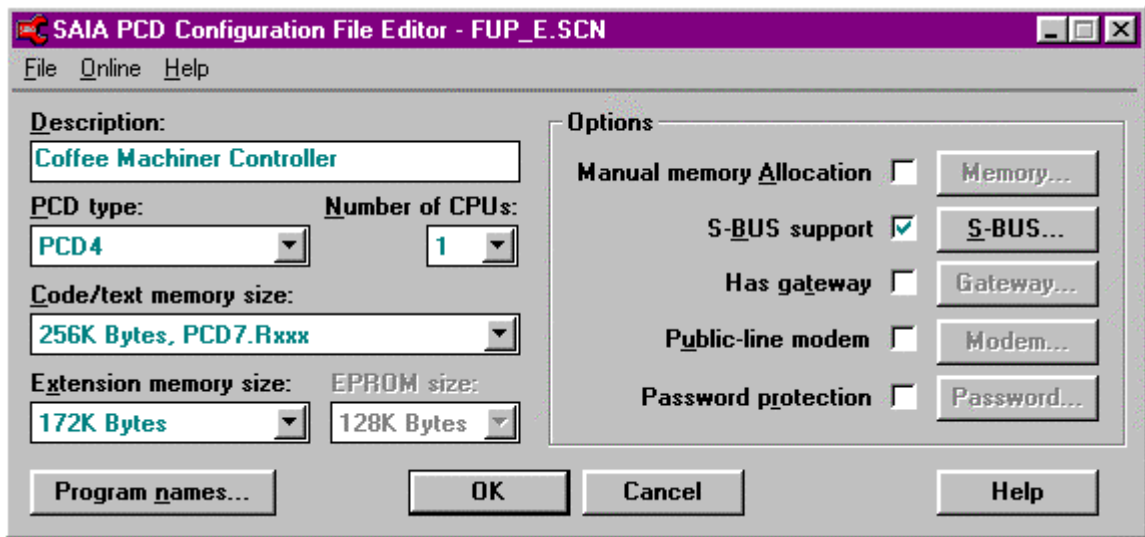


For further information, consult 'Help' and the EPROM programmer documentation.

Configure submenu



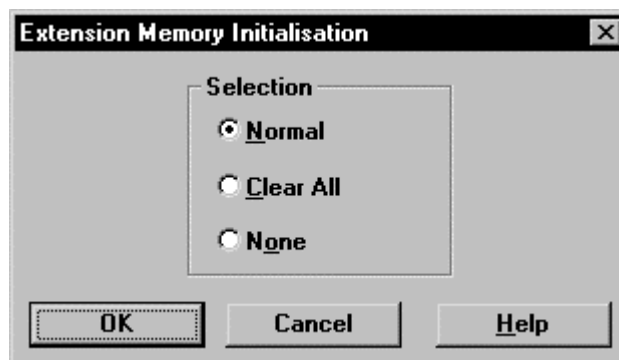
PCD Configuration... : Loads the configuration file editor:



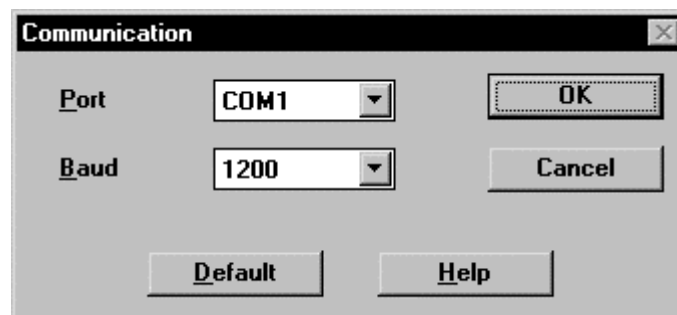
Refer to the Configuration File Editor's description.

Extension Memory Initialisation... :

Active only when extension memory has been selected under 'PCD Configuration...'. Defines how extension memory is initialized using this window:



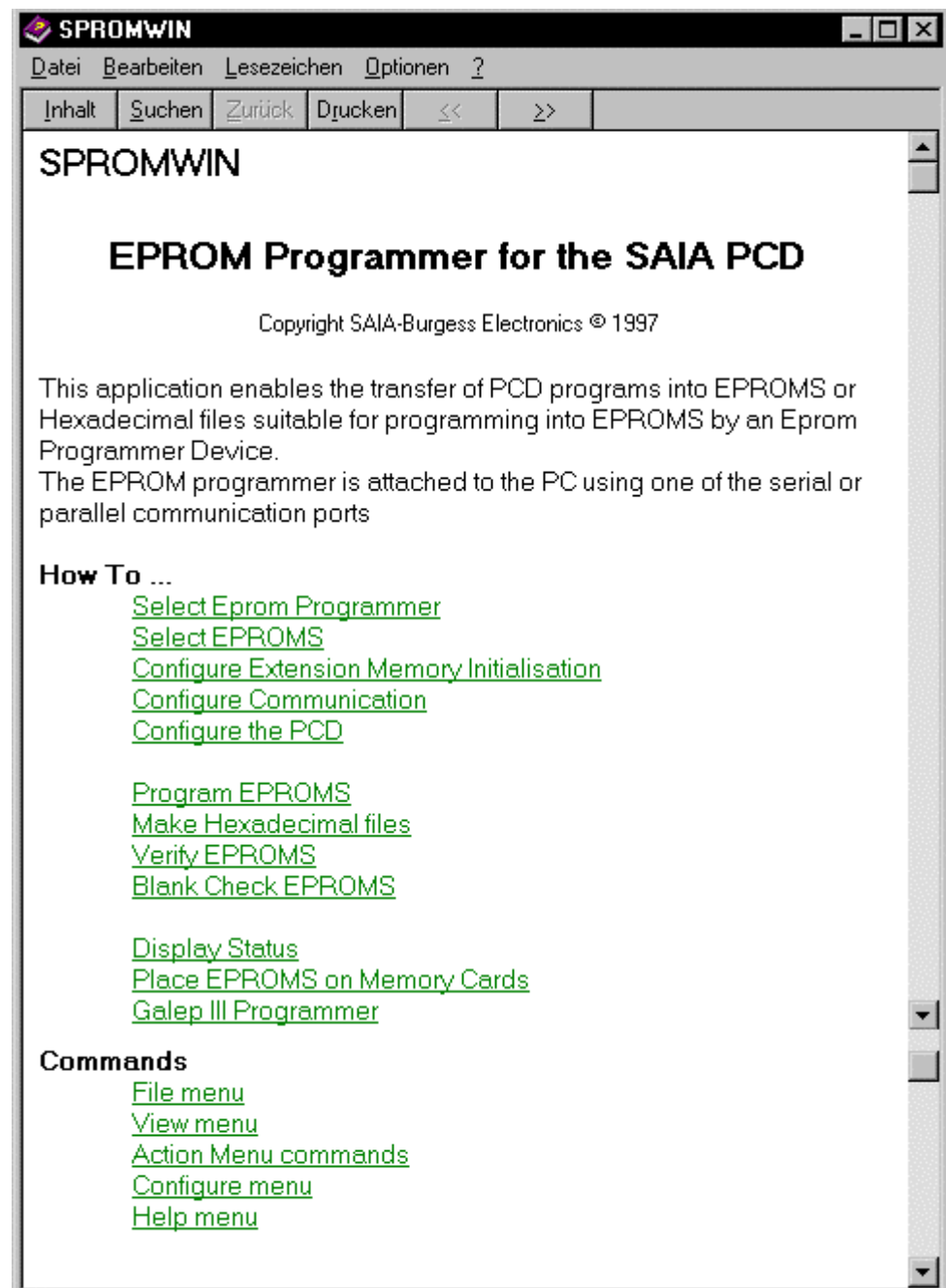
Communication... : Active only when the PGS 49 is selected. It defines the serial connection to the PGS 49. This window is displayed:



Help submenu



Index: The list of topics in the interactive help file (in English) is displayed.

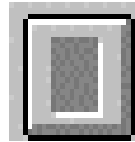
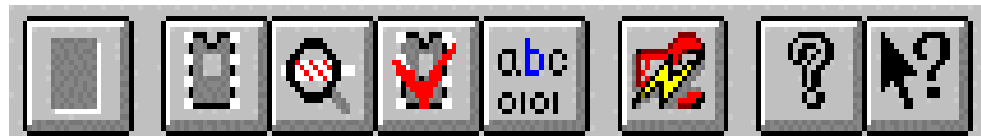


Using Help: Displays the standard WINDOWS description for using help files in the installed WINDOWS language version.

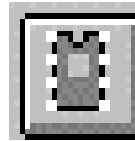
About Spromwin... : Displays the current version and licensed user's name:



The toolbar and short-cut keys



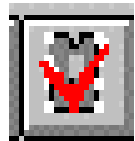
Displays the report
same as 'View' - 'Report...'



Programs an EPROM using the selected configuration
same as 'Action' - 'Program EPROM...'



Checks whether EPROMs are blank
same as 'Action' - 'Blankcheck EEPROM...'



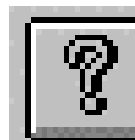
Compares the data in EPROMs with the user program and configuration
same as 'Action' - 'Verify EPROM...'



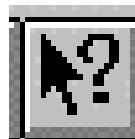
Creates hex files
same as 'Action' - 'Create Hex File...'



Configures the PCD
Same as 'Configure' - 'PCD Configuration...'

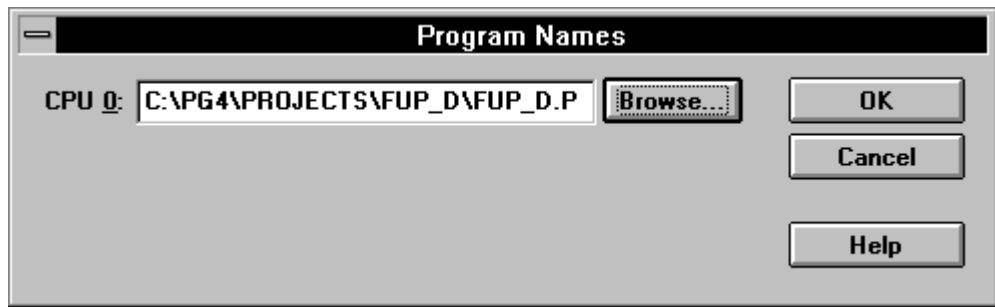


Displays current version and user's name
same as 'Help' - 'About Spromwin...'

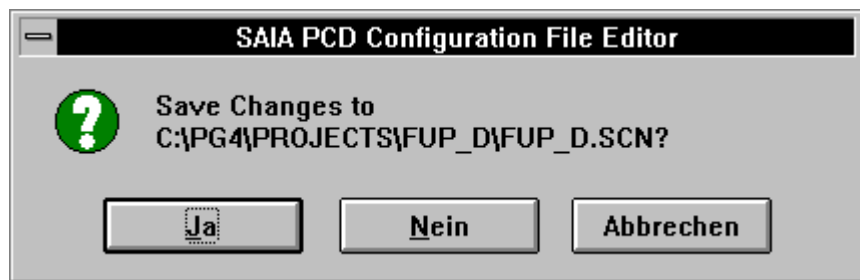


For help on a specific item.

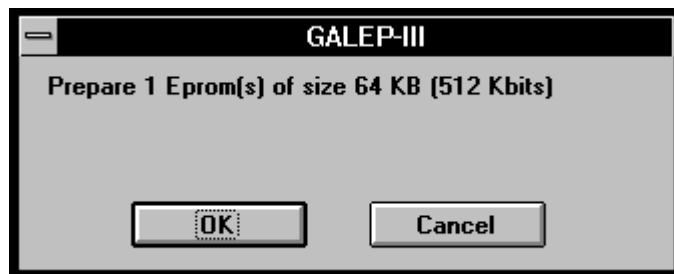
The program name and path should be shown:



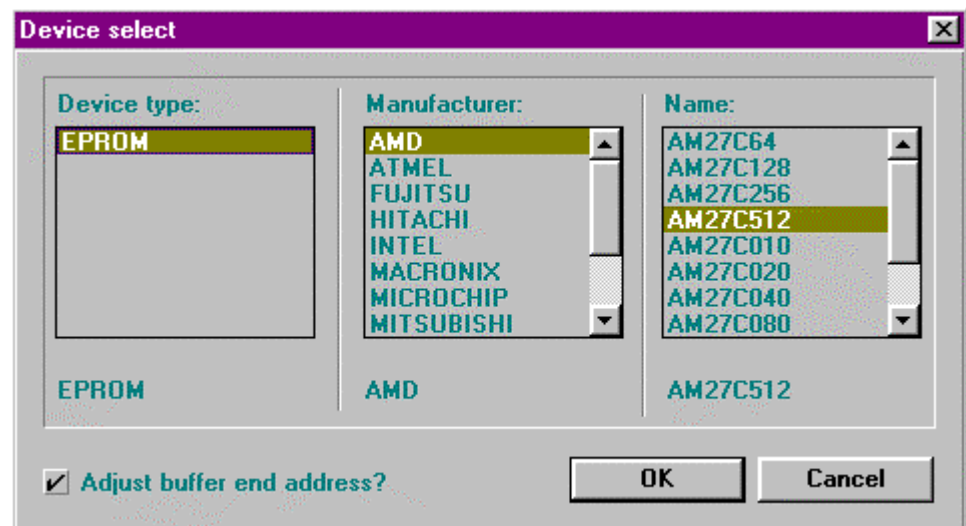
- The configuration is terminated and saved with 'OK'.



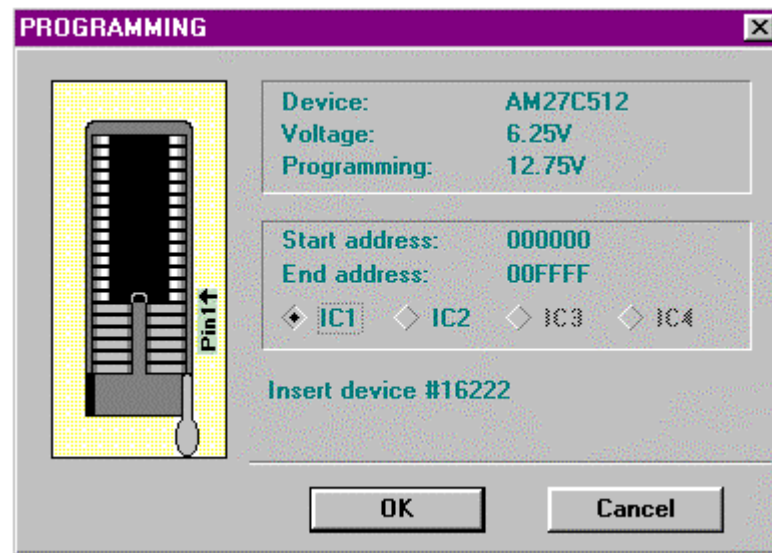
- The EPROM can now be programmed. From the 'Action' menu, select 'Program EPROM...' or press the EPROM button on the tool bar. The following window is displayed:



- Press 'OK' to display the 'Device select' window:



- The make of EPROM to be programmed is "AMD" and its type is "AM27C512". This type is preselected in the table, but can be changed to any other 512 K bit EPROM.
- After 'OK' the following box is displayed.

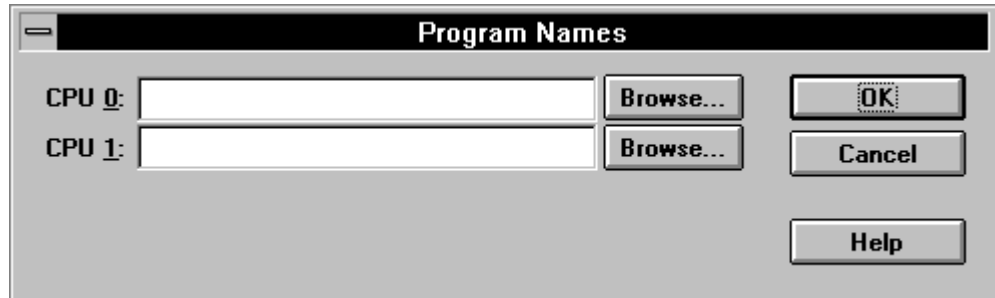


- The blank EPROM should be inserted as shown in the diagram, and the socket locked. Programming commences when 'OK' is pressed. "Programming..." is displayed and 'Start address' shows the address being programmed. When complete, "Comparing..." is displayed and the EPROM data is compared with the files.
- If another EPROM of the same type is to be programmed with the same program, insert the new EPROM and press 'OK'.
- To exit, press 'Cancel' or Shift+Enter.
- The programmed EPROM can now be inserted into the PCD2's user program socket. Jumper 'J2' should be put in position 'E' (for EPROM).

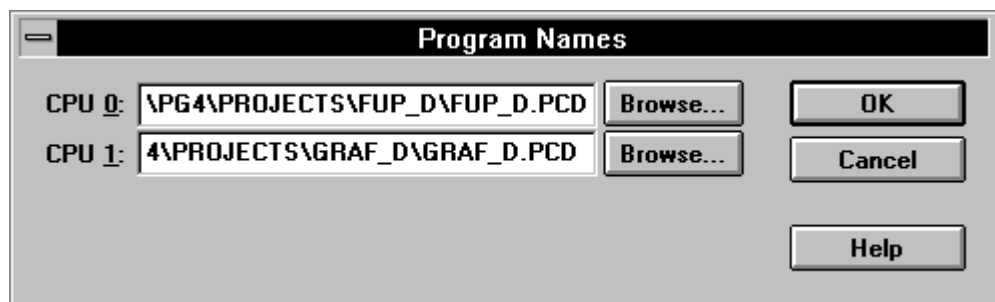
Practical application example 2

EPROMs are to be programmed for a PCD4.M445 (dual CPU). The program for CPU 0 is to be "fup_d" and for CPU 1 "graf_d". The "GALEP-III" EPROM programmer will be used.

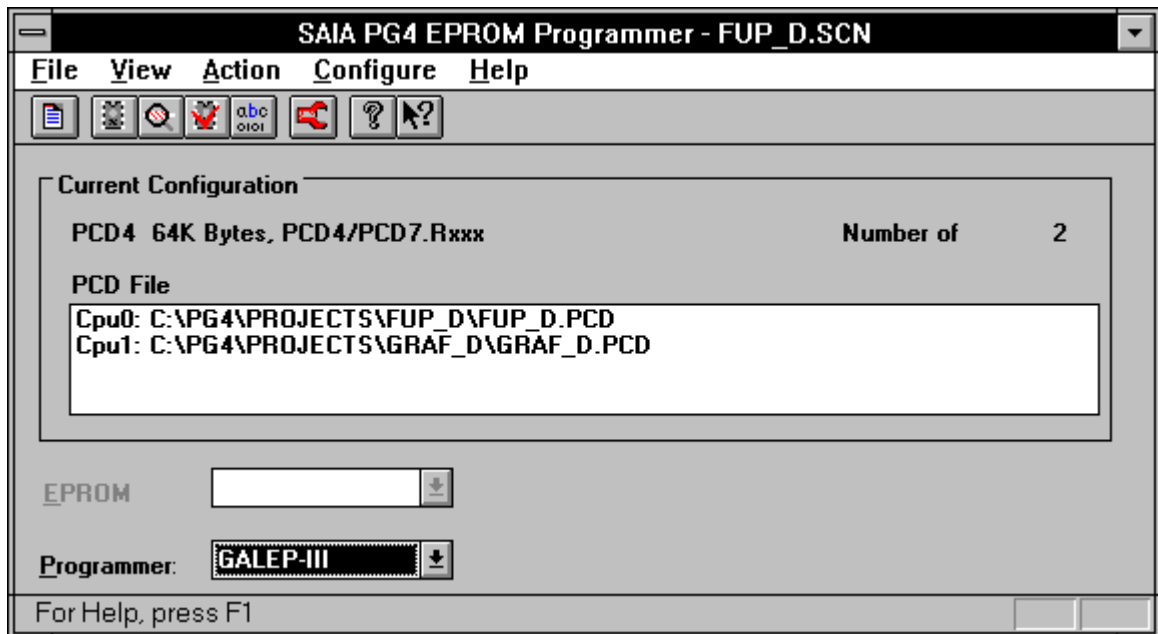
- For each of the two projects, execute a 'Make' from the "SAIA Project Manager". A loadable program, "fup_d.pcd", must be generated for CPU 0 and another loadable program, "graf_d.pcd" for CPU 1. The 'Download-Option' must first be switched off in the 'Project' - 'Make Options...' menu.
- From project "fup_d" call the "SAIA PG4 EPROM Programmer".
- Select the 'GALEP-III' programmer.
- From the 'Configure' menu, select 'PCD Configuration...'.
 - Select PCD type: 'PCD4'.
 - At 'Number of CPUs:', select '2'.
 - At Code/text memory size: select '64K Bytes, PCD4/PCD7/Rxxx' as the smallest EPROM memory.
 - Extension memory size is left as 'None'.
 - 'Options' are not used for this project.
- Indicate the names of both user programs. Click on the 'Program names...' button.



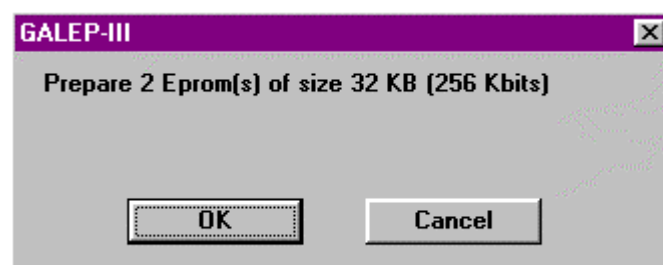
- Use 'Browse...' to find the project "fup_d" for CPU 0, then 'Browse...' again to find "graf_d" for CPU 1. Save the filenames with 'OK'.



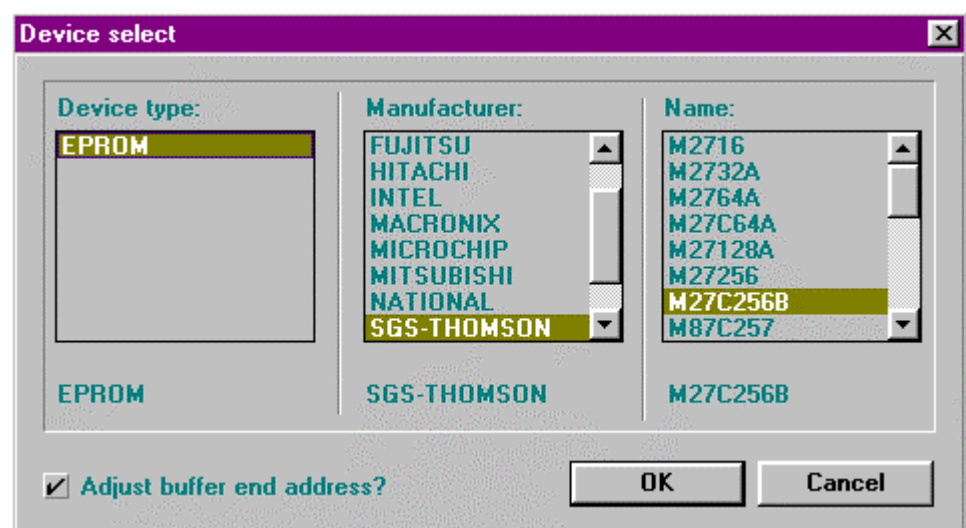
- Save the configuration by pressing 'OK' on the configurator's main window.
- The EPROM Programmer window should now appear as follows:



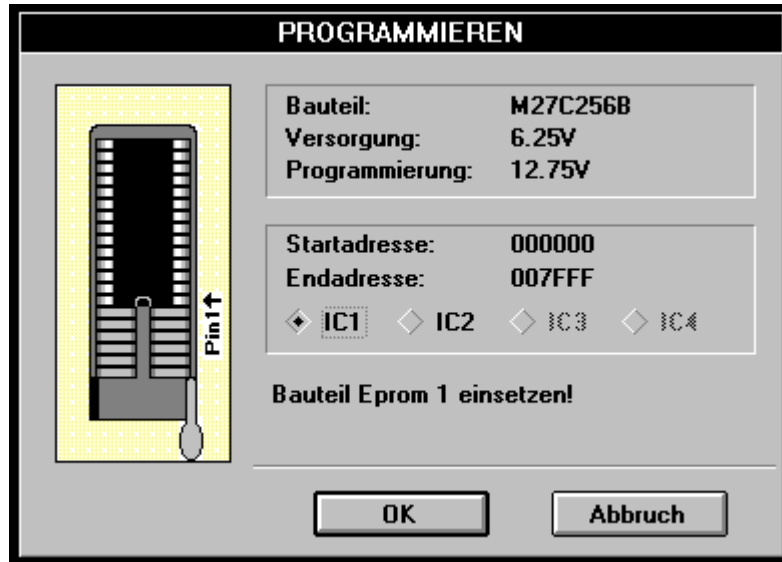
- The EPROMs can now be programmed. From the 'Action' menu, select 'Program EPROM...' or click on the EPROM button on the tool bar. The following window is displayed:



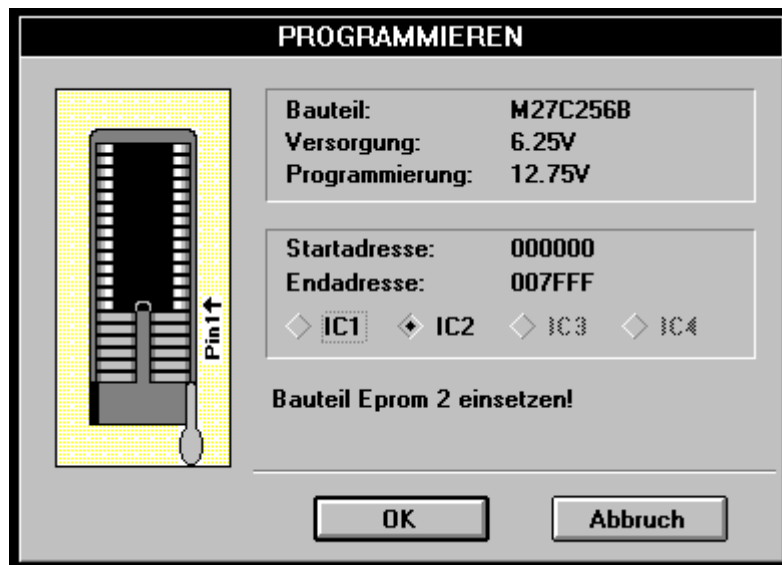
- Press 'OK' to display the 'Device select' window:



- The make of EPROM to be programmed is "SGS-THOMSON" and its type is "M27C256B". This type is preselected in the table, but can be changed to any other 256 K bit EPROM.
- Press 'OK' to display the 'Programming' window:



- Press 'OK' to program the first EPROM. When complete, the window looks like this:



- Insert the second EPROM and press 'OK'.
- The message: "Component programmed" appears.
- Both EPROMs can now be inserted in the public memory module of the PCD4, which is slotted into a PCD4.M445 CPU and the process put into 'Run'.
- If for any reason only one of the two EPROMs should be burned, it is possible to click directly on 'IC1' or 'IC2'.

Important: The PCD4 and PCD6 both require 2 EPROM or 2 RAM memory chips, even for the smallest user program and only a single CPU. This is because for each program line the lower 16 bits are stored in one memory chip and the upper 16 bits in the other.

It is **not** the case that the program of CPU 0 is located in one chip while that of CPU 1 is located in the other, as might be assumed from these examples.

Notes :

From :

Company :

Department :

Name :

Address :

Tel. :

Date :

Send back to :

SAIA-Burgess Electronics Ltd.

Bahnhofstrasse 18

CH-3280 Murten (Switzerland)

<http://www.saia-burgess.com>

BA : Electronic Controllers

Programming tools for MS WINDOW

PG4 - Version 1.3 and 1.4

If you have any suggestions concerning the SAIA[®] PCD, or have found any errors in this manual, brief details would be appreciated.

Your suggestions :