

SAIA PCA-ASSEMBLER

Version 1.0

Provisional Manual

<u>Contents</u>

A) Summary

B) Importants MS-DOS Commands

C) Software installation on a Hard Disk

D) Software installation on a system with 2 Flopy Diskettes

- E) Using directories on a Hard Disk
- F) Using the menues (with Error Messages)
- G) Connection of the PCA to the IBM PC

H) Listing of Help Screen and Sub-Menues

Nov 21, 1988



- HARRISTER

10

SAIAPLC Programmable controllers

A) SUMMARY

This Provisional Manual contains installation details, and notes on using the new backage.

This first release of the new SAIA FCA ASSEMBLER package has:

- Easy to use menus. batch files are not needed anymore.
- Help screens.
- New Uploader and Downloader with verification.
- New Eprom Programming utility.
- New ONLINE debug program.
- New utility programs, P10 Simulator, PCLIST Program Lister etc.
- New improved versions of all the original programs.
- Now runs on IBM PS/2 systems.

Hardware requirements:

- IBM PC/XT/AT or compatible, or an IBM PS/2.
- Minimum 512K of memory.
- Hard disk, or two diskettes of 360K.
- 2 parallel printer ports (one for the printer, and one for the PCA). I parallel port can be used.
- Any monochrome or colour monitor (with MCGA, CGA, EGA, VGA or Hercules card).
- MS-DOS Version 3.0 or above.
- SAIA PCA2.K43 cable for connection to the PCA.

You must also have an Editor program, such as IBMs Personal Editor (recommended), or WordStar etc.

The PCA ASSEMBLER is available as three separate packages:

- PCASS 1 ONLINE	The base package, which contains debug tools. With this package alone only small programs can be developed.
- PCASS 2 PROGRAMMING	The main backage, which includes PCASS 1, and contains the Assembler and Linker. This package allows the development of large programs, with the comfort of using mnemonics and symbolic names instead of numbers for instructions, and includes documentation aids.
- PCASS 3 TRANSFER	Provides PCA computications and Encom Pro-

PCASS 3 TRANSFER Provides PCA communications and Eprom Programming utilities.

Package PCASS 1 - ONLINE

Supplied on 1 × 360K diskette:

Contents:

README	This tile.
FOUINI.CUM	initializes the barafiel interface to the run.
PLA.EXE	menu program.
PCAHLP.0xx	Menu helo files
PCUPLD.EXE	Uploads programs from the PCA.
PCDNLD.EXE	Downloads programs into the PEA.
ONLINE.EXE	Online aebug, for RUN mode.
CI.EXE	Comms Interface, debug when not in RUN mode.
PCXRF.EXE	Creates cross reference listings of orograms.
PCFILE.EXE	File mandling utilities, copy, erase etc.
PCLIST.EXE	Creates a listing file from the PCAs memory.
P10.EXE	Simulates the P10 on your (BM PC.

Package PCASS 2 - PROGRAMMING

Supplied on 2 x 360K diskettes: - Disk 1 PCASS 2-1 ONLINE - Disk 2 PCASS 2-2 ASSEMBLER

Contents of Disk 1 (same as PCASS 1):

README	This file.
PGUINI.COM	Initializes the parallel interface to the PCA.
PCA.EXE	Menu program.
PCAHLP. Úxx	Menu helo files
PCUPLD.EXE	Uploads programs from the PCA.
PCDNLD.EXE	Downloads programs into the FCA.
ONLINE.EXE	Online depud, for RUN mode.
CI.EXE	Comms Interface, debug when not in RUN mode.
PCXRF.EXE	Creates cross reference listings of programs.
PCFILE.EXE	File handling utilities, copy, erase etc.
PCLIST.EXE	Creates a listing file from the PCAs memory.
P10.EXE	Simulates the P10 on your IBM PC.

Contents of Disk 2:

PCKEY.PWD	Your registered users identification.
PCAS.EXE	PCA Macro Assembler.
PCAS.OOx	PCAS overlay files.
PCLINK.EXE	PCA Linker.
PCLINK.00x	PCLINK overlay files.
PCTEXT.EXE	Text Assembler.
PCDIS.EXE	Disassembler.
PCCOMP.EXE	Program compare utility.
PCFLOW.EXE	Flow chart generator.
××××.SRC	Source program examples.

Package PCASS 3 - FRANSFER

Supplied on 1 x 360K diskette.

Contents:

	README	This file.
	PCPROM.EXE	Eorom Programming Utility.
	ERASE.OBJ	File or FFFFs for "erasing" RAM memory.
	RTA.COM	Runtime Analysis.
ę	PCCONVST.EXE PCCONV.EXE	Converts OBJ file into Standard Intel Hex format. Converts OBJ file into Intel MCS-86 Hex format.

B) IMPORTANTS MS-DOS COMMANDS

To install the backade on an IBM PC/XT/AT (or PS/2), a minimum knowledge of DOS (PC-DOS or MS-DOS) is required. You need to know about directories (if you have a hard disk), now to format diskettes and copy files, and about Batch files.

This information is easily obtained from your DOS Reference Manual. If you don't already know DOS, read about the following items in your DOS manual before installing the package:

USING TREE STRUCTURED DIRECTORIES

DOS COMMANDS:

CHDIR	(or CD - change directory)
COPY	(copies files from one disk or directory to another)
FORMAT	(initializes a diskette before vou can use it)
MKDIR	(or MD - creates a new directory on your disk)
RMDIR	(or RD - deletes an empty directory from your disk)

BATCH FILES (files containing several DOS commands, saves typing)

AUTOEXEC.BAT (Start-up batch file)

Once the backage is correctly installed, almost no knowledge of DOS is required to use it.

C) INSTALLATION ON A HARD DISK

This is a fairly simple procedure. Essentually, copy all the files on the supplied PCA diskettes into a special PCA directory, then set the PATH so that DOS can find them.

Follow these steps:

1) Create an emoty directory to hold the backage. Preferably name it "NPCA", but you can use any other name. such as "NNEWPCA". In the following examples, the directory "C:NPCA" is assumed. Create the directory using the command:

MD PCA

2) Copy ALL the PCA Assembler backage files from all the diskettes you have received into this directory, by inserting each diskette in turn into drive A, then using the command:

COPY A: *. * C:\PCA

5) Edit the following lines into a batch file in the top level directory (perhaps into the AUTOEXEC.BAT file, see Step 4). Use the path name of the directory created in Step 1 in place of C:\PCA in this example:

APPEND C: YPCA PATH C: YPCA: C: YDDS SET PCA=C: YPCA

The PATH command defines the directories in which the PCA Menu program will search when trying to locate programs. The PATH command above is an example, you should provide a list of all the directories that contain programs you wish to execute, separated by semi-colons ":". Refer to your DOS Reference Manual.

When you run this batch file, it will configure the environment of your system for using the PCA ASSEMBLER package. Programmers often have several batch files in the top level directory, for configuring their systems for several different uses. For example TC.BAT to set up for using Turbo C. TP.BAT to use Turbo Pascal, PCA.BAT for using the PCA ASSEMBLER etc.

4)

Edit the systems AUTOEXEC.BAT file to contain the following line:

C:\PCA\PGUINI

This initializes the parallel port used to communicate with the PCA. If not initialized, and a PCA which is in RUN mode is connected to the IBM PC, the PCAs program execution will be paused.

The lines in Step 3 may also be placed in the AUTOEXEC.BAT file, to set up your system for using the PCA ASSEMBLER package when your computer is switched on (see Step 7).

5) Edit CONFIG.SYS to contain the following line:

DEVICE=ANSI.SYS

This allows support for the standard ANSI escape seduences which are used for display control (see your DOS manual).

- NOTE: Certain IBM compatibles have a program called ANSI.EXE (or ANSI.COM) instead of ANSI.SYS. This prodram should be executed from the AUTOEXEC.BAT file. instead of placing the above line in CONFIG.SYS.
- 6) Reboot your system by typing Ctrl-Alt-Del. to execute the new CONFIG.SYS and AUTOEXEC.BAT files.
- 7) Now read the Section E) USING DIRECTORIES ON A HARD DISK below. then create a work directory, move into it (using CD), and type the following command to enter the PCA menus:

PCA

Alternatively, the batch file created in Step 3 could move you into your working directory, and automatically invoke the PCA ASSEMBLER menus. For example, if a batch file PCA.BAT contains: $r_i^2 \mathcal{P} \in N\mathcal{D} \subset \mathbb{N} \cong \mathcal{A}$ PATH C:\PCA:C:\DOS SET PCA=C:\PCA CD %1 C:\PCA\PCA

You can type a command like following command. Which will move you into your PROJECT1 work directory, and invoke the PCA Menus:

C:\>PCA PROJECT1

The "%1" parameter in the batch file is the name of the work directory you are using for your project.

8) When on the main menu, select the "Setup" menu by pressing "S", and enter the name of your Editor program (available only if you have PCASS 2), and your printers page size.

The Editor is invoked supplying the name of the file you wish to edit, for example if your Editor is called PE, and you edit the file FRED.SRC, the menu executes the command "PE FRED.SRC". If this is not suitable for your Editor, the Editor can be invoked using a batch file to supply any additional information, enter the name of the batch file as the Editor name.

9) Read each Help screen. Help screens are displayed by selecting the option on the main menu, then pressing F1.

10) You are now ready to start work.

D) INSTALLATION ON A SYSTEM WITH 2 FLOPPY DISKETTES

Since the usual 2-diskette system has diskettes which can contain only 360K of data, which is not enough to contain all the programs, it is necessary to separate the backage into several diskettes. These diskettes must be swapped in and out of Drive A, depending on which programs you want to execute.

Drive A should be used to hold the DOS operating system. certain DOS programs, and some SAIA ASSEMBLER backage programs. Drive B should hold your work disk, containing your own files (SRC, LNK, OBJ etc).

You should stay logged onto Drive A all the time, this means that you must use the "B:" disk drive name in every file name entry field on the menus, so that your work files can be found.

It is recommended that you create 3 diskettes for use in Drive A. Each diskette must be formatted to contain the operating system using the "File handling" option "Format", or the DOS command "FORMAT B:/S" to initialize the diskettes before copying the files onto them.

It is not recommended to use directories when working on systems witch only 360K floppy diskettes.

The lists below show the suggested diskette contents, although the contents of each diskette depends on which programs you use most often.

If the files listed below will not fit onto a single diskette, move one or two files onto one of the other diskettes. Remember that all the PCAS.xxx files must be on the same diskette, and the PCLINK.xxx files must also be on the same diskette. If you need them, the Help files PCAHLP.xxx should be on each of the diskettes, but are not required to run the programs. It is also use+ul to have PCFILE.EXE on each diskette.

If you get a "PROGRAM NOT AVAILABLE" error when you try to execute a program from the menus, it means that you must enter the diskette containing the desired program into Drive A.

Disk 1: Startup Disk, in Drive A at the start.

COMMAND.COM	
AUTOEXEC.BAT	(see below)
CONFIG.SYS	(see below)
ANSI.SYS	(or ANSI.EXE on some systems)
KEYBxx.COM	(your keyboard definition program, eq. KEYBGR.COM)
PCA.EXE	
PCKEY.PWD	(if you have the PCASS2 or 3 package)

Disk 2: Editor, Assempler, Linker and Downloader, in Drive A to Edit, Assemble, Link and Download your program. COMMAND.COM PRINT.COM FORMAT.COM PGUINI.COM PCAHLP.001 - PCAHLP.018 PCAS.EXE PCAS.001 - PCAS.004 PCLINK.EXE PCLINK.001 - PCLINK.003 PCDNLD.EXE PCFILE.EXE PE.EXE 3 These are your editor files, PE in this example. PE.PRO 3 PE.HLP 3 Disk 3: Debug. Downloader. Uploader and Utilities. in Drive A while you are testing your program. COMMAND.COM PRINT.COM FORMAT.COM PCAHLP.001 - PCAHLP.018 PCFILE.EXE ONLINE.EXE CI.EXE PCUPLD.EXE PCDNLD.EXE PCXRF.EXE PCDIS.EXE PCCOMP.EXE PCFLOW.EXE RTA.COM PCCONV.EXE PCCONVST.EXE PCPROM.EXE The AUTOEXEC.BAT file should contain the following lines: PGUINI PATH A:\:B:\ SET PCA=A:\ PCA CONFIG.SYS must contain this line (but see note in hard disk installation Step 5 above): DEVICE=ANSI.SYS Start up your system using Disk 1 in drive A, and your work disk in Drive B. The menus will be entered directly (AUTOEXEC.BAT does this for you), then insert Disk 2 or 3 into Drive A. With Disk 2, the Edit, Assemble, Link, Download and File nandling commands may be done. To use the other commands, insert the relevant diskette into Drive A before executing the command from the menu.



E) USING DIRECTORIES UN A HARD DISK

When correctly installed, and the FATH and "SET PCA=directory" commands nave been executed from the batch files created in the installation Steps 3 or 4, the PCA backage can be run from ANY directory.

It is recommended that you use a different directory for each project or large program. NEVER work in the SAIA PCA programs directory, unless you want a directory containing a tangled mixture of SAIA programs and your own SRC. LNK, OBJ files etc. ugn!

Each work directory must contain the following files. In order to run the Assembler and Linker (for PCASS 2 owners only). and may also need some files for your Editor:

PCAS.001 - PCAS.004	Assembier overlav files.
PCLINK.001-PCLINK.003	Linker overlay files.
PCKEY.PWD	Your user identification file.
PE.PRO (or PE2.PRO)	Editor Key profile) If using PE.
PE.HLP (or PE2.HLP)	Editor help file)

The original cooles of these files must still reside in the SAIA FCA programs directory, so that they can be copied easily into any work directory. You can write a batch file to do this, which you execute whenever you move into a new work directory.

Here's an example, this batch file should reside in the PCA programs directory, it is assumed that the SAIA PCA programs are in directory "C:\PCA". To use this batch file, first move into the new work direct-ory using the CD command, then type "MAKEWORK $\langle CR \rangle$ ".

Batch file MAKEWORK.BAT:

COPY C:\PCA\PCLINK.00? COPY C:\PCA\PCAS.00? COPY C:\PCA\PCKEY.PWD COPY C:\PCA\PE.PRO COPY C:\PCA\PE.HLP

If you have DOS V3.3 or above, you can use the APPEND command to temporarily merge the work directory with the PCA directory, it is then not necessary to copy the files into each directory.

You cannot refer to other directories from the SAIA menus, because some of the SAIA programs do not recognize directory names, only disk drive names. In order to reference other directories, you can use the SUBST command to substitute a drive name for a directory name. The drive name can then be used from the menus in place of the directory name. Drives D and E (and above) can be used. Drive B is NOT supported on a system with only one floopy disk Drive A, due to a nasty MS-DOS feature which automatically re-assigns Drive A to be Drive B if you reference Drive B and it doesn't physically exist. If you reference Drive B from the menus, and it doesn't physically exist, an "Invalid file name" error will occur, you can also substitute drives F = Z for directories, providing the DOS command LASTDRIVE=n is in your CONFIG.SYS file, see your DOS Reference Manual. This allows you to reference drives up to "n", the default last drive is E.

WARNING: When substitution is in effect, you must NOT use any of the following commands to reference a SUBSTituted drive: ASSIGN, BACKUP, DISKCOMP, DISKCOPY, FDISK, FORMAT, JOIN, LABEL or RESTORE.

K,

F) USING THE MENUS

To run the menu program. enter the command "PCA" at the DOS prompt. First you will see the SAIA Logo and Copyright notice, pressing any key will display the main menu. If you press another key immediately after entering the PCA command (eg. PCA(CR)(CR)), the Logo is not displayed.

The menus can now be used to invoke the various SAIA PCA programs. Before you begin working, if you have package PCASS 2, you should enter the name of your Editor (eg. PE or WS) and your printers page size in the "Setup" menu.

Once you have finished work, the "Quit" command will save all the information you have entered into the menus, and return you to DOS. The menu information is saved in a file named PCAMENU.DAT. In the current directory. The next time you run the PCA menu program from this directory, the PCAMENU.DAT file is automatically loaded, restoring the information in the menus exactly as it was when you last used "Quit".

Three types of input fields are used on the menus. When the cursor is on a field, the entire field is always displayed in reverse video, indicating it is the selected field. Movement from one field to another is by using the arrow, tab or space keys.

1) Select fields

These appear on the main menu. They allow selection of a sub-menu, program or operation, each field being a description of the process. The selected field is highlighted in reverse video. To select another field, the space bar can be used, as well as the tab or arrow keys. Pressing <CR> (ENTER) displays the selected sub-menu or executes the program or operation. Alternatively, each description contains one capital letter (the command letter), displayed in bold video. This single command letter key can be pressed to select the operation directly.

2) Entry fields

Numeric or text data, such as an address or file name can be input into these fields. Only valid Keys are accepted, invalid key depressions are beeped.

Full editing facilities are provided, using the ins. Jel. Dackesoace. left and right arrow. Home and End keys.

For file name +reids, only valid MS-DOS file name characters are accepted. NOTE: Only one "." character is allowed in a file name, if one already exists in the field, another cannot be entered, the existing "." must first be deleted.

MS-DOS file names consist of an 8-character file name with an optional 3-character file type. If more than 8 or 3 characters are entered into the field, the extra characters are ignored. If a file name ends in ".", no default extension is appended.

Pressing the $\langle SPACE \rangle$ bar clears to the end of the entry field. Moving the cursor to the next field.

3) Switch field

This type of field is an option "switch", pressing the SPACE bar selects the switch position. Normally these fields have two positions "Yes-No", they may also be used to select an operation, such as on the "Up/download" menu.

Menu Help

From any position on the main menu, or from any sub-menu, function Key Fl may be pressed to display a Heip screen. Help screen texts are contained in files FCAHLP.001-PCAHLP.018, there is one file for each Help screen.



EDITOR MENU

Name of file to edit (.SRC)

Enter file name. <CR> executes. <ESC> aborts.

EDITOR MENU

EDITOR MENU HELP

Invokes the Editor program, supplying a file name. The default file type (or extension) is SRC, this is automatically appended to the entered name. To edit other types of files, the file type must be supplied, for example "FRED.TEX". To edit a file without a type end the file name with ".", for example "FRED.". An optional drive specification (A:, B: etc) may be given. When working with a hard disk, you can refer to any directory by SUBSTituting the drive name D: or E: for the directory (drive B: is NOT supported on systems with only one floppy disk drive A), see your DOS Reference Manual (SUBST command) for more details, some commands must be avoided when SUBST is in effect.

Examples: NAME File NAME.SRC on the current drive is edited. B:NAME File NAME.SRC on drive B: is edited. NAME.TEX File NAME.TEX on the current drive is edited. E:NAME.ABC File NAME.ABC on SUBSTituted drive E: is edited. The above file naming conventions are true for all file name entry fields on the PCA menus.

The name of your Editor program must be entered in the "Setup" menu.

Press any key to return to the menu

Menu Error Messages

When an error is detected, the reverse video cursor is placed over the entry field containing the error, if applicable.

ERROR 1: CAN'T OPEN FILE: <file name> The file does not exist in the current directory or on the specified disk drive.

ERROR 2: NO FILE NAME SPECIFIED A file name must be entered before the operation can be started.

ERROR 3: CANNOT PRINT PGD AND LST AT THE SAME TIME From the Assembler menu, you can only print either the documentation or the listing file, but not both at the same time. You must do the print as two separate operations.

ERROR 4: PROGRAM LOAD FAILURE ON: corogram name>
The program cannot be loaded and executed. This occurs if the
operating system file COMMAND.COM cannot be found, or if there
is not enough memory to load and execute the program.

- ERROR 5: WRITE ERROR ON FILE: <file name>
 - The disk(ette) is full.
 - The diskette (or destination file) is write protected.
 - No diskette in the drive.

ERROR 6: OUTPUT FILE HAS SAME NAME AS INPUT FILE When creating an intermediate LNK file or a PGD file using the Linker menu, the output file name used has the same name as an input file. The output file name must not be the same any of the input files.

- ERROR 7: COMMAND.COM NOT FOUND. OR NOT ENOUGH MEMORY Similar to Error 4, but caused because DOS (not the menus) cannot execute the command:
- ERROR 8: INVALID VALUE The value in the highlighted field is out of range.

ERROR 9: INVALID ADDRESS Addresses may be between 0 and 15999, according to your PCA type.

ERROR 10: INVALID FILE NAME: <file name>

- The highlighted file name is not a valid MS-DOS file name: - Device names PRN, LPT1, LPT2, LPT3, CON, AUX, COM1 and
 - COM2 cannot be used as input file names.
- CON, AUX, COM1 and COM2 cannot be used as output file names.
- You are trying to reference Drive B: and it doesn't physically exist.
- You cannot use the SUBST command to substitute Drive B: for a directory, you must use Drive D: or above.

ERROR 11: PROGRAM NOT AVAILABLE: sprooram name>

The indicated program does not exist (is not included in the package you have purchased). Or cannot be incated. The PATH command, usually executed from the AUTOEXEC or other patch file, must indicate the directory where the SAIA programs can be found, refer to the installation notes, or to your DOS Reference Manual (PATH command).

If this error occurs for the Edit command, check that the correct program name of your Editor has been entered in the "Setup" menu, and that the Editor can be found via the PATH.

ERROR 12: CANNOT PRINT RTA AND MOD AT THE SAME TIME On the "Runtim analysis" menu, you can send either the RTA file OR the MOD file to the printer, but not both at the same time.

6) CONNECTION OF THE PCA TO THE IBM PC

You should have a free parallel port on your IBM PC to connect to the PCAs PGU connector, via caple PCA2.K43.

The last available parallel port is always used (LPT1, 2 or 3). For example, if you have two parallel ports (LPT1 and LPT2), use LPT1 for the printer, and connect the PCA to LPT2.

If you have only one parallel port, you won't be able to use both the PCA and the printer at the same time.

*** WARNING ***

When the PCA is in RUN, it is important to initialize the barallel port of your IBM PC BEFORE you connect the PCA2.K43 cable to the PCA, by executing FGUINI (see Section C-4).

To prevent damage to the PCA, the cable or your IBM PC, take care of the following:

- The grounds of the PCA and the IBM PC must be connected to the SAME source. There must be no potential difference between the grounds.
- When connecting or disconnecting the cable, ensure that the pins in the connector do not touch the earthed sides of the socket if power is on.
- Do NOT connect the PCA2.K43 cable to a SERIAL PORT (a male to female adaptor is required to make this mistake).

SAIA PCA ASSEMBLER V1.0

MAIN MENU

*** SAIA AG. MURTEN, PRE-RELEASE VERSION FOR SAIA USE ONLY ***

Directory: C:\

25.11.88 16.40

Edit	Text assembler	Compare programs
Assemble	Disassembier	Xref listing
Link	floW chart	File handling
Up/downicad	Runtime analysis	Ms-dos command
Online debug	Hex converter	Setup
comms interface	Program eoroms	Quit

<ARROW>. <SPACE> or <Tab> selects operation, <CR> or <Command letter> executes

SAIA PCA ASSEMBLER MENU

File(s) to assemble (.SRC)

Create	link	fil	e	•	•	•	•	•	•	•	(.	LNK)	Yes
Create	listı	na	•	•	•	•	•	•	•	•	(.)	LST)	No
	Listi	ng	Wi	tn	:			Sy	′MC	01	t	abl	e	No
				Cr	09	5	re	+e	ere	nc	e	lıs	t	No
Greate	aocun	nent	at	10	п	•	•	•	•	•	(.	PGD)	No
Print d	locume	enta	atı	on	C	R	11	st		na				No

Enter file name. (ARROW) or (Tab) moves cursor, (CR) executes, (ESC) aborts.

SAIA PCA ASSEMBLER MENU

ASSEMBLER MENU HELP

Converts SRC type source files into LNK type files ready for the Linker. Up to 9 source files may be assembled at one time. Optional LST listing files or PGD documentation files can also be produced.

The Yes-No switches control the creation of LNK, LST or PGD files. and the destination of LST and PGD files (file or printer). The page format for LST and PGD files or printouts is as defined on the "Setup" menu.

The "Symbol table" and "Cross reference list" switches are ignored unless a a listing is being created, when "Create listing" is "Yes".

To send the LST listing or PGD documentation (if created) to the printer, set the "Print documentation OR listing" switch to "Yes". If printed, the LST or PGD files on the disk will be unchanged, and may now be out of date. It is not possible to print both the LST and the PGD documents at the same time.

NOTE: For DOS versions without the APPEND command, the Assembler overlay files PCAS.001-PCAS.004, and PCKEY.PWD, must be in the current directory.

Press any key to return to the menu

SAIA

Enter file name. $\langle ARROW \rangle$ or $\langle ab \rangle$ moves cursor, $\langle CR \rangle$ executes, $\langle ESC \rangle$ aborts.

SAIA PCA LINKER MENU

LINKER MENU HELP

Links separate LNK type files (created by the Assembler) together. to produce a single OBJ type loadable file. or a relocatable and re-linkable LNK type file, according to the "Relocatable file" switch. SYM symbol files and global PGD documentation files can also be produced, but not printed.

Up to 24 LNK files can be linked at one time. If more than 24 files must be linked, a sub-link can be done by setting "Relocatable file" to "Yes". This links up to 24 files into an intermediate LNK file, which can then be linked again with other LNK files.

The output file name is optional if only OBJ and/or SYM files are produced. In this case, if no output file name is entered, the name of the first file to be linked is used. If creating an LNK or PGD file, the output file name MUST be given, and it must not have the same name as an existing LNK or PGD file. To produce a global PGD file, valid (and up-to-date) PGD files, produced by the Assembler, must exist for every LNK file that is linked.

An optional base address for RSEG relocatable segments can be given. Files containing ASEG absolute segments MUST be linked in the correct order.

NOTE: For DOS versions without the APPEND command. the Linker overlay files PCLINK.001-PCLINK.003, and PCKEY.PWD, must be in the current directory.

Press any key to return to the menu

SAIA'PLC Programmable controllers

SAIA PCA UPLOADER AND DOWNLOADER MENU
Oberation. SPACE to select Download from OBJ file to PCA
Download Oberation Information:
Object file name . . . (.OBJ)
Start and End address 0 _____ 8191_
Fill unused instructions . . . No fill
Upload Oberation Information:
Object file name . . . (.OBJ)
Start and End address 0 _____ 8191_
Remove instructions Remove nothing

<SPACE> selects operation, <ARROW>/<Tab> moves, <CR> executes, <ESC> aborts.

SAIA PCA UPLOADER AND DOWNLOADER MENU

UP/DOWNLOAD MENU HELP

"Download" loads an OBJect file, produced by the Linker, into the PCA. "Ubload" reads PCA memory, and writes it to an OBJect file. The Upload or Download operation is selected by pressing the SPACE bar.

The PCA must be in PROGram mode, and must be connected via the PGU connector to the IBM PCs parallel interface, using cable PCA2.K43, pefore starting an Upload or Download.

For a Download, the name of the OBJ file must be entered. For an Obload, the name of the destination OBJ file must be entered. Ensure that the destination file name is not the same as an existing OBJ file, this will be overwritten by the uploaded file.

The Start and End addresses should be specified for an upload, so that areas of PCA memory which don't contain program steps or texts are not uploaded. When Downloading, you can optionally fill unused memory with NOP 0000s or FFFFs (important for PAS 31..46 checksum calculation). When Uploading, you can optionally remove any NOP 0000s and/or FFFFs.

Press any key to return to the menu



SAIA PCA ONLINE DEBUG V3.4

_Dispiav____

אא מא אא שט מש חח אש בא

-Clock-

_Program____

Write____

Display Write Program clock Save Load comms-Interface Reset Quit

SAIA PCA ASSEMBLER V1.0

MAIN MENU

ONLINE DEBUG HELP

When the PCA is in RUN mode, executing your program. Unline can be used to display or modify the values of elements, timers, counters, registers, program code or the internal clock. Before running Unline, the PCA must be in RUN mode and must be connected via the PGU connector to the IBM PCs barallel interface, using the cable PCA2.K43. Commands are entered by pressing the capital command letter of the command word displayed on the promot line at the bottom of the Unline screen. Values displayed in the Display and Clock windows are continually updated, the refresh rate depends of the number of values displayed. The Display window configuration can be loaded or saved in a file using the Load/Save commands (enter a filename).

For debugging in modes other than RUN, the "comms Interface" program (CI.EXE) can be executed by pressing "I", or the P10 Simulator program (P10.EXE) can be executed by pressing function key F10.

NOTE: PCAs containing old firmware, and the PCA13 and PCA210 series, do not support the Online program, see the Online documentation.

Press any key to return to the menu

SAIA

(C) SAIA AG. MURTEN

PCA COMMUNICATION INTERFACE VERSION 1988.03

SAIA PCA ASSEMBLER V1.0	MAIN MENU
COMMUNICATIONS INTERFACE HELP - COMMAND SUMMARY	
OPERATING MODES: OPERATING MODE SWITCH MUST BE CORRECTLY SET	
RUN Run Set start address with SST.	
BRK Break AAA:yyyy Executes to yyyy. KCR>/K+> steps, KANY CH	AR> stop.
SST Step AAA:xxxx Jumps to xxxx, then <cr>/<+> steps, displ</cr>	ays ACCU.
MAB ManBit AAA:eee Selects element, EEE:0/1, <cr>/<+> next.</cr>	<-> prev.
MAW ManBcd AAA:rrr Selects register. EEE:nn enters new value	
TEX Text AAA:tctc Character at tctc, EEE:nn enters char nn	decimal.
LCM LoadCopyMem LCM:xxxx.:yyyy Copies from xxxx to yyyy, PCA2.M1/	M3 only.
COM Compare Mem COM:xxxx,:yyyy Compares xxxx to yyyy, displays la	st adds.
PRG Prog AAA:aaaa Address, EEE:mnemo ooa Writes new prog	ram line,
CCC Clears un-entered line. <cr>/<+> next. <-> pre</cr>	٧.
IAD Inserts at adoress. KAD Kills address.	
COMMANDS: OPERATING MODE SWITCH POSITION MAY NEED CHANGIN	G
SCH:xxxx,:yyyy,:mnem Search for mnemonic CLM:xxxx,:yyyy Clear me	mory
LST:xxxx,:yyyy List flow chart LWC:xxxx.:yyyy List wit	h comment
DME:xxxx,:yyyy Display program DWC:xxxx,:yyyy Disp wit	h comment
DID:eee.:eee Display elements DTX:ttt.:ttt Display	texts
DRG:rrr,:rrr Display registers PAG Page for	mat
MET: yyyy Memory top for IAD/KAD	
Ctrl-P Print off/on Ctrl-S Pause Ctrl-Q Continue * Stop	G Go
Press any key to return to the menu	

5

A

SAIA[®]PLC Programmable controllers

:

SAIA PCA TEXT ASSEMBLER MENU

Text file to assemble (.TEX)

Text memory address. (SPACE) to change

4096

Enter file name, <ARROW> or <Tab> moves cursor, <CR> executes, <ESC> aborts.

SAIA PCA TEXT ASSEMBLER MENU

С

L

U

TEXT ASSEMBLER HELP

Assembles a text source file of type TEX, and produces an OBJ file which can be downloaded into the FCA from the "Ub/download" menu. A text documentation file of type TED is also produced. The correct text memory base address for your FCA must be selected before the text is assembled:

PCA0: 2048 PCA14: 4096 PCA2.M22 and M31: 4096 PCA2.M32: 8192

The TEX source file is written using your Editor. These text instructions can be used, see the PCA Assembler Manual for details:

- Comment Text line
- Text subroutine
- End of text subroutine
- T Text
- J Text line without line feed
- M Text line without carriage return or line feed
- e End of text
- Ax Absolute segment, at address "x"
- Press any key to return to the menu

SAIA'PLC Programmable controllers

SAIA FCA DISASSEMBLER MENU

Object file to disassemble . . (.OBJ) Output file name (.DIS)

Enter file name. <ARROW> or <Tab> moves cursor. <CR> executes. <ESC> aborts.

SAIA PCA DISASSEMBLER MENU

DISASSEMBLER HELF

Converts an OBJ type file into a Source format file of type DIS. This is the reverse of the Assembly process: a file containing symbols, labels and mnemonic code is created, but the DIS file does not contain the original comments. The DIS file can be edited. re-commented, assembled. linked and re-loaded into the PCA. To re-assemble the file, it must first be renamed to an SRC type file. from the "File handling" or "Ms-dos command" options.

To disassemble a program which exists in the memory of the PCA, the program must first be uploaded from the "Up/download" menu, or you can use PCLIST.

When the Disassembler is invoked from this menu, the entire OBJ file is disassembled. Small sections of the program may be disassembled by invoking the Disassembler (PCDIS) directly from MS-DOS, or from the "Ms-dos command" on the main manu. Refer to the PCA Assembler Manual for details. To disassemble a large OBJ file your IBM PC must have sufficient memory, about 50K is needed for every 1000 lines disassembled.

To disassemble to the printer, use PRN or LPT1/2/3 as the DIS file name.

Press any key to return to the menu

SAIA PCA FILE HANDLING UTILITIES V1.3

Directory	: C:\	405\PCA	SSTO		Files:	14	8 ytes	free:	4130816
		<dir></dir>	4.11.88	16.34	PCLINK	003	3456	੪.07.	87 1.00
• •		<dir></dir>	4.11.88	16.34	PE2	HLF	17817	23.05.	86 15.48
ONLINE	DAT	704	25.11.88	10.15	PE2	PRO	7312	23.03.	87 16.40
PCAMENU	DAT	801	25.11.88	10.15					
PCAS	001	9216	29.07.87	1.00					
PCAS	002	9600	7.07.87	1.00					
PCAS	003	10240	7.07.87	1.00					
PCAS	004	5504	7.07.87	1.00					
PCKEY	PWD	128	15.10.88	1.00					
PCLINK	001	5248	29.07.87	1.00					
PCLINK	002	7936	8.07.87	1.00					
Сару	5(2)		Display	y file		Form	at disk	ette	
Eras	e fil	e		Print	file				
Rename file co					e files		Quit		

 $\langle ARROW \rangle_* \langle SPACE \rangle$ or $\langle Tab \rangle$ selects, $\langle CR \rangle$ or $\langle Command letter \rangle$ executes. View directory with $\langle PgUp \rangle$ and $\langle PgDn \rangle_*$

SAIA PCA ASSEMBLER V1.0

MAIN MENU

FILE HANDLING HELP

The file handling menu provides the following functions:

- Display of the directory contents in a window, use PgUp and PgDn to view.
- Copy files from one drive or directory to another drive or directory, the wildcard filename characters "*" or "?" can be used, the copy is verified.
 Erase a file.
- Rename a file.
- Display a file. Up, down, left and right display scrolling is provided.
- Print a file. Text files of type SRC, PGD, CRF, FLW, DIF etc. can be sent to the printer. Do not try to print binary files, type OBJ, LNK etc.
- Compare two files. Does a character-by-character file compare, and shows if the files are the same. For comparing OBJect files, use the "Compare programs" option on the main menu.
- Format a diskette in drive A (on a one diskette system), or drive B (on systems with two diskettes).

All these operations can be also done directly from MS-DOS, or from the main menu "Ms-dos command" option, providing you know the MS-DOS commands.

NOTE: COPY.COM and FORMAT.COM must be available to use Copy or Format.

Press any key to return to the menu

SAIA PLC Programmable controllers

SAIA PCA FLOW CHART GENERATOR MENU

Documentation file to process . . (.PGD) _____

Flow cnart file name (.FLW)

Enter file name. <ARROW> or <Tab> moves cursor, <CR> executes. <ESC> aborts.

SAIA PCA FLOW CHART GENERATOR MENU

FLOW CHART HELP

Converts a PGD program documentation file into a flow chart file of type FLW.

The flow chart is a structured representation of the program. Which indicates program flow graphically. The page titles, subtitles and comments from the PGD documentation files are included in the flow chart.

Flow charts may be either created from individual source program PGD files produced by the Assembler, or from the global documentation PGD file produced by the Linker.

The page format of the flow chart is as specified in the "Setup" menu.

To print the flow chart, use PRN or LPT1/2/3 as the FLW file name.

Press any key to return to the menu

SAIA

SAIA'PLC Programmable controllers

SAIA PCA RUNTIME ANALYSIS MENU

Name of file to analyse (.OBJ)	
Name of output file (.RTA)	
Create modified source file	No
Modified source file name (.MOD)	
PLC type	PCA0
Baud rate	110
Character length	9 Bits
Communications mode	С

Enter file name, <ARROW> or <Tab> moves cursor, <CR> executes, <ESC> aborts.

SAIA FCA RUNTIME ANALYSIS MENU

RUNTIME ANALYSIS HELP

If your PCA uses serial communications, characters being received may be lost while the PCA is executing instructions, the check for a recieved character is done only at certain times. If "NOP 1111" instructions are inserted at critical points in your program, the PCA will check for received characters. This program examines your PCA program (OBJ file), and using the entered details of communications and PCA execution speed, can indicate if characters are likely to be lost. A file (type RTA) is produced, which shows the minimum time between received characters, and lists any critical areas in the program.

A modified SRC file can also be produced (type MOD), which contains the added "NOP 1111" instructions. This file is a disassembly of the OBJ file, and does NOT contain comments and equates from the original source file, and so should be used only as an example. The final modifications MUST be made in your SRC file(s).

Either the RTA or the MOD file can be sent to the printer by using PRN or LPT1/2/3 as the RTA or MOD file name.

Press any key to return to the menu

SAIA

SAIA PCA OBJECT TO INTEL HEX CONVERTER MENU

Object file to convert to intel Hex (.OBJ)

Hex file format Standard Intel Hex

Enter file name, <ARROW> or <Tab> moves cursor. <CR> executes. <ESC> aborts.

SAIA PCA OBJECT TO INTEL HEX CONVERTER MENU

PCA OBJECT TO INTEL HEX CONVERTER HELP

Converts an OBJ type file into an Intel Hex format file with type HEX.

The PCA Eprom Programming Utility (PCPROM, see the "Program eproms" option on the main menu) supports several Eprom Programmer types. If you have an Eprom programmer which is not supported, you can create a HEX file using this utility, which can be programmmed into Eproms using software supplied with your Eprom Programmer, such as "PromLink".

A single Hex file is created from the OBJect file. If this is too large to be programmed onto a single Eprom, it may be necessary to split your program into several smaller OBJect files, so that one Hex file can be created for each Eprom to be programmed. You can create smaller OBJect files using the Uploader, by uploading into Eprom-sized OBJect files.

You can create either "Standard Intel Hex" or "Intel MCS-86 Hex" format files, choose the one which is supported by your Eprom Programmer. Almost all Eprom Programmers will accept Standard Intel Hex format files.

Press any key to return to the menu

SAIA PCA EPROM PROGRAMMING UTILITIES V1.0

Operation, press SPACE to select	Program EPROM(s) from OBJect file
Object file name (.OBJ)	
Start address	0
End address	15999
EPROM type (shows steps per EPROM)	2764 (4096 Steps)
EPROM Programmer type	PCA2.P16
Serial port (COM1 or COM2)	COM1
Baud rate (2400, 4800 or 9600)	9600

Press SPACE to select operation. Arrow keys move, ENTER executes, ESC aborts.

SAIA PCA EFROM PROGRAMMING UTILITIES V1.0

EPROM PROGRAMMING HELP

Programs, reads, compares or blankchecks EPROMs of type 2716, 2732A, 2764, 27128, 27256, and non-volatile RAM types PCA2.R95 and PCA2.R96. Code or text object files can be programmed, compared or created.

The Start and End addresses should be set to the start and end addresses of your PCAs memory as default values. If the object file requires more than one EPROM of the selected type, you will be promoted to insert each EPROM.

To program an individual EPROM, set the Start and End addresses to the first and last steps contained in the EPROM. For example, to program Texts into the second 2764, set the Start address to 4096 and the End address to 8191.

It is possible to program the same EPROM with sections of code from different object files, if the code or text does not overlap data that has already been programmed.

When reading EPROMs to produce an object file, unprogrammed steps (FFFF in hex) are not stored in the object file.

Press any key to return to menu

SAIA

SAIA PCA COMPARE PROGRAMS MENU

First file .	• •	• •	• •	•••	• • •	(.OBJ)	
Second file	H .	• •	• •	••		(.OBJ)	
Name of file Default is CC	for Nso	aif 1e.	fere	nces	• • •	(.DIF)	
Start address		••	•••				٥
End address .			• •				4095

Enter file name. <ARROW> or <Tab> moves cursor. <CR> executes, <ESC> aborts.

SAIA PCA COMPARE PROGRAMS MENU

COMPARE PROGRAMS HELP

Compares two OBJ files, listing the differences to the console, or to a file of type DIF. The two files are compared between the entered Start and End addresses, which can be set to compare the entire files, or only small parts of the files. If the End address is not known, set the End address to the maximum (8191 or 15999), the actual End address will then be used.

If any steps are found which do not compare, the step address, the numeric codes, and the ASCII and decimal text values of the step in each file are listed to the DIF file or console.

The compare can be aborted by pressing any key. The compare is also aborted if more than 1000 compare errors are found.

When displaying the differences on the console, the display can be paused by pressing Ctrl-S, then any key to continue.

The differences list can be sent to the printer by using PRN or LPT1/2/3 as the DIF file name.

Press any key to return to the menu

SAIA

SAIA PCA CROSS REFERENCE LIST MENU

Instruction selectors:

PAS instructions	Yes	Text instructions	Yes
Jumo instructions	Yes	Wara processor instructions	Yes
JMS instructions	Yes	Other instructions	Yes
Logic instructions	Yes		

Enter file name, <ARROW> or <Tab> moves cursor. <CR> executes, <ESC> aborts.

SAIA PCA CROSS REFERENCE LIST MENU

CROSS REFERENCE LISTING HELP

Reads an OBJ file, and creates a cross reference listing file of type CRF. If the cross reference listing file name is not specified, a file with the same name as the OBJ file is created, but with type CRF.

The CRF file contains lists of specified types of instructions, where each instruction is used, and which elements are referenced by the instructions. The Yes-No switches can be used to select which instructions will appear in the cross reference listing.

By specifying Start and End addresses, a cross reference listing of a section of the program can be produced. To process an entire file, if the end address is not known, set the End address to the maximum possible address (8191 etc), the actual end address will then be used.

The page format of the cross reference listing is as specified in the "Setup" menu.

To print the listing, use PRN or LPT1/2/3 as the CRF file name.

Press any key to return to the menu

SAIA

MS-DOS COMMAND MENU

COMPAQ Personal Computer DOS Version 3.10

Press ESCAPE to return to the PCA menu.

C:\405\PCASSTO>>dir

SAIA PCA ASSEMBLER V1.0

MAIN MENU

MS-DOS COMMAND HELP

Executes MS-DOS commands, as it would be executed if entered at the MS-DOS prompt (eg. $C:>_$), here are some examples:

DIR *.LNK /W Displays the names of all files of type LNK. Any file type can be used (.OBJ, .CRF etc). The /W is optional, and causes 5 file names to be displayed on each line (useful for large directories). TYPE NAME.SRC Lists the contents of file NAME.SRC to the screen. DEL *.* Deletes all files in the current directory, be VERY careful with this one ! DEL *.OBJ Deletes all OBJ type files in the current directory, any file type can be used (.LNK, .CRF etc). PRINT NAME.SRC Prints the file NAME.SRC as a background operation. NOTE: This may interfere with serial communications.

Refer to your DOS Reference Manual for details of these and other commands. NOTE: The PCLIST Lister and the P10 Simulator can be executed from this menu, see your PCA documentation for details.

Press any key to return to the menu

SAIA*PLC Programmable controllers

SAIA PCA ASSEMBLER V1.0

MAIN MENU

*** SAIA AG. MURTEN. PRE-RELEASE VERSION FOR SAIA USE ONLY ***

Directory: C:\405\PCASSTO

25.11.88 17.12

Edit	Text assembler	Compare programs
Assemble	Disassempler	Xref listing
Link	flow chart	File handling
Up/downicad	Runtime analysis	Ms-dos command
Online debug	Hex converter	Setup
comms intertace	frogram eproms	Quit

Exit to MS-DOS (Y or N) ?

SAIA PCA ASSEMBLER V1.0

MAIN MENU

QUIT COMMAND HELP

The "Quit" command exits the PCA Assembler menus, returning to DOS. You must first verify that want to exit, by typing "Y" or "N".

All data which has been entered into the menu entry fields is saved in a file named PCAMENU.DAT, in the current directory or logged diskette. The next time you use the PCA menus from this directory or diskette, the file PCAMENU.DAT is read, and all the entry field information is restored, as it was when the PCA menus were last used. If PCAMENU.DAT is not present (or has an invalid checksum), the menu entry fields will be empty, and default switch settings are used.

Regularly used information, such as the names of files to be linked, can thus be permanently entered into the menu. It is a good idea to use a different work directory (or diskette) for each project, so that the different menu information can be saved and restored for each project.

Refer to the installation instructions for details of how to organize your project directories.

Press any key to return to the menu

SAIA

SAIA PLC Programmable controllers



SAIA

A-7

2.3 FORMAT OF THE SOURCE PROGRAM

2.3.1 LINE FEATURES

For writing a source program the following rules must be observed:

- One line of the source program consists of max. 80 characters.

- Each line is split up into 4 fields.
- Each line must be terminated with <CR>.

- The length of a field is optional, but it must not exceed the maximum number of characters (8 \emptyset). If you use the "Personal Editor", the correct line length is determined by means of the instruction "SET MARGINS 1 8 \emptyset <CR>".

- The contents of the fields 1 to 3 are defined.
- Field 4 starts with ";" and is available for any comments.

Each field must be terminated with a <BLANK> or a <TAB>.
 Exception: The last field must be terminated with <CR> (line end).

<CR> = Carriage return + line feed <BLANK> = Blank <TAB> = Tabulator function

Examples:

- 1) START STH E1 ; Interrogation start button FIELD 1 FIELD 2 FIELD 3 FIELD 4
- 2) ; This line contains only comments!

ENDM

Fields 1 to 3 have the lengths \emptyset , therefore field 4 may contain max. 8 \emptyset characters (including semicolon).

3)

; End of macro

Field 2 contains an assembler instruction. Fields 1 and 3 are empty.

2.3.2 DEFINITION OF THE FIELDS

Field 1: Field 1 contains labels, symbols, the name of the macro definition or the control character "\$" characterizing an assembler control instruction.

Field 2: Field 2 contains the operation code (Mnemonic or numerical code of the SAIA-PLC), assignment functions, and control instructions usable in the source program.

Field 3: Field 3 contains the operands. An operand is identified by arguments of PLC-instructions, or by assembler control instructions e.g. inputs, timer values, destination addresses of jump instructions (labels), etc. The operands can be indicated as absolute values, symbolically or as arithmetic expressions. Field 4: Field 4 contains any comment. This field starts with ";" and ends with <CR>. 2.3.3 LABELS AND SYMBOLS LABELS are symbolic addresses which are used for assignment to start addresses of subroutines, parallel programs and destination addresses of jumps. SYMBOLS are names for constants and variables. Labels and symbols may consist of max. 7 alphanumerical characters. The first character has to be a letter. Example: START EQU 1 VALVE1 EQU 33 LIM2 12 EQU BEGIN STH START ; Start button on ANH ; and limit switch 2 on END2 OUT VALVE1 : Valve 1 on • • • . . . JMP BEGIN ; Jump back to the beginning of the program The following examples are wrong: LIMSWIT2 ; more than 7 characters ON/OFF ; contains an inadmissible character 1A ; starts with a number LABEL ; because of the blanks at the beginning it is not interpreted as a label — — error message

SAIA

saia°pc

2.4 ASSEMBLER INSTRUCTION SET

The assembler instruction set comprises the operating code of the SAIA-PLC as well as the instructions SET, EQU, PUBL, EXTN, ORG, ASEG and RSEG the functions of which will be described in the following. The instructions MACRO, ENDM and LOCAL belong to the same cathegory and will be described in the chapter "Macros".

All of the above instructions are in <u>field 2</u> and must be entered in capitals.

2.4.1 OPERATION CODE

The whole instruction set of the SAIA-PLC (bit and word processor) is permitted as operation code.

BIT PROCESSOR:

NC MNEMOCODE

ØØ ØØ Ø1 Ø2	NOP SWP STH STL	(IF OP >< 1248) (IF OP = 1248) START HIGH START LOW
ø4	ANL	AND LOW
Ø5	ORH	OR HIGH
Ø6	URL	OR LOW
10 10	XUK	EXCLUSIVE OR
ØQ	DYN	DYNAMIC CONTROL
10	OUT	SET OUTPUT W ACCU
11	SE0	SET OUTPUT
12	REO	RESET OUTPUT
13	C00	COMPLEMENT OUTPUT
14	STR	*SET TIMER
15	SCR	*SET COUNTER
10	SEL	SEI INDEX
18		DECREMENT COUNTER
19	SEA	SET ACCII
20	JMP	*UNCONDITIONAL JUMP
21	JIO	*JUMP IF ACCU = 1
22	JIZ	*JUMP IF ACCU = Ø
23	JMS	*JUMP TO SUBROUTINE
24	RET	RETURN FROM SUBROUT
25	WIH	WAIT IF HIGH
26	WIL TNT	WAIL IF LOW
21	101	DECOEMENT INDEX
20	DEI	*DDOCDAM ASSIGNMENT
30	DOP	DISPLAY OPERAND
31	DTC	DISPLAY TIMER OR
	•	COUNTER

WORD PROCESSOR:

NC MNEMOCODE

ØØ	NOP	(IF OP > < 1248) NO OPERATION
ØØ	SWP	(IF OP = 1248) START WORD PROCESSOR
Ø1	RRG	READ REGISTER
Ø2	WRG	WRITE INTO REGISTER
Ø3	RRE	READ REG AND WRITE IN ELEM
Ø4	WRE	WRITE REG WITH ELEMENTS
Ø5	LAR	LOAD AO WITH REGISTERS
Ø6	SAR	STORE AO INTO REGISTERS
Ø7	LAC	LOAD AO WITH COUNTER
Ø8	SAC	STORE AO INTO COUNTER
Ø9	WEL	WRITE ELEM WITH LOWER DIGIT
1Ø	WEU	WRITE ELEM WITH UPPER DIGIT
11	INR	INCREMENT REGISTER
12	DER	DECREMENT REGISTER
13	SNC	SKIP IF NO CARRY OR IF O
14	CLA	CLEAR A
15	LAI	LOAD A IMMEDIAT
16	DBN	DECIMAL TO BINARY
17	BND	BINARY TO DECIMAL
18	ROR	ROTATE REGISTERS
19	ROA	ROTATE A
2Ø	EXG	EXCHANGE AO WITH A1
21	CLK	CLOCK SOURCE
22	SHI	SHIFT REGISTERS
23	тхт	TEXT
24	SEW	SKIP TO EWP OR SWP
25	SCP	COMPARE AØ WITH A1
26	SQR	SQUARE ROOT FROM A1
27	ADD	ADD AØ TO A1
28	SUB	SUBTRACT AØ FROM A1
29	MUL	MULTIPLY A1 BY AØ
ЗØ	DIV	DIVIDE A1 BY AØ
A		

31 EWP END WORD PROCESSOR

SAIA

B-5

The instructions fitted with * consist of two or more lines. Depending on the instruction various particularities must be noted. Further information on this is provided in the following chapters.

Timer and counter instructions:

1. LINE:	2. LINE	
NC MNC OPRD 14 STR 256 - 287 15 SCR 256 - 287 (256 - 319 for PCA23 and PCA14)	NC MNC ØØ - 15 TCV * 16 IUM 17 ITM 18 IHM 19 IFD 2Ø OFD 21 OEB 22 OTB 23 OSB 24 IEB 25 ITB 26 ISB 27 ADD 28 SUB 29 MUL 3Ø DIV 21 ICV	MEANING: TIMER/COUNTER VALUE IN : 1 * 8 BITS BCD IN : 1 \emptyset * 8 BITS BCD IN : 1 \emptyset * 8 BITS BCD IN : 2 \emptyset BITS BCD OUT : 2 \emptyset BITS BCD OUT : 2 \emptyset BITS BCD OUT : 8 BITS BINARY OUT : 16 BITS BINARY IN : 16 BITS BINARY IN : 16 BITS BINARY IN : 16 BITS BINARY IN : 16 BITS BINARY Z = Z + X Z = Z - X Z = Z * X Z = Z / X
		L - THELY COUNTER THEOL

*: TCV-values can be defined from \emptyset ...32767. Code conversion ensues automatically.

For PCA14/23 the following is valid:

The timer and counter registers have been extended from 15 to 16 bits which corresponds to an extension (in decimal) from 32767 to 65535. The extended timing and counting area is available only in the second line as a result of an arithmetic PLC-operation or via the input instructions IFD (19) or ISB (26). In other words, no values greater than 32767 are allowed in the assembler code. This restriction depends on the system.

The mnemo code used by STR and SCR in the second line is <u>exclusively</u> used in the SAIA-PLC assembler. It is impossible to use this mnemo code in the communication interface PCA2.T13 as well. Contrary to that, it is of course possible to use the numerical code in the second line in the PLC-assembler.

AA

Example:	STR	256		STR	256
	TCV	31562	corresponds to	15	842

B-6

1	NC	MNC	Meaning
ØØ	15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	TCV IUM ITM IFD OFD OEB OTB OSB IEB ISB ADD SUB MUL DIV ICV	Timer Counter Value In Units Multiply In Tenth Multiply In Hundreds Multiply In Five Digits Out Five Digits Out Eight bits Binary Out Twelve bits Binary Out Sixteen bits Binary In Eight bits Binary In Twelve bits Binary In Sixteen bits Binary Add Subtract Multiply Divide Index/Counter-Value
Jump instr	ructions:		
	سلا تفلفا العاسي		I, THE THURD THNEFOCLIDHS ARE THEN CALLED OVER
ddress ra address is address is Examples:	and JMSL ange of th therefor JMPL	. From a PCA2.1 re 15999	assembler version C8C1/1.1.05 on the extended M32 is also supported. The highest admissible corresponds to JMP Ø 2 224
ddress ra ddress is address is	and JMSL ange of th therefor JMPL JIOL	. From a le PCA2.1 re 15999 432Ø	corresponds to JIO Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø
AS-instru	and JMSL ange of th therefor JMPL JIOL	. From he PCA2.1 he 15999 432Ø	corresponds to JMP Ø 2 224 corresponds to JIO Ø Ø Ø
AS-instru	and JMSL ange of th therefor JMPL JIOL actions:	. From ne PCA2.1 ne 15999 432Ø	2. LINE:
AS-instru AS-instru L LINE: MC MNC PAS PAS PAS PAS PAS PAS	and JMSL ange of th s therefor JMPL JIOL dctions: OPRD Ø - 1 16 - 9 100 - 204	. From PCA2.1	2. LINE: NC MNC $\emptyset \emptyset - 31 \emptyset \emptyset - 31 (\text{the following 9 lines)}$

SAIA

 \sim

B-7

2.4.2 DEFINITION OF A CONSTANT

The definition EQU is used to assign a constant value to a symbol. This value cannot be altered anymore. Furthermore, it is impossible to assign another value to this symbol in the rest of the source program with the instruction SET. Numerical values, symbols and constants already defined and arithmetic expressions are permitted for assignment. Externally defined values (see EXTN) cannot be assigned. These values can also be assigned after using the corresponding symbol, i.e. they may appear anywhere in the program.

saia°pc

Examples: EMSTOP EQU 8 LAMP1 EQU 32 LAMP2 EQU LAMP1 + 1

The following value assignment is <u>not permitted</u> (upwards reference):

VALVE1 EQU VALVE2 + 1 VALVE2 EQU 38

2.4.3 DEFINITION OF A VARIABLE

The definition SET is used to assign a variable value to a symbol which can be altered as desired in the source program by means of other SET definitions.

Numerical vatues, symbols and constants already defined and arithmetic expressions are permitted for assignment.

Externally defined values (see EXTN) cannot be assigned.

These values may appear anywhere in the program, but they have to be used before the corresponding symbol.

Examples:	DELAY	SET	25	
	STR TCV	256 DELAY		
	DELAY	SET	25 + 1ø	
	STR TCV etc.	256 DELAY		
	The fol	llowing	value assignment is <u>not perm</u>	<u>itted</u> :

DELAY SET DELAY + 10

EQU

2.4.4 EXTERNALLY USABLE LABELS AND SYMBOLS

The assembler permits linking of several program parts which have been assembled individually to form a single program. Therefore, labels and symbols can also be defined in a program part, which can be activated or used from another program part assembled separately. These labels and symbols must be referred to as PUBL (public) in the program part in which they were defined. A <u>label</u> is considered defined if it is in the <u>first field</u>. An example is given in the following chapter. In the <u>assembler listing</u> in column UI a "P" is indicated for symbols and labels defined as public. See programming example in chapter 6.

Important: The instruction PUBL must be at the end of the corresponding program part.

2.4.5 LABELS AND SYMBOLS OUTSIDE THE MODULE

Labels and symbols which are defined as PUBL in another program part (module) must be referred to as EXTN (external) in all other modules in which they are used. A symbol referred to as external, may appear as an operand <u>only in an addition</u>. Subtraction, multiplication and division are not permitted. In the <u>assembler listing</u> in column UI an "E" appears for symbols and

labels defined as external. See programming example in chapter 6.

Important: The instruction EXTN must be at the end of the corresponding program part.

Example:

PROGRAM A

! Field 1	! Field 2	! Field 3 !	! Field 4
SYM1 SYM2	EQU EQU	12 36	
	STH ANH etc.	SYM1 SYM2	
	PUBL	SYM1, SYM2	; These symbols are also ; used in program B

B-9

EXTN

PUBL

B-10

-10	saia°pc	
	PROGRAM B	
	STH SYM2 XOR SYM1 etc.	
	EXTN SYM1, SYM2 ; These symbols are ; defined in program A	
2.4.6	DEFINITION OF A START ADDRESS	ORG
	The start address of a program can be defined by ORG (origi address may be given only in an absolute segment (see chapt ORG is in field 2, the start address is in field 3.	n). A start er 2.4.7).
	Examples: ORG 100 ORG PSTART ORG PSTART + 10	
	In ORG-instructions no externally defined symbol must be used.	s and labels
2.4.7	ABSOLUTE SEGMENT	ASEG
	A program part which still has to keep the addresses determ assembling after linking, is referred to as ASEG (absolute segment). ASEG characterizes the beginning of an absolute segment. Th instruction is used to determine the start address of the a segment.	ined by e ORG- bsolute
	Example: ASEG ; This program starts ORG 100 ; at address 100	
2.4.8	RELATIVE SEGMENT	RSEG
	RSEG indicates the beginning of a program part, the positio is not determined from the start. The absolute addresses ar by the linker. ASEG identifies the end of a relative segmen instruction is not admissible in a relative segment. If a program part is neither referred to as ASEG nor RSEG, matically defined as relative segment.	n of which e assigned t. The ORG- it is auto-

SAIA

2.4.9 PRESENTATION OF OPERANDS

With symbols of operands, a maximum number of 7 alphanumeric characters is possible. The first character must be a letter. The assembler accepts operands as absolute, symbolic or arithmetic expressions. Absolute values from \emptyset ...2 \emptyset 47 are admissible for all instructions.

Exceptions:

- Jump instructions and the 2nd line of PAS Ø...15: Ø...15999 : Ø...32767
- TCV (Timer/Counter Value)

Arithmetic expressions with symbols can also be used as operands. The following operations are possible:

- Addition - Subtraction
- Multiplication *
- (whole-number result) - Division 1
- Modulo
- Expressions in brackets ()
- Special character \$ The character \$ is replaced in the operand by the address valid for this program line.

Ex.: JIO \$ - 4 e.g. jump 4 lines back

In the case of the jump instructions JMPL, JIOZL, JIOL, JMSL (they cause 2-line instructions) the second line must also be taken into account.

Ex.: JIOL END 🔫 STH 1 ANL 12 JIZL \$-5 -

Exception for arithmetic operations:

Externally defined symbols can only be used for addition.

Examples: STL FLAG1 + 1000 ; indexing ORG (\$ + 9)/10 * 10; next program step ; falls on an address ; of 1Ø

> SYMBOL1 EQU SYMBOL3 - SYMBOL2



Example:

\$ TITLE DEMOPROGRAM 11.4
\$ STITLE DELAY

results in the following page heading in the assembler listing and the program documentation:

*** SAIA - PC *** DEMOPROGRAM V11.4 PAGE 1 MACRO - ASSEMBLER DELAY

PROCONTROL 3000 Bern Switzerland

The version number of the user program (V11.4) must be added to the title.

The company address is written in compliance with the custromer's specifications before supplying the software. It appears automatically and cannot be altered.

2.5.4 FORM FEED TO THE BEGINNING OF A PAGE

The instruction EJECT has the result that a new page is started in the assembler listing and the program documentation. The instruction EJECTL has the result that only in the assembler listing a new page is started.

2.5.5 ADDITION OF n BLANK LINES

With the instruction SPACE(n) n blank lines can be added to the assembler listing and program documentation.

Example:

\$ SPACE(12) ; causes 12 blank lines in the assembler listing.

2.5.6 INCLUSION OF A SOURCE PROGRAM FILE

The instruction INCLUDE is used to include other source programs or macros in the assembler listing. The programs must be called up in the source program in the respective position by entering

\$ INCLUDE d:name.type

The macros or source programs to be included must be specified giving the complete drive unit designation, name and data type. They must have source format. The data type is optional.

INCLUDE

SPACE(n)



B-14

Examples:

\$	INCLUDE	A:text.txt	; inclusion of a text file
\$ \$ \$	NOLIST INCLUDE LIST	B:delay.src	; inclusion of a source program ; the contents of which do not ; appear in the assembler ; listing

saia°pc

The use of the instruction INCLUDE in connection with macros is described in chapter 2.6.

2.5.7 SUBTITLE IN THE PROGRAM DOCUMENTATION

The instruction MARK is used to define subtitles in the source program, which can only be used in the program documentation PGDOK at the corresponding points. The first character of field 3 is automatically repeated 19 times before the actual subtitle in the program documentation.

MARK MARKL

Example:

ASEG ORG	1150		
MARK	-START DELAY	;	subtitle
STH	E1	;	input
DYN	PULS	;	generate 1 pulse
STR	TR1	;	timer
VAL	T1	;	delay time
ANH	PULS		
OUT	A1	;	delayed output
	ASEG ORG MARK STH DYN STR VAL ANH OUT	ASEG ORG 115Ø MARK -START DELAY STH E1 DYN PULS STR TR1 VAL T1 ANH PULS OUT A1	ASEG ORG 115Ø MARK -START DELAY ; STH E1 ; DYN PULS ; STR TR1 ; VAL T1 ; ANH PULS OUT A1 ;

SALA

has the following result in the program documentation:

			START DELAY
115Ø Ø1	STH	ø	Input
1151 Ø9	DYN	5ØØ	Generate 1 pulse
1152 14	STR	256	Timer
1153 16	16	1ø	Delay time
1154 Ø3	ANH	5ØØ	•
1155 1Ø	OUT	32	Delayed output

A blank line is added with MARKL before the subtitle.

2.5.8 DATE and TIME

The instructions DATE and TIME read in the system date and the system time when assembling and add them to the assembler listing and program documentation.

Activation in the source program:

\$ DATE
\$ TIME

Assembler listing:

\$ DATE tt-MM-yyyy
\$ TIME hh:mm

Program documentation:

DATE tt-MM-yyyy TIME hh:mm

(tt = day, MM = month, yyyy = year, hh = hours, mm = minutes)

DATE

2.6 MACROS

The performance of the SAIA-MACRO ASSEMBLER is considerably increased due to the use of macros. Thus, it is possible to create a comprehensive software library with program modules. Even an individual useroriented programming language can be defined in such a way. Constantly recurring program sequences are programmed in one line. In addition to high-speed programming, the employment of macros also guarantees uniformity among the program modules. An existing program module can always be adapted to the problem individually.

2.6.1 MACRO DEFINITION

MACRO-ENDM

A program part written by the user, which can be called up anywhere in the source program as often as desired, is referred to as macro. A macro program may contain up to 10 parameters, i.e. up to 10 names of constants or variables (symbols) can be transferred from the source program during assembly.

A macro program must be defined in the activating program, i.e. it must be available as complete text during assembly. There are two ways of defining a macro program:

- 1) The macro program is defined somewhere in the source program according to the following rules.
- 2) The macro program is defined as a complete file outside the source program calling up and stored on a disk (e.g. as part of a macro library). In this case, this file must be included in the assembling time by means of the instruction INCLUDE.

A macro definition is started with by the instruction MACRO and finished with the instruction ENDM. The instruction MACRO and ENDM are in the 2nd field.

Structure of a macro:

mname MACRO p1,p2,...p1Ø

... code

•••

ENDM

mname: arbitrary macro name consisting of up to 7 alphanumeric characters.

B-18

MACRO: reserved word, indicates the beginning of a macro ENDM : reserved word, indicates the end of a macro p1,p2,...p1Ø: list of parameters (often referred to as formal parameters). These reserve space for values which are used in the main program during assembly (so-called actual parameters). The syntax of the parameters follows the same rules as that of symbols and labels (see chapter 2.3.3).

sala°pc

2.6.2 PARAMETERS IN A MACRO

All symbols used as parameters must be fitted with the control character "%" in the macro source code. In the parameter list they are listed without this control character.

2.6.3 LOCAL LABELS AND SYMBOLS IN A MACRO

Labels and symbols which are used only in a macro must be identified by the instruction LOCAL (field 2) at the beginning of a macro. The corresponding labels and symbols are listed in field 3. In other words, due to the instruction LOCAL the scope of a symbol or a label is restricted to a macro. Thus, conflicts with the same symbols and labels outside the macro are avoided. In the macro program itself the local labels and symbols must be characterized by the control character "?". In the assembler listing they are replaced by the supplementary symbols (supplementary labels) "??Ø", ??1, etc. (see example chapter 6). A maximum number of 10 local symbols or labels can be defined. Arithmetic operations are not permitted together with local symbols and labels.

Example: LOCAL SYM1, LAB1, LAB2

2.6.4 INCLUSION OF A MACRO

INCLUDE

%

LOCAL

By means of the instruction INCLUDE all required macros are included as text in the source program, as far as they are not defined in the source program itself. INCLUDE is an assembler control instruction and must therefore be fitted with "\$" in the 1st field (see chapter 2.3.5). Due to the INCLUDE instruction the macros are included in the source program in PASS1, with no parameters being transferred. PASS 1 creates a temporary file of the EXP type, which will be erased automatically at the end of assembling.

Example:

Field 1 Field 2 Field 3 Field 4 \$ INCLUDE d:MTR1A1.SRC ; includes the macro file ; in the source program

It must be noted that this instruction does <u>not</u> actually call up the macro in the program code.

2.6.5 CALLING A MACRO AND PARAMETER TRANSFER

At the point in the program code where the macro program has to become active it is called up by its name (without mentioning the data type) in field 2.

In field 3 the complete actual parameters are listed in the correct sequence according to the definition.

If PAS-instructions are used in macros note especially that symbols can be used in the operation code (code in field 2), which are defined as parameters in the macro definition.

When calling up the macro in the source program, however, these parameters must be given as absolute values $(\emptyset \dots 31)$.

Example: Macro for setting or reading the time

Marco definition: SLTIME MACRO STD, MIN, SEC PAS 5Ø %STD 256 PAS 5Ø %MIN 257 PAS 5Ø %SEC 258 ENDM

Call in the source program: (Reading out the time)

SLTIME 15,16,17

Macro calls can be nested down through $1\emptyset$ levels and may also contain macro definitions again.

But parameters cannot be replaced down through several macro levels. The parameter of a nested macro called up cannot influence the parameters of the superior macro directly, but only via its parameter list. In other words, the control character "%" may also appear in a parameter list of a macro.

B-20

saia°pc

2.6.6	EXAMPLE OF A MACRO DEFINITION AND MACRO CALL						
	Macro def	Macro definition					
	By means delay tim file is o the file those two macros wi	By means of the editor a macro definition is written including a rise- delay time. It is assumed that the file name is "DELSTAR.SRC" and the file is on the disk in drive unit B. The INCLUDE instruction refers to the file name, the call to the macro name. In contrast to the example, those two names need not be identical. A macro file may contain several macros with different macro names.					
	FIELD 1	FIELD 2	FIELD 3	FIELD 4			
	DELSTAR	MACRO	E1,M1,TIMER,TVALUE,A1				
		STH DYN STR TCV STL ANH OUT ENDM	%E1 %M1 %TIMER %TVALUE %TIMER %M1 %A1	; Edge ; triggering ; Start of timer ; Delay time ; Delayed output			
	<u>Macro cal</u>	<u>l in the s</u>	ource program				
	FIELD 1	FIELD 2	FIELD 3				
	; Equates	: প					
	MAIN KEY1 FLAG1 MOTOR TMR TIMELAG	EQU EQU EQU EQU EQU SET	1 5 6ØØ 34 26Ø 15	· · ·			
	\$	INCLUDE	B:DELSTAR.SRC				
	BEGIN	STH ANL JIZ DELSTAR JMP	MAIN MOTOR BEGIN KEY1,FLAG1,TMR,TIMELAG, BEGIN	MOTOR			
	; End of	program					
	Transfer symbols o to the co the <u>posit</u>	of the act f the para rrespondin <u>ion</u> and no	ual values ("equates") i meter list in the call. Ig parameters in the macr it on the name.	s effected via the Allocation of the values to definition depends on			

MA



MA

A-16

sala°pc C-6 3.3 SYNTAX CHECKS OF THE ASSEMBLER In PASS 1 and PASS 2 the assembler checks the operands and the operation code for syntax and if the operation code and/or the operands are not admissible, an error message is output. 3.3.1 SYNTAX CHECKS OF THE OPERANDS One-line instructions: range Ø...2047 Multi-line instructions: Jump instructions JMP, JMS, JIO, JIZ: The operand is in the range 1...2047 Jump instructions JMPL, JMSL, JIOL, JIZL: The operand is in the range \emptyset ...15999 PAS ØØ...15: Operand in the 2nd line: Ø...15999 PAS 15...99: Operand in the 2nd line: Ø...2Ø47 PAS > 99:Operand in the line 2 to 10: 0...2047 STR/SCR instruction Operand in the 2nd line: Ø...32767 3.3.2 SYNTAX CHECKS OF THE OPERATION CODE One-line instructions: Numerical operation code: ØØ...31 Whole mnemo code of the bit and word processor PAS-instructions: Lines 2 to 10: only numerical code 00...31. If PAS-instructions are called in macros with parameters referring to the lines 2 to 10, these parameters have to be given in numerical code in the range \emptyset ...31 in the macro call. They must not be symbols which were assigned a value by means of a SET or EQU-instruction. The number of lines which have to follow a PAS-instruction will be checked.

3.3.3 ERROR MESSAGES

Error messages of the operating system MS-DOS and error messages of the assembler must be differentiated. Error messages of the DOS-operating system are described in chapter "Messages" in the DOS-manual.

The assembler outputs the following types of error messages:

- FATAL ERROR

- ERROR
- WARNING

A FATAL ERROR leads to the abortion of the running program. ERROR and WARNING are indicated, but the program continues running.

List of the FATAL ERRORS

All of these FATAL ERRORS lead to an immediate interruption of the assembler program.

All files already opened are <u>not</u> closed.

ARGUMENT STACK OVERFLOW: A maximum number of 10 arguments per macro may be given.

ARGUMENTABLE OVERFLOW: The sum of the argument characters is limited.

CALLSTACK OVERFLOW: A maximum number of 10 additional macros may be called in a macro.

CANNOT CREATE FILE: A destination file cannot be opened.

CANNOT OPEN FILE: A file to be assembled or a file which is to be included in the source program by means of INCLUDE does not exist.

CANNOT WRITE TO DISK: Error when writing to disk e.g. disk is full.

HEAP OVERFLOW: The number of used symbols and symbol references (and macro definitions, too) is too large for the storage space still available.

ILLEGAL COMMAND LINE SYNTAX: The syntax of the assembler call is not correct.

INCLUDE CANNOT BE NESTED: The instruction INCLUDE cannot be executed.

MACROTABLE OVERFLOW: Too much storage space is required for the macro definitions.

NO MACRO-CMD IN ARGUMLIST: A macroname was used in the parameter list.

SOURCE FILE NOT SPECIFIED: The file to be assembled has not been specified.

C-7



TOO MANY MACROS: The number or the storage requirement of the macro definitions is too large.

TOO MANY SYMBOLS: The number of the symbols used is too large for the string storage still available.

List of ERROR messages

Execution of the assembler program is not interrupted. However, no value is allocated to the faulty symbol or label.

ILLEGAL EXPR WITH RELATIVE OR EXTN SYM: The respective symbol was used in an inadmissible expression e.g. multiplication with an externally defined symbol.

ILLEGAL OR MISSING OPCODE OR ASSEMBL-DIR: The mentioned operation code or the assembler instruction is not permitted or missing.

ILL. SYNTAX OR TOO MANY ARGS IN ARG-LIST: The syntax of the parameter list was not followed, or too many parameters are listed.

MISSING FIRST OPERAND: The first operand is missing.

MISSING PARENTHESIS IN EXPRESSION: In the mentioned expression a parenthesis is missing.

MULTIPLE DEFINED SYMBOL: The mentioned symbol was defined several times.

OPCODE NOT OF APPROPRIATE CLASS: The mentioned operation code is not permitted at this point.

OPERAND OUT OF RANGE: The value of the mentioned operand is not within the admissible range.

PARAM NOT SPEC. IN MACRO- OR LOCAL LIST: The mentioned parameter was not specified in the list of the macro definition or in the list of the locally used symbols.

SYMBOL OR ASSEMBL-CNTL EXPECTED: A symbol or an assembler control character is expected in the given position.

List of WARNINGS

Like the errors, the warnings do not lead to an interruption of the assembler program. The value of the respective symbol, however, is kept, even if it is wrong.

ILLEGAL ASSEMBL-DIR WITHIN REL SEGM: The mentioned assembler instruction must not be used in a relative segment.

NO LABEL ALLOWED HERE: No label is allowed at this point.

UNDEFINED SYMBOL: The mentioned symbol was not defined.

sala°pc

16. SHORT DESCRIPTION OF THE MS-DOS INSTRUCTIONS

This is a summary of the most important DOS-instructions used in conjunction with the assembler.

d = drive unit (A or B)

ASSIGN d1=d2

reroutes instructions given to drive unit d1 to drive unit d2 BREAK ON checks for program interruption < CTRL> < BREAK ---BREAK OFF the above check only for standard I/O

CHKDSK d: CHKDSK d:/F

CLS

COMP d:f1.src d:f2.src COPY d:name.type d: COPY d:*.* d: COPY d:f1.src+d:f2.src d:f3.src

DATE DEL

DIR d: DIR d:name.* DIR d:>LPT1: DIR d:>d:contents.txt

DISKCOMP d: d: DISKCOPY d: d:

ERASE d:name.src ERASE d:*.obj ERASE d:*.* FORMAT d: FORMAT B:/S FORMAT B:/V

KEYBxx

MODE MODE LPT1:132 MODE LPT1:=COM1: MODE COM1:12,N,B,1,P

MORE < d:name.src</pre> PRINT d:name.src

PRINT /T

checks disks and creates the state report checks disks, creates the state report and rectifies errors clears the screen display

compares file f1.src to file f2.src copies file name.type copies all files links the files fl.src and f2.src to form f3.src

in- and output of the date see ERASE

· contents of the disk contents of all files name outputs the contents to the printer writes contents into the file contents.txt

compares disks copies disks

erases file name.src erases all files with the extension obj erases all files formats disk formats disk B and copies operating system formats disk B with system request for data record

loads auxiliary file for country-specific keyboard (xx = GR for German keyboard)

sets mode for printer or screen sets IBM(-compatible) printer to condensed letters reroutes printer output to serial interface sets parameters for serial interface to 1200 Baud, no parity bit, 8 bit/character 1 stop bit, printer mode

displays file name.src pagewise on the screen includes the file name.src in the queue and prints it out deletes all files in the print queue

saia®pc

RENAME B:name1.src b:name2.src SYS d: TIME TYPE d:name.type renames file name1.src into name2.src

copies operating system in-/output of the time displays the file contents on the screen

10. SAIA-TEXT ASSEMBLER (PCTEXT)

10.1 GENERAL DESCRIPTION

- PCTEXT enables you to produce a text file with the aid of a standard editor (Personal Editor, Wordstar etc.) by following a few simple rules. Thus, it is possible to present the text having output format already while editing.
- PCTEXT converts the produced text file into an object file (.OBJ) compatible with the SAIA-MACRO-ASSEMBLER which can then be loaded into the SAIA-PLC by means of "PGLINK". Download In addition to the object file a text documentation file (.TED) similar to the SAIA-PLC-text editor (PCA23 and PCA14) is produced.

PCTEXT helps you create and modify the PLC-text in a simple way.

- You can : insert characters
 - delete characters
 - add texts
 - address texts with absolute addresses

10.1.1 PRODUCED FILES

"PCTEXT" produces the following files with the extension:

- .TED Text documentation file which is similar to the format of the PCA-editor
- .OBJ Files which can be loaded into the SAIA-PLC with "PCLINK".

"Download"

10.2 CREATION OF THE SOURCE FILE

The source file (source) is produced as a text file with the aid of an editor or a text processing program. As editor either the line editor being part of the MS-DOS or a commercially available processing program suitable for editing can be used. The relevant rules will be dealt with in detail in the following chapters.

For the production of a source file no tabs must be used.

Important: With the Personal Editor the parameter <notabs> must be entered when storing the source file.

MA





MA

G-13

G-14

saia°pc

10.2.3	TEXT ASSEMBLER INSTRUCTION SET
	C - comment C,c
	Identifies a text line in the source file which is not processed.
	Source : C c comment line in the source file C
	- text line
	Identifies a processable text line in the source file.
	Source : . **********************************
	L - text subroutine L,I
	Identifies the address of a text subroutine in the text documentation (.TED) with '=>'.
	Source : L **********************************
	(.TED): => 0 .\$0.1.0.\$.*.0.1.0 1 ^M^J.\$0.1.0.*P 2 .C.A1.4*^M^J.\$ 30.1.0.\$.*.0.1.0^M 4 ^J

saia°pc U,u U - end of the text subroutine Ends the text subroutine with '\$U' and fills up the rest of the line with '^@'. If the control character 'U' is in the last line of the text subroutine, Carriage Return/Line Feed of this line are suppressed. Source : L ****** * PCA 14 * • **** U ***** L * PCA 14 * U ****** 0 \$.*.0.1.0^M 1 ^J. . . . *. .P.C.A 2 . .1.4. .*^MJ. . . => 0 (.TED) : **3** . .\$.*.0.1.0^MJ.\$.U => 4 \$.*.0.1.0^M 5 ^J. *. .P.C.A 6 . .1.4. .*^M^J. . . 7 . .\$.*.0.1.0.\$.U^@^@ T.t T - Text Identifies the start address of a text in the text documentation (.TED) with -> . Source : TSODECO-SAIA LANDIS & GYR 0 .S.O.D.E.C.O.-.S.A.I (.TED) : -> 1 .A.L.A.N.D 2 .I.S. .&. .G.Y.R^M^J J,j J - Text line without Line Feed Suppresses Line Feed (^J) at the end of a text line in the source file. PCA14 for complex tasks Source : J (.TED) 0 P.C.A.1.4. .f 1 .o.r. .c.o.m.p.l.e.x 2 . .t.a.s.k.s^M^J^M

SAIA

G-15

G-16

saia°pc

M.m M - Text line without Carriage Return/Line Feed Suppresses Carriage Return (^M)/Line Feed (^J) at the end of a line in the source file. More control with a PCA14 Source : Μ M.o.r.e. .c.o (.TED) 0 .n.t.r.o.l. .w.i.t.h 1 2 . .a. .P.C.A.1.4[^]M[^]J 0 @ - End of text Fills up the rest of the line with `^@`. If the control character `@` is in the last text line, Carriage Return/Line Feed of this line are suppressed. Source : Т PCA14 for complex tasks . 0 1)T More control with a PCA14 0 (.TED) -> 0P.C.A.1.4. .f .o.r^M^J. . . .c.o.m 1 2 .p.l.e.x. .t.a.s.k.s 3 ^M^J^@^@^@^@^@^@^@@ M.o.r.e. .c.o -> 4 5 .n.t.r.o.l^M^J. . . .w.i.t.h. .a. .P.C.A 6 7 .1.4^@^@^@^@^@^@^@^@ 1) T<CR> without ^M/^J T <CR> with ^M/^J

MA

A,a A - Absolute segment Format : Ax or ax ^A identifies the beginning of an absolute segment. ^x determines the start address of the absolute segment. The start addresses may be defined only in ascending order. Source : A100 Т \$H Check sum of • the system program. @ A110 Т \$H . Check sum of the user program. @ (.TED) -> 100 . . .\$.H^M^J. . .C.h 101 .e.c.k. .s.u.m. .o.f ^M^J. . .t.h.e. .s.y .s.t.e.m. .p.r.o.g.r 102 103 104 .a.m..^M^J^@^@^@^@ . . .\$.H^M^J. . .C.h -> 110 111 .e.c.k. .s.u.m. .o.f ^M^J. . .t.h.e. .u.s 112 .e.r. .p.r.o.g.r.a.m ..^M^J^@^@^@^@^@@ 113 114

G-17



\$H

The \$LO01PLC series is the newest \$L001product of the \$L001PLC-family. \$L001PLC is available in \$c269 countries!

SAIA

LANDIS & GYR

.

The jump address of the text subroutine (\$L001) is defined by the absolute address (A1).

1) L < CR >

10.5

10.5.1 Source

С C

С С С

A1 L

С С \mathbf{C} .

С A10 Т

@

USAIA-

APPLICATION EXAMPLE

text :

text subroutine :

G-20

saia°pc

10.5.2 Text documentation (.TED) B:fname.TED => 1 .S.A.I.A.-.\$.U^@^@^@ -> 10 ^M^J^M^J. 11 .\$.H^M^J^M^J. . . . 12 . . .T.h.e. .\$.L.0.0 13 .1.P.L.C. .s.e.r.i.e 14 .s. .i.s. .t.h.e. .n 15 .e.w.e.s.t^M^J. . . 16\$.L.0.0.1.p.r 17 .o.d.u.c.t. .o.f. .t .h.e. .\$.L.0.0.1.P.L 18 19 .C.-.f.a.m.i.l.y..^M ^J.\$.L.0 20 .0.1.P.L.C. .i.s. .a 21 .v.a.i.l.a.b.l.e. .i 22 23 .n. .\$.c.2.6.9. .c.o .u.n.t.r.i.e.s.!^M^J 24 25 ^M^J^M^J^M_J.\$. .0.0 26 .6. .S.A.I.A^M^J^M^J 27 28 29 .R^M^J^@^@^@^@^@^@

10.6	SUMMARY	0F	THE	PCTEXT	INSTRUCTION	SET

<u>;</u>

í ____

C,c	Comment in the source file
•	Text line in the source file
L,1	Text subroutine
U <u>,</u> u	End of the text subroutine
T,t	Text
J , j	Text line without Line Feed
M,m	Text line without Carriage Return/Line Feed
Ø	End of text
Ax,ax	Absolute segment with the address 'x'

G-21