1 Overview

STRUC G is the **graphics oriented configuring language** for the digital control system **SIMADYN D**.

STRUC G describes and specifies the interaction of the **hardware** components such as processor or peripheral modules and the **software** components such as controller, arithmetic or logic blocks.

The list oriented configuring language STRUC L is available, in addition to STRUC G for the configuring of SIMADYN D. Full compatibility exists between STRUC G and STRUC L. Configuring generated by STRUC L can be further processed, without restrictions, with STRUC G and vice versa.

STRUC G supports all **configuring steps** that are necessary to:

- **Design** a solution with SIMADYN D for the set control, regulation and communication tasks
- **Create** a **program** for SIMADYN D and **load** it into the program memory sub-modules of a SIMADYN D rack
- Subsequently implement the **commissioning** and finally
- Automatically generate the complete **documentation** of the configuration

Additional configuring and documenting tools are not absolutely necessary.

PC s are utilized as configuring tools for STRUC G.

1.1 Configuring Hierarchies

Your configuring data is **hierarchically** managed under STRUC G. STRUC G supplies an integrated set of tools for processing this configuring data.

STRUC G sub–divides configuring data into the following hierarchical levels:

Project

- Group term belonging to a **plant** or a **plant section**, under which the master programs of all racks are managed.
- **Macros** (see below) can be created for each project, that simply group frequently repeating sub-functions together.

Master Program

- Describes the hardware configuration of **one** rack
- The configuring of the master program is implemented in the **graphics oriented master program editor** The positions (slots) and types of the processor modules, the sub-modules, the communication and I/O modules are defined here (**arrangement drawing**)
  In addition, the parameters of the individual boards, i.e. the sampling times of processor modules, are spe cified in the master program (**module parameter drawing**)
- The designation ‘Master Program ’’is also utilized as a group term for all function packets in the processor modules within a rack

Function Packet

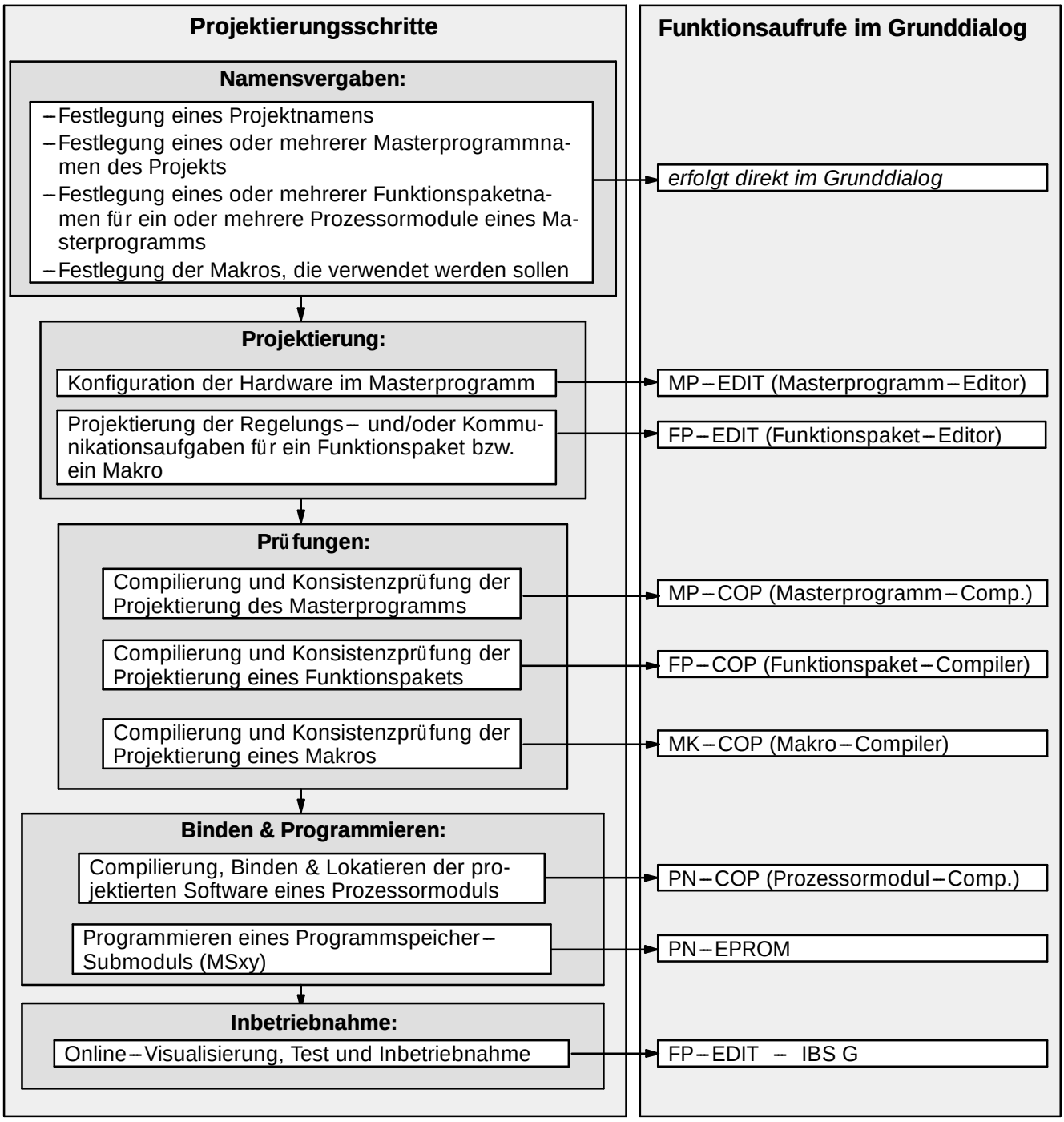
- Describes one or several configured **control or communication tasks** for a processor module. A processor module can be allocated several function packets. All configured control or communication tasks are created from groups of **function blocks** and/or **macros**.
- The configuring of a control or communication task is implemented in the **graphics oriented function packet** editor. The function blocks are selected from libraries, inserted onto drawing pages and their param eters set. The signal inter-connections are created by clicking on the start and end locations of the signal connections in the editor; the line routing and text positioning is automatically implemented by the editor (auto-routing).

Macros

- Permit the possibility of grouping several function blocks together to make one unit. A macro is configured in the **graphics editor** just as a function packet and can subsequently be reused as often as necessary. Macros permit the designer to create new function block types.

1.2 Configuring Steps with STRUC G

The starting point for all STRUC G work is the **basic dialog**.
The basic dialog manages the users projects, master programs, function packets and macros as well as displaying them in a clearly defined structure. All **tools,** necessary for up to the programming of the programme memory sub-modules or commissioning of the configuration, are invoked from the basic dialog:



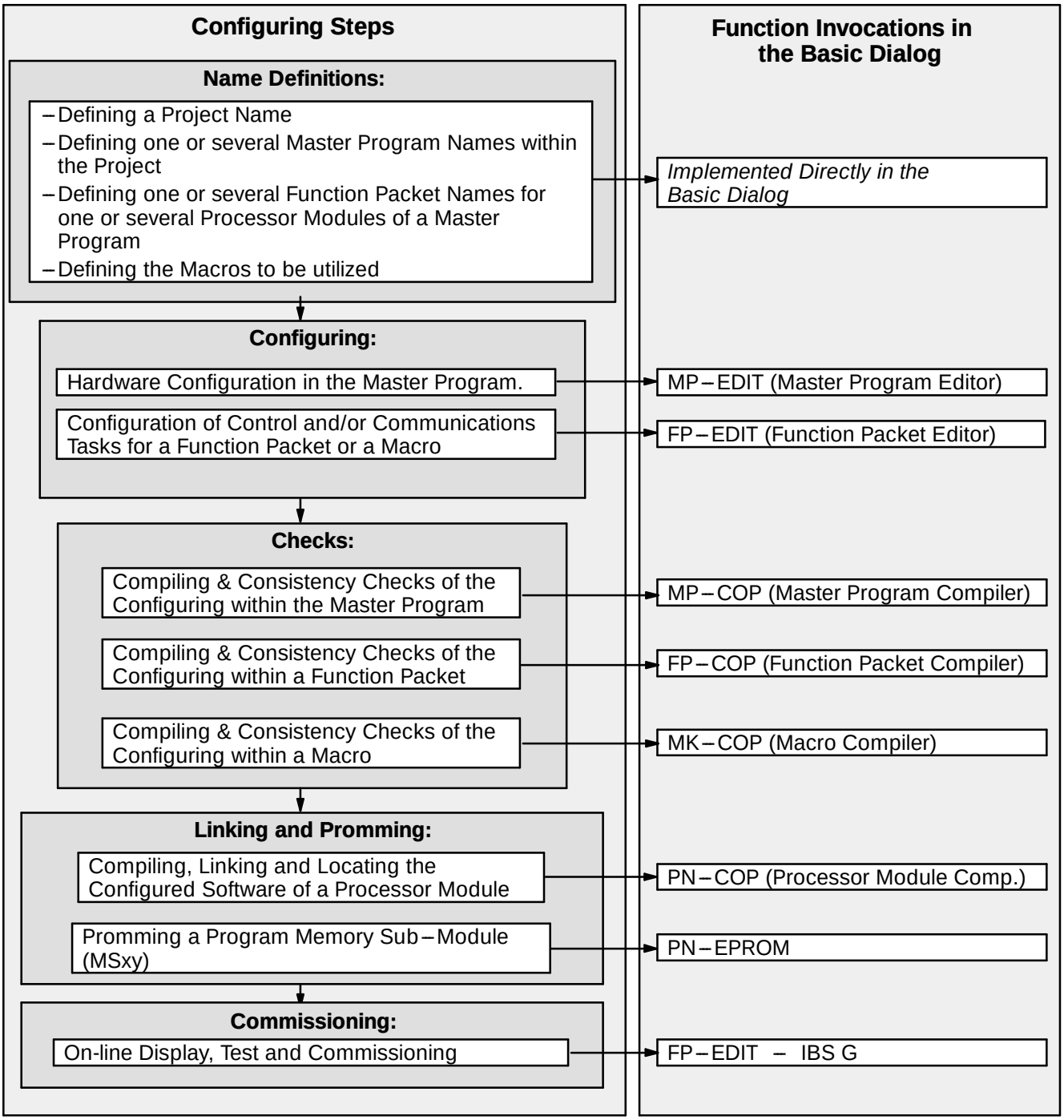


Diagram 1/1: Configuring Steps with STRUC G

1.3 STRUC Workstation

The following hardware components are necessary for working with STRUC G:

- A PC with the STRUC G configuring software
- Internal or external prommer for programming the P16 and P32 program memory sub-modules
- Printer / plotter for printing the configuring as documentation

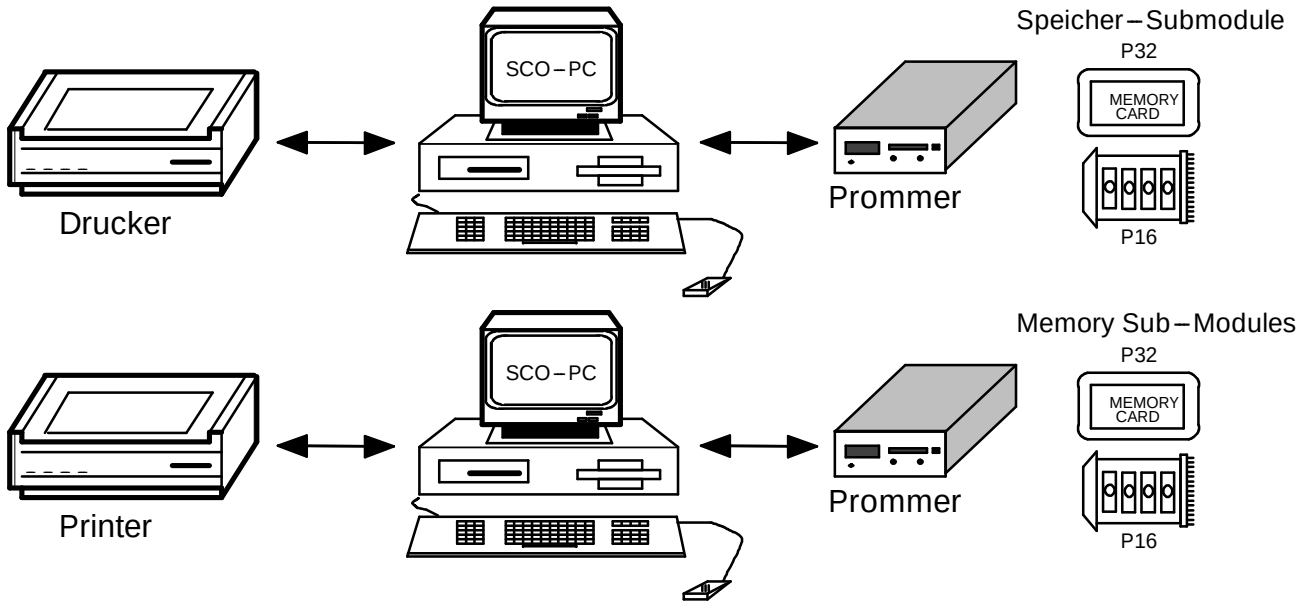


Diagram 1/2: Components for a STRUC G Workstation

1.3.1 Hardware Components Required For STRUC

The following list describes the hardware configuration required by STRUC G.

PC Minimum Equipment: ` PC486 with operating system with SCO ODT V3.0
20 MByte main memory
200 MByte hard disk
High resolution monitor (1024 x 786 Pixel)
3.5 ”disk drive,
CD ROM drive (with **SCSI** interface),

Optional: MOD drive (3.5 ”, 128 Mbyte)
High resolution color monitor (1280 x 1024 Pixel)

Prommer: ` Internal prommer PP1I (order no. 6DD1672–0AF0) or
` External prommer PP1X (order no. 6DD1672–0AD0)

Printer: ` WDV2400–CP (CGM, Postscript format) or
` Postscript printer

optional: **Configuring Devices Network**

` Ethernet Board: Western Digital WD LAN–EP 16 E
(compatible with the 'SCO Hardware Compatibility Handbook')

SINEC H1: ` Board SINEC CP1411 or equivalent type (see product information "SOFTNET H1–TF/UNIX for SCO", item number S79200–A0XXX–X–01–7437), only necessary when IBS–G or a download via SINEC H1 is to take place.

1.3.2 Software Components for STRUC G

The STRUC G software runs on the UNIX operating system SCO–ODT V3.0
(Open Desktop Personal System V3.0) and SCO Open Server 5.0.

When a board SINEC CP1411 has been installed for the IBS–G or download via SINEC H1, then the SINEC H1 driver and the SINEC H1 installation software is additionally necessary and can be obtained via the order number 6GK1704–1TC00–0EA0 (German) or 6GK1704–1TC01–0EA0 (English). The order designation for the CP1411 is 6GK1 141–1A00. The current order numbers can be reviewed in the catalog IK10.

1.3.3 Workstations in an Interactive Network Group

STRUC has been designed for single operation as well as for working in an interactive network group. The structure and layout of the network is relatively unrestricted; the following minimum equipment must be available:

- A PC with installed STRUC G
- A streamer tape, a magneto optical or a disk drive on any one of the configuring devices
- A printer (plotter) connection on any one of the configuring devices
- An internal or external prommer

Working in a network offers the following advantages with respect to the individual workstation:

+ Several users can utilize common data and peripheral devices. Therefore each PC does not require its own printer, disk drives, streamer tape drives or parallel prommers.
+ The data in a project is saved only once, even when several users are working on one project.

Diagram 1/3 shows **one example** of a network constellation with differing STRUC workstations.

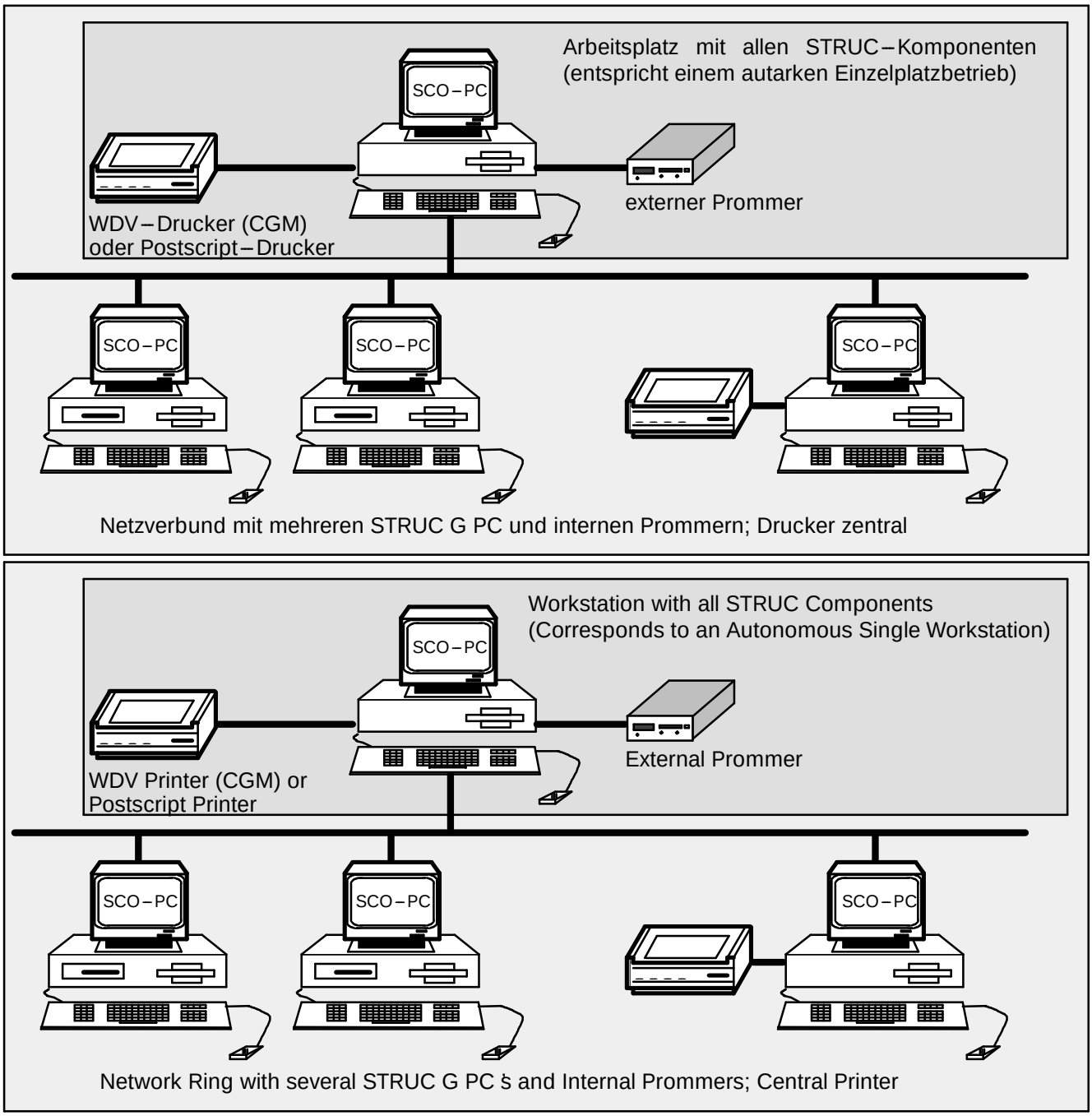


Diagram 1/3: STRUC G Workstations in a Network Group

1.3.4 Individual and Group Utilization

Every user requires a login identification to the configuring devices in order to work on STRUC projects.

If **one user** supervises a STRUC project **alone**, then all work is processed under his login identification. Configuring data, such as master programs and function packets, are stored in a user data area in his home directory. The required data is always inserted into a temporary data area when it is to be processed.

If **several STRUC users** work on a project **together**, then it is practical to have the system administrator assign access login identifications with the same group ID (i.e. Smith.Group 1 ', Jones.Group 1 'and Bass.Group 1).

The configuring data is then saved only **once** per group in the user data area of **one** user. All other group members can access this data.
The processing always occurs in the temporary data areas of the individual group members.

Please take note that it is not possible for two users to simultaneously process the same master program or function packet, since the modified data is saved into the configuring data area when they are finished. Therefore modifications from the first version to be saved, of the simultaneously processed data, is lost.

1.3.5 Starting the Configuring Session

The user must identify himself to the system, at the start of the configuring session, with his user ID (i.e. Smith.Gr1) and his password.
The valid user ID and the password are assigned by the system administrator.
At all STRUC–PC or STRUC–Notebooks, preinstalled by SNI, the user ID and the password is "gast ".

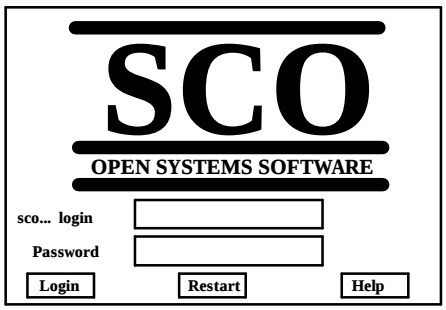


Diagram 1/4: Starting the Configuring Session: 'login " under SCO Unix

1.4 Quick Menu

Power up both PC and monitor and wait till the prompt:

Boot
:

appears on the screen. Acknowledge with ↵ (Return).

Wait for the PC prompt to continue the start-up with *CNTRL d*:
Enter CNTRL d (without ↵).

Next break-point is the prompt to enter a new time for the computer:
Enter ↵ .

The start-up is completed with the prompt for the login/password entry. It is essential to switch the STRUC–PC to console 2 before making these entries. This is implemented by simultaneously pressing the three keys CNTRL, ALT and F2. If the screen does not change, then the PC has already automatically switched to this console.
Now make the login / password entries:

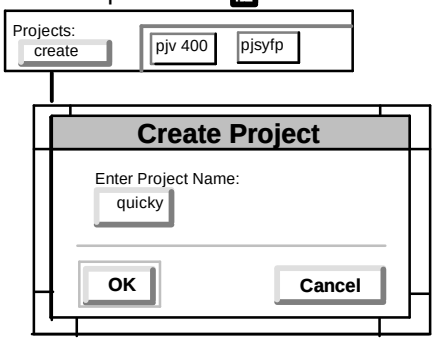
login: *guest* ↵
password: *guest* ↵

A screen is displayed with two lines ("Continue my last Session " [Default] and "Start a New Session "); position mouse to "OK " and *click left-hand mouse key* ↵ .

The UNIX environment platform has now been created. Now search for the STRUC symbol and select it with a ***double*** .

The basic dialog screen will then be created. A **new project** (with the name "Quicky") is created:

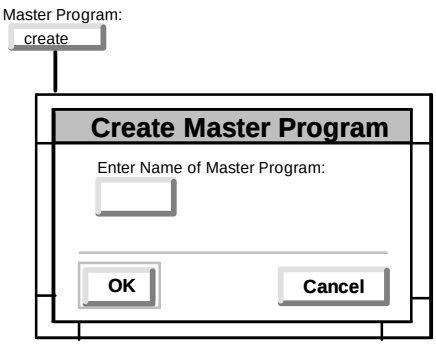
Use on the button Create in the line "Projects" and enter the name "Quicky" (upper case characters are automatically converted to lower case characters) in the name window, that is subsequently popped onto screen, and complete with :



The button quicky then appears in the projects panel.

A **new master program** (with the name "Single") must subsequently be created, in which all HW and system definitions are to be made:

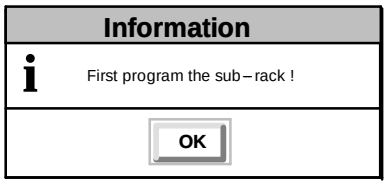
- Select quicky with (button becomes green).
- Use on the button create in the line "Master Program" and enter the name "S*ingle*" (upper case characters are automatically converted to lower case characters) in the name window, which is subsequently popped onto screen, and complete with :



The button single then appears in the panel "Master Program".

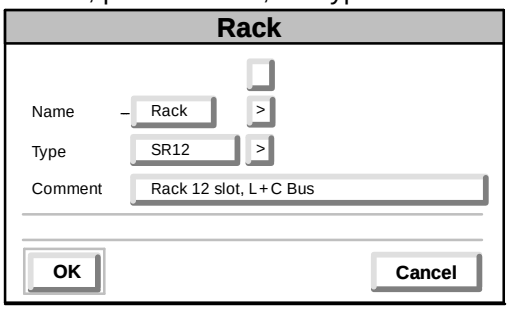
Now define the **hardware configuration** in the master program single:

- Select single with (button becomes green).
- Select the button 

in the menu strip (on the upper edge of the basic dialog screen) using and acknowledge the automatically selected panel "Configure" with .
- The subsequently displayed information panel 

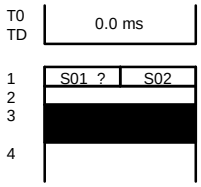
is acknowledged with .

G The rack screen is then displayed. Define a name and rack type.
The empty panel behind 'Name" is selected with ↓ (or by pressing the TAB key [Tabulator]) and enter 'Rack", press 2x TAB, the type 'SR12" is inserted; complete with ↵:



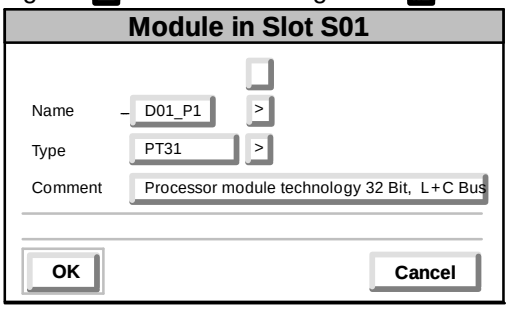
G A rack screen with 12 slots (SR1..SR12) is then displayed.

A **processor module** is configured as follows:



G Select the column 'S01" and line '3", in the RACK screen, using the left-hand mouse key.

G The module screen is displayed:
Go to the button > in the line 'Type" with 3x TAB, select the menu window 'Module Library" using ↵ and move to the " at the end of the line 'IBSLIB Standard Modules" using the mouse. A further window containing module types is automatically popped onto screen. Select 'P32 Processors" using the mouse and acknowledge with ↵. A window with all 32 bit processors is then displayed. Go to the line 'PT31" using the ↓ and acknowledge with ↵. The PT31 is then inserted into the module screen:

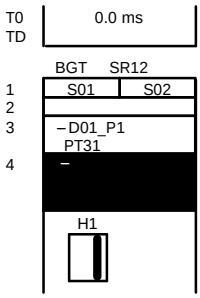


Use 3x TAB to go to the button 'OK" and insert the PT31 screen into the rack using ↵.

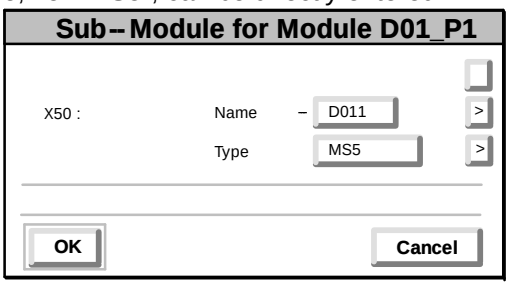
Each processor module requires one **program memory sub-module MSxy**. The configuring is implemented as follows:

- Select the empty panel in the column "S01/S02" and line "4" in the PT31 screen using ↵.

  A sub-module window is popped onto screen. Move to the empty panel for the name by pressing 2x TAB.



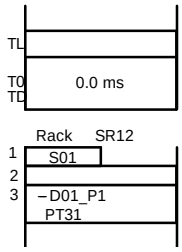
Enter the name "D011" and then branch to the type panel with TAB.
Running through the menus, analogous to the selection of the processor type, is also possible here or the type, i.e. "MS5", can be directly entered:



Press ↵ to insert the sub-module into the PT31 screen.

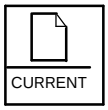
Entry of the **basic sampling time T_0** (cycle time):

- Select the empty panel in the column "S01/S02" and line "T0/TD", in the PT31 screen, using ↵. A window is popped onto screen. Use ↵ to first eliminate the "?" in the panel in the upper right-hand corner of the screen.



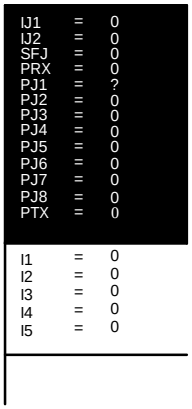
Press 8x TAB (or direct selection via the mouse and ↵) to go to the panel for the basic sampling time. Enter "1". Use ↵ to accept the entry and insert into the PT31 screen.

Function packet name PJ_x and **sampling times $T_1..T_5$** definition:

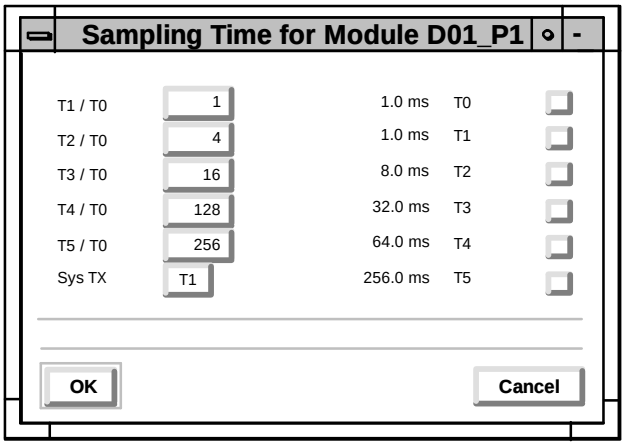
- The entries are made in another menu, the "Module Parameter Drawing". Go to the upper menu strip using the mouse and select the button 

with ↵. A new menu strip is popped onto screen. Go to the line *"Module Parameters..?"* with ↵ and acknowledge using **2x** ↵. The module parameter drawing is then popped onto screen.

- Enter function packet name:
  Select the 1st. panel (IJ1=0 ... PTX=0) in the 2nd. column using . The FP table is then presented. Go to the ? panel (line PJ1) using the mouse or TAB, enter the name, i.e. "RUN" (automatic upper case characters) and eliminate the "?" with the keys and *(backspace)*.
  Use to insert these entries into the module parameter drawing.



- Define the sampling times by using the mouse to go to the line "? T1=?" and select with . The sampling time menu window is then popped onto screen.
  The sampling times T_1..T_5 are then to be defined in relation to the previously configured basic sampling time T_0:
  All possible definitions are popped onto screen with . Go to "1" using the mouse pointer and acknowledge (accept) with . Subsequently use to go to "?" in the right-hand panel and eliminate with . Return with and TAB to go to the line for "T2/T0" and repeat the same procedure for the sampling time T2. This time enter the number *"4"*. Repeat for T3, T4 and T5 with the entries *"16"*, *"128"* and *"256"* (and eliminate all "?").
  Finally eliminate the "?" in the Sys–TX line (use the mouse to position and eliminate with .



  Branch to OK using TAB and accept all the entries into the module parameter drawing with .
- No further "?" must be visible in the module parameter drawing !

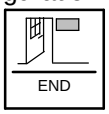
To complete the master program processing, **the cover sheet must be created**, in which a version date must always be entered:

- This entry is implemented in another menu, the "Text Panel..?". Use the mouse to go to the upper menu strip and select the button CURRENT with . A new menu strip is popped onto screen. Go to the line "Text Panel..?" with and acknowledge with . The text panel will then be presented.
- **Warning: the position of the mouse pointer is decisive for all entries in this menu !**
  The mouse pointer always enables the specific panels for the keyboard entries.

- Position the mouse pointer in the panel with the brown background, behind the panel "Date", and enter the date, i.e. "30.11.95" (do not forget to eliminate the "?"):

	Date	05.09.95
	Drawn	ASI 1 R
	Appr.	
Name	Std.	

  All further entries are optional, i.e. not mandatory.
- Accept the entries by positioning the mouse pointer on the panel "OK" and acknowledge with .

The module parameter drawing is now popped onto screen once again. The processing of the master program is now complete for this hardware configuration. **Exit the processing** by placing the mouse pointer in the upper menu strip and select the button 

with . All entries are acknowledged and saved and the processing exited with **2x** . The screen will now display the basic dialog.

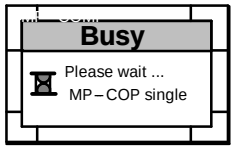
Subsequently **compile the master program** to check all the entries:

- The project quicky and the master program single are still selected. This can be seen by the green background in both panels.

  Select the button 

with to compile. The line "Selected MP" appears below the button.

  The compilation is started with .

  A window 

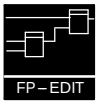
is temporarily displayed.

  The result of the compiler run is shown in the right-hand lower box "Message Output Basic Dialog". If all definitions have been correctly made, then the following appears in the black panel:

```
****          0 Errors   0 Warnings
```

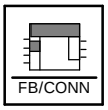
  The master program processing has then been successfully completed.

The technological functions and the communication are configured in the **function packets** using the available function blocks:

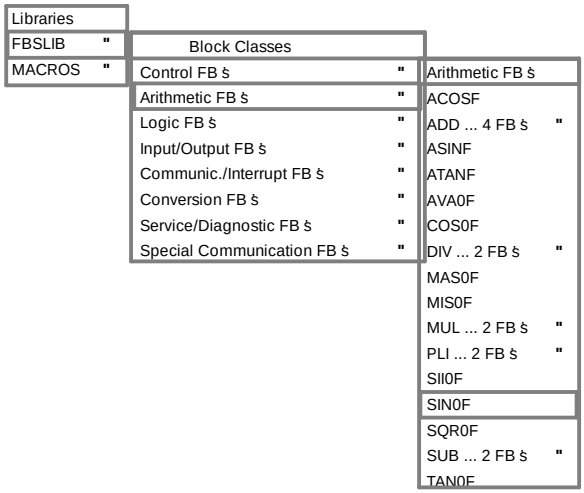
- Define the function packet names in the master program. This example only shows how to configure the function packet with the name "RUN". It is already created as a light brown button in the first column. Move the mouse pointer to the button "RUN" to select the function packet. Activate with (button becomes green).
- Processing the function packet requires a different screen layout. Place the mouse pointer on the button 

and select with . The new screen layout, the function packet editor, is activated with (menu item "Configure").

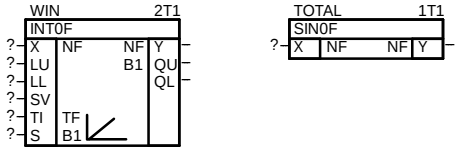
Configuring **function blocks (FB)**:

- Select 

in the menu line with . Go to the line "Invoke" using # and invoke with ; . Place the mouse pointer in the large window in the middle of the screen and activate the sheet for "Invoke block" using . A window is displayed. Switch to the empty panel for the FB name using TAB and enter the name, i.e. "TOTAL" (upper case characters automatic).
- Go to the next empty panel (FB type) using TAB and **either** directly enter the type.
  Or: place the mouse pointer on the & panel and select with . Place the mouse pointer in the newly popped up library window on the line "FBSLIB " " and select with . Surf through the menus until the desired FB is selected, i.e. "SIN0F":

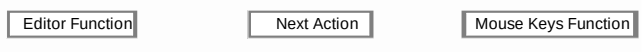


  After "SIN0F", the type is automatically accepted. The default "T1" (see window) is retained as the sampling time.
  Place the mouse pointer on the panel "OK" and complete the entry with . The window is then closed and the frame of the block lies at the mouse pointer. Move the mouse pointer to the position where the block is to be placed and insert the block with the **right-hand** . The block is then displayed.
- The "Invoke block" is continued with the . This procedure is also used for configuring the block type "INT0F" (class: control FB's) with the name "WIN". Placing the block using the mouse pointer then results in:



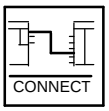
- "Invoke block" is completed using the **middle** . The upper graphical menu line can now be reselected.
- **One small tip** regarding the screen layout and the allocation of the mouse keys in the menu 

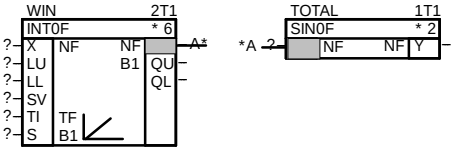
is as follows:
  Information is located below the menu line at the upper edge of the screen layout:

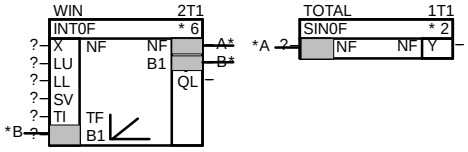


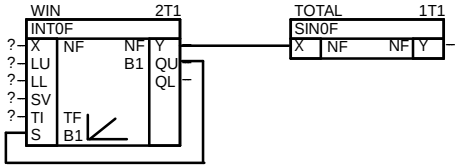
  The *editor function* defines which function is currently selected, i.e. "Invoke Function Blocks". The *next action* implies the actions to be activated next and the right-hand panel *mouse keys* indicate the allocation of the three mouse keys.

Connecting Function Blocks:

- Select 

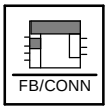
in the menu line using and invoke with .
- The Y output of the INT0F block is to be connected to the X input of the SIN0F block. In addition the QU output of the INT0F is to be connected to the S input of the INT0F.
- Bring the mouse pointer in the large window into the middle of the screen and select the Y output of the INT0F with (click directly onto "Y"). Drag the mouse pointer to the X input of the SIN0F and create the connection with : 

- Position the mouse pointer on the QU output, for the 2nd. connection, and select with . Drag the mouse pointer to the S input and create the connection with : 

- The block connect is completed with the **middle** and the connections are then displayed: 

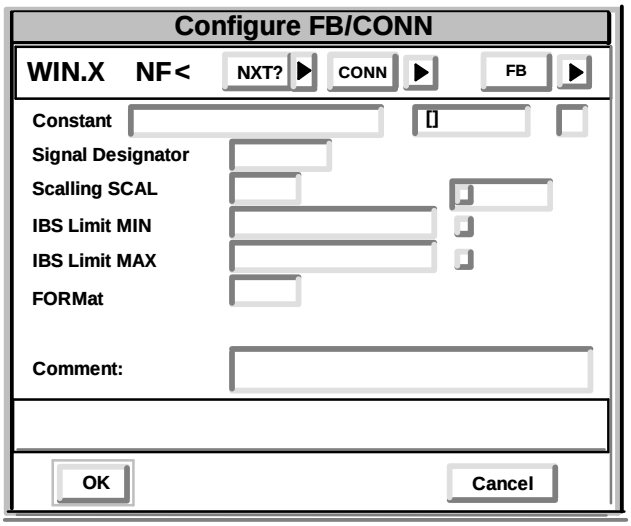
  The upper graphical menu line can now be reselected.

Setting the Function Block Inputs:

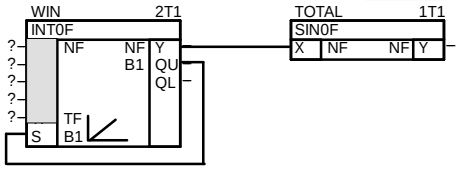
- Select 

in the menu line with and go to the line "Parameters" with TAB. Invoke the menu with .
- Parameters should only be set at inputs displaying a "?" (only at INT0F) and only in sequence starting from the top with "260%", "359.64%", "0%", "0%" and "1".

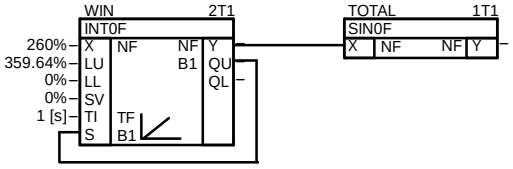
G Bring the mouse pointer in the large window to the middle of the screen and select the X input of the INTOF with (directly click on the X). A parameter window is popped onto screen:



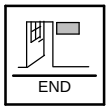
G Select the empty panel for the constants using and enter '260 ". Then use on " of NXT? to switch to the next connector and correspondingly repeat the procedure.
The parameters of all the other inputs are set in the same manner. After setting the TI input, the mouse pointer is placed on the button OK in the window and all entries are accepted with :



G Use the **middle** (the mouse pointer must be **inside** the sheet) to exit the block parameters and display all the input values:



The upper graphical menu line can now be reselected.

All the necessary steps for processing this function packet have now been completed. Use the mouse to select the button 

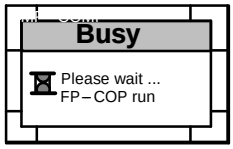
with to **exit** the processing. Press **2x** to acknowledge and save all entries and terminate the processing.
The screen for the basic dialog is then displayed.

Check all entries by subsequently **compiling the function packet**:

G The project quicky, the master program single and the function packet run are still selected. This can be seen by the green background of the three panels.

Select the button 

with to compile. The line 'Selected FP " appears below the button.

The compiling is activated with . The window 

is temporarily displayed.

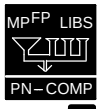
The result of the compiler run is displayed in the right-hand lower box "Message Output Basic Dialog". If all definitions have been correctly implemented, then the following appears in the black panel:

```
****          0 Errors   0 Warnings
```

The function packet processing has then been successfully completed.

The next step is to **compile** the total **processor program**:

- The project quicky, the master program single and the processor D01_P1 are still selected. This can be seen by the green background of the three panels.

  Select the button 

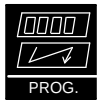
  with to compile. The line "Selected PN" appears below the button.

  Activate compiling with . The result of the compiler run is displayed in the right-hand lower box "Message Output Basic Dialog". If all definitions have been correctly made, then the following appears in the black panel as a completion message:

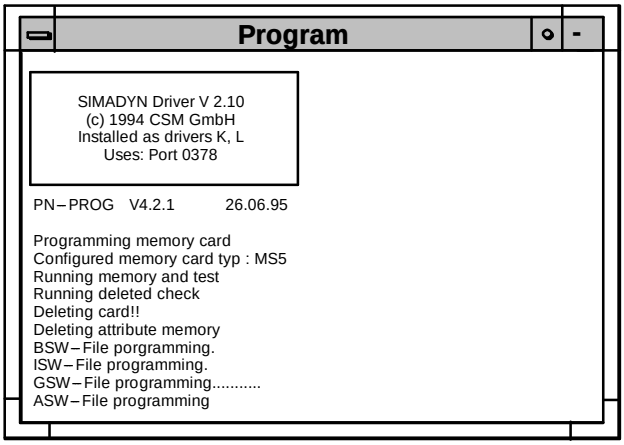
```
****          0 Errors   0 Warnings
```

Now the last step, **programming the program memory sub-module MSxy**:

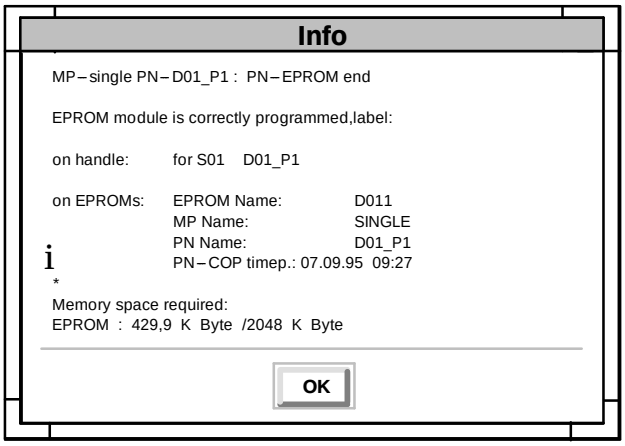
- First insert the program memory sub-module into the corresponding slot of the parallel prommer.
- The basic dialog screen still shows that the project quicky, the master program single and the processor D01_P1 are selected. This can be seen by the green background of the three panels.

  Select the button 

  with to compile. Three lines are displayed below the button, the line "Selected PN" is the automatic preselection.
  Activate the compiler with , whereby the following window is initially popped onto screen and sequentially filled:

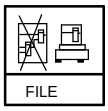


The window is automatically closed and processing continued by opening the following window:



The processor program has now been successfully compiled. The programming and the configuring phase are then completed with ↵.

Exiting the STRUC configuring software and **powering down the STRUC-PC**:

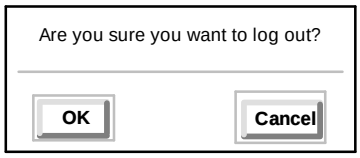
G Select the left-hand button 

with ↵ in the basic dialog screen. A menu strip is popped onto screen.

Use # to go to the line "Exit STRUC...." and acknowledge with ↵ (or place the mouse pointer on this location and acknowledge with ↵). A further window for acknowledging is popped onto screen.
Acknowledge with ↵.
The basic dialog screen is closed. Select "File", on the UNIX shell in the upper left-hand corner with ↵

File Edit View Help

and go to the last line "Exit" using #. Acknowledge with ↵.

The following window 

is acknowledged with ↵.

Finally power down the STRUC-PC in a controlled manner. First switch the console to console 1 by simultaneously activating the keys CNTRL, ALT and F1. Then enter

login: ***root*** ↵
password: ***root*** ↵

The PC will now run until a "#" appears, enter "***init 0***" (init space zero) and ↵. Wait until the screen

** Press Any Key to Reboot **

appears, then the **PC can be powered down**.
If an immediate restart up is desired (i.e. no power down), then press any key to activate.