SIEMENS

SIMATIC S5

COM 525
Programming Package for the Communications Processors CP 524 and CP 525 (S5-DOS)

Manual

Volume 2/2

Order No. 6ES5998-1DB21
Release 08

We have checked the contents of this manual for agreement with the hardware and software described. Since deviations cannel be precluded entirely, we cannot guarantee full agreement. However, the data In this manual are reviewed regularly and any necessary corrections included h subsequent editions. Suggestions for improvement are welcomed

Technical data ● bjea to change

Siemens Aktiengesellschaft

6ES5 998-1DB21
Elektronikwerk-Karlsruhe
Printed in the Federal Republic of Germany

Contents

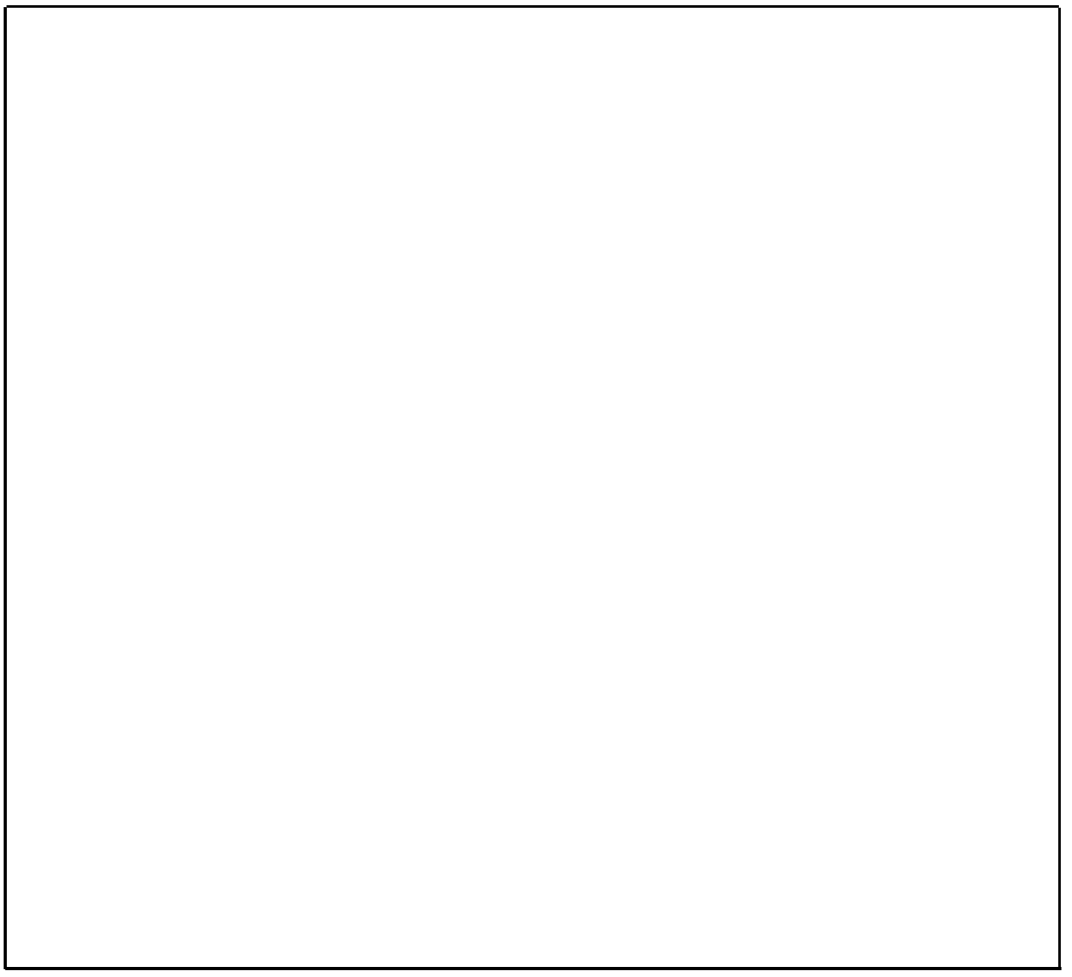
Volume 1

Volume 2

SIEMENS

SIMATIC S5
Computer Link with RK 512

Example of Application **C79000-B8576-C540-05**



CONTENTS

1 Introduction

This is an example of an application to illustrate the use of the communications processor CP 525 for a point- to-point computer link. When you work through and implement this example you will have a finished CP 525 user program in which all the most important functions of the computer link have been implemented.

First, the required hardware settings are explained. Then, not only the generation of a CP 525 user program with the COM 525 programming software, but also the corresponding STEP 5 user program for the CPU of the programmable controller are explained.

For this example you require the following hardware:

two CP 525 modules with RAM memory submodules
- one CP 525 <--> CP 525 cable connector (TTY or V.24)
- one PG 685 programmer
- one cable connector PC <--> CP 525
- one S5-135U programmable controller
- one R processor with RAM memory submodule
  one cable connector PG <--> R processor
- one S5-150U programmable controller
- one cable connector PG <--> AS 511

and if possible

- one digital input module (24 V, at least 8 channels)
- one digital output module (24 V, at least 8 channels)
- one SIMATIC S5 simulator

the following software is also required:

- the S5 DOS programming package COM 525
- the STEP 5 **basic package for the PG 685 programmer**
  (supplied with the programmer)
- handling blocks for the R processor
- handling blocks for the S5-150U

2 Aims

First you set up the hardware. Then you install the software necessary for writing your user programs.

The first step is to send data from a data block in one PC to a data block in the other PC. Then data is fetched from the data block of the other PC and stored in a data block. The next step is to send a flag word to a data word on the partner cyclically. The reception of this data sent cyclically is then disabled by a coordination flag.

To demonstrate the PSEUDO READ/WRITE function, a PSEUDO WRITE job is programmed to send an input byte to a data word on the partner.

You implement the reading of the error message area in the SYSTAT.

Both the reading and setting of the CP 525 hardware clock is demonstrated and the synchronization of several CP hardware clocks explained.

At the end of the example there is an explanation of the information and documentation functions available with the CPM 525 programming software.

The individual tasks are restricted to what is absolutely necessary and in some respects (though not all) build on one another.

The DIRECT jobs (i.e., jobs which trigger job processing) are entered in the CP 525 queue, which holds 10 jobs, and are processed in the order in which they were entered. Jobs already in the queue are not entered a second time. Since in this example of an application, there are less than 10 DIRECT jobs programmed, a DIRECT job will be processed whenever it is triggered. Interlocking of jobs is therefore not necessary in this example.

3 Hardware

In the course of this example, a point-to-point link between two SIMATIC S5 programmable controllers is set up, The two programmable controllers used are an S5-135U with R processor and an S5-150U.

In principle, the procedure for S5-115U, 135U and 150U is identical. Nothing whatsoever changes in the CP 525 user program. The handling blocks have different block numbers with the different PCs, however, they are identical in terms of programming. Only the handling of interprocessor communication (IPC) flags is different in the various PC'S (see Section 8).

Plug in the Reprocessor into one of the slots (11, 19, 27 or 35) in the S5-135U. Carry out an overall reset of the CPU and switch the mode selector to STOP.

Plug in the central processor and memory modules in the correct slots in the S5-150U frame. Carry out an overall reset of the S5-150U.

On one CP 525 set the module address 2; i.e., insert jumper 7-10 on jumper block 16. On this CP 525 enable the IPC flag bytes 0 to 31. To do this, insert jumper 8-9 at jumper block 25 on this module. Plug this CP into the S5-135U. The slots available for the CP are 11, 19, 27, 35, 43, 51, 59 and 67.

On the other CP set the module address 0, and disable all the IPC flags (i.e., do not insert jumpers, either at jumper block 16 or block 25). Plug the CP 525 into the S5-150U. The slots available are 107, 115, 123 and 131.

Connect the upper device interfaces (IF 1) of both CPS using the corresponding cable connector. Whether your transmission is carried out with TTY or V.24 signals depends only on the type of cable connector used. Both signals are always available at the port of the CP 525. If the transmission distance exceeds 20 meters, you must use TTY signals.

Set the module address) on the digital input module and the digital output module. This makes input byte IB) and output byte QB 0 available. Plug in the modules in any free slots on the S5-135U. Connect the simulator to the 1/0 modules.

Assignment of input byte IB 0:

Bit 0 : not used
Bit 1 : trigger SEND 1 PC job (see Section 5)
Bit 2 : trigger FETCH 2 PC job (see Section 6)
Bit 3 : trigger PSEUDO-WRITE PC job (see Section 9)
Bit 4 : reset error message area in the SYSTAT (see Section 10)
Bit 5 : trigger date/time synchronization (see Section 11.3)
Bit 6 : trigger setting of date/time (see Section 11.2)
Bit 7 : reset coordination flags (see Section 8)

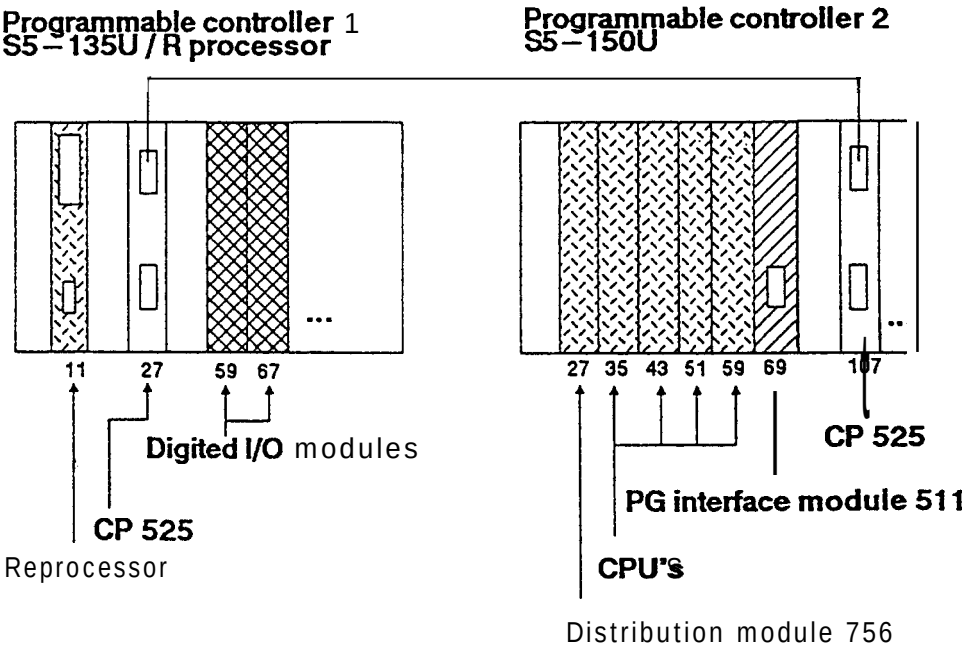
Assignment ofoutputbyte QBO:

Bit 0 : SYNCHRON parameter assignment error (see Section 5)
Bit 1 : error in SEND 1 PC job (see Section 5)
Bit 2 : error in FETCH 2 PC job (see Section 6)
Bit 3 : error in PSEUDO-WRITE Pc job (see Section 9)
Bit 4 : error reading the error message area SYSTAT (see Section 10)
Bit 5 : error reading the CP 525 clock (see Section 11.1)
Bit 6 : error setting the CP 525 clock (see Section 11.2)
Bit 7 : error synchronizing the slave clocks (see Section 11.3)

If you have no I/O modules orno simulator available, then insteadof input byte IB 0, use flag byte FY 0. These flags can be set and reset with the STEP 5 online function CONTROL VARIABLE.

In the remainder of this section the S5-135U, which includes the digital 1/0 modules, is called PC 1 and the S5-150U is called PC 2.

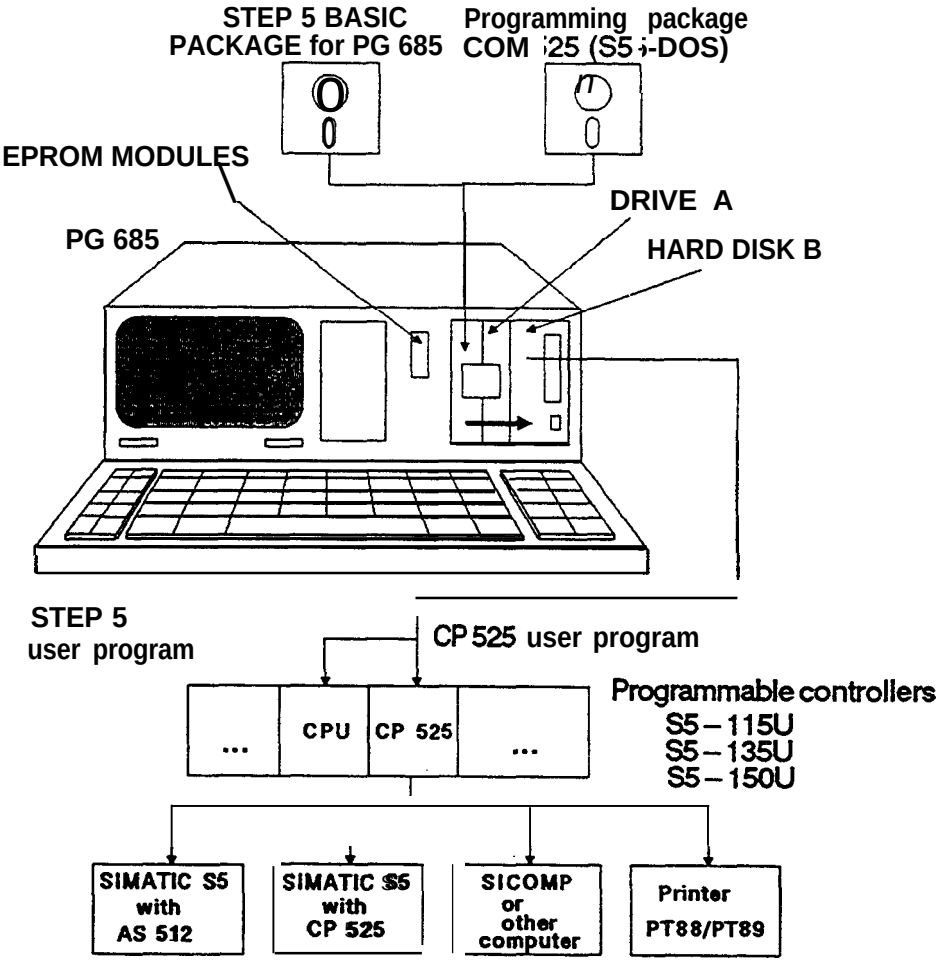
SYSTEM CONFIGURATION:



Connect the PG 685 using the corresponding cable connector to device interface (II? 2) of the CP 525; or the programming interface of the R processor of the 135U; or the programming device interface module AS 511 in the 150U, depending on which module you wish to program. When programming the CP 525, you must switch its mode selector to PGR.

Instead of a PG 685, you can also use a PG 635, PG 675 or PG 695.

GENERATING PROGRAMS:



4 Programming Software COM 525

This chapter explains how to set up the software for programming the computer link jobs based on the example.

4.1 Installing COM 525 on the PG 685

A PG 685 programmer in which the STEP 5 basic package has already been installed must be available.

Go to drive B: user number 0 by entering

```
0:
```

Copy the contents of the three COM 525 floppy disks with

```
PIP B:=A:*.*[RV]

Option V= verify copying
       R = copy SYS files
```

onto the hard disk user number 0.

Assign the system and read only attributes to the files with

```
SET S5?EC?5X.CMD[SYS RO]
SET S5PEP05X.CMD[SYS RO]
SET COMLIB*.525[SYS RO]
```

You can work with COM 525 under any user number.

Exit user area 0 which should be reserved for system files, with

```
n:      (n = required user number).
```

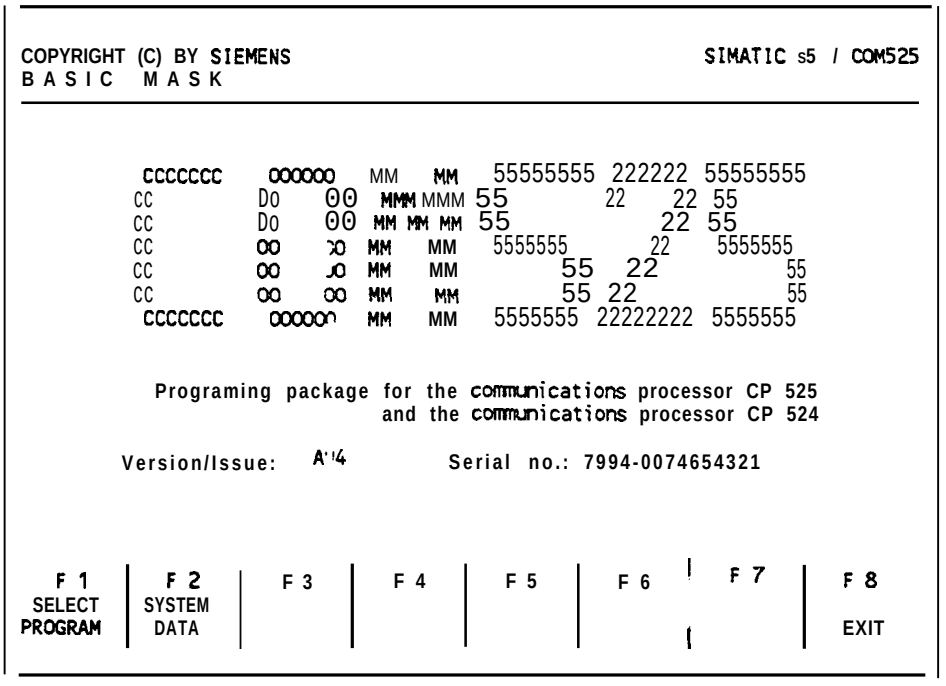
4.2 Calling COM 525

By entering

S5

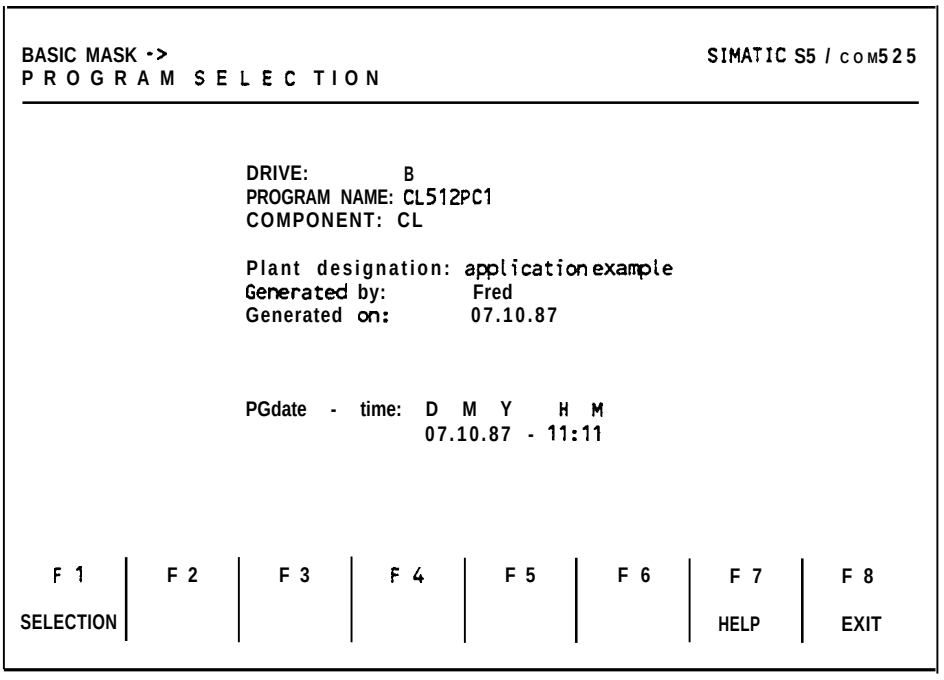
you call the package selection mask. Place the cursor in the line **'COM 525** programming package for the CP **525/524'** and using the function key F1 (PACKAGE) select the COM 525 programming software for the CP **525.**

The COM 525 basic mask then appears:



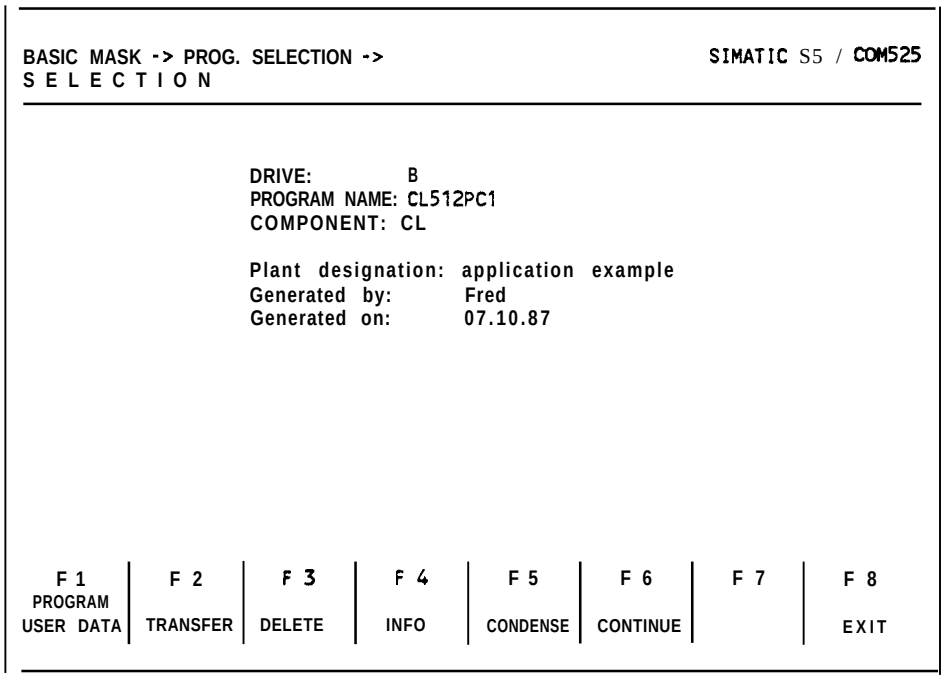
4.3 CP 525 User Program for PC 1 (135U)

Press F1 (SELECT PROGRAM) *to* call the mask 'PROGRAM SELECTION'.



As drive, select the hard disk "B". Enter a program name (in this case "CL512PC1") and the required component "CL" for computer link. The plant designation is called 'application example" in this case. Under "generated by" enter your own name. The date and time of the PG 685 hardware clock can be set by overwriting the input fields.

In the 'SELECTION' mask, which you call with F1, you can select the basic COM 525 functions.



By pressing F6 (CONTINUE) you call the second set of function keys (second menu).

By pressing F6 you return to the basic set of function keys.

When programming with COM 525 you generate the user program first on floppy or hard disk (storage medium FD), and then transfer it to the CP 525 user memory and then test it.

The S5 DOS file on the FD is automatically named program name.525, in this case CL512PC1.525.

4.3.1 Interpreter and Procedure

Without an interpreter and procedure no data exchange is possible between the CP 525 and a partner.

4.3.1.1 Copying from COMLIB into the User Program

The interpreter and procedure must first be copied into the user program. They are located in the library COMLIBn (n = version number) which is supplied on a separate floppy disk with the COM 525 software.

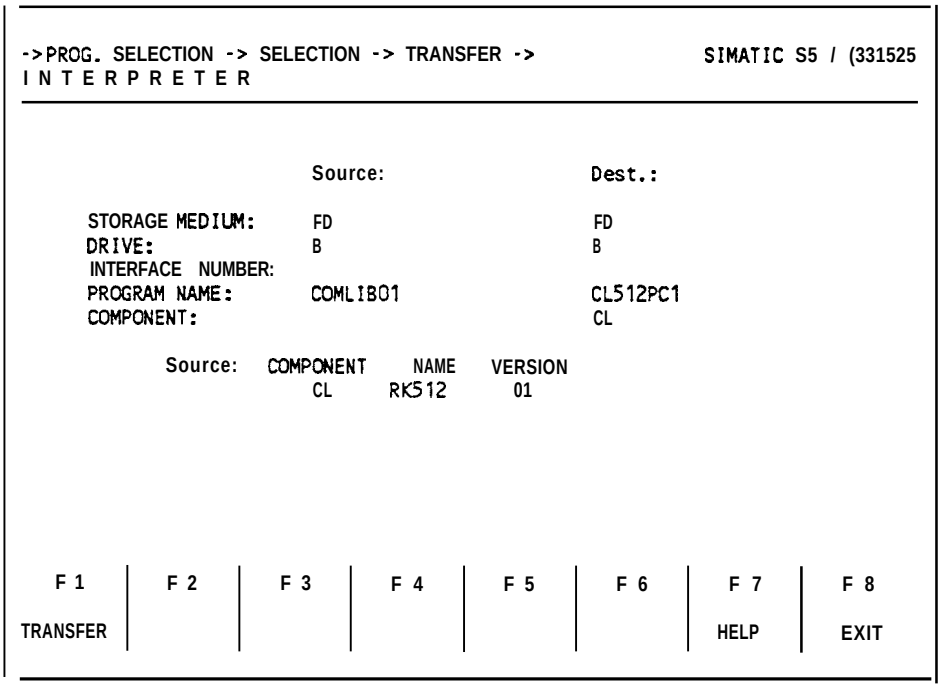
Ress F2 (TRANSFER) and F5 (FD->FD) in the 'Selection mask to call the function for transferring from FD to FD. Press F7 (HELP) in the 'TRANSFER' mask and enter drive "B" and program name "COMLIBn" (n = version number) as source. The destination is automatically the program CL512PC1 selected in the 'PROCRAM SELECTION' mask.

->PROG.SELECTION-> SELECTION -> TRANSFER SIMATIC S5 / COM525

	Source:	DeSt.:
STORAGE MEDIUM:	FD	FD
DRIVE:	B	B
INTERFACE NUMBER:		
PROGRAM NAME:	COMLIB01	CL512PC1
COMPONENT:		CL
Plant designation:	standard library	Application example
Generated by:	GW Karlsruhe	Fred
Generated on:	03.09.87	07.10.87

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
		INTER-PRETER	PROCEDURE		JOB BLOCK	HELP	EXIT

Press F3 to call the 'INTERPRETER' transfer mask. Press F7 (HELP) in this mask to select the interpreter "RK512" (component "CL") to be transferred F1 (TRANSFER) starts the transfer. The successful transfer is displayed with 'MESS.002: Completed'.



With F8 (EXIT) you return to the 'TRANSFER' mask.

With F4 you call the 'PROCEDURE' transfer mask. Select procedure "P3964R" (procedure 3964 with block check) using F7 (HELP). The procedureistransferred the same way that the interpreter is.

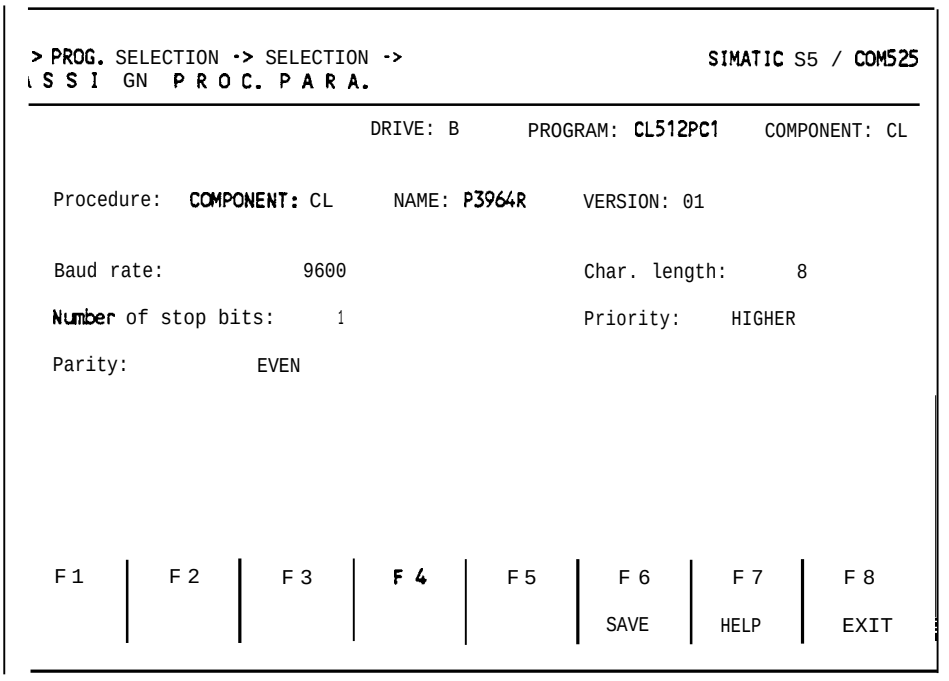
Press F8 (EXIT) twice to return to the 'SELECTION' mask from the 'PROCEDURE' mask.

4.3.1.2 Assigning Parameters to the Procedure

With F6 (CONTINUE) call the second set of function keys in the 'SELECTION' mask,

F2 branches to the 'ASSIGN PROC. PARA.'. The procedure P3964R belonging to your program CL512PC1 is displayed with its version number. Select the data rate "9600" and priority "HIGHER".

In PC 2 (150U) the priority must be programmed as LOWER. If there is an initialization conflict, i.e., both CP 525s attempt to transmit simultaneously, the CP 525 in PC 2 (150U) desists.



Save the procedure parameters on the hard disk with F6.
'MESS.003: Saved!' indicates that the information has been saved.
With F8 (EXIT) and F6 (CONTINUE) return to the 'SELECTION' mask
(first set of function keys).

No parameters can be assigned to the RK512 interpreter.

4.3.1.3 Transferring to the CP 525 User Memory

The interpreter and procedure must still be transferred to the user memory of the CP 525 in PC 1 (135U). The CP 525 must be connected to the PG 685 and its mode selector must be set to "PGR" (programming).

Using F2 (TRANSFER) and F3 (FD->CP) in the 'SELECTION' mask call the function for transferring from FD to CP 525. In the 'TRANSFER' mask enter the CP 525 interface "1" as the destination. Make the transfer with F3 (INTERPRETER) and F4 (PROCEDURE).

```
->PROG. SELECTION -> SELECTION ->                          SIMATIC S5 / COM525
TRANSFER

                          Source:                Dest.:

      STORAGE MEDIUM:     FD                     CP525
      DRIVE:              B
      INTERFACE NUMBER:                          1
      PROGRAM NAME:       CL512PC1
      COMPONENT:          CL

      Plant designation:  application example
      Generated by:       Fred
      Generated on:       07.10.87

ERR.724:  Interpreter/procedure not present
| F 1 | F 2 | F 3 | F 4 | F 5 | F 6 | F 7 | F 8 |
| COLD RESTART | TOTAL PROGRAM | INTER- PRETER | PROCEDURE | | JOB BLOCK | HELP | EXIT |
```

The message ERR.724 appears only if you have not yet loaded data in the CP.

'MESS.002: Completed!' shows that the transfer is completed. When the interpreter and procedure are transferred to the CP 525 memory you must cold restart the CP 525 with F1 (COLD RESTART). If you switch the CP 525 mode selector to RUN, the red LED of interface 1 (IF 1) must go out,

Press F8 (EXIT) twice to return to the 'PROGRAM SELECTION' selection. Press F8 (EXIT) again so that the prompt 'ACK.001: terminate program?' appears on the PG 685 screen. Confirm that you wish to terminate the program with F1 (YES) and exit the COM 525 programming package by again pressing F8 (EXIT),

Press F8 (BACK) and confirm the prompt 'EXIT S5 COMMAND INTERPRETER?' with the enter key to return to the PCP/M Operating system,

All the preparations are now made on the CP in PC 1 for programing the computer link functions,

4.4 CP 525 User Program for PC2 (150U)

Copy the CP 525 user program for PC 1 into a file for the CP 525 user program for PC 2 with

PIP CL512PC2.525=CL512PC1.525

The programs differ from each only in the priority of the procedures. By typing in

S5

call the package selection mask. Using the cursor select the programming package COM 525 and press F1 (PACKACE).

From the 'BASIC MASK', F1 (SELECT PROCRAM) leads to the 'PROGRAM SELECTION' mask.

BASIC MASK ->
P R O G R A M S E L E C T I O N

SIMATIC S5 / COM525

DRIVE: B
PROGRAM NAME: CL512PC2
COMPONENT: C L

Plant designation: application example
Generated by: Fred
Generated on: 07.10.87

PGdate - tire: D M Y H M
07.10.87- 13:13

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
SELECTION						HELP	EXIT

Press F7 (HELP) to select the program name "CL512PC2" in this mask.

For the component, plant designation and "generated by" no new entries are required. They are the same as those for program C1512PC1.

4.4.1 Interpreter and Procedure

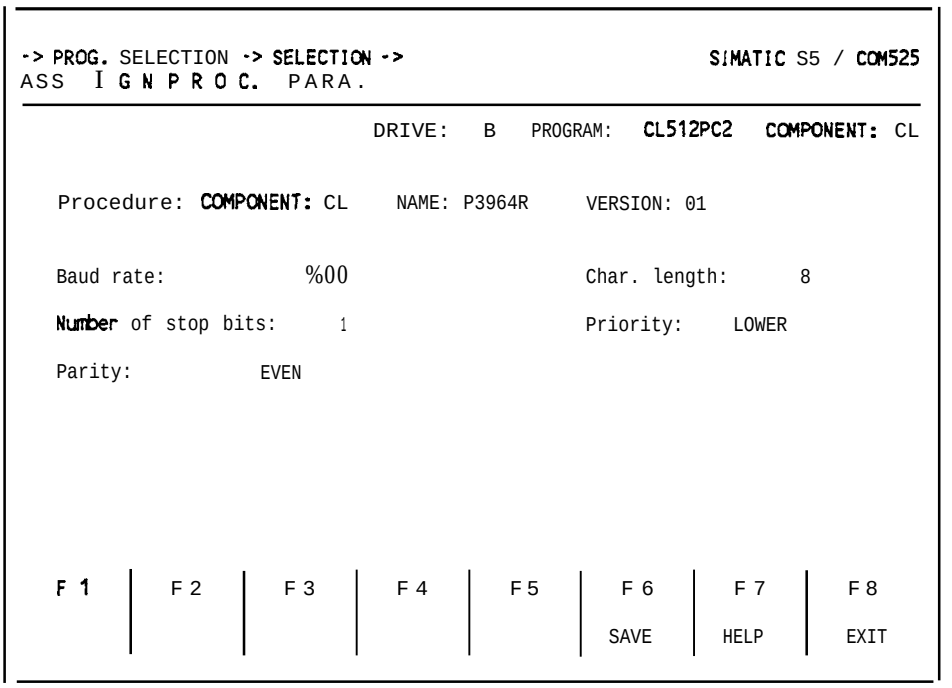
The interpreters in PC 1 and PC 2 are identical. The procedures in both communications processors must have the same data rate but different priority.

4.4.1.1 Changing the Procedure Parameters

Press F1 to call the 'SELECTION' mask and then F6 (CONTINUE) to call the second set of function keys.

F2 branches into the mask 'ASSIGN PROC. PARA.'; the procedure belonging to program CL512PC2 is displayed along with the version number. Leave the baud rate of 9600 as it is but change the priority using F7 (HELP) to "LOWER",

In PC 1 (135U) the priority was programed as HIGHER. If there is an initialization conflict, i.e. , both CP 525s attempt to transmit simultaneously, CP 525 in PC 2 (150U) desist.

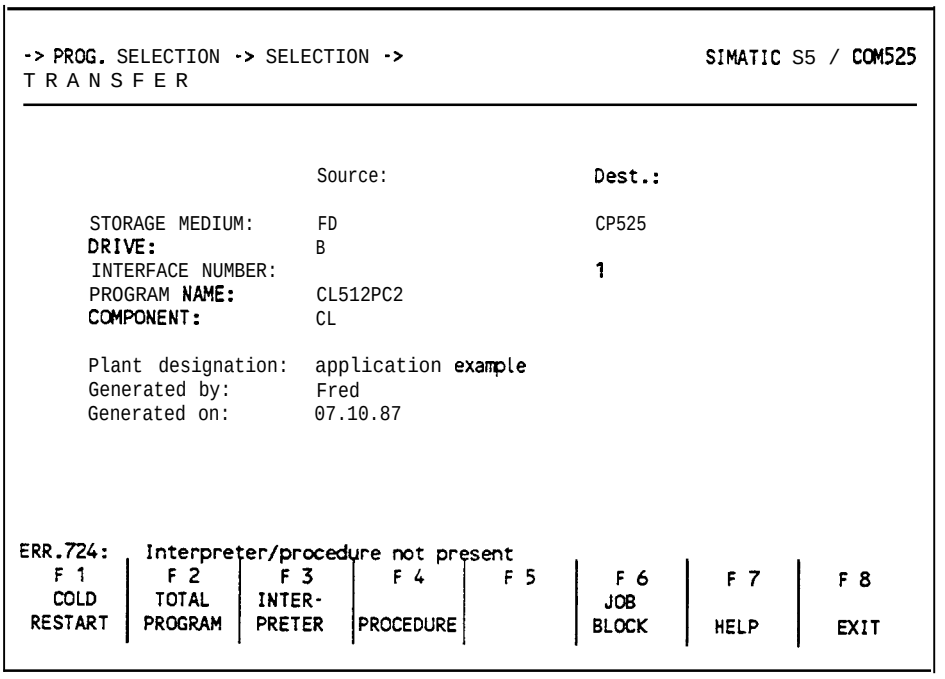


Save the procedure parameters on the hard disk with F6. 'MESS.003: Saved!' indicates that the information is saved. With F8 (EXIT) and F6 (CONTINUE) return to the '.SELECTION' mask (first *set* of function keys).

4.4.1.2 Transfer to the CP 525 User Memory

The interpreter and procedure must still be transferred to the user memory of the CP 525 in PC 2 (150U). The CP 525 must be connected to the PG 685 and its mode selector must be set to PGR (programming).

Press F2 (TRANSFER) and F3 (FD->CP) in the 'SELECTION' mask to call the function for transferring from FD to CP 525. In the 'TRANSFER' mask enter the CP 525 interface "1" as the destination. Make the transfer with F3 (INTERPRETER) and F4 (PROCEDURE).



The message ERR.724 appears only if you have not yet loaded data in the CP.

'MESS.002: Completed!' shows that the transfer has been completed. When the interpreter and procedure have been transferred to the CP 525 memory you must cold restart the CP 525 with F1 (COLD RESTART). If you switch the CP 525 mode selector to RUN the red LED of interface 1 (IF 1) must go out.

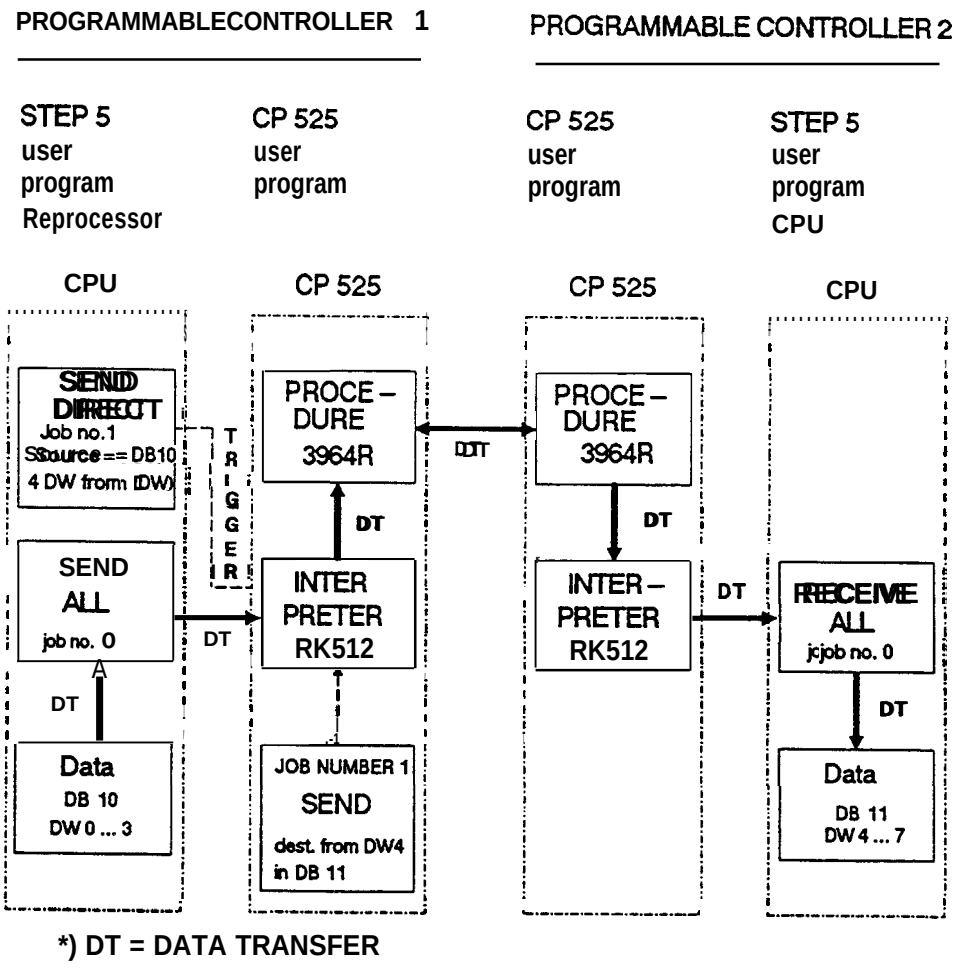
You can now program the computer link functions on the CP 525 in Pc 2.

5 SEND **Job from Data Block to Data Block**

Begin with a SEND job from PC 1 (135U) to PC 2 (lSOU).

Four data words of data block DB 10 (from data word DW 0 onwards) in the R processor are to be transferred to data block DB 11 (from data word DW 4) in the S5-150U. You can see that the source and destination specified can be completely different.

For this the following program structure is required:



*) DT = DATA TRANSFER

The interpreter and procedure are already loaded in the user memories of the two CP 525s (see Sections 4.3.1.3 and 4.4.1.2).

The SEND job needs to be programmed only on the CP 525 of the active PC. Active means that this PC initiates the data exchange and that you need to program a SEND job only for the CP 525 in PC 1 (135U).

The SEND job is initiated when the CPU calls the handling block SEND DIRECT with the job number 'n'. Select job number 1. The same job number 'n' is also given to the corresponding job on the CP 525. This job contains information about the destination of the data in PC 2 (150U).

CP 525 fetches the data only when called by the handling block SEND ALL. This call transfers the data from the CPU/DB 10 to the CP 525 in PC 2.

The CP 525 in PC 2 (150U) receives the data and calls the RECEIVE ALL to store it in the destination indicated by PC 1.

The SEND AIL (also known as SEND 0) has the job number A-NR 0. It is called unconditionally in every program cycle and checks whether the CP 525 (dual-port RAM) requires data. If it does, the SEND ALL fetches it from the specified source and transfers it to the CP 525, If it does not, the function block is exited and the cyclic program continues.

The RECEIVE ALL (also known as RECEIVE 0) has the job number A-NR 0. It is called unconditionally in every program cycle and checks whether the CP 525 (dual-port RAM) needs to transfer data to the CPU. If it does, the RECEIVE ALL stores the data in the destination address specified by the CP 525. If it does not, the function block is exited and the cyclic program continues.

5.1 CP 525 User Program for PC 1 (135U)

Switch the mode selector on the CP 525 to PGR.

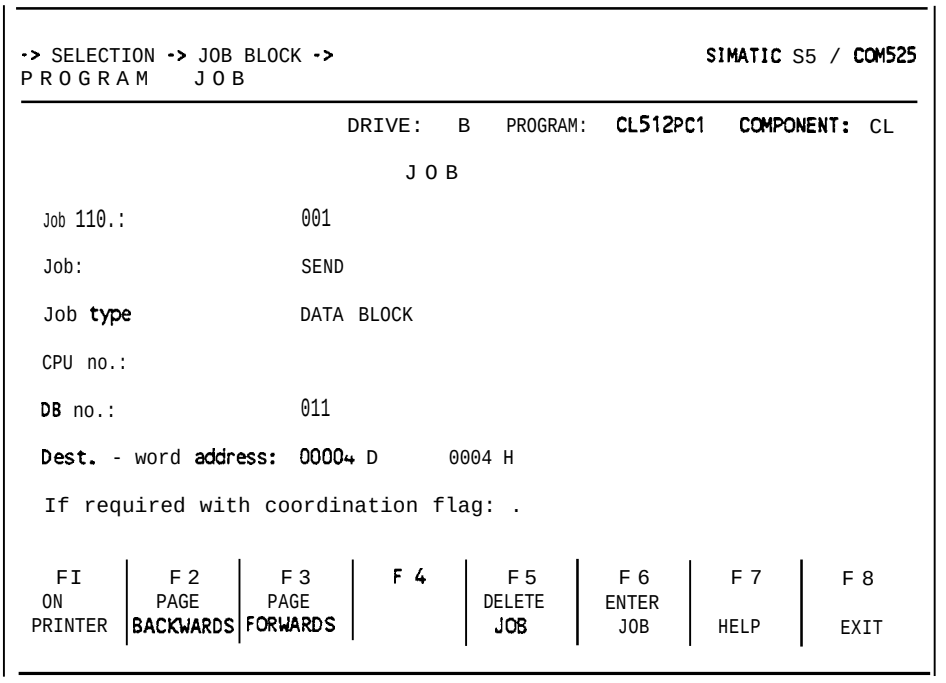
Press F1 (PROGRAM USER DATA) in the 'SELECTION' mask to call up the 'JOB BLOCK'. Enter the job number with which this job is to be triggered by the CPU, Use the default 'job no. :' "1".

Press F5 (PROGRAM JOB) to call the 'PROGRAM JOB' mask.

Use the defaults 'job:' "SEND" and 'job type' "DATA BLOCK". The 'dest. - word address' is "4" D (decimal) and "4" H (hexa-decimal); the data block number is "11". No entry is necessary for the CPU number since you are not concerned with multi-processor operation. Do not specify any coordination flags.

Press F6 (ENTER JOB) to write the job to the RAM memory of the PG 685. With F8 (EXIT) you return to the 'JOB BLOCK' mask. With F6 (SAVE JOB BLOCK) the SEND job is written to the hard disk.

Press F8 (EXIT) to return to the 'SELECTION' mask.



Transfer the job with F2 (TRANSFER) and F3 (FD -> CP) *to the user* memory of the CP 525. This is similar to transferring the interpreter and procedure (see Section 4.3.1.3).

Press F6 (JOB BLOCK) in the 'TRANSFER' mask (see Section 4,3.1.3) *to* transfer all the jobs to the CP 525 user memory.

Perform a CP 525 cold restart by pressing F1 in the 'TRANSFER' mask.

Switch the CP 525 to "RUN". The red LED for device interface 1 (IF 1) must now go off. If it does not, check that the interpreter and procedure are actually present.

5.2 STEP 5 User Program for PC 1 (Reprocessor)

Call the S5 package LAD, CSF, STL. Program in STL (statement list) and call your STEP 5 program file "B:CLPC1RST. S5D".Transfer the handling blocks for the R processor into this program file. Connect the R processor to the PG 685.

The STEP 5 program must perform the following:

- synchronize the CPU and CP 525 during the PC start-up
- trigger the SEND job
- transfer the data from the CPU to the CP 525.

5.2.1 Program Start-up

All the start-up organization blocks OB 20, OB 21 and OB 22 call the SYNCHRON handling block unconditionally.

The interface number SSNR for device interface 1 (IF 1) is the same as that set for the module address "2". Select field length "2" which allows data to be sent in strings of up to 32 bytes at once between the CPU and CP 525. This number is sufficient for this example since no more than 10 words (20 bytes) will be transferred at any time, Any other field length could also be used.

If you wish to transfer more data than the field Length set allows, the CP 525 requests the remaining data with further SEND ALL calls.

With the selection of the field length you can decide the amount of data to be transferred with each SEND ALL (and also RECEIVE AIL). With a smaller field length the data exchange may be distributed over several cycles, but puts very little load on the cycle. With larger field lengths more data at a time can be transferred. The cycle time may well be extended.

Use flag byte "FY10" as the parameter assignment error byte PAFE.

The start-up organization blocks OB 20, OB 21 and OB 22 appear as follows:

```
      :JU  FB125                SYNCHRONIZATION CP 525 <--> CPU
W     :SYNCHRON
SSNR :     KY0,2                INTERFACE NUMBER 2
BLGR :     KY0,2                FIELD LENGTH 2 (MAX, 32 BYTES)
P m  :     FY10                 PARAMETER ASSIGNMENT ERROR BYTE

      :BE
```

A parameter assignment error is to be indicated at digital output Q 0.0. In the start-up organization blocks there is no process image; therefore no digital 1/0s are processed and the indication can be made only in the cyclic program (OB 1, see Section 5.2.2). Normally an error evaluation program is called up at this point. If an error occurs, check whether the interface number and module address are the same. If they are correct a hardware fault is suggested. In this case a computer link cannot be established.

5.2.2 cyclic Program

The beginning of the cyclic program checks whether a parameter assignment error occurred during the start-up (see Section 5.2.1).

The SEND job is triggered by a SEND DIRECT call, i.e., the handling block SEND is called with a job number other than 0; in this case, job number 1. The job number in the SEND DIRECT must be the same as the job in the CP 525. The job is to be carried out on the positive going edge at digital input I 0.1.

The interface number SSNR "2" and the job number A-NR "1" must be programmed. Select condition codeword ANZW "FW 11" and parameter assignment error byte PAFE "FY 15". The condition codeword requires two words. The job status and any error messages are indicated in flag word FW 11. The number of pieces of data to be transferred per SEND ALL call is indicated in flag word FW 13.

In the handling block there must be details about the source of the data in the CPU for the SEND job. The data source should be a data block (source type QTYP "DB") with the number DBNR "10". The source start QANF is data word "0" and source length QLAE is "4" data words.

The SEND job is triggered with the result of logic. operation (RLO) 1. If RLO equals 0 when the call is made, only the condition codeword will be updated.

The termination of the job with an error and the occurrence of a parameter assignment error is indicated at digital output Q 0.1. In this situation, an error evaluation program tailored to the particular application is usually called. If an error occurs, evaluate the error numbers in the parameter assignment error byte PAFE, condition codeword ANZW and error message area in the SYSTAT (see also Section 10).

The actual data transfer from data block DB 10 to the CP 525 is carried out by the SEND ALL call. The interface number SSNR must also be "2". The job number A-NR is "0". Select condition codeword ANZW "FW16" and parameter assignment error byte PAFE "FY18". In the ALL function the condition codeword is only one word long. The *lower* order byte (FY 17) indicates the number of the job for which the SEND AIL is currently transferring data from the CPU to the CP; in this case, job number 1.

The SEND ALL can be called up with any RLO.

On the following page you can see the program in the cyclic [redacted]tion block OB 1.

```
      :A   F 10,0          IF SYNCHRON-PARA. ASS. ERROR
      :=   Q 0.0           OCCURRED --> SET OUTPUT

      :AN I 0.1            EDGE EVALUATION:
      :R   F 2.1           THE RESULT OF LOGIC OPERATION
      :A   10.1            (RLO) IS SET FOR ONE CYCLE, IF
      :AN  F 2.1           THE SIGNAL CHANGES FROM 0 TO 1 AT
      :S   F 2.1           INPUT I 0.1.
      .
      :JU  FB120           TRIGGER SEND JOB 1
NAME :SEND
SSNR :     KY0,2          INTERFACE NUMBER 2
A-NR :     KY0,1          JOB NUMBER 1
ANZW :     FW11           CONDITION CODEWORD
QTYP :     KSDB           SOURCE TYPE DATA BLOCK
DBNR :     KY0,10         SOURCE DATA BLOCK DB 10
QANF :     KF+0           SOURCE START DATA WORD 0
QLAE :     KF+4           SOURCE LENGTH 4 DATA WORDS
PAFE :     FY15           PARAMETER ASSIGNMENT ERROR BYTE

      :O   F 12.3          JOB 1 TERMINATED WITH ERROR
      :O   F 15.0          OR PARAMETER ASSIGNMENT ERROR
      :=   Q 0.1           INDICATE AT OUTPUT Q 0.1

      :JU  FB126           SEND ALL CALL
NAME :SEND-A
SSNR :     KY0,2          INTERFACE NUMBER 2
A-NR :     KY0,0          JOB NUMBER 0
ANZW  :    FW16           CONDITION CODEWORD
PAFE  :    FY18           PARAMETER ASSIGNMENT ERROR BYTE

      :BE
```

All that is still missing is the data source, data block DB 10. To be sure that it is long enough for the other examples, it should be at least 31 data words long. Enter values other than 0 in the first four data words DW 0 to DW 3.

Transfer the R processor handling blocks FB 120 to FB 127, OB 20, OB 21, OB 22, OB 1 and DB 10 to the user memory of your R processor. Carry out a cold restart on the R processor. The green LED (RUN) must light up.

5.3 CP 525 User Program for PC 2 (150U)

Nothing is changed in the CP 525 user program in Section 4.4, since no job is required in the passive PC 2.

Switch the mode selector on the CP 525 to "RUN". The red LED on the CP 525 should now go out. If it does not, the interpreter and/or procedure is missing or there is a CP 525 hardware fault.

5.4 STEP 5 User Program for PC 2 (150U)

Connect the interface module AS 511 to the PC 685.

Call the S5 package IAK), CSF, STL. Program in STL (statement list) and call your STEP 5 program file "B:CLPC25ST.S5D". Transfer the handling blocks for the R processor into this program file.

The STEP 5 program must perform the following:

- synchronize the CPU and CP 525 during the PC start-up
- transfer the data received from the CP 525 to the CPU

5.4.1 Program Start-up

All the start-up organization blocks OB 20, OB 21 and OB 22 call the handling block SYNCHRON unconditionally.

The interface number SSNR for device interface 1 (IF 1) is the same as that set for the module address "01'. Select the smallest field length "1" which allows data to be sent in strings ofup to 16 bytes at once between the CPU and CP 525. This number is sufficient for this example since no more than four words (8 bytes) will be transferredat any time. Any other field length could also be used.

If you want to transfer more than 16 bytes, the CI 525 simply stores the remaining data in the CPU with further RECEIVE ALL calls.

With the selection of the field length you can decide the amount of data to be transferred with each RECEIVE ALL (and also SEND ALL). With a smaller field length the data ex-change may be distributed over several cycles, but puts very little load on the cycle. With larger field lengths more data can be transferred at once but the cycle time may well be extended.

Use flag byte "FY10" as the parameter assignment error byte P m .

The start-up organization blocks OB 20, OB 21 and OB 22 appear as follows:

```
     :JU  FB185                SYNCHRONIZATION CP 525 <--> CPU
NAME :SYNCHRON
SSNR :    KY0,0                INTERFACE NUMBER 0
BLGR :    KY0,1                FIELD LENGTH 1 (MAX. 16 BYTES)
P m  :    FY10                 PARAMETERASSICNMENT ERROR BYTE

     :BE
```

A parameter assignment error is to be indicated by flag F 10.0. If an error occurs, check that the interface number and module address are the same. If they are correct, a hardware fault is suggested. In this case a computer link cannot be established.

5.4.2 Cyclic Program

The data transfer from the CP 525 to the data block DB 11 is carried out by a RECEIVE ALL call. The interface number SSNR must be "0". The job number A-NR is "0". Select condition codeword ANZW "FW 11". In the ALL function the condition code is only one word long. The coordination flag byte number of the job that has just sent data (see Section 8) is indicated in the lower order byte (FY 12). In this example no coordination flags have been programed so FFH appears here. The parameter assignment error byte PAFE is flag byte "FY 13".

The data destination is specified by the CP 525. Therefore, there is no need to enter a destination in the handling block (specify dest. type ZTYP as "NN"). The following three parameters - data block number DBNR, destination start ZANF and destination length ZLAE - are therefore automatically not evaluated.

The RECEIVE ALL can be called up with any result cf logic operation (RLO).

The program in the cyclic organization block OB 1 appears as follows:

```
      :JU FB181                RECEIVE ALL CALL
NAME :RECEIVE
SSNR :    KY0,0                INTERFACE NUMBER 0
A-NR :    KY0,0                JOB NUMBER 0
ANZW  :   FW11                 CONDITION CODEWORD
ZTYP  :   KSNN                 NO DESTINATION SPECIFIED
DBNR :    KY0,0                NOT EVALUATED
ZANF  :   KY+0                 NOT EVALUATED
ZLAE  :   KY+0                 NOT EVALUATED
"Pm   :   FY13                 PARAMETER ASSIGNMENT ERROR BYTE
      :
      :BE
```

Only the data destination data block DB 11 is now missing. This block must be at least 8 data words long.

Transfer the S5-150U handling blocks FB 180, FB 181 and FB 185, OB 20, OB 21, OB 22, OB 1 and DB 11 to the user memory of your S5 150U. Carry out a cold restart. The green LED (RUN) must light up.

5.5 Test

Using the PG 685 online function CONTROL VARIABLE observe the data words DW 4 to DW 7 of data block DB 11 in PC 2 (lSOU), Then switch the digital input I 0.1 on PC 1 from O to 1. The values from PC 1 DB 10 now appear in PC 2 DB 11.

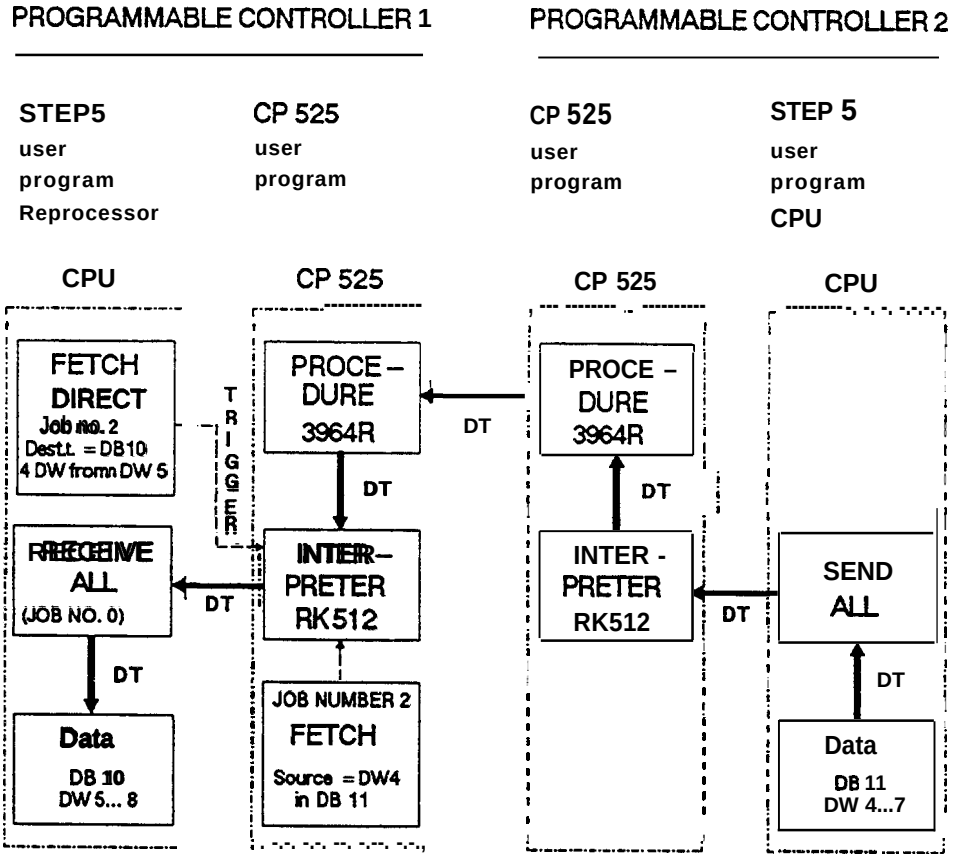
If there is no digital input module with a simulator available, use the flag F 0.1 instead of input I 0.1. Set (ox reset) the flag with the PG 685 online function CONTROL VARIABLE. Observe the condition codeword FW 11 and the parameter assignment byte FY 15 using CONTROL VARIABLE.

6 FETCH Job from Data Block to Data Block

The data sent to PC 2 in Section 5 will be fetched back from PC 2 (150U) to PC 1 (135U) using a FETCH job.

This means that four data words from DB 11 starting from data word DW 4 in PC 2 (150U) are fetched and transferred to data block DB 10 from data word DW 5 in PC 1 (135U/R processor). The source and destination specified are different.

The following program structure is required:



*) DT = DATA TRANSFER

The interpreter and procedure are already loaded in the user memory of both CP 525s (see Section 4).

The FETCH job needs to be programmed only on the CP 525 in the active PC. Active means that this PC triggers the exchange of data and that you do not need to program a FETCH job for the CP 525 on PC 1.

The FETCH job is triggered when the CPU calls the handling block FETCH DIRECT with the job number 'n'. Select job number 2.

The corresponding job on the CP 525 has the same job number 'n'. This job includes specifications about the data source in PC 2; the data destination in PC 1 is specified in the FETCH DIRECT.

The CP 525 in PC 1 (135U) sends a request telegram to the CP 525 in PC 2 (150U) with details about the location from which the data is to be fetched. The CP 525 in PC 2 (150U) fetches the requested data using a SEND ALL from data block DB 11 of the CPU, and sends the data to PC 1. The CP 525 receives the data in PC 1 and writes it into data block DB 10 with a RECEIVE ALL.

The SEND ALL (also known as SEND 0) has the job number A-NR 0. It is called unconditionally in every program cycle and checks whether the CP 525 (dual-port RAM) requires data. If the CP does, the SEND ALL fetches it from the specified source and transfers it to the CP 525. If the CP does not, the function block is exited and the cyclic program continues.

The RECEIVE ALL (also known as RECEIVE 0) has the job number A-NR 0. It is called unconditionally in every program cycle and checks whether the CP 525 (dual port RAM) needs to transfer data to the CPU. If it does, the RECEIVE ALL stores the data in the destination address specified by the CP 525. If it does not, the function block is exited and the cyclic program continues.

6.1 CP 525 User Program for PC 1 (135U)

Switch the mode selector on the CP 525 to "PGR".

Call the program "CL512PC1" in the COM 525 mask 'PROGRAM SELECTION' with F7 (HELP) .

Press F1 (PROGRAM USER DATA) in the 'SELECTION' mask to call the 'JOB BLOCK' mask. Enter the job number with which the job will be triggered by the CPU. Select 'Job no. :' "2".

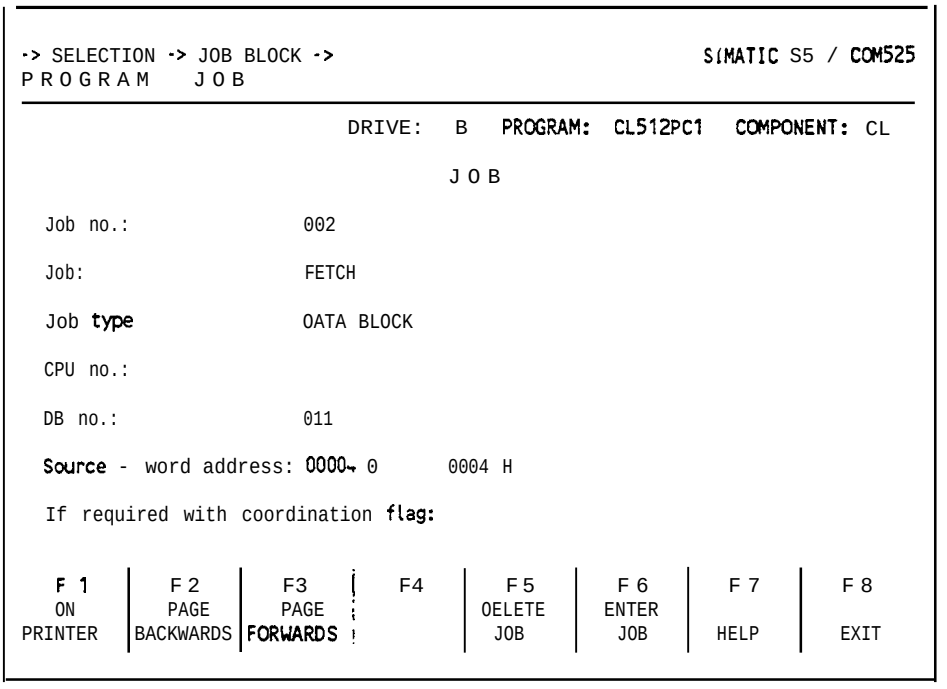
Press F5 (PROGRAM JOB) to call the 'PROCRAM JOB' mask.

Using F7 (HELP) enter the 'job: ' "FETCH" and 'job type' "DATA BLOCK". The 'source - word address' in PC 2 is "4" D (decimal) and also "4" H (hexadecimal). The data block number is "11". No CPU number need be specified since PC 2 (150U) is capable only of single processor operation. Do not enter any coordination flags.

The job is written to the RAM on the PC 685 with F6 (ENTER JOB). Press F8 (EXIT) to return to the 'JOB BLOCK' mask,

Press F6 (SAVE JOB BLOCK) to save the whole job block; i.e. , the FETCH and the SEND job (see Section 5) are saved on the hard disk. Depending on whether you wish to overwrite the existing job block, answer the prompt 'ACK.011: Job block already exists - overwrite?' with F1 (YES) or F3 (NO).

Press F8 (EXIT) to return to the 'SELECTION' mask.



Transfer the job in the 'SELECTION' mask with F2 (TRANSFER) and F3 (F'D -> CP) to the user memory of the CP 525. This is similar to transferring the' interpreter and procedure (see Section 4.3.1.3).

Press F6 (JOB BLOCK) in the 'TRANSFER' mask (see Section 4.3.1.3) to transfer all the jobs to the CP 525 user memory.

Performa CP 525 cold restart by pressing F1 in the 'TRANSFER* mask.

Switch the CP 525 to "RUN". The red LED for device interface 1 (IF 1) must now go off, If it does not, check that the interpreter and procedure are actually present.

6.2 STEP 5 User Program for PC 1 (R Processor)

The FETCH job is triggered by a FETCH DIRECT call with a job number other than 0, in this case, job number 2. The job number in the FETCH DIRECT block and the number of the job in the CP 525 must be the same. The job is to be carried out on the positive going edge at digital input I 0.2.

Interface number SSNR "2" and job number A-NR "2" must be programmed. Select the flag word "FW 20" as the condition codeword ANZW and "FY 24" as the parameter assignment error byte PAFE. The condition codeword indicates the job status and any error messages.

The FETCH DIRECT requires details of the destination of the data in PC 1. The data destination should be the data block (dest. type QTYP "DB") with the number DBNR "10". The start of the destination ZANF is data word "5" and the destination length ZLAE is "4" data words.

The FETCH job is triggered with the result of logic operation (RLO) 1. If RLO equals 0 when the call is made, only the condition codeword is updated

The tennination of the job with an error and the occurrence of a parameter assignment error is indicated at digital output Q 0.2. In this situation, an error evaluation program tailored to the particular application is usually called. If an error occurs, evaluate the error numbers in the parameter assignment error byte PAFE, condition codeword ANZW and error message area in the SYSTAT (see also Section 10).

The data sent from PC 2 (150U) is written to the Reprocessor by the CP 525 of PC 1 (135U) using RECEIVE ALL. The interface number SSNR must also be "2". The job number A-NR is "0". Select condition codewordANZW "FW25° and parameter assignment error byte PAFE "FY 27". In the AIL function the condition codeword is only one word long. The lower order byte (FY 26) indicates the number of the job for which the RECEIVE ALL is currently transferring data from the CP to the CPU; in this case job number 2.

The RECEIVE AIL can be called up with any RLO.

From the STEP 5 presetting mask, call the program file "B:CLPC1RST.S5D" and add the following statements to the organization block OB 1:

```
     :AN I 0.2          EDGE EVALUATION:
     :R   F 2.2         THE RESULT OF LOGIC OPERATION
     :A   I 0.2         (RLO) IS SET FOR ONE CYCLE IF THE
     :AN F2.2           SIGNALAT INPUT I 0.2 CHANGES FROM
     :S   F 2.2         0 TO 1.

     :JU  FB122         TRIGGER FETCH JOB 2
NAME :FETCH
SSNR :    KY0,2        INTERFACE NUMBER 2
A-NR :    KY0,2        JOB NUMBER 2
ANZW  :   FW20         CONDITION CODEWORD
ZTYP :    KSDB         DEST. TYPE DATA BLOCK
DBNR :    KY0,10       DEST. DATA BLOCK DB 10
ZANF :    KF+5         DEST. START DATA WORD 5
ZLAE  :   KF+4         DEST. LENGTH 4 DATA WORDS
PAFE :    FY24         PARAMETER ASSIGNMENT ERROR BYTE

     :O   F 21.3       JOB 2 TERMINATED WITH ERROR
     :O   F 24.0       OR PAMMETER ASSIGNMENT ERROR
     :=   Q0.2         INDICATED AT OUTPUT Q 0.2
     .
     :JU  FB127        RECEIVE ALL CALL
W    :REC-A
SSNR :    KY0 ,2       INTERFACE NUMBER 2
A-NR  : KY0, 0         JOB NUMBER 0
ANZW  :   FW25         CONDITION CODEWORD
PAFE : FY27            PARAMETER ASSIGNMENT ERROR BYTE
```

Data block DB 10 was programed with sufficient length in the previous example.

Transfer the extended OB 1 to the user memory of your R processor. Carry out a cold restart on the R processor. The green LED (RUN) should light up.

6.3 CP 525 User Program for PC 2 (150U)

Nothing is changed in the CP 525 user program in Section 4.4 since no job is required in the passive PC 2.

Switch the mode selector on the CP 525 to "RUN". The red LED on the device interface 1 of the CP 525 should now go out. If it does not, the interpreter and/or procedure is missing or there is a CP 525 hardware fault.

6.4 STEP 5 User Program for PC 2 (150U)

The transfer of the data requested by PC 1 (135U) from the CPU (data block DB 11) to the CP 525 in PC 2 (150U) is performed bya SEND ALL call. The interface number SSNR must be "0". The job number A-NR is "0". Select "FW14" as the condition codeword ANZW. In ALL functions the condition codeword is one word long. The coordination flag byte number of the job for which the data is fetched is indicated in the lower order byte (J?Y 15). In this example there are no coordination flags in the FETCH job for the CP 525 in PC 1 (135U) so FFH appears here.

CP 525 specifies the data source for the SEND ALL. No source need be specified in the handling block (specify source type QTYP "NN"). Thenext three parameters - data block number DBNR, source start QANF and source length QLAE - are then automatically not evaluated.

The parameter assignment error byte PAFE is flag byte "FY 16".

Connect the interface module AS 511 with the PG 685 and call up the program file "B: :CLPC525ST.S5D" in the STEP 5 presettings mask.

Organization block OB 1 must be extended as follows:

```
     :JU  FB180                SEND ALL CALL
NAME :SEND
SSNR :    KY0,0                INTERFACE NUMBER 0
A-NR :    KY0,0                JOB NUMBER 0
ANZW  :   FW14                 CONDITION CODEWORD
QTYP  :  KSNN                  NO SOURCE SPECIFIEL
DBNR :    KY0,0                NOT EVALUATED
QANF :    KY+0                 NOT EVALUATED
QLAE :    KY+0                 NOT EVALUATED
PAFE  :  FY16                  PARAMETER ASSIGNMENT ERROR BYTE
```

You already programmed the data block **DB 11 in the** previous **example.**

Transfer the extended OB 1 to the user memory of your **S5-150U.** **Carry out** a cold restart The green LEO (RUN) must light up.

6.5 Test

Observe the data words DW 0 to DW 3 and DW 5 to DW 9 of data block DB 10 in PC 1 (135U) using the PC 685 online ***function*** CONTROL VARIABLE.

Switch the digital input I 0.1 and then input I 0.2 on PC 1 from 0 to 1. Four data words from data word DW 0 to DW 3 are then sent to PC 2 (150U). Data words DW 5 to DW 9 are transferred back (see also Section 5). The values of data words DW 0 to DW 3 must now be located in data words DW 5 to DW 9.

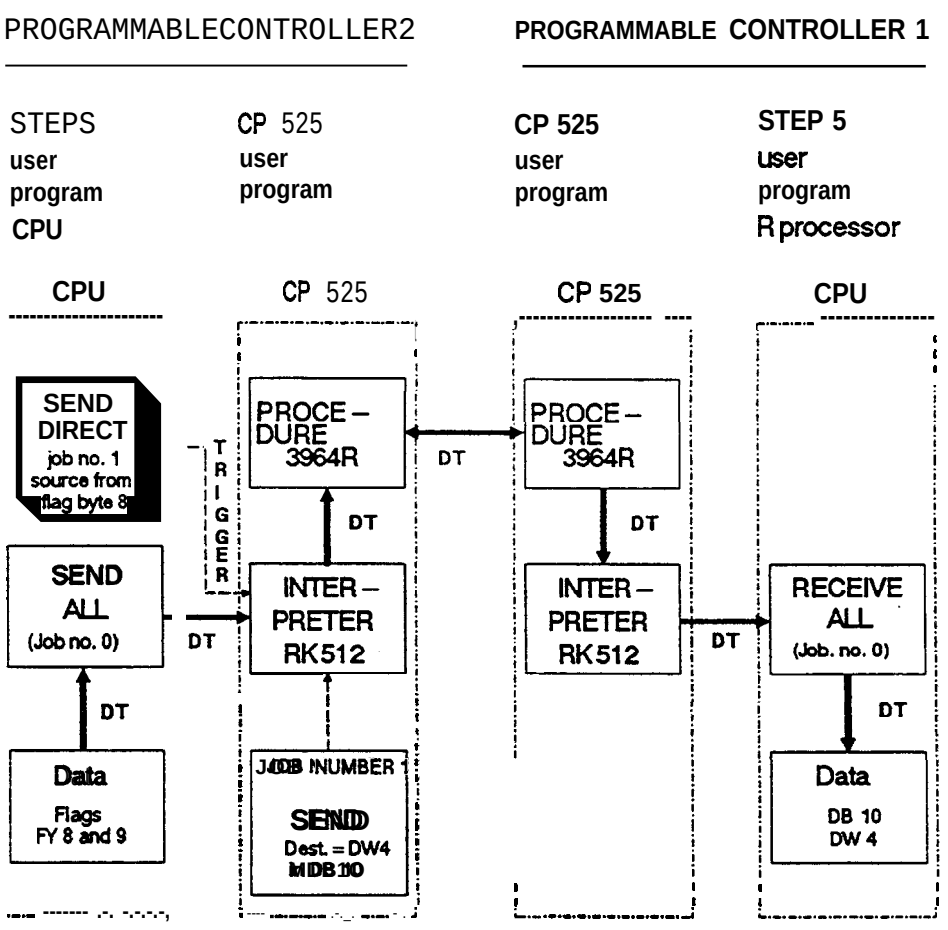
If you do not have any digital 1/0 **modules** with a simulator available, then use **flags** F 0.1 and F 0.2 instead of the inputs I 0.1 and I 0.2. Set (or reset) the flags with the **PG** 685 online function CONTROL **VARIABLE.** Observe the condition codewords FW 11 and FW 20 and the **parameter** assignment error bytes FY 15 and **FY** 24 with **CONTROL VARIABLE.**

7 SEND Job, Flags to Data Block

A SEND job is to be programmed in the opposite direction, from PC 2 to PC 1.

In every second program cycle, flag word 8 (i.e. flag bytes 8 and 9) are to be transferred from PC 2 (150U) to data block DB 10 - data word DW 4 - in PC 1 (135U).

The following program structure is required:



*) DT = DATA TRANSFER?

The interpreter and procedure are already loaded in the user memories of the two CP 525s (see Section 4) .

The SEND job needs to be programed only on the CP 525 of the active PC. Active means that this PC initiates the data exchange and that you need to program a SEND job only for the CP 525 in PC 2.

The SEND job is initiated when the CPU calls the handling block SEND DIRECT with the job number 'n'. Select job number '1'. The same job number 'n' is also given to the corresponding job on the CP 525. This job contains information about the destination of the data in PC 1 (135U).

The CP 525 fetches the data only when called by the handling block SEND ALL which transfers the data from the CPU/FW8 to the CP 525 in PC 2.

The CP 525 in PC 1 (135U) receives the data and calls the RECEIVE ALL to store it in the destination indicated by PC 2.

The **SEND ALL** (also known as SEND 0) has the job number A-NR 0. It is called unconditionally? in every program cycle and checks whether the CP 525 (dual-port RAM) requires data. If the CP does, the SEND ALL fetches it from the specified source and transfers it to the CP 525. If the CP does not, the function block is exited and the cyclic program continues.

The RECEIVE ALL (also known as RECEIVE 0) has the job number A-NR 0. It is called unconditionally in every program cycle and checks whether the CP 525 (dud-port RAM) wishes to transfer data to the CPU. If it does, the RECEIVE ALL stores the data in the destination address specified by the CP 525, If it does not, the function block is exited and the cyclic program continues.

The content of flag word FW 8 is to be incremented by 1 in each program cycle. In this way it is easy to recognize in PC 1 that the SEND job has been carried out successfully.

7.1 CP 525 User Program for PC 1 (13 U)

Nothing is changed in the CP 525 user program since no job is required in the passive PC 1.

Switch the mode selector on the CP 525 to "RUN". The red LED on the CP 525 should now go out, If this is not the case, the interpreter and/or procedure is missing or there is a CP 525 hardware fault.

7.2 STEP 5 User Program for PC 1 (Reprocessor)

The transfer of the data received by the CP 525 to data block DB 10 is performed by the RECEIVE ALL function which is called up cyclically. Since this, and the data block DB 10, were programmed in the previous examples, nothing needbechanged in the STEP 5 programon the R processor.

Switch the R processor to "RUN". The green LED must light up.

7.3 CP 525 User Program for PC 2 (135U)

Switch the mode selector on the CP 525 to "PCR" ard call the program "CL512PC2" in the COM 525 mask 'PROGRAM SELECTION' .

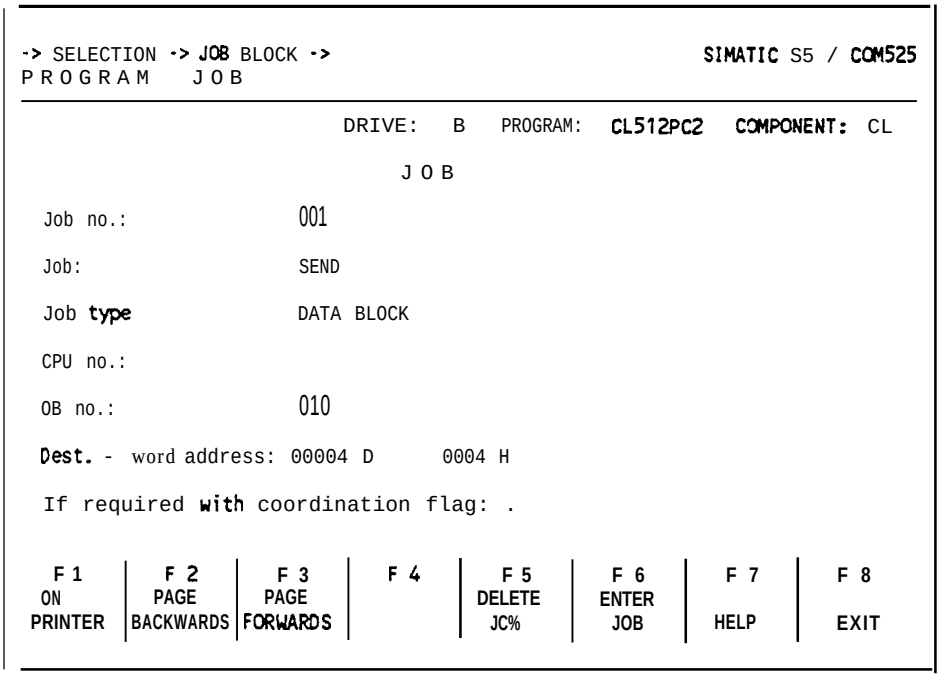
Press F1 (PROCRAM USER DATA) in the 'SELECTION' mask to call up the 'JOB BLOCK' . Enter the job number with which this job is to be triggered by the CPU. Select 'Job no.:' "1".

Press F5 (PROGRAM JOB) to call the 'PROGRAM JOB' mask.

Using F7 (HELP) enter 'job:' "SEND" and 'job type' 'DATA BLOCK". The 'dest. - word address' is "4" D (decimal) and the data block number is "10". No entry is necessary for the CPU number as PC 1 (135U) is in single processor operation. Do not specify any coordination flags.

The job is written to the RAM memory of the PG 685 by pressing F6 (ENTER JOB). With F8 (EXIT) you can return to the 'JOB BLOCK' mask. With F6 (SAVE JOB BLOCK) the SEND job is written to the hard disk.

F8 (EXIT) returns you to the 'SELECTION' mask,



Transfer the job with F2 (TRANSFER) and F3 (FD -> CP) to the user memory of the CP 525. This is similar to transferring the interpreter and procedure (see Section 4.4.1.2).

Ress F6 (JOB BLOCK) in the 'TRANSFER' mask (see Section 4,4.1.2) to transfer all the jobs to the CP 525 user memory.

Perform a CP 525 cold restart by pressing F1 in the 'TRANSFER' mask.

Switch the CP 525 to "RUN". The red LED for device interface 1 (IF 1) must now go off. If it does not, check that the interpreter and procedure are actually present.

7.4 STEP 5 User Program for PC 2 (15JU)

The SEND job is triggered by a SEND DIRECT call, with a job number other than 0, in this case job number 1. The job number in the SEND DIRECT must be the same as that of the job in the CP 525. The job is to be carried out in each second program cycle,

The interface number SSNR "0" and job number A-NR "1" must be programmed. Select condition codeword ANZW "FW 20" and parameter assignment error byte PAFE "FY 25". The condition codeword requires two words. The job status and any error messages are indicated in flag word FW 20. The number of pieces of data transferred during the current SEND ALL call is indicated in flag word FW 22.

In the handling block there must be details about the source of the data in the CPU for the SEND job. The data source should be flag bytes (source type QTYP "FY"). The parameter data block DBNR is therefore not evaluated. The source start QANF is flag byte "8" and source length QLAE is "2" flag bytes,

The SEND job is triggered with the result of logic operation (RLO) 1. If RLO equals 0 when the call is made, only the condition codeword is updated. Inverting the flag bit F 3.0 means that the RLO is 0 in one cycle and 1 in the next.

The termination of the job with an error is indicated in the condition codeword in bit 3. The occurrence of a parameter assignment error can be recognized by the setting of bit 0 in the parameter assignment error byte. In this situation, an error evaluation pro-gram tailored to the particular application is usually called. If an error occurs, evaluate the error numbers in the parameter as-signment error byte PAFE, condition codeword ANZW and error roes-sage area in the SYSTAT (see also Section 10).

The actual data transfer of the flag to the CP 525 is carried out by the SEND AIL call. This was programed in the last example.

Connect the PG 685 with the programmer interface module AS 511, and call up the program file "B:CLPC25ST.S5D" in the STEP 5 presetting mask. Extend the organization block OB 1 with the following statements.

```
     :L   FW8
     :L   KF+1
     :+F                    INCREMENT CONTENT OF THE FLAG WORD
     :T   FW8               BY 1 IN EACH PROG. CYCLE

     :AN F 3.0              INVERT FLAG BIT IN EACH
     :=   F 3.0             PROGRAM CYCLE

     :JU  FB180             TRIGGER SEND JOB 1
NAME :SEND
SSNR :    KY0,0            INTERFACE NUMBER 0
A-NR :    KY0,1            JOB NUMBER 1
ANZW :    FW20             CONDITION CODEWORD
QTYP :    KSMB             SOURCE TYPE FLAG
DBNR :    KY0,0            NOT EVALUATED
QANF :    KF+8             SOURCE START FLAG BYTE 8
QLAE :    KF+2             SOURCE LENGTH 2 FLAG BYTES
PAFE :    FY24             PARAMETER ASSIGNMENT ERROR BYTE
```

Transfer OB 1 to the user memory of your S5 -150U. Perform a cold restart on the PC. The green LED (RUN) must light up.

7.5 Test

Observe the data word DW 4 in data block DB 10 on PC 1 (135U) using the PG 685 online function CONTROL VARIABLE. The value in the data word must increase constantly.

8 Coordination Flags

Coordination flags with a SEND job

By using coordination flags you can protect the data destination of a SEND job (if this is a data block) from being overwritten. The coordination flag is specified in the SEND job. The byte and bit number of this flag are transferred to the partner along with the data.

The byte number of the coordination flag appears in the destination PC for one cycle in the condition codeword of the RECEIVE ALL. Based on this byte number, you can recognize in the destination CPU which job is responsible for the data that has arrived. If you set the coordination flag belonging to the job in the STEP 5 user program, you can prevent the job from being repeated. As seen as the data from the destination area is saved or otherwise processed, you can have the user program reset the coordination flag; the job is then free to be executed again.

Coordination flags with a FETCH job

By using coordination flags you can also protect the data source of a FETCH job (only possible for DB or DX) in the source CPU from being read. A coordination flag is specified in the FETCH job of the CP 525. The byte and bit number of *this* flag are transferred in the request telegram to the partner.

The byte number of the coordination flag appears in the PC in which the data source is located. It appears for one cycle in the condition codewordoftheSENDAIL Basedonthisbytenumberyou can recognise when and by which job data is requested. If you set the coordination flag belonging to this job in the STEP 5user program, you can prevent the source data area for this job being read. Assoonasthedatain the source area is ready, you can have the coordination flag reset; data is then once again transferred in response to the request telegram from the job.

Note that in the condition codeword of the SEND ALL or RECEIVE ALL only the byte and not the bit number of the coordination flag appears. You must therefore make sure that the byte number allocated to a coordination flag is unique if you wish to evaluate the information in the condition codeword in the program. You can use any flag as a coordination flag, however, only byte numbers from 1 to 223 can be indicated in the condition codeword.

The coordination flags must be declared in the STEP 5 program as IPC output flags and enabled on the CP 525 with jumpers at jumper block 25. The IPC flags (and therefore also the coordination flags) are part of the normal flag area.

IPC flags with the R processor

The IPC flags must be entered in data block DB 1 as IPC outputs. As soon as DB 1 is programmed, the digital 1/0 bytes must also be entered. The IPC flags are then handled by the operating system as digital I/Os. Before the cyclic program is run, the IPC input flags are transferred from the dual-port RAM of the CP to the flag area of the CPU. IPC input flags are not used with the CP 525. At the end of the cyclic program IPC output flags are transferred from the flag area of the CPU to the dual-port RAM.

IPC flags with the S5-115U

The data block DB 1 must also be programmed. Digital I/Os do not need to be entered with the S5-115U.

IPC flags with the S5-150U

The IPC flags must be transferred in each program cycle between the CPU and the dual-port RAM of the CP 525 by function block FB 186 or by system commands.FB 186 is supplied with the handling blocks. ADB 1 for exchanging IPC flags and/or digital I/Os does not exist in the S5-150U.

In every second program cycle, PC 2 (150U) triggers a SEND job (job number 1). It is transferredto data block DB 10 - data word DW 4 in PC 1 with the flag word FW 8. The job was programed in Section 7 and only needs to have the details of the coordination flag F 5.3 added. The content of the flag word is incremented by 1 in each program cycle.

The program in PC 1 (R processor) recognises from the condition codeword of the RECEIVE ALL that data with the coordination flag byte number5fromaSEND jobhasarrived. CoordinationflagF 5.3 is then set. The jobs with the same coordination flag numbers (same byte and bit number) are then rejected with the error number 32H in the reply telegram, 9H in the condition codeword and 30H in the SYSTAT. Therefore data word DW 4 is protected from being overwritten.

With digital input I 0.7 you can reset the coordination flag again so that the SEND job is again accepted.

8.1 CP 525 User Program for PC 1 (135U)

The CP 525 program does not need to be changed. In any case no job is required on PC 1. The IPC flag bytes 0 to 31 have already been enabled by means of jumpers (see Section 3).

Switch the CP 525 to “RUN”. The red LED on the CP 525 must go out. If this is not the case, either the interpreter and/or procedure is missing or there is a hardware fault on the CP 525.

8.2 STEP 5 User Program for PC 1 (R Processor)

The transfer of the data received by the CP 525 to the data block DB 10 is performed by the RECEIVE ALL, which is called up cyclically. The RECEIVE ALL and data block DB 10 have already been programmed.

As soon as data arrives from the SEND job (with job number 1) of PC 2 the coordination flag byte number appears in the lower order byte of the RECEIVE ALL condition codeword. If the byte number is 5, the coordination flag F 5.3 is set and SEND jobs from PC 2 are rejected.

The coordination flag is reset by a positive edge at digital input I 0.7 allowing the job to be accepted again.

FOUowing the next SEND job from PC 2 (with job number 1) the coordination flag is set again immediately.

Extend organization block OB 1 as follows:

```
:L   FY26          COORDINATION FIAGBYTENUMBERFROM
:L   KF+5          CONDITION CODEWORD OF THE RECEIVE
:!=F               ALL EQUALS 5? IF YES, THEN SET
:S   F 5.3         COORDINATION FLAG

:                  AS LONG AS THE COORDINATION FLAG
                   IS SET, THE DATADESTINATIONIS
:                  PROTECTED FROM BEING OVERWRITTEN,
:                  THE SEND JOB WITH COORDINATION
:                  FLAG F 5.3 WILL EJECTED.
:
:AN  10.7          EDGE EVALUATION:
:R   F 2.7         THE RESULT OF LOGIC OPERATION
:A   I 0.7         (RLO) IS SET FOR ONE CYCLE, IF
:AN  F 2.7         THE SIGNALAT I 0.7 CHANGES FROM
:S   F 2.7         0 TO 1.

:R   F 5.3         RESET COORDINATION FLAG WITH 0->1
                   EDGEATINPUTI0.7
```

IPC output flag byte 5, input byte 0 and output byte 0 must be entered in data **block DB 1** with the STEP 5 programming function INPUT (F1) **MASK** (F4) :

DB 1 ASSIGNMENT OF 1/0s

DIGITAL INPUTS	: ,	**0**, , , , . . .
DIGITAL OUTPUTS	: ,	0, , , , . . .
IPC FLAG **INPUT's**	: ,	, , ,
IPC FLAG **OUTPUTS**	: ,	5: , , , . . .
TIMER BLOCK LENGTH	: ,	

Transfer the extended OB 1 and DB 1 to the R processor. Perform a cold restart on the R processor. The green LED must now light up.

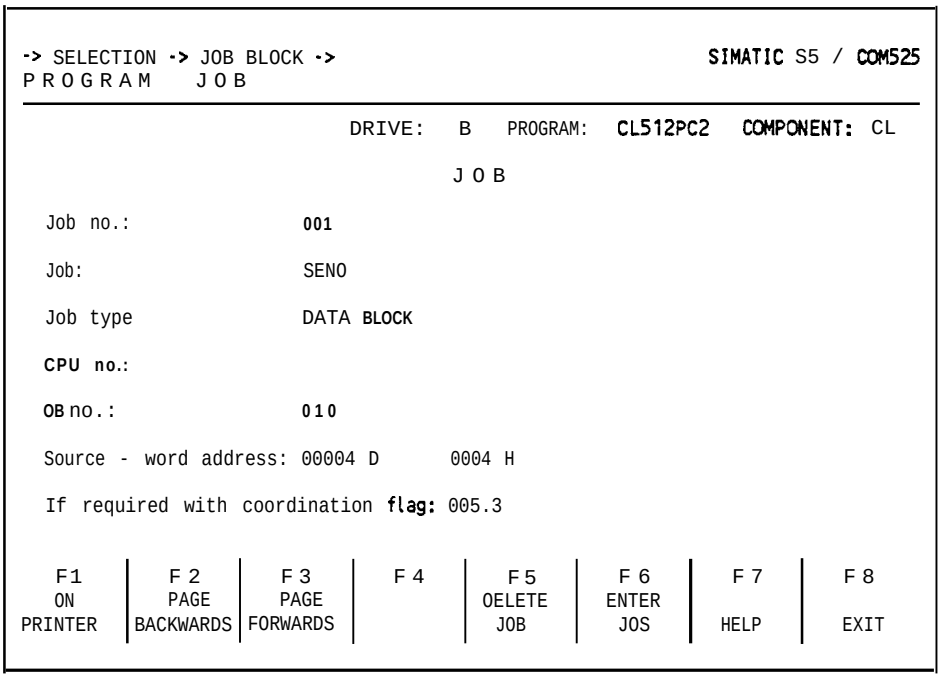
8.3 **CP525 User Program in PC2 (150U)**

Switch the mode selector of the CP 525 to "PGR" and call up the program "CL512PC2" in the COM 525 mask ' PROGRAM SELECTION' .

Using F1 (PROGRAM USER DATA) in the 'SELECTION' mask, you call up the 'JOB BLOCK' mask. Enter the job number "l" of the SEND job programmed in Section 7.3.

Press F5 (PROGRAM JOB) to call up the 'PROGRAM JOB ' mask.

Leave the destination parameters as they are and enter the coordination flag F "5. 3" .



The job is entered in the RAM of the PG 685 with F6 (ENTER JOB). With F8 (EXIT) you rem to the 'JOB BLOCK' mask. With F6 (SAVE JOB BLOCK) the SEND job is written to the hard disk. Return to the 'SELECTIONI mask with F8 (EXIT).

Transfer the job in the 'SELECTION' mask with F2 (TRANSFER) and F3 (FD -> CP) to the user memory of the CP 525. This is similar to transferring the interpreter and procedure (see Section 4.4.1.2). Ress F6 (JOB BLOCK) in the 'TRANSFER' mask (see Section 4.4.1.2) to transfer all the jobs to the CP 525 user memory. Perform a CP 525 cold ~ by pressing F1 in the 'TRANSFER' mask.

Switch the CP 525 to "RUN". The red LED for device interface 1 (IF 1) must now go off. If it does not, check that the interpreter and procedure are actually present.

8.4 STEP 5 User Program for PC 2 (150U)

Nothing needs changing on the previous STEP 5 program. The SEND job with job number 1 is triggered as before (see Section 7.4).

Switch the CPU to "RUN". The green LED must light up.

8.5 Test

Observe the data word DW 4 in data block DB 10 on PC 1 (135U) with the PG 685 online function CONTROL VARIABLE. The value does not change.

Switch input I 0.7 from 0 to 1. The value in data word DW 4 must change once and then remain constant since the coordination flag is set immediately after the job is accepted.

Also observe condition codeword FW 20 of the SEND job with job number 1 on the S5-150U. If the coordination flag is set in the R processor, this SEND job is rejected by PC 1 (135U) and KM = 00001001 00101000 appears briefly in the condition codeword. This means that the job is completed with error 9H, DB or DX inhibited by coordination flag. The error number 30H appears in the error message area of the SYSTAT (see Section 10) .

9 PSEUDO WRITE Job, Input to Data Block

You can have the source and destination of a job specified by the STEP 5 program and change them while the program is running. For this purpose there are special jobs on the CP 525 with the reserved job numbers 190 to 199. To distinguish them from normal SEND and FETCH jobs, these functions are known as PSEUDO WRITE and PSEUDO READ. These functions have nothing to do with the READ/WRITE functions of the handling blocks. In the READ/WRITE functions, neither CPU numbers nor coordination flags can be specified.

No job needs to be programmed on the CP 525.

The source and destination must be specified in a data block or extended data block (the parameter data block) as follows

n :	KS= QTYP	source type (not XX, RW or NN)
n + 1 :	KY= DBNR	data block number for QTYP DB or DX
n + 2 :	KF= QANF	source start address
n + 3 :	KF= QLAE	source length SEND (any length for FETCH)
n + 4 :	KS= ZTYP	dest. type (only DB, DX, RS, AS)
n + 5 :	KY= DBNR	data block number for ZTYP DB
n + 6 :	RF= ZANF	dest. start address
n + 7 :	KF= ZLAE	dest. length FETCH (any length for SEND)
n + 8 :	KY= BYTE, BIT	coordination flag byte and bit number
n + 9 :	KF= CPU No	CPU number in partner

The data type DB or DX, the data block number and the start word address n of the set of parameters must be specified when the SEND DIRECT or FETCH DIRECT is called. The CP 525 automatically fetches the 10 data words with the transfer parameters.

The PSEUDO WRITE function is triggered by a SEND DIRECT call with a job number between 190 and 199. The special job first fetches the source and destination parameters from the parameter data block specified with the SEND DIRECT by means of a SEND ALL. With a second SEND ALL call, the CP 525 fetches the data to be transferred from the source specified in the parameter data block

and transfers it to the partner. At the partner the data is stored under the specified destination address in the parameter data block by a REcEIVE ALL.

The PSEUDO READ function is triggered by a FETCH DIRECT call with a job number between 190 and 199. The special job first fetches the source and destination parameters from the parameter data block specified with the FETCH DIRECT by means of a SEND AIL. The CP 525 then sends a request telegram with the source address specified in the parameter data block to the partner. This uses a SEND ALL to fetch the data from the data source and sends it to the requesting CP 525. The CP stores the data using a RECEIVE ALL under the address specified in the parameter data block.

For example, input byte IB O in in PC 1 (135U) is sent to data word DW O in data block DB 11 of PC 2 (150U) using a PSEUDO WRITE job.

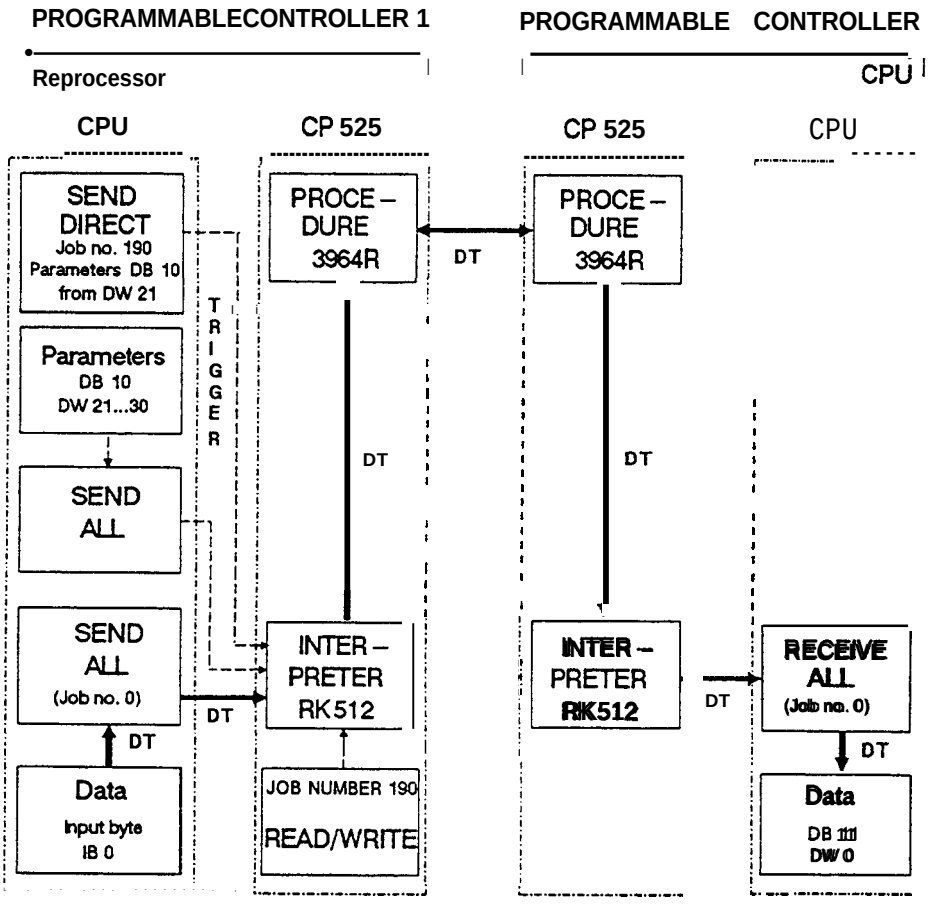
The interpreter and procedure are already loaded in the user memory of both CP 525s (see Section 4).

The special jobs 190 to 199 for the PSEUDO READ and PSEUDO WRITE jobs are permanently installed on the CP 525 so they do not need programming and transferring to the CP user memory. The job numbers 190 to 199 are reserved in the computer link for PSEUDO READ and PSEUDO WRITE functions.

The PSEUDO WRITE job is triggered by calling SEND DIRECT. Select the job number 190. The CP 525 first fetches the source and destination parameters from the parameter data block by means of a SEND AIL. Then the input byte IB O is fetched with a second SEND ALL and transferred to the CP 525 in PC 2 (150U).

The CP 525 in PC 2 (150U) receives the data and stores it in the destination specified by PC 1 (135U) by means of a RECEIVE ALL.

When only one byte is transferred to a data word, the CP 525 writes the data into the left data byte DL and overwrites the right data byte DR with O.



') DT = DATA TRANSFER

The SEND AIL (also known as SEND 0) has the job number A-NR 0. It is called unconditionally in every program cycle and checks whether the CP 525 (dual-port RAM) requires data. If it does, the SEND AIL fetches it from the specified address and transfers it to the CP 525. If the CP does not require data, the function block is exited and the cyclic program continues.

The RECEIVE ALL (also known as RECEIVE 0) has the job number A-NR 0. It is called unconditionally in every program cycle and checks whether the CP 525 (dual-port RAM) wishes to transfer data to the CPU. If it does, the RECEIVE ALL stores the data in the destination address specified by the CP 525. If it does not, the function block is exited and the cyclic program continues.

9.1 CP 525 User Program for PC 1 (135U)

The job 190 is already available on the CP 525 and does not need tobeprogrammed.

Switch the CP 525 to "RUN". The red LED on device interface 1 (IF 1) must now go off. If this is not the case check whether the interpreter and procedure actually exist.

9.2 STEP 5 User Program for PC 1 (R Processor)

The PSEUDO WRITE job is to be triggered by a signal change from 0 to 1 at digital input I 0.3.

The PSEUDO WRITE job is triggered by a SEND DIRECT call; i.e., the handling block SEND with a job number in the range between 190 and 199 is called. Select job number 190.

The interface number SSNR must be programmed as "2" and the job number A-NR "190". Select "FW 30" as the condition codeword ANZW and "FY 34" as the parameter assignment error byte PAFE. The condition codeword requires two words. The flag word FW 30 indicates the job status and any error messages. The number of data to be transferred with each SEND ALL call is located in flag word FW 32.

The start address of the transfer parameters must be specified for the PSEUDO WRITE job. The parameters should be located in the data block (QTYP "DB") with the number DBNR "10" from data word QANF "21". The CP 525 automatically requests 10 data words as transfer parameters; the parameter QLAE is therefore not evaluated.

The SEND job is called unconditionally and triggered by result of logic operation (RLO) 1. If the RLO is 0 at the time of the call, only the condition codeword is updated.

The termination of the job with errors and the occurrence of a parameter assignment error are indicated at digital output Q 0.3. In this situation an error evaluation program tailored to the particular application is usually called. If an error occurs, evaluate the error numbers in the parameter assignment error byte PAFE, condition codeword ANZW and the error message area in the SYSTAT (see also Section 10).

Connect the PG 685 with the R processor and call up the program file "B:CLPC1RST.S5D" in the STEP 5 presetting mask. Extend the organization block OB 1 with the following statements.

```
     :AN I 0.3              EDGE EVALUATION:
     :R  F 2.3              THE RESULT OF LOGIC OPERATION
     :A  10.3               (RLO) IS SET FOR ONE CYCLE, IF THE
     :AN F 2.3              SIGNALAT INPUT I 0.3 CHANGES
     :S  F 2.3              FROM 0 TO 1.
     :
     :JU FB120              TRIGGER SEND JOB 190
NAME :SEND
SSNR :    KY0,2             INTERFACE NUMBER 2
A-NR :    KY0,190           JOB NUMBER 190
ANZW  :   FW30              CONDITION CODEWORD
QTYP :    KSDB              SOURCE TYPE DATA BLOCK
DBNR :    KY0,10            PAMMETERFROMDATABLOCKDB10
QANF :    KF+21             START OF PARAMETER BLOCK DW 21
QLAE :    KF+0              NOT EVALUATED
P m :     FY34              PARAMETER ASSIGNMENT ERROR BYTE
     :
     :O  F 31.3             JOB 190 TERMINATED WITH ERROR
     :O  F 34.0             OR PARAMETER ASSIGNMENT ERROR
     :=  QO.3               INDICATE AT OUTPUT Q 0.3
```

The SEND AIL call required for transferring the transfer parameters and the input byte are already programmed.

Program the source and destination in data block DB 10:

```
21 :    KS= IB        QTYP source type input byte
22 :    KY= 0,0       DBNR DB number not necessary
23 :    KF= +0        QANF source start is input byte IB 0
24 :    KF= +1        QLAE source length is 1 byte
25 :    KS= DB        ZTYP dest. type data block
26 :    KY= 0,11      DBNR dest. data block is DB 11
27 :    KF= +0        ZANF dest. start is data word DW 0
28 :    KH= FFFF      ZLAE irrelevant with SEND
29 :    KY= 255,255   no coordination flag
30 :    KF= +0        CPU number
```

Transfer organization block OB 1 and data block DB 10 to the user memory of your S5-135U. Perform a cold restart on the PC. The green LED (RUN) must light up.

9.3 CP 525 User Program for PC 2 (150U)

Nothingneeds changing intheexisting CP525 program, since no job is required on PC 2.

Switch the CP 525 to "RUN". The red LED on device interface 1 on CP 525 must now go off. If it does not, check whether the interpreter and procedure actually exist or if there is a CP 525 hardware error.

9.4 STEP 5 User Program for PC 2 (150U)

The transfer of the data received by the CP 525 to data block DB 11 is carried out by the RECEIVE ALL which is called cyclically. RECEIVE ALL and data block DB 11 are already programmed; therefore nothing needs to be changed in the STEP 5 pr-.

Switch the CPU to "RUN". The green RUN LED must light up.

9.5 Test

Observe the data word DW 0 of data block DB 11 in PC 2 (150U) with the PG 685 online function CONTROL VANABLE.

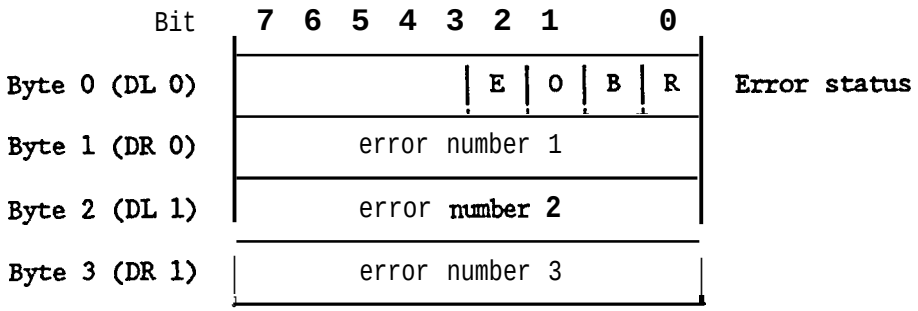
Switch the digital input I 0.3 on PC 1 (135U) from 0 to 1. The bit pattern of input byte IB 0 appears in the higher order part of data word DW 0 (left-hand data DL 0). The data to the right DR 0 is always overwritten with 0.

If you do not have any digital 1/0 modules with a simulator available in PC 1 (135U), then use flag byte FY 0 instead of the input byte IB 0. Set (or reset) the flags with the PG 685 online function CONTROL VARIABLE. Observe the condition codeword FW 30 and the parameter assignment error bytes FY 34 with CONTROL VARIABLE.

10 Reading the Error Message Area in the SYSTAT

The SYSTAT is a memory area on the CP 525 that contains status information separately for each interface.

Within the error message area of the SYSTAT the causes of errors are coded in detail. The error message area is 4 bytes or 2 words long and appears as follows



E - set, if error is entered in the SYSTAT
O - set, if there is an error overflow (more than 3 errors)
B - set, if there is a BREAK on the interface
R - irrelevant for the computer link

The PC job RECEIVE DIRECT 200 is reserved for reading the error message area. This job is carried out immediately; i.e. , it is not entered in the CP 525 queue and does not require a RECEIVE ALL. No job needs programming on the CP 525.

10.1 STEP 5 user Program for PC 1 (R Processor)

The error information is to be transferred to data block DB 10 into data words DW 9 and DW 10. The job is called unconditionally in every program cycle but executed only if an error occurs. It must be called by the result of logic operation (RLO) 1.

When RESET 200 is called, the error message area of the interface addressed (in this example SSNR = 2 for device interface IF 1 of the CP 525) is reset. The data words DW 9 and DW 10 in data block DB 10 in which the error numbers are written by RECEIVE 200, must be reset separately.

The error message area is reset by a positive going edge at digital input I 0.4.

Extend the organization block OB 1 by the following statements:

```
      :O   F 3.0
      :ON F 3.0              GENERATE RLO = 1
      :
      :JU  FB121             READ ERROR MESSAGE AREA SYSTAT
NAME :RECEIVE
SSNR :     KY0,2             INTERFACE NUMBER 2
A-NR :     KY0,200           JOB NUMBER 200
ANZW  :  FW35                CONDITION CODEWORD
ZTYP :     KSDB              THE DEST. IS A DATA BLOCK
DBNR :     KY0,10            DATA BLOCK NUMBER 10
ZANF :     KF+9              FROM DATA WORD 9
ZLAE :     KF+2              2 DATA WORDS
PAFE :     FY39              PARAMETER ASSIGNMENT ERROR BYTE
      :
      :O   F 36.3            JOB TERMINATED WITH ERROR
      :O   F 39.0            OR PARAMETER ASSIGNMENT ERROR
      :=   QO.4              INDICATEDAT  OUTPUTQ0.4
      :
      :AN  10.4              EDGE EVALUATION:
      :R   F2.4              THE RESULT OF LOGIC OPERATION
      :A   I 0.4             (RLO) IS SET FOR ONE CYCLE IF THE
      :AN  F2.4              SIGNAL AT INPUT I 0.4 CHANGES
      :S   F2.4              FROM 0 TO 1.
      :
      :JC FB100
NAME :R-SYSTAT
```

Function block FB 100

```
      :JU  FB124              RESET SYSTAT
NAME :RESET
SSNR :     KY0,2              INTERFACE NUMBER 2
A-NR :     KY0,200            JOB NUMBER 200
PAFE :     FY40               PAwMETER ASSIGNMENT ERROR BYTE

      :C   DB10               IN DATA BLOCK DB 10
       :L  KF+0               DELETE DATA WORDS DW 9 AND DW 10
      :T   DW9
      :T   DWIO
      :BE
```

Transfer the organisation block OB 1 and function block FB 100 to the user memory of the R processor. Perform a cold restart. The green RUN LED must light up.

10.2 Test

Observe the data block DB 10 DW 9 and DW 10 with the PG 685 online function CONTROL VARIABLE and disconnect the cable connector between the two CP 525s.

In data word DW 9, KH = 0EFF appears, and in DW 10, KH = FFFF appears, (BREAK). This error number is signalled repeatedly until the situation has been resolved. For this reason the error buffer in the SYSTAT overflows; bit 2 in the first byte is set. Transfer jobs are terminated with error FH in the ANZW.

Reconnect the two CP 525s. Reset the SYSTAT error message area by switching input I 0.4 from 0 to 1. Check that it has been reset using CONTROL VARIABLE. Initiate the SEND job with job number 1 on PC 1 at input I 0.1.

Switch the mode selector of the CP 525 in the S5-150U to STOP and trigger the SEND job with the job number 1 on PC 1 (135U). KH = 0838 appears in data word DW 9 of DB 10, i.e. the error message that the partner CP is switched to "PGR" or "STOP".

11 Date/Time PC Jobs

The CP 525 has an internal battery-backed hardware clock that continues to run even if there is a power failure, If necessary it can be used as a calendar and timer. The clock can be set, read and synchronized with the hardware clocks of other CPS in the same PC.

The PC job number 218 is reserved for reading and writing the date and time on the CP 525. This special job is executed immediately; i.e., it is not entered in the job buffer of the CP 525 anddoesnotrequire a SEND ALL or RECEIVE ALL. The job is initiated and the transfer carried out in one call. You do not need to program a job on the CP 525.

11.1 Read Date/Time

The current date and time values are to be transferred cyclically from the CP 525 hardware clock to data block DB 10 from data word DW 11. In DB 10 they are stored in the following format:

11 :	KH= 0100	master identifier
12 :	KH= mmss	1/10 s / 1/100 s / seconds
13 :	KH= mmhh	minutes / hours
14 :	KH= ddmm	day / month
15 :	KH= yy00	year / --

When the time is to be transferred to a master, which is to remain a master and which can read the time, the master identifier must be set. When the power is switched on the master identifier on the CP 525 is reset. It is, however, not influenced by the SYNCHRON function block. The setting of the master identifier is described in Sections 11.2 and 11.3.

The transfer is to take place in every cycle. The termination of the job with an error is tobe indicated at output Q 0.5.

Extend OB 1 by the statements below to trigger a corresponding PC job:

```
     :O   F 3.0
     :ON F 3.0              RLO = 1
     :
     :JU  FB121
NAME :RECEIVE
SSNR :    KY0,2             INTERFACE NUMBER 2
A-NR :    KY0,218           JOB NUMBER 218
m    :    FW41              CONDITION CODEWORD
ZTYP :    KSDB              THE DEST. IS A DATA BLOCK
DBNR :    KY0,10            DATA BU3CK NUMBER 10
ZANF :    KF+11             FROM DATA WORD 11
ZLAE :    KF+5              5 DATA WORDS
PAFE :    FY45              PARAMETER ASSIGNMENT ERROR BYTE
     :
     :O   F 42.3            JOB TERMINATED WITH ERROR
     :O   F 45.0            OR PARAMETER ASSIGNMENT ERROR
     :=   QO.5              INDICATED AT OUTPUT Q 0.5
```

Transfer the extended organization block OB 1 to the R processor and perform a cold restart. The green "RUN" LED must light up.

Test

Observe data block DB 10 fromDW 11 to DW 15 with the PG 685 online function CONTROL VARIABLE. The date and time are not entered because the master identifier is not set after the CP 525 has started following power up.

The job is terminated without errors in the condition codeword; nothing is entered in the destination data block.

11.2 Setting the Date/Time

Enter the value to be set (31st December, 1987, 23 hours, 58 minutes and 30 seconds) in data block DB 10 from data word DW 16 in BCD format. The CP 525 is to be the master. The master identifier (bit 8) in data word DW 16 must be set (see also Sections 11.1 and 11.3) .

```
16 :    KH= 0100          master identifer set
17 :    KH= 0030          1/10 s, 1/100 s / seconds
18 :    KH= 5823          minutes/hours
19 :    KH= 3112          day/month
20 :    KH= 8700          year/--
```

The set function is to be triggered by a signal change from 0 to 1 at input bit I 0.6. If the job is terminated with an error, the error is indicated at output Q 0.6.

Extend OB 1 with the statements below to trigger a corresponding PC job:

```
     :AN  10.6          EDGE EVALUATION:
     :R   F 2.6         THE RLO IS SET FOR ONE CYCLE IF
     :A   10.6          THE SIGNALAT INPUT I 0.6 CHANGES
     :AN  F 2.6         FROM 0 TO 1.
     :S   F 2.6
     :
     :JU  FB120         CALL PC JOB SET TIME
NAME :SEND
SSNR:     KY0,2        INTERFACE NUMBER 2
A-NR :    KY0,218      JOB NUMBER 218
ANZW :    FW46         CONDITION CODEWORD
QTYP :    KSDB         SOURCE IS ADATA BLOCK
DBNR :    KY0,10       DATA BI.XX'KNUK8ER10
QANF :    KF+16        FROM DATA WORD 16
QLAE :    KF+5         5 DATA WORDS
P m :     FY50         PARAMETER ASSIGNMENT ERROR BYTE
     :
     :O   F 47.3       JOB TERMINATED WITH ERROR
     :O   F 50.0       OR PARAMETER ASSIGNMENT ERROR
     :=   Q0.6         INDICATEDAT  OUTPUTQ0.6
```

Transfer organization block OB 1 and data block DB 10 to the user memory of the R processor. Perform a cold restart. The green "RUN" LED must light up.

Test

Switch the digital input I 0.6 from 0 to 1. The master identifier, date and time of the CP 525 hardware clock is set to the time programmed in data block DB 10. The success of the operation can be recognised by the date and time values that are written cyclically to data words DW 11 to DW 15; tenths of seconds, seconds and minutes must change continuously. Shortly afterwards the CP 525 indicates a change of year. The CP 525 recognizes a change of year and also leap years; it counts 28 or 29 days in February correctly.

11.3 Synchronizing Several CP Hardware Clocks

To synchronize the hardware clocks of several CP modules, the CPU must read the date and time from a CP declared as time master and transfer these values to other communications processors which are declared as time slaves.

Reading the time is carried out with a RECEIVE DIRECT 218 (see Section 11.1) and setting the time with a SEND DIRECT 218 (see Section 11.2).

A CP is declared master by means of a SEND DIRECT 218 if bit 8 is set in the first data word transferred. If this bit is 0 the CP is a time slave.

If only one data word is transferred with SEND DIRECT 218 (QLAE = 1), only the master identifier is affected; the time remains unchanged. If five data words are transferred with SEND DIRECT 218 (QLAE = 5) the master identifier, date and time are transferred. The hardware clock is set correspondingly and continues to run with the new value.

For example, suppose there are three CP 525 modules plugged into an S5-135U with R processor. They have the module addresses 0, 2 and 4. The CP 525 with module address 2 is already declared master and has the correct time.

The date, time and master identifier are read by the master clock (interface 2) in every program cycle in the command string from Section 11.1 and written into data block DB 10 from data word DW 11. Before the date and time of the master clock can be transferred to the slave clocks (interfaces 0 and 4), the master identifier in data block 10 data word DW 11 must be reset. Carry out the synchronization on a positive going edge at input I 0.5.

To have an exact synchronization, the time read by the master clock must be transferred to the slaves in the same program cycle; it is, however, not essential that the two slaves are synchronized in the same program cycle.

Following the 0->1 edge at input I 0.5, the initiation flags F 3.2 for interface 0 (SSNR 0) and F 3.3 for interface 4 (SSNR 4) are set.

The two set jobs (SEND DIRECT 218) are executed only if the corresponding initiation flags are set and the read job (RECEIVE DIRECT 218) has been executed. You can recognise that the time has been read successfully because the master identifer that is reset in each cycle in DB 10 DW 11 is set immediately following the RECEIVE DIRECT call. Once a slave has been synchronized (complete without error), the initiation flag is reset.

Output Q 0.5 indicates any errors that occur when the error clock is being read. Output Q 0.7 indicates any errors that occur when the slaves are being set. Normally in this situation en error evaluation program tailored to the particular application is called up. In the case of errors, evaluate the error numbers in the condition codewords ANZW, parameter assignment error bytes PAFE and the error message area of the SYSTAT for the different interfaces (see Section 10).

The program in organization block OB 1 for synchronizing the three CP 525s appears as follows

```
     :O   F 3.0                                          *)
     :ON F 3.0              RLO = 1                      *)

     :Ju FB121                                           *)
NAME :RECEIVE                                            *)
SSNR :    KY0, 2           INTERFACE NUMBER 2           *)
A-NR :    KY0, 218         JOB NUMBER 218               *)
ANZW :    FW41             CONDITION CODEWORD           *)
ZTYP :    KSDB             DEST. IS A DATA BLOCK        *)
DBNR :    KY0, 10          DATA BLOCK NUMBER 10         *)
ZANF :    KF+11            FROM DATA WORD 11            *)
ZLAE :    KF+5             5 DATA WORDS                 *)
PAFE :    FY45             PARAMETER ASSIGNMENT ERROR BYTE *)

     :O   F 42.3           JOB TERMINATED WITH ERROR      *)
     :O   F 45.0           OR PARAMETER ASSIGNMENT ERROR  *)
     :=   Q 0.5            INDICATED AT OUTPUT Q 0.5      *)

     :C   DB10
     :A   D 11.8           IF MASTER IDENTIFIER 1, THEN
     :=   F 3.1            SET "READING SUCCESSFUL" BIT
     :R   D 11.8           AND RESET MASTER IDENTIFIER

     :AN  I 0.5            EDGE EVALUATION:
     :R   F 2.5            THE RLO IS SET FOR ONE CYCLE IF
     :A   I 0.5            THE SIGNAL AT I 0.5 CHANGES
     :AN  F 2.5            FROM 0 TO 1
     :S   F 2.5

     :S   F 3.2            SET INITIATION FLAG SSNR 0
     :S   F 3.3            SEX' INITIATION FLAG SSNR 4
```

*) Already programed in Section 11.1

```
      :
      :A   F 3.2          IF INITIATION FLAG SET AND
      :A   F 3.1          DATE/TIME READ, SET SLAVE
      :JU  FB120          CLOCK SSNR 0
W     :SEND
SSNR  :    KY0,0          INTERFACE NUMBER 0
A-NR  :    KY0,218        JOB NUMBER 218
ANZW  :    FW51           CONDITION CODEWORD
QTYP  :    KSDB           SOURCE IS A DATA BLOCK
DBNR  :    KY0,10         DATA BLOCK NUMBER 10
QANF  :    KF+11          FROM DATA WORD 11
QLAE  :    KF+5           5 DATA WORDS
PAFE  :    FY55           PARAMETER ASSIGNMENT ERROR BYTE

      :A   F 52.2         SET SSNR 0 COMPLETE WITHOUT ERROR,
      :R   F 3.2          THEN RESET INITIATION FLAG

      :A   F 3.3          IF INITIATION FIAG SET AND
      :A   F 3.1          DATE/TIME READ, SET SLAVE
      :JU  FB120          CLOCK SSNR 4
NAME  :SEND
SSNR  :    KY0,4          INTERFACE NUMBER 4
A-NR  :    KY0, 218       JOB NUMBER 218
ANZW  :    EW56           CONDITION CODEWORD
QTYP  :    KSDB           SOURCE IS A DATA BLOCK
DBNR  :    KY0, 10        DATA BLOCK NUMBER 10
QANF  :    KF+11          FROM DATA WORD 11
QLAE  :    KF+5           5 DATA WORDS
PAFE  :    FY60           PARAMETER ASSIGNMENT ERROR BYTE
      :
      :A   F 57.2         SET SSNR 4 COMPLETE WITHOUT ERROR,
      :R   F 3.3          THEN RESET INITIATION FLAG

      :O   F 52.3         JOB SSNR 0 TERMINATED WITH ERROR
      :O   F 55.0         OR PARAMETER ASSIGNMENT ERROR
      :O   F 57.3         JOB SSNR 4 TERMINATED WITH ERROR
      :O   F 60.0         OR PARAMETER Assignment' ERROR
      :=   Q 0.7          INDICATED AT OUTPUT Q 0.7
```

The values set could also have been sent to interface number SSNR 1 and 5 since the CP 525 has only one hardware clock for both interfaces.

12 Info

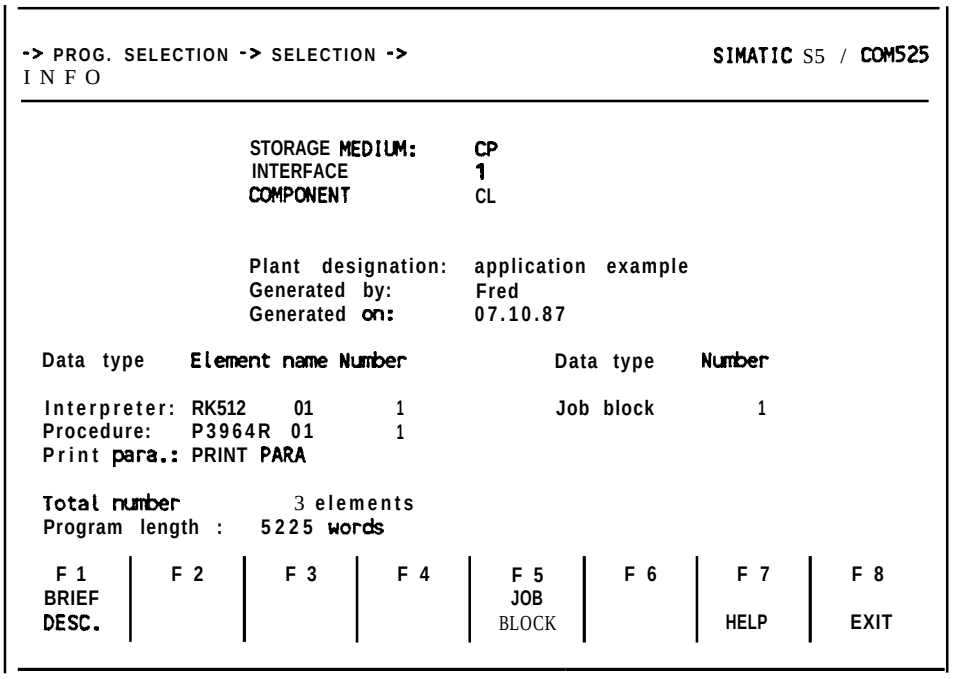
Using the COM 525 programming software you can obtain information about user data which has already been programmed and have it displayed on the PG monitor.

Starting from the 'SELECTION' mask, press F4 to call the ' INFO mask. In this mask you can further decide whether you require information about user data on the CP or a program file (FD).

The data programmed on the CP for device interface 1 can be output on the monitor.

```
-> PROG. SELECTION -> SELECTION ->                SIMATIC S5 / COM525
I N F O

            STORAGE MEDIUM:  CP
            INTERFACE:       1
            COMPONENT:       CL

            Plant designation: application example
            Generated by:      Fred
            Generated on:      07.10.87

 F 1   | F 2 | F 3 | F 4 | F 5   | F 6 | F7   | F 8
 BRIEF |     |     |     | JOB   |     |      |
 DESC. |     |     |     | BLOCK |     | HELP | EXIT
```

Press F1 (BRIEF DESC.) to obtain a brief description of the CP 525 interface 1.



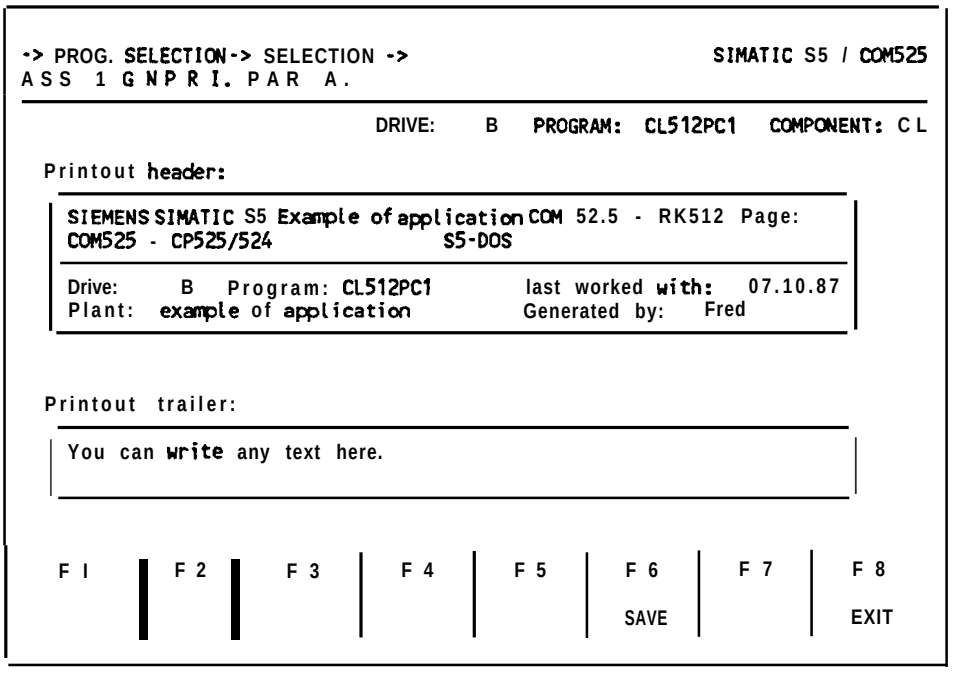
In this mask you can read the interpreter and procedure names as well as the version number and see whether a job block has been programmed or not. In addition you can see the program length displayed in words.

Press F5 (JOB BLOCK) in the 'INFO' mask to display a table of all the programmed PC job numbers. With F1 (ON PUNTER) in the 'JOB BLOCK' mask you can have this information printed out on a PT88 or PT89 printer connected to the PG.

13 Program Documentation

All the elements of a COM 525 program file can be documented in detail. The printouts can have either a header or a trailer programed that is output on a PT88 printer.

In the 'SELECTION' mask F6 (CONTINUE) and F3 (ASSIGN PRI. PAPA.) call up the following mask



For the printout header and trailer you can program any two lines of text.

Ress F6 to save the printout header and trailer in your user program file CL512PC1.525.

Press F6 (CONTINUE) and F4 (LISTING) in the 'Selection mask to call the 'LISTING' mask. Using the keys

- F1 (TOTAL PROGRAM)
- F2 (INTERPRT PROCEDURE)
- F4 (JOB BLOCK)

you can list the whole program or parts of it.

Try out the various options.

By pressing F4 the following list of all the programmed PC jobs will be printed out on the printer connected to the PG 685:

```
+----------------------------------------------------------------------------+
! SIEMENS SIMATIC S5 Example of application COM 525 ... Page:            1 !
! COM525 - CP525/524                RK512                          10.30 ..s7 !
+----------------------------------------------------------------------------+
! Drive: B            Program: CL512PC1        Last worked with:   10.30.87 !
! Plant: application● .wrlp.le                 Generated by:    Fred
+----------------------------------------------------------------------------+
```

JOB BLOCK

Name: JOB BLOCK

A-NR.	JOB	JOB TYPE	CPU NO.	DB/DX NO.	SOURCE ADDR. HEX.	SOURCE ADDR. DEC.	DEST. ADDR. HEX.	DEST. ADDR. DEC.	COORD. FLAG
001	Send	Data block		011	-	-	0004	00004	-
OD2	Fetch	Data block		011	0004	00004	-	-	-

```
+----------------------------------------------------------------------------+
! You can write any text here
+----------------------------------------------------------------------------+
```

SIEMENS

We have checked the contents of this manual for agreement with the hardware and software described. Since deviations cannot be precluded entirely, we cannot guarantee full agreement. However, the data in this manual are reviewed regularly and any necessary corrections included in subsequent editions. Suggestions for improvement are welcomed.
Technical data subject to change.

Siemens Aktiengesellschaft

C79000-B8576-C540-05
Gerätewerk Karlsruhe
Printed in the Federal Republic of Germany

SIEMENS

SIMATIC S5

CP 525: Event Output and Listing with the PT88/PT89 Printer

Example of Application

Order No. C79000-B8576-C542-05

Contents

1 Introduction

This is an example of an application using the communications processor CP 525 for listing and monitoring technical processes. When you have worked through this example you will have a finished CP 525 user program in which all the most important functions are implemented.

To start with, the required hardware settings are explained. After this, not only the generation of a CP 525 user program with the COM 525 programing software but also the corresponding STEP 5 program in the CPU will be explained.

For this example you require the following hardware

one CP 525 module with RAM memory submodule
- one PC 685 programmer
- one cable connector PC <--> CP 525
- one PT88 or PT89 printer without memory expansion
- one cable connector PT88 <--> CP 525
- one S5-135U programmable controller
- one R processor with RAM memory submodule
- one cable connector PG <--> CPU

and if possible

- one digital input module (24 V, 16 channels)
- cme digital output module (24 V, 16 channels)
- one SIMATIC S5 simulator

the following software is also required

- the S5-DOS programming package COM 525
- the STEP 5 basic package for the PC 685 programmer (supplied with the programmer)
- handling blocks for the R processor
- standard function blocks 'signaling functions for standard CPs'

2 Aims

First of all, you will set up the hardware. After this, you will install the software necessary for writing your user programs.

The first step is to program a PC job for a form feed.

Based on the example of a process status list (PSL), current process values will be output on the printer; all possible s5 formats will be used.

You will program 11 process messages to be output on the printer when a particular digital input is set. One message will have no process variables and the others will each have one. All possible S5 formats will be used. The messages will include the time and message status (coming or going) and will be output whenever the signal at the corresponding digital input changes (sequential message list - SML).

Using a current message list (CML) you can list all the messages on the printer which are still 'active at the time the CML is called.

The sequential message list, current message list and process status list may also have frames, i.e. , header and trailer. The programming of frames is also explained based on an example.

How to set and read the CP 525 hardware clock and read the error message area in the SYSTAT will also be demonstrated.

Finally, the example covers the information and documentation facilities provided by the COM 525 programming software.

The texts of the messages of the process status list and the frame are intended to illustrate the CP 525 function. For this reason no terminology from process engineering w-ill be used.

3 Hardware

As the programmable controller, an S5-135U with R processor is used.

Fundamentally the procedure for the S5-115U,S5-135U, S5-150U and S5-155U programmable controllers is identical. Nothing changes in the CP 525 user program. The handling blocks and standard function blocks "signaling functions for standard CPs" have different numbers for different programmable controllers but they are identical in terms of programming.

The CP 525-2 cannot be Used in conjunction with tile S5 135U s processor.

Plug the R processor into one of the CPU slots (11, 19, 27 or 35) in the S5-135U. Carry out an overall reset of the CPU and switch the mode selector to "STOP".

On the CP 525, set the module address to zero; i.e. , no jumpers are plugged in at jumper block 16. All the CP 525 printer functions can be implemented without IPC flags. You should, therefore, inhibit all the IPC flags to avoid multiple addressing when using several CPs and/or a coordinator module. Remove all the jumpers on jumper block 25. Plug the CP 525 into one of the slots (11, 19, 27, 35, 43, 51, 59 or 67) in your S5-135U. Switch the mode selector to "PGR" (programming) .

On the digital input module and the digital output module, set the module address to 0. Input bytes IB 0 and IB 1 and output bytes QB 0 and QB 1 are then available. The ties can be plugged into any slot on the S5-135U. Connect the simulator to the 1/0 modules.

If you have no I/O modules or no simulator available, then instead of using IB 0 and IB 1, use flag bytes FY 0 and FY 1. These flags can be set and reset using the STEP 5 online function CONTROL VARIABLE.

Connect the printer to interface 1 (IF 1) of the CP 525.

Connect the PG 685 either to the programming interface (IF 2) of the CP 525 or to the programing interface of the R processor, (depending on which module you wish to program), using the correct cable connector. The mode selector on the CP 525 must be set to PGR during programming.

3.1 Assigning Parameters to the Printer

The coding switch cm the central controller of the printer is located under the front hinged cover of the printer in front of the platen. Here the following settings must be made:

	1	2	3	4	5	6	7	8	9	10	
ON		X		X	X	X			X	X	PT88
OFF	X		X				X	X			

	1	2	3	4	5	6	7	8	9	10	
ON		X		X	X	X			X		PT89
OFF	X		X				X	X		X	

Among other things, the printer is then set for the German character set, even parity and 12" paper length. (For other character sets refer to the description of the PT88/PT89).

The printer has different interface adapter cards depending on whether it is intended for use with TTY, V.24 or combined TTY/V.24 signals. To check the settings on these cards, you must completely remove the cover of the printer.

The mode switches on the interface adapter SAP-S1 (V.24/V.28) must be set as follows for this example:

Switch 1

	1	2	3	4	5	6
ON	X	X	X			X
OFF				X	X	

Switch 2

	1	2	3	4	5	6
ON		X		X		
OFF	X		X		X	X

If your printer has the interface adapter SAP-S2 (TTY/20 mA) the mode switches must be set as follows:

Switch 1

	1	2	3	4	5	6	7	8
ON	X	X	X	X		X		
OFF					X		X	X

Switch 2

	1	2	3	4	5	6	7	8
ON			X	X			X	X
OFF	X	X			X	X		

With the mode switch S4 on the combined interface adapter SAP-S3 (V.24/TTY), you can select between V.24 (switch position 1) and TTY (switch position 2). The settings for the mode switches S1, S2 and S3 must be as follows for this example:

Switch 1

	1	2	3	4	5	6	7	8
ON	X	X	X	X			X	
OFF					X	X		X

Switch 2

	1	2	3	4
ON	X	X	X	X
OFF				

Switch 3

	1	2	3	4
ON				
OFF	X	X	X	X

The settings of switches 5 and 6 are irrelevant.

In all three cases the printer is set to a data rate of 9600 bps and to use the X-ON/X-OFF protocol.

The cable connector CP 525 <--> PT88/89 is always the same regardless of which interface adapter is used.

4 COM 525 Programming Software

4.1 Installing COM 525 on the PG 685

A PG 685 programmer must be available in which the STEP 5 basic package has already been installed,

Go to drive B: user number 0 by entering

```
0:
```

Copy the contents of the three COM 525 floppy disks with

```
PIP B:=A:*.*[R V]

Option V = verify copying
       R = copy SYS files
```

onto the hard disk, user number 0.

Assign the system and read only attributes to the files with

```
SET S5?EC?5X. CMD[SYS RO]
SET S5PEP05X. CMO[SYS RO]
SET COMLIB*. 525[SYS RO]
```

You can work with COM 525 on any user area.

Exit user number 0, which should be reserved for system files, with

```
n:      (n = required user number).
```

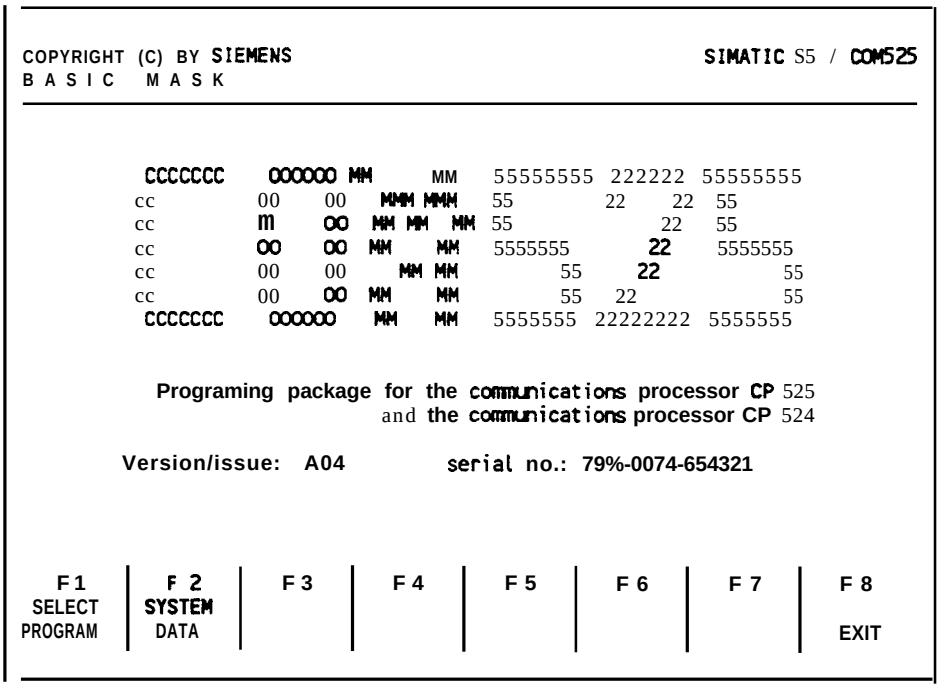
4.2 Calling COM 525 and Program Selection

By entering

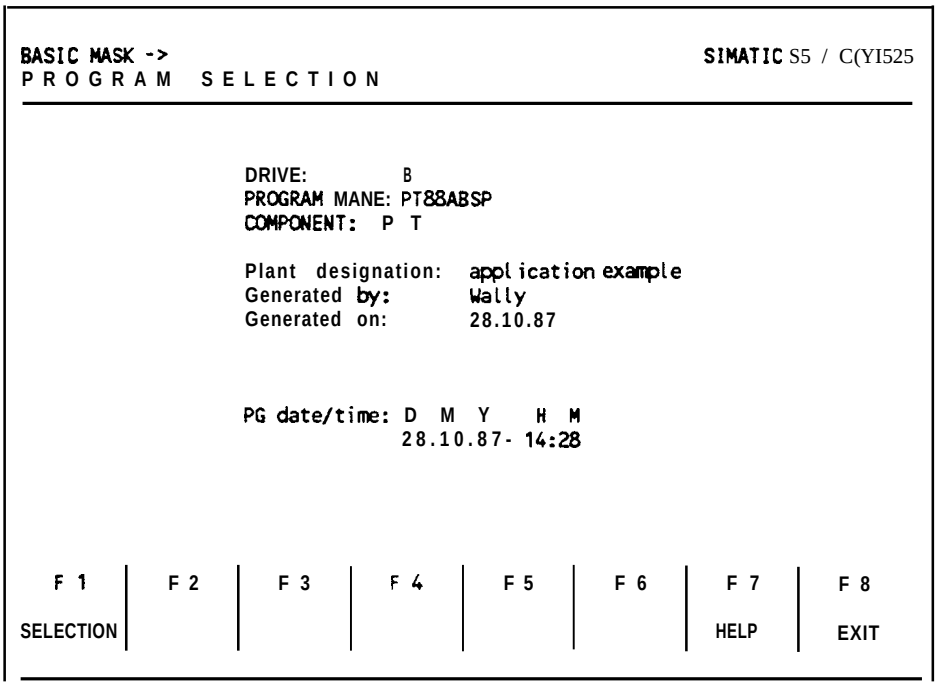
S5

you call the SIMATIC programing packages. Place the cursor in the line 'C O M 5 2 5 . . . 'and use the function key F1 (PACKAGE) to select the COM 525 programming software for the CP 525.

The COM 525 basic mask then appears as follows:

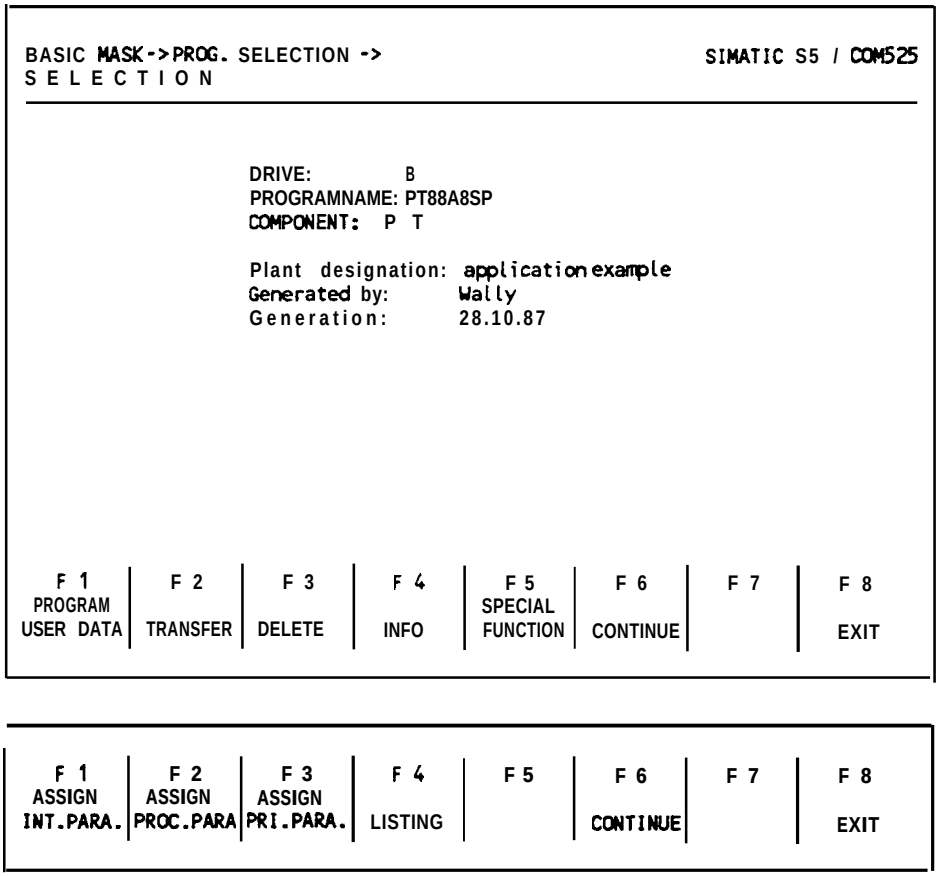


Press F1 (SELECT PROGRAM) to call the mask 'PROGRAM SELECTION' .



A program name (in this case "PT88ABSP") and the component "PT" for the event output and listing function must be entered. As plant designation enter “application example” and enter your own name beside ‘Generated by”.

In the 'SELECTION' mask, which you call with F1 (SELECT), you can select the basic COM 525 functions.



With the COM 525 programming software the user program is first generated on floppy or hard disk (storage medium FD) and then transferred to the CP 525 user program where it is then tested.

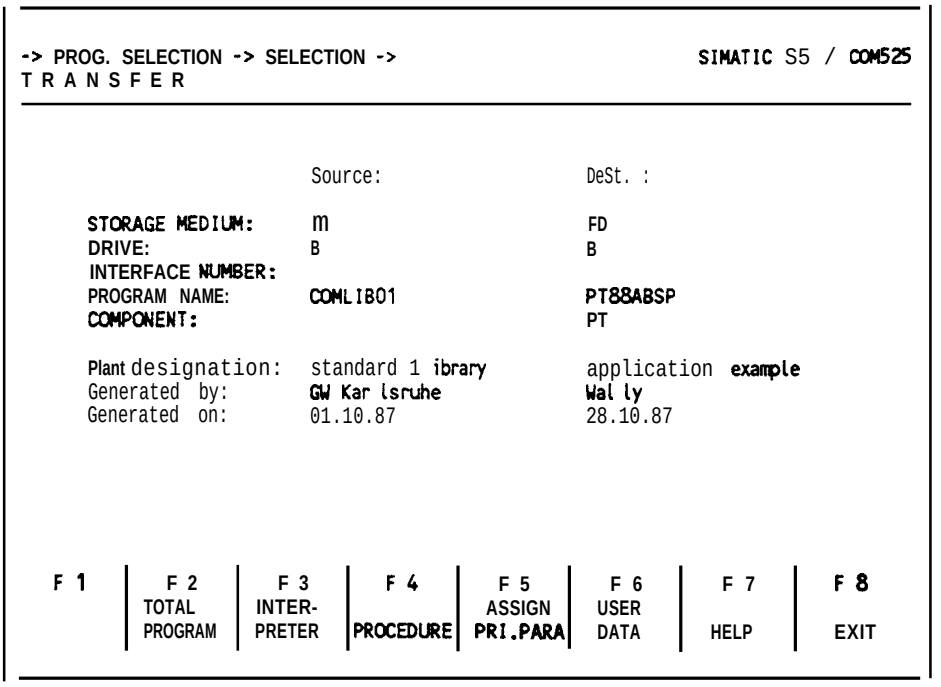
The S5-DOS file on the storage medium FD automatically has the program name .525, in this case PT88ABSP. 525.

4.3 Interpreter and Procedure

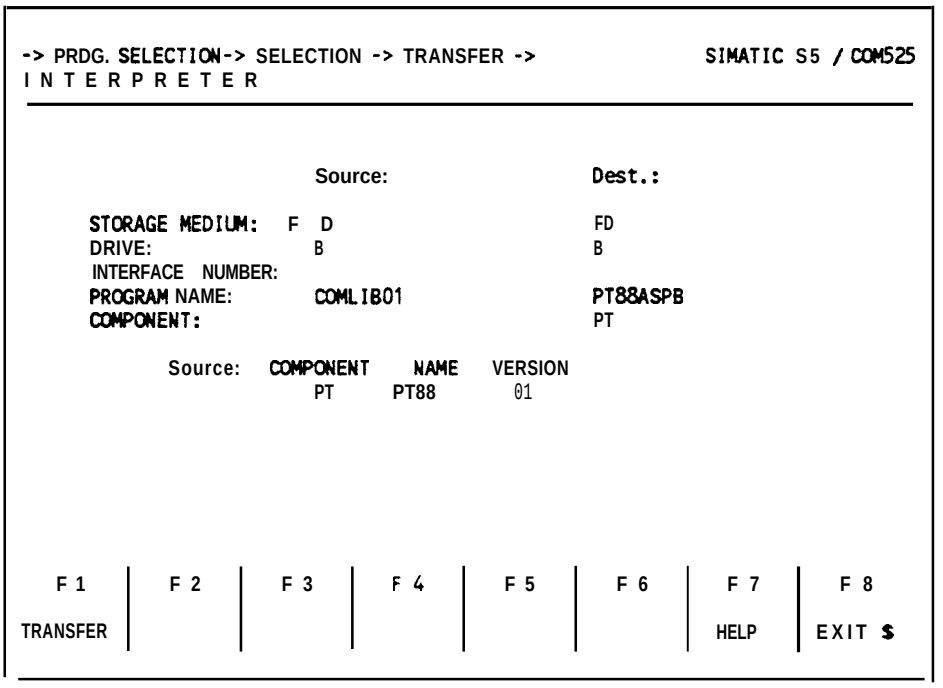
4.3.1 Copying from the Library into the User Program

The interpreter and procedure must first be copied into the user program. They are located in the library COMLIB. n (n = version number), which is part of the COM 525 software (in drive B, user 0).

Ress F2 (TRANSFER) and F5 (FD->FD) in the ' SELECTION' mask to call the function for transferring from FD to FD. Press F7 (HELP) in the 'TRANSFER' mask and enter drive "B" as the source and program name "COMLIB01" (the library in our example has version number 01). The destination is automatically the program PT88ABSP selected in the 'PROGRAM SELECTION' mask.



Press F3 to call the 'INTERPRETER' mask. By pressing F7 (HELP), in this mask the interpreter to be transferred "PT88" (component "PT") is selected. F1 (TRANSFER) starts the transfer. The successful trsnsfer is displayed with 'MESS.002 completed' .



With F8 (EXIT) you return to the 'TRANSFER' mask.

With F4 (PROCEDURE) you call the 'PROCEDURE' mask. The procedure "LAUFPT88" is transferred in the same way as the interpreter.

Press F8 (EXIT) twice to return to the 'SELECTION* mask from the 'PROCEDURE' mask.

4.3.2 Parameter Assignment

With F6 (CONTINUE) in the 'SELECTION' mask, call the second set of function keys .

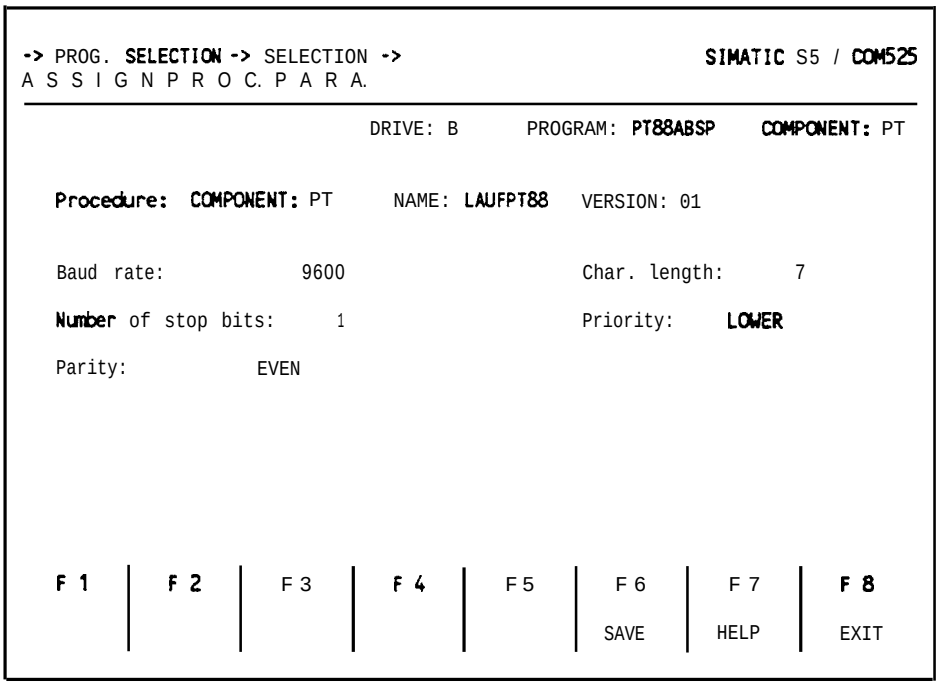
F1 (ASSIGN INT. PARA.) branches to the 'ASSIGN INTERP. PARA.' mask. In this mask, the interpreter belonging to program PT88ABSP is displayed. The mask already has standard default values entered. Most of these are used in this program.

The entry in the "printer type" field may have to be changed to "N" (needle head). The separator for the time should be changed to ":" and the character set to 'ENGLISH". All the entries can be made using F7 (HELP). Using F6 (SAVE) you write the interpreter to the hard disk. 'MESS. 003: saved ! ' indicates that this is complete.

```
-> PROG. SELECTION -> SELECTION ->                         SIMATIC S5 / COM525
ASS I GN I NT ERP. PAR A.
                                   DRIVE:    B   PROGRAM: PT88ABSP   COMPONENT: P T

   Interpreter:      COMPONENT: P T    NAME: PT88       VERSION: 01

                                        Presetting for interpreter PT88:

   Printer model:              PT 88 PT 88 or   PT 89
   Printer type:                   I  N=needle head       I=inkjet

   Lines per inch:                 6
   Page length:                   72 lines
   Page width:                   080 from 1 to80 characters per line
   Format (date):              O M Y D=day       M=month     Y=Year
   Format (time):              H M S  H=hour     M=minute    S=second
   Separator (date):
   Separator (time):               :
   Time format:       ENGLISH AM/PM    GERMAN 24 H,      ENGLISH AH/PM
   Character set: ENGLISH

   F1   | F2   | F3   | F4   | F5   | F6   | F7   | F8
        |      |      |      |      | SAVE | HELP | EXIT
```

With F8 (EXIT) you return to the 'SELECTION' mask.

With F2 (ASSIGN PROC. PARA) in the second set of function keys, call the 'ASSIGN PROC. PARA' mask. In this mask the procedure and version number belonging to our program PT88ABSP are displayed. You must enter the same parameters that were set on the printer with the switches (baud rate "9600", "1" stop bit, "EVEN" parity). These are also the standard defaults in the mask.

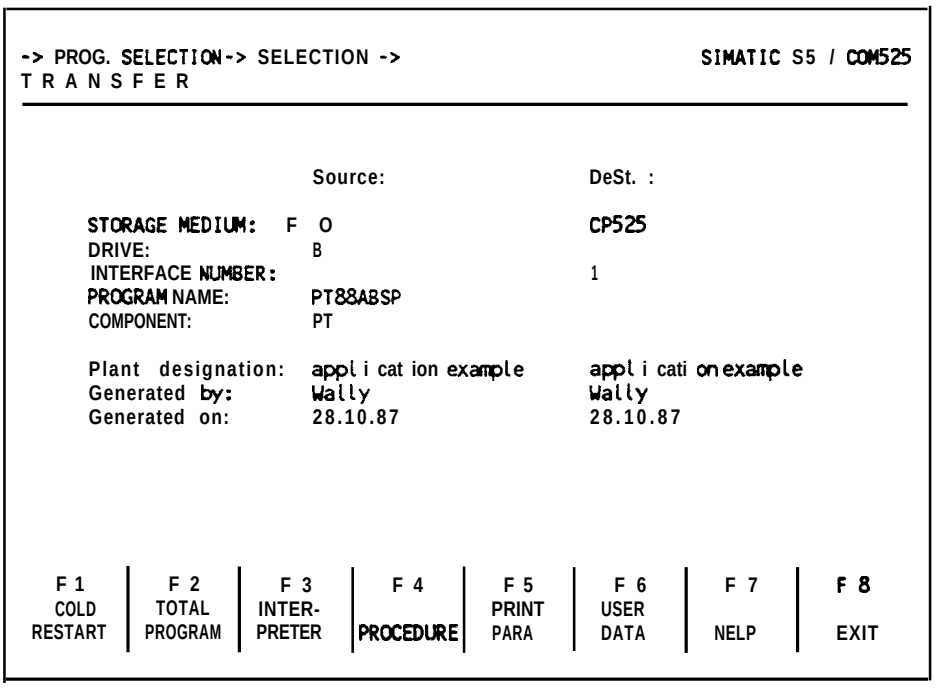


Ress F6 to save the procedure parameters on the hard disk and press F8 (EXIT) and F6 (CONTINUE), to return to the 'Selection mask.

4.3.3 Transfer to- CP 525 User Memory

The interpreter and procedure still must be transferred to the user memory of the CP 525. The CP 525 must be connected to the PC 685 and its mode selector must be set to "PGR" (programming).

Press F2 and F3 in the ' SELECTION' mask to call the function for transferring from FD to CP 525. In the 'TRANSFER' mask enter the CP 525 interface "1" as the destination. Make the transfer with F3 (INTERPRETER) and F4 (PROCEDURE).



'MESS.002 completed!' shows that the transfer has been completed.

!IMPORTANT!: when the interpreter and procedure are transferred to the CP 525 memory you must cold restart the CP 525 with F1 (COLD RESTART).

If you switch the CP 525 mode selector to "RUN", the red LED of interface 1 (IF 1) must go out.

All the preparations have now been made for programing the actual event output and listing functions.

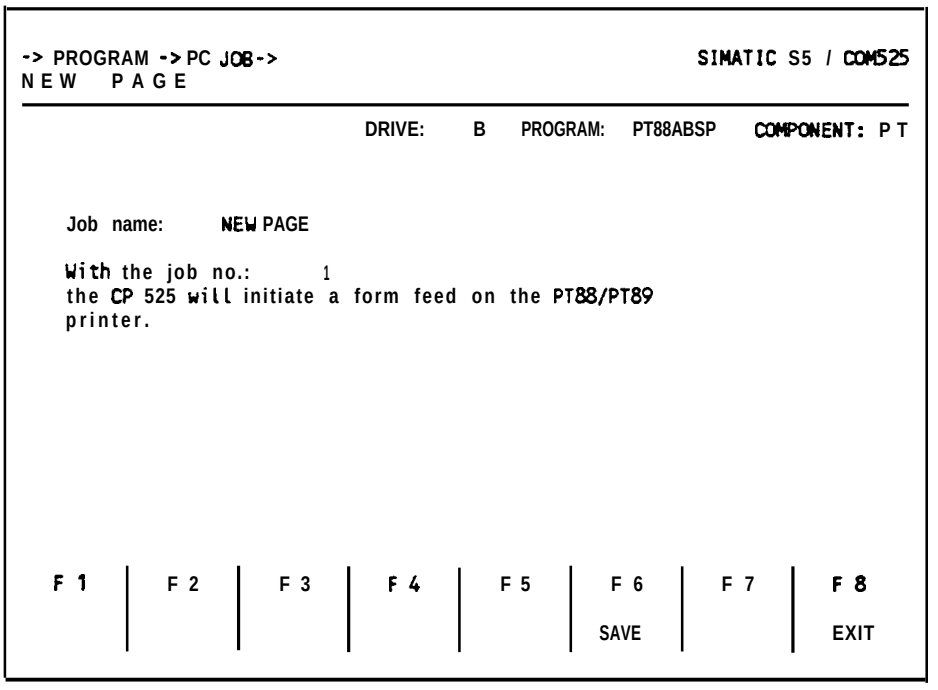
5 PC Job NEW PAGE

Using the PC job NEW PAGE the CPU can trigger a form feed on the printer.

5.1 CP 525 User Program

Switch the mode selector on the CP 525 to "PGR".

Press F1 (PROGRAM USER DATA) in the 'SELECTION' mask followed by F2 (PC JOB), F6 (CONTINUE) and F2 (NEW PAGE) to call the 'NEW PAGE mask. The job number must be entered with which the form feed will be initiated by the CPU. Select job number "1". The PC job is automatically named NEW PAGE.

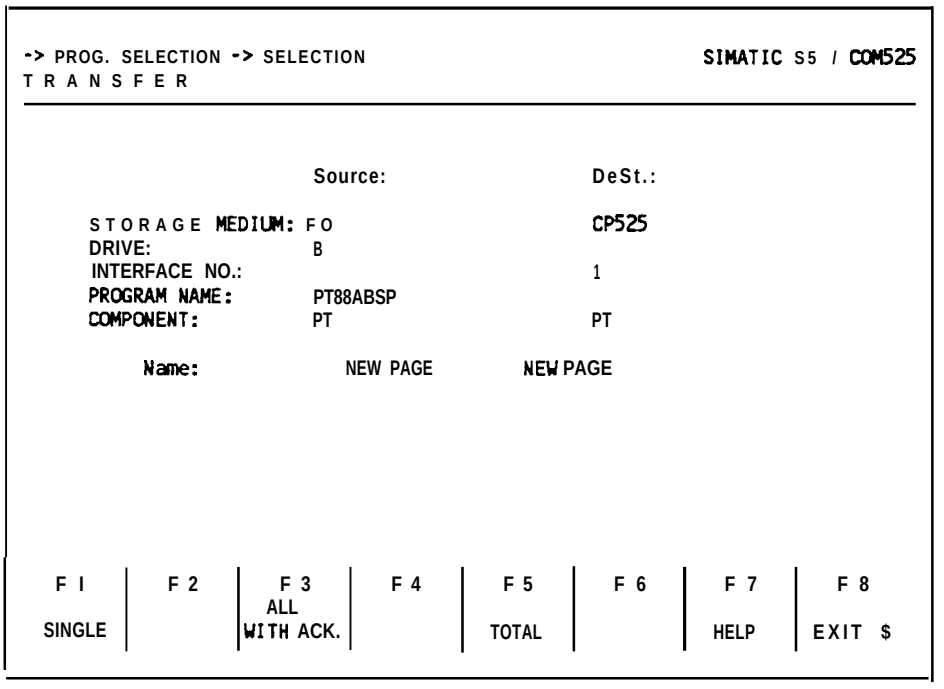


The PC job is saved on the hard disk with F6 (SAVE). Press F8 (EXIT) three times to return to the 'SELECTION' mask .

Press F2 (TRANSFER) and F3 (FD->CP) to transfer the PC job to the user memory of the CP 525. This is similar to transferring the interpreter and procedure (cf. Section 4.3 .3). Press F6 (USER DATA) in the 'TRANSFER' mask to display a new set of function keys .

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
COLD RESTART	MESSAGE	Pc JOB	FRAME				EXIT

Press F3 (PC JOB) and press F7 (HELP) in the 'PC JOB to enter the name of the PC job NEW PAGE in the input column of the source. The PC job is transferred with F1 (SINGLE),



With F8 (EXIT) return to the 'TRANSFER' mask. Carry out a cold restart on the CP 525 with F1.

Switch the CP 525 to the "RUN" mode. The red LED on interface 1 (IF 1) must now go off.

5.2 STEP 5 User Program

Call the S5 package LAD, CSF, STL. Program in STL (statement list) and call your STEP 5 program file V: PT88R1ST.S5D" . Transfer the handling blocks for the R processor into this program file. Connect the R processor to the PG 685.

The STEP 5 program must do the following:
- synchronize the CPU and CP 525 during the PC start-up
- trigger the form feed job.

5.2.1 Program Start-up

The start-up [redacted]tion blocks OB 20, OB 21 and OB 22 call the SYNCHRON handling block unconditionally.

```
     :JU  FB125                SYNCHRONIZATION CP <--> CPU
NAME: SYNCHRON
SSNR :    KY0,0                INTERFACE NUMBER 0
BLGR :    KY0 ,6               FIELD LENGTH 6 (MAX. 512 BYTES)
PAFE  :   FY10                 PARMETER ASSIGNMENT ERROR BYTE

     :BE
```

The interface number SSNR for interface 1 (IF 1) is the same as the module address "0". Select the largest field length '6". This means that data can be sent in strings of up 51.2 bytes (one complete data block) at one time between the CPU and CP. Use flag byte "FY10" as the parameter assignment error byte PAFE.

A parameter assignment error is to be indicated at digital output Q 0.0. In the start-up organization blocks, there is no process image so no digital 1/0s are addressed. Therefore this indication can be made only in the cyclic program (see Section 5.2.2). Normally an error evaluation program is called during start-up. If a parameter assignment error occurs, check whether the interface number and module address are the same. If they are correct, a hardware fault is suggested.

5.2.2 Cyclic Program

If a parameter assignment error occurs during the start-up, it is indicated at digital output Q 0.1 at the beginning of the cyclic program (see Section 5.2.1).

A PC job is triggered by a SEND DIRECT call; i.e. the handling block SEND DIRECT is called with a job number other than 0. The job number for the SEND DIRECT and the number of the job on the CP 525 must be the same. Therefore you program job number A-NR '1". The job is performed if a positive going edge appears at digital input I 0.3.

The interface number SSNR is "0"; "FW11" is selected as the condition codeword ANZW and "FY15" as the parameter assignment byte PAFE. The condition codeword requires two words for a SEND DIRECT call. The job status and any error messages that occur are indicated in flag word FW 11.

Since for a form feed no data needs to be transferred from the CPU to the CP 525, no source needs to be specified. "NN" must then be programmed as the source type QTYP, which means that the three following source parameters are not evaluated.

The job is triggered with the result of logic operation (RLO) 1. If RLO = 0 when the call is made, only the condition codeword will be updated.

The termination of the job with an error and the occurrence of a parameter assignment error is indicated only at digital output Q 0.1. In this situation an error evaluation program tailored to the particular application is usually called. If an error occurs, evaluate the error numbers in the condition codeword, the error message area of the SYSTAT (see Section 11) and in the parameter assignment error byte.

The corresponding program in the cyclic organization block OB 1 is then as follows:

```
      :A   F 10.0          IF SYNCHRON PARAMETER ASS . ERROR
      := Q 0.0             OCCURRED --> SET OUTPUT
      :
      :AN I 0.3            FLANK EVALUATION:
      :R   F 2.3           THE RESULT OF LOGIC OPERATION
      :A   I 0.3           (RLO) IS SET FOR ONE CYCLE IF THE
      :AN F 2.3            SIGNAL CHANGES FROM 0 TO 1 AT
      :S   F 2.3           INPUT I 0.3.
      :
      :JU  FB120           TRIGGER PC JOB NEW PAGE
NAME :SEND
SSNR :    KY0,0            INTERFACE NUMBER 0
A-NR :    KY0,1            JOB NUMBER 1
ANZW :    FW11             CONDITION CODEWORD
QTYP  :   KSNN             NO SOURCE SPECIFIED
DBNR :    KY0,0            NOT EVALUATED
QANF  :   KF+0             NOT EVALUATED
QLAE  :   KF+0             NOT EVALUATED
PAFE :    FY15             PARAMETER ASS. ERROR BYTE
      :
      :O   F 1.2.3         JOB 1 TERMINATED WITH ERROR
      :O   F 15.0          OR PAMMETER ASSIGNMENT ERROR
      := Q 0.1             INDICATED AT OUTPUT Q 0.1
      :
      :BE
```

Transfer the R processor handling blocks FB 120 to FB 127, OB 20, OB 21, OB 22 and OB 1 to the user memory of your R processor. Carry out a cold restart on the R processor.

5.3 Test

If you now switch the signal at digital input I 0.3 from zero to me, a new page command is output to the printer.

!Note! The CP 525 does not check whether a form feed has been carried out or even if the printer has been switched on. To avoid paper being wasted, a form feed only takes place following a printout or after you press the LF key on the printer.

If you have no digital input module with a simulator available, use flag F0.3 instead of input I 0,3. Set (or reset) the flag with the PG 685 online function CONTROL VARIABLE. Observe the condition codeword FW U and the parameter assignment error byte FY 15 using CONTROL VARIABLE .

6 Process Status List (PSL)

When a PSL is called the current process values sre printed out on the printer with static text previously programed on the CP 525.

6.1 CP 525 User Program

Switch the mode selector on the CP 525 to "PGR".

Ress F1 (PROGRAM USER DATA), F2 (PC JOB) and F4 (PSL) in the 'SELECTION' mask to call the following mask:

```
-> PROGRAM -> PC JOB ->                                   SIMATIC S5 / COM525
PROCESS STATUS LIST

                         DRIVE:  B   PROGRAM : PT88ABSP   COMPONENT: PT

   Name of list:   PSL-EXAMPLE        Name to be stored: PSL-EXANPLE

   With the job no.:       2
   the output of the process status list with the
   name selected above will be started.

   The dynamic data are on CPU no.: 1
   in data block            DB no.: 30

   List header is frame:
   List trailer is frame:

   Output of messages possible while the PSL is being output   (Y/N) N
   Form feed on PT88/PT89 at start of list                     (Y/N) Y
```

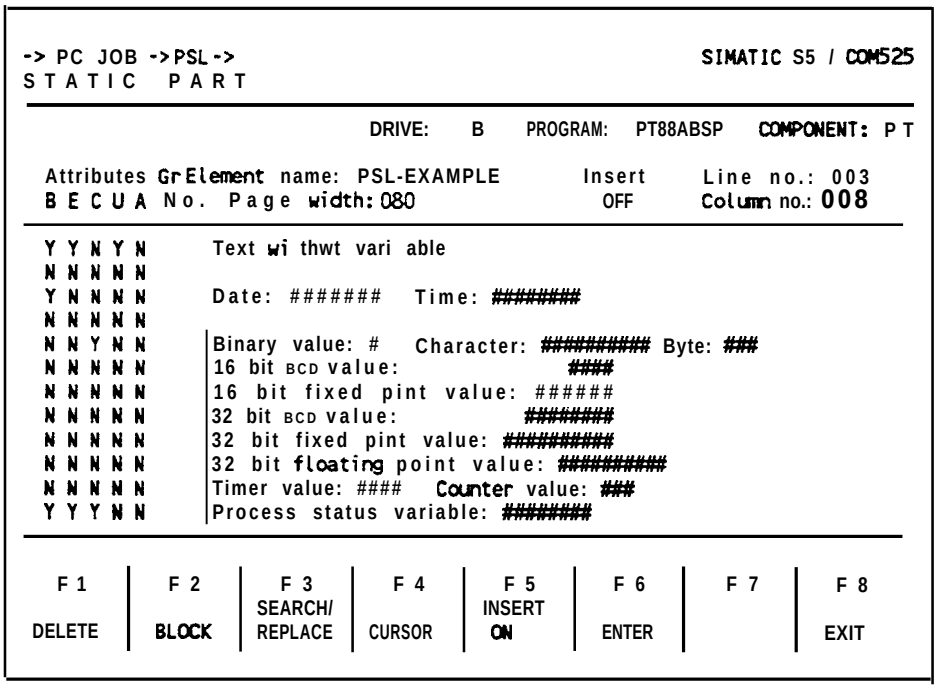
F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
STAT. PART	DYN. PART	FRAME	COMPLETE DYN.PART		SAVE	HELP	EXIT

Select the list name "PSL-EXAMPLE" (this is automatically entered in the input field 'name to be stored") and the job numb& '2".

The CPU number is "1", since only one CPU is plugged into the S5-135U. The data block for the process data is DB '30".

Initially do not enter a list header or trailer. The output of the PSL is not to be interrupted by process messages (entry: "N"). Before the list is output a form feed should be carried out (entry: 'Y").

Press F1 (STAT. PART) to select the mask 'STATIC PART' . In this mask you can enter the following attributes, static texts and reserve characters (#) for dynamic values:



With the function keys F1 to F4 for editing functions the above function key set can be replaced by others.

The following pages explain the most important editing functions. A detailed description of the editor functions can lx-found in the user's guide for the programming package COM 525 in this manual.

When you use the cursor keys with single arrows, the cursor is roved character by character; when you use the double arrow keys it is moved word by word.

The character on which the cursor is currently positioned can be deleted with the delete key (in the cursor key block).

If you called a nev set of function keys using function keys F1 to F4 the cursor cannot be moved nor can entries be made in the editing window. These keys are explained as follows:

F1 (DELETE) calls the delete menu as follows:

- **F2 (WORD)**
  deletes all the characters from the current cursor position to the end of the word
- **F3 (LINE)**
  deletes the line in which the cursor is positioned
- **F4 (BLOCK)**
  deletes the block of lines marked

F2 (BLOCK) calls the block handling menu as follows:

- **F1 (SET START)**
  the start of a block is marked in the mask with S in the line in which the cursor is positioned
- **F2** (SET **END)**
  the end of a block is marked with **E** in the first column of the mask in the line in which the cursor is located
- **F3 (COPY)**
  a block of lines is copied in before the line in which the cursor is located
- **F4 (MOVE)**
  a block of lines is roved in before the line in which the cursor is located

F3 (SEARCH) branches to the search menu as follows:

- **F1 (SEARCH)**
  allows a string of up to 20 characters to be searched for
- **F2 (REPLACE)**
  allows a string of up to 20 characters to be replaced

F4 **(CURSOR)** moves the cursor:

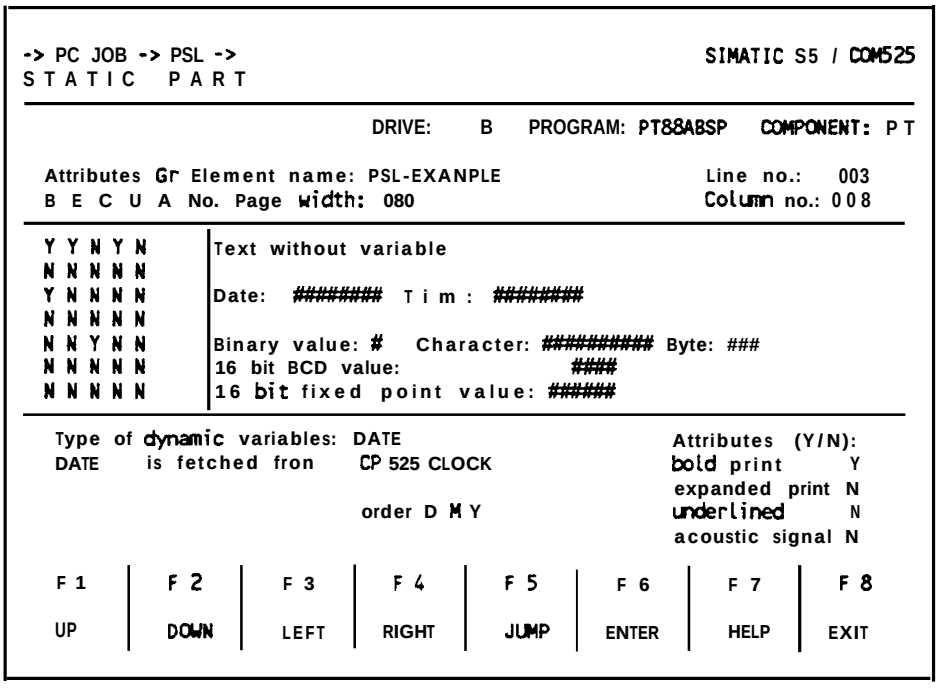
- **F1 (LEFT MARGIN)**
  to the beginning of the line
- **F2 (RIGHT MARGIN)**
  to the end of the line
- **F3 (INPUT FIELDS)**
  into the input f ieMs for the attributes and for the group number. Using the **PG** 685 cursor keys you can now move from column to column and line to line within the attributes. With the function key F4 **(TO ED.WINDOW)** you return to the text editing window.

F5 (INSERT ON/OFF)
switches the INSERT writing mode on or off. Characters entered are either inserted before the current cursor position (ON) or they overwrite text that is already there (OFF).

In the input fields for the attributes, you fix the way **in which** static texts in a list line will be printed, You can select bold **print** (B), expanded print (E), compressed print (C), underlining (U) and acoustic signal (A) by entering **"Y"** (= yes) or **"N"** (= no).

Save the static part of the PSL with F6 (ENTER) in the **PG** 685 **RAM** and exit the mask 'STATIC PAN with F8 **(EXIT)**.

Press F2 (DYN. PART) in the *PROCESS STATUS LIST' mask to call the 'DYN. PART' mask. Using the function keys F1 to F5 you select the parameter you wish to specify. By pressing F5 (JUMP) you will be prompted to specify where you wish to jump ("jump to line ##, field ##"). Press F5 (JUMP) again to jump to the selected field. The specified variable is displayed in the text window inversely.



The following presettings are shown: in the field 'type of dynamic variables" - PROCESS VARIABLE, "format" - 16BIT FXP, 'address" DW and as 'no. of characters" the number of reserve characters (#). All the attributes have the default "N". You can change the type of dynamic variables with F7 (HELP). The liner part of the mask then changes automatically.
These defaults must be extended and possibly changed. The address, format and attributes can be entered using F7 (HELP).

The specification of a variable is stored in the PG 685 RAM with F6 (ENTER). The next variable is then selected with F1 to F5. This is then specified and entered. When all the variables are specified, you exit the 'DYN. PART' with F8 (EXIT). The whole PSL is saved with F6 (SAVE) on the hard disk in the 'PROCESS STATUS LIST' mask.

The date and time should be taken from the CP 525 hardware clock and printed out in bold print. Seconds should not be shown.

```
Type of dynamic variable: DATE                    Attributes (Y/N):
DATE     is fetched from  CP 525 CLOCK            bold print        Y
                                                  expanded print    N
                          order D M Y             underlined        N
                                                  acoustic signal   N
```

Press F6 (ENTER) to store the specification in the PG 685 RAM. Using F4 (WRITE), select the next input field.

```
Type of dynamic variable: TIME                    Attributes (Y/N):
TIME     is fetched from  CP 525 CLOCK            bold print        Y
                                                  expanded print    N
                          order H M               underlined        N
                                                  acoustic signal   N
```

Ress F6 (ENTER) to store the specification in the PG 685 RAM. Using F2 (DOWN), select the next input field,

All other variables in the PSL are of the type PROCESS VARIABLE. It has already been specified in the 'PROCESS STATUS LIST' mask that all the process variables will be fetched from CPU 1 data block DB30. The number of reserve characters for dynamic variables (#) is adopted as the number of characters (before the point) .

Program the type of access and representation as follows:

Type of dynamic variable: PRDCESS VARIABLE		**Attributes (Y/N):**
		bold print N
CPU no. 1	**Format** BINARY	**expanded print N**
DB no. 30	**No. of characters 1**	**tit-lined N**
Address BI 1.4		**acoustic signal N**

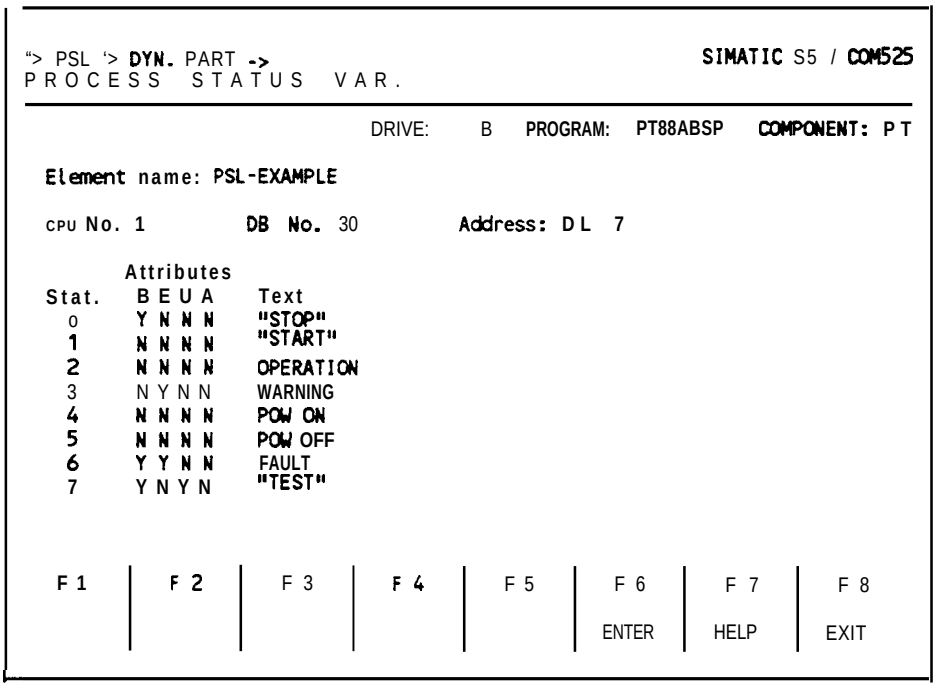
The specification is stored in the PG 685 **RAM with F6** (ENTER). Select the next input field with F4 (WRITE) and continue specifying the following variables accordingly.

The table below shows all the variables of the type PROCESS **VARIABLE**, in the order **in** which they appear in the PSL.

Format	Address	No. of characters	Attributes
BINARY	BI 1.4	1	
CHAR	DL 2.	10	
PROCESS sTATus	DL 7.	8	
BYTE	DR 7.	3	
16BIT BCD	**DW** 8.	4.0	
16BIT FXP	**DW** 9.	6.0	
32BIT BCD	DD 10.	8.0	
32BIT FXP	**DD** 12.	10.0	
32BIT FLP	**DD** 14.	5.4	
TIMER	**DW** 16.	3.1	underlined
COUNTER	**DW** 17,	3	underlined

Select type PROCESS STATUS for the variable in the last line of the list. Ress F6 to branch to the PROCESS STATUS VARIABLE mask. Uptoeight different texts with individual **attributes can** be **programmed in this mask. These texts** cannot exceed the maximum **number** of reserved spaces (**in** this case eight characters).

Enter the texts as shown in the following mask:



Which text is printed out depends on the value of the left data byte DL 7.

Ress F6 to enter the stares variable in the PG memory and press F8 EXIT to return to the DYN. PART mask.

Ress F4 (COMPLETE DYN. PART) in the 'PROCESS STATUS LIST' mask to check whether every variable field marked # in the 'STATIC PART mask is also specified in the 'DYNAMIC PART' mask.

If all the variables have been specified, the message "MESS .017: All fields have already been programed!" appears above the function keys .

If not all variables are specified, the ' COMPLETE DYN. PART' mask appears. Here you can specify the variable fields for which you have not yet programmed the format, address and representation.

The whole process status list is written to the hard disk with F6 (SAVE) .

After saving the list, return to the ' SELECTION' mask by pressing F8 (EXIT) three times; then press F2 to branch to the *TRANSFER' mask. Transfer the PC job PSL-EXAMPLE in the same way that you transferred the NEW PACE PC job (see Section 5.1) to the CP 525 user memory.

Following each transfer, carry out a cold restart on the CP 525 with F1 and switch the CP 525 to "RUN". The red LED for interface 1 (IF 1) must go out.

"6.2 STEP 5 User Program

The STEP 5 program must achieve the following:

- the job for the PSL mat be triggered
- the process data must be transferred from the CPU to the CP.

The triggering of the process status list is achieved in the same way as the triggering of the PC job for form feed (using a SEND DIRECT job). The only cliff erence is that a different job number and different addresses for the condition codeword (ANZW) and parameter assignment error word (PAFE) must be used. The job number A-NR "2" is identical with the number of the PC job progr-d on the CP. Use the digital input I 0.4 and start the PSL on a positive going signal edge.

The SEND DIRECT job is triggered with the result of logic operation (RLO) 1. If the RLO is 0 when the call is made, only the condition codeword is updated.

With the SEND DIRECT call, the condition codeword requires two words. The job status and any error messages are indicated in flag word FW 16. The number of pieces of data transferred is indicated in flag word FW 18.

If the job is to be terminated with the status " job terminated with error" or if a parameter assignment error occurs, this is indicated at digital output Q 0.2. Normally, an error routine tailored to the particular application is called. If an error occurs, evaluate the error numbers in the condition codeword, in the error message area of the SYSTAT (see Section 11) and in the parameter assignment error byte.

The transfer of the process data is executed with a SEND ALL handling block call. The SEND ALL (also known as SEND 0) is a SEND call with the job number A-NR '0'. It is called unconditionally in every program cycle and checks whether the CP (dual-port RAM) requires data. If it does, the SEND ALL fetches it from the source specified by the CP and sends it to the CP. If it does not, the function block is exited and the cyclic program continued.

The SEND AIL is called unconditionally and is executed when it is called regardless of the result of logic operation. Its condition codeword is only one word long. The number of the job is entered for which the SEND ALL trensfers data from the CPU to the CP.

OB 1 must be expsnded by the following statements:

```
     :AN  10.4          EDGE EVALUATION:
     :R   F 2.4         THE RESULT OF LOGIC OPERATION
     :A   I 0.4         (RLO) IS SET FOR ONE CYCLE IF THE
     :AN  F 2.4         SIGNAL AT INPUT I 0.4 CHANCES
     :S   F 2.4         FROM 0 To 1.

     :JU  FB120         CALL PROCESS STATUS LIST
NAME :SEND
SSNR :    IW0, 0       INTERFACE NUMBER 0
A-NR :    KY0 ,2       JOB NUMBER 2
ANZW :    FW16         CONDITION CODEWORD
QTYP :    KSNN         NO SOURCE SPECIFIED
DBNR :    KY0, 0       NOT EVALUATED
QANF :    KF+0         NOT EVALUATED
QLAE :    KF+0         NOT EVALUATED
PAFE :    FY20         PAMMETER ASS. ERROR BYTE

     :JU  FB126         TRANSFER PROCESS DATA TO CP
NAME :SEND-A
SSNR :    KY0, 0       INTERFACE NUMBER 0
A-NR :    KY0 ,0       JOB NUMBER 0
ANZW :    FW21         CONDITION CODEWORD
PAFE :    FY23         PARAMETER ASS. ERROR BYTE

     :O   F 17.3        JOB 2 TERMINATED WITH ERROR
     :O   F 20.0        OR PARMETER ASSIGNMENT ERROR
     := Q 0.2           INDICATED AT OUTPUT Q 0.2
```

The process data is located in data block DB 30. This must be programed with a length of at least 32 data words: it is then long enough for all our example programs. Assign data words DW 1 to DW 17 as follows:

```
 0 :   KH= 0000;
 1 :   KM= 0000000000010000;   bit 4 = 1
 2 :   KS= EXAMP. TEXT         character string
 7 :   KY= 005,123;            process variable 5, byte variable 1.23
 8 :   KH= F123;               -123 in 16 bit BCD format
 9 :   la?= +12345;            +12345 in 16 bit fixed pt. format
10 :   KH= F123;               -1234567 in 32 bit BCD format
11 :   KH= 4567;
12 :   KF= +00001;             +77881 = 12345 + 65536
13 :   KF= +12345;             in 32 bit fixed point format
14 :   KG=+1234567+02;         1.2.34567 floating point format
16 :   KT= 123.3;              timer value 1230 seconds
17 :   KC= 123;                counter value 1.23
```

Transfer DB 30 and the expanded OB 1 to the user memory of your R processor.

6.3 Test

Carry out a cold restart on the R processor to synchronize the CPU and CP 525. When you switch the signal at digital input I 0.4 from zero to one, the PSL is output on the printer:

```
Text without variable

Date:  01  .01.88 Time  00:00 am

Binary value: 1  Character: EXAMP.TEXT  Byte: 123
16 bit BCD value:                  -123
16 bit fixet point value:         12345
32 bit BCD value:             -1234567
32 fixed point value:             77881
32 bit floating point value:        12.3457

Timer value: 123.3  Counter value: 123
Process status variable:  POW OFF
```

If you have no digital input module with a simulator available, use flag F 0.4 instead of input I 0.4. Set (or reset) the flag with the PG 685 online function CONTROL VARIABLE. Observe condition codeword FW 16 and the parameter assignment error byte FY 20 with CONTROL VARIABLE.

Change the values in data block DB 30 and start the process status list again.

7 Sequential Message List (SML)

The SML is a series of single messages that are printed out immediately after an event has been detected.

First program the messages on the CP 525 (Section 7.1). You will thenbeshown the various ways of programming your STEP5 program.

1. Either use the standard function blocks "signaling functions for standard CPs" as follows:

   - without the S5-DOS programming package COM PMC (Section 7.2),
   - with the S5-DOS programming package COM PMC (Section 7.3).

2. The other option is not to use the 'signaling functions for standard CPs" (Section 7.5).

7.1 CP 525 User Program

Switch the mode selector of the CP 525 to "PGR".

Before you can program process messages you must program the PC job 'SML'. Among other things the positions of the date, time and status in the message text are fixed for all the messages.

7.1.1 PC Job SEQ.MESSAGE

Press F1 (PROGRAM USER DATA) in the 'SELECTION' mask to call the 'PROGRAM' mask. Press F2 (PC JOB) in this mask to obtain the 'PC JOB' mask. In this mask, press F1 (SML) to call the 'SEQUENTIAL MESSAGE USC'. Some standard default values are already entered.

```
-> PROGRAM -> PC JOB ->                                          SIMATIC S5 / COM525
S E Q U E N T I A L   M E S S A G E   L I S T
DRIVE:  B   PROGRAM: PT88ABSP   COMPONENT: PT

 List name:        SEQ. MESSAGE
 With the job number: 3
 messages are output as events take place.
 List header is frame:
 List trailer is frame:
 Form feed on PT88/PT89 as start of list (Y/N) N
 Position and format of date, time, status is as below for all messages:
                      Attributes Order
                Column   B E U A
   Date:           1     N N N N  DD.MM
   Time:           7     N N N N  HH:MM:SS
   ------------------------------------------------------------
   Status:        65              Text (enclose in inverted comas)
     caning:             Y N N N  '1'
     going:              Y N N N  '0'
     acknowledged:       N N N N  ' '

 F 1 | F 2 | F 3   | F 4 | F 5 | F 6  | F 7  | F 8
     |     | FRAME |     |     | SAVE | HELP | EXIT
```

First of all, a job number must be assigned with which the CPU starts the SML. All PC jobs must have different job numbers. For this example select '3".

In this case do not enter anying the in the list header and trailer.

A form feed before the SML is printed out is not needed so "N" is entered.

The date is to appear in column "1" without the year being specified; the time should appear in column "7" and the status in column "65".

Messages which are "coming" (positive going edgeat digital input) should have the status '1"; messages which are "going" (falling edge) should have the status '0". The status is tobe printed in bold print: a "Y" must be entered in the first attribute column.

Messages cannot be acknowledged using the CP 525 alone. For this reason we delete the text entered in this line.

The following format is therefore determined for all process messages:

```
Printer column 1  7        15                65       80
               !  !        !                 !        !
               DDDDDxTTTTTTTTxxxxx . . . . . xxxxxSxxxxxxxxx

               D - date      T - time        s - status
               x - still to program
```

The number of columns reserved for the status is determined by thelongest ofthethree status texts; in this case one character.

The page width of 80 characters and the separators for the date (.) and time (:) were determined when the parameters were assigned to the interpreter (see Section 4.3.2).

In the input fields for attributes you can specify how the date, time and status are to be represented. You can select bold print (B), expanded print (E), underlining (U) and acoustic signal (A) by entering “Y” (yes) or "N" (no). The message status should be printed in bold print.

The SML PC job is written to the hard disk with F6.

Ress F8 (EXIT) twice to return to the 'PROGRAM' mask.

7.1.2 Messages

Press F1 (MESSAGE) in the 'PROGRAM' mask to call the 'MESSAGES' mask. Enter the number of the first message; select '100' as the number for the first message.

Press F1 (STAT. PART) to call the 'STATIC PART' mask.
Enter 11 messages.

```
-> PROGRAM -> MESSAGES ->                                        SIMATIC S5 / COM525
S T A T I C   P A R T
------------------------------------------------------------------------------------
                        DRIVE:  B   PROGRAM:  PT88ABSP   COMPONENT: PT

 Mess attributes  Gr.                                Insert      Line no.: 0001
 No.  B E C U A   No.  page width: 080               OFF         Column no.: 006
------------------------------------------------------------------------------------
 0100 Y N N Y Y  | 00000 TTTTTTTT MESSAGE 100 static text and ac. sign. char.
 0101 N N N N N  | DDDDD TTTTTTTT MESSAGE 101 binary value:              #
 0102 N N N N N  | DDDDD TTTTTTTT MESSAGE 102 character:                 #
 0103 N N N N N  | DDDDD TTTTTTTT MESSAGE 103 byte variable:             ###
 0104 Y N N N N  | DDDDD TTTTTTTT MESSAGE 104 16-bit BCD number:         ####
 0105 Y N N N N  | DDDDD TTTTTTTT MESSAGE 105 16-bit fixed pt. no.:      ######
 0106 Y N N N N  | DDDDD TTTTTTTT MESSAGE 106 32-bit BCD number:         ########
 0107 N N Y N N  | DDDDD TTTTTTTT MESSAGE 107 32-bit fixed pt. no.:      ###########
 0108 N N Y N N  | 00000 TTTTTTTT MESSAGE 108 floating point number:     ###########
 0109 N N Y N N  | DDDDD TTTTTTTT MESSAGE 109 timer value:               #####
 0110 N N N Y N  | DDDDD TTTTTTTT MESSAGE 110 counter value:             ###
                 |
------------------------------------------------------------------------------------
```

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
DELETE	BLOCK	SEARCH/ REPLACE	CURSOR	INSERT ON	ENTER		EXIT

The text editing window can be up to 136 columns wide. Of these, only 59 columns can be displayed on the PG screen. As soon as the cursor is moved beyond the margin of the editing window, the visible area is moved by 20 columns.

The character # is the reserve character for message parameters. Each message can have one parameter. Messages with a parameter must be numbered consecutively when the standard function blocks 'signaling functions for standard CPs' are used (see also Section 7.2.4).

In the input fields for the attributes you specify the way in which the static parts of a message will be represented. You can select bold print (B), expanded print (E), compressed print (C), underlining (U) and acoustic signal (A) by entering "Y" (yes) or "N" (no). Various combinations of attributes are programed. Do notenteranygroupnumbers.

By using the function keys F1 to F4 for editing functions the set of function keys previously described is replaced by a different set.

The operator environment of the message editor is the same as that for the PSL editor. The most important functions are explained below. A more detailed description of the message editor can be found in the user's guide for the COM 525 programming: package in Sections 5.3.2 and 5.3.3.

Using the cursor keys with single arrows moves the cursor character by character; the double arrow keys moves the cursor word by word.

The character on which the cursor is currently positioned can be deleted with the delete key (in the cursor key block).

If you call anew set of function keys using function keys F1 to F4, the cursor cannotbemovednorcanentriesbemadeinthe editing window. The following keys are defined:

F1 (DELETE) calls the delete menu:

- **F2 (WORD)**
  deletes all the characters from the current cursor position to the end of the word
- **F3 (LINE)**
  deletes the line in which the cursor is positioned
- **F4 (BLOCK)**
  deletes the block of lines marked

F2 (BLOCK) calls the **block** kindling **menu**:

- **F1 (SET START)**
  the start of a block is **marked** in the mask with S **in** the line in which the cursor is positioned
- **F2** (SET **END)**
  the end of a block **is** marked with **E in** the first column of the mask in the line in which the cursor is located
- **F3 (COPY)**
  a block of lines are copied in before the line in which the cursor is located.
- **F4 (MOVE)**
  a block of lines are moved **in** before the line **in** which the **cursor** is located.

F3 (SEARCH) branches to the search menu:

- **F1 (SEARCH)**
  allows a string of up to 20 characters to be searched for
- **F2 (REPLACE)**
  allows a string of up to 20 characters to be replaced

F4 (CURSOR) roves the cursor:

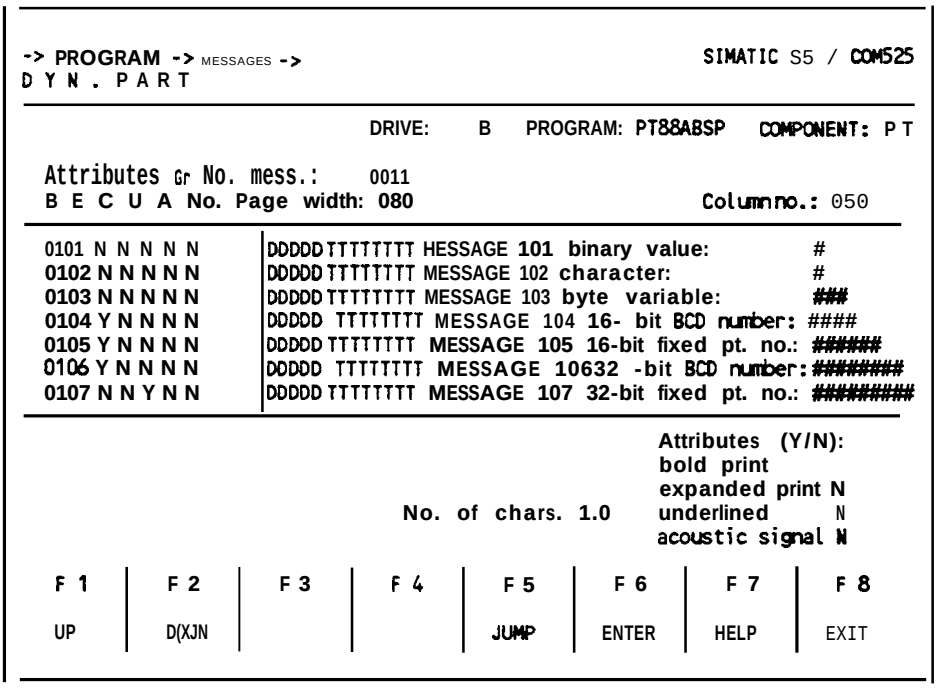
- **F1 (LEFT MARGIN)**
  to the start of the message line
- **F2 (RIGHT** MARGIN)
  to the end of the message line
- **F3 (INPUT FIELDS)**
  into the input fields for the message **number,** the attributes and the message **group** number. Using the **PC** 685 cursor keys you **can** now **move** from **column** to **column** and line to line within the attributes. With the function key **F4** ('10 **ED.WINDOW)** you return to the text editing window.

F5 (INSERT ON/OFF)

switches the INSERT writing **mode** on or off. Characters entered are either inserted before the current cursor position (ON) or they overwrite text that is already there (OFF).

F6 (ENTER) saves all **the programme**d **messages** on the hard disk.

F8 (EXIT) returns you to the 'MESSAGES ' mask. Press F2 (DYN. PART) in this mask to call the 'DYN. PART' mask.



In this mask only the type of representation (attributes and places before and after the point) of the parameters is determined. The number of characters resenred for message parameters (#) is adopted as the default for the number of places (before the point). The parameter type is determined only when you program the STEP 5 program (data block DB M-PARAM, see section 7.2.4).

With F1, F2 and F5 you select the parameter to be programed; it is displayed on the screen inversely. Then the attributes and number of characters are programmed. These are written into the PG RAM individually sing F6. The next parameter is then selected with F1, F2 or F5, programmed and saved. Continue in this way until all the parameters have been programmed. When you exit the mask with F8 (EXIT) all the dynamic parts are written to the hard disk.

Specify the following attributes (input "Y") and number of characters for the parameters:

Binary value:	BOLD	no. of char.: 1.
Character:	EXPANDED	no. of char.: 1.
Byte variable:	UNDERLINED	no. of char.: 3.
16-bit BCD number:	BOLD	no. of char.: 4.
16-bit fixed pt. no.:	EXPANDED	no, of char.: 6.
32-bit BCD number:	UNDERLINED	no. of char.: 8.
32-bit fixed pt. no.:	BOLD	no. of char.: 10.
Floating point no.:	EXPANDED	no. of char.: 5.4
Timer Value:	UNDERLINED	no. of Char.: 3.1
Counter value:		no. of char.: 3.

Press F4 (COMPLETE DYN. PAM') in the 'MESSACES' mask to check whether all the variable fields in the 'STATIC PART' marked with # have been specified in the 'DYN. PART'.

If all the variables have been specified, the message 'MESS. 016: All fields from message 0100 to 0110 already programmed!' appears above the function keys.

If variables are still unspecified, the 'COMPLETE DYN. PART' mask appears. Here, you can specify the variable fields for which no type of representation has yet been programmed.

Transfer the process messages and the PC job SEQ.MESSAGE (sequential message list) to the user memory of your CP 525.

For the process messages, press the function keys F6 (USER DATA) and F2 (MESSAGE) in the 'TRANSFER' mask. Transfer all the process messages with F5 (TOTAL) in the 'MESSAGE' mask. The PC job SEQ.MESSAGE is transferred in the same way as the NEW PAGE PC job (see Section 5.1).

After the transfer, carry out a cold restart on the CP 525 by pressing F1 (COLD RESTART) in the 'TRANSFER' mask. Then the red LED on interface 1 (Xl? 1) must go out.

7.2 STEP 5 User Program without COM PMC

The listing of messages can be carried out in the CPU by standard function blocks. In addition to the handling blocks, the signalling functions for standard CPs are then also required (see 'Ordering data in this manual) .

You Can output messages on a printer without the 'signaling functions for standard CPs'. To do this, you must use the interface between the CPU and CP 525 as described in the user guide 'Event output and listing with the PT88/PT89 printer'. (See also Section 7.5 in this example).

In this example of an application and in Section 7.3 the "signalling functions for standard CPs" are used. This means that the program runs completely differently from the STEP 5 programs you have seen up to now. You must therefore open a new STEP 5 program file, "PT88R2ST. S5D". Transfer the handling blocks, the function blocks, program and organization blocks (not the data blocks!) for "signaling functions for standard CPs" for the R processor to this program file.

7.2.1 Signalling Functions for Standard CPs

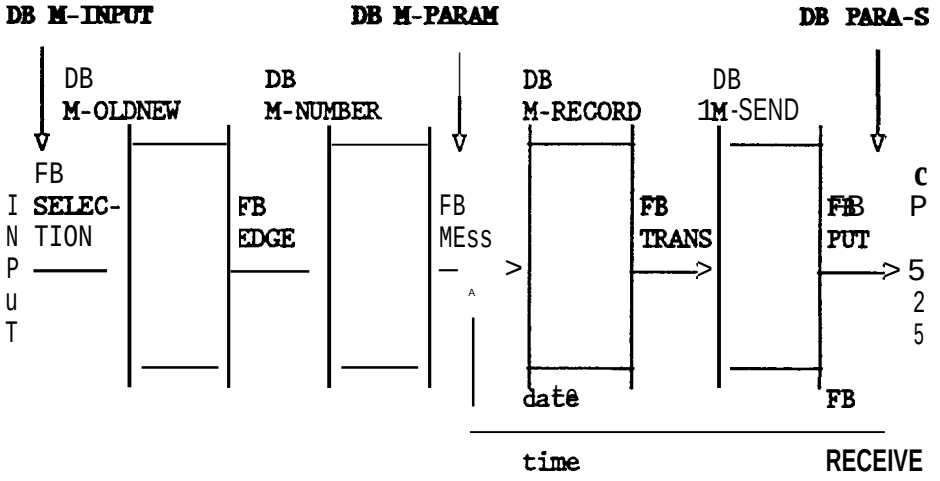
These function blocks, which must be ordered separately, detect changes in the message statuses and generate 6 or 8 data word long message records from these changes. The message records contain the following:

message number
- message status
- date and time of their occurrence
- parameter type
- parameter (if present)

They are transferred to the CP 525 in the data block DB M-SEND.

The structure of a message record and of DB M-SEND can be found in Section 2.2 of the user's guide "Event output and listing with the PT88/89 printer" and in Section 7.5 of this example.

From the event, i.e. the change of status in a data, input, output or flag word, the data block DB M-SEND is generated in a chain of function and data blocks as follows:



The function and data blocks are supplied as part of a simple example program, which must be adapted to your own particular situation. Block numbers and variables are for the most part retained in this example.

The programing or parameter assignment of these blocks can be carried out easily with the programming package COM PMC, which must be ordered separately in addition to the standard function blocks (see Section 7.3). PMC stands for "Process monitoring and control system" .

The following sections describe the STEP 5 programming and parameter assignment of the blocks without COM PMC. In this way you gain a better insight into the way in which the blocks function. The terms used come mainly from the description of the "signalling functions for standard CPs".

The function blocks must be called in succession in a cyclic program. Program block PB 3 is used. The PB 3 supplied is reduced to the calls that are absolutely necessary.

```
      :C   DB6          SELECT DB PMC
      :
      :JU  FB151
NAME  :SELECTION
      :
      :JU  FB152
NAME  :EDGE
      :
      :JU  FB154
NAME  :MESS
      :
      :JU  FB155
NAME  :TRANS
      :
      :A   D5.8         MESSAGE ENABLE FOR CP 525 (COMMUNICATION
      :JC  FB141                          & MONITORING DEVICE 1)
NAME  :PUT
BLNR  :    KF+5         TRANSFER DB M-SEND TO CP 525
      :
      :JU  FB143
NAME  :PUT-A            SEND ALL FUNCTION
      :
      :BE
```

The function blocks have parameters assigned indirectly in data block DB PMC (DB 6). DB 6 must be open before the function blocks are called.

The CP 525 is the only operator communication and monitoring (C & M) device. The transfer of the message records to the CP is achieved by calling the PUT function block; the handling block parameters for the transfer must be located in parameter field 5 of the data block DB PARA-S (see also section 7.2.5). The enabling of a message is controlled automatically by the function blocks using data bit 5.8 in data block DB PMC DB 6 (see also Section 7.2.2).

PB 3 is called unconditionally in every program cycle in OB 1.

With the S5-115U, the cycle time must be triggered extra in PB 3.

Select the following data block numbers and set up the data blocks with the following lengths:

DB M-OLDNEW: DB 201 255 data words
DB M-ACKN1: DB 202 79 data words only with S5-150U
DB M-NUMBER: DB 204 256 data words
DB M-RECORD: DB 205 256 data words
DB M-SEND: DB206 256 data words

Assign the following data words in the data block DB M-OLDNEW DB 201:

253 : KF= +00100 lower limit of message numbers
254 : KF= +00001 number of words occupied by message bits in DB M-OLDNEW

The CP 525 requires this information for outputting the current message list (see also Section 8.2).

7.2.2 Assigning Parameters to the Signaling Function Blocks DB PMC

The assignment of parameters to the signaling function blocks is carried out indirectly in a data block, DB PMC. Select DB 6. The listing with the CP 525 is the only communication and monitoring function. Since you have not connected an operator-process communication device (e.g. CP 526), it is not possible to acknowledge (nevertheless, a DB M-ACKN1 - DB 202 will be required in the S5-150U).

Set up DB 6 in its full length (256 data words).

The following entries must be made in the 256 word long data block DB PMC (DB 6):

```
 0 :   KM= 0000000000000100;   Signaling package present
 1 :   KM= 0000000000000100;   The interface c (printer) is
                               present. This must have params.
                               assigned from DW 136 onwards.

 4:    KH= 0001;               The printer is C + M device 1.
                               (No further C + M devices).
 5:    KM= 0000000000000001;   Message enable C + M device 1.
 6:    KM= 0000000000000001;   Device 1 has the master clock.

20 :   KS= DB                  The condition codewords of the
                               handling blocks are written into
                               data blocks.

33 :   KF= +00208;             DB M-INPUT is DB 208.

38 :   KF= +00201;             DB M-OLDNEW is DB 201.
39 :   KF= +00000;             no DB M-ACKN1
                               (DB 202 with S5-150U)

41 :   KF= +00204;             DB M-NUMBER is DB 204.
42 :   KF= +00205;             DB M-RECORD is DB 205.
43 :   KF= +00206;             DB M-SEND is DB 206.
44 :   KF= +00099;             DB M-PARAM is DB 99.
45 :   KF= +00014;             DB PARA-R is DB 14.
46 :   RF= +00015;             DB PARA-S is DB 15.
47 :   KF= +00100;             Lower limit of message numbers
48 :   KF= +00110;             Upper limit of message numbers
49 :   KF= +00101;             Mess. no. with lower limit param.
50 :   KF= +00110;             Mess no. with upper limit param.

90 :   KF= +00006;             DB PMC is data block DB 6.
91 :   KF= +00007;             PB-HDB is program block PB 7.

136 :  KF= +00000;             Interface number of the CP 525
                               Printer interface (IF 1) =0
137 :  K)?= +00006;            Block size = 512 bytes
```

Another data words can be preassignedwithO.

7.2.3 Determining the Origin of the Message, DB M-INPUT

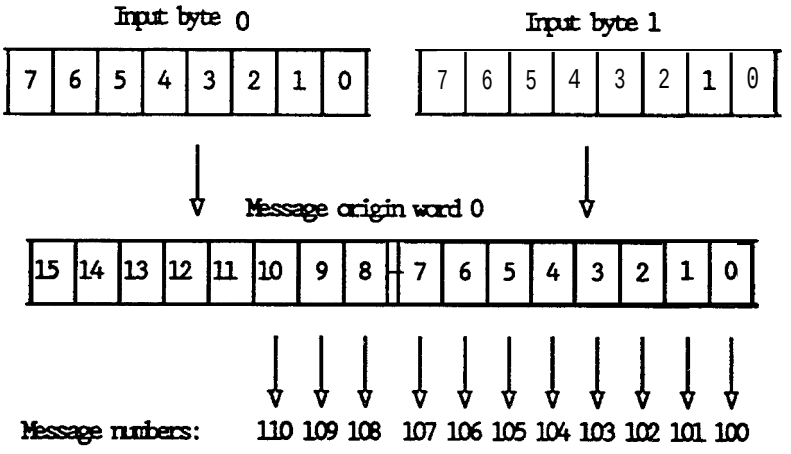
The origin of the message is entered in DB M-INPUT according to type, area start and area length. Messages are tobe triggeredby input word IW 0. Select data block DB 208 as follows:

```
0 :   KY= 001,000;     input word type
1 :   KY= 000,001;     from input word 0, one word
2 :   KH= FFFF,FFFF;   end identifier (two data words)
```

With the DB M-INPUT the **origin of a message is automatically assigned to a message number**. If the cause of a message is located in a data block, bit 0 of the first word in area 1 controls the message with the number located in DB PMC (DB 6) DW 47 (upper limit of message numbers). The following bits control the messages with the following message numbers. The message numbers of the maximum 10 input areas follow on without gaps.

In contrast to data words, the flags, inputs and outputs are organised in bytes. When imaging the bytes on a message origin word, the byte with the number programmed in DB M-INPUT is imaged in the higher order part of the word. The byte with the next number is transferred to the lower order part of the word.

In the example the following assignment is obtained:



7.2.4 Message Parameters DB M-PARAM

The message parameters are programmed in data block DB M-PARAM. You specified the message range by means of parameters in DB PMC (DB 6) in DW 49 and DW 50. Beginning with the lowest message number in DW 49 (in the example, message 101), the parameters for the successive messages are fixed. Select data block DB 99 as the DB M-PARAM data block.

```
 0:    KH= 0004;          message 101: bit variable; bit 4 of
 1 :   KY= 030,001;        data block DB 30, data word DW 1
 2:    KH= 0108;          message 102: char. variable; left byte
 3:    KY= 030, 002;       data block DB 30, data word DW 2
 4:    KH= 0200;          message 103: byte variable; right byte
 5:    KY= 030,007;        data block DB 30, data word DW 7
 6:    KH= 0300;          message 104: 16-bit BCD value
 7 :   KY= 030,008;        data block DB 30, data word DW 8
 8 :   KH= 0500;          message 105: 16-bit fixed point number
 9:    KY= 030,009;        data block DB 30, data word DW 9
10 :   KH= 0400;          message 106: 32-bit BCD value
11 :   KY= 030,010;        data block DB 30, from DW 10
12:    KH= 0600;          message 107: 32-bit fixed point number
13 :   KY= 030,012;        data block DB 30, from DW 12
14 :   KH= 0700;          message 108: floating point number
15 :   KY= 030,014;        data block DB 30, from DW 14
16 :   KH= 0800;          message 109: timer value
17 :   KY= 030,016;        data block DB 30, data word DW 16
18 :   KH= 0900;          message 110: counter value
19 :   KY= 030>017;        data block DB 30, data word DW 17
```

If there is not sufficient room in the data block to write the parameters (more than 128 messages with parameters) you can continue to write the parameters into the data block with the next higher data block number. If there are more than 256 messages with parameters takethenext higher data block and so on.

7.2.5 Assigning Parameters in DB MBA-R and DB MBA-S for Triggering Jobs

PC jobs are triggered by the ' signaling functions for standard CPs' on the CP 525 with PUT or GET function block calls. The function blocks trigger the job and the data transfer with the help of handling blocks.

The function blocks PUT and GET have parameters assigned to them indirectly. The function block parameter is a field number. You must program the handling block parameters for the PUT call (GET call) in the parameter field of the data block DB PARA-S (DB PARA-R) with the seine number.

A parameter field is 10 data words long. The first data word in the parameter field N has the number (N - 1) * 10. The handling block parameters are programmed in the parameter field; the condition codeword and parameter assignment error byte of the corresponding handling block call are also indicated here (see also program examples).

The FB-PUT function block transfers data directly to the CP 525, or triggers a job which is completed with a PUT-A function. The FB-PUT monitors the job and repeats it up to three times if an error occurs. FB-PUT uses the SEND handling block for the data transfer.

The function block FB-PUT is called unconditionally, PUT jobs are triggered by the setting of bit 0 in the first data word of the corresponding parameter field of DB PAM-S for one cycle.

If no data is to be transferred to the CP with the PUT call, then bit 2 in the first data word of the parameter field in DB PARA-S must also be set before each PUT call. Instead of a SEND DIRECT with the source type XX, a SEND DIRECT with source type NN is called.

'Job running is indicated in bit 3 of the f irst data word, 'job complete without errors in bit 4, 'job aborted' in bit 5 and the number of repetitions (0 . . . 2) in data byte DL 0.

The function block FB-GET tranfers data directly from the CP 525 to the CPU. FB-GET uses the handling block RECEIVE. GET calls are required only for special jobs such as reading the date/time, SYSTAT or SYSID.

The actual handling block calls must take place in a program block, PB HDB. The program block number must be entered in DB PMC, data word DW 91 (see also Sections 7.2.2 and 7.2.6).

Parameter field 1 in DB PARA-R and parameter fields 2 and 5 in DB PARA-S are necessary for listing messages.

You can also program the triggering of other PC jobs such as form feed and the output of process status lists with PUT or GET calls. These jobs can be carried out immediately following a handling block call; they are not automatically repeated if an error occurs. To achieve a uniform structure for triggering PC jobs, all PC jobs are triggered by PUT and GET function blocks.

The parameter fields up to number 5 in DB PARA-R and up to and including number 13 in DB PARA-S are reserved for further PMC functions. The fields immediately following these fields are available for triggering additional PC jobs.

Set up data block DB 14 as DB PARA-R with a length of 10 data words and data block DB 15 as DB PARA-S with a length of 150 data words.

Data block DB PARA-R DB 14, field 1 (read time from the CP 525 hardware clock into DB PMC; see also Section 10,2)

```
 0:    KM= 0000000000000000;  job status
 1 :   KF= +00000;            interface number 0
 2:    KF= +00218;            job number 218
 3:    KS= DB                 destination type data type
 4:    KF= +00006;            destination data block DB PMC DB 6
 5:    KF= +00051;            destination start data word DW 51
 6:    KF= +00005;            destination length 5 data words
 7:    KM= 0000000000000000;  condition codeword (reply)
 8 :   KM= 0000000000000000;  condition codeword (reply)
 9:    KH= 0000;              parameter assignment error (reply)
```

Data block DB PARA-S DB15 field 2, (set master identifier in CP 525 in FB-START FB 130; see Sections 10.1 and 10.3)

```
10 :   KM= 0000000000000000;  job status
11 :   KF= +00000;            interface number 0
12:    KF= +00218;            job number 218
13 :   KS= DB                 source type data block
14 :   KF= +00006;            source data block DB PMC DB 6
15 :   KF= +00051;            source start data word DW51
16 :   KF= +00001;            source length 1 data word
17 :   KM= 0000000000000000;  condition codeword (reply)
18 :   KM= 0000000000000000;  condition codeword (reply)
19 :   KH= 0000;              parameter assignment error (reply)
```

Data block DB MBA-S, field 5 (trigger sequential message list)

```
40 :   KM= 0000000000000000;  job status
41 :   KF= +00000;            interface number 0
42 :   m?= +00003;            job number 3
43 :   KS= DB                 source type data block
44 :   KF= +00206;            source DB DB M-SEND DB 206
45 :   KF= +00000;            source start data word DW 0
46 :   KF= +00000;            source length is entered
                              dynamically by FB-TRANS, FB 155
47 :   KM= 0000000000000000;  condition codeword (reply)
48:    KM= 0000000000000000;  condition codeword (reply)
49 :   KH= 0000;              parameter assignment error (reply)
```

The PC jobs for a new page and for the PSL which have up to now been triggered by a SEND DIRECT call are now also triggered by PUT calls. The corresponding parameter fields must be programed as follows. Select the first completely free usable fields 14 and 15 in DB PARA-S:

Parameter field 14 for the PC job NEW PACE

```
130 :    KM= 0000000000000000;   job stares
131 :    KF= +00000;             interface number 0
132 :    KF= +00001;             job number 1
133 :    KS= NN                  source type NN (no data)
134 :    KF= +00000;             source DB not relevant
135 :    KF= +00000;             source start not relevant
136 :    KF= +00000;             source length not relevant
137 :    KM= 0000000000000000;   condition codeword (reply)
138 :    KM= 0000000000000000;   condition codeword (reply)
139 :    KH= 0000;               para. assignment error (reply)
```

Parameter field 15 for the PC job PSL-EXAMPLE

```
140 :    KM= 0000000000000000;   job status
141 :    KF= +00000;             interface number 0
142 :    KF= +00002;             job number 2
143 :    KS= NN                  source type NN (no data)
144 :    KF= +00000;             source DB not relevant
145 :    KF= +00000;             source start not relevant
146 :    m?= +00000;             source length not relevant
147 :    KM= 0000000000000000;   condition codeword (reply)
148 :    KM= 0000000000000000;   condition codeword (reply)
149 :    KH= 0000;               para. assignment error (reply)
```

7.2.6 Handling Block Calls PB-HDB

Handling blocks are called by the function blocks START-UP, PUT, GET, PUT-A and GET-A. The handling block calls mat take place in a program block whose block number is entered in data block DB PMC DB 6 data word DW 91. Select program block PB 7.

The program block PB-HDB PB 7 is supplied with the 'signaling functions for standard CPs".

```
     :C   DB6                  SELECT DB PMC
     :
     :A   D 92.8               SEND ALL ENABLED ?
     :JC  FB126
NAME :SEND-A
SSNR :    KY255 ,18            INTERFACE NO. PROM DB 6 DW 18
A-NR :    KY0,0                JOB NUMBER FROM DB 6 DW 19
ANZW :    FW0                  C. CODEWORD ADDR. DB 6 DW 20/21
PAFE :    FY199                PARAMETER ASSIGNMENT ERROR BYTE
     .
     :A   D92.9                RECEIVE ALL ENABLED ?
     :JC  FB127
NAME :REC-A
SSNR :    KY255 ,18            INTERFACE NO. FROM DB 6 DW 18
A-NR :    KY0,0                JOB NUMBER FROM DB 6 DW 19
ANZW :    FW0                  C. CODEWORD ADDR. DB 6 DW 20/21
PAFE :    FY199                PARAMETER ASSIGNMENT ERROR BYTE

     :A   D 92.10              CONTROL ENABLED?
     :JC  FB123
NAME :CONTROL
SSNR :    KY255 ,18            INTERFACE NO. PROM DB 6 DW 18
A-NR :    KY0,0                JOB NUMBER FROM DB 6 DW 19
ANZW :    FW0                  C. CODEWORD ADDR. DB 6 DW 20/21
P m :     FY199                PARAMETER ASSIGNMENT ERROR BYTE
     :
     :A   D 92.11              SEND XX ENABLED?
     :JC  FB120
NAME :SEND
SSNR :    KY255 ,18            INTERFACE NO. PROM DB 6 DW 18
A-NR :    KY0,0                JOB NUMBER FROM DB 6 DW 19
ANZW :    FW0                  C. CODEWORD ADDR. DB 6 DW 20/21
QTYP :    KSXX                 INDIRECT PARAMETER ASSIGNMENT!
DBNR :    KY0, 6               SOURCE PARAMETER IN DB PMC DB 6
QANF :    KF+22                FROM DATA WORD DW 22
QLAE :    KF+0
PAFE :    FY199                PARAMETER ASSIGNMENT ERROR BYTE
     :
```

Continuation PB-HTB:

```
     :A   D 92.12          SEND NN ENABLED?
     :JC  FB120
NAME :SEND
SSNR :    KY255,18         INTERFACE NO. FROM DB 6 DW 18
A-NR :    KY0, o           JOB NUMBER FROM DB 6 DW 19
ANZW  :   FW0              C . CODEWORD ADDR. DB 6 DW 20/21
m     :   KSNN
DBNR :    KY0, o
QANF  :   KF+0
QLAE  :   KF+0
PAFE :    FY199            PARAMETER ASSIGNMENT ERROR BYTE

     :A   D 92.13          RECEIVE ENABLED ?
     :JC  FB121
NAME :RECEIVE
SSNR :    KY255,18         INTERFACE NO. FROM DB 6 DW 18
A-NR :    KY0, o           JOB NUMBER FROM DB 6 DW 19
ANZW  :   FW0              C . CODEWORD ADDR. DB 6 DW 20/21
ZTYP :    KSXX             INDIRECT PARMETER ASSIGNMENT
ZBNR :    KY0,6            DEST. PARAMETER IN DB PMC DB 6
ZANF :    KF+22            FROM DATA WORD DW 22
ZLAE  :   KF+0
PAFE :    FY199            PARMETER ASSIGNMENT ERROR BYTE

     :A   D 92.14          SYNCHRON ENABLED?
     :JC  FB125
NAME :SYNCHRON
SSNR :    KY255,18         INTERFACE NO. FROM DB 6 DW 18
BLGR :    KY0, o           FIELD LENGTH FROM DB 6 DW 137
PAFE :    FY199            PARAMETER ASSIGNMENT ERROR BYTE
     :
     :L   MB199            PARMETER ASSIGNMENT ERROR BYTE
     :T   DR92             RECOPYING TO DB PMC DB 6 DW 92
     :
     :L   KB0              RESET HANDLING BLOCK ENABLE BITS
     :T   DL92
     :
     :BE
```

The handling block calls are enabled by the corresponding bit in data byte DL 92 of DB PMC (DB 6) . The enable bits are set automatically by the signaling function blocks. Flag byte FY 199 is selected as the parameter assignment error byte. Before the program exits PB-HDB, FY 199 is copied by DB PMC (DB 6) into DR 92 and all the enable bits are reset.

The handling blocks fetch the parameters SSNR, A-NR, QTYP/ZTYP, QANF/ZANF and QLAE/ZLAE in DB PMC from DW 18 or DW22. They are written by the function blocks START-UP, PDT, GET and POT-A. You specify the parameters in DB PARA-S for the PUT triggers (SEND DIRECT jobs) and in DB PARA-R for the GET triggers (RECEIVE DIRECT jobs). The replies to the handling block calls (condition code ANZW and parameter assignment error byte PAFE) are also located there. You can use these to check the processing of the jobs. Their evaluation is explained in the description of the handling block.

7.2.7 Start-up Organization Blocks

Start-up identifiers must be set and the function block START-UP called unconditionally in the start-up organization blocks OB 20, OB 21 and OB 22. FB-START-UP sets defaults and synchronieses the printer interface (and other interfaces, if programed).

You specify the SYNCHRON parameters SSNR and BLGR in DB PMC (DB 6) in data words DW 136 and DW 137 (interface c printer). The parameter assignment error byte of the SYNCHRON appears in data word DW 138.

The start-up routine is different for a cold restart (OB 20) and a warm restart (OB 21 and OB 22):

organization block OB 20

```
:JU PB6
:BE
```

Organization blocks OB 21 and OB 22

```
:JU PB8
: BE
```

The cold restart program block PB 6 is supplied with the "signaling functions for standard CPs".

```
:C   DB6                SELECT DB PMC
:L   KB0
: T  DL8                RESET START-UP IDENTIFIERS
:AN  D 8.15
:S   D 8.15             SET COLD RESTART IDENTIFIER
:
:JU  FB130
NAME :START-UP

:A   D 8.15
:R   D 8.15             RESET COLD BESTART IDENTIFIER
:AN  D 8.13
:S   D 8.13             SET AUXILIARY IDENTIFIER
: BE
```

Warm restart program block PB 8 is also supplied

```
:C   DB6                SELECT DB PMC
:L   KB0
:T   DL8                RESET START-UP IDENTIFIERS
:AN  D 8.14
:S   D 8.14             SET WARM RESTART IDENTIFIER
:
:JU  FB130
NAME :START-UP
:
:A   D 8.14
:R   D 8.14             RESET WARM RESTART IDENTIFIER
:AN  D 8.13
:S   D 8.13             SET AUXILIARY IDENTIFIER
:BE
```

FB 130 must also be called in organization block OB 1 at the beginning of the cyclic program (see Section 7.2.8).

7.2.8 Cyclic Program

The triggering of the PC jobs must be programmed in organization block OB 1. In addition, you must also call up the start-up function block FB 130 and the program block PB-MELD PB 3.

The digital output Q 0 .n (n = job number) indicates whether the job was aborted or not. Normally in this situation an error routine tailored to the particular application is called up. If an error occurs, evaluate the error numbers in the condition codeword, parameter assignment error byte (words 7 and 9 in the corresponding parameter field of the DB PARA-S) and the error message area of the SYSTAT (see Section 11).

```
     :C   DB6                  SELECT DB PMC

     :JU  FB130                CALL START-UP FB
NAME : START-UP
     :
     :AN I 0.3                 EDGE EVALUATION:
     :R   F 2.3                THE RESULT OF LOGIC OPERATION
     :A   I 0.3                (RLO) IS SET FOR ONE CYCLE IF THE
     :AN F 2.3                 SIGNAL AT INPUT I 0.3 CHANGES
     :S   F 2.3                FROM 0 TO 1.

     :C   DB15                 SELECT DB PARA-S
     :=   D 130.0              SET INITIATION BIT
     :S   D 130.2              NO DATA TRANSFER
     :
     :C   DB6                  SELECT DB PMC
     :JU  FB141                CALL PC JOB NEW PACE
NAME :PUT
BLNR :     KF+14              FIELD NUMBER IN DB PARA-S
     :
     :C   DB15                 SELECT DB PARA-S
     :A   D 130.5              JOB ABORTED AFTER 3 ATTEMPTS,
     := Q 0.1                  INDICATE AT OUTPUT Q 0.1
```

OB 1 continued:

```
     :AN  10.4          EDGE  EVALUATION:
     :R   F 2.4         THE RESULT OF LOGIC OPERATION
     :A   I 0.4         (RLO) IS SET FOR ONE CYCLE IF THE
     :AN  F2.4          SIGNAL AT INPUT I 0.4 CHANGES
     :S   F 2.4         FROM 0 TO 1.

     :=   D 140.0       SET INITIATION BIT IN DB PARA-S
     :S   D 140.2       NO IMMEDIATE DATA TRANSFER

     :C   DB6           SELECT DB PMC
     :JU  FB141         CALL PC JOB PSL-EXAMPLE
NAME :PUT
BLNR :     KF+15       FIELD NUMBER IN DB PARA-S

     :C   DB6           SELECT DB PMC
     :JU  FB143         SEND ALL FUNCTION
NAME :PUT-A

     :C   DB15          SELECT DB PARA-S
     :A   D 140.5       JOB ABORTED AFTER 3 ATTEMPTS,
     :=   Q 0.2         INDICATE AT OUTPUT Q 0.2

     :JU PB3            PB-MELD : SIGNALLING FUNCTION BLOCKS

     :C   DB15          SELECT DB PARA-S
     :A   D 40.5        SML PC JOB ABORTED
     :=   Q 0.3         INDICATE AT OUTPUT Q 0.3
     :BE
```

7.2.9 Transmit Clock Pulse

Message records will be transferred to the CP 525 if

- DB M-RECORD is 90% full of message records or
- the transmit clock pulse (DB PMC DW 8 bit 8) is "1".

By inverting data bit D 8.8 in DB PMC, generate a transmit clock pulse of200ms in organization block OB 13 as foil-:

```
:C   DB6              SELECT DB PMC
:
:AN  D 8.8
:=   D 8.8            INVERT TRANSMIT CLOCK PULSE BIT
:
:BE
```

Transfer all the organization, program, signalling function and message data blocks to the user memory of the CPU. Copy data block DB 30 for the variables in the messages and the process status list (see Section 6.2) from the file PT88R1ST.S5D and transfer them to the CPU. Switch the CPU to "RUN".

7.3 STEP5 User Program with COM PMC

In this section the same messages will be triggered as were programed with the "signaling functions for standard CPs" in Section 7.2, however, in this case with the S5-DOS programming package COM PMC.

With COM PMC, all the parameters for the "signaling functions for standard CPs" are assigned in user-friendly screen menus. COM PMC generates the data blocks required for the signaling functions and saves the parameters in these data blocks.

The CP 525 user program remains unchanged (see Section 4.3, 5.1, 6.1 and particularly 7.1).

The triggering of the messages is to be integrated into the already existing program PT88R1ST.S5D (see Section 5.2 and 6.2). Transfer all function, program, data and organization blocks except OB 1 of the "signaling functions for standard CPs" MEL135ST.S5D for the Reprocessor to the file PT88R1ST.S5D.

The jobs for a new page and a process status list are already programed in OB 1 of PT88R1ST.S5D. These remain part of the program.

During the start-up, the SYNCHRON handling block is no longer called immediately, but rather the START-UP function block (see Section 7.2. 7). The parameter assignment error byte of the SYNCHRON for the printer interface is now in DB PMC DB 6 in data word 138.

Delete the first statement in organization block OB 1:

```
:A   F 10.0
```

For message processing, only the function block START-UP and the program block PB-MELD PB 3 must be called in organization block OB 1. Digital output Q 0.3 indicates whether the SML PC job was aborted (see Section 7.2 .8).

```
:C   DB6              SELECT DB PMC

:JU  FB130            CALL START-UP FB
W : START-UP

:C   DB6              SELECT DB PMC
:A   D 138.0          IF SYNCHRON PARA , ASS . ERROR
:= Q 0.0              OCCURRED --> SET OUTPUT

:JU PB3               PB-MELD: SIGNALING FUNCTION BLOCKS

:C   DB15             SELECT DB PARA-S
:A   D 40.5           SML PC JOB ABORTED
:= Q 0.3              INDICATE AT OUTPUT Q 0.3


:Triggering PC job NEW PAGE see Section 5.2

:
: Triggering PC job PSL EXAMPLE see Section 6.2
:
:BE
```

7.3.1 Installing COM PMC on the PG 685

You *require* a PG 685 programmer, on which the STEP5 basic package is already installed.

Go to drive B: user area 0 by inking the following entry:

```
0:
```

Copy the contents of the COM PMC diskette to the hard disk drive, user area 0 with the following command:

```
PIP B:=A:*.*[R V]
```

```
Option V = verify
       R = copy SYS files
```

Declare the files as write-protected system files with the following commands :

```
SET S5??C?4X. CMD[SYS RO]
SET S5??C?4X. DAT[SYS RO]
```

You can now work in any user area with COM PMC.

User area 0 should be reserved for system files. Exit this area with the command as follows:

```
n:      n = required user area.
```

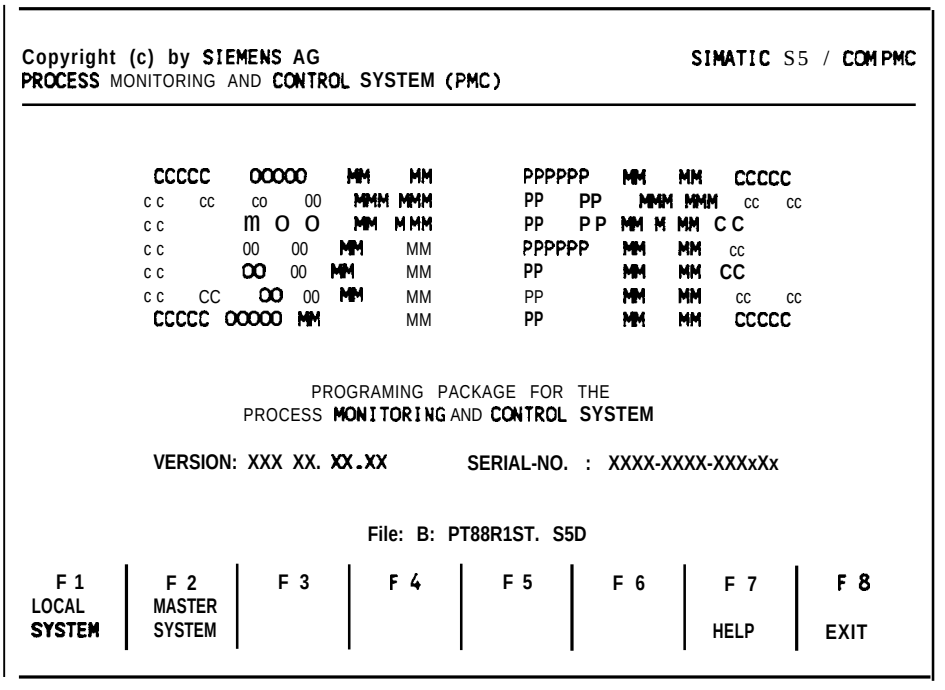
7.3.2 Parameter Assignment with COM PMC

Call the "PACKACE SELECTION" mask with the following command:

S5

Position the cursor on the "COM PMC ., ," line and select the COM PMC programing software with F1 (PACKACE).

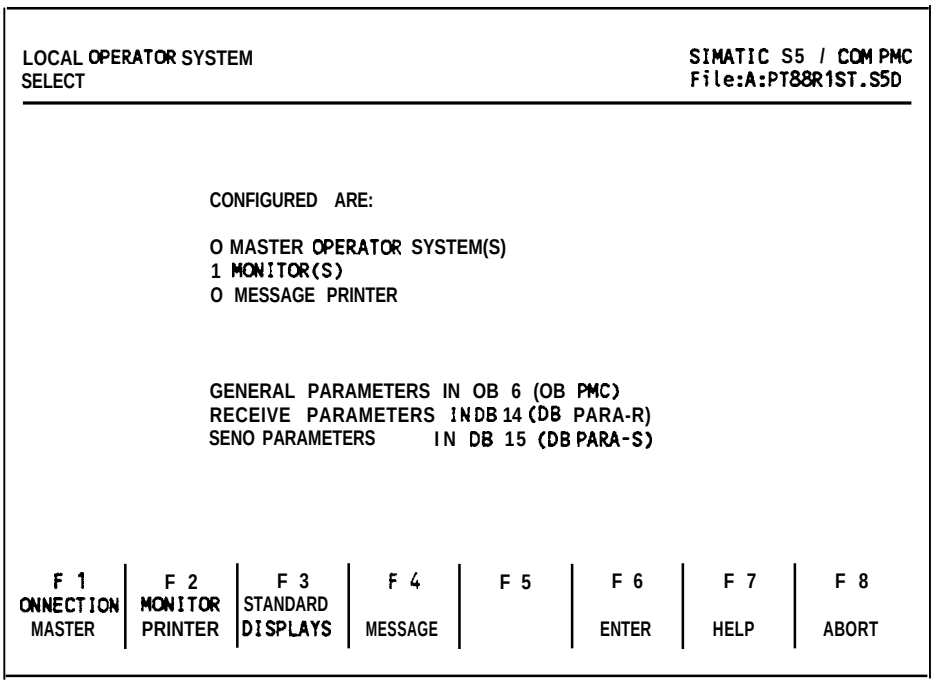
The COM PMC logo mask appears.



Enter drive "B" and the program name "PT88R1"ST. S5D" in the input field. You can enter the file name with F7 (HELP).

Listing belongs to the local system functions. Ress function key F1 (local system) to branch to the “select” mask.

This mask already has standard values as defaults that can be used for this example.

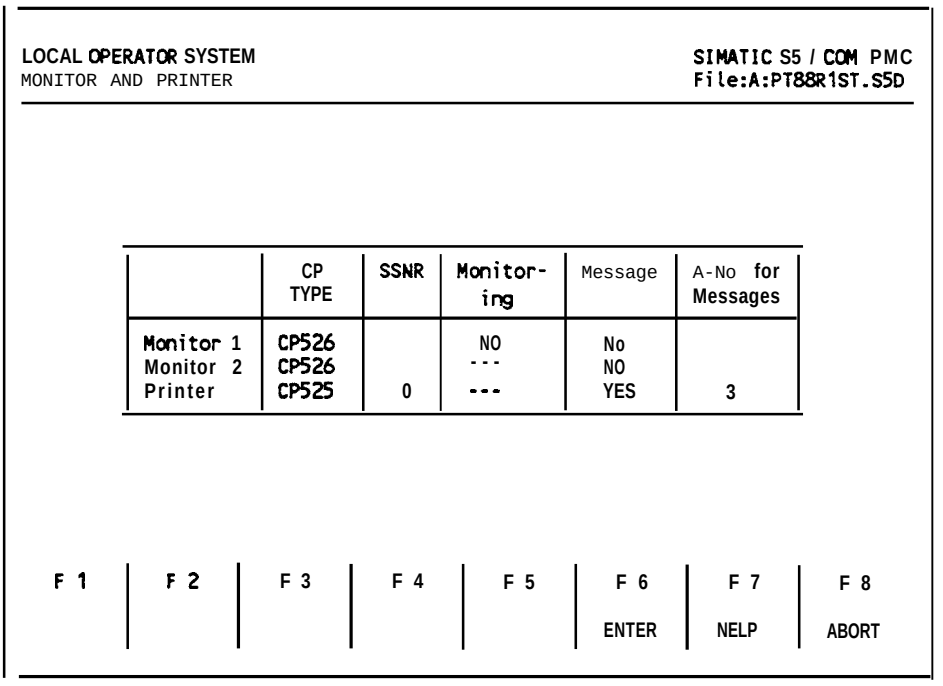


This mask also shows that a monitor but no printer is planned.

Press F2 (monitor printer) to branch to the "monitor and printer" mask.

The messages are not required on the monitor via the CP 525 but only on the printer. Overwrite the interface number of monitor 1 with blanks and enter "no" in the "message" column on the same line.

In the "printer" line select "0" as the interface number (SSNR) of the CP 525. This is the number of the upper interface 1 (IF 1) set cm your module. Press F7 (HELP) to enter "yes" in the "message" column. The job number for the messages must be the same as the job number for the sequential message list on the CP 525. The preset job number 199 must therefore be changed to "3" in the "A-NO" column. Enter the parameters with F6 (ENTER).



LOCAL OPERATOR SYSTEM
MONITOR AND PRINTER

SIMATIC S5 / COM PMC
File:A:PT88R1ST.S5D

	CP TYPE	SSNR	Monitor-ing	Message	A-No for Messages
Monitor 1	CP526		NO	No	
Monitor 2	CP526		---	NO	
Printer	CP525	0	---	YES	3

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
					ENTER	NELP	ABORT

The next stage is to specify the message numbers and data blocks required by the "signaling functions for standard CPS" in the messages mask. Select the "messages" mask with F4 (message) in the "select" mask.

The "message numbers" must be overwritten. "100" is entered for the first message. The COM PMC package can only process the messages in words; therefore assign "115" as the last message although you have only programmed up to message 110. The messages with parameters are located in the area between 101 and 110; therefore enter "101" and "110" in the "messages with parameter" field. Do not enter anything in the status or system messages fields.

Enter the data block numbers as used in the file MEL135ST. S5D and add DB "99" for the message parameters. DB M-ACKN1 and DB M-ACKN2 are required since they are called in PB MELD PB 3 by FB QUIT FB 153.

LOCAL OPERATOR SYSTEM — SIMATIC S5 / COM PMC
MESSAGES — File:A:PT88RI ST. S5D

MESSAGE NUMBERS:	100 - 115
MESSAGES WITH PARAMETER:	101 - 110
STATUS MESSAGES:	
SYSTEM MESSAGES:	NO

ADDRESSES OF INPUT FIELDS	IN DB 208	(DB H-INPUT)
MESSAGE BITS FOR EDGE EVALUATION	IN DB 201	(DB M-OLDNEW)
MESSAGE AND ACKNOWLEDGEMENT NUMBERS	IN DB 204	(DB M NUMBER)
MESSAGE SETS	IN DB 205	(DB H-RECORD)
MESSAGE OUTPUT BUFFER	IN DB 206	(DB U-SEND)
ACKNOWLEDGEMT. RECEIVE BUFFER FOR CP526/1	IN DB 202	(DB M-ACKN1)
ACKNOWLEDGEMT. RECEIVE BUFFER FOR CP526/2	IN DB 203	(DB M-ACKN2)
DESCRIPTION OF MESSAGE PARAMETERS	IN DB 16	(DB M-PARAM)

F1	F2	F3	F4	F5	F6	F7	F8
INPUT FIELDS	MESSAGE PARAs				ENTER	HELP	ABORT

The input fields (the flag, data, input and/or output areas) that can trigger messages must now be specified.

Press F1 (input fields) in the "messages" mask to branch to the "input fields" mask. In this mask, the causes of messages are specified according to type, start of area and length.

The messages are to be triggered by input word IW 0. Press F7 (HELP) to enter "I" (input) in the first input field of the "1/O/F/D" column. The number of the first word "IW 0" and the length "1" word are entered in the "word no." and "length" fields.

LOCAL OPERATOR SYSTEM Messages — SIMATIC S5 / COM PMC
INPUT FIELDS — File:A:PT88R1ST.S5D

MESSAGE NUMBERS: 100 - 115 STATUS MESSAGES:
MESSAGES WITH PARAM: 101 - 110 SYSTEM MESSAGES:

Ser. No	I/O/F/D	DB-No	Word No	Length	Message No
1	I		IW O	1	100 - 115
2					116 -
3					
4					
5					
6					
7					
8					
9					
10					

F 1	F 2 INSERT LINE	F 3 ERASE LINE	F 4 PAGE BACKWARDS	F 5 PAGE FORWARDS	F 6 ENTER	F 7 HELP	F 8 ABORT

The message numbers are displayed automatically by COM PMC.

Ress F6 (enter) to enter the input fields and return to the "messages" mask.

You must now specify the message parameters. Press F2 (message parameters) in the "messages" mask to branch to the 'message parameters" mask.

Press F3 (enter parameter) to move to the 'parameter type" column and the first message with parameter; in this case message 101.

Press F7 (HELP) to enter the appropriate parameter type in the input field. Enter the parameter sources in the messages as follows :

LOCAL OPERATOR SYSTEM Messages — SIMATIC S5 / COM PMC
MESSAGE PARAMETERS — File:A:PT88R1ST.S5D

MESSAGE NUMBERS: 100 - 115 MESSAGES WITH PARAN: 101 - 110

Mess. No.	Bit Address of message I/O/F/D	DB-No	WORD BIT	Parameter type	Bit Address of message I/O/F/D	DB-No	WORD BIT
100	I		I 1.0	———			
101	I		I 1.1	BINARY	D	DB 30	D 1.4
102	I		I 1.2	CHAR	D	DB 30	DL 2
103	I		I 1.3	BYTE	D	DB 30	OR 7
104	I		I 1.4	16 BIT BCD	D	OB 30	DU 8
105	I		I 1.5	16 BIT FXP	D	DB 30	DW 9
106	I		I 1.6	32 BIT BCD	D	DB 30	00 10
107	I		I 1.7	32 BIT FXP	D	DB 30	DD 12
108	I		I 0.0	32 BIT FLP	D	DE 30	DD 14
109	I		I 0.1	TIMER	D	DB 30	DW 16

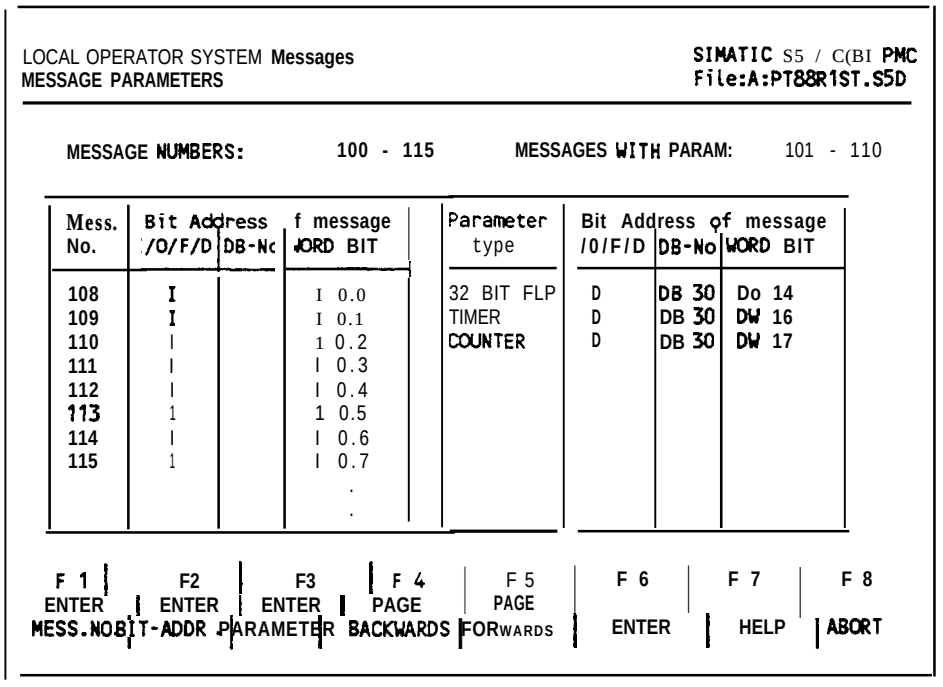
F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
ENTER MESS.NO.	ENTER BIT-ADDR.	ENTER PARAMETER	PAGE BACKWARDS	PAGE FORWARDS	ENTER	HELP	ABORT

Press F1 (enter mess. no.) to position the cursor in the 'mess. no." input field. Enter the required message number and press F2 or F3. The message then appears at the first position.

Press F2 (enter bit-addr.) to move the cursor to the "1/O/F/D" input field. Enter the required input and then the bit address in the next input fields and press F1 or F3. Following this, the message appears at the first position.

Press F5 (page forwards) in the “message parameters” mask to obtain the higher message numbers.

Press F4 (page backwards) to obtain lower message numbers.



Ress F6 (enter) to return to the "messages" mask. Now press F6 (enter) again to branch to the “select” mask. The parameter assignments are saved on the hard disk in the file B:PT88R1ST.S5D by pressing F6 (enter) in the “select” mask. Data blocks that do not exist in this file are generated; in the example this applies to data block M-PARAM DB 99.

Transfer all the blocks in the user memory of the CPU. Transfer DB 30 from Section 6.3.1 to the CPU as source for the message variables.

switch the CPU to “RUN”.

7.4 Test

Switch all the digital inputs which trigger messages from 0 to 1. All the messages will be printed with the status coming (1):

```
12.31 11:58:42 pm MESSAGE 100 static text and acoustic signal 1
12.31 11:58:43 pm MESSAGE 101 binary value:          1          1
12.31 11:58:44 pm MESSAGE 102 character              A          1
12.31 11:58:45 pm MESSAGE 103 byte variable:         123        1
12.31 11:58:46 pm MESSAGE 104 16-bit BCD number:     -123       1
12.31 11:58:&6 pm MESSAGE 105 16-bit fixed pt. no.:  12345      1
12.31 11:58:47 pm MESSAGE 106 32-bit BCD number:     -1234567   1
12.31 11:58:48 pm MESSAGE 107 Z-bit fixed pt. no.:   77881  1
12.31 11:58:52 pm MESSAGE 108 floating point number  12.3457  1
12.31 11:58:53 pm MESSAGE 109 timer value            @ J   1
12.31 11:58:53 pm MESSAGE 110 counter value          -123       1
```

If you now switch all the inputs which trigger messages from 1 to 0 the CP prints out all the messages with the status going (0):

```
01.01 00:00:13 pm MESSAGE 100 static text and acoustic signal 0
01.01 00:00:14 pm MESSAGE 101 binary value:          1          0
01.01 00:00:15 pm MESSAGE 102 character              A          o
01.01 00:00:16 pm MESSAGE 103 byte variable:         123        Q
01.01 00:00:19 pm MESSAGE 104 16-bit BCD number:     -123       0
01.01 00:00:20 pm MESSAGE 105 16-bit fixed pt. no.:  12345      o
01.01 00:00:22 pm MESSAGE 106 32-bit BCD number: ,   -1234567   o
01.01 00:00:24 pm MESSAGE 107 32-bit fixed pt. ea.:  77881  D
01.01 00:00:29 pm MESSAGE 108 floating point number  12.3457  0
01.01 00:00:30 pm MESSAGE 109 timer value            123.3  0
01.01 00:00:31 pm MESSAGE 110 counter value          123        o
```

If you have not connected a simulator, select flag word 0 for triggering the messages. Switch the flag bits with the PG 685 online function CONTROL VARIABLE. Type 3 (flagword)mustbe programed in DB M-INPUT as follows:

```
DB M-INPUT DB 208 data word DUO: KY= 003,000;
```

In this case do not use any digital outputs. Observe the condition codewords (DU 47, DW 137 and DW 147 in DB PARA-S DB 15) belonging to the jobs and the *parameter* assignment error bytes (DW 49, DW 139 andDW 149 DB PARA-S DB 15) using CONTROL VARIABLE.

Assuming, for example, that messages 100 and 101 arrive in one program cycle, the data block DB M-SEND DB 206 appears as follows (see also "Interface between CPU and CP 525" in the instructions "Event output and logging with the PT88/PT89 printer" in this manual):

```
 0 :    KY= 051,014;     identifier 33H/body length 14 data words
 1 :    ICY= 255,001;    no parameter / message coming
 2:     KY= 000,100;     disturbance message /message 100
 3:     KH= 3045         300 milliseconds / 45 seconds
 4:     KH= 3210         32 minutes / 10 hours
 5:     KH= 2810         28th October
 6:     KH= 8700         1987
 7:     KY= 000,001;     binary variable /message coming
 8:     KY= 000,101;     disturbance message /message 101
 9:     KH= 3045         300 milliseconds / 45 seconds
10 :    KH= 3210         32 minutes / 10 hours
11 :    KH= 2810         28th October
12 :    KH= 8700         1987
13 :    KM= 0010         binary variable data bit  4=1
14 :    KM= 0400         bitaddress4
```

If you transfer this data block using a SEND DIRECT job with job number A-NR 3, source tgpe QTYP DB, data block number DBNR 206, source start QANF 0 and source length QLAE 15 to the CP 525, the messages 100 and 101 are printed out as 'coming" (see Section 7.5).

7.5 STEP5 User Program without 'Signaling Functions for Standard CPs"

This section explains how to trigger the messages without "signalling functions for standard CPs". The simplest case is demonstrated in which the DB M SEND contains only one message record and not up to 31 or 42 (without parameter or with parameter) as when messages are triggered with the standard function blocks. This message record is transferred to the CP 525 by a handling block.

By varying this message record you can get to know how the interface between the CPU and CP 525 functions when messages are triggered (see also Section 2.2 in the user's guide 'Event output and listing with the PT 88/PT 89 printer" in this manual).

Call the S5 package IAI), CSF, STL. Program in STL (Statement List) and call your new STEP5 program "B:PT88R3ST.S5D". Transfer the handling blocks for the R processor to this program file. Connect the Reprocessor with the PG 685.

The STEP5programmust

synchronize the CPU and CP 525 during start-up and trigger the PC job for the output of messages.

7.5.1 Program Start-up

All the start-up organization blocks OB 20, OB 21 and OB 22 call the handling block SYNCHRON unconditionally.

```
     :JU  FB125                    SYNCHRONIZATION CP <--> CPU
NAME : SYNCHRON
SSNR :    KY0 ,0                   INTERFACE NUMBER 0
BLGR :    KY0 ,2                   BLOCK SIZE 2 (MAX. 32 BYTES)
PAFE :    FY10                     PARMETER ASSIGNMENT ERROR BYTE

     :BE
```

The interface number SSNR for IF 1 of the CP 525 is the same as the set module address '0". As block size select "2". This means that data can be transferred between the CPU and CP in groups of a maximum of 32 bytes. If there are more than 32 bytes to be transferred, the CP 525 requests the excess bytes with another SEND AIL. The block size is selected so that DB M-SEND can be transferred in one operation. Any other block size could be selected.

Use "FY10" as the parameter assignment error byte PAFE.

Any parameter assignment error is to be indicated at digital output Q 0.0. Since there is no process image in the start-up organization blocks and therefore no digital 1/0s can be addressed, this indication can only occur in the cyclic program (see Section 7.5. 2). Normally an error evaluation program is called during the start-up. If a parameter assignment error occurs during the start-up, check that the interface number and module address are the same ("0"). If both are set to '0', there is probably a hardware fault.

7.5.2 Cyclic Program

Any parameter assignment error detected during the start-up is indicated at digital output Q 0.0 at the beginning of the cyclic program (see Section 7.5 .1).

The PC job sequential message list is triggered by a SEND DIRECT; i.e. by calling a SEND handling block with a job number other than 0. The job number of the SEND DIRECT and the number of the SML PC job on the CP 525 must be the same. Program the job number as A-NR "3". The job is executed if a positive-going edge is detected at input I 0.3.

The interface number is SSNR "0". Select condition codeword ANZW "FW11" and parameter assignment byte PAFE "FY15". The condition codeword occupies two words for a SEND DIRECT call. The job status and any error messages are indicated in flag word FW 11.

For each message output, data block M-SEND must be transferred from the CPU to the CP 525; select data block DB 20. "DB" must be programmed as the source type QTYP; the source parameters are DBNR "20", QANF "0" and QLAE "9". (If there are several message records of six or eight data words in DB M-SEND, the source length must be adjusted accordingly.) The SEND DIRECT call triggers the job; i.e. the job is entered in the job queue. Data M-SEND DB 20 is only requested by the CP 525 by a SEND ALL.

The job is triggered on result of logic operation (RLO) "1". If the RLO is "0" when the call occurs, only the condition codeword of the SEND DIRECT is updated.

If the job is completed with an error or if a parameter assignment error occurs, this is displayed at digital output Q 0.3. Normally, in this case, a special error routine is called. If an error occurs, evaluate the error numbers in the condition codeword in the error message area of the SYSTAT (see Section 11) and in the parameter assignment error byte.

The program in the cyclic organization block OB 1 is as follows:

```
      :A   F 10.0          IF SYNCHRON PARAMETER ASSIGNMENT
      :=   Q 0.0           OCCURRED --> SET OUTPUT

      :AN  10.3            EWE EVALUATION
       :R  F 2.3           THE RESULT OF LOGIC OPERATION
      :A   10.3            (RLO) IS SET FOR ONE CYCLE,
      :AN  F 2.3           IF THE SIGNAL AT INPUT I 0.3 CHANGES
      :S   F 2.3           FROM o TO 1

      :JU  FB120           TRIGGER SML PC JOB
NAME :SEND
SSNR :     KY0,0          INTERFACE NUMBER 0
A-NR :     KY0,3          JOB NUMBER 3
ANZW  :    FW11           CONDITION CODEWORD
QTYP :     KCDB           SOURCE m DB
DBNR :     KY0,20         DATA BLOCK DB 20
QANF  :    KF+0           SOURCE START DW 0
QLAE :     KF+9           SOURCE LENGTH 9 DATA WORDS
PAFE :     FY15           PARMETER ASSIGNMENT ERROR BYTE

      :JU  FB126           TRANSFER DB M-SEND TO CP 525
NAME: SEND-A
SSNR :     KY0 ,0         INTERFACE NUMBER 0
A-NR :     KY0 ,0         JOB NUMBER 0
ANZW  :    FW16           CONDITION CODEWORD
PAFE  :    FY18           PARAMETER ASSIGNMENT ERROR BYTE
      :
      :
      :O F 1.2.3           JOB FINISHED WITH ERROR
      :O F 15.0            OR PARAMETER ASSIGNMENT ERROR
      := Q 0.3             INDICATE AT OUTPUT Q 0.3

      :BE
```

Structure of DB M-SEND

	Left byte DL	Right byte DR
DUO	Message ID = 33H	Body length = 6 or 8 DW
DW 1	Parameter type	Message status 0 or 1
DW 2	Message type = 0	Message number
DW 3	100 ms	Seconds
DW 4	Minutes	Hours
DW 5	Day	Month
DW 6	Year	not used
DW 7	PARAMETER VALUE	
DW 8	PARAMETER VALUE	

DB M-SEND parameters for the messages from Section 7.1

Mess. no. (DW 2)	Parameter type	HEX (DL 1)	PARAMETER (DW 7)	PARAMETER (DW 8)	status (DR 1)
100	No. var.	FF	———	———	o or 1
101	Binary	00	KH= 0010	KH= 0400	0 or 1
102	Character	01	KS= BE	KH= 0800	0 or 1
103	Byte	02	KY= 0,123	KH= 0000	0 or 1
104	16 bit bcd	03	KH= o123	KH= 0000	0 or 1
105	16 bit fpt	05	KF= +12345	KH= 0000	0 or 1
106	32 bit bcd	04	KH= F123	KH= 4567	0 or 1
107	32 bit fpt	06	KF= +00001	KF= +12345	0 or 1
108	floating pt.	07	KG= +123456	+02	0 or 1
109	timer	08	KT= 123.3	KH= 0000	0 or 1
110	counter	09	KC= 123	KH= 0000	0 or 1

For parameter type FF (no parameter) the message record is only six (instead of eight) data words long; i.e. DR O = 6 (instead of DW O = 8).

You can read the date and time using the PC job RECEIVE DIRECT 218 (see Section 10. 2) from the CP and enter them in data words 3 to 6. Enter the value "O" for day so that the CP 525 reads the date and time directly from its hardware clock.

7.5.3 Testing the Program

Using the PG 685 online function CONTROL VARIABLE, supply the data block M-SEND DB 20 with data (see table). Monitor the condition codeword FW 11 and the PAFE byte FY 15 using the same CONTROL VARIABLE function.

If you switch the signal at digital output I 0.3 fram O to 1, the printer will print the appropriate message.

8 Current Message List (CML)

Using the CML you can print out the disturbance status of a process at a particular time. All the messages that have cane and not yet gone again, i.e. are still "active" on the CPU at the time of the CML call are output to the printer by the CP 525.

The CP 525 initiates the printout by requesting the data block DB M-OLDNEW immediately after the PC job is triggered by the CPU. The CP outputs to the printer the message texts belonging to all the bits set in the new value area of the DB M-OLDNEW (see also Section 8.3 and the description of "The interface between CPU and CP 525" in the user's guide "Event output and listing with the PT88/PT89 printer" in this manual) .

Instead of the dynamic data (date, the, message status and process variable) question marks are printed.

Note: if you use the "signaling functions for standard CPs", DB M-OLDNEW is generated and updated automatically. Without the "signaling functions for standard CPS" you must ensure that the required data block is available cm the CPU.

8.1 CP 525 User Program

Switch the mode selector of the CP 525 to "P(2R".

Press F1 (PROGRAM USER DATA) in the 'SELECTION' mask to call the 'PROGRAM' mask and press F2 (PC JOB) to call the 'PC JOB' mask. F3 (CML) then calls the 'CURRENT MESSAGE LIST' :

```
-> PROGRAM ->PC JOB ->                                  SIMATIC S5 / COM525
CURRENT  MESSAGE  LIST

                                DRIVE: B     PROGRAM:  PT88ABSP   COMPONENT: P T

   List name:          CUR. MESSAGE

   With the job no.:       4
   the output of the current message list is started.

   List header is frame:
   List trailer is franc:

   Output of messages possible while CML is being output          (Y/N) N
   Form feed on PT88/PT89 at start of list                        (Y/N) Y

 F 1 | F 2 | F 3  | F 4 | F 5 | F 6  | F 7  | F 8
     |     | FRAME |     |     | SAVE | HELP | EXIT
```

Enter the job number '4". Do not enter anything for list header and trailer. Block the output of process messages (enter "N") and program a form feed before each CML (enter "Y"). Save the PC job CUR.MESSAGE on the hard disk with F6 (SAVE).

Press F8 (EXIT) three times to return to the 'SELECTION' mask. Transfer the PC job CUR. MESSAGE in the same way that the PC job NEW PACE was transferred to the user memory of the CP 525 (see Section 5.1). After the transfer, carry out a cold restart on the CP 525 and switch the CP 525 to "RUN".

8.2 STEP 5 User Program

Trigger the current message list with a SEND DIRECT job with job number A-NR "4". The data block DB M-OLDNEW is transferred from the Reprocessor to the CP 525 by a SEND ALL call; the SEND ALL is already programmed for the process status list (see Section 6.2) and the sequential message list (see Section 7.2.1).

The termination of the job with an error is indicated at digital output Q 0.4. Normally in this situation an error evaluation program tailored to the particular application is called. If an error occurs, evaluate the error numbers in the condition code-word (FW 24), parameter assignment error byte (PY 28) and the error message area of the SYSTAT (see Section 11).

Add the statements to trigger the PC job to OB 1 as follows:

```
      :AN I 0.5              EDGE EVALUATION:
      :R F 2.5               THE RESULT OF LOGIC OPERATION
      :A  I 0.5              (RLO) IS SET FOR ONE CYCLE IF THE
      :AN F 2.5              SIGNAL AT I 0.5 CHANGES FROM
      :S  F 2.5              0 TO 1
      :
      :JU  FB120             TRIGGER CURRENT MESSAGE LIST
NAME :SEND
SSNR :    KY0,0              INTERFACE NUMBER 0
A-NR :    KY0, 4             JOB NUMBER 4
ANZW :    FU24               CONDITION CODEWORD
QTYP :    KCDB               SOURCE TYPE DATA BLOCK
DBNR :    KY0, 201           SOURCE DATA BLOCK DB M-OLDNEW
QANF :    KF+0               SOURCE START DATA WORD 0
QLAE :    KF+255             SOURCE LENGTH 255 DATA WORDS
PAFE :    FY28               PARAMETER ASSIGNMENT ERROR BYTE
      :
      :O   F 25.3            JOB 4 FINISHED WITH ERROR
      :O   F 28.0            OR PAMMETER ASSIGNMENT ERROR
      :=   Q 0.4             INDICATE AT OUTPUT Q 0.4
```

8.3 Test

Switch on all the inputs which trigger a message (I 0.0 to I 0.2 and I 1.0 to I 1.7).

AS soon as the signal at digital input I 0.5 is switched from zero to one, the following CML is printed out:

```
??.?? ??:??:?? MESSAGE 100 static text and acoustic signal_____?
??. ?7??: ??:?? MESSAGE 101 binary value:          ?          ?
??.?? ??:??:?? MESSAGE 102 character               ?           ?
??.?? ??:??:?? MESSAGE 103 byte variable:          ???        ?
??.?? ??:??:?? MESSAGE 104 16-bit BCD number:      ????       ?
??.?? ??:??:?? MESSAGE 105 16-bit fixed pt. no.: ?  ?'??'?          ?
??.?? ??:??:?? MESSAGE 106 32-bit BCD number: ????????       ?
??.?? ??:??:?? MESSAGE 107 32-bit fixed pt. no.: ?????????? ?
??.?? ??:??: ?? MESSAGE 108 floating point number ??????? ???    ?
??.?? ??:??:?? MESSAGE 109 timer value          ?????       ?
??.??_??:??:??_MESSAGE 110 counter value_____???___/_____?
```

The data block DB M-OLDNEW DB 201 may appear as follows in this situation:

```
 0:    KH= 0001;              new values from data word DW 128
 1 :   KF= +xxxxx             write pointer not relevant
 2:    K = +xxxxx             readpoi.nternotrelevant

 3:    KM= 0000010111111011;  old values
 4:    KM= 0000000000000000;
   :
   :
127 :  KM= 0000000000000000;  end of old value area

128 :  KM= 0000011111111111;  new values
1.29 :  KM= 0000000000000000;
   :
   :
252 :  KM=      0000000000    end of new value area

253 :  KF= +00100             lower limit of message numbers
254 :  KF= +00001             1 word occupied with message bits
```

9 Frames

Each list can have a header and/or a trailer. In SMLs and CMLs, the header is printed out at the beginning of a list and at the beginning of each new page; the trailer is printed out at the end of a list and at the end of each page. In PSLs, they are printed out only at the beginning and end of the list. The list header and trailer are known as frames. They are programed as a PSL, with a maximum of 9 lines, and cannot be started separately by the CPU.

9.1 CP 525 user program

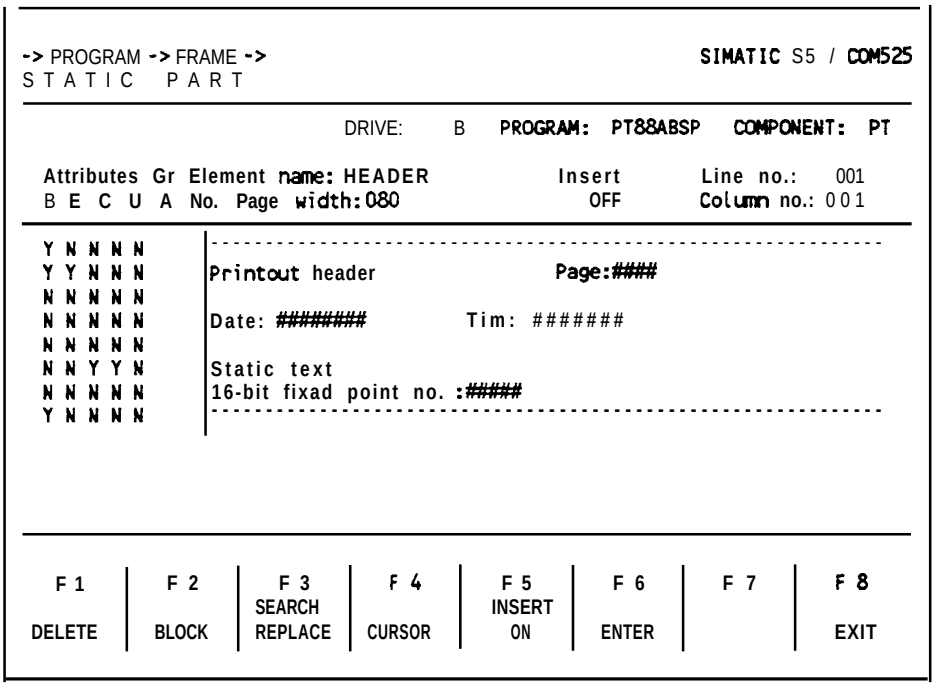
Press F1 (PROGRAM USER DATA) and F3 (FRAME) in the 'SELECTION' mask to call the following mask:

```
-> SELECTION -> PROGRAM ->                                   SIMATIC S5 / COM525
F R A M E
--------------------------------------------------------------------------------
                         DRIVE:   B   PROGRAM: PT88ABSP   COMPONENT: P T

  Frame name: HEADER                Name to be stored: HEADER

  The dynamic data are on CPU no.: 1

  in data block         DB no.: 30

  Frames can be used as a header and/or trailer for process
  status lists, sequential message lists and currant message l i sts.
```

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
STAT. PART	DYN. PART		COMPLETE DYN. PART		SAVE	HELP	EXIT

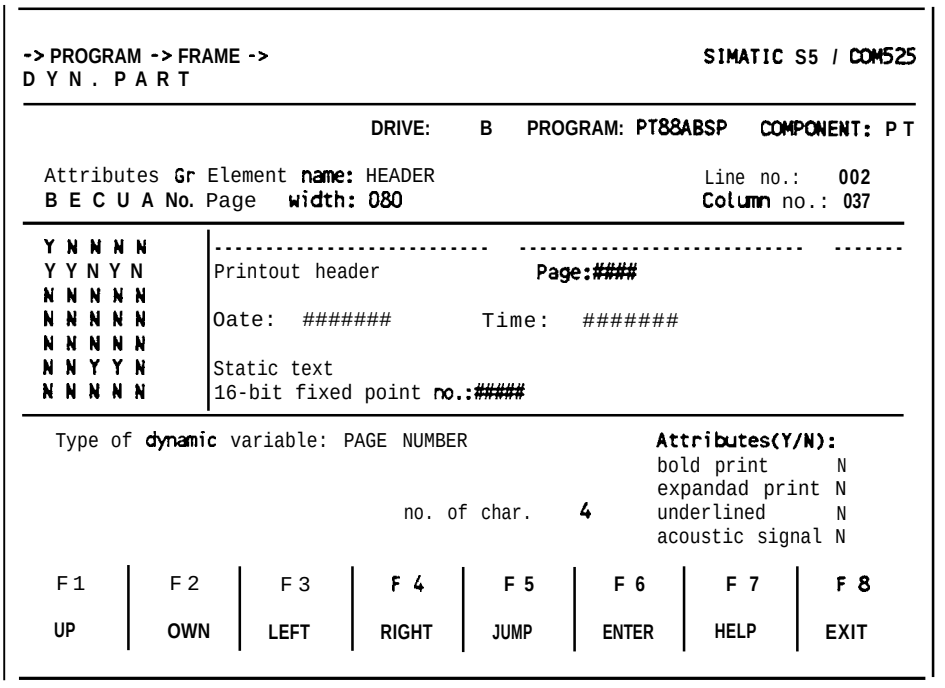
Select the frame name 'HEWER", CPU number "1" and data block DB "30" for the dynamic variables. "HEADER" is automatically the name for saving the frame unless you specify a different name.

Press F1 (STAT. PART) to call the 'STATIC PART' mask. Program the following frame. The editing is performed in just the same way as for the PSL (see Section 6 .1).



Store the static part of the frame on the PG 685 RAM with F6 (ENTER) and exit the 'STATIC PART' mask with F8 (EXIT).

Press F2 (DYN. PART) in the 'FRAME' mask to call the 'DYN. PART' mask. The dynamic variables are programmed in just the same way as those for the PSL (see Section 6.1).



Press F7 (HELP) to enter the variable type, e.g. "PAGE NUMBER". The CP counts the pages printed out up to page 9999. It then returns automatically to page 1. Using the PC job RESET PAGE NUMBER you can reset his counter to 1 at any time from the CPU.

The date and time are to be taken from the CP525 hardware clock and printed out in bold print.

Type of dynamic variable:	DATE	Attributes(Y/N):	
DATE is fetched from	CP 525 - CLOCK	bold print	Y
		expanded print	N
	order D M Y	underlined	N
		acoustic signal	N

Store the specification on the PG 685 RAM with F6 (ENTER) and with F4 (RIGHT) select the next variable.

Type of dynamic variable:	TIME	**Attributes(Y/N):**
TIME is fetched from	CP 525 - CLOCK	bold print Y
		expanded print N
	order H M S	underlined N
		acoustic signal N

Store the specification on the PG 685 RAM with F6 (ENTER) and with F2 (DOWN) select the next variable.

As specified in the 'FRAME' mask, process variables are fetched from CPU 1 data block **DB 30. The** number of reserve characters for dynamic variables (#) is taken as the default for the number of places.

Type of dynamic variables:	PROCESS VARIABLE	**Attributes(Y/N):**
		bold print N
CPU No. 1	Format 16BIT FXP	expanded print N
DB No. 30	no. of char. 6.0	underlined N
Address DW 9.		acoustic signal N

Store the specification on the PG 685 RAM with F6 (ENTER) and exit the mask with F8 (EXIT). Save the static and dynamic part on the hard disk by pressing F6 (SAVE) in the 'FRAME' mask.

Program a purely static frame "TRAILER". Enter the frame name "TRAILER" in the 'FRAME' mask. The CPU number and data block number are not necessary since there are no dynamic fields in this frame.

The frame is to be printed out as follows:

```
------------------------------------------------------------------
Static text
---- ------ -- ------------------- -------- ---- ------------ ---------
```

The broken lines are to be printed out in bold print.

Ress F6 to save the static part, exit the 'STATIC PART' mask with F8 (EXIT) and save the frame on the hard disk with F6 (SAVE) .
Call the PC job CML and enter "HEADER" as the list header and "TRAILER" as the list trailer in the 'CURRENT MESSAGE **LIST'** mask (see Section 8.1). Save the PC job CUR. **MESSAGE** and transfer **it** to

the user memory of the CP 525. Overwrite the CML already in the CP 525 user memory, F1 (YES, in response to the prompt 'ACK. 010: element already exits! - overwrite?').

Transfer both frames to the CP 525 user memory and perform a cold restart on the CP 525 with F1 in the 'TRANSFER' mask.

9.2 Test

The test is performed in the same way as it is for the CML (see Section 8.3). The printout should be as below:

```
---------------------------------------------------------------------
Printout header                                              Page: 7

Date : 12.3 1. 87            Time:      11 :59 : 30 am

static text
1. 6-bit fixed point no.:12345
---------------------------------------------------------------------
??.?? ??:??:?? MESSAGE 100 static text and acoustic signal ?
??.?? ??:??:?? MESSAGE 101 binary value :                   ?
??.?? ??:??:?? MESSAGE 102 character                 ?        ?
??.?? ??:??:?? MESSAGE 103 byte variable:            ???      ?
??.?? ??:??:?? MESSAGE 104  16-bit  BCD  number:     ????     ?
??.?? ??:??:?? MESSAGE 105 16-bit fixed pk. no. :  ??????          ?
??.?? ??:??:?? MESSAGE 106 '32-bit BCD number:       ????????  ?
??.?? ??:??:?? MESSAGE 107 32-bit fixed pt. no.: ????????? ?
??.?? ??:??:?? MESSAGE 108 floating point numb r????????? ?
??.?? ??:??:?? MESSAGE 109 timer value        ?????      ?
??.?? ??:??:?? MESSAGE 110 counter value             ???      ?
---------------------------------------------------------------------
Static  text
---------------------------------------------------------------------
```

10 Date/Time PC Jobs

The CP 525 has an internal hardware clock that is used as the calendar and clock for lists. This clock can be set, read and synchronized with the clocks of other CPs in the same PC.

The PC job number 218 on the CP 525 is reserved for reading and writing the date and time. These special jobs are carried out immediately; i.e. they are not entered in the ten-job queue of the CP 525. No jobs need to be programmed on the CP 525.

10.1 Setting the Date/Time

Enter the value 31st December 1987, 11 p.m. , (23 hours) 58 minutes and 30 seconds in BCD format in data block DB 30 from data word DW 20. The CP 525 is to remain the master clock. For this reason, the master identifier (bit 8) must be set in data word DW 20.

```
20 :    KH= 0100              master identifier set
21 :    KH= 0030              1/10 s, 1/100 s / seconds
22 :    KH= 5823              minutes / hours
23 :    KH= 3112              day / month
24 :    KH= 8700              year / --
```

The set function is to be triggered when the signal at input I 0.6 changes from 0 to 1. The termination of the job is to be indicated at output Q 1.0.

Add the statements for triggering the PC job to OB 1 as follows:

```
      :AN  10.6          EDGE EVALUATION:
      :R   F 2.6         THE RESULT OF LOGIC OPERATION
      :A   I 0.6         (RLO) IS SET FOR ONE CYCLE IF THE
      :AN  F2.6          SIGNAL AT INPUT I 0.6 CHANGES
      :S   F 2.6         FROM 0 TO 1.
      :
      :JU  FB120         TRIGGER SET DATE/TIME
NAME:SEND
SSNR :     KY0,0        INTERFACE NUMBER 0
A-NR :     KY0,218      JOB NUMBER 218
ANZW :     FW29         CONDITION CODEWORD
QTYP :     KCDB         SOURCE m DATA BLOCK
DBNR :     KY0, 30      SOURCE DATA BLOCK DB 30
QANF .     KF+20        SOURCE START DATA WORD 20
QLAE :     KF+5         SOURCE LENGTH 5 DATA WORDS
PAFE :     FY33         PARAMETER ASSIGNMENT ERROR BYTE

      :O   F30.3         JOB FINISHED WITH ERROR
      :O   F 33.0        OR PARAMETER ASSIGNMENT ERROR
      := Q 1. 0          INDICATE AT OUTPUT Q 1.0
```

Test

Switch the digital input I 0.6 from 0 to 1. The date and time of the CP 525 hardware clock are set to the value in data block DB 30. Check that the setting of the date/time was successful simply by reading the date and time on the message printout. After one and a half minutes the CP 525 indicates that the year is changed.

The CP 525 recognizes leap years and when necessary automatically counts 29 days in February.

10.2 Read Date/Time

The current time is to be transferred cyclically from the CP 525 hardware clock to data block DB 30 from data word DW 25. It is stored in the same format as the values prograннnedstartingat DW 20 (see Section 10.1).

The transfer is to take place in every cycle. If an error occurs when the job is being processed it is indicated at output Q 1.1. Add the following statements for triggering the PC job to organization block OB 1:

```
     :JU  FB121                TRIGGER READ DATE/TIME
NAME :RECEIVE
SSNR :    KY0, 0               INTERFACE NUMBER 0
A-NR :    KY0,218              JOB NUMBER 218
ANZW :    FW34                 CONDITION CODEWORD
ZTYP :    KCDB                 DEST TYPE DATA BLOCK
DBNR :    KY0,30               DEST DATA BLOCK DB 30
ZANF :    KF+25                DEST START DATA WORD 25
ZLAE :    KF+5                 DEST LENGTH 5 DATA WORDS
PAFE :    FY38                 PARAMETER ASSIGNMENT ERROR BYTE
     .
     :O   F 35.3               JOB FINISHED WITH ERROR
     :O   F 38.0               OR PARAMETER ASSIGNMENT ERROR
     :=   Q 1.1                INDICATE AT OUTPUT Q 1.1
```

The time canbe read only if the master identifier is set on the CP 525. After the CP 525 module has started up following power up, the master identifier is reset. In the example of message listing (see Sections 7.2 and 7.3), the rester identifier is set during the start-up. You established in the data block DB PMC DB 6 in data word DW 6 that the CP 525 is the master clock.

Test

Observe data block DB 30 from DW 25 to DW 29 with the PG 685 online function CONTROL VARIABLE. Tenths of seconds, seconds and minutes must change constantly.

10.3 Synchronizing Several CP Hardware Clocks

To synchronize the hardware clocks of several CP modules, the date and time must be transferred from a CP that is declared as time master to the other communications processors.

The time is read by a RECEIVE DIRECT 218 call or by a corresponding GET call (see Section 7.2. 5). 'l'he time is set by a SEND DIRECT 218 call or a PUT call (see Section 7.2. 5).

A CP is declared time master with a SEND DIRE(7I 218 if bit 8 is set in the first data vord transferred. If this bit is O the CP is declared time slave.

If only one data word is transferred with the SEND DIRECT 218 (QLAE = 1), only the master identifier is affected and the time is not set.

If five data words are transferred with the SEND DIRECT 218 (QLAE = 5), the master identifier, date and time are transferred. The hardware clock is set accordingly.

11 Reading the Error Message Area in the SYSTAT

The SYSTAT is a memory area on the CP 525 that contains information separately for each interface.

Within the error message area of the SYSTAT the causes of errors are coded in detail. The error message area is 4 bytes = 2 words long and appears as below:

Byte 0 (DL 0)					E	O	B	R
Byte 1 (DR 0)	error number 1							
Byte 2 (DL 1)	error number 2							
Byte 3 (DR 1)	error number 3							

E - set, if error is entered in the SYSTAT
O - set, if there is an error overflow (more than 3 errors)
B - set, if there is a BREAK on the interface
R - set, if the printer is not ready to receive for longer than one minute (XOFF status)

The PC job RECEIVE DIRECT 200 is reserved for reading the error message area. This job is performed immediately, i.e. it is not entered in the 10-job queue of the CP 525, No job needs programming on the CP 525.

Transfer the error information in data block DB 30 starting at data word DW 30. The job is called unconditionally in every program cycle but is executed only if an error occurs.

When RESET DIRECT 200 is called, the error message area of the interface addressed in the SYSTAT (in our example SSNR = 0) is reset. The data words DW 30 and DW 31 in data block DB 30, into which the error messages are written by RECEIVE DIRECT 200, must be reset separately.

Reset the error message area with a positive going edge at digital input I 0.7.

Add the following statements to organization block OB 1.

```
     :O   F 3.0
     :ON F 3.0              GENERATE RLO = 1

     :JU  FB121             TRIGGER READ DATE/TIME
NAME :RECEIVE
SSNR :    KY0,0            INTERFACE NUMBER 0
A-NR :    KY0,200          JOB NUMBER 200
ANZW  :  FW39              CONDITION CODEWORD
ZTYP :    KCDB             DEST TYPE DATA BLOCK
DBNR :    KY0,30           DEST DATA BLOCK DB 30
ZANF :    KF+30            DESTSTART  DATAWORD30
ZLAE :    KF+2             DEST LENGTH 2 DATA WORDS
PAFE :    FY43             PAMMETER ASSIGNMENT ERROR BYTE

     :O   F40.3            JOB FINISHED WITH ERROR
     :O   F 43.0           OR PAMMETER ASSIGNMENT ERROR
     :=  Q  1.2            INDICATE AT OUTPUT Q 1.2

     :AN 10.7              EDGE EVALUATION
     :R  F2.7              THE RESULT OF LOGIC OPERATION
     :A   10.7             (RLO) Is SET FOR ONE CYCLE,
     :AN F2.7              IF THE SIGNAL AT INPUT I 0.7
     :S   F2.7             CHANGES FROM o To 1

     :JC  FB100            RESET SYSTAT OF INTERFACE 0
NAME :R-SYSTAT
```

Function block FB 100

```
NAME :R-SYSTAT

     :JU  FB124              RESET SYSTAT
NAME :RESET
SSNR :    KY0,0              INTERFACE NUMBER 0
A-NR :    KY0,200            JOB NUMBER 200
PAFE :    FY198              PARAMETER ASSIGNMENT ERROR BYTE

     :C   DB30               IN DATA BLOCK DB 30
     :L   KF+0               DELETE DATA WORDS DW 30 AND DW 31
     :T   DW30
     :T   DW31

     :BE
```

Transfer organization block OB 1 and function block FB 100 to the user memory of the R processor. Carry out a cold restart. The green RUN LED on the R processor must light up.

Test

Observe data block DB 30, DW 30 and DW 31 with the PG 685 online function CONTROL VARIABLE and then unplug the printer cable at the CP 525.

KH= 0AFF appears immediately in data word 30; i.e. the bits 'error' and 'BREAK' are set and the error FF 'BREAK on the line' is displayed.

As soon as you trigger messages you obtain KH= OEFF in data word DW 30 and KH= FFFF in DW 31; further FF errors are entered and for this reason the 'overflow' bit is set in the first byte.

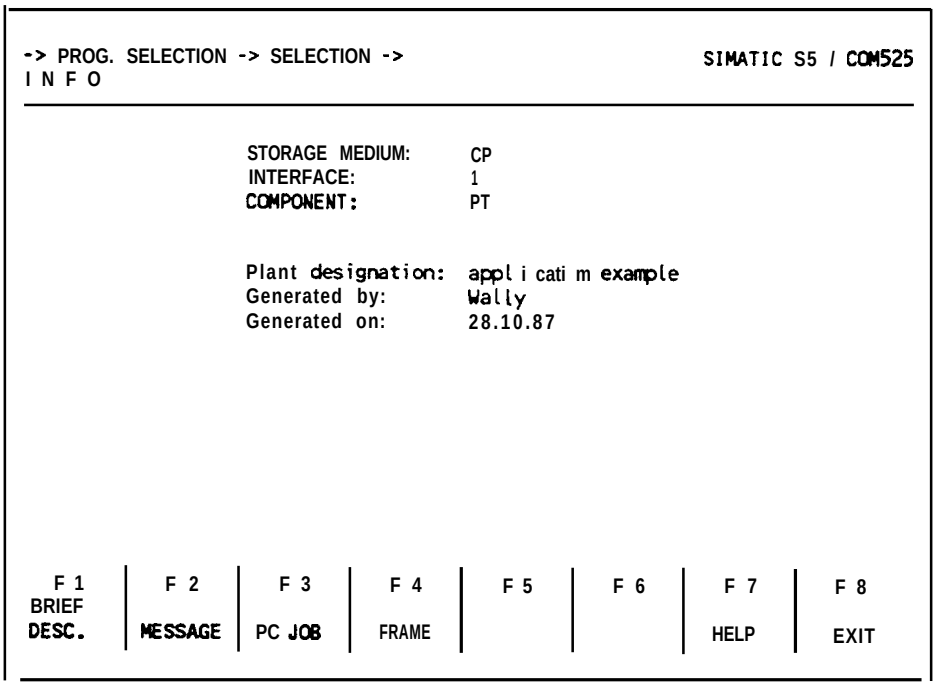
Plug in the printer cable again. The 'break bit in DL 30 is reset immediately. Reset the error message area of interface 1 in the SYSTAT, by switching input I 0.7 from zero to one.

12 Information

The COM 525 programming software displays information on the PC monitor about user data that is already programmed.

Press F4 in the **' SELECTION'** mask to call the 'INFO' mask. In this mask you can decide whether you wish to have information about user data on the **CP** or from a program file on **FD**.

Press **F1 (CP)** to display the data **programme**d **on** the **CP** for interface 1.



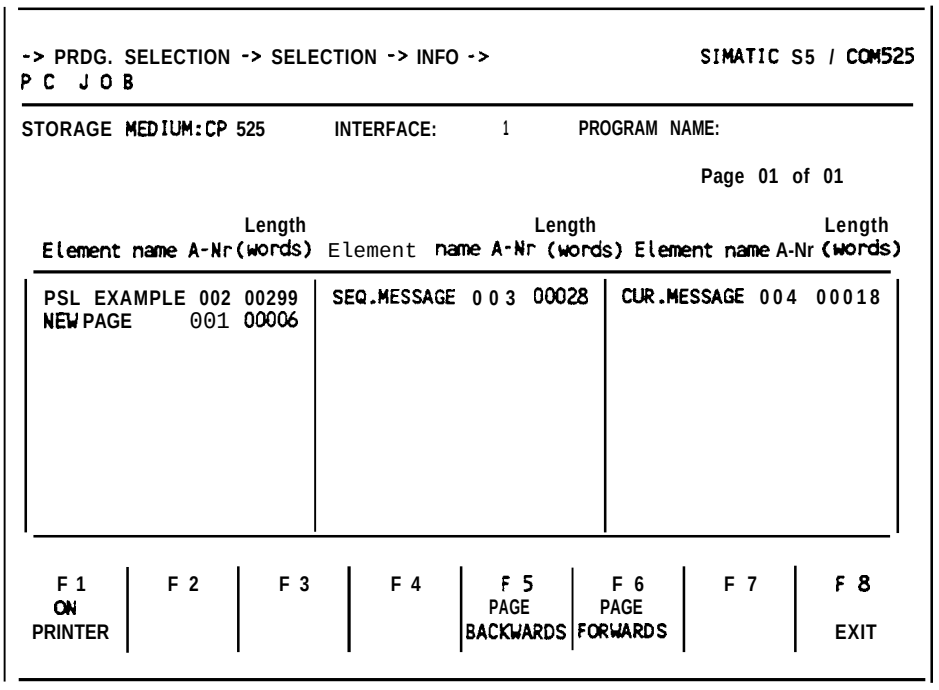
Press F1 to display a submask with a brief description of interface 1 on the CP 525.

```
-> PROG. SELECTION -> SELECTION ->                    SIMATIC S5 / COM525
I N F O
-------------------------------------------------------------------------

                    STORAGE MEDIUM:     CP
                    INTERFACE           1
                    COMPONENT           PT


                    Plant designation:  application example
                    Generated by:       Wally
                    Generated on:       28.10.87

 Data type    Element name Number            Data type    Number

 Interpreter: PT88 01        1              Message:        11
 Procedure:   LAUFPT88 01    1              PC job:          4
 Print para.: PRINT-PARA     1              Franc:           2

 Total number        20 elements
 Program length    9800 words

  F1       F2  |  F3  |  F4  |  F5  |  F6  |  F7  |  F8
BRIEF          |
DESC.  | MESSAGE  PC   J08   FRAME |      |      | HELP | EXIT
```

You can see the interpreter and procedure name in this mask and you can obtain a list of the existing elements for all types of user data. In addition, you see the total number of elements and the length of your program.

Call up detailed information about allthedatatypes.

Press F3 in the 'INFO' mask to display the names and job numbers for all PC jobs and the memory space required for each job in words.

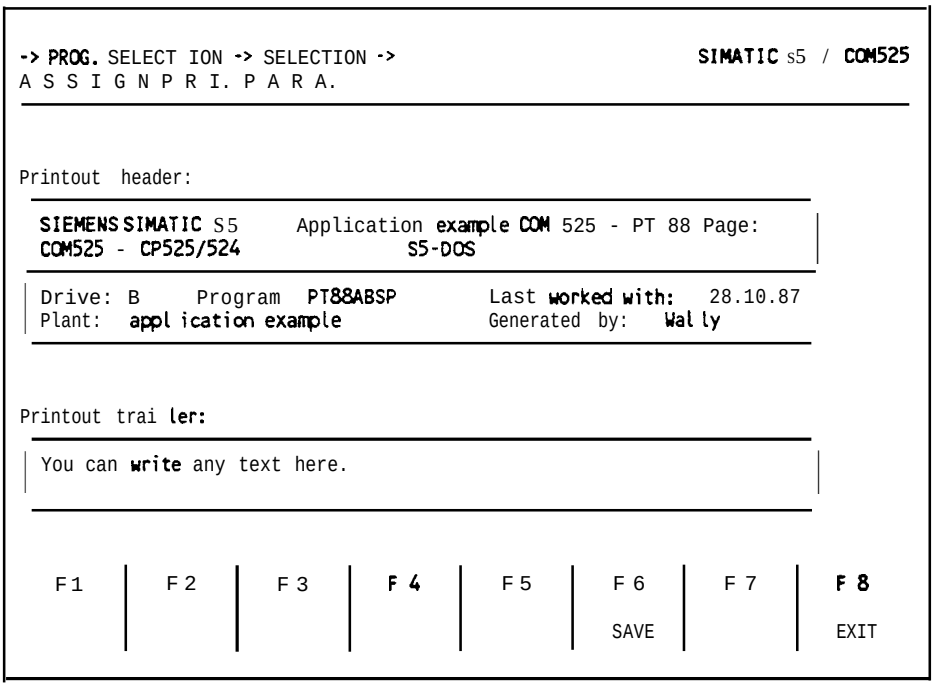


Press F1 (ON PRINTER) to have this information printed out on a PT88 printer connected to the PG.

13 Program Documentation

AU the elements of the COM 525 program file can be documented in detail. The printouts are output with a printout header and, if programmed, with a printout trailer at the bottom of the page. To achieve this, the settings of the mode switches on the interface adaptor of your printer must be the same as the settings on your PG (see PG manual). The settings on the coding switch of the central controller on the printer can be taken from Section 3.1

F6 (CONTINUE) and F3 (ASSIGN PRI. PARA.) in the 'SELECTION' mask call the mask below:



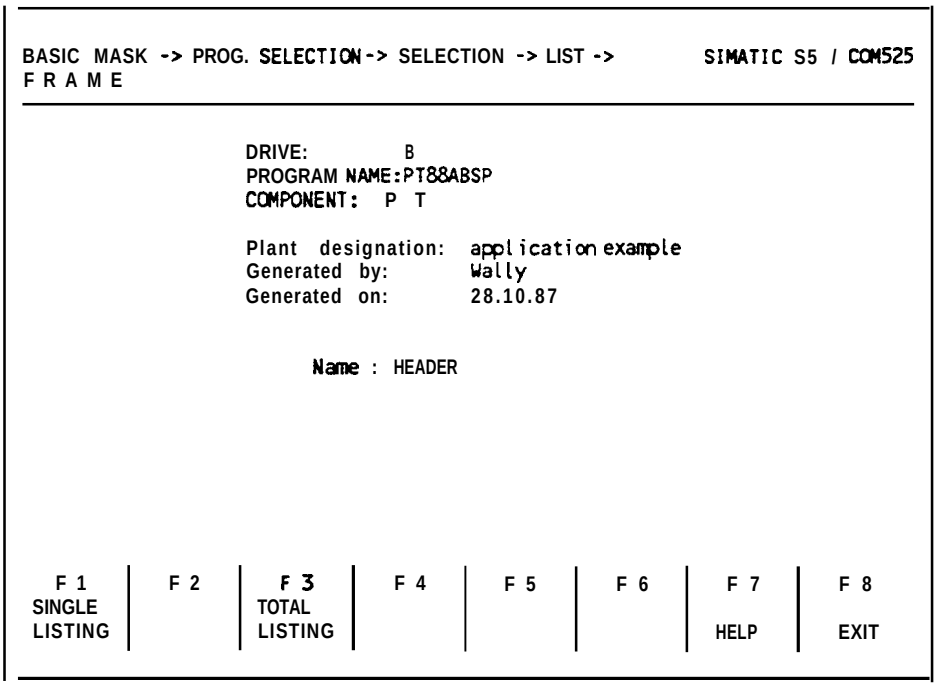
You can program two lines of text for the header and trailer. F6 saves the header and trailer in your user program file PT88ABSP. 525.

Press F6 (CONTINUE) and F4 (LIST) in the 'SELECTION' mask to call the 'LISTING' mask. The softkeys

- F1 (TOTAL PROGRAM)
- F2 (INTERPRT / PROCEDURE)
- F3 (MESSAGES)
- F4 (PC JOB)
- F5 (FRAME)
- F6 (SUMMARY LISTING)

can be used to list the whole program or parts of it.

Try out the various options. As an example the 'FRAME mask is shown below:



With F3 (TOTAL LISTING) all the frames sre listed on the printer. To list the single frame HEADER, enter the name **'HEADER'** in the mask with F7 (HELP) and press **F1 (SINGLE** LISTING).

The following listing will be printed out on the printer connected to the **PG** 685:

```
+----------------------------------------------------------------------------+
! SIEMENS SIMATIC : S5 Application example COM 525-PT88 Page  ,  1
! COM525 - CP525/524                   S5-DOS                 10. 30. S-
+----------------------------------------------------------------------------+
, Drive : 3          Program: PT88ASSP          Last worked with:  10. 30. 87
, Plant: application '2, ample                  Generated  by:  Wally
+----------------------------------------------------------------------------+

===========
F  R A M E S
===========

Frame                       Name:             HEADER
-----                       CPU no.:          1
                            DB no.:           030

Line:    01                     Group no.:  -        Attributes: B - - - -

-------------------------------------------------------------------------


Dynamic fields:none

Line:    02                     Group no.:  -        Attributes: B E - - -

Printout header              Page: ###

Dynamic fields: 01

Field  Variable type    Source  Address    Format/  No of   Attributes
                                           Order    chars.
-------------------------------------------------------------------------
01     Page  number     CP                          03      B E - -

Line:    03                     Group no.: -         Attributes: - - - - -


Dynamic fields: none

Line:    04                     Group no.: -         Attributes: - - - - -

Date:  ########        Time  :  ########

Dynamic fields: 02

Field  Variable type    Source  Address    Format/  No. of  Attributes
                                           Order    chars.
-------------------------------------------------------------------------
01     Date             CP   clk -         DMY      08      B - - -
02     Time             CP   clk -         HMS      08      B - - -

Line:    05                     Group no. :  -       Attributes:  -


Dynamic  fields:none

+------------------------------------------------------------------------+
! You can  write any  text here
+------------------------------------------------------------------------+
```

```
+--------------------------------------------------------------------------+
' SIEMENS SIMATIC S5  Aoolicati cati on a::amole COM 525 -PTS8   Page: 2  ,
: COMS25 - CP525/ 524              S5-DOS                        LO. 30. a: ,
+--------------------------------------------------------------------------+
3.,..0 : S         Program:PT88ABS P          Last worked with:  10 .30. &? ,
. Plan t: aoolication a::amole                Generated by ,    Wally
+--------------------------------------------------------------------------+

Line:   06                 Group no.:  -        Attributes: - - C U -

Static text

Dynamic fields: none

Line:   07                 Group no.:  -        Attributes: - - - - -

16-bit fixed point no.: ******

Dynamic fields: 01

Field  Variable type   Source  Address   Format/    No. of  Attributes
                                          Order      chars.
-------------------------------------------------------------------------
01     Process var .   DB      DW 007    16-BI T FXP 06      - - - -

Line  : 08                 Group no. : -        Attributes: B - - - -

--------------------------------------------------------------------------
--------

Dynamic fields: none


+--------------------------------------------------------------------------+
' You can wri te any text here                                             '
+--------------------------------------------------------------------------+
```

We have checked the contents of this manual for agreement with the hardware and software described. Since deviations cannot be precluded entirely, we can. not guarantee lull agreement. However, the data in this manual are reviewed regularly and any necessary corrections included in subsequent editions. Suggestions for improvement are welcomed.
Technical data subject to change.

Siemens Aktiengesellschaft

C79000-B8576-C542-05
Gerätewerk Karlsruhe
Printed in the Federal Republic of Germany

SIEMENS

SIMATIC S5

Notes on the Operating Systems PCP/M-86 and S5-DOS

User's Guide

Order No. C79000-B8576-C543-03

CONTENTS

1 What Does COM 525 Include?

1.1 COM 525 Parts

COM 525 includes the main COM program, five overlays, PROM 525 snd a standard library with interpreters and procedures. The following table shows the names of the overlays and the functions they contain.

Filename	Functions
S5PEC05X.CMD	Main program
S5OECP5X.CMD	Listing (of programs)
S5OECR5X.CMD	Computer link
S5OECD5X.CMD	Printer
S5OECM5X.CMD	Message editor
S5OECZ5X.CMD	List editor
S5PEPO5X.CMD	EPROM programmer PROM 525
COMLIBn.525	Standard library (interpreters, procedures)

1.2 Distribution on Floppy Disks

The distribution of COM 525 on floppy disks depends on the programmer.

There are three floppy disks tf you are using any of the programmers PG 675, PG 685 and PG 695. The COM 525 files are distributed on these disks as follows:

Disk 1 (1/3)

- S5PEC05X.CMD COM 525 main program
- S5PEP05X.CMD EPROM programer PROM 525
- S5OECR5X.CMD overlay **computer* link'
- S5OECP5X.CMD *overlay 'listing'*

Disk 2 (2/3)

- S5OECD5X.CMD overlay 'printer?
- S5OECM5X.CMD overlay 'message editor'
- S5OECZ5X.CMD overlay 'list editor'

Disk 3 (3/3)

- COMLIBn.525 (n = release) standard library

For programer PG 635, COM 525 is distributed cm two disks as follows:

Disk 1 (1/2)

- S5PEC05X.CMD	COM 525 main program
- S5PEP05X. CMD	EPROM programmer PROM 525
- S5OECR5X.CMD	overlay 'computer link'
- S5OECP5X.CMD	overlay 'listing'
- S5OECD5X.CMD	overlay 'message printer'
- S5OECM5X.CMD	overlay 'message editor'
- S5OECZ5X.CMD	overlay 'list editor'

Disk 2 (2/2)

- COMLIBn.525 (n= release)	standard library

2 Making Back-up Copies

When you are using a programmer without a hard disk drive (PG 635 and PG 675) you should make back-up copies of the original diskettes as soon as you receive them. To do this, you must first format an empty floppy disk by calling the DSKMAINT program.

Press function key F7 to select the formatting function; with F1 or F3 you can select the drive in which the floppy disk to be formatted has been inserted. You are prompted to confirm your intention.

The copying function is called with function key F3. Again F1 and F3 are used to specify the source and destination drive. When you are sure that you are going to copy in the correct direction, confirm your intention.

3 COM 525 Files

COM 525 initializes one file for each generated program. The name of this file is the program name you selected and the file is automatically given the file type '.525'.

Example: program name: TURBINE
file name: TURBINE.525

Files with the name COMLIB??.525 are recognized as libraries by COM 525. "?" stands for any alphanumeric character. The term "library" is explained in the section "Libraries" in the COM 525 instructions in this manual.

4 Working with Files under PCP/M-86

The following sections briefly explain the PCP/M-86 functions, directory, erase and copy. These functions are also made available by COM 525; however, they are then restricted to COM 525 programs. To use these PCP/M-86 functions fully you should have some experience of this operating system.

4.1 Directory (DIR)

Using the DIR function, you can have all files orgroupsoffiles listed that have the DIR attribute. When specifying the files or groups of files, you can replace a single unknown character by ? and a group of unknown characters by * (? and * are known as wild cards).

Syntax: **DIR d:filename[Options]** d: drive

Example:	**DIR C:**	*all* files on drive C:
	DIR *.525	all COM 525 files on the current drive
	DIR S5?EC?5X.CMD	COM 525 main program and all overlays

Files with the SYS attribute can be listed if you use the function DIRSYS. You can also use wild cards with this function.

Example:	**DIRSYS *.CMD**	all *system* files of the file type CMD
	DIRSYS	all system files on the current drive

The permissible options and further examples can be found in the PCP/M-86 description.

4.2 Erase (ERA)

The ERA function allows you to erase single files or groups of files. You can also use wild cards with this function.

syntax:	ERA d:filename[options]	d: drive
Example:	ERA PLANT.525	deletes the COM 525 file PLANT
	ERA B:*.525	deletes all COM 525 files on drive B:
	~ * . *	after an acknowledgement is prompted all files on the current drive and current user area are erased
	ERA *.*[CONFIRM]	this option means that every erase job is comfirmed

Further options can be found in the PCP/M-86 description.

4.3 Copy (PIP)

The PIP function allows you to copy single files or groups of files. You also possible to rename individual files when they are copied.

The following options are available:

- Single file without renaming

  PIP d: = s:sourcefilename — d: dest. drive, s: source drive

- Single file with renaming

  PIP d:destfilename = s:sourcefilename

- Single file with renaming and specification of the USER area

  PIP d:destfilename[Gn] = s:sourcefielname[Gn]

  n: number of the user area of the destination or source

- Single system file with renaming

  PIP d:destfile = s:sourcefile[R]

- Group of files (the group is selected using wild cards)

  PIP d: = S: filegroup

  Example: PIP C: = A:*.525 all COM 525 programs on the current user area are copied from drive A to C

- Group of files with system files and specification of the user areas

  PIP d:[Gn] = s:filegroup[Gn R]

For further information, particularly about other options, refer to the PCP/M-86 description.

5 Loading COM 525 with the PG 635 and PG 675

5.1 Loading PCP/M-86

After powering up, the programmer requests the operating system disk to be inserted in drive A:.

If PCP/M-86 has never been loaded in your PG, insert the PG-TEST disk in drive A:. You are, among *other* things, prompted to specify the memory capacity of your programmer by entering plus and minus characters. When this is completed, insert the operating system disk in drive A: and then switch the device off-and on. The operating system is then loaded.

5.2 Calling COM 525

Wait until the operating system has been completely loaded. Then remove the PCP/M-86 disk and insert the following disks:

with PG 635

in drive A: disk 2/2 of the S5-DOS
in drive B: disk 1/2 of COM 525

with PG 675

in drive A: disk 2/3 of S5-DOS
in drive B: disk 1/3 of COM 525.

Enter

S5

to load the S5-DOS operating system. The S5-DOS command interpreter (S5-KOMI) then becomes active. S5-KOMI allows you to select the available packages. You can now call COM 525 by positioning the cursor in front of the COM 525 package and pressing the function key F1 (PACKAGE) . If you wish to program an EPROM, position the cursor in front of the PROM 525 package and press F1.

Once COM 525 has been called, you can replace the S5-DOS disk in driveA: with the disk on which to generate your COM 525 programs.

6 Loading COM 525 with the PG 685 and PG 695

6.1 Loading PCP/M-86

Before you load COM, the PCP/M-86 and S5-DOS operating systems on the hard disk should be in user area 0. The files for this operating system must have the attribute SYS (system). Check this with the command

DIR d:[FULL]	d: hard disk drive (with PG 685 always B:; with PG 695, usually C:)

After powering up, ignore the PG request to insert the operating system disk in drive A: and wait approximately 10 seconds. After this time the PG switches over to the hard disk drive and loads the operating system.

If PCP/M-86 has never been loaded on your PG, then follow the instructions in Section 5.1. After switching on the programmer, transfer the operating system to user area 0 on your hard disk. Enter the command:

PIP d:[G0] = A:*.*[G0 R V]	d: hard disk drive

6.2 Using S5-DOS for the First Time

If the S5-DOS operating system is not yet cm user area 0 of your hard disk, transfer it to this area. Insert disk 2/2 of S5-DOS in drive A: and enter the following command:

PIP d:[G0] = A:*.*[G0 R V]	d: hard disk drive

The files located on disk 1/2 of S5-DOS are not required for working with COM 525. A detailed descripticm can be found in the instructions for S5-DOS.

6.3 Setting up a Working Area

To generate programs with COM 525, select an area other than user area 0 (if possible select a user area that is empty or only contains COM 525 programs). Insert disk 1/3 in drive A: and copy the contents to the user area you selected on your hard disk.

PIP d: [Gn] = A:*.*[G0 R V]	d: hard disk drive n: selected user area

Copy the contents of disks 2/3 and 3/3 in the same way.

6.4 Calling COM 525

Enter

d:	d: hard disk drive

to select the hard disk drive and follow this with

USER n	n: user area on which COM 525 is located

to set the required user area. Load the operating system S5-DOS, by entering

S5

The S5-DOS command interpreter (S5-KOMI) is now called and you can select one of the packages available on this user area. You select a package by placing the cursor in front of the required package and pressing function key F1 (PACKAGE).

6.5 COM 525 on Several User Areas

If, to simplify matters, you want to work with COM 525 on several user areas, a different procedure is recommended.

In this case, copy disks 1/3, 2/3 and 3/3 of COM 525 to user area 0 with

```
PIP d:[G0] = A:*.*[G0 R V]
```

d: hard disk drive

and assign the system attribute to the files with

```
SET d:S5?EC?5X.CMD[SYS RO]
SET d:S5PEP05X.CMD[SYS RO]
SEX d:COMLIB*.525[SYS RO]
```

d: hard disk drive

You can now work with COM 525 on every user area. Area 0 should, however, be reserved for system files.

SIEMENS

We have checked the contents of this manual for agreement with the hardware and software described. Since deviations cannot be precluded entirely, we cannot guarantee full agreement. However, the data in this manual are reviewed regularly and any necessary corrections included in subsequent editions. Suggestions for improvement are welcomed.
Technical data subject to change.

Siemens Aktiengesellschaft

C79000-B8576-C543-03
Gerätewerk Karlsruhe
Printed in the Federal Republic of Germany

SIEMENS

SIMATIC S5

Programming Package COM 525

User's Guide C79000-B6576-C544-07

Contents Page

1 The Programming Package COM 525

1.1 Introduction

The software package COM 525 supports the programming of the CP 525 and CP 524 communications processors,

When connected to the PT88/PT89 printer, the CP 525 can list process statuses and messages originating in the process.

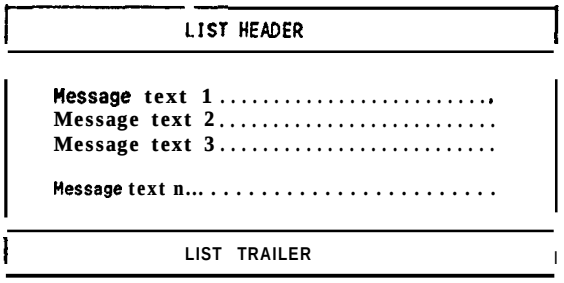
The messages acquired by the CPU of the programmable controller are transferred to the CP 525 in a data block (DB); the CP 525 evaluates the DB and initiates the printing out of the messages on the printer.

The message text includes static texts and dynamic parts such as the date, time, message status (coming, going, acknowledged) and a process variable,

To highlight parts of the message texts, you can select attributes such as bold print, expanded print, compressed print or underlining. The messages can also be accompanied by an acoustic signal on the printer.

The static texts programmed with COM 525 are stored in the user memory of the CP 525.

List printout:



Messages can be grouped together; for example, you could group together messages for a particular part of the plant. Each of the maximum 16 message groups can be inhibited or scanned.

All functions to be executed by the CP 525 (e.g. output of messages on the printer) can be programmed when you complete interactive masks on the PG 685 (also the PG 635 *or* a PG 7xx).

More information on the types of lists and functions can be found in the user's guide "Event output end listing with the PT88/PT89 printer" in this manual.

The computer link with RK 512 allows the data exchange between two programmable controllers or between a programmable *controller* and a central process computer (see user's guide "Computer link with RK 512" in this manual). Using the COM 525 software package you can generate the required CP 525 user program.

If you are not familiar with the terms or abbreviations used, refer to Section 6, "Terms and definitions" in this user guide.

The operating system expansion S5-DOS allows COM 525 to be read on several programmers without being adapted. Before using COM 525 you should get used to working with the S5-DOS command interpreter (KOMI). You require this function to be able to work with a STEP 5 package, Read through the section "Getting started" in the manual for your PO.

If you have not worked with PCP/M, CP/M or a similar operating system, refer to the description of the operating system to get at least a working knowledge of the following utility programs:

- DSKMAINT (formatting and copying disks)
- PIP (copying single files)
- DIR (displaying the file directory)
- SET (setting file attributes)
- ERASE (erasing files)
- USER (changing the user area number)

1.2 The Layout of the Masks with COM 525

To work with COM **525 you** do not need programming experience. All inputs are prompted by COM 525 in plain language in the masks; if you make errors a text is displayed to indicate the nature of the **errors. The 'PROGRAM JOB'** mask, for example, appears as below:

-> SELECTION -> JOB BLOCK -> SIMATIC S5 / COM525
P R O G R A M J O B

DRIVE: $ PROGRAM: $$$$$$$$ COMPONENT: SS

J O B

Job no.: $ss

Job: ##.##

Job type ################

CPu 170.: #

DB 110.: ###

Dest. - word address: ##### D ###H

If required with coordination flag: ###.#

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
ON PRINTER	PAGE BACKWARDS	PAGE FORWARD:		DELETE JOB	ENTER JOB	HELP	EXIT

AU masks have basically the same layout. In the top left corner the "route" to the currently displayed mask is displayed. In the line below this (in spaced letters) is the name of the current mask, for example:

-> SELECTION -> JOB BLOCK ->
P R O G R A M J O B

You can press the function key F8 (EXIT) to follow the same route back; i.e. you return to the last mask displayed (this is shown farthest right in the first line).

In the right-hand top corner, you see the name of the programming package

SIMATIC S5 / COM525

In the lower part of the mask there is a softkey menu. This corresponds to the function keys F1 to F8 on your keyboard. The labelling of the keys on the screen corresponds to the current function of the individual keys, Using the function keys, you can call a follow-on mask or second set of softkey functions or you can execute the function currently assigned to the key.

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
ON PRINTER	PAGE BACKWARDS	PAGE FORWARDS		DELETE JOB	ENTER JOB	HELP	EXIT

The keys F6, F7 and F8 have special functions that remain the same in all the masks as follows:

- F6 allows an alternative set of function keys to displayed if there are more than six functions available in one mask. This is displayed on the softkey by the word 'CONTINUE'. Press this key to obtain the second set of softkey functions. Press it again to return to the first set.
  F6 is also reserved for storing your input; e.g. 'SAVE' or in the example 'ENTER JOB'. If 'ENTER' is displayed, your input in the mask is transferred to the RAM of the programmer; and if 'SAVE' is displayed, to the storage medium (floppy or hard disk).

- F7 (HELP) allows you to make entries in certain input fields; (in the currently displayed mask you can enter either 'SEND' or 'FETCH' in the "Job" field with F7).

• **F8**(EXIT) allows you return to the previous ma sk.

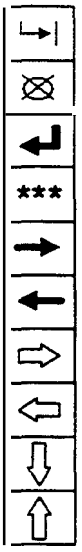
Messages, error displays and acknowledgement prompts are displayed in the line above the function key menu. If you press key F4 with the mask in the example, the error message MF.001 'illegal key' is displayed since this key has no function in this menu. The section "Messages" in this manual provides a list of all possible messages and an explanation of their causes.

The central section of the mask consists of input fields, explanatory texts and output fields.

Input fields are always displayed on the screen of the PG inversely, in the manual they are represented as "########". You must complete these.

Output fields contain specifications which COM 525 has taken over from previously completed masks. These are always represented as "$$$$$$$$" in the manual.

The cursor keys and the RETURN key can be used in all the masks for moving the cursor. They have the following significance:



INSERTCHARACTER

DELETE CHARACTER

(RETURN) completes input and has the same effect as FIELD REGISTER

RUB-OUT

CHARACTERRIGHT

CHARACTER LEFT

FIELDRIGHT

FIELDLEFT

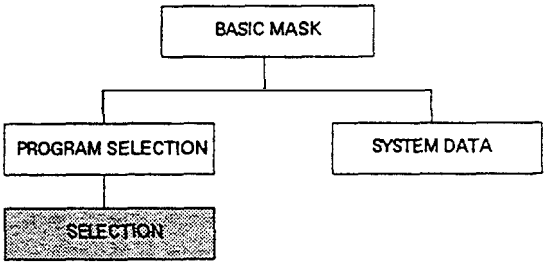
FIELD DOWN

FIELDUP

With all other keys the input depends on which characters are permissible for the particular field (no special characters, only numbers etc.). In file and element names, lower case characters are automatically converted to upper case. The exact specifications for each field can be found in the description of the particular mask (Sections 2 to 5 in this guide).

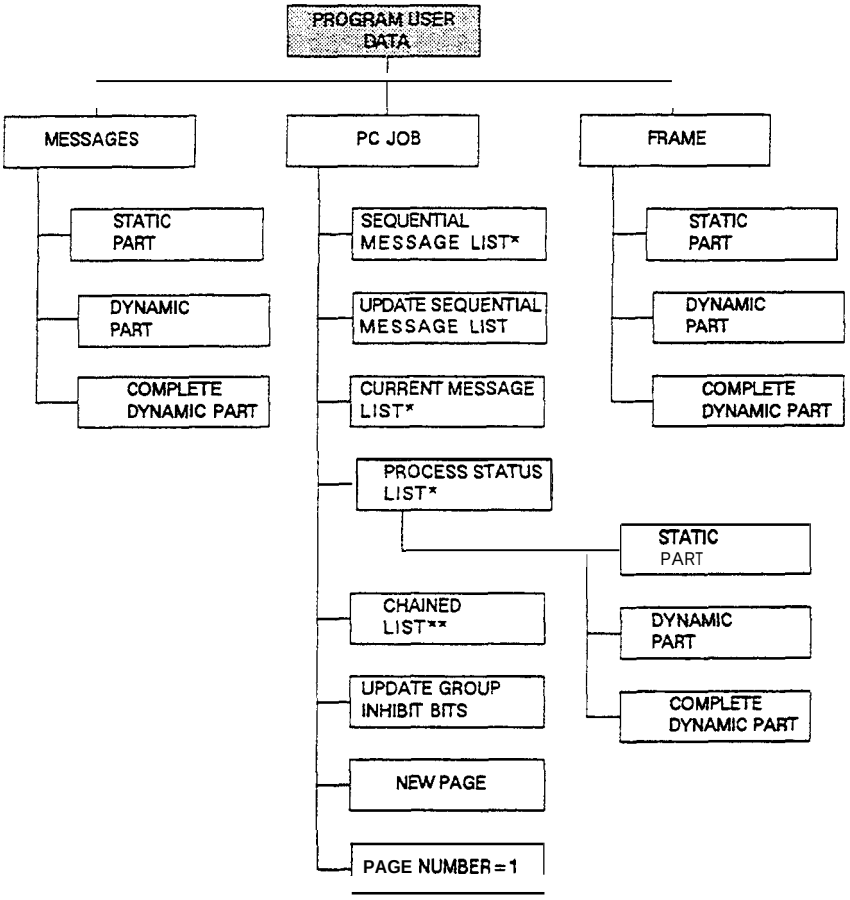
1.3 Overview of the Masks

The following overviews show the masks of COM 525 and the route you must follow to reach a particular mask. Please note that not all the functions, but only the COM 525 masks are listed. The route for the components printer (pT), computer link (CL) and for libraries (see 1.6) is identical, i.e. via the basic mask.

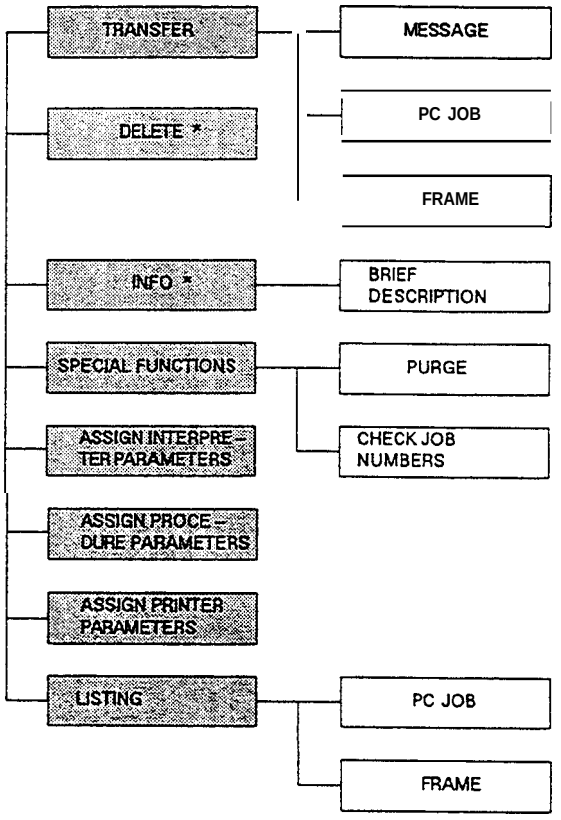


The program branches to different masks from the SELECTION mask depending cm which component, "PT" or "CL" or a library (see Section 1.7) is specified. The shaded fields in the following overviews indicate the masks which can be branched to from the SELECTION mask.

Component, printer

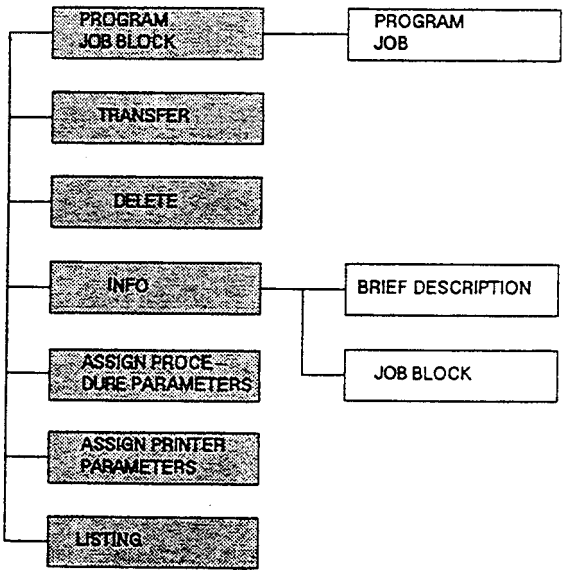


*From these masks it is possible to branch to the FRAME mask
** From this mask it is possible to branch to the PROCESS SATUS LIST mask



* From these mask it is possible to branch to the MESSAGES, PC JOB and FRAME masks just as from the TRANSFER mask

Component, computer link



1.4 File Organization

Most modern programmers are equipped with hard disk drives containing a capacity of 10 to 40 Mbytes, COM 525 uses a file format which is adapted to these capacities.

You can divide the hard (or floppy) disk into 16 "USER areas" by means of **USER ?** ("?" stands for a number between 0 and 15).

This allows a more orderly storage of files.

You should reserve user 0 for the following:

- the operating system
- the operating system utility programs
- the COM and STEP 5 packages
  (with the PG 685 the STEP 5 software is already loaded)

you assign system attributes to these files using the command SET *.*[SYS]. Then you can start all programs from any **USER** area.

When you generate programs with COM 525 the program name you select is automatically given the file type '.5251. This type allows you to recognize your COM 525 programs when you list the files using DIR.

1.5 Structure of a COM 525 Program

A program generated with COM 525 is structured hierarchically. It includes data types that are then subdivided into elements. A program always contains the following data types: an interpreter, a procedure and depending on the component, either user data (printer) or a job block (computer link).

Printer	Computer link
Interpreter	Interpreter
Procedure	Procedure
Printer parameter assignment	Printer parameter assignment
User data messages PC jobs sequential message list update sequential message list current message list process status list chained list update group inhibit bits new page page number = 1 Frames	Job block

The data type PRINT_PARA, i.e. the parameter assignment for the printer (specifying the labelling of the printout header or trailer for lists and listings with COM 525) can, but does not have to be part of a program.

1.6 Libraries

Libraries are files in which parts of programs are collected and from where they can be transferred to the individual programs. They are not assigned to any particular interface.

You should initialize a library to store e.g invariable parts of the program, which are required regularly, and which can then be transferred to the individual programs when required. This means that interpreters, procedures, printer parameter assignments, messages, PC jobs (same PC job several times), frames and job blocks can be stored in a library.

You can rename data types when they are transferred from a program to a library (and vice-versa) ; this does not, however, apply to interpreters and procedures named by SIEMENS. These cannot be changed.

The data type, printer parameter assignment, always has the name PRINT-PARA in a program. When you transfer it to a library it can, however, be changed. When this data type is transferred back to the program, COM 525 automatically changes it back to PRINT-PARA.

If you wish to establish a library you must give it the name COMLIB?? ("?" stands for any alphanumeric character). COM 525 then recognizes that this is a library.

For processing libraries, two “fail-safe” procedures have been included as follows:

- when you specify a library name in the ‘PROGRAM SELECTION’ mask, COM 525 outputs the prompt “process library?". If you acknowledge positively, COM 525 then makes available masks and functions which are to some extent different from those for programs (see overview of masks at the end of this section. Remember that only the masks and not all the functions are listed.)

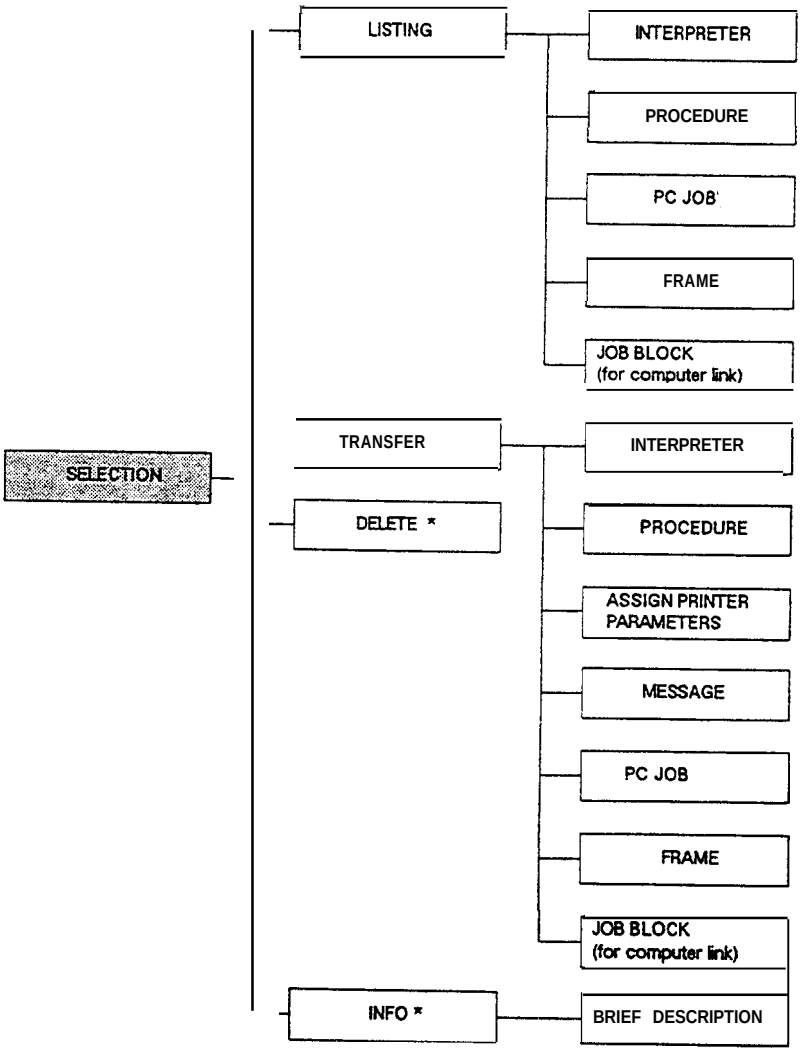
a library cannot be 'purged'. Purging is the deleting of parts of a user program which are no longer required. From the structure of a program it is possible to determine whether, for example, frames are present which are not used with any list. Since a library rarely contains a consistent program the whole library would be deleted if it were purged.

You should store libraries on user 0 and set them to [SYS], to allow them to be accessed from every user area.

Also, you should assign the IRO] attribute (read only) to libraries to avoid them being changed accidentally.

The COM 525 programming package already contains a standard library with the name COMLIB02. This contains the interpreters and procedures you require togenerate a program.

Overview of masks, libraries



* From these masks it is possible to branch to the same masks as from TRANSFER

1.7 Handling User Data

After you generate user data, you want to be able to see it and change it if necessary. COM 525 makes available the following functions in the interactive masks:

- deleting single elements
- transferring elements or whole data types between user programs and/or libraries
- transferring the whole program or parts of it from disk to disk or from disk to CP and vice-versa
- information about programs on disk or CP
- listing programs or parts of programs
- purging (logical) - unnecessary elements can be deleted
- condensing (physical) - necessary because of the file organization and is carried out (particularly in conjunction with the delete function) without the user having to take any action
- programing (if you wish to change user data)

The corresponding masks and programming options canbe found in Sections 2 to 5 in this guide.

1.8 Standard Outputs when Working with COM 525

overlays

After you select the function PROGRAM or LISTING in the 'SELECTION' mask the message 'Loading overlays!' is displayed. Certain COM 525 overlays must be loaded to allow COM 525 to run on programmers that have a memory capacity of only 384 Kbytes. If you select a function that is one of the overlays and has already been loaded, it will not be loaded again.

COM 525 includes a main program and 5 overlays:

S5PEC05X.CMD	main program
S50ECD5X.CMD	programming frames and PC jobs
S50ECM5X.CMD	programming messages
S50ECZ5X.CMD	programing the static parts of PSLs and frames
S50ECR5X.CMD	programming computer link (job block)
S50ECP5X.CMD	listing functions

Since loading overlays takes time, the functions have been distributed so that the loading of overlays is restricted as much as possible. On devices that have no hard disk, you may need to change the floppy disk to load an overlay. If an incorrect floppy disk is inserted or if an incorrect COM 525 version (e.g. German) is on the floppy disk, then an error message is output.

If you intend to use COM 525 very often, you should use a hard disk (PG 685 or PG 695).

'ACTIVE' message

Whenever you process user data on a floppy disk with COM 525 or when data is exchanged in any way with the CP 525, the 'ACTIVE' message is displayed.

Data transferred from and to the CP 525

If the data transfer between the PG and CP is not functioning correctly and one of the actions affecting the CP is triggered (TRANSFER, DELETE or INFO), an associated error message is displayed.

In this case check the cable connection and that the CP 525 is switched on.

2 Selecting the User Program

The masks described in this section (except SYSTEM info) lead to the 'SELECTION' mask, From this mask you can select all the other functions.

2.1 General Information

File names (called program names in the masks)

File names must correspond to the PCP/M conventions for file names.

Lower case letters, the characters < > = , ! * ? / $ ( ) . : ; and spaces and square brackets must not be used in file names.

Letters separated by a space in a file name are automatically moved together.

Names of data types

Messages, PC jobs, frames etc. are known as data types.

Within this name only single spaces are allowed. If several spaces appear between characters they are automatically reduced to one space.

Messages do not use a tame but rather a 4-digit message number (including initial zeros).

Note

If only upper case letters are permitted, then lower case letters that are input are automatically converted.

If there is no other specification, then:

- **F7 (HELP) -** you call the help mode to allow various entries to be made. In the tables describing the fields in the individual masks, the fields marked with '*' can have an entry made using F7 (HELP)

- **F8 (EXIT) -** you return to the previous mask. This function key can be used (for example when transferring or deleting several elements) as an abort key.

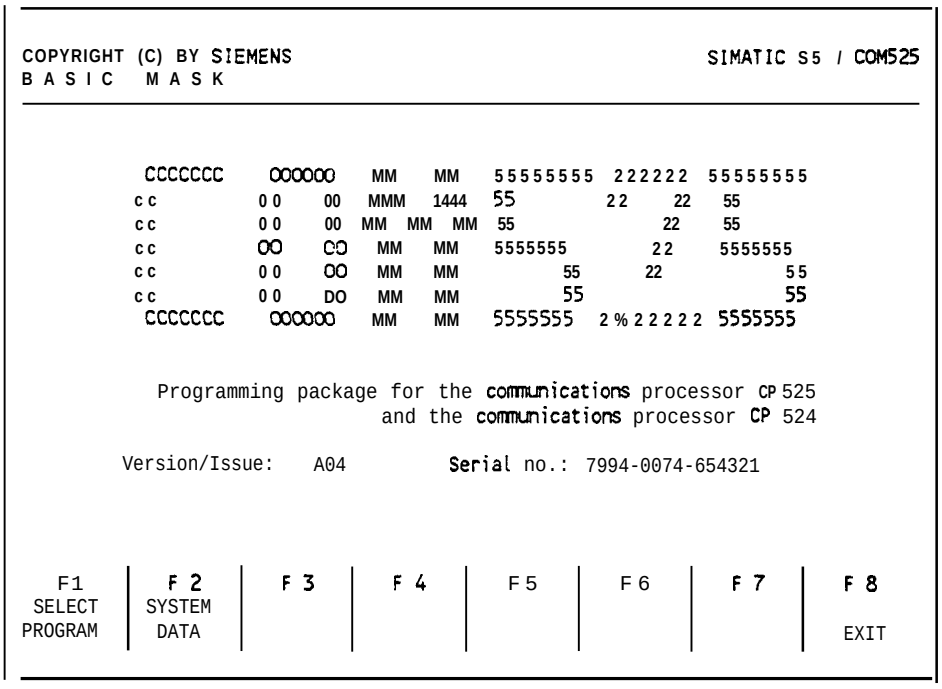
The '-> acknowledgement' prompt means that the following acknowledgement menu will be displayed:

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
YES		NO					

When this menu is displayed you must answer a prompt in the message line. Generally the answer 'YES' means-an action is to be executed, while 'NO' simply returns you to the previous menu,

2.2 Basic Mask

Call the 'BASIC MASK' with F1 (PACKAGE) or the enter key in the S5 command interpreter



F1: change to the 'program selection' mode;
branch to the 'PROGRAM SELECTION' mask (see 2.4)

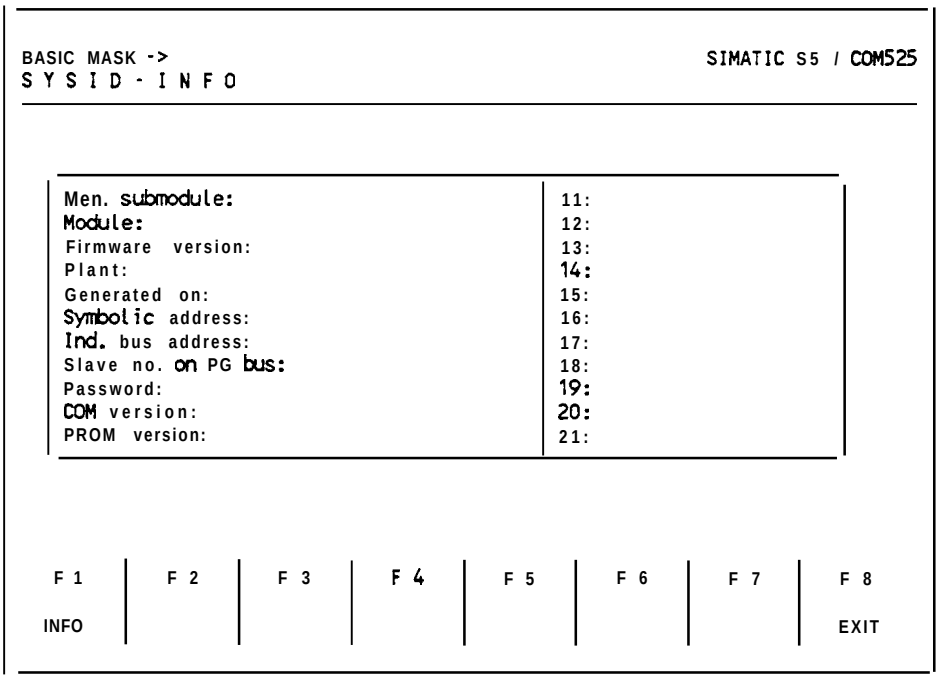
F2: change to the 'SYSID info' mode;
read SYSID from the CP 525 (see 2.3);
the CP must be connected

F8: return to the S5 command interpreter

2.3 SYSID Info

The SYSID contains the total system data required to identify a module and a program.

Call the 'SYSID INFO' mask with F2 (SYSTEM DATA) in the 'BASIC MASK'.



F1: SYSIDinfo isoutputagain.

For the CP 525, the following fields in the lower part of the mask are empty:

- Symbolic address
- Ind. bus address
- Slave no. on PG bus
- Password
- COM version

If present, the following data are output in the numbered fields on the right-hand side of the mask:

Device interface 1

Field U: component
Field 12: name of the interpreter
Field 13: version of the interpreter
Field 14: name oftheprocedure
Field 15: version of the procedure

Device interface 2

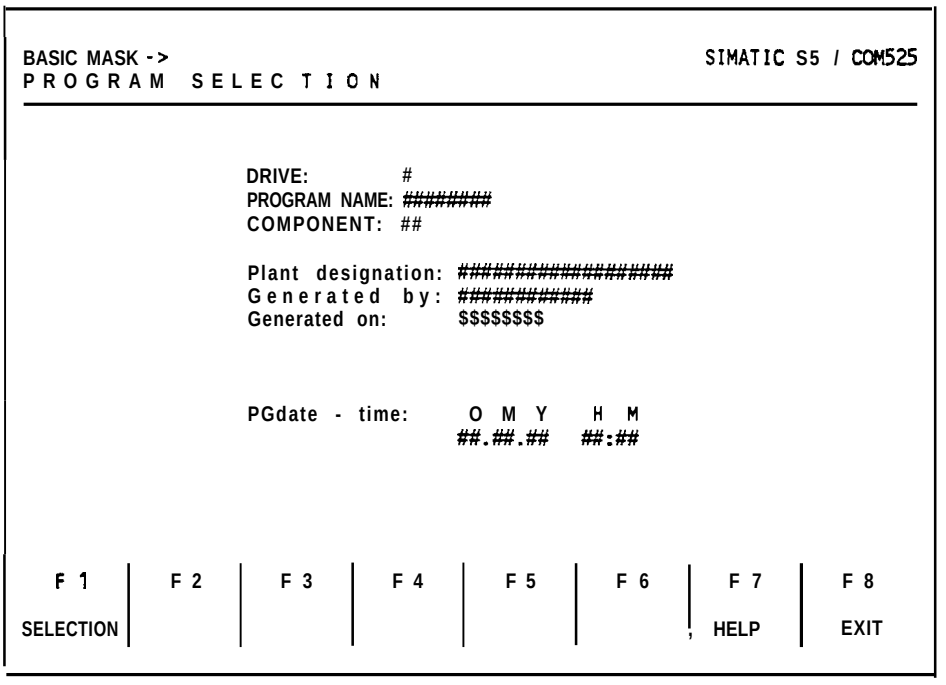
Field 16: component
Field 17: name of the interpreter
Field 18: version of the interpreter
Field 19: name of the procedure
Field 20: version of the procedure

2.4 Program Selection

All the data entered here defines a program and is part of the program (except the time).

If the program already exists, the program name can be entered with F7; COM 525 then outputs the component, plant designation, who generated it and when.

Call the 'PROGRAM **SELECTION'** mask with F1 **(SELECT** PROGRAM) **in** the **'BASIC MASK'.**



If a library name is specified in the PROGRAM NAME field then an '-> acknowledgement prompt' appears.

When generating anewprogramyoumust specify in the COMPONENT field whether a printer program (PT) or a computer link program (CL) is to be generated. If the program already exists the field will be written by COM 525.

F1: branch to the 'SELECTION' mask (see 2.5).

Field	Field type keys allowed	Limit value	Alternatives
DRIVE	upper case letters		A,B,C,D,E,F, . . dependent on device type *
PROGRAM	file name (see 3.1)		*
COMPONENT	upper case letters		PT,CL *
Plant designation	any (must b; entered)		
Generated by	any (must be entered)		
Generated on	OUTPUT FIELD		
PG date D	numbers	1 - 31	
M	numbers	1 - 12	
Y	numbers		
PG time H	numbers	0 - 23	
M	numbers	0 - 59	

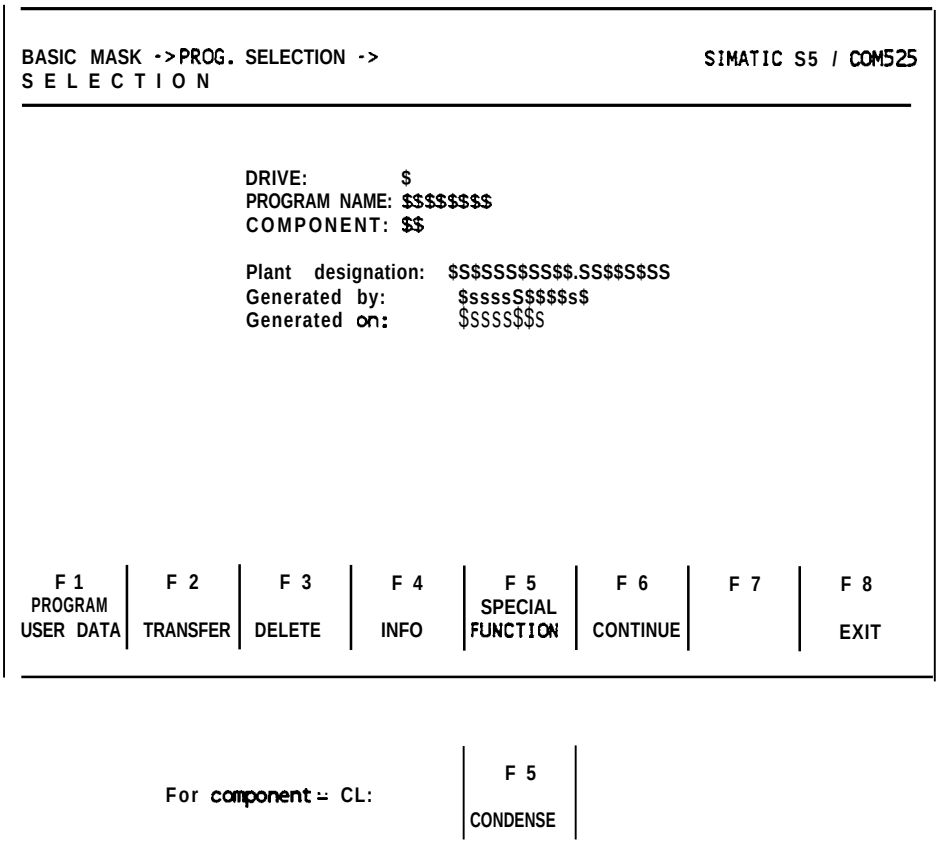
F7 (HELP) can be used to make an entry in the fields marked '*'.

2.5 Selection

In this mask you can select the function with which a program is to be processed. There are some hierarchical menus (TRANSFER, DELETE, **INFO);** i.e. after you press the corresponding function key the mask remains the same but a second menu is output.

The fields **DRIVE, PROGRAM,** COMPONENT, plant designation, generated by, generated on are output fields and are taken from the **'PROGRAM SELECTION'** mask.

Call the **'SELECTION'** mask with F1 **(SELECTION) in** the 'PROGRAM SELECTION' mask,



F1: load the specific overlay and
branch to the 'PROGRAM' mask (PT, see 5.2)
or 'JOB BLOCK' mask (RK, see 4.1)

F2: branch to the 'TRANSFER' mask (see 3.2)

F3: branch to the 'DELETE' mask (see 3.3)

F4: branch to the 'INFO' mask (see 3.4)

F5: branch to the 'SPECIAL FUNCTIONS' mask (see 3.5)

Press F6 to obtain the following menu:

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
ASSIGN INT. PARA	ASSIGN PROC.PARA	ASSIGN PRI.PARA	LISTING		CONTINUE		EXIT

El: branch to the 'ASSIGN INT. PARA' mask
(only with PT component, see3.6)

F2: branch to the 'ASSIGN PROC. PARA' mask
(see 3.7)

F3: branch to the 'ASSIGN PRI. PARA' mask
(see 3.8)

F4: branch to the 'LISTING' mask
(see 3.9)

Press F6 to obtain the original menu again.

With a library, the following menu is output in the 'SELECTION' mask:

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
LISTING	TRANSFER	DELETE	INFO	CONDENSE			EXIT

F1: branch to the 'LISTING' mask
(see 3.9)

F2: branch to the 'TRANSFER' mask
(see 3.2)

F3: branch to the 'DELETE' mask
(see 3.3)

F4: branch to the 'INFO' mask
(see 3.4)

F5: physical condensing of the library
(gaps in the files are closed)

3 Functions in the Main COM Section

3.1 General Information

This section provides an overview of the standard functions of COM 525 (transfer, delete, info, special functions, interpreter, assign parameters to the procedure and printer and list).

The '-> acknowledgement prompt' means that COM 525 displays a prompt and the acknowledgement menu as follows:

a) with TRANSFER, the destination element exists with the same name but with different contents. The following acknowledgement prompt is displayed:

   ACK.010: Element already exists! - overwrite?

b) for the TRANSFERof a whole program, the prompt appears:

   ACK.023: Transfer all elements?

c) to DELETE a whole program or an interface, or to delete all the elements of a data type you must answer the following prompt:

   ACK.023: Delete all elements?

The prompts are acknowledged in the following menu:

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
YES		NO					

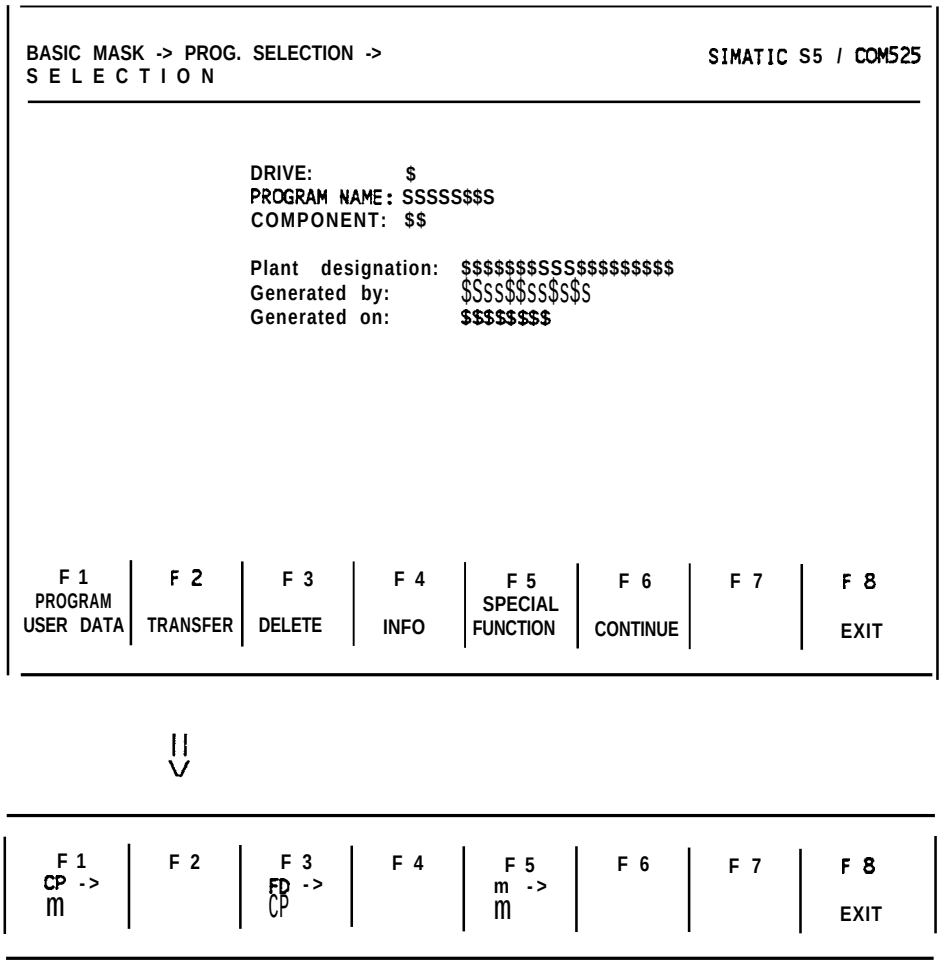
Press F1 (YES) to execute the action described; press F3 (NO) to return to the original menu.

The fields marked with '*' in the masks in this manual can have entries made using F7 (HELP).

3.2 Transfer

The transfer function allows the transfer of programs or parts of programs both on the hard disk and floppy disk as well as between the hard disk (or floppy) and the CP (in both directions).

The 'TRANSFER' mask is called with F2 (TRANSFER) in the 'SELECTION' mask.



The 'working program' is the program specified in the 'SELECTION' mask .

This is the SOURCE for transferring FD->CP, but
DESTINATION for transferring CP->FD and
FD->FD
(FD is used for both floppy and hard disk).

The working program can be changed only in the 'SELECTION' mask, not within the 'transfer' function.

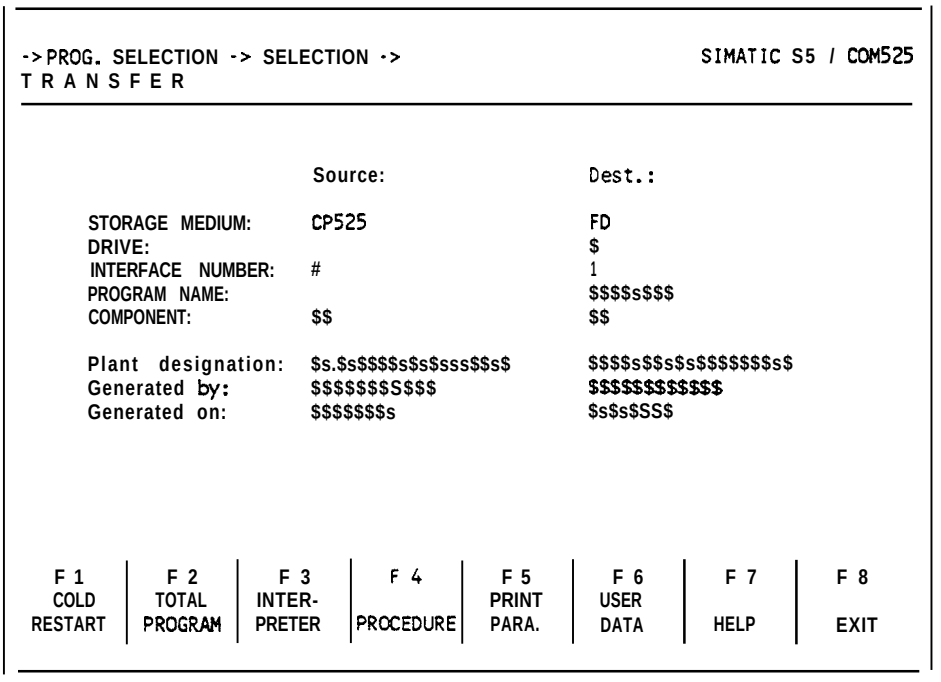
F1: change to the 'transfer from CP 525 to FD' mode

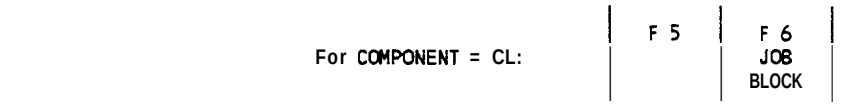
F3: change to the 'transfer from FD to CP 525' mode
(see 3.2.2)

F5: change to the 'transfer from FD to FD' mode
(see 3.2.3)

3.2.1 Transferring from CP 525 to FD

Call this menu with F1 (CP->FD) in the 'TRANSFER' mask.





F1: cold restart of the selected device interface on the CP 525
(must be carried out after every transfer to the CP)

F2: transfer total program,
-> acknowledgement prompt

F3: transfer interpreter,
if destination element already *exists*
-> acknowledgement t prompt

F4: transfer procedure,
if element already exists
-> acknowledgement prompt

F5: transfer print parameters
-> acknowledgement prompt
(With this function the specifications made in the 'ASSIGN PRI. PARA.' mask (see 3.8) are transferred).

F6: **case 1: component = PT**

The user data is transferred,
thefollowingmenuisoutput:

F1	F2	F3	F4	F5	F6	F7	F8
COLD RESTART	MESSAGE	Pc JOB	FRAME				EXIT

F1: cold restart of the selected device interface on the CP 525

F2: transfer messages (see following page)

F3: transfer PC **jobs (see following page)**

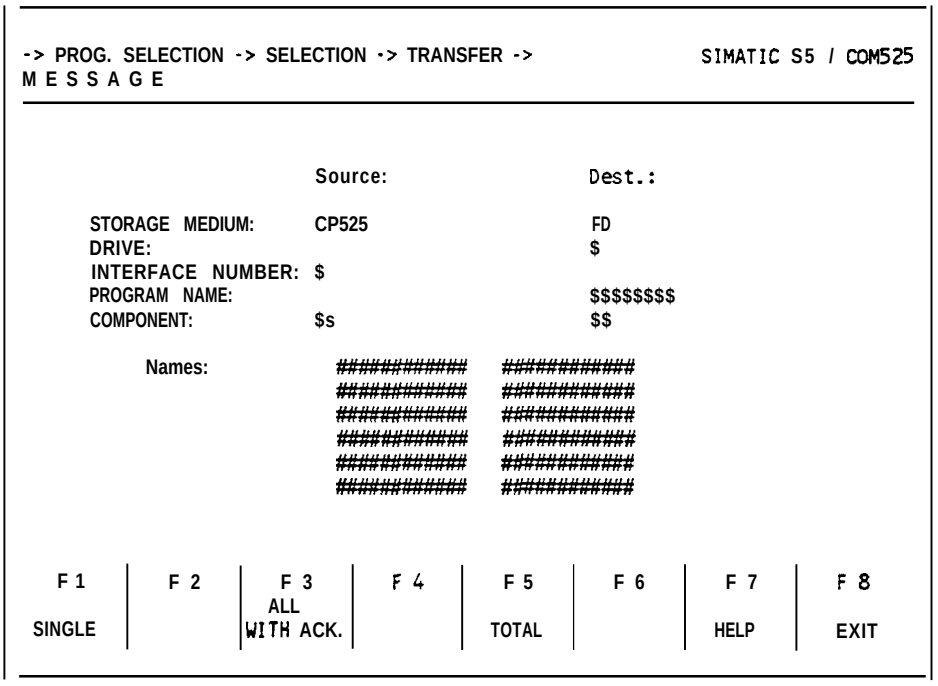
F4: **transfer frames (see following page)**

F8: **return *to* original menu**

Case 2: component = CL

The job block is transferred.
All the **jobs are always** transferred, i.e. not **only** new or changed **jobs.** If the destination already exists, the **destination is compared with the source to establish whether they are identical**.

The following mask is used with the PT component for transferring the element types MESSAGE, PC JOB and FRAME (in each case with the corresponding designation in the mask header):



F1: transfer single elements according to the entries in the 'names' fields.
If the **destination** element already exists (with different contents), -> acknowledgement prompt.

F3: transfer all elements.
If a destination element already exists (with different contents), -> **acknowledgement** prompt.

F5: transfer all elements without acknowledgement prompt (existing elements are overwritten).

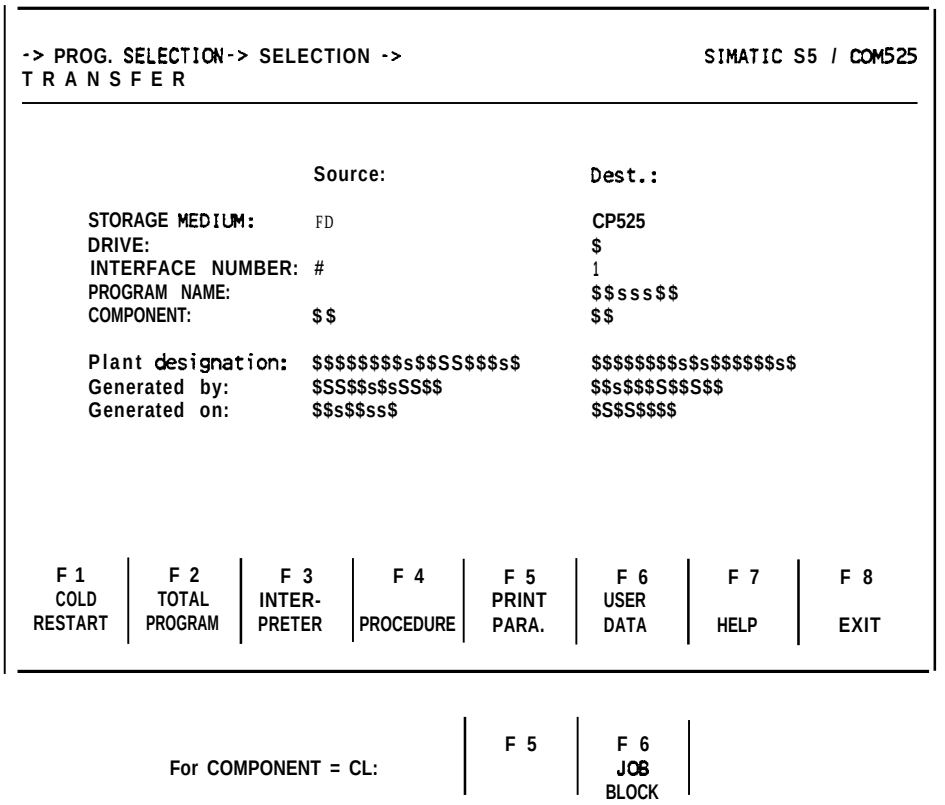
Field	Field type keys allowed	Limit value	Alternatives
DRIVE	OUTPUT FIELD		
INTERFACE NUMBER	numbers	1-2	1, 2 *
PROGRAM NAME	OUTPUT FIELD		
COMPONENT	OUTPUT FIELD		
Name	alphanumeric characters	-	* (in source)

F7 (HELP) can be used to make an entry in the fields marked '*'.

3.2.2 Transferring from FD to CP 525

Call this menu with F3 (FD->CP) in the 'TRANSFER' mask.

The description of the fields (except the output fields "plant designation", 'generated by" and "generated on") can be found in Section 3.2.1



F1: cold **restart** of the selected device interface on the **CP** 525 (must be carried out after **every** transfer)

F2: transfer **total program, interface must first be deleted** -> **acknowledgement** prompt

F3: transfer interpreter,
-> acknowledgement prompt

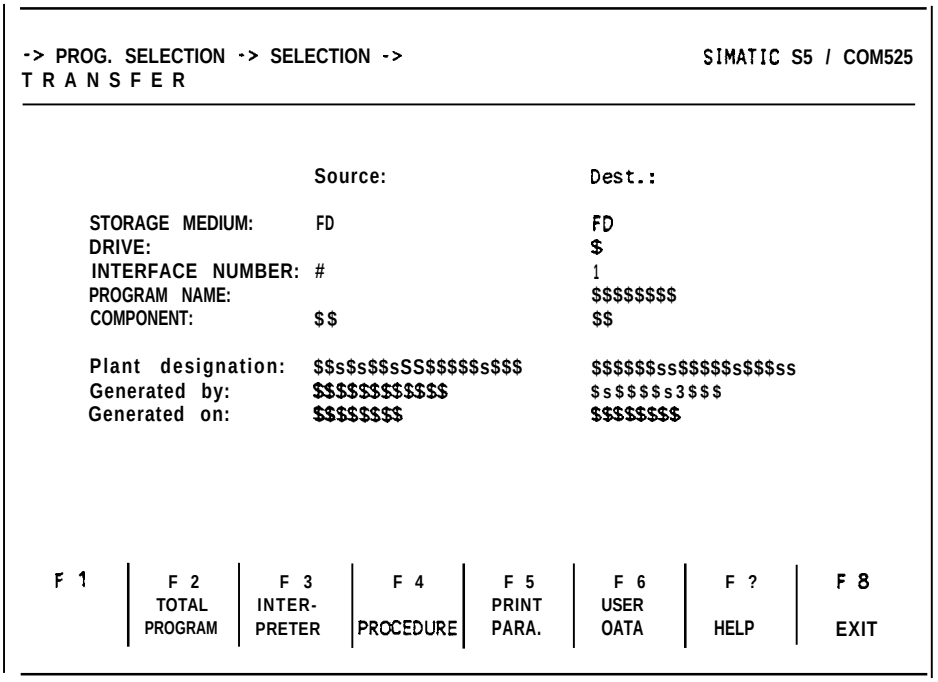
F4: transfer procedure,
-> acknowledgement prompt

F5: transfer print parameters
-> acknowledgement prompt
(With this function, the specifications made in the 'ASSIGN PRI. PARA.' mask (see 3.8) are transferred).

F6: see 3.2.1.

3.2.3 Transferring from FD to FD

Call this menu with F5 (FD->FD) in the 'TRANSFER' mask.



F2: transfer total program
-> acknowledgement prompt

F3: transfer interpreter,
-> acknowledgement prompt

F4: transfer procedure,
-> acknowledgement prompt

F5: transfer print parameters
-> acknowledgement prompt.

F6: see 3.2.1.

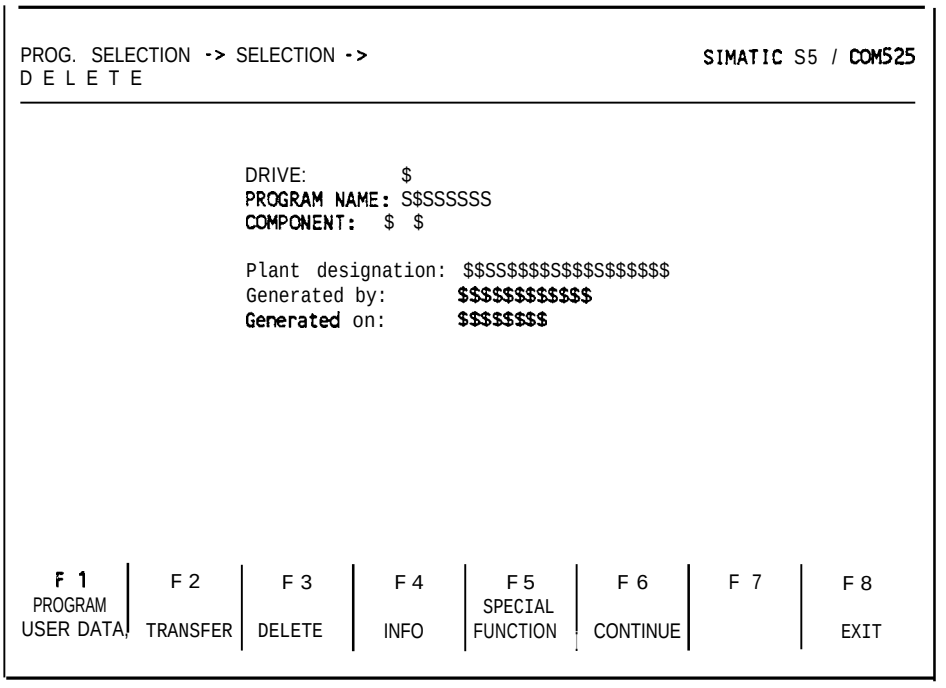
Field	Field type keys allowed	Limit value	Alternatives
DRIVE (source)	alphanum. characters		A,B,C,D,..,J
DRIVE (dest.)	OUTPUT FIELD (from the PROGRAM SELECTION mask)		
PROGRAM NAME (source)	alphanum. characters		*
PROGRAM NAME (dest.)	OUTPUT FIELD		
COMPONENT	OUTPUT FIELD		
Plant designation	OUTPUT FIELD		
Generated by	OUTPUT FIELD		
Generated on	OUTPUT FIELD		

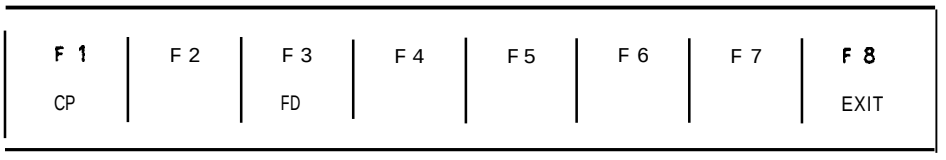
F7 (HELP) can be used to make an entry in the fields marked '*' .

3.3 Delete

This function allows programs or parts of programs to be deleted on the CP, hard disk or floppy disk.

Call the **'DELETEI** mask with F3 (DELETE) in the 'Selection mask (see **2.4**).



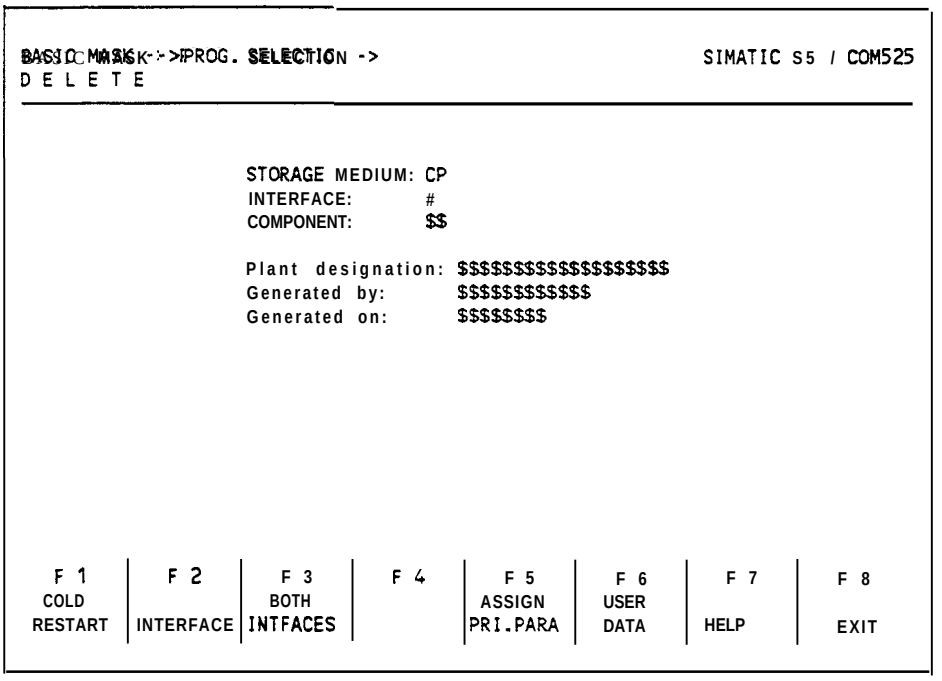


F1: change to 'delete CP 525' mode
(see 3 .3.1)

F3: change to 'delete FD' mode
(floppy disk or hard disk, see 3.3.2)

3.3.1 Delete CP 525

Call the 'DELETE' mask (CP 525) with F1 (CP) in the delete menu.



For COMPONENT = CL: F 5 | F6 JOB BLOCK

F1: cold restart on the selected device interface on the CP 525 (must be carried out following each deletion)

F2: delete selected interface
-> acknowledgement prompt

F3: delete whole user RAM
-> acknowledgement prompt

F5: delete print parameters
(with this function, the specifications made in the 'ASSIGN PRI. PARA' mask (see 3.8) will be deleted) .

F6 : **case 1:** component = PT

Delete user data,
the following menu is displayed:

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
COLD RESTART	MESSAGE	Pc J08	FRAME				EXIT

F1: cold restart on the selected device interface on the CP 525 (must be carried out following each deletion)

F2: delete messages (see following page)

F3: delete PC jobs (see following page)

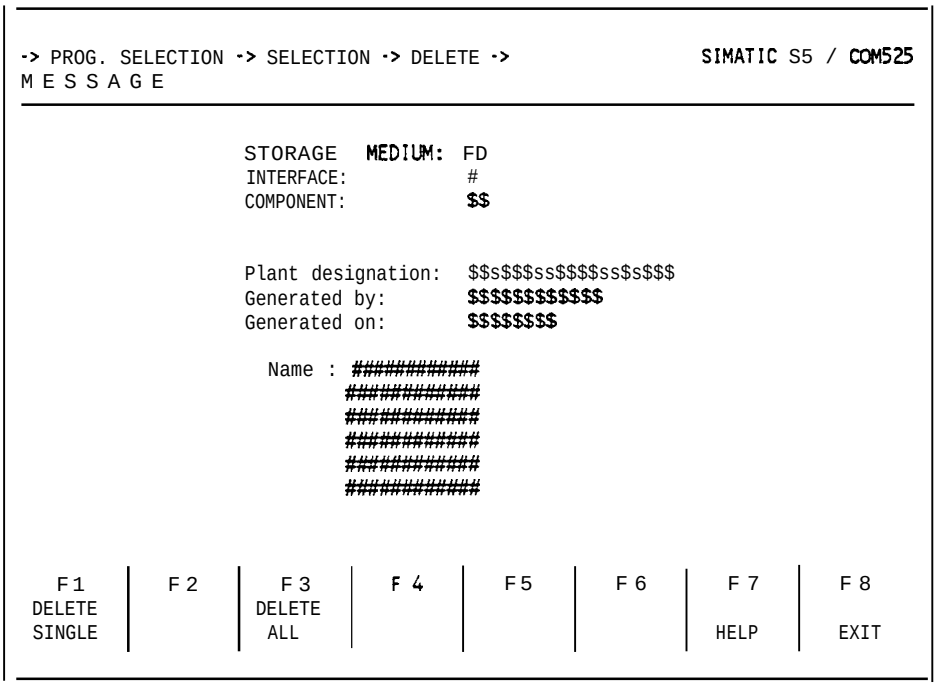
F4: delete frames (see following page)

F8: return to original menu

Case 2: component = CL

Delete job block
(all jobs are deleted)

The following mask is output with the PT component to DELETE the element types MESSAGE, PC JOB and FRAME (in each case with the corresponding designation in the mask header):



F1: delete individual elements according to the entries in the 'name' fields

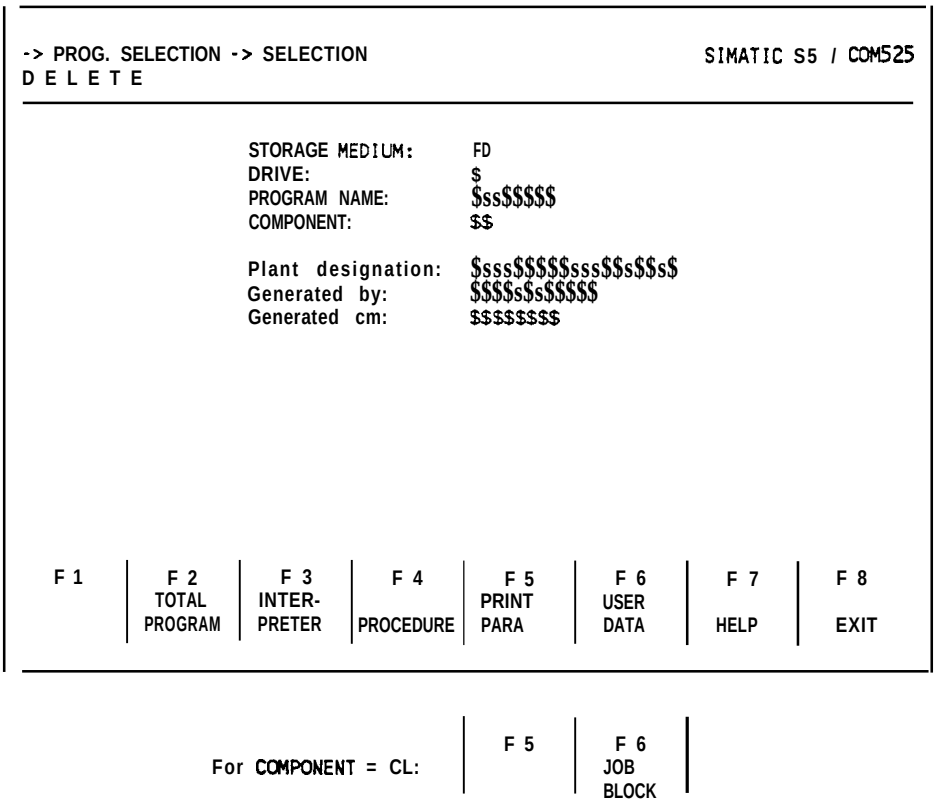
F3: delete all elements
-> acknowledgement prompt

Field	Field type keys allowed	Limit value	Alternatives
INTERFACE NUMBER	numbers	1-2	1, 2 *
PROGRAM NAME	OUTPUT FIELD		
COMPONENT	OUTPUT FIELD		
Plant designation	OUTPUT FIELD		
Generated by	OUTPUT FIELD		
Generated on	OUTPUT FIELD		
Name	alphanum. characters		*

F7 (HELP) can be used to make an entry in the fields marked '*'.

3.3.2 Deleting on FD

Call the 'DELETE' mask (FD) with the function key F3 (FD) in the delete menu.



F2: delete total program
-> acknowledgement prompt

F3: delete interpreter

F4: delete procedure

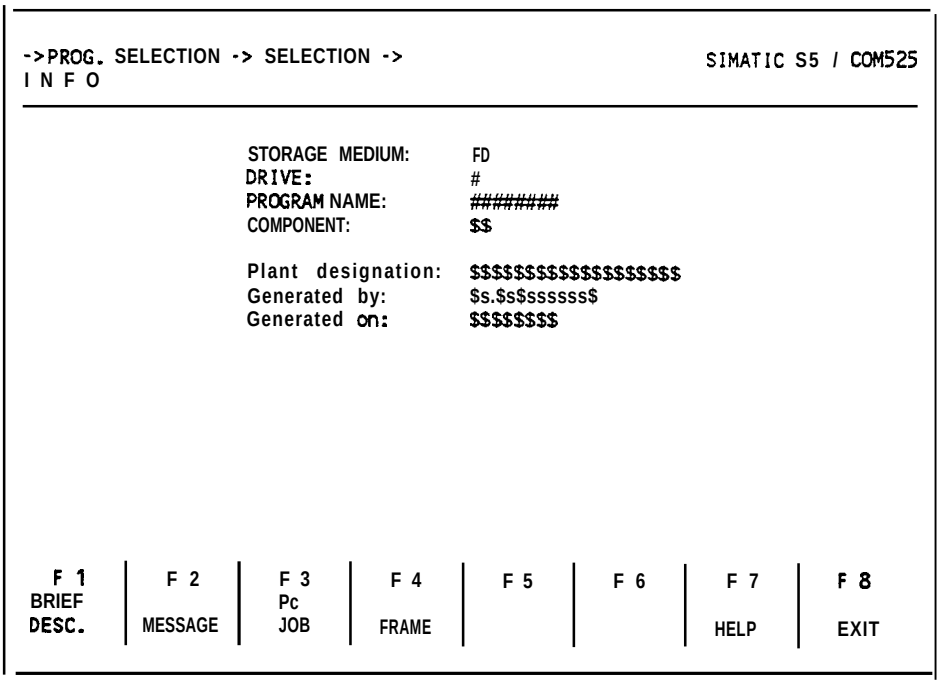
F5: delete print parameters

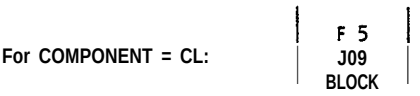
F6: see 3.3.1, but without F1 (COLD RESTART)

3.4 Info

In this mask, you can call up information about parts of the program (e.g. the names of all the PC jobs and their lengths). It is also possible to call up an overview of a whole program with F1 (BRIEF DESC). You can also have this information printed out (see 3.4.2 to 3.4.5), If you require information about further programs enter their names in the 'PROGRAM NAME' field.

Call the 'INFO' mask with F4 (INFO) and F1 (CP) or F3 (FD) in the 'SELECTION' mask.





F1: brief description of a user program

F2: info about messages (only for component PT)

F3: info about PC jobs (only for component PT)

F4: info about frames (only for component PT)

F5: info about a job block (only for component CL, in this case keys F2 to F4 have no function)

If the name of a library is specified as the program name the function keys F2 *to* F4 and F5 are assigned functions and F6 has the function 'CONTINUE'.

F6: output of a second menu:

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
BRIEF DESC.		INTER-PRETER	PROCEDURE	PRINT PARA	CONTINUE		EXIT

F1: brief description of the library

F3: info about interpreters

F4: info about procedures

F5: info about parameters

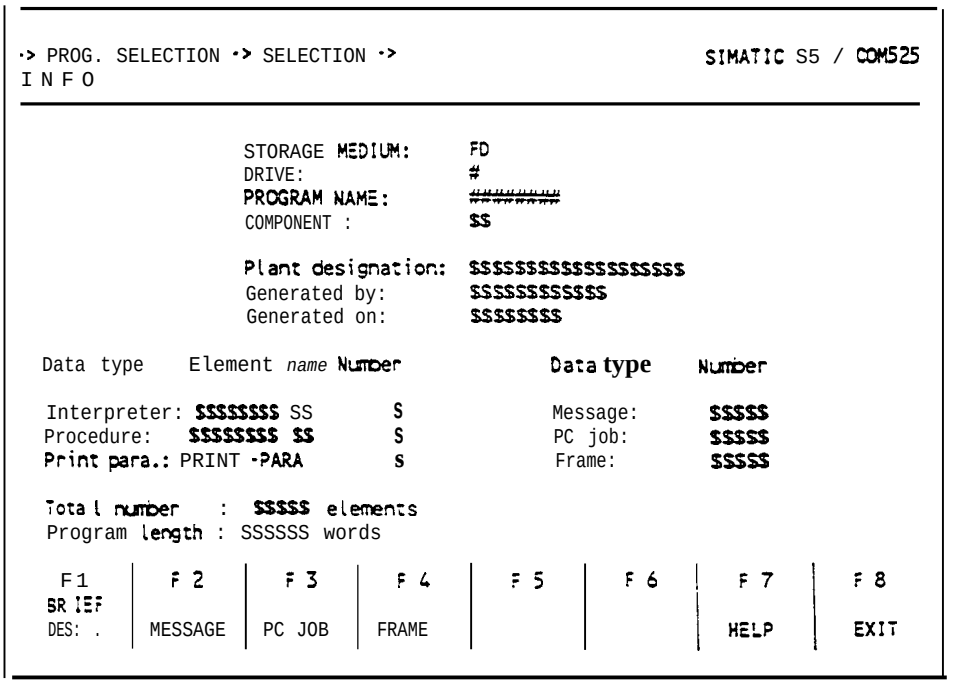
F6: returntofirstmenu

Field	Field type keys allowed	Limit value	Alternatives
DRIVE	alphanum. characters (only for FD info)	A - J	A,B,C,D, ...J *
INTERFACE NUMBER	numbers (only for CP info)	1-2	1, 2 *
PROGRAM NAME	alphanum. characters (only for FD info)		*
COMPONENT	OUTPUT FIELD		
Plant designation	OUTPUT FIELD		
Generated by	OUTPUT FIELD		
Generated on	OUTPUT FIELD		

F7 (HELP) can be used to make an entry in the fields marked '*' .

3.4.1 Brief Description

Press F1(BRIEF DESC.) in the 'INFO mask to call up the brief description submask.

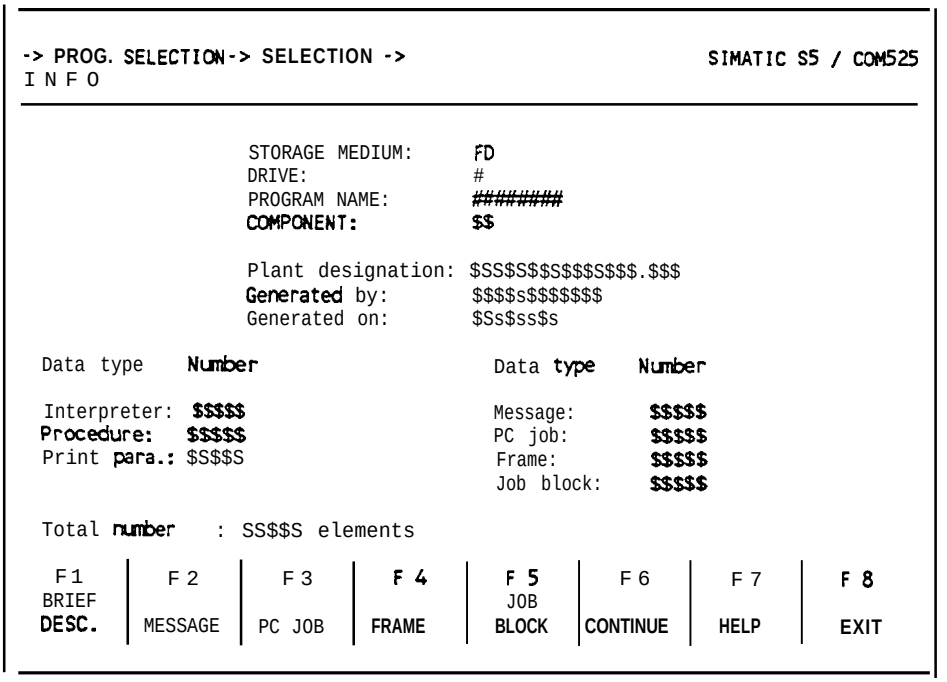


The names of the interpreted and procedure are displayed (in each case with version number) and the number of elements. In addition, COM 525 displays how much memory space is occupied by the whole program in the CP 525.

Note: the specified program length always includes 2 x 4 Kwords (2 x 8 Kbytes) for interpreters and procedures of both device interfaces (regardless of whether one or both interfaces are being used). The program length also includes the memory required for the memory manager. In the 'program length' field, the length of your program is specified, but not the memory space actually taken up; this is always a multiple of 8 Kbytes.

The name for print parameters is always PRINT-PARA in COM 525 programs (this is automatically assigned by COM 525) . It can only berenamed when it is being transferred to libraries.

If the name of a library is specified as the program name the brief description appears as follows:

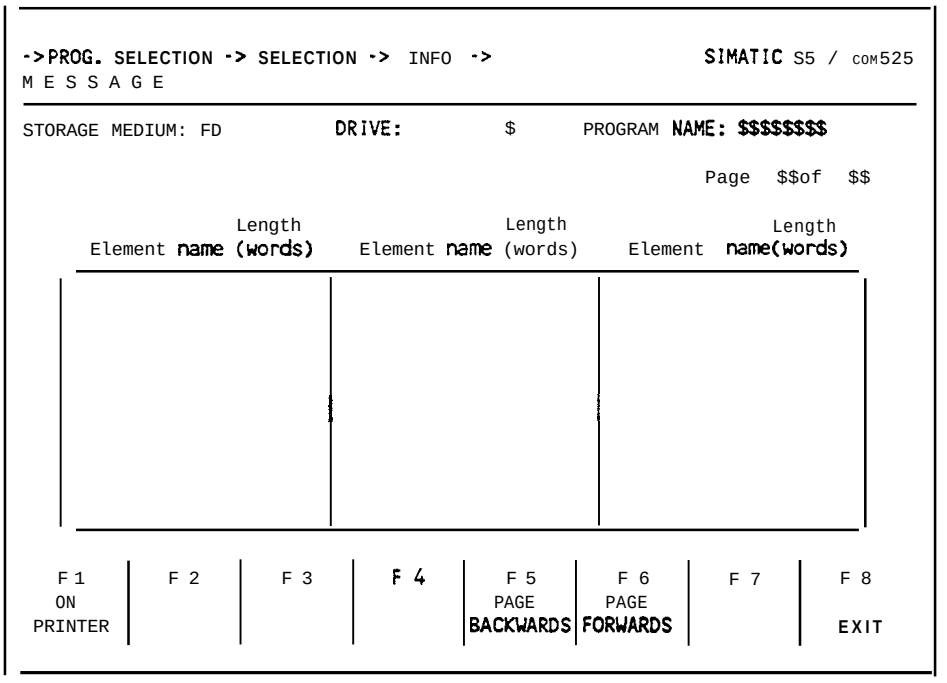


In addition to the number of data types, the number of interpreters, procedures and print parameters contained in the library are also displayed.

3.4.2 Information about Messages

Call the 'MESSAGE' mask with F2 (MESSAGE) in the 'INFO' mask.

The functions paging forwards and paging backwards will be required if there are more elements than can be displayed.



F1: output the displayed information on the printer

F5: page backwards

F6: page forwards

3.4.3 Information about PC Jobs

Call the 'PC JOB' mask with F3 (PC JOB) in the 'INFO' mask.

For the function key assignment see 3.4.2.

In this mask the job number belonging to each element is also output.

3.4.4 Information about Frames

Call the 'FRAME' mask with F4 (FRAME) in the 'INFO' mask.

For the function key assignment see 3.4.2.

3.4.5 Information about a Job Block

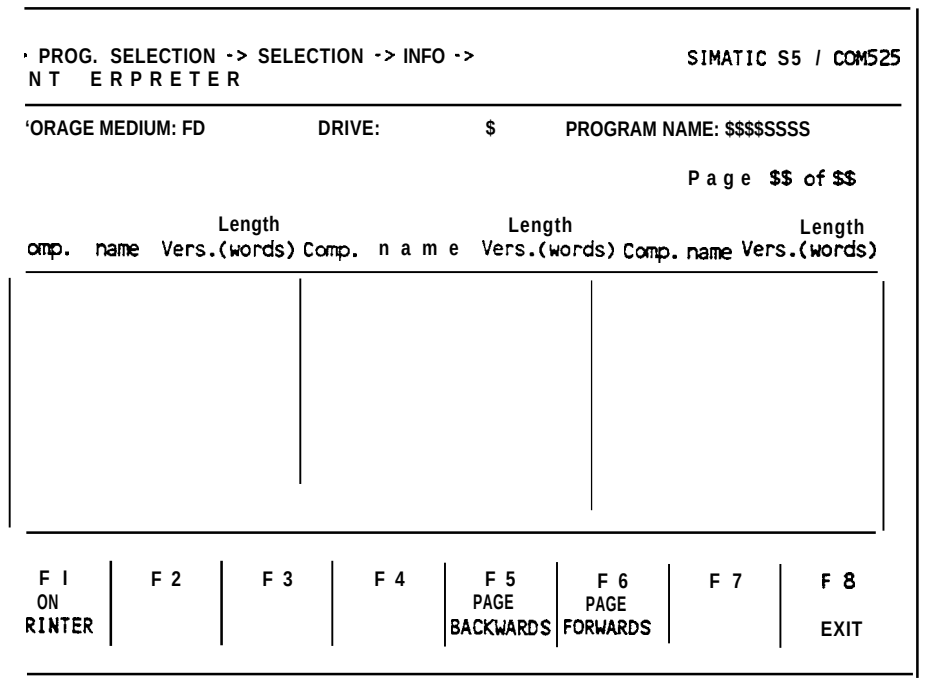
Call the 'JOB BLOCK' mask with F5 (JOB BLOCK) in the 'INFO' mask. (Only for component CL).

For the function key assignment see 3.4.2.

The numbers of all programmed jobs will be displayed.

3.4.6 Information about Interpreters/Procedures

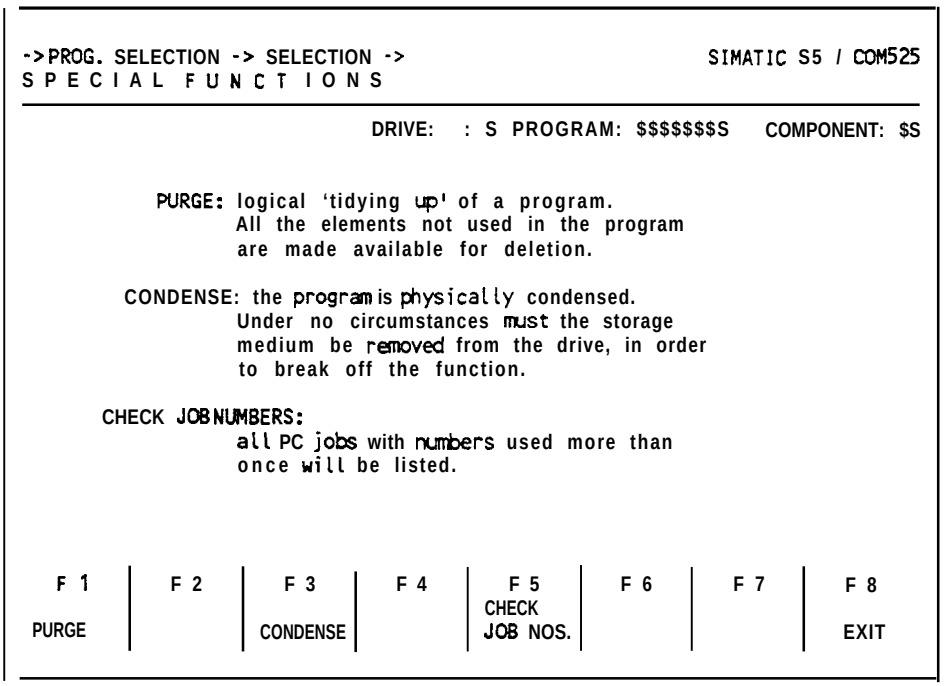
Call this mask with F6 (CONTINUE) and F3 (INTERPRETER) or F4 (PROCEDURE) in the 'INFO' mask. This information function is only available with libraries. It provides the following mask:



This mask provides information about components (PT or CL), names, version and length of the interpreters or procedures contained in the library.

3.5 Special Functions

Call the 'SPECIAL FUNCTIONS' mask with the function key F5 (SPECIAL FUNCTIONS) in the 'SELECTION' mask.



F1: branch to the 'PURGE' mask (logical tidying up)

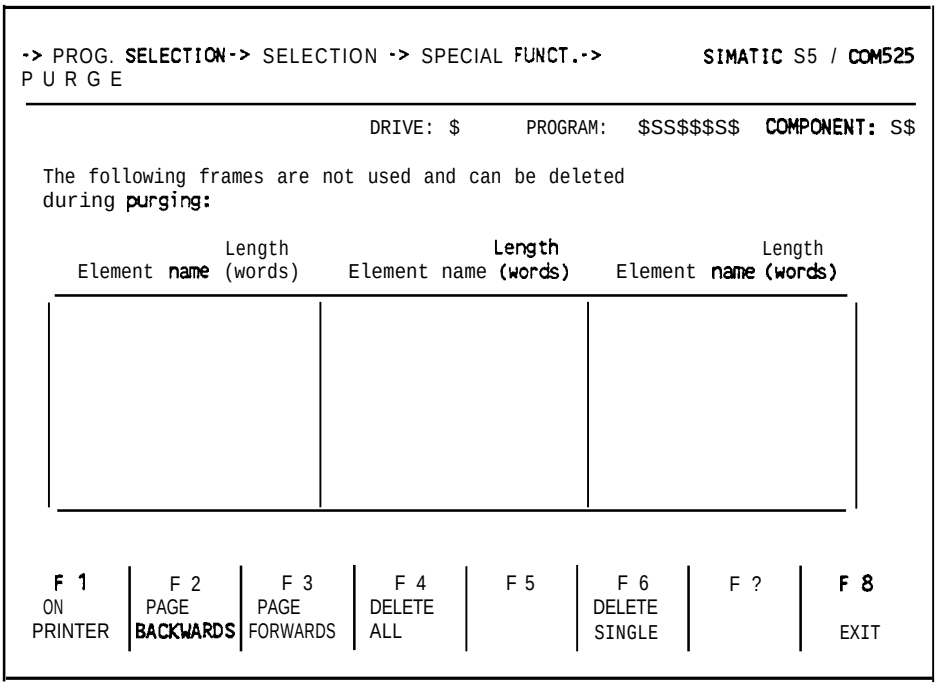
F3: condense a file (physically)

F5: check the PC jobs for numbers used more than once

3.5.1 Purge

Call the 'PURGE' mask with F1 (PURGE) in the 'SPECIAL FUNCTIONS' mask.

The functions page forwards and backwards are required if there are more elements present than can be displayed,



F1: output of the displayed information on the printer

F2: page backwards

F3: page forwards

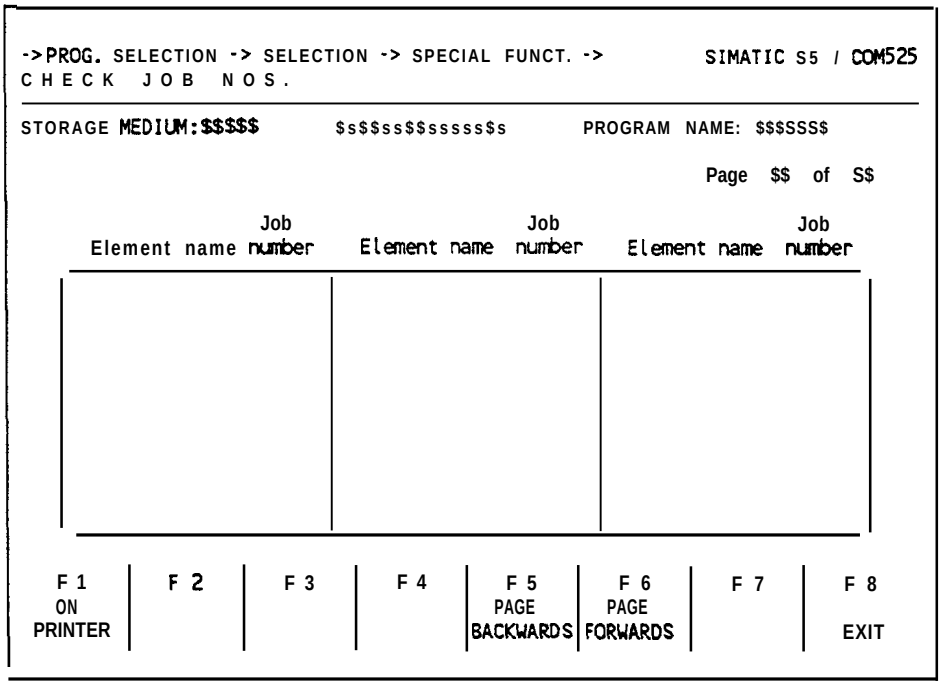
F4: delete all the elements made available
-> acknowledgement prompt

F6: delete single elements (these are made available for deleting one after the other) -> acknowledgement prompt

3.5.2 Check Job Numbers

Call the 'CHECK JOB NOS. ' mask with F5 (CHECK JOB NOS.) in the 'SPECIAL FUNCTIONS' mask.

The functions page forwards and page backwards are required if there are more elements present than can be displayed.



This mask displays all the PC jobs which (incorrectly) have the same job number, arranged according to job numbers and in alphabetical order.

F1: output of the displayed information on the printer

F5: page backwards

F6: page forwards

3.6 Assign Interpreter Parameters

This function is available only for programs with the PT component; this component is used to assign parameters to the printer (should not be confused with the function 'assign printer parameters' for listings which are output on the printer connected to the PG). The input fields on the left-hand side of the mask contain the default values of the parameters for the interpreter present in this program. The alternatives are displayed on the right-hand side of the mask.

The defaults for the assignment of date and time parameters corresponds to the German format.

If you require the English format, the specification should appear as below:

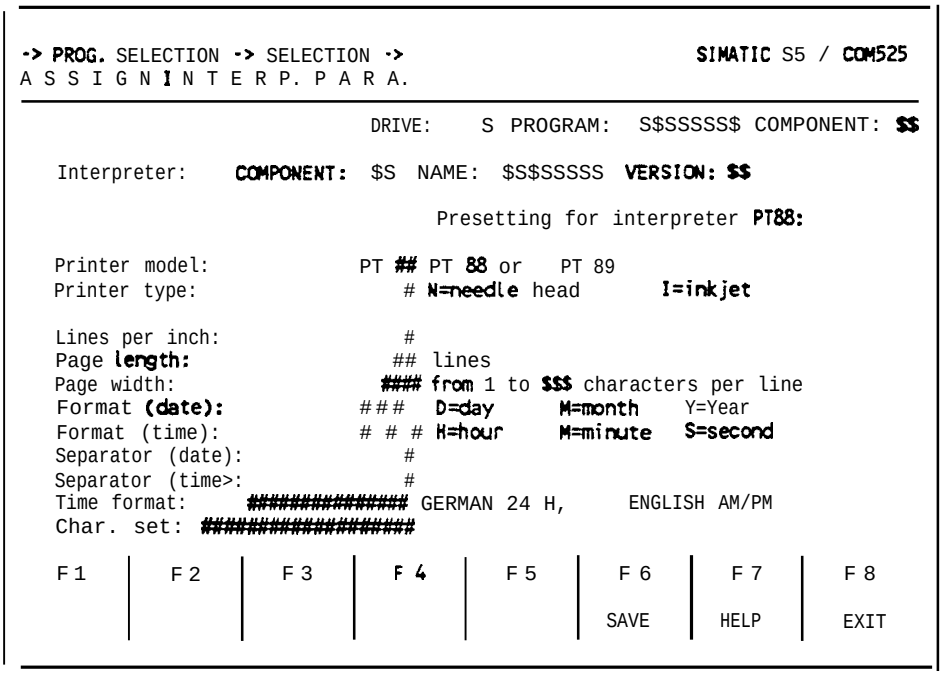
format (date): M D Y
format (time): H M S
separator (date): /
separator (time): :
time format: ENGLISH AM/PM

Example:

date / time German: **30.06.87 14.30.22**

date / time English: **06/30/87** 02:30:22 **PM**

Call the *ASSIGN INTERP. PARA. ' mask with F6 (CONTINUE) and F1 (ASSIGN INT. PARA.) in the 'SELECTION' mask.



F6: save the interpreter parameter assignment

F8: does not save (if **data** have been changed without saving them an acknowledgement prompt appears);
return to the 'SELECTION' **mask**

Field	Field type keys allowed	Limit value	Alternatives
COMPONENT	OUTPUT FIELD		
NAME	OUTPUT FIELD		
VERSION	OUTPUT FIELD		
Printer model	numbers		88, 89 *
Printer type	letters		T, N *
Lines per inch	numbers		6, 4, 3 *
Page length	numbers	dependent on 'lines per inch' 1)	*
Page width	numbers	1 - 80 (for PT88) 1 - 136 (for PT89)	*
Format (date)	letters		D, M, Y *

F7 (HELP) can be used to make an entry in the fields marked '*'.

1) 6 lines/inch . . . 1 - 99 (default withheld: 72)
4 lines/inch . . . 1- 66 (default with HELP: 48)
3 lines/inch . . . 1 - 49 (default withheld: 36)

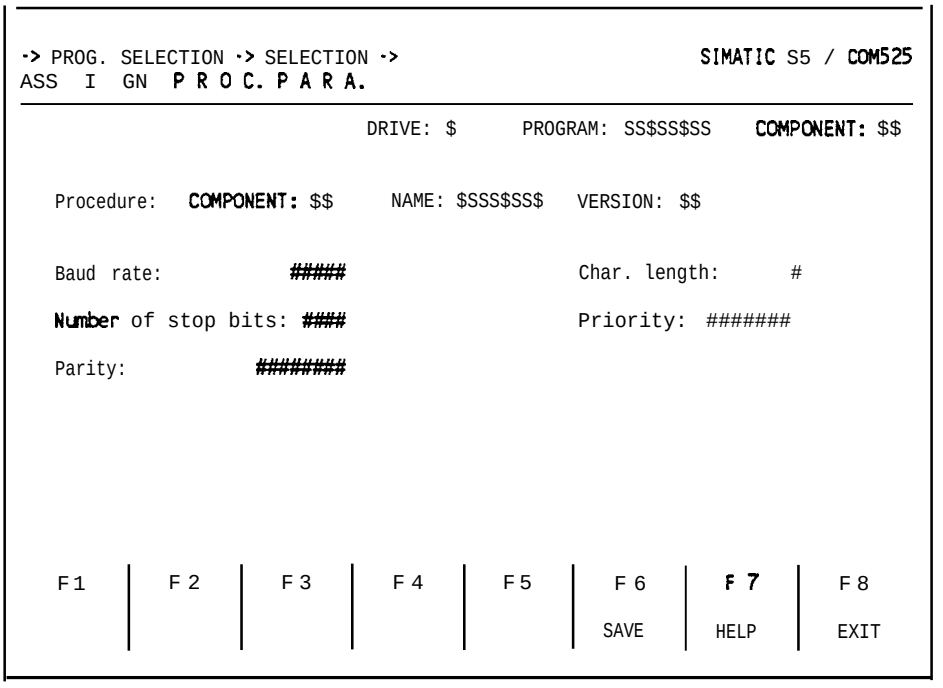
Field	Field type keys allowed	Limit value	Alternatives
Format (time)	letters		H, M, S *
Separator (date, time)	letters		* + . , ' / : ; | *
Time format	alphanumeric characters		GERMAN 24 H ENGLISH AM/PM *
Char. set	letters		GERMAN, ENGLISH, ASCII, FRENCH, SPANISH, NORWEG./DANISH, SWEDISH, FINNISH, INTERNATIONAL *

F7(HELP)can be used to make an entry in the fields marked '*'. In the 'separator' field the character '*' is also one of the alternatives.

3.7 Assigning Procedure Parameters

Just as with the function 'assigning interpreter parameters' the parameters for the procedure can also be changed. The input fields contain defaults and display the parameters for the procedureintheprogram named in the mask.

Call the 'ASSIGN PROC. PARA' mask with F6 (CONTINUE) and F2 (ASSIGN PROC. PARA) in the 'SELECTION' mask.



F6: save the procedure parameter assignment

F8: does not save (if data have been changed, without saving them an acknowledgement prompt appears), return to the 'SELECTION' mask

Field	Field type keys allowed	Limit value	Alternatives
COMPONENT	OUTPUT FIELD (ace. to prog. name)		
NAME	OUTPUT FIELD		
VERSION	OUTPUT FIELD		
Data rate	numbers		9600,4800,2400, 1200, 600, 300, 200, 150, 110, 100, 75, 50 * 1)
Char. length	numbers		8, 7, 6, 5 * 1)
Number of stop bits	numbers		1, 1,5, 2 * 1)
Priority	letters		HIGHER, LOWER * 1)
Parity	letters		EVEN, ODD * 1)

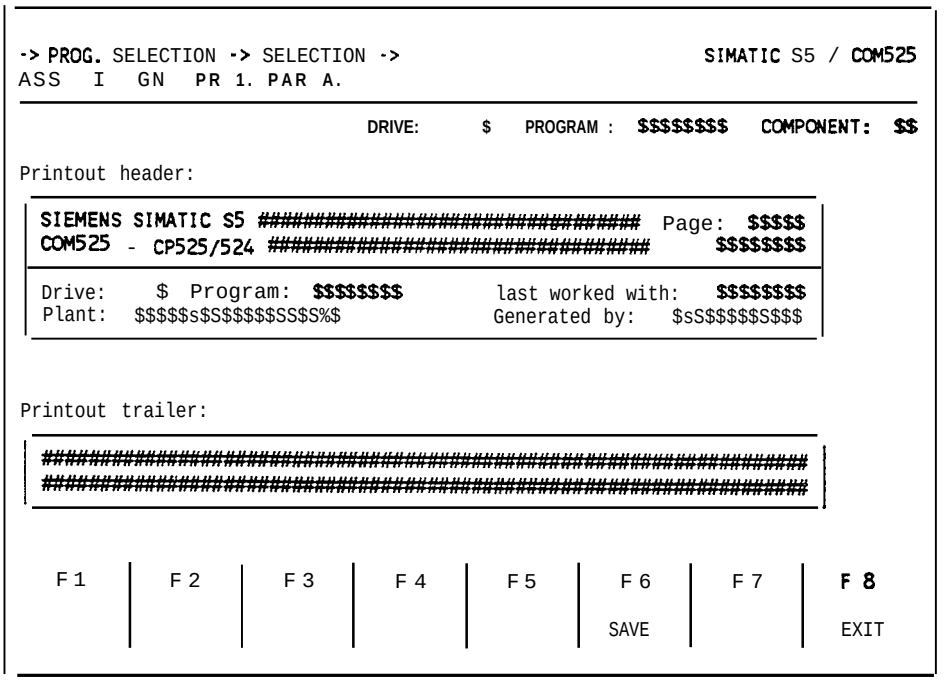
F7(HELP) can be used to make an entry in the fields marked '*'.

1) The options depend on the procedure. Only those parameters which can be changed in the particular procedure aremade available for changing.

3.8 Assigning **Printer Parameters**

You can use this function to determine the printout header and trailer of listings to be output **on the** printer **connected to the PG**. This has nothing to do with the function **'event** output and **listing on** the **PT88/PT89 printer'** . After this mask is saved, **all** the lists are printed out with the specifications made in this mask (see 3.4, 3.5, 3.9 and 4.2).

Call the 'ASSIGN **PRI. PARA.** ' mask with F6 (CONTINUE) and F3 (ASSIGN **PRI. PARA.**) in the 'SELECTION' mask.



F6: save the printer parameter assignment

F8: **abort (if data** is changed without saving it, an acknowledgement prompt appears);
return to the **'SELECTION'** mask

Field	Field type keys allowed	Limit value	Alternatives
Printout header	ASCII upper/lower case		
Printout trailer	ASCII upper/lower case		
Drive	OUTPUT FIELD (defaults from the PROGRAM SELECTION mask)		
Program	OUTPUT FIELD (defaults from the PROGRAM SELECTION mask)		
Component	OUTPUT FIELD (defaults from the PROGRAM SELECTION mask)		
Iast worked with	OUTPUT FIELD (defaults from the PROGRAM SELECTION mask)		
Plant	OUTPUT FIELD (defaults from the PROGRAM SELECTION mask)		
Generated by	OUTPUT FIELD (defaults from the PROGRAM SELECTION mask)		

3.9 Listing

Using the listing function, you can have a summary of your program printed out. This shows the structure of the program and how the individual parts of the program access each other.

When parts of programs are listed the corresponding elements are listed one after the other.

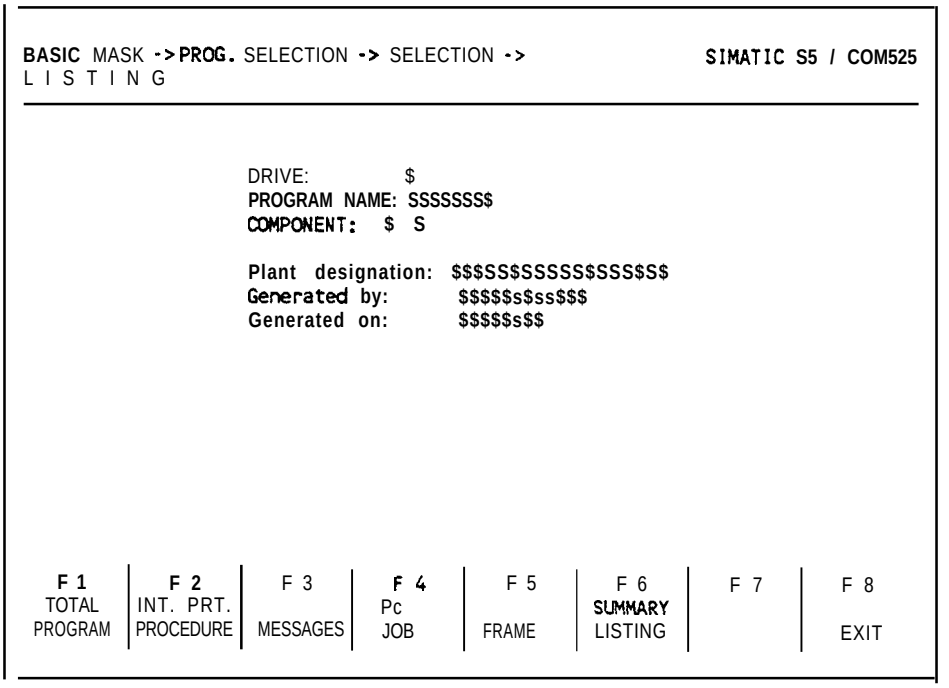
Remember that these listings are printed out on a printer connected to the PG, i.e. not on the listing printer.

The pages of all listings have a printout header and trailer that is specified with the 'assign printer parameters' function (see 3.8) and that can be saved on floppy or hard disk. The print parameters of a PT program can also be stored on the CP 525 (CP 524).

The page width of a listing is always 72 characters. Message or PSL lines which are longer than this are broken up arbitrarily after 72 characters.

All the listing functions can be aborted with F8 (EXIT).

Call the 'LISTING' mask with F6 (CONTINUE) -> F4 (LISTING) in the 'SELECTION' mask.



For COMPONENT = CL: | F 4 JOB BLOCK |

F1: listthetotalprogramonthepri.nter

F2: print out the parameter assignments for the interpreter and procedure

F3: print out all messages

This starts with the printout of the parameters for the PC job, sequential message list (SML). These parameters determine the position am-structure of the date, time and status for all messages. The messages are then printed out in ascending order with their static and dynamic parts.

F4: with component PT: a new mask is displayed:

F1: all the PC jobs whose names have been entered manually or with HELP are printed out.
F3: printout of all the PC jobs. The PC jobs are printed out in the order in which they were made available during programming (see 5.4),

For component CL: printout of the job block.

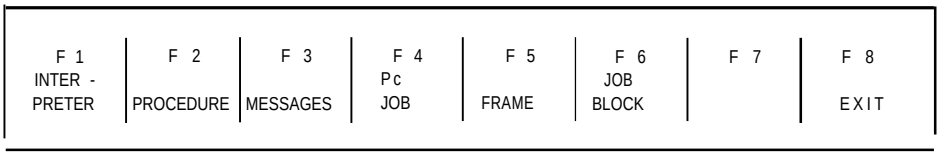
F5: a new mask is displayed:

F1: all the frames whose names have been entered manually or with HELP are printed out.
F3: all frames are printed out,

F6: printout ofa summary listing.
In additicm to the interpreter and procedure parameters, a table listing the PC jobs is also printed out.

F8: abort the current listing or return to the 'SELECTION' mask.

If the program name is a library then **F1** (LISTING) calls the menu shown as follows:



F1: change to the mode 'listing of the interpreters contained in the library' .
Analogous ***to the*** **'DELETE'** function (see 3.3.1) in the follow-on menu, F1 (SINGLE LISTING) is used to list single interpreters or F3 (TOTAL LISTING) for all interpreters.

F2: changetothemode 'listing of the procedures contained in the library',
Branch see **F1.**

F3: printout of all messages (see previous page).

F4: printout of all PC jobs (see previous page).

F5: printout of all frames (see previous page).

F6: change to mode 'listing the job blocks contained in the library'.
Branch see **F1.**

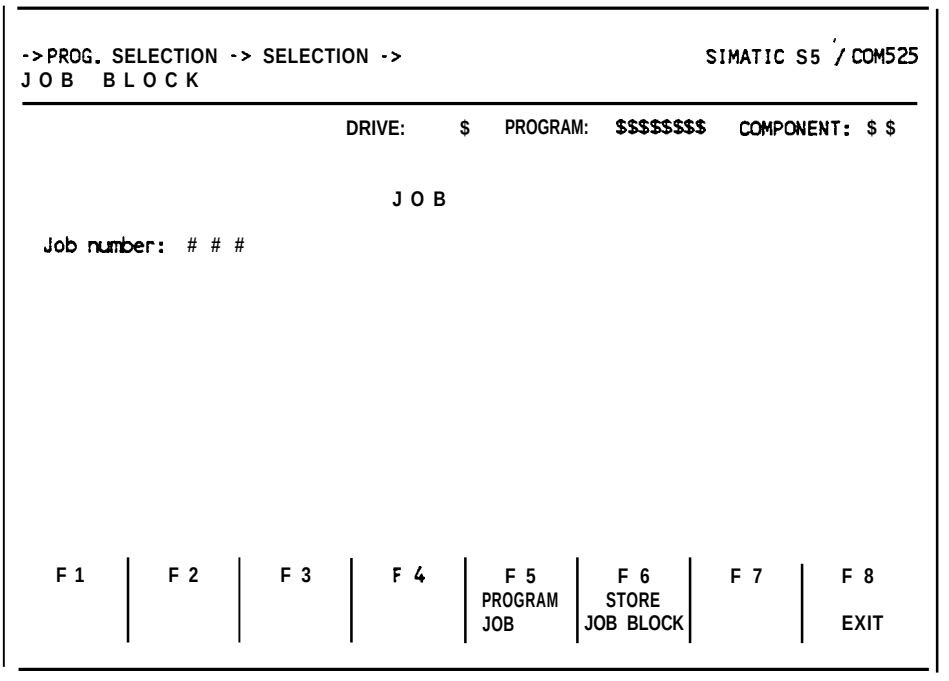
F8: abort the current listing or
return to the **'SELECTION'** mask.

4 Functions with the Computer Link Component

For further information about the computer link component refer to the user's guide "Computer link with RK 512" in this manual.

4.1 Programming the Job Block

Call the 'JOB BLOCK' mask with F1 (PROGRAM USER DATA) in the 'SELECTION' mask.



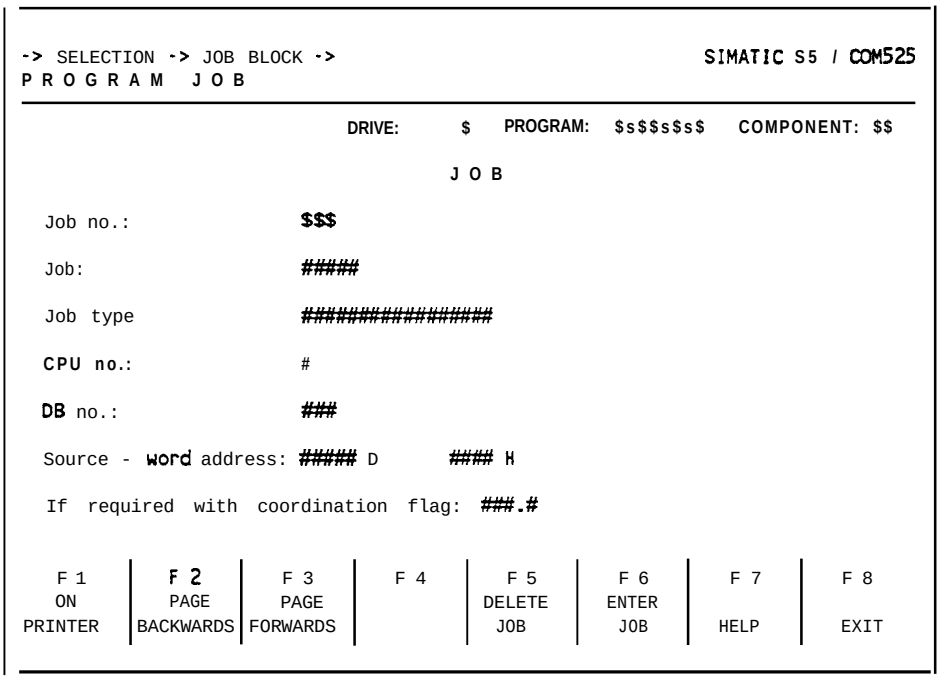
F5: change to the 'program job' mode; branch to the 'PROGRAM JOB' mask

F6: save a job block

Field	Field type keys allowed	Limit value	Alternatives
Job number	numbers	1 - 189	

4.2 Programming a Job

Call the ' PROGRAM JOB' **mesk with** F5 (PROGRAM JOB) in the 'JOB BLOCK' mask.



F1: the job is output on the printer

F2: if it exists, the job with the next lower job number is displayed on the screen

F3: if it exists, the job with the next higher job number is displayed on the screen

F5: delete the current job

F6: enter the current job

Field	Field type keys allowed	Limit value	Alternatives
Job number	numbers	1 - 189	
Job	upper case letters		SEND FETCH *
Job type (see Table 1)	upper case letters		*
CPU no.	numbers	1 - 4	1, 2, 3, 4 *
DB no.	numbers	3 - 255	
Address	numbers / hex. numbers	(see Table 1)	
Coordination flag	numbers	bytes: 1 - 223 bits: 0 - 7	

F7 (HELP) can be used to make an entry in the fields marked '*'.

JOB	JOB TYPE	max. ADDRESS	ADDR. TYPE
s	DATA BLOCK	255	WORDS
E	SYSTEM DATA	511	WORDS
N	ABSOLUTE ADDRESSES	65535	WORDS
D	EXT. DATA BLOCK	255	WORDS
	DATA BLOCK	255	WORM
	FLAGS	255	BYTES
F	INPUTS	127	BYTES
E	OUTPUTS	127	BYTES
T	1/0'S	255	BYTES
c	TIMER LOCATIONS	255	WORDS
H	COUNTER LOCATIONS	255	WORDS
	SYSTEM DATA	511	WORDS
	ABSOLUTE ADDRESSES	65535	WORDS
	EXT. DATA BLOCK	255	WORDS
	EXT. 1/0' S	255	BYTES

Table 1: Job types

If the specified limit values for addresses are exceeded then a warning is output. The jobs will nevertheless be accepted if the address is less than or equal to FFFFH.

5 Functions with the Printer Component

5.1 General Information

At the top right of all the masks the three output fields "DRIVE", "PROGRAM" and "COMPONENT" appear. They always contain the defaults specified in the 'PROGRAM SELECTION mask.

If a data element already exists, you are prompted to decide whether it should be overwritten and the acknowledgement menu is displayed.

ACK.010: Element already exists! - overwrite?

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
YES		NO					

If you exit a mask using F8 (EXIT) in which fields have been filled in without having entered the data, then an acknowledgement is also prompted.

ACK.002: Loss of data - exit mask?

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
YES		NO					

If you answer 'YES' the action described is carried out, if you answer 'NO' you return to the original menu.

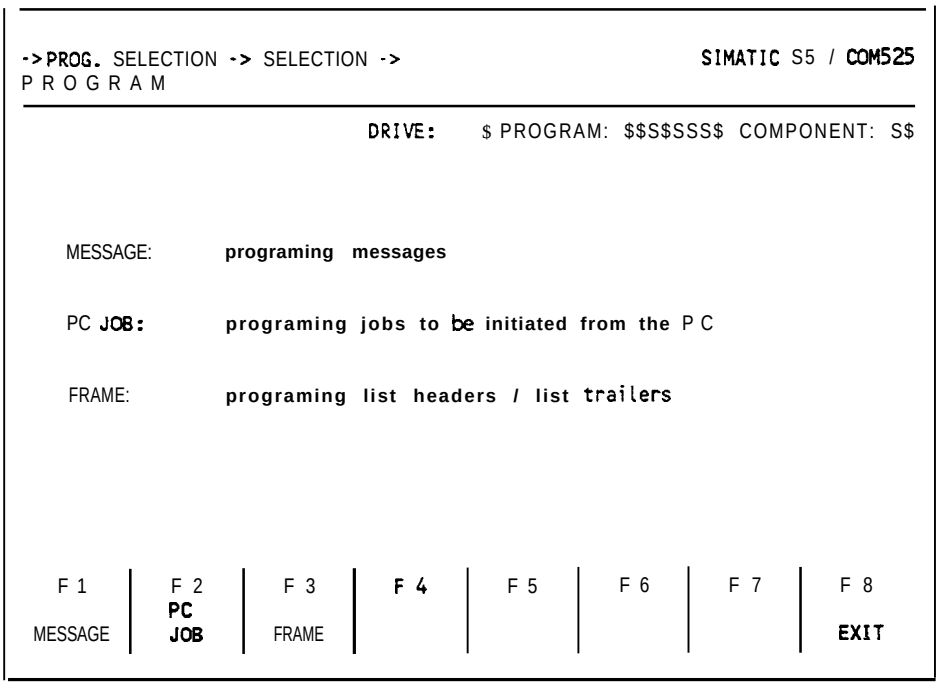
The fields in the masks marked with '*' in this manual can have inputs made using F7 (HELP).

5.2 Programming

Call the 'PROGRAM' mask with **F1** (PROGRAM USER DATA) in the 'SELECTION' mask.

In the **'PROGRAM'** mask you select what you wish to program:

- messages
- PC jobs
- frames



F1: change to 'programming messages' mode;
branch to 'MESSAGES' mask

!Important! before programming messages forthefirst tine, the PC_job that determinesthe position and **structure of the date, time** and **status** for all messages must be generated (see 5.3 and 5.4).

F2: change to the ‘program PC jobs’ mode;
branch to the ‘PC JOB' mask

F3: change to the ‘program frames' mode;
branch to the 'FRAME' mask

5.3 MESSAGES

Messages are made up of the following:

- static parts, i.e. fixed texts
- 'fixed' dynamic data (date, time, message status)
- dynamic parts, i.e. variable data which can be inserted at any point between the static parts.

static parts

The structure and contents of the static parts are determined when you program them on the PG using the text editor.

'Fixed' dynamic data

The position and structure of the 'fixed' dynamic data are the same for all messages. They must be specified globally before you program the messages. You do this when you program the PC job 'sequential message list' (SML, see Section 5.4.1). Before you program the first messages, you must therefore program the PC job SML.

The specifications for

- the date (position, order, print attributes)
- the time (position, order, print attributes)
- the message status (position, length, texts, print attributes)

apply for all messages in a program and can be modified only in the 'SEQ.MESSAGE LIST' mask, i.e. they cannot be changed by the editor.

In the editing window of the PG screen, the corresponding fields are displayed inversely.

Dynamic part

Each message can have a maximum of one dynamic parameter. When programming the static parts, you can decide where you wish to insert this parameter.

The number of reserve characters inserted in the static text for a dynamic field serves as the default for the number of characters when you program the dynamic part.

You can select the following features for each dynamic field:

number of characters in the list
- print attributes (bold, expanded, underlined, acoustic signal)
- position in the list

All other data (data type, source) are supplied to the CP by the CPU. They are therefore not prompted during programming.

Messages are programmed in two steps as follows:

1. The static parts are programmed first. Press F1 (STAT. PART) in the 'MESSAGES' mask you call the 'STATIC PART' mask.

2. The dynamic part is then specified for the static parts which have already been programmed. This is achieved by branching to the 'DYN. PART' mask using F2 (DYN. PART) in the 'MESSAGES' mask.

WithF4 (COMPLETE DYN. PART) in the 'MESSAGES' mask you can select the 'COMPLETE DYN. PART' mask. This function automatically searches for dynamic fields which have not yet been programmed although the static parts are already specified.

Notes on programming messages

To reduce the processing time of the editor to a minimum, up to 100 messages (static and dynamic parts) can be buffered in the RAM memory of the PG (called editing buffer below). Transfers from and to the mass storage of the PG (floppy or hard disk) are performed only at the beginning and end of an editing session:

at the beginning when you start the editor in the 'MESSAGES' mask with F1 (STAT.PART), F2 (DYN. PART) and F4 (COMPLETE DYN. PART)
at the end of the session in the masks 'STAT. PART', 'DYN. PART' or 'COMPLETE DYN PART' with F8 (EXIT).

In these cases the time required for the transfer must be taken into account. The amount of time depends on the following:

the number of messages programed, i.e. the more messages to be transferred the longer the transfer time
the speed of the external. PG memory (hard disk is faster than floppy)

It is possible to optimise the editing and transfer times by following certain procedures, as shown in the following examples.

Example 1: programming a large number of new messages

Read in only a few (e.g. one) of the already programmed messages.
- Program your new messages (up to 99) in one session.
- Save the programmed message.

Advantages:

- At the beginning there are almost no transfer times, since you only read in a few messages.
- You can then edit up to 99 messages without being disturbed (i.e. without losing time through transfers), before you complete the session and have to save the messages.

Disadvantages:

- At the end of the editing session the transfer time is particularly long. Depending on the number of messages and the type of external storage on the PG the time may exceed 10 minutes.

Example 2: **programming a few new messages**

- Read in only a few (e.g. one) of the already programmed messages.
- Program your new messages.
- Save the programmed messages.

Advantages:

- **At the** beginning there **are almost no transfer times** since you only read in a few messages.
- At the end there are also only short transfer times since only a few messages have been programed.

Example 3: [illegible] individual messages

- Read in the appropriate message singly.
- Change this message.
- Save the modified message.

Advantages:

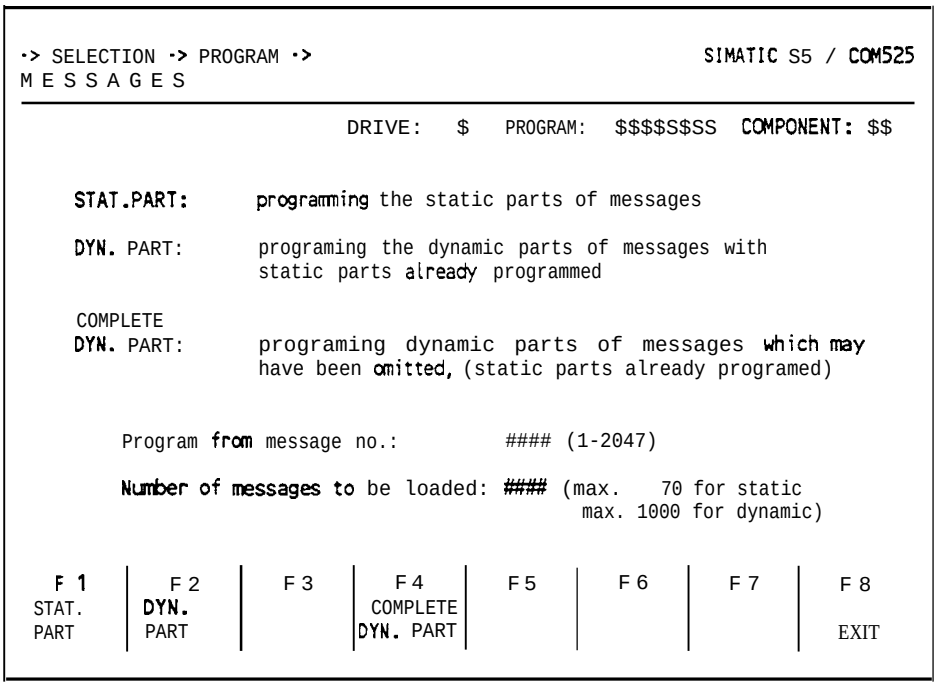
- **At the** beginning there **are almost no transfer times** since onlyonemessagehasbeenread in.
- **At the end there are** only short **transfer times since only one** message **must be** saved.

Example 4: changing a lot of messages

- If the messages are consecutive, **follow the procedure** as described in **Example** 1.

- If the messages are separated, follow the procedure as described in Example 3.:

You call the 'MESSAGES' mask with the function key F1 (MESSAGE) in the 'PROGRAM' mask.



The field "program from message no." has the default '1' or the number of the first message programmed. If you specify the "number of messages to be loaded" (default '12') you can select a specific number of messages.

F1: change to 'generation of static text' mode;
branch to 'STATIC PART' mask.

F2: change to 'generation of dynamic text' mode;
branch to 'DYN. PART' mask.

F4: change to *completion of dynamic text' mode;
branch to 'COMPLETE DYN. PART' mask.

Field	Field type keys allowed	Limit value	Alternatives
Program from message no.	numbers	1 - 2047	
Number of messages to be loaded 1)	numbers	1 - 70 or 1 - 1000	

1) By 'messages to be loaded' the following is meant:

- STAT. PART: - number of messages to be read into the editing buffer,

- DYN. PART or COMPLETE DYN. PART: - number of messages whose dynamic fields (or not yet programmed dynamic fields) are to be checked. Only the messages which have a dynamic field (or a not yet programed dynamic field) are read into the buffer.

5.3.1 Static Part of Messages

Before you can branch to the 'STAT. PART' mask, you must complete the following fields in the 'MESSAGES' mask:

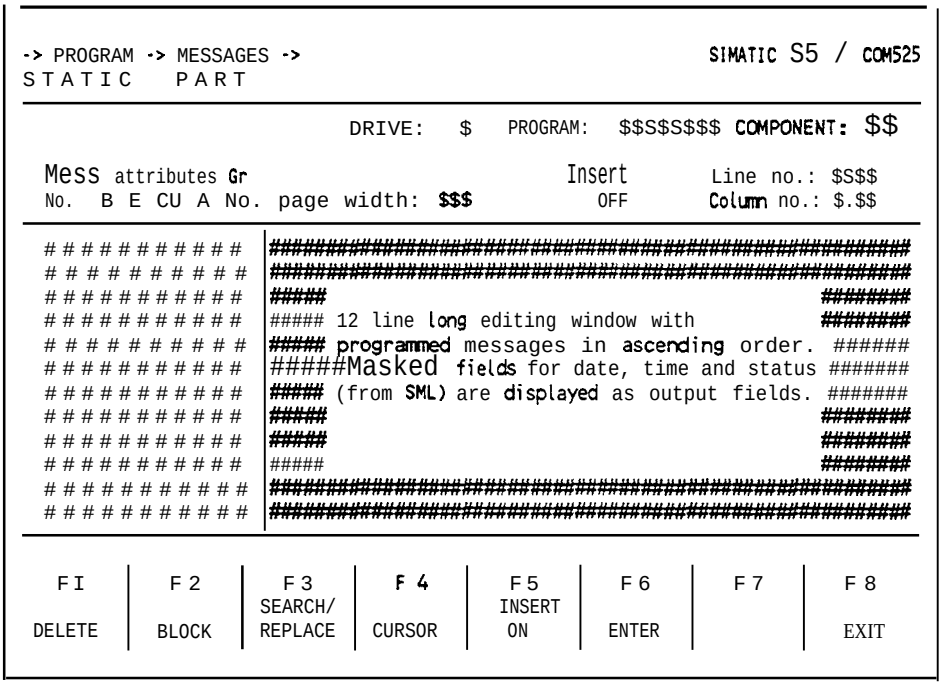
- 'Program from message no.: #/## (1-2047)'

  - Read in areas with a maximum of 70 successive messages. Specify the first message number of the area to be read in.

- 'Number of messages to be loaded: ## (max. 70)'

Here you enter the number of messages to be read in from the already specified message number. The same number of already programmed messages will be read in as you have specified. There can be gaps between the message numbers. The messages will be read in in ascending order of the message numbers. If there are less messages present than you specified then only those present will be read in.

You call the 'STATIC PART' mask with the function key F1 (STAT. PART) in the 'MESSAGES' mask.



5.3.2 Description oft&? Editor

In COM 525 a text editor is available for the programming of messages. An editing window is displayed on the PG screen, 12 messages long and 80 characters wide.

When you call the mask it may appear as follows

- It is empty if no message has been programmed in the selected area.
- It contains the first 12 messages in the specified area, or all the programmed messages if fewer than 12 have been programmed.

If you reach the margin of the editing window, the screen is automatically 'scrolled', i.e. the editing window is moved vertically by one message, i.e. if the upper edge is reached by one message upwards and if the lower edge is reached by one message downwards.

Remember the following points when you edit the static parts:

- Edit the static texts in the way in which you want them printing out on the PT88 or PT89.
- Mark the field within the static text of a message line in which the dynamic parameter is to be printed out with a string of # or $ to reserve the required number of characters.

  !Important!
  When marking a dynamic field use only # or $, but not both.

  Correct: ######## Wrong: ####$$$$

  The # and $ characters must not be used for other purposes, i.e. they must not be part of your message.

- Mark a maximum of one dynamic field in each message line.
- Assign the print attributes individually for each message if you wish. These then apply for all static parts of this message.
  The attributes for the dynamic part are set when this is programmed (see 5.3.2).
- Allocate each message line to one of 16 message groups if you wish.

5.3.3 Editor Functions

The message editor makes available a number of functions to make editing easier. These are called up using function keys. Depending on which function is called, submenus are displayed with which you can start the subfunctions.

The following functions are available:

o F1 in the 'STATIC PART' mask:

editor function DELETE;
branch to a submenu:

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
	WORD	LINE	BLOCK				ASORT

Words, lines or even whole blocks of your edited text can be deleted.

The following subfunctions can be started using the submenu displayed:

F2: WORD
The word on which the cursor is positioned is deleted.

F3: **LINE**
The line on which the cursor is positioned is deleted.

F4: BLOCK
The block is deleted which you markedby setting startandendmarkers.

• **F2** in the 'STAT. **PART'** mask:

editor function BLOCK,
branch to a **submenu**:

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
SET START	SET END	COPY	MOVE				ABORT

You can define one or more successive message lines as a block. You can then manipulate this block as a unit, i.e. copy the **whole** block or delete the whole block etc.

The following **subfunctions** can be started using the displayed submenu:

F1: SET START
The start of the block is marked. The start of the block is always the start of the message in which the cursor is currently located. The cursor itself need not be at the start of the message. The start of the block is marked to the left of the message line with the letter S on the screen.

F2: SET END
The end of the block is marked. The end of the block is always the end of the message on which the cursor is currently positioned. The cursor itself need not be at the end of the message. The **block** end is marked by the letter E to the left of the message line on the screen.

!Important! - You can only **mark** block end after you have marked the start of the block.
- The start and end of the block can be **marked on** the same message. The block then simply **consists** of one message.

F3: COPY

The marked block as described above is copied in before the line in which the cursor is currently positioned. The numbers of the messages copied are set as follows:

- The first copied message receives the number of the first non-programmed message after the cursor position.
- The second message copied receives the number of the second non-programmed message after the cursor position etc.

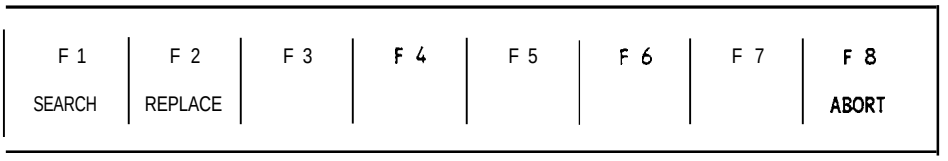
The original block is not deleted. The block *now* exists twice.
If the cursor is positioned on a message which has not yet been programmed (empty except for the masked fields for date and/or time and/or status) the first message copied will receive the number of this non-programmed message.

F4: MOVE

The marked block as described above is moved to the position immediately before the line on which the cursor is positioned. The original block is deleted. The block only exists once.
The message numbers will have changed as described under 'F3: COPY'.

• **F3** in the 'STATIC PART' mask:

editor function SEARCH
branch to a submenu.



You can search for characters or strings of characters in the edited text and replace these by other characters or strings of characters. The following subfunctions can be started from the submenu displayed:

F1: SEARCH

The 'search line' is displayed above the editing window. Here you can enter:

- A maximum 12-character long string which is to be searched for in the text.
- Whether you wish to search forwards or backwards from the cursor.

When you exit the search line, e.g. after it is completed, the editor searches for the string of characters you entered. If the string exists in the text the cursor is positioned on the character stringthathasbeen found. Ifthestringisnot contained in the text, an error message is displayed. The cursor is then positioned in the original position.

F2: REPLACE

Above the editing window the 'replace line' is displayed. Here you can enter:

- A maximum 12-character long string of characters to be searched for in the text,
- A maximum 12-character long string to be inserted in place of the string which has been searched for.
- Whether the string is to be searched forwards or backwards from the cursor.

- The number of times the string is to be searched for and replaced.
- Whether an operator acknowledgement is required before the string is replaced.

When you exit the 'replace line' e.g. after it is complete, the editor searches for the specified string of characters in the text and replace this with the second string of characters. Depending on what has been specified, this may occur several times.

F4 in the 'STATIC PART' mask:

editor function CURSOR,
branch to a submenu:

F 1	F 2	F3	F4	F 5	F 6	F ?	F 8
LEFT MARGIN	RIGHT MARGIN	INPUT FIELDS		MESSAGE NUMBER	CONTINUE		ABORT

With these functions you can position the cursor at any point in the text.

The following subfunctions can be started using the displayed submenu:

F1: LEFT MARGIN
The cursor is positioned on the left margin of the line in which it was previously located.

F2: RIGHT MARGIN
The cursor is positioned on the right margin of the line in which it was previously located.

F3: INPUT FIELDS
A switchover is made so that yu can enter the message number, attributes and group number. The cursor is positioned on the message number in the line in

which it was previously located. You can now program only in these input fields. To return to the editing window press F4 (TO ED. WINDOW).

F5: MESSAGE NUMBER
An input field is displayed above the editing window in which you can enter the number of the message on which the cursor is to be positioned. The cursor is then positioned at the start of the specified message, if the message is in the editing buffer. Otherwise an error message is output.

F6: CONTINUE
A switchover is made to the second submenu. Here further cursor functions are made available:

F 1	F 2	F3	F4	F 5	F 6	F 7	F 8
BLOCK START	BLOCK END	TEXT START	TEXT END		CONTINUE		ABORT

Fl: BLOCK START
The cursor moves to the start of the block. This must already have been marked.

F2: BLOCK END
The cursor moves to the end of the block. This must alxeadyhavebeenmarked.

F3: TEXT START
The cursor moves to the start ofthetextinthe editing buffer.

F4: TEXT END
The cursor movestotheendof the text in the editing buffer.

• **F3** in the 'STATIC PART' mask:

editor function SEARCH
branchtoasubmenu.

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
SEARCH	REPLACE						ASORT

You can search for characters or strings of characters in the edited text and replace these by other characters or strings of characters. The following subfunctions can be started from the submenu displayed:

F1: SEARCH
The 'search line' is displayed above the editing window. Here you can enter:
- A maximum 12-character long string which is to be searched for in the text.
- Whether you wish to search forwards or backwards from the cursor.

When you exit the search line, e.g. after it is completed, the editor searches for the string of characters you entered. If the string exists in the text the cursor is positioned **on the character stringthathasbeen found. Ifthestringisnot contained in the text, an error message is displayed. The cursor is then positioned in the original position.**

F2: REPLACE
Above the **editing window the 'replace line' is displayed. Here you can enter:**
- **A maximum 12-character long string of characters to be searched for in the text,**
- **A maximum 12-character long string to be inserted in place of the string which has been searched for.**
- **Whether the string is to be searched forwar& or backwards from the cursor.**

- The number of times the string is to be searched for and replaced.
- Whether an operator acknowledgement is required before the string is replaced.

When you exit the 'replace line' e.g. after it is complete, the editor searches for the specified string of characters in the text and replace this with the second string of characters. Depending on what has been specified, this may occur several times.

F4 in the 'STATIC PART' mask:

editor function CURSOR,
branch to a submenu:

F 1 LEFT MARGIN	F 2 RIGHT MARGIN	F3 INPUT FIELDS	F4	F 5 MESSAGE NUMBER	F 6 CONTINUE	F 7	F 8 ABORT

With these functions you can position the cursor at any point in the text.

The following subfunctions can be started using the displayed submenu:

F1: LEFT MARGIN
The cursor is positioned on the left margin of the line in which it was previously located.

F2: RIGHT MARGIN
The cursor is positioned on the right margin of the line in which it was previously located.

F3: INPUT FIELDS
A switchover is made so that yu can enter the message number, attributes and group number. The cursor is positioned on the message number in the line in

which it was previously located. You can now program oniy in these input fields. To return to the editing window press F4 (TO ED. WINDOW).

F5: MESSAGE NUMBER
An input field is displayed above the editing window in which you can enter the number of the message on which the cursor is to be positioned. The cursor is then positioned at the start of the specified message, if the message is in the editing buffer. Otherwise an error message is output.

F6: CONTINUE
A switchover is made to the second submenu. Here further cursor functions are made available:

F 1	F 2	F3	F 4	F 5	F 6	F 7	F 8
BLOCK START	BLOCK END	TEKT START	TEXT END		CONTINUE		ABORT

F1: BLOCK START
The cursor moves to the start of the block. This must already have been marked.

F2: BLOCK END
The cursor moves to the end of the block. This must already have been marked.

F3: TEXT START
The cursor moves tothestart ofthetext in the editing buffer.

F4: TEXT END
The cursor moves totheendofthetext in the editing buffer.

F6: **CONTINUE**
You return to the first submenu.

- F5 in the 'STATIC **PART'** mask:

editor function INSERT (ON/OFF);
press this function key to switch the editor from insert on to insert off and vice-versa, so that

edited characters are inserted in the existing text (INSERT ON)
- edited characters overwrite existing characters in the text **(INSERT OFF)**

Default setting: 'INSERT ON'

- F6 in the 'STATIC PART' mask:

saves the static parts of the message lines which have been edited or changed.

The total edited text (contents of the editing buffer with maximum 99 messages) is entered so that all edited or changed messages are transferred to the external storage.

- **F8** in the 'STATIC **PART'** mask:

exits the editing **mask. You** return to the basic menu. If you have not already transferred the edited or changed messages to the external storage with F6 **(ENTER)** an acknowledgement **prompt will** be displayed.

Description **of the fields in the 'STATIC PART mask (5.3.1)**

Field	Field type keys allowed	Limit value	Alternatives
Page width	OUTPUT FIELD (from interpreter parameter assignment, see 3.6)		
Column no.	OUTPUT FIELD		
Mess. no.	numbers	1 - 2047	*
Attributes	Letters B=bold, E=expanded, C=compressed, U=underlined, A=acoustic signal		Y, N
Group number	numbers	0 - 15	

F7 (HELP) can be used to make an entry in the fields marked '*'

5.3.4 The **Dynamic Part of Messages**

You must have already completed the following fields in the 'MESSAGES' mask:

- 'Program from message no.: #### (1-2047)'

  Specify the first message number of the area to be read in. You can read in areas with a maximum of 1000 successive messages.

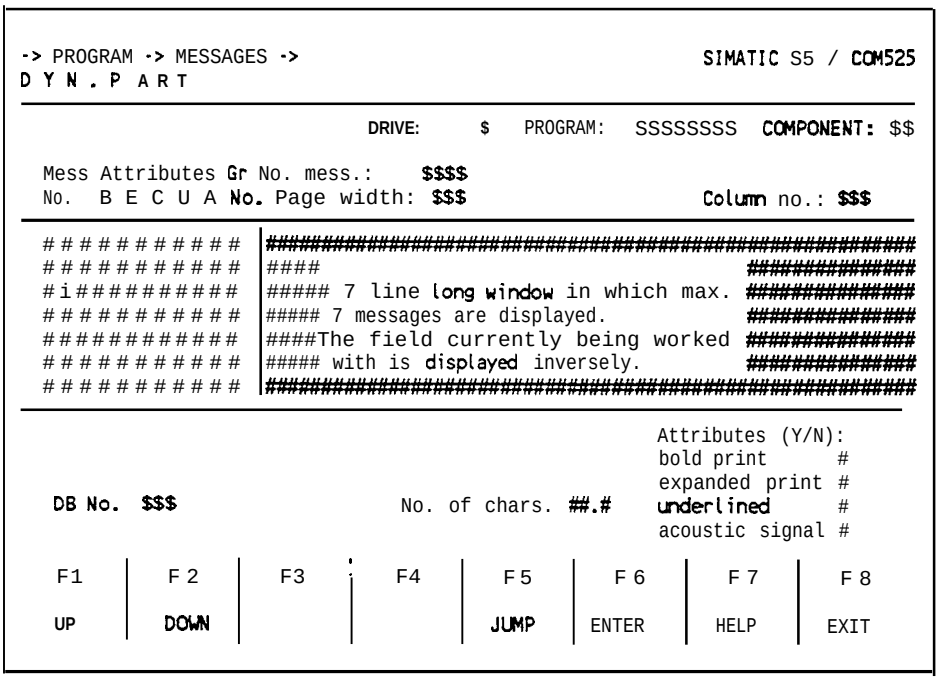
- 'Number of messages to be loaded: ## (max. 1000)[1]

  Enter the number of messages to be read in starting from the already specified message number.
  The same number of already programmed messages are read in that you specified, however, only those which include a dynamic field marked with reserve characters,
  There can be gaps between the message numbers.
  The messages are read in in ascending order of the message numbers.
  If there are fewer messages present than you specified, then only those present are read in.

Call the 'DYN. PART' mask with the function key F2 (DYN. PART) in the 'MESSAGES' mask.

Depending on the type of function ' UP/DOWN' or 'JUMP the mask appears as follows:

UP/DOWN



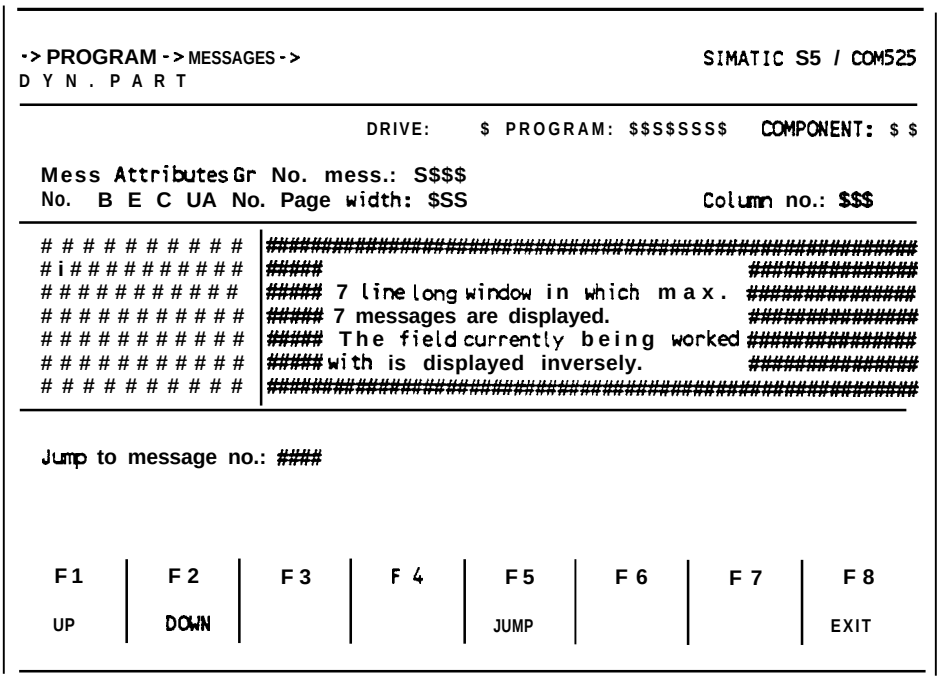
An editing window 7 lines long and 80 characters wide is displayed on the PG screen. It appears as follows after the mask has been called:

It displays the first seven messages which have already been programmed in the selected area or less than seven if a shorter area has been selected or if the area contains less than seven messages.
It inversely displays the dynamic field.

- Below the 7-line editing window, it also displays a submask. Here you can specify the number of characters and the print attributes for the current (inversely displayed) field.

!Important!
After you program a field, you must enter the values with F6 (ENTER). Otherwise a warning is displayed that you must acknowledge.

JUMP



With the following keys you can jump to fields in neighboring messages:

F1 (UP) to the field in the previous message
F2 (DOWN) to the field in the next message,

With F5 (JUMP) you can also jump to any field in any message. COM 525 prompts you to enter the required message number and field number. Specify these and then press F5 (JUMP) again.

F1: the field in the message before is processed.

F2: the field in the message following is processed.

F5: the submask 'JUMP' is processed or
the field in the specified message is processed,

F6: the dynamic part is saved on disk along with the static part

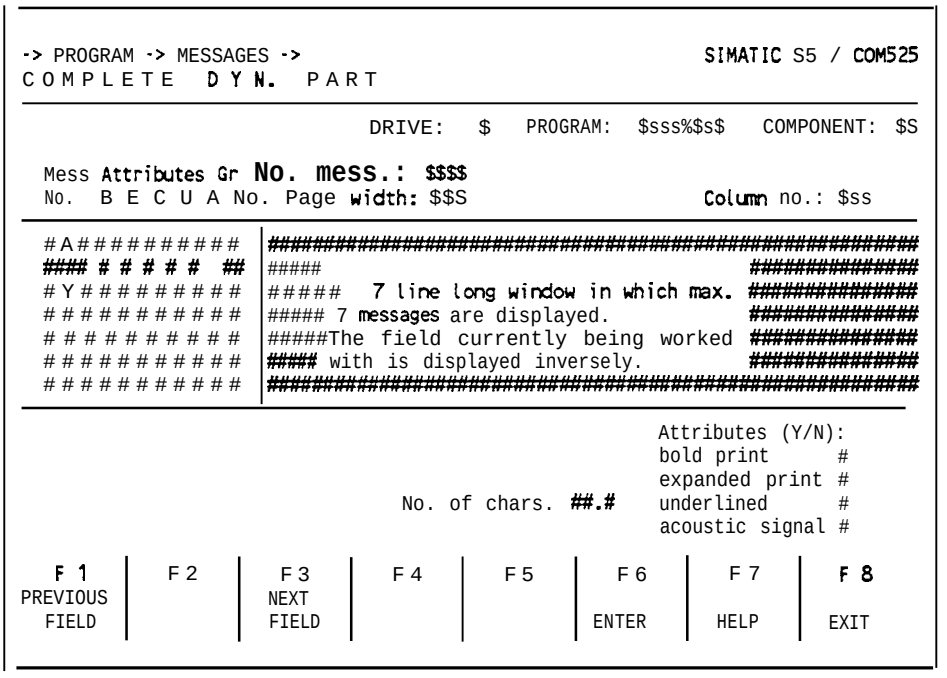
F8: abort:
the data are not entered, and
you return to the 'MESSAGES' mask.

Field	Field type keys allowed	Limit value	Alternatives
No. mess.	OUTPUT FIELD	1 - 1000	
Page width	OUTPUT FIELD (from interpreter parameter assignment, see 3.6)		
Column no.	OUTPUT FIELD		
Mess. no.	numbers	1 - 2047	
Attributes	Letters B=bold, E=expanded, C=compressed, U=underlined, A=acoustic signal		Y, N *
Gr. no.	numbers	0 - 15	
Bold, expanded, underlined, acoustic signal	letters		Y, N *
No. of chars.	numbers	01.0 - 80.9	
Jump to message no.	numbers	1 - 2047	

F7(HELP) can be used to make an entry in the fields marked '*'.

5.3.5 Completing the Dynamic Parts of Messages

Call the 'COMPLETE DYN. PART mask with the function key F4 (COMPLETE DYN. PART) in the 'MESSAGES mask.



The layout and distribution of the screen as well as the submasks forthe different types of data correspond to the 'DYN. PART' mask.

The mask is, however, different since not every dynamic field can be programmed but only the fields for which no dynamic part has as yet been programmed.

Only messages that have not had their dynamic fields programmed are read in and displayed in the editing window.

Once you have programmed such a field you must enter the input with F6 (ENTER) .

Once all the fields have been programmed, COM 525 indicates this with a message; it does not then indicate any further fields.

F1: the previous non-programmed field is completed

F3: the next non-programmed field is completed

F6: the dynamic part is saved on disk along with the static part

F8: abort:
the data is not entered,
return to the 'MESSAGES' mask

Field	Field type keys allowed	Limit value	Alternatives
Page width	OUTPUT FIELD (from interpreter parameter assignment, see 3.6)		
Column no.	OUTPUT FIELD		
Mess, no.	numbers	1 - 2047	
Attributes	Letters B=bold, E=expanded, C=compressed, U=underlined, A=acoustic signal		Y, N *
Gr. no.	numbers	0 - 15	
Bold, expanded, underlined, acoustic signal	letters		Y, N *
No. of chars.	numbers	01.0 - 80.9	

F7(HELP) can be used to make an entry in the fields marked '*'.

5.4 PC JOB

Call the 'PC JOB' mask with the function key F2 (PC job) in the 'PROGRAM' mask.

In the 'PC JOB' mask, you select which PC job you wish to program. The list of programmable PC jobs can be seen in the following mask. Depending on which selection is made the program will branch to different masks.

```
-> SELECTION -> PROGRAM ->                                   SIMATIC S5 / COM525
P C   J O B
--------------------------------------------------------------------------------
                         DRIVE:   $   PROGRAM:  $ss$$ss$   COMPONENT:  $$

  SML:            Sequential Message List
                  continuous output of the messages (with dynamic parts)
  UPDATE SML:     updating of dynamic data in the frames of the SML.

  CML:            Current Message List
                  output of all the messages currently on the CPU
  PSL:            Process Status List
                  output of process statuses with comments.
  CL:             Chained List
                  chained output of several process status lists (PSL).
  UPDATE GIB:     updating of group inhibit bits.

  NEW PAGE:       for form feed on PT88/PT89 printer.

  PAGE NO.=1:     set page numbering on PT88/PT89 printer to 1.

  F 1  | F 2    | F 3 | F 4 | F 5 | F 6      | F 7 | F 8
       | UPOATE |     |     |     |          |     |
  SML  | SML    | CML | PSL | CL  | CONTINUE |     | EXIT
```

F1: **programming** a PC job for output of the **sequential message list**:
branch to the 'SEQ. MESSAGE LIST' mask

!Important! Before messages are programed for the first time (see 5.3), the SML (position and structure of date, time and status for all messages) must be specified.

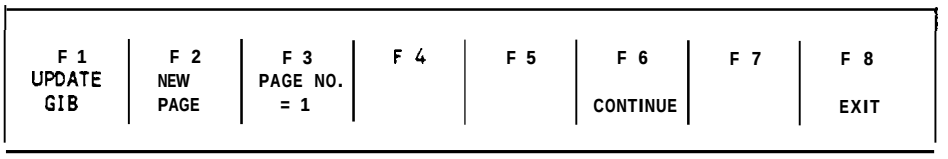
F2: programming a PC job for **updating the frames of the sequential message list,**
branch to the 'UPDATE SML' mask
(see 5.4,2),

F3: programming a PC job for output of the current message **list,**
branch to the 'CUR .MESSAGE LIST ' mask
(see 5.4.3).

F4: programming a PC job for output of a process **status** list
branch to the 'PROCESS STATUS LIST' mask
(see 5.4.4).

F5: programmingg a PC job for output of several **chained** process **status** lists
branch to the 'CHAINED LISTS ' mask
(see 5.4.5).

F6: output of a menu with further PC jobs:



F1: programming **a** PC job for **updating the group inhibit bits**
branch to the *UPDATE GIB' mask
(see 5.4.6).

F2: programming **a Pc job** to initiate a **form feed**
branch to the 'NEW PAGE' mask
(see 5.4.7).

F3: programming **a PC** job to **reset the page numbering**
branch to the 'RESET PAGE NO. ' mask
(see 5.4.8).

F6: return to the previous menu

F8: abort:
return to the 'PROGRAM' mask

Note: If several PC jobs were programmed with the same job number only the last job stored can be executed. With F5 (check job number) in the SPECIAL FUNCTIONS mask, you can check whether job numbers have been used more than once.

5.4.1 Sequential Message List (SML)

Call the 'SEQ. MESSAGE LIST' mask with the function key F1 (sML) in the 'PC JOB' mask.

Before you can program a message, the global data valid for all messages must be specified as follows:

- Number of the PC job, with which the CPU is to send the message records to the CP 525.
- Name of the list header and/or list trailer which is to be printed out at the beginning or end of a page filled with messages.
- The position, status and structure of the date, time and message status for all messages. These specifications determine the position and length of the masked fields that cannot be changedwhenprogramming the individual messages (see Section 5.3).

```
-> PROGRAM -> PC JOB->                                  SIMATIC S5 / COM525
SEQ. MESSAGE LIST

                                  DRIVE:    $   PROGRAM:  $$$$$$$$   COMPONENT: $$

  List name:      SEQ.MESSAGE
  With the job number: ###
  messages are output as events take place.
  List header is frame: ############
  List trailer is frame: ############
  Form feed on PT88/PT89 as start of list (Y/N) #
  Position and format of date, time, status is as below for all messages:
                        Attributes  Order
                Column   B E U A
    Date:         ###    # # # #    ##$##$##
    Time:         ###    # # # #    ##$##$##
  ------------------------------------------------------------
    Status:       ###                Text (enclose in inverted coronas)
       coming:            # # # #    ############
       going:             # # # #    ############
       acknowledged:      # # # #    ############

 F1 | F 2 | F 3  | F 4 | F 5 | F 6  | F ? | F 8
    |     | FRAME |     |     | SAVE | HELP | EXIT
```

F3: generate a frame.
You branch directly to the mask for programming frames (see 5.5) .
You can then program or change the frames (list header and/or list trailer) that are to be printed out with the sequential message list (you can program other frames at this point).

F6: save the PC job on disk

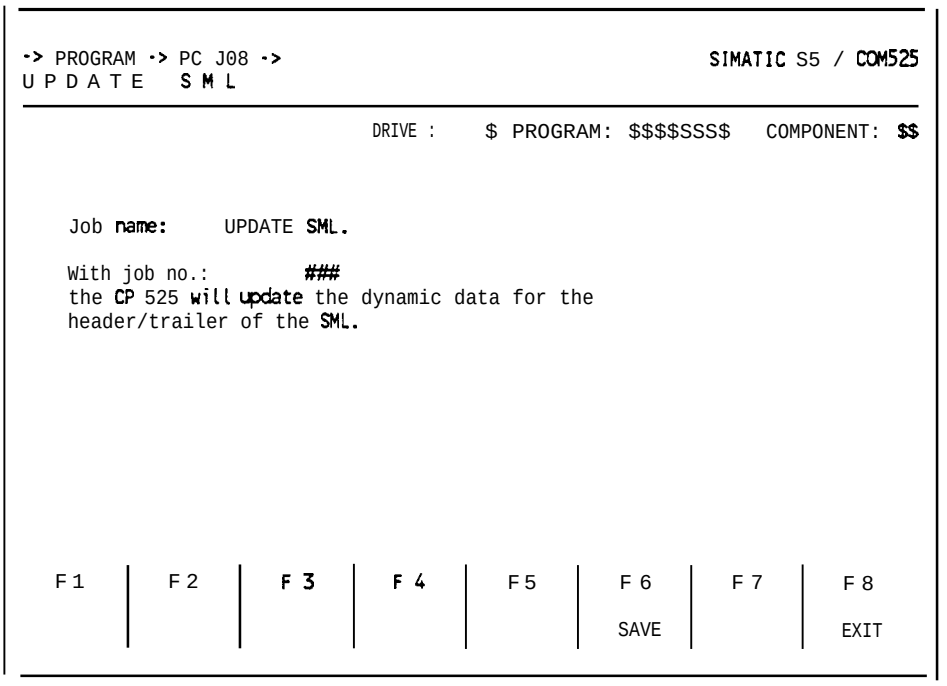
F8: abort:
the data is not entered; return to the 'PC JOB' mask

Field	Field type keys allowed	Limit value	Alternatives
Job no.	numbers	1 - 199	
List header/ list trailer	element name (see 5,1)		*
Form feed	letters (default 'Y')		Y, N *
Column	numbers	1 - 136	
Attributes	letters B=bold E=expanded, U=underlined, A=acoustic signal		Y, N *
Order - date	letters (default 'DD.MM.YY')		DD,MM,YY *
Order - time	letters (default 'HH:MM:SS')		HH,MM,SS *
Text	any (defaults)		

F7 (HELP) can be used to make an entry in the fields marked '*'.

5.4.2 Update SML

Call the *UPDATE SML' mask with the function key F2 (UPDATE SML) in the 'PC JOB' mask.



F6: the PC job is savec on the storage medium

F8: abort:
return to the 'PC JOB' mask

Field	Field type keys allowed	Limit value	Alternatives
Job no.	numbers	1 - 199	

5.4.3 Current Message List (CML)

Call the *CURRENT MESSAGE LIST' mask with the function key F3 (CML) in the 'PC JOB' mask.

```
-> PROGRAM -> PC JOB ->                                  SIMATIC S5 / COM525
CURRENT  M E S S A G E  L I S T
                                DRIVE: $     PROGRAM:  $ss$ss$$   COMPONENT: S$

   List name:      CUR.MESSAGE

   With the job no.: ###
   the output of the current message list is started.

   List header is frame: ############
   List trailer is frame: ############

   Output of messages possible while CML is being output        (Y/N) #
   Form feed on PT88/PT89 at start of list                      (Y/N) #

 F 1 | F 2 | F 3  | F 4 | F 5 | F 6  | F 7  | F 8
     |     | FRAME |     |     | SAVE | HELP | EXIT
```

F3: generate a frame.
You branch directly to the mask for programming frames (see 5.5).
You can then program *or change* the frames (list header and/or list trailer) which are to be printed out with the sequential message list (you can program other frames at this point).

F6: save the PC job on disk.

F8: abort:
the data is not entered; return to the 'PC JOB' mask

Field	Field type keys allowed	Limit value	Alternatives
Job no.	numbers	1 - 199	
List header/ list trailer	element name (see 5.1)		*
Output of messages. , .	letters (default 'Y')		Y, N *
Form feed . . .	letters (default 'Y')		Y, N *

F7 (HELP) can be used to make an entry in the fields marked '*'.

5.4.4 Process status List (PSL)

Process status lists include the following:

- static parts, i.e. fixed texts
- dynamic parts, i.e. variable data, which can be distributed at any point within the static parts

Static parts

You determine the structure and contents of the static parts when programming on the PG with the text editor. They can then no longer be changed on the CP 525,

Dynamic parts

When programming the static parts you specify where the dynamic parts are to be inserted as follows:

- per list line maximum 40 dynamic fields
- per PSL maximum; approximately 1400 dynamic fields (dependent on data type)

The number of reserve characters inserted in the static text for a dynamic field is used as the default for the number of characters when programming the dynamic part.

The following information can be freely programmed for the dynamic data:

- source (address in the data block of the CPU)
- data type (bit, byte, characters, 16-bit FXP etc.)
- number of characters in the list
- printer attributes (e.g. bold, expanded etc.)
- position in list

Process status lists and frames are programmed in two steps:

1. First the static parts are programmed. Press F1 (STAT. PART) in the 'PROCESS STATUS LIST' to call the 'STATIC PART' mask.

2. The dynamic parts for the static parts which have already been programmed are specified. Press F2 (DYN. PART) in the 'PROCESS STATUS LIST' to branch to the 'DYN. PART' mask.

In addition, you can call the 'COMPLETE DYN. PART' mask with F4 (COMPLETE DYN. PART) in the 'PROCESS STATUS LIST' mask. When you use this function, dynamic fields that have not yet been programmed within the already programed static parts are searched for.

Call the 'PROCESS STATUS LIST' mask with the function key F4 (PSL) in the 'PC JOB' mask,

```
-> PROGRAM -> PC JOB ->                                    SIMATIC S5 / COM525
P R O C E S S   S T A T U S   L I S T
--------------------------------------------------------------------------------
                                 DRIVE:   $   PROGRAM : $$$$$$$$  COMPONENT:  $S

   Name of list: ###########        Name to be stored: ###########

   With the job no.: ###
   the output of the process status list with the
   name selected above will be started.

   The dynamic data are on CPU no.: #
   in data block          DB no.: ###

   List header is frame: ###########
   List trailer is frame: ###########

   Output of messages possible while the PSL is being output     (Y/N) #
   Form feed on PT88/PT89 at start of list                       (Y/N) #

  F 1    |  F 2   |  F 3   |  F 4    |  F 5   |  F 6   |  F 7   |  F 8
 STAT.   | DYN.   |        |COMPLETE |        |        |        |
 PART    | PART   | FRAME  |DYN.PART |        | SAVE   | HELP   | EXIT
```

F1: change to the mode 'generating the static part of a process status list' ; branch to the 'STATIC PART' mask (see a) in this section)

F2: change to the mode "generating the dynamic part of a process status list'; branch to the 'DYN. PART' mask (see b) in this section)

F3: change to the 'programming frames' mode; branch to the 'FRAME' mask (see 5.5)

F4: change to the mode 'completing the dynamic part of a process status list' ;
branch to the 'COMPLETE DYN. PART' mask;
(see c) in this section)

F6: store the PC job on disk

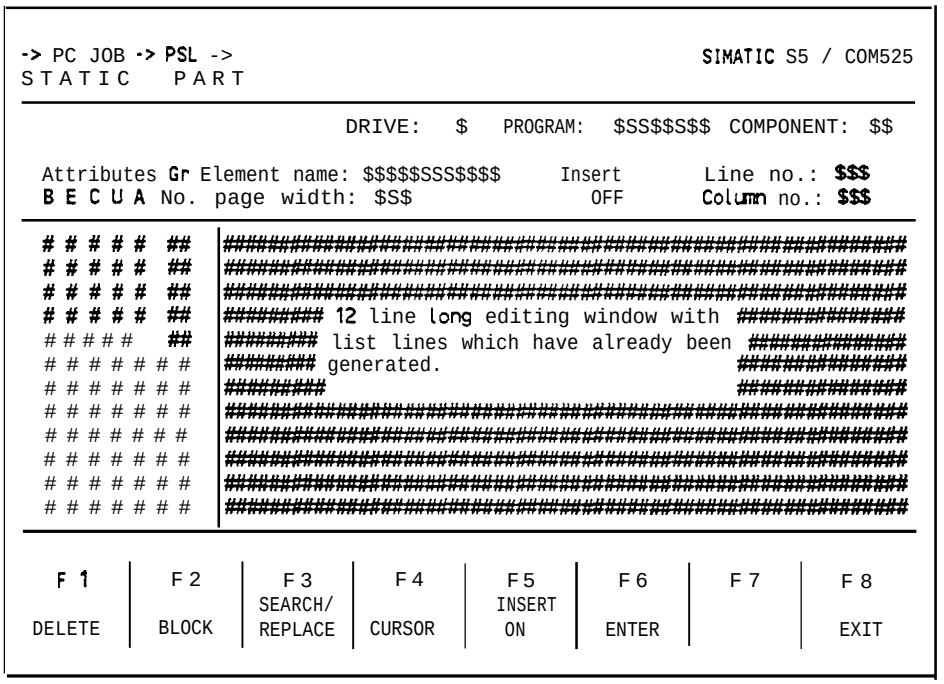
F8: abort:
data is not entered; return to the 'PC JOB' mask.

Field	Field type keys allowed	Limit value	Alternatives
List name	element name (see 5.1)		*
Name to be stored	element name (see 5.1)		
Job no,	numbers	1 - 199	
CPU no.	numbers	1 - 4	*
DB no.	numbers	1 - 255 (the DB numbers 1 and 2 are reserved for the system)	
List header/ list trailer	element name (see 5.1)		*
Output of messages . . .	letters (default 'Y')		Y, N *
Form feed . . .	letters (default 'Y')		Y, N *

F7(HELP) can be used to make an entry in the fields marked '*'.

a) Static part of a process status list

Call the 'STATIC PART' mask with the function key F1 (STAT. PART) in the 'PROCESS STATUS LIST' mask.



An editing window that is 12 lines long and 80 characters wide is displayed on the screen of the programmer.

When the mask is displayed, it can appear as follows:

- it is empty, if a new PSL is edited
- it contains the first 12 lines of a PSL which has already been programmed or less if the PSL is less than 12 lines long

You can edit a maximum of 99 list lines each with a maximum of 136 characters.

If the cursor reaches the margin of the editing window the screen is automatically 'scrolled'; i.e. the editing window is scrolled vertically, one message upwards if the lower margin has been reached and one message downwards if the upper margin is reached.

The same thing happens if the *cursor* reaches the right or left margin of the editing window (scrolls one word to the right or one word to the left).

You edit the static texts as you wish to have them output on the PT88 or PT89.

- Fields within the static text in which you want *to* print out dynamic data have their positions and lengths marked with the characters # or $ to reserve space for them.

!Important!

.

- One field must contain only # or $ but not both

  Correct: ######## Incorrect: ####$$$$

- Dynamic fields which are immediately adjacent to each other (i.e. there is no gap between them) must have different reserve characters, either # or $.

  Correct: $$$######$$ Incorrect: ############

- The characters #and $ must not be used for other purposes; i.e. they cannot be part of a PSL.

- A maximum of 40 dynamic fields can be marked in each line; however, there must not be more than approximately 1400 dynamic fields in any PSL. The maximum value depends on the type of variables pro-grammed for the fields, If status variables are used, the number of dynamic fields which can be programmed is considerably reduced since these can involve up to eight statuses. Such a variable is pro-grammed eight times and therefore occupies eight times more memory space.

- The print attributes can be set individually for each line. They then apply for all static parts in that line.

- **The attributes** for the dynamic parts are set when they are programmed (see b) in this section).
- Each list line can be assigned to one of 16 message groups.

Editor functions

A text editor is available in COM 525 for programming process status lists.

The editor makes available a number of functions in order to make editing easier. These are called up using function keys. Depending on which function is called, submenus will be displayed with which you can start the subfunctions.

The following functions are available:

- **F1** in the 'STATIC PART' mask:

  editor function DELETE,
  branch to a submenu:

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
	WORD	LINE	BLOCK				ABORT

Words, lines or even whole blocks of your edited text can be deleted.

The following subfunctions can be started using the submenu displayed:

F2: WORD
The word on which the cursor is positioned is deleted.

F3: LINE
The line on which the cursor is positioned is deleted.

F4: BLOCK
The block that you marked by setting start and end markers is deleted.

- **F2** in the 'STATIC PART' mask:

editor function BLOCK,
branch to a submenu:

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
SET START	SET END	COPY	MOVE				ABORT

You can define one or more successive message lines as a block. You can then manipulate this block as a unit, i.e. copy the whole block or delete the whole block etc.

The following subfunctions can be started using the displayed submenu:

F1: SET START
The start of the block is marked. The start of the block is always the start of the text line in which the cursor is currently located. The cursor itself need not be at the start of the line.
The start of the block is marked to the left of the text line with the letter S on the screen,

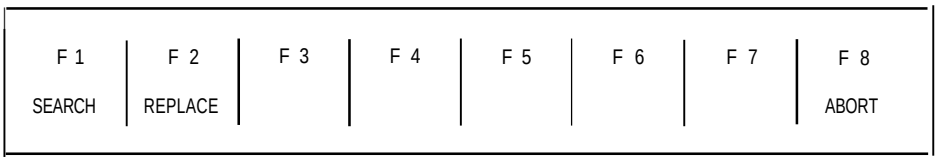
F2: SET END
The end of the block is marked.
The end of the block is always the end of the text line on which the cursor is currently positioned. The cursor itself need not be at the end of the line.
The block end is marked by the letter E to the left of the text line on the screen.

!Important! - You can only mark block end after you have marked the start of the block.
- The start and end of the block can be marked on the same text line. The block then simply consists of one line.

F3: COPY
The block marked as described above is copied in before the line in which the cursor is currently positioned.
The original block is not deleted. The block now exists twice.

F4: MOVE
The block marked as described above is moved to the position immediately before the line on which the cursor is positioned.
The original block is deleted, The block only exists once.

• **F3** in the 'STATIC PART' mask:
editor function SEARCH
branch to a submenu:



You can search for characters or strings of characters in the edited text and replace these by other characters or strings of characters. The following subfunctions can be started from the submenu displayed:

F1: SEARCH
The 'search line' is displayed above the editing window. Here you can enter:
- a maximum 12-character long string which is to be searched for in the text
- whether you wish to search forwards or backwards from the cursor

When you exit the search line, e.g. after it is completed, the editor searches for the string of characters entered. If the string exists in the text, the cursor will be positioned after the character string which has been found. If the string is not contained in the text an error message is displayed. The cursor is then positioned in the original position.

F2: REPLACE
Above the editing window the 'replace line' is displayed. Here you can enter:
a max. 12-character long string of characters to be searched for in the text,
- a max. 12-character long string to be inserted in place of the string which has been searched for.
- Whether the string is to be searched forwards or backwards from the cursor.

- How often the string is to be searched for and replaced.
- Whether an operator acknowledgement is required before the string is replaced.

When you exit the 'replace line' e.g. after it is complete, the editor searches for the specified string of characters in the text and replacees this with the second string of characters. Depending on what has been specified, this may take place several times.

- F4 in the 'STATIC PART' mask:

editor function CURSOR,
branch to a submenu:

F 1 LEFT MARGIN	F 2 RIGHT MARGIN	F 3 INPUT FIELDS	F 4	F 5 LINE NUMBER	F 6 CONTINUE	F 7	F 8 ABORT

With these functions you can position the cursor at any point in the text.

The following subfunctions can be started using the displayed submenu:

F1: LEFT MARGIN
The cursor is positioned on the left margin of the line in which it was previously located.

F2: RIGHT MARGIN
The cursor is positioned on the right margin of the line in which it was preciously located.

F3: INPUT FIELDS
A switchover is made so that you can enter the attributes and group number. The cursor is positioned on the first attribute in the line in which it was previously located. You can now program only in these input fields.
To return to the editing window press F4 (TO ED. WINDOW) .

F5: LINE NUMBER
An input field is displayed above the editing window in which the number of the text line can be entered on which the cursor is to positioned.
The cursor is then positioned at the start of the specified line, providing this is in the editing buffer. Otherwise an error message is output.

F6: CONTINUE
A switchover is made to the second submenu. Here further cursor functions are made available:

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
'BLOCK START	BLOCK END	TEXT START	TEXT END		CONTINUE		ABORT

F1: BLOCK START
The cursor moves to the start of the block. This must alreadyhave been marked.

F2: BLOCK END
The cursor moves to the end of the block. This must already have been marked.

F3: TEXT START
The cursor moves to the start of the process status list.

F4: m END
The cursor moves to the end of the process status list.

F6: CONTINUE
You return to the first submenu.

- **F5** in the 'STATIC PART' mask:

  editor function INSERT (ON/OFF);
  press this function key to switch the editor from insert on to insert off and vice-versa, so that

  - edited characters are inserted in the existing text (INSERT ON)
  - edited characters overwrite existing characters in the text (INSERT OFF)

  Default setting: 'INSERT ON'

- F6 in the 'STATIC PART' mask:
  saves the static parts of the text lines which have been edited or changed.
  The total edited text (static part of the edited PSL) is entered. This means that all PSLs are transferred to the external storage.

- **F8** in the 'STATIC PART' risk:
  exits the editing mask. You return to the basic menu. If you have not already transferred the edited PSL to the external storage with F6 (ENTER), an acknowledgement prompt will be displayed.

Description of the fields in the 'STATIC PART' mask

Field	Field type keys allowed	Limit value	Alternatives
Attributes	letters B=bold, E=expanded, C=compressed, U=underlined, A=acoustic signal		Y, N *
Gr no.	numbers	0 - 15	
Line no.	OUTPUT FIELD		
Page width	OUTPUT FIELD (from interpreter - parameter assignment see 3.6)		
Column no.	OUTPUT FIELD		

F7 (HELP) can be used to make an entry in the fields marked '*'.

b) **Dynamic part of a process status list**

You call the 'DYN. PART' mask with the function key F2 (DYN. PART) in the 'PROCESS STATUS LIST' mask.

An editing window is displayed on the screen of the PG. This is 7 lines long and 80 characters wide and appears as follows after the mask has been called.

It displays the first seven lines of a PSL which has already been programed or less if the PSL has less than seven lines.

The first dynamic field is displayed inversely.

Below the seven-line editing window, a submask is displayed that is laid out depending on the type of dynamic variable. There are various submasks for:

- process variable
  (always displayed if there are fields which have not yet been programmed)
- date/time
  (source is the CP 525 clock)
- date/time
  (source is the &tab Lock)
- process stares
- page number

!Important! When you have completely programmed a field you must enter the values with F6 (ENTER). Otherwise a warning is displayed which you must acknowledge.

You can jump in any direction to any neighbouring field with

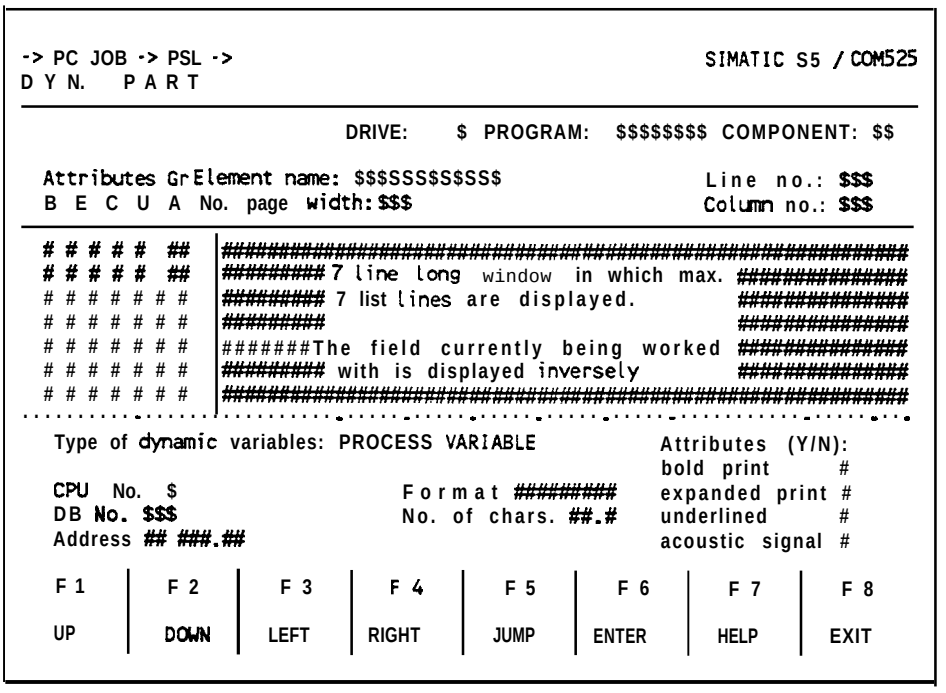
F1 (UP) to the first field in the line before
F2 (DOWN) to the first field in the next line
F3 (LEFT) to the next fieldto the left in the same line or to the last field in the previous line
F4 (RIGHT) to the next field to the right inthesamelineor to the firstfieldinthe next line

If there are no more fields in the direction you selected, this is indicated by a message.

By using F5 (JUMP) you can also jump to any field in any line. COM 525 then prompts you to specify the required line number and field number. Input these numbers and press F5 (JUMP) again,

Depending on the type of dynamic variables you selected the mask appears as one of the following masks:

PROCESS VARIABLE



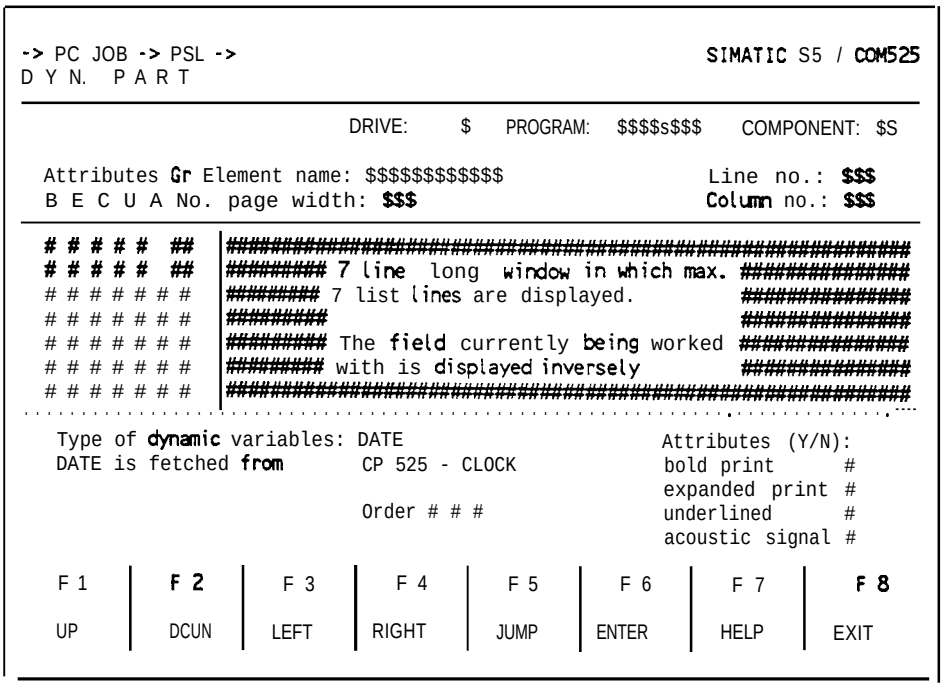
DATE / TIME

You can select the location from which the date and time are to be fetched:

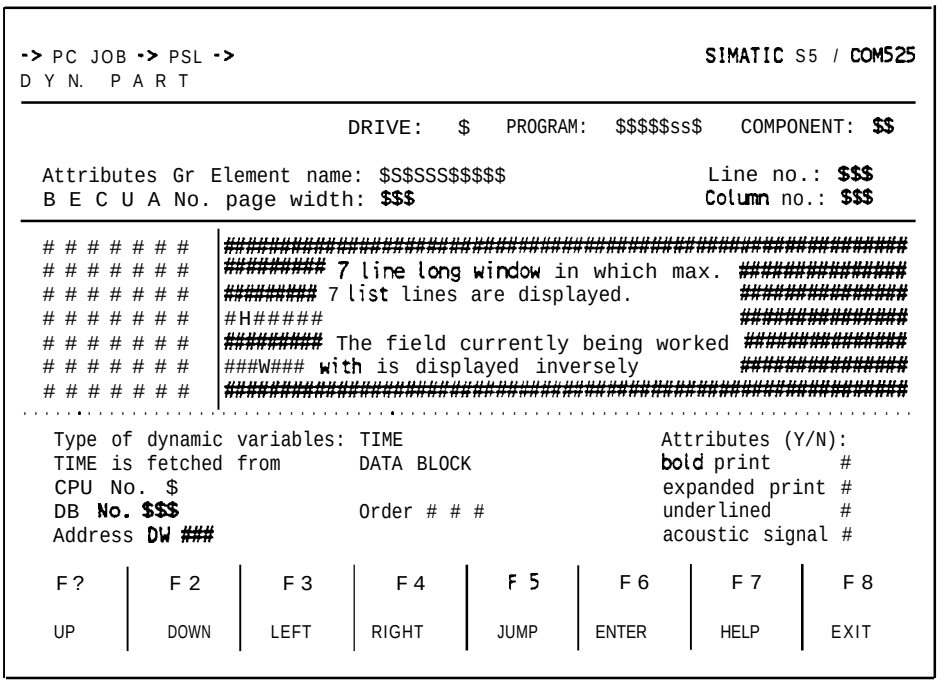
- the data block
- the internal CP 525 clock

Depending on the selected source, the mask appears as below:

DATE / TIME from the CP 525 clock
(example for date)

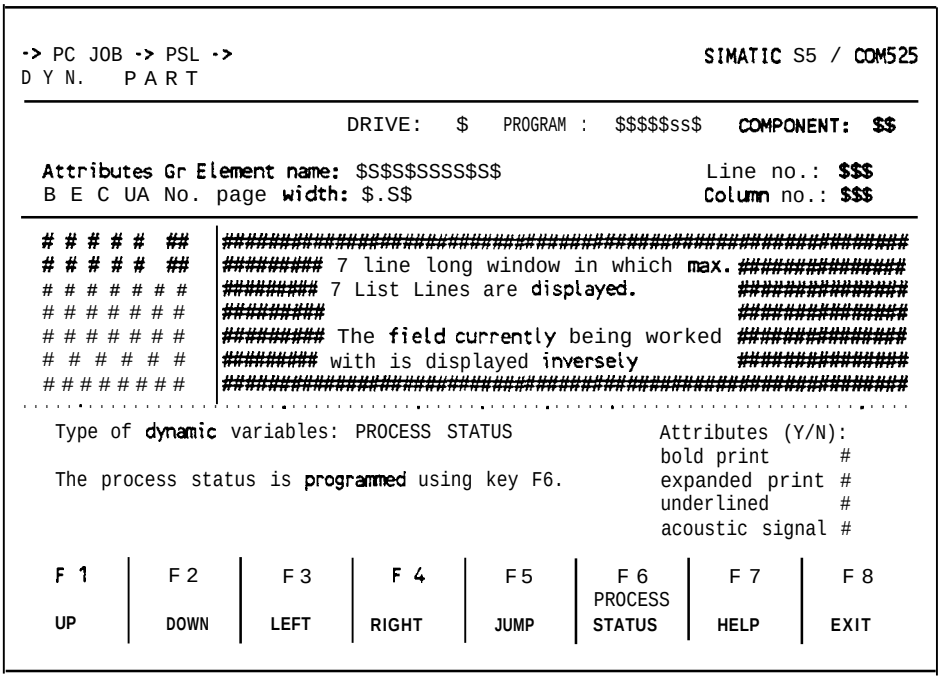


DATE / TIME from the data block
(example for time)



PROCESS STATUS

In the 'PROCESS STATUS' **mask the function key** F6 **is renamed** 'PROCESS sTATus'.



F6: branch to the 'PROCESS STATUS VAR.' mask

Process status - programming the text

You call the 'PROCESS STATUS VAR' mask by entering ' PROCESS STATUS in the ' type of dynamic variables field and with the function key F6 (PROCESS STATUS) in the 'DYN. PART mask.

The length of the input fields below the ' text corresponds to the number of characters reserved with $ or #, that you programed for this dynamic field in the static part of the PSL.

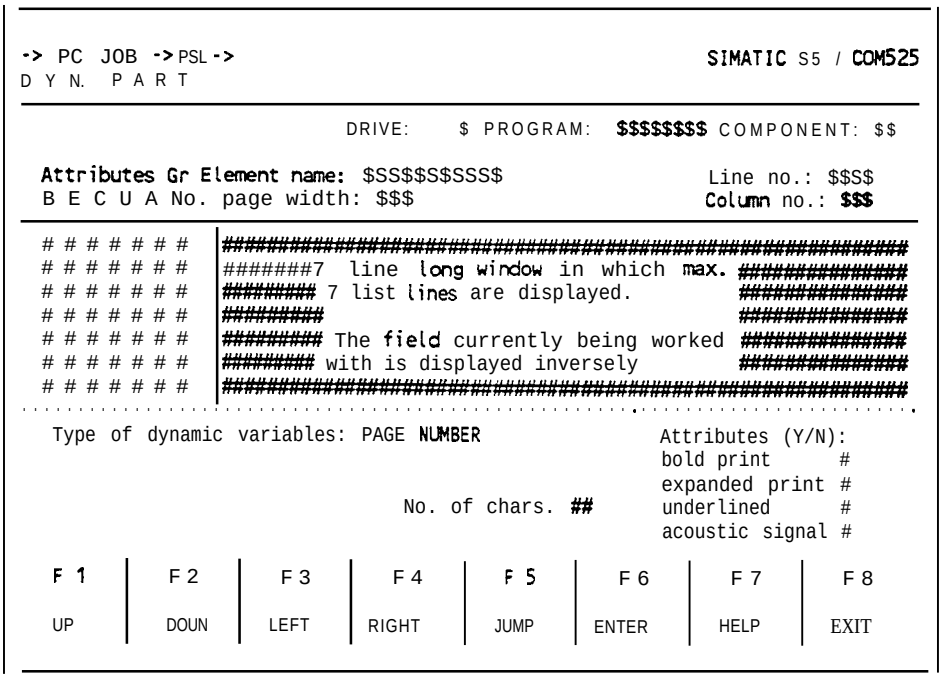
```
-> PROCESS STATUS LOG ->DYN. PART ->                    SIMATIC S5 / COM525
P R O C E S S   S T A T U S   V A R.

                                DRIVE:    $   PROGRAM :  $$$ss$s$   COMPONENT: $ $

Element name: $$$$$$$$$$$$

CPU no. $       DB no. $$$       Address: ## ###

     Attributes
Stat.  B E U A   Text
 o     # # # #   ##################################################
 1     # # # #   ##################################################
 2     # # # #   ##################################################
 3     ####      ##################################################
 4     ####      ##################################################
 5     ####      ##################################################
 6     # # # #   ##################################################
 7     # # # #   ##################################################


  F 1  |  F 2  |  F 3  |  F 4  |  F 5  |  F 6  |  F 7  |  F 8
       |       |       |       |       | ENTER | HELP  | EXIT
```

F6: all texts and attributes are entered, but are only saved along with the static and dynamic part with F6 (SAVE) in the 'PROCESS STATUS LIST ' mask.

F8: abort:
the data is not entered; return to the ' DYN. PART' mask.

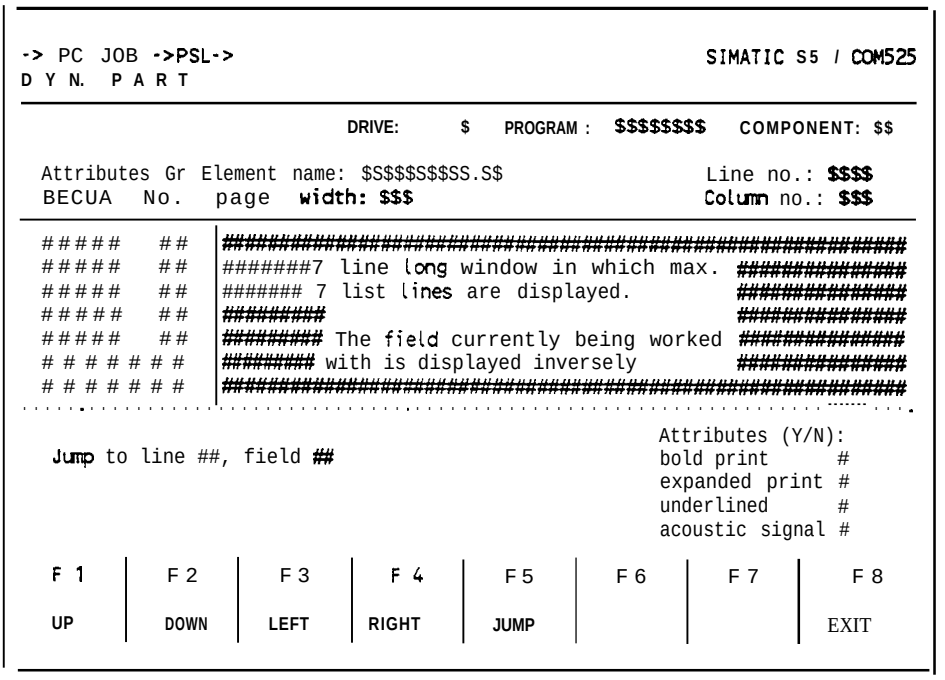
PAGE UMBER

The page number is entered in this field when the lists are output by the CP 525,



Jump to any field in a PSL

Press F5 (JUMP) in the 'DYN. PART' mask.



F5: a jump is made to the field specified in the fields **'line'** and 'field'.
If this field does not exist (line and/or field not present) an error message is displayed.

significance of the function keys in the 'DYN. PART' masks

F1: the first field of the previous line is processed.

F2: the first field of the next line is processed.

F3: the field to the left of the current field is processed (this can also be the last field of the line before).

F4: the field to the right of the current field is processed (this can also be the first field of the following line).

F5: processing of the submask 'jump to any field'
or the field specified in the specified line will be processed.

F6: the dynamic part is entered, but saved on disk with the static part only if you press F6 (SAVE) in the 'PROCESS STATUS LIST' mask or if you program a process status variable. Branch to the 'PROCESS STATUS VAR. ' mask.

F8: abort:
the data is not entered,
return to the 'PROCESS STATUS LIST' mask.

Description of the fields in the 'DYN. PART' masks and the 'PROCESS STATUS VAR. ' mask.

Field	Field type keys allowed	Limit value	Alternatives
Attributes	letters (default 'N')		Y, N
Gr no.	numbers	0 - 15	
Element name	OUTPUT FIELD		
Line no.	OUTPUT FIELD		
Page width	OUTPUT FIELD (from interpreter parameter assignment see 3 .6)		
Column no.	OUTPUT FIELD		
CPU no.	OUTPUT FIELD	1 - 4	
Format	letters (default '16BIT FXP')	(see Table 2)	
DB no.	OUTPUT FIELD	1 - 255	
No. of chars	numbers (default: no. of reserve characters)	01.0 - 80.9	

F7 (HELP) can be used to make an entry in the fields marked '*'.

Field	Field type keys allowed	Limit value	Alternatives
Address = memory access mode and DB address	letters (default 'DW')		DW, DL, DR, DD, BI *
	numbers	000.00 - 255.15	
bold/expanded underlined/ acoustic signal	letters (default 'N')		Y, N *
DATE/TIME is fetched from	letters (default 'CP 525 CLOCK')		CP 525 CLOCK DATA BLOCK *
Order	letters (default 'DMY' or 'HMS')		D, M, Y, or H, M, S *
Type of dyn. variables	letters (default 'PROCESS VARIABLE')		DATE, TIME, PROCESS VARIABLE, PROCESS STATUS, PAGE NUMBER *
Text	any		

F? (HELP) can be used to make an entry in the fields marked '*'.

Field	Field type keys allowed	Limit value	Alternatives
Jump to line	numbers	1 - 99 (last list line)	
Field	numbers	1 - 40	

Note on the defaults: if the submask of the selected type has already been edited this data will be entered as defaults. An exception to this is the 'number of characters', which always has the number of reserve characters.

Note on 'number of characters' in the 'PAGE NUMBER' mask: if a number of characters greater than four is selected, the maximum four-digit page number is output right-justified in a field with the width 'number of characters'.

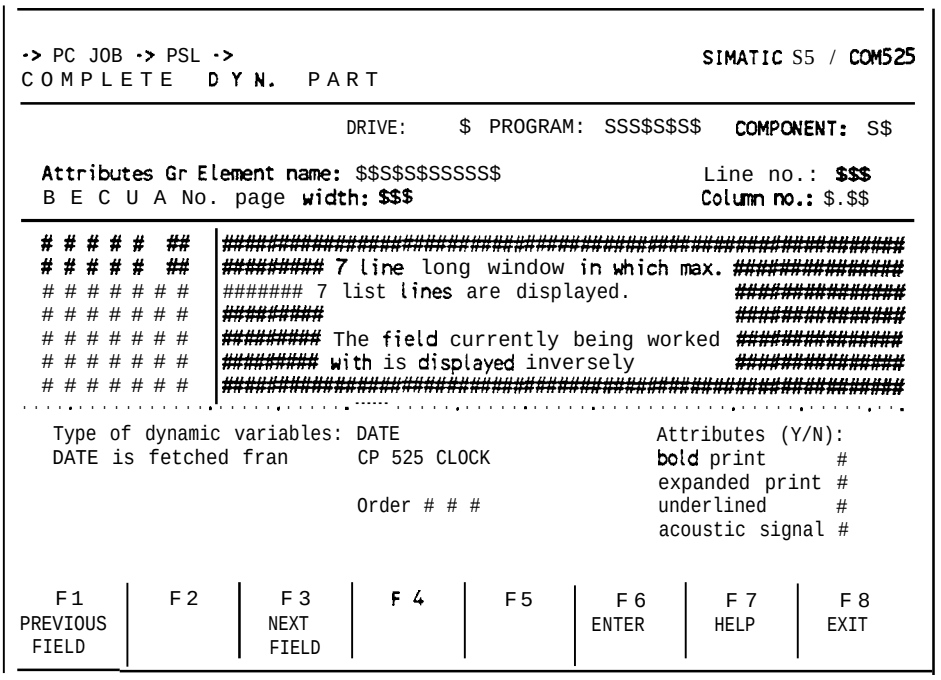
Format	Significance	Range of values	For memory access mode
31 NARY	1 bit binary	0.1	BI
CHAR	Character	ASCII	DR, DL
BYTE	1 byte binary	0 to 255	DR, OL
16BIT BCD	16 bit in BCD cede	-999 to +999	DW
16BIT FXT	16 bit in fixed point format	-32768 to +32767	DW
32BIT BCD	32 bit in BCD code	-9999999 to +9999999	DD
32BIT FXT	32 bit in fixed point format	-2^{31} to $2^{31}-1$	DD
32BIT FLP	32 bit in floating point format	-999999 * 10^{9} to -999999' 10^{-9} 0 +999999 * 10^{-9} to +999999 * 10^{9}	OD
TIMER	16 bit BCD code + time base (example: 154.2)	0 to +999 0 to 3	DW
COUNTER	16 bit BCD code	0 to +999	DW

Table 2: Format (all S5 formats can be specified)

When you enter the data with F6 (ENTER), COM 525 checks whether the format specified is compatible with the mode of memory access.

c) **Completing** the **dynamic** parts of a process status list

Call the 'COMPLETE DYN. PART' mask with the function key F4 (COMPLETE DYN. PART) in the 'PROCESS STATUS LIST' mask.



The layout and distribution of the screen as well as the **submasks** for the various data types correspond to the 'DYN. PART' mask for a process status list (see **b**).

There is, however, one major difference - you cannot **program** every dynamic field, but only those for which no dynamic part has yet been programmed.

The dynamic fields which have not yet been programmed are made available for **programming automatically** (displayed inversely) and you can position the cursor on these fields with

- **F1** (PREVIOUS FIELD)
- **F3 (NEXT FIELD)**

When you program one of these fields you must enter the specification with F6 (ENTER). Only then can you have the next field made available.

When all these fields are programmed, COM 525 outputs a confirming message. It then makes no further fields available.

F1: the last non-programmed field is processed

F3: the next non-programmed field is processed

5.4.5 Chained List (CL)

Call the 'CHAINED LISTS' mask with the function key F4 (CL) in the 'PC JOB' mask. You can have a maximum of eight chained process lists output on the printer.

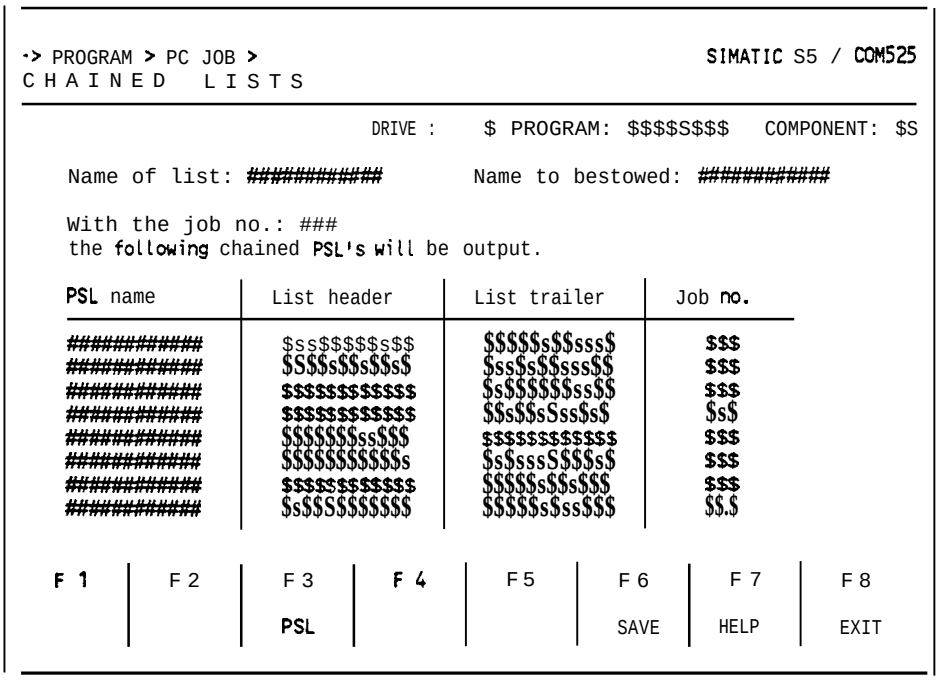
If a PSL was stored under a specified name, COM 525 displays the following programmed parameters for this PSL in the column next to the PSL name.

- Name of the list header (if no list header has been specified the field remains empty).
- Name of the list trailer (if no list trailer has been specified the field remains empty).
- Number of the PC job with which this PSL is to be startedby the CPU (if no job number has been programmed for this PSL the field remains empty).

The only lines in the mask that are completed are those for which you entered the name of a PSL. All other lines remain empty.

You can read in existing chained lists. The name of the chained list must be specified in the 'list name' field. You can modify this chained list and store it again under a different name. This new name must be entered in the field 'name to be stored'.

The 'name to be stored' field always has as a default the name specified in the 'list name' field.



F3: generate a **PSL.**
You branch directly to the mask for programming a **PSL** (see 5.4.4).
This allows you to program or modify process status lists **which you wish to have output as a chained list.**
Exiting this mask you return to the ‘chained lists’ mask.

F6: save the PC job on the storage medium.

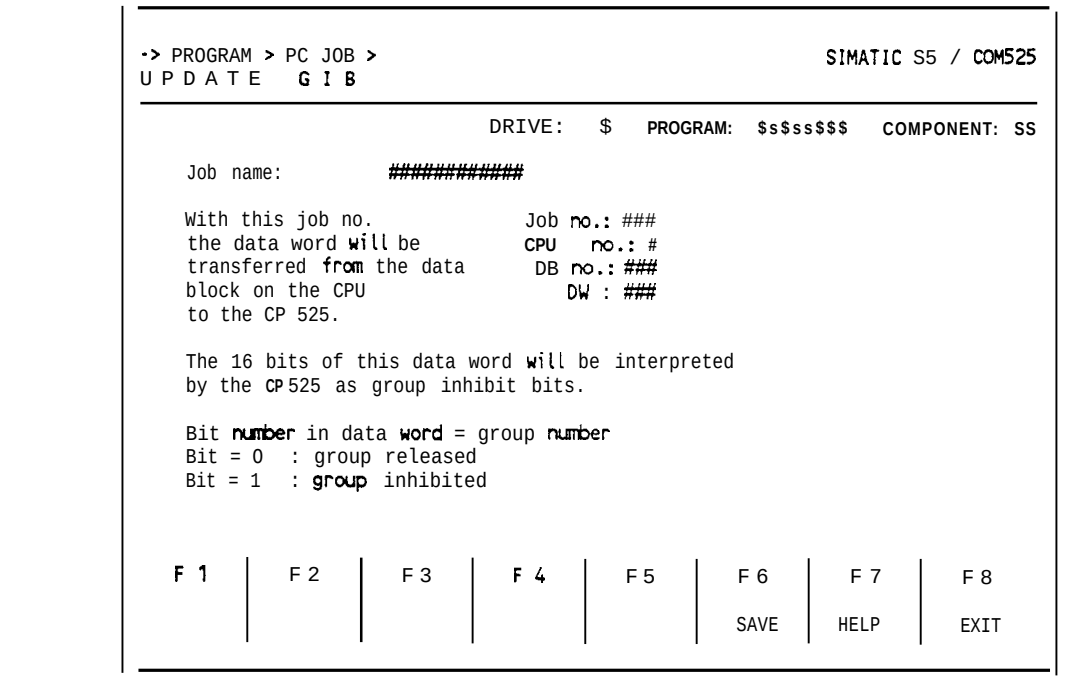
F8: abort:
the data is not entered;
return to the **'PC** JOB’ mask.

Field	Field type keys allowed	Limit value	Alternatives
Name of list	element name (see 5.1)		*
Name to be stored	element name (see 5.1)		
Job no.	numbers	1 - 199	
PSL name	element name (see 5.1)		*
List header	OUTPUT FIELD (as for corres. PSL)		
List trailer	OUTPUT FIELD (as for corres. PSL)		
Job no.	OUTPUT FIELD (as for corres. PSL)		

F7 (HELP) can be used to make an entry in the fields marked '*'.

5.4.6 Updating the Group Inhibit Bits

Call the 'UPDATE GIB' mask with the function key F6 (CONTINUE) and **F1** (UPDATE GIB) in the 'PC JOB' mask.



F6: the PC **job** is saved on the storage medium

F8: abort:
return to the 'PC JOB' mask

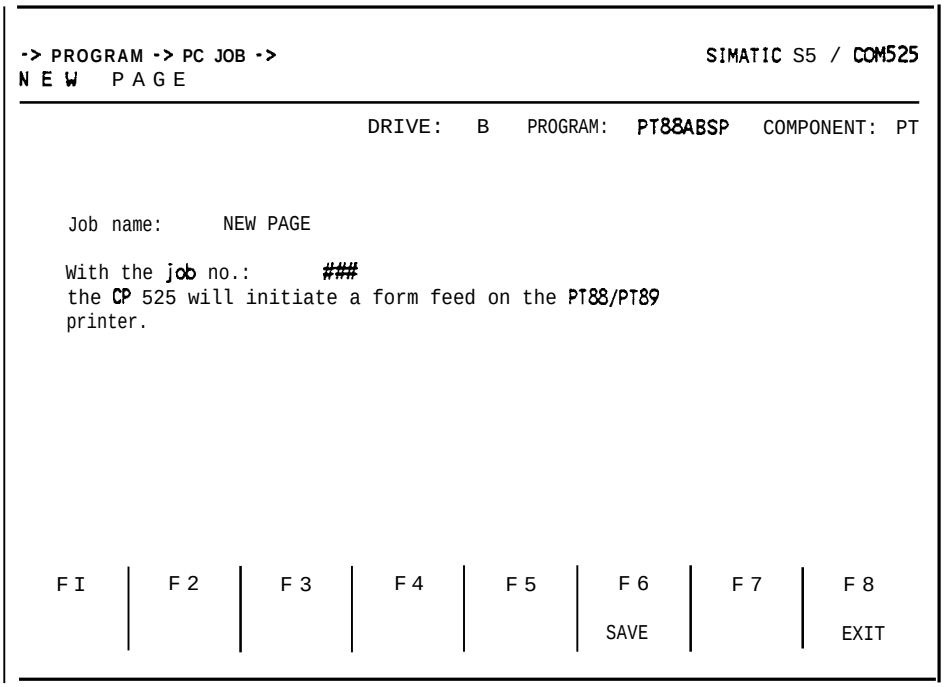
Field	Field type keys allowed	Limit value	Alternatives
Job name	element name (see 5.1)		*
Job no.	numbers	1 - 199	
CPU no.	numbers	1 - 4	*
DBno.	numbers	1 - 255 (DB numbers 1 and 2 are reserved for the system)	
DW	numbers	0 - 255	

F7 (HELP) can be used to make an entry in the fields marked '*'.

5.4.7 New Page

Call the 'NEW PAGE mask with the function key F6 (CONTINUE) and F2 (NEW PAGE) in the 'PC JOB' mask.

You can only program one 'NEW PAGE' PC job in a user program. The name NEW PAGE is assigned by COM 525 and cannot be changed.



F6: save the PCjobon thedatamedium

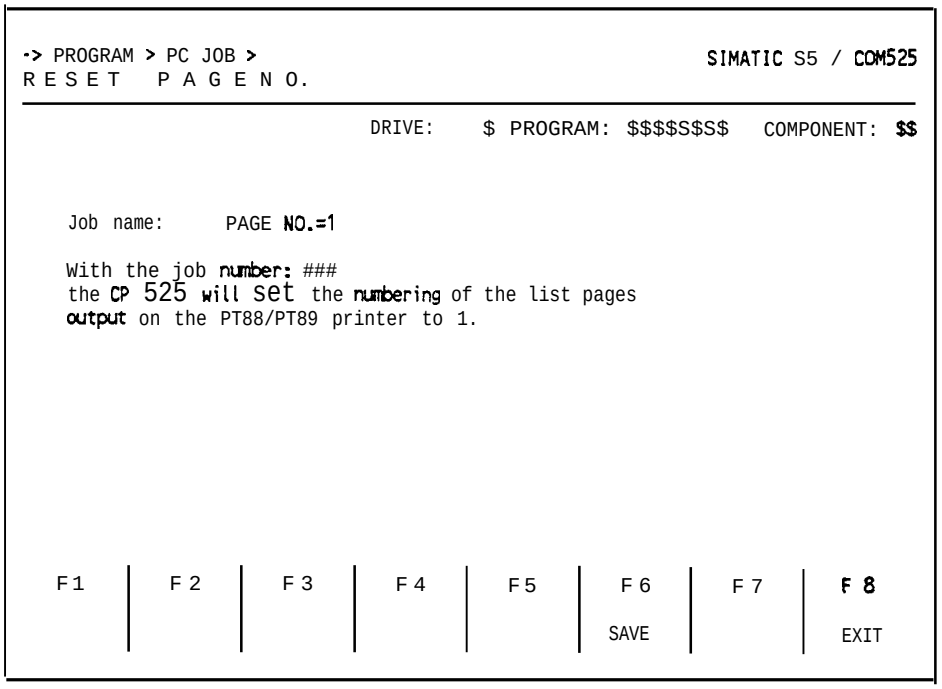
F8: abort:
return to the 'PC JOB' mask

Field	Field type keys allowed	Limit value	Alternatives
Job no.	numbers	1 - 199	

5.4.8 Reset Page Number

Call the 'RESET PAGE NO. ' mask with the function key F6 (CONTINUE) and F3 (PAGE NO. = 1) in the 'PC JOB' mask.

You can program only one 'RESET PAGE NUMBER' PC job in a user program. The name PAGE NO. = 1 is assigned by COM 525 and cannot be changed.



F6: save PC job on the storage medium

F8: abort:
return to the 'PC JOB' mask

Field	Field type keys allowed	Limit value	Alternatives
Job number	numbers	1 - 199	

5.5 FRAMES

Call the 'FRAME' mask with the function key F3 (FRAME) in the following masks

- 'PROGRAM'
- 'PROCESS STATUS LIST'
- 'CUR. MESSAGE LIST'
- 'SEQ. MESSAGE LIST'

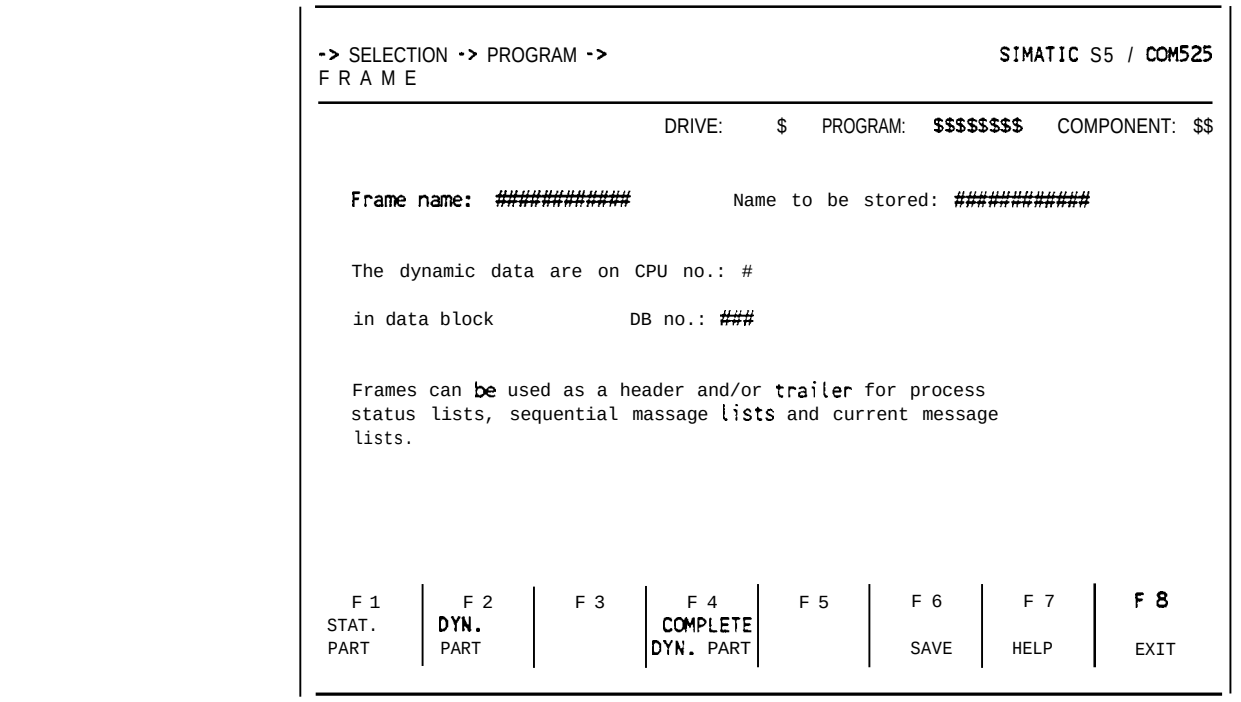
Frames are processed in the same way that process status lists are, i.e. with the same text editor (see 5.4.4). There are, however, certain restrictions as follows:

- frames can be only maximum nine lines long
- the dynamic data in a frame is expected in a data block, which the CP 525 automatically requests from the CPU. This block must not exceed 128 bytes.

You can enter the CPU number and the number of the data block in the mask.

It is possible to read in existing frames. You specify the name of the frame in the 'frame name' field. You can modify the frame which has been read in and save it again under a different name. The new name must be entered in the 'name to be stored' field.

The 'name to be stored' field initially always has the name specified in the 'frame name' field as a default.



F1: generate the static part,
branch to the 'STATIC PART' mask

F2: generate the dynamic part,
branch to the 'DYN. PART' mask

F4: complete the dynamic part,
branch to the 'COMPLETE DYN. PART' mask

F6: save the frame on disk

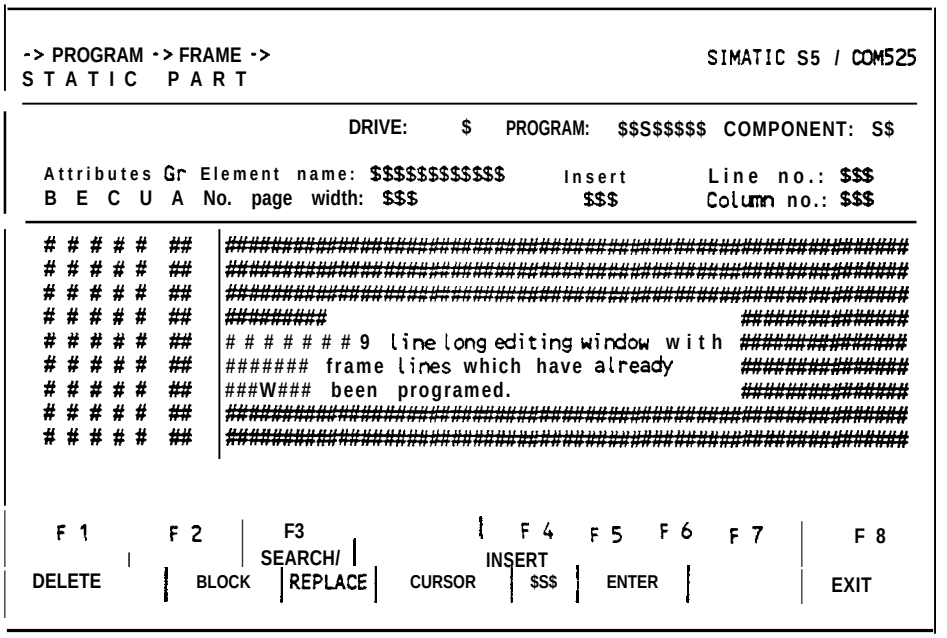
F8: return to the mask: 'PROGRAM'
or 'PROCESS STATUS LIST'
or 'CUR.MESSAGE LIST'
or 'SEQ.MESSAGE LIST'

Field	Field type keys allowed	Limit value	Alternatives
Frame name	element name (see 5.1)		*
Name to be stored	element name (see 5.1)		
CPU no.	numbers	1 - 4	*
DB no.	numbers	1 - 255 (the DB numbers 1 and 2 are reserved for the system)	

F7 (HELP) can be used to make an entry in the fields marked '*'.

5.5.1 Static Part of a Frame

Call the 'STATIC PART' mask with the function key F1 (STAT. PART) in the 'FRAME' mask.



F1: editor function DELETE (see **5.3.1)**

F2: editor function BLOCK (see **5.3.1)**

F3: editor function SEARCH (see 5.3.1)

F4: editor function CURSOR (see **5.3.1)**

F5: editor function INSERT (ON/OFF),
switches the insert mode on or off

F6: enter the static part of a frame

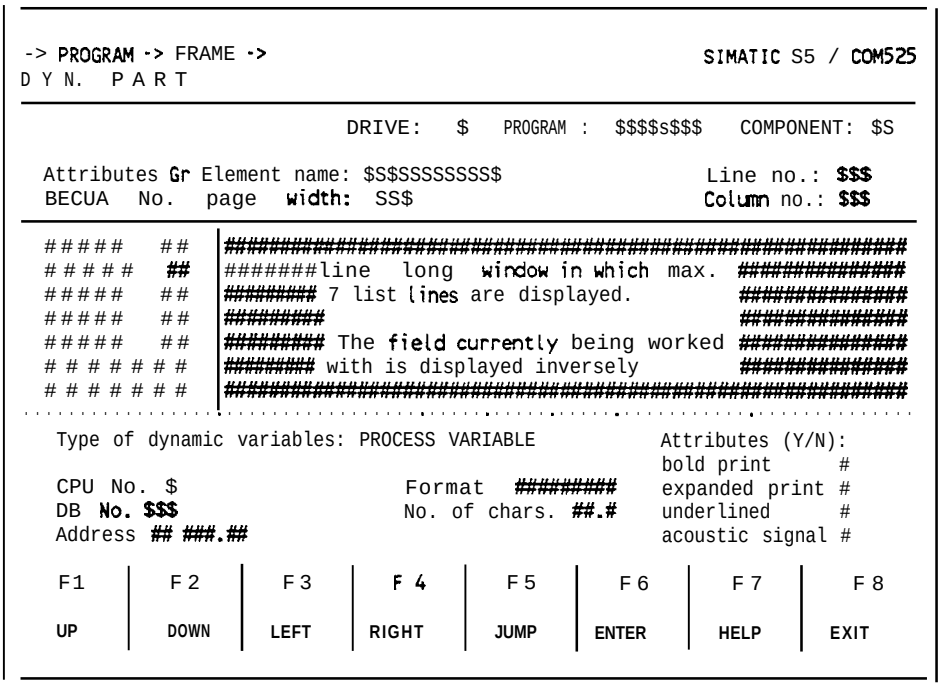
F8: abort:
the data is not saved;
return to the 'FRAME' mask

Field	Field type keys allowed	Limit value	Alternatives
Attributes	letters B=bold, E=expanded, C=compressed, U=Underlined, A=acoustic signal		Y, N *
Or no.	numbers	0 - 15	
Element name	OUTPUT FIELD		
he no.	OUTPUT FIELD		
Page width	OUTPUT FIELD (from interpreter parameter assignment see 3.6)		
Column no.	OUTPUT FIELD		

F7 (HELP) can be used to make an entry in the fields marked '*'.

5.5.2 Dynamic Part of a Frame

Call the 'DYN. PART' mask with the function key F2 (DYN. PART) in the 'FRAME' mask.



The function and processing of this mask corresponds to the 'DYN. PART' mask of a process status list (see 5.4.4 b).

The limit values for the DB address and line are, however, different as follows:

memory access mode	DB address
DD	0 - 126
DR,DL,DW	0 - 127
BI	0.0 - 127.15

a) Frame - programming the text for the process status

The function and processing of this mask corresponds to the 'PROCESS STATUS VAR. ' mask (see 5.4.4 b).

The limit values for the DB address are, however, different:

memory access mode	DB address
DR,DL	0 - 127

5.5.3 Complete lb Dynamic Parts of a Frame

The function and processing of this mask corresponds to the 'COMPUTE DYN. PART' mask in a PSL (see 5.4.4 c).

The limit values for the DB address are, however, different:

memory access mode	DB address
DD	0 - 126
DR,DL	0 - 127
BI	0.0 - 127.15

6 Terms and Definitions

COM 525
Programming package for the communications processors CP 525 and CP 524

Component
Indicates the function of a CP 525 interface:

PT = printer
CL = computer link

CP 525/CP 524
communications processors

Data type
The user program is divided into data types (message, PC job, frame or job block).

Element
Single partofadatatype

Function keys
Keys that have a specific significance in each mask. The labelling of the key indicates which action will be executed if the key is pressed.

Identification header
Header that contains the designation of the plant, who generated the program and when. The identification header is included on the disk and in the EPROM and is used to distinguish between programs with the same name on different storage media (volumes).

Interpreter
interchangeable interface driver that defines the characteristics of the component

KOMI
S5-DOS command interpreter: see description of the operating system S5-DOS

Library
Collection of user data which can be transfer-red to several user programs. Various interpreters and procedures can also be collected here. A library is identified by the name COMLIB?? ("?" = any alphanumeric character). A detailed description of these can be found in the section "Libraries" in this user's guide.

Listing
Overview of a user program output on a printer by COM 525 (not to be confused with current message list and sequential message list).

Mask
Template displayed on the PG monitor. COM 525 prompts inputs in the masks and enters defaults from previous masks.

Menu
I'Unction keys F1 to F8 with the corresponding labelling.

Module (software)
See **element**.

overlay
Part of a program which cannot run independently. Is only loaded inthememorywhenit is required.

PCP/M
Operating system for your programmer (pG). Is loaded when you switch on the PG and insert the PCP/M floppy disk.

Procedure
Interchangeable software driver, which implements the transmission procedure on the line.

S5-DOS
Operating system extension for PCP/M to standardize all the software packages running on programmers.

user area
A hard or floppy disk can be divided into 16 user areas; brings order to large numbers of files.

user data
User program
Total of all the data generated by the user with COM 525; required to operate a CP 525/CP 524 interface.

SIEMENS

We have checked the contents of this manual for agreement with the hardware and software described. Since deviations cannot b-s precluded entirely, we cannot guarantee full agreement. However. the data in this manual are reviewed regularly and any necessary corrections included in subsequent editions. Suggestions for improvement are welcomed.
Technical data subject 10 change.

Siemens Aktiengesellschaft

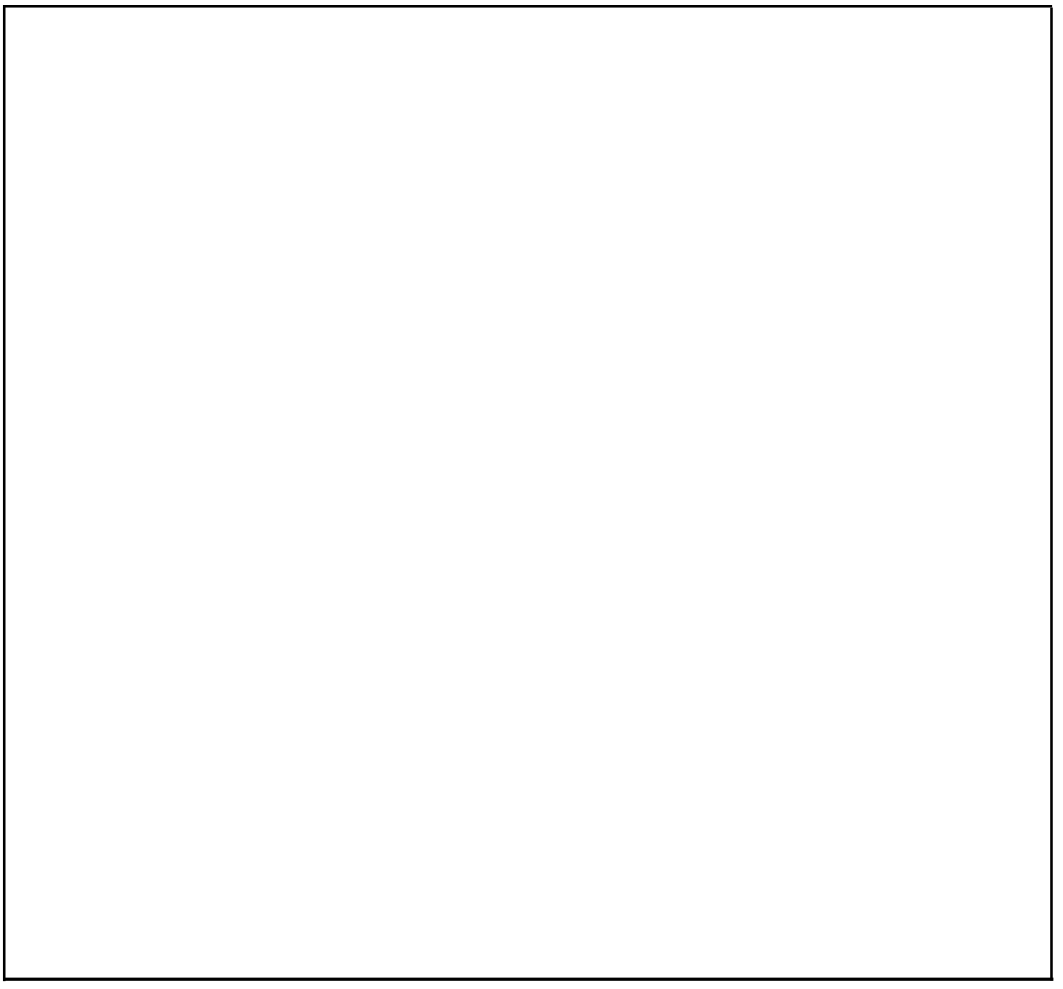
C79000-B8576-C544-06
Gerätewerk Karlsruhe
Printed in the Federal Republic of Germany

SIEMENS

SIMATIC S5
COM 525 Messages

Reference Manual

Order No. C79000-B8576-C545-03



Contents Page

1 Introduction

When you are working with the COM 525, messages are output in the message line (the line above the function key menu) under the following conditions:

-whenanerrorhasoc curred (error message)

- to avoid errors (warnings)

  as information for the user, for example to show that an action has been carried out successfully (general message)

- to ask whether the user “really” intends to carry out an action, for example when deleting (prompts).

A message includes an identifier (e.g. MF, ERR), a number and a short text.

The foilowing section lists and describes all the COM 525 messages.

2 Messages Common to All Masks

Message number	Description / Remedy
MF.001	Illegal key! - Illegal characters have been entered in the field in which the cursor is positioned. Refer to Sections 2 to 5 of the COM 525 user's guide in this manual to find the permissible characters.
MF.002	Illegal entry! - The entry which has been made in the field on which the cursor is located does not correspond to any of the permissible alternatives, or it exceeds or falls below the permissible limit values. Refer to Sections 2 to 5 of the COM 525 user's guide in this manual for the limit values and the permissible alternatives.

3 General Situations

3.1 General Error Messages

Message number	Description / Remedy
ERR.001	Overlay has wrong version number! - Use the correct COM 525 system diskette.
ERR.004	Program not found - The overlay cannot be found (see also Section 'Standard outputs when ***working with*** COM 525' in the COM 525 user's guide in this manual).
ERR.009	Element destroyed! - When reading in an element from the file the length was found to be wrong. Delete the element! The error can also occur when the 'already existing" destination element is destroyed during transfer. (This is read in the acknowledgement mode in order to achieve a comparison.)
ERR.010	Error in user program The file with the user data has been corrupted and can no longer be interpreted.

Message number	Description / Remedy
ERR.011	User data generated with old COM version - no longer valid.
ERR.012	***Error*** in SYSID - An error in the memory submodule of the CP 525. With the EPROM: the memory submodule must be replaced. With the RAM: if a cold restart on the CP 525 does not bring about an improvement, the memory module should be replaced.
ERR.013	NotaCP525!
ERR.014	Interpreter not present - fetch from library - See Sections 'Structure of a COM 525 program' and 'Libraries' in the COM 525 user's guide in this manual.
ERR.015	Procedure not present - fetch from library - See ERR.014.

Message number	Description / Remedy
ERR.020	Reserved name not allowed here - The following PC jobs can only exist once per program and therefore have resewed names: SEQ.MESSAGE UPDATE SML CUR.MESSAGE NEW PAGE RESET PAGE
ERR.021	**Name** is illegal or reserved! - See ERR.020.
ERR.024	CP not programmed - interpreter missing! - Neither interface 1 nor interface 2 has been programmed, The CP 525 does not contain an interpreter, procedure or user data. Transfer the interpreter!
ERR.025	Interface does not exist - Only with the CP 525 can two interfaces be programmed.
ERR.026	Interface is not programmed - No interpreter has as yet been transferred to the selected CP 525 interface.

Message number	Description / Remedy
ERR.030	At x lines per inch: page length of y to z lines possible!
ERR.101	First generate static part => F1
ERR.102	No dynamic fields in static part
ERR.103	There are no dynamic fields in messages x to y
ERR.104	Name already assigned
ERR.105	Illegal column number - The column number entered exceeds the maximum defined during interpreter parameter assignment.
ERR.107	Illegal message number! - The message number must be between 1 and 2047. When making a jump: the specified message does not exist.

Message number	Description / Remedy
ERR.108	Illegal line number -The line number must be between 1 and 2047. When making a jump: thespecifiedline number does not exist.
ERR.109	Illegal field number - The specified field number is greater than the number of existing fields.
ERR.110	First generate sequential message list => F2
ERR.111	Text must be enclosed in inverted commas!
ERR.112	Illegal entry: fields overlap
ERR.113	Bit not specified!

Message number	Description / Remedy
ERR.114	Combination of memory access and format illegal - See Section 'Process status list', Table 2 in the COM 525 user's guide in this manual.
ERR.115	Different types of memory access! - See Section 'Process status list', Table 2 in the COM 525 user's guide in this manual.
ERR.116	Value too high - cannot be represented!
ERR.117	Job not saved - The job can only be printed out after it has been entered. First enter the job with F6!
ERR.118	Line is too long - The required structure for the date, time orstatusstarting at the specified column exceeds the maximum length of a line (136 characters per line). Change the structure or position accordingly!

Message number	Description / Remedy
ERR.119	DB boundary exceeded (No. of characters)!
ERR.120	No more fields in this direction!
ERR.121	All fields already programmed in this direction!
ERR.122	No more messages loaded in this direction!
ERR.123	All loaded messages programmed in this direction!
ERR.124	Message specified is not loaded!
ERR.125	No field in the line specified!

Message number	Description / Remedy
ERR.126	Field specified not in this line!
ERR.127	No field in the message specified!

3.2 General Warnings

With warnings the data input will generally be entered or saved.

Message number	Description / Remedy
WARN.001	Source/dest. address cannot be addressed - See user's guide 'Computer link with RK 512' in this manual.
WARN.002	Coord. flag too high as partner for CP 525
WARN.003	DB number used by system - Depending on the CPU, DB 1 and possibly DB 2 may be reserved for the system.
WARN.004	Less characters than number of positions reserved
WARN.005	More characters than number of positions reserved
WARN.006	Field too long - max. 60 characters used!

Message number	Description / Remedy
WARN.007	Only 80 characters will be printed!
WARN.013	From message x to message y read in! The remainder was not loaded!

3.3 General Prompts

Prompts must be answered with F1 (YES) or F3 (NO) in the acknowledgement menu.

Message number	Description / Remedy
QUIT.001	Terminate program? - 'YES' = COM 525 returns to the basic mask. A program which contains no more data will be deleted as a file.
QUIT.002	Loss of data - exit mask? - 'YES' = the entries made in the current mask will not be entered. 'NO' = the data can be entered with F6 (ENTER or SAVE).
QUIT.003	Process library? Libraries do not have to contain consistent programs; they are used to collect different parts of programs (see also Section 'Libraries' in the COM 525 user's guide in this manual).
QUIT.004	Loss of data - exit field? - As for QUIT.002

Message number	Description / Remedy
QUIT.010	Element already exists! - overwrite?
QUIT.011	Job block already exists - overwrite?
QUIT.012	No comparison of elements possible! Overwrite? - There is insufficient memory to compare the contents of the source and destination element.

3.4 General Messages

Message number	Description / Remedy
MESS.001	Active! - This message is output while a function is being executed, when COM 525 user data are being processed on disk or data are being exchanged with the CP 525.
MESS.002	Completed!
MESS.003	Saved!
MESS.004	Active - abort with F8 - The function can be aborted with F8.
MESS.005	Deleted!
MESS.006	Element being processed:
MESS.007	Entered!

Message number	Description / Remedy
MESS.008	Aborted!
MESS.009	Loading overlays! - Is output while COM 525 overlays are being loaded.
MESS.010	Program being condensed - A program is automatically condensed when it is too “broken up” by deleting and storing.
MESS.011	Program already purged
MESS.012	First page! - When trying to page backwards from the first page.
MESS.013	Last page! -When trying topageforwardsfromthe last page.
MESS.014	Number of messages to be read from FD:

Message number	Description / Remedy
MESS.015	Number of messages to write/delete on FD
MESS.016	All fields from message x to y already programmed!
MESS.017	All fields have already been programmed!
MESS.018	The fields cannot be programmed from message x to y! - All fields which appear in the specified messages are located after the endofthelineowing tochangesin the sequential message list and cannot, therefore, be programmed. The fields can be programmed again if the sequential message list is changed again (e.g. reducing the length of the status text).
MESS.019	Active - deleting may take several minutes! - If there is a large number of elements (up to 1000) on the CP 525, deleting messages can take several minutes - the elements must be searched in the RAM.

4 Editing

4.1 Errors During Editing

Message number	Description / Remedy
ERR.100	Data cannot be interpreted - The code of the stored data has an error and cannot therefore be re-translated. This data can only be deleted.
ERR.106	Max. 9 (frame) or99 (PSL) lines possible.
ERR.130	Macro too long -> changes will not be entered - A maximum of 35000 bytes are available for programming a process status list. Once this number has been reached (too many process statuses programed) no further lines can be acceptedly the editor.
ERR.131	Macro too long, changes in the line will not be entered - See ERR.130.

Message number	Description / Remedy
ERR.132	Too many dynamic fields A maximum of 40 dynamic fields per line are possible with the process status list and with frames. For each message only one dynamic field is permitted,
ERR.133	Start of block not set
ERR.134	Start of block after end of block
ERR.135	Block not set
ERR.136	No line saved yet
ERR.137	Destination for block transfer is within the block

Message number	Description / Remedy
ERR.138	Block cannot be saved - macro too long - A maximum of 35000 bytes are available for programming a process status list. Once this number has been reached (too many process statuses programmed) no further lines can be accepted by the editor.
ERR.139	Block cannot be saved - too many lines
ERR.140	No more found - press ENTER key
ERR.141	Reserve characters not allowed - The characters used to reserve positions for dynamic fields must not be searched for using the 'search' and 'search/replace' functions.
ERR.142	No reserve characters allowed in word to be deleted
ERR.143	Insertion of a line not possible

Message number	Description / Remedy
ERR.144	Insertion only possible when cursor is at start or end of line
ERR.146	Message number already assigned
ERR.148	Message number already assigned in area read in -> HELP
ERR.150	Max. number of messages already programmed! -Amaximum of 1000 messages (from 1 - 2047) can be programmed.
ERR.151	Message number assigned outside area read in -> HELP
ERR.152	Message number not in area read in!
ERR.153	Block will not be copied - too many messages! - Max. 99 messages can be entered in the editing buffer of the PG. By copying blocks this number has been exceeded.

4.2 **Warnings during Editing**

Message number	Description / Remedy
WARN.006	Field too long - max. 60 characters used!
WARN.008	Lines longer than page width - If, when editing a process status list a frame or a message there are more characters edited per line than determined in the 'ASSIGN INTERP. PARA.' mask as the page width, the characters which extend beyond the end of the line will be printed in the next line. Remedy: increase the page width in the 'ASSIGN INTERP. PARA.'
WARN.009	Masked area changed - dyn. part has been deleted - cf. WARN.012
WARN.010	Message numbers higher than 2000 are system messages!
WARN.011	Copied messages have system message numbers higher than 2000!

Message number	Description / Remedy
WARN.012	SML was modified! Inconsistency of data in messages with ">" - By changing the parameter assignment for the sequential message list when messages already exist it is possible that the following inconsistencies may occur (such messages are marked with ">") 1) If messages were programmed with DATE, TIME or STATUS and the jobs were later deleted in the sequential message list, this warning points out that the jobs in these messages no longer exist (displayed by a narrow inverse field with the identifier D, T or S). They will not, however, be triggered by the CP. Remedy: change the parameters in the sequential message list. 2) If messages have been programmed with DATE, TIME and/or STATUS and later the date, time or status field was extended in the sequential message list, it is possible that a message line which has already been programmed will be longer than 136 characters. The characters which would exceed this if anything is changed in the message are truncated. (It is possible that a dynamic field which has already been programmed may be lost). Remedy: change the parameters in the sequential message list (to see the original text of the truncated message, the date, time or status fields in the SML must be reduced again).

Message number	Description / Remedy
WARN.013	From message x to message y read in! The remainder was not loaded!
WARN.015	Dyn. field in line x too long (max. 80 res. pos. per field)!
WARN.016	Dyn. field in message x *too long (max.* 80 res. pos. per field)!

4.3 **Prompts during Editing**

Message number	Description / Remedy
QUIT.040	Dyn. part already exists, delete?
QUIT.041	Field with a dyn. part in the last column, delete? - In the insert mode or when replacing a word a line may sometimes become longer than 136 characters. These characters and any programmed dynamic fields would be lost during the insertion. The characters may possibly be located outside the currently visible characters on the screen.
QUIT.042	Delete last character in the line? - See QUIT.041
QUIT.043	REPLACE?
QUIT.044	Line too long - replace?

Message number	Description / Remedy
QUIT.045	Field with dyn. part in column before masked field, delete? - Inhibited fields are fields which cannot be changed for the date, time and status in the editing window for messages.
QUIT.046	Delete character before masked field?
QUIT.047	New text exceeds masked area, replace? - When replacing a word it is possible that the following text would go beyond the date, time or status field, since the text following the word to be replaced is shifted only as far as the next date, time or status field. These characters may possibly be outside the visible characters displayed on the screen.
QUIT.050	**Save?**
QUIT.051	Changed messages marked ">" will be truncated! ENTER? - See WARN.012

Message number	Description / Remedy
QUIT.052	Abort reading in of messages from FD?
QUIT.053	Abort saving of messages on FD?

5 Deleting

5.1 Warnings when Deleting

Message number	Description / Remedy
WARN.014	Before deleting SEQ.MESSAGE: check whether messages exist for it! - Do not forget that the position of the date, time and status is fixed with the PC job SML (sequential message list). If the SML is missing no messages can be output even if they have been programmed and stored on the CP 525/ 524.

5.2 Prompts when Deleting

Message number	Description / Remedy
QUIT.020	Delete both interfaces?
QUIT.021	Delete all elements?
QUIT.030	Delete element:
QUIT.031	No element will be deleted! - abort function?
QUIT.032	Delete all elements?

6 Transfer

6.1 General Transfer Errors

Message number	Description / Remedy
ERR.016	Function illegal: component wrong - An attempt has been made to transfer a PT element to a CL program or a CL element to a PT program.
ERR.022	Source =destination?
ERR.023	Different components in source and destination - See ERR.016
ERR.024	CP not programmed - interpreter missing! - The CP 525 has no interpreter, procedure or user data on interface 1 and interface 2.
ERR.025	Interface does not exist - Two interfaces can be programmed only on the CP 525-2, with other CP types only one can be programmed.

Message number	Description / Remedy
ERR.026	Interface is not programmed - No interpreter has been transferred to the selected CP 525 interface.
ERR.027	EPROM plugged in - No data can be written to the EPROM on the CP 525! Plug in a RAM card.

6.2 **Transfer Errors, Floppy and Hard Disk**

Message number	Description / Remedy
ERR.701	Drive cannot be addressed! - Check whether the drive compartment is closed!
ERR.702	Element directory not present - No element exists for the data type selected!
ERR.703	Element not present
ERR.704	FD is write-protected With a floppy disk: remove the write-protect tab. With hard disk: see description of the operating system PCP/M.
ERR.706	Maximum number of elements exceeded - No more elements of the selected data type can be stored.
ERR.707	File not present

Message number	Description / Remedy
ERR.708	FD external full - No more entries can be made in the file directory. Check whether files can be deleted or use a new floppy disk,
ERR.709	Working diskette/CP memory full With FD: use a new floppy disk. With CP: transfer the new data to floppy or hard disk and use a larger RAM submodule.
ERR.710	Element already exists
ERR.712	FD has been changed - program no longer available - The floppy disk was changed while processing a program. Insert the correct floppy disk again!
ERR.713	File is write-protected - Remove the write-protect attribute (see descriptionof the operating system PCP/M).

Message number	Description / Remedy
ERR.714	Not a COM 525 file - The file was not generated by this COM 525.
ERR.715	Illegal file name - The file name specified does not correspond to the PCP/M conventions. See Section 'Selecting the user program' in the COM 525 user's guide in this manual.
ERR.716	File already exists
ERR.717	Invalid date - The entry in the "date" field is illegal.

6.3 Errors when Transferring Between COM 525 and the CP 525

Message number	Description / Remedy
ERR.718	No RAM card inserted!
ERR.720	SYSID block does not match specification Error in memory submodule. With EPROM: the memory submodule must be replaced. With RAM: if a cold restart on the CP 525 does not bring improvement, the memory submodule must be replaced.
ERR.721	CP switch set to STOP - Switch the mode selector on the CP to RUN.
ERR.722	Transfer error - Error or fault on the CP 525 or the link between the CP 525 and PG which cannot be identified more exactly,
ERR.723	Initiation of function negatively acknowledged - Please check whether there is. a connection to the CP 525.

Message number	Description / Remedy
ERR.724	Interpreter/procedure not present - Before data can be transferred the interpreter must first be transferred to the CP 525.
ERR.725	Cable not connected/wrong baud rate
ERR.727	Job number already exists on CP! -A PCjobwiththesame number has already been transferred to the CP 525.
ERR.728	Interface cannot be deactivated at present! - Repeat the transfer. When transferring between the CP 525 and a partner follow the rule of thumb: large volume of data -> low data rate.
ERR.729	Error in interpreter/procedure - Following this message, the interface on the CP 525 is completely deleted. Possible causes: error in the source file (interpreter or procedure on the PG) - error during transfer from PG to CP 525 error in the memory submodule of the CP 525

6.4 Prompts during Transfer

Message number	Description / Remedy
QUIT.022	Delete interface on CP and start program transfer? - 'YES' = before the program transfer the interface will be completely deleted.
QUIT.023	Transfer all elements?

7 Saving

7.1 Errors when Saving

Message number	Description / Remedy
ERR.130	Macro too long -> changes will not be entered - A maximum of 35000 bytes are available for programming a process status list. Once this number has been reached (too many process statuses programmed) no further lines can be accepted by the editor.
ERR.131	Macro too long, changes in the line will not be entered - See ERR.130

7.2 Prompts when Saving

Message number	Description / Remedy
QUIT.010	Element already exists overwrite?
QUIT.011	Job block already exists - overwrite?
QUIT.051	Changed messages marked 'Y' will be truncated! ENTER? - See WARN.012

8 Internal Errors

Notes on these errors

These errors can be caused by a corrupted COM 525. If, despite reloading from a back-up disk the problem cannot be remedied, please send a description of the problem to the nearest SIEMENS service department or technical office.

Message number	Description
IF.001	Mask not present!
IF.002	Field not present
IF.003	Illegal MAINT call
IF.004	Overlay identifier is illegal!
)2?.005	Buffer not long enough!
IF.006	Data type riot defined!

Message number	Description \ Remedy
IF.007	CP function unknown - Wrong function call caused by transfer error. Repeat the function call. If the error message occurs again **see** 'Notes on these errors' on the previous page.

9 Operating System Errors

Message number	Description
ERR.002	Program cannot be closed - Internal error (see Section 8).
ERR.003	Program cannot be executed
ERR.004	Programnotfound
ERR.005	Error loading program - not enough memory - See ERR.007
ERR.006	Program cannot be removed - Internal error (see Section 8).
ERR.007	Not enough memory for buffer - press ENTER key - Attempt to restart the system (RESET or switch off and then on). If the error message appears again there is insufficient memory capacity.

Siemens Aktiengesellschaft

C79000-B8576-C545 -03
Gerätewerk Karlsruhe
Printed in the Federal Republic of Germany

SIEMENS

SIMATIC S5

PROM 525

User's Guide C79000-B6576-C546-04

contents

1 Introduction

The program PROM 525 supplements the programming package COM 525. It is used to transfer COM 525 programs, stored on floppy or hard disk, to an EPROM. With PROM 525 the transfer in the opposite direction is also possible, i.e. transferring a program back from an EPROM to a floppy or hard disk.

Using PROM 525 you can test whether a aubmodule has already been written to or whether it is empty. PROM 525 also has two information functions. The first one outputs en identification header for the programs located on the floppy or hard disk. The second function provides you with information about EPROMs which have already been written to.

You can have the length of your program displayed, and therefore check whether the aubmodules you want to program still have sufficient free memory capacity. PROM 525 informs you of the length of time required for programming without any transfer being made. While you program an EPROM, the programming time still left to run is displayed.

Before you start to work with PROM 525 you have probably already generated programs with COM 525 end know about the masks and menus. If you are unsure about using the masks and menus or are still unfamiliar with terms or abbreviations that are used in this part of the manual please refer to 'Terms and definitions in the user's guide to COM 525.

With the PG 695 it is possible to connect an external programmer (prommer) to the device. More information can be found in the installation guide and instructions supplied with the device. The initialisation (the setting up of the communications link) is carried out by PROM 525 (even with the integrated unit). This takes approximately 60 to 70 seconds.

A program generated with COM 525 is initially stored either on a floppy or hard disk. The following EPROM submodules can be programmed with PROM 525:

EPROM submodule number	capacity	
6ES5 373-0AA41	16 Kwords	
6ES5 373-0&461	32 Kwords	
6ES5 373-0AA81	64 Kwords	*
6ES5 373-0AA42	16 Kwords	
6ES5 373-0AA62	32 Kwords	
6ES5 373-1AA41	16 Kwords	*
6ES5 373-1AA61	32 Kwords	*
6ES5 373-1AA81	64 Kwords	*

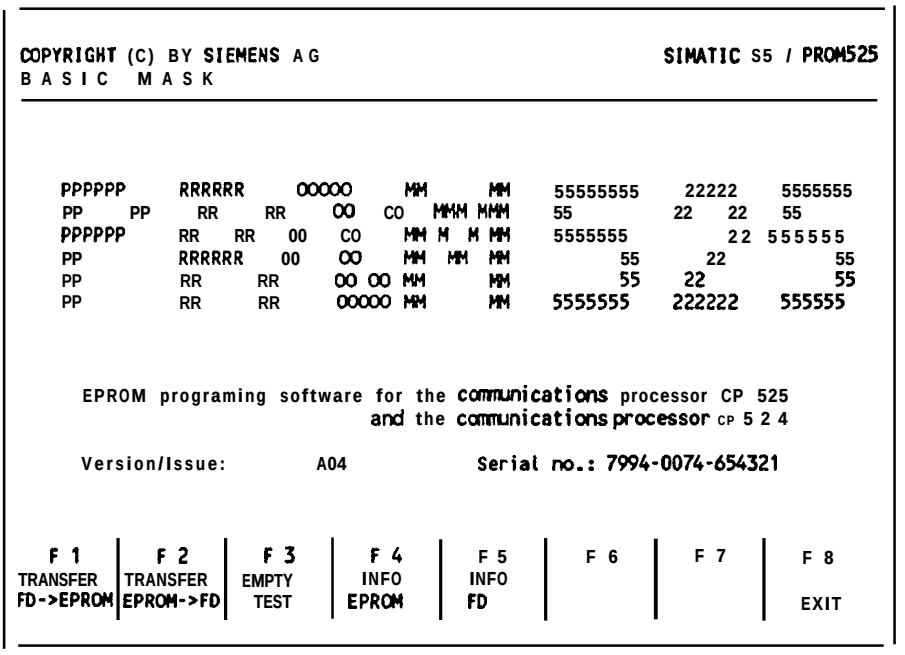
* With PG 685 RELEASE < 8 only possible with MEP adapter.

2 Example of Programming an EPROM

In this section you are introduced to the fictions of the PROM 525 step by step. You can implement the example on your own screen. If you find that an explanation is missing at any point (e.g. the explanation of the function key) you can find this in the reference section of this user's guide.

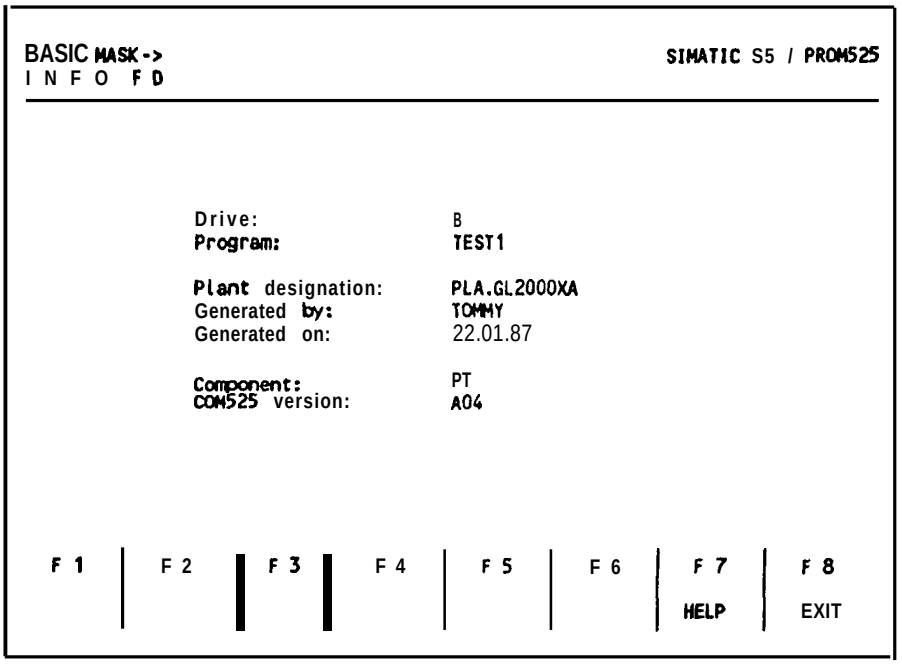
The PROM 525 program is loaded when you enter the co-d "S5" to call the S5 command interpreter.

Position the cursor on the programming package COM 525. Then press the function key F1 (PACKAGE) to call the 'BASIC MASK' of the PROM 525.



Use the function keys F1 to F5 to branch to the PROM 525 follow-on masks. Using F8 (EXIT) you always return to the previous mask, or from the 'BASIC MASK' to the S5 command interpreter.

For the example, press F5 (INFO FD). PROM 525 then outputs the ' INFO FD ' mask.



This mask displays information about your program,

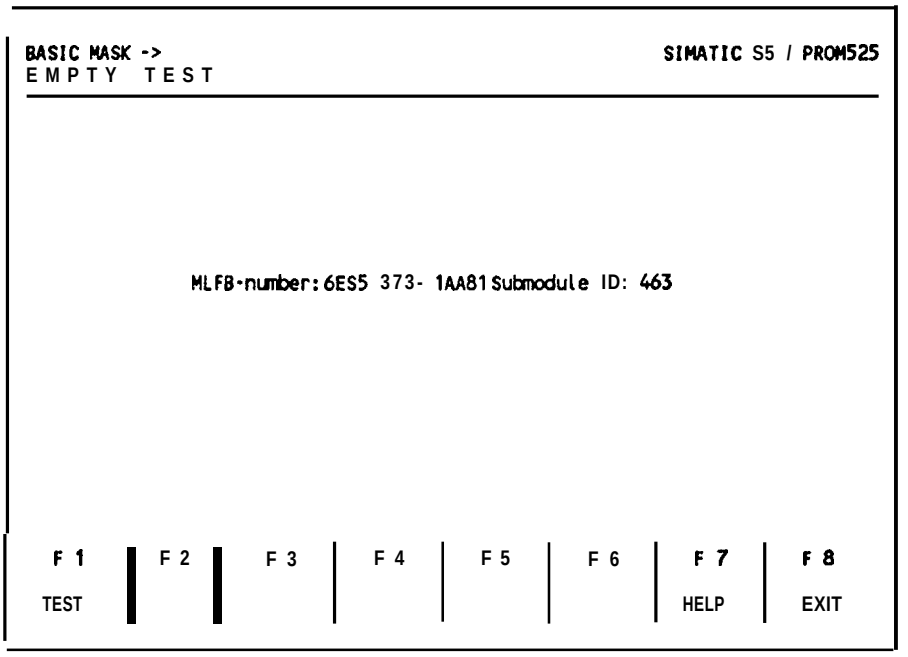
The 'drive" field shows the drive in which your ' 'PROM 525 program-~ package is located and needs to be changed only if for example the package is on the hard disk, end your program is on a floppy disk.

The cursor is positioned on the "program" field. You can enter the name of the file about which you wish to obtain information (for the example, take the program to be transferred to an

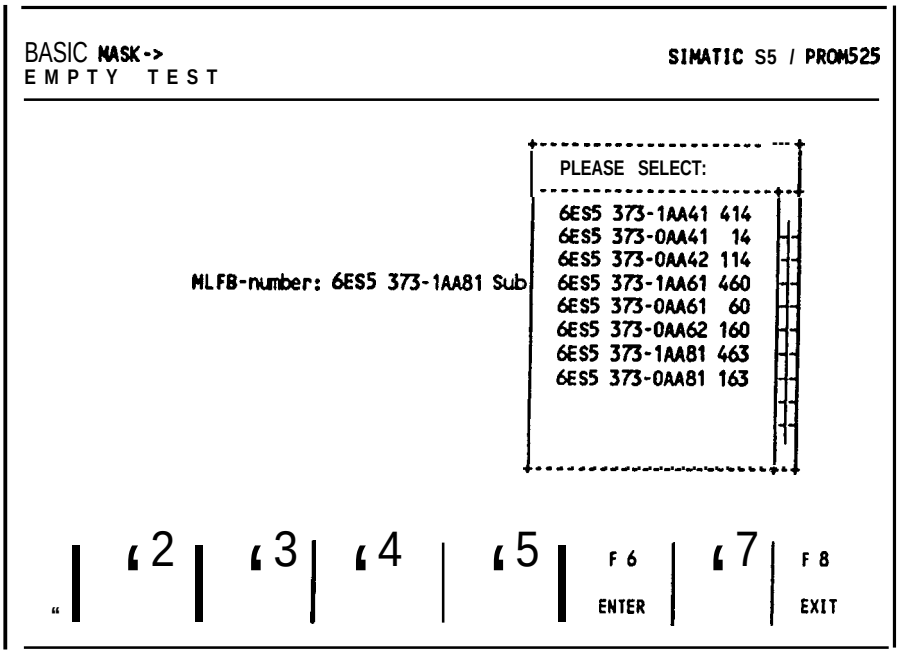
EPROM). This can also be achieved with F7 (HELP), which you already know through working with COM 525. In the reference section of this user's guide you can see which fields can be completed using F7 (HELP). If the COM 525 file you have, specified is present, the identification header of the program is now displayed.

If you have programed both interfaces, i.e. you wish to transfer two programs, you can also have the identification header of the second program displayed. Press F8 (EXIT) to return to the basic mask.

As the next step, check whether the EPROM you wish to program is empty. With F3 (EMPTY TEST) select the 'EMPTY TEST' mask and plug in one of the EPROMs which has already been written to in the receptacle (in the example an EPROM with the EPROM number 6ES5 373-1AA81 is used).



When pressing F7 (HELP) the following mask appears with a selection list.



Using the arrow keys you can select an MLFB number and acknowledge it by pressing F6 (ENTER).

If the number of the EPROM you have plugged in is not the same as the default displayed, enter the correct number with F7 (HELP) and press F1 (TEST). In the field "current address" which now appears on the screen you can follow all the addresses being searched until the following message is output in the message line:

'EPROM is empty! '

If the EPROM has already been written to, the search stops at the first address which is not empty and the message

'EPROM not empty!

is output.

Once you are sure that your EPROM is empty, exit the 'EMPTY TEST' mask with F8 (EXIT). You can now program the EPROM.

Press F1 (TRANSFER FD->EPROM) in the 'BASIC MASK' to call the 'TRANSFER FD - EPROM' mask.

Mask with the CP 525-2

BASIC MASK -> SIMATIC S5 / PROM525
TRANSFER FD-EPROM

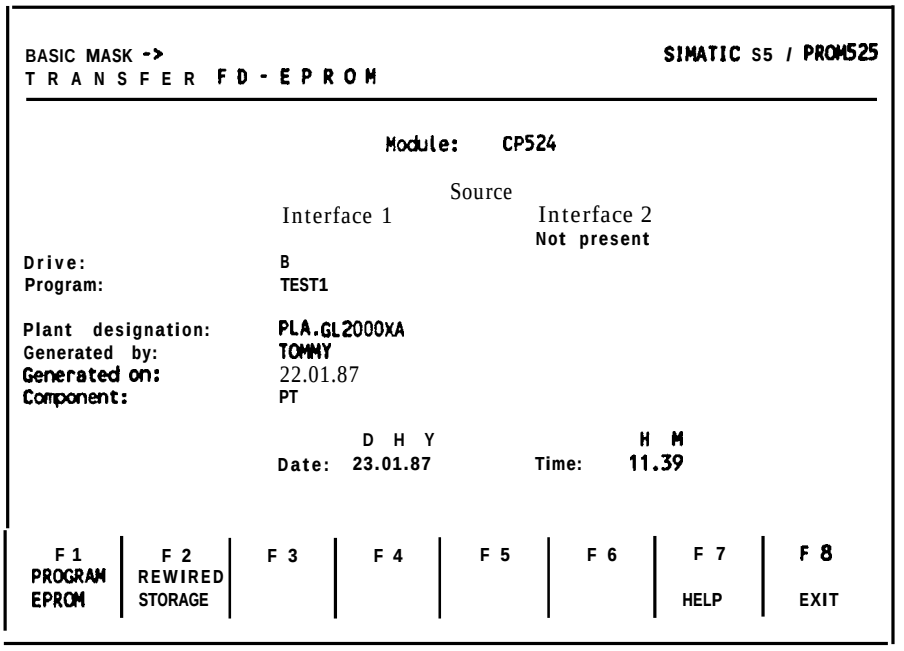
Module: CP525-2

	Source Interface 1	Interface 2
Drive:	B	B
Program:	TEST1	TEST2
Plant designation:	PLA.GL2000XA	PLA.GL2000XA
Generated by:	TOMMY	TOMMY
Generated on:	22.01.87	22.01.87
Component:	PT	CL

D M Y   H M
Date: 23.01.87   Time: 11.39

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
PROGRAM EPROM	REQUIRED STORMS					HELP	EXIT

Mask with CP 524



The "module" field on which the cursor is Positioned has the default 'CP 525-2', If you are using a CP 524 module enter this in this field with F7 (HELP) and continue reading at b) below.

a) CP 525-2

The fields "drive" below "interface 1" and 'interface 2" already have the default of the drive for the PROM 525 as seen in the 'INFO FD' mask. After you leave the "module" field the cursor is positioned on the 'program" field. If you want to transfer one or two programs (for both interfaces) when leaving the 'module' field, enter one program name or two program names one after the other using F7 (HELP). The plant designation, generator, date of generation and component for the programs are then displayed.

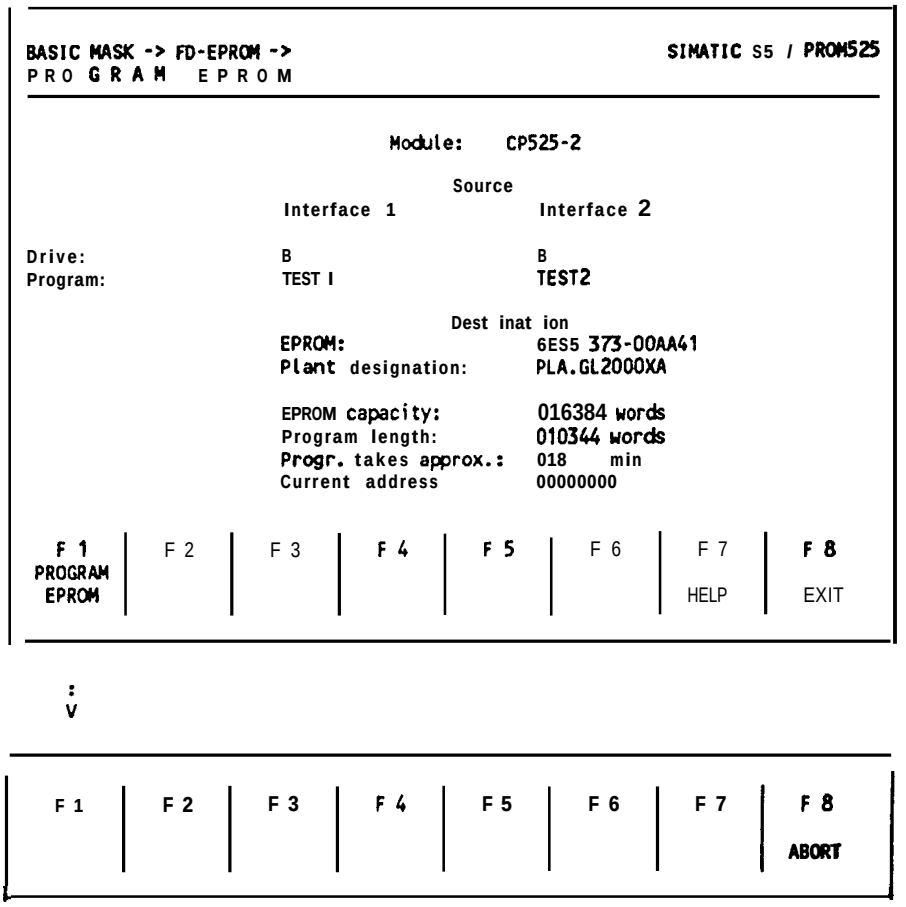
b) **CP 524**

No second interface is available for this module. Enter the name of your program with F7(HELP)- the "drive" field has as a default the drive with the PROM 525 package. The fields 'plant designation", 'generated by', 'generated on" end "component" now appear.

If the entries in the fields "date" and "time" are not up to date you should correct them (date in the form DD MM YY and the time HH MM).

Press F2 (REQUIRED STORAGE) to display the minimum storage required to store the specified program in the message line. You can then check whether the submodule you have selected has sufficient capacity.

Press F1 (PROGRAM EPROM) to branch to the 'PROGRAM EPROM' mask.



:
V

All the entries above 'destination" have been taken over from the previous mask (i.e. are different forCP 525-2 and CP 524).

If you change the 'EPROM" field using F7 (HELP), notice that a different value is output in the "EPROM capacity" field; this is because you plugged in a larger or smaller submodule. The field "plant designation" has defaults entered from the previous mask but can still be changed.

Now press F1 (PROGRAM EPROM). PROM 525 sets up the directories of the programs which are to be transferred. Following this, the storage, required for your program, the programing time end the current address are displayed in a submask.

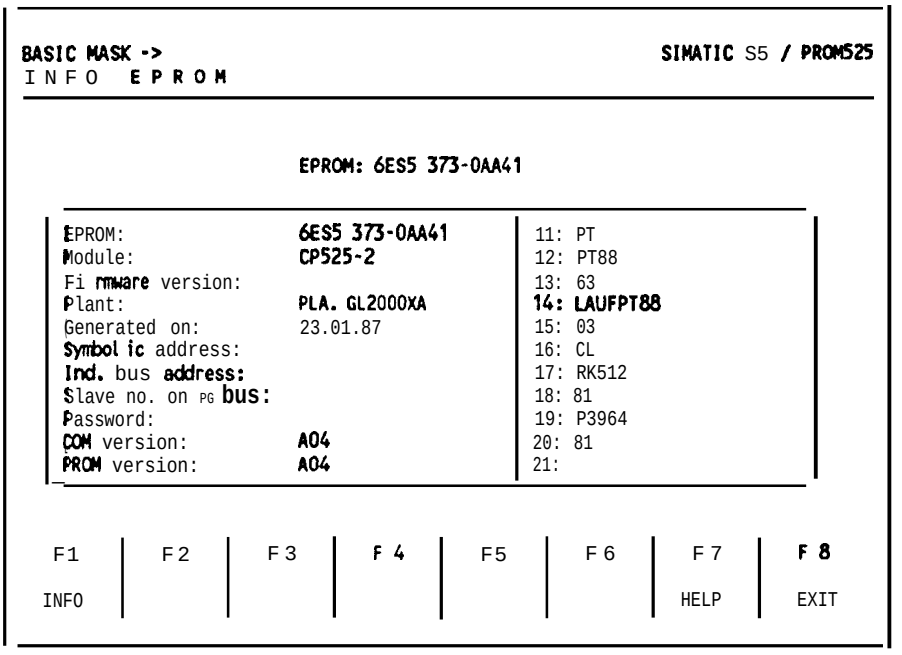
Now check to see that your EPROM is still plugged in and that it is the right one.

PROM 525 first checks that the EPROM is empty and only then starts the programming. From this point forward yOU can still interrupt the progreining using F8 (ABORT). (The programming is not aborted immediately but only after the transfer of the element currently being processed).

During the whole programming process the remaining programing time and the current address are displayed. The message line displays the name of the element that is currently being processed. Once the programming of the EPROM is completed, the message 'completed' will be displayed.

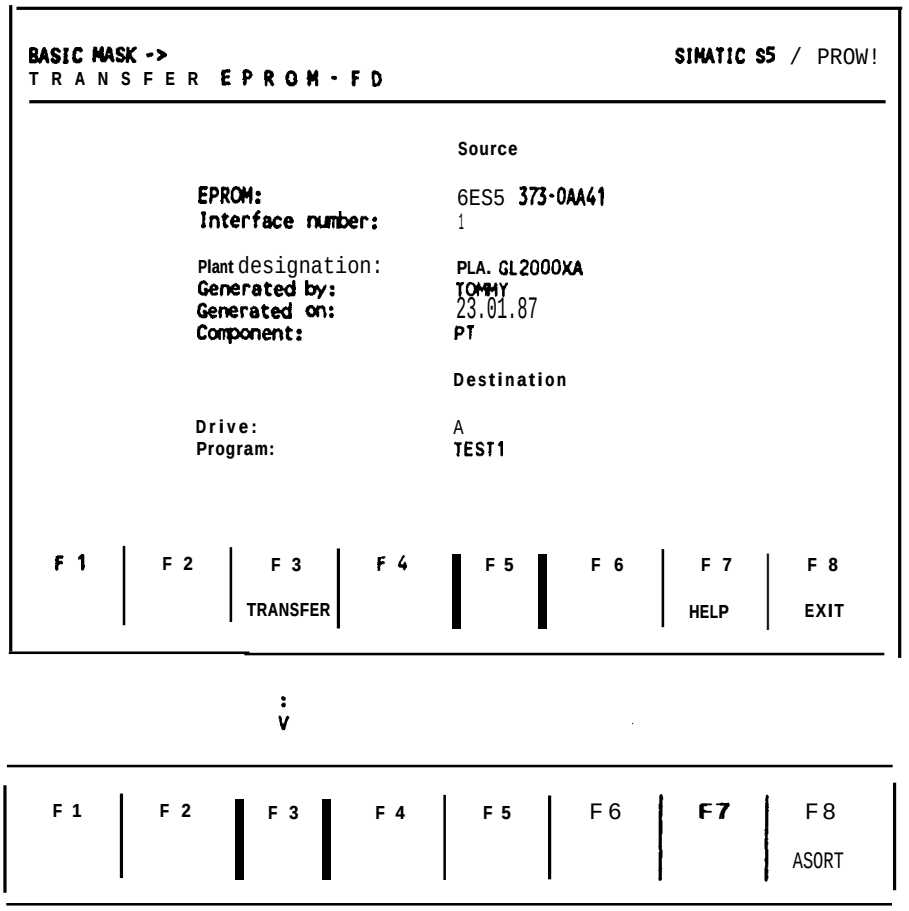
After programming the EPROM, you can press F8 (EXIT) (PROM 525 will prompt you to confirm that you wish to leave the mask) to return to the 'BASIC MASK' via the 'TRANSFER FD - EPROM' mask in **which you once again press F8,**

Pressing F4 (INFO EPROM) to branch to the 'INFO EPROM' mask. Now you can obtain information about the EPROM you have just written to.



Enter the EPROM number in this mask using F7 (HELP). Press F1 (INFO) and the PROM 525 outputs the EPROM and module parameters (SYSID) for the EPROM. The significance of the numbered fields can be found in the reference section of these instructions.

Return to the 'BASIC MASK'. There you see that only the function 'transfer EPROM -> FD' is now missing. You can use this function to transfer the contents of an EPROM-to a floppy or hard disk. In this case you will transfer the program which is now loaded in the EPROM to a floppy disk. Press F2 (TRANSFER EPROM ->J?D).



If necessary, enter the EPROM tier in the 'EPROM' field using F7 (HELP). If the EPROM belongs to a module of the type CP 525-2, the field 'interface number' appears below this. This is assigned a default (when both or only the first interface is programed = "1"; only interface 2 is programmed = "2"). You can change the default with F7 (HELP), At the same time the plant designation, generated by, generated on and component fields are output (also applies to the CP 524),

In the example you will transfer the contents of the EPROM to a floppy disk. Insert a formatted disk in drive A: and enter 'A' in the 'drive" field using F7 (HELP). Write the name you wish to give your program on the floppy disk in the 'program" field (F7 cannot be used here) and then press F3 (ENTER). At this point the PROM 535 prompts you to confirm whether you wish tostartthe transfer or not. With "YES" you start the transfer. The second menu now appears on the screen with which it is possible to stop the transfer using F8 (ABORT). The name of the element currently being transferred appears in the message line, where the end of the transfer will also be displayed.

This completes the example. You can exit the PROM 525 program with F8 (EXIT) and return to the 'BASIC MASK' and from there to the S5 command interpreter.

3 Reference Section

This section contains all the masks displayed by PROM 525. There is also a description of the function keys and fields in the masks.

3.1 General Notes

The input fields (displayed inversely on the screen) are represented in the manual as '##########' and the output fields as '$$$$$$$$'.

Before aborting a transfer and before overwriting existing parts of the program PROM 525 outputs a request or prompt and the following acknowledgement menu:

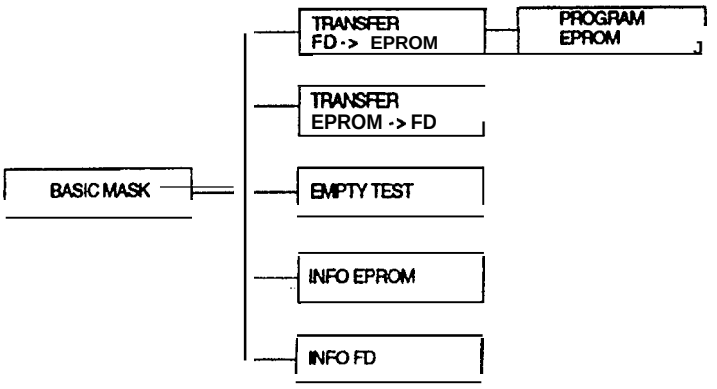
F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
YES		NO					

PROM 525 requires a positive or negative acknowledgement before it continues the program.

Some fields can have entries made using F7 (HEW); these fields are marked with '*' in the tables describing the fields.

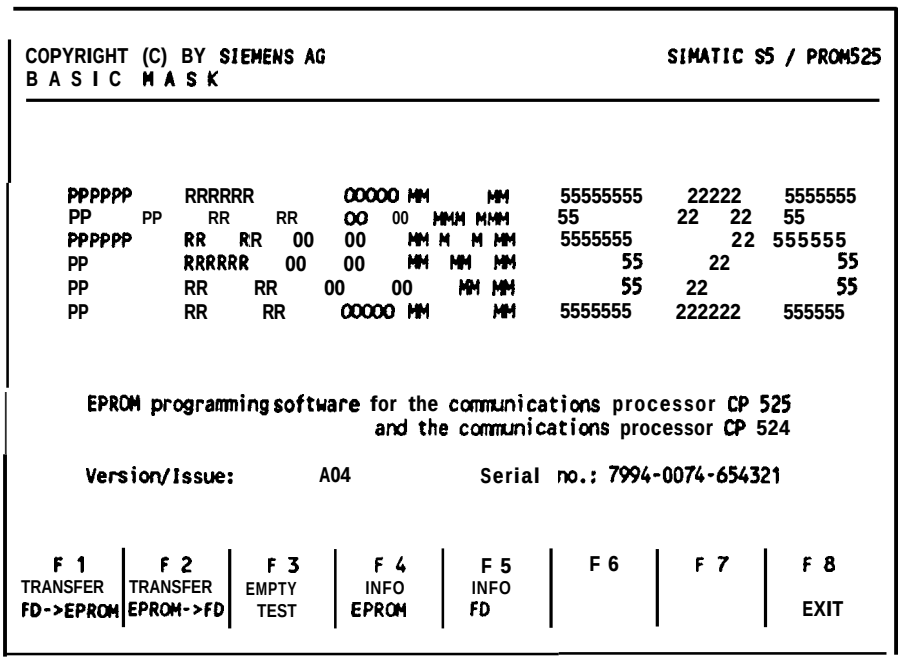
3.2 Overview of the Masks

The following overview shows you how to call each mask:



3.3 Basic Mask

0s11 the 'BASIC MASK' (after positioning the cursor on the line 'PROM 525 EPROM programing CP 525/524*) with the function key F1 (PACKAGE) in the S5 command interpreter.



F1: branch to the 'TRANSFER FD - EPROM' mask.

F2: branch to the **'TRANSFER EPROM - FD' mask.**

F3: branch to the **'EMPTY TEST' mask.**

F4: **branch** to the **'INFO EPROM'** mask.

F5: branch to the 'INFO FD' mask.

F8: return to the S5 command interpreter.

3.4 Transfer FD->EPROM

Call the 'TRANSFER FD - EPROM' mask with the function key F1 (TRANSFER FD->EPROM) in the 'BASIC MASK'.

Mask for CP 525-2

BASIC MASK ->
TRANSFER FD-EPROM
SIMATIC S5 / PROM525

Module: #######

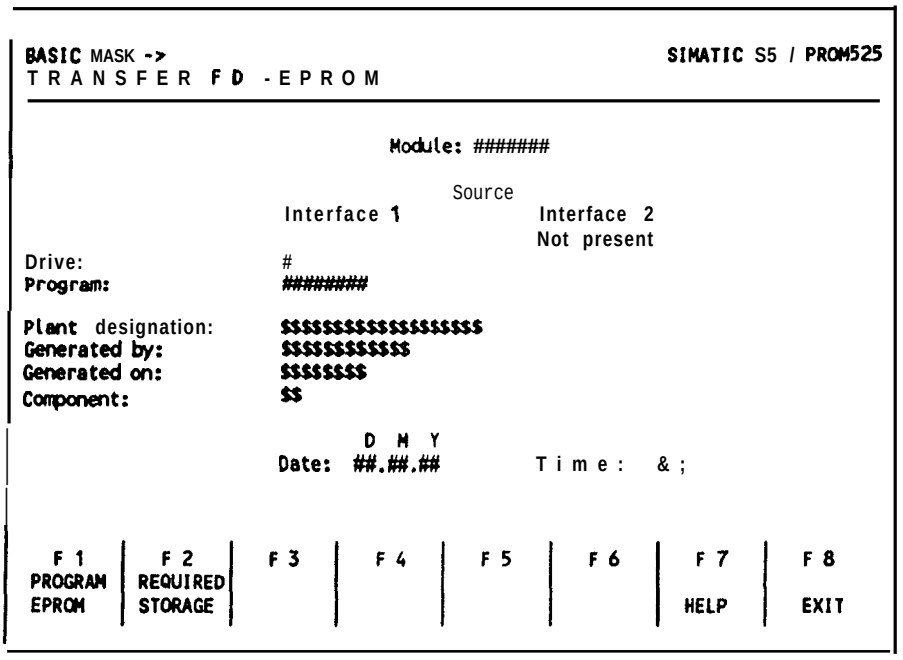
	Source	
	Interface 1	Interface 2
Drive:	#	#
Program:	########	########
Plant designation:	$$ssssssssssssssssss	$$$$$$$$$$$$$$$$$$$$
Generated by:	$$$$$$$$$$$$$$	$$$$$$$$$$$$$$
Generated on:	$$$$$$$$	$$$$$$$$
Component:	$$	$$

D M Y
Date: ##.##.##

H M
Time: ##.##

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
PROGRAM EPROM	REQUIRED STORAGE					HELP	EXIT

Mask for CP 524



F1: branch to the **'PROGRAM EPROM'** mask.

F2 display of the minimum memory requirement (program length) **of the** specified program or the two programs if both interfaces are used,

Note: the specified program length always contains 2 x 8 **Kbytes** for the interpreters and procedures of both interfaces. The actual memory space **required** by the program is always rounded up to a whole multiple of 8 **Kbytes** for each interface, since the memory is divided into memory pages of 8 **Kbytes.**

F8: return to the *BASIC **MASK'.**

Field		Field type keys allowed	Unlit value	Alternatives
Module		any		CP524/ CP525-2 *
Drive 1)		upper case letters		dependent on device type *
Program 1)		letters/ numbers		*
Plant designation 1)		OUTPUT FIELD		
Generated by 1)		OUTPUT FIELD		
Generated on 1)		OUTPUT FIELD		
Component 1)		OUTPUT FIELD		
Date	D	numbers	1 - 31	
	M	numbers	1 - 12	
	Y	numbers		
Time	H	numbers	0 - 23	
	S	numbers	0 - 59	

F7 (HELP) can be used to make an entry in the fields marked '*'.

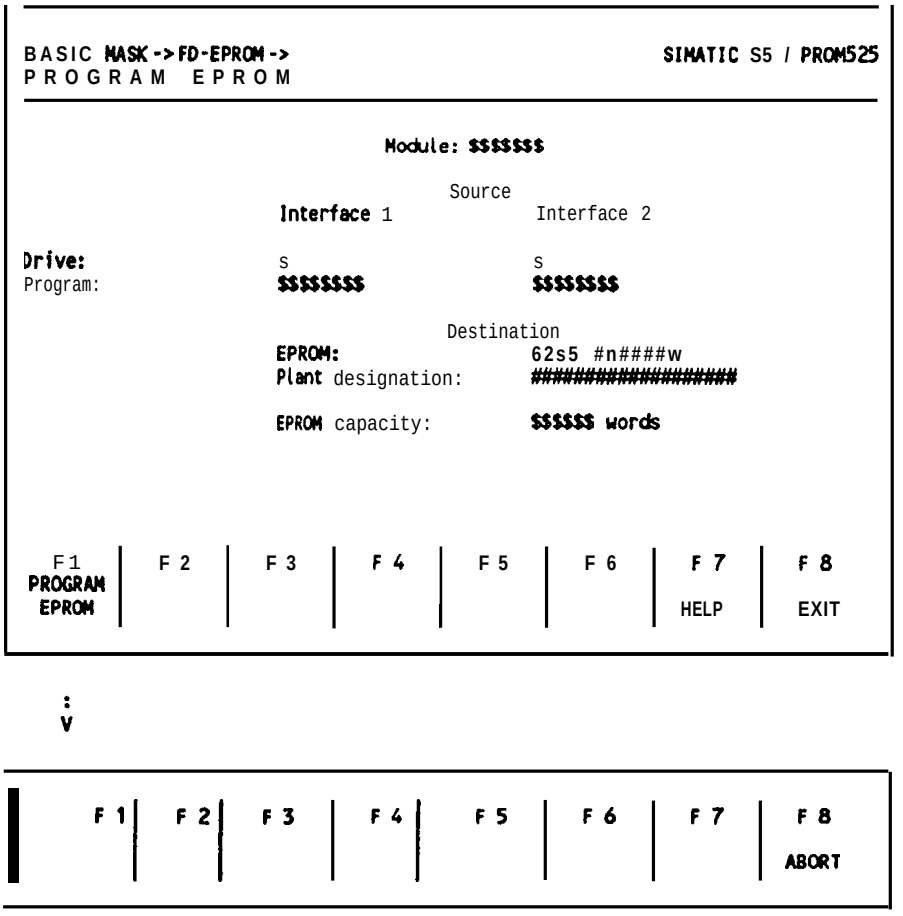
1) If two interfaces are being used (CP 525-2) these specifications apply to the fields under 'interface 1' and under ' interface 2.

3.5 Programming the EPROM

Call the 'PROGRAM EPROM' mask with the function key F1 (PROGRAM EPROM) in the 'TRANSFER FD - EPROM' mask.

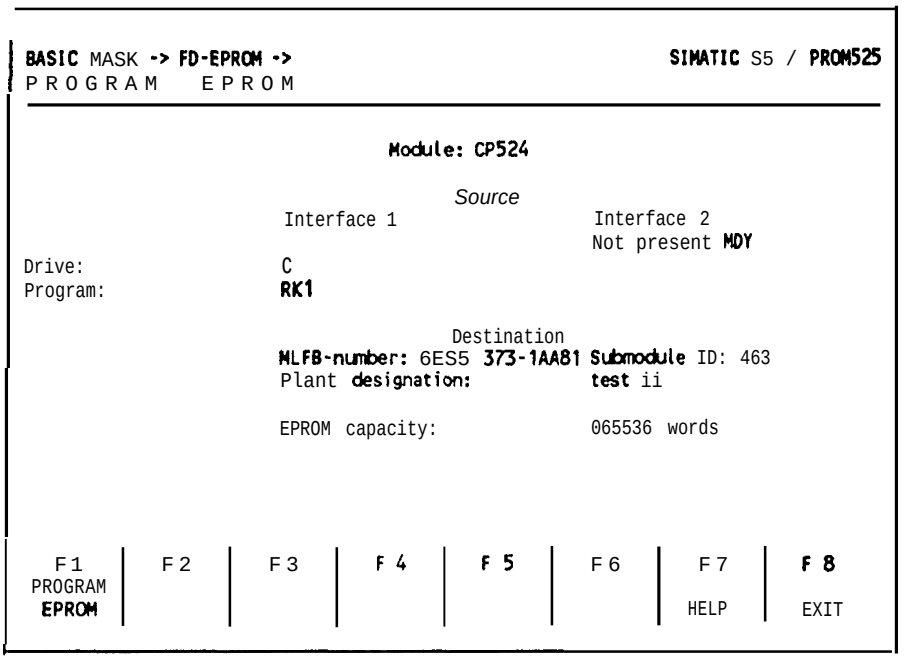
The floppy disk must not be changed while this mask is displayed!

Mask for CP 525-2

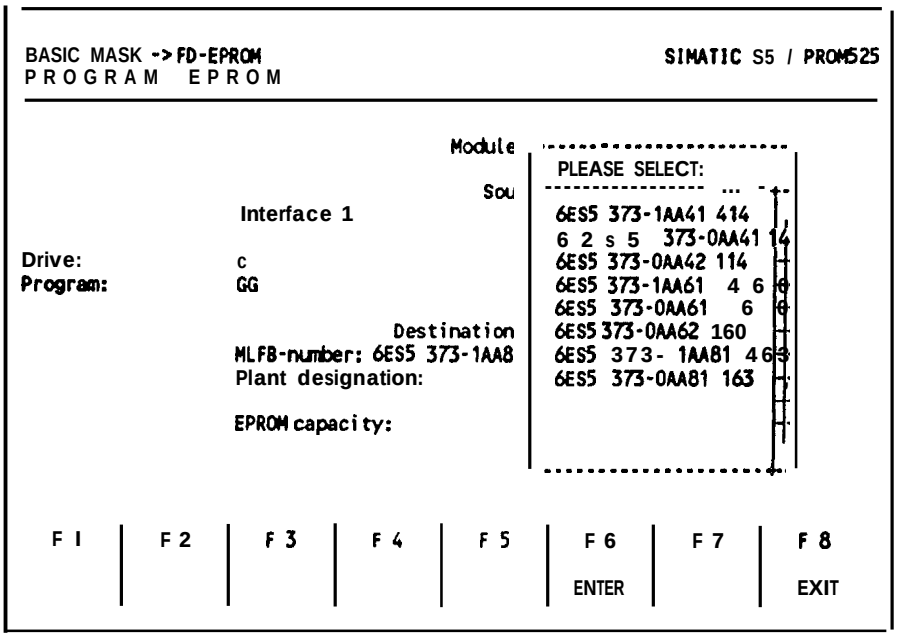


:
V

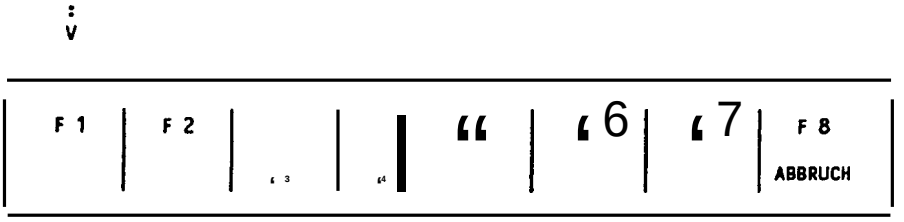
Mask for COM 524



When pressing F7 (HELP) the following mask appears with a selection list.



Using the arrow keys you can select en MLFB number and acknowledge it by pressing F6 (ENTER).



First menu:

F1: setting up the directory of the user programs to be transferred.
A submask is displayed that indicates the memory required, the programing time end the current address.
A check is carried out to establish whether en EPROM is plugged in end whether the EPROM type 5.s permissible.
A further check establishes whether the EPROM is empty.

The programming is started.
The second menu is displayed.
During the entire programming time, the remaining programming time is displayed as well as the current address and name of the element currently being processed.
The message 'completed' is then output.
Note: the actual memory space occupied is displayed. This includes the memory required for the manager and the actual program. The memory required always includes 2 x 8 Kbytes for the interpreter and procedures of both interfaces.

F8: return to the 'TRANSFER FD - EPROM' mask.

Second menu:

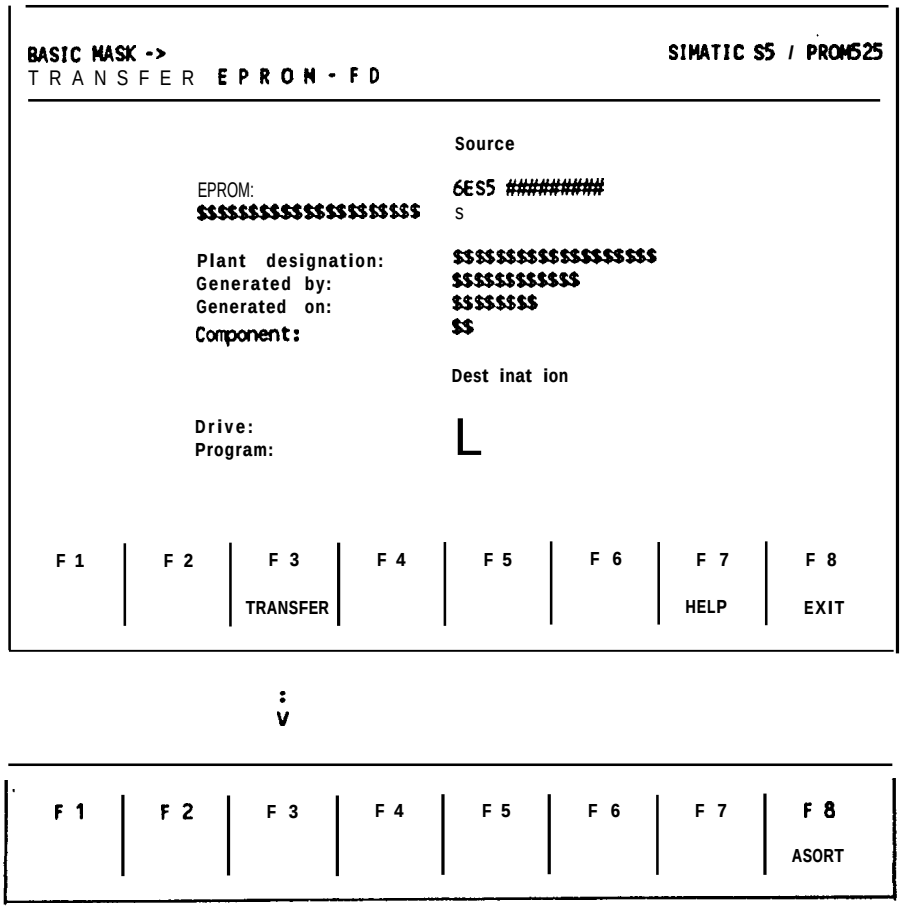
F8: after the current element is programmed, the transfer is aborted the next time the hard or floppy disk is accessed.

Field	Field type keys allowed	Limit value	Alternatives
Module	OUTPUT FIELD	(default from the 'TRANSFER FD - EPROM' mask)	
Drive 1)	OUTPUT FIELD	(default from the 'TRANSFER FD - EPROM' mask)	
Program 1)	OUTPUT FIELD	(default from the 'TRANSFER FD - EPROM' mask)	
Programing number	numbers		
MLFB number	OUTPUT FIELD		
Plant designation	any		
EPROM capacity	OUTPUT FIELD		
Required storage	OUTPUT FIELD		
Programming time	OUTPUT FIELD		
current address	OUTPUT FIELD		-

1) If two interfaces are being used, the specifications apply to the fields under 'interface 1 ' end under 'interface 2'.

3.6 Transferring EPROM->FD

Call the 'TRANSFER EPROM - FD' mask with the function key F2 (TRANSFER EPROM->FD) in the 'BASIC MASK'.



First menu:

F3: checks whether an EPROM is plugged in and whether the EPROM type is permissible,
end whether the EPROM contains a COM 525 program.
The prompt "Start transfer EPROM->FD?" is displayed as well as the acknowledgement menu,
With a positive acknowledgement:
the transfer is started
the second menu is output.
During the transfer: the name of the element currently being processed is displayed.
The message 'completed' is output.
With negative acknowledgement:
the first menu is displayed.

F8: return to the 'BASIC MASK'.

Second menu:

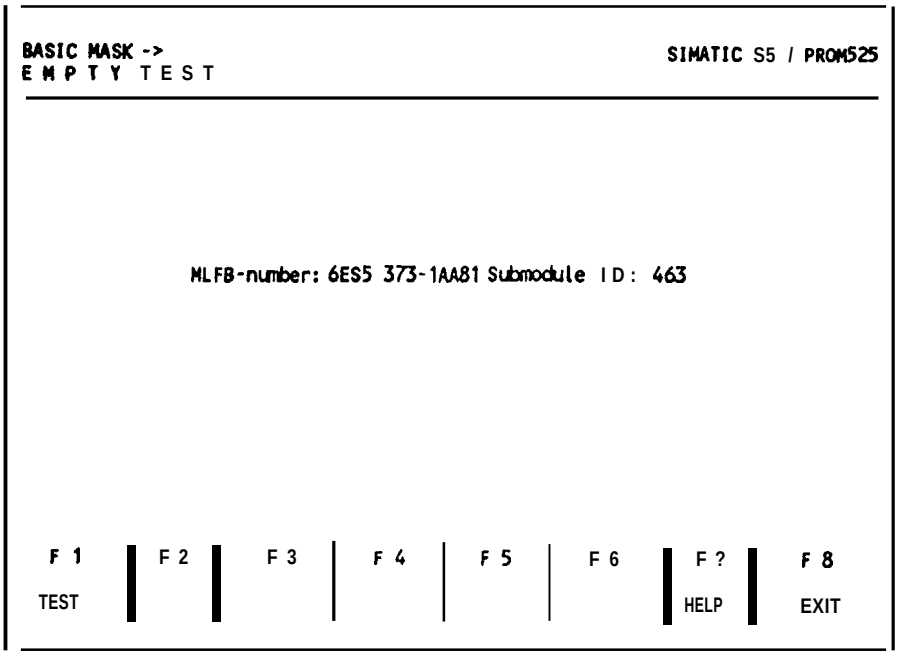
F8: abort the transfer,

Field	Field type keys allowed	Limit value	Alternatives
Programming number	numbers	-	-
MLFB number	OUTPUT FIELD		
Interface number (only with CP 525-2)	numbers		1 / 2 *
Plant designation	OUTPUT FIELD		
Generated by	OUTPUT FIELD		
Generated on	OUTPUT FIELD		
Component	OUTPUT FIELD		
Drive	upper case letters		dependent on device type *
Program	letters/ numbers		

F7 (HELP) can be used to make an entry in the fields marked '*'.

3.7 Empty Test

Call the 'EMPTY TEST' mask with the function key **F3 (EMPTY** TEST) in the 'BASIC MASK'.



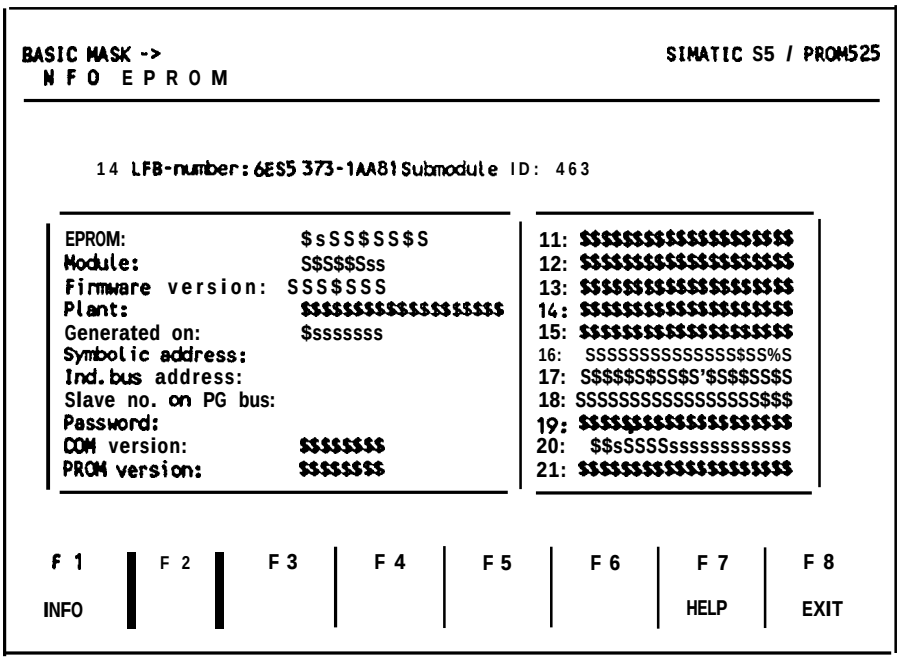
F1: a check is made as to whether an EPROM is plugged in and whether the EPROM type is permissible,
The empty test is started.
During the empty test: the current address is output.
The message 'EPROM is empty! ' or'EPROM not empty! ' is output at the end of the empty test,

F8: return to the 'BASIC MASK'.

Field	Field type keys allowed	Limit value	Alternatives
Programming number	numbers	-	-
MLFB number	OUTPUT FIELD		
Current address	OUTPUTFIELD		

3.8 INFO EPROM

Call the 'INFO EPROM' mask with the function key F4 (INFO EPROM) in the 'BASIC MASK'.



F1: a check is made as to whether an EPROM is plugged in and whether the EPROM type is permissible.
The EPROM and module parameters (SYSID) of the EPROM are output .

F8: return to the 'BASIC MASK' .

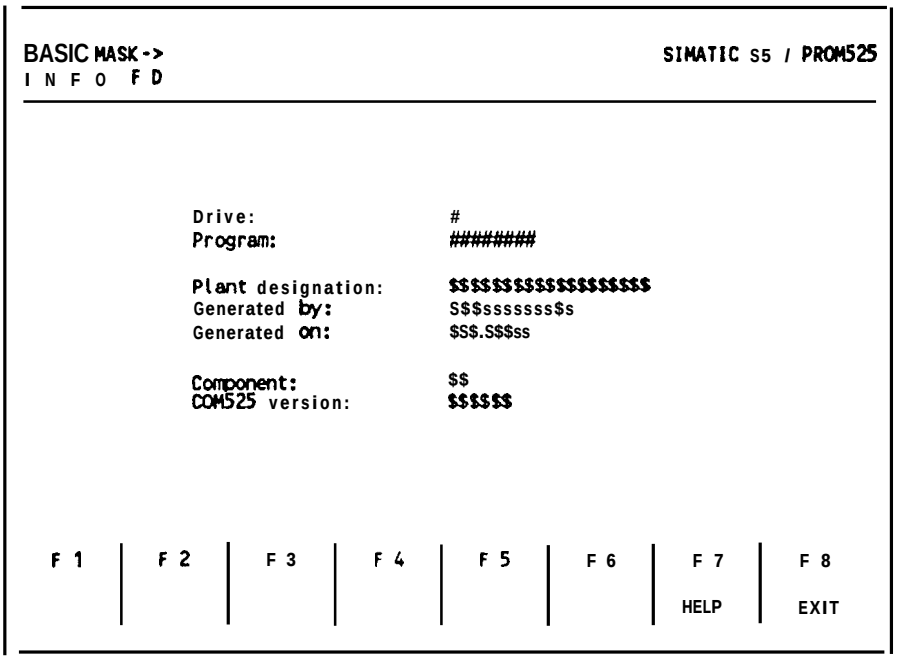
Field	Field type keys allowed	Limit value	Alternatives
EPROM (central field at top)	any		
EPROM (field on left)	OUTPUT FIELD		
Module	OUTPUT FIELD		
Firmware version	OUTPUT FIELD		
Plant	OUTPUT FIELD		
Generated on	OUTPUT FIELD		
symbolic address	OUTPUT FIELD		
Ind. bus address:	OUTPUT FIELD		
Slave no.	OUTPUT FIELD		
COM version	OUTPUT FIELD		
PROM version	OUTPUT FIELD		

Field	Field type keys allowed	Limit value	Alternatives
Data for additional identification:			
11 (component/ interface 1)	OUTPUT FIELD	-	-
12 (name of the interpreter)	OUTPUT FIELD	-	-
13 (version of the interpreter)	OUTPUT FIELD	-	-
14 (name of the procedure)	OUTPUT FIELD	-	-
15 (version of the procedure)	OUTPUT FIELD	-	-
16 (component/ interface 2)	OUTPUT FIELD	-	
17 (name of the interpreter)	OUTPUT FIELD	-	
18 (version of the interpreter)	OUTPUT FIELD	-	
19 (name of the procedure)	OUTPUT FIELD	-	
20 (version of the procedure)	OUTPUT FIELD	-	
21 1)	OUTPUT FIELD	-	

1) This field is not used with the current version of PROM 5250

3.9 INFO FD

Call the 'INFO FD' mask with the function key F5 (INFO FL)) in the *BASIC MASK'.



F8: return to the 'BASIC MASK'.

Field	Field type keys allowed	Limit value	Alternatives
Drive	upper case letters		dependent on device type *
Program	letters/ numbers		*
Plant designation	OUTPUT FIELD		
Generated by	OUTPUT FIELD		
Generated on	OUTPUT FIELD		
Component	OUTPUT FIELD		
COM525 version	OUTPUT FIELD		

F7 (HELP) can be used to mske en entry in the fields marked '*',

4 PROM 525 Messages

When you work with the PROM 525, messages are output in the message line (the line above the fiction key menu) for the following reasons:

- when an error has occurred (error message)

  as information for the user, for example to show that an action has been carried out successfully (messages)

  to prompt the user to confirm "yes" or "no" in the acknowledgement menu (acknowledgement prompts)

A message includes an identifier (e.g. MF, MESS), a number and a short text.

The following section lists and describes all the PROM 525 messages.

4.1 Messages Common to All Masks

Message number	Description / Remedy
MF.001	illegal key! - Illegal characters have been entered in the field in which the cursor is positioned. Refer to the reference section of this user's guide to find the permissible characters.
MF.002	illegal entry! - The entry which has been made in the field on which the cursor is located does not correspond to any of the permissible alternatives, or it exceeds or falls below the permissible limit values. Refer to the reference section of this user's guide for the limit values and the permissible alternatives.

4.2 Messages

Message number	Description
MESS .001	Completed - The end of the transfer is displayed.
MESS .002	Element being processed: - The element named is currently being transferred.
MESS ,003	Aborted - The transfer FD->EPROM has been aborted as required.
MESS .101	EPROM is empty! - Result of the empty test.
MISS .201	HELP will not help here! - Here only the name of an existing program should be specified, if it is to be overwritten.

Message number	Description
MESS.202	EPROM->FD transfer aborted / file deleted! - The transfer was aborted as requested, the file which was initialized will be deleted.
MESS.301	Memory required for #,#; # words - The length of the specified programs will be displayed.
MESS.302	Directories being established ,..
MESS.303	Memory required for #: # words - The length of the specified program will be displayed.
MESS.401	Transfer link being established (takes approx. 70 see)!
MESS.402	External EPROM programmer is ready!

4.3 Acknowledgement Prompts

Acknowledgement prompts are always displayed with the acknowledgement menu; they must be answered with F1 (YES) or F3 (NO), before PROM 525 will continue.

Acknowledgement prompt number	Description
QUIT.001	Exit mask?
QUIT .002	Exit PROM 525?
QUIT .003	FD has been changed - correct FD inserted again?
QUIT.004	Plug in correct EPROM or remove adapter!
QUIT. 005	Please plug in adapter and confirm!
QUIT. 006	Please check EPROM and confirm!

Acknowledgement prompt number	Description
QUIT. 101	EPROM has been changed - correct EPROM plugged in?
QUIT. 102	EPROM not erased - empty EPROM plugged in?
QUIT.103	Break off programing?
QUIT.201	Program already exists - delete before transfer?
QUIT.202	Start trsnsf er EPROM->FD?

4.4 Error Messages

Error message number	Description
ERROR 001	No program name specified - The name of the program must be entered in the field on which the cursor is located.
ERROR 002	Invalid date - time specification! Refer to the reference section of this user's guide for the permissible entries.
ERROR 003	Illegal entry! - Refer to the reference section of this user's guide for the permissible entries.
ERROR 004	No drive specified! - The drive identifier must be entered in the field on which the cursor is currently located.
ERROR 005	Not a COM525 file! - The file specified does not contain a COM 525 program.

Error message number	Description
ERROR 006	Program does not exist! - No program exists under the program name specified.
ERROR 007	Library must not be processed - The name specified is not the name of a program but the name of a library.
ERROR 008	Illegal key! - See MF.001
ERROR 101 ERROR 102	Drive cannot be addressed! - Possible causes: - the floppy disk has not been inserted in the specified drive - the drive has not been closed - the drive or the hard disk is defect.
ERROR 103	Aborted! - The floppy disk was changed during the programing.

Error message number	Description
ERROR 104	Drive is write-protected! - Please remove the write-protect from the drive or from the floppy disk.
ERROR 105	File is write-protected!
ERROR 106 ERROR 107	FD is full! There is not sufficient space on the floppy or hard disk.
ERROR 108	EPROM not empty! - Result of the empty test.
ERROR 109	EPROM - read after write error! - The element which has just been programed does not retch the element in the file. Possible causes: - the EPROM submodule is defect - the EPROM interface is defect - the device is defect.

Error message number	Description
ERROR 110	Wrong EPROM plugged in! - The EPROM number of the submodule plugged in does not match the entry in the mask,
ERROR 111	Adapter not plugged in! - In order to use the 64 Kword sub-module, an MEP adapter must be plugged in.
ERROR 112	Remove adapter !
ERROR 113	S5-DOS diskette removed! - The floppy disk with the S5 operating system should have been removed (see Section 'Notes on the operating systems PCP/M-86 and S5 DOS' in this manual) .
ERROR 114	S5-DOS - drive cannot be addressed! - Applies to the floppy or hard disk with the operating system, see ERROR 101.
ERROR 115	Hardware error! - EPROM submodule or device is defect.

Error message number	Description
ERROR 116	No readable program! - The SYSID of the element is not readable
ERROR 117	Interface not programed! - Enter a different interface number with F7 (HELP).
ERROR 118	No space for interface directory! - The capacity of the submodule is not sufficient for the interface directory.
ERROR 119 ERROR 120	EPROM does not have enough capacity! - The capacity of the submodule is not sufficient for the specified program.
ERROR 121	EPROM type not intended for programing! - The submodule plugged in is not one of the three permitted module types (see Section 'Introduction' in this user's guide).

Error message number	Description
ERROR 1.22	EPROM exchanged! - After reading SYSID a different EPROM submodule was plugged in.
ERROR 201	No program specified! - The name of the program must be entered in the field on which the cursor is positioned.
ERROR 202	Invalid component specified! - The program contains neither the component PT nor the component CL. Check (and if necessary change) the specification in COM 525.
ERROR 203	# missing on interface #! - The program is not complete. The element named in the message is not present.
ERROR 204	No plant designation specified! - The plant designation must be entered in the field on which the cursor is positioned.

Error message number	Description
ERROR 401	Error in transfer to external EPROM programmer! - Possible causes: - the device was switched off after initialization - the connecting cable is not plugged in correctly - the connecting cable is defect.
ERROR 402	Fault on external EPROM programmer! - Possible causes: - the device is not switched on - the connecting cable is not plugged in correctly - the connecting cable is defect.
ERROR 403	Error in transfer to external EPROM programmer ! an error has occurred in the transfer of the S5 files to the external programmer,
ERROR 404	External EPROM programmer not ready! - The transfer link could not be established.

Error message number	Description
ERROR 405 ERROR 406	S5 DOS not addressable! - One *or more* S5-DOS files are not present. Reload the operating system S5-DOS!

4.5 Internal Errors

Error message number	
INTERNAL ERROR 301 to INTERNAL ERROR 336	

If PROM 525 displays one of these error messages please contact your nearest SIEMENS service department or nearest technical office and inform them of the error message.

SIEMENS

We have checked the contents of 11118 manual for ● graamwu with the hardware ● nd software described. Since deviations cannot be precluded entirely, we can. not guarantee full agreement. However, the data in this manual are reviewed regularly and ● ny necessary cor· motions included in subsequent editions. Suggestions for Imp rovement are welcomed.
Technical date subject 10 change.

Siemens Aktiengesellschaft

C79000-B8576-C546
Gerätewerk Karlsruhe
Printed in the Federal Republic of Germany

SIEMENS

SIMATIC S5
Using the Handling Blocks

Description **Order No. C79000-B8576-C547-06**

Contents Page

1 Introduction

Handling blocks (HDBs) are standard function blocks in the STEP 5 user program. HDBs trigger jobs and carry out the data exchange between the CPU and the CP 525.

"Using the handling blocks" is intended as an overview and as a brief introduction to the functions available with the HDBs. It in no way claims to be comprehensive. The handling blocks are described in detail in their own instruction manuals (see /1/ in the List of further relevant documentation),

Please remember that you cannot use the CP 525 in association with the CPU 921 (S processor) of the S5-135U.

2 The Handling Blocks with the Computer Link

2.1 Overview

You can call handling blocks in your STEP 5 program (mode of representation STL) . The following handling blocks are used with the computer link:

HDB	Function	S5-115U	S5-135U S5-155U	S5-150U
SYNCHRON	synchronises CP and CPU	FB248	FB125	FB185
SEND DIRECT n	starts the SEND job with job number n	FB244	FB120	FB180
FETCH DIRECT n	starts the FETCH job with job number n	FB246	FB122	FB182
SEND ALL	transfers the data from the CPU to the CP	FB244 A-Nr. =0	FB120 A-Nr.=0 or FB126	FB180 A-Nr. =0
RECEIVE ALL	transfers the data from the CP to the CPU	FB245 A-Nr.=0	FB121 A-Nr.=0 or FB127	FB181 A-Nr. =0
CONTROL n	copies the job status of a job into the specified condition codeword	FB247	FB123	FB184
RECEIVE DIRECT 200/218/221/223	only for special jobs	FB245	FB121	FB181
RESET DIRECT 200	resets the error entries in the SYSTAT	FB248	FB124	FB183

n = job number (value range 1 to 223)

n = job number (value range 1 to 223)
The HDBs are located as follows:

- S5-115U and S5-135U/CP 922 (R processor) and CPU 928

The HDBs are in the operating system; with the S5-115U they even have the block headers. With the S5-135U/CPU 922 (R processor) the block headers are on floppy disk and must be loaded in the PC.

- S5-150U and S5-155U:

The HDBs are available as a STEP 5 program on floppy disk and must be loaded in the PC.

2.2 SYNCHRON

This HDB synchronizes the interface between the CPU and CP - during cold restart (OB 20), during a manual warm restart (OB 21) or in an automatic warm restart following power failure (OB 22). The SYNCHRON block must be called up for every CP interface in the start-up organization block of the CPU (OB 20, OB 21 and OB 22). During the synchronization, the maximum field length for the data transfer between the CPU and CP 525 is set.

Note on multiprocessor operation:
it is sufficient when the SYNCHRON is programmed for each interface in the start-up organization blocks (OB 20, OB 21 and OB 22) of one CPU.

Tobesure thattheinterface CPU/CP really is synchronized, observe the PAFE byte of the SYNCHRON (PAFE byte - see Section 2.7.1).

Assigning parameters to SYNCHRON:

The following table shows a description of the parameters that you must specify when the HDB is called. The form and order of these parameters are the same that you see when programming on the'programmer. The letters x and y are variables to be replaced by values when you are programming. For the FB number (PB 125) the ***number*** of the CPU 922 (R processor)/S5-135U is used in this example.

Further information canbefoundinthe instruction manuals for the handling blocks (see /l/ in List of further relevant documentation).

SSNR: KYx,y ;y= number of the required interface according to the jumper setting for the interface number on the CP (for interface number - see CP 525 instructions in this manual)
value range: x > 0 : indirect parameter assignment (see /1/ in list of documentation)
x = 0: Y = interface number values from 0 to 255

BLGR: KY0,y ;field length for data transfer between CP and CPU
value range: : y = o to 255
significance: y= 0 field length (BLGR) - see /1/ in list of docum.
· 1: field length 16 bytes
· 2: field length 32bytes
· 3: field length 64 bytes
= 4: field length 128 bytes
= 5: field length 256 bytes
· 6: field length 512 bytes *)
= 7 *to* 254: field length 512 bytes *)
= 255:field length 51.2 bytes *)

Pm: xxy ; parameter assignment error - see /1/ in list of docum.

*) means: differfrominformation in the instruction manuals for the handling blocks: with the computer link max. 256 bytes are transferred.

2.3 SEND DIRECT n and FETCH DIRECT n

SEND DIRECT n (FETCH DIRECT n) triggers the job with the job number 'n' (n = 1 to 223). Jobs for triggering the processing of a job are termed DIRECT jobs.

For every DIRECT job with the job number 'n' (except special jobs) that you programed in the STEP 5 program, there must be a job with the same job number 'n' stored in the user memory of the CP. You program these jobs with the programming package COM 525.

You start a DIRECT job by calling SEND DIRECT n or FETCH DIRECT n. The job is then entered in the internal job queue in the CP.

Queue

The CP has an internal queue for each interface in which the maximum 10 DIRECT jobs (SEND DIRECT and FETCH DIRECT jobs) can be entered. The CP notes the order in which the jobs are initiated and entered in the queue, the corresponding handling block parameter and sets bit 1 "job running" in the condition codeword for each of these jobs.

The CP processes the jobs in the order in which they are entered in the queue. For SEND jobs and PSEUDO READ/WRITE jobs the CP then requests all the data from the CPU using a SEND ALL. When the job is completed, bit 2 "job finished without error" is set in the condition codeword, otherwise bit 3 "job finished with error" is set. If an error occurs, an error number is entered in bits 8 to 11 of the condition codeword and in the error message area of SYSTAT.

If there are already 10 jobs in the queue, every further job is rejected with the error number 7H in the condition codeword, 15H in the error message area of SYSTAT and CH in the parameter assignment error byte.

A job can only occur once in the queue, since there is only one job status maintained on the CP for each job number.

The CPU cannot query the number of jobs in the queue and jobs cannot be deleted from the queue. After a cold restart or when the CP is synchronized, all entries in the queue are deleted.

If a job depends on another job being processed without errors, you must include an interlock in your STEP5 user program.

Jobs sent by the partner are processed outside the queue,

Special jobs are processed directly and are not entered in the queue. The following jobs are special jobs:

RESET DIRECT 200
RECEIVE DIRECT 200
SEND DIRECT 218
RECEIVE DIRECT 218
RECEIVE DIRECT 221
RECEIVE DIRECT 223

Jobs the partner sends are processed outside the queue,

Assigning parameters to SEND DIRECT n and FETCH DIRECT n

The following table shows description of the parameters that you must specify when the HDB is called. The form and order of these parameters are the same as those you see when you are programming on the programer. The letters x and y are variables to be replaced by values when programming. For the FB numbers the numbers of the CPU 922 (R processor) /S5-135U are used in this example.

Further information can be found in the instruction manuals for the handling blocks (see /l/ in List of further relevant documentation).

SEND DIRECT n: FB120

SSNR: KYx, y ; interface number
value range: x > 0: indirect para. assignment see /1/ in list of documentation
x = 0: y = interface number values from 0 to 255 (see instructions CP 525)

A-NR: KY0,x ;number of the direct job
value range: x=lto223

ANZW: xxy ;condition codeword (a doubleword is occupied)
value range see /1/ in list of documentation
xx = flag word (FW) or data word (DW) (caution: with DW, DB/DX must be called first)
y= number (dependent on xx)

QTYP: KSyy ;source type
value range yy = DB,DX,CB,TB,BS,AS,FY,QB,IB,OB or PB (only these permitted)
XX indirect para. assignment see /1/ in the list of documentation
(Note: with PSEUDO READ/WRITE function A-Nr. 190 to 199 - DB or DX allowed. See also special jobs)

DBNR: KYo,x ;source data block no. with QTYP DB, DX, (XX see above)
value range: x=3t0255
with other QTYP parameter assignment irrelevant

QANF: KF+x ;startaddress of the source
value range: x=dependentonQTYP and PC type; see job tables in the user's guide "Computer link with RK 512"

QLAE: KF+x ;source length - numberofdatatobetrsnsferred
value range: x = dependent on QTYP und PC type; see job tables in the user's guide “Computer link with RK 512"

PAFE: xxy ;parameter assignment error
value range: see /1/ in list of documentation

FETCH DIRECT n: FB122

```
SSNR: KYx, y   ; interface number
               value range: x > 0: indirect pars. assignment
                                   see /1/ in list of
                                   documentation
                            x = 0: y = interface number values
                                   from 0 to 255 (see in-
                                   structions CP 525)
A-NR:  KY0,x   ;number of the direct job
               value range: x = 1 to-223
ANZW:  xxy     ; condition codeword (a doubleword is occupied)
               value range see /l/ in list of documentation
                           xx = flag word (FW) or data word
                                (DW) (caution: with DW, DB/
                                DX must be called first)
                            y = number (dependent on xx)
ZTYP:  KSyy    ;destination type
               value range yy = DB,DX,RS,AS
                                XX indirect para. assignment
                                   see /1/ in the list of
                                   documentation
               (Note: with PSEUDO READ/WRITE function (A-Nr.
               190 to 199) only DB or DX allowed. See also
               special jobs)
DBNR:  KY0, x  ; dest. data block no. with dest. type DB ,DX
                                     (XX see above)
               value range:  x = 3 to 255
               with other ZTYP parameter assignment irrelevant
ZANF:  KF+x    ; start address of the destination
               value range: x = dependent on ZTYP and PC type;
                                see job tables in the
                                user's guide "Computer link
                                with RK 51.2"
ZLAE:  KF+x    ;number of data to be transferred
               value range: x . dependent on ZTYP und PC type;
                                see job tables in the
                                user's guide "Computer link
                                with RK 512"
PAFE:  xxy     ;parameter assignment error
               value range:  see /l/ in list of documentation
```

2.4 SEND ALL and RECEIVE ALL

These jobs are responsible for the data exchange between the CP 525 and the CPU. For ALL functions, the HDBs are assigned the job number 0.

First, the HDB SEND DIRECT n or FETCH DIRECT n triggers a job on the CP 525. When the job is triggered, the CP 525 stores the source or destination parameters you assigned to the SEND DIRECT n or FETCH DIRECT n.

Then the CP 525 performs the job, i.e. it sends an "ALL request" to the CPU. It informs the CPU of the required data source or data destination. The SEND ALL then transfers the requested data from the CPU (source) to the dual-port RAM of the CP. The RECEIVE ALL transfers the data from the dual-port RAM to the CPU (destination). With larger amounts of data, several ALL functions may be necessary to transport the data (see field length).

The source/destination parameters that you must specify for the direct functions are irrelevant for ALL functions (any values can be entered here).

Note: if you have a long cycle time with your STEP 5 program, it may be a&sable to call up SEND ALL and RECEIVE ALL several times per cycle. SEND ALL and RECEIVE ALL calls extend your cycle time only when there is actually data to be transferred.

The SEND ALL and RECEIVE ALL HDB's in association with partner jobs

These HDBs are used not only to transport data for jobs from their own CPU, but when working with the computer link, they also transfer data the partner is sending or fetching on its own initiative.

If the partner sends a telegram, the RK 512 interpreter analyses it. If it recognises that the partner wishes to send or fetch data, it extracts the source and destination parameters contained in the telegram header. The interpreter sends an 'ALL request" to the CPU so that the information data can be transferred.

The ALL HDB then executes the data exchange between the CPU and dual-pm% RAM.

Assigning parameters to **SEND ALL and RECEIVE ALL:**

The following tables contain a description of the parameters that you must specify when the HDB is called. The foxm and order of these parameters are the same that you see when programming on the PG. The letters x and y are variables that must be replaced by values when you program. For the FB numbers the numbers of the CPU 922 (R processor)/S5-135U are used in this example.

Further **information can be** found **in the instruction** manuals for the handling blocks (see /l/ in List of further relevant documentation).

SEND ALL: **FB126**

(with the **S5-115U/150U** the call of the SEND DIRECT **n with** job number 0 is used. The specifications for the source parameters are then unnecessary).

SSNR: **KYx,y** ;**interface number**
value range: x > 0: **indirect para.** assignment see /1/ in list of documentation
x = 0: y = interface number values from 0 to 255

A-NR: **KY0,0** ;**number** of the job

ANZW: **xxy** ;**condition** codeword
value range: see /1/ in list of d**ocumentation**
xx = flag word (FW) or data word (DW) (caution: **with DW,** DB/DX **must be** called **first!)**
y =number (dependent **on xx)**

PAFE: **xxy** ;**parameter** assignment error
value range: see /1/ in list of d**ocumentation**

RECEIVE ALL: FB127

(with the S5-115U/150U, the call of the RECEIVE DIRECT n with job number 0 is used. The specifications for the source parameters are then unnecessary).

```
SSNR:  KYx,y    ;interface number
                 value range: x > 0: indirect para. assignment
                                     see /1/ in list of
                                     documentation
                              x = 0: y = interface number values
                                         from 0 to 255 (see in-
                                         structions CP 525)
A-NR:  KY0,0    ;number of the job

ANZW:  xxy      ;condition codeword
                 value range: see /l\ in list of documentation
                              xx = flag word (FW) or data word
                                   (DW) (caution: with DW, DB/
                                   DX must be called first.)
                               y =number (dependent on xx)
PAFE:  xxy      ;parameter assignment error
                 value range: see /1/ in list of documentation
```

2.5 CONTROL

The CONTROL HDB copies the job status of a job into the specified condition codeword. The job status provides information about operating or processing statuses.

CONTROL can be called at any point in the program.

- - parameters to CONTROL:

The following table shows a description of the parameters that you must specify when the HDB is called, The form and order of these parameters are the same that you will see when you program on the PG. The letters x and y are variables that must be replaced by values when you program. For the FB number (FB123), the number of the CPU 922 (R processor) /S5-135U is used in this example.

Further information can be found in the instruction manuals for the handling blocks (see /l/ in List of further relevant documentation) .

```
SSNR:  KYx, y    ; interface number
                 value range: x > 0:  indirect para. assignment
                                      see /1/ in list of
                                      documentation
                              x = 0:  Y = interface number values
                                          from 0 to 255 (see in-
                                          structions CP 525)
A-NR:  KY0 ,x    ;number of the job to be controlled
                 value range: x = 1 to 223
                              x = 0 number of the job currently
                                    being processed is copied
                                    into the ANZW

ANZW:  xxy       ;condition codeword
                 value range: see /1/ in list of documentation
                              xx = flag word (FW) or data word
                                   (DW) (caution: with DW, DB/
                                   DX must be called first!)
                               Y = number (dependent on xx)
```

2.6 RECEIVE DIRECT n and RESET DIRECT 200

These are used only with special jobs. The following special functions are carried out with RECEIVE DIRECT n:

- reading the error message area in SYSTAT
- reading the whole status area SYSTAT
- reading the identification area. SYSTAT
- reading the date and time

The special job RESET DIRECT n allows you to reset the error entries in SYSTAT.

The description of the special jobs can be found in the user's guide "Computer link with RK 512" in this manual.

Assigning parameters to RECEIVE DIRECT n and RESET DIRECT n:

The following two tables give a description of the parameters that you must specify when the HDB is called. The form and order of these parameters are the same as those you see when you program on the programmer. The letters x and y are variables that must be replaced by values when you program. For the FB numbers the numbers of the CPU 922 (R processor) /S5-135U are used in this example.

Further information can be found in the instruction manuals for the handling blocks (see /1/ in List of further relevant documentation).

RECEIVE DIRECT n: FB121 - only for special jobs

SSNR: KYx,y ;interface number
value range: x > 0: indirect para. assignment
see /1/ in list of
documentation
x = 0: y = interface number values
from 0 to 255 (see in-
structions CP 525)

A-NR: KY0,x ;numberof the direct job
value range: x = 200, 218, 221, 223
(callable special jobs)

ANZW: xxy ;condition codeword (a doubleword is occupied!)
value range see /1/ in list of documentation
xx = flag word (FW) or data word
(DW) (caution: with DW, DB/
DX must be called first)
y = number (dependent on xx)

ZTYP: KSyy ;destination type
value range yy = DB orDX
XX indirect para. assignment
see /1/ in the list of
documentation
(special jobs)

DBNR: KY0,x ;dest. DB no. with ZTYP DB, DX; (XX see above)
value range: x = 3 t0 2 5 5

ZANF: KF+x ;start address of destination
value range: x = 0 to length of dest. type minus
ZLAE

ZLAE: KF+x ;number of data to be transferred
value range: x= dependent on special job

PAFE: xxy ;parameter assignment error
value range: see /l/ in the list of docum.

- DIRECT 200: FB124 - only for special jobs

SSNR: KYx,y ;interface number
value range: x > 0: indirect para. assignment
see /1/ in list of
documentation
x = 0: y = interface number values
from 0 to 255 (see in-
structions CP 525)

A-NR: KY0,200 ;number of the direct job
(special job)

PAFE: xxy ;parameter assignment error
value range: see /1/ in the list of docum.

2.7 Evaluation

The following sections contain information about how to evaluate the parameter assignment error byte (PAFE byte) and the condition codeword ANZW.

2.7.1 The Parameter Assignment Error Byte (PAFE Byte)

When using the handling blocks SEND DIRECT n, FETCH DIRECT n, RECEIVE DIRECT n, RESET DIRECT 200, SEND ALL, RECEIVE ALL and SYNCHRON, you must specify a **parameter assignment error byte (PAFE-BYTE)**.

If the HDB recognizes an error, it writes the corresponding error number into the PAFE byte. If the block runs through without errors, the HDB writes OOH into the PAFE byte.

In addition to the errors that can be attributed directly to the parameter assignment, the PAFE byte also indicates errors that occur during the communication CPU/CP 525. If such errors occur you should immediately remedy the situation to avoid other messages appearing during operation.

In certain exceptional cases, errors are indicated in the PAFE byte during operation. These errors are caused by the software of the CP 525 in special situations as follows:

if more than 10 DIRECT jobs are processed in the cycle, i.e. 10 jobs are already signalled as "running" and you want to start an eleventh job

if the STEP 5 program sets the IPC (coordination flag) flag when the CPhas just received a telegram for this coordination flag

if the whole or part of the CP 525 user program is transferred to the CP 525 with COM 525 during normal operation

if a SEND or FETCH job is started before the SYNCHRON job has run through once without errors

During multiprocessor operation you can also receive a PAFE message if several CPUs access the same CP 525 interface at the same time. The access is prevented (with a PAFE message) until the current CPU/CP communication is completed. This PAFE message does not indicate an error but shows that access is momentarily blocked.

The following list provides a brief explanation of the PAFE error messages. Note that the significance of each error number depends on the PC' type. Check the instructions for the handling blocks for your PC (see /1/ in the list of further relevant documentation).

PAFE no.	Significance
11H	Wrong ORG format: source/destination parameter type wrong, area (start address, length) not permissible.
21H	DB/DX or memory area not present,
31H	Source/destination too small.
41H	Source/destination area not present (not plugged in) Timeout (QVZ) from this area,
51H	Error in condition codeword.
61H	No source/destination parameters with SEND/RECEIVE ALL.
71H	Interface does not exist (QVZ for the CP 525). Check the jumper settings for the interface number on the CP 525.
81H	Interface no: ready or not synchronized. Check that the SYNCHRON HDB is programmed in the start-up OBS and runs through without errors.
91H	Interface overload; part of a program is being transferred from the PC to the CP 525.

91H Interface overload; part of a program is being transferred from the PC to the CP 525.

A1H CP interface being used by a different CPU in multiprocessor operation (S5-135U and S5 155U) when several CPUs access one CP interface.

B1H Job number too high.

C1H Error in handshake with the CP 525:

- CP does not reply within the monitoring time
- or CP rejects handshake, e.g. because more than 10 jobs were triggered simultaneously (when 10 jobs are running, bit 1 is set in the ANZW)
  or
- the coordination flag was set when a telegram was received for this coordination flag

D1H Field length (SYNCHRON) wrong and group error for other handshake errors.

E1H DB call missing with indirect *parameter* assignment and group error for HDB software error.

F1H Double block call (when a block can be interrupted at command boundaries).

2.7.2 The Condition Codeword (ANZW)

The condition codeword with direct jobs:

The first part of the condition codeword contains information about the status of the job processing - the second part contains information about the length of the data transferred.

If the direct jobs SEND DIRECT n, FETCH DIRECT n, RECEIVE DIRECT n are being used, the condition codeword requires a doubleword.

Significance of the bits in the condition codeword of a DIRECT job:

Bit 0 : RECEIVE job ready (handshake acceptable)
0 = RECEIVE disabled
1 = RECEIVE enabled

Bit 1 : job running
0 = SEND/FETCH enabled
1 = SEND/FETCH disabled

Bit 2 : job complete without error
Bit 3 : job complete with error

Bit 4 : data transfer/data reception running

Bit 5 : data transfer complete
Bit 6 : data reception complete

Bit 7 : 0 = data transfer/reception enabled
1 = data transfer/reception disabled

Bit 8-11 : error messages
see the error tables in the user's guide 'Computer link with RK 512" in this manual

Bit 12-15 : not used

In the second half of the condition code doubleword, the SEND ALL and RECEIVE ALL enter the. length of the transferred data. SEND ALL enters the number of pieces of data of the current transmission, and RECEIVE ALL adds the current number of pieces of data to the number of pieces of data already received. If several active ALL functions are required for a RECEIVE ALL to carry out the direct job, the value in the second part of the condition code doubleword increases centinuously.

Every direct job should have its own condition code doubleword reserved. The following areas can be used:

- the flag area FW 0 to 252
  or
- the data words 0 to 254 in the currently open data block (DB or DX).

To ensure reliable data exchange it is sufficient with the CPU 922/S5 135U when the STEP 5 program evaluates bits 1, 2 and 3 of the ANZW:

- bit 1 indicates that the job was accepted for processing by the CP. The job can be triggered again only after bit 1 has been reset

- bits 2 and 3 indicate that the CP completed processing the job.

  On completion of the job without errors (bit 2 = 1), new information data can be prepared for the next transfer or the data arriving can be processed.

  If the job is terminated with an error (bit 3 = 1), the error number is entered in bits 8 to 11 of the ANZW. The list of error numbers can be found in the user's guide "Computer link with RK 512" in this manual.

The condition codeword with ALL jobs:

In ALL jobs, RECEIVE ALL or SEND AIL, the condition codeword requires only one data word or flag word.

While the ALL function is running, the number of the DIRECT job for which the data is being transferred is indicated in the ANZW. The value 0 is entered if the function is run through without any data transfer.

If the ALL function is active for its own CPU, it also updates the condition codeword of the corresponding direct job.

If the ALL function is active for a partner job it indicates the byte number of the coordination flag specified in the telegram in the AM?. This number is treated as a job number by the HDB and must therefore have values only between 1 and 223. If no coordination flag was specified (or the number is too high) FFH is entered in the ANZW.

If several ALL functions are required for a partner job with a coordination flag, the byte number of the coordination flag is entered in the ANZW only with the last data transfer. From this it is possible to recognise that the partner job is complete.

Then you can set the IPC flag (which corresponds to the coordination flag) with the STEP 5 program and process the information data that has arrived (if the partner activated a SEND job) or make new information data available (if the partner has activated a FETCH job). The set coordination flag prevents the partner accessing the data area again.

When the IPC flag is reset, this data area can be accessed tmce again.

2.8 Example, Using a Coordination Flag

A SEND job, sent by the Partner is to be protected against overwriting at the receiver by coordination flag 101.0 in the STEP 5 progrsm as follows:

1. The IPC flag must be enabled on the receiver CP by means of jumper settings, i.e. in this example jumper 5-12 cm jumper block number 25 cm the CP must be inserted. Correspondingly, the flag byte 101 in DB 1 of the S5-115U, S5-135U and S5-155U or with standard FB 186 in the S5-150U must be defined as an IPC output flag in the STEP 5 program.

2. Evaluation in the PC cycle of the receiver:

```
e.g. : FB  1            FB to be written to by CP 525
NAME :AUSWERT
     : .                any program
     : .
     :JU  FB127         RECEIVE ALL call
NAME :REC-ALL
       .
ANZW:FW12               parameter assignment of condition codeword
       .
       .
     :L   FY13          load low byte of condition codeword
     :L   KF+101        load byte number of coordination flag
     : !=F              corresponding telegram received? "
     :S   F 101.0       set coordination flag
     : .
     : .                evaluation of the data
     :
     :A  F 1. O         all data evaluated
     :R F1O1. O         i.e. enable coordination flag
     : .
     : .                program continues
     :BE
```

The procedure is similar to prevent data being prepared for a FETCH job from being accessed. In the programmable controller from which the data is to be fetched, the coordination flag must be evaluated in the ANZW of the SEND ALL. The FETCH job of the partner does not receive any data as long as the coordination flag is set.

3 The Handling Blocks with the Printer

3.1 Overview

You csn call handling blocks in your STEP 5 program (mode of representstion STL). The following handling blocks are used with the printer:

HDB	Function	S5-115U	S5-135U S5-155U	S5-150U
SYNCHRON	synchronises CP and CPU	FB248	FB125	FB185
SEND DIRECT n	starts the SEND job with job number n	FB244	FB120	FB180
SEND ALL	trsnsfers the data from the CPU to the CP	FB244 A-Nr. =0	FB120 A-Nr.=0 or FB126	FB180 A-Nr. =0
CONTROL n	copies the job status of a job into the specifiec condition codeworc	FB247	FB123	FB184
RECEIVE DIRECT 200/ 218/221/223	only for special jobs	FB245	FB121	FB181
RESET DIRECT 200	resets the error entries in the SYSTAT	FB248	FB124	FB183

n = job number (value range 1 to 223)

The HDBs are located as follows:

S5-115U and S5-135U/CPU 922 (R processor) and CPU 928:

The HDBs are in the operating system; in the S5-115U they have block headers. With the S5-135U/CPU 922 (Reprocessor) the block headers are on floppy disk and must be loaded in the Pc.

S5-150U and S5-155U:

The HDBs are available as a STEP 5 program on floppy disk and must be loaded in the PC.

3.2 SYNCHRON

This HDB synchronizes the interface between the CPU and CP during cold restart (OB 20), during a manual warm restart (OB 21) or in an automatic warm restart following power failure (OB 22). The SYNCHRON block must be called up for every CP interface in the start-up organisation block of the CPU (OB 20, OB 21 and OB 22). During the synchronization, the maximum field length for the data transfer between the CPU and CP 525 is set.

Note on multiprocessor operation:
it is sufficient when the SYNCHRON is programmed for each interface in the start-up organisation blocks (OB 20, OB 21 and OB 22) of one CPu.

To be sure *that the* interface CPU/CP really is synchronised, observe the PAFE byte of the SYNCHRON (PAFE byte - see Section 3.8. 1) .

Assigning parameters to SYNCHRON:

The following table contains a description of the parameters you must specify when the HDB is called. The form and order of these parameters are the same as those you see when you program on the PG. The letters x and y are variables which must be replaced by values when you program. For the FB number (FB 125) the number of the CPU 922 (R processor)/S5-135U is used in this example.

Further information can be found in the instruction manuals for the handling blocks (see /1/ in List of further relevant documentation).

SSNR: KYx,y ;y =number of the required interface according to the jumper setting for the interface number on the CP (for interface number - see CP 525 instructions in this manual)
value range: x > 0: indirect parameter assignment (see /1/ in list of documentation)
x = 0: y = interface number values from 0 to 255 (see instructions CP 525)

BLGR: KYx,y ;field length for data transfer between CP and CPU
value range:: x= 0
y = 0 to 255
significance: y = 0 field length (BLGR) - see /1/ in list of docum.
= 1: field length 16 bytes
= 2: field length 32 bytes
= 3: field length 64 bytes
= 4: field length 128 bytes
= 5: field length 256 bytes
= 6: field length 512 bytes
= 7 to 254: field length 512 bytes
= 255:field length 512 bytes

PAFE: xxy ;parameter assignment error - see /1/ in list of &~.

3.3 SEND DIRECT n

SEND DIRECT n triggers the job with the job number 'n' (n = 1 to 223), The job for triggering the processing of a job is termed DIRECT job.

For every DIRECT job with the job number 'n' (except special jobs) you programmed in the STEP 5 program, there must be a job with the same job number 'n' stored in the user memory of the CP. These jobs are programmed with the programming package COM 525.

You start a DIRECT job by calling SEND DIRECT n. The job is then entered in the internal job queue in the CP.

Queue

The CP has an internal queue for each interface in which the maximum 10 DIRECT jobs (SEND DIRECT jobs) can be entered. The CP notes the order in which the jobs are initiated and entered in the queue, the corresponding handling block parameter and sets bit 1 "job running" in the condition codeword for each of these jobs.

The CP processes the jobs in the order in which they are entered in the queue. With jobs for sequential message logs and current message logs the CP then requests all the data from the CPU using a SEND AIL. When the job is completed, bit 2 “job finished without error" is set in the condition codeword, otherwise bit 3 “job finished with error” is set. If an error occurs, an error number is entered in bits 8 to 11 of the condition codeword and in the error message area of SYSTAT.

If there are already 10 jobs in the queue, every further job is rejected with the error number 7H in the condition codeword, 15H in the error message area of SYSTAT and CH in the parameter assignment error byte.

A job can only occur once in the queue, since there is only one job status maintained on the CP for each job number.

The CPU cannot query the number of jobs in the queue and jobs cannot be deleted from the queue. After a cold restart or when the CP is synchronized, all entries in the queue are deleted.

If a job depends on another job being processed without errors, you must include an interlock in your STEP5 user program.

Special jobs are processed directly and are not entered in the queue. The following jobs are special jobs:

RESET DIRECT 200
RECEIVEDIRECT200
SEND DIRECT 218
RECEIVE DIRECT 218
RECEIVE DIRECT 221
RECEIVEDIRECT223

Assigning parameters to SEND DIRECT n

The following table contains a description of the parameters that you must specify when the HDB is called. The form and order of these parameters are the same that you see when you program on the PG. The letters xandy are variables that must be replaced by values when you program. For the FB number (FB120) the number of the CPU 922 (R processor)/S5-135U has been used in this example.

Further information can be found in the instruction manuals for the handling blocks (see /l/ in List of further relevant documentation).

SEND DIRECT n: FB120

SSNR: KYx,y ;interface number
value range: x > 0: indirect para. assignment see /1/ in list of documentation
x = 0: y = interface number values from 0 to 255 (see instructions CP 525)

A-NR: KY0,x ;number of the direct job
value range: x = 1 to 223

ANZW: xxy ;condition codeword (a doubleword is occupied)
value range see /1/ in list of documentation
xx = flag word (FW) or data word (DW) (caution: with DW, DB/DX must be called first)
y = number (dependent on xx)

QTYP: KSyy ;source type
value range yy= DB (only data block permitted)
NN (only job initiation, no data transfer)
XX indirect para. assignment see /1/ in the list of documentation

DBNR: KY0,x ;source data block no.
value range: x=3t0255
with other QTYP parameter assignment irrelevant

QANF: KF+x ;start address of the source
value range: x=0datatransfer always from DW0

QLAE: KF+x ;source length - number of data to be transferred
value range: x=1to256

PAFE: xxy ;parameter assignment error
value range: see /1/ in list of documentation

3.4 SEND ALL

This is responsible for the data exchange between the CP 525 and the CPU. For the ALL function, the HDB is assigned the job number 0.

First, the HDB SEND DIRECT n triggers a job on the CP 525. When the job is triggered, the CP 525 stores the source parameters which you have assigned to the SEND DIRECT n.

Following this, the CP 525 carries out the job, i.e. it sends an "ALL request" to the CPU. It informs the CPU of the required data source. The SEND ALL then transfers the requested data from the CPU (source) to the dual-port RAM of the CP. With larger amounts of data, several ALL functions may be necessary in order to transport the data (see field length).

The source parameters which you must specify for the direct functions are irrelevant for ALL functions (any values can be entered here).

Note: if you have a long cycle time with your STEP 5 program it may be advisable to call up SEND ALL several times per cycle. SEND ALL calls only extend your cycle time when there is actually data to be transferred.

Assigning parameters to SEND ALL:

Below, there is a description of the parameters that you must specify when the HDB is called. The form and order of these parameters are the same as those you see when you program on the PG. The letters x and y are variables to be replaced by values when you program. For the FB number (FB 126) the number of the CPU 922 (R processor)/S5-135U is used in this example.

Further information can be found in the instruction manuals for the handling blocks (see /1/ in List of further relevant documentation).

SEND ALL: FB126

(With the S5-115U/150U the call of the SEND DIRECT n with job number 0 is used. The specifications for the source parameters are then unnecessary.)

```
SSNR:  KYx,y    ;interface number
                value range: x > 0: indirect para. assignment
                                    see /1/ in list of
                                    documentation
                             x = 0: y = interface number values
                                        from 0 to 255 (see in-
                                        structions CP 525)
A-NR:  KY0, x   ;number of the job
                value range: x = 0 for ALL function
ANZW:  xxy      ;condition codeword
                value range: see /1/ in list of documentation
                             xx = flag word (FW) or data word
                                  (DW) (caution: with DW, DB/
                                  DX must be called first!)
                             y = number (dependent on xx)
PAFE:  xxy      ;parameter assignment error
                value range: see /l/ in list of documentation
```

3.5 CONTROL

The CONTROL HDB copies the job status of a job into the specified condition codeword. The job status provides information about operating or processing statuses.

CONTROL can be called at any point in the program.

Assigning parameters to CONTROL:

The following table contains a description of the parameters that you must specify when the HDB is called. The form and order of these parameters is the same that you see when you program on the PG. The letters x and y are variables that must be replaced by values when you program. For the FB number (FB123) the number of the CPU 922 (R processor)/S5-135U has been used in this example.

Further information can be found intheinstruction manuals for the handling blocks (see /1/ in List of further relevant documentation).

```
SSNR:  KYx,y    ;interface number
                value range: x > 0: indirect para. assignment
                                    see /1/ in list of
                                    documentation
                             x = 0: y = interface number values
                                    from 0 to 255 (see in-
                                    structions CP 525)
A-NR:  KY0,x    ;number of the job to be controlled
                value range: x=1to223
                             x=0   numberofthejob currently
                                   being processed is copied
                                   into the ANZW

ANZW:  xxy      ;condition codeword
                value range: see /1/ in list of documentation
                           %x= flag word (FW) or data word
                               (DW) (caution: with DW, DB/
                               DX must be called first)
                            y =number (dependent on xx)
                The condition code of the job to be
                controlled must be specified.
```

3.6 RECEIVE DIRECT n and RESET DIRECT 200

These are only used with special jobs. The following special functions are carried out with RECEIVE DIRECT n:

- reading the error message area in SYSTAT
- reading the whole status area SYSTAT
- reading the identification area SYSTAT
- reading the date and time

The special job RESET DIRECT n allows you to:

- reset the error entries in SYSTAT

The description of the special jobs can be found in the instructions "Event output and listing with the PT88/PT89 printer" in this manual.

Assigning parameters to RECEIVE DIRECT n and RESET DIRECT n:

The following table contains a description of the parameters that you must specify when the HDB is called. The form and order of these parameters are the same that you see when you program on the PG. The letters x and y are variables that must be replaced by values when you program. For the FB numbers the numbers of the CPU 922 (R processor)/S5-135U have been used in this example.

Further information can be found in the instruction manuals for the handling blocks (see /1/ in List of further relevant documentation).

RECEIVE DIRECT n: FB121 - only for special jobs!

SSNR: KYx,y ;interface number
value range: x > 0: indirect para. assignment
see /1/ in list of
documentation
x = 0: y = interface number values
from 0 to 255 (see in-
structions CP 525)

A-NR: KY0, x ;number of the direct job
value range: x = 200, 218, 221, 223
(callable special jobs)

ANZW: xxy ;condition codeword (a doubleword is occupied)
value range see /1/ in list of documentation
xx = flag word (FW) or data word
(DW) (caution: with DW, DB/
DX must be called first)
y = number (dependent on xx)

ZTYP: KSyy ;destination type
valuerangeyy=DB
XX indirect para. assignment
see /1/ in the list of
documentation
(special jobs)

DBNR: KY0,x ;dest. DB no, with ZTYP DB
value range: x = 3 to 255

ZANF: KF+x ;start address of destination
value range: x = 0 to length of dest. type minus
ZLAE

ZLAE: KF+x ;number of data to be transferred
value range: x = dependent on special job

PAFE: xxy ;parameter assignment error
value range: see /1/ in the list of docum.

RESET DIRECT 200: **FB124** - only for special jobs

```
SSNR:  KYx,y   ;interface number
               value range: x > 0: indirect para. assignment
                                   see /1/ in list of
                                   documentation
                            x = 0: y = interface number values
                                       from 0 to 255 (see in-
                                       structions CP 525)

A-NR:  KY0,x   ;number of the direct job
               value range: x = 200 (special job)

PAFE:  xxy     ;parameter assignment error
               value range: see /1/ in the list of docum.
```

3.7 Examples - HDBs with Parameters Assigned

You start a job by calling the HDB SEND DIRECT n on the CPU (in your STEP 5 user program). The parameters to be assigned to the HDB SEND DIRECT n depend on the type of job to be started.

Jobs without data transfer

This group includes the following PC jobs

- new page
- page no. = 1

The HDB SEND DIRECT n must have parameters assigned as follows:

```
                                    GENERATE RLO=1 !!
     :
     :JU  FB120
NAME :SEND
SSNR :    KY0,0                     CP 525 INTERFACE 1
A-NR :    KY0,10                    JOB NUMBER 10
ANZW  :  FW6
QTYP :   KSNN                       NO DATA TRANSFER
DBNR :    KY0,0                     IRRELEVANT SINCE NO DATA
QANF:     KF+0                      IRRELEVANT SINCE NO DATA
QLAE :    KF+0                      IRRELEVANT SINCE NO DATA
PAFE :    FY199                     PARAM. ASS. ERROR IN FY199
     .
     .
```

Under the job number above (here: 10) you must program the required job belonging to this group in your CP 525 user program and transfer it to the corresponding interface.

Jobs for which data are fetched from the CPU

This group includes:

- process status list (XL)
- chained list (CL)
- update group inhibit bits (update GIB)
- update sequential message list (update SML)

To execute these jobs the CP 525 requires data from the CPU. Where the data is to be fetched from is decided when you program the job with COM 525.

The HDB SEND DIRECT n must have parameters assigned as follows:

```
                                    GENERATE RLO = 1 !!

     :JU  FB120
NAME :SEND
SSNR :     KY0,0                    CP 525 INTERFACE 1
A-NR :     KY0,10                   JOB NUMBER 10
ANZW  :  FW6
QTYP   :   KSNN                     INFORMATION COMES FROM CP 525
DBNR :     KY0,0                    IRRELEVANT
QANF  :  KF+0                       IRRELEVANT
QLAE  :  KF+0                       IRRELEVANT
PAFE :     FY199                    PARAM. Ass. ERROR IN FY199
```

Under the job number above you must program **the** required job belonging **to this group in your** CP 525 user **program and transfer** it to the corresponding interface.

Jobs for which data are sent by the CPU

These PC jobs include the following:

- sequential message list (sML)
- current message list (CML)

The CP 525 requires data from the CPU to execute these jobs. Which data the CPU is to send is decided when you assign parameters to the SEND DIRECT n.

The HDB SEND DIRECT n must have parameters assigned as follows:

```
     .                                  GENERATE RLO = 1. ! !
     :
     :JU  FB120
NAME :SEND
SSNR :    KY0,0                        CP 525 INTERFACE 1
A-NR :    KY0,10                       JOB NUMBER 10
ANZW : FW6
QTYP :    KSDB                         SOURCE = DATA BLOCK
DBNR :    KY0 ,11                      Data block number: 11
QANF :    KF+0                         START ADDRESS ALWAYS 0 ! !
QLAE :    KF+200                       TRANSFER 200 DATA WORDS
PAFE :    FY199                        PARAM. ASS. ERROR IN FY199
```

Under the job number above you must program the required job belonging to this group in your CP 525 user program and transfer it to the corresponding interface.

3.8 Evaluation

The following sections contain information about how to evaluate the parameter assignment error byte (PAFE byte) and the condition codeword ANZW.

3.8.1 The Parameter Assignment Error Byte (PAFE Byte)

When working with the handling blocks SYNCHRON, SEND DIRECT n, SEND ALL, RECEIVE DIRECT n and RESET DIRECT 200, you must specify a parameter assignment error byte (PAFE-BYTE) .

If the HDB recognizes an error, it writes the corresponding error number into the PAFE byte. If the block runs through without errors the HDB writes OOH into the PAFE byte.

In addition to the errors that can be attributed directly to the parameter assignment, the PAFE byte also indicates errors which occur during the CPU/CP 525 communicat ion.

If such errors occur you should immediately remedy the situation to avoid other messages appearing during operation.

In certain exceptional cases, errors are indicated in the PAFE byte during operation. These errors are caused by the software of the CP 525 in special situations as follows:

if more than 10 DIRECT jobs are processed in the cycle, i.e. ten jobs are already signalled as “running” and you want to start an eleventh job

if the whole or part of the CP 525 user program is transferred to the CP 525 with COM 525 during normal operation

i f a SEND job is started before the SYNCHRON job runs through once without errors.

During multiprocessor operation you can also receive a PAFE message if several CPUS access the same CP 525 interface at the same time. The access is prevented (with a PAFE message) until the current CPU/CP communication is completed. This PAFE message does not indicate an error but shows that access is momentarily blocked.

The following list provides a brief explanation of the PAFE *error* messages. Note that the significance of each error number depends on the PC type. You should check the instructions for the handling blocks for your PC (see /1/ in the list of further relevant documentation).

PAFE no.	Significance
11H	Wrong ORG format: source/destination parameter type wrong, area (start address, length) not permissible
21H	DB/DX or memory area not present
31H	Source/destination too small
41H	Source/destination area not present (not plugged in) Timeout (QVZ) from this area
51H	Error in condition codeword
61H	No source parameters with SEND ALL
71H	Interface does not exist (QVZ for the CP 525) Check the jumper settings for the interface number on the CP 525
81H	Interface not ready or not synchronized. Check that the SYNCHRON HDB is programmed in the start-up OBs andnms without errors.
91H	Interface overload; part of a program is being transferred from the PC to the CP 525.

A1H CP interface being used by a different CPU in multiprocessor operation (S5-135U and S5-155U) when several CPUs access one CP interface.

B1H Job number too high

C1H Error in handshake with the CP 525:

- CP does not reply within the monitoring time or
- CP rejects handshake, e.g. because more than 10 jobs were triggered simultaneously (when 10 jobs are running bit 1 is set in the ANZW)

D1H Field Length (SYNCHRON) wrong and group error for other handshake errors

E1H DB1 call missing with indirect parameter assignment and group error for HDB software error

F1H Double block call (when a block can be interrupted at command boundaries)

3.8.2 The Condition Codeword (ANZW)

The condition codeword with direct jobs:

The first part of the condition codeword contains information about the status of the job processing - the secondpart contains information about the length ofthedata transferred.

If you are working with the direct jobs SEND DIRECT n, FETCH DIRECT n, RECEIVE DIRECT n the condition codeword requires a doubleword.

Significance of the bits in the condition codeword of a DIRECT job:

Bit 0 : RECEIVE job ready (handshake acceptable)
0 = RECEIVE disabled
1 = RECEIVE enabled

Bit 1 : **job running**
0 = SEND/FETCH enabled
1 = SEND/FETCH disabled

Bit 2 : job ***complete*** **without error**
Bit 3 : job complete with error

Bit 4 : data transfer/data reception running

Bit 5 : data transfer complete
Bit 6 : data reception complete

Bit 7 : 0 = data transfer/reception enabled
1 = data transfer/reception disabled

Bit 8-11 : error messages
see the error tables in the user's guide "Event output and listing with the PT88/PT89 printer" in this manual

Bit 12-15 : not used

In the second half of the condition code doubleword the SEND ALL enters the length of the transferred data. This is the number of pieces of data currently being transmitted.

Every direct job should have its own condition code doubleword reserved. The following areas can be used:

- the flag area FW 0 to 252
or
- the data words 0 to 254 in the currently open data block **(DB).**

To ensure **reliable data exchange it is sufficient when the STEP 5 program evaluates bits 1, 2 and 3 of the ANZW as follows:**

- bit 1 indicates that the job was accepted for processing by the CP. The **job can be triggered again only after bit 1 is reset**

- bits 2 and 3 indicate that the CP completed processing the job.

On completion of the job without errors (bit 2 = 1), new information data can be prepared for the next transfer.

If the job is terminated with an error (bit 3 = 1), the error number is entered in bits 8 to 11 of the ANZW. The list of error numbers can be found in the user's guide "Event output and listing with the PT88/PT89 printer" in this manual.

The condition codeword with the ALL job:

With the SEND ALL, the condition codeword requires only one data word or flag word.

While the ALL function is running, the number of the DIRECT job for which the data is being transferred is indicated in the ANZW. If the function is run through without any data transfer, the value 0 is entered.

In addition, the ALL function updates the condition codeword of the corresponding direct job.

SIEMENS

We have checked the contents of this manual for agreement with the hardware and software described. Since deviations cannot be precluded entirely, we cannot guarantee full agreement. However. the data in this manual are reviewed regularly and any necessary corradons included in subsequent editions. Suggestions for improvement are welcomed
Technical data subject to change.

Siemens Aktiengesellschaft

C79000-B8576-C547 .06
Gerätewerk Karlsruhe
Printed in the Federal Republic of Germany