SIEMENS

EB11 – STANDEXEMPLAR

SIMATIC S5

Manual
E MA-W

Contents

E89120-F1944-U11-C Operation Manual Bubble Memory Module 513-3M...	1
E89120-F1944-U61-C Operation Manual Standard Function Blocks for the AG 135 U/S Processor	2
E89120-F1944-U81-D Operation Manual Standard Function Blocks for the AG 135 U/R Processor	3
E89120-F1944-U71-B Operation Manual Standard Function Blocks for AG 150 U	4
E89120-F1944-U91 Operation Manual Standard Function Modules for the AG 115 U	5
E89100-A503-U11-B Operation Manual Process Control Keyboard PBT 982 PBT 982/kyrillic	6
E89100-A611-U1 Instruction Manual Process Communication Keyboard PBT 982/2 PBT 982/3	7

Issue: 01

SIEMENS AKTIENGESELLSCHAFT

S I E M E N S

SIMATIC S5

Bubble Memory Module 513-M 6ES5513-3MA11
6ES5513-3MB11

Operation Manual Order number: E89120-F1944-U11-C

Contents

1 T e c h n i c a l D e s c r i p t i o n
1.1 Field of Application
1.2 Accessories
1.3 Design
1.4 Operation Mode
1.4.1 Addressing Method
1.4.2 Internal Memory Management
1.4.3 Interface to Simatic User Program
1.4.4 Interface to Programmer
1.5 Technical Data

2 M o u n t i n g a n d O p e r a t i o n
2.1 Removal and Insertion of the Module
2.2 Slots in the Programmer
2.3 Display Elements
2.4 Jumper Assignment
2.5 Connection to the Programmer

3 C o m m i s s i o n i n g
3.1 Setting the Interface Number
3.2 Synchronization
3.3 Loading

4 Operating with Programmer

4.1 General

4.2 Preparation Settings

4.3 Possible Functions

4.4 Examples of Operations

5 Maintenance

5.1 Connector Pin Assignments

5.2 Signal Names and Abbreviations

1 Technical Description

1.1 Field of Application

In control engineering, there is an increasing number of data that are not required permanently in the main memory of the controller. These data may be either process data, images, etc., and also user programs.

To record these data, it is necessary to have a mass storage device with following characteristics:

- pluggable into programmable controller
- No information loss on power or buffer voltage failure
- Robustness and maintenance-free operation
- Bidirectional data transfer between programmable controller and mass storage.

For this purpose, bubble memory module 513 has been designed for application in the S5 115 U, S5 135 U and S5 150 U programmable controllers. Due to the addressing method (page frame principle), it is possible to install several bubble memory devices in one programmable controller.

1.2 Accessories

For operation of the bubble memory in the various programmed controllers there are standard function blocks that have to be ordered separately, as well as the according data handling blocks.

AG	Function block BUBBLE	Data handling block
135U/ S processor	6ES5842-0GA10	6ES5842-0CA10
135U/ R processor	6ES5852-0GB10	contained in operating system
150U	6ES5844-0GA10	6ES5844-0CA10

Observe that only data handling blocks of version A2 or higher may be used.

1.3 Design

The bubble memory module contains following function units (see Figure 1.3):

- The bubble memory system with a capacity of maximum 256 k*bytes (= 128 K*words).

- One dual-port RAM as for the termediate buffer to transfer data between bubble memory and the central module.

- One 8085 microprocessor for the handling of data transfer between dual-port RAM and bubble memory as well as between programmer unit and bubble memory.

- One multi-function block 8256 for interface to the programmer.

The Simatic S5 bus is connected via two base connectors. Two LEDs are provided on the front panel to indicate operation modes.

Figure 1.3 Functional block diagram

1.4 Operation Mode

1.4.1 Addressing Method

Data transfer between main memory and bubble memory is carried out via the dual-port RAM located on the bubble memory module.
The dual-port RAM has a capacity of 1K (=1024) bytes and occupies the address range from F400H to F7FFH in the programmable controller.
The addressing mode with page frame addressing enables plugging several modules within the same address range.
With page frame addressing the module selection is effected by writing the ident number into vector register address FEFFH.
The ident number is compared with the respective jumper configuration on all page frame modules. If the ident number in the vector register matches the jumper configuration on the module, this module is thus selected. It remains selected until a new ident number is loaded into the vector register (see Figure 1.4.1).
This addressing method is used in SIMATIC S5 also by communication processors (CP) and intelligent peripheral modules (IP).
So the bubble memory can be mounted together with these modules without any problems.

Figure 1.4.1

1.4.2 Internal Memory Organization

The bubble memory module has a capacity of 2048 resp. 4096 blocks of 64 bytes each. Of that, 256 blocks are reserved for internal memory management. The memory management contains, among other a directory for a maximum of 984 blocks. However, only the first 579 sectors are output.
Memory compression is not necessary since memory organization recognizes that a location is free as soon as the respective block is cleared.
If a block already existing is replaced by a longer one, the old memory location is overwritten and the remainder added to the next free memory location. This ensures an optimum memory allocation.

1.4.3 Interface to Simatic User Program

For the Simatic user wishing to store data on the bubble memory, the structure of the memory area is as follows:

	logical sector name	
DB1 DB2 • DB254	"ANTON1" sector	(1st sector)
DB5 DB7	"ANTON2" sector	
DB5	"ANTON3" sector	
DB1 DB2 DB3	•	
• •	•	
DB 1 DB25 •	•	
	"CAESAR5" sector	(255th sector)

The memory area is divided into sectors so that as many data blocks etc. as possible can be stored on the bubble memory. The length of the single sectors is variable.
Into each sector a maximum of 254 blocks can be stored (for 135U: data blocks only). Each block may have a length of up to 2015 words.

So if a block is to be stored on the bubble memory, it must be assigned a sector name.

By combining the block number with a sector name, it is possible to store on the bubble memory several blocks of same type (e.g. data blocks) and with same block number under different sector names.
So it is possible to address a maximum of 255 x 254 blocks of each type on the bubble memory module.

The sectors can also be used, for example, to combine data from different program parts.

Data transfer between Simatic user program (that is main memory) and bubble memory is performed by a function block.

The logical name of the sector to be referenced is parameterized in the function block, i.e. ANTON1.

1.4.4 Interface to Programmer

The bubble memory is provided with a serial interface for the programmer. This enables direct transmission of blocks from user diskette to bubble memory.
The assignment of the block to the sectors is performed by first transmitting the name of the desired sector to the bubble before transmitting the blocks designed for one specific sector.
Data block 255 is fixedly reserved for this purpose.

Data block 255 contains the name of the sector in following format:

```
0: KC=A:XXXXXX
4:      Name of the sector (must be 6 ASCII characters)
```

So the use of data block number 255 on the bubble memory is prohibited.

Each time another sector has to be selected on the programmer, DB255 must first be transmitted to the bubble memory with another sector name.

1.5 Technical Data

General data

Order numbers:	
64 K*words	6ES5513-3MA11
128 K*words	6ES5513-3MB11
(order numbers of function blocks see 5.3)	
Dimensions	233mm x 160mm
Weight	approx 550 g
Format	Double-height Eurocard
Standard Plug-in Stations	1 1/3 SPS
Supply voltage	+ 5V ± 5 %
Current consumption	+ 5V / max. 2.3 A

Connectors

Base connectors X1, X2	48-pin male multipoint connector row 2
Front connector X3	15-pin cannon connector (sockets)

Dual-port RAM

Capacity	max. 1 K*byte RAM

Progammer interface

Type of interface	TTY (20mA), passive
Baud rate	9600 baud
Character frame	8 bits, 1 parity bit (even), 2 stop bits

Microcomputer

Microprocessor	8085 AH-2
Program memory	16 K*bytes of EPROM
Data memory	8 K*bytes of RAM

Bubble memory system

Bubble memory module(s)	7110AZ-1 (Intel)
Capacity	128/256 K*bytes
Block length *)	64 bytes
Number of blocks *)	2048/4096
Error rate	$1x10^{-14}$ errors per bit
Mean access time *) per block	50 ms including processor processing time
Minimum access time *) per block	10 ms including processor processing time
Maximum access time *) per block	90 ms including processor processing time
Mean data throughput:	8.5 K*bytes/s

*) These data apply to the i n t e r n a l data transfer of the bubble memory, and not to the data transfer bubble memory - main memory - function block.

Environmental conditions

Temperature	
Operating	0°C to +55°C
Storage	-40°C to +70°C
Gradient	≤ 1.1°C/min
Humidity class	F according to DIN 40040
Air pressure	≥ 700mbar = 3000m Height above sea level ≥ 260mbar = 10000m Height above sea level
Permissible magnetic fields	≥ 16A/cm

2 Mounting and Operation

2.1 Removal and Insertion of the module

The module may only be removed or inserted when power is off.

2.2 Slots in the Programmer

The slots for the bubble memory module depend on the programmable controller used.
The bubble memory module can be plugged into the same slots as the communication processors (see figure 2.2).

2.3 Display Elements

Two LEDs located on the front panel indicate certain bubble memory states and errors.

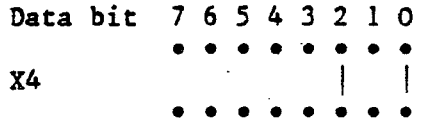
LED No.	LED lights when
LED1 (green)	Access to bubble memory
LED2 (red)	Failure (Data error in bubble memory, microprocessor failure)

2.4 Jumper Assignment

- X4 Jumper connector

  The ident number (= interface number) of the page frame module is set at connector X4.

  Example: Ident number 5



- X5 Jumper connector

  EPROM selection:

  Jumper inserted

  X5/1 - 2 EPROM 27128 (standard setting)

  X5/2 - 3 EPROM 27256

- X6 Jumper connector

  Test jumper, must always be inserted.

  For jumper layout see figure 2.4

2.5 Connection to Programmer

Cable 6ES5731-1...0 has to be used to establish the connection to the programmer.

Figure 2.4 Jumper layout

3 Commissioning

3.1 Setting the interface number

Before commissioning of the module, it is necessary to check the jumper setting for the interface number (see 2.4). If in the programmable controller communication processors (CP) or intelligent peripheral modules (IP) are connected, care must be taken that the interface number is specified only once.

3.2 Synchronization

If the bubble memory is mounted in the programmable controller and the voltage switched on, first the SYNCHRON block must be run in the programmable controller for the interface number selected (see Operation Instructions: Standard function blocks). With that, the bubble memory performs its start-up.

During start-up, the LEDs on the front panel light shortly: first alternatingly, then together, and then again the green one lights (flickering now).

If both LEDs are dark again, the bubble memory is ready for operation and can now also be operated with the programmer.

3.3 Loading

Loading the bubble memory can be carried out directly via the programmer. The operation is described in chapter 4.

When connecting the programmer to the bubble memory, care must be taken that range B resp. AG 150 S is the defaulted selection.

4 Operating with Programmer

4.1 General Considerations

As already described in Sections 1.3.3 and 1.3.4, the blocks can be stored on the bubble memory in different sectors.

When the bubble memory is operated with the programmer (e.g. for transfer, information), it must be possible to switch over between the individual sectors.

Data block 255 is reserved for this purpose. The name of the sector must be specified in data block 255 in following format:

```
DB 255
0: KC = A : XXXXXX        (8 bytes = 4 data words)
4:          |
            |  sector name
            |  Mandatory length: 6 printable ASCII
            |  characters
            |  e.g.: ANTON1
            |
            |-- This ASCII character is used to dis-
               criminate on the diskette between
               several bubble memories.
               For instance: all data for bubble
                             memory No.1 are
                             under A.
```

After DB255 is transferred to the bubble memory, all blocks transferred subsequently are stored under the sector name indicated in DB255. All operations (Input/output, clearing, directory) refer to this sector name.
This sector name is valid until anew a DB255 with another sector name is transferred to the bubble memory.

The bubble memory is addressed by the programmer as "AG".

Attention: With the use of the PG 685, problems result with following functions:
Output (manager): not possible, error number 3
Output DB 255 (sector directory): not possible, error number 3
Output DB 255 (sector selection): possible, but error number 4 call of non-existing blocks: error number 3

4.2 Preparation Settings

Set programmer to ON-LINE
Operating system PG 675
S79200-A003-A254-01:
Set AG150S
Operating system PG 675 U-S0-A02 P1-545:
Set range B

4.3 Possible Functions

- Input to AG (= bubble memory)

|F1| Block, block number

Attention: DB255 is reserved for sector name.

- Output from AG (= bubble memory)

|F2| Block, block number

Particularity: Output DB 255

The function "Output DB255" serves for giving the user an overview of all sector names existing on the bubble memory:
Presetting: Data format KC
Output DB 255: the sector names are output to the programmer as ASCII characters. The first name each is the current one set by the programmer.

- Block comparison

|F3| between

FDx AG
PG AG
AG FDx
AG PG

- Transfer

|F4|

from	to
PG	AG
FDx	AG
AG	PG
AG	FDx

The block numbers of source and destination blocks may be different.

- Clearing

|F5| Device: AG block: B
DB xxx
•
•
•

When using Clear B, all blocks stored under the currently valid name will be cleared including DB 255 which contains this name. Afterwards the name is no longer valid.

- Information functions

|F7|

- Directory

|F7| |F2| Device: AG Block: B
DB
•
•
•

The blocks stored under the current name are output. The number of the first block section on the AS 513 is output (in hexadecimal) under "starting address AG".

Directory of names: see output function.

- Memory configuration (only possible with Operating System S79200-A0003-A254-01)

F7	F4	
		1) AG RAM extension/size up to: 1000H (800H)
		2) AG RAM reserved up to: XXX

For 1:

Here the last block number in the AS 513 is given.

1000H for 256 K*bytes
800H for 128 K*bytes

For 2:

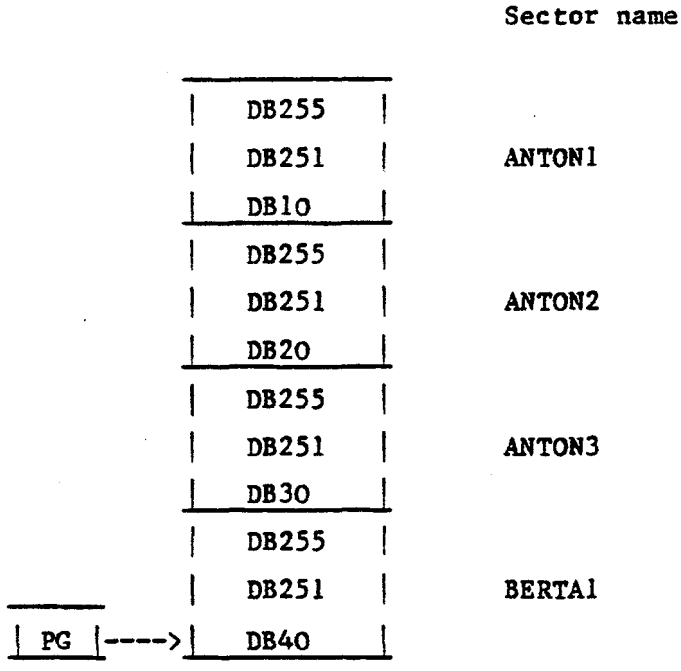
Here the number of blocks occupied (1 block = 64 bytes) is given. 256 blocks (= 100H) are reserved fixedly for the directory of the bubble memory.

Each block requires at least 1 block as "pointer block" i.e. cross-reference block, depending on the length of the block.

4.4 Examples of Operations

Application:

BUBBLE MEMORY



Data blocks DB 251 and DB 40 shall be entered into the BERTA1 sector of the bubble memory via the programmer.

Data block DB255 in the bubble memory is used as "Switch-over block". The sector name must be entered into DB255. Then it is possible to store various data blocks in this sector.

Example 1:
"BERTA1" sector shall be prepared

1. Connect cable to bubble memory

2. |F1| (Input) |AG| Block: |DB255|

=> DB255:
0: |AG| = |A:BERTA1| | |
4: Sector name = 6 ASCII characters

Now the "BERTA1" sector is selected

3. Input of DB251 and DB40 as usual
with input | AG | | DB251 |
0: 255 < KH = 0000,0000,0000
1020: | |

If the bubble memory is accessed or a transfer occurs, the green LED flickers.

Before beginning to enter DBs, synchronization must be performed (OB 20).

Example 2: Information about the "ANTON 2" sector

1. Select corresponding sector

   -> proceed as described in example 1:

```
|F1| |AG| |DB255| -> 0:|KC| = |A:ANTON2| |  |
```

Subsequently the directory can be output with output Buch.

5 Maintenance

5.1 Connector Pin Assignment

- Base connector X1

	d	b	z
2		Ground	+5V
4	UBAT		
6	ADB12	ADB0	CPKL
8	ADB13	ADB1	/MEMR
10	ADB14	ADB2	MEMW
12	ADB15	ADB3	/RDY
14		ADB4	DB0
16		ADB5	DB1
18		ADB6	DB2
20		ADB7	DB3
22		ADB8	DB4
24		ADB9	D55
26		ADB10	DB6
28	/DSI	ADB11	DB7
30		BASP	
32		Ground	

- Base connector X2

	d	b	z
2		Ground	+5V
4			
6			
8			
10			
12			
14			/NAU
16			/BAU
18			
20			
22	TxD *	/STOPPA	
24			GEP
26		RxD *	
28			
30			GROUND 24V
32		Ground	+24V

* Programmer interface

- Front connector X3

		1	
TTY receiver +	9		
		2	TTY receiver-
24V ground	10		
		3	
	11		
		4	+24V from bus
	12		
		5	
	13		
		6	TTY transmitter+
	14		
		7	TTY transmitter-
	15		
		8	

5.2 Signal Names and Abbreviations

Signal names without / : H = active
Signal names with / : L = active

ADB0 to ADB15	Address bus bits 0 to 15
BASP	Instruction output inhibit
/BAU	Battery failure
CPKL	Clear CPU Signal generated by the power supply to reset the central modules and the peripheral units.

DB0 to DB7	Data bus bits 0 to 7
/DSI	Data save
GEP	Buffered
/MEMR	Memory read
/MEMW	Memory write
/NAU	Power failure
PESP	Peripheral coding Is decoded from address bits ADB12 to ADB15 on central module
/RDY	Ready acknowledge signal
RxD	Receiver programmer interface
TxD	Transmitter programmer interface
/STOPPA	Stop request
UBAT	Battery voltage

SIMATIC S5

Bubble Memory Module 513-M
Standard Function Blocks
for the AG135U/S processor 6ES5842-0GA10

Operation Manual Order number: E89120-F1944-U61-C

Contents

1 General Considerations
1.1 Field of Application
1.2 Program Structure
1.2.1 Synchronization
1.2.2 Data Transfer
1.3 Data Structure
1.4 Parameter Handling for the FB-BUB
1.5 Handling of Several Bubble Memories Plugged into the PC
1.6 Special Features

2 Function Blocks
2.1 FB 199 FB-BUB 135S
Data Handling between PC and Bubble
2.2 FB 125 SYNCHRON
Preset SSDB
2.3 FB126 ACTIVE
Queue Processing

3 Application Example

1 General Considerations

1.1 Application

The 513-M bubble memory in the SIMATIC S5-135U Programmable Controller (PC) provides the facility for storing data and programs. Data transfer between the CPU and the bubble memory is accomplished by the BUB function block.

The BUB function block uses the SEND, RECEIVE, TECH, SYNCHRON, ACTIVE and ACTIVE-UP handling blocks. These blocks must be ordered separately under MLFB No 6ES5842-0CA10.

In the version for the S processor, only data blocks (DB1 to 254) with a maximum length of 2015 words can be transferred (completely or partially).

The data block number 255 must not be used in the CPU and in the bubble memory to store useful data since the DB 255 is used to transfer the sector name from the PC to the bubble memory and as a buffer for the FB-BUB.

1.2 Program Structure

1.2.1 Synchronization

First, the SYNCHRON function block must be called when the programmable controller is started (first start, warm start). The FB SYNCHRON controls the presetting of a data block (SSDB) and maintains a queue in which it is entered as first request. The FB-BUB function block contains, as a parameter, the data block which is assigned to this interface. The data block (SSDB) controls the request for data handling.
(See the description of the handling blocks for the AG 135U, S processor, order number C790000-B8500-C325-03).

1.2.2 Data Transfer

Data transfer is accomplished by the BUB function block in conjunction with the handling blocks.

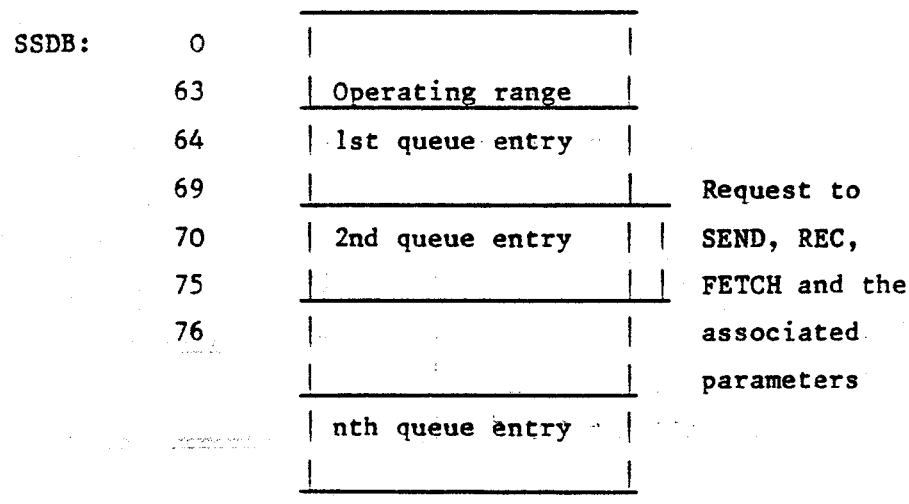
The requests are processed by the ACTIVE function block. This block must be called once per cycle and per interface.

The transfer of 1k words of data lasts approx. 6 to 10 seconds.

1.3 Data Structure

When handling blocks are used, one data block must be determined for each interface (interface data block SSDB: queue block). This block need not be pre-assigned and called by the user.

The data block consists of a 64-word operating range and a queue area of variable length. Each queue entry occupies 6 words, the minimum block length is then 70 words. The capacity of the queue determines the length of the data block. The maximum DB length is 256 words.



Moreover, the following data blocks are required: According to the type of request, the data block containing the flag word (refer to the Section dealing with the flag words); the data block containing the parameters in the case of indirect parameter assignment (see source/destination parameters) and the data block which provides the data to be transmitted or received. All these data blocks are called by FB-BUB and by the handling blocks.

The DB255 block is used as a data buffer for the FB-BUB. This block must not be used by the user.

DB 255	0	Operating range
	19	
	20	Data buffer
	149	

1.4 Parameters of the FB-BUB

The FB bubble is used to perform data handling between the PC and the bubble memory. The SEND, FETCH and RECEIVE data handling blocks are subordinated to it.

The ACTIVE FB which maintains the queue runs at the same time as the FB-BUB. Both are called in the OB12 block for time reasons. The FB parameters are based on the handling blocks.

The following parameter designations are used:

ANST = Initiation bit
SSDB = Interface data block
ANZW = Condition code word
BETR = Operating mode

NAM1 = Name of data set
NAM2 = (also sector) on the
NAM3 = bubble memory
e.g. AN
TO = ANTON1
N1
TYP = Block type (only DB)
QU-B = Source data block
QUDW = Source data word
ZI-B = Destination data block
ZIDW = Destination data word
ZAHL = Number of words to be transferred

The FB includes a direct and indirect parameter assignment. Only the initiation bit (ANST) has to be entered directly.

In the case of the direct parameter assignment, the function block processes directly the parameters specified in the block call. In the case of the indirect parameter assignment, a pointer to a parameter field is transferred to the function block with the block parameter.
The high byte (left byte) of the SSDB parameter is used as a switching criterion for direct/indirect parameter assignment.

High byte of SSDB = 0 direct parameter assignment:
SSDB, ANZW, BETR and NAM1, NAM2, NAM3 are parameterized directly

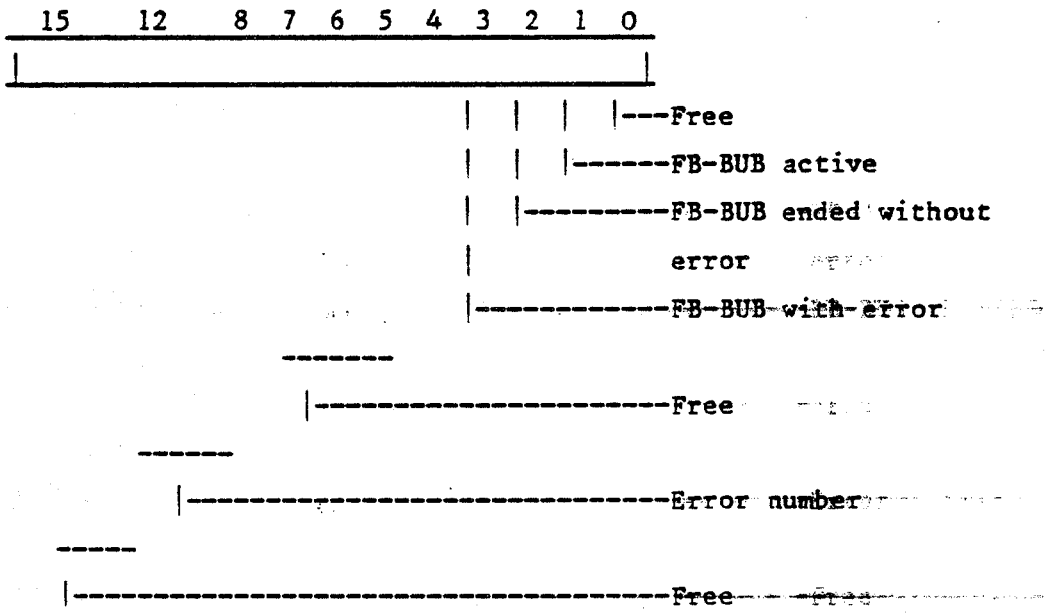
High byte of SSNR = 0 indirect parameter assignment:
SSDB, ANZW, BETR and NAM1, NAM2, NAM3 are stored in the data block at the data word indicated in the low byte of SSDB.

Indirect parameter assignment of the source and destination data is achieved with parameter TYP = XX.QU-B gives the data block number and QUDW the word number at which the TYP, QU-B, QUDW, ZI-B, ZIDW and ZAHL parameters are stored.

a) The initiation bit (ANST) is set by the user if data is to be transferred from the PC to the bubble memory.
FB-BUB resets the initiation bit after the request has been executed.

b) The interface data block (SSDB) corresponds to the data block, the interface number of which has been parameterized at the SYNCHRON block in OB20. The FB-BUB block does not check whether SSDB and interface number match (see SYNCHRON block).

c) The flag word (ANZW) contains all the information about the FN-BUB.



Error numbers:

0 = no error
1 = parameter error
2 = block not existing in PC resp. too short
3 = number > 2015 or = 0 resp.
PC DB longer than 2020 DW (incl. header)
4 = number <> PC DB length (only with operation mode HS, if DB not existing on bubble)
5 = Bubble DB too short
6 = error on hardware
7 = Bubble memory not formatted
8 = Bubble memory full
9 = operation mode MS -- HS:
DB not existing resp. data set name not existing on bubble
10 = free
11 = erroneus request
12 = handshake error

The flag word can be a flag word or a data word in a data block.
The FB-BUB expects that data block and data word exist.

d) The operating mode (BETR) indicates a data transfer from or to the bubble.

MS = Data transfer from the bubble to the PC
HS = Data transfer from the PC to the bubble

e) The data blocks are stored in the bubble under a data set name (also sector name; NAM1, NAM2, NAM3). Hence, the data blocks (3 to 254) can be stored under as many data block names as desired. The data set name is a 6-digit ASCII name. This name must exist on the bubble (see bubble programming unit).

f) The type (TYP) describes the block type. In this version, only "DB" is admitted. Indirect parameter assignment of data source and data destination can be indicated with "XX".

g) QU-B: Source data block number
DB3...254 are admitted

h) QUDW: Source data word number
DWO...2014 are admitted

i) ZI-B: Destination data block number
DB3...254 are admitted

j) ZIDW: Destination data word number
DWO...2014 are admitted

k) ZAHL: Number of data words to be transferred

1.5 Use of Several Bubble Memories in the PC

The SYNCHRON FB must be called in the OB20 block for each bubble memory. Each interface is assigned an interface data block. The user can call the FB-BUB block conditionally once for each interface or amend the parameters of the interface data block by using the indirect parameter assignment. As long as a bubble interface is processed, the FB-BUB must not be called for another interface.

Note: Due to the data handling blocks of the S processors, only one of the CPUs in a multiprocessor system can exchange data with one or several bubble memories.

Make sure that the FB-BUB is not active (bit 2 is set in the flag word when the FB has terminated) before switching to another interface.

Example:

Two bubble memories are connected to the programmable controller at interface 0 and 2

```
      OB 20
    : A   DB 255
    : L   KB 0          Internal data in the bubble buffer must
    : T   DW 0          be deleted on system startup or restart
    : T   DW 8
    : T   DD 9
    : SPA FB 125
NAME: SYNCHRON
SSDB:     DB 17
SSNR:     KY 0,0
ANZW:     KY 0,10       Flag word 10
PAFE:     M 1.0
    : SPA FB 125
NAME: SYNCHRON
SSDB:     DB 18
SSNR:     KY 0,2
ANZW:     KY 0,12       Flag word 12
PAFE:     M 1.2
    :SPA  FB210         User program with call of FB215 ACTIVE
          •             and error analysis (see Section 2.2,
          •             SYNCHRON handling block)
          •
          BE
```

OB 1

```
    : SPA FB 126
NAME: ACTIVE
SSDB:     DB 17
    : SPA FB 126
NAME: ACTIVE
SSDB:     DB 18
    : SPA FB WECHSEL
    : A   DB 100
    : SPA FB 130
NAME: BUB-135 U
ANST:     M 1.7
SSDB:     KY 255.1    DB 100: DW 1 : KY = 0.17 Interface 1
ANZW:     KY 0.0      DB 100: DW 2 : KY = 0.7
BETR:     KC..        DB 100: DW 3 : KC = HS
NAM1:     KC..        DB 100: DW 4 : KC = AU
NAM2:     KC..        DB 100: DW 5 : KC = GU
NAM3:     KC..        DB 100: DW 6 : KC = ST
TYP :     KC DB
QU-B:     KF 252
QUDW:     KF 0
ZI-B:     KF 252
ZIDW:     KF 0
ZAHL:     KF 20
          •
          •
          •
          BE
```

FB WECHSEL

```
: U  M 8.1        (FB-BUB still active at interface 1)
: BEB
: A  DB 100       SSDB for interface 2
: L  KB 18
: T  DW 1
     •
     •
     •
     BE
```

1.6 Special features

- In the MS operating mode (data transfer from the bubble to the PC grammable controller), the destination data block must be available in the PC.

- If the DB is not available in the bubble in the HS mode, the number of data words to be transferred must be equal to the DB length - 5 (header).

- The DB255 is used as a buffer and must not be used to store data in bubble memory.

- The FB-BUB expects that the queue is correct (see FB SYNCHRON block).

- The programming unit function "KOMP" and "DB LOAD" may lead to errors when FB-BUB and the handling block are run simultaneously. The programming unit must be brought into the stop status before.

2 Function Blocks

2.1 FB-BUB - 135 S (FB 199)

The FB-BUB is responsible for data transfer between the AG 135 U and the bubble memory 513-M.

The FB-BUB is called in the OB1 block and is activated by the initiation bit (indicated as a parameter). To perform data transfer, the FB-BUB will run several times. The flag word informs the user that the FB-BUB is active.

Explanation of the input and output parameters:

NAME	Designation	Kind, Type
ANST	Activation of a data transfer	E, BI
SSDB	Interfaces DB (queue) Direct parameter assignment Left byte = • Right byte = DB 3...254 Indirect parameter assignment of SSDB, ANZW, BETR, NAM1, NAM2, NAM3: Left byte = not equal to 0 Right byte = DW No. at which the parameters are stored. DW0 ...250	D, KY
ANZW	Address of the code condition word: Left byte = 0 Right byte = flag word or Left byte = DB No. Right byte = DW No.	

NAME	Designation	Kind, Type
BETR	Operating mode KC = HS: Transfer from the main memory of the PC to the bubble memory KC = MS: Transfer from the bubble memory to the main memory for the PC	D, KC
NAME1	Data set name, also sector name (6 characters = Name 1, 2, and 3) of the bubble memory to and from which the data is transferred (KC = RE)	D, KC
NAME2	(KC = ZE) = REZEPT	D, KC
NAME3	(KC = PT)	D, KC
TYP	Type of the block to be transferred; for the present, data block only (KC = DB) KC = XX for indirect parameter assignment of source and destination parameters	D, KC
QU-B	Source block (KF = 3 ..254)	D, KF
QUDW	Only relevant for the data blocks: First word in the source block at which data transfer starts (KF = 0 .. 2014)	D, KF

NAME	Designation	Kind, Type
ZI-B	Destination block (KF = 3 ...254)	D, KF
ZIDW	Only relevant for data blocks: First word in the destination block where the data transferred starts to be written to (KF = 0 ..2015)	D, KF
ZAHL	Only relevant for data blocks: Number of the words to be transferred (KF = 1...2015)	D, KF

Flags assigned: MW 200 to MW 220

Data blocks called: Interface-related data block SSDB
Data block for the code condition word
Source/destination data block DB 255 FB-BUB buffer

Function blocks called: FB 120 = SEND
FB 121 = RECEIVE
FB 122 = FETCH

System data assigned: BS 240

Nesting depth: zero

Program example of FB-BUB:

When the FB-BUB is called, the FB ACTIVE block must always be called (to process the queue).

If the user sets the initiation flag M1.7, the FB-BUB is run. The initiation flag must remain set for one cycle. It can then be reset.

The interface DB is DB 14, which corresponds to the interface 0. 10 data words of DB 252, starting at DWO, must be transferred from the programmable controller to the bubble memory in DB 251, starting at DW 30.

a) Direct parameter assignment

OB 1 Cycle

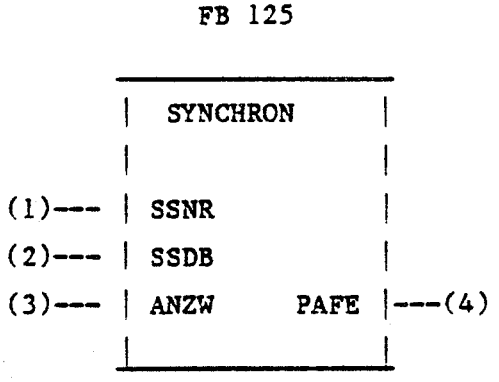
```
    : SPA FB 126
NAME: ACTIVE
SSDB:     DB 14
    : SPA FB 199
NAME: BUB-135 S
ANST:     M 1.7
SSDB:     KY 0.14
ANZW:     KY 0.8      Flag word 8
BETR:     KC HS       Transfer the data from the programmable
                      controller to the bubble
NAM1:     KC RE
NAM2:     KC ZE       =REZEPT
NAM3:     KC PT
TYP :     KC DB
QU-B:     KF 252
QUDW:     KF 0
ZI-B:     KF 251
ZIDW:     KF 30
ZAHL:     KF 10
```

b) Indirect parameter assignment

```
      :     A DB 100        The parameters SSDB, ANZW, BETR,
      :   SPA FB 199        NAM1, NAM2, and NAM3 are stored
NAME:   BUB-135 S           in DB100, starting at DW1
ANST:       M 1.7
SSDB:       KY 255.1
ANZW:       KY 0.0
BETR:       KC ..
NAM1:       KC ..
NAM2:       KC ..
NAM3:       KC ..
TYP :       KC XX           The parameters TYP, QU-B ,QUDW,
QU-B:       KF 100          ZI-B, ZIDW and ZAHL are stored in
QUDW:       KF 7            DB100, starting at DW1
ZI-B:       KF 0
ZIDW:       KF 0
ZAHL:       KF 0

DB 100

DW  1:      KY 0.14   SSDB
DW  2:      KY 0.8    ANZW
DW  3:      KC HS     BETR
DW  4:      KC RE     NAM1
DW  5:      KC RE     NAM2
DW  6:      KC PT     NAM3
DW  7:      KC DB     TYP
DW  8:      KF 252    QU-B
DW  9:      KF 0      QUDW
DW 10:      KF 251    ZI-B
DW 11:      KF 30     ZIDW
DW 12:      KF 10     ZAHL
```

2.2 FB 125 Data Handling Block, Presetting SYNCHRON

Block diagram



Functional description

The SYNCHRON block checks if the interface-related data block (SSDB) exists and if its length is within the prescribed limits. If so, the interface (SSNR) can be assigned to the data block (SSDB) by entering the interface number (SSNR) in the working area of the SSDB. The working area and the queue area are deleted or preset existing requests are also deleted. SYNCHRON is the new request to be entered in the queue. Bit 14 is set in the flag word (ANZW), all other bits are cleared.

However, if the SSDB does not exist or if its length is not admitted, the program leaves the block and the output bit of the PAFE parameter error is set. The flag word is not affected and the SYNCHRON request cannot be queued to the queue since no queue can be ceated. In this case, it is recommended, for safety reasons, to put the programmable controller to stop. Otherwise, the following bubble memory (SEND/ RECEIVE/FETCH/ ACTIVE) calls would lead to errors since they expect a correct queue.

The SYNCHRON function checks if any CP (bubble memory) is connected; if so, a check is made to know if there exists an in-

terface with number. If one of these checks is not successful, bit 12 is set in the ANZW flag word. The SYNCHRON function is used to initialize the interface. If an error is encountered, the SYNCHRON request is not deleted but remains at the first place in the queue. The requests that may follow can only be serviced if the interface is synchronized properly.

Flag word: (bit 12 to bit 15)

Bits 0 to 11 are irrelevant

	Bit 15	14
Do not enter handling function in the queue	0	0
Enter handling function in the queue	0	1
Handling function started	1	1
End of handling function	0	0

If these handling functions are terminated, the bit position 12 and 13 give the following information.

	Bit 13	12
Handling function was performed correctly	0	0
Handling function ended in error or was interrupted	0	1
Handling function could not be performed	1	0

The flag word bits 12 and 13 must not be evaluated as long as bit 15 is set.

Explanation of the inputs and outputs (parameters)

Parameter	Designation	Kind	Type
SSDB	Interface-related data block, it maintains one queue which contains the SYHCHRON request	B	
SSNR	Interface number, number of the logic interface, it can be set on the CP (bubble memory) by jumpers, only the low byte is evaluated, admitted: KY = 0 to 255, 0 to 255	D	KY
ANZW	Condition code word Flag word admitted: KY 0, 0 to 254	D	KY
PAFE	Parameter error, admitted: inputs, outputs flags	A	BI

Flags assigned: MW200, MW202, MW206

Data blocks called: Interface-related data block SSDB, data block for the flag word

Function blocks called: None

Nesting depth: Zero

Sample program

Synchronization of an interface

The SYNCHRON block must be the first handling block to be called in the start OB (startup or restart).

The interface number (parameters SSNR: page frame address, can be set on the CP module by means of jumpers) is 4; the data block assigned to this interface (parameter SSDB: queue block) is DB 14.

```
    :A   DB 255
    :L   KB 0
    :T   DW 0    Internal data must be deleted on first start
    :T   DW 8    or restart in the bubble buffer
    :T   DD 9
    :SPA FB125
NAME:SYNCHRON
SSDB:    DB 14
SSNR:    KY0,4
ANZW:    KY0,6    Flag word 6
PAFE:    M 1.0
    :        :
    :SPA FB210   User block
NAME:ANLAUFE:ANLAUF
    :BE


FB210    B210
ANLAUF   ANLAUF

    : UN M 1.0   Note 1, see below
    : SPB =MOO1
    : STP        STOP: Abort first start
MOO1: SPA FB126  Note 2, see below
NAME:ACTIVE
SSDB:    DB14
```

```
:U   M6.6      Condition code word bit 14
:SPB =M001
:U    M6.4     Errors?
:      •
:      •       Error processing
:      •
:BE
```

Note 1: If this parameter error bit (M1.0) is set, the first start is interrupted for safety reasons because the length of SSDB is not admitted or the SSDB does not exist.

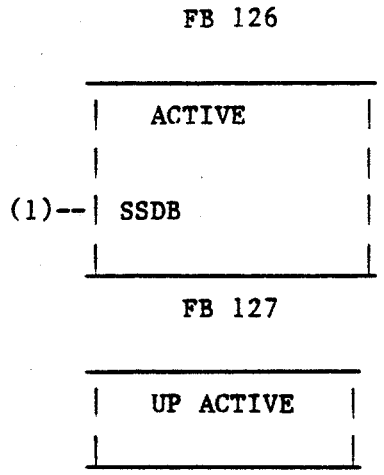
Note 2: This loop causes the SYNCHRON function to be processed immediately and completely. Finally, the ANZW bit 12 enables to determine whether this interface exists and is initialized properly. If not, the calls of the FB bubble (with this SSNR number) are irrelevant.

Complementary information

Startup SEND/RECEIVE requests are to be serviced when the programmable controller is started. The FBs can be called in the startup OB, but only after the SYNCHRON function has been called without error (PAFE bit M1.0 = 0).
If these requests still have to be serviced at startup time, for instance to provide received data, it is also possible to program a repeated call of FB ACTIVE; the calls are made conditionally depending on the ANZW bit 14 of the last request entered in the queue.

This construction of putting the FB in a loop is only admitted at startup time; in the cycle, the total runtimes might exceed the maximum permissible cycle time.

2.3 Handling blocks FB 126, Queue ACTIVE and FB 127



Functional description

The requests contained in the queue are serviced by the ACTIVE function block. The ACTIVE block calls a subroutine, the UP ACTIV subroutine. A watchdog or process alarm could be processed at such a block limit. To avoid a nesting of the ACTIVE block, the function block can be called either in the cycle (OB1 or FB0) or in the timed program.

Since the FB ACTIVE block calls the UP ACTIV subroutine, the interrupt-driven program parts called must be programmed in such a way that they do not destroy the flag area specified. I.e.: If the FB ACTIVE block is called in the cycle (OB1 or FB0) and the time-controlled or process interrupt-driven programs called also use the flags, starting at flag byte 200 (standard FBs, other handling blocks), the flags assigned several times must be saved at the beginning of the interrupt processing and written back at the end of the processing.

Data transfer between the programmable controller and the CP within one ACTIVE run is accomphised in blocks of up to 256 bytes. If 400 bytes (DW 0 to DB 199) have to be transferred, the first block will contain 256 bytes (DW 0 to DW 127), and the second one 144 bytes (DW 128 to DB 199).

Explanation of the inputs and outputs (parameters)

Parameter name	Designation	Kind	Type
SSDB	Interface-related data block, it contains the request queue which is processed by the ACTIVE function block	B	

Flags assigned: MB200 to MB255

System data assigned: BS254

Area assigned interface: byte 0 to byte 511

Data blocks called: Interface-related data block SSDB,

Data block for the flag word

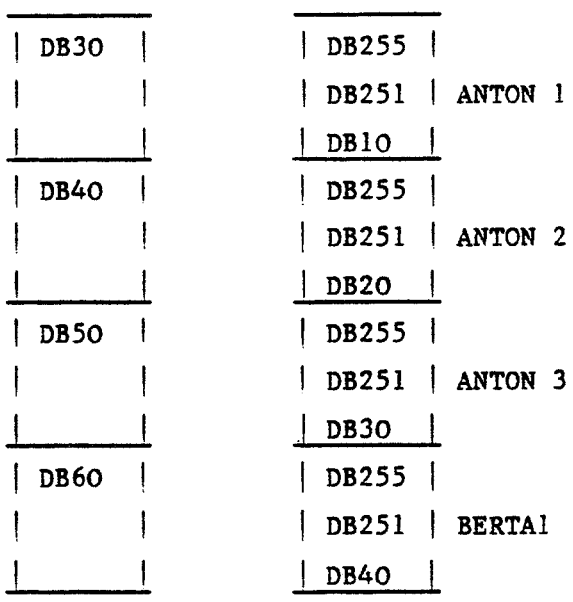
Data block, contains the source or destination parameter in the case of an indirect parameter assignment

Data block for transmit or receive data

Function blocks called: FB127 UP ACTIV

Nesting depth: One

3 Example of Application



Data direction: MS <---
HS --->

Task:

1000 words are to be transferred from DB30 in the programmable controller memory to DB 10 of sector "ANTON1" on the bubble memory.

Configuration for this example

Programmable controller AG 135 U with S processor
Jumpers on the bubble:
X4: No jumper inserted, i.e. Identno = 0
X5: Jumper 1-2 inserted, i.e. EPROM 27128
X6: Jumper inserted

Slot for the bubble memory
Like for CP EP 2..9

In the case of NAM1, NAM2, NAM3 in the FB199, the sector name is indicated in the bubble, e.g. ANTON1
Important: a max. of 6 characters
The transfer direction is indicated in the BETR = operating mode

HS = PC to bubble
MS = bubble to PC

OB1

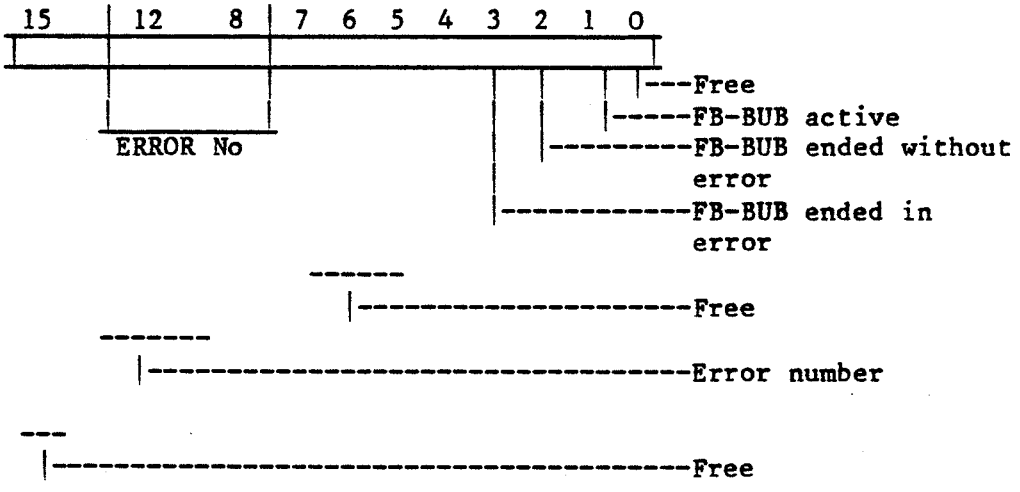
```
0000      :
0001      : SPA FB126
0002 NAME: ACTIVE
0003 SSDB:   DB35          SSDB must be the same as in the
0004      :                OB20
0005      :
0006      : SPA FB199
0007 NAME: BUB-135 S
0008 ANST:   M 1.7         Initiation bit
0009 SSDB:   KY0,35        Interface DB
000A ANZW:   KY0,8         ANZW = MW8
000B BETR:   KCHS          Transfer direction HS or MS
000C NAM1:   KCAN          ) Sector name here ANTON1
000D NAM2:   KCTO ANTON1   )
000E NAM3:   KCN1          )
```

```
000F TYP :   KCDB
0010 QU-B:   KF+30          Source DB here DB30
                            in the PC DA HS
0011 QUDW:   KF+0           Starting at DWOI
0012 ZI-B:   KG+10          Destination DB here DB10
                            in the bubble
0013 ZIDW:   KF+F           Store at DWO
0014 ZAHL:   KF+1000        Transfer length in DW
0015     :                  Max length = 2015
0016     :
0017     :   BE

OB20

NETZWERK 1       NEUANLAUF
0000     :A DB255
0001     :L KBO             Reset of internal data
0002     :T DWO             Buffer
0003     :T DW8
0004     :T DD9
0005     :SPA FB125
0006 NAME:SYNCHRON
0007 SSDB:    DB35          Interface DB like
                            in FB 130
0008 SSNR:    KYO,0         Interface 0
0009 ANZW:    KYO,254       ANZE = MW254
000A PAFE:    M1.0
000B
000C     :SPA FB210
000D NAME:START
000E     :
000F     :
0010     :
0011     :BE
```

```
FB210          SPRM-B

NETZWERK 1     START FB
NAME: START

0005      :UN M1.0           IF PAFE
0006      :SPB =M001
0007      :STP               THEN STOP
0008 M001:SPA FB126
0009 NAME:ACTIVE
000A SSDB:    DB35
000B      :U   M254.6        Flag word, bit 14
000C      :SPB M001
000D      :
000E      :
000F      :
0010      :BE                The transfer takes place when the
                             START bit is set.


OPERANDS: CONTROL PROCESS IMAGE:

                  |--(*)
                  |
MB1           KM=10000000
MW254         KM=00000000 00000000 Flag word in Synchron
MW8           KM=00000000 00000100 Flag from FB199
                                |--(*)
              (*) ended without error
```

DBs	DB30	
in PC	DW0	KH=0000
	DW1	KH=0000
	DB40	
	DW0	KH=0000
	DW1	KH=0000
	DB50	
	DW0	KH=0000
	DW1	KH=0000
	DB60	KH=0000
	DW1	KH=0000

The flag word informs the user of how the FB-BUB is processed.



Error numbers:

0 = no error
1 = parameter error
2 = block not existing in PC resp. too short
3 = number > 2015 or = 0 resp.
    PC DB longer than 2020 DW (incl. header)
4 = number <> PC DB length (only with operation
    mode HS, if DB not existing on bubble)
5 = DB too short
6 = error on hardware
7 = Bubble memory not formatted
8 = Bubble memory full
9 = operating mode MS -- HS:
    DB not existing resp. data set name not existing
    on bubble
10 = free
11 = erroneus request
12 = handshake error

ANZW in SYNCHRON

(here MW254)

When the handling functions are terminated, the bit positions 13 and 12 give the following information:

	Bit 13	12
Handling function was performed correctly	0	0
Handling function ended in error or was interrupted	0	1
Handling function could not be performed	1	0

Condition code word: (bit 12 to bit 15)

Bits 0 to 11 are irrelevant

	Bit 15	14
Do not enter handling function in the queue	0	0
Enter handling function in the queue	0	1
Handling function started	1	1
End of handling function	0	0

SIMATIC S5

Bubble Memory Module 513-M
Standard Function Blocks
for the AG135U/R processor

6ES5842-0GB10

Operation Manual Order number: E89120-F1944-U81-D

Contents

1 General Considerations
1.1 Field of Application
1.2 Program Structure
1.3 Data Structure

2 Function Block BUB-135 R
2.1 Functional Description
2.2 Parameters
2.2.1 Direct and Indirect Parameter Assignment
2.2.2 Parameterization of SSNR, ANZW, BETR, NAM1, NAM2 and NAM3
2.2.3 Parameterization of TYP, QU-B, QUDW, ZI-B, ZIDW and ZAHL
2.2.4 Parameter Description
2.3 Sample Program of the FB-BUB 135 R
2.3.1 Direct Parameter Assignment
2.3.2 Indirect Parameter Assignment
2.3.3 First Start and Restart

3 S Y N C H R O N F u n c t i o n B l o c k
3.1 Functional Description
3.2 Parameters
3.3 Parameter Description

4 M u l t i p r o c e s s o r O p e r a t i o n
4.1 General Considerations
4.2 Synchronization of an Interface
4.3 Coordination in Multiprocessor Operation
4.4 SES/SEF Semaphore Commands
4.5 Sample Program in Multiprocessor Mode
4.5.1 Configuration
4.5.2 First Start and Restart
4.5.3 Cyclic Program Section

5 S o f t w a r e

6 T e c h n i c a l D a t a

1 General Considerations

1.1 Application

The 513-M bubble memory in the SIMATIC S5-135U Programmable Controller (PC) provides the facility for storing data.

The BUB-135R function block to be parameterized by the user controls the data transfer (receipt/transmission of data) between the programmable controller and the bubble memory.

The FB-BUB-135R can communicate with a maximum of eight bubble memories.

The user can store data blocks (DB) on the bubble memory under a six-digit name (ASCII characters are permitted).

The time required to transfer 1k bytes of data (512 data words) is approx. 3.5 to 23 seconds (for a constant cycle time of 100 ms) according to the block size or the operating mode used (see table, next page).

BLGR	BETR = MS ZAHL = 512	BETR = HS ZAHL = 512	
		approx.23 s	Data block does not exist on bubble
1 (16 bytes)	approx. 10.5 s	approx.12 s	Extend data block on memory
		approx. 9.5s	Data block exists on bubble
		approx.16.5s	Data block does not exist on bubble
6 (512 bytes)	approx. 4.5 s	approx. 6 s	Extend data block on bubble memory
		approx. 3.5s	Data block exists on bubble

The FB BUB-135R uses the following standard data handling blocks for the R processor (order number: 6ES5842-0CB10):

FB 120	SEND	Send request
FB 122	FETCH	Fetch request
FB 127	REC-A	Receive request

Moreover, it is necessary to use the FB 125 SYNCHRON which must be called during cold restart or restart.

1.2 Program Structure

The FB BUB 135R can only be executed after the user has assigned parameters (interface and block size) to the SYNCHRON block during startup (first start OB20, restart OB21, OB22) of the programmable controller (PC) and after this block is processed without errors.

The SYNCHRON handling data block must be called with the appropriate parameters for each interface existing in the PC (bubble memory module).

The synchronization must be successful so that the next SEND, FETCH, REC-A and RESET data handling blocks can communicate properly with an interface.

The FB-BUB block is called in the cyclic program section (e.g. OB1) and the interface number must match the number assigned in the SYNCHRON block.

The FB-BUB-135R can be assigned parameters either directly or indirectly.

Direct parameter assignment is performed by entering directly in the BUB-135R function block.

In the direct parameter assignment, a pointer to a parameter field is transferred with the block parameters (see description of the block parameters).

With the indirect parameter assignment, the user can initialize the FB BUB-135R once.

The parameters are then changed accordingly in a data block (containing an image of the parameters).

The FB-BUB-135R is initialized via an initiation bit if this bit is set by the user. To ensure a proper servicing of the request, the initiation bit must not be reset by the user (during request servicing).

Several cycles are required by the FB-BUB-135R so that data transfer can be accomplished completely.

During this interval, the parameters of the FB-BUB-135R block must not be changed; i.e. a new request can only be initiated after the previous request for the relevant interface (bubble memory module) has ended successfully or has been interrupted by an error.

When the request has been serviced, this initiation bit is reset automatically by the FB-BUB-135R.

1.3 Data Structure

The DB255 data block is reserved for the FB BUB-135 R for communication purposes and is used as sector name and data buffer.

According to the number of interfaces (bubble memories connected) and to the interface number assigned, the DB255 block contains an working area for each interface as well as a data buffer.

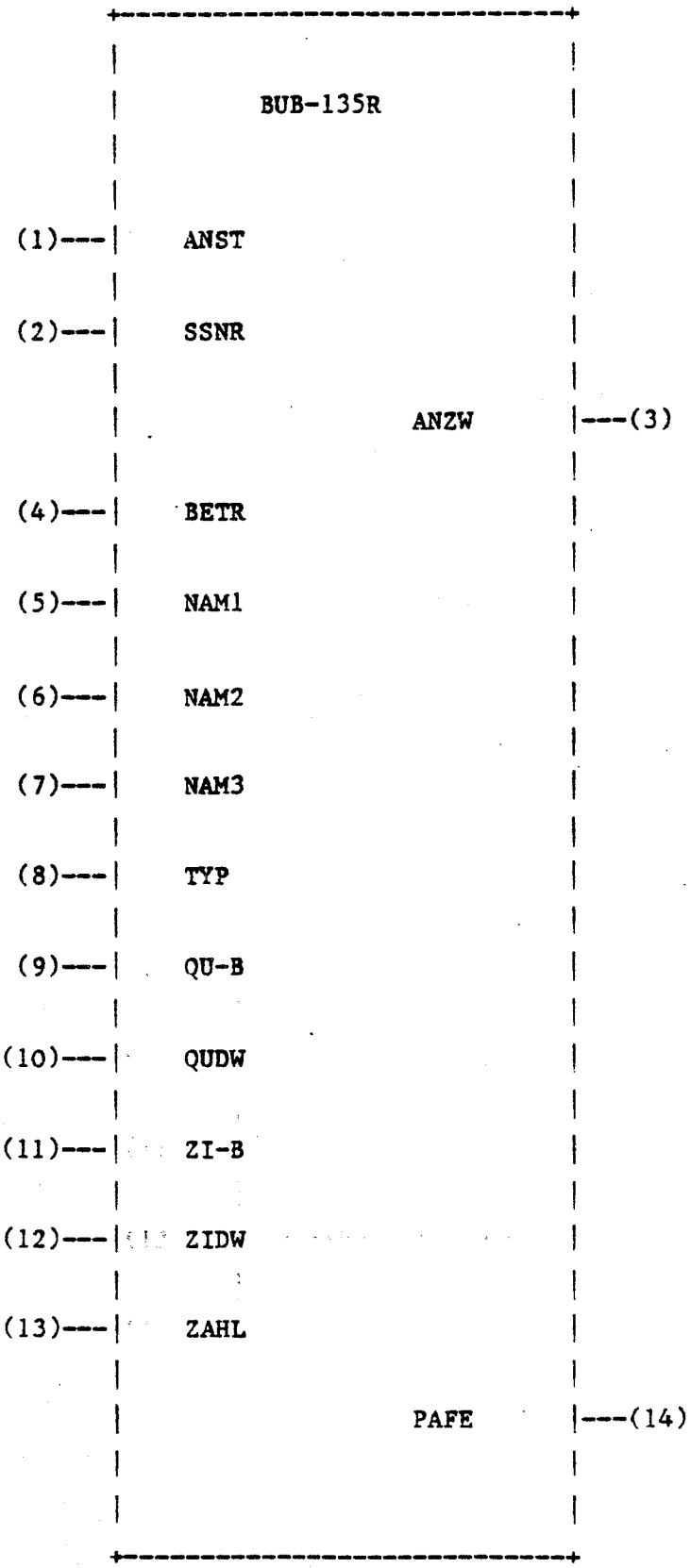
Its minimum length is 66 data words (DWO to DW65) for one interface.

Its maximum length is 283 data words (DWO to DW282) for eight interfaces).

DB 255

DW 0 DW 34	Working area
DW 35 DW 65	Data buffer Interface 0
DW 66 DW 96	Data buffer Interface 1
DW 97 DW 127	Data buffer Interface 2
DW 128 DW 158	Data buffer Interface 3
DW 159 DW 189	Data buffer Interface 4
DW 190 DW 220	Data buffer Interface 5
DW 221 DW 251	Data buffer Interface 6
DW 252 DW 282	Data buffer Interface 7

2 BUB-135R Function Blocks



2.1 Functional Description

The FB-BUB-135R block is responsible for transfer of DBs between the PC 135 U/R processor and the 513M bubble memory. The FB BUB-135R can communicate with a maximum of 8 bubble memories.

Loading blocks from the bubble memory into the programmable controller (PC) is only possible if the data block to be loaded (destination data block) is already stored in the memory of the PC.

Writing data blocks from the PC to the bubble memory is also possible if the data block to be written (destination block) is not stored on the bubble memory yet:

It is then created in the bubble memory according to the appropriate length (prerequisite: the offset address is zero).

Data blocks, in whole or in part, can be transferred:

The following rules must be observed:

- The source data block number and destination data block number can be different.

- The length of the destination data block is obtained from the maximum length of the existing destination block (if it does not exist: length = 0) and the length of the block to be transferred plus an offset address which can be specified (offset relative to the beginning of the destination data block).
  Appropriate corrections are made in the block header.
  These conventions apply to a transfer of a data from the PC memory to the bubble memory.

- During transfer, no free space should be left undefined. For instance, a partial transfer to a non-existing block is not permitted if the offset address is greater than zero (this would lead to an undefined "header gap").
  In the same way, it is not permitted to transfer a part block with an offset address that is greater than the length of the destination block (this would lead to an undefined "intermediate gap").

- The maximum length of a data block from/to which data can be transferred is 2048 data words (DW0 to DW4095).

- The maximum number of data words which can be transferred is 2048 data words.

2.2 Parameters

ANST = Initiation bit
SSNR = Interface data block
ANZW = Flag word
BETR = Operating mode
NAM1 =)
NAM2 =) > Name of data set (also sector)
NAM3 =) on the bubble memory
TYP = Block type
QU-B = Source data block number
QUDW = Source data word number
ZI-B = Destination data block number
ZIDW = Destination data word number
ZAHL = Number of words to be transferred
PAFE = Parameter error byte

2.2.1 Direct and Indirect Parameter Assignment

The FB BUB-135R includes a direct and indirect assignment of the block parameters.

Only the initiation bit (ANST) and the parameter error byte (PAFE) have to be entered directly.

In case of the direct parameter assignment, the FB BUB-135 R processes directly the parameters specified in the block call.

In case of the indirect parameter assignment, a pointer to a parameter field is transferred to the FB BUB-135 R with the block parameters.
The "actual" parameters are arranged without any gap in the same order as for the direct parameter assignment.

2.2.2 Parameter Assignment of SSNR, ANZW, BETR, NAM1, NAM2 and NAM3

The high byte (left byte) of the SSDB parameter is used as switching criterion for direct/indirect parameter assignment.

High byte of SSDB = 0: direct parameter assignment:
SSDB, ANZW, BETR and NAM1, NAM2, NAM3 are parameterized directly

High byte of SSNR <>0: indirect parameter assignment:
SSNR, ANZW, BETR, NAM1, NAM2, NAM3 are stored in the data block which has been opened before calling the FB BUB-135R, starting at the data word indicated in the low byte of SSDB.

The ANZW, BETR, NAM1, NAM2 and NAM3 block parameters are irrelevant in the indirect parameter assignment.

2.2.3 Parameter Assignment of TYP, OU-B, QUDW, ZI-B, ZIDW and ZAHL

In direct parameter assignment, the FB BUB-135R processes directly the source and destination parameters specified in the block call; in indirect parameter assignment, the block parameters constitute a pointer to a parameter field in a data block which contains the "actual" source and destination parameters.

Parameter sequence and data format are similar to those used in direct parameter assignment (data format KC).

Indirect parameter assignment of the source and destination parameters is achieved with parameter TYP = XX.
The QU-B parameter gives the data block number and QUDW the word number at which the TYP = "XX" and ZAHL parameters are assumed by the FB BUB-135 R.

The block parameters ZI-B, ZIDW and ZAHL are irrelevant for the indirect parameter assignment.

2.2.4 Parameter Description

* ANST: Initiation bit

The initiation bit (ANST) is set by the user if data is to be transferred from the PC to the bubble memory.

This bit must be set by the user.
It is reset by the FB BUB-135R block after the request has been serviced.

Kind of parameter/Type	:Input/bit
Possible area	:E 0.0 ... E 127.7 A 0.0 ... A 127.7 M 0.0 ... M 255.7
Admissible area	:A 0.0 ... A 127.7 (if available) M 0.0 ... M 255.7

* SSNR : Interface number

This number is the number of the interface used to address the bubble memory module.

The interface number must match the interface number which has been parameterized for this interface in the SYNCHRON data handling block.

Kind of parameter/Type:	Datum / KY
Possible area:	0.0 ... 255.255
Admissible area:	0.0 ... 0.7
High byte = 0:	direct parameter assignment: low byte = SSNR
High byte <> 0:	indirect parameter assignment, low byte = pointer to parameter field

* ANZW: Address of flag word

Address of the flag word containing the flags indicating the request servicing carried out by the FB-BUB-135R.

It contains the status and error messages which are generated by the bubble memory firmware and by the FB-BUB-135R. These messages are available to the user once the request has been terminated or interrupted.

The user can place the flag word (16-bits long) either into the flag area or into the data block.

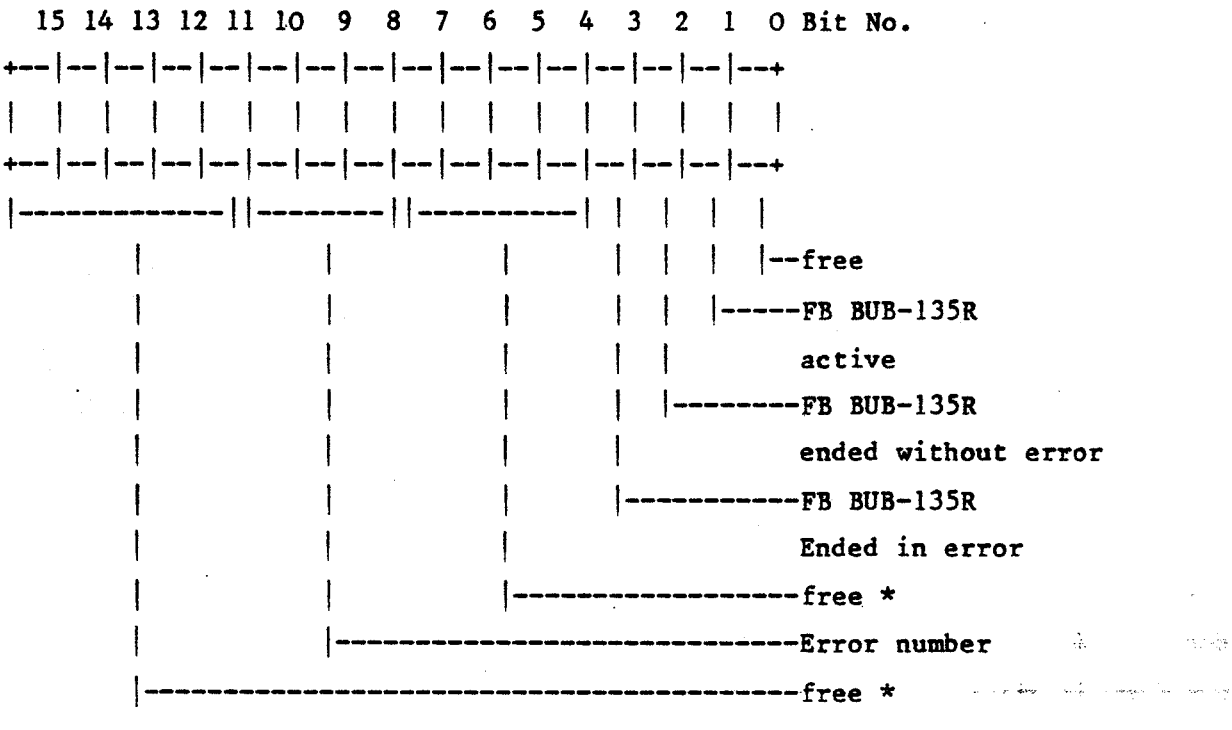
The high byte (left byte) of the ANZW parameter is the switching criterion:

High byte = 0 : Flag word is located in the flag area, and the low byte (right byte) indicates the flag word address of the flag word

High byte <> 0 : Flag word is located in the data block and the high byte contains the number of this data block; the low byte contains the data word address of the flag word

Kind of parameter/Type:	Datum /KY
Possible area:	0.0 ... 255.255
Admissible area:	0.0 ... 0.254 (flag area) 3.0 ... 254,255 (data block)

Structure of the flag word:



Note: * These bits are cleared by the FB BUB-135R when the flag word is written.

The FB-BUB-135R issues the message "Request ended in error" in bit 3 of the flag word, the tetrad from bit 8 to 11, contains an error number which gives information on the cause of the interrupt.

Error numbers in the flag word:

(1 to 5: errors detected by the programmable controller)

(6 to CH: errors detected by the bubble memory)

0 = No error
1 = Parameter error
3 = Operating mode HS ---> MS: Data block does not exist on bubble; danger of "header gap"

4 = Operating mode MS ---> HS: Data block on bubble too short
Operating mode HS ---> MS: Data block on bubble too short: Danger of "intermediate gap"
5 = ZAHL = "wild card length": Source data block (QU-B) too long: data block without block header greater than 2048 data words

6 = Hardware error
7 = Bubble memory not formatted
8 = Bubble memory full
9 = Operating mode MS ---> HS: data block or name of data set does not exist on bubble

AH = No. not used
BH = Erroneous request
CH = Handshake error

* BETR : Operating mode

There are two operating modes for the FB BUB-135R:

HS --> MS: Transfer form the main memory of the PC (HS) to the bubble memory (MS)

MS --> HS: Transfer form the bubble memory (MS) to the main memory of the PC (HS)

Kind of parameter/Type: Data / KC

Possible area: All ASCII characters

Admissible area: HS (HS --> MS)
MS (MS --> HS)

*NAM1, NAM2, NAM3: Name of data set, sector name

The data blocks are stored on the bubble under a data set name (also sector name) comprising NAM1, NAM2 and NAM3.

Hence, data blocks can be stored under as many data block names as desired.

The data set name consists of 6-digit ASCII characters

* NAM1 : Name part 1

Characters 1-2 of the six-digit name (sector name) of the bubble data set into/from which the block is to be transferred.

Kind of parameter/Type: Data / KC

Possible area: All ASCII characters

Admssible area: All ASCII characters

* NAM2 : Name part 2

Characters 3 and 4 of the six-digit name (sector name) of the bubble data set into/from which the block is to be transferred.

(NAM2)

Kind of parameter/Type: Data / KC

Possible area: All ASCII characters

Admissible area: All ASCII characters

* NAM3 : Name part 3

Characters 5 and 6 of the six-digit name of the (sector name) bubble data set into/from which the block is to be transferred.

Kind of parameter/Type: Data / KC

Possible area: All ASCII characters

Admissible area: All ASCII characters

* TYP : Block type

This parameter is the switching criterion used for switching between direct and indirect parameter assignment of TYP, QU-B, QUDW, ZI-B, ZIDW and ANZ:

If TYP contains the value KC = "XX", the parameters are assigned indirectly.

Kind of parameter/Type: Data / KC

Possible area: All ASCII characters

Admissible area: KC = "DB" : direct parameter assignment
KC = "XX" : indirect parameter assignment

* QU-B : Number of source data block

In direct parameter assignment, this parameter contains the number of the source data block from which the data is transferred; in indirect parameter assignment, it contains the number of the data block which contains the parameter list (TYP = XX).

Kind of parameter/Type: Data / KF

Possible area: - 32768 ... + 32767

Admissible area. + 2 ... + 254 (BETR = HS ---> MS)
+ 1 ... + 254 (BETR = MS ---> HS)
+ 3 ... + 254 (indirect parameterizing)

* QUDW : Number of the source data word

First word in the source data block which is transferred (for direct parameter assignment); or data word address at which the parameter list is stored (for indirect parameter assignment).

Kind of parameter/Type: Data / KF

Possible area: - 32768 ... + 32767

Admissible area: + 0 ... + 2048 (direct parameterizing)
+ 0 ... + 250 (indirect parameterizing)

* ZI-B : Number of the destination block

This parameter contains the number of the source data block to which the data is transferred (for direct parameter assignment); it is irrelevant in the case of the indirect parameter assignment (TYP = XX).

Kind of parameter/Type: Data / KF

Possible area: - 32768 ... + 32767

Admissible area: + 1 ... + 254 (BETR = HS ---> MS)
+ 2 ... + 254 (BETR = MS ---> HS)

* ZIDW : Number of the destination word

First word in the destination block to which data is transferred (for direct parameter assignment); this parameter is irrelevant in the case of the indirect parameter assignment (TYP = XX).

Kind of parameter/Type: Data / KF

Possible area: - 32768 ... + 32767

Admissible area: + 0 ... + 2048

* ZAHL : Number of data words to be transferred

This parameter contains the number of words to be transferred (for direct parameter assignment).

(ZAHL)
If wild card length is used (ZAHL = -1), the actual data block length of the source block is transferred from the data block header; the QUDW parameter is irrelevant, i.e. it is set to zero by the FB-BUB-135R: The transfer begins in the source data address at DW 0.
This parameter is irrelevant in the indirect parameter assignment.

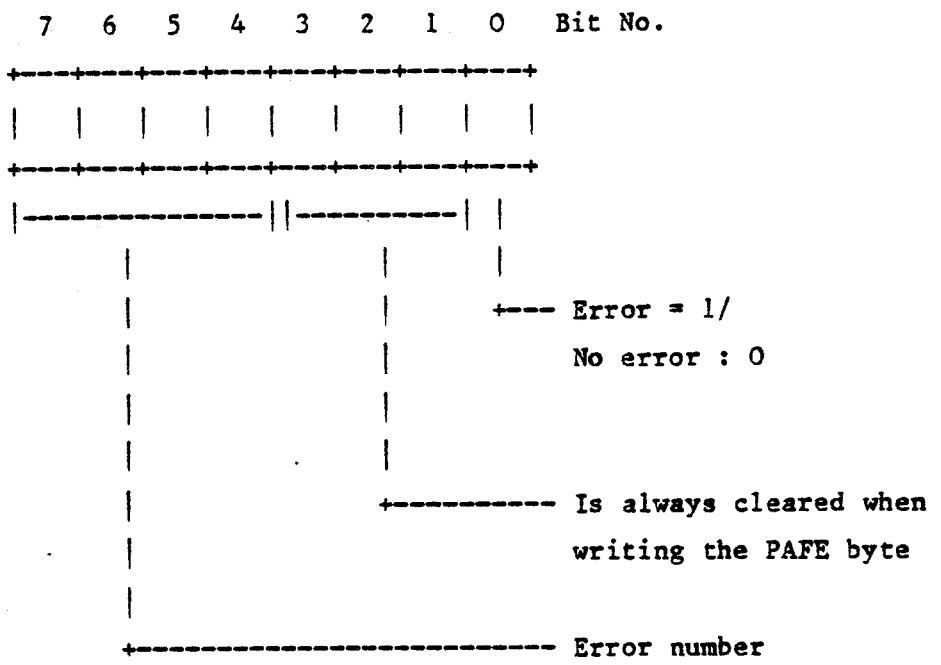
Kind of parameter/Type:	Data / KF
Possible area:	- 32768 ... + 32767
Admissible area:	+ 1 ... + 2048 - 1 (wild card length)

* PAFE : Parameter error byte

The byte indicated here (flag byte, output byte, input byte) is set when the FB BUB-135R block detects a parameter error.

Kind of parameter/Type:	Output /BY
Possible area:	EB 0 ... EB 127 AB 0 ... AB 127 MB 0 ... MB 255
Admissible area:	EB 0 ... EB 127 (if available) AB 0 ... AB 127 (if available) MB 0 ... MB 255

Structure of the PAFE byte:



The following error numbers may occur in the PAFE byte:

(PAFE error numbers)

0 = No error

1 = Data block for indirect parameter assignment does not exist or is too short

2 = Interface number (SSNR) > 7 or interface SSNR does not exist

3 = Data block for the ANZW parameter does not exist

4 = Wrong operating mode (BETR):
BETR : "HS" ; BETR = "MS"

5 = Wrong block type (TYP):
TYP = "DB"; TYP = "XX"

6 = Illegal block number (QU-B, ZI-B):

BETR = HS ---> MS : QU-B < 2; QU-B > 254;
ZI-B < 1; ZI-B > 254

BETR = MS ---> HS : QU-B < 1; QU-B > 254;
ZI-B < 2, ZI-B > 254

7 = Data block does not exist in the programmable controller or is too short:

BETR = HS ---> MS : QU-B ; QUDW > DB length;
QUDW + ANZ > DB length;

BETR = MS ---> HS : ZI-B ; ZIDW > DB length;
ZIDW + ANZ > DB length;

8 = Illegal data word number (QUDW; ZIDW):

QUDW/ZIDW < 0 ; QUDW/ZIDW > 2048

9 = Number (ZAHL) illegal:

ZAHL <= 0; ZAHL = -1; ZAHL > 2048

(PAFE error numbers)

AH Data handling block DB 255 does not exist or is too short

BH ZAHL = "wild card length" : Source data block (QU-B) is too long: length of data block without header is greater than 2048 data words

2.3 Sample Program of the FB BUB-135R

By setting the initiation flag M100.0, the user issues a request to the FB BUB-135R.

100 data words of the DB 35 data block are to be transferred, starting at DW 15, from the main memory of the PC to the bubble memory in DB 40 starting at DW 30.

The interface number 2 has been parameterized for the bubble memory.

The flag word is stored in flag word MW 180.

Parameter errors are stored in the flag byte MB 50.

2.3.1 Direct Parameter Assignment

OB 1

```
•
•
•
                : U  M   90.0 | Flag for initiation activation
                : UN M  100.0 | Initiation flag
                : R  M   90.0 | Reset activation flag
                              |
                : S  M  100.0 | Set initiation flag
                : U  M  100.0 | Initiation ?
                : SPB FB199   | Call of FB-BUB-135R
NAME            : BUB-135R    |
ANST            :    M 100.0  | Initiation bit : M100.0
SSNR            :    KY0,2    | Interface number : 2
ANZW            :    KY0,180  | Condition code word address: MW 180
BETR            :    KCHS     | Operating mode: HS ---> MS
NAM1            :    KCDA     | )
NAM2            :    KCTU     | )> Sector name : DATUM1
NAM3            :    KCM1     | )
TYP             :    KCDB     | Block type : DB
QU-B            :    KG+35    | Source data block number : 35
QUDW            :    KF+15    | Source data word number : DW 15
ZI-B            :    KF+40    | Destination data block number: 40
ZIDW            :    KF+30    | Destination data word number : DW 30
ZAHL            :    KF+100   | Number: 100 data words
PAFE            :    MB50     | PAFE byte : MB 50
•
•
                :    BE       |
```

The initiation bit (M100.0) is reset by FB BUB-135R once the request is completed.

2.3.2 Indirect Parameter Assignment

The SSNR, ANZW, BETR, NAM1, NAM2, NAM3 parameters are stored in DB 101, starting at DW 10.

The TYP, QU-B, QUDW, ZI-B, ZIDW and ANZ parameters are stored in DB 110, starting at DW 1.

```
OB 1
•
•
                : U     90.0    | Flag for initiation
                                | Activation
                : UN M 100.0    | Initiation flag
                : R  M  90.0    | Reset activation flag
                : S  M 100.0    | Set initiation flag
                : A  DB101      | Call of parameter DB <-------|
                : U  M 100.0    | Initiation ?                 |
                : SPB FB199     | Call of FB BUB-135R          |
   NAME         : BUB 135R      |                              |
   ANST         :    M 100.0    | Initiation bit: M 100.0      |
   SSNR         :    KY255,10   | Parameters are in DB 101-----|
                                | starting at DW 10 (left para-
                                | meter byte <> 0---> indirect
                                | parameter assignment)
   ANZW         :    KY0,0      | )
   BETR         :    KCHS       | )
   NAM1         :    KC99       | )>>> Irrelevant
   NAM2         :    KC         | )
   NAM3         :    KCM1       | )
   TYP          :    KCXX       | Indirect parameter assignment
   QU-B         :    KF+110     | Parameter DB: DB 110
   QUDW         :    KF+1       | Parameter list begins with DW | 1
   ZI-B         :    KF+0       | )
   ZIDW         :    KF+0       | >>> Irrelevant
   ZAHL         :    MB50       | PAFE byte : MB 50
   •
   •
                :    BE
```

The initiation bit (M100.0) is reset by the FB BUB-135R once the request is serviced.

Parameter list SSNR, ANZW, BETR, NAM1, NAM2, NAM3

DB 101

```
DW 10 :   KY= 0.2          Interface number : SSNR = 2
   11 :   KY= 0,180        Condition code word address MW 180
   12 :   KC= HS           Operating mode = HS ---> MS
   13 :   KC= DA           )
   14 :   KC= TU           )> Sector name: DATUM1
   15 :   KC= M1           )
```

Parameter list TYP, QU-B, QUDW, ZI-B, ZIDW, ANZ

DB 110

```
DW  1 :   KC= DB           Block type = DB
    2 :   KF= +35          Source data block: DB 35
    3 :   KF= +15          Source data word : DW 15
    4 :   KF= +40          Destination data block: DB 40
    5 :   KF= +30          Destination data word:  DW 30
    6 :   KB= +100         Number = 100 data words
```

2.3.3 First Start and Restart

During first start and restart, the SYNCHRON function block must be called for each interface available (bubble memory).

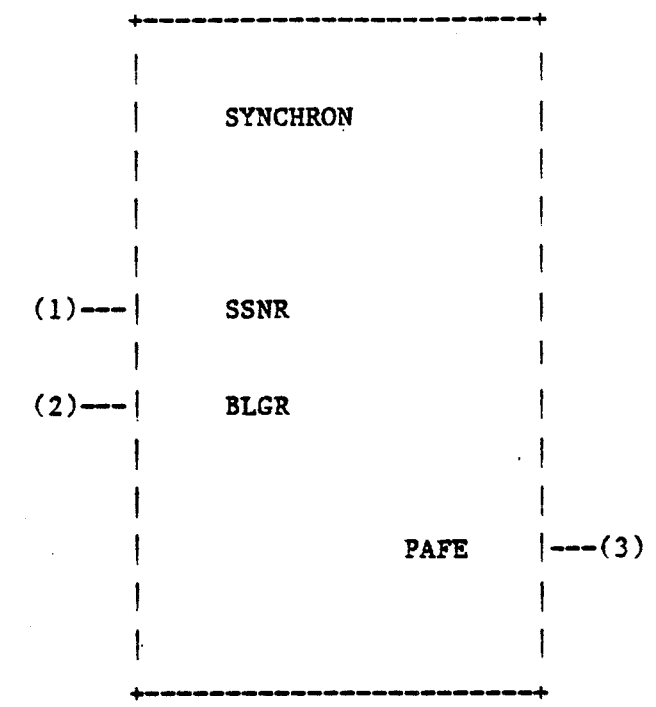
Error messages of the SYNCHRON function block must be evaluated by the user since a correct synchronization of the interface is required by the FB BUB-135R.

Moreover, the internal status word of the FB BUB-135R must be deleted for each interface.

OB 20, OB 21; OB 22:

```
        •
        •
        •
        :SPA FB 125      Call of the handling block
NAME    :SYNCHRON        SYNCHRON
SSNR    :KYO,2           Interface number : 2
BLGR    :KYO,4           Block size : e.g.: 4 (128 bytes)
PAFE    :MB255           Parameter error byte:

        •
        •
        •
        •
        •
        :A DB255         Data buffer for the FB BUB-135R
        :L KBO
        :T DW31          Clear internal status word
        :T DW35          Delete status word interface 0
        :T DW66          Delete status word interface 1
        :T DW97          Delete status word interface 2
        :T DW127         Delete status word interface 3
        :T DW159         Delete status word interface 4
        :T DW190         Delete status word interface 5
        :T DW221         Delete status word interface 6
        :T DW252         Delete status word interface 7
        :
        •
        •
        •
        •
        •
```

3 SYNCHRON Function Block



3.1 Functional Description

The SYNCHRON function block (FB 125) is responsible for synchronizing the central module and the interface (bubble memory).

The interface between the central module and the interface is deleted or preset.

The interface must be synchronized in OB 20 (first start), in OB21 (manual restart) as well as in OB 22 (restart following power failure).

This handling block can only be called at startup time, i.e. during first start or restart.

3.2 Parameters

SSNR = Interface number
BLGR = Block size
PAFE = Parameter error byte

3.3 Parameter Description

* SSNR : Interface number

Number of the interface for accessing the bubble memory.

Kind of parameter/Type: Data / KY

Possible area: 0.0 ... 255,255

Admissible area: 0.0 ... 0.7

High byte = 0 : Direct parameter assignment
Low byte = SSNR

High byte <> : Indirect parameter assignment;
Low byte = pointer to parameter field;
The parameters SSNR and BLGR are located in the data block opened prior to the call of FB SYNCHRON, in the same way (sequence, format) as for the direct parameter assignment.

* BLGR : Block size

On system startup, the block size is determined by the FB SYNCHRON.

The size can be affected by the BLGR parameter.

To determine the block size, the SYNCHRON block transfers a "desired" block size to the interface, according to the BLGR parameter.

This size is then checked and changed if necessary.

The "resulting" block size constitutes the upper limit for the SEND and RECEIVE blocks; it limits the maximum (net) number of data bytes to be received/sent for each block call.

If the area to be transferred is longer than this block, subsequent blocks are then transferred.

To obtain a fast throughput, select a greater block size (load of programmable controller cycle).

To reduce the run time of the function blocks, set a small block size (slow data throughput).

Kind of parameter/Type: Data / KY

Possible area: 0,0 ... 0,255

Permissible area: 0,0 ... 0,255

The block size is set according to the table shown on the next page.

Table for block size

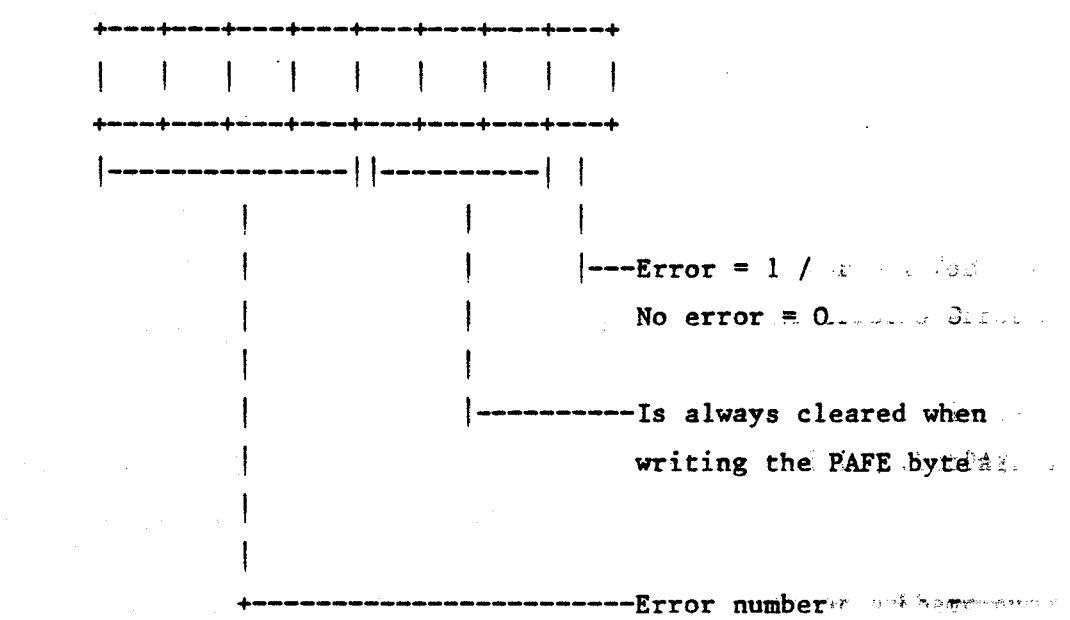
Parameter value BLGR	Meaning	
0	Standard value (256 bytes)	
1	16 bytes	
2	32 bytes	desired block
3	64 bytes	> size
4	128 bytes	
5	256 bytes	
6	512 bytes	
7...254	Standard value (256 bytes)	
255	Block size set to 512 bytes	

* PAFE : Parameter error byte

The byte indicated here (flag byte, output byte, input byte) is set when the FB-BUB-135U block detects a parameter error.

Kind of parameter/Type:	Output / BY
Possible area:	EB 0 ... EB 127 AB 0 ... AB 127 MB 0 ... MB 255
Admissible area:	EB 0 ... EB 127 (if available) AB 0 ... AB 127 (if available) MB 0 ... MB 25

Structure of the PAFE byte:



The following error numbers may occur in the PAFE byte:

0	No error
1 to 4	Erroneous source/destination parameters
1	Wrong area
2	DB or DX non-existent
3	Area too small
4	Area does not exist
5	Erroneous flag word
6	Not reserved
7	Interface does not exist
8	Interface not ready
9	Interface overloaded
AH	Interface is set by another module
BH	Request number is not permitted
CH	Interface does not react in time
DH	Other interface errors, such as - Erroneous (not admitted) handshake acknowledge - Block size of interface does not lie within permissible range 1...255
EH	Other errors at the handling block, such as - Call of a data block missing for indirect parameter assignment - Software error of central module or of handling block

(continue next page)

FH FB call not admitted; such as
- Double call in case of interruptibility
- Invalid amendment of standard function blocks

Note: For more details, refer to the operating manual "S5-135U programmable controller handling blocks for the R processor". Order No. C79000-B8500-C366-01.

4 Multiprocessing

4.1 General Considerations

The FB BUB-135R supports multiprocessing; i.e. several central modules can access a bubble memory by means of a call of the FB BUB-135R.

Multiprocessing is not supported by the bubble memory; the latter is only capable of coordinating and processing one current request at a time. Consequently, the user must see to it that an interface (bubble memory) receives a request from only one central module. I.e. as soon as a request is initiated for an interface (bubble memory) (ANZW FB BUB-135R = FB active), care must be taken that a request activation is not performed by one/several other central modules for this interface as long as the request is being serviced.

4.2 Synchronizing an Interface

To synchronize an interface, refer to sections 4.1 and 4.3.

More information about multiprocessing:

For run-time purposes, an interface (bubble memory) should be synchronized by one central module only.
The block size (parameter BLGR) must then be selected in order to come up to all the central modules which have exchanged data with the interface (bubble memory).
If an interface is synchronized several times in a row, the last value used for BLGR is then valid.

If an interface is synchronized, i. e. as long as, on a central module, a SYNCHRON FB is running for this interface, handling blocks which are also accessing this interface must not be called on other central modules.

4.3 Coordination in Multiprocessing

When bubble memories AS 523 are used in multiprocessing, the difference is made between:

- Accessing an interface (bubble memory) by one central module only

- Accessing an interface (bubble memory) by several modules

In the first case, data exchange between the central module and the bubble memory is similar to single processor operation; i.e. it is not necessary to take further measures.

In the second case (several central modules), the data exchange must be organized in such a way that, as soon as an interface (bubble memory) is active for a central module, it cannot be activated by other central modules (all of FB BUB-135R). i.e. FB BUB-135R must be inhibited for other central modules within this period of time.

This coordination between the central modules is effected through the so-called semaphore instructions SES and SEF.

4.4 SES/SEF Semaphore Instructions

The SES nn (set semaphore) and SEF nn (release semaphore) are part of the system operations and control the data exchange between the central module and the Cps (bubble memories) in multiprocessing.

By setting a semaphore SES nn (nn = 0...31), a data area or an FB call can be inhibited for other central modules.

The "release semaphore" instruction (SEF nn) causes the semaphore previously set to be reset and, consequently, to be read and written to.

A semaphore can only be set if it is not set already, i.e. if it has not been set by any other central module.

A semaphore can only be released by the central module which set it.

The SES/SEF instructions affect the flag bits as follows:

ANZ1	ANZ0	Meaning
0	0	Semaphore is set by another central module and cannot be set or reset
0	1	Semaphore is set/reset

The flags can be evaluated by the jump functions.

Important:

Choose a different semaphore number for each bubble memory module connected to the central rack, which is accessed by several (different) central modules).

4.5 Sample Program in Multiprocessor Mode

4.5.1 Configuration

KOOR	ZBG 1	ZBG 2	AS513 (1)	AS513 (2)	AS513 (3)

ZBG 1 : R processor
reads from/writes to AS513 (1)
reads from/writes to AS513 (2)
reads from/writes to AS513 (3)

ZBG 2 : R processor
reads from/writes to AS513 (1)
reads from/writes to AS513 (2)

AS513 (1) : is parameterized as follows:
SSNR = 0

AS513 (2) : is parameterized as follows:
SSNR = 4

AS513 (3) : is parameterized as follows:
SSNR = 6

Assignment AS513 (1) : Semaphore 31

Assignment AS513 (2) : Semaphore 30

The interface is synchronized by ZBG 1

4.5.2 First Start and Restart

First start and restart ZBG 1:

OB 20; OB 21; OB 22:

```
     :
     :
     : SPA FB 254       Call of function block for
NAME : NEUSTART         first start / restart
```

The NEUSTART FB is set up as follows:

```
FB254                   First start / restart

NAME : NEUSTART

     : L   KH0000       Delete the semaphore area
     : L   KHF33F
     : TIR  3
     : O   KHF33E
     : TNB 31
     :
     : SPA FB125        Synchronization AS513 (1)

NAME : SYNRHON
SSNR : KY0,0            Interface number : 0
BLGR : KY0,4            Block size: e.g. 4 (128 bytes)
PAFE : MB255
     :
     : SPA FB 125       Synchronization AS 513 (2)
NAME : SYNCHRON
```

```
SSNR : KYOI,4          Interface number : 4
BLGR : KYO,5           Block size e.g. 5 (256 bytes)
PAFE : MB254
     :
     : SPA FB 125      Synchronization AS513 (3)
NAME : SYNCHRON
SSNR : KYO,6           Interface number : 6
BLGR : KYO,6           Block size e.g. 6 (512 bytes)
PAFE : MB253
     :
     : A DB255         Data buffer for the FB BUB-135R
     : L KBO
     : T DW31          Clear internal Status word
     : T DW35          Delete status word interface 0
     : T DW159         Delete status word interface 4
     : T DW221         Delete status word interface 6
     :
     : BE
```

First start and restart ZBG 2:

Only the status words of the data buffers have to be deleted for the interface since the interfaces are not synchronized and the semaphore area need not be deleted.

OB 20, OB 21, OB 22

```
     :
     : A DB255         Data buffer for the FB BUB-135R
     : L KBO
     : T DW31          Clear internal status word
     : T DW35          Delete status word interface 0
     : T DW159         Delete status word interface 4
     :
     : BE
```

4.5.3 Cyclic Program Part

Cyclic program part ZBG 1:

```
OB 1
     :
     :
     : SPA FB 210        Call coordination FB ZBG 1 for multi-
NAME : BUB-KOOR          processing FB BUB-135R
     : BE
```

The BUB-KOOR is set up as follows:

```
FB210                    Coordination ZBG1 FB BUB-135R
                         multiprocessing

NAME : BUB-KOOR

     : UN M 100.0        Initiation for FB BUB-135R for AS 513 (1)?
     : SPB =SEF1         No ---> go on
     : SES  31           Set semaphore 31
     : SPP =AN 1         Action successful ?
     : SPA =TES1         No ---> go on
AN 1 : SPA FB199         Call FB BUB-135R for AS 513 (1)
NAME : BUB-135R          This interface is assigned semaphore 31
ANST :    M 100.0
SSNR :   KY0,0           SSNR = 0
ANZW :   KY0,240
BETR :   KCMS
NAM1 :   KCF0
NAM2 :   KCLG
NAM3 :   KCE2
TYP  :   KCDB
QU-B :   KF+70
QUDW :    KF+0
ZI-B :    KF+20
```

```
ZIDW :    KF+100
ZAHL :    KF+70
PAFE :    MB247
     : U  M 100.0      FB BUB-135R for AS513 (1) active ?
     : ON M 247.0      Parameter error ?
     : SPB =TES1       ----> go on
SEF1 : SEF  31         Release semaphore 31 for other central
     :                 module
     :
TES1 :
     : UN M 100.0      Initiation for FB-BUB-135R for AS513 (2)?
     : SPB =SEF2       No ---> go on
     : SES   30        Set semaphore 30
     : SPP =AN 2       Action successful ?
     : SPA =TES2       No ---> go on
AN2  : SPA FB199       Call FB BUB-135R for AS513 (2)
NAME : BUB-135R        This interface is assigned semaphore 30
ANST :    M 100.1
SSNR :    KY0,4        SSNR = 4
ANZW :    KY0,242
BETR :    KCHS
NAM1 :    KCTE
NAM2 :    KCST
NAM3 :    AKCO5
TYP  :    YRCDB
QU-B :    QUKE+30
QUDW :    KF+10
ZI-B :    ZIKF+20
ZIDW :    ZIKF+0
ZAHL :    ZAKF-1
PAFE :    MB248
     :
     : U  M 100.1      FB BUB-135R for AS 513 (2) active?
     : ON M 248.0      Parameter error?
     : SPB =TES2       ----> go on
     :
SEF2 : SEF 30          Release semaphore 30 for other central
```

```
TES2 :                  module
     : SPA FB199        Call FB-BUB-135R for AS 513 (3)
NAME : BUB-135R
ANST :    M 100.2
SSNR :    KY0,6        SSNR = 6
ANZW :    KY0,244
BETR :    KCHS
NAM1 :    KCRE
NAM2 :    KCZE
NAM3 :    KCPT
TYP  :    KCDB
QU-B :    KF+20
QUDW :    KF+5
ZI-B :    KF+35
ZIDW :    KF+10
ZAHL :    KF+40
PAFE :    MB249
     :
     : BE
```

Cyclic program part ZBG 2:

```
OB 1
     :
     :
     : SPA FB210        Call coordination FB ZBG1 for multi-
NAME : BUB-KOOR         processing FB BUB-135R
     : BE
```

The BUB-KOOR is set up as follows:

```
FB 210                  Coordination ZBG2 FB BUB-135R
                        multiprocessing

NAME : BUB-KOOR

TESO : UN M 90.0        Initiation for FB BUB-135R for AS513 (1)?
     : SPB =SEF1        No ---> go on
     : SES  31          Set semaphore 31
     : SPP =AN 1        Action successful?
     : SPA =TES1        No ---> go on
AN 1 : SPA FB199        Call FB BUB-135R for AS513 (1)
NAME : BUB-135R         This interface is assigned semaphore 31
ANST :     M 90.0
SSNR :     KY0,0        SSNR = 0
ANZW :     KY0,120
BETR :     KCMS
NAM1 :     KCAU
NAM2 :     KCTO
NAM3 :     KC22
TYP  :     KCDB
QU-B :     KF+77
QUDW :     KF+21
ZI-B :     KF+22
ZIDW :     KF+27
ZAHL :     KF+270
PAFE :     MB130
     : U   M 90.0        FB BUB-135R for AS513 (1) active?
     : ON  M130.0        Parameter error?
     : SPB =TES1         ---> go on
SEF1 :
     : SEF 31            Release semaphore 31 for other
     :                   central module
TES1 :
```

```
     : UN M 90.1        Initiation for FB-BUB-135R for AS513 (2)?
     : SPB =SEF2        No ---> go on
     : SES   30         Set semaphore 30
     : SPP =AN 2        Action succesful ?
     : SPA =TES2        No ---> go on

AN 2 : SPA FB199        Call FB BUB-135R for AS513 (2)
NAME : BUB-135R         This interface is assigned semaphore 30
ANST :     M 90.1
SSNR :     KY0,4        SSNR = 4
ANZW :     KY0,122
BETR :     KCHS
NAM1 :     KCHA
NAM2 :     KCLL
NAM3 :     KCE3
TYP  :     KCDB
QU-B :     KF+55
QUDW :     KF+0
ZI-B :     KF+130
ZIDW :     KF+0
ZAHL :     KF+1000
PAFE :     MB131
     : U   M 90.1        Call FB BUB-135R for AS513 (2) active?
     : ON  M 13N.0       Parameter error?
     : BEB               ----> END
SEF2 :
     : SEF   30FF        Release semaphore 30 for other central
TES2 :                   module
     :BE
```

5 S o f t w a r e

Software required for a maximum of 8 bubble memory modules in the AG 135 U (R processor):

- OB 20, OB 21, OB 22 (first start, restart) parameterized for 8 interfaces with the SYNCHRON FB 125
- FB 199 BUB-135R
- FB 120 SEND
- FB 122 FETCH
- FB 127 REC-A
- DB 255 Data buffer
- FB NEUSTART *)
- FB BUB-KOOR *)

*) only in multiprocessing mode, when several central modules access an interface.

6 Technical Data

Block number:	FB 199
Block name:	BUB-135R
Block length:	818 words
Call length:	16 words
Processing time:	Approx. 4 ms to 18 ms
Nesting depth:	1
Function blocks called:	FB 120 SEND FB 122 FETCH FB 127 REC-A
Data blocks called:	DB 255 Data buffer DB "PARA" (indirect parameter assignment)
Organization blocks called:	OB 217 Read data from page frame
Flag area allocated: (scratchpad area)	MW 200 - MW 220 MB 255
System data allocated:	BS 240 (scratchpad cell)

S I E M E N S

SIMATIC S5

Bubble Memory Module 513-M
Standard Function Blocks
for AG 150U 6ES5844-0GA10

Operation Manual Order number: E89120-F1944-U71-B

Contents

1 G e n e r a l C o n s i d e r a t i o n s
1.1 Field of Application
1.2 Program Structure
1.3 Data Structure

2 BUB-150U F u n c t i o n B l o c k
2.1 Functional Description
2.2 Parameters
2.3 Sample Program with the BUB-150U Function Block

3 K O M P R I M I F u n c t i o n B l o c k
3.1 Functional Description

4 S Y N C H R O N F u n c t i o n B l o c k
4.1 Functional Description
4.2 Parameters

5 F i r s t S t a r t a n d R e s t a r t

6 S a m p l e P r o g r a m w i t h t w o
B u b b l e M e m o r i e s

7 S o f t w a r e

1 General Considerations

1.1 Field of Application

The 513-M bubble memory in the SIMATIC S5-150U Programmable Controller (PC) offers the possibility to store data and programs.
The FB-BUB function block to be parameterised by the user controls data transfer (receiving/transmitting of data) between the programmable controller and the bubble memory.

The FB-BUB can communicate with a maximum of eight bubble memories.
The user can store various types of blocks (DB, SB, PB, FB, OB) on the bubble memory under a six-digit data record name (all ASCII characters are permitted).

The data of the SB, PB, FB and OB block types of the AG150U are transferred according to their respective current length. A partial transfer is only possible for data blocks.

Blocks which are not existing in the PC are installed before being transferred from bubble memory to PC.

After an update operation ('Overwriting' of an existing block) in the PC, the FB-BUB initiates compression and calls the KOMPRIMI program which compresses the PC memory. The data transfer of 1 K*byte takes up approx. 1 to 7 seconds.

The FB-BUB uses the following standard data handling blocks (have to be ordered separately; order no.: 6ES5844-0CA10):

FB 180	SEND	Send request
FB 181	RECEIVE	Receive request
FB 182	FETCH	Fetch request
FB 183	RESET	Reset of the interface

The FB 185 SYNCHRON must be called during startup (OB20/21/22).

1.2 Program Structure

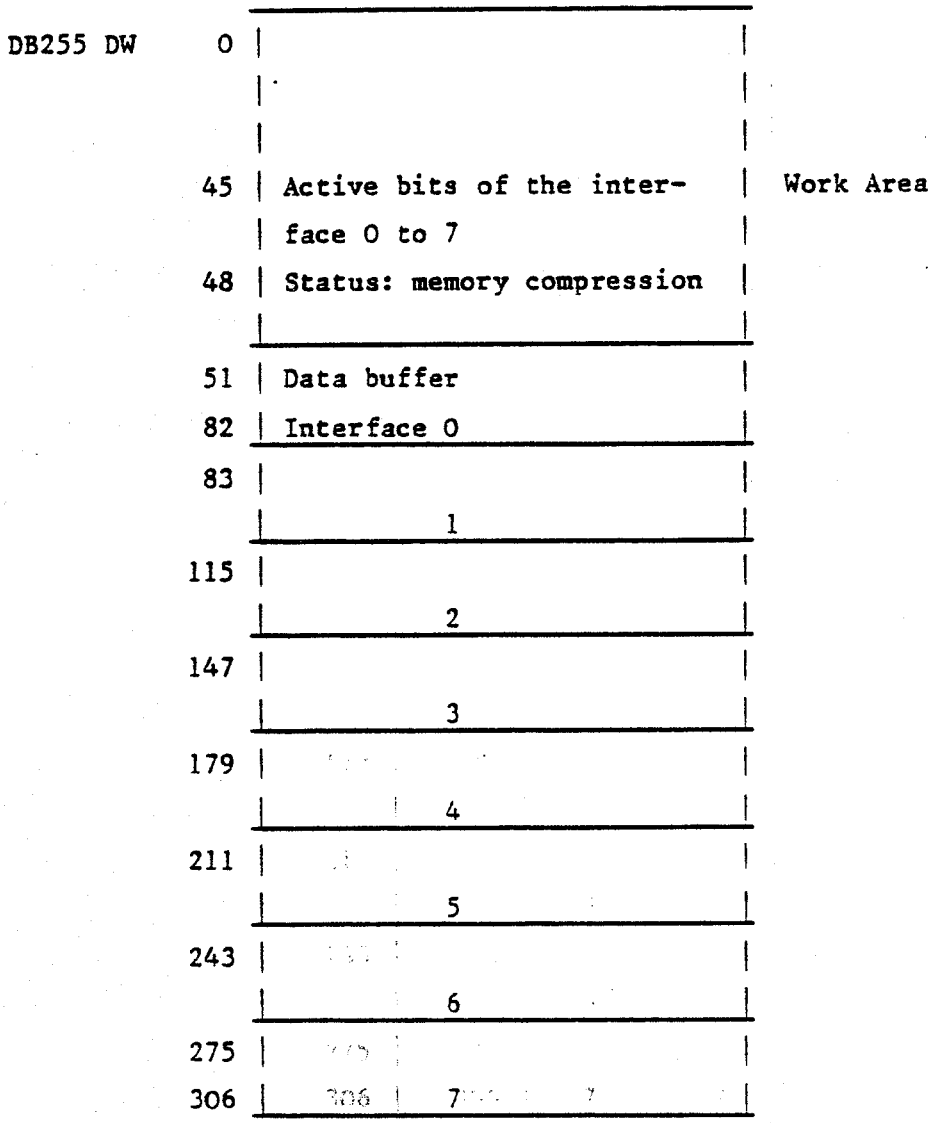
The FB-BUB is only executeable after the user has assigned parameters (interface and block size) to the SYNCHRON data handling block during startup (first start, restart, OB20/21/22) of the programmable controller (PC); and this is necessary as many times as there are interfaces (bubble memories).

The SYNCHRON data handling block is used to synchronise the PC with the bubble memory. The FB-BUB is called in the cyclic program section (e.g. OB1) where the paramterized interface number corresponds to the number assigned in the SYNCHRON.

The FB-BUB calls a PC memory compression program and the SEND, FETCH, RECEIVE and RESET standard data handling blocks.

1.3 Data Structure

The DB255 data block is used as sector name and data buffer. It mey be shortened according to the number of interfaces (bubble memories connected). Its minimum length is 82 data words.



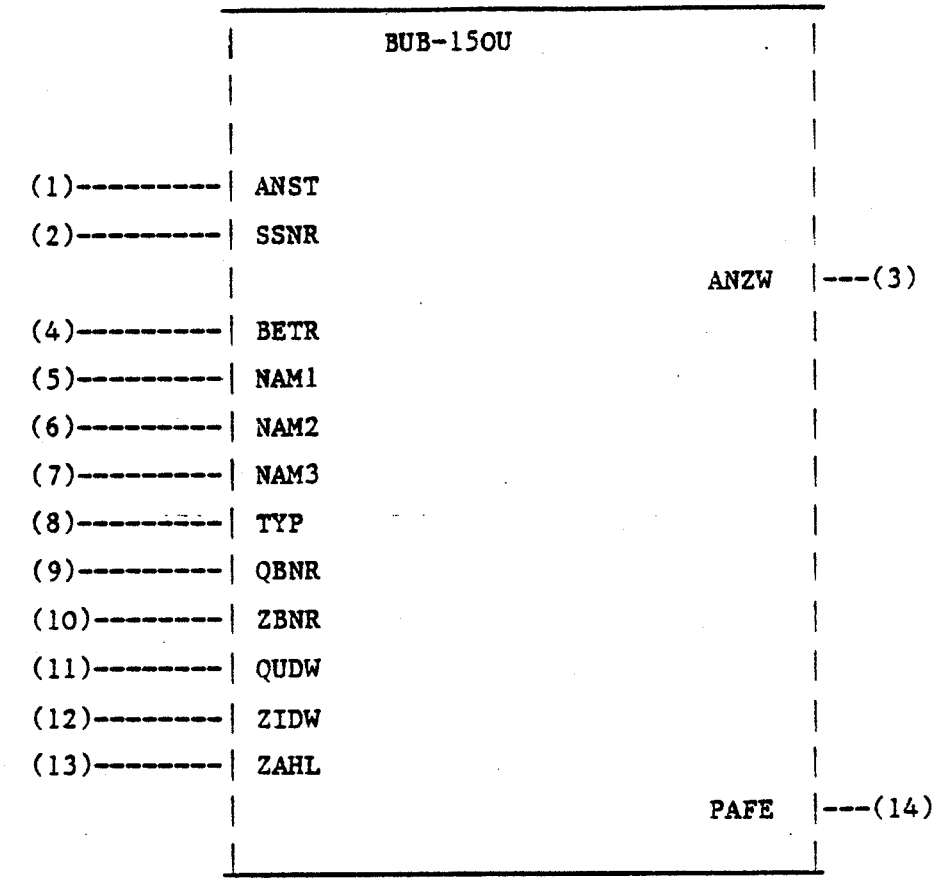
DW45 active bits of interface 0 to 7

Bit 0 - Interface 0 active
Bit 1 - Interface 1 active
Bit 2 - Interface 2 active
Bit 3 - Interface 3 active
Bit 4 - Interface 4 active
Bit 5 - Interface 5 active
Bit 6 - Interface 6 active
Bit 7 - Interface 7 active
Bits 8 to 15 free

DW48 Status Memory Compression

Bit 6 - Compression terminated without error
Bit 7 - Compression terminated in error
The remaining bits are used internally.

2 The BUB-150U Function Block (F B 1 9 9)



2.1 Functional Description

- The FB-BUB implements the data transfer of block types DB, SB, PB, FB and OB between the AG150U and the 513-M bubble memory. The FB-BUB can communicate with a maximum of 8 bubble memories.

- Loading blocks from the bubble memory into the programmable controller (PC) is even possible if the block to be loaded does not exist in the PC. The FB-BUB then installs this block in the length according to the block on the bubble, transfers the whole block, and enters the starting address into the block list of the PC.

- If the block to be loaded already exists in the PC, it is declared as invalid. The block to be loaded from the bubble is written to the PC to the last occupied memory address and the block list is updated. Before the transfer, data blocks are copied up to the destination address onto the target address, if it is >0. The PC memory is then compressed.

- The blocks of the SB, PB, FB and OB type are transferred in their current length.

- Data blocks can be transferred as a whole or in parts:

  - The length of the destination block is always the maximum of the length of the existing destination block (if not existing, length = 0) and the length of the block to be transferred plus an optional displacement (offset relative to start of destination block). Appropriate corrections are made in the block header.

  - During transfer, no undefined gaps may occur. For instance, a partial transfer to a non-existing block with an offset address larger than zero is not permitted (this would lead to an undefined "header gap"). It is also not permitted to transfer a block part with a displacement address larger than the length of the destination block (this would lead to an undefined "intermediate gap").

- The FB-BUB is neither re-entrant nor restart-capable since these restrictions apply to the standard data handling blocks used by FB-BUB.

Technical Data:

Block number	FB199
Block name	BUB-150U
Block length	1270
Call length	16
Processing time	3 to 20 ms
Nesting depth	2
FB called	FB 180 SEND
	FB 181 RECEIVE
	FB 182 FETCH
	FB 183 RESET
	FB 197 KOMPRIMI
DB called	DB 255 Bubble data buffer
	DB "0"
Flag words reserved	MW 200-220
(scratch flags)	MB 255
System data	BS 8.0
	BS 32-38
	BS 240 (scratchpad area)

2.2 Parameters

ANST = Initiation bit
SSNR = Interface number
ANZW = Flag word
BETR = Operating mode
NAM1 =) Name of data set (also sector name) on the
NAM2 =) bubble memory
NAM3 =)
TYP = Block type
B-NR = Number of Source block
ZBNR = Number of Destination block
QUDW = Source data word
ZIDW = Destination data word
ZAHL = Number of words to be transferred
PAFE = Parameter error

Direct and indirect parameter assignment

- With direct parameter assignment, the function block directly processes directly the parameters specified in the block call.
  The initiation bit (ANST) and the parameter error (PAFE) are only subjected to direct parameter assignment.

- With indirect parameter assignment, a pointer to a parameter field is transferred to the function block with the block parameter.
  The high byte (left byte) of the SSNR parameter is used as switching criterion for direct/indirect parameter assignment of SSNR, ANZW, BETR, NAM1, NAM2 and NAM3.

High byte of SSNR = 0: direct parameter assignment
High byte of SSNR <>0: indirect parameter assignment- the low byte (right byte) of SSNR points to the data word which contains the parameter assignment in the data block opened.

The indirect parameter assignment of source and destination data is performed with the parameter TYP = XX.
The QBNR parameter describes the data block number and the QUDW parameter describes the word address at which the TYP, QBNR, ZBNR, QUDW, ZIDW and ZAHL parameters are stored.

Parameter description

ANST The initiation bit is set by the user if data is to be transferred between PC and bubble memory.
ANST is reset by FB-BUB after the request has been executed. An intermediate reset by the user is not allowed, because otherwise the request can no longer be processed.
e.g. M 17.3

SSNR The interface number is the address set on the bubble memory for the select register or the Ident register.

The interface number must match the interface number parameterised at FB-SYNCHRON for this interface (bubble memory).

Permissible range: High Byte, Low Byte
0,0 0,7

High byte = 0: Direct parameter assignment,
low byte = SSNR

High byte <>0: Indirect parameter assignment,
low byte = DW pointer of SSNR, ANZW,
BETR, NAM1-3

Permissible range: Low byte: 0.....250

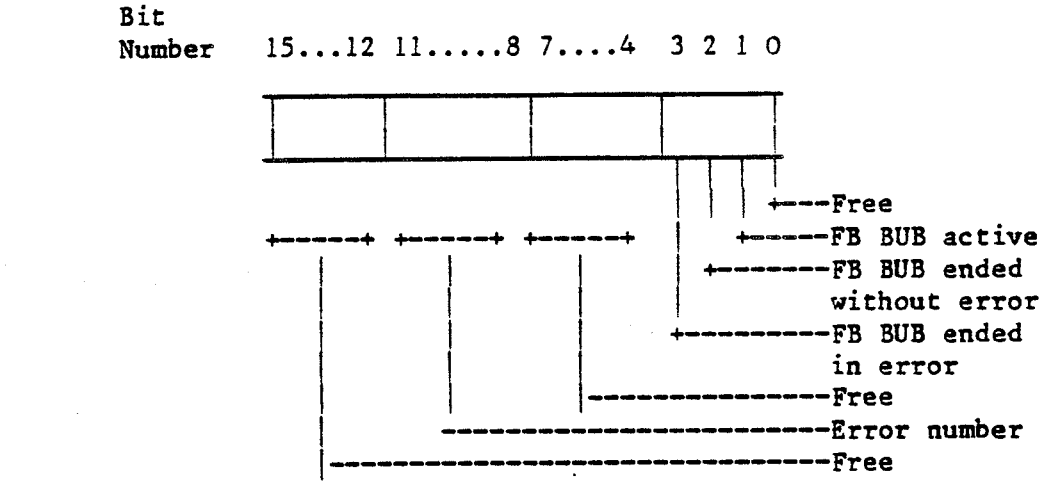
ANZW Flag word for status and error messages which are generated by the bubble memory firmware as well as by the FB-BUB and are provided to the user on termination or abortion of the request.

Permissible range: High byte, low byte
0,0 254,255

High byte = 0: Flag word is in the flag area and the low byte indicates the flag word address (MB 240 to 255 are used by the data handling blocks, so they have to be saved beforehand)

High byte = 0: High byte = Number of data block containing the flag word;
Low byte = data word address

Structure of the flag word:



Error number:

(1 to 6: errors recognised by the PC)

(7 to 12: errors recognised by the bubble memory)

0 = no error

1 = Parameter error of the FB-BUB (see PAFE)

2 = Overload (is set if the request could not be processed within 32.000 cycles)

3 = Block on bubble too short (intermediate gap, operating mode HS)

4 = Data word starting address > data block length, resp. block on bubble too short (MS)

5 = PC memory full

6 = Hardware error (module defective)

7 = Bubble memory not formatted

8 = Bubble memory full

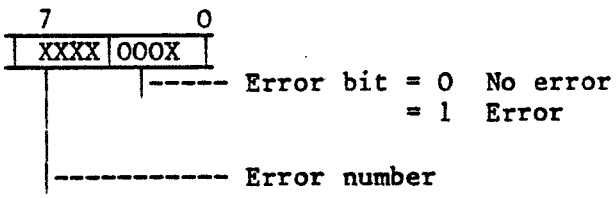
9 = Block not existing on bubble memory (operating mode MS)
Name of data set no existing on bubble

11 = Erroneous request

12 = Handshake error

PAFE Parameter errors are stored as error bit in the right tetrad and as error number in the left tetrad of a byte.

Permissible range: E80....EB127
AB0....AB127
MB0....MB255
(MB240 to 255 are used by the data handling blocks, so they have to be saved beforehand)



Error number

0 = No error

1 = DB for indirect parameter assignment not existing or too short

2 = Interface number > 7 or interface not existing

3 = DB/DW for the ANZW parameter not existing resp. too short

4 = BETR (Operating mode) <> HS/MS

5 = TYP is not DB, SB, PB, FB, OB, XX

6 = B-NR (block number) >255
<1
>254 for DB
<2 for DB

7 = block to be transferred not existing in the PC resp. too short (operating mode = HS)

8 = Q/Z-DW (source/destination DW) >2048
<0
resp. Z-DW <> if DB not on PC (MS), resp. block > 2048 DW (FB, OB, PB, SB)

9 = ZAHL (Number) >2048
=0
<-1 only for DB

10 = DB 255 too short, resp. not existing

PAFE must be evaluated after the FB-BUB call.

BETR Operating mode
KC = HS: Transfer from main memory of PC to bubble memory.
KC = MS: Transfer from bubble memory to main memory of PC.

NAM1 Characters 1-2 of the six-character name of the bubble data set (sector name) into/from which the block is to be transferred.
e.g. KC=RE

NAM2 Characters 3-4 of the six-character name of the bubble data set (sector name) into/from which the block is to be transferred.
e.g. KC=ZE

NAM3 Characters 5-6 of the six-character name of the bubble data set (sector name) into/from which the block is to be transferred.
e.g. KC=PT

The blocks are stored in the bubble memory under a data set name or sector name (NAM1, NAM2, NAM3). So blocks bearing the same number can be stored under as many data set names as desired.
The name of the data set is 6 characters long with ASCII characters. Anyhow the dataset name must exist on the bubble memory (see description of programmer for bubble)

TYP Block type to be transferred

KC=DB: Data block

KC=PB: Program block

KC=SB: Sequence block

KC=FB: Function block

KC=OB: Organisation block

Note: Reading out the OBs on the programmer (PG) is restricted to the OBs executable in the programmed controller (PC): OB1-50, 54, 63

KC=XX: Indirect parameter assignment of BYTP, QBNR, ZBNR, QUDW, ZIDW and ZAHL

QBNR Direct parameterization: Number of source block

Indirect parameterization (Typ:XX): Number of the data block containing the parameters.

($1 \leq$ DB number ≤ 254

$1 \leq$ SB/PB/FB number ≤ 255

$1 \leq$ OB number ≤ 63)

ZBNR Direct parametrization: Number of destination block

($1 \leq$ DB number ≤ 254

$1 \leq$ SP/PB/FB number ≤ 255

$1 \leq$ OB number ≤ 63)

QUDW Only relevant for data blocks: First word in the source block at which the data transfer shall be started (for direct parameterization) (0...2048).

For indirect parameterization: Data word address from which on the parameters are stored (0...250).

ZIDW Only relevant for data blocks: First word in the destination from which on the transfered data rewritten (0...2048).

ZAHL Only relevant for data blocks: Number of words to be transferred (1...2048).
If number = -1 (length of wild card) the actual data block length (1..2053 including header) of the source block is transferred from the data block header; the address of the source data word is assumed to 0.

2.3 Sample Program with the BUB-150U Function Block

1. By setting the initiation flag M1.7, the user transfers a request to the FB-BU-150U. Via interface 2 80 data words of the DB 99 starting at DW0 are to be transferred from the main memory of the PC to the bubble memory in DB 99 starting at DW 30. The flag word is MW8 and parameter errors are stored in MB20.
The module is stored under the ALBERT sector name on the bubble memory.

a) direct parameter assignment

```
OB1     •
        •
           :A DB 255
           :U M  100.0     condition for initiation of
                           FB-BUB
           :UN D 45.2      active bit interface 2
           :S  M  1.7
           :R  M 100.0
           :SPA FB 199
        NAME: BUB-150U
        ANST: M  1.7
        SSNR: KY 0,2
        ANZE: KY 0,8
        BETR: KC HS
        NAM1: KC AL
        NAM2: KC BE
        NAM3: KC RT
        TYP:  KC DB
        QBNR: KF 99
        ZBNR: KF 99
        QUDW: KF 0
        ZIDW: KF 30
        ZAHL: KF 80
        PAFE: MB 20
            •
            •
            : BE
```

b) indirect parameter assignment

The parameters ANST, SSNR, BETR, NAM1, NAM2, NAM3 are stored in DB 100 starting at DW 1.
The parameters TYP, QBNR, ZBNR, QUDW, ZIDW, ZAHL are stored in DB 100 starting at DW 7.

```
OB1          •
             •
             :A  DB 255
             :U  M  100      condition for initiation
                             of FB-BUB
             :UN D  45.2     active bit interface 2
             :S  M  1.7
             :R  M  100.0

             :A  DB 100      parameters in DB 100

             :SPA FB 199
            NAME:BUB-150U
            ANST:M 1.7
            SSNR:KY 255,1
            ANZW:KY 0,0
            BETR:KC
            NAM1:KC
            NAM2:KC
            NAM3:KC
            TYP :KC XX
            QBNR:KF +100
            ZBNR:KF 0
            QUDW:KF 7
            ZIDW:KF 0
            ZAHL:KF 0
            PAFE:MB 20
               •
               •
               • BE
```

```
DB 100
DW 1 : KY 0,2 SSNR
DW 2 : KY 0,8 ANZW
DW 3 : KC HS  BETR
DW 4 : KC AL  NAM1
DW 5 : KC BE  NAM2
DW 6 : KC RT  NAM3
DW 7 : KC DB  TYP
DW 8 : KF+99  QBNR
DW 9 : KF+99  ZBNR
DW 10: KF+0   QUDW
DW 11: KF+30  ZIDW
DW 12: KF+80  ZAHL
```

If a conditional call of the FB-BUB is made, care must be taken that if necessary the compression program is called again by the FB-BUB block after a request has been processed.

3 K O M P R I M I F u n c t i o n B l o c k (FB197)

3.1 Functional Description

If a block has been declared invalid after a transfer of data to the programmable controller (PC), it still occupies physical memory space.
To avoid a memory overflow, the FB-BUB calls the FB-KOMPRIMI to compress the PC memory (simulation of the KOMP function of the AS511).
Compressing the PC memory can last several cycles during which requests from FB-BUB cannot be processed.
The following information is provided to the user in DB 255, DW 48.

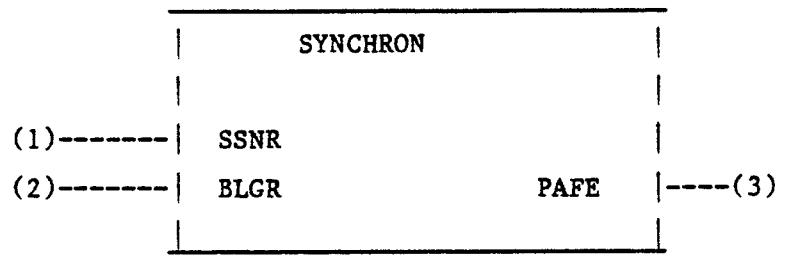
Bit 6 = Compression terminated without error
Bit 7 = Compression terminated in error

Note:
Compression by programmer function while the FB-KOMPRIMI is running internally leads to error.
Deleting blocks in the PC with the programmer function leads to error during compression.

Technical Data:

Block number	FB197
Block name	KOMPRIMI
Block length	201 words
Call length	2
Processing time	up to 0.4 ms
Nesting depth	1
Blocks called	
FBs called	-
DBS called	DB255
Reserved flag words	None
System data	BS5, BS6, BS33, BS40, BS48, BS49, BS50, BS51, BS120, BS121

4 S Y N C H R O N F u n c t i o n B l o c k



4.1 Functional Description

This block generates a synchronization between the programmed controller (PC) and the bubble memory (AS) on startup of the PC. Simultaneously the transfer area of the interface is deleted and preset, and the block size is determined by AS and PC. Runtimes of up to 10 seconds may occur for synchronization. So the synchron block may only be called in the startup branch (OB20, restart).

4.2 Parameters

The following block size may be set:

0: Standard value such that runtime of blocks less than 10 ms (without considering acknowledge times)
1: 16 Bytes
2: 32 Bytes
3: 64 Bytes
4: 128 Bytes
5: 256 Bytes
6: 512 Bytes
>6..254: Standard value like 0
255: Block size is fixedly set to 512 bytes

(high block size = high data throughput but high PC load,
small block size = low data throughput but small PC load).

PAFE: Error flag for parameter errors

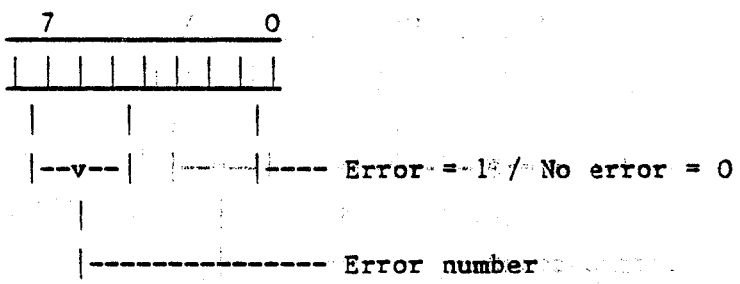
The "BYTE" indicated here (output, input, flag) is set if the block recognizes a parameter error.

e.g. Interface does not exist
wrong parameterization of
QTYP/ZTYP; QANF/ZANF; QLAE/ZLAE

programmer flag:	Output, byte
permissible	EBO ... EB 127
	ABO ... AB 127
area:	MBO ... MB 255

(MB240 to 255 are used by the data handling blocks, so they have to be saved beforehand)

Structure PAFE byte:



0 = No error
1 = Erroneous ORG format
2 = Area not existing
(DB not existing)
3 = Area too small
4 = Timeout error
5 = Erroneous flag word
6 = No source/destination parameter
for SEND/RECEIVE ALL
7 = Interface not existing
8 = Interface not ready
9 = Interface overload
AH = Interface reserved by other CPU
BH = A-NR not permitted
CH = Interface not acknowledged resp. not enabled
DH = Free
EH = Free
FH = Double call

5 First Start and Restart

The SYNCHRON function block must be parametrized in the OB20 block for each of the maximum 8 interfaces (bubble memories connected).

OB 20 and/or OB 21 and/or OB 22

```
  SPA  FB
NAME: SYNCHRON
SSNR: KY 0.0        0.0 ... 0,7
BLGR: KY 0.0        0.0 ... 0,6
PAFE: MB
```

Occurring error messages of the FB SYNCRHON must be evaluated by the user since the FB-BUB assumes that the synchronization has been carried out without any error.

A data transfer interrupted by a restart/power failure must be aborted by the user in OB 20 (and/or OB 21,22) as follows:

Delete data words in DB 255

DW31	clear internal status word	
DW49	clear internal status word	
DW45	Active bits of interfaces 0 to 7	
DW51	Status word Interface	0
DW83	•	1
DW115	•	2
DW147	•	3
DW179	•	4
DW211	•	5
DW243	•	6
DW275	Status word Interface	7

See example FB 253: BUB-N/W

```
FB253                LAE=41
NETWORK 1            FIRST START AND RESTART
NAME: BUB-N/W

:L     BS34
:L     KB255
:+F
:LIR   1
:T     MW200         ABS.ANF.ADR.DB 255
:L     KB0           (absolute start address)
: =F
:BEB                 DB 255 not existing
:A     DB 255
:L     KB0           Delete:
:T     DW31          clear status words
:T     DW49          clear status words
:T     DW45          Active bits interfaces 0 - 7
:T     DW51          Status interface 0
:T     DW83                         1
:T     DW115                        2
:T     DW147                        3
:T     DW179                        4
:T     DW211                        5
:T     DW243                        6
:L     MW200
:ADD   BF-1
:LIR   1
:L     KF+281
:<F
:BEB                 DW275 not existing
:L     KB0
:L     MW200
:ADD   KF+275        0 -> DW275: Status word interface 7
:TIR   3
:BE•
```

6 Sample Program with two Bubble Memories

2 bubble memories on interfaces 5 and 6.
For each interface the FB SYNCHRON has to be parameterized in the first start/restart-OB.

OB 20 and/or OB21 and/or OB 22

```
SPA FB 185
NAME : SYNCHRON
SSNR : KY 0,5
BLGR : KY 0,4
PAFE : MB 100
       •
       •   <-------- error evaluation of PAFE
       •
SPA FB 185
NAME : SYNCHRON
SSNR : KY 0,6
BLGR : KY 0,4
PAFE : MB 102
       •
       •   <-------- error evaluation of PAFE
       •
     : A DB 255
     : L KB 0
     : T DW 45     active bits of interfaces 0 to 7
     : T DW 211    status word interface 5
     : T DW 243    status word interface 6

     : BE
```

For each interface the FB-BUB is parameterized in OB 1.

```
OB 1
•
•
•
     : A DB 255        bubble memory DB
     : U  M 13.5       condition for initiation of FB-BUB,
                       interface 5
     : UN D 45.5       active bit interface 5
     : S  M  1.3       initiation bit for FB-BUB, interface 5
     : R  M 13.5
     : SPA FB 255
NAME : BUB-15OU
ANST : M1.3
SSNR : KY 0,5
ANZW : KY 0,8
BETR : KC HS
NAM1 : KC SE
NAM2 : KC KO
NAM3 : KC 03
TYP  : KC DB
QBNR : KF 99
ZBNR : KF 99
QUDW : KF 0
ZIDW : KF 30
ZAHL : KF 80
PAFE : MB 20
       •
       •
       •              error evaluation
       •
       •
     : U  M  40.6      condition for initiation of FB-BUB,
                       interface 6
     : UN D  45.6      active bit interface 6
     : S  M   1.4      initiation bit for FB-BUB, interface 6
     : R  M  40.6
```

```
      : SPA   FB 255
NAME  : BUB-150U
ANST  : M  1.4
SSNR  : KY 0,6
ANZW  : KY 0,1
BETR  : KC MS
NAM1  : KC SE
NAM2  : KC KO
NAM3  : KC 03
TYP   : KC DB
QBNR  : KF 123
ZBNR  : KF  99
QUDW  : KF   0
ZIDW  : KF  31
ZAHL  : KF  10
PAFE  : MB  21
        •
        •             error evaluation
        •
        •
      : BE
```

7 Software

Software for use of maximum 8 bubble memories 513-M in the 150 U programmed controller:

- OB 20 first start parameterized for 8 interfaces
- FB 199 BUB-150 U
- FB 197 KOMPRIMI
- FB 253 BUB-N/W sample first start/restart
- FB 185 SYNCHRON
- FB 183 RESET
- FB 182 FETCH
- FB 181 RECEIVE
- FB 180 Send

- DB 255 data buffer

SIMATIC S5

S5-115U Programmable Controller
Standard Function Modules
for the AG115U

Operating Manual Best.-Nr. E89120-F1944-U91

Table of contents

1 G e n e r a l

1.1 Field of Application

1.2 Program Structure

1.3 Data Structure

2 T h e B U B - 1 1 5 U F u n c t i o n B l o c k

2.1 Functional Description

2.2 Parameters
2.2.1 Direct and Indirect Parameterization
2.2.2 Parameterization of SSNR, ANZW, BETR, NAM1, NAM2, and NAM3
2.2.3 Parameterization of TYP, QU-B, QUDW, ZI-B, ZIDW, and ZAHL
2.2.4 Parameter Description

2.3 Program Example for FB BUB-115U
2.3.1 Direct Parameterizing
2.3.2 Indirect Parameterizing
2.3.3 Cold Start and Restart

3 The SYNCHRON Function Block

3.1 Functional Description

3.2 Parameters

3.3 Parameter Description

4 The SEND Function Block

4.1 Functional Description

4.2 Parameters

4.3 Parameter Description

5 Software

6 Technical Data

1 General

1.1 Field of Application

The AS513-M bubble memory in the SIMATIC S5-135U Programmable Controller (PC) provides the facility for storing data.

The BUB-115U function block to be parameterized by the user controls the data transfer (receipt/transmission of data) between the programmable controller and the bubble memory.

The FB-BUB-115U can communicate with a maximum of eight bubble memories.

The user can store data blocks (DB) on the bubble memory under a six-digit name (ASCII characters are permitted).

The time required to transfer 1k bytes of data (512 data words) is approx. 5 to 23 seconds (with a constant cycle time of 100 ms) according to the block size and the operating mode used:

BLGR	BETR = MS NUMBER = 512	BETR = HS NUMBER = 512	
1 (16 bytes)	approx. 11 s	approx. 23 s	Data block does not exist on bubble
		approx. 13 s	Extend data block on bubble memory
		approx. 10 s	Overwrite data block on bubble
6 (512 bytes)	approx. 5 s	approx. 17 s	Data block does not exist on bubble
		approx. 6.5 s	Extend data block on bubble memory
		approx. 4 s	Overwrite data block on bubble

Figure 1: Data transfer intervals of the FB BUB-115U

The FB-BUB-115U uses the following standard data handling blocks for the AG 115U:

FB	244	SEND	Send request
FB	245	RECEIVE	Fetch request
FB	246	FETCH	Reset request
FB	247	CONTROL	Receive request

Moreover, it is necessary to use the FB 249 SYNCHRON which must be called during cold start or restart.

1.2 Program Structure

The FB-BUB-115U can only be executed after the user has assigned parameters to the SYNCHRON block when starting (OB 21, OB22) the programmable controller (PC) and after this block has been processed without errors.

The SYNCHRON handling data block must be called with the appropriate parameters for each interface existing in the PC (bubble memory module).

The synchronization of the interface must be performed without error to allow proper transfer between the SEND, FETCH, RECEIVE and CONTROL handling data blocks and the interface.

The FB-BUB block is called in the cyclic program section (OB1) and the interface number must match the number mentioned in the SYNCHRON call.

The FB-BUB-115U can be assigned parameters either directly or indirectly.

Direct assignment of the parameters is carried out by entering them directly in the BUB-115U function block.

Indirect assignment is performed through a pointer to a parameter field which is transferred together with the block parameters (see description of the block parameters).

Indirect parameter assignment allows the user to initialize the FB-BUB-115U once.

The parameters are then changed accordingly in a data block (containing an image of the parameters).

The FB-BUB-115U is initialized via an initiation bit if this bit is set by the user. To ensure proper servicing of the request, the initiation

bit should not be reset by the user during request servicing.

It will then take several cycles to the FB-BUB-115U to perform total transfer of the data.

The parameters of the FB-BUB-115U block must not be changed during request servicing; i.e. it will only be possible to initiate a new request if the previous request has ended successfully for the relevant interface (bubble memory module) or if it has been interrupted by an error.

Once the request has been serviced or if it has been interrupted, the initiation bit is reset automatically by the FB-BUB-115U.

1.3 Data Structure

The DB255 data block is reserved for communication purposes and is used as sector name and data buffer.

According to the number of interfaces (bubble memories connected) and to the interface number specified, the DB255 block contains for each interface, a data buffer in addition to a work area.

Its minimum length is 66 data words (DWO to DW65) for one interface.

Its maximum length is 283 data words (DWO to DW282) for eight interfaces.

DB 255

DW 0	Operating range
DW 34	
DW 35	Data buffer
DW 65	Interface 0
DW 66	Data buffer
DW 96	Interface 1
DW 97	Data buffer
DW 127	Interface 2
DW 128	Data buffer
DW 158	Interface 3
DW 159	Data buffer
DW 189	Interface 4
DW 190	Data buffer
DW 220	Interface 5
DW 221	Data buffer
DW 251	Interface 6
DW 252	Data buffer
DW 282	Interface 7

Figure 2: Data structure of the DB 255

2 The BUB-115U Function Block

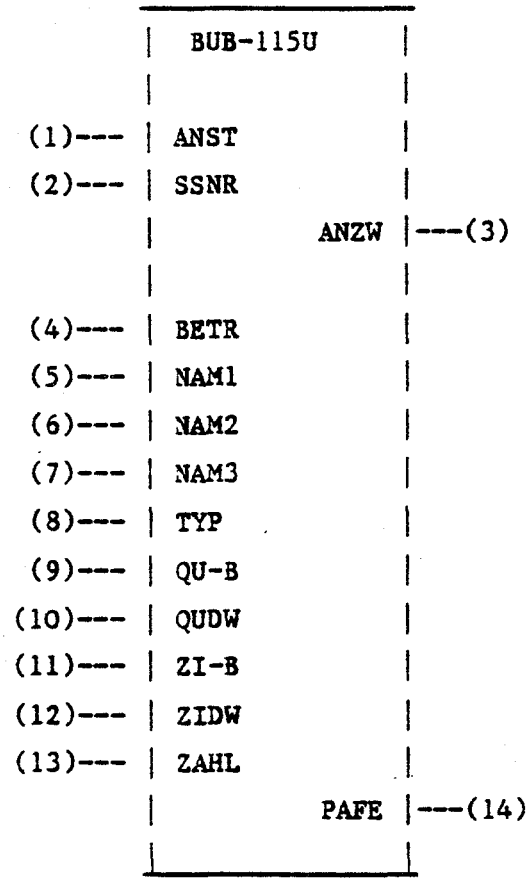


Figure 3: Function block FB BUB-115U

2.1 Functional Description

The FB BUB-115U block is responsible for transfering DBs between the AG 115U and the AS513-M bubble memory. The FB BUB-115U can communicate with a maximum of 8 bubble memories.

- Loading blocks from the bubble memory into the programmable controller (PC) is only possible if the data block to be loaded (destination data block) is already existing on the memory of the PC.

- Writing data blocks from the PC to the bubble memory is also possible if the data block to be written (destination block) does not exist on the bubble memory:

  It is then created in the bubble memory according to the appropriate length. Prerequisite: the offset address (ZIDW parameter) must be zero.

- Data blocks can be transferred in whole or in part:

- The following rules must be observed:

  - The source and destination data block numbers may be different.

  - The length of the destination data block is calculated according to the maximum length of the existing destination block (if it does not exist: length = 0) and to the length of the block to be transferred plus an offset address which can be specified (offset relative to the beginning of the destination data block). Appropriate corrections are made in the block header (writing a DB into the bubble memory).
    These conventions apply to a transfer of data from the PC memory to the bubble memory.

  - During transfer, no free space should be left undefined. For instance, a partial transfer to a non-existing block is not

permitted if the offset address is greather than zero (this would lead to an undefined "header gap").
In the same way, it is not permitted to transfer a block with an offset address that is greater than the length of the destination block (this would lead to an undefined "intermediate gap").

- The maximum length of a data block from/to which data can be transferred is 4096 data words (DW0 to DW4095).

- The maximum number of data words which can be transferred is 2048 data words.

2.2 Parameters

ANST	=	Initiation bit
SSNR	=	Interface data block
ANZW	=	Flag word
BETR	=	Operating mode
NAM1	=)	Name of data set (also sector) on the bubble memory
NAM2	=)	
NAM3	=)	
TYP	=	Block type
QU-B	=	Source data block number
QUDW	=	Source data word number
ZI-B	=	Destination data block number
ZIDW	=	Destination data word number
ZAHL	=	Number of words to be transferred
PAFE	=	Parameter error byte

2.2.1 Direct and Indirect Parameterization

The FB BUB-155U Conception includes direct and indirect assignment of the block parameters.

Only the initiation bit (ANST) and the parameter error byte (PAFE) have to be entered directly.

In the case of the direct parameter assignment, the parameters specified in the block call will be directly processed by the function block.

In the case of the indirect parameter assignment, a pointer to a parameter field is transferred to the function block with the block parameters.
The "actual" parameters are arranged without any gap in the same order as for the direct parameter assignment.

2.2.2 Parameterization of SSNR, ANZW, BETR, NAM1, NAM2 and NAM3

The high byte (left byte) of the SSDB parameter is used as switching criterion for direct/indirect parameter assignment.

High byte of SSNR = 0: direct parameter assignment:
SSNR, ANZW, BETR, NAM1, NAM2 and NAM3
are parametrized directly

High byte of SSNR <> 0: indirect parameter assignment,
SSNR, ANZW, BETR, NAM1, NAM2, and NAM3
are stored in the data block specified
in this byte starting at the data word
indicated in the low byte of SSNR.

The ANZW, BETR, NAM1, NAM2 and NAM3 block parameters are irrelevant in the indirect parameter assignment.

2.2.3 Parameterization of TYP, QU-B QUDW, ZI-B, ZIDW, and ZAHL

In direct parameter assignment, the FB BUB-115U processes directly the source and destination parameters specified in the block call; in indirect parameter assignment, the block parameters build a pointer to a parameter field in a data block that contains the "current" source and destination parameters.

Parameter sequence and data format are similar to those used in direct parameter assignment.

Indirect parameter assignment of the source and destination parameters is achieved with parameter TYP = "XX" (data format KC).
The QU-B parameter gives the data block number and QUDW the word number at which the TYP, QU-B, QUDW, ZI-B, ZIDW and ZAHL parameters are stored.

The block parameters ZI-B, ZIDW and ZAHL are irrelevant for indirect parameter assignment.

2.2.4 Parameter Description

*) ANST: Initiation bit

The initiation bit (ANST) is set by the user if data is to be transferred from the PC to the bubble memory.

This bit must be set by the user.
It is reset by the FB BUB-115U block after the request has been serviced.

Kind/type of parameter: Input/bit

Possible range: E 0.0 ... E 127.7
A 0.0 ... A 127.7
M 0.0 ... M 255.7

Admissible range: A 0.0 ... A 127.7 (if available) on the CPU 942/943
A 0.0 ... A 63.7 (if available) on the CPU 941
M 0.0 ... M 255.7

*) SSNR: Interface number

This number is the number of the interface used to address the bubble memory module.

The interface number must match the interface number which has been parametrized for this interface (bubble memory module) at the SYNCHRON data handling block.

Kind of parameter/Type Data / KY

Possible range: 0,0 ... 255,255

Admissible range: 0,0 ...0,7

High byte = 0: direct parameter assignment: low byte = SSNR

High byte <> 0: indirect parameter assignment,
High byte = Data block number; the parameters are stored in this DB
Low byte = Pointer to the data word at which the parameters are stored

* ANZW: Address of condition code word

Address of the condition code word containing the flags indicating that the request servicing is carried out by the FB BUB-115U.

It contains the status and error messages which are generated by the bubble memory firmware and by the FB BUB-115U, and which are available to the user once the request has been terminated or interrupted.

Kind of parameter/Type: Data / KY

Possible range: 0,0 ... 255,255

Admissible range: 0,0 ... 0,254 (flag area)
2,0 ... 254,255 (data block)

The user can place the condition code word (16-bits long) either into the flag area or into the data block.

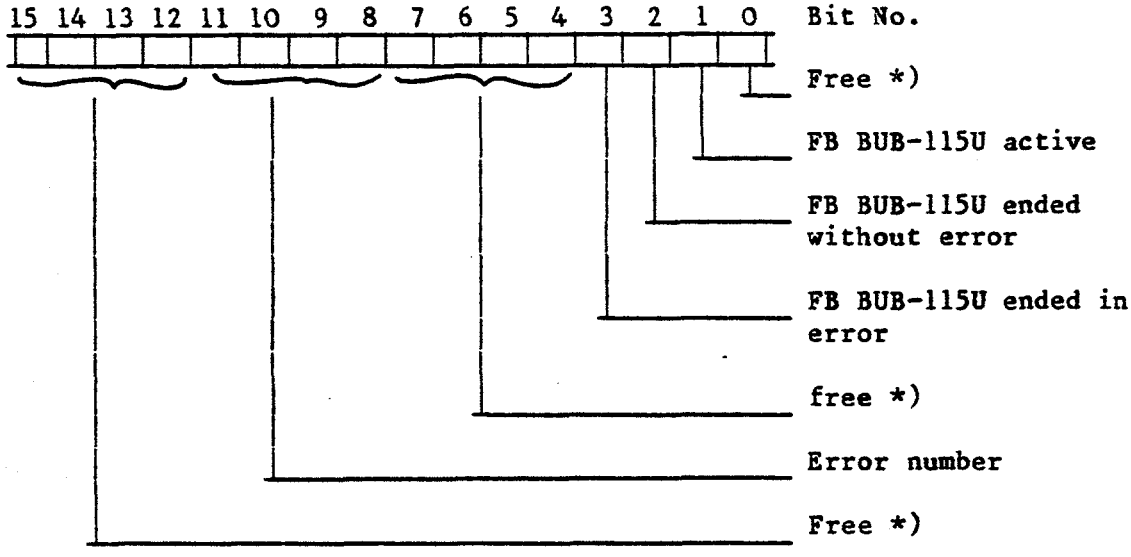
The high byte (left byte) of the ANZW parameter is the switching criterion:

High byte = 0 : Condition code word is located in the flag area and the low byte (right byte) indicates the flag word address of the condition code word

High byte <> 0 : Condition code word is located in the data block and the high byte contains the number of this data

block; the low byte contains the data word address of the condition code word

Structure of the flag word.



Note: *) These bits are cleared by the FB-BUB-115U block when the condition code word is written.

If the FB BUB-115U block issues the message "Request ended in error" in bit 3 of the flag word, the tetrad bit 8 to bit 11 contains an error number which gives information on the cause of the interrupt.

Error numbers in the flag word:

(1 to 5: errors detected by the programmable controller)
(6 to CH: errors detected by the bubble memory)

0 = no error

1 = Parameter error (for more information, see PAFE)

2 = No. not used

3 = Operating mode HS --> MS: Data block does not exist on bubble;
danger of "header gap"

4 = Operating mode MS --> HS: Data block on bubble too short
Operating mode HS --> MS: Data block on bubble too short
Danger of "intermediate gap"

5 = ZAHL = "Joker length": Source data block (QU-B) too long:
Data block without block header
greater than 2048 data words

6 = Hardware error

7 = Bubble memory not formatted

8 = Bubble memory full

9 = Operating mode MS --> HS: Data block or name of data set does
not exist on bubble

AH = No. not used

BH = Erroneous request

CH = Handshake error

* BETR: Operating mode

There are two operating modes:

HS --> MS: Transfer from the main memory of the PC (HS) to the bubble memory (MS)

MS --> HS: Transfer from the bubble memory (MS) to the main memory of the PC (HS)

Kind of parameter/Type: Data / KC

Possible range: All ASCII characters

Admissible range: HS (HS--> MS)
MS (MS--> HS)

* NAM1, NAM2, NAM3: Name of data set, sector name

The data blocks are stored on the bubble under a data set name (also sector name) comprising NAM1, NAM2 and NAM3.

Hence, data blocks can be stored under as many data block names as desired.

The data set name consists of 6-digit ASCII characters.

* NAM1: Name part 1

Characters 1 and 2 of the six-digit name of the data block (also sector name) into/from which the block is to be transferred.

Kind of parameter/Type: Data / KC

Possible range: All ASCII characters

Admissible range: All ASCII characters

* NAM2: Name part 2

Characters 3 and 4 of the six-digit name of the bubble data set (also sector name) into/from which the block is to be transferred.

Kind of parameter/Type: Data / KC

Possible range: All ASCII characters

Admissible range: All ASCII characters

* NAM3: Name part 3

Characters 5 and 6 of the six-digit name of the bubble data set (also sector name) into/from which the block is to be transferred.

Kind of parameter/Type: Data / KC

Possible range: All ASCII characters

Admissible range: All ASCII characters

* TYP: Block type

This parameter is the switching criterion used for switching between direct and indirect parameter assignment of TYP, QU-B, QUDW, ZI-B, ZIDW and ANZ:

If TYP contains the value KC="DB", the parameters are assigned directly.

If TYP contains the value KC="XX", the parameters are assigned indirectly.

Kind of parameter/Type: Data / KC

Possible range: All ASCII characters

Admissible range: DB: direct parameter assignment
XX: indirect parameter assignment

* QU-B: Number of source data block

In direct parameter assignment, this parameter contains the number of the source data block from which the data is transferred; in indirect parameter assignment, it contains the number of the data block which contains the parameter list (TYP = XX).

Kind of parameter/Type: Data / KF

Possible range: -32768 ... +32767

Admissible range: +1 ... + 254

* QUDW: Number of the source data word

First word in the source data block which is transferred (for direct parameter assignment); or data word address at which the parameter list is stored (for indirect parameter assignment).

Kind of parameter/Type: Data / KF

Possible range: -32768 ...+32767

Admissible range: +0 ... +2048

* ZI-B: Number of the destination block

This parameter contains the number of the source data block to which the data is transferred (for direct parameter assignment); it is irrelevant in the case of the indirect parameter assignment (TYP = XX).

Kind of parameter/Type: Data / KF

Possible range: -32768 ... + 32767

Admissible range: +1 ... +254

* ZIDW: Number of the destination word

First word in the destination block to which data is transferred (for direct parameter assignment); this parameter is irrelevant in case of indirect parameter assignment

Kind of parameter/Type: Data / KF

Possible range: -32768 ... +32767

Admissible range: +0 ... +2048

* ZAHL: Number of data words to be transferred

This parameter contains the number of words to be transferred (for direct parameter assignment).
If the so-called "wildcard length" is used (ZAHL = -1), the actual data block length of the source block is transferred from the data block header; the QUDW parameter is irrelevant, i.e. it is set to zero by the FB BUB-115U block. The transfer begins at DW 0 in the source data address.
This parameter is irrelevant with indirect parameter assignment.

Kind of parameter/Type: Data / KF

Possible range: -32768 ... +32767

Admissible range: +1 ... +2048
-1 (wildcard length)

* PAFE: Parameter error byte

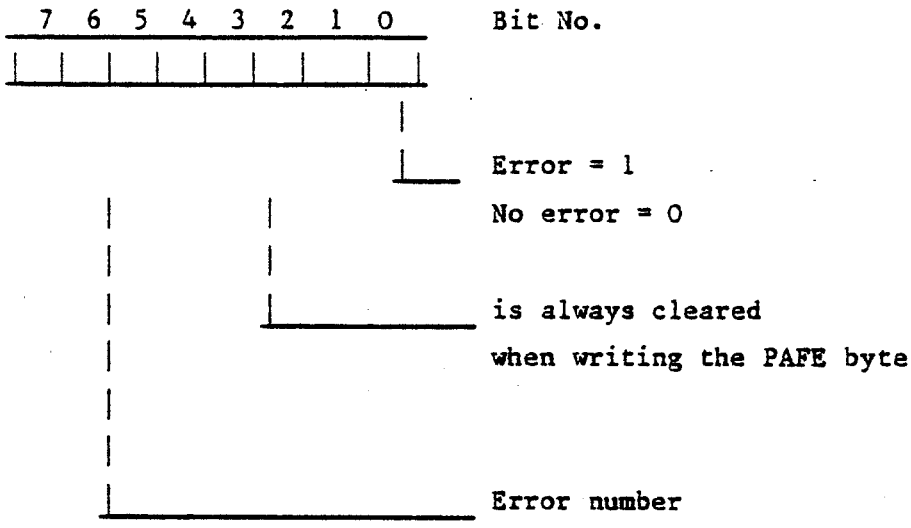
The byte indicated here (flag byte, output byte, input byte) is set when the FB BUB-115U block recognises a parameter error.

Kind of parameter/Type: Output / BY

Possible range: EB 0 ... EB 127
AB 0 ... AB 127
MB 0 ... MB 255

Admissible range: EB 0 ... EB 127 (if available) CPU942/943
AB 0 ... AB 127 (if available) CPU942/943
EB 0 ... EB 63 (if available) CPU941
AB 0 ... AB 63 (if available) CPU941
MB 0 ... MB 255

Structure of the PAFE byte:



The following error numbers may occur in the PAFE byte:

Figure 5: Structure of the PAFE byte

The following error numbers might appear in the PAFE byte:

0 = No error

1 = Data block for indirect parameter assignment does not exist or is too short; DB 255 used resp. non-admissible range

2 = Interface number (SSNR) > 7 or interface SSNR does not exist

3 = Data block for ANZW parameter does not exist; DB 255 used resp. non-admissible range

4 = Wrong operating mode (BETR):
BETR ≠ 'HS' BETR ≠ 'MS'

5 = Wrong block type (TYP):
TYP ≠ 'DB'; TYP ≠ 'XX'

6 = Block number (QU-B, ZI-B) not admitted:
BETR = HS --> MS: QU-B < 1; QU-B > 254;
ZI-B < 1; ZI-B > 254
BETR = MS --> HS: QU-B < 1; QU-B > 254
ZI-B < 1; ZI-B > 254

7 = Data block does not exist in the programmable controller or it is too small:
BETR = HS --> MS: QU-B; QUDW > DB length;
QUDW + ZAHL > DB length
BETR = MS --> HS: ZI-B; ZIDW > DB length;
ZIDW + ZAHL > DB length

8 = Data word number (QUDW, ZIDW) not admitted:
QUDW/ZIDW < 0; QUDW/ZIDW > 2048

9 = Number (ZAHL) not admitted:
ZAHL < = 0; ZAHL ≠ 1; ZAHL > 2048

AH = Data handling block DB 255 does not exist or is too small

2.3 Program Example for FB BUB-115U

By setting the initiation flag M100.0, the user issues a request to the FB BUB-115U.

100 data words starting at DW 15 in the DB 35 data block are to be transferred from the main memory of the PC to the bubble memory in DB 40, starting at DW 30 and stored under the name "DATUM1".

The interface number 2 has been parametrized for the bubble memory.

The condition code word is stored in flag word MW 180.

Parameter errors are stored in flag byte MB 50.

2.3.1 Direct Parameterizing

```
      OB 1
       •
       •
       •
         :U      M  90.0     Flag for initiation activation
         :UN     M 100.0     Initiation flag
         :R      M  90.0     Reset activation flag
         :S      M 100.0     Set initiation flag
         :U      M 100.0     Initiation?
         :SPB    FB199       Call of FB BUB-115U
   NAME :BUB-115U
   ANST :        M 100.0     Initiation bit: M 100.0
   SSNR :        KY0,2       Interface number: 2
   ANZW :        KY0,180     Condition code word address: MW 180
   BETR :        KCHS        Operating mode: HS --> MS
   NAM1 :        KCDA  }
   NAM2 :        KCTU  }     Sector name: DATUM1
   NAM3 :        KCM1  }
   TYP  :        KCDB        Block type : DB
   QU-B :        KF+35       Source data block number: DB 35
   QUDW :        KF+15       Source data block number: DW 15
   ZI-B :        KF+40       Destination data block number: DB 40
   ZIDW :        KF+30       Destination data word number: DW 30
   ZAHL :        KF+100      Number: 100 data words
   PAFE :        MB50        PAFE byte: MB 50
    .
    .
    .
```

The initiation bit (M 100.0) is reset by the FB BUB-115U once the request is completed.

2.3.2 Indirect Parameterizing

The SSNR, ANZW, BETR, NAM1, NAM2, NAM3 parameters are stored in DB 101, starting at DW 10.

The TYP, QU-B, QUDW, ZI-B, ZIDW and ZAHL parameters are stored in DB 110, starting at DW 1.

```
      OB 1
       •
       •
       •
         :U      M  90.0     Flag for initiation activation
         :UN     M 100.0     Initiation flag
         :R      M  90.0     Reset activation flag
         :S      M 100.0     Set initiation flag
         :U      M 100.0     Initiation?
         :SPB    FB199       Call of FB BUB-115U
NAME     :BUB-115U
ANST     :       M 100.0     Initiation bit: M 100.0
SSNR     :       KY101,10    Parameter are in DB 101 at DW 10
ANZW     :       KY0,0       (left parameter byte >< 0 -->
BETR     :       KC          indirect parameter assignment
                             ≙ DB number
NAM1     :       KC          Irrelevant parameters
NAM2     :       KC
NAM3     :       KC
TYP      :       KCXX        Indirect parameter assignment
QU-B     :       KF+110      Parameter DB: DB 110
QUDW     :       KF+1        Parameter list begins at DW 1
ZI-B     :       KF+0        Irrelevant
ZIDW     :       KF+0        Irrelevant
ZAHL     :       KF+0        Irrelevant
PAFE     :       MB50        PAFE byte: MB 50
       .
       .
       .
```

The initiation bit (M 100.0) is reset by the FB BUB-115U once the request is serviced.

```
Parameter list SSNR, ANZW, BETR, NAM1, NAM2, NAM3

DB 101

DW 10: KY = 0.2      Interface number: SSNR = 2
   11: KY = 0.180    Condition code word address MW 180
   12: KC = HS       Operating mode= HS --> MS
   13: KC = DA  }
   14: KC = TU  }    Sector name: DATUM1
   15: KC = M1  }

Parameter list TYP, QU-B, QUDW, ZI-B, ZIDW, ZAHL

DB 110

DW 1 : KC = DB       Block type = DB
   2 : KF = +35      Source data block: DB 35
   3:  KF = +15      Source data word:  DW 15
   4 : KF = +40      Destination data block: DB 40
   5 : KF = +30      Destination data word: DW 30
   6 : KF = +100     Number = 100 data words
```

2.3.3 Cold Start and Restart

During cold start and restart, the SYNCHRON function block must be called for each available interface (bubble memory).

Error messages of the SYNCHRON function block must be evaluated by the user since a correct synchronization of the interface is required by the FB BUB-115U block.

After the interface has been synchronized, a system-specific SEND request must be transmitted to the AS 513 interface.

Moreover, the internal status word of the FB-BUB-115U block must be deleted for each interface.

```
    OB 21; OB22:
     .
     .
     .
     : SPA FB 249          Call of the handling block
NAME : SYNCHRON            SYNCHRON
SSNR : KY0,2               Interface number : 2
BLGR : KY0,4               Block size: e.g. : 4 (128 bytes)
PAFE : MB255               Parameter error byte: MB 255
     :
     :
     :A       DB255        Data buffer for the FB BUB-115U
     :L       KB0
     :T       DW 1         Internal status flag
     :T       DW 31        Internal status flag
     :T       DW 35        Delete status word interface 0
     :T       DW 66        Delete status word interface 1
     :T       DW 97        Delete status word interface 2
     :T       DW 128       Delete status word interface 3
     :T       DW 159       Delete status word interface 4
     :T       DW 190       Delete status word interface 5
     :T       DW 221       Delete status word interface 6
     :T       DW 252       Delete status word interface 7
     :
     :
     :SPA FB244            Call of the SEND handling block with start
NAME :SEND                 request for the AS 513 (bubble memory module)
SSNR :KY0,2                Interface number: 2 (corresponds to that of
                           the FB SYNCHRON)
A-NR :KY 0,10              Request number: 10 (fix)
ANZW :MW 250               Condition code word: e.g. MW 250
QTYP :KC NN                Source type (fix)
DBNR :KY 0,0               Irrelevant
```

```
QANF :KF 0              Irrelevant
QLAE :KF 0              Irrelevant
PAFE :MB 255            Parameter error byte: e.g. MB 255
```

3 The SYNCHRON Function Block

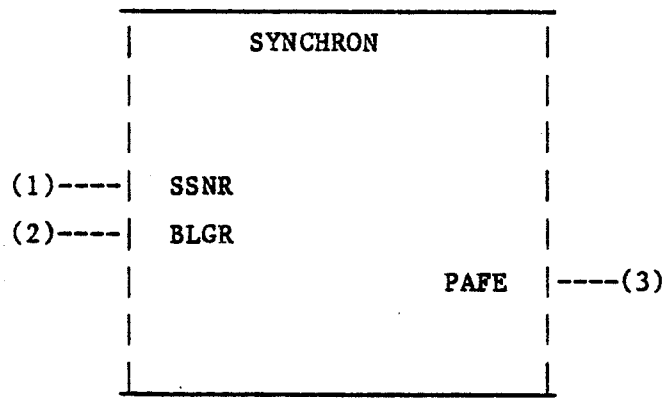


Figure 6: Function block FB SYNCHRON

3.1 Functional Description

The SYNCHRON function block (FB 249) is responsible for synchronizing the central module and the interface (bubble memory).

The interface between the central module and the interface is deleted or preset.

The interface must be synchronized during cold start in the stop state (OB 21) and after "power on" (OB 22).

This handling block can only be called at startup time, i.e. during cold start or restart.

3.2 Parameters

SSNR = Interface number
BLGR = Block size
PAFE = Parameter error byte

3.3 Parameter Description

*) SSNR: Interface number

Number of the interface used to access the bubble memory.

Kind of parameter/Type Data / KY

Possible range: 0,0 ... 255, 255

Admissible range: 0,0 ...0,7

High byte = 0: Direct parameter assignment: Low byte = SSNR

High byte <> 0: Indirect parameter assignment:
Low byte = point to parameter field;
The parameters SSNR und BLGR are located in the data block opened which has been opened before calling the FB SYNCHRON in the same way (sequence, format) as for the direct parameter assignment.

* BLGR: Block size

On system startup, the block size is determined by the FB SYNCHRON block.
This size can be affected by the BLGR parameter.

To determine the block size, the SYNCHRON block transfers a "desired" block size to the interface, according to the BLGR parameter. This size is then checked and changed, if necessary.

The "resulting" block size constitutes the upper limit for the SEND and RECEIVE blocks; it limits the maximum (net) number of data bytes to be received/sent for each block call.

If the area to be transferred is longer than this block, subsequent blocks are then transferred.

To obtain a high throughput, select a greater block size (load of programmable controller cycle).

To reduce the runtime of the function blocks, set a small block size (low data throughput).

Kind of parameter/Type: Data / KY

Possible range: 0.0 ... 0,255

Admissible range: 0,0 ... 0,255

The block size can be set as follows:

Parameter value/BLGR	Meaning	
0	Standard value (64 bytes)	block size wanted
1	16 bytes	
2	32 bytes	
3	64 bytes	
4	128 bytes	
5	256 bytes	
6	512 bytes	
7...254	Standard value (64 bytes)	
255	Block size set to 512 bytes	

Fig. 7: Block size of the FB SYNCHRON

* PAFE: Parameter error byte

The byte indicated here (flag byte, output byte, input byte) is set when the FB BUB-115U block detects a parameter error.

Kind of parameter/Type: Output / BY

Possible range:
EB 0 ... EB 127
AB 0 ... AB 127
MB 0 ... MB 255

Admissible range:
EB 0 ... EB 127 (if available) for the CPU942/943
AB 0 ... AB 127 (if available) for the CPU942/943
EB 0 ... EB 63 (if available) for the CPU941
AB 0 ... AB 63 (if available) for the CPU941
MB 0 ... MB 255

Structure of the PAFE byte:

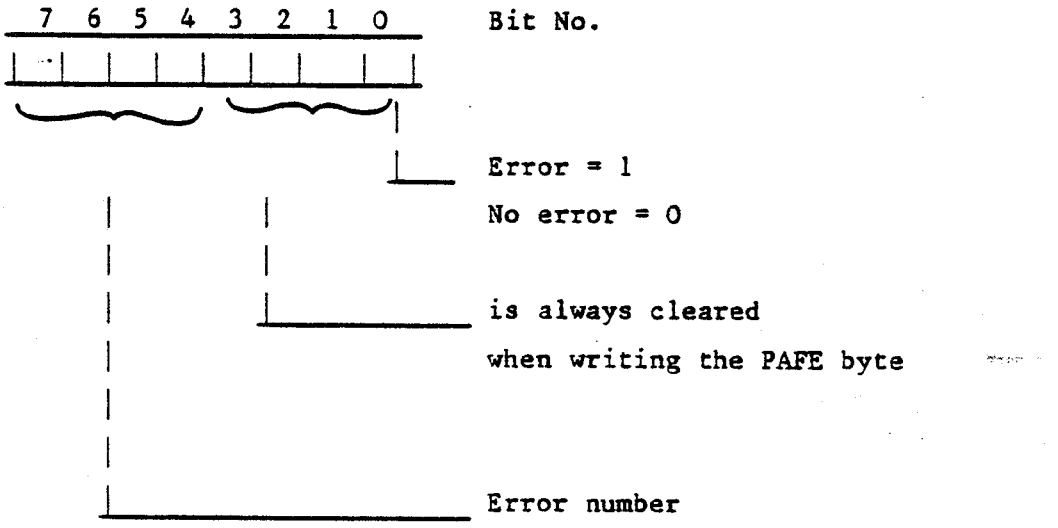


Fig. 8: Structure of the PAFE byte

The following error numbers may occur in the PAFE byte:

0 = No error

1 to 4 Erroneous source/destination parameters

1 = Wrong ORG format

2 = range doesn't exist (DB doesn't exist)

3 = Range too small

4 = Timeout

5 = Erroneous condition code word

6 = No source/destination parameters for SEND/RECEIVE-ALL

7 = Interface does not exist

8 = Interface not ready

9 = Interface overloaded

AH = Free

BH = Request number is not permitted

CH = Interface not acknowledged or enabled

4 The SEND Function Block

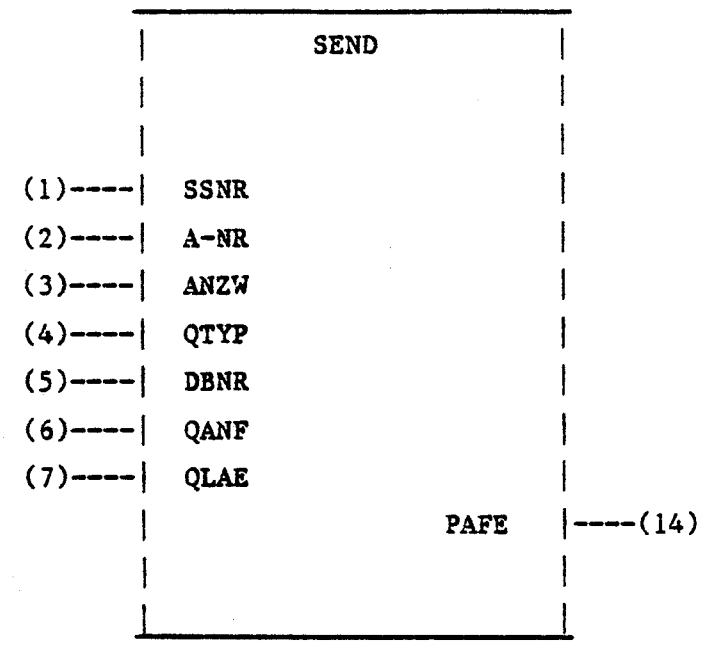


Figure 9: Function block FB SEND

4.1 Functional Description

The SEND function block is responsible for initiating a request on an interface (bubble memory).

In addition to the SSNR, A-NR, ANZW and PAFE parameters, the block requires the QTYP parameter.

4.2 Parameters

SSNR	=	Interface number
A-NR	=	Request of the interface
ANZW	=	Address of the condition code word indicating the processing of the initiated request
QTYP	=	Type of data source from which data is transferred
DBNR	=	Data block number
QANF	=	"relative starting address" of the data source
QLAE	=	Number of source data (in bytes or words)
PAFE	=	Error flag in case of parameter error

4.3 Parameter Description

*) SSNR: Interface number

Number of the interface for accessing the bubble memory.

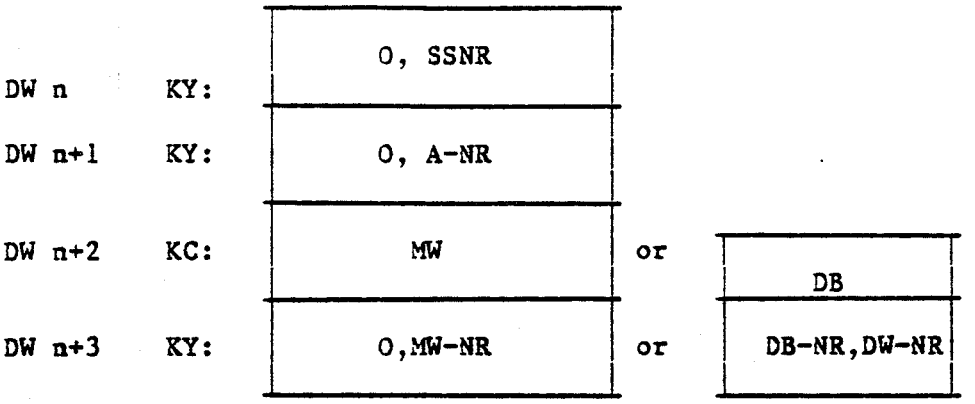
Kind of parameter/Type Data / KY

Possible range: 0,0 ... 255; 255

Admissible range: 0,0 ...0,7

High byte = 0: Direct parameter assignment:
Low byte = SSNR

High byte ≠ 0: Indirect parameter assignment;
Low byte = pointer to parameter field;
The parameters SSNR, ANR and ANZW are located in the data block opened prior to the call of FB SEND, according to the following order and format:



* A-NR: Request number

Number of the request for the interface addressed (bubble memory module):

Kind of parameter/Type: Data / KY

Possible range: 0,0 ... 0, 255

Admissible range: 0,10

The request number "10" performs the special function "Setting of the AG 115U" in case one programming device is connected to the bubble memory module.

* ANZW: Condition code word (double word)

Address of the condition code word containing the processing of the request specified under A-NR.

Kind of parameter/Type: Input/W

Possible range: DW 0 ... DW 255
MW 0 ... MW 254

Admissible range:	DW 0 ... DW 254	Flag word is in a data block
	MW 0 ... MW 252	Flag word is in the flag area

DW is related to the data block opened before the SEND block has been called.

Note: The specified word and the following one are used (DW n and DW n+1 or MW n and MW n+1).
Request servicing is shown in DW n or MW n.
The SEND block enters the number of data transferred in the word following the flag word ("length word").

Structure of the flag word.

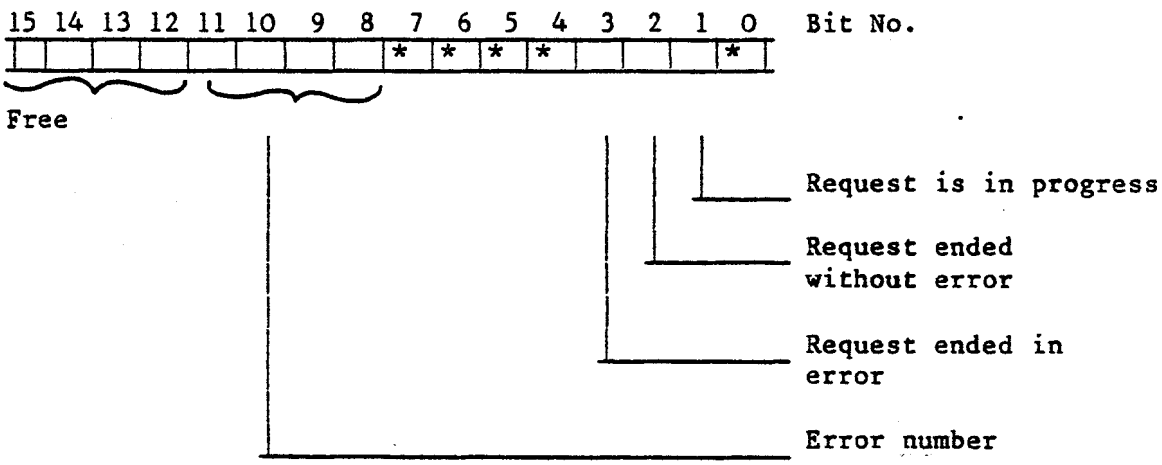


Figure 10: Structure of the flag word

*) These bits are irrelevant in the case of this SEND request.

The flag word is divided into four tetrads:

- Tetrad 1, bits 0 to 3, status flag:
  The codes indicate if a request has been started, if errors have occurred or if the request is already finished.

- Tetrad 2, bits 4 to 7, data management:
  Irrelevant

- Tetrad 3, bits 8 to 11, is available for error flags:
  The error flags of the request are displayed. These error flags are only valid when bit "request ended in error" is set in the status tetrad.

  The following flag may occur:

  0: No error
     If the bit "Request ended in error" is still set, it means that the interface has restructured the request after a cold start or a restart.

  3: Area in PC too small or does not exist

- Tetrad 4, bits 12 to 15, not used at present.

* QTYP: Type of data source

Default type of the data source

Kind of parameter/Type: Data / KC

Possible range: All ASCII characters

Admissible range: NN

The parameter QTYP = "NN" indicates that no further parameter is available; the parameters are available on the bubble memory module.

* DBNR: Data block number

Irrelevant

* QANF: Starting address of the source data block

Irrelevant

* QLAE: Length of the source data block

Irrelevant

* PAFE: Parameter error byte

The byte indicated here (flag byte, output byte, input byte) is set when the block detects a parameter error.

Kind of parameter/Type: Output / BY

Possible range:	EB 0 ... EB 127	
	AB 0 ... AB 127	
	MB 0 ... MB 255	
Admissible range:	EB 0 ... EB 127	(if available) for the CPU942/943
	AB 0 ... AB 127	(if available) for the CPU942/943
	EB 0 ... EB 63	(if available) for the CPU941
	AB 0 ... AB 63	(if available) for the CPU941
	MB 0 ... MB 255	

For the structure of the PAFE byte, refer to Section 3.3

The status flags of tetrad 1 perform the following functions:

Bit 1: Request is being processed

Set: By the SEND handling block when the corresponding request has been made.

Clear: By the SEND handling block when the appropriate request has been processed or interrupted.

Bit 2: Request ended without error

Set: By the SEND handling block when the request has been ended without error.

Clear: By the SEND handling block when the request is initiated again.

Bit 3: Request ended in error

Set: By the SEND handling block when the request has been interrupted due to an error.

Clear: By the SEND handling block when the request has been initiated again.

5 Software

Software required for a maximum of 8 bubble memory modules in the AG 115U:

- OB 21, OB 22 cold start, restart parametrized for 8 interfaces with the FB 249 SYNCHRON and the system specific SEND request
- FB 199 BUB-115U
- FB 244 SEND
- FB 245 RECEIVE
- FB 246 FETCH
- FB 247 CONTROL
- DB 255 Data buffer

6 Technical Data

Block number:	FB 199
Block name:	BUB-115U
Library number:	P71200-S 199-A-1
Block length:	832
Call length:	16
Processing time:	Typ. 24 ms to 66 ms CPU 941 Typ. 15 ms to 60 ms CPU 942 Typ. 8 ms to 50 ms CPU 943
Function blocks called:	FB 244 SEND FB 245 RECEIVE FB 246 FETCH FB 247 CONTROL
Data blocks called:	DB 255 Data buffer DB "PARA" (indirect parameter assignment) DB "ANZW" (flag word data block)
Flag area allocated: (scratchpad area)	MW 200 - MW 220

S I E M E N S

SIMATIC S5

Process Control Keyboard (PBT 982) 6ES5 982-2AA11
Process Control Keyboard (PBT 982) 6ES5 982-2BA11
with additional kyrillic layout

Operating Manual Order numbers: E89100-A503-U11-B

Table of Contents

1 T e c h n i c a l D e s c r i p t i o n
1.1 Field of Application
1.2 Design
1.3 Theory of Operation
1.3.1 State after Switch On
1.3.2 Matrix Processing
1.3.3 Interface Processing
1.4 Technical Data

2 O p e r a t o r I n p u t
2.1 Keypads
2.2 Keyswitches, LEDs

3 P u t t i n g I n t o O p e r a t i o n
3.1 Connection to the VDU (DS 075)
3.2 Connection to the Communications Processor (CP526)
3.3 Switch On

4 M a i n t e n a n c e
4.1 Connector Assignment
4.2 PBT 982-CP526 Connection Cable
4.3 Switches and Jumpers
4.3.1 State of Delivery
4.3.2 Selectable Transmission Speeds
4.3.3 Selectable Character Frames
4.3.4 Switch Off of the Alphanumeric Keys
4.4 Keyboard Coding
4.5 Keyboard Receive Signals
4.6 Test Possibilities

1 Technical Description

1.1 Field of Application

The PBT process control keyboard was designed for the control and observation of processes in a rough industrial environment.

The characteristics of the keyboard are as follows:

* High degree of protection, strong housing
* Keyboard layout with the most essential functions

The process control keyboard shall be used in conjunction with the CP 525 and CP 526 communications processors.

Configurations:

Operation at the CP 526 (1)
In this case the CP 526 is connected to the keyboard interface.

Operation at the CP 525 (2)
In this case the PBT is connected to the 3975 data display unit instead of the standard keyboard. The data display unit is connected to the CP 525 processor.

Operating modes switchover is performed automatically.

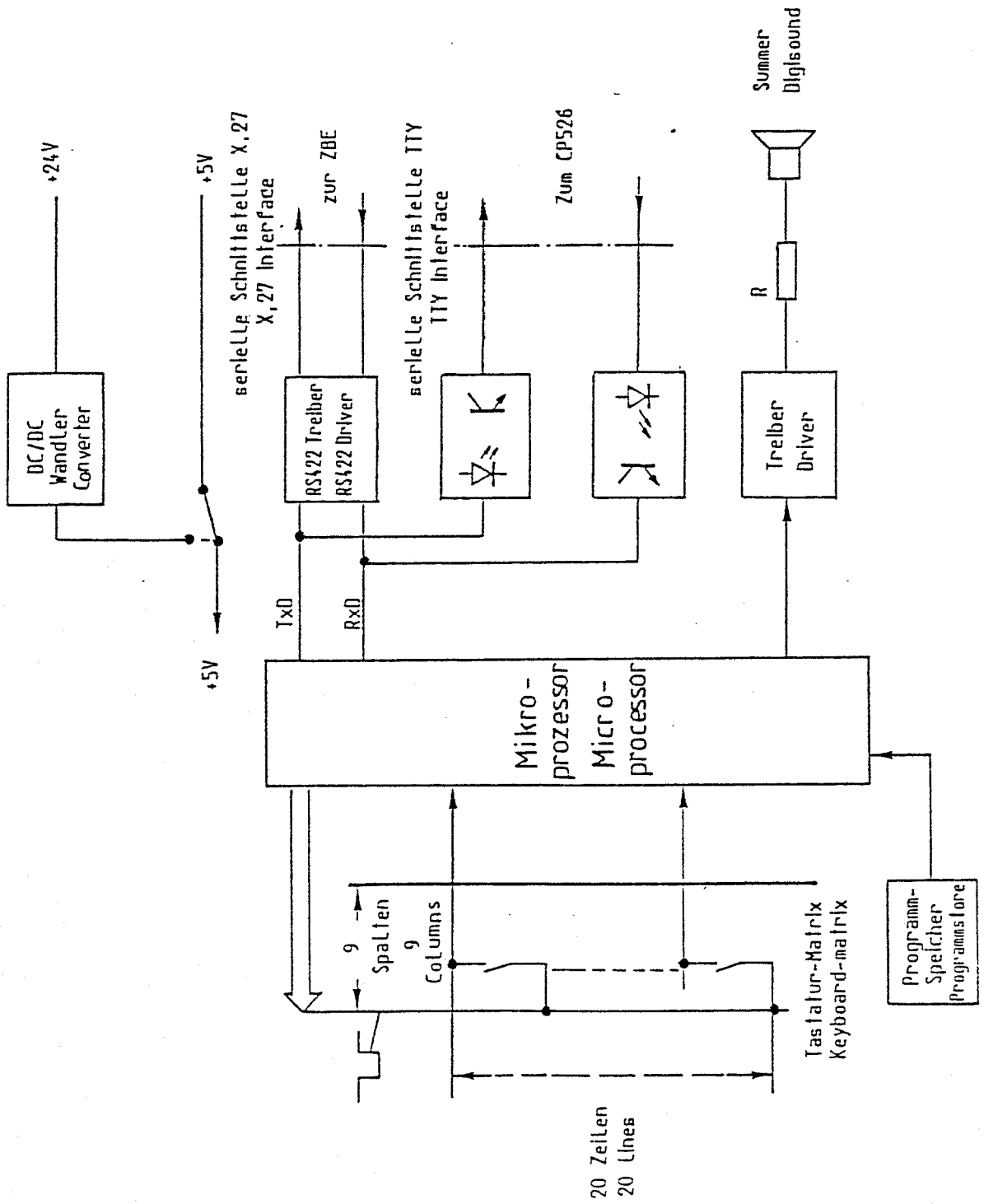
1.2 Design

The process control keyboard is provided with the following function units (see Figure 1.2):

- DC/DC converter
- TTY and RS422 serial interface
- Matrix and DIL switch logic
- Digisound logic
- Microprocessor with EPROM and RAM.

The connection to the 3975 display unit is established by the keyboard cable; a connection cable is required for connecting the CP 526.

Figure 1.2 Block diagram



1.3. Theory of Operation

1.3.1 After Switch On

Immediately after the unit has been switched on, still before entering the initial state, the whole device executes a self-diagnostic routine. Therefore it is necessary to connect the keyboard to the DS 075 display unit or to the CP 526 communications processor, so that the supply voltage is applied; set keyswitch to position "Keyboard ON" (hardware restart).

The self-diagnostic routine comprises a BUS test, an EPROM and a RAM test; the runout of the keys is tested (short-circuit). This test program is part of the device firmware.
The execution time amounts to approx. 1.5 sec. No character can be received by the keyboard during this period of time; the self-diagnostic routine cannot be interrupted. If defaults occur during this test, the red LED lights up with approx. 2Hz and keyboard operations are rejected.

If the self-diagnostic routine is performed successfully, the accoustic signal sounds shortly and the PBT process control keyboard enters the following initial state:

- Green LED lights up (power supply O.K.)
- International assignment
- Capital letters in UNSHIFT level
- Sound signal can be activated via the microprocessor by means of the BEL control character (sound signal has the maximum volume). The sound signal is twice as long as the acknowledge signal (approx. 300 ms).
- Acknowledge sound signal on (duration approx. 150 ms). (Normal sound volume of the acknowledge sound signal.)
- Brightness control (only in the case of CP 525) divided into 16 levels; can be adjusted for foreground and background via CTRL with the keys cursor control (brightness control on medium).

1.3.2 Matrix Processing

The 16 function keys, the alphanumeric keys, the digital keys and the operator keys are arranged according to the matrix layout (6 columns, 24 lines).
There are three other columns for the status lines (SHIFT, LOCK CTRL).
They are scanned cyclically by the processor, timer controlled. Therefore the column lines are set to "0" one after the other.If, during this process, a key is pressed, the corresponding line is also set to "0".

The hardware coordinates are obtained from these informations and the debouncing logic is started.

During the next two scan cycles further key depressions of the keyboard are checked. If no key is identified, this means that there is only one key actuation.

- Keyboard scan operation, 2 key lockout

However, if a further switch is identified as closed, none of both keys can be interpreted in order to prevent multiactuation. This mode applies to all keys, except when coordinating with the CTRL, SHIFT and SHIFT-lock status lines.

- FIFO memory

If a keystroke is found "all right", the key position together with the state of the status lines can be entered into a FIFO memory for a maximum of 8 characters. The key actuation is acknowledged by means of a sound signal (duration approx. 150 ms). If the FIFO memory is full, the key signal is no longer input and an error flag is set.

- Repetitive keys

By means of the software parameter assignment of the keys stored in EPROM, check is made as to whether the key may be repeated. If this is the case, the firmware supplies the desired repeat frequency.

Repeat generator:

Key shortly depressed	- repeat generator off (single function)
Key = depressed for 1 sec.	- after 1 sec. the key code is transmitted with a 15 Hz repeat frequency (see Key Assignment)

1.3.3 Interface Processing

- Automatic initialization of the interfaces

The power supply determines which of the 2 interfaces (CP 525 via DS075 display unit) is to be operated.

Interface for CP 525:

Transmission speed	: 2400 baud
Mode of transmission	: X.27 (asynchronous, serial)
Character frame	: 1 start bit, 7-bit ASCII code, 1 parity bit (even), 2 stop bits
Power supply	: +5V via VDU 3975

Interface for CP 526:

Transmission speed	: 2400 baud
Mode of transmission	: TTY passive (asynchronous, serial)
Character frame	: 1 start bit, 8-bit code, 1 parity bit (even), 2 stop bits
Power supply	: +24 V via CP 526

If the interface shall be changed, only the appropriate plug connection has to be established. The firmware automatically generates the new values specific to the interfaces after the unit has run through the above-mentioned self-diagnostic routine without error.

- Manual initialization of the interfaces

If transmission speeds and character frames, other than those automatically set are required, this can be achieved manually by means of two DIL switches after the present interface has been removed and the keyswitch is on position "Keyboard OFF".

- Information transmission

A code assigned to the key is transmitted via the TxD line drivers of the serial interface. The keyboard can receive data via the RxD line receiver of the serial interface (the format of the transmitted and received data is basically the same).
If during the transmit routine a receive interrupt occurs, it cannot be taken into consideration.

1.4 Technical Data

Order numbers:	
- Process control keyboard 982	6ES5982-2AA11
- Process control keyboard 982 with additional kyrillic layout	
- Connecting cable for the connection to the CP526	6ES5727-2...0 (length code)
Dimensions (in mm):	height x width x depth 48/23,5 x 490 x 234 (desk form)
Weight:	approx. 1.5 kg
Supply voltage:	
- Connection to 3975:	+5VDC, ±5 %
- Connection to CP526:	+24VDC, +25 % / -10 %
Power consumption:	+5V, 250mA +24V, 180mA
Degree of protection:	IP54
Environmental temperature:	
- during operation	0° to 40° C (32° F to 104° F)
- during storage and transport	-40° C to 70° C (-40° F to 158° F)

2 Operator Input

2.1 Keypads

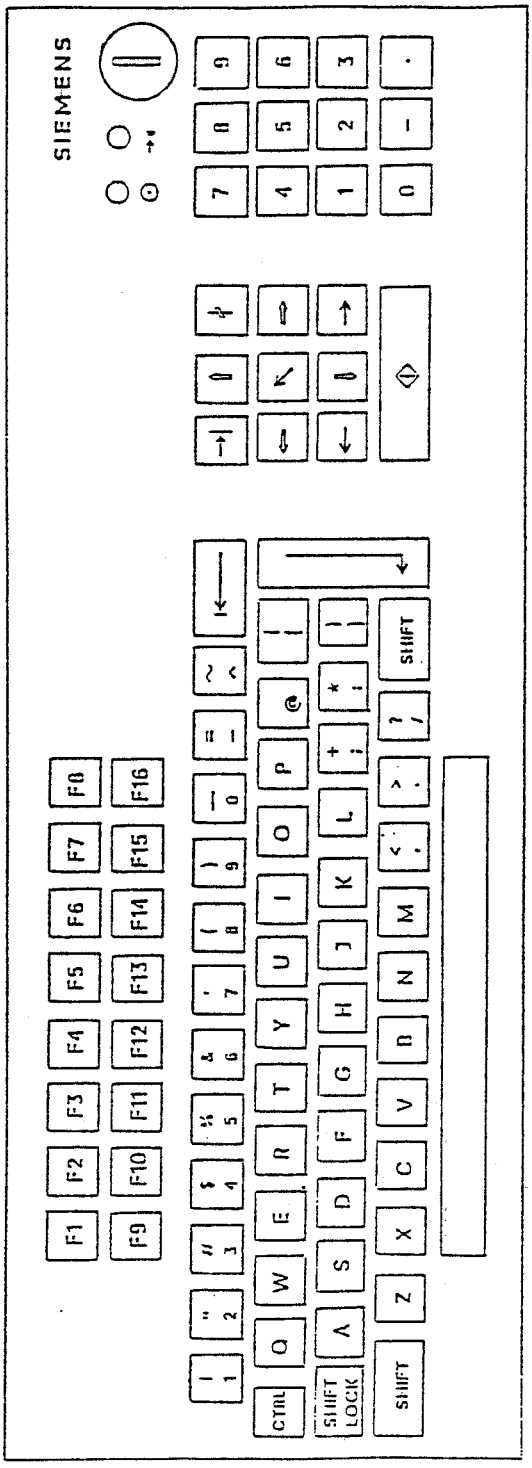


Figure 2.1 Keyboard layout (standard version)

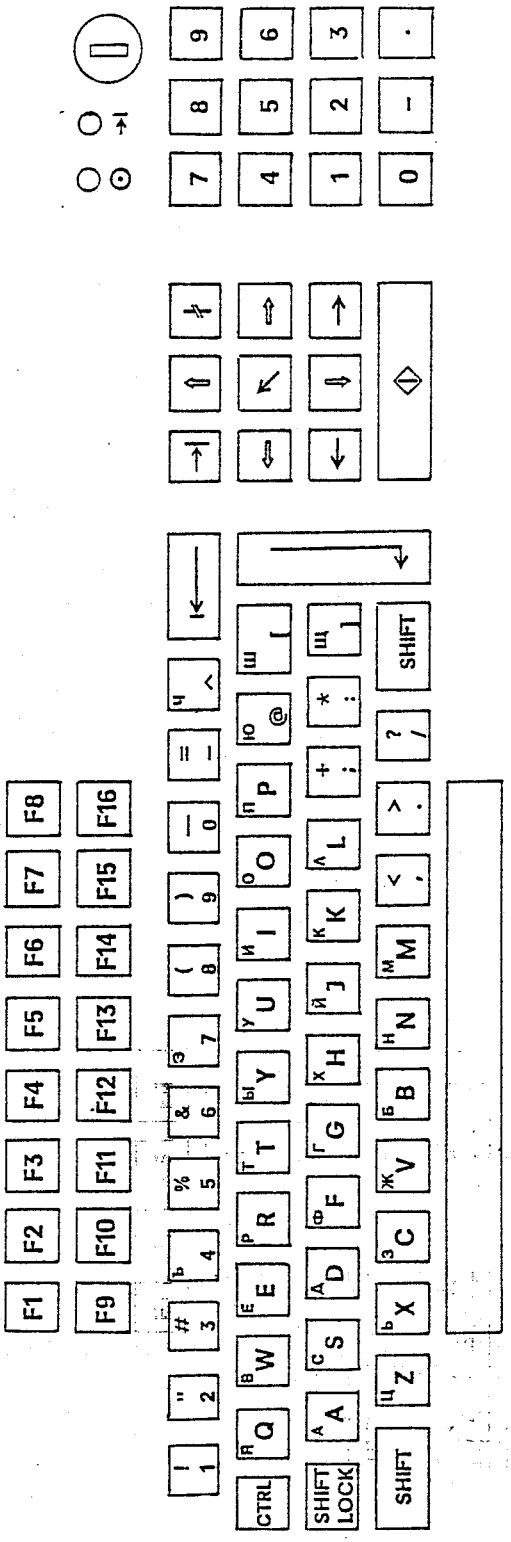
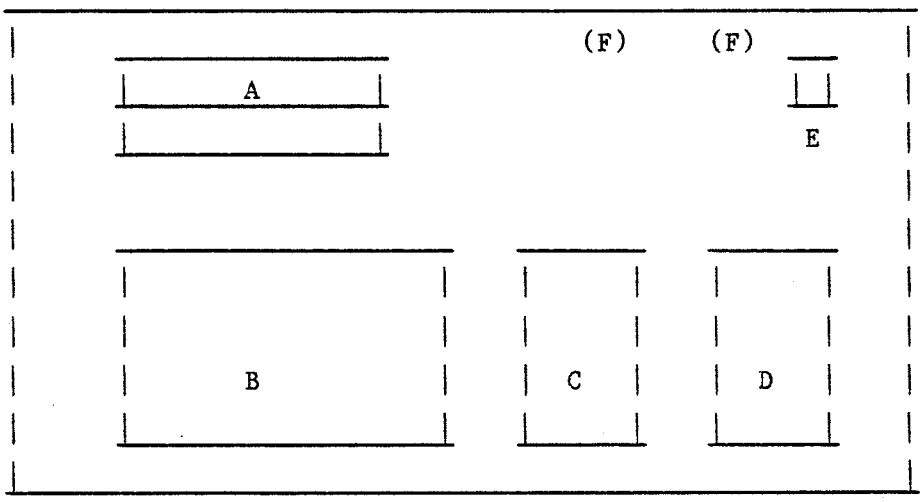


Figure 2.1a Keyboard layout with kyrillic characters
(in upper left corner of each key)

The keyboard is structured as follows:

A 16 function keys
B Alphanumeric keys
C Keys for cursor control and input
D Digital keys
E Keyswitches
F Optical displays (ready-to-operate, input lockout)



A: The 16 function keys are arranged in 2 rows of 8 below each other.

B: Alphanumeric keys (can be switched off, see Chapter 4).

For international layout resp. kyrillic layout see following page

i) International layout

Default after switching on: Capital letters

"Shift-Lock" - key

- alpha characters: capital letters <---> small letters
- special characters, number row: always the lower layout

"Shift"-key

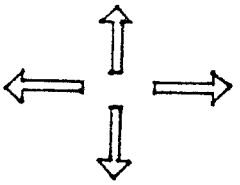
- alpha characters: capital letters <---> small letters
- special characters, number row: always the upper layout

ii) Additional kyrillic layout

Default after switching-on: capital letters.
When using the keyboard version with additional kyrillic layout the same ASCII codes are generated as for the international layout. Yet the reaction to "Shift" and "Shift Lock" is different:
The "Shift" and the "Shift-Lock" key each shift the complete alphanumeric block.

- alpha characters: capital letters <---> small letters
- special characters, number row: lower <---> upper layout

C: Keys for cursor control and input



Cursor control (cursor movement from input field to input field)

Cursor control within an input field

Cursor positioning to the first input field

a)For operation at the DS075:
initiation of a data input

b)For operation at the CP 526:
cursor positioning in the command line

Data transfer

Abort

D: The digital keys comprise the digits 0...9 as well as a decimal point and a minus sign.

2.2 Keyswitches, LEDs

Position keyswitch	Meaning
vertical	input lockout
horizontal	keyboard release

The ready status is displayed by the green LED. The green LED generally lights up when the power supply is on.

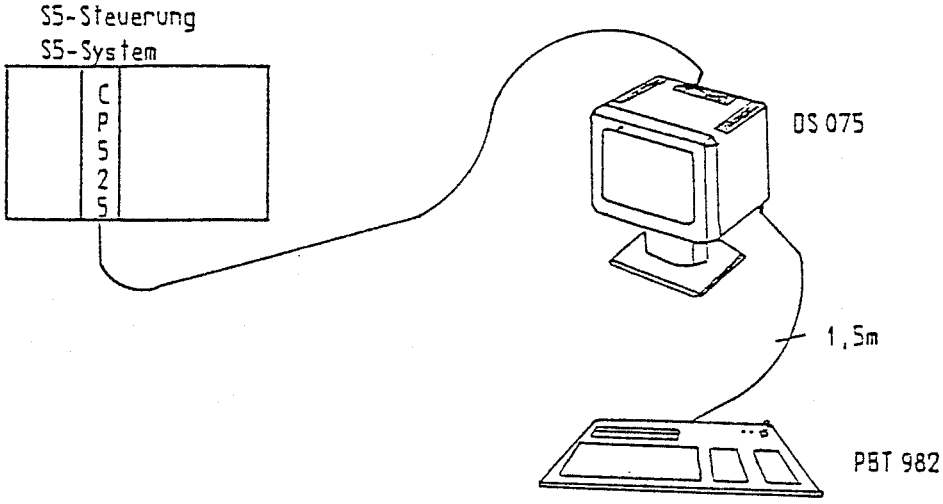
The red LED displays errors occuring during a self-diagnostic routine (blinking).

Keyboard operation is specified in the CP 525/526 operating manuals.

3 Putting Into Operation

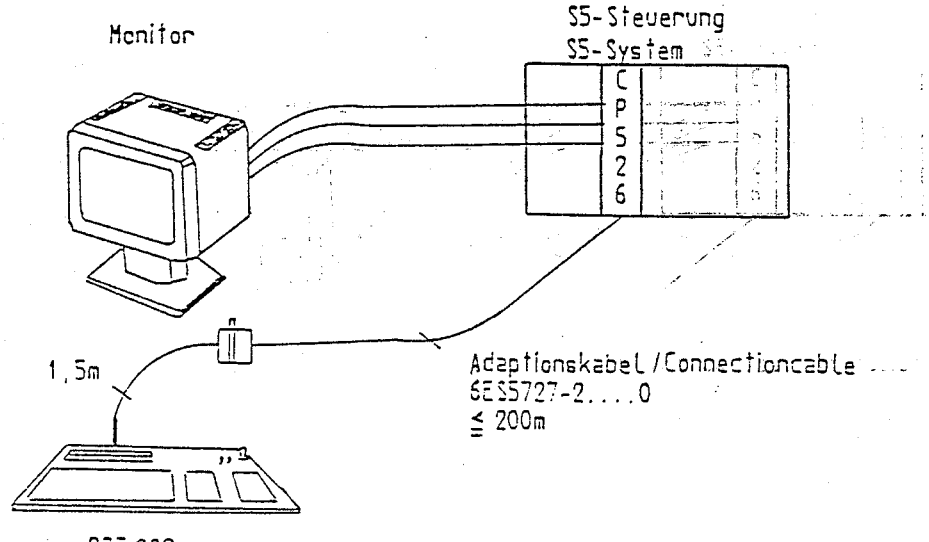
3.1 Connection to the DS 075 Display Unit

The keyboard cable is directly connected to the keyboard connection of the DS 075.



3.2 Connection to the CP 526 Communications Processor

The connection cable of the keyboard must not be used for direct connection to the CP 526. The connection has to be performed via the 6ES5727-2...0 connection cable. This cable can be delivered with a length of up to 200m.



3.3 Switch On

After the supply voltage has been switched to the DS 075 resp.CP 526 the keyboard is supplied with voltage via the connection cable.

The transmission speed and the character frame are selected according to the supply voltage.

The keyboard runs a self-diagnostic routine and is ready-to-operate approx. 1.5 second after successful completion.

The ready status is displayed by means of a sound signal as well as by the lighting of the green LED.

If the red LED also lights up (steady light, input lockout), the keyswitch has to be actuated in order to release the keyboard.

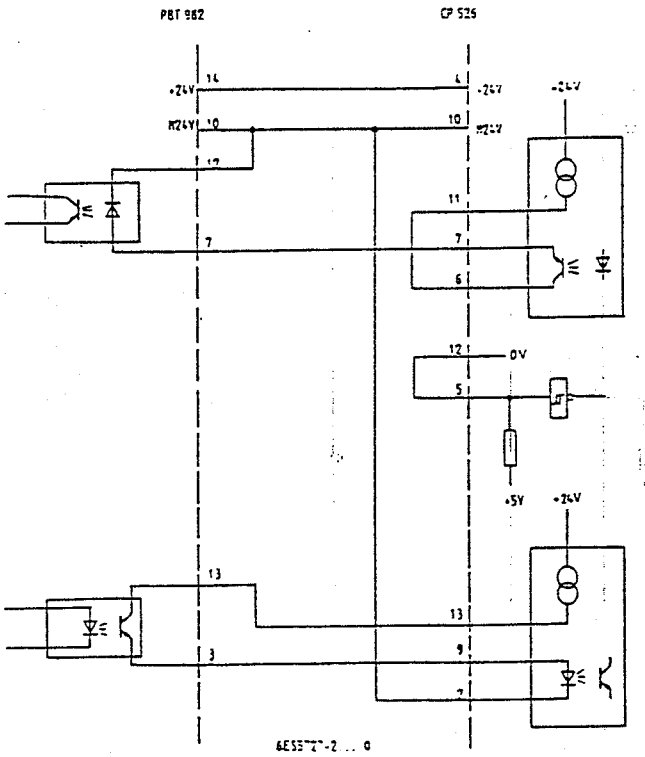
If an internal error is detected during the self-diagnostic routine, the red LED blinks with a frequency of approx. 2Hz; in this state operations cannot be performed.

4 Maintenance

4.1 Connector Assignment

not used	1		
		9	RS422 Receiver A
RS422 Receiver B	2		
		10	Ground 24 V
TTY Transmitter -	3		
		11	RS422 Transmitter A
RS422 Transmitter B	4		
		12	TTY Receiver -
not used	5		
		13	TTY Transmitter +
not used	6		
		14	+24V
TTY Receiver +	7		
		15	+5V
Ground 5V	8		

4.2 PBT 982-CP526 Connection Cable



4.3 Switches and Jumpers

4.3.1 State of Delivery

- S1 DIL switch

  4 3 2 1

  0 0 0 0 automatic Baud rate setting

- S2 DIL switch

  8 7 6 5 4 3 2 1

  0 0 0 0 0 0 0 0

								Automatic character frame
								setting
								Not used
								Alphanumeric keys active

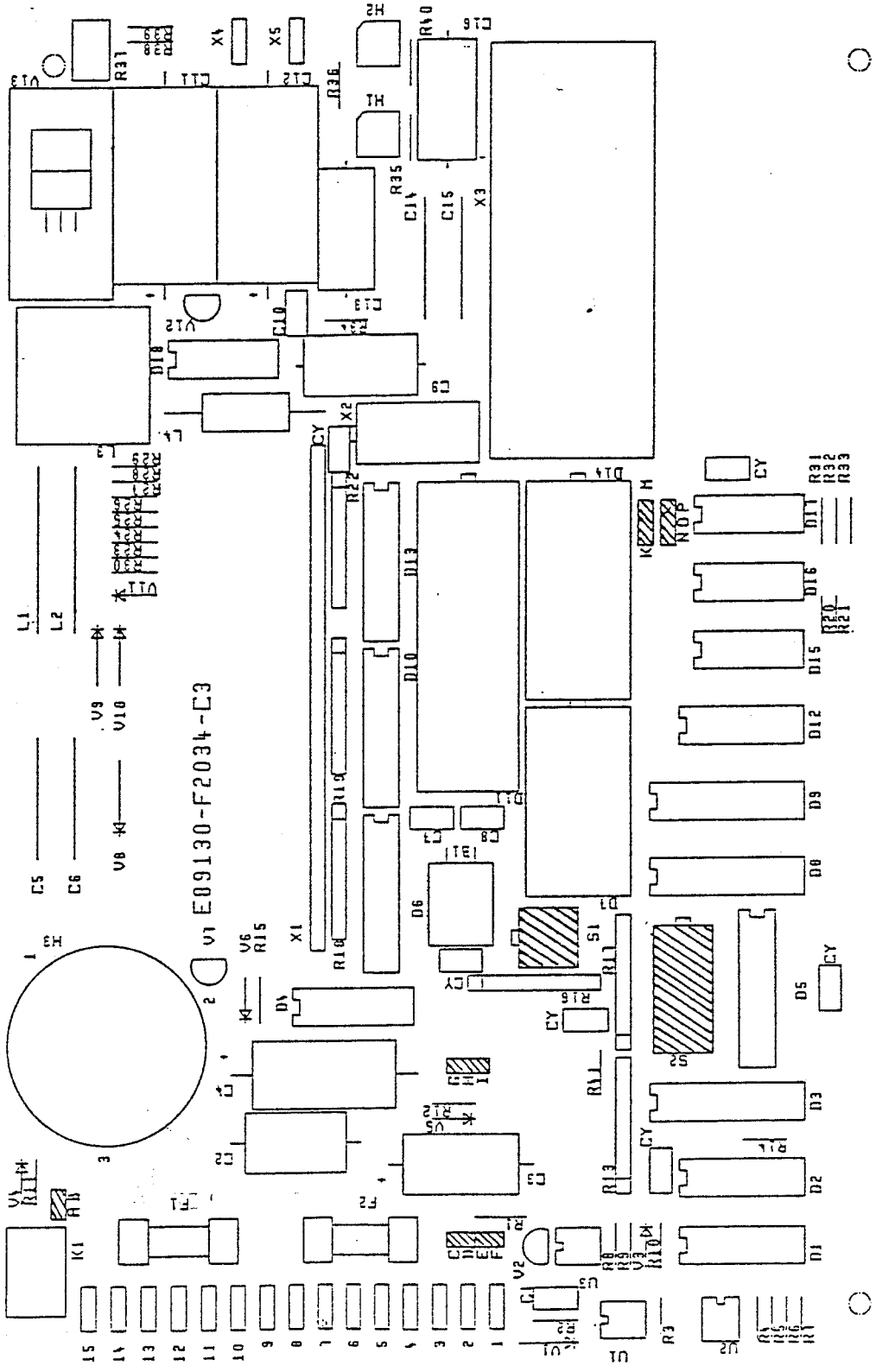
- Jumpers

  : = jumper removed, I = jumper inserted

Jumper	State of delivery	Meaning
A-B	I	Test jumper
C-D	I	No galvanic isolation
E-F	:	M not linked to screen
G-H	I	Test jumper
H-I	:	Test jumper
K-L	:	EPROM selection
L-M	I	EPROM selection
N-O	:	EPROM selection
O-P	I	EPROM selection

Baud rate and character frames can be modified and the alphanumeric keys be switched off for other applications. Arrangement of switches and jumpers, see Figure 4.3.1.

Bild 4.3.1 Arrangement of switches and jumpers



4.3.2 Selectable Transmision Speeds

S1 DIL switch 4 3 2 1	Function
0 0 0 0	Manual Baud rate selection inactive
0 0 0 1	19200 baud
0 0 1 0	9600 baud
0 0 1 1	4800 baud
0 1 0 0	2400 baud
0 1 0 1	1200 baud
0 1 1 0	600 baud
0 1 1 1	300 baud
1 0 0 0	150 baud
1 0 0 1	110 baud
:	Not used
:	
1 1 1 1	Special application: manufacturing test I (see 4.5)

4.3.3 Selectable Character Frames

S2 DIL switch 4 3 2 1	Function
0 0 0 0	Manual character frame selection inactive
0 0 0 1	7 bits + even parity + 2 stop bits
0 0 1 0	7 bits + odd parity + 2 stop bits
0 0 1 1	7 bits + even parity + 1 stop bit
0 1 0 0	7 bits + odd parity + 1 stop bit
0 1 0 1	7 bits + 2 stop bits
0 1 1 0	8 bits + 2 stop bits
0 1 1 1	8 bits + even parity + 1 stop bit
1 0 0 0	8 bits + odd parity + 1 stop bit
1 0 0 1	8 bits + even parity + 2 stop bits
1 0 1 0	8 bits + odd parity + 2 stop bits
:	Not used
:	
1 1 1 1	Special application: manufacturing test II (see 4.6)

The switches 5, 6 and 7 at the S2 DIL switch are not used.

4.3.4 Switch Off of the Alphanumeric Keys

The alphanumeric keys of the keyboard can be switched to inactive via the switch 8 of the S2 DIL switch.

S2 DIL switch Switch 8	
0	Alphanumeric keys active
1	Alphanumeric keys inactive

4.4 Keyboard Coding

- For connection to the DS 075 see Figure 4.4.a
- For connection to the CP 526 see Figure 4.4.b

Figure 4.4 a Keyboard coding for connection to DS075

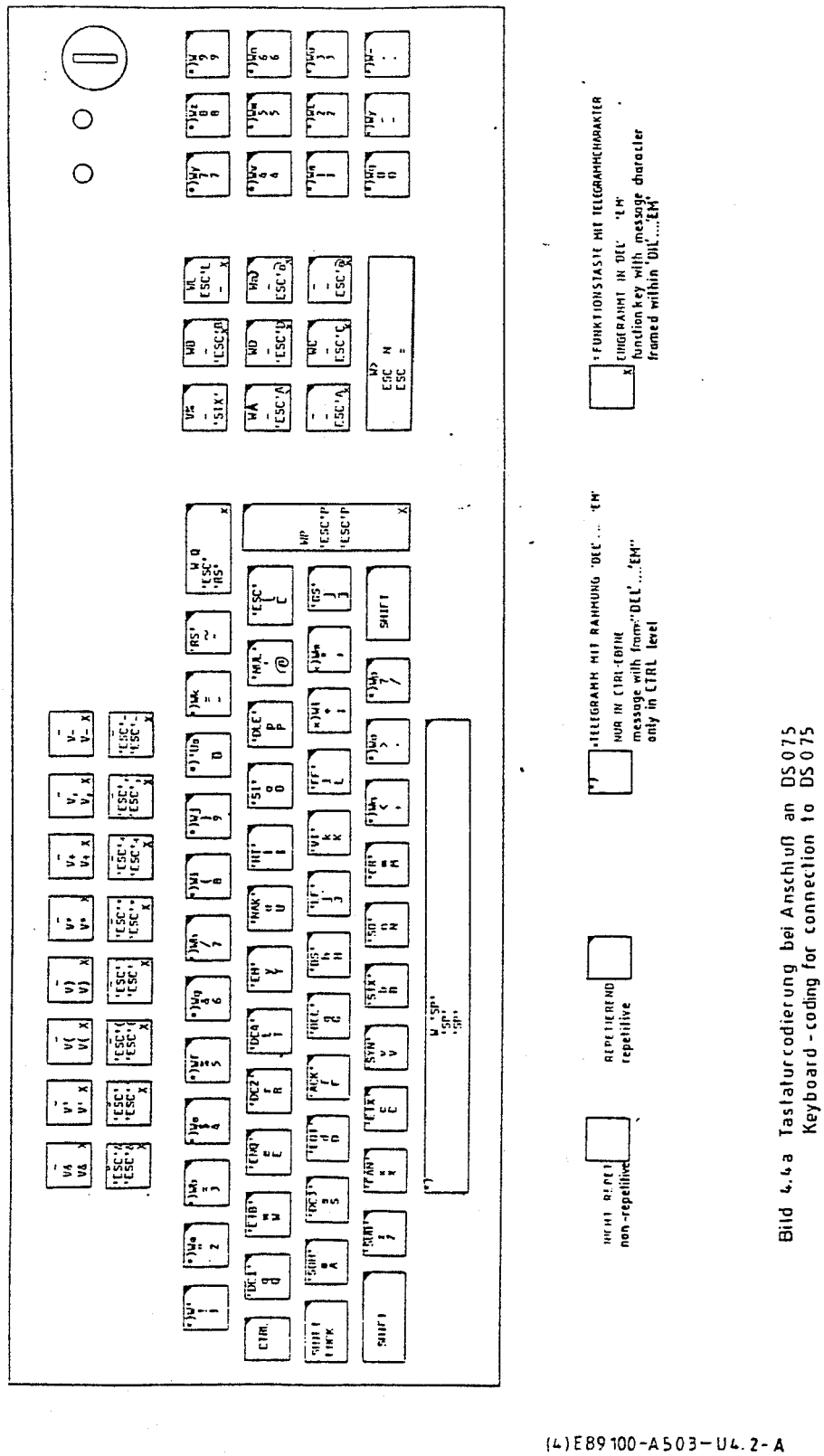


Bild 4.4 a Tastaturcodierung bei Anschluß an DS075
Keyboard-coding for connection to DS075

(4)E89100-A503-U4.2-A

Figure 4.4 b Keyboard coding for connection to CP526

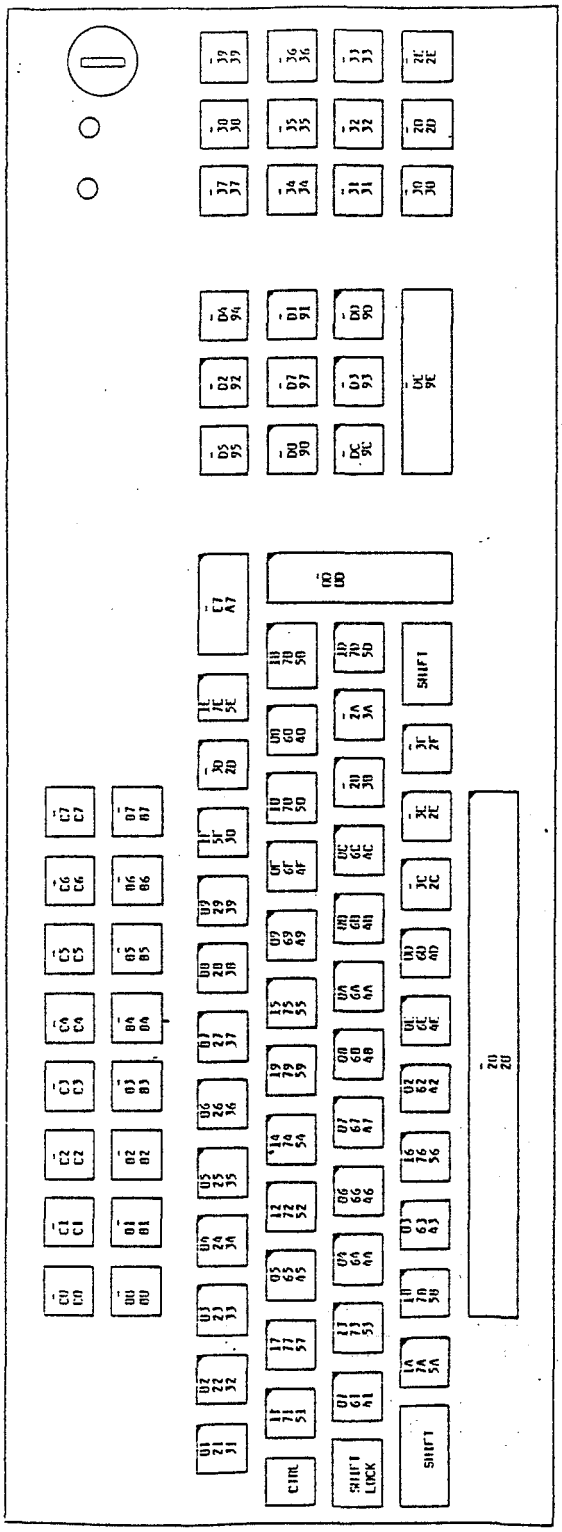


Bild 4.4 b Tastaturcodierung bei Anschluß an CP526
Keyboard-coding for connection to CP526

(4) E B9100-A503-U4.3 - A

4.5 Receive Signals of the Keyboard (only for CP 525)

The receive signals of the keyboard comprise parameter assignment messages and control characters.

- Parameter assignment messages

• Parameter assignment of the acknowledge signal and sound volume

Acknowledge signal loud : 'SUB' D a 'EM'
Acknowledge signal normal : 'SUB' D b 'EM'
Acknowledge signal low : 'SUB' D c 'EM'
Acknowledge signal off : 'SUB' D d 'EM'

• Parameter assignment capital-small letters

Capital letters in Unshift : 'SUB' D e 'EM'
Small letters in Unshift : 'SUB' D f 'EM'

• Parameter assignment of the repeat frequency

Repeat frequency f1=20 Hz : 'SUB' D g 'EM'
Repeat frequency f2=15 Hz : 'SUB' D h 'EM'
Repeat frequency f3=10 Hz : 'SUB' D i 'EM'
No key repetition : 'SUB' D j 'EM'

• Keyboard lockout/release

Keyboard lockout : 'SUB' D p 'EM'
Keyboard release : 'SUB' D q 'EM'

- Control characters

  Unlike the parameter assignment messages the control characters to the keyboard are not acknowledged.

  • Triggering of the alarm signal

    The alarm signal is triggered with the character "BEL".

  • Transmission inhibit

    Character transmission can be inhibited from the keyboard by means of the "DC3" character. If the FIFO memory is full, inputs to the keyboard are also inhibited.

  • Transmission release

    The transmision can be released by using the "DC4" character.

  • Acknowledgement signals

    Parameter assignment messages to the keyboard are acknowledged by it with:

    - Message understood : 'DEL' Q + 'EM'
    - Message not understood : 'DEL' Q - 'EM'

- Brightness control (only for CP 525)

  The brightness control is performed with the CRTL key as well as with the cursor control keys, according to the 3975 VDU specification.
  The brightness is adjustable in sixteen levels, separately for foreground and background.

4.6 Test Possibilities

For maintenance and manufacturing purposes there are provided two separate test possibilities which can be started independently, in addition to the self-diagnostic routine.

- Manufacturing test I (DIL switch test)

The interface to the CP 526 communications processor or to the DS 075 display unit has to be removed, i.e. the supply voltage is turned off.
Set the keyswitch to position "Keyboard OFF".
The DIL switch 1 is to be set as follows:

S1 DIL switch	Function
1 1 1 1	Manufacturing test 1

Establish connection to DS 075.
Set keyswitch to position "Keyboard ON". Now the unit runs the self-diagnostic routine, then records the DIL switch setting "Manufacturing test I" and acknowledges it with a single sound signal (approx. 500 ms).

Transmission speed, mode of transmission and character frame are adjusted to the DS 075 interface.

Now all DIL switches loose their original meaning; so their electromechanical function can be checked by chosen test settings. Each setting is displayed in binary form on the DS 075 display unit and can only be checked visually.

Example

DIL switch 1				DIL switch 2							
1	2	3	4	1	2	3	4	5	6	7	8
0	1	0	0	1	1	1	0	1	0	0	1

If the manufacturing test I is to be concluded, the interface to DS 075 has to be removed (the supply voltage is turned off).
Set keyswitch to position "Keyboard OFF".
Then all DIL switches are to be set again to position "0" (OPEN).
Automatic setting of the transmission speed or a manual Baud rate setting is to be performed according to Section 4.3.
Now the manufacturing test II or the normal operation can be started.

- Manufacturing test II (interface test)

The interface to the CP 526 communications processor or to the DS 075 display unit has to be removed (the supply voltage is turned off).
Set keyswitch to position "Keyboard OFF".
The DIL switch 2 is set as follows:

S2 DIL switch				Function
4	3	2	1	
1	1	1	1	Manufacturing test II

One of both short-circuit connectors can be connected to the interface optionally.
Set keyswitch to position "Keyboard ON".
The unit runs the self-diagnostic routine, then records the DIL switch setting "Manufacturing test II" and acknowledges it with a two time sound signal (approx. 300 ms.)

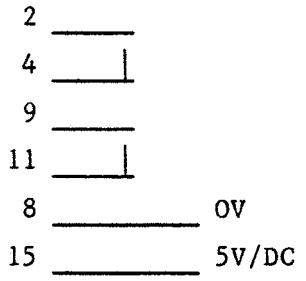
Transmission speed, mode of transmission and character frame are set automatically to the selected interface.

At the actuation of a key the corresponding code is transmitted and reflected to the unit by the short-circuit connector. The processor checks if the two messages are equal. If the transmitted and received messages are identical, this is acknowledged by a sound signal (duration approx. 150 ms).
A negative acknowledgement is performed in case of an error, i.e. the red LED remains blinking until the key is depressed.
In order to test the other interface, only the short-circuit connector has to be changed.

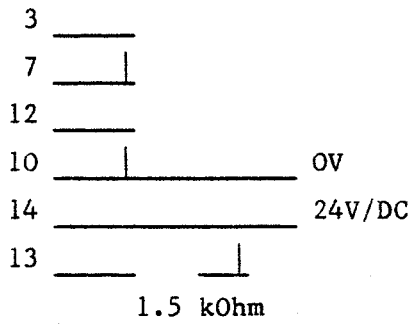
To conclude the manufacturing test II, the short-circuit connector has to be removed.
Set keyswitch to position "OFF". Then all DIL switches are to be set to position "0" (OPEN). Automatic setting of the transmission speed or a manual setting of the character frame is to be performed according to Section 4.3.

- Short-circuit connector for manufacturing test II

PBT 982 - Test RS422 interface (DS 075)



PBT 982 - Test TTY interface (CP526)



SIEMENS

SIMATIC S5

Process Communication Keyboard 982/2 6ES5982-3AA11
982/3 6ES5982-3BA11

Instruction Manual Order number: E89100-A611-U11-A

Contents

1 T e c h n i c a l D e s c r i p t i o n
1.1 Scope of Application
1.2 Design
1.3 Method of Operation
1.3.1 State after Power-up
1.3.2 Matrix Processing
1.3.3 Interface Processing
1.4 Technical Data

2 O p e r a t i n g
2.1 Keyboard Layout
2.2 Keylock Switch, LEDs

3 I n s t a l l a t i o n
3.1 Connection to CP526
3.2 Power-up
3.3 982/3 Desk Type

4 M a i n t e n a n c e
4.1 Pin Assignments
4.2 PBT 982 - CP526 Adapter Cable
4.3 Switches and Jumpers
4.3.1 State of Delivery
4.3.2 Selectable Bit Rates
4.3.3 Selectable Character Frames
4.3.4 Input Lockout
4.4 Keyboard Codes
4.5 Test Possibilities

1 Technical Specifications

1.1 Scope of Application

The 982/2 process communication keyboard was designed for controlling and monitoring processes in a rugged industrial environment. The keyboard thus has following inherent features:

* a high degree of protection and a robust casing
* a keyboard layout that is restricted to essentials.

The process communication keyboard is designed for use on the CP 526 communications processor.
The keyboard is connected to the keyboard interface on the CP 526 via an adapter cable.

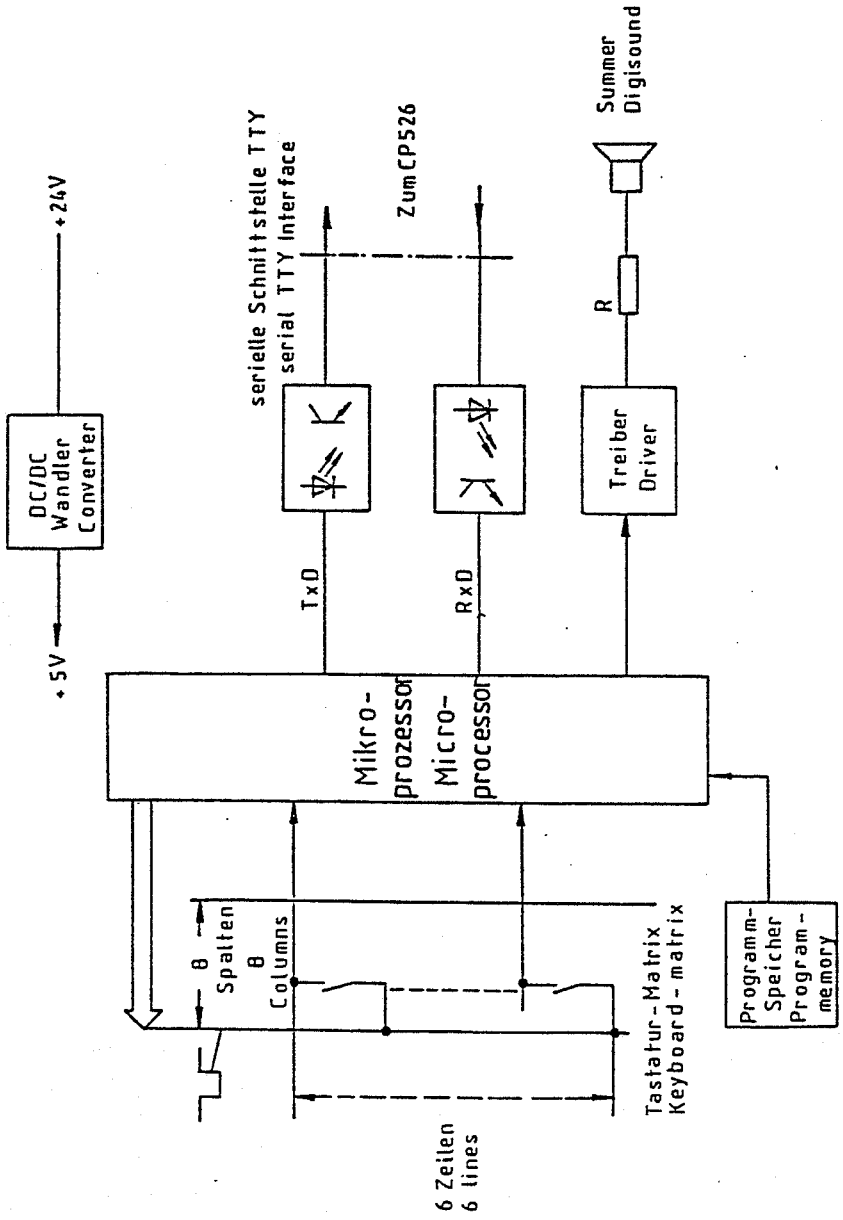
1.2 Design

The process communication keyboard contains the following functional units (see Fig. 1.2):

- DC/DC converter
- serial TTY interface
- matrix and DIL switch logic
- buzzer logic
- microprocessor with EPROM and RAM

The keyboard must be interfaced to the CP 526 via adapter cable 6ES5727-2...0.

Fig. 1.2 Block diagram



1.3 Method of Operation

1.3.1 State after Power-up

Immediately after power-up, still before entering the initial state, the entire unit is subjected to a self check. Prerequisite is that the keyboard be connected to the CP 526, so that operating voltage is on. Set keylock switch to the "Keyboard ON" position.

The self check comprises a bus test, an EPROM test and a RAM test; the keys are tested for unwanted repeats (short circuit). The self check routine is integrated in the device firmware, and requires approximately 1.5 seconds to execute.
During this interval, no characters will be accepted from the keyboard; the self check cannot be interrupted.
If defects are detected, the red LED flashes with approximately 2Hz and any attempted keyboard entries are rejected.

If the self check is o.k. the acoustic signal sounds briefly and the PBT enters the initial state.

1.3.2 Matrix Processing

The 16 function keys, the numeric keypad, and the operator control keys are arranged as matrix (8 columns, 6 lines).
The matrix is scanned cyclically by the processor timer-controlled. Herefore the column lines are connected consecutively to "0". If during that interval a key is pressed, the corresponding line is also brought to "0".

The hardware coordinates are set up from this information and the anti-bounce logic started.

The keyboard is checked for other pressed keys during the next two scan cycles. If none are determined, this indicates that only a single key was pressed.

- Keyboard scanning mode, 2-key lockout

If another closed switch is detected, however, neither of the two keys can be evaluated in order to exclude the possibility of multiple actuation.

- FIFO buffer

If the keystroke is regarded as being "all right", the key position, together with the state of the status lines, can be entered in a FIFO buffer for up to 8 characters. The keystroke is acknowledged by an acoustic signal (lasting approximately 150 ms). If the FIFO buffer was already full, the keyborad signal is not input any more and an error flag set.

- Repetitive keys

Based on the keys' software parameterization in the EPROM, the key is checked as to whether or not it may perform repetition (see key layout). If it may, the firmware provides the required repetition rate.

Repeat generator:

Key depressed briefly	- Repetition generator off (single function)
Key depressed for 1 s	- The key code is transmitted with 15Hz repetition frequency after 1 second.

1.3.3 Interface Processing

- State of Delivery

Transmission rate	2400 Baud
Transmission Mode	TTY passive (asynchronous serial)
Character Frame	1 start bit, 8-bit code,
	1 parity bit (even)
	2 stop bits
Power supply	+24V via the CP526

- Manual interface initialization

The default transmission rate and character frame may be changed, i.e. set manually via 2 DIL switches after the existing interface has been disconnected and the key-lock switch set to "Keyboard OFF" (see 4.3).

- Information transfer

Via the serial interface's TXD line driver, a code allocated to a key is transferred.
The keyboard can accept data via the serial interface's RXD receiver (the format of transmit and receive data is principally the same).

If a receiver interrupt occurs during the transmit routine, it cannot be taken into account.

1.4 Technical Data

Order numbers	
- Process communication keyboard	6ES5982-3AA11
- Adapter cable for connection to the CP526	6ES5727-2...0
Dimensions	Height x width x depth 42/20 x 328 x 210 (desk-type)
Weight	Approx. 1.5 kg
Supply voltage	+24V DC, +25 °/o/-10°/o
Current input	+24V, 50mA
Degree of protection	IP54
Environmental temperature	
- During operation	0 to 40°C
- During transport and storage	-40°C to 70°C

2 Operating

2.1 Keyboard layout

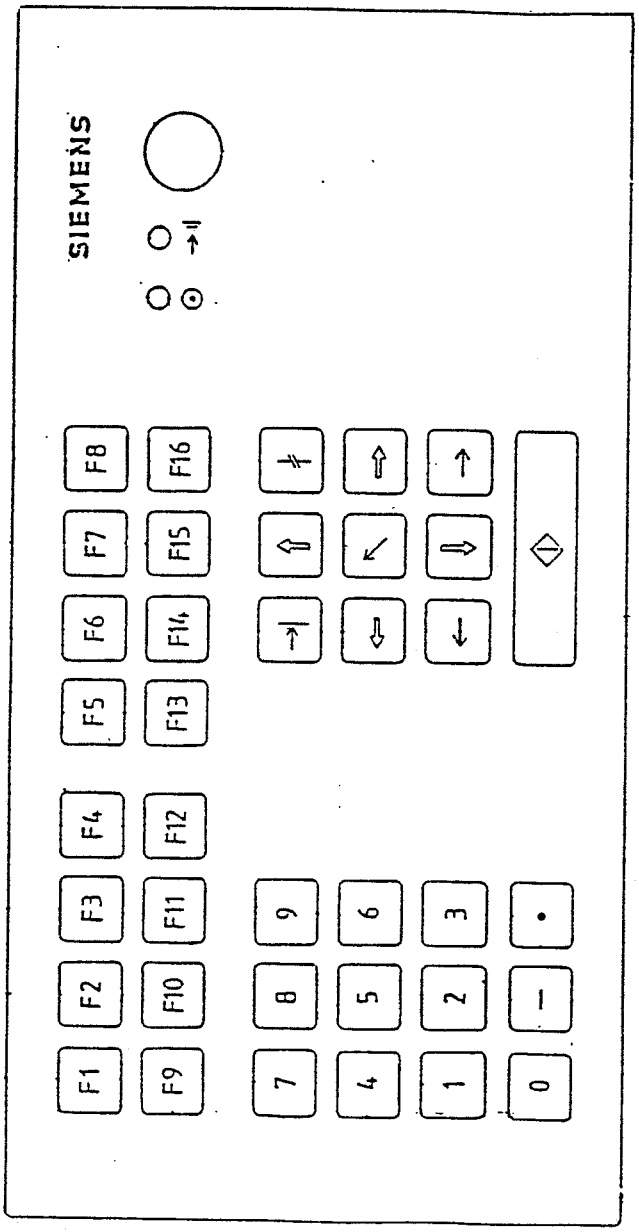
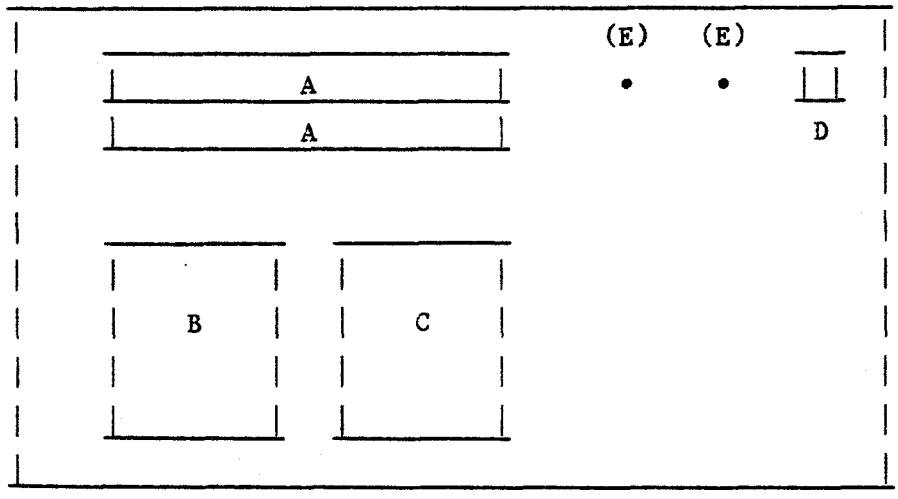


Bild 2.1 Tastenanordnung
Keyboard - Layout

Fig. 2.1 Keyboard layout

The keyboard is subdivided into the following keypads:

A 16 function keys
B Numeric keypad
C Keypad for cursor control and input
D Keylock switch
E Indicators (readiness for operation, input lockout)



Concerning A: The 16 function keys are arranged in two rows of 8 keys each, one row below the other.

Concerning B: The numeric keypad contains the digits 0...9, a decimal point and a minus sign.

Concerning C: Keypad for cursor control and input

Cursor control (to move cursor from input field to input field)

Moves cursor within an input field

Positions cursor to first input field

Positions cursor to command line

Data acception

Abort

2.2 Keylock Switch, LEDs

Position of keylock switch	Meaning
Vertical	Input lockout
Horizontal	Keyboard enable

The green LED indicates the ready status. The green LED is always on when the voltage supply is on.

The red LED flashes to indicate errors detected during the self check. The red LED lights constantly when the keylock switch is in the "Input lockout" position.

Details on keyboard operating can be found in the CP526 Instruction Manual.

3 Installation

3.1 Connection to CP526

The keyboard's connecting cable must not be used to connect the keyboard directly to the CP526. The keyboard must be connected to the CP526 via the adapter cable 6ES5727-2...0. This cable is available in lengths of up to 200 m.

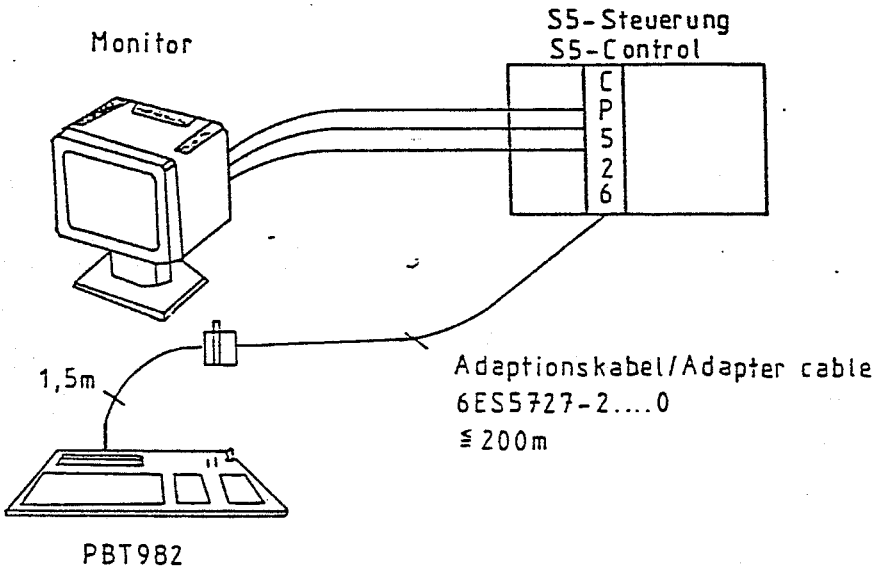


Fig. 3.1 Connection to CP526

3.2 Power up

After the supply voltage for the CP526 is switched on, the keyboard receives its supply voltage via the connecting cable.

The transmission rate and the character frame are preset (see 1.3.3).
The keyboard executes self check, and after successful completion is ready for operation within approximately 1.5 seconds. The ready state is indicated by an acoustic signal and lighting of the green LED.
If the red LED also lights (continuous lighting, input lock-out), the keylock switch must be set to enable the keyboard.

If an internal error was detected during the self check, the red LED flashes with a frequency of approximately 2Hz. Any attempted keyboard entries are rejected in this state.

3.3 982/3 Desk Type

The process communication keyboard is also available as desk type 982/3. This type is electrically identical to the 982/2 keyboard. The mechanical construction of the desk type is shown in figure 3.2.

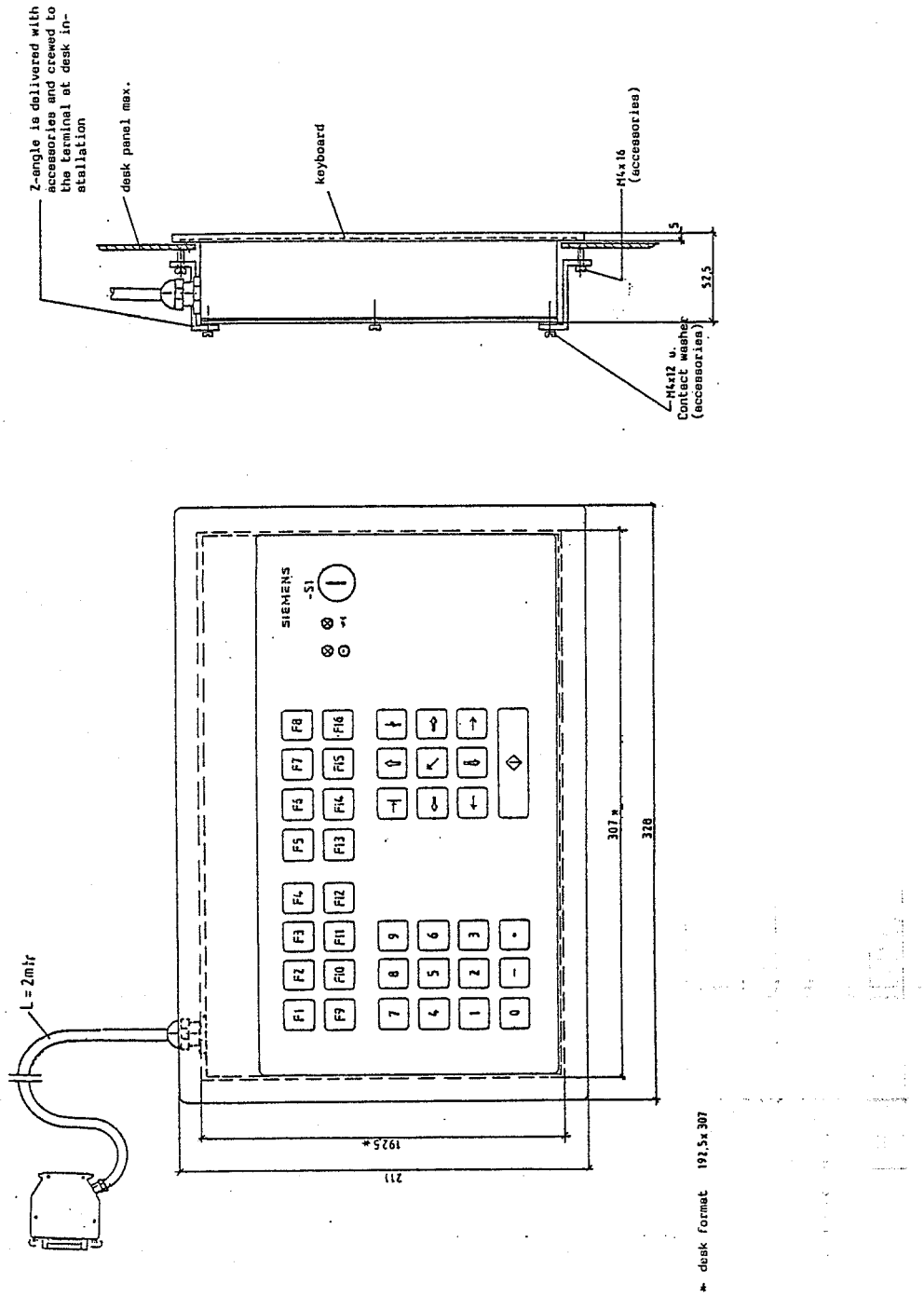


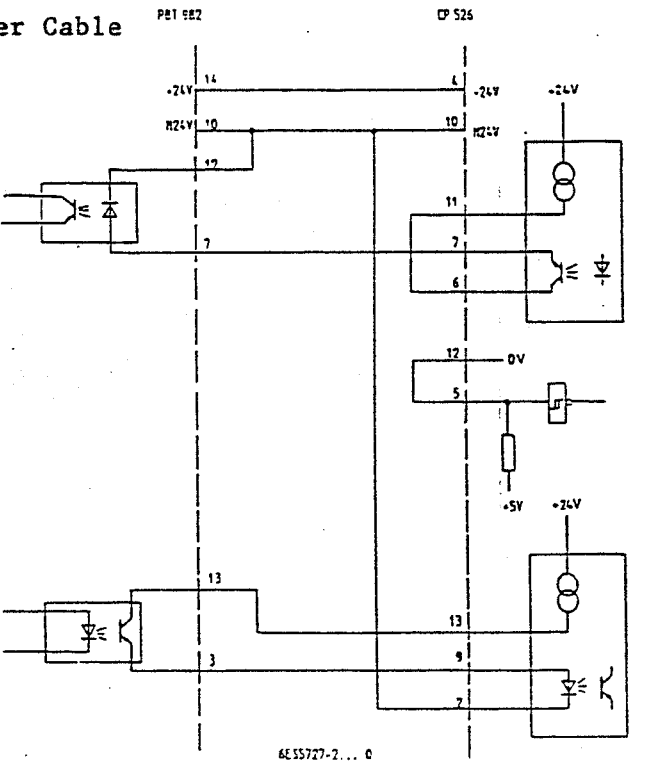
Fig. 3.2 Process communication keyboard, desk type 982/3

4 Maintenance

4.1 Pin Assignments

	1		
		9	
	2		
		10	Ground 24V
TTY transmitter -	3		
		11	
	4		
		12	TTY receiver -
	5		
		13	TTY transmitter +
	6		
		14	+24V
TTY receiver +	7		
		15	
	8		

4.2 PTB982 - CP526 Adapter Cable



4.3 Switches and Jumpers

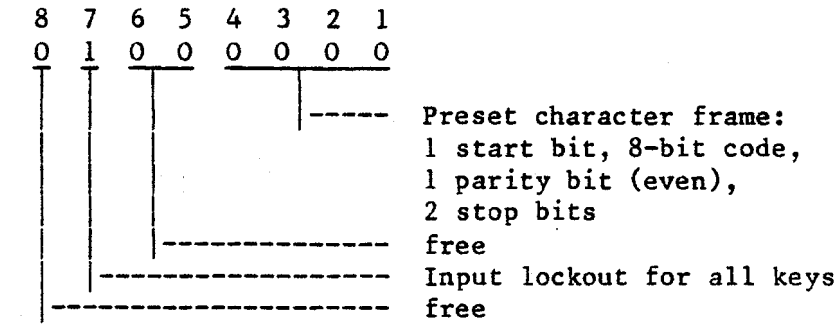
4.3.1 State of Delivery

- DIL switch S1

  4 3 2 1

  0 0 0 0 Preset Baud rate: 2400 Baud

- DIL switch S2



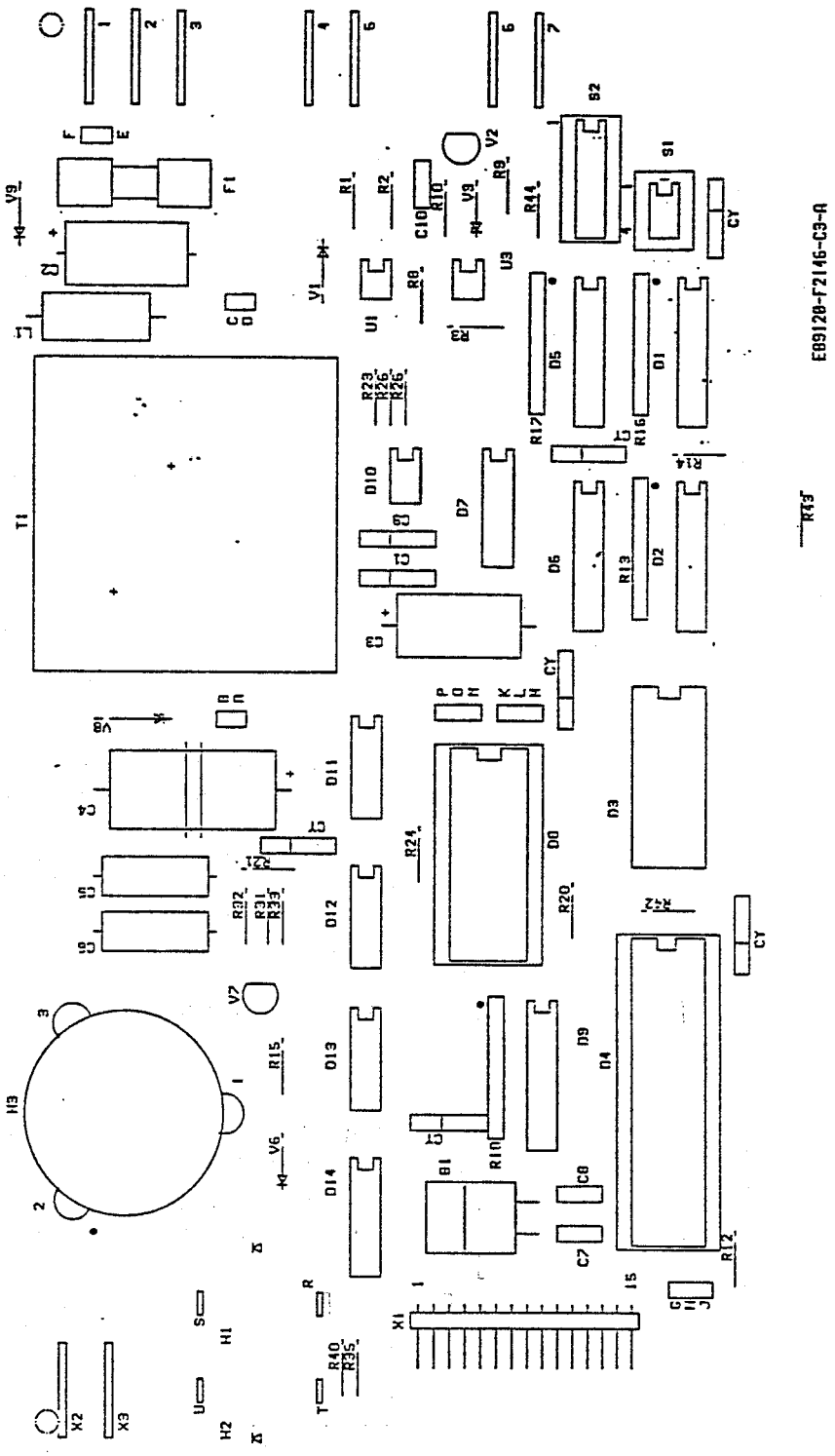
- Jumpers

  : = jumper removed, I = jumper inserted

Jumper	Standard version	Meaning
A-B	I	Test jumper
C-D	:	Galvanic Isolation
E-F	:	M not connected to shield
G-H	I	Test jumper
H-I	:	Test jumper
K-L	:	EPROM select
L-M	I	EPROM select
N-O	:	EPROM select
O-P	I	EPROM select

Baud rate and character frame can be changed, and the alpha keypad shut down, for other applications.
For locations of the switches and jumpers see Fig. 4.3.1.

Fig. 4.3.1 Locations of switches and jumpers



4.3.2 Selectable Bit Rates

DIL switch S1				-	Function
4	3	2	1		
0	0	0	0	-	Manual baud rate setting inactive
0	0	0	1	-	19200 baud
0	0	1	0	-	9600 baud
0	0	1	1	-	4800 baud
0	1	0	0	-	2400 baud
0	1	0	1	-	1200 baud
0	1	1	0	-	600 baud
0	1	1	1	-	300 baud
1	0	0	0	-	150 baud
1	0	0	1	-	110 baud
	:			-	free
	:				
1	1	1	1	-	Special application: Production test I (see 4.5)

4.3.3 Selectable Character Frames

DIL switch S2				-	Function
4	3	2	1		
0	0	0	0	-	Manual character frame setting inactive
0	0	0	1	-	7 bits + even parity + 2 stop bits
0	0	1	0	-	7 bits + odd parity + 2 stop bits
0	0	1	1	-	7 bits + even parity + 1 stop bit
0	1	0	0	-	7 bits + odd parity + 1 stop bit
0	1	0	1	-	7 bits + 2 stop bits
0	1	1	0	-	8 bits + 2 stop bits
0	1	1	1	-	8 bits + even parity + 1 stop bit
1	0	0	0	-	8 bits + odd parity + 1 stop bit
1	0	0	1	-	8 bits + even parity + 2 stop bits
1	0	1	0	-	8 bits + odd parity + 2 stop bits
	:			-	free
	:			-	
1	1	1	1	-	Special application: Production test II (see 4.5)

Switches 5,6 and 8 on DIL switch S2 are not in use.

4.3.4 Input Lockout

Via switch 7 on DIL switch S2 the function of the input lockout can be set as follows:

DIL switch S2
Switch 7

0	Input lockout does not apply for function keys
1	Input lockout applies for all keys (hardware reset)

4.4 Keyboard Codes

See Fig. 4.4

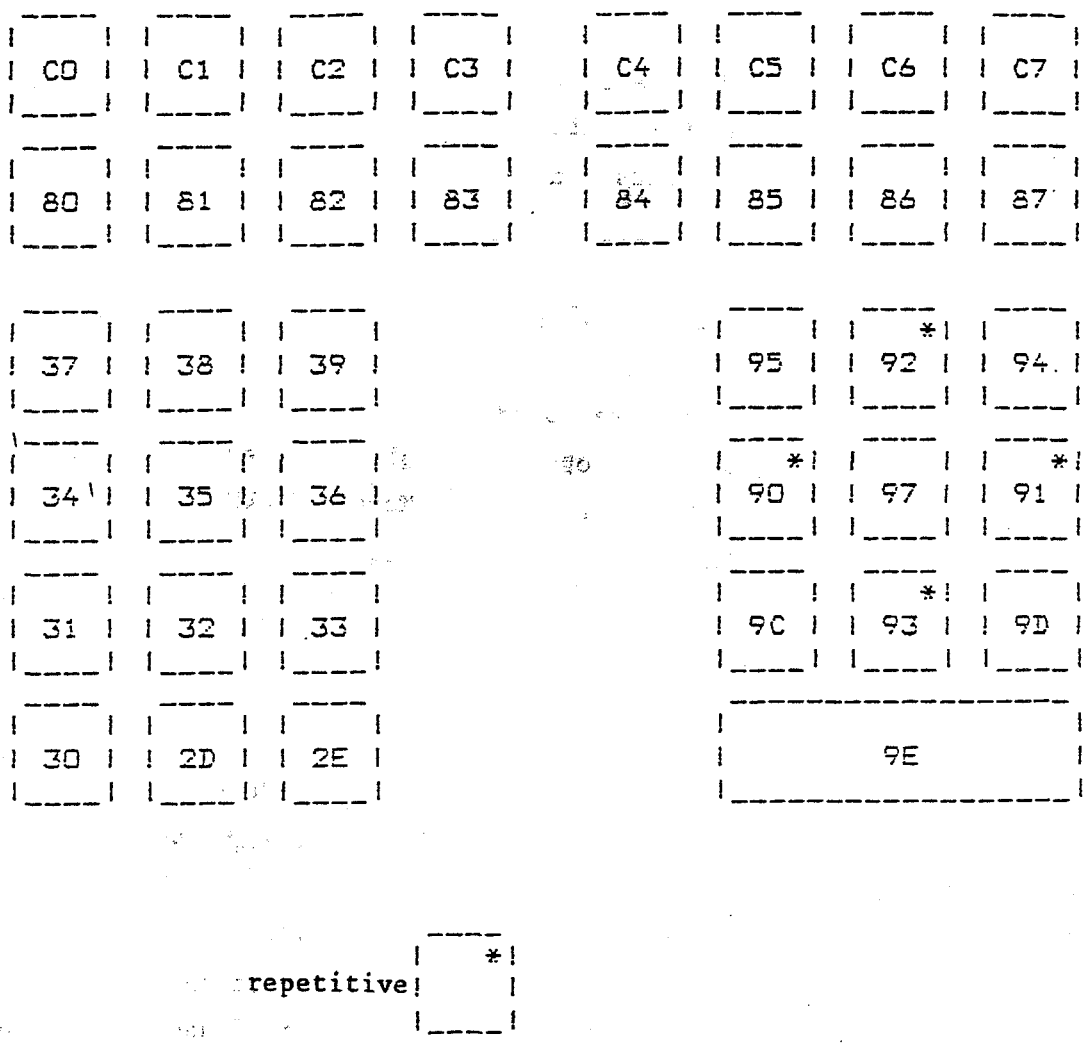


Fig. 4.4 Keyboard codes

4.5 Test Possibilities

In addition to the self check, the device firmware provides two separate test features for maintenance and production. Each of these tests can be started independently of the other.

- Production test I (DIL switch test)

For this test the keyboard must first be disconnected from the CP526, i.e. the operating voltage is switched off.
Set the keylock switch to the "Keyboard OFF" position.
Set the following on DIL switch 1:

DIL switch S1	Function
1 1 1 1	Production test I

Establish a connection to a terminal with a TTY interface (terminal active). Supply the keyboard with 24V:

Set the keylock switch to "Keyboard ON".
The keyboard first runs the self check. It then registers the DIL switch setting "Production test I" and acknowledges with a single acoustic signal (approx. 500 ms).

Transmission rate, transmission mode and character frame are preset.

A 1 1 DIL switches now loose their inherent significance and can be adjusted arbitrarily in order to test their electro-mechanical functions.
Each setting is displayed in binary at the terminal, and can now be checked visually.

Example:

DIL switch 1				DIL switch 2							
1	2	3	4	1	2	3	4	5	6	7	8
0	1	0	0	1	1	1	0	1	0	0	1

To terminate production test I, the interface to the terminal and the 24V voltage supply must be disconnected. Set the keylock switch to "Keyboard OFF".
All DIL switches must now be returned to position "0" (OPEN). Refer to 4.3 for manual or automatic setting of the baud rate. It is now possible to proceed to production test II or to return to normal operation.

- Production test II (interface test)

The keyboard must first be disconnected from the CP526 communications processor (operating voltage is switched off).
Keylock switch: s set to "Keyboard OFF".
Set the following on DIL switch 2:

DIL switch S2				Function
4	3	2	1	
1	1	1	1	Production test II

The short-circuit plug (TTY) can now be connected to the interface.
Set the keylock switch to "Keyboard ON".
The keyboard runs the self check, registers that the DIL switch is set to "Production test II" and acknowledges with a dual acoustic signal (approx. 300ms).

Transmission rate, transmission mode and character frame are preset automatically.

When a key is pressed, the corresponding code is transmitted and echoed back into the device via the short-circuit plug. There the processor checks the two codes to their equality. If the code transmitted and the code received are identical, this is acknowledged by an acoustic signal (approximately 150ms duration).

The short-circuit plug must be removed to terminate production test II.
Set the keylock switch to "OFF".
All DIL switches must then be returned to the "0" position (OPEN). Refer to section 4.3 for automatic setting of the transmission rate or manual character frame setting.

- Short-circuit plug for production test II

PBT982 TTY test interface (CP526)