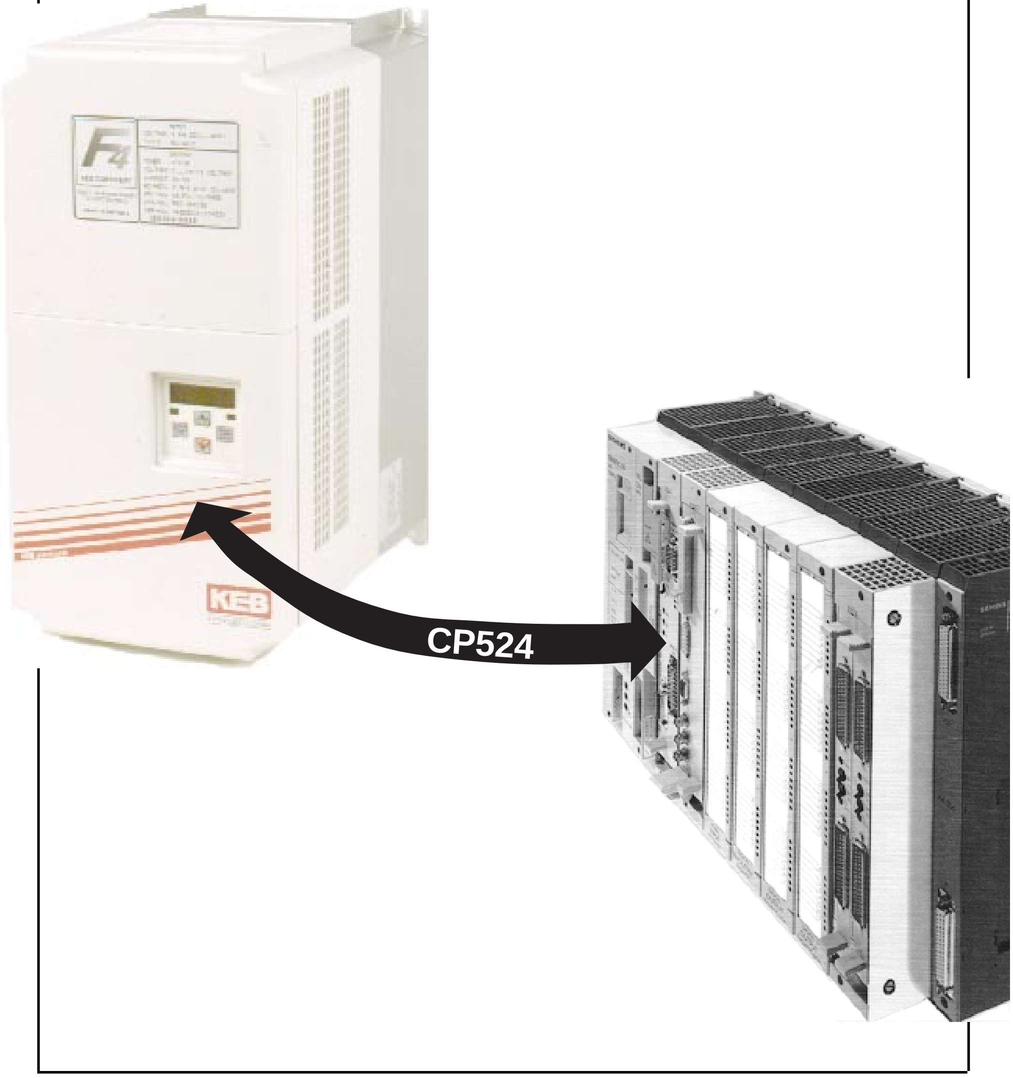
BETRIEBSANLEITUNG

INSTRUCTION MANUAL



KEB COMBICOM **SIMATIC S5 / KEB-DIN66019**

0S.58.038-K005

KEB
ANTRIEBSTECHNIK

12/93

FUNCTION PACKET FOR THE COMMUNICATION OF KEB FREQUENCY INVERTERS WITH SIMATIC S5 VIA KEB-DIN66019 BY MEANS OF CP524

Table of Contents

1. General

The documentation as well as the hardware and software are developments on behalf of the Karl E. Brinkmann GmbH. Errors and omissions excepted! The Karl E. Brinkmann GmbH has prepared this documentation to the best of their knowledge, but no guarantee is given that the specifications will bring the user the efficiency aimed at. The Karl E. Brinkmann GmbH reserves the right to change the specifications without obligation. All rights reserved!

KEB Antriebstechnik develope, manufacture and distribute worldwide static frequency inverters in the industrial power ranges. The inverters of the types F0, F1, F2, 56C and 56D are equipped with an integrated serial interface through which the frequency inverter can be completely parameterized by means of KEB-DIN66019 protocol.

With the aid of the S5 driver package it is possible to parameterize KEB frequency inverters from a SIMATIC S5 without any additional hardware expenditure for the inverter control.

1.1 Ordering Directions

Order Number

0S.58.038-0005:	KEB-S5 driver software for communication via DIN66019 by means of CP524.
0S.58.038-K005:	This Instruction Manual

Required SIEMENS components

6ES5 524 3UA13, as of issue state 5:	Communications processor CP524.
6ES5 752 0AA42:	RS422/RS485 interface module for CP524.
6ES5 897 2MB11:	Special driver S5R00T for CP524.

2. Description of Function Modules

2.1 General

The user can copy the function modules into other block numbers or call them repeatedly in the program so that several CP524 can be used as well. The function packet is designed in such a manner that it can run on a AG S5-115U as well as on a AG S5-135U. When using it on a AG S5-135U the SIEMENS data handling blocks FB120, FB125, FB126, FB127 must be loaded in the AG. For the AG S5-115U these functions are already integrated in the operating system.

2.2 Function Module FB230:CP524SYN for the System Start

Normally the function module in the start-up block OB20-OB22 is called once for each CP524 to be supplied, in this context different parameters are to be specified in the block. The FB230:CP524SYN determines first in which automation unit the function packet shall be used. If no known CPU type is identified then the error bit ECPU is set. After that a check is made whether the data block parameterized as DBNR is loaded in the AG. All necessary internally required data will be deposited and processed in the DB later on.
If a value smaller than 10 or larger than 255 is specified as data block number then the error bit EDBN is set. Subsequently the user has to check the parameterizing of the specified parameter. If the parameterizing is valid and if the DB exists then the processing of the block is continued. But if the DB does not exist then the further process depends on the automation unit: that means with a AG S5-135U the block can be created by way of indirect parameterizing, which is not possible with a AG S5-115U. Consequently an error bit EDBA is set at AG S5-115U and further processing is impossible. Whereas the block is created with a AG S5-135U. In the event that the aforementioned error occurs in the AG S5-115U the user has to create the DB with a length of 150 data words once by hand.

With AG S5-135U it is then being checked whether the data handling blocks FB120:SEND, FB125:SYNCHRON, FB126:SEND-A, FB127:REC-A are loaded in the automation unit as these blocks handle the further process of communication with the CP524. During this enquiry only the existence of the block numbers is checked, for that reason it is not allowed to copy these blocks to other block numbers. If one of the mentioned blocks is missing then the error bit ERR is set and in the DW1 of the parameterized DBNR the table below can be read that shows which block was not found.

If all of the functions mentioned so far have been executed without error then the boot values for the cyclic operation are written on the DBNR thus the data block contains all basic settings for the function module FB231:CP524KEB. The interface number SSNR of the CP524 to be supplied is also entered in this data block as a result the block FB231:CP524KEB can supply several CP524 in the case of multiple calls.

As last function the FB230:CP524SYN carries out a synchronization between the CP524 under the interface number specified with SSNR and the CPU specified under the CPUN. This parameter is only of interest for AG S5-135U when several CPUs are inserted. If an error occurs during the synchronization process then the error bit ERR is set and the error code can be read in DW3 (see below) of the DBNR that has been returned by the synchronization block.

If no error has occurred in the functions listed so far then a bit is set internally in the data block which acts as release for all further cyclic functions to be executed.

2.2.1 Module Parameter of FB230:CP524SYN

SSNR:
Interface number, the CP524 to be supplied must be adjusted to it (see hardware description). The parameter is entered in KY format as 2-byte number, values from “0,0” ... “0,255” are allowed. The high-byte must always be “0”. The inerface number is entered in the low-byte. If several CP524 are to be used then the FB230:CP524SYN in the start-up program OB20-OB22 must be called repeatedly, for that different interface numbers SSNR and data clock numbers are to be specified.

DBNR:
Number of the data block in which all parameters and internal data for the start-up and the cyclic operation are deposited. The user must make certain that the block with a length of 150 bytes is already in the AG at the first start-up. The parameter is entered in KY format as 2-byte number, values from “0,10” ... “0,255” are allowed. The high byte must always be “0”. The data block number is entered in the low-byte, it can be in the range from DB10 to DB255. If several CP524 are used in one control then the FB230:CP524SYN must be called repeatedly. For each CP524 a data block must be set up (see above).

EDBN:
Binary output signal of the FB230:CP524SYN. It indicates an error in the parameterizing of the DBNR when the permissible value range was exceeded or fallen below.

EDBA:
Binary output signal of the FB230:CP524SYN. It indicates a missing DBNR. The reason for it is that for a AG S5-115U or an unidentified CPU type (then the bit ECPU is also set) the parameterized DBNR could not be found in the memory of the AG. For the AG S5-115U the user must create the block with a length of 150 data words by hand and then the start-up must be carried out again.

ECPU:
Binary output signal of the FB230:CP524SYN. It indicates that the function block of the CPU type could not be identified. The block recognizes the CPU 941, 941B, 942, 942B, 943, 943B, 944, 944B, 945 in the AG S5-115U and the CPU928 and 928B in the AG S5-135U. The block is executable only with these CPUs.

ERR:
Binary output signal of the FB230:CP524SYN that is to be classified as general fault bit. It is set at each of the above listed errors. Furthermore, it has the signal status “1” when at least one of the data handling blocks in the AG S5-135U is not loaded in the AG or when an error occurred during the synchronization process. With regard to the two last mentioned cases the data word 1 (listing of the not loaded data handling blocks) and data word 3 (error description of the synchronization block) are to be evaluated for precise error specification.

CPUN:
Input parameter of the FB230:CP524SYN. With it the number of the CPU must be specified for the AG S5-135U in multiprocessor operation with which the CP524 must be synchronized. In all other cases parameterize a “0” here.

2.2.2 Assigned Temporary Flags of FB230:CP524SYN

MW254	system data word BS41
MB255	for AG S5-115U: first digit of CPU label
MW252	system data word BS42
MB253	for AG S5-115U: thrid digit of CPU label
MB252	for AG S5-115U: second digit of CPU label
MW250	system data word BS43
MB250	for AG S5-115U: identifier character of CPU (e.g. “b”)
MW248	buffer of status word (see DBNR/DW1)
MW246	system data word BS29
MB247	for AG S5-135U: label AG-Typ
MB245	for AG S5-135U: buffer CPU label
MB244	is not being changed
MW242	DBNR filing
MW240	address of data block DBNR in the AG memory
MW238	CPU label
MW236	is not being changed
MW234	parameterizing error PAFE of HTB-Synchron (explanation see DBNR/DW3)
MB235	PAFE for AG S5-135U
MB234	PAFE for AG S5-115U
MW232	DBNR in the high byte

2.2.3 Entries of FB230:CP524SYN in the Data Block DBNR

DW1 Status word of the starting module

Bit 0	: AG-Typ is S5-115U
Bit 1	: AG-Typ is S5-135U
Bit 2	: no CPU type identified
Bit 3	: data handling block FB120 is not loaded in the AG
Bit 4	: data handling block FB126 is not loaded in the AG
Bit 5	: data handling block FB127 is not loaded in the AG
Bit 6	: data handling block FB125 is not loaded in the AG
Bit 7	: HTB:SYNCHRON was successful
Bit 8	: temporary flag for pulse generation from Bit 7 (is written on by FB231:CP524KEB)
Bit 9	: pulse from Bit 7 (is written on by FB231:CP524KEB)
Bit 10	: triggering of INIT function (is written on by FB231:CP524KEB)
Bit 11	: INIT function was successful (is written on by FB231:CP524KEB)
Bit 12	: free
Bit 13	: free
Bit 14	: free
Bit 15	: free

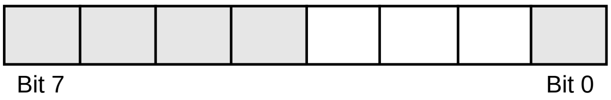
DW2 CPU label (e.g. “943B”$_{hex}$); in the event of unidentified CPU type “FFFF”$_{hex}$ is entered here

DW3 parameterizing error of synchronization block HTB:SYNCHRON (for AG S5-115U FB249; for AG S5-135U FB125)

DL3 PAFE of FB249:SYNCHRON is entered in the left byte of the DW3 if the block runs in a AG S5-115U

DR3 PAFE of FB125:SYNCHRON is entered in the right byte of the DW3 if the block runs in a AG S5-135U

2.2.4 Explanation of Error Bytes PAFE of Data Handling Blocks



Bit 0 : “0” = no error; “1” = error occurred (explanation in Bit 4-7)
Bit 1-3 : without significance
Bit 4-7 : hexadecimal error code according to the table below

Hex value in Bit 4-7	Meaning
0	no error
1	source/target parameter faulty (QTYP/ZTYP, DBNR, QANF/ZANF or QLAE/ZLAE) or pointer to source/target parameter faulty (at QTYP/ZTYP XX and RW) or source/target parameter of CP faulty
2	range does not exist (DB does not exist or inadmissible)
3	range too small or sum of starting address and length too large
4	QVZ error / no access possible since the range does not exist
5	wrong indicator word (address)
6	no source or target parameter at SEND/RECEIVE ALL (system error) ONLY AT AG S5-115U !!!
7	no interface
8	interface unclear
9	interface overloaded
A	interface occupied by another processor (only at multiprocessor operation) ONLY AT AG S5-135U !!!
B	inadmissible A-NR or inadmissible block size (at HTB:SYNCHRON)
C	interface (CP) does not acknowledge or gives negative acknowledgement
D	parameter / BLGR inadmissible (1.Byte)
E	other errors in data handling block, that includes • no data block opened at indirect parameterizing of parameters SSNR, A-NR, ANZW, BLGR. • software error of processor or data handling block
F	FB call up inadmissible, that includes • double call at possible interruption at instruction limits • inadmissible alteration of the standard function module

2.3 Function Module FB231:CP524KEB for Cyclic Operation

This function module represents the interface between user program and CP524. It is called absolute in the periodic program and takes over the initialization of the driver loaded on the CP524 after the start-up of the PLC and the successful synchronization of the interface between CPU and CP524.

For each inserted and supplied CP524 the FB231:CP524KEB must be called once in the periodic program, at that the data block belonging to the corresponding CP524 must be passed on as parameter DBNR. This DB and the interface number of the CP524 belonging to it has been parameterized and adjusted by the start-up block FB230:CP524SYN.

The FB231:CP524KEB supports the KEB unit function status enquiry, parameter read and parameter write and in each case it can execute one of these orders. The start of the function is triggered by the user by setting the corresponding bits in the order code. For that the required data must be handed over to the block parameters. From the beginning of the setting of the start bit the entire process is executed in the FB, without any action on the part of the user; upon completion of the function an acknowledge bit is set in the status byte of the FB for successful or unsuccessful communication with the corresponding bus user.
After successful status enquiry or successful writing of a parameter the parameters PARA and DATA are cancelled; when parameter reading was successful then PARA contains the address of the KEB parameter which was transmitted from the unit and DATA contains the received parameter value. In the event of an error an error specification is entered in PARA that is to be evaluated by the user: DATA may contain helpful information about the error (see below).

ATTENTION:
The DBNR of the data block to be used must be entered by the user at several marked positions in the function module FB231:CP524KEB. This is necessary as some of the data handling blocks cannot be parameterized directly.
However, this intervention into the FB231:CP524KEB is only necessary

a) when at a CP524 to be used the data block number DB230 is occupied by other user blocks respectively
b) when several CP524 are to be used the FB231:CP524KEB must be copied to other block numbers and then proceed in the first FB231 as stated under a), in the other new blocks CP524KEB the block number of the DBNR to be manipulated must be entered.

2.3.1 Module Parameter of FB231:CP524KEB

DBNR:
Specification of the data block in which the data belonging to the CP524, that is to be supplied, has been deposited by the start-up function module FB230:CP524SYN

CODE:
In this order code the user sets the bit for the function to be executed:

Bit 0	: status enquiry of user
Bit 1	: reading the parameter of the user
Bit 2	: writing on the parameter of the user
Bit 3-6	: free
Bit 7	: clear error counter

Please note that the block with the error message (bit 4 in ZUST) aborts if more than one order bit (bit 0 - 2) is set simultaneously. When an order was carried out correctly then the order bit will be cleared after the transmission and receiving phase. At the same time the bit 6 (communication successful) is set in ZUST. After this bit has taken on the status “1” the user can send the next order by writing of SLAD, PARA and DATA as well as by setting the order bit in CODE.

In the event that an error occurred during the communication the order bit remains set and in ZUST the bit 7 (communication finished with error) is set. Now the user must first clear all order bits in CODE, after that the block reports again with bit 6 (communication successful). The same procedure is necessary when the error bit 4 (see above) was set.

SLAD:
Slave address of the user from which the status or a parameter shall be interrogated or whose parameter shall be written. This parameter can be preset as binary-coded number. In FB231:CP524KEB the FB232:HEX>ASCI is called which converts the number into ASCII characters, so that these data will be accepted by the KEB unit.

PARA:
Specification of parameter address which shall be interrogated or written. Upon successful status enquiry or successful parameter writing PARA is cleared; if an error occurred at one of the possible functions then a general error information is given that informs the user at which function the error occurred. Under certain circumstances the value contained in DATA should be investigated more closely in order to better localize the error source.
In the event of an error the value “FF”$_{hex}$ stands in the left byte and a hexadecimal-coded error mesage stands in the right byte:

10_{hex} error at status enquiry (slave has not answered with “ACK”)
Then the characters transmitted from the KEB unit are contained in DATA.
21_{hex} error at parameter read (slave answer does not begin with “STX”)
Then the characters transmitted from the KEB unit are contained in DATA.
22_{hex} error at parameter read (error in check total BCC)
31_{hex} error at parameter write (slave has not answered with “ACK”)
Then the characters transmitted from the KEB unit are contained in DATA.

ZUST:

Output of the status byte of the function module in which, if necessary or desired, the current operating condition of the function module FB231:CP524KEB can be interrogated.

Bit 0: status enquiry runs (internal BUSY-Bit)
Bit 1: parameter reading runs (internal BUSY-Bit)
Bit 2: parameter writing runs (internal BUSY-Bit)
Bit 3: free
Bit 4: error bit: More than one order is active
Bit 5: SYNCHRON and INIT were successful (order bits can be set)
Bit 6: communication finished without error
Bit 7: communication finished with error

If an order was finished with error (Bit 7 = “7”), because the CP524 has determined an error, the indicator word ANZW of the corresponding SEND order can be evaluated:

Status enquiry with error:	DW74
Parameter reading with error:	DW94
Parameter writing with error:	DW114

Bit 0: without significance
Bit 1: order runs
Bit 2: order completed without error
Bit 3: order completed with error
Bit 8-11: error code

The hexadecimal coded error numbers in Bit 8-11 have following meaning:

Bit 8-11	Meaning
1..5	the CPU of the AG identifies parameterizing errors that are reported to the CP
6	CP identifies error in the data transmission between CPU and CP
7	order not executable, error in the order parameter
8, 9	not occupied
A	error at RECEIVE-ALL
B	not occupied
C	no answer from the slave within the answering surveillance time
D	switch at CP in position STOP / PGR
E	error in answering telegram of the slave
F	irrelevant at RS485

DATA:

On successful reading of a parameter the read value is entered here. When writing a parameter the user must enter the parameter value here. The function module converts the value into the ASCII representation required for transmission (when reading a parameter the ASCII characters transmitted from the KEB unit are converted into readable representations and outputted in DATA). If an error occurs at one of the possible functions, a more precise error specification, if possible, is put out here, which gives the user the chance to localize the error exactly.

2.3.2 Assigned Temporary Flags of FB231:CP524KEB

MW254	status word starting block
MW252	parameterizing error PAFE of “INIT” function
MB253	parameterizing error PAFE of “INIT” function at AG S5-135U
MB252	parameterizing error PAFE of “INIT” function at AG S5-115U
MW250	indicator word ANZW of “INIT” function
MW248	parameterizing error PAFE of “SEND-ALL” function
MB249	parameterizing error PAFE of “SEND-ALL” function at AG S5-135U
MB248	parameterizing error PAFE of “SEND-ALL” function at AG S5-115U
MW246	parameterizing error PAFE of “RECEIVE-ALL” function
MB247	parameterizing error PAFE of “RECEIVE-ALL” function at AG S5-135U
MB246	parameterizing error PAFE of “RECEIVE-ALL” function at AG S5-115U
MB245	temporary buffer of parameter CODE
MB244	temporary buffer of parameter SLAD
MW242	temporary buffer of parameter PARA
MB241	temporary buffer of parameter ZUST
MW238	temporary buffer of parameter DATA
MW236	parameterizing error PAFE of “status enquiry” function
MB237	parameterizing error PAFE of “status enquiry” function at AG S5-135U
MB236	parameterizing error PAFE of “status enquiry” function at AG S5-115U
MW234	parameterizing error PAFE of “parameter read” function
MB235	parameterizing error PAFE of “parameter read” function at AG S5-135U
MB234	parameterizing error PAFE of “parameter read” function at AG S5-115U
MW232	parameterizing error PAFE of “parameter write” function
MB233	parameterizing error PAFE of “parameter write” function at AG S5-135U
MB232	parameterizing error PAFE of “parameter write” function at AG S5-115U
MW230	status word CP operation
M231.0	starting pulse status enquiry
M231.1	auxiliary flag for M231.0
M231.2	start status enquiry
M231.3	starting pulse parameter reading
M231.4	auxiliary flag for M231.3
M231.5	start parameter reading
M231.6	starting pulse parameter writing
M231.7	auxiliary flag für M231.6
M230.0	start parameter reading
M230.1	starting pulse error at status enquiry
M230.2	auxiliary flag for M230.1
M230.3	starting pulse error at parameter reading
M230.4	auxiliary flag for M230.3
M230.5	starting pulse error at parameter writing
M230.6	auxiliary flag for M230.5
M230.7	error occurred in answering telegram of the KEB unit (e.g. BCC error)
MW228	indicator word ANZW for status enquiry
MW226	indicator word ANZW for parameter reading
MW224	indicator word ANZW for parameter writing
MW222	Slave address to be addressed in ASCII format
MW220	High-Word parameter address to be addressed in ASCII format
MW218	Low-Word parameter address to be addressed in ASCII format
MW216	High-Word parameter value to be transmitted in ASCII format
MW214	Low-Word parameter value to be transmitted in ASCII format

2.3.3 Entries of FB231:CP524KEB in the Data Block DBNR

DW4	status word CP operation
DW5	counter: error occurred at status enquiry
DW6	counter: error occurred at parameter reading
DW7	counter: error occurred at parameter writing
DW8	counter: BCC error occurred
DW9	counter: CP524 signals error during transmission (e.g. participant does not answer)
DW20	parameter filing SSNR for “INIT” function
DW21	parameter filing A-NR for “INIT” function
DW22	deposit parameter filing ANZW for “INIT” function in DB
DW23	deposit parameter filing ANZW for “INIT” function in DB/DW
DW24	indicator word ANZW for “INIT” function
DW25	word length LAEW for “INIT” function
DW26	driver parameter EMPFDB/EMPFDW for “INIT” function
DW27	driver parameter ECPUNR/EKENN for “INIT” function
DW28	driver parameter ANTTIM/RS485 for “INIT” function
DW29	driver parameter GERADE/ENDEKEN for “INIT” function
DW30	driver parameter ENDEZ1/ENDEZ2 for “INIT” function
DW31	driver parameter TELLEN for “INIT” function
DW32	driver parameter ZVZTIM for “INIT” function
DW33	parameterizing error PAFE for “INIT” function (compare MW252)
DL33	for AG S5-135U PAFE is entered in the left data byte
DR33	for AG S5-115U PAFE is entered in the right data byte
DW40	parameter filing SSNR for “SEND-ALL” function
DW41	parameter filing A-NR for “SEND-ALL” function
DW42	deposit parameter filing ANZW for “SEND-ALL” function in DB
DW43	deposit parameter filing ANZW for “SEND-ALL” function in DB/DW
DW44	indicator word ANZW for “SEND-ALL” function
DW45	word length LAEW for “SEND-ALL” function
DW46	parameterizing error PAFE for “SEND-ALL” function (compare MW252)
DW50	parameter filing SSNR for “RECEIVE-ALL” function
DW51	parameter filing A-NR for “RECEIVE-ALL” function
DW52	deposit parameter filing ANZW for “RECEIVE-ALL” function in DB
DW53	deposit parameter filing ANZW for “RECEIVE-ALL”- function in DB/DW
DW54	indicator word ANZW for “RECEIVE-ALL” function
DW55	word length LAEW for “RECEIVE-ALL” function
DW56	parameterizing error PAFE for “RECEIVE-ALL” function (compare MW252)
DW60- DW69	receive range for RECEIVE-ALL (receive data from CP524)
DW70	parameter filing SSNR for “STATUS ENQUIRY” function
DW71	parameter filingA-NR for “STATUS ENQUIRY” function
DW72	deposit parameter fiing ANZW for “STATUS ENQUIRY” function in DB
DW73	deposit parameter filing ANZW for “STATUS ENQUIRY” function in DB/DW
DW74	indicator word ANZW for “STATUS ENQUIRY” function
DW75	word length LAEW for “STATUS ENQUIRY” function
DW76	parameterizing error PAFE for “STATUS ENQUIRY” function (compare MW252)
DW80	transmission telegram EOT / ADR High-Byte for status enquiry
DW81	transmission telegram ADR Low-Byte / ENQ for status enquiry

DW90	parameter filing SSNR for “PARAMETER READ” function
DW91	parameter filing A-NR for “PARAMETER READ” function
DW92	deposit parameter filing ANZW for “PARAMETER READ” function in DB
DW93	deposit parameter filing ANZW for “PARAMETER READ” function in DB/DW
DW94	indicator word ANZW for “PARAMETER READ” function
DW95	word length LAEW for “PARAMETER READ” function
DW96	parameterizing error PAFE for “PARAMETER READ” function (compare MW252)

DW100	transmission telegram EOT / ADR High-Byte for parameter read
DW101	transmission telegram ADR Low-Byte / CMD1 for parameter read
DW102	transmission telegram CMD2 / CMD3 for parameter read
DW103	transmission telegram CMD4 / ENQ for parameter read

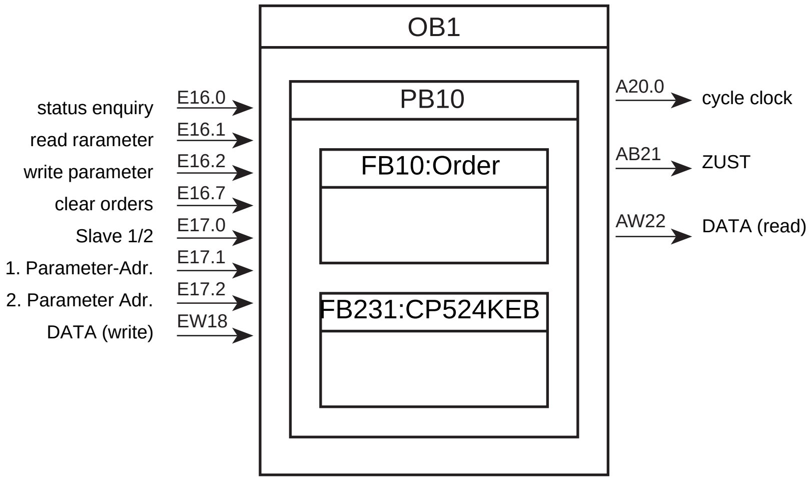
DW110	parameter filing SSNR for “PARAMETER WRITE” function
DW111	parameter filing A-NR for “PARAMETER WRITE” function
DW112	deposit parameter filing ANZW für “PARAMETER WRITE”function in DB
DW113	deposit parameter filing ANZW for “PARAMETER WRITE” function in DB/DW
DW114	indicator word ANZW for “PARAMETER WRITE” function
DW115	word length LAEW for “PARAMETER WRITE” function
DW116	parameterizing error PAFE for “PARAMETER WRITE” function (compare MW252)

DW120	transmission telegram EOT / ADR High-Byte for parameter write
DW121	transmission telegram ADR Low-Byte / STX for parameter write
DW122	transmission telegram CMD1 / CMD2 for parameter write
DW123	transmission telegram CMD3 / CMD4 for parameter write
DW124	transmission telegram DATA 1 / DATA 2 for parameter write
DW125	transmission telegram DATA 3 / DATA 4 for parameter write
DW126	transmission telegram ETX / BCC for parameter write

3. The Example Application

The present S5 software package contains an example application which is realized in the modules **PB10** and **FB10:ORDER**. Digital inputs/outputs of the PLC form the interface of the user for this example application. By way of inputs it is possible to initiate either a **status enquiry**, the **reading of a parameter** or the **writing of a parameter** either to the inverter with the **slave address 1** or **2**. Two more inputs changeover between two parameter addresses. The selection which of these two parameters this will be is adjusted absolute in **FB10**. For writing a parameter the value is taken from an input word.

An output byte (status byte) informs about success or failure of the communcations order. In the event of the successful reading of a parameter the read value is contained in another output word. The general program structure (cyclic part) presents itself as follows:



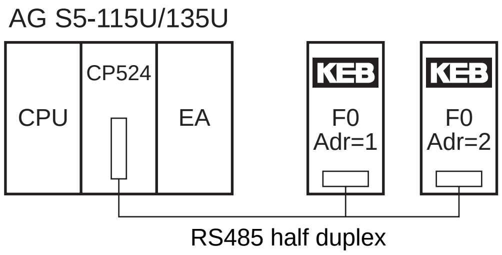
The internal interface is realized on the following flags:

M0.0 : EDBN/FB230	: 1 : error in DBNR programming
M0.1 : EDBA/FB230	: 1 : parameterized DBNR does not exist
M0.2 : ECPU/FB230	: 1 : CPU label not recognized
M0.3 : ERR/FB230	: 1 : general group fault (ERR = (EDBN OR EDBA OR ECPU)).
MB10 : CODE/FB231	: order code :
M10.0	: start status enquiry
M10.1	: start parameter read
M10.2	: start parameter write
M10.3	: not assigned
“	: “ “
M10.6	: not assigned
M10.7	: clear error counter
MB11 : SLAD/FB231	: slave address
MW12 : PARA/FB231	: parameter address
MW14 : DATA/FB231	: parameter value
MB16 : ZUST/FB231	: status byte for communications orders
M16.0	: status enquiry runs
M16.1	: parameter read runs
M16.2	: parameter write runs
M16.3	: not assigned
M16.4	: error occurred, more than one order active
M16.5	: CP524 ready
M16.6	: communication finished **WITHOUT ERROR**
M16.7	: communication finished **WITH ERROR**
MB166 :	: auxiliary flag

3.1 Commissioning of Example Application

3.1.1 General Structure

The following picture shows the basic structure of the test application:



The example application can be used without any change if in the AG the **CP524** is inserted in slot **0**, a **32-bit input module** is inserted in **slot 4** and a **32-bit output module** is inserted in slot **5**.
Other system structures require slight changes in the program. With regard to wiring refer to the instruction manual of the **CP524** and of the **F0 frequency inverter**. For half duplex wiring make certain that the jumper **X3** of the **CP524 module** is inserted in position **2-3**. The example application requires that the **CP524** can be addressed under interface number **200 (dec)**. This is indicated by insertion of the bridges **1–16**, **2–15**, **5–12** in the plug-in station **J53** of the hardware.

3.1.2 Test of the Example Appliation

Preparation:
First of all make sure that an EPROM module with the special driver **6ES5 897 2MB11** is inserted in the **CP524.**
Load the complete program **KEB485ST.S5D** from the supplied disk into the **AG**. **CP524** and **CPU** must be switched into condition **RUN**. After a faultless start-up the green **RUN-LED** lights up at the **CPU** and the two red control lights **IF1** and **IF2** of **CP524** do not light nor do they blink.

For debug purposes the following variables should be indicated on **PG** in the menu **Control Variable**:

– **MB10**	: order code (see above.)
– **MB11**	: slave address (see above.)
– **MW12**	: parameter address
– **MW14**	: parameter value
– **MB16**	: status byte (see above)

Starting Point:

E16.0,E16.1,E16.2 = 0 **==>** no communication orders initiated
E16.7 = 0 **==>** no reset of error counters
E17.0 = 0 **==>** slave address **1** is selected
E17.1 = 1,E17.2 = 0 **==>** 1. parameter address is selected (here **0201 (hex)**).
For a single software reset the input **E16.7** can be set and reset once.

Test of Communication Service Status Enquiry:

E16.0 = 1 **==>** status enquiry requested
If **M16.6 = 1** and **M16.7 = 0** then the service could be executed without error.
If **M16.6 = 0** and **M16.7 = 1** then an error occurred. In this case the **DW74** of **DBNR (**here **DB230)** provides more information about the error.
E16.0 = 0 **==>** resetting of the service request in order to prepare the next service
In the event of an error the input **E16.7** must be additionally set and reset again once.

Test of Communication Service Parameter Read:

E17.0 = 0 ==> inverter **1** selected
E17.1 = 1, E17.2 = 0 ==> 1. parameter address selected (here **0201 (hex) : Inverter-Status**).
E16.1 = 1 ==> parameter read requested
If **M16.6 = 1** and **M16.7 = 0** then the service could be executed without error. In this case the read value of the parameter (**DATA**) can be taken from the **AW22** or the **MW14**.
If **M16.6 = 0** and **M16.7 = 1** then an error occurred. In this case the **DW94** of **DBNR (**here **DB230)** provides more information about the error.
E16.1 = 0 **==>** resetting of the service request in order to prepare the next service
In the event of an error the input **E16.7** must be additionally set and reset again once.

Test of Communication Service Parameter Write:

E17.0 = 0 ==> inverter **1** selected.
E17.1 = 0, E17.2 = 1 ==> 2. parameter address selected (here **0213 (hex) : Frequency-Reference Value**).
EW18 = 0FA0h ==> parameter value = **4000 = 50 Hz.**
E16.2 = 1 ==> parameter write requested
If **M16.6 = 1** and **M16.7 = 0** then the service could be executed without error. In this case the value can be checked on the frequency inverter under parameter **r.19**.
If **M16.6 = 0** and **M16.7 = 1** then an error occurred. In this case the **DW114** of **DBNR (**here **DB230)** provides more information about the error.
E16.2 = 0 **==>** resetting of the service request in order to prepare the next service
In the event of an error the input **E16.7** must be additionally set and reset again once.

KEB
ANTRIEBSTECHNIK

Karl E. Brinkmann GmbH
Postfach 11 09 • Försterweg 36 - 38
D - 32677 Barntrup • Telefon 0 52 63 / 4 01-0
Teletex 5 263 811 keb • Telefax 4 01 - 116