AMK AMKASYN

VARIABLE SPEED DRIVES

AMKASYN

Programmable Controller PS

AMK Digital Parallel Interface (ADPS)

Version AZ-PSx V2.06e

Subject to change without prior notice

2295

AMK Arnold Müller, Antriebs- und Steuerungstechnik GmbH & Co. KG, D-73230 Kirchheim/Teck,
Tel.: 07021/5005-0, Telefax: 07021/5005-176

TABLE OF CONTENTS

Used abbreviations:

AZ-PSx	AMKASYN optional card for implementation of the PS function
ADB	Output data byte of the higher ranking system
ADPS	AMKASYN Digital Parallel Interface
!AFehl	Output signal error of the higher ranking system
AInfo	Output user information of the higher ranking system
ASB	Output control byte of the higher ranking system
AStb	Output signal strobe of the higher ranking system
AW	AMKASYN inverter module
DB	Data block
E/A/M	Input/Output/Flag (I/O/F)
ED	Input double word
EDB	Input data byte of the higher ranking system
!EFehl	Input signal error of the higher ranking system
EInfo	Input error information of the higher ranking system
ESB	Input control byte of the higher ranking system
EStb	Input signal strobe of the higher ranking system
FB	Function block
F-Code	Function code
ID	Parameter of the basic system
LB	Less significant byte
HB	More significant byte
LW	Less significant word
HW	More significant word
PS	Programmable Controller
!xxx	Negated signal xxx
0x..	Hexadecimal notation

1 Introduction

The AMKASYN DIGITAL PARALLEL INTERFACE (ADPS) serves the standardized access to the AMKASYN drive concept which supports the AMK specific features. This description of the interface includes the definition of the physical instance of the information exchange protocol as well as the structure of the transmitted data packets and their information contents.

2 Physical instance of the ADPS interface

The ADPS interface is based on binary input and output signals (16 input and 16 output signals). Because of the interface width the transmission information which is divided into 2 byte blocks is transmitted via ADPS. There is only one ADPS interface per AMKASYN system.

The input and output signal leads consist of the following signal groups (cf. diagram 1):

Input signals: - Control byte (ASB[1)])
- Data byte (ADB[1)]))

Output signals: - Control byte (ESB[1)])
- Data byte (EDB[1)]))

1) "Input or output" refers to the higher ranking system.

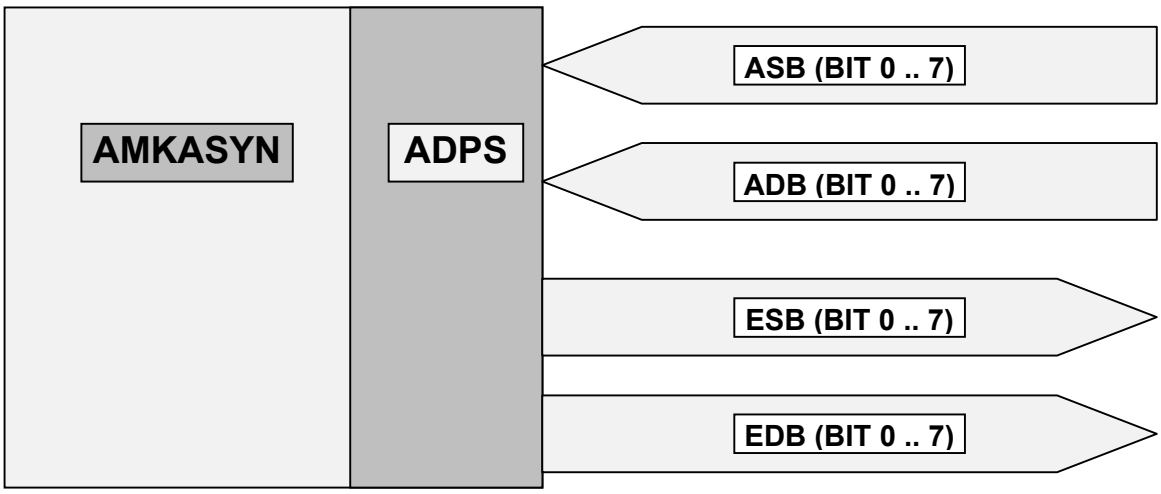

Diagram 1: ADPS interface scheme

The structure of the single signal groups is defined as follows:

- Output control byte ASB (control byte output by the higher ranking system):

Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
AStb	!AFehl	f2	f1	f0	a2	a1	a0
					AW address/ extended function code		
		Function code					
	Error output (! = active low)						
Strobe output							

- Output data byte ADB (data byte output by the higher ranking system):

Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
d7	d6	d5	d4	d3	d2	d1	d0
Data byte							

- Input control byte ESB (control byte which the higher ranking system receives from the AMKASYN system):

Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
EStb	!EFehl	f2	f1	f0	a2	a1	a0
					AW address/ extenden function code		
		Function code					
	Error input (! = active low)						
Strobe input							

- Input data byte EDB (data byte which the higher ranking system receives from the AMKASYN sytem):

Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
d7	d6	d5	d4	d3	d2	d1	d0
Data byte							

3 Information exchange protocols in the scope of elementary cycles

The information exchange protocol in the scope of elementary cycles consists of:
- **Write cycles** for transmission of information from a higher ranking system to the AMKASYN system and
- **Read cycles** for transmission of information from the AMKASYN system to a higher ranking system.

The AMKASYN system only starts a read cycle if it was arranged for from the higher ranking system by a write cycle with a control code for e.g. a read command.

In addition it is distinguished between
- Strobe and error signals for control of the information exchange in the most significant bits (bit 6 and 7) of the control bytes (ESB/ASB) and
- the user information to be transmitted in the remaining bits of the control bytes (bit 0 to 5) and the data bytes (EDB/ADB).

The strobe signal characterizes a transmission function. At the beginning of a transmission function it is set active. All other signals of the control byte and the entire data byte must be valid at this time. At the end of a transmission cycle the strobe signal becomes inactive and thus characterizes the remaining signal leads as being invalid.

The error signal (active low) characterizes a faulty transmission function in case of which send data and echo differ. If the transmitter recognizes such a fault he must activate the error signal and wait for the receiver to activate his error signal for handshake, too.

Attention: The error signal (!EFehl or !AFehl) is defined active low, i.e. a physical "0" level signalizes logically an error.

After switching on the AMKASYN system the error signal becomes inactive (physically "1" level) as soon as the ADPS interface is ready. Only afterwards may the higher ranking system start the first transmission cycle. In case the error signal is activated by the AMKASYN system while no transmission cycle is active, an error status in the AMKASYN system is signalled (caused by e.g. an unpermissible value input).

In this chapter firstly the temporal sequence of the information exchange for the elementary write and read cycle is shown. The contents to be transmitted as user information as well as the sequence to be maintained for the transmission protocols composed of several write and read cycles will be treated in chapter 4.

3.1 Description of a write cycle

A write cycle (cf. diagram 2) is initiated by the higher ranking system and is executed in principle as follows:

The transmitter puts his user information out to the outputs and characterizes them as being valid after a minimum waiting time t_0 by an edge change of the strobe signal. These signals are read in by the AMKASYN system. If configured „with echo“ (cf. paragraph 5) the user information is written back for error recognition as an echo to the inputs of the higher ranking system. It compares the echo with his send data. In case they comply, the transmission is

finished by a "0" level at the strobe signal. If send data and echo differ the error signal must be activated (!AFehl = "0"). With faulty transmissions the AMKASYN system rejects the receive data.

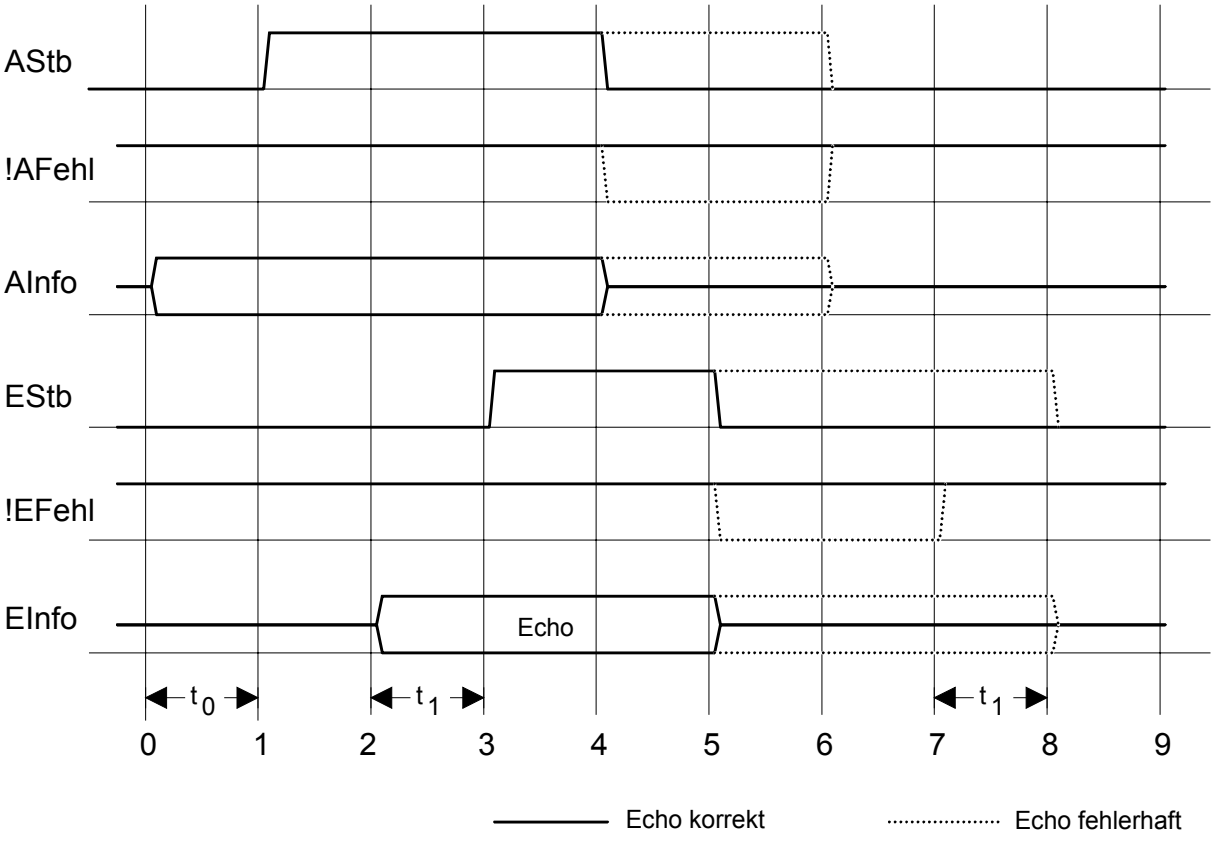

Diagram 2: Write cycle

Instant 0 (basic status):

Input signals [1] : Strobe (EStb) = "0", error (!EFehl) = "1", input information (EInfo) in any ESB (without EStb and !EFehl) and EDB

Output signals [1] : Strobe (AStb) = "0", error (!AFehl) = "1", output information (AInfo) in any ASB (without AStb and !AFehl) and ADB

1) "Input or output" refers to the higher ranking system.

The basic status is recognized by the higher ranking system and the output information is injected.

Instant 1:

The output information is adjacent (the signal AStb may at the earliest be adjacent simultaneously with the remaining output information; $t_0 \geq 0$).

Instant 2:

Signal AStb = "1" is recognized. After a waiting time the output information is read in and retransmitted to inputs of the higher ranking system in the form of an echo. If configured „without echo" the retransmission of the input information does not apply.

Instant 3:
After a waiting time ($t_1 \geq 0$) signal EStb is set. With that the echo is valid or, if configured „without echo“, the data reception is characterized.

Instant 4:
The higher ranking system recognizes EStb = "1". If configured „with echo“, it is now valid and can be compared with the sent information. If no error is recognized the write cycle was successful and the output information (ASB and ADB) is taken away. This is obvious through AStb = "0". Without echo, signal Astb = „0“ can be set immediately by the higher ranking control.

If in contrary an error was recognized, the signal !AFehl = "0" is set by the higher ranking control.

Instant 5:
The AMKASYN system recognizes AStb = "0". Then the input information is taken away and this is obvious through EStb = "0". If in contrary signal !AFehl = "0" is recognized, this means that a transmission error has occured and that the user information is to be rejected. Then signal !EFehl = "0" is set.

Instant 6:
The higher ranking system recognizes !EFehl = "0" and then sets signal !AFehl = "1" and ends the transmission by resetting the strobe signals (AStb = "0"). At the same time the output information (ASB and ADB) can be taken away.

Instant 7:
The AMKASYN system recognizes AStb = "0". After that signal !EFehl = "1" is set.

Instant 8:
After a waiting time ($t_1 \geq 0$) signal EStb and the output information (ASB and ADB) is deleted.

Instant 9:
The initial status is reached again.

3.2 Description of the read cycle

The read cycle (cf. diagram 3) follows a write cycle with a control code for a read cycle and is executed once or several times depending on the length of the information to read (cf. paragraph 4).

The AMKASYN system writes its user information to the inputs of the higher ranking system and characterizes them valid after a minimum waiting time t1 by an edge change of the strobe signal. These signals are read in by the higher ranking system. If configured „with echo“ the user information has to be retransmitted to the AMKASYN system for error recognition by means of the outputs in the form of an echo. It compares the echo with the send data. If they comply, the transmission is finished by a "0" level at the strobe signal. In case send data and echo differ, the error signal is activated (!EFehl = "0"). With faulty transmission the higher ranking system has to reject the receive data.

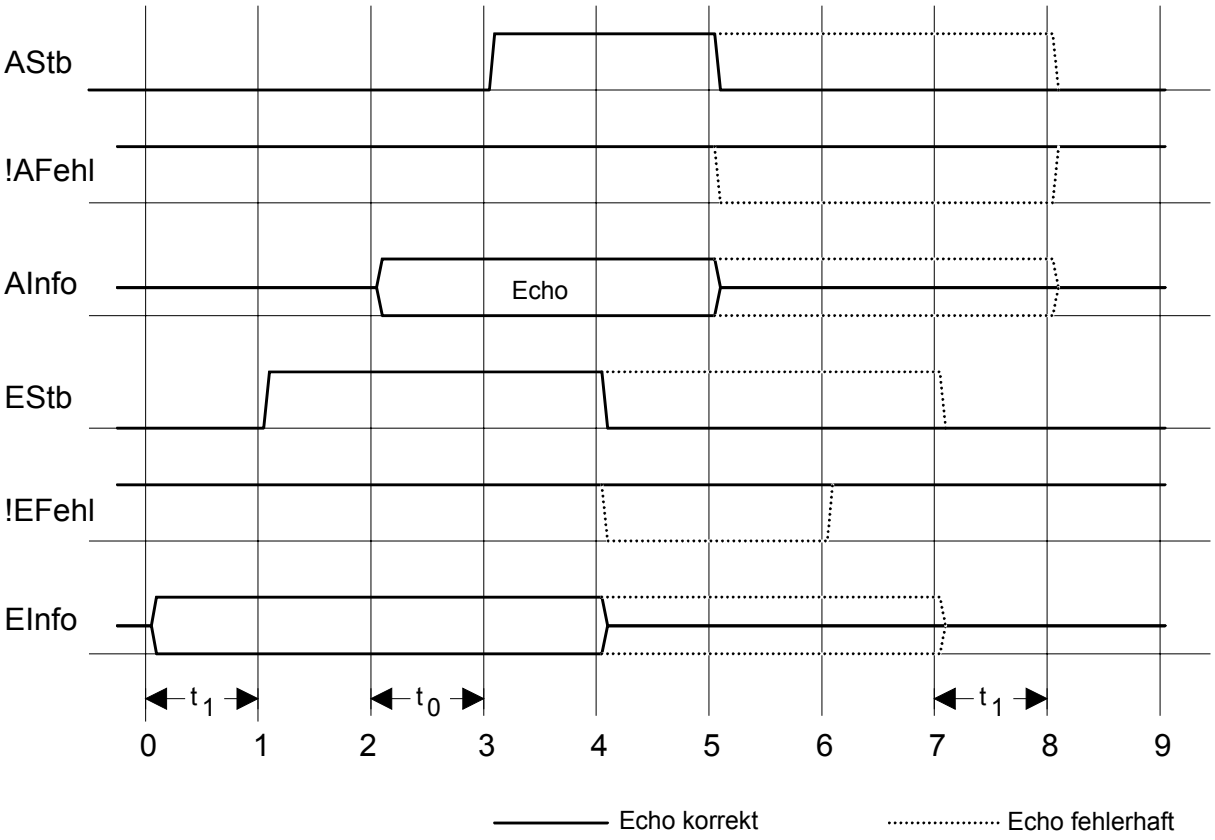

Diagram 3: Read cycle

Instant 0 (basic status):
Input signals [1] : Strobe (EStb) = "0", error (!EFehl) = "1", input information (EInfo) in any ESB (without EStb and !EFehl) and EDB
Output signals [1] : Strobe (AStb) = "0", error (!AFehl) = "1", output information (AInfo) in any ASB (without AStb and !AFehl) and ADB

1) "Input or output" refers to the higher ranking system.

After request of a read cycle by the higher ranking system and recognition of the basic status, the user information is put against the inputs of the higher ranking system by the AMKASYN system.

Instant 1:
After a waiting time ($t_1 \geq 0$) signal EStb = "1" is set.

Instant 2:
The higher ranking system recognizes EStb = "1", reads in the input information and transmits it by means of the outputs to the AMKASYN system as an echo. If configured „without echo" (cf. paragraph 5), the retransmission of the input information can be dropped.

Instant 3:
After a waiting time ($t_0 \geq 0$) the signal Astb is set by the higher ranking system. The echo is thus valid or the reception of the data is acknowledged.

Instant 4:
The AMKASYN system recognizes AStb = "1". This means that the echo, if configured „with echo“, is valid and can be compared with the sent information. If no error was recognized, the read cycle was successful and output information is taken away (ESB and EDB). This is obvious through EStb = "0". Without echo, Estb = “0“ can be set immediately.

If in contrary an error was recognized, signal !EFehl = "0" is set by the AMKASYN system.

Instant 5:
The higher ranking system recognizes EStb = "0". After that it takes away the output information and shows this by AStb = "0" . If in contrary the signal !EFehl = "0" is recognized, this means that a transmission error has taken place and that the input information is to be rejected. The higher ranking system then sets the signal !AFehl = "0".

Instant 6:
The AMKASYN system recognizes !AFehl = "0" and then sets signal !EFehl = "1".

Instant 7:
After a waiting time ($t_1 \geq 0$) the transmission is finished by resetting the strobe signal (EStb = "0"). At the same time the ouput information (ESB and EDB) is taken away.

Instant 8:
The higher ranking system recognizes EStb = "0". After that it must set the signal AStb = "0" and the signal !AFehl = "1". At the same time the output information (ASB and ADB) can be deleted.

Instant 9:
The initial state is reached again.

In case the error signal is activated by the AMKASYN system after read command, this signals an error in the AMKASYN system (e.g. as a sequence of a read command with unpermissible parameter). In this case no read information is transmitted to the higher ranking system.

4 Information contents and protocols, based on several elementary cycles

4.1 Overview of the protocols' structure

The information contents to be transmitted to and from the AMKASYN system (or the operations resulting from this) are defined according to the following function codes (F code; cf. ASB, bit 3 .. 5), the AW address (cf. ASB, bit 0 .. 2) and the parameter value in the output data byte (ADB) according to the following table 1.
The protocols are mostly composed of several elementary cycles. In the first cycle of a protocol the control byte contains the corresponding function code for a command and the data byte of the first parameter.
If in column "additional parameters in ADB" further parameters are indicated, these are transmitted in the scope of autarkic write cycle with the function code "write sequential value" (F code: 000, extended F code: 001) or "write last sequential error" (F code: 000, extended F code: 010). The AMKASYN system only starts with a resulting operation if all parameter bytes have completely been transmitted.

Exception:
The write cycle "write last sequential value" can be used in case the following values are to be supplemented automatically with "0", i.e. the protocol is finished with this transmission.

F code[1)]	AW addr. / exp. code	Significance	First para-meter byte	Addit. parameter bytes[2)]
000	000	I/O test	Value	
000	001	Write sequential value	Value	
000	010	Write last sequential value	Value	
000	011	Generate, select or delete[3)] DB	DB n°	Length in words (2 bytes)
000	100	Receive read information	Value	
000	101	Write remanent drive parameter	Parameter set number	2 byte ident. n°, 4 byte value
000	110	Read remanent drive parameter	Parameter set number	2 byte ident n°
000	111	Error deletion AZ-PSx; in case PS in error status		
001	nnn	Currently not used		
010	yyy[4)]	Write system info of category yyy	Info addr.	Value[5)]
011	yyy[4)]	Read system info of category yyy	Info addr.	Value[6)]
100	aaa	Call commanding with reg. to AW aaa	Comm. DB n°	
101	aaa	Change temporary drive parameters with reg. to AW aaa	Para. DB n°	
110	aaa	Currently not used		
111	aaa	Currently not used		

Table 1: Overview of the function codes

1) The F codes 000 .. 011 are extended by the extended F code, the F codes 100 .. 111 work AW-related.

2) If an additional parameter consists of several bytes the less significant byte is firstly transmitted (sequence LB/LW, HB/HW, LB/HW, HB/HW)

3) If the DB does not exist it is generated with the length of the parameter and selected with regard to ADPS.
If the DB exists and the length of the parameter value does not become zero the DB is selected with regard to ADPS.
If DB exists and the length of the parameter value is zero the DB is deleted.

4) Currently the following info categories are distinguished:
- 000: I/O info (byte)
- 001: M info (byte)
- 010: I/O info (word)
- 011: M info (word)
- 100: I/O info (double word)
- 101: M info (double word)¦
- 110: DB info (word)
- 111: Special info (info addr. = 1: user list 1)

5) One or several byte sequential values (depending on the info category):
- 00x: 1 Byte
- 01x: 2 Byte (LB, HB)
- 10x: 4 Byte (LB/LW, HB/LW, LB/HW, HB/HW)
- 110: 2 Byte (LB, HB)
- 111 and info addr. = 1: n byte (n = number of the user list1 bytes to be transmitted; corresponds to the current length of the list in the head info; the indicated length comprises 4 bytes head info)

6) Only necessary with info category 111 and info addr. = 1: n = number of the user list 1 bytes to be transmitted

Table 1 (continuation): Overview of the function codes

4.2 Description of the single function codes

I/O test
With protocols containing this function code, no operation is executed. The function code can be used for test of the transmission route.

Read sequential value, write last sequential value
These function codes consisting of several elementary cycles are used for characterization of additional parameter bytes (sequential values) (see previous paragraph).

Generate, select or delete DBs
Via the function „generate, select or delete DB“ (F code: 000; extended F code: 011) the data blocks can be generated, selected or deleted. It is distinguished as follows:

- if the DB does not exist, it is generated with the length of the parameter value and selected with regard to ADPS,
- if the DB exists and the length in the parameter value is not zero the DB is selected with regard to the ADPS,
- if the DB exists and the length of the parameter value is zero the DB is deleted.

Read remanent drive parameters, write remanent drive parameters:
Remanent drive parameters can be read with the command “read remanent drive parameters (F code: 000, extended F code: 110)“ and the data record number as a parameter. After the command the programmable controller (PS) sends back the 4 byte read information (F code: 000, extended F code: 100).

The writing of remanent drive parameters takes place with the command “write remanent drive parameters (F code: 000, extended F code: 101)“. The data record number must be transmitted as a first parameter byte and the Sercos ident. number (2 bytes) as well as the value (4 bytes) as sequential bytes.

Error deletion:
If the PS is in error status (!EFehl = “0“ and EStb = “0“) a resetting of the PS can take place with the command “error deletion“. At the same time the whole PS is reset and in case of AZ errors even the AZ computer. If an AZ error occurs, DC-Bus enable (UE) and all Inverters On (RF) must be inactive so that the command “error deletion“ can be executed correctly.

Note: The command “error deletion“ is always executed without echo independently of the parameterization (with/without echo). The command is only executed if the PS is in error status, otherwise it is ignored.

Read system info, write system info:
After a command "read system info of the category yyy" (F code: 011) the PS send the read information (F code: 000, extended F code: 100). The number of the read cycles depends on the info category indicated in the extended F code of the read command as well as possibly on the necessary sequential value (cf. table 1).
With the command “write system info of category yyy“ it can be written to outputs and flags. The number of the sequential cycles depends on the info category indicated in the extended F code of the write command as well as possibly on the necessary sequential value (cf. table 1).

Under the F codes "010" and "011" thus the complete I-/O-/F image of the PS is available; i.e. above all also the drive parameters in the I-/O-/F image (cf. documentation PS instruction set, paragraph 6.4: drive specific I-/O-/F image).

Note: The F image can be read and written, whereby the I image can only be read and the O image only be written.

Commanding drive:
For standardized commanding FB 220 provides already predefined data blocks (DB 1 .. DB 15) for the AMKASYN basic functions, each data block is then logically combined with a special drive function. Currently the functions are defined in table 2 (cf. paragraph 5).

A variation of the respective data block parameters (cf. documentation AMK specific function blocks, paragraph 2.2: FB201 commanding drive) can take place, through function "write system info of category yyy" (F code: 010; extended F code or info after selection of

category: 110; info addr. = DW address)", after selection of the corresponding commanding DBs via the function "generate, select or delete DB".

Comm. DB n°	Drive function
1	Operation mode change
2	Parameter set switching over
3	Digital torque control
4	Digital torque control
5	Homing
6	Spindle positioning
7	Absolute positioning
8	Relative positioning
9	Currently not used (reserve)
10	Synchronous control
11	Actual position value addition
12	Actual position value offset
13..15	Currently not used (reserve)

Table 2: Overview of the commanding DB

Change temporary drive parameters:
Temporary drive parameters can be written with the aid of the F code "101". For this a data block with the corresponding values (cf. documentation AMK specific function blocks, paragraph 2.3.1: FB202 write temporary drive parameters) must be available. This one can be generated or selected by means of the function "generate, select or delete DB" (F code: 000; extended F code: 011; value = 6[2)]) and can be changed by the function "write system info of category yyy" (F code: 010; extended F code or info category: 110; info addr. = DW address)".

2) FB202 expects an active DB with a length of 6 data words.

5 Initialization of the ADPS interface

The ADPS interface is initialized with the aid of the AMK specific function block FB220. The initialization data must be passed to this FB in active DB. The DB for the ADPS initialization must be 6 data words long.

The data words are defined as follows:

Data double word	High word High byte	High word Low byte	Low word High byte	Low word Low byte
0	Strobe low time	Mode Select	max. number of replications	reserve
2	A DB address	A SB address	E DB address	E SB address
4	reserve		Timeout for strobe and error	
6	*Double word sum*			

Max. number of replications (DL 0)
- 0: no replications in case of transmission errors
- 1..255: max. number of replications in case of transmission errors before a fatal error is generated

Mode select (DR 1)
- Bit0 = 0:Protocol with echo
- Bit0 = 1:Protocol without echo[1)]
- Bit1 = 0: without ready evaluation during drive commanding
- Bit1 = 1: with ready evaluation during drive commanding[2)]

1) The write/read cycles according to diagram 2 and 3 are reduced to the signal characteristic for "correct echo". The echo information, however, must not be generated or checked. The strobe signal necessary for the handshake (EStb during writing and AStb during reading) remains part of the signal characteristics.

2) By this option a renewed drive commanding with regard to this very drive is only executed if the drive status shows "ready" (cf. documentation AMK spec. function blocks: paragraph 2.1, FB200 "Update drive status").

Strobe low time (DL 1)
- 0..255 (ms): time (resolution 1 ms), for which the input strobe signal (EStb) is at least kept inactive (low) between the change from write to read cycle. [3)]

3) In order to be able to tell the transition from a write to a read cycle (in the scope of a request "read system info") the "Low" time of the Estb signal (during cycle change) may not become smaller than the time resolvable by the higher ranking system (e.g. limited by the cycle time of a higher ranking PLC).
Attention: During change over from a read to a write cycle (in the scope of a sequential command after a command „read system info" it must in addition be guaranteed a minimum "Low" time of the Astb signal (generated by the higher ranking system) of 4ms!

E SB address (DR 2)
- 0..31: Address of the E byte[4)] for the ADPS control word (ASB)

E DB address (DL 2)
- 0..31: Address of the E byte[4)] for the ADPS data word (ADB)

A SB address (DR 3)
- 0..31: Address of the A byte[4)] for the ADPS control word (ESB)

A DB address (DL 3)
- 0..31: Address of the A byte[4] for the ADPS data word (EDB)

4) I/O refer to the AMKASYN system.

Timeout for strobe and error (DW 4)
- 0: No timeout monitoring for the strobe and error signal
- 1..65535 (ms): Time (resolution 1 ms), in which the higher ranking control must have reacted to the changes of the strobe or error signal. After that a timeout error is generated.

Note: By initialization of the AMK digital parallel interface the commanding data blocks DB01 and DB15 of 16 words length and with the following structure are implicitely generated:

```
;DB01
;KMD DB for start of the operation mode change after HBA0;
;generated by FB220

   :KD 0       ;ANTRK      = 0   (AW1)
   :KB 1       ;KMDCODE    = 1   (Start)
   :KB 1       ;KMDFKT     = 1   (Operation mode change)
   :KB 0       ;KMDBA      = 0   (HBA0)
   :KB 0       ;KMDBIT     = 0   (Default ID values)
   :KD 0       ;KMD_VAR1   = 0   (2 Words Reserve)
   :KD 0       ;KMD_VAR2   = 0   (2 Words Reserve)
   :KD 0       ;KMD_VAR3   = 0   (2 Words Reserve)
   :KD 0       ;KMD_VAR4   = 0   (2 Words Reserve)
   :KD 0       ;KMD_VAR5   = 0   (2 Words Reserve)
   :KD 0       ;KMD_VAR6   = 0   (2 Words Reserve)


;DB02
;KMD DB for start of parameter set change over;
;generated by FB220

   :KD 0       ;ANTRK      = 0   (AW1)
   :KB 1       ;KMDCODE    = 1   (Start)
   :KB 2       ;KMDFKT     = 2   (Parameter set
               ;                  change over)
   :KB 0       ;KMDBA      = 0   (HBA0)
   :KB 0       ;KMDBIT     = 0   (Default ID values)
   :KD 0       ;KMD_VAR1   = 0   (2 Words Reserve)
   :KD 0       ;KMD_VAR2   = 0   (2 Words Reserve)
   :KD 0       ;KMD_VAR3   = 0   (2 Words Reserve)
   :KD 0       ;KMD_VAR4   = 0   (2 Words Reserve)
   :KD 0       ;KMD_VAR5   = 0   (2 Words Reserve)
   :KD 0       ;KMD_VAR6   = 0   (2 Words Reserve)
```

```
;KMD DB for start of dig. torque control;
;generated by FB220

   :KD 0        ;ANTRK      = 0   (AW1)
   :KB 1        ;KMDCODE    = 1   (Start)
   :KB 3        ;KMDFKT     = 3   (torque control)
   :KB 7        ;KMDBA      = 7   (NBA7)
   :KB 0        ;KMDBIT     = 0   (Default ID values)
   :KD 0        ;KMD_VAR1   = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR2   = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR3   = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR4   = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR5   = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR6   = 0   (2 Words Reserve)


;DB04
;KMD DB for start of dig. speed control;
;generated by FB220

   :KD 0        ;ANTRK      = 0   (AW1)
   :KB 1        ;KMDCODE    = 1   (Start)
   :KB 4        ;KMDFKT     = 4   (Speed control)
   :KB 9        ;KMDBA      = 9   (NBA9)
   :KB 0        ;KMDBIT     = 0   (Default ID values)
   :KD 0        ;KMD_VAR1   = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR2   = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR3   = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR4   = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR5   = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR6   = 0   (2 Words Reserve)


;DB05
;KMD DB for start of homing cycle;
;generated by FB220

   :KD 0        ;ANTRK      = 0   (AW1)
   :KB 1        ;KMDCODE    = 1   (Start)
   :KB 5        ;KMDFKT     = 5   (homing cycle)
   :KB 8        ;KMDBA      = 8   (NBA8)
   :KB 0        ;KMDBIT     = 0   (Default ID values)
   :KD 0        ;KMD_VAR1   = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR2   = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR3   = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR4   = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR5   = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR6   = 0   (2 Words Reserve)
```

```
;DB06
;KMD DB for start of spindle positioning;
;generated by FB220

   :KD 0        ;ANTRK      = 0   (AW1)
   :KB 1        ;KMDCODE    = 1   (Start)
   :KB 6        ;KMDFKT     = 6   (Spindle positioning)
   :KB 8        ;KMDBA      = 8   (NBA8)
   :KB 0        ;KMDBIT     = 0   (Default ID values)
   :KD 0        ;KMD_VAR1   = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR2   = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR3   = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR4   = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR5   = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR6   = 0   (2 Words Reserve)


;DB07
;KMD DB for start of absolute positioning;
;generated by FB220

   :KD 0        ;ANTRK      = 0   (AW1)
   :KB 1        ;KMDCODE    = 1   (Start)
   :KB 7        ;KMDFKT     = 7   (Absolute positioning)
   :KB 8        ;KMDBA      = 8   (NBA8)
   :KB 0        ;KMDBIT     = 0   (Default ID values)
   :KD 0        ;KMD_VAR1   = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR2   = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR3   = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR4   = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR5   = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR6   = 0   (2 Words Reserve)


;DB08
;KMD DB for start of relative positioning;
;generated by FB220

   :KD 0        ;ANTRK      = 0   (AW1)
   :KB 1        ;KMDCODE    = 1   (Start)
   :KB 8        ;KMDFKT     = 8   (Relative positioning)
   :KB 8        ;KMDBA      = 8   (NBA8)
   :KB 0        ;KMDBIT     = 0   (Default ID values)
   :KD 0        ;KMD_VAR1   = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR2   = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR3   = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR4   = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR5   = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR6   = 0   (2 Words Reserve)
```

```
;DB09
;KMD DB reserved;
;generated by FB220

   :KD 0       ;ANTRK      = 0   (AW1)
   :KB 1       ;KMDCODE    = 1   (Start)
   :KB 9       ;KMDFKT     = 9   (reserve)
   :KB 0       ;KMDBA      = 0   (HBA0)
   :KB 0       ;KMDBIT     = 0   (Default ID values)
   :KD 0       ;KMD_VAR1   = 0   (2 Words Reserve)
   :KD 0       ;KMD_VAR2   = 0   (2 Words Reserve)
   :KD 0       ;KMD_VAR3   = 0   (2 Words Reserve)
   :KD 0       ;KMD_VAR4   = 0   (2 Words Reserve)
   :KD 0       ;KMD_VAR5   = 0   (2 Words Reserve)
   :KD 0       ;KMD_VAR6   = 0   (2 Words Reserve)
```

```
;DB10
;KMD DB for start of synchronous control;
;generated by FB220

   :KD 0       ;ANTRK      = 0   (AW1)
   :KB 1       ;KMDCODE    = 1   (Start)
   :KB 10      ;KMDFKT     = 10  (Synchronous control)
   :KB 0       ;KMDBA      = 0   (HBA0)
   :KB 0       ;KMDBIT     = 0   (Default ID values)
   :KD 0       ;KMD_VAR1   = 0   (2 Words Reserve)
   :KD 0       ;KMD_VAR2   = 0   (2 Words Reserve)
   :KD 0       ;KMD_VAR3   = 0   (2 Words Reserve)
   :KD 0       ;KMD_VAR4   = 0   (2 Words Reserve)
   :KD 0       ;KMD_VAR5   = 0   (2 Words Reserve)
   :KD 0       ;KMD_VAR6   = 0   (2 Words Reserve)
```

```
;DB11
;KMD DB for start of actual position value offset;
;generated by FB220

   :KD 0       ;ANTRK      = 0   (AW1)
   :KB 1       ;KMDCODE    = 1   (Start)
   :KB 11      ;KMDFKT     = 11  (Actual position value offset)
   :KB 0       ;KMDBA      = 0   (HBA0)
   :KB 0       ;KMDBIT     = 0   (Default ID values)
   :KD 0       ;KMD_VAR1   = 0   (2 Words Reserve)
   :KD 0       ;KMD_VAR2   = 0   (2 Words Reserve)
   :KD 0       ;KMD_VAR3   = 0   (2 Words Reserve)
   :KD 0       ;KMD_VAR4   = 0   (2 Words Reserve)
   :KD 0       ;KMD_VAR5   = 0   (2 Words Reserve)
   :KD 0       ;KMD_VAR6   = 0   (2 Words Reserve)
```

```
;DB12
;KMD DB reserve;
;generated by FB220

   :KD 0        ;ANTRK       = 0   (AW1)
   :KB 1        ;KMDCODE     = 1   (Start)
   :KB 12       ;KMDFKT      = 12  (Reserve)
   :KB 0        ;KMDBA       = 0   (HBA0)
   :KB 0        ;KMDBIT      = 0   (Default ID values)
   :KD 0        ;KMD_VAR1    = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR2    = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR3    = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR4    = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR5    = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR6    = 0   (2 Words Reserve)


                .
                .
                .

;DB15
;KMD DB reserve;
;generated by FB220

   :KD 0        ;ANTRK       = 0   (AW1)
   :KB 1        ;KMDCODE     = 1   (Start)
   :KB 15       ;KMDFKT      = 15  (Reserve)
   :KB 0        ;KMDBA       = 0   (HBA0)
   :KB 0        ;KMDBIT      = 0   (Default ID values)
   :KD 0        ;KMD_VAR1    = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR2    = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR3    = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR4    = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR5    = 0   (2 Words Reserve)
   :KD 0        ;KMD_VAR6    = 0   (2 Words Reserve)
```

The following AWL program example serves to show the operation mode of the FB220:

```
;OB22
;Example for FB 220 (ADPS initialization in the
;scope of the start OB22)

   :A   DB  36        ;ADPS init DB
   :SPA FB  220

   :BE
```

```
;DB36
;DB for ADPS initialization;
;generated with APROS

   :KB 0       ;(DR0) Reserve              = 0
   :KB 0       ;(DL0) Max. number of repl.  = 0 (no replication in case of error)
   :KB 1       ;(DR1) Mode select          = 1 (without echo and ready evaluation)
   :KB 5       ;(DL1) Strobe low time =     5 (ms)
   :KB 0       ;(DR2) ESB address          = 0 (EB 0)
   :KB 8       ;(DL2) EDB address          = 8 (EB 8)
   :KB 0       ;(DR3) ASB address          = 0 (AB 0)
   :KB 8       ;(DL3) ADB address          = 8 (AB 8)
   :KF 0       ;(DW4) Timeout              = 0 (no timeout monitoring)
   :KF 0       ;(DW5) Reserve              = 0
```

6 Annex

6.1 Examples

The following examples show the handling of the ADPS interface to be organized by the higher ranking system.

The syntax in the scope of the exemplarily written processes is:

[S [xxx.yyy,DB]] [;comment]
or
[L [000.100,DB]] [;comment]

with: S = write cycle
L = read cycle

xxx = Function code (F code) (cf. ASB: Bit 3..5)
yyy = extended function code (for F code: 000..011) / AW address (for F code: 100..111; cf. ASB: Bit 0..2)

000.100 = Function code and extended function code for read cycles (cf. ESB: Bit 0..5)

DB = Data byte value (0x.. = Hexadecimal notation)
[] = optional input
{} = alternative input

Exemplary targets are:

A) AW1 in speed control with n_{set} = 2500 rev/min and reduced torque limit of 50% nominal torque.

B) Configuration of the secondary operation mode 1 (setpoint value source = actual position value of AW1; operation mode = position control) as a remanent parameter change (e.g. as a condition for example C).
Note: The change only becomes effective with activation of the Inverter On.

C) Commanding AW8 in synchronous control with alignment to master (AW1); master slave offset = 90°. Later the offset is changed to 95°.

D) The actual position value of AW3 is to be acquired.

Example A:

```
S 000.001, 0x04  ;Comm. DB04 (select dig. speed control)
S 000.010, 0x10  ;Write last sequential value "length 16 Words"
                 ;
S 010.110, 0x04  ;Write DW04 (parameter 1, LW)
S 000.001, 0xC4  ;Write sequential value "2500 (LB/LW)"
S 000.001, 0x09  ;Write sequential value "2500 (HB/LW)"
                 ;
S 000.011, 0x14  ;Generate DB (e.g. DB20) for temporary parameter change
S 000.001, 0x06  ;Write sequential value "length 6 Words (LB)"
S 000.001, 0x00  ;Write sequential value "length 6 Words (HB)"
S 010.110, 0x02  ;Write DW02 (parameter number)
S 000.001, 0x52  ;Write sequential value "ID 82 (LB)"
S 000.001, 0x00  ;Write sequential value "ID 82 (HB)", ID82 (positive torque limit)
S 010.110, 0x04  ;Write DW04 (parameter number)
S 000.010, 0x32  ;Write last sequential value "value 50 (LB)"
                 ;
S 101.000, 0x14  ;Temp. parameter change with regard to AW1, via DB20
;                ;for negative torque limit only ID in DB is changed
S 010.110, 0x02  ;Write DW02 (parameter number)
S 000.010, 0x53  ;Write last sequential value "ID 83 (LB)", ID83 (torque limit negative)
;
S 101.000,0x14 ;Temp. parameter change with regard to AW1, via DB20
;
S 100.000, 0x04  ;Call commanding (dig. torque control),
                 ;with regard to AW1
```

Example B:

Conditions for AW8:
- AW8 works with the parameter set n° 8

```
S 000.101, 0x08  ;Write remanent drive parameter in parameter set 8
                 ;ID 32801 (0x8021H), value (0x00240004H :setpoint value source =
                 ;actual position value of AW1 / operation mode = position control)
S 000.001, 0x21  ;Write sequential value "ID 32801 LB"
S 000.001, 0x80  ;Write sequential value "ID 32801 HB"
S 000.001, 0x04  ;Write sequential value "value 240004H LB/LW"
S 000.001, 0x00  ;Write sequential value "value 240004H HB/LW"
S 000.010, 0x24  ;Write last sequential value "value 240004H LB/HW"
```

Example C:

Conditions for AW8:
- ID 116 (resolution motor encoder) = 20000
- ID 32801 (secondary operation mode1 set according example B)
  - Setpoint value source = 0x24 (actual position value of AW1)
  - Operation mode = 0x0004 (position control)

```
S 000.011, 0x0A ;Select comm. DB10 (Sync. with/without WA)
S 000.010, 0x10 ;Write last sequential value "length 16 words (LB)"
;
S 010.110, 0x04 ;Write DW04 (parameter 1, LW)
S 000.001, 0x88 ;Write sequential value "5000 (LB/LW)"
S 000.001, 0x13 ;Write sequential value "5000 (HB/LW)"
                ;5000 = 90° with motor encoder resolution of 20000 inc
                ;
S 100.111, 0x0A ;Call commanding (Sync. with/withotu WA), with regard to AW8
                ;Command. works per default value with AMKsecondary operation mode 1
                ;
S 010.110, 0x04 ;Write DW04 (Parameter 1, LW)
S 000.001, 0x9E ;Write sequential value "5278 (LB/LW)"
S 000.001, 0x14 ;Write sequential value "5278 (HB/LW)"
                ;5278 = 95° with motor encoder resolution of 20000 Inc
                ;
S 100.111, 0x0A ;Call commanding (Sync. with/without WA), with regard to AW8
                ;Command. works per default value with AMK secondary operation
                ;mode 1
```

Example D:

```
S 011.100, 0x88 ;System info of category "E" (inputs) / read double word with the
                ;info addr. 136 (actual position value AW3)
L 000.100, 0x20 ;Actual value AW3 LB/LW (for e.g. 20000 Inc)
L 000.100, 0x4E ;Actual value AW3 HB/LW (for e.g. 20000 Inc)
L 000.100, 0x00 ;Actual value AW3 LB/HW (for e.g. 20000 Inc)
L 000.100, 0x00 ;Actual value AW3 HB/HW (for e.g. 20000 Inc)
```

7 Impressum

Title **AMK Digital Parallel Interface (ADPS)**

Objective **Description of the AMK Digital Parallel Interface (ADPS)**

Part-Number **27886**

History

Date
1995/22

Disclaimer We reserve the right to change the contents of the documentation and the availability of products at any time without prior notice.

Service Tel.: **+49/(0)7021 / 5005-191, Fax –193**

Business Hours:
Mo-Fr 7.30 - 16.30, On weekends and holidays calls are forwarded to an emergency response number by the automated answering system.

To assure a fast and accurate response to solve customer problems we ask for your cooperation in providing us with the following information:

- Nameplate data
- Software version
- System configuration and application
- Description of problem and presumed cause of failure
- Diagnostic message (error code)

Publisher AMK Arnold Müller Antriebs- und Steuerungstechnik GmbH & Co. KG
Gaußstraße 37 – 39, 73230 Kirchheim/Teck

Tel.: 07021/5005-0, Fax: 07021/5005-176
E-Mail: info@amk-antriebe.de

For further information www.amk-antriebe.de