AMKASYN

VARIABLE SPEED DRIVES

AMKASYN

Digital Converter in Modular Construction

Programmable Control PS

AMK-Specific Function Blocks

Version PSx V02.14

4200.E

Part No. :27889

Arnold Müller, Antriebs- und Steuerungstechnik GmbH & Co. KG, D-73230 Kirchheim/Teck,
Tel.: 07021/50 05-0, Telefax: 07021/50 05-176

Contents

Illustrations

ABBREVIATIONS USED

ABIS	AMK operator panel AB202L start-up software
Anz0	Display Bit0 in the PS status
Anz1	Display Bit1 in the PS status
AWx	Inverter module (1...8)
AWL	Instruction list
AZ	AMKASYN central module
AZB	AZ operator panel
AZ-PSx	AMKASYN option card for implementing the PS functions
DB	Data block
DD	Data double word
DL	Data word (left byte)
DR	Data word (right byte)
ED	Input double word
EOT	End of text
EOF	End of file
E/A/M	Input/output/flag
FB	Function block
FIPW	Function interpolator for distance-distance function
FIPZ	Function interpolator for distance-time functions
ID	Parameter of the basic system
LSW	Least significant word
MB	Flag byte
MSW	Most significant word
Ov	Overflow bit in the PS status
PS	Programmable control
SAK	Following error compensation
SF	Fast function
STX	Start of text
SF-Nr-Max	Largest possible permissible SF number (currently = 15)
SF type-Max	Largest possible permissible SF type number (currently = 11)
TTY	Teletype (20mA current interface)

1 Overview

Function blocks (FBs) are available to the user for frequently recurring or complex functions. They are divided into two groups. Function blocks which the user himself programs belong to the first group. The second group comprises the so-called AMK-specific FBs. They contain special functions programmed in the system software which can be used by the user. FB200 to FB255 are provided for this second group. The following table provides an overview of the currently available AMK-specific FBs. A detailed description of the different function blocks is provided in Section 2.

AMK-specific function blocks (in machine code)	
Block	**Description**
FB200	Updating drive status
FB201	Commanding drive
FB202	Write/read drive parameters
FB203	Write/read user list 1
FB204	Initialize serial interface
FB205	Send through serial interface
FB206	Receive through serial interface
FB207	Intializing fast function (SF)
FB208	Commanding fast function (SF)
FB209	Synchronization after BA change
FB210	Table value calculation
FB211	Floating point arithmetic
FB212	Diagnosis
FB213	Reserved
FB220	Initialize AMK digital parallel interface (ADPS)

2 AMK-specific function blocks

2.1 FB200 "Updating drive status"

This function block provides the AWL programmer the current status and error state of all drives in a system. A commanding interface the data of which both the PS and the drive control access is assigned to each of the drives. The status and the error byte of this interface always indicate the current status of the relevant drive. To make this current status available to the AWL programmer for possible program branches and other evaluations, this current status per drive is transferred by FB200 in each case into a status and an error flag byte:

AW No	STATUS flag byte	ERROR flag byte
AW1	MB 240	MB 241
AW2	MB 242	MB 243
AW3	MB 244	MB 245
AW4	MB 246	MB 247
AW5	MB 248	MB 249
AW6	MB 250	MB 251
AW7	MB 252	MB 253
AW8	MB 254	MB 255

The meaning of the individual bit information is defined as follows (cf. documentation: PS commanding drive):

– **MB 240, 242,.., 254: Status (commanding AW1, AW2, .., AW8)**

Bit	Meaning	
.0	x	1: Command set in drive, order accepted 0: No command set in the drive
.1	x	1: Command not interrupted 0: Command interrupted
.2	x	1: Command not ready 0: Command ready
.3	x	1: Command is in error status 0: Command is error-free
.4	n	Currently not used
.5	n	Currently not used
.6	n	1: Command ready for signal processing (e.g. cams)
.7	x	1: FB201 selection not yet processed 0: Selection processed, status valid

For monitoring the less significant 4 bits as well as the most significant Bit7 of the command status must be evaluated as follows:

0xxx0000: Basic status (after switching on) "RESET"
0xxx0011: Command executed in the drive correctly "READY"
0xxx0101: Command interrupted in the drive "HALT"
0xxx0111: Command in execution in the drive "ACTIVE"
0xxx1111: Error status "ERROR"

x = arbitrary

- **MB 241, 243, .., 255: Error (commanding AW1, AW2, .., AW8)**

Bit		Meaning	
.0	x	= 0 no error	≠ 0 error
.1	x		
.2	x		
.3	x		
.4	x		
.5	x		
.6	x		
.7	x		

The function block FB200 is processed automatically, cyclically by the system after the transfer of the E/A image and before processing the OB01. In this way the drive status is updated in each case. In addition FB200 can also be selected in the AWL program.

2.2 FB201 "Commanding drive"

This function block implements the commanding of a selected drive as well as the updating of the status and of the error flag byte of this drive corresponding to the status and the error byte in the commanding interface.

To place the drive in another drive function or to change into another operating mode, the required data must be entered and a defined protocol must be maintained in its commanding interface. For simple handling, an indirect access through a data block is provided the user on the part of the PS.

All required commanding data must be provided in a DB by means of programming unit or by the AWL program for this purpose. Before selection of the FB201 this DB must be activated on the AWL side. The FB201 when selected then causes the transfer of the data from the current DB into the commanding interface and thus triggering drive commanding. At the same time the statuses of the status and of the error byte are transferred from the commanding interface of the drive into the status and the error flag byte of this drive (updated). In this update Bit 7 of the status flag byte is also set as identification that the commanding of the drive was triggered by the PS.

The AWL program must be compiled so that with reset Bit 7 (drive ready for new commanding) the FB201 is selected only once and then only the Bit 7 of the status flag byte of the corresponding drive is evaluated. However, independently of this, the contents of the status and error flag byte of the drive, or the AW status messages on the AWL program side can also be evaluated (e.g. to detect the end of a started positioning of AW1: Status "MB 240" = 0xxx0011, Error "MB 241" = 0, In Position "E 224.6" = 1; cf. documentation: PS command set, Section 6.5).

The structure of a data block which is required for selecting FB201 looks as follows:

Data double word	High word H-Byte	High word L-Byte	Low word H-Byte	Low word L-Byte
0	DL1 Reserve = 0	DR1 Reserve = 0	DL0 KMDMODE	DR0 ANTRK
2	DL3 KMDBIT	DR3 KMDBA	DL2 KMDFKT	DR2 KMDCODE
4	DD4		KMD_VAR1	
6	DD6		KMD_VAR2	
8	DD8		KMD_VAR3	
10	DD10		KMD_VAR4	
12	DD12		KMD_VAR5	
14	DD14		KMD_VAR6	

----
16 data words

The meaning of the individual DB bytes is defined as follows:

– **ANTRK (DR 0): Drive identifier = AW No**
  0 - 7: Corresponding to drive 1 to 8

– **KMDMODE (DL 0): Commanding mode**
  0: Commanding without change of the KMD_VAR1 to KMD_VAR6
  1: Commanding with change of the KMD_VAR1 to KMD_VAR6, if information is transferred from the drive in the variables (e.g. in a position command value determination "KMDFKT=12")

– **KMDCODE (DR 2): Commanding code**
  0: No commanding
  1: Start commanding
  2: Abort commanding
  3: Stop commanding (IPO)
  4: Continue commanding (IPO)
  5: Commanding value change

- **KMDFKT (DL 2): Commanding function**
  1: Operating mode change
  2: Parameter set switch-over
  3: Digital torque control
  4: Digital speed control
  5: Homing run
  6: Spindle positioning
  7: Absolute positioning
  8: Relative positioning
  9: Reserved
  10: Synchronous control
  11: Position feedback value shift
  12: Position command value determination
  13: Lifting axis
  14: Flying saw
  15: Reserved

- **KMDBA (DR 3): Commanding operating mode**
  0: AMK main operating mode according to ID 32800
  1 - 5: AMK secondary operating modes NBA1 - NBA 5 according to ID 32801 - ID 32805
  6: NBA6 preallocated by AMK according to ID 32806; free
  7: NBA7 preallocated by AMK; torque control with digital command value according to ID 32807
  8: NBA8 preallocated by AMK; position control with motor encoder and fine interpolation according to ID 32808
  9: NBA9 preallocated by AMK; speed control with digital command value according to ID 32809

- **KMDBIT (DL 3): Parameter valid bit mask**
  0: Standard data according to database ID definition
  1: Datum in KMD_VAR1 valid
  2: Datum in KMD_VAR2 valid
  4: Datum in KMD_VAR3 valid
  8: Datum in KMD_VAR4 valid
  16: Datum in KMD_VAR5 valid
  32: Datum in KMD_VAR6 valid

The individual bit information can be superimposed additively.

Remarks: It must be observed that the commandable functions depend upon the relevant product level of the basic drive system! Refer to the description of the commanding interface of the drives for further information about the individual data of this DB as well as about status and error bytes (cf. Docum.: PS commanding drive / drive functions).

The following AWL program example serves for illustrating the mode of operation of FB201:

Assignment list

```
;Assignment list
;for example FB 201

;Inputs
E_Kmd          E  0.0          ;Command triggering at 0->1

;Outputs
A_Fehl         A  0.0          ;Error indication

;Flags
M_SteuKmd      M  0.0          ;Commanding control flag
M_FlKmd        M  0.1          ;Commanding edge flag
```

Cyclic organization block OB01

```
;OB01
;Referencing AW1 with default parameters, on
;edge change 0->1 an $E_Kmd
;Prerequisite:  - Inverter on
;               - Controller enable on

:U             $E_Kmd          ;Command input
:UN            $M_SteuKmd      ;Interlocking during the check
:UN            $M_FlKmd        ;Edge flag
:S             $M_FlKmd        ;Set edge flag
:SPB           FB 01           ;Commanding on 0->1, if allowed
:UN            $E_Kmd
:R             $M_FlKmd        ;Reset edge flag

:U             $M_SteuKmd      ;Commanding check?
:SPB           FB 02           ;Commanding check

:BE
```

Function block FB01

```
;FB01
;Referencing AW1 with default parameters

:U             M  240.7        ;Commanding not permitted?
:S             $A_Fehl         ;Error: commanding not permitted
:BEB

:R             $A_Fehl         ;Reset error indication
:A             DB 16           ;Activate DB for commanding
:SPA           FB 201          ;Commanding

:S             $M_SteuKmd      ;Control flag for commanding check

:BE
```

Function block FB02

```
;FB02
;Commanding check

        :U          M  240.7
        :BEB                          ;Waiting for FB201 selection assumed

        :L          KB 15
        :L          MB 240
        :!=FD                         ;Error in the drive status
        :SPB        =  Fehl

        :L          KB 0
        :L          MB 241
        :><FD                         ;Error in drive error identifier
        :SPB        =  Fehl

        :R          $A_Fehl           ;Reset error indication
        :R          $M_SteuKmd        ;Reset error monitoring
        :BEA

Fehl    :S          $A_Fehl           ;Set error indication
        :R          $M_SteuKmd        ;Reset error monitoring

        :BE
```

Data block DB16

```
;DB16
;Command-DB for referencing AW1 with default
;parameters; generated with APROS

:KB 0                 ;ANTRK       = 0   (AW1)
:KB 0                 ;KMDMODE     = 0   (without KMD_VAR change)
:KF 0                 ;Reserve     = 0
:KB 1                 ;KMDCODE     = 1    (Start)
:KB 5                 ;KMDFKT      = 5    (homing run)
:KB 8                 ;KMDBA       = 8    (position control)
:KB 0                 ;KMDBIT      = 0    (default ID values)
:KD 0                 ;KMD_VAR1    = 0    (2 words spare)
:KD 0                 ;KMD_VAR2    = 0    (2 words spare)
:KD 0                 ;KMD_VAR3    = 0    (2 words spare)
:KD 0                 ;KMD_VAR4    = 0    (2 words spare)
:KD 0                 ;KMD_VAR5    = 0    (2 words spare)
:KD 0                 ;KMD_VAR6    = 0    (2 words spare)
```

2.3 FB202 "Write/read drive parameters"

The FB202 "Write/read drive parameters" replaces, upwards compatible, the FB202 "Changing temporary drive parameters" implemented up to version 2.04.

Generally two types of drive parameters must be distinguished in the AMKASYN system:

- remanent parameters and
- temporary parameters

Remanent parameters are filed in the AMKASYN on a non-volatile storage medium. They are not lost when the system is switched off and are effective directly on switching back on. If a remanent parameter is changed with active controller enable, then it is initially filed on the non-volatile storage medium. However, it becomes effective in the drive only on a transition of the controller enable 0->1. In contrast to this, a temporary parameter becomes effective immediately in the drive even with activated controller enable. Temporary parameters can only be written. They are lost when the system is switched off. (However not with inactive controller enable, as long as no remanent parameters are changed!)

Remarks: If with inactive controller enable both parameters (temporary and remanent parameters) are newly written e.g. for the "Synchronous ratio" in the drive, then the remanent parameter always becomes effective in the drive on activating the controller enable.

Furthermore the different storage structures of temporary and remanent parameters in the AMKASYN must be observed. Complete sets (sum of all data relevant for a drive) of remanent parameters can be filed in 10 different data sets (currently: 0...9). The assignment of a data set to a certain drive (AW module) is determined by Ident No. ID32813..ID32820.

Default setting is: Data set 1..8 -> Main parameter set AW1..AW8

There is therefore always access to one out of 10 data sets when reading and writing remanent parameters. When writing a temporary parameter, there is no change of the contents of a data set, but a direct change of the parameter in the "physical drive" (AW No. 0...7).

Access to the parameters is through function block FB202. All required parameter access data must be provided in a DB by means of programming unit or by the AWL program. This DB must be activated in the AWL program before selecting FB202.

The structure of the data block for function block FB202 has the following structure:

Data double word	High word		Low word	
	H byte	L byte	H byte	L byte
0	Reserved = 0		PARATYP	ANTRK/DATK
2	Reserved = 0		IDNR	
4	PARAWERT			

----
6 data words

Meaning of the individual bytes in the data block:

ANTRK/DATK (DR 0): Drive identifier/data set identifier

0..7: Drive identifier (AW1...AW8) on access to temporary parameters (PARATYP = 0)

0..9: Data set identifier (parameter set 0...9) on access to remanent parameters (PARATYP = 1,2)

PARATYP (DL 0): Parameter type

DL 0 (PARATYP) determines whether access is to temporary or remanent drive parameters.

0: Parameter type = write temporary parameter
1: Parameter type = write remanent parameter
2: Parameter type = read remanent parameter

IDNR (DW 2): Ident number

xxxxx: Ident number of the parameter to which access should be made (on temp. parameter change the ID must be contained in the list of the temporary parameters ID 32891).

PARAWERT (DD 4): Parameter value

yyyyy write: New value for the selected ID number
read: Parameter value which is read from the selected Ident number.

2.3.1 Writing temporary drive parameters

- The purpose of this function (PARATYP = 0) is to make a temporary drive parameter change during operation. The change becomes effective directly. Temporary parameters can only be written, not read. The list "Temp. parameters ID 270" contains all parameters which can be changed temporarily. In system booting the drives are configured, i.e. they are loaded initially with the remanent parameters. Readiness is indicated with the binary signal SBM. The temporary parameters can then be changed. A temporarily changed parameter remains effective until it is changed anew or until the system is reconfigured. A system reconfiguration always takes place:
- on switching on,
- if after occurrence of a fatal error (SBM=0) the error deletion function (input FL) was activated,
- the transition of the controller enable 0->1, if previously a remanent parameter was changed.

  If a remanent parameter change must also be effective after switching off and on, the parameter (ID No.) must also be overwritten in addition as remanent parameter (PARATYP = 1).

It must be observed that only the parameter value (cf. Section 2.3.2) can be changed temporarily with the aid of FB 202. All other elements are permanently agreed in the AMKASYN and filed in the EPROM and thus cannot be changed. The FB 202 triggers when selected the transfer of the data from the current DB to the drive control. Furthermore this FB has the effect that in the status flag (MB 238) the bit to be assigned in each case to a drive is set as identification of a triggered, but not yet acknowledged parameter change. Bit 0 in the flag byte 238 corresponds to drive 1 (AW1), bit 1 to drive 2 (AW2), ... The programmer is given with this the possibility of taking into account in his AWL program a concluded or not yet concluded parameter change triggering. The bit corresponding to the drive in the same manner is also reset in an error flag (MB 239) by FB 202 (no error). According to whether the new parameter data which were taken over by the drive control can be realized by the drive or not, there is a positive or negative acknowledgement in turn by SBUS communication. This acknowledgement is received by a special SBUS routine of the PS. It always resets the bit corresponding to the drive in the status flag (MB 238) (triggering parameter change ended). The corresponding bit is also reset in the error flag (flag byte 239) on a positive acknowledgement (no error, parameter change made) and set in the case of a negative acknowledgement (error, parameter change not made). If the bit corresponding to the intended drive is not checked in the status flag for an already triggered parameter change (bit set) in the AWL program before selecting FB 202, then the corresponding bit is still set in the error flag by the FB 202, in addition to the bit set in the status flag (error, parameter change already triggered and yet no acknowledgement received).

In summary therefore the following evaluation instructions to be observed by the AWL programmer result:

MB238 [1]	MB 239 [2]	Remarks
1	0	Parameter change triggered; no acknowledgement as yet; there may be no new triggering as yet.
0	0	Parameter change made without error, there can be new triggering.
0	1	Parameter change could not be made (e.g. wrong parameter, or ID or drive No.), there can be no new triggering.
1	1	It was attempted to trigger a renewed parameter change, although no acknowledgement of the previously triggered change has yet been made.

[1] Status of the bit corresponding to the drive in the status flag byte 238
[2] Status of the bit corresponding to the drive in the error flag byte 239

The following AWL program example serves to illustrate the mode of operation of FB202:

Assignment list

```
;Assignment list
;for example FB 202, PARATYP=0 (Temp. para.)

;Inputs
E_Tmp           E  0.1          ;Triggering a temp. parameter
                                ;change at 0->1

;Outputs
A_Fehl          A  0.0          ;Error indication

;Flags
M_SteuTmp       M  0.2          ;Control flag temp. parameter change
M_FlTmp         M  0.3          ;Edge flag temp. parameter change
```

Cyclic organization block OB01

```
;OB01
;Write temporary parameters (ID32780, acceperation ramp)
;AW1

:U              $E_Tmp          ;Read input for temp. parameter change
:UN             $M_StTmp        ;Interlocking during the check
:UN             $M_FlTmp        ;Edge flag
:S              $M_FlTmp        ;Set edge flag
:SPB            FB 03           ;Change at 0->1, if allowed
:UN             $E_Tmp
:R              $M_FlTmp        ;Reset edge flag

:U              $M_SteuTmp      ;Parameter change check?
:SPB            FB 04           ;Parameter change check

:BE
```

Function block FB03

```
;FB03
;Temporary parameter change of AW1

:U        M  238.0      ;Temp. parameter change permitted?
:S        $A_Fehl       ;Error: parameter change not permitted
:BEB

:R        $A_Fehl       ;Reset error indication
:A        DB 17         ;Activate DB for temporary parameter change
:SPA      FB 202        ;Write/read drive parameters

:S        $M_SteuTmp    ;Control flag for parameter change check
:BE
```

Function block FB04

```
;FB04
;Parameter change check

     :U        M  238.0
     :BEB                     ;Waiting for FB202 call accepted

     :U        M  239.0       ;Error in FB202 call processing
     :SPB      =  Fehl

     :R        $A_Fehl        ;Reset error indication
     :R        $M_SteuTmp     ;Reset error monitoring
     :BEA

Fehl :S        $A_Fehl        ;Set error indication
     :R        $M_SteuTmp     ;Reset error monitoring

     :BE
```

Data block DB17

```
;DB17
;DB for temperature parameter change of the acceleration
;ramp ID 32780 / AW1; generated with APROS

:KB 0              ;ANTRK    = 0       (AW1)
:KB 0              ;PARATYP  = 0       (temp. parameter)
:KF 0              ;Reserved
:KD 32780          ;IDNR     = 32780   (acceleration ramp)
:KD 1000           ;PARAWERT = 1000    (100ms)
```

2.3.2 Write/read remanent drive parameters

A remanent parameter consists of a total block of information, designated below as parameter or data block. A data block contains essentially the following elements:

- Parameter value
- An ASCII string as text identifier of the parameter
- Attribute for determining the data format of a parameter.
- Minimum and maximum values
- Unit code
- Additional AMK-specific administrative information

The parameter value can be accessed exclusively with the aid of FB 202 (PARATYP=1: write; PARATYP=2: read). A 16-bit wide Ident number (ID number) is used for identifying a parameter.

A further distinction is made for the remanent parameters into:

- 2-byte parameters
- 4-byte parameters
- Lists of arbitrary lengths (max. 64 KBytes)

It must be observed that with the aid of FB 202 only parameters which are not lists can be accessed.

Access to the remanent parameters may be made only consecutively. Since apart from FB 202 also FB 203 ("Read/write user list 1“) accesses the remanent parameters, the accesses must be synchronized. For this reason the accesses to the remanent drive parameters are organized by the same flag byte "MB 237" for FB 202 as for FB 203. The flag bit "M 237.7“ identifies the current status of a read or write operation and must be queried by the AWL programmer before selecting FB 202:

- - M 237.7 = 0: A read or write operation can take place.
- - M 237.7 = 1: A commanded read or write operation is not yet concluded, a renewed operation is not permitted.

The flag bit "M 237.0" identifies the current error status:

- M 237.0 = 0: No error
- M 237.0 = 1: Error

Organization form of the flag byte "MB 237":

MB 237	Flag byte	
.0	x	Error status 0: No error 1: Error detected
.1	n	Currently not used (= 0)
.2	n	Currently not used (= 0)
.3	n	Currently not used (= 0)
.4	n	Currently not used (= 0)
.5	n	Currently not used (= 0)
.6	n	Currently not used (= 0)
.7	x	Write/read status 0: Starting status 1: Execution not yet completed

x: 0, 1

The following AWL program examples serve for illustrating the mode of operation of FB202:

Assignment list

```
;Assignment list
;for example FB 202, PARATYP=1 (write)

;Inputs
E_RemS          E  0.2          ;Triggering a reman. parameter
                                ;change at 0->1

;Outputs
A_Fehl          A  0.0          ;Error indication

                                ;Flag
M_SteuRemS      M  0.4          ;Write reman. parameter control flag
M_FlRemS        M  0.5          ;Write reman. parameter edge flag
```

Cyclic organization block OB01

```
;OB01
;Write remanent parameter (ID32780, acceleration ramp)
;data record 1

:U              $E_RemS         ;Input for read reman. parameter change
:UN             $M_SteuRemS     ;Interlocking during the check
:UN             $M_FlRemS       ;Edge flag
:S              $M_FlRemS       ;Set edge flag
:SPB            FB 05           ;Change at 0->1, if allowed
:UN             $E_RemS
:R              $M_FlRemS       ;Reset edge flag

:U              $M_SteuRemS     ;Parameter change check?
:SPB            FB 06           ;Parameter change check

:BE
```

Function block FB05

```
;FB05
;Remanent parameter change data set 1

:U          M  237.7       ;Reman. parameter change permitted?
:S          $A_Fehl        ;Error: parameter change not permitted
:BEB

:R          $A_Fehl        ;Reset error indication
:A          DB 18          ;Activate DB for remanent parameter change
:SPA        FB 202         ;Write/read drive parameters

:S          $M_SteuRemS    ;Control flag for parameter change check
:BE
```

Function block FB06

```
;FB06
;Parameter change check

     :U     M  237.7
     :BEB                  ;Waiting for FB202 call accepted

     :U     M  237.0       ;Error at FB202 call processing
     :SPB   =  Fehl

     :R     $A_Fehl        ;Reset error indication
     :R     $M_SteuRemS    ;Reset error monitoring
     :BEA

Fehl :S     $A_Fehl        ;Set error indication
     :R     $M_SteuRemS    ;Reset error monitoring

     :BE
```

Data block DB18

```
;DB18
;DB for remanent parameter change of the acceleration
;ramp ID 32780 / data set 1; generated with
;APROS

:KB 1           ;DATK        = 1       (AW1)
:KB 1           ;PARATYP     = 1       (write remanent
                ;                      parameter)
:KF 0           ;Reserved
:KD 32780       ;IDNR        = 32780   (acceleration ramp)
:KD 1000        ;PARAWERT    = 1000    (100ms)
```

Assignment list

```
;Assignment list
;for example FB 202, PARATYP=2 (read)

;Inputs
E_RemL          E  0.3          ;Triggering a reman. parameter
                                ;query at 0->1

;Outputs
A_Fehl          A  0.0          ;Error indication

;Flags
M_SteuRemL      M  0.6          ;Read reman. parameter control flag
M_FlRemL        M  0.7          ;Read reman. parameter edge flag
MD_Anzeige      MD   4          ;For display of the parameter in the viewing
                                ;window
```

Cyclic organization block OB01

```
;OB01
;Read remanent parameter (ID32780, acceleration ramp)
;data set 1

:U              $E_RemL         ;Input for read rema. parameter query
:UN             $M_SteuRemL     ;Interlocking during the check
:UN             $M_FlRemL       ;Edge flag
:S              $M_FlRemL       ;Set edge flag
:SPB            FB 07           ;Query at 0->1, if allowed
:UN             $E_RemL
:R              $M_FlRemL       ;Reset edge flag

:U              $M_SteuRemL     ;Parameter query check?
:SPB            FB 08           ;Parameter query check
:BE
```

Function block FB07

```
;FB07
;Remanent parameter query data set 1

:U              M  237.7        ;Reman. parameter query permitted?
:S              $A_Fehl         ;Error: parameter query not permitted
:BEB

:R              $A_Fehl         ;Reset error indication
:A              DB 19           ;Activate DB for remanent parameter query
:SPA            FB 202          ;Write/read drive parameters

:S              $M_SteuRemL     ;Control flag for parameter query check
:BE
```

Function block FB08

```
;FB08
;Parameter query check

       :U        M  237.7
       :BEB                      ;Waiting for FB202 call accepted

       :U        M  237.0        ;Error at FB202 call processing
       :SPB      =  Fehl

       :R        $A_Fehl         ;Reset error indication
       :R        $M_SteuRemL     ;Reset error monitoring
       :A        DB  19          ;Acceleration ramp can be read from
       :L        DD  4           ;DD4/DB19 (1000 = 100ms)
       :T        $MD_Anzeige     ;Display in the viewing window
       :BEA

Fehl   :S        $A_Fehl         ;Set error indication
       :R        $M_SteuRemL     ;Reset error monitoring
       :BE
```

Data block DB19

```
;DB19
;DB for remanent parameter query of the acceleration
;ramp ID 32780 / data set 1; generated with
;APROS

:KB 1          ;DATK          = 1        (AW1)
:KB 2          ;PARATYP       = 2        (read remanent parameter)
:KF 0          ;Reserved
:KD 32780      ;IDNR          = 32780    (acceleration ramp)
:KD 0          ;PARAWERT      = 0        (10=1ms)
               ;contains parameter value after successful
               ;query
```

2.4 FB203 "Read/write user list 1"

FB203 facilitates reading or writing the drive-global, remanent user list 1 (see: AMKASYN documentation: Digital pulse converter in modular construction; parameters) through the AWL program.

Access to the user list 1 is organized by the current data block as well as the flag byte "MB 237“ apart from the FB 203.

The flag bit "M 237.7" identifies the current status of a read or write operation and must be queried by the AWL programmer before selecting the FB 203:

- M 237.7 = 0: A read or write operation can take place.
- M 237.7 = 1: A commanded read or write operation is not yet concluded, a renewed operation is not permitted.

The flag bit "M 237.0" identifies the current error status:

- M 237.0 = 0: No error
- M 237.0 = 1: Error

Organization form of the flag byte "MB 237":

MB 237	Flag byte	
.0	x	Error status 0: No error 1: Error detected
.1	n	n: Currently not used
.2	n	n: Currently not used
.3	n	n: Currently not used
.4	n	n: Currently not used
.5	n	n: Currently not used
.6	n	n: Currently not used
.7	x	Write/read status 0: Starting status 1: Execution not yet completed

x: 0, 1

Organization form of the current data block to be created:

Data word	High byte	Low byte
0	Header datum 1 (current length of the list)	
1	Header datum 2 (maximum length of the list)	
2	Useful datum 1	
3	Useful datum 2	
..	..	
129[1)] 255[2)]	Useful datum 128[1)] / 254[2)]	
----------- 130[1)] 256[2)]		

[1)] up to version AZR V02.04 or PS1 V02.04
[2)] from version AZR V02.05 or PSx V02.05

2 words header information (current and maximum length of the list in bytes) as well as 128[1)] /254[2)] words useful data of the user list 1 are filed in this data block. Therefore in principle 130[1)] /256[2)] words (260[1)] /512[2)] bytes) must be provided for the length of the data block to be created currently in connection with FB 203.

- In the reading case (DW 0 == 0) the complete user list 1 (header information + useful data) is made available as from "DW 0".
- In the writing case the data to be transferred must be written as from "DW 0" in the user list 1. The current length (in bytes) of the user list to be transferred must be stated in "DW 0“. The value 5555Hex must be entered in "DW 1" for protection against unwanted multiple writing of the user list 1. This value is checked by FB 203 in the case of a write order. If a value unequal to 5555Hex is found, then there is no write process and the error bit is set, if the value of "DW 1" is equal to 5555Hex, then the values as from "DW 0" are written in the user list 1, whereby "DW 1" is overwritten previously with the maximum length of the list (value = 260[1)] / 512[2)]).

[1)] up to version AZR V02.04 or PS1 V02.04
[2)] from version AZR V02.05 or PSx V02.05

Remarks: In a combination of version AZR > V02.04 and version PS1 = V02.04 the header information of user list 1 (ID 32798; current and maximum length) must be reduced to 260 if FB203 is used!

The following AWL program example serves for illustrating the mode of operation of FB203:

Assignment list

```
;Assignment list
;for example FB 203, write and read
;user list 1; DB20 / DB21 generated with start OB22

;Inputs
E_Anw1S      E  0.0        ;Triggering writing in the
                           ;user list 1 at 0->1
E_Anw1L      E  0.1        ;Triggering reading of the
                           ;user list 1 at 0->1

;Outputs
A_Fehl       A  0.0        ;Error indication

;Flags
M_Steu0      M  0.0        ;Control flag 0
M_FlankS     M  0.1        ;Writing edge flag
M_FlankL     M  0.2        ;Reading edge flag
```

Cyclic organization block OB01

```
;OB01
;Write and read user list 1

:U           $E_Anw1S      ;Input for writing
:UN          $E_Anw1L      ;Input for reading
:UN          $M_Steu0      ;Interlocking during the check
:UN          $M_FlankS     ;Edge flag
:S           $M_FlankS     ;Set edge flag
:A           DB 20         ;Activate DB for writing user list 1
:SPB         FB 01         ;Writing at 0->1, if allowed
:UN          $E_Anw1S
:R           $M_FlankS     ;Reset edge flag

:U           $E_Anw1L      ;Input for reading
:UN          $E_Anw1S      ;Input for writing
:UN          $M_Steu0      ;Interlocking during the check
:UN          $M_FlankL     ;Edge flag
:S           $M_FlankL     ;Set edge flag
:A           DB 21         ;Activate DB for reading user list 1
:SPB         FB 02         ;Reading at 0->1, if allowed
:UN          $E_Anw1L
:R           $M_FlankL     ;Reset edge flag

:U           $M_Steu0      ;Write/read check?
:SPB         FB 03         ;Write/read check

:BE
```

Cyclic organization block OB022

;OB22

;Start OB for creating DB20 and DB21

```
:L      KB    0
:E      DB    20            ;Delete DB 20, if created
:E      DB    21            ;Delete DB 21, if created
:L      KF    130 1)/256 2)
:E      DB    20            ;Create DB 20 with 130 1)/256 2) words
:E      DB    21            ;Create DB 21 with 130 1)/256 2) words
                            ;User list 1 with 260 bytes length

:A      DB    20            ;Activate DB 20
:L      KF    512
:T      DW    0             ;Identifier for writing access

:A      DB    21            ;Activate DB 21
:L      KF    0
:T      DW    0             ;Identifier for reading access

:BE
```

[1] up to version AZR V02.04 or PS1 V02.04
[2] from version AZR V02.05 or PSx V02.05

Function block FB01

```
;FB01
;Write user list 1

:U      M 237.7       ;Writing permitted?
:S      $A_Fehl       ;Error: writing not permitted
:BEB

:L      KH 5555
:T      DW 1          ;Cancel write protection

:R      $A_Fehl       ;Reset error indication
:SPA    FB 203        ;Write/read user list 1
                      ;corresponding to active DB

:S      $M_Steu0      ;Control flag for write/read
                      ;check

:BE
```

Function block FB 02

```
;FB02
;Read user list 1

:U          M  237.7      ;Reading permitted?
:S          $A_Fehl       ;Error: reading not permitted
:BEB

:L          KF 0
:T          DW 0          ;Identifier for reading

:R          $A_Fehl       ;Reset error indication
:SPB        FB 203        ;Write/read user list 1
                          ;corresponding to active DB

:S          $M_Steu0      ;Control flag for write/read
                          ;check

:BE
```

Function block FB03

```
;FB03
;Write/read check

     :U          M  237.7
     :BEB                      ;Waiting for FB203 call accepted

     :U          M  237.0      ;Error at FB203 call processing
     :SPB =      Fehl

     :R          $A_Fehl       ;Reset error indication
     :R          $M_Steu0      ;Reset error monitoring
     :BEA

Fehl:S           $A_Fehl       ;Set error indication
     :R          $M_Steu0      ;Reset error monitoring

     :BE
```

Note:
The user list 1 is organized word-wise in the AZR.
Enter in each case in ID 32798 (AZR) in DW0 and DW1 (or is preset on delivery of the AMKASYN system depending upon the AZR version):
"260" up to version AZR V02.04 / PS1 V02.04
"512" from version AZR V02.05 / PSx V02.05

2.5 FB204 / FB214 "Activate serial interface"

FB204 facilitates setting up a "Serial interface" (RS422/RS485) to which there can be access through the AWL program (interface mode = 0..2) or through which it is possible to communicate with the operator panel AB 202L (interface mode = 3).

Before initialization of the interface (selection of FB204) both a data block "DB x"

- for holding the characters to be received (interface mode = 0..2, 5..8: receive mailbox)
- and handling DB (interface mode = 3)

as well as the data block "DB y"

- for holding the characters to be sent (interface mode = 0..2: send mailbox)
- as variable DB (interface mode = 3)

must be provided.

In Mode 4 (RK512 mode) the data block address is taken directly from the RK512 protocol!

As from AZ-PS4 version 02.11 it is possible to select the 3964 protocol with the Mode 7 and the 3964R protocol with the Mode 8; whereby the priority is "HIGH" in both modes (cf. Section 2.5.4).

As from AZ-PS4 version 02.13 it is possible to communicate with an operator panel (touch panel, types "VTxxx") of the ESA company (interface mode =.9; cf. Section 2.5.5).

As from version AZ-PS5 V02.14/3500 it is possible to initialize a second "Serial interface" (RS232) with FB214 (cf. documentation: AZ-PS5). The use of the second interface is very largely identical to the previous interface from the viewpoint of the user program. Functional differences (or restrictions) are summarized at the end of this section.

Organization form of the current DB to be created

Data word	High byte	Low byte
0	Receive mailbox/ handling DB number/slave address	Send mailbox/ variable DB number
1	Parity	Interface mode
2	Stop bits	Data bits
3	Enable	EOT character
4	Baud rate (LSW)	
5	Baud rate (MSW)	

6		

For initializing the interface the following parameters must be provided in the current data block before selecting FB204:

Send mailbox DB number / variable DB number (DR 0)

– Meaning:	Interface mode = 0..2, 5..8	⇒ Send mailbox DB number
	Interface mode = 3, 9	⇒ Variable DB number
	Interface mode = 4	⇒ no meaning
– Value range:	Interface mode = 4	⇒ 0.. 255
	Other	⇒ 1.. 63
– Example:	Interface mode = 1: Value = 16	⇒ DB No. = 16 (send mailbox)

Receive mailbox DB number / handling DB number / slave address (DL 0)

– Meaning:	Interface mode = 0..2, 5..8	⇒ Receive mailbox DB number
	Interface mode = 3	⇒ Handling DB number
	Interface mode = 4	⇒ no meaning
	Interface mode = 9	⇒ Slave address
– Value range:	Interface mode = 4	⇒ 0.. 255
	Interface mode = 9	⇒ 0.. 31
	Other	⇒ 1.. 63
– Example:	Interface mode = 1: Value = 16	⇒ DB No. = 16 (receive mailbox)
	Interface mode = 9: Value = 1	⇒ Slave address = 1

Interface mode (DR1)

– Meaning: Selection of the interface functionality

– Value range:
0 = Exchange receive mailbox and send mailbox DB
1 = Standard mode (RS422; full duplex)
2 = Half duplex mode (RS485-capable; on parallel connection of the transmission and reception lines and taking into account the bus termination **Caution:** Possible only with AZ-PS3 module!
3 = Operator panel mode (for communication with the operator panel AB 202L)
4 = RK512 mode (DB access)
5 = 3964 mode (priority = low)
6 = 3964R mode (priority = low)
7 = 3964 mode (priority = high)
8 = 3964R mode (priority = high)
9 = Modbus mode (operator panel "ESA VTxxx")

– Remarks: Mode 2..9 available only in connection with 1st serial interface!

Parity (DL1)

– Meaning: Parity of the serial interface.

– Value range:
0 = no parity
1 = even parity

Data bits (DR2)

– Meaning: Data bit number of the serial interface.

– Value range:
7 = 7 data bits (parity = 1 required!)
8 = 8 data bits

Stop bits (DL2)

– Meaning:	Number of the stop bits of the serial interface.
– Value range:	1 = 1 stop bit 2 = 2 stop bits

EOT character (DR3)

– Meaning:	EOT character of the serial interface on receiving.
– Value range:	0 = no EOT character monitoring Other = EOT character

Enable (DL3)

– Meaning:	Receiver enable and enable of the error monitoring modes of the serial interface.
– Value range:	bit-coded Bit0 =1: Receiver Enable Bit1 =1: Parity Check Enable Bit2 =1: Framing Check Enable Bit3 =1: Overrun Check Enable
–Example:	Value = 15 ⇒ all monitorings and receiver enable are active

Baud rate (DD4)

– Meaning:	Setting of the transmission rate of the serial interface in baud (bit/s).
– Value range:	300 ≤ transmission rate ≤ 76800
–Example:	19200 ⇒ transmission rate =19200 baud

2.5.1 Access to the serial interface (Mode = 0..2)

Access to the serial interface is organized on the AWL program side by means of the send and receive mailboxes provided in the form of data blocks as well as send and receive control bytes implemented in the form of flag bytes (SBS/SBE).

As from version AZ-PS5 V02.14/3500 it is possible to initialize a second serial interface (RS232) with the FB214. However, only the modes 0 and 1 are permitted in connection with this (see above).

Organization form of control byte transmission (SBS):

MB 235 or MB208	SBS (1st serial interface) SBS (2nd serial interface; if AZ-PS5)	
.0	x	Error in last data transfer 0: No error 1: Error detected
.1	n	Currently not used
.2	n	Currently not used
.3	n	Currently not used
.4	n	Currently not used
.5	n	Currently not used
.6	n	Currently not used
.7	x	Send mailbox access right 0: Program can access send mailbox 1: Operating system can access send mailbox

x: 0, 1

Organization form of the receive control byte (SBE):

MB 236 or MB209	SBE (1st serial interface) SBE (2nd serial interface; if AZ-PS5)	
.0	x	Error in last data transfer 0: No error 1: Error detected
.1	n	Currently not used
.2	n	Currently not used
.3	n	Currently not used
.4	n	Currently not used
.5	n	Currently not used
.6	n	Currently not used
.7	x	Receive mailbox access right 0: Program can access receive mailbox 1: Operating system can access receive mailbox

x: 0, 1

Organization form of the data blocks for sending (send mailbox) and receiving (receive mailbox):

Data word	High byte	Low byte
0	Length of the data package (number of the useful data in bytes)	
1	reserved	
2	Useful datum 2	Useful datum 1
..	..	..
n/2 + 1	Useful datum n	Useful datum n-1

----
n/2 + 2

n: Block length (even number, in bytes)

<u>Example for the standard mode (1st serial interface):</u>

```
;PB00
;Activate serial interface

;Delete DBs, if created
:L          KB  0
:E          DB  22        ;Init DB
:E          DB  23        ;Send mailbox DB
:E          DB  24        ;Receive mailbox DB

;Preallocate DB23 with send text
:L          KB  5
:E          DB  23        ;Create and select DB 23
:A          DB  23        ;with 5 words

:L          KB  6
:T          DW  0         ;Send 6 useful data
:L          KC  eT        ;Send text init. (Test'\r''\n')
:T          DW  2
:L          KC  ts
:T          DW  3
:L          KH  0A0D
:T          DW  4

;Create DB24 for receive text
:L          KB  52
:E          DB  24        ;Create and select DB 24
:A          DB  24        ;with 52 words

:L          KB  16
:T          DW  0         ;Receive maximum 16 useful data
```

```
;Preallocate DB22 for Init. serial interface
:L          KB  6
:E          DB  22      ;Create and select DB 22
:A          DB  22      ;with 6 words

:L          KB  23
:T          DR  0       ;Send mailbox = DB 23
:L          KB  24
:T          DL  0       ;Receive mailbox = DB 24
:L          KB  1
:T          DR  1       ;Mode = 1 (RS422; full duplex)
:L          KB  1
:T          DL  1       ;Parity = Even
:L          KB  7
:T          DR  2       ;Data bits = 7
:L          KB  1
:T          DL  2       ;Stop bits = 1
:L          KB  13
:T          DR  3       ;EOT character = '\r' (Carriage Return)
:L          KB  15
:T          DL  3       ;Enable = Receiver, Parity, Over, Fram
:L          KD  9600
:T          DD  4       ;Baud rate = 9600

:SPA        FB 204      ;Activate 1st serial interface

:BE
```

Example for the standard mode (2nd serial interface; only in connection with AZ-PS5):

```
;PB00
;Activate serial interface

;Delete DBs, if created
:L          KB  0
:E          DB  22      ;Init DB
:E          DB  23      ;Send mailbox DB
:E          DB  24      ;Receive mailbox DB

;Preallocate DB23 with send text
:L          KB  5
:E          DB  23      ;Create and select DB 23
:A          DB  23      ;with 5 words

:L          KB  6
:T          DW  0       ;Send 6 useful data
:L          KC  eT      ;Send text init. (Test'\r''\n')
:T          DW  2
:L          KC  ts
:T          DW  3
:L          KH  0A0D
:T          DW  4
```

```
;Create DB24 for receive text
:L          KB  52
:E          DB  24          ;Create and select DB 24
:A          DB  24          ;with 52 words

:L          KB  16
:T          DW  0           ;Receive maximum 16 useful data
;Preallocate DB22 for Init. serial interface
:L          KB  6
:E          DB  22          ;Create and select DB 22
:A          DB  22          ;with 6 words

:L          KB  23
:T          DR  0           ;Send mailbox = DB 23
:L          KB  24
:T          DL  0           ;Receive mailbox = DB 24
:L          KB  1
:T          DR  1           ;Mode = 1 (RS232)
:L          KB  1
:T          DL  1           ;Parity = Even
:L          KB  7
:T          DR  2           ;Data bits = 7
:L          KB  1
:T          DL  2           ;Stop bits = 1
:L          KB  13
:T          DR  3           ;EOT character = '\r' (Carriage Return)
:L          KB  15
:T          DL  3           ;Enable = Receiver, Parity, Over, Fram
:L          KD  9600
:T          DD  4           ;Baud rate = 9600

:SPA        FB 214          ;Activate 2nd serial interface

:BE
```

2.5.2 Operator panel mode (Mode = 3)

The **procedure 3964R** (cf. Section 2.5.4) and the computer coupling protocol **RK512** (cf. Section 2.5.3) was simulated for the connection with the operator panel "AB 202L". RK512 corresponds to the standard slave protocol, but can be used only for two defined data blocks (see below: variable DB number, handling DB number).

The protocol can be configured using the function block FB204 "Activate serial interface" with "Interface mode = 3".

Caution: To avoid character delay time errors on the part of the AB202L, a PS cycle time of 200ms must not be exceeded!

The individual data words of the current DB for activating the mode and setting the serial interface must be initialized as follows :

DR 0:	xx	Variable DB number (e.g. 51)
DL 0:	xx	Handling DB number (e.g. 50)
DR 1:	3	Operator panel mode
DL 1:	1	Even parity
DR 2:	8	8 data bits
DL 2:	1	1 stop bit
DR 3:	0	No EOT character
DL 3:	15	All monitorings are active
DD 4:	19200	Baud rate e.g. 19200 baud

----
6 data words

In "Mode = 3" protocol errors are reported to the PS program through the flag bytes 235/236 (MB 235 / MB 236). These errors must be acknowledged by the AWL program in the PS by deleting the flag byte (MB 235 = 0, or MB 236 = 0).

Remarks: These errors are not displayed on the AMKASYN operator panel (AZB). The PS goes into the error state!

The individual error codes of the MB 235 or MB 236 have the following meaning:
– Procedure 3964R

1	Timeout of the character delay time on reception
2	Timeout of the character delay time on transmission
3	Timeout of the acknowledgement delay time
4	No STX received
5	Faulty telegram length
6	Wrong acknowledgement on STX after 6 times repetition
7	Wrong acknowledgement on STX
8	Wrong acknowledgement for transmission after 6 times repetition
9	Wrong acknowledgement for transmission
10	No ETX received
11	Faulty transmission
12	Character error on reception (fault)
13	Inadmissible acknowledgement character after STX
14	Inadmissible acknowledgement character after transmission

– RK512

20	Handling or variable DB not generated or wrong length
21	Unknown DB number in the SEND command
22	Unknown telegram code in the SEND command
23	Error in the special command
24	Unknown DB number in the FETCH command
25	Unknown telegram code in the FETCH command
26	No SEND or FETCH telegram
27	No DB addressed
28	Character number > max. handling / or variable DB byte number; or character number > 138 bytes (max. 128 bytes data + 10 bytes RK512 header information)

Example for the operator panel mode:
Initializing the interface through DB1 and FB204

```
:L      KB   0
:E      DB   1        ;Deletion of the DB1
:L      KB   6        ;Length of the DB
:E      DB   1        ;Setting up the DB1
:A      DB   1        ;Activating the DB1
:L      KB   51       ;Variable DB number
:T      DR   0
:L      KB   50       ;Handling DB number
:T      DL   0
:L      KB   3        ; Operator panel mode
:T      DR   1
:L      KB   1        ;Even parity
:T      DL   1
:L      KB   8        ;8 data bits
:T      DR   2
:L      KB   1        ;1 stop bit
:T      DL   2
:L      KB   0        ;No EOT
:T      DR   3
:L      KB   15       ;Interface monitoring active
:T      DL   3
:L      KD   19200    ;Baud rate
:T      DD   4

:SPA    FB   204      ;Initializing the interface and selection
                      ;of the operator panel mode
```

2.5.2.1 Remarks on using the operator panel AB 202L:

Refer to the operating instructions for the description of the AMK operator panel start-up software (ABIS) for the AB 202L.

The following notes must be observed:

1. SIMATIK S5 must be selected as parameter for target control.

2. Settings of the driver software :
- Select the TTY interface as hardware interface
- Use the 3964R (RK512) protocol
- The baud rate is 19200 baud

3. Select the handling or variable block number corresponding to the AWL program in the PS.

4. All menus, status texts or messages can be activated in the PS only by an entry in the relevant data word of the handling block in the AWL program.

Load operator panel project:

1. Connect AB 202L with the PC through the connection cable "AB 202L↔PC".

2. Start the AMK operator panel start-up software ABIS. Observe that the interface with which the PC is connected with the AB 202L is correctly selected.
   (Option -Cx with x=1 for COM1 and with x=2 for COM2)

3. Switch on AB 202L with simultaneous operation of the "Shift", "Clear" and "Help" keys. Transferring the project file is then requested.

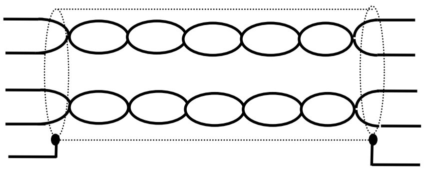
4. Press key "F8" (load project) of the PC for selecting the project.

5. Using the "TAB" key select e.g. the test project *p90test.prd* in a created project folder.

6. Press the "Shift F3" key (transfer project) for transferring the project into the AB 202L.

It is possible in the "Driver parameters of ABIS" menu item ("Shift F6") to set a switch-on delay of the AB 202L between 0 .. 60 seconds.

<u>**Connection cable**</u>
- Connection "AB 202L ↔ PC" through the programming interface (RS232C): an assembled cable is supplied for this.
- Connection "AB 202L ↔ PS" through the communication interface (RS422) according to the following pin assignment:

AB 202L (communication interface, D-SUB 25-pin)			AMKASYN PS (X72, D-SUB 15-pin)	
Pin 10	TxD+		RxD+	Pin 13
Pin 19	TxD-		RxD-	Pin 3
Pin 13	RxD+		TxD+	Pin 12
Pin 14	RxD-		TxD-	Pin 2
Shield applied to metallized plug housing			Shield applied to metallized plug housing	

The signal wire pairs "TxD+" and "TxD-" as well as "RxD+" and "RxD-" must be twisted in each case and have a minimum cross-section of 0.14 mm^2 and be provided with a common shield.
The shield must be applied to the metallized plug housing at both ends.

The type "UNITRONIC Li2YCY (TP)" from Lapp is recommended as cable.

Notes:

- A metallized D-SUB plug housing with lateral cable outlet must be used on the AMKASYN PS side.
- The housing of the AB202L must be connected with RF-wise good PE connection with the PE star point of the AMKASYN system.

2.5.3 RK512 mode (Mode = 4)

The RK512 mode (Mode = 4) is largely identical with the operator panel mode (Mode =3); cf. Section 2.5.2. However, in contrast to the operator panel mode, access can be made to arbitrary data words of each data block present in the PS. (The DB number is only determined by the RK512 protocol; planning this DB in the scope of the interface activation is not required.)

The protocol can be configured with "Interface mode = 4" through the function block FB204 "Activate serial interface".

The individual data words of the current DB for activating the mode and the setting of the serial interface must be initialized as follows:

DR 0:	0	Reserved = 0
DL 0:	0	Reserved = 0
DR 1:	4	RK512 mode
DL 1:	1	Even parity
DR 2:	8	8 data bits
DL 2:	1	1 stop bit
DR 3:	0	No EOT character
DL 3:	15	All monitorings are active
DD 4:	4800	Baud rate = 4800 baud

----
6 data words

In "Mode = 4" protocol errors are reported to the PS program through the flag bytes 235/236 (MB 235 / MB 236). These errors must also be acknowledged by the AWL program in the PS by deleting the flag byte (MB 235 = 0, or MB 236 = 0).

Remarks: These errors are not displayed on the AMKASYN operator panel (AZB). The PS does not go into the error status!

The individual error codes of the MB 235 or MB 236 have the following meaning:

– Procedure 3964R

1	Timeout of the character delay time on reception
2	Timeout of the character delay time on transmission
3	Timeout of the acknowledgement delay time
4	No STX received
5	Faulty telegram length
6	Wrong acknowledgement on STX after 6 times repetition
7	Wrong acknowledgement on STX
8	Wrong acknowledgement for transmission after 6 times repetition
9	Wrong acknowledgement for transmission
10	No ETX received
11	Faulty transmission
12	Character error on reception (fault)
13	Inadmissible acknowledgement character after STX
14	Inadmissible acknowledgement character after transmission

– RK512

20	Handling or variable DB not generated or wrong length
21	Unknown DB number in the SEND command
22	Unknown telegram code in the SEND command
23	Error in the special command
24	Unknown DB number in the FETCH command
25	Unknown telegram code in the FETCH command
26	No SEND or FETCH telegram
27	No DB addressed
28	Character number > max. handling / or variable DB byte number; or character number > 138 bytes (max. 128 bytes data + 10 bytes RK512 header information)
30	Unplausible character number in 3964R procedure received
31	Unplausible character number in FETCH telegram
32	Unplausible character number in reaction telegram

Example for the RK512 mode:

Initializing the interface through DB1 and FB204

```
:L      KB    0
:E      DB    1        ;Deletion of the DB1
:L      KB    6        ;Length of the DB
:E      DB    1        ;Setting up the DB1
:A      DB    1        ;Activating the DB1
:L      KB    0
:T      DR    0        ;without meaning
:T      DL    0        ;without meaning
:L      KB    4        ;RK512 mode
:T      DR    1
:L      KB    1        ;Even parity
:T      DL    1
:L      KB    8        ;8 data bits
:T      DR    2
:L      KB    1        ;1 stop bit
:T      DL    2
:L      KB    0        ;No EOT
:T      DR    3
:L      KB    15       ;Interface monitoring active
:T      DL    3
:L      KD    19200    ;Baud rate (e.g. 19200 baud)
:T      DD    4

:SPA    FB    204      ;Initializing the interface and selection
                       ;of the RK512 mode
```

2.5.4 3964(R) mode (Mode = 5, 6 or Mode = 7, 8)

The transmission protocol 3964(R) from Siemens allows the coupling of two communication partners through a serial interface. It is an asynchronous protocol for equally entitled partners, i.e. each can send data on its own, without request by the other communication subscriber. The transmission protocol 3964R differs from the protocol 3964 due to an additionally sent block test character which makes the transmission safer.

The 3964 protocol is selected with Mode 5 and the 3964R protocol with Mode 6. The priority is "LOW".

As from AZ-PS4 version 02.11 it is possible to select the 3964 protocol with the Mode 7 and the 3964R protocol with the Mode 8; whereby in both modes the priority is "HIGH". (This is required, for instance, if two AMKASYN systems communicate by means of the protocol 3964(R).)

Caution: To guarantee the organization and the correct sequence of the transmission protocol, some protocol parameters are set permanently on the part of the AMKASYN system. These must agree for both subscribers, except for the priority.

Character delay time	220	ms
Acknowledgement delay time	2000	ms
Block waiting time	4000	ms
Number of the build-up attempts	6	
Number of the transmission attempts	6	
Priority	LOW / HIGH	
Block size	128	bytes

The protocol can be configured with "Interface mode = 5" or with "Interface mode = 6" (alternatively 7 or 8: see above) according to 3964 or 3964R respectively using the function block FB204 "Activate serial interface".

The individual data words of the current DB for activating the mode and setting the serial interface must be initialized as follows:

DR 0:	xx	Send mailbox DB number
DL 0:	yy	Receive mailbox DB number
DR 1:	5 / 6 (7 / 8)	3964 / 3964R mode, low priority (high priority)
DL 1:	0	No parity
DR 2:	8	8 data bits
DL 2:	1	1 stop bit
DR 3:	0	No EOT character
DL 3:	15	All monitorings are active
DD 4:	4800	Baud rate = 4800 baud

----
6 data words

Structure of the send and receive mailbox (DB)

The length of the send and receive mailbox can be maximum the length of a data block (256 words). The maximum data blocks to be sent or received are limited to 128 bytes. The data in the data block are filed word-oriented in the Intel format. This means that firstly the less significant byte (DR = data word right byte) and then the more significant byte (DL = data word left byte) stands. The length of the data package (number of the bytes to be sent or received) stands in the data word 0 (DW0). This must be entered in the send mailbox through the AWL program. The length results from the length of the received data package in the receive mailbox.

Organization form of the data blocks for sending (send mailbox) and receiving (receive mailbox):

Data word	**High byte**	**Low byte**
0	Length of the data package (number of the useful data in bytes)	
1	Useful datum 2	Useful datum 1
..	..	..
n/2	Useful datum n	Useful datum n-1
---------		
n/2 + 1		

n: Block length (even number, in bytes)

Organization form and function of the flag bytes MB235 and MB236

The flag bytes MB 235 (SBS = send control byte) and MB 236 (SBE = receive control byte) are responsible for coordinating the transmission and reception of data.

Organization form MB 235

MB 235	SBS	
.6 .. .0		Error code (0: No error in last data transfer)
.7	x	Send mailbox access right 0: Program can access send mailbox (send data can be entered in DB) 1: Operating system can access send mailbox (send data are sent from DB)

x: 0, 1

Function of MB 235

The data package of the send mailbox is sent with "M 235.7 = 1". After successful transmission the bit is reset by the operating system. If an error occurs during the transmission, then the error code (see below) is filed in the bits 0 to 6.

Organization form of MB 236

MB 236	SBE	
.6 .. .0		Error code (0: No error in last data transfer)
.7	x	Receive mailbox access right 0: Program can access receive mailbox (received data can be read from DB) 1: Operating system can access receive mailbox (received data are entered in DB)

x: 0, 1

Function of MB 236

A data package can be received with "M 236.7 = 1". After successful reception and after entry of the data in the receive mailbox DB, the bit is reset by the operating system. If an error occurs during the transmission, then the error code (see below) is filed in the bits 0 to 6.

The individual error codes of the MB 235 or MB 236 have the following meaning:

1	Timeout of the character delay time on reception
2	Timeout of the character delay time on transmission
3	Timeout of the acknowledgement delay time
4	No STX received
5	Faulty telegram length
6	Wrong acknowledgement on STX after 6 times repetition
7	Wrong acknowledgement on STX
8	Wrong acknowledgement for transmission after 6 times repetition
9	Wrong acknowledgement for transmission
10	No ETX received
11	Faulty transmission
12	Character error on reception (fault)
13	Inadmissible acknowledgement character after STX
14	Inadmissible acknowledgement character after transmission
20	EF or SF DB not generated or inadmissible length
28	Character number > max. EF / or SF DB byte number; or character number > 138 bytes (max. 128 bytes data + 10 bytes RK512 header information)

Programming

Initially the send and receive mailbox DBs must be set up for the longest protocol to be sent or received. The serial interface must then be initialized by means of FB204, the 3964(R) mode selected and send and receive mailbox DB numbers allocated (see above). The flags MB 235 (SBS) and MB 236 (SBE) take over the control of the interface.

After the FB204 selection "M 236.7 = 1" must be set, so that data can be received. If a complete data package has been received, then this is identified by "M 236.7 = 0" by the operating system. The data can now be taken out from the receive mailbox. The length of the received data package (in bytes) is entered in data word DW0.

Data are sent in an analogous manner. The data to be sent as well as the byte number of the data to be sent (in DW0) are entered in the send mailbox DB. These data are sent by setting "M 235.7 = 1". After successful transmission of the data "M 235.7 = 0" is set by the operating system.

If errors occur during the transmission, then an "Error code unequal to 0" is entered in the bits 0 to 6 of the relevant flag byte. After reading the error code, the flag byte must be deleted ("MB 235 = 0" or "MB 236 = 0").

The PS module is always set in the slave mode (priority = LOW). If it can happen that both subscribers want to send simultaneously (collision), the higher ranking communication subscriber must be switched into the master mode (priority = HIGH).

2.5.5 Modbus mode (Mode = 9)

As from AZ-PS4 version 02.13 it is possible to communicate with an operator panel (touch panel, type "VTxxx") of the ESA company.

"Telegrams" are sent in the Modbus mode (see also description of operator panel start-up software "VTWIN" of the ESA company) and allows the coupling of an operator panel with a PS4 (RS422) .
The connection between PS4 and operator panel is activated through function block FB204 "Activate serial interface" with "Interface mode = 9".

The individual data words of the current DB for activating the mode and the setting of the serial interface must be initialized as follows:

DR 0:	16	Variable DB number (e.g. DB 16)
DL 0:	1	Slave address (e.g. 1)
DR 1:	**9**	**Modbus mode**
DL 1:	1	Parity (e.g.: even parity)
DR 2:	8	Data bits (e.g.: 8 data bits)
DL 2:	1	Stop bits (e.g.: 1 stop bit)
DR 3:	**0**	**No EOT character**
DL 3:	**15**	**All monitorings are active**
DD 4:	9600	Baud rate (e.g. 9600 baud)

----
6 data words

The Modbus mode as well as DR3 and DL3 must be assigned the above values. All other parameters can be selected within the scope of the permitted value range (cf. Section 2.5).

Caution: It must be observed that the settings of the serial interface correspond to those of the operator panel interface.

In the "Mode = 9" transmission errors are reported to the PS program through the flag byte MB 235. These errors must be acknowledged by the AWL program in the PS by deleting the flag byte. (MB 235 = 0).

Remarks*:* These errors are not displayed on the AMKASYN operator panel (AZB). The PS goes into the error status!

The individual error codes of the MB 235 have the following meaning:

Error No. 0: no error

Error No. 1..31: bit-coded
- Bit 0 =1: Parity error
- Bit 1 =1: Framing error
- Bit 2 =1: Overrun error
- Bit 3 =1: Error in receive FIFO
- Bit 4 =1: BREAK condition

Error No. 32: Modbus CRC error
33: Send buffer full
34: Inadmissible data length Read n Words
35: Inadmissible data length Write n Words

36: Inadmissible data length Read n Bits
37: Inadmissible data length Write 1 Bit
38: Inadmissible address value Read n Words
39: Inadmissible address value Write n Words
40: Inadmissible address value Read n Bits
41: Inadmissible address value Write 1 Bit
42: Inadmissible function code
43: Receive buffer overflow

Read / write areas for the Modbus mode:

- Transmission telegrams READ n WORDS and WRITE OF n WORDS:
  n words starting from the word address of the data block which has been initialized as variable DB No. (DR 0) by means of function block FB 204 stated in the telegram is read or written.
  The permissible address area is determined by the length of the created variable DB.

- Transmission telegrams READ DI n BITS and WRITE OF 1 BIT:
  n bits or 1 bit starting from the bit address stated in the telegram are read or written.
  The bit address area lies between 0 and 511 and corresponds to the flag byte area of the PS from MB 128 to MB 191 (bit address 0 corresponds to M 128.0, bit address 1 corresponds to M 128.1, .., bit address 8 corresponds to M 129.0, .., bit address 511 corresponds to M 191.7.).

Example for the Modbus mode:
Initializing the interface through DB1 and FB204

```
:L      KB   0
:E      DB   1      ;Deletion of the DB1
:L      KB   6      ;Length of the DB
:E      DB   1      ;Setting up the DB1
:A      DB   1      ;Activating the DB1

:L      KB   16     ;Send and receive DB number
:T      DR   0
:L      KB   1      ;Slave address (own address)
:T      DL   0
:L      KB   9      ;Modbus mode
:T      DR   1
:L      KB   1      ;Even parity
:T      DL   1
:L      KB   8      ;8 data bits
:T      DR   2
:L      KB   1      ;1 Stop bit
:T      DL   2
:L      KB   0      ;No EOT
:T      DR   3
:L      KB   15     ;Interface monitoring active
:T      DL   3
:L      KD   9600   ;Baud rate
:T      DD   4

:SPA    FB   204    ;Initializing the interface and selection
```

;of the Modbus mode

2.5.5.1 Remarks on using the operator panel VTxxx:

Refer to the operating instructions of the ESA company for the description of the operator panel start-up software (VTWIN) for the "VTxxx".

Load operator panel project:

1. Connect VTxxx with the PC through the "VTxxx↔PC" connection cable.

2. Start the ESA operator panel start-up software VTWIN, load project and possibly translate.

3. Switch on VTxxx and after booting of the display simultaneously touch 2 diagonally located corner points of the touch panel (VTxxx) as well as afterwards the touch field "TRAN PAGE".

4. After "WAITING FOR DOWNLOAD" appears on the panel, trigger "Load project" on PC (VTWIN).

5. After successful loading the display of the first page of the project appears on the panel.

Connection cable

- Connection "VTxxx ↔ PC" through MSP ↔ COMx-PC interface (RS232C): an assembled cable is delivered by ESA for this.
- Connection "VTxxx ↔ PS" through MSP ↔ X72-AZ-PS4 interface (RS422) according to following pin assignment:

VTxxx (MSP), D-SUB 25-pin)			**AMKASYN PS (X72, D-SUB 15-pin)**	
Pin 23	Tx+		Rx+	Pin 13
Pin 12	Tx-		Rx-	Pin 3
Pin 13	Rx+ 220		Tx+	Pin 12
Pin 24	Rx- Ohm		Tx-	Pin 2
Shield applied to metallized plug housing			Shield applied to metallized plug housing	
Pin 25	GND			
Pin 7	GND		GND	Pin 7
Pin 4	RTS		RTS+	Pin 4
Pin 5	CTS		CTS+	Pin 5
Pin 15			RTS-	Pin 14
Pin 18			CTS-	Pin 15

The signal wire pairs "Tx+" and "Tx-" as well as "Rx+" and "Rx-" or TxRx+ and TxRx- must be in each case twisted, with a minimum cross-section of 0.14 mm^2 and with a common shield.
The shield must be applied at both ends to the metallized plug housing.

The type "UNITRONIC Li2YCY (TP)" from the Lapp company is recommended as cable.

Notes:

- A metallized D-SUB plug housing with lateral cable outlet must be used on the AMKASYN PS side.
- The housing of the VTxxx must be connected with HF-wise good PE connection with the PE star point of the AMKASYN system.

2.6 Sending or receiving through serial interface in Mode 1 and 2

Sending or receiving in Mode 1 and 2 is under use of the DB for sending or receiving as well as evaluating the control byte SBS or SBE. Furthermore there exists one function block each for sending (FB205) and receiving (FB206) characters. These FBs can be selected after initialization through FB204 (Interface mode = 1 or 2). One parity bit can be generated or checked per character for data protection. Both 7-bit ASCII (7 data bits mode and even parity) as well as 8-bit values (8 data bit mode, with even or without parity) can be sent as useful data.

As from version AZ-PS5 V02.14/3500 it is possible to initialize a second "Serial interface" (RS232) in Mode 1 with the FB214 (cf. Section 2.5). Sending or receiving through the second interface must be organized from the viewpoint of the user program analogously to the previous interface. In contrast to the first interface, only FB215 must be used instead of FB205 and FB216 instead of FB206. Further MB208 must be used instead of the SBS MB235 as well as MB209 instead of the SBE MB236.

2.6.1 FB205 / FB215 "Sending through serial interface"

Prerequisite:

Successful initialization by FB204 (with interface mode = 1 or 2 in the active DB with the word length 6) or with FB214 (with interface mode = 1 in the active DB with the word length 6).

Procedure:
Step (1): Testing flag bit 7 of the send control byte (SBS)
1: Preceding send order is not yet processed,
the send mailbox may not yet be accessed
on the part of the program. > Step (7)
0: The send mailbox is available for the program side. > Step (2)

Step (2): Enter length of the useful data package to be sent in "DW 0"
of the send mailbox.
Enter useful data as from "DW 2" (byte 0) continuously. > Step (3)

Step (3): Set flag bit 7 of the SBS to "1" to enable the send mailbox
for the operating system. > Step (4)

Step (4): Select function block FB205 for triggering the send process.
(Bit 7 is reset to "0" by the operating system after processing
the send order. Further the error bit 0 of the SBS is set to "1"
on occurrence of an error.) > Step (5)

Step (5): Testing flag bit 7 of the SBS
1: Preceding send order is not yet processed. > Step (7)
0: The send mailbox is available for the program side. > Step (6)

Step (6): Testing flag bit 0 of the SBS
1: Preceding send order was faulty. > Step (8)
0: Preceding send order was performed correctly. > Step (7)

Step (7): Further with cyclic program processing.

Step (8): Error handling (reset flag bit 0 of the SBS to "0")

2.6.2 FB206 / FB216 "Receiving through serial interface"

Prerequisite:

Successful initialization by FB204 (with interface mode = 1 or 2 in the active DB with the word length 6) or with FB214 (with interface mode = 1 in the active DB with world length 6).

Procedure:
Step (1): Testing flag bit 7 of the receive control byte
1: Preceding receive order is not yet processed,
the receive mailbox may not yet be accessed
writing on the part of the program. > Step (7)
0: The receive mailbox is available for the program side. > Step (2)

Step (2): Enter length of the newly to be received useful data package in "DW 0"
of the receive mailbox. (FB 206 transmits, corresponding to the
size stated in "DW 0" or until the EOT character, e.g. "\r" (0x0D),
is detected, received characters in the receive mailbox). > Step (3)

Step (3): Set flag bit 7 of the SBE for enabling the receive mailbox for the
operating system to "1". > Step (4)

Step (4): Select function block FB206 for triggering the receive process.
(Bit 7 is reset to "0" by the operating system after processing
the receive order. Further on occurrence of an error the error bit 0
of the SBE is set to "1", if the corresponding error was enabled
on activating the interface). > Step (5)

Step (5): Testing flag bit 7 of the SBE
1: Preceding receive order is not yet processed,
the receive mailbox may not be accessed writing
on the part of the program. > Step (7)
0: The receive mailbox is available for the program side. > Step (6)

Step (6): Testing flag bit 0 of the SBE
1: Preceding receive order was faulty. > Step (8)
0: Preceding receive order was performed correctly.
Enter number of the received useful data in "DW 0" of the receive
mailbox. Useful data are entered continuously starting
from "DW 2" (Low Byte) and can be interpreted on the part of the
AWL program. > Step (7)

Step (7): Further with cyclic program processing.

Step (8): Error handling (reset flag bit 0 of the SBE to "0".)

The following AWL program examples serve for illustrating the mode of operation of the FB204..FB206 or FB214..FB216, whereby reference is made to the previously described PB00 (activate serial interface):

Example 1: First serial interface (RS422)

Assignment list

```
;Assignment list
;for example FB 204..FB206 (transmitt, receive through serial interface)

;Inputs
E_SerA     E  0.0          ;Activate serial interface
                           ;at 0->1
E_SerS     E  0.1          ;Send through serial interface
                           ;at 0->1
E_SerE     E  0.2          ;Receive through serial interface
                           ;at 0->1

;Outputs
A_FehlS    A  0.0          ;Error indication on send error
A_FehlE    A  0.1          ;Error indication on receive error

;Flags
M_SteuS    M  0.0          ;Send control flag
M_SteuSÜ   M  0.1          ;Send monitoring control flag
M_SteuE    M  0.2          ;Receive control flag
M_SteuEÜ   M  0.3          ;Receive monitoring control flag
M_FlankA   M  0.4          ;Activate edge flag
M_FlankS   M  0.5          ;Send edge flag
M_FlankE   M  0.6          ;Receive edge flag
```

Cyclic organization block OB01

```
;OB01
;Send/receive through serial interface

:U         $E_SerA         ;Input for activating
:UN        $M_FlankA       ;Edge flag
:S         $M_FlankA       ;Set edge flag
:SPB       PB 00           ;Activate interface at 0->1
:UN        $E_SerA
:R         $M_FlankA       ;Reset edge flag

:U         $E_SerS         ;Activate input for sending
:UN        $M_SteuS        ;Sending not active
:UN        $M_SteuSÜ       ;Send monitoring not active
:UN        $M_FlankS       ;Edge flag
:S         $M_FlankS       ;Set edge flag
:S         $M_SteuS        ;Receive through interface at 0->1
:UN        $E_SerS
:R         $M_FlankS       ;Reset edge flag

:U         $E_SerE         ;Activate input for receiving
:UN        $M_SteuE        ;Receiving not active
```

```
:UN     $M_SteuEÜ    ;Receive monitoring not active
:UN     $M_FlankE    ;Edge flag
:S      $M_FlankE    ;Set edge flag
:S      $M_SteuE     ;Send through interface at 0->1
:UN     $E_SerE
:R      $M_FlankE    ;Reset edge flag

:U      $M_SteuS
:SPB    PB 01        ;Send through interface

:U      $M_SteuSÜ
:SPB    FB 01        ;Send monitoring

:U      $M_SteuE
:SPB    PB 02        ;Receive through interface

:U      $M_SteuEÜ
:SPB    FB 02        ;Receive monitoring
:BE
```

Program block PB01

```
;PB01
;Sending (FB 205)

:U      M   235.7     ;Send order not yet
:BEB                  ;processed

:S      M   235.7     ;Access FB205
:SPA    FB  205       ;Send contents of send mailbox

:R      $M_SteuS      ;Reset sending
:S      $M_SteuSÜ     ;Set send monitoring

:BE
```

Program block PB02

```
;PB02
;Receiving (FB 206)

:U      M   236.7     ;Receive order not yet
:BEB                  ;processed

:A      DB  24        ;Select receive mailbox
:L      KB  16
:T      DW  0         ;Receive maximum 16 useful data
                      ;or up to EOT character

:S      M   236.7     ;Access FB206
:SPA    FB  206       ;Enter received characters in receive
                      ;mailbox

:R      $M_SteuE      ;Reset receiving
:S      $M_SteuEÜ     ;Set receive monitoring

:BE
```

Function block FB01

```
;FB01
;Send monitoring

:U      M   235.7     ;Send order not yet
:BEB                  ;processed

:R      $M_SteuSÜ     ;Reset send monitoring
:R      $A_FehlS      ;Reset send error

:U      M   235.0
:S      $A_FehlS      ;Send error
:R      M   235.0     ;Reset error flag
                      ;Characters to be sent, corresponding
                      ;to the DW0 of the send mailbox,
                      ;can be entered
:BE
```

Function block FB02

```
;FB02
;Receive monitoring

:U      M   236.7     ;Receive order not yet
:BEB                  ;processed

:R      $M_SteuEÜ     ;Reset receive monitoring
:R      $A_FehlE      ;Reset receive error

:U      M   236.0
:S      $A_FehlE      ;Receive error
:R      M   236.0
                      ;Received characters, corresponding
                      ;to DW0 of the receive mailbox,
                      ;can be read

:BE
```

Example 2: Second serial interface (RS232; only in connection with AZ-PS5)

Assignment list

```
;Assignment list
;for example FB 214..FB216 (send, receive through serial interface)

;Inputs
E_SerA     E  0.0     ;Activate serial interface
                      ;at 0->1
E_SerS     E  0.1     ;Send through serial interface
                      ;at 0->1
E_SerE     E  0.2     ;Receive through serial interface
                      ;at 0->1

;Outputs
A_FehlS    A  0.0     ;Error indication on send error
A_FehlE    A  0.1     ;Error indication on receive error

;Flags
M_SteuS    M  0.0     ;Send control flag
M_SteuSÜ   M  0.1     ;Send monitoring control flag
M_SteuE    M  0.2     ;Receive control flag
M_SteuEÜ   M  0.3     ;Receive monitoring control flag
M_FlankA   M  0.4     ;Activate edge flag
M_FlankS   M  0.5     ;Send edge flag
M_FlankE   M  0.6     ;Receive edge flag
```

Cyclic organization block OB01

```
;OB01
;Send/receive through serial interface

:U      $E_SerA     ;Input for activating
:UN     $M_FlankA   ;Edge flag
:S      $M_FlankA   ;Set edge flag
:SPB    PB 00       ;Activate interface at 0->1
:UN     $E_SerA
:R      $M_FlankA   ;Reset edge flag

:U      $E_SerS     ;Activate input for send
:UN     $M_SteuS    ;Send not active
:UN     $M_SteuSÜ   ;Send monitoring not active
:UN     $M_FlankS   ;Edge flag
:S      $M_FlankS   ;Set edge flag
:S      $M_SteuS    ;Receive through interface at 0->1
:UN     $E_SerS
:R      $M_FlankS   ;Reset edge flag

:U      $E_SerE     ;Activate input for receive
:UN     $M_SteuE    ;Receive not active
:UN     $M_SteuEÜ   ;Receive monitoring not active
:UN     $M_FlankE   ;Edge flag
:S      $M_FlankE   ;Set edge flag
:S      $M_SteuE    ;Send through interface at 0->1
:UN     $E_SerE
:R      $M_FlankE   ;Reset edge flag

:U      $M_SteuS
:SPB    PB 01       ;Send through interface

:U      $M_SteuSÜ
:SPB    FB 01       ;Send monitoring

:U      $M_SteuE
:SPB    PB 02       ;Receive through interface

:U      $M_SteuEÜ
:SPB    FB 02       ;Receive monitoring
:BE
```

Program block PB01

```
;PB01
;Sending (FB 215)

:U      M   208.7      ;Send order not yet
:BEB                   ;processed

:S      M   208.7      ;Access FB215
:SPA    FB  215        ;Send send mailbox contents

:R      $M_SteuS       ;Reset send
:S      $M_SteuSÜ      ;Set send monitoring

:BE
```

Program block PB02

```
;PB02
;Receiving (FB 216)

:U      M   209.7      ;Receive order not yet
:BEB                   ;processed

:A      DB  24         ;Select receive mailbox
:L      KB  16
:T      DW  0          ;Receive maximum 16 useful data
                       ;or up to EOT character

:S      M   209.7      ;Access FB216
:SPA    FB  216        ;Enter received characters in
                       ;receive mailbox

:R      $M_SteuE       ;Reset receive
:S      $M_SteuEÜ      ;Set receive monitoring

:BE
```

Function block FB01

```
;FB01
;Send monitoring

:U      M   208.7      ;Send order not yet
:BEB                   ;processed

:R      $M_SteuSÜ      ;Reset send monitoring
:R      $A_FehlS       ;Reset send error

:U      M   208.0
:S      $A_FehlS       ;Send error
:R      M   208.0      ;Reset error flag
                       ;Characters to be sent,
                       ;corresponding to DW0 of the send mailbox,
                       ;can be entered
:BE
```

Function block FB02

```
;FB02
;Receive monitoring

:U      M   209.7     ;Receive order not yet
:BEB                  ;processed

:R      $M_SteuEÜ     ;Reset receive monitoring
:R      $A_FehlE      ;Reset receive error

:U      M   209.0
:S      $A_FehlE      ;Receive error
:R      M   209.0
                      ;Received characters,
                      ;corresponding to DW0 of the receive mailbox,
                      ;can be read

:BE
```

2.7 FB207 "Initializing fast function SF"

FB207 is used for initializing fast functions SF. It facilitates

- integrating the SF in the call that is time-synchronous with the drive control,
- interfacing the SF to the internal data storage box of the drives or to the E/A/M image or to further SF
- as well as interfacing the SF to the binary E/A signals or to the flag image.

As from the AZ-PS4 software level AZ-PS4-V02.09 it is possible to delete already initialized SF (see below: SF type=255) and / or to initialize anew SF under the same SF number.

As from version AZ-PS5 V02.14/3500 it is possible to operate on SF by means of extended source and sink addresses through the CAN bus. (Prerequisite is an AZ-PS5-C option module.)

The information required for this is transferred in the current DB to the FB207:

Data double word	High word H byte	High word L byte	Low word H byte	Low word L byte
0	SF cycle time factor	E/A-M mode	SF number	SF type
2	Sink address	Sink address offset	Source address	Source address offset
4	Output Byte address	Output bit mask	Input byte address	Input bit mask
6	DB number Tab3	DB number Tab2	DB number Tab1	DB number Tab0
8	DB number Tab7	DB number Tab6	DB number Tab5	DB number Tab4
-------------- 10 data words				

SF type (DR 0)
– Meaning: SF type
– Value range: 0 .. SF type-Max (SF type-Max depends upon the AZ-PSx version; cf. documentation: Fast functions (SF))
255 A SF already initialized under this SF number is deleted (deinitialized).

SF number (DL 0)
– Meaning: Number under which the SF is processed cyclically (for the cyclic call it applies that: 0 = first function; 1 = second function; ...)
– Value range: 0 to SF-Nr-Max (cf. "Abbreviations and designations")

E/A-M mode (DR 1)
– Meaning: Binary signal selection
– Value range: 0 .. 3
bit-coded:
– Bit 0 = 0: Binary input in the E image
= 1: Binary input in the M image
– Bit 1 = 0: Binary output in the A image
= 1: Binary output in the M image

SF cycle time factor (DL 1)
– Meaning: Factor by which the cycle time of the SF based on ID 2 is lengthened.
– Remarks: The SF cycle time factor facilitates e.g. combining a SF "FIPW" in the time grid of 1ms (ID2=1; DL1=0) with a SF "REGL" in the grid of 20ms (ID2=1; DL1=20).
– Value range: 0: Factor formation ineffective (factor = 1)
1 .. 255: Factor = 1 .. 255

Source address offset (DR 2)
– Meaning: Selection of the input value channel (related to the source according to DL2; source address)
– Value range:

Source address=0:	1, 3, 5, 7	AZ message 1, .., 4 (High-Word)
	16,18,20,22	AZ message 1, .., 4 (Low- and High-Word)
Source address=1..8:	2	Configurable 16-bit value according to ID 32785
	3	Low-Word of the configurable 32-bit value according to ID 32786
	5	Low-Word of the 32-bit position feedback value
	19	Configurable 32-bit value according to ID 32786
	21	32-bit position feedback value
Source address=128:	0..126	MW 0 .. MW 126
Source address=129:	0..124	MD 0 .. MD 124
Source address=130:	0..126	EW 0 .. EW 30
Source address=131:	0 124	ED0 .. ED28
Source address=132:	0..254	OEW 0 .. OEW 254
Source address=133:	0 252	OED 0 .. OED 252
Source address=255:	0..15	Internal output value of SF0 to SF15

Source address (DL 2)

– Meaning: Selection of the input value source (AZ., AW, E/M, OE or SF)

– Value range:

0	AZ
1..8	AW1 .. AW8
128	MW (Flag range; word format)
129	MD (Flag range; double word format)
130	EW (Input range; word format)
131	ED (Input range; double word format)
132	OEW (Option input range; word format)
133	OED (Option input range; double word format)
255	SF

– Remarks: Currently supported option input range = CAN input range

Sink address offset (DR 3)

– Meaning: Selection of the output value channel (related to the sink according to DL3; see under sink address)

– Value range:

Sink address=0:	0	No sink assigned
Sink address=1..8:	4	16-bit command value (SWQ1)
	16:	32-bit command value (SWQ2)
	18:	EDG reference input variable (only in connection with SF EDG)
Sink address=64:	0..3	AA1 .. AA4
Sink address=128:	0..126	MW 0 .. MW 126
Sink address=129:	0..124	MD 0 .. MD 124
Sink address=132:	0..254	OAW 0 .. OAW 254
Sink address=133:	0 252	OAD 0 .. OAD 252
Sink address=255:	0	Internal output value sink (without AW feed through)

Sink address (DL 3)

– Meaning:	Selection of the output value sink (AZ., AW, A/M, OA, or SF)	
– Value range:	0	AZ
	1.. 8	AW1 .. AW8
	64	AAx (Analog output)
	128	MW (Flag range; word format)
	129	MD (Flag range; double word format)
	132	OAW (Option output range; word format)
	133	OAD (Option output range; double word format)
	255	SF
– Remarks:	Currently supported option output range = CAN output range	

Input bit mask (DR4)

– Meaning:	Marking the position of the first SF input bit in the input byte according to DL4	
– Value range:	0	No input bit assigned
	1,2,4,8,16,32,64,128	First input bit of the SF is input bit 0, 1, 2, 3, 4, 5, 6, 7

Input byte address (DL 4)

– Meaning:	Number of the input or flag byte for SF binary inputs	
– Value range:	E/A-M mode, Bit 0 = 0	
	0 ..31:	EB 0 .. EB 31
	E/A-M mode, Bit 0 = 1	
	0 ..127:	MB 0 .. MB 127

Output bit mask (DR 5)

– Meaning:	Marking the position of the first SF output bit in the output byte according to DL 4	
– Value range:	0	No output bit assigned
	1,2,4,8,16,32,64,128	First output bit of the SF is output bit 0, 1, 2, 3, 4, 5, 6, 7

Output byte address (DL 5)

– Meaning:	Number of the output or flag byte for SF binary outputs	
– Value range:	E/A-M mode, Bit 1 = 0	
	0 ..31:	AB 0 .. AB 31
	E/A-M mode, Bit 1 = 1	
	0 ..127:	MB 0 .. MB 127

DB number Tab0, .., DB number Tab7 (DR 6, DL 6, .., DR 9, DL 9)

– Meaning:	DB number, of the Table 0 to 7 used by the SF.	
– Value range:	0	No table assigned
	1..63	DB number 1 .. 63

2.7.1 Example

Task*:* The fast function of the type 1 (FIPW) should be effective in the AW as distance-time function. (With regard to the calculation of the table interpolation points cf. Section 2.10.)

```
;OB22
;for example FB 207 (SF initialization; in the course of the start OB22)

;Table value calculation (distance-time table)
          :L        KB  0
          :E        DB  35      ;Delete, if created
          :L        KF  256
          :E        DB  35      ;Generate DB35 for distance-time table
                                ;with 127 interpolation points (256 words)

          :A        DB  34      ;Activate DB for calculating the
                                ;distance-time table
          :SPA      FB  210     ;Table value calculation

          :A        DB  28      ;SF-init DB
          :SPA      FB  207

          :BE


;DB28
;generated with APROS

          :KB 1                 ;(DR0) SF type (FIPW)
          :KB 0                 ;(DL0) SF number (SF0)
          :KB 0                 ;(DR1) E/A-M mode
          :KB 0                 ;(DL1) SF cycle time factor
          :KB 0                 ;(DR2) Source address offset (assigned
                                ;       for distance-time)
          :KB 0                 ;(DL2) Source address (assigned
                                ;       for distance-time)
          :KB 4                 ;(DR3) Sink address offset (16-bit
                                ;       command value)
          :KB 1                 ;(DL3) Sink address (AW1)
          :KB 1                 ;(DR4) E bit mask (E-Bit 0)
          :KB 8                 ;(DL4) E byte address (EB 8)
          :KB 32                ;(DR5) A bit mask (A-Bit 4)
          :KB 8                 ;(DL5) A byte address (AB 8)
          :KB 35                ;(DR6) Tab0 DB No (DB35-distance-
                                ;      time table)
          :KB 0                 ;(DL6) Tab1 DB No
          :KB 0                 ;(DR7) Tab2 DB No
          :KB 0                 ;(DL7) Tab3 DB No
          :KB 0                 ;(DR8) Tab4 DB No
          :KB 0                 ;(DL8) Tab5 DB No
          :KB 0                 ;(DR9) Tab6 DB No
          :KB 0                 ;(DL9) Tab7 DB No
```

```
;DB34
;DB for table value calculation;
;generated with APROS

        :KB 4            ;Tab. type (distance-time table)
        :KB 0            ;Init. type (acceleration time
                         ;default)
        :KB 35           ;DB No. (DB35 = distance-time table)
        :KB 0            ;Reserve
        :KF 128          ;Number of the table elements
                         ;(127 interpolation points)
        :KF 0            ;Reserve
        :KD 20000        ;Output interval (20000 incr. per
                         ;cycle time)
        :KF 20000        ;Encoder resolution (20000 incr. per
                         ;revolution)
        :KF 0            ;Reserve
        :KD 1000         ;Para1: Cycle time (1000 ms)
        :KD 10           ;Para2: Acceleration time
                         ;(10% cycle time)
        :KD 100000       ;Para3: max. acceleration
                         ;(100 U/ss)
        :KD 0            ;Para4: Reserve
        :KD 0            ;Para5: Output of |v|max
                         ;(10000=1rpm)
        :KD 0            ;Para6: Output of |r|max
                         ;(1=1U/sss)
```

2.8 FB208 "Commanding fast function SF"

FB 208 is used for commanding initialized fast functions SF. The required information for this must be transferred in the current DB at FB selection:

Data double word	High word H byte	High word L byte	Low word H byte	Low word L byte
0	SFKMD-BA	SFKMD code	SFKMD fct.	SF number
2	Parameters			
.	.			
.	.			
.	.			
32				

----
34 words

SF number (DR 0)
- 0 .. SF-Nr-Max: Position of the SF to be commanded with regard to the cyclic call (0: first function .. 15: last function; from AZ-PS4V02.09 the max. possible number of fast functions "SF-Nr-Max + 1" was increased to 16: - cf. documentation description of fast functions SF)

SFKMD fct (DL 0)
- 0 : No selection required
- Other: Selection of a SF type specific version

SFKMD code (DR 1)
- 0: Reset
- 1: Start
- 5: Value new (SF type dependent)
- 6: Value reading (SF type dependent)
- 255:Selection SF status and error

SFKMD-BA (DL 1)
- 0 : No BA differentiation required
- Other: SF type specific operating mode

Parameters (DW 2 .. DW 33)
- SF type, SFKMD fct and SFKMD-BA dependent parameters in DW or DD format

The SF number, the status as well as the error byte of the last commanded SF are displayed in the flags MB 231 .. MB 233. It is possible to switch over to the display of the status and error byte of the corresponding SF number by entry of a permissible SF number in MB 231 (SF must be initialized) or by SF commanding with SFKMD code = 255.

The meaning of the individual bit information is defined as follows:

SF number (MB 231: commanding SF)

MB 231		
.0.. .2	x	SF number: 0..15
.3	n	Currently not used
.4	n	Currently not used
.5	n	Currently not used
.6	n	Currently not used
.7	n	Currently not used

– Status (MB 232: commanding SF)

MB 232		
.0	n	Currently not used
.1	x	1: SF-ACTIVE
.2	n	Currently not used
.3	n	Currently not used
.4	n	Currently not used
.5	n	Currently not used
.6	x	1: SF error
.7	x	FB208 selection not yet processed

– Error (MB 233: commanding SF)

<table>
<tr><td>MB 233</td><td colspan="3"></td></tr>
<tr><td>.0</td><td>x</td><td rowspan="8">= 0
no error</td><td rowspan="8">≠ 0
Error</td></tr>
<tr><td>.1</td><td>x</td></tr>
<tr><td>.2</td><td>x</td></tr>
<tr><td>.3</td><td>x</td></tr>
<tr><td>.4</td><td>x</td></tr>
<tr><td>.5</td><td>x</td></tr>
<tr><td>.6</td><td>x</td></tr>
<tr><td>.7</td><td>x</td></tr>
</table>

M 232.7 is set independently with call of the FB208 and reset automatically on transfer of the current SF number, SF status as well as SF error byte into the process image.

The following AWL program example, which is based on SF initialization according to Section 2.7 and a table value calculation according to Section 2.10 serves to illustrate the mode of operation of FB208:

Assignment list

```
;Assignment list
;for example FB 208 (SF commanding: Start
;with regard to SF number = 0)

;Inputs
E_SFKmd         E  0.0          ;SF commanding start at 0->1

;Outputs
A_Fehl          A  0.0          ;Error indication

;Flags
M_SteuKÜ        M  0.0          ;SF commanding monitoring control flag
M_FlankK        M  0.1          ;SF commanding edge flag
```

Cyclic organization block OB01

```
;OB01
;SF0-commanding Start

:U              $E_SFKmd        ;Input for SF commanding
:UN             $M_SteuKÜ       ;SF commanding monitoring not active
:UN             $M_FlankK       ;Edge flag
:S              $M_FlankK       ;Set edge flag
:SPB            PB 01           ;SF commanding at 0->1
:UN             $E_SFKmd
:R              $M_FlankK       ;Reset edge flag


:U              $M_SteuKÜ
:SPB            FB 01           ;SF commanding monitoring

:BE
```

Program block PB01

```
;PB01
;SF commanding: Start regarding SF number = 0

:A              DB  29          ;SF commanding DB
:SPA            FB  208         ;SF commanding

:O              M   0.0
:ON             M   0.0
:S              $M_SteuKÜ       ;Set SF commanding monitoring

:BE
```

Function block FB01

```
;FB01
;SF commanding monitoring

:U        M   232.7     ;SF commanding not yet acknowledged
:BEB

:R        $MSteuKÜ      ;Reset SF commanding monitoring

:L        MB  232
:L        KB  2         ;SF active
:><FD
:=        $A_Fehl

:BE
```

Data block DB29

```
;DB29
;DB for SF commanding (Start SF0 = SF type 1), generated with APROS

:KB 0       ;(DR0) SF number                = 0  (FIPW corresponds to
                                                  SF initialization)
:KB 0       ;(DL0) SFKMD fct                = 0
:KB 1       ;(DR1) SFKMD code               = 1  (Start)
:KB 132     ;(DL1) SFKMD-BA                 = 132(Start with zeroing)
:KF 0       ;(DW2) SAK factor               = 0  (without SAK)
:KF 2       ;(DW3) SF cycle time            = 2  (1ms)
:KF 1       ;(DW4) Total table number       = 1  (1 table)
:KF 0       ;(DW5) Start table number       = 0
:KF 0       ;(DW6) Division exponent        = 0  (/ 1)
:KF 0       ;(DW7) Reserved= 0
:KD 1       ;(DD8) Multiplication factor    = 1  (* 1)
:KD 2000    ;(DW10) Cycle time              = 2000  (1000 ms)
:KF 0       ;(DW12) Master offset angle     = 0  (incr.)
:KF 0       ;(DW13) Reserved                = 0
:KF 0       ;(DW14) Start angle             = 0  (incr.)
:KF 0       ;(DW15) Reserved                = 0
:KF 0       ;(DW16) Stop angle              = 0  (incr.)
:KF 0       ;(DW17) Reserved                = 0
:KD 0       ;(DD18) Reserved                = 0
:KD 0       ;(DD20) Reserved                = 0
:KD 0       ;(DD22) Reserved                = 0
:KD 0       ;(DD24) Reserved                = 0
:KD 0       ;(DD26) Reserved                = 0
:KD 0       ;(DD28) Reserved                = 0
:KD 0       ;(DD30) Reserved                = 0
:KD 0       ;(DD32) Reserved                = 0
```

2.9 FB209 "Synchronization after BA change"

Caution: FB209 is no longer required as from version AZ-PS4V02.09 (all functions to be organized by the FB through the user program are performed automatically by the system software as from this software level). However, the FB can still be called for compatibility reasons. (However, the call is without effect! The following description is of significance thus only for older AZ-PSx software versions.)

This function block implements, in conjunction with a BA change, the synchronization of a drive with a new reference value input (e.g. position command value). In this way step changes in the reference input variable (e.g. at start of the SF) can be avoided in a BA change (e.g. for selecting a new command value source for a SF).

FB209 must always be used after "BA change with external synchronization" commanding. This means that after a FB201 selection with operating mode change (cf. documentation: "Commanding drive"):

- KMDFKT = 1,
- KMDBIT = 1,
- KMD_VAR1 = 0 (or KMD_VAR1 = 4).

FB209 requires a DB interface based on drive commanding and upwards compatible with regard to the BA change.

The structure of a data block which is required for calling FB209 looks as follows:

Data double word	**High word**		**Low word**	
	H byte	**L byte**	**H byte**	**L byte**
0	Reserve = 0	Reserve = 0	Reserve = 0	ANTRK
2	X	X	X	X
4	X			
6	Synchronous value			
8	Reserve = 0		Command value source	SF number
10	X			
12	X			
14	X			

----
16 data words

with X: arbitrary value

The meaning of the individual DB bytes is defined as follows:

ANTRK (DR 0): Drive identifier = AW number
- 0..7: corresponding to AW 1 to 8

Synchronous value (DD 6)
- $-2^{15}..+2^{15}-1$: Reference input command value after BA switch-over
  for 16-bit command value source (DL 8 = 0)
- $-2^{31}..+2^{31}-1$: Reference input command value after BA switch-over
  for 32-bit command value source (DL 8 = 1)

SF number (DR 8)
- 0..15: Number of the SF to be synchronized
  (Sink address of the SF initialized under the SF number
  must be compatible to the drive identifier; cf.
  Section 2.8)

Command value source (DL 8)
- 0: 16-bit command value source (SWQ1)
- 1: 32-bit command value source (SWQ2)

AWL program example for demonstration of an operating mode switch-over with synchronization:

Assignment list

```
;Assignment list
;for example FB 209 (synchronization after BA switch-over);
;Activation by "S $M_Steu0"

;Outputs
A_Fehl          A  0.0          ;Error indication

;Flags
M_Steu0         M  0.2          ;Control flag 0; commanding BA change
M_Steu1         M  0.3          ;Control flag 1; monitoring commanding
                                ;BA change
M_Steu2         M  0.4          ;Control flag 2; Sync. after BA change
```

Program block PB02

```
;PB02
;Synchronization after BA change

;Step 0 ?
    :UN         $M_Steu0
    :SPB =      St1

;BA change AW1 with KMD_VAR1 = 0
    :U          M  240.7        ;Commanding not permitted?
    :S          $A_Fehl         ;Error: Commanding not permitted
    :BEB

    :R          $A_Fehl         ;Reset error indication
    :A          DB 30           ;Activate DB for BA change
    :SPB        FB 201          ;Commanding

    :R          $M_Steu0        ;Control flag for commanding
    :S          $M_Steu1        ;Control flag for commanding check
    :BEA
```

```
;Step 1?
 St1 :UN        $M_Steu1
     :SPB       =  St2

;Commanding check
     :U         M  240.7
     :BEB                       ;Waiting for FB201 call accepted

     :L         KB 15
     :L         MB 240
     :!=FD                      ;Error in the drive status
     :SPB       =  Fehl

      :L        KB 0
      :L        MB 241
      :><FD                     ;Error in drive error identifier
      :SPB      =  Fehl

      :R        $A_Fehl         ;Reset error indication
      :R        $M_Steu1        ;Reset error monitoring
      :S        $M_Steu2        ;Synchronization after BA change
      :BEA

Fehl :S         $A_Fehl         ;Set error indication
      :R        $M_Steu1        ;Reset error monitoring
      :BEA

;Step 2 ?
 St2 :UN        $M_Steu2
     :BEB

;Synchronization after BA change
     :A         DB 30           ;Activate DB for BA change and FB209
     :L         KD 0
     :T         DD 6            ;Synchronous value=0
     :L         KB 0
     :T         DR 8            ;SF number= 0
     :L         KB 0
     :T         DL 8            ;16-bit command value source
     :SPA       FB209           ;Synchronization after BA change

     :R         $M_Steu2        ;Reset Sync. after BA change

     :BE
```

Data block DB30

```
;DB30
;Command DB for BA change according to NBA1 (ID32801 =
;3C0004); generated with APROS

:KD 0          ;ANTRK    = 0   (AW1)
:KB 1          ;KMDCODE  = 1   (Start)
:KB 5          ;KMDFKT   = 1   (BA change)
:KB 1          ;KMDBA    = 1   (NBA1)
:KB 1          ;KMDBIT   = 1   (KMD_VAR1 = active)
:KD 4          ;KMD_VAR1 = 0   (ext. synchronization, SWQ1 + SWQ2)
:KD 0          ;KMD_VAR2 = 0   (2 words spare)
:KD 0          ;KMD_VAR3 = 0   (2 words spare)
:KD 0          ;KMD_VAR4 = 0   (2 words spare)
:KD 0          ;KMD_VAR5 = 0   (2 words spare)
:KD 0          ;KMD_VAR6 = 0   (2 words spare)
```

2.10 FB210 "Table value calculation"

Different "Fast functions" (SF) (cf. documentation: Description of fast function blocks SF) are based on dependencies described in tabular form (such as distance-distance or distance-time relations in the form of position command value tables. These tables are provided in the form of data blocks. The table contents (values) can be generated by manual input through the APROS editor, on-line generation through the AWL program (e.g. using FB211) or on-line generation through the FB210.

As from AZ-PS4 V02.12/1999, FB210 was extended so that, apart from the equidistant interpolation point tables described below (Y tables), the not equidistant interpolation point tables (XY tables) processed by the SF "XFIPW" can be generated (cf. documentation: Description of fast function blocks SF; SF "XFIPW"). For this purpose the information to be transferred to the FB210 is extended in the current DB by the selection of the wanted table type in DL1 (see below: Tab form).

Further a positioning profile corresponding to "Table type 4", which allows the calculation of XY tables for a SF "XFIPW" in a "Distance-distance mode" was created with the "Table type 5". For this purpose the information to be transferred to FB210 was extended by an additional entry parameter in DD14 in the current DB (see Section 2.10.5).

Moreover, a movement profile corresponding to the table types 4 and 5 was created with the table types 6 and 7. The difference consists in that for the table types 4 and 5 a linearly increasing and decreasing acceleration (limited jerk) is achieved (cf. Figure 4), whereas with table types 6 and 7 a constant acceleration (unlimited jerk) is achieved (cf. Figure 5).

Basic structure of an interpolation point table for equidistant interpolation points

Data double word	High word (H byte, L byte)	Low word (H byte, L byte)
0	Table form = 0	Interpolation point number+1
2	Interpolation point 1	
4	Interpolation point 2	
:		
4000	Interpolation point 2000	
4002		

Interpolation point number+1 (DW 0)
– 3..2001: Number of the table interpolation points +1.

Table form (DW 1)
– 0: Equidistant interpolation point table (Y table)

Interpolation point 1 (DD 2)
– 0: Interpolation point value of the function filed in the table at the interpolation point 1 (start of the table in the origin).

Interpolation point n; with n = 2..2000 (DD 4 to DD 4000)
– $-2^{31}..+2^{31}-1$: Interpolation point value of the function filed in the table at the interpolation point n.

Basic structure of an interpolation point table for non-equidistant interpolation points

Data double word	High word (H byte, L byte)	Low word (H byte, L byte)
0	Table form = 1	Interpolation point number+1
2	X value interpolation point 1	
4	Y value interpolation point 1	
:		
3998	X value interpolation point 1000	
4000	Y value interpolation point 1000	
4002		

Interpolation point number+1 (DW 0)
– 3..1001: Number of the table interpolation points +1.

Table form (DW 1)
– 1: Not equidistant interpolation point table (XY table)

X value interpolation point 1 (DD 2)
– 0: Interpolation point X value of the function filed in the table at the interpolation point 0 (start of the table in the origin).

- **Y value interpolation point 1 (DD 4)**
- 0: Interpolation point Y value of the function filed in the table at the interpolation point 0 (start of the table in the origin).

X value interpolation point n; with n = 2..1000 (DD 6 to DD 3998)
- $0..+2^{31}-1$: Interpolation point X value of the function filed in the table at the interpolation point n.

Y value interpolation point n; with n = 2..1000 (DD 8 to DD 4000)
- $-2^{31}..+2^{31}-1$: Interpolation point Y value of the function filed in the table at the interpolation point n.

For simplified on-line specification of table values, table values are calculated for different SF in the form of defined function curves by calling FB 210. The information to be transferred for this purpose in the current DB to the FB210 is listed below:

Data double word	High word H byte	High word L byte	Low word H byte	Low word L byte
0	Tab. form	DB No.	Init. type	Tab. type
2	Reserve = 0		Interpolation point number+1	
4	Output interval (output incr. per table processing)			
6	Reserve = 0		Incr. per master drive period / encoder resolution	
8	Input/output parameter1			
10	Input/output parameter2			
12	Input/output parameter3			
14	Input/output parameter4			
16	Input/output parameter5			
18	Input/output parameter6			
20				

Tab. type (DR 0)
- Meaning: Type of table which should be generated
- Value range:
  1: Coupling table (SF "FIPW / XFIPW")
  2: Working table (SF "FIPW / XFIPW ")
  3: Decoupling table (SF "FIPW / XFIPW ")
  4: Positioning table with limited jerk for distance-time operating modes (SF "FIPW / XFIPW" / SF "FIPZ")
  5: Positioning table with limited jerk for distance-distance operating modes (SF "FIPW / XFIPW")
  6: Positioning table with unlimited jerk for distance-time operating modes (SF "FIPW / XFIPW" / SF "FIPZ")
  7: Positioning table with unlimited jerk for distance-distance operating modes (SF "FIPW / XFIPW")

Init. type (DL 0)

– Meaning: Initialization type; for varying the parameter input or the course of the function.

– Value range: 0,1,2 (only relevant for Tab. type 1, 3, 4, 5, 6 and 7)

Tab. type 1 and 3:	Parameters 1, 4 and 5 have a different meaning according to Init. type (see below)
Tab. type 4, 6:	Parameters 2 and 5 have a different meaning according to Init. type (see below)
Tab. type 5, 7:	Parameters 2 and 5 have a different meaning according to Init. type Parameter 4 is relevant in XY tables (Tab. form = 1; see below)

DB No. (DR 1)

– Meaning: Number of the DB in which the table values calculated by FB210 are filed (the DB must be generated on calling FB210 already with the required word number "2 x number of the table double word element")

– Value range: 1..255

Tab. form (DR 0)

– Meaning: Type of the table which should be generated

– Value range: 0 = Y table: Equidistant point spacing (defined by stating the Y point values)

1 = XY table: Arbitrary point spacing (defined by stating the X and Y point values)

Interpolation point number+1 (DW 2)

– Meaning: Number of the table interpolation points to be calculated + 1 (in this case the first table element is also counted (+1); it serves for stating the table element number and the table form for the SF.)

– Value range: 5..2001

Output interval (DD 4)

– Meaning: Increments which the drive controlled by the table moves through (table end value).

– Value range: $-2^{31}..+2^{31}-1$

Incr. per master drive period / encoder resolution (DW 6)

– Meaning: Tab.Type = 1..3: Increments per master drive period (input increment number at which the table is processed)

Tab.Type = 4..7: Increments per revolution of the drive which is controlled by means of the table (output increment number at which the controlled drive turns one revolution)

– Value range: 50..65535 (is limited in the distance operating modes on the part of the SF commanding to 32767!; cf. docu. description of fast functions, SF FIPW / XFIPW)

Input parameter 1 (DD/DW 8)

– Meaning: Tab. type and Init. type dependent (see below)

Input parameter 2 (DD/DW 10

– Meaning: Tab. type and Init. type dependent (see below)

Input parameter 3 (DD/DW 12)

– Meaning: Tab. type and Init. type dependent (see below)

Input parameter 4 (DD/DW 14)

– Meaning: Tab. type and Init. type dependent (see below)

Output parameter 5 (DD/DW 16)

– Meaning: Tab. type and Init. type dependent (see below)

Output parameter 6 (DD/DW 18)

– Meaning: Tab. type and Init. type dependent (see below)

A description of the function curves (table values) that can be generated depending on the parameters as well as of the parameters depending on table type is provided in the following sections.

2.10.1 Table type 1 (coupling table)

The purpose of the coupling table is to couple a stationary axis into a movement which is controlled e.g. by the SF "FIPW".
The output values for the coupling table are calculated depending upon the input parameters either according to one or two parabolas ($y = ax^2$) ($\Rightarrow$ constant acceleration). The apex of the first parabola always lies in the point in which the coupling process is started (point S according to Figure 1). The coupling process is concluded in the synchronous point (point P according to Figure 1).

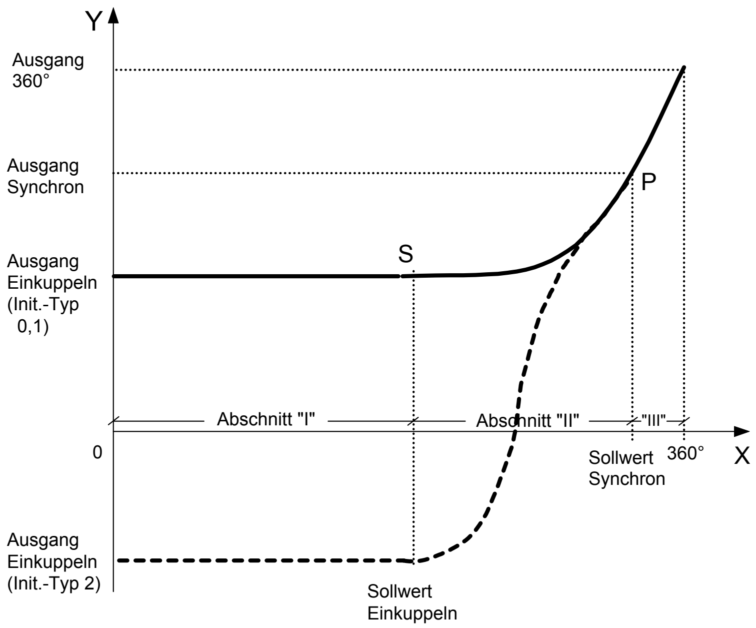


Figure 1: Coupling process

The sequence of the coupling process according to Figure 1 can be divided into three sections:

- In the first section (command value 0 until command value = coupling command value) the output value remains unchanged.
- In the second section (command value = coupling command value up to command value = synchronous command value) the coupling process runs.
- In section 3 the coupling process is ended and the output value is changed linearly to the command value (with "synchronous factor" slope).

Different parameters must be input for calculating the coupling table with the aid of parabolas. In this case a distinction is made between the initialization versions 0..2 (Init. type = 0..2). The initialization versions for calculating the coupling curve differ in the input parameters, the return parameter and the curve calculated from these.

Initialization type 0

In the initialization type 0 the coupling curve is calculated with the aid of one parabola (cf. Figure 1). Input parameters in this case are the output value before the coupling process, the command value at which the coupling process should be concluded and the synchronous factor.

Calculation of the parabola
The factor a of the parabola $y = ax^2$ is calculated from the equation y'= 2ax. The following applies in the synchronous point P:
Synchronous factor = 2a * (synchronous command value - coupling command value) * 100
coupling command value = synchronous command value - 200 * (synchronous output - coupling output) /
synchronous factor
Synchronous output = output interval - (synchronous factor * (output interval -
synchronous command value) / 100

It follows for a:
a = synchronous factor / 200 / (synchronous command value - coupling command value)

The following parameters must be specified:

Data double word	High word H byte	High word L byte	Low word H byte	Low word L byte
0	Reserved = 0	DB No.	Init. type = 0	Tab. type = 1
2	Reserve = 0		Interpolation point number+1	
4	Output interval (output incr. per table processing)			
6	Reserve = 0		Incr. per master drive period	
8	Synchronous command value			
10	Coupling output			
12	Synchronous factor			
14	reserved = 0			
16	Coupling command value			
18	reserved = 0			
20				

Input parameter1: Synchronous command value (DD 8)
– Meaning: Command value angle in which the coupling process should be concluded
– Value range: 0 .. 360
– Unit: degrees

Input parameter2: Coupling output (DD 10)
– Meaning: Angular position of the slave drive before coupling
– Value range: 0 .. +360
– Unit: degrees

Input parameter3: Synchronous factor (DD 12)
– Meaning: Slope of the coupling curve in the linear section after the synchronous point is reached
– Value range: 100 .. 32767
– Unit: Percent, i.e. the slope of 1 (= 45 degrees slope) corresponds to an input value of 100.

Output parameter5: Coupling command value (DD 16)

– Meaning: Largest command value angle in which coupling in the current cycle is still possible
– Value range: 0 .. 360
– Unit: degrees

The following dependencies and marginal conditions apply for the initialization parameters:

1) Synchronous command value ≥ 360 degrees + (coupling output - 360 degrees) / synchronous factor / 100
2) Synchronous command value ≤ 2 * (360 degrees + (coupling output - 360 degrees) / synchronous factor / 100)

Initialization type 1

In the initialization type 1 the coupling curve is calculated with the aid of <u>one</u> parabola (cf. Figure 1). Input parameters in this case are the output value before the coupling process, the command value in which the coupling process should be started and the synchronous factor.

Calculation of the parabola

The factor a of the parabola $y = ax^2$ is calculated from the equation y'= 2ax. The following applies in the synchronous point P:
Synchronous factor = 2a * (synchronous command value - coupling command value) * 100
synchronous command value = coupling command value + 200 * (synchronous output - coupling output) /
synchronous factor

Synchronous output = 2 * coupling output - output interval +
(synchronous factor * (output interval -coupling command value) / 100)

It follows for a:
a = synchronous factor / 200 / (synchronous command value - coupling command value)

The following parameters must be specified:

Data double word	High word H byte	High word L byte	Low word H byte	Low word L byte
0	Reserved = 0	DB No.	Init. type = 1	Tab. type = 0
2	Reserve = 0		Interpolation point number+1	
4	Output interval (output incr. per table processing)			
6	Reserve = 0		Incr. per master drive period	
8	Coupling command value			
10	Coupling output			
12	Synchronous factor			
14	reserved = 0			
16	Synchronous command value			
18	reserved = 0			
20				

Input parameter1: Coupling command value (DD 8)
– Meaning: Command value angle in which the coupling process should be started
– Value range: 0 .. 360
– Unit: degrees

Input parameter2: Coupling output (DD 10)
– Meaning: Angular position of the slave drive before coupling
– Value range: 0 .. +360
– Unit: degrees

Input parameter3: Synchronous factor (DD 12)
– Meaning: Slope of the coupling curve in the linear section after the synchronous point is reached
– Value range: 100 .. 32767
– Unit: Percent, i.e. the slope of 1 (= 45 degrees slope) corresponds to an input value of 100.

Output parameter5: Synchronous command value (DD 16)
– Meaning: Command value angle in which a coupling process is ended
– Value range: 0 .. 360
– Unit: degrees

The following dependencies and marginal conditions apply for the initialization parameters:
1) Coupling command value ≤ 360 degrees + (coupling output - 360 degrees) / synchronous factor / 100
2) Coupling command value ≥ 2 * (180 degrees + (coupling output - 360 degrees) / synchronous factor / 100)

Initialization type 2

In the initialization type 2 the coupling curve is calculated with the aid of <u>two</u> parabolas (cf. Figure 1). An extended definition range for the parameters of the coupling process results from this (cf. dependencies and marginal conditions section). Thus for this initialization type the start valve for the coupling process can be less than 0 degrees. The parabolas are calculated so that during the entire coupling process the acceleration remains constant in amount. Input parameters are the output value before the coupling process, the command value in which the coupling process should be started, the command value in which the coupling process should be concluded and the synchronous factor.

Calculation of parabolas
The factor a of the parabola $y = ax^2$ is calculated from the equation y'= 2ax. The following applies in the synchronous point P:
Synchronous factor = 2a * (apex command value - synchronous command value) * 100

It follows for a:
a = synchronous factor / 200 / (apex command value - synchronous command value)

The following parameters must be specified:

Data double word	High word H byte	High word L byte	Low word H byte	Low word L byte
0	Reserved = 0	DB No.	Init. type = 2	Tab. type = 0
2	Reserve = 0		Interpolation point number+1	
4	Output interval (output incr. per table processing)			
6	Reserve = 0		Incr. per master drive period	
8	Synchronous command value			
10	Coupling output			
12	Synchronous factor			
14	Coupling command value			
16	reserved = 0			
18	reserved = 0			
20				

Input parameter1: Synchronous command value (DD 8)
– Meaning: Command value angle in which the coupling process should be completed
– Value range: 0 .. 360
– Unit: degrees

Input parameter2: Coupling output (DD 10)
– Meaning: Angular position of the slave drive before coupling
– Value range: -360 .. +360
– Unit: degrees

Input parameter3: Synchronous factor (DD 12)
– Meaning: Slope of the coupling curve in the linear section after the synchronous point is reached
– Value range: 100 .. 32767
– Unit: Percent, i.e. the slope of 1 (= 45 degrees slope) corresponds to an input value of 100.

Input parameter4: Coupling command value (DD 14)
– Meaning: Command value angle in which the coupling process should be started
– Value range: 0 .. 360
– Unit: degrees

The following dependencies and marginal conditions apply for the initialization parameters:

1) Synchronous command value ≥ 360 degrees + (coupling output - 360 degrees) / synchronous factor / 100)
2) Synchronous command value ≥ 2 * (360 degrees + (coupling output - 360 degrees) / synchronous factor / 100) - coupling command value
3) Synchronous command value ≥ coupling command value

The following AWL program example serves for illustrating the mode of operation of FB210:

```
;Table value calculation (coupling table)
:A          DB  31      ;Activate DB for calculating coupling table
:SPA        FB  210     ;Table value calculation
```

Data block DB31
;DB31
;DB for table value calculation;
;generated with APROS

:KB 1	;DR00:	Tab. type = 1	(Coupling table)
:KB 0	;DL00:	Init. type = 0	(Synchronous command value input)
:KB 25	;DR01:	DB No. = 25	(DB25 = coupling table)
:KB 0	;DL01:	Reserve = 0	
:KF 52	;DW02:	Interpolation point number+1	(51 interpolation points)
:KF 0	;DW03:	Reserve = 0	
:KD 20000	;DD04:	Output interval	(20000 incr. of the slave per
	;		master revolution)
:KF 20000	;DW06:	Incr. per master drive period	(20000 incr. of the master
	;		per revolution)
:KF 0	;DW07:	Reserve = 0	
:KD 340	;DD08:	Para1: Synchronous command value	(340 degrees)
:KD 180	;DD10:	Para2: Coupling output	(180 degrees)
:KD 150			
	;DD12:	Para3: Synchronous factor	(Slope = 1,5)
:KD 0	;DD14:	Para4: Reserve = 0	
:KD 0	;DD16:	Para5: Output of coupling	
	;	command value	
:KD 0	;DD18:	Para6: Reserve = 0	

2.10.2 Table type 2 (working table)

The position command values for the periodic movement of an axis are filed in the working table. The output values for the working table are calculated with the aid of a sine squared function. Marginal conditions in this case are that the function has at command value 0 an inputtable slope (velocity) and also retains this over a parameterizable range ("Sine start" parameter; cf. Figure 2).

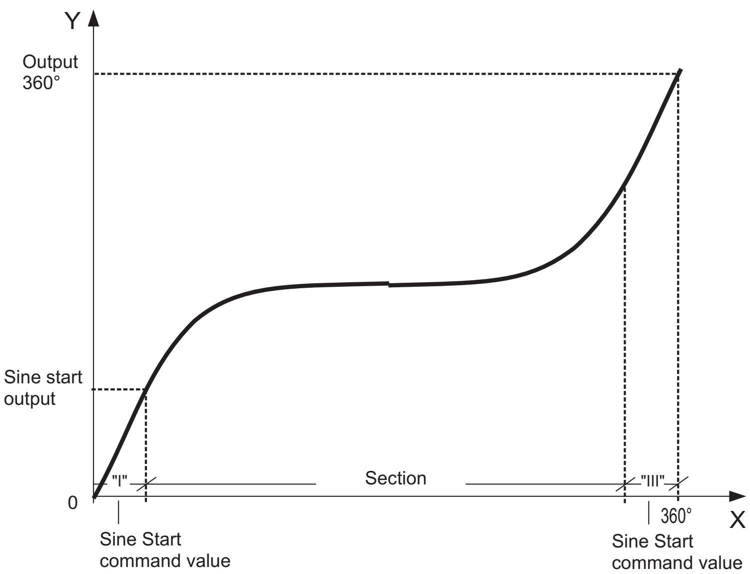


Figure 2: Working function

The function for the working table can be divided into three sections (cf. Figure 2):

- In the first section (command value 0 to command value = sine start command value) the output value is changed linearly to the command value (with slope of synchronous factor).
- The second section (command value = sine start command value to command value = 360 degrees - sine start command value) runs according to the function
  y = synchronous factor * x * (synchronous factor / 100 - 1) * $\sin^2$(arg)
  where: arg = (x - sine start command value) * π / 2 / (360 degrees - 2 * sine start command value)
- In section 3 the output value is changed again linearly to the command value (with slope of synchronous factor).

The following parameters must be input for calculating the working table:

Data double word	High word H byte	High word L byte	Low word H byte	Low word L byte
0	Reserved = 0	DB No.	Init. type = 0	Tab. type = 2
2	Reserve = 0		Interpolation point number+1	
4	Output interval (output incr. per table processing)			
6	Reserve = 0		Incr. per master drive period	
8	Sine start command value			
10	reserved = 0			
12	Synchronous factor			
14	reserved = 0			
16	reserved = 0			
18	reserved = 0			
20				

Input parameter1: Sine start command value (DD 8)

– Meaning: Command value angle in which the sine squared function should be started
– Value range: 0 .. 180
– Unit: degrees

Input parameter3: Synchronous factor (DD 12)

– Meaning: Slope of the working curve in the zero passage and in the linear section before and after the sine squared function
– Value range: 100 .. 32767
– Unit: Percent, i.e. the slope of 1 (= 45 degrees slope) corresponds to an input value of 100.

The following dependencies and marginal conditions apply for the parameters:

1) Sine start command value * synchronous factor / 100 $<$ 360 degrees

The following AWL program example serves to illustrate the mode of operation of FB210:

```
;Table value calculation (working table)
:A          DB  32        ;Activate DB for calculating working table
:SPA        FB  210       ;Table value calculation
```

Data block DB32

```
;DB32
;DB for table value calculation;
;generated with APROS

:KB 2         ;DR00:  Tab. type = 2                 (Working table)
:KB 0         ;DL00:  Init. type= 0                 (not relevant)
:KB 26        ;DR01:  DB No.   = 26                 (DB26 = Working table)
:KB 0         ;DL01:  Reserve  = 0
:KF 52        ;DW02:  Interpolation point number+1  (51 interpolation points)
:KF 0         ;DW03:  Reserve  = 0
:KD 20000     ;DD04:  Output interval               (20000 incr. of the slave per
              ;                                      master revolution)
:KF 20000     ;DW06:  Incr. per master drive period (20000 incr. of the master
                      ;                              per revolution)
:KF 0         ;DW07:  Reserve  = 0
:KD 20        ;DD08:  Para1: Sine start command value          (20 degrees)
:KD 0         ;DD10:  Para2: Reserve  = 0
:KD 150       ;DD12:  Para3: Synchronous factor =   (150%)
:KD 0         ;DD14:  Para4: Reserve  = 0
:KD 0         ;DD16:  Para5: Reserve  = 0
:KD 0         ;DD18:  Para6: Reserve  = 0
```

2.10.3 Table type 3 (decoupling table)

The decoupling table serves for decoupling an axis which is moved in a working table with the aid of the SF "FIPW" from this movement and to stop at a defined angle position. The output values for the decoupling table are calculated according to one or two parabolas ($y = ax^2$). The apex of a parabola lies in the point in which the decoupling process is completed (point S). The decoupling process is started in the decoupling point (point P; cf. Figure 3).

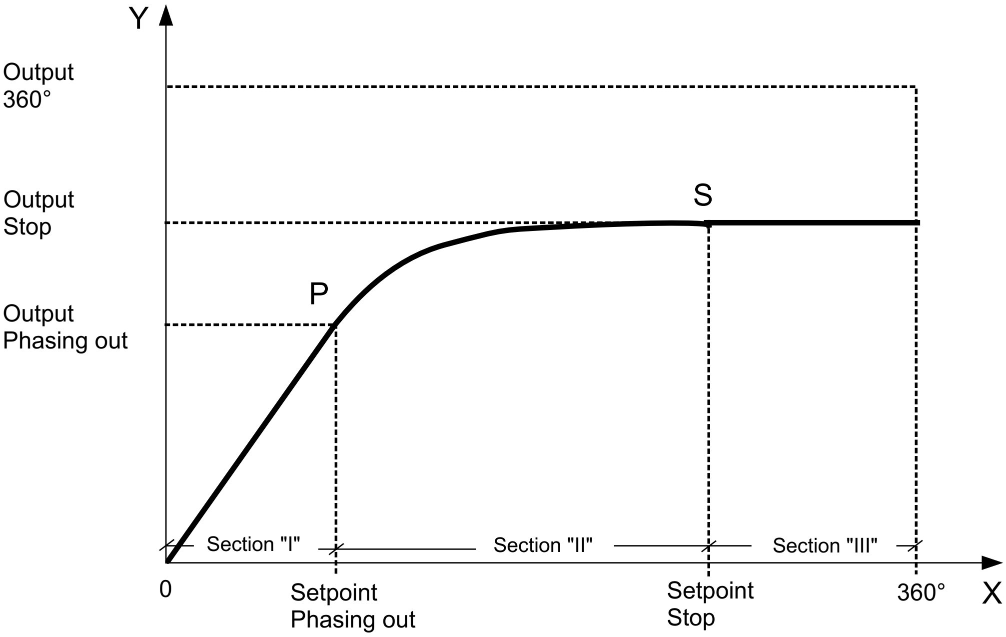


Figure 3: Decoupling process

The decoupling process can be divided into three sections (cf. Figure 3):

- In the first section (command value 0 to command value = decoupling command value) the output value is changed linearly to the command value (slope of synchronous factor).
- In the second section (command value = decoupling command value to command value = brake command value) the decoupling process runs.
- In section 3 the decoupling process is ended, the value is no longer changed.

Different parameters must be input to calculate the decoupling table with the aid of a parabola. A distinction is made between the three initialization versions 0..2 (Init. type = 0..2). The initialization versions for the calculation of the decoupling curve differ in the input parameters, the return parameter and the curve calculated from these.

Initialization type 0

In the initialization type 0 the decoupling curve is calculated with the aid of one parabola (cf. Figure 3). Input parameters in this case are the command value in which the decoupling process should be started and the output value after the decoupling process.

Calculation of the parabola

The factor a of the parabola $y = -ax^2$ is calculated from the equation $y' = -2ax$. The following applies in the decoupling point P, in which the decoupling process is started:

Synchronous factor = -2a * (decoupling command value - brake command value) * 100
brake command value = (200 * brake output / synchronous factor) - decoupling command value

It follows for a:

a = synchronous factor / 200 / (brake command value - decoupling command value)

The following parameters must be specified:

Data double word	High word H byte	High word L byte	Low word H byte	Low word L byte
0	Reserved = 0	DB No.	Init. type = 0	Tab. type = 3
2	Reserve = 0		Interpolation point number+1	
4	Output interval (output incr. per table processing)			
6	Reserve = 0		Incr. per master drive period	
8	Decoupling command value			
10	Brake output			
12	Synchronous factor			
14	reserved = 0			
16	Brake command value			
18	reserved = 0			
20				

Input parameter1: Decoupling command value (DD 8)

– Meaning: Command value angle in which the decoupling process should be started
– Value range: 0 .. 360
– Unit: degrees

Input parameter2: Brake output (DD 10)

– Meaning: Angular position of the slave drive after the decoupling n
– Value range: -360 .. +360
– Unit: degrees

Input parameter3: Synchronous factor (DD 12)

– Meaning: Slope of the decoupling curve in the linear section before the decoupling point is reached
– Value range: 100 .. 32767
– Unit: Percent, i.e. the slope of 1 (= 45 degrees slope) corresponds to an input value of 100.

Output parameter5: Brake command value (DD 16)

– Meaning: Command value angle in which the decoupling process is ended and the output is no longer changed
– Value range: 0 .. 360
– Unit: degrees

The following dependencies and marginal conditions apply for the initialization parameters:

1) Decoupling command value ≤ brake output / synchronous factor / 100
2) Decoupling command value ≥ 2 * brake output / synchronous factor / 100 -360 degrees

Initialization type 1

In the initialization type 1 the decoupling curve is calculated with the aid of one parabola (cf. Figure 3). Input parameters are the command value in which the decoupling process should be ended and the output value after the decoupling process.

Calculation of the parabola

The factor a of the parabola $y = -ax^2$ is calculated from the equation y'= -2ax. The following applies in the decoupling point P in which the decoupling process is started:

Synchronous factor = -2a * (decoupling command value - brake command value) * 100
decoupling command value = 200 * brake output / synchronous factor - brake command value

It follows for a:

a = synchronous factor / 200 / (brake command value - decoupling command value)

The following parameters must be specified:

Data double word	High word H byte	High word L byte	Low word H byte	Low word L byte
0	Reserved = 0	DB No.	Init. type = 1	Tab. type = 3
2	Reserve = 0		Interpolation point number+1	
4	Output interval (output incr. per table processing)			
6	Reserve = 0		Incr. per master drive period	
8	Brake command value			
10	Brake output			
12	Synchronous factor			
14	reserved = 0			
16	Decoupling command value			
18	reserved = 0			
20				

Input parameter1: Brake command value (DD 8)

– Meaning: Command value angle in which the decoupling process should be ended
– Value range: 0 .. 360
– Unit: degrees

Input parameter2: Brake output (DD 10)

– Meaning: Angular position of the slave drive after decoupling
– Value range: -360 .. +360
– Unit: degrees

Input parameter3: Synchronous factor (DD 12:)

– Meaning: Slope of the decoupling curve in the linear section before the synchronous point is reached
– Value range: 100 .. 32767
– Unit: Percent, i.e. the slope of 1 (= 45 degrees slope) corresponds to an input value of 100.

Output parameter5: Decoupling command value (DD 16)

– Meaning: Command value angle in which the decoupling process is started
– Value range: 0 .. 360
– Unit: degrees

The following dependencies and marginal conditions apply for the initialization parameters:

1) Brake command value ≥ brake output / synchronous factor / 100
2) Brake command value ≤ 2 * brake output / synchronous factor / 100

Initialization type 2

In the initialization type 2 the decoupling curve is calculated with the aid of two parabolas (cf. Figure 3). In this way there results an extended definition range for the parameters of the decoupling process (cf. dependencies and marginal conditions section). Thus in this initialization type the end value for the decoupling process can be less than 0 degrees. The apex point of the 1st parabola (apex command value) is calculated so that the acceleration remains constant in amount during the entire decoupling process. Input parameters are the command value in which the decoupling process should be started, the command value in which the decoupling process should be ended and the output value after the decoupling process.

Calculation of the parabolas

The factor a of the parabola $y = -ax^2$ is calculated from the equation $y' = -2ax$. The following applies in the decoupling point P in which the decoupling process is started:

Synchronous factor = -2a * (apex command value - decoupling command value) * 100

It follows for a:

a = synchronous factor / 200 / (apex command value - decoupling command value)

In this case the following table and initialization type dependent parameters must be specified:

Data double word	High word H byte	High word L byte	Low word H byte	Low word L byte
0	Reserved = 0	DB No.	Init. type = 2	Tab. type = 3
2	Reserve = 0		Interpolation point number+1	
4	Output interval (output incr. per table processing)			
6	Reserve = 0		Incr. per master drive period	
8	Decoupling command value			
10	Brake output			
12	Synchronous factor			
14	Brake command value			
16	Overshoot output			
18	reserved = 0			
20				

Input parameter1: Decoupling command value (DD 8)
– Meaning: Command value angle in which the decoupling process should be started
– Value range: 0 .. 360
– Unit: degrees

Input parameter2: Brake output (DD 10)
– Meaning: Angular position of the slave drive after the decoupling n
– Value range: -360 .. +360
– Unit: degrees

Input parameter3: Synchronous factor (DD 12:)
– Meaning: Slope of the decoupling curve in the linear section after the synchronous point
– Value range: 100 .. 32767
– Unit: Percent, i.e. the slope of 1 (= 45 degrees slope) corresponds to an input value of 100.

Input parameter4: Brake command value (DD 14)
– Meaning: Command value angle in which the decoupling process should be ended
– Value range: 0 .. 360
– Unit: degrees

Output parameter5: Overshoot output (DD 16)
– Meaning: Overshoot of the slave drive during return
– Value range: 0 .. 360
– Unit: degrees

The following dependencies and marginal conditions apply for the initialization parameters:
1) Decoupling command value ≥ 2 * brake output / synchronous factor / 100 - brake command value
2) Decoupling command value ≤ brake command value

The following AWL program example serves for illustrating the mode of operation of FB210:

```
;Table value calculation (decoupling table)
:A          DB  33      ;Activate DB for calculating decoupling table
:SPA        FB  210     ;Table value calculation
```

Data block DB33

```
;DB33
;DB for table value calculation
;generated with APROS

:KB 3          ;DR00:  Tab. type = 3                   (Decoupling table)
:KB 0          ;DL00:  Init. type = 0                  (Decoupling command value input)
:KB 27         ;DR01:  DB No.                          (DB27 = decoupling table)
:KB 0          ;DL01:  Reserve = 0
:KF 52         ;DW02:  Interpolation point number+1    (51 interpolation points)
:KF 0          ;DW03:  Reserve = 0
:KD 20000      ;DD04:  Output interval                 (20000 incr. of the slave per
               ;                                        master revolution)
:KF 20000      ;DW06:  Incr. per master drive period   (20000 incr. of the master
               ;                                        per revolution)
:KF 0          ;DW07:  Reserve = 0
:KD 20         ;DD08:  Para1: Decoupling command value           (20 degrees)
:KD 180        ;DD10:  Para2: Brake output             (180 degrees)
:KD 150        ;DD12:  Para3: Synchronous factor       (150%
:KD 0          ;DD14:  Para4: Reserve = 0
:KD 0          ;DD16:  Para5: Output of decoupling command value
:KD 0          ;DD18:  Para6: Reserve = 0
```

2.10.4 Table type 4 (positioning table with limited jerk for the distance-time mode)

The SF "FIPW / XFIPW" (in the "Time" operating mode) or the SF "FIPZ" replace the input distance by a time value for one axis, i.e. the table values are assigned to the values of the periodical input variable of time (a table value is assigned to each point in time between 0 and the cycle time).

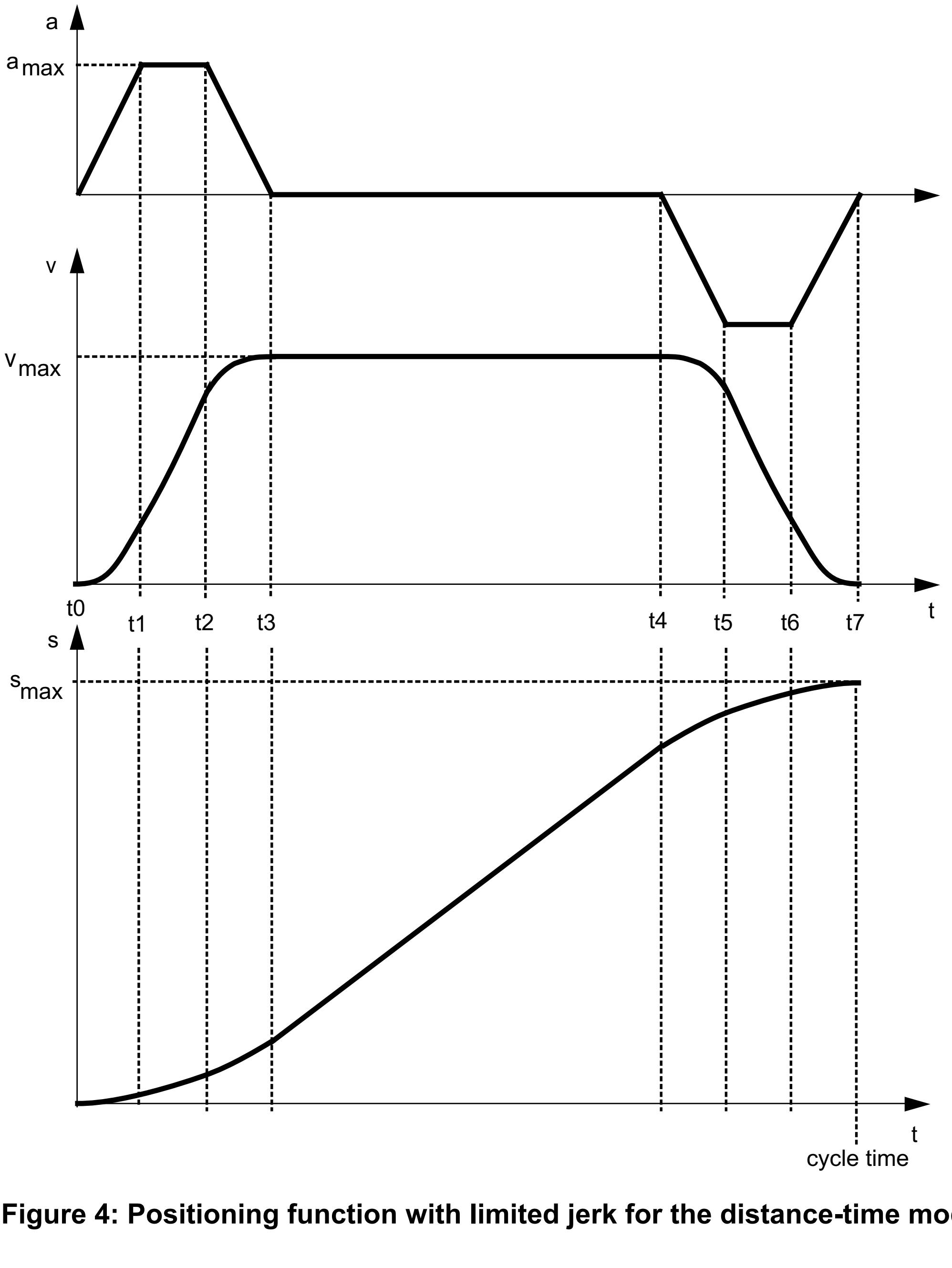


Figure 4: Positioning function with limited jerk for the distance-time mode

FB 210 "Table value calculation" calculates under table type 4, in one function passage, all output values (table values) for a distance-time table. A positioning process with limited jerk, which is run through once per cycle is filed in the distance-time table.

The distance-time function can be divided into 7 sections (cf. Figure 4):

- 1st section (t=0 to $t=t_1$) : Acceleration a increases linearly (jerk constant)
- 2nd section ($t=t_1$ to $t=t_2$): Acceleration a is constant, velocity v increases linearly
- 3rd section ($t=t_2$ to $t=t_3$): Acceleration a decreases linearly (jerk constant)
- 4th section ($t=t_3$ to $t=t_4$): Acceleration is zero, velocity $v=v_{max}$ is constant and the distance “s“ increases linearly
- 5th section ($t=t_4$ to $t=t_5$): Acceleration decreases linearly (jerk constant)
- 6th section ($t=t_5$ to $t=t_6$): Acceleration a is constant, velocity v decreases linearly
- 7th section ($t=t_6$ to $t=t_7$): Acceleration a increases linearly (jerk constant)

Parameters for the distance-time function

The following standardizations must be taken into account:

– Jerk	r:	1	= 1 U/sss
– Acceleration	a:	1000	= 1 U/ss
– Velocity	v:	10000	= 1 rpm
– Distance	s:	1	= 1 increment

A distinction is made between two initialization versions (Init. type = 0, 1) for the calculations of the distance-time table. The two initialization versions for calculating the table differ in the input parameter 2 and in the return parameter 5.

Initialization type 0

Initialization type 0 (Init. type = 0) has as parameter 2 the acceleration time t_3 (time in percent of the cycle time in which the acceleration process should be performed). The following parameters must be input to calculate the distance-time table:

Data double word	High word H byte	High word L byte	Low word H byte	Low word L byte
0	Tab. form	DB No.	Init. type = 0	Tab. type = 4
2	Reserve = 0		Interpolation point number+1	
4	Output interval (output increments pro cycle time)			
6	Reserve = 0		Encoder resolution (Inkr. pro U.)	
8	Cycle time T			
10	Acceleration time TB			
12	max. acceleration a_{max}			
14	reserved = 0			
16	max. velocity v_{max}			
18	max. jerk r_{max}			
20				

Input parameter1: Cycle time (DD 8)
– Meaning: Time for the positioning process
– Value range: 1 .. 65535
– Unit: ms

Input parameter2: Acceleration time (DD 10)
– Meaning: Time for the acceleration process in percent cycle time
– Value range: 1 .. 50
– Unit: Percent

Input parameter3: Max. acceleration (DD 12)
– Meaning: $|a|_{max}$ Absolute value of the max. acceleration for the positioning process
– Value range: 1 .. 65536000
– Unit: 1/1000 U/ss

Output parameter5: Max. velocity (DD 16)
– Meaning: $|v|_{max}$ Absolute value for the max. velocity during the positioning process
– Value range: 1 .. 65536000
– Unit: 1/10000 rpm

Output parameter6: Max. jerk (DD 18)
– Meaning: $|r|_{max}$ Absolute value of the max. jerk during the positioning process
– Value range: 1 .. 65536000
– Unit: U/sss

The following parameters must be input for both initialization types for the calculations of the tables:
- Encoder resolution of the axis to be moved
- Output interval s_{max} (number of increments by which the axis should be moved)
- Cycle time T (time in ms in which the movement process is performed)
- $|a|_{max}$ (Absolute value of the maximum acceleration in the movement process)

To be able to perform the acceleration process in the stated acceleration time, the following marginal conditions apply for the maximum acceleration a_{max}:

- Lower limit: $|a|_{max}$ / 1000 >
  $|v|_{max}$ / 10000 / 60 / (cycle time / 1000 * acceleration time / 100)
- Upper limit : $|a|_{max}$ / 1000 <
  2 * $|v|_{max}$ / 10000 / 60 / (cycle time / 1000 * acceleration time / 100)

whereby for $|v|_{max}$:
$|v|_{max}$ / 10000 / 60 =
|Output interval| / (cycle time / 1000 * (1 - acceleration time / 100)) / encoder resolution

with:
- Output interval in increments
- Encoder resolution in increments / revolution
- Cycle time in ms
- Acceleration time in percent of the cycle time

If the stated value for $|a|_{max}$ is less than the lower limit, then the FB is ended with an error message. If the stated value is greater than the upper limit, $|a|_{max}$ is set equal to the upper limit value. There is no error message.

Initialization type 1

The initialization type 1 (Init. type = 1) has as parameter 2 the maximum velocity $|v|_{max}$. The following parameters must be input for calculating the distance-time table:

Data double word	High word H byte	High word L byte	Low word H byte	Low word L byte
0	Tab. form	DB No.	Init. type = 1	Tab. type = 4
2	Reserve = 0		Interpolation point number+1	
4	Output interval (output increments pro cycle time)			
6	Reserve = 0		Encoder resolution (Inkr. pro U.)	
8	Cycle time T			
10	max. velocity v_{max}			
12	max. acceleration a_{max}			
14	reserved = 0			
16	Acceleration time TB			
18	max. jerk r_{max}			
20				

Input parameter1: Cycle time (DD 8)
– Meaning: Time for the positioning process
– Value range: 1 .. 65535
– Unit: ms

Input parameter2: Max. velocity (DD 10)
– Meaning: $|v|_{max}$ Absolute value of the velocity for the positioning process
– Value range: 1 .. 65536000
– Unit: 1/10000 rpm

Input parameter3: Max. acceleration (DD 12)
– Meaning: $|a|_{max}$ Absolute value of the max. acceleration for the positioning process
– Value range: 1 .. 65536000
– Unit: 1/1000 U/ss

Output parameter5: Acceleration time (DD 16)
– Meaning: Time for the acceleration process in percent cycle time
– Value range: 1 .. 50
– Unit: Percent

Output parameter6: Max. jerk (DD 18)
– Meaning: $|r|_{max}$ Absolute value of the max. jerk during the positioning process
– Value range: 1 .. 65536000
– Unit: U/sss

To be able to end the positioning process in the stated cycle time
the following marginal conditions apply for vmax:
– Lower limit: $|v|_{max}$ / 10000 > |Output interval| * 60 / (cycle time / 1000 *
(1 - acceleration time / 100)) / encoder resolution

whereby: acceleration time = min. acceleration time = 1%

– Upper limit : $|v|_{max}$ / 10000 < |Output interval| * 60 / (cycle time / 1000 * (1 - acceleration time / 100)) / encoder resolution

whereby: acceleration time = max. acceleration time = 50%

with:
- Output interval in increments
- Encoder resolution in increments / revolution
- Cycle time in ms
- Acceleration time in percent of the cycle time

If the stated for $|v|_{max}$ is less than the lower limit, then the FB is ended with an error message. If the stated value is greater than the upper limit, $|v|_{max}$ is set equal to the upper limit value. There is no error message.

For |a|max it applies analogously to the initialization type 0:
- Lower limit: $|a|_{max}$ / 1000 > $|v|_{max}$ / 10000 / 60 / (cycle time / 1000 * acceleration time / 100)
- Upper limit : $|a|_{max}$ / 1000 < 2 * $|v|_{max}$ / 10000 / 60 / (cycle time / 1000 * acceleration time / 100)

with:
- Cycle time in ms
- Acceleration time in percent of the cycle time

Whereby for die acceleration time:
Acceleration time TB = 100 * (1 - (|Output interval| / ($|v|_{max}$ / 10000 / 60 * cycle time / 1000 * Encoder resolution)))

The following AWL program example serves for illustrating the mode of operation of FB210:

;Table value calculation (distance-time table)

```
:L      KB  0
:E      DB  35     ;Delete, if created
:L      KF  256
:E      DB  35     ;Generate DB35 for distance-time table
                   ;with 127 interpolation points (256 words)
:A      DB  34     ;Activate DB for calculating the distance-time table
:SPA    FB  210    ;Table value calculation
```

Data block DB34

;DB34
;DB for table value calculation;
;generated with APROS

:KB 4	;DR00:	Tab. type = 4	(Distance-time table)
:KB 0	;DL00:	Init. type= 0	(Acceleration time input)
:KB 35	;DR01:	DB No. = 35	(DB35 = Distance-time table)
:KB 0	;DL01:	Tab. form = 0	(Y table)
:KF 128	;DW02:	Interpolation point number+1	(127 interpolation points)
:KF 0	;DW03:	Reserve = 0	
:KD 20000	;DD04:	Output interval	(20000 incr. pro cycle time)
:KF 20000	;DW06:	Encoder resolution	(20000 incr. per revolution)
:KF 0	;DW07:	Reserve = 0	
:KD 1000	;DD08:	Para1: Cycle time	(1000 ms)
:KD 10	;DD10:	Para2: Acceleration time	(10% cycle time)
:KD 100000	;DD12:	Para3: max. acceleration	(100 U/ss)
:KD 0	;DD14:	Para4: Reserve = 0	
:KD 0	;DD16:	Para5: Output of |v|max	(10000=1rpm)
:KD 0	;DD18:	Para6: Output of |r|max	(U/sss)

2.10.5 Table type 5 (positioning table with limited jerk for the distance-distance mode)

The SF "FIPW / XFIPW" (in the "Distance-distance" operating mode) assign an output value to the input distance, i.e. the table values are assigned to the input variable values (a table value is assigned to each input value between 0 and the number "Increments per master drive period").

Table type 5 is required only in connection with XY tables (table form = 1). Since in connection with XY tables the "Increments per master drive period" are taken from the last table interpolation point X value, this is required as additional parameter for the table calculation (see below: Increments per master drive period; DD14).

The table curve corresponds to the curve of table type 5 (cf. Section 2.10.4, Figure 4).

Parameters for the distance-distance function

The following standardizations must be taken into account:

– Jerk	r:	1	= 1 U/sss
– Acceleration	a:	1000	= 1 U/ss
– Velocity	v:	10000	= 1 rpm
– Distance	s:	1	= 1 increment

A distinction is made between two initialization versions (Init. type = 0, 1) for the calculations of the distance-distance table. The two initialization versions for the calculation of the table differ in the input parameter 2 and in the return parameter 5.

Initialization type 0

The initialization type 0 (Init. type = 0) has as parameter 2 the acceleration time t_3 (time in percent of the cycle time, in which the acceleration process should be performed). The following parameters must be input for calculating the distance-distance table:

Data double word	High word H byte	High word L byte	Low word H byte	Low word L byte
0	Tab. form = 1	DB No.	Init. type = 0	Tab. type = 5
2	Reserve = 0		Interpolation point number+1	
4	Output interval (output increments pro cycle time)			
6	Reserve = 0		Encoder resolution (Inkr. pro U.)	
8	Cycle time T			
10	Acceleration time TB			
12	max. acceleration a_{max}			
14	Increments per master drive period			
16	max. velocity v_{max}			
18	max. jerk r_{max}			
20				

Input parameter1: Cycle time (DD 8)

– Meaning: Time for the positioning process. (Time in which the "Increments per master drive period" arrive; or in which the positioning table is processed.)
– Value range: 1 .. 65535
– Unit: ms

Input parameter2: Acceleration time (DD 10)

– Meaning: Time for the acceleration process in percent cycle time
– Value range: 1 .. 50
– Unit: Percent

Input parameter3: max. acceleration (DD 12)

– Meaning: $|a|_{max}$ Absolute value of the max. acceleration for the positioning process
– Value range: 1 .. 65536000
– Unit: 1/1000 U/ss

Input parameter4: Increments per master drive period (DD 14)

– Meaning: Number of incoming increments at which the positioning process is completed (X value of the last table interpolation point of the XY table).
– Value range: 1 .. 2^{31}-1 (The max. permissible value is limited to 32767 by the SF FIPW / XFIPW!)
– Unit: Increment

Output parameter5: max. velocity (DD 16)

– Meaning: $|v|_{max}$ Absolute value of the max. velocity during the positioning process
– Value range: 1 .. 65536000
– Unit: 1/10000 rpm

Output parameter6: max. jerk (DD 18)

– Meaning: $|r|_{max}$ Absolute value of the max. jerk during the positioning process
– Value range: 1 .. 65536000
– Unit: U/sss

The following parameters must be input for both initialization types for the calculations of the tables:

– Encoder resolution of the axis to be moved
– Output interval s_{max} (number of increments by which the axis should be moved)
– Cycle time T (time in ms in which the movement process is performed)
– $|a|_{max}$ (Absolute value of the maximum acceleration in the movement process)

To be able to perform the acceleration process in the stated acceleration time, the following marginal conditions apply for the maximum acceleration a_{max}:

– Lower limit: $|a|_{max}$ / 1000 >
$|v|_{max}$ / 10000 / 60 / (cycle time / 1000 * acceleration time / 100)
– Upper limit : $|a|_{max}$ / 1000 <
2 * $|v|_{max}$ / 10000 / 60 / (cycle time / 1000 * acceleration time / 100)

whereby for $|v|_{max}$:
$|v|_{max}$ / 10000 / 60 =
|Output interval| / (cycle time / 1000 * (1 - acceleration time / 100)) / encoder resolution

with:
- Output interval in increments
- Encoder resolution in increments / revolution
- Cycle time in ms
- Acceleration time in percent of the cycle time

If the stated value for $|a|_{max}$ is less than the lower limit, then the FB is ended with an error message. If the stated value is greater than the upper limit, $|a|_{max}$ is set equal to the upper limit value. There is no error message.

Initialization type 1

The initialization type 1 (Init. type = 1) has as parameter 2 the maximum velocity $|v|_{max}$. The following parameters must be input for calculating the distance-time table:

Data double word	High word		Low word	
	H byte	L byte	H byte	L byte
0	Tab. form = 1	DB No.	Init. type = 1	Tab. type = 5
2	Reserve = 0		Interpolation point number+1	
4	Output interval (output increments pro cycle time)			
6	Reserve = 0		Encoder resolution (incr. pro U.)	
8	Cycle time T			
10	max. velocity v_{max}			
12	max. acceleration a_{max}			
14	Increments per master drive period			
16	Acceleration time TB			
18	max. jerk r_{max}			
20				

Input parameter1: Cycle time (DD 8)
- Meaning: Time for the positioning process. (Time in which the "Increments per master drive period" arrive; or in which the positioning table is processed.)
- Value range: 1 .. 65535
- Unit: ms

Input parameter2: max. velocity (DD 10)
- Meaning: $|v|_{max}$ Absolute value of the velocity for the positioning process
- Value range: 1 .. 65536000
- Unit: 1/10000 rpm

Input parameter3: max. acceleration (DD 12)
- Meaning: $|a|_{max}$ Absolute value of the max. acceleration for the positioning process
- Value range: 1 .. 65536000
- Unit: 1/1000 U/ss

Input parameter4: Increments per master drive period (DD 14)

– Meaning: Number of incoming increments at which the positioning process is completed (X value of the last table interpolation point of the XY table).
– Value range: 1 .. 2^{31}-1 (The max. permissible value is limited to 32767 by the SF FIPW / XFIPW!)
– Unit: Increment

Output parameter5: Acceleration time (DD 16)

– Meaning: Time for the acceleration process in percent cycle time
– Value range: 1 .. 50
– Unit: Percent

Output parameter6: max. jerk (DD 18)

– Meaning: $|r|_{max}$ Absolute value of the max. jerk during the positioning process
– Value range: 1 .. 65536000
– Unit: U/sss

To be able to end the positioning process in the stated cycle time
the following marginal conditions apply for vmax:

– Lower limit: $|v|_{max}$ / 10000 > |Output interval| * 60 / (cycle time / 1000 * (1 - acceleration time / 100)) / encoder resolution

whereby: acceleration time = min. acceleration time = 1%

– Upper limit : $|v|_{max}$ / 10000 < |Output interval| * 60 / (cycle time / 1000 * (1 - acceleration time / 100)) / encoder resolution

whereby: acceleration time = max. acceleration time = 50%

with:

– Output interval in increments
– Encoder resolution in increments / revolution
– Cycle time in ms
– Acceleration time in percent of the cycle time

If the stated value for $|v|_{max}$ is less than the lower limit, then the FB is ended with an error message. If the stated value is greater than the upper limit, $|v|_{max}$ is set equal to the upper limit value. There is no error message.

There applies for |a|max analogously to initialization type 0:

– Lower limit: $|a|_{max}$ / 1000 > $|v|_{max}$ / 10000 / 60 / (cycle time / 1000 * acceleration time / 100)
– Upper limit : $|a|_{max}$ / 1000 < 2 * $|v|_{max}$ / 10000 / 60 / (cycle time / 1000 * acceleration time / 100)

with:

– Cycle time in ms
– Acceleration time in percent of the cycle time

Whereby for die acceleration time:
Acceleration time TB = 100 * (1 - (|Output interval| / ($|v|_{max}$ / 10000 / 60 * cycle time / 1000 * encoder resolution)))

2.10.6 Table type 6 and 7 (positioning table with unlimited jerk for the distance-time and distance-distance mode)

The table type 6 permits the calculation of a positioning table with unlimited jerk (constant acceleration; see Figure 4) for the distance-time operating modes of the SF FIPW / XFIPW and FIPZ.

The type 7 allows the calculation of a positioning table with unlimited jerk (constant acceleration; see Figure 4) for the distance-distance operating modes of the SF FIPW / XFIPW.

The function curve according to Figure 5 results as limiting case of the curve according to Figure 4, if the phase of the linear increase or decrease of the acceleration tends towards 0 (t1 → t0, t2 → t3, t5 → t4, t6 → t7) in such a way that the area remains constant.

Thus the parameters or statements for table type 4 (cf. Section 2.10.4) apply for table type 6. Analogously the parameters or statements for table type 5 (cf. Section 2.10.5) apply for table type 7.

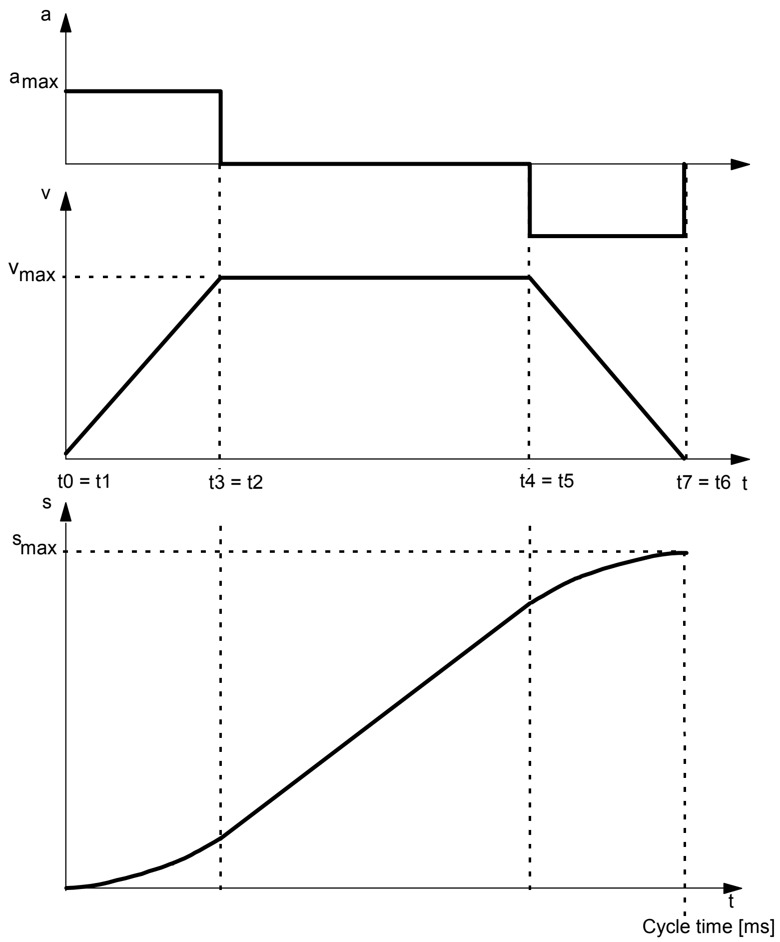


Figure 5: Positioning function with unlimited jerk for the distance-time / distance-distance mode

The positioning profile according to Figure 4 offers advantages in the calculation of periodical movements with constant direction reversal (cf. Figure 6).

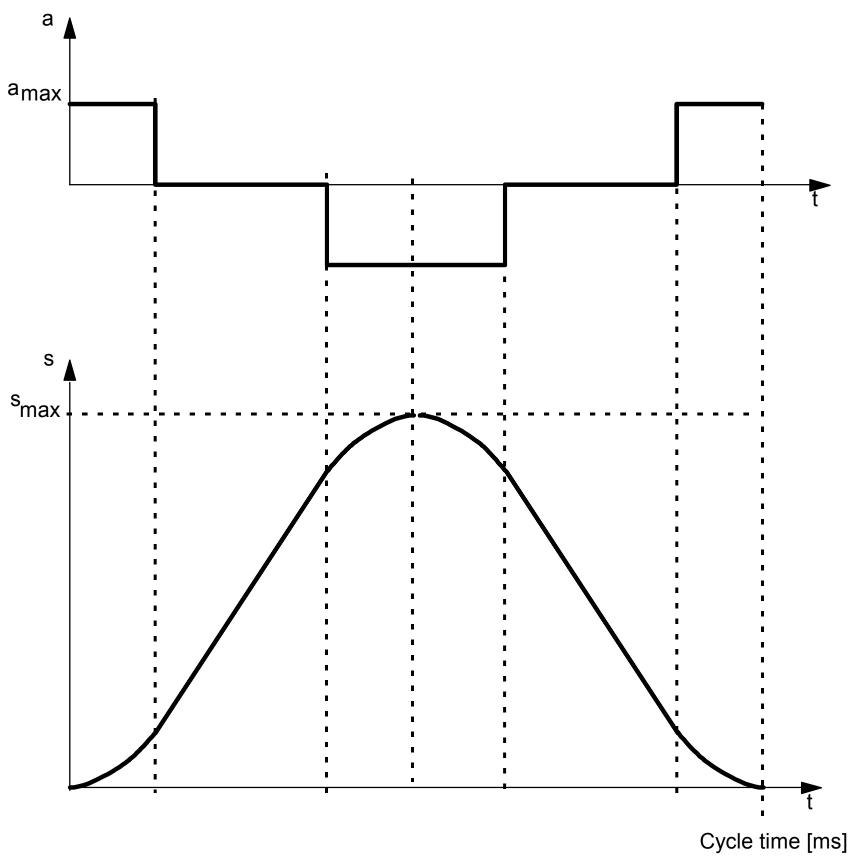


Figure 6: Periodical pendulum movement

2.10.7 Error codes

A separate error module number (111) has been allocated for the error codes of the FB210 (cf. documentation: PS error description for version ≥ AZ-PSx V02.05).

2.11 FB211 "Floating point arithmetic"

FB 211 "Floating point arithmetic" allows processing of a mathematical expression by calling FB211 and describing the mathematical expression (operators, operands) in the current DB.

In the current DB

- the operands and
- the operators

of the mathematical expression are filed. The order of the operands and operators in the current DB corresponds to the order of their occurrence in the mathematical expression – commencing from left to right.

The operands must be input as signed 32-bit fixed point numbers.

The internal calculation takes place in a 64-bit floating point format. The logic operation of the operands through the operators is based on a "Polish Notation"; i.e. the operands are operated on corresponding to the relevant operator, commencing from left to right. The logic operation of the following operand takes place in each case related to the result of the previous logic operation, i.e. there is no weighting of the operators (such as e.g. "Dot before line").

Due to the operators "(" and ")" (cf. Section 2.11.2) it is possible to perform a logic operation of operands corresponding to the customary bracket rules deviating from the "Polish Notation".

It is possible using the operators "=SL", "=S" and "S" (cf. Section 2.11.2) to write a floating point number into the memory field, or to load a temporarily stored floating point number as operand out from the memory field.

The result of the operation(s) is transferred as rounded 32-bit fixed point number through the DB interface (result field) by the operator "=" and the calculation is ended.

An error acknowledgement is also made through the DB interface. Further the PS status (Anz0, Anz1, Ov; cf. documentation: PS command set, Section 3.2.4) is formed depending upon the result.

2.11.1 Description of the DB interface

DD <x>	**DL <x+1>**	**DR <x+1>**	**DL <x>**	**DR <x>**
DD 0	Reserve = 0	Reserve = 0	Error acknowledgement (FehlR)	Operand number (OpAnz)
DD 2	Reserve = 0			
DD 4	Result field			
DD 6	Reserve = 0			
DD 8	Memory field (less significant)			
DD 10	Memory field (more significant)			
DD 12	Operand 0			
DD 14	Operand 1			
:	:			
DD 2*(n+6)	Operand n			
DD 2*(n+7)	Operator 3	Operator 2	Operator 1	Operator 0
:	:	:	:	:
DD 2*(n+6+m/4)	Reserve = 0	Reserve = 0	Operator m+1	Operator m

with:
- 22 + **OpAnz** * 2 + [number of the operators / 2] $\leq$ DB length in data words
- **FehlR** = 0: No error
  = 1: Exceeding the maximum real value (value $\rightarrow \infty$)
  = 2: Not reaching the minimum real value (value $\rightarrow 0$)
  = 3: Inadmissible operand (e.g. $\sqrt{}$ -1)
  = 4: Division by 0

2.11.2 Operators

The following operators are defined:

Operand group	Operator mnemonic	Meaning	Decimal code
0-operands	=	Assignment to the result field (end of the calculation)	0
	=SL	Store deleting: Assign result in real format to the memory field and ignore previous calculation process	1
	=S	Store: Assign result in real format to the memory field	2
	S	Store operand: Operate on real operand from the memory field	3
	Up	Round up: Round current real value up to the next larger real value which can be represented as fixed point number	4
	Down	Round down: Round current real value down to the next smaller real value which can be represented as fixed point number	5
	(	Opening bracket	6
	)	Closing bracket	7
	.	.	.
1-operand	sin	Sine function: Operand in radian measure	64
	cos	Cosine function: Operand in radian measure	65
	√	Root function	66
	| |	Absolute value formation	67
	tan	Tangent function: Operand in radian measure	68 1)
	sinh	Hyperbolic sine function	69 1)
	cosh	Hyperbolic cosine function	70 1)
	tanh	Hyperbolic tangent function	71 1)
	asin	Arc sine function: Operand in radian measure	72 1)
	acos	Arc cosine function: Operand in radian measure	73 1)
	atan	Arc tangent function: Operand in radian measure	74 1)
	exp	Natural exponential function	75 1)
	log	Natural logarithmic function	76 1)
	log10	Decadic logarithmic function	77 1)
	.	.	.
2-operands	+	Addition	128
	-	Subtraction	129
	*	Multiplication	130
	/	Division	131
	atan2	Arc tangent function y/x: Operand in radian measure	132 1)
	pow	Exponential function y^x	133 1)
	fmod	Modulo division y%x	134 1)
	.	.	.

Operand group	Operator mnemonic	Meaning	Decimal code
Constants	1		192
	$\pi/2$		193
	π		194
	$3\pi/2$		195
	2π		196
	.	.	.
	.	:	255
[1] Only in AZ-PS4 module, from version AZ-PS4V02.09			

2.11.3 Examples

Different calculation examples are shown below. The procedure in this case is:

- Statement of the mathematical expression to be processed as example.
- Description of the input order of the operands and operators in the interface DB resulting from this.
- Order of the operands and operators to be entered.
- Example of programming in AWL.

2.11.3.1 Example1

– Mathematical expression 1:
$y_1 = ((((a + b) - c) * d) / e)$

– Entry syntax taking account of the "Polish Notation":
a + b - c * d / e =

– Order of the operands to be entered:
Operand0 = a
Operand1 = b
Operand2 = c
Operand3 = d
Operand4 = e

– Order of the operators to be entered:
Operator0 = +
Operator1 = -
Operator2 = *
Operator3 = /
Operator4 = =

– AWL example for expression 1 (y_1)
with:
a = 25; b = 22; c = 11; d = -5; e = 7⇒ $y_1 = -25{,}714$

```
;DB 16 preallocation by APROS for calculating y1
:KD     5       ;OpAnz = 5
:KD     0       ;Reserve
:KD     0       ;Result
:KD     0       ;Reserve
:KD     0       ;Memory less significant (0; default)
:KD     0       ;Memory more significant (0; default)
:KD     25      ;Operand0 (a = 25)
:KD     22      ;Operand1 (b = 22)
:KD     11      ;Operand2 (c = 11)
:KD     -5      ;Operand3 (d = -5)
:KD     7       ;Operand4 (e = 7)
:KB     128     ;Operator0 (+)
:KB     129     ;Operator1 (-)
:KB     130     ;Operator2 (*)
:KB     131     ;Operator3 (/)
:KB     0       ;Operator4 (=)
:KB     0       ;Reserve
:KB     0       ;Reserve
:KB     0       ;Reserve


;Calculation of y1

:A      DB 16   ;Activate DB 16
:SPA    FB 211  ;Floating point arithmetic
:L      DD 4    ;Transfer result to ACCU1 (y1 = -26)
```

2.11.3.2 Example2

– Mathematical expression 2:
$y_2 = a_3 * x^3 + a_2 * x^2 + a_1 * x + a_0$
$= ((((((a_3 * x) + a_2) * x) + a_1) * x) + a_0)$

– Entry syntax taking account of the "Polish Notation":
$a_3 * x + a_2 * x + a_1 * x + a_0 =$

– Order of the operands to be entered:
Operand0 = a_3
Operand1 = x
Operand2 = a_2
Operand3 = x
Operand4 = a_1
Operand5 = x
Operand6 = a_0

– Order of the operators to be entered:
Operator0 = *
Operator1 = +
Operator2 = *
Operator3 = +
Operator4 = *
Operator5 = +
Operator6 = =

– AWL example for expression 2 (y_2)
with: $a_3 = 4$; $a_2 = -3$; $a_1 = 7$; $a_0 = 12$;
x = 5 ⇒ $y_2 = 472$

```
;DB 16 preallocation by APROS for calculating y2
:KD     7      ;OpAnz = 7
:KD     0      ;Reserve
:KD     0      ;Result
:KD     0      ;Reserve
:KD     0      ;Memory less significant (0; default)
:KD     0      ;Memory more significant (0; default)
:KD     4      ;Operand0 (a3 = 4)
:KD     0      ;Operand1 (x = 0; default)
:KD     -3     ;Operand2 (a2 = -3)
:KD     0      ;Operand3 (x = 0; default)
:KD     7      ;Operand4 (a1 = 7)
:KD     0      ;Operand5 (x = 0; default)
:KD     12     ;Operand6 (a0 = 12)
:KB     130    ;Operator0 (*)
:KB     128    ;Operator1 (+)
:KB     130    ;Operator2 (*)
:KB     128    ;Operator3 (+)
:KB     130    ;Operator4 (*)
:KB     128    ;Operator5 (+)
:KB     0      ;Operator6 (=)
:KB     0      ;Reserve
```

;Calculation of y_2

```
:A      DB 16    ;Activate DB 16
:L      KD 5     ;x = 5
:T      DD 14    ;Operand1
:T      DD 18    ;Operand3
:T      DD 20    ;Operand5
:SPA    FB 211   ;Floating point arithmetic
:L      DD 4     ;Transfer result to ACCU1 (y2 = 472)
```

2.11.3.3 Example3

– Mathematical expression3:
y_3 = (a + b) * (c + d)

– Entry syntax taking account of the "Polish Notation":
a + b =SL c + d * S =

– Order of the operands to be entered:
Operand0 = a
Operand1 = b
Operand2 = c
Operand3 = d
– Order of the operators to be entered:
Operator0 = +
Operator1 = =SL
Operator2 = +
Operator3 = *
Operator4 = S
Operator5 = =

– AWL example for expression 3 (y_3)
with:
a = 25; b = 22; c = 11; d = -5 ⇒ y_3 = 282

```
;DB 16 preallocation by APROS for calculating y3
:KD     4       ;OpAnz = 4
:KD     0       ;Reserve
:KD     0       ;Result
:KD     0       ;Reserve
:KD     0       ;Memory less significant (0; default)
:KD     0       ;Memory more significant (0; default)
:KD     25      ;Operand0 (a = 25)
:KD     22      ;Operand1 (b = 22)
:KD     11      ;Operand2 (c = 11)
:KD     -5      ;Operand3 (d = -5)
:KB     128     ;Operator0 (+)
:KB     1       ;Operator1 (=SL)
:KB     128     ;Operator2 (+)
:KB     130     ;Operator3 (*)
:KB     3       ;Operator4 (S)
:KB     0       ;Operator5 (=)
:KB     0       ;Reserve
:KB     0       ;Reserve


;Calculation of y3

:A      DB 16   ;Activate DB 16
:SPA    FB 211  ;Floating point arithmetic
:L      DD 4    ;Transfer result to ACCU1 (y3 = 282)
```

2.11.3.4 Example4

– Mathematical expression4:
$y_4 = (\sin((a + b) / \pi)) * c$

– Entry syntax taking account of the "Polish Notation":
a + b / π sin * c =

– Order of the operands to be entered:
  Operand0 = a
  Operand1 = b
  Operand2 = c

– Order of the operators to be entered:
  Operator0 = +
  Operator1 = /
  Operator2 = π
  Operator3 = sin
  Operator4 = *
  Operator5 = =

– AWL example for expression 4 (y_4)
with:
a = 2; b = 7; c = 1000
⇒ y_4 = 273,282

```
;DB 16 preallocation by APROS for calculating y4
:KD     3       ;OpAnz = 3
:KD     0       ;Reserve
:KD     0       ;Result
:KD     0       ;Reserve
:KD     0       ;Memory less significant (0; default)
:KD     0       ;Memory more significant (0; default)
:KD     2       ;Operand0 (a = 2)
:KD     7       ;Operand1 (b = 7)
:KD     1000    ;Operand2 (c = 1000)
:KB     128     ;Operator0 (+)
:KB     131     ;Operator1 (/)
:KB     194     ;Operator2 (π)
:KB     64      ;Operator3 (sin)
:KB     130     ;Operator4 (*)
:KB     0       ;Operator5 (=)
:KB     0       ;Reserve
:KB     0       ;Reserve


;Calculation of y4

:A      DB 16   ;Activate DB 16
:SPA    FB 211  ;Floating point arithmetic
:L      DD 4    ;Transfer result to ACCU1 (y4 = 273)
```

2.11.3.5 Example5

– Mathematical expression5:
$y_5 = \text{atan}(1 / \tan(\pi / a)) * b$

– Entry syntax taking account of the "Polish Notation":
1 / (π / a tan) atan * b =

– Order of the operands to be entered:
Operand0 = a
Operand1 = b

– Order of the operators to be entered:
Operator0 = 1
Operator1 = /
Operator2 = (
Operator3 = π
Operator4 = /
Operator5 = tan
Operator6 =)
Operator7 = atan
Operator8 = *
Operator9 = =

– AWL example for expression 5 (y_5)
with:
a = 4; b = 20000000;
$\Rightarrow$ y_5 = 15707963

```
;DB 16 preallocation by APROS for calculating y5
:KD         2           ;OpAnz = 2
:KD         0           ;Reserve
:KD         0           ;Result
:KD         0           ;Reserve
:KD         0           ;Memory less significant (0; default)
:KD         0           ;Memory more significant (0; default)
:KD         4           ;Operand0 (a = 4)
:KD         20000000    ;Operand1 (b = 20000000)
:KB         192         ;Operator0 (1)
:KB         131         ;Operator1 (/)
:KB         6           ;Operator2 (()
:KB         194         ;Operator3 (π)
:KB         131         ;Operator4 (/)
:KB         68          ;Operator5 (tan)
:KB         7           ;Operator6 ())
:KB         74          ;Operator7 (atan)
:KB         130         ;Operator8 (*)
:KB         0           ;Operator9 (=)
:KB         0           ;Reserve
:KB         0           ;Reserve

;Calculation of y5
:A          DB 16       ;Activate DB 16
:SPA        FB 211      ;Floating point arithmetic
:L          DD 4        ;Transfer result to ACCU1 (y5= 15707963)
```

2.11.3.6 Example6

– Mathematical expression6:
$y_6 = \lg e^1 * \ln a * 10^9$

– Entry syntax taking account of the "Polish Notation":
1 exp log10 * (a log) * (b pow c) =

– Order of the operands to be entered:
Operand0 = a
Operand1 = b
Operand2 = c

– Order of the operators to be entered:
Operator0 = 1
Operator1 = exp
Operator2 = log10
Operator3 = *
Operator4 = (
Operator5 = log
Operator6 =)
Operator7 = *
Operator8 = (
Operator9 = pow
Operator10 =)
Operator11 = =

– AWL example for expression 6 (y_6) with:
a = 100; b = 10; c = 9;
⇒ y_6 = 2000000000

```
;DB 16 preallocation by APROS for calculating y6
:KD     3       ;OpAnz = 3
:KD     0       ;Reserve
:KD     0       ;Result
:KD     0       ;Reserve
:KD     0       ;Memory less significant (0; default)
:KD     0       ;Memory more significant (0; default)
:KD     100     ;Operand0 (a = 100)
:KD     10      ;Operand1 (b = 10)
:KD     9       ;Operand2 (b = 9)
:KB     192     ;Operator0 (1)
:KB     75      ;Operator1 (exp)
:KB     77      ;Operator2 (log10)
:KB     130     ;Operator3 (*)
:KB     6       ;Operator4 (()
:KB     76      ;Operator5 (log)
:KB     7       ;Operator6 ())
:KB     130     ;Operator7 (*)
:KB     6       ;Operator8 (()
:KB     133     ;Operator9 (pow)
:KB     7       ;Operator10 ())
:KB     0       ;Operator11 (=)

;Calculation of y6
:A      DB 16   ;Activate DB 16
:SPA    FB 211  ;Floating point arithmetic
:L      DD 4    ;Transfer result to ACCU1 (y6= 2000000000)
```

2.12 FB212 "Diagnosis"

This function block facilitates reading or writing the AMKASYN error stack by means of AWL program. Apart from FB212, the current data block as well as the flag bits "M 237.6" and "M 237.1" are of importance for this.

Organization form of the flag byte "MB 237":

Bit	Meaning
.0	Error status FB202 or FB203 0: No error 1: Error
.1	Error status FB212 0: No error 1: Error
.2	0: Currently not used
.3	0: Currently not used
.4	0: Currently not used
.5	0: Currently not used
.6	Writing/reading status 0: Starting status (FB can be called) 1: Execution of FB212 not yet completed
.7	Writing/reading status 0: Starting status (FB can be called) 1: Execution of FB202 or FB203 not yet completed

The organization is analogous to "MB 237" within the scope of FB202 or FB203. Therefore refer to the description of AMK-specific functional blocks, Section 2.3.2 or Section 2.4 for further information on this.

The different FB212 functions are differentiated by the mode code in the data word DL 0 of the current DB:
- Mode = 0: Function = read error stack
- Mode = 1: Function = write error stack
- Mode = 2: Function = read diagnostic information

2.12.1 Read error stack (Mode = 0)

Select (activate) a data block with a length of at least 82 data words and call FB 212 (prerequisite: M 237.6 = 0) to read the error stack. As soon as "M 237.6 = 0" and "M 237.1 = 0" anew, the DB contains all errors which are displayed under diagnosis in the operator panel of the AZ computer.

The structure of the current DB for FB212 (Mode = 0) looks as follows:

Data double word	High word High byte	High word Low byte	Low word High byte	Low word Low byte	
00	reserved = 0		Mode = 0	Error number	Header
02	reserved = 0		Error module		1st error message
04	Error number				
06	Additional information				
08	reserved = 0		Error class		
10	reserved = 0		Error module		2nd error message
12	Error number				
14	Additional information				
16	reserved = 0		Error class		
..					
74	reserved = 0		Error module		10th error message
76	Error number				
78	Additional information				
80	reserved = 0		Error class		

Total number: 82 data words

Error number (DR 0)
– Meaning: Number of the error messages entered in the error stack (= number of the valid error messages in the current DB).
– Value range: 0 no error message
1..10 1..10 error messages

Mode (DL 0)
– Meaning: Function selection for FB212.
– Value range: 0 read error stack
(1 write error stack)
(2 read diagnostic information)

Error module (DW 2, DW 10, ..)
– Meaning: Number of the error-reporting module.
– Value range: 0..255

Error number (DD 4, DD 12, ..)
– Meaning: Error number within the error-reporting module.
– Value range: 0..255

Additional information (DD 6, DD 14, ..)
– Meaning: Additional information for determining the cause of error.
– Value range: 0..65535

Error class (DW 8, DW 16, ..)
– Meaning: Bit-coded information for classifying the error.
– Value range: 0..65535

2.12.2 Write error stack (Mode = 1)

Enter the required data in the current DB which is structured as below before writing the error stack (call FB212; Mode = 1; prerequisite: M 237.6 = 0):

Data double word	High word: High byte	High word: Low byte	Low word: High byte	Low word: Low byte	
00	reserved = 0		Mode	Error number	Header
02	Additional information		reserved = 0	Error number	1st error message
04	reserved = 0		reserved = 0	Error class	
06	Additional information		reserved = 0	Error number	2nd error message
08	reserved = 0		reserved = 0	Error class	
..					
38	Additional information		reserved = 0	Error number	10th error message
40	reserved = 0		reserved = 0	Error class	

Total number: 2 + 4 * error number (data words)

Error number (DR 0)
– Meaning: Number of the error messages entered in the error stack (= number of the valid error messages in the current DB).
– Value range: 0 no error message
1..10 1..10 error messages

Mode (DL 0)
– Meaning: Function selection for FB212.
– Value range: (0 read error stack)
1 write error stack
(2 read diagnostic information)

Error number (DR 2, DR 6, ..)
– Meaning: Error number within the error-reporting module.
– Value range: 0..255

Additional information (DW 3, DW 7, ..)
– Meaning: Additional information for determining the cause of error.
– Value range: 0..65535

Error class (DR 4, DR 8, ..)

– Meaning: Bit-coded information for classifying the error.
– Value range: 0..255
– Remarks: Errors of Class 1 and 2 lead to a display in the AZ operator panel
Errors of Class 4 (warnings) lead to no display in the AZ operator panel

The minimum length of the current DB to be created (in data words) results from 2 data words for the header, plus 4 data words per error message.

The number of errors to be written in the error stack must be entered in the data word DR 0 (error number). This number (max. 10) further states how many error messages (per 2 data double words) have to be filled out in the current DB starting from data double word 2.

113 is inserted automatically as error module when writing the error stack (FB212; Mode = 1).

Caution: The order in which the errors are written in the stack starts with the error message entered in the highest two data double words (corresponding to the entered error number) and ends with the data double words 2 and 4.

The error number in DR 0 is decremented per completed entry in the error stack.

Currently a maximum of 2 error messages can be entered in the error stack per FB212. (Further error messages can be entered only after "Delete errors" of the AMKASYN system.)

2.12.3 Read diagnostic information (Mode = 2)

Create a data block with a length of at least 22 data words and select it before calling FB212 (Mode = 2) for reading the diagnostic information (as it can be read e.g. by means of the binary E/A of the AMKASYN system) (prerequisite: M 237.6 = 0). As soon as "M 237.6 = 0" and "M 237.1 = 0" anew, the DB contains all diagnostic information which is displayed in the operator panel of the AZ computer.

The structure of the current DB to be selected looks as follows:

Data double word	**High word High byte**	**Low byte**	**Low word High byte**	**Low byte**	
00	reserved = 0		Mode	Error number	Header
02	Diagnosis number		Error source		1st error message
04	Diagnosis number		Error source		2nd error message
..					
18	Diagnosis number		Error source		
20	Diagnosis number		Error source		10th error message

Total number: 22 data words

Error number (DR 0)
– Meaning: Number of the error messages to be entered in the error stack (= number of the valid error messages in the current DB).
– Value range: 0 no error message
1..10 1..10 error messages

Mode (DL 0)
– Meaning: Function selection for FB212.
– Value range: (0 read error stack)
(1 write error stack)
2 read diagnostic information

Error source (DW 2, DW 4, ..)
– Meaning: Error source .
– Value range: 0: Fault message AZ
1..8: Fault message AW1 .. AW8
9: Fault message option card

Diagnosis number (DW 3, DW 5, ..)
– Meaning: Diagnosis number according to the AZ operating display (or the documentation: AMKASYN; digital pulse converter in modular construction; diagnostic messages)
– Value range: 0 .. 4095

2.12.4 Read error code list (Mode = 3)

Create a data block with a minimum length of 2 data words and activate it before calling FB212 (Mode = 3) (prerequisite: M237.6 = 0) for reading the error code list (as it is displayed by means of AZ operator panel under Sercos parameter ID 32962).
After "M 237.6 = 0" and "M 237.1 = 0" anew, the DB contains the corresponding data of the error code list. The current number of the error messages is entered as decimal number in DR 0. The read error data are filed as ASCII code starting from DW 2 – according to created DB length and existing number of error messages.

A maximum of 162 data words are required for reading the complete list of all error codes (10 error messages). If the number of the read error data exceeds the range of the created DB, these are limited to the DB length.

After the last text character of an error message the remaining bytes of the error message (see below: structure of the current DB) are filled up with the code NUL (decimal 0). If the number of read error messages is less than the created DB, the remaining data bytes (up to max. DW 161) are also preassigned with the code NUL.

Structure of current DB for FB 212 (Mode = 3):

Data	High word		Low word		
double word	High byte	Low byte	High byte	Low byte	
00	reserved = 0		Mode = 3	Error number	Header
02	Error number				1st error
04	Error text		Error source		message
06	Further error text				
08	Further error text				
10	Further error text				
12	Further error text				
14	Further error text				
16	Further error text				
18	Error number				2nd error
20	Error text		Error source		message
22	Further error text				
24	Further error text				
26	Further error text				
28	Further error text				
30	Further error text				
32	Further error text				
..					
..					
146	Error number				10th error
148	Error text		Error source		message
150	Further error text				
152	Further error text				
154	Further error text				
156	Further error text				
158	Further error text				
160	Further error text				

Minimum number: 2 data words
Number of max. list: 162 data words

Error number (DR 0)

– Meaning: Number of the error messages entered in the error stack (= number of the valid error messages in the current DB).
– Value range:

0	no error message
1..10	1..10 error messages

Mode (DL 0)
– Meaning: Function selection for FB212.
– Value range:

(0	read error stack)
(1	write error stack)
(2	read diagnostic information)
3	read error code list

Error number (DD 2, DD 18, ..)
– Meaning: Diagnosis number according to the AZ operation display (or the documentation: AMKASYN; digital pulse converter in modular construction; diagnostic messages)
– Value range: 00000000hex .. 34303935hex ASCII (corresponds to.: 0000dec .. 4095dec)

Error source (DW 4, DW 20, ..)
– Meaning: Error source .
– Value range:

3030hex:	Fault message AZ corresponds to: 00dec.)
3031hex..3038hex:	Fault message AW1 .. AW8 (corresponds to: 01dec .. 08dec)
3039hex:	Fault message option card corresponds to: 09dec)

Error text (DW 5 to DW 17, DW 21 to DW 33, ..)
– Meaning: Diagnostic text according to the AZ operation display (or the documentation: AMKASYN; digital pulse converter in modular construction; diagnostic messages)
– Value range: 00hex .. 7Fhex ASCII (per byte)

2.13 FB220 "Initializing the ADPS interface"

The ADPS interface (cf. documentation: AMK digital parallel interface "ADPS") is initialized by calling the AMK-specific function block 220. The initialization data must be transferred to this FB in the current DB. The DB for ADPS initialization must be 6 data words in length.

The data words are defined as follows:

Data double word	High word H byte	High word L byte	Low word H byte	Low word L byte
0	Strobe-Low-T.	Mode-Select	Max. number rep.	Reserv. = 0
2	A-DB address	A-SB address	E-DB address	E-SB address
4	Reserve = 0		Timeout for strobe and error	

----
6 data words

Max. number rep. (DL 0)

- 0 : No repetitions on transmission errors

1..255: Maximum number of repetitions on 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 on drive commanding
- Bit1 = 1: With ready evaluation on drive commanding [2)]
-

Strobe-Low-Time (DL 1)

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

E SB-Adr. (DR 2)

- 0..31: Address of the E byte for the ADPS control word

E DB-Adr. (DL 2)

- 0..31: Address of the E byte for the ADPS data word
- **A SB-Adr. (DR 3)**
- 0..31: Address of the A byte for the ADPS control word

A DB-Adr. (DL 3)

- 0..31: Address of the A byte for the ADPS data word

Timeout for strobe and error (DW 4)

- 0 : No timeout monitoring for strobe and error signal
- 1..65535 (ms): Time (resolution 1 ms) in which the higher level control must have reacted to changes of the strobe or error signal. A timeout error is then generated.

[1] The write/read cycle according to Fig. 2 and 3 (documentation: AMK digital parallel interface "ADPS") are reduced to the signal curve for "correct echo". However, the echo information must not be formed or checked. However, the strobe signal required for acknowledgement (EStb for writing and AStb for reading) remains part of the signal curves.

[2] Due to this option a renewed drive commanding is performed with regard to the same drive only if the drive status is "Ready" (cf. Section 2.1).

[3] To be able to detect by a higher level system the transition from a write cycle to a read cycle (in the course of a "Read system info" request), the "low" time of the EStb signal (on cycle change) may not be less than the time that can be resolved by the higher level system (e.g. limited by the cycle time of a higher level SPS) (cf. documentation: AMK digital parallel interface "ADPS").

Caution:
In the transition from a read cycle to a write cycle (in the course of a following order after a "Read system info" request), a minimum "low" time of the AStb signal (formed by the higher level system) of 4ms must be guaranteed!

Remarks: The commanding data blocks DB 01 to DB 15, with a word length of 16 words, are generated implicitly by initializing the AMK digital parallel interface.

;**Data block DB01**

```
;DB01
;Command-DB for operating mode change start after HBA0;
;generated by FB220

:KD 0          ;ANTRK    = 0   (AW1)
:KB 1          ;KMDCODE  = 1   (Start)
:KB 1          ;KMDFKT   = 1   (BA change)
:KB 0          ;KMDBA    = 0   (HBA0)
:KB 0          ;KMDBIT   = 0   (default ID values)
:KD 0          ;KMD_VAR1 = 0   (2 words spare)
:KD 0          ;KMD_VAR2 = 0   (2 words spare)
:KD 0          ;KMD_VAR3 = 0   (2 words spare)
:KD 0          ;KMD_VAR4 = 0   (2 words spare)
:KD 0          ;KMD_VAR5 = 0   (2 words spare)
:KD 0          ;KMD_VAR6 = 0   (2 words spare)
```

;**Data block DB02**

```
;DB02
;Command-DB for parameter set switch-over start;
;generated by FB220

:KD 0          ;ANTRK    = 0   (AW1)
:KB 1          ;KMDCODE  = 1   (Start)
:KB 2          ;KMDFKT   = 2   (Parameter set switch-over)
:KB 0          ;KMDBA    = 0   (HBA0)
:KB 0          ;KMDBIT   = 0   (default ID values)
:KD 0          ;KMD_VAR1 = 0   (2 words spare)
:KD 0          ;KMD_VAR2 = 0   (2 words spare)
:KD 0          ;KMD_VAR3 = 0   (2 words spare)
:KD 0          ;KMD_VAR4 = 0   (2 words spare)
:KD 0          ;KMD_VAR5 = 0   (2 words spare)
:KD 0          ;KMD_VAR6 = 0   (2 words spare)
```

;**Data block DB03**

```
;DB03
;Command-DB for dig. torque control start;
;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 spare)
:KD 0          ;KMD_VAR2 = 0   (2 words spare)
:KD 0          ;KMD_VAR3 = 0   (2 words spare)
:KD 0          ;KMD_VAR4 = 0   (2 words spare)
:KD 0          ;KMD_VAR5 = 0   (2 words spare)
:KD 0          ;KMD_VAR6 = 0   (2 words spare)
```

;**Data block DB04**

```
;DB04
;Command-DB for dig. speed control start;
;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 spare)
:KD 0          ;KMD_VAR2 = 0   (2 words spare)
:KD 0          ;KMD_VAR3 = 0   (2 words spare)
:KD 0          ;KMD_VAR4 = 0   (2 words spare)
:KD 0          ;KMD_VAR5 = 0   (2 words spare)
:KD 0          ;KMD_VAR6 = 0   (2 words spare)
```

;**Data block DB05**

```
;DB05
;Command-DB for homing run start;
;generated by FB220

:KD 0          ;ANTRK    = 0   (AW1)
:KB 1          ;KMDCODE  = 1   (Start)
:KB 5          ;KMDFKT   = 5   (Homing run)
:KB 8          ;KMDBA    = 8   (NBA8)
:KB 0          ;KMDBIT   = 0   (default ID values)
:KD 0          ;KMD_VAR1 = 0   (2 words spare)
:KD 0          ;KMD_VAR2 = 0   (2 words spare)
:KD 0          ;KMD_VAR3 = 0   (2 words spare)
:KD 0          ;KMD_VAR4 = 0   (2 words spare)
:KD 0          ;KMD_VAR5 = 0   (2 words spare)
:KD 0          ;KMD_VAR6 = 0   (2 words spare)
```

;**Data block DB06**

```
;DB06
;Command-DB for spindle positioning start;
;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 spare)
:KD 0        ;KMD_VAR2 = 0   (2 words spare)
:KD 0        ;KMD_VAR3 = 0   (2 words spare)
:KD 0        ;KMD_VAR4 = 0   (2 words spare)
:KD 0        ;KMD_VAR5 = 0   (2 words spare)
:KD 0        ;KMD_VAR6 = 0   (2 words spare)
```

Data block DB07

```
;DB07
;Command-DB for absolute positioning start;
;generated by FB220

:KD 0        ;ANTRK    = 0   (AW1)
:KB 1        ;KMDCODE  = 1   (Start)
:KB 7        ;KMDFKT   = 7   (Absolute pos.)
:KB 8        ;KMDBA    = 8   (NBA8)
:KB 0        KMDBIT    = 0   (default ID values)
:KD 0        ;KMD_VAR1 = 0   (2 words spare)
:KD 0        ;KMD_VAR2 = 0   (2 words spare)
:KD 0        ;KMD_VAR3 = 0   (2 words spare)
:KD 0        ;KMD_VAR4 = 0   (2 words spare)
:KD 0        ;KMD_VAR5 = 0   (2 words spare)
:KD 0        ;KMD_VAR6 = 0   (2 words spare)
```

Data block DB08

```
;DB08
;Command-DB for relative positioning start;
;generated by FB220

:KD 0        ;ANTRK    = 0   (AW1)
:KB 1        ;KMDCODE  = 1   (Start)
:KB 8        ;KMDFKT   = 8   (Relative pos.)
:KB 8        ;KMDBA    = 8   (NBA8)
:KB 0        ;KMDBIT   = 0   (default ID values)
:KD 0        ;KMD_VAR1 = 0   (2 words spare)
:KD 0        ;KMD_VAR2 = 0   (2 words spare)
:KD 0        ;KMD_VAR3 = 0   (2 words spare)
:KD 0        ;KMD_VAR4 = 0   (2 words spare)
:KD 0        ;KMD_VAR5 = 0   (2 words spare)
:KD 0        ;KMD_VAR6 = 0   (2 words spare)
```

Data block DB09

```
;DB09
;Command-DB reserved;
;generated by FB220

:KD 0          ;ANTRK    = 0   (AW1)
:KB 1          ;KMDCODE  = 1   (Start)
:KB 9          ;KMDFKT   = 9   (reserved)
:KB 0          ;KMDBA    = 0   (HBA0)
:KB 0          ;KMDBIT   = 0   (default ID values)
:KD 0          ;KMD_VAR1 = 0   (2 words spare)
:KD 0          ;KMD_VAR2 = 0   (2 words spare)
:KD 0          ;KMD_VAR3 = 0   (2 words spare)
:KD 0          ;KMD_VAR4 = 0   (2 words spare)
:KD 0          ;KMD_VAR5 = 0   (2 words spare)
:KD 0          ;KMD_VAR6 = 0   (2 words spare)
```

Data block DB10

```
;DB10
;Command-DB for synchronous control start;
;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 spare)
:KD 0          ;KMD_VAR2 = 0   (2 words spare)
:KD 0          ;KMD_VAR3 = 0   (2 words spare)
:KD 0          ;KMD_VAR4 = 0   (2 words spare)
:KD 0          ;KMD_VAR5 = 0   (2 words spare)
:KD 0          ;KMD_VAR6 = 0   (2 words spare)
```

Data block DB11

```
;DB11
;Command-DB for position feedback value shift start;
;generated by FB220

:KD 0          ;ANTRK    = 0   (AW1)
:KB 1          ;KMDCODE  = 1   (Start)
:KB 11         ;KMDFKT   = 11  (Position feedback value shift)
:KB 0          ;KMDBA    = 0   (HBA0)
:KB 0          ;KMDBIT   = 0   (default ID values)
:KD 0          ;KMD_VAR1 = 0   (2 words spare)
:KD 0          ;KMD_VAR2 = 0   (2 words spare)
:KD 0          ;KMD_VAR3 = 0   (2 words spare)
:KD 0          ;KMD_VAR4 = 0   (2 words spare)
:KD 0          ;KMD_VAR5 = 0   (2 words spare)
:KD 0          ;KMD_VAR6 = 0   (2 words spare)
```

Data block DB12

```
;DB12
;Command-DB reserved;
;generated by FB220

:KD 0          ;ANTRK    = 0    (AW1)
:KB 1          ;KMDCODE  = 1    (Start)
:KB 12         ;KMDFKT   = 12   (reserved)
:KB 0          ;KMDBA    = 0    (HBA0)
:KB 0          ;KMDBIT   = 0    (default ID values)
:KD 0          ;KMD_VAR1 = 0    (2 words spare)
:KD 0          ;KMD_VAR2 = 0    (2 words spare)
:KD 0          ;KMD_VAR3 = 0    (2 words spare)
:KD 0          ;KMD_VAR4 = 0    (2 words spare)
:KD 0          ;KMD_VAR5 = 0    (2 words spare)
:KD 0          ;KMD_VAR6 = 0    (2 words spare)
```

Data block DB15

```
;DB15
;Command-DB reserved;
;generated by FB220

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

The following AWL program example serves for illustrating the mode of operation of FB220

Data block DB22

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

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

:BE
```

Data block DB36

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

:KB 0        ;(DR0) Reserve             = 0
:KB 0        ;(DL0) Max. num. rep.      = 0 (no repetition on
             ;                               error)
:KB 1        ;(DR1) Mode-Select         = 1 (without echo and without
             ;                               ready evaluation)
:KB 5        ;(DL1) Strobe-Low-Time     = 5 (ms)
:KB 0        ;(DR2) E-SB address        = 0 (EB 0)
:KB 8        ;(DL2) E-DB address        = 8 (EB 8)
:KB 0        ;(DR3) A-SB address        = 0 (AB 0)
:KB 8        ;(DL3) A-DB address        = 8 (AB 8)
:KF 0        ;(DW4) Timeout             = 0 (no timeout
             ;                                monitoring)
:KF 0        ;(DW5) Reserve             = 0
```

3 Impressum

Title **AMK specific function blocks**

Objective **Description of the AMK specific function blocks**

Part-Number **27889**

History

Date
2000/42

Copyright

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