



AMKASYN

VARIABLE SPEED DRIVES

Application Note No. AP_2005_03-1e

Date: 21.01.2004

Author: A. Noll

Rights reserved to make technical changes

This note replaces the Application note AP_2004_50-1e

Memory allocation for the electronic name plate on EnDat and Hiperface motor encoders

1 Introduction

The memory allocation of the electronic name plate for AMK motors is described in this document.

Beside the AMK type designation and the AMK serial number the electronic name plate also contains all the necessary motor data which are required for the drive parameter setting.

2 Realization

The electronic name plate is available on AMK motors with HEIDENHAIN EnDat encoders (type E and F) and STEGMANN Hiperface encoders (type S and T). The electronic name plate is stored in an OEM memory location on these encoder types.

The encoder documentation of the respective manufacturer provides the instruction how the electronic name plate data can be read directly, e. i. without use of the AMK inverter module.

2.1 Starting address

The starting address of the electronic name plate is depending on the used encoder type:

Encoder type	Starting address	Remark
EnDat	Beginning of the OEM memory location	Must be read out according to the manufacturer's instruction
Hiperface	Addresses 0 / Data field 0	

2.2 Memory allocation for EnDat and Hiperface encoder

The physical address on the encoder results from the addition of the starting address according to chapter 2.1 plus the offset from the following table (word or byte offset, depending on encoder make):

Offset		Parameter name Motor data	AMK ID No.	Unit	Remark
Word	Byte				
0	0	Reserved			
	1				
1	2	Reserved			
	3				
2	4	Reserved			
	5				
3	6	Version number Low byte		Hex	Part 2 of the version number e.g. 05h --> Encoder data base has version 1.05
	7	Version number High Byte		Hex	
4	8	Reserved			
	9				
5	10	Checksum of the data base			Low word at the addition result of all memory cell starting at word 6 and ending at word 63 for EnDat encoder & Hiperface with EEPROM size => 256 Byte word 47 for Hiperface encoder with EEPROM size < 256 Byte
	11				
6	12	Reserved			Enters into the checksum
	13				
7	14	Reserved			Enters into the checksum
	15				
8	16	Part number character 1	34160	ASCII	Motor part number, max. 10*ASCII characters, e. g. D352AD. "20h" is entered for not used characters.
	17	Part number character 2			
9	18	Part number character 3			
	19	Part number character 4			
10	20	Part number character 5			
	21	Part number character 6			
11	22	Part number character 7			
	23	Part number character 8			

Offset		Parameter name Motor data	AMK ID No.	Unit	Remark
Word	Byte				
12	24	Part number character 9			
	25	Part number character 10			
13	26	Motor manufacturing date	34161	--	Year and calender week YYWW e. g. 0449,
	27				
14	28	Motor serial number	34162	--	Consecutive number, max. 4'294'967'296, e. g. 388001
	29	Low word			
15	30	Motor serial number			
	31	High Word			
16	32	Speed controller:	100	--	Value for motor without load, depending on the application
	33	Proportinal gain K_P			
17	34	Speed controller:	101	0,1 ms	Value for motor without load, depending on the application
	35	Integral-action time T_N			
18	36	Speed controller:	102	0,1 ms	Value for motor without load, depending on the application
	37	Derivative-action time T_D			
19	38	Position loop gain K_V	104	1/min	Value for motor without load, depending on the application
	39				
20	40	Maximum motor current 1)	109	exponential 0,001 A	
	41	I_{max}			
21	42	Motor nominal current I_N 1)	111	exponential 0,001 A	
	43				
22	44	Nominal motor voltage	32768	0,1 V	e. g. 190V or 350V
	45				
23	46	Magnetizing current I_M 1)	32769	exponential 0,001 A	
	47				
24	48	Magnetising current I_{M1} 1)	32770	exponential 0,001 A	
	49				
25	50	Nominal torque M_N	32771	0.1Nm	
	51				
26	52	Nominal speed n_N 1)	32772	exponential 0,0001 rpm	
	53				
27	54	Rotor time constant T_R	32774	0.0001s	Only for asynchronous motor
	55				
28	56	Number of motor poles	32775	--	
	57				
29	58	Sine encoder periods	32776	--	Encoder periods available from the EnDat encoder, on HIPERFACE encoder

Offset		Parameter name Motor data	AMK ID No.	Unit	Remark
Word	Byte				
	59				(Stegmann) find out via encoder identifier. This value can be used for calculation of the encoder gear ratio.
30	60 61	Encoder type	32953	--	Motor principle (SM synchronous motor / AM asynchronous motor) respectively encoder type (for AIPEX).
31	62 63	Cross field inductance L_q	34045	0,1 mH	Only for synchronous motor
32	64 65	Main axis inductance. L_d	34046	0,1 mH	Only for synchronous motor
33	66 67	Current controller T_{nq}	34050	0,1 ms	
34	68 69	Current controller T_{nd}	34052	0,1 ms	
35	70 71	Continous standstill current I_0 1)	34096	exponential 0,001 A	
36	72 73	Voltage controller K_{pu}	34148	0,001 A / V	
36	74 75	Voltage controller T_{nu}	34149	0,1 ms	
38	76 77	Current controller K_{pq}	34151	0,01 V / A	
39	78 79	Current controller K_{pd}	34152	0,01 V / A	
40	80 81	Terminal resistance R_{tt}	34164	0,01 Ohm	
41	82 83	Brake holding torque	34165	0.1 Nm	0 = No brake
42	84 85	Temperature sensor	34166	--	Coding (decimal): Temperature (0 - 654), Quantity (0 - 9), Type (0 – 9 e. g.: 13013 = 130-1-3 Type: No sensor = 0, THW = 1, PTC = 3, NTC = 5, KTY = 6, PT100 = 7, Type J = 8, other = 9
43	86 87	Terminal inductance L_{tt}	34167	0,1 mH	
44	88 89	Time period of I_{max}	34168	0,1 s	I^2T monitoring
45	90	Reserved			

Offset		Parameter name Motor data	AMK ID No.	Unit	Remark
Word	Byte				
	--				
46 and following	--	Reserved			
	--				

Comment:

1) The 32 bit value (ID content) is stored on the encoder as a 16 bit floating-point mantissa value. The unit (resolution) then must be considered.

The mantissa value is represented by the 12 bit mantissa and the 4 bit exponent. The mantissa is limited to 3 digits:

Mantissa value range: 0 ... 4095 respectively -2048 ... 2047; limited to +/- 999

Exponent value range: 0 ... 16 (for 10^0 ... 10^{16})

Example (current): 4,52 [0.001A] → internal 4520 → 452×10^1 → Mantissa = 452 Exponent = 1

Example (speed): 3120 [0.0001 rpm] → internal 3120000 → 312×10^5 → Mantissa = 312 Exponent = 5