


Overview over the changes and new functionality of Motion Controller V3.0 firmware revision J.

Changes and fixes

No.	Affected Component	Change Description
1	Object dictionary	Corrected some parameter names and typos
2	Trace support	When using negative trigger delay it is now checked to have values only out of the current recording in the buffer
3	Homing Offset	The offset did not care for the settings of the factor group. So the value had to be given in internal position units – this is fixed now.
5	CAN-timing	The internal timing of the 250kBit was previously according to the official recommendations but error frames were still observed. The internal resolution has been increased in this revision. The rate of error frames is reduced significantly.
6	Detection of AES CRC error	CRC errors of the AES had not been reposted as encoder errors up to revision I.
7	Handling of following error and speed deviation	Following error and speed deviation error are reset to 0 when the related controller is disabled. The timing of the reset has been optimized to ensure a reset under all conditions.
8	RS232 net-mode	When using a segmented transfer, the answering controller did not release the RS232 Tx line again. The net-mode did not work any longer in such a case.
9	Touch-probe control	<p>The touch-probes require a position source to be selected in the bits 6+7 / 14+15 of 0x60B8.</p> <p>The selection is now:</p> <ul style="list-style-type: none"> 00: use the motor encoder (new default) 01: use the motor encoder (unchanged) 10: use the ref-encoder (unchanged) 11: reserved <p>This will allow for a direct compatibility of a TwinCAT NC axis.</p>

New Functions

No.	Affected Component	Description															
1	Target position handling	<p>Support for round table operation in CSP mode has been added. Via the object Positioning option code 0x60F2 the movement type standard or shortest can be selected.</p> <table border="1"> <thead> <tr> <th>Bit 7</th> <th>Bit 6</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Normal positioning similar to linear axis; If reaching or exceeding the position range limits (607B_n) the input value shall wrap automatically to the other end of the range. Positioning can be relative or absolute. Only with this bit combination, the movement greater than a modulo value is possible.</td> </tr> <tr> <td>0</td> <td>1</td> <td>Positioning only in negative direction; if target position is higher than actual position, axis moves over the min position limit (607D_n, sub-index 01_n) to the target position.</td> </tr> <tr> <td>1</td> <td>0</td> <td>Positioning only in positive direction; if target position is lower than actual position, axis moves over the max position limit (607E_n, sub-index 01_n) to the target position.</td> </tr> <tr> <td>1</td> <td>1</td> <td>Positioning with the shortest way to the target position. NOTE If the difference between actual value and target position in a 360° system is 180°, the axis moves in positive direction.</td> </tr> </tbody> </table> <p>Figure 44 shows movement examples depending on settings of the bits 6 and 7. Here the min position range limit (607B_n, sub-index 01_n) is 0° and the max position range limit (607E_n, sub-index 02_n) is 360°.</p>  <p>Only negative or only positive are not yet implemented.</p>	Bit 7	Bit 6	Definition	0	0	Normal positioning similar to linear axis; If reaching or exceeding the position range limits (607B _n) the input value shall wrap automatically to the other end of the range. Positioning can be relative or absolute. Only with this bit combination, the movement greater than a modulo value is possible.	0	1	Positioning only in negative direction; if target position is higher than actual position, axis moves over the min position limit (607D _n , sub-index 01 _n) to the target position.	1	0	Positioning only in positive direction; if target position is lower than actual position, axis moves over the max position limit (607E _n , sub-index 01 _n) to the target position.	1	1	Positioning with the shortest way to the target position. NOTE If the difference between actual value and target position in a 360° system is 180°, the axis moves in positive direction.
Bit 7	Bit 6	Definition															
0	0	Normal positioning similar to linear axis; If reaching or exceeding the position range limits (607B _n) the input value shall wrap automatically to the other end of the range. Positioning can be relative or absolute. Only with this bit combination, the movement greater than a modulo value is possible.															
0	1	Positioning only in negative direction; if target position is higher than actual position, axis moves over the min position limit (607D _n , sub-index 01 _n) to the target position.															
1	0	Positioning only in positive direction; if target position is lower than actual position, axis moves over the max position limit (607E _n , sub-index 01 _n) to the target position.															
1	1	Positioning with the shortest way to the target position. NOTE If the difference between actual value and target position in a 360° system is 180°, the axis moves in positive direction.															
2	Block commutation	<ul style="list-style-type: none"> • The commutation itself has been accelerated. • Current measurement is improved around the commutation event itself. • The resolution of the speed measurement has been increased to allow for high speed motors. 															
3	PWM	<p>The frequency of the motor PWM can now be changed via 0x250A.00:</p> <ul style="list-style-type: none"> • 1: 20 kHz • 2: 40kHz • 3: 60kHz • 4: 80kHz • 5: 100kHz 															
5	Back-EMF compensation	<p>Compensation of the typical differences of the back-EMF of BX4 motors. Is observable in volt-mode only. Will lead to better compensation of A-Hall signals too. Can be switched on/off using 0x2505.01.</p>															
8	Homing Support	<p>Options to be used during homing: 0x233F.00 bit 4: ignore SoftPosLimits during homing 0x233F.00 bit 5: use homing torque limits (0x2350/0x2351) during homing sequence. This is recommended for homings -1 ... -4 using a mechanical block.</p>															
9	Use of target-offset in AxC modes	<p>Target value offsets (0x60B0 ... 0x60B2) are applied in AxC modes too. This allows for following-error compensation during APC mode.</p>															

BASIC 2

No.	Improvement	Description			
1	Support of loops	Standard BASIC DO ... LOOP is supported having the variants: <table border="1" data-bbox="624 371 1445 506"> <tr> <td>DO ... LOOP</td> <td>DO WHILE <cond> ... LOOP</td> <td>DO ... LOOP UNTIL <cond></td> </tr> </table>	DO ... LOOP	DO WHILE <cond> ... LOOP	DO ... LOOP UNTIL <cond>
DO ... LOOP	DO WHILE <cond> ... LOOP	DO ... LOOP UNTIL <cond>			
2	Support of functions	Sub-functions can be defined using the key-word FUNCTION: FUNCTION (Param1, Param2, ...) ... RETURN result END FUNCTION			
3	Support of symbolic variables	Using DIM symbolic variables can be created DIM counter ... counter = counter +1 DIM is supported in functions too. When used in functions it will allocate a local variable which is visible within the function only			
4	#DEFINE	Can be used to assign a value to a symbolic name #DEFINE OpModePP 1 Will assign the value 1 to the symbolic name 1. The symbolic name can now be used wherever the numeric value would have to be used.			
5	#INCLUDE	Can be used to include support files. These can either be the ones supplied by FAULHABER or files defined by the user. The included files can contain predefined sets of symbolic names or even complete functions.			
7	Shift Operator >> and <<	Introducing a C-like operator for bit-wise shift a = a<<2 will shift the contents of a by 2 positions to the left b = b>>2 will shift the contents of b by 2 positions to the right			
8	Use of GETOBJ in expressions	GETOBJ can now also be used as a part of an expression like a condition e.g. IF ((GETOBJ \$6041.00 & \$6F) = \$40) THEN			
9	Use of protection	If a script has been protected using the key 0x3003.00 the read- and write access to the variables using 0x3004 and 0x3005 is still possible now.			

Use of Analog-Inputs

No.	Improvement	Description
	Use of AnIn1/2 as DigIn	<p>AnIn1/2 can be selected to be used as emulated DigIns. The number of emulated inputs is selected via 0x2300.04: = 0: no emulated DigIn = 1: AnIn1 is used to emulate DigIn4 = 2: additionally AnIn2 is used to emulate DigIn5</p> <p>The threshold of the emulated DigIns is configured using 0x2310.05: Bit 0: threshold of emulated DigIn4 (0: TTL / 1: PLC) Bit 1: threshold of emulated DigIn5 (0: TTL / 1: PLC)</p> <p>Emulated DigIn5 is not available for MC 5004 STO</p>
	Simulated AnInx	<p>The inputs of AnIn1 and AnIn2 can be switched to simulated values. The values to be used are configured in 0x2313.09: simulated raw input value of AnIn1 0x2313.19: simulated raw input value of AnIn2</p> <p>The simulated values are activated when a value > 0 is written to 0x2313.xA.</p>