Dokumentation

Technikerarbeit 2003 / 2004

Handbuch

# Automatisierung von Akustikmessungen



## 1 Wichtige Hinweise

## 1.1 Zweck des Handbuches

Die Informationen dieses Handbuches ermöglichen es Ihnen :

- den Messautomat zur Messung vorzubereiten und aufzubauen.
- den Messautomat über das integrierte Touchpanel zu Bedienen.
- Funktionsabläufe des Messsystems kennenzulernen.
- technische Daten des Messautomaten und seiner Komponenten nachzuschlagen.

## **1.2** Dokumentation zur Programmierung

Um Änderungen im Ablauf des Messverfahrens prozesssicher durchführen zu können, ist diese Dokumentation nicht ausreichend. Auch Änderungen der Bedienoberflächen des integrierten Touchpanels sollte ausschließlich durch eingewiesenes Fachpersonal mit Hilfe der herstellerspezifischen Dokumentation durchgeführt werden.

Weitere Hinweise zu herstellerbezogenen Dokumentationen erhalten Sie in den betreffenden Kapiteln.

## 1.3 CD-ROM

Diese Dokumentation ist auch auf beiliegender CD-ROM enthalten.

Weitere technische Daten zu Komponenten, die dieser Messautomat beinhaltet und nicht in der Handbuchausgabe enthalten sind, finden Sie ebenso auf der CD-ROM.

## 1.4 Wegweiser

Um Ihnen einen schnellen Zugriff auf spezielle Informationen zu erleichtern, enthält das Handbuch folgende Zugriffshilfen :

- Am Anfang des Handbuches steht Ihnen ein Gesamtinhaltsverzeichnis zur Verfügung.
- Ein Abbildungsverzeichnis ist im Gesamtinhaltsverzeichnis integriert.

## 1.5 Weitere Unterstützung

Bei Fragen zur Nutzung der im Handbuch beschriebenen Produkte, die Sie hier nicht beantwortet finden, wenden Sie sich bitte jeweils an den entsprechenden Komponentenhersteller.

Bei Fragen, die das Komplettsystem betreffen, wenden Sie sich bitte an Ihren **\_\_\_\_\_\_\_**Ansprechpartner.

## 1.6 Ständig aktuelle Informationen

Ständig aktuelle Informationen zu diesem Messsystem erhalten Sie über Ihren **BOSSE**<sup>1</sup>- Ansprechpartner.</sup>

# Inhalt

1	WICHTIGE HINWEISE	3
1.1	Zweck des Handbuches	3
1.2	Dokumentation zur Programmierung	3
1.3	CD-ROM	3
1.4	Wegweiser	3
1.5	Weitere Unterstützung	4
1.6	Ständig aktuelle Informationen	4
2	SINN UND ZWECK	13
<b>2.1</b> 2	Warum werden Akustikmessungen durchgeführt ?         1.1       Was ist "tunen" ?	<b> 13</b> 13
2.2	Warum Automatisierung des Messsystems ?	14
3	DER MESSAUFBAU	15
3.1	Messaufbau Blockschaltbild (Abb. 3 – 1)	<b> 15</b> 15
4	ÜBERSICHT AUFBAU MESSAUTOMAT	17
4 4.1	ÜBERSICHT AUFBAU MESSAUTOMAT	17 17
4 4.1 4.2	ÜBERSICHT AUFBAU MESSAUTOMAT Allgemein Teil 1 : Mechanik	17 17 17
4 4.1 4.2 4 4	ÜBERSICHT AUFBAU MESSAUTOMAT.         Allgemein.         Teil 1 : Mechanik.         2.1 Gehäuse         1.2 Antriebseinheiten.	<b>17</b> <b>17</b> <b>17</b> 17 17
4 4.1 4.2 4 4 4.3 4 4	ÜBERSICHT AUFBAU MESSAUTOMAT.         Allgemein.         Teil 1 : Mechanik.         2.1 Gehäuse         1.2 Antriebseinheiten.         Teil 2 : Elektrik         3.1 Hauptplatine         3.2 HMI – Human Machine Interface.         3.3 externe Anschlüsse der Frontplatte	17 17 17 17 17 17 17 18 18
4 4.1 4.2 4 4 4 4 4 4 4 4.4	ÜBERSICHT AUFBAU MESSAUTOMAT	17 17 17 17 17 17 17 18 18 19
4 4.1 4.2 4 4 4 4 4 4 4 4.4 4.4	ÜBERSICHT AUFBAU MESSAUTOMAT	17 17 17 17 17 17 17 18 18 19 19 20
4 4.1 4.2 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5.1	ÜBERSICHT AUFBAU MESSAUTOMAT.         Allgemein.         Teil 1 : Mechanik.         2.1 Gehäuse         1.2 Antriebseinheiten.         Teil 2 : Elektrik         3.1 Hauptplatine         3.2 HMI – Human Machine Interface.         3.3 externe Anschlüsse der Frontplatte.         Blockschaltbild : neue Komponenten des Messautomaten         (Abb. 4-1).         KURZFUNKTIONSBESCHREIBUNG.         Kurzfunktionsbeschreibung Ablauf Messautomat	17 17 17 17 17 17 18 18 18 19 20 20
4 4.1 4.2 4 4 4 4 4 4 4 4.4 5 5.1 6	ÜBERSICHT AUFBAU MESSAUTOMAT.         Allgemein.         Teil 1 : Mechanik.         2.1 Gehäuse         1.2 Antriebseinheiten.         Teil 2 : Elektrik         3.1 Hauptplatine         3.2 HMI – Human Machine Interface.         3.3 externe Anschlüsse der Frontplatte.         Blockschaltbild : neue Komponenten des Messautomaten         (Abb. 4-1).         KURZFUNKTIONSBESCHREIBUNG         Kurzfunktionsbeschreibung Ablauf Messautomat         DETAIL AUFBAU MESSAUTOMAT.	17 17 17 17 17 17 17 18 18 19 20 20 22

(Abb. 6-1)	22
6.1.2 Antriebseinheiten	22
(Abb. 6-2)	23
(Abb. 6-3)	23
(Abb. 6-4,5)	23
(Abb. 6-6)	24
6.1.3 Dimensionierung der mechanischen Komponenten	24
6.1.3.1 Dimensionierung des Gehäuses	24
6.1.3.2 Dimensionierung der Antriebseinheit	24
6.1.4 Sonstige mechanische Komponenten	26
6.1.4.1 Mikrotonnaiterung	26
6.1.5 Lechnische Zeichnungen	27
6. I.S. I Technische Zeichnungen Anthebseinneit	21
(ADD. 0-7)	21
(ADD. 0-8)	20
(ADD. 0-9)	20
(ADD. 0-10)	29
(ADD. 0-11)	29
(ADD. 0-12)	20
(ADD. 0-13)	3U 24
	31 21
(ADU. 0-14)	51
6.2 Toil 2 : Elektrik	วา
6.2.1 Spappungsvorsorgung	3 <b>∠</b> 22
6.2.1 Spannungsversorgung Messautomat	32 32
	32 22
(ADD. 0-15)	32 22
6.2.1.2 Spannungsversorgung des Netzteils der Mikrofone	32
6.2.1.4 Spannungsversorgung der HMI	32
6.2.2 Hauntolatine	33
6.2.2 Teil 1 · Basisnlatine	33
(Δhh 6-16)	33
6222 Teil 2 Prozessorolatine	34
(Abb 6-17)	34
(Abb. 6-18)	34
(Abb. 6-19)	35
623 Spindelantrieb	36
6231 DC-Motor	36
(Abb. 6-20)	36
(Abb. 6-21).	37
6.2.3.2 Inkrementalgeber	37
(Abb. 6-22).	37
(Abb. 6-23)	38
6.2.3.3 Getriebe	38
(Abb. 6-24)	38
(Abb. 6-25)	39
6.2.3.4 Kupplung	39
(Abb. 6-26)	39
6.2.4 Sensorik	39
6.2.4.1 Referenzpunktschalter	40
(Abb. 6-27)	40
6.2.4.2 Endlagenschalter	40
6.2.4.3 Deckelschalter	41
6.2.4.4 Neigungssensor	41
(Abb. 6-28)	41
(Abb. 6-29)	41
(Abb. 6-30)	42
6.2.5 HMI – Human Machine Interface	42
(Abb. 6-31)	42

	(Abb. 6-32) (Abb. 6-33)	
	(, , , , , , , , , , , , , , , , , , ,	
7	DETAIL BASISPLATINE	45
7.1	Spannungsversorgung	
	(Abb. 7-1)	
7.2	Logikteil	
	(Abb. 7-2)	
	(Abb. 7-3)	
	(Abb. 7-4) (Abb. 7-5)	
73	l eistungsteil	48
	(Abb. 7-6)	
	(Abb. 7-7)	
	(Abb. 7-8)	
	(Abb. 7-9)	
71	Anschlussmöglichkoiton zur Porinhorio	40
/.4	(Abb. 7-10)	
7.5	Serielle Schnittstellen	50
	(Abb. 7-11)	
7.6	Debugbereich	
	(Abb. 7-12)	51
7.7	Reservierter Bereich	
	(Abb. 7-13)	
7.8	Schaltpläne der Basisplatine	
	(ADD. 7-14) (Δbb. 7-15)	
	(Abb. 7-16)	53
	(Abb. 7-17)	
7.9	Layout Basisplatine	
	(ADD: 7-18)	
7.10	0 Bestückungsdruck Basisplatine	
	(Abb. 7-19)	
8	DETAILS PROZESSORPLATINE	
• •		
8.1	Allgemein	
8.2	Blockschaltbild Prozessorplatine	
	(Abb. 8-1)	
82	Pinbalagung Mikrocontroller	E7
0.3	(Abb. 8-2)	
9	PROGRAMMIERUNG DES MIKROCONTROLLERS C509	58
-		

9.1	Allgemeines	. 58
	(Abb. 9-1)	. 58
	(Abb. 9-2)	. 58
	(/	
92	Tasking	59
•	(Abb. 9-3)	59
	(100.00)	. 00
03	Ablaufdiagrammo C. Projokt	60
9.5	Ablautulayrainine C-Projekt	. 00
	(ADD. 9-4)	. 02
	(ADD. 9-5)	. 63
	(ADD. 9-6)	. 64
	(ADD. 9-7)	. 65
	(Abb. 9-8)	. 66
	(Abb. 9-9)	. 67
	(Abb. 9-10)	. 68
	(Abb. 9-11)	. 69
	(Abb. 9-12)	. 70
	(Abb. 9-13)	. 71
	(Abb. 9-14)	. 72
	(Abb. 9-15)	. 73
	(Abb. 9-16)	. 74
	(Abb. 9-17)	. 75
	(Abb 9-18)	76
	(Abb 9-19)	77
	(Abb 9-20)	78
	$(\Delta bb, 0.21)$	70
	$(\Delta bb, 0.22)$	80
	(Abb, 0.22)	Q1
	(ADD, 9-23)	.01
	(ADD. 9-24)	. 82
	(ADD. 9-25)	. 83
	(ADD. 9-26)	. 84
	(ADD. 9-27)	. 85
	(Abb. 9-28)	. 86
	(Abb. 9-29)	. 87
	(Abb. 9-30)	. 88
	(Abb. 9-31)	. 89
	(Abb. 9-32)	. 90
	(Abb. 9-33)	. 91
	(Abb. 9-34)	. 92
	(Abb. 9-35)	. 93
	(Abb. 9-36)	. 94
	(Abb. 9-37)	. 95
	(Abb. 9-38)	. 96
	(Abb. 9-39)	. 97
	(Abb. 9-40)	. 98
	(Abb. 9-41)	99
	(Abb 9-42)	100
	(Abb. 9-43)	101
	$(\Delta bb, Q_{-}AA)$	102
	(Abb, 0.45)	102
	(ADD, 9-45)	103
	(ADD, 9-40)	104
	(ADD. 9-47)	100
	(ADD. 9-48)	100
	(ADD. 9-49)	107
	(ADD. 9-5U)	108
	(Abb. 9-51)	109
	(Abb. 9-52)	110
9.4	Details zur Programmierung	111
9.4	.1 Allgemeines	111

9.4.3 9.4.4 9.4.5 9.4.6 9.4.7 9.4.8 9.4.9 9.4.1 9.4.1 9.4.1 9.4.1 9.4.1 9.4.1 9.4.1 9.4.1 9.4.1 9.4.1 9.4.1	<ul> <li>Modul 1: Initialisierung der seriellen Schnittstellen</li></ul>	111 113 113 115 116 116 116 117 118 120 125 127 130 131 132 134 135
9.4.1	9 Modul 19: Fahrbefehle in Betriebsart Automatik auslesen	136
9.4.2	1 Modul 30: Umdrehungszähler der Achsen 1 – 3 aktualisieren	139
9.4.2	2 Modul 31: Umdrehungszähler der Achse 4 aktualisieren	141
9.4.2	<ul> <li>Modul 32. Storung durch Endlagenschalter auswerten</li> <li>Modul 33: Leistungsteil Überlast</li> </ul>	141
9.4.2	5 Modul 34: Deckelschalter	143
9.4.2	6 Serielle Schnittstellen	144
10	KOMMUNIKATION	146
10.1	Allgemein	146
10.2	Kommunikation: HMI zu Mikrocontroller	146
	(Abb 10-1)	146
10.2		1/7
10.2.	1 Aufbau eines Befehls	147
10.2.	1       Aufbau eines Befehls         Kommunikation: Mikrocontroller zu HMI	147 148
10.2. <b>10.3</b> 10.3.	1 Aufbau eines Befehls         Kommunikation: Mikrocontroller zu HMI         1 Aufbau eines Befehls	147 <b>148</b> 148
10.2. <b>10.3</b> 10.3. <b>10.4</b>	1 Aufbau eines Befehls         Kommunikation: Mikrocontroller zu HMI.         1 Aufbau eines Befehls         1 Aufbau eines Befehls         Kommunikation: Externe Software und Mikrocontroller	147 <b>148</b> 148 <b>149</b>
10.2. <b>10.3</b> <b>10.4</b> 10.4	1 Aufbau eines Befehls         Kommunikation: Mikrocontroller zu HMI	147 <b>148</b> 148 <b>149</b> 149
10.2. <b>10.3</b> <b>10.3</b> <b>10.4</b> 10.4. 10.4.	1       Aufbau eines Befehls         2       Übersicht der Befehle         2       Übersicht der Befehle	<ul> <li>147</li> <li>148</li> <li>148</li> <li>149</li> <li>149</li> <li>149</li> <li>149</li> <li>149</li> <li>149</li> </ul>
10.2. <b>10.3</b> <b>10.4</b> 10.4. 10.4. 10.4. 10.4.	1       Aufbau eines Befehls         Kommunikation: Mikrocontroller zu HMI	147 <b>148</b> 148 <b>149</b> 149 149 149 150
10.2. <b>10.3</b> <b>10.4</b> 10.4. 10.4. 10.4. 10.4. 10.4.	<ul> <li>Aufbau eines Befehls</li></ul>	147 <b>148</b> 148 <b>149</b> 149 149 149 150 151
10.2. <b>10.3</b> <b>10.4</b> 10.4. 10.4. 10.4. 10.4. 10.4.	1       Aufbau eines Befehls         1       Aufbau eines Befehls         1       Aufbau eines Befehls         1       Aufbau eines Befehls         1       Allgemein         2       Übersicht der Befehle         2       Übersicht der Befehle         3       Aufbau eines Befehls         4       Sicherheit bei der Datenübertragung         (Abb. 10-3)       Aufbau eines Befehls	147 <b>148</b> 148 149 149 149 149 150 151
10.2. <b>10.3</b> <b>10.4</b> 10.4. 10.4. 10.4. 10.4. 10.4. 10.4.	1       Aufbau eines Befehls         Kommunikation: Mikrocontroller zu HMI	147 <b>148</b> 148 149 149 149 150 151 151 152
10.2. <b>10.3</b> <b>10.4</b> 10.4. 10.4. 10.4. 10.4. 10.4. 10.4.	1       Aufbau eines Befehls         Kommunikation: Mikrocontroller zu HMI	147 <b>148</b> 148 149 149 149 150 151 151 152
10.2. <b>10.3</b> <b>10.4</b> 10.4 10.4 10.4 10.4 10.4 <b>10.4</b> <b>10.4</b>	1       Aufbau eines Befehls         Kommunikation: Mikrocontroller zu HMI.         1       Aufbau eines Befehls         1       Aufbau eines Befehls         1       Allgemein.         2       Übersicht der Befehle.         (Abb. 10-2).       3         3       Aufbau eines Befehls         4       Sicherheit bei der Datenübertragung         (Abb. 10-3).       5         5       Sicherheit beim Prozess	147 <b>148</b> 148 149 149 149 150 151 151 152 <b>153</b>
10.2. <b>10.3</b> <b>10.4</b> 10.4. 10.4. 10.4. 10.4. 10.4. <b>11.1</b>	1       Aufbau eines Befehls         Kommunikation:       Mikrocontroller zu HMI	147 148 148 149 149 149 150 151 151 152 <b>153</b>
10.2. 10.3 10.4 10.4 10.4 10.4 10.4 10.4 10.4 10.4 10.4 11.1 11.1	1       Aufbau eines Befehls         Kommunikation: Mikrocontroller zu HMI	<ul> <li>147</li> <li>148</li> <li>149</li> <li>149</li> <li>149</li> <li>150</li> <li>151</li> <li>151</li> <li>152</li> <li>153</li> <li>153</li> <li>153</li> </ul>
10.2. 10.3 10.4 10.4 10.4 10.4 10.4 10.4 10.4 10.4 10.4 11.1 11.1	1       Aufbau eines Befehls         Kommunikation: Mikrocontroller zu HMI	147 148 148 149 149 149 150 151 151 152 153 153 153
10.2. <b>10.3</b> <b>10.4</b> 10.4. 10.4. 10.4. 10.4. 10.4. <b>10.4.</b> <b>11.1</b> <b>11.1</b> <b>11.2</b>	1       Aufbau eines Befehls         Kommunikation: Mikrocontroller zu HMI	147 <b>148</b> 149 149 149 149 150 151 151 152 <b>153</b> <b>153</b> <b>153</b> 154 155
10.2. 10.3 10.4 10.4 10.4 10.4 10.4 10.4 10.4 10.4 11.1 11.1	1       Aufbau eines Befehls         Kommunikation: Mikrocontroller zu HMI	147 <b>148</b> <b>149</b> 149 149 149 150 151 151 152 <b>153</b> <b>153</b> <b>153</b> <b>153</b> <b>153</b> <b>153</b> <b>154</b> 155 156
10.2. 10.3 10.4 10.4 10.4 10.4 10.4 10.4 10.4 10.4 11.1 11.1 11.2	1       Aufbau eines Befehls         Kommunikation: Mikrocontroller zu HMI	147 148 148 149 149 149 150 151 151 152 153 153 153 154 155 156 157
10.2. 10.3 10.3 10.4 10.4 10.4 10.4 10.4 10.4 10.4 11 11.1 11.2	1       Aufbau eines Befehls         Kommunikation: Mikrocontroller zu HMI	147 <b>148</b> <b>149</b> 149 149 149 150 151 151 152 <b>153</b> <b>153</b> <b>153</b> <b>153</b> <b>153</b> <b>155</b> 156 157 158 159
10.2. 10.3 10.4 10.4 10.4 10.4 10.4 10.4 10.4 10.4 11.1 11.1	1       Aufbau eines Befehls         Kommunikation: Mikrocontroller zu HMI	147 <b>148</b> <b>149</b> 149 149 149 149 150 151 151 <b>153</b> <b>153</b> <b>153</b> <b>153</b> <b>153</b> <b>153</b> <b>154</b> 155 156 157 158 159 160

11.3	Details zum Quelltext	161
11	3 1 Befehlstabelle	161
	(Abb 11-8)	161
	(Δbb 11-9)	162
11	(XBC 11 0)	163
11	3.3 Ouelltextauszug: Intro	164
	$(\Delta hh 11_10)$	165
11	(Abb. 11-10)	165
	(Abb. 11 11)	165
11	(ADD. 11-11)	166
		166
11	(ADD. 11-12)	167
	(Abb. 11.12)	167
11	(ADD. TT-TS)	167
		107
11	(ADD. 11-14)	100
11		109
	(ADD. 11-15)	109
11.4	Compilieren und Laden des Projekts	170
11.4	Testen der erstellten Bilder	170
	(Abb. 11-16)	171
	(Abb. 11-17)	171
12	BEDIENUNG DES MESSAUTOMATEN	172
12.1	Allgemein	172
12.2	Bedienung über die HMI	172
12	2.2.1 Betriebsartenwahl	172
	(Abb. 12-1)	172
12	2.2.2 Betriebsartenwahl nach Störung	173
	(Abb. 12-2)	173
12	2.2.3 Betriebsart Teilautomatik	174
	(Abb. 12-3)	174
12	2.2.4 Betriebsart Hand	175
	(Abb. 12-4)	176
	(Abb. 12-5)	177
	(Abb. 12-6,7.8)	178
12	2.2.5 Online-Hilfe zur Betriebsart Teilautomatik	178
	(Abb. 12-9)	178
	(Abb. 12-10)	179
12	2.2.6 Online-Hilfe zur Betriebsart Hand	180
	(Abb. 12-11)	180
12	2.2.7 Bedjenerführung: Störungen	180
. –	Störungen durch angefahrene Endlagen	180
	(Abb 12-12)	180
	Störungen durch Paarfehler	181
	Störungen durch Fehlbedienung	181
	(Abb 12-13)	182
	Störung Deckelüberwachung	183
	(Ahh 12-14)	183
	Störung Neigungssensor	18/
	(Δhb 12-15)	12/
	Störung Überleet	12/
10	0.010119 Oberlast	186
12	Moldungon hoim Start dos Systems	100
	(Abb. 12 16 17)	100
	الماطيية: المعاد المع	100
		100

	(Abb. 12-18)	186
N	leldung: Position erreicht	187
_	(Abb. 12-19)	187
N	leldung: Achse aktiv	187
N	leldung: Messung starten	188
	(Abb. 12-20)	188
12.3	Bedienung über OLLICONTROL	189
12.3	3.1 Installation	189
12.3	3.2 Vorbereitung	189
12.3	3.3 Bedienung OLLICON I ROL	189
	(Abb. 12-21)	189
	(Abb. 12-22)	190
	(Abb. 12-23)	190
_	(Abb. 12-24)	191
В	etriebsart Automatik	192
	(Abb. 12-25)	192
	(Abb. 12-26)	193
	(Abb. 12-27)	193
	(Abb. 12-28)	194
	(Abb. 12-29)	194
	(Abb. 12-30)	195
	(Abb. 12-31)	195
	(Abb. 12-32)	196
	(Abb. 12-33)	196
В	etriebsart Teilautomatik	197
	(Abb. 12-34)	197
	(Abb. 12-35)	197
	(Abb. 12-36)	198
	(Abb. 12-37)	198
	(Abb. 12-38)	199
U	ngeplante Ereignisse	200
	(Abb. 12-39)	200
	(Abb. 12-40)	200
	(Abb. 12-41)	201
40		~~~
13	STRUMLAUFPLANE	202
12 1	Allgemein	202
10.1		202
13 2	Stromlaufnläne Messautomat	202
10.2	(Abb 13-1)	202
	(Abb. 13-2)	203
	(Abb. 13-3)	203
	(Abb. 13-4)	200
	(Abb. 13-5)	204
	(Abb. 13-6)	205
	(Abb. 13-7)	205
	(Abb. 13-8)	206
		200
14	ADAPTERPLATINE	207
	A.U	oc-
14.1	Allgemein	207
14.2	Allgemeine Funktionen	207
14.3	Frontplatte	207
14.3	Frontplatte	<b>207</b> 207

14.4	Schaltplan der Adapterplatine	<b>208</b> 208
14.5	Layout der Adapterplatine	<b>208</b> 208
ANHA	ANG	209
A 1 –	KUGELGEWINDETRIEB	210
A 2 –	THK LAGER	214
A 3 –	FEDERSTEGKUPPLUNG2	218
A 4 –	GETRIEBE	221
B 1 –	ANTRIEBSMOTOR	222
B 2 –	IMPULSGEBER2	223
B 3 –	НМІ2	226
B 4 –	NETZTEIL	236
С – Е	LEKTRONIK	238
D 1 –	STÜCKLISTE BASISPLATINE	239
D 2 –	STÜCKLISTE ADAPTERPLATINE	240
D 3 –	STÜCKLISTE SONSTIGES	241
E – N	OTIZEN	242
F – D	OKUMENTATION AUF CDROM	243

## 2 Sinn und Zweck

## 2.1 Warum werden Akustikmessungen durchgeführt?

Um das akustisch hochwertige Ergebnis eines Bose Sound Systems zu erhalten, sind viele Entwicklungsschritte notwendig. Das Entwickeln eines Sound Systems für ein Kraftfahrzeug stellt die Bose Automotive Systems Division immer wieder vor neue Herausforderungen.

Das Klangbild in einem Fahrzeug wird von vielen Faktoren beeinflusst. An erster Stelle sind natürlich die Lautsprecher und deren Einbauort zu nennen. Jeder Fahrzeuginnenraum "klingt" anders, das Interieur beeinflusst den Klang maßgeblich: Leder- oder Stoffausstattung, Form der Kopfstützen, Art der Seitenverkleidungen, Armaturen und die Neigungswinkel der Scheiben sind Faktoren, welche die Akustik im Fahrzeug beeinflussen.

Um das gewünschte Klangbild eines Sound Systems in einem Kraftfahrzeug zu erhalten, muss das Sound System an den Fahrzeugtyp angepasst werden. Hierzu sind zuerst die akustischen Vorrausetzungen im Fahrzeuginneren festzustellen. Dies erreicht man durch Messen der Frequenzgänge aller Lautsprecher im Fahrzeuginnenraum.

Die aus diesen Messungen gewonnenen Daten bilden die Grundlage für die Anpassung, das "tunen", des Sound Systems.

Um möglichst allen Fahrzeuginsassen besten Klang zu Bieten, müssen die Akustikmessungen nicht nur auf den vorderen Sitzplätzen, sondern auch auf dem linken und rechten Sitzplatz im Fond durchgeführt werden.

### 2.1.1 Was ist "tunen"?

Zum "tunen" eines Sound Systems gibt es verschiedene Möglichkeiten.

- Filter setzen
- Phasengänge korrigieren
- Delays einstellen

Durch das Messen der Frequenzgänge und das Auswerten der gewonnenen Daten kann man feststellen, dass ein Lautsprecher einen nichtlinearen Frequenzgang hat. Diese Nichtlinearität gilt es zu korrigieren.

Durch das Einsetzen von Equalizerfiltern versucht man, den Frequenzgang linear zu gestalten.

Durch das Einsetzen von Hoch-, Tief- und Bandpässen wird erreicht, dass ein Lautsprecher nur in diesem Frequenzbereich eingesetzt wird, für den er auch konzipiert wurde.

Über "Allpässe" werden in kritischen Frequenzbereichen Phasengänge geändert.

Als weitere Möglichkeit bleibt, Zeitverzögerungen, sogenannte "delays", einzusetzen. Durch diese Einstellungen werden entsprechende Lautsprecher zeitverzögert angesprochen. Dadurch kann ein räumliches Klangbild hervorgerufen werden.

## 2.2 Warum Automatisierung des Messsystems ?

Die Messungen werden durchgeführt mit Hilfe einer geschlossenen "Kiste", die durch ihre Größe in etwa einen menschlichen Oberkörper darstellt und auf deren oberen Ende zwei Mikrofone installiert sind.

Um verschiedene Sitzhaltungen und Körpergrößen der Passagiere zu berücksichtigen, werden die Messungen nicht nur auf jedem Sitzplatz, sondern auch an fünf verschiedenen Positionen auf dem jeweiligen Sitzplatz durchgeführt.

Ziel der Messungen ist es, die Frequenzgänge aller im Kraftfahrzeug eingebauten Lautsprecher zu erfassen – an jeder Position, an der sich der Kopf eines Fahrzeuginsassen während der Fahrt befinden könnte.

Die verschiedenen Positionen erreichen die Mikrofone durch nach vorne kippen und durch zur Seite neigen der "Kiste". Zusätzlich kann die Höhe der Mikrofone durch Verstellen eines versenkbaren Halses variiert werden.

Diese Positionsänderungen geschahen seither durch manuellen Eingriff des Bedieners der Messeinrichtung. Dieser legte nach eigenem Ermessen die Kipp- und Neigungswinkel durch z.B. Unterlegen von Schaumstoffkeilen fest.

Je nach Bediener und Wiederholungen der Messungen variierten somit die Messpositionen im Fahrzeug, Vergleichsmessungen nach Änderungen am Tuning des Sound Systems konnten nur unzureichend durchgeführt werden.

Zu jeder Positionsänderung mußte der Bediener den Messaufbau manuell bewegen, kontrollieren und der Messsoftware bestätigen, dass die Messung fortgesetzt werden konnte.

Zusätzlich wurden an weiteren Geräten wie Notebook oder Psion, verschiedene Eingaben, wie z. B. Umschaltung der zu messenden Frequenz auf andere Lautsprecher des Systems, verlangt.

Für die Messungen bestanden somit keine nachvollziehbaren Standards, dadurch ergab sich Handlungsbedarf. Durch Automatisierung des Messvorganges wurde dieser optimiert :

- Vergleichsmessungen nach Änderungen am Sound System können zuverlässig durchgeführt werden.
- Durch Automatisierung wird eine selbstständige Positionierung des Messsystems im Fahrzeug möglich. Der Bediener kann sich auf die Überwachung des Systems konzentrieren.
- Durch selbstständige Kommunikation zwischen Messsoftware und Messsystem, wird Fehlbedienung durch den Benutzer ausgeschlossen.
- Durch Automatisierung werden Funktionen von externer Hardware implementiert, d. h., Geräte zur Eingabe von Daten, wie z. B. einem Psion, können eingespart werden.
- Der Messaufbau wird bedienerfreundlicher, während der Messung ist kein Bedienereingriff mehr notwendig.

Ziel der Automatisierung ist es, die Gewinnung der Rohdaten zu standardisieren, Peripherie zu rationalisieren und Arbeitsaufwand zu minimalisieren.

## 3 Der Messaufbau

## 3.1 Messaufbau Blockschaltbild



eingesparte Verbindungen / hardware neue Verbindungen / hardware

(Abb. 3 – 1)

Durch die Automatisierung des Messvorganges und Änderungen in der Art der Kommunikation der einzelnen Komponenten des Messaufbaus untereinander ist es möglich, Hardwarekomponenten einzusparen. Zum Teil wird auch Hardware gegen Software ersetzt.

Wie aus der Übersicht im Blockschaltbild zu erkennen ist, werden Geräte wie der Handheld Psion, der seither zum Umschalten auf den zu messenden Lautsprecher benötigt wurde, eingespart.

Ebenso kann auf den bisherigen Preq, der Umschalter der Lautsprecherkanäle, in Zukunft verzichtet werden. Diese Aufgabe übernimmt der Hasheraccelerator, der bauartbedingt kleiner und leichter ist, als sein Vorgänger. Außerdem lassen sich mit dem Hasheraccelerator höhere Umschaltgeschwindigkeiten der Lautsprecherkanäle erzielen.

Die Messvorrichtung wird zum Messautomat ausgebaut und kommuniziert direkt über eine RS 232 - Schnittstelle mit der Messsoftware, programmiert in Matlab, auf einem Notebook.

Der Ausbau der Messvorrichtung zum Messautomat wird in Kapitel 4 verdeutlicht.

# 4 Übersicht Aufbau Messautomat

## 4.1 Allgemein

Der Messautomat ist in Modulbauweise aufgebaut. Der Aufbau läßt sich weiter gliedern. Eine Übersicht über den Aufbau können Sie nachfolgend entnehmen, genauere Beschreibungen und Detailinformationen sind in Kapitel 5 ersichtlich.

## 4.2 Teil 1 : Mechanik

### 4.2.1 Gehäuse

- Rahmen aus Aluminiumprofilen
- Füllung aus Mehrschichtplatten

### 4.1.2 Antriebseinheiten

- Doppelte Linearführungen
- Kugelrollspindel mit entsprechender Lagerung
- Kupplung zur Kraftübertragung zwischen Antriebsmotor und Getriebe
- Antriebsschlitten mit Gleitlager
- Getriebe
- DC Antriebsmotor
- Inkrementalgeber zur Lageregelung
- Seitliche Schiene als Endschalterhalterung

## 4.3 Teil 2 : Elektrik

### 4.3.1 Hauptplatine

- Netzteil zur Bereitstellung verschiedener Spannungen
- Steuerungsteil
- Leistungsteil
- Kommunikation über serielle Schnittstellen
- Anschluss der Peripherie

### 4.3.2 HMI – Human Machine Interface

- Zum Bedienen des Systems und zur Bedienerführung
- Touchpanel mit beleuchtetem LCD Display

### 4.3.3 externe Anschlüsse der Frontplatte

- Spannungsversorgung, Netzeinspeisung 240 V~
- Mikrofonausgänge
- Serielle Schnittstelle RS 232

## 4.4 Blockschaltbild : neue Komponenten des Messautomaten



(Abb. 4-1)

## 5 Kurzfunktionsbeschreibung

## 5.1 Kurzfunktionsbeschreibung Ablauf Messautomat

Der Messautomat dient zum akustischen Ausmessen eines Fahrzeuginnenraumes. An fünf Messpositionen auf jedem der vier Standartsitzplätze im Kraftfahrtzeug werden Messungen vorgenommen. Der Messautomat soll sich selbstständig zwischen den fünf Messpositionen eines Sitzplatzes bewegen.

Verschiedene Betriebsarten stehen hierzu zur Verfügung :

- Betriebsart Automatik
- Betriebsart Teilautomatik
- Betriebsart Hand

Der Messautomat verfügt über ein beleuchtetes LCD-Touchpanel als HMI (Human Machine Interface). Über die HMI werden Eingaben durch den Benutzer angefordert. Die HMI ist die dominante Schnittstelle zur Bedienung des Messautomaten. Die umfangreichsten Eingaben können über diese Schnittstelle getätigt werden.

Die HMI dient selbstverständlich auch zur Bedienerführung. Meldungen werden hier ausgegeben, genauso wie Störungen dargestellt.

Als weiteres Werkzeug zur Bedienung des Messautomaten steht die Applikation OLLICONTROL (OC) zur Verfügung. Genauso wie die HMI kommuniziert OC über eine serielle Schnittstelle mit der Steuerung.

Mit OC kann im Servicefall das Matlab-Hasherprogramm simuliert werden. Der Messautomat kann also autark von einem Messaufbau wie im Blockschaltbild in Kapitel 3.1 (Abb. 3-1) dargestellt, betrieben werden.

OC wird über die Steuerung in Abhängigkeit der HMI zur Bedienung des Automaten freigeschalten bzw. gesperrt.

Befehle, die über das Hasherprogramm aus Matlab, OC, oder die HMI an die Steuerung gesendet werden, werden dort weiterverarbeitet. In der Steuerung werden Reaktionen auf die Befehle ausgelöst.

Entsprechend den Befehlen werden logische Ausgänge der Steuerung gesetzt bzw. rückgesetzt oder Befehle über die seriellen Schnittstellen an entsprechende Adressen weitergeleitet.

Nach dem Verbinden des Automaten mit der Netzeinspeisung führt dieser jedes Mal einen Selbsttest aus.

Durch automatische Freigabe der Bewegungen nach einem erfolgreichen Selbsttest wird eine Referenzpunktfahrt ausgelöst. Bei dieser Referenzpunktfahrt werden die Antriebseinheiten auf ihre mechanischen Bezugspunkte gefahren und mit der Steuerung abgeglichen.

Nach der Referenzpunktfahrt ist die Betriebsart Automatik vorgewählt. Ohne zusätzlichen Bedienungsaufwand kann nun mit der Messung durch das Hasherprogramm begonnen werden.

Durch Fahrbefehle werden Positionsänderungen des Messautomaten ausgelöst. Entsprechend dieser Befehle werden die Treiberstufen angesteuert. Die Treiberstufen stellen die Leistung für die Antriebsmotoren zur Verfügung, welche nun in Verbindung mit den Antriebseinheiten eine Positionsänderung durchführen.

Die Achsen verfahren nicht interpolierend, da kein NC-technisches Bahnfahren zur Positionierung des Messautomaten notwendig ist.

Entsprechende Kommunikation zwischen Mechanik und Steuerung sorgt für Prozesssicherheit.

Detailinformationen über die Antriebseinheiten bietet Ihnen Kapitel 6.1.2.

## 6 Detail Aufbau Messautomat

## 6.1 Teil 1 : Mechanik

### 6.1.1 Gehäuse

Da der Messautomat nicht nur für den stationären, sondern auch für den mobilen Einsatz gedacht ist, ist eine außerordentlich stabile und gegen Umwelteinflüsse robuste Bauweise nicht nur für das Gehäuse, sondern für alle Komponenten des Messautomaten, erforderlich.

Das Gehäuse ist eine Rahmenkonstruktion aus Aluminiumprofilen, welche an den Knotenpunkten mit Würfelprofilen verschraubt sind. Diese Rahmenkonstruktion bringt ein Höchstmaß an Stabilität mit sich.



(Abb. 6-1)

Als Füllung wurden Mehrschichtplatten verwendet, die eine hohe Verwindungssteifigkeit des Körpers garantieren. Diese Mehrschichtplatten wurden mit der Oberfräse bearbeitet, um sie in die Rahmenprofile einlassen zu können.

Die eigentlichen Seitenverkleidungen sind auf die eingelassenen Mehrschichtplatten geschraubt. Dadurch wurde im Inneren des Messautomaten deutlich an Raum gewonnen.

Tragegriffe sind im Gehäuse integriert.

Der Deckel, der die Mikrofone vor Beschädigungen beim Transport des Messautomaten schützt, ist verdrehsicher mit Schnellspannverschlüssen montiert. Dieser beinhaltet auch ein separates Fach, um Anschlussleitungen für die Netzversorgung und eine Datenleitung sicher aufbewahren zu können.

### 6.1.2 Antriebseinheiten

Um den Messautomat in seinem Arbeitsumfeld bewegen zu können, sind vier Linearachsen notwendig. Prinzipiell sind alle vier Achsen Z-Achsen, also hängende Achsen, da sie sich im Koordinatensystem des Messautomaten senkrecht bewegen und somit eine Auf- / Abbewegung auslösen.

Anstatt die Achsen somit mit Z1 – Z4 zu bezeichnen, wurden sie einfach durchnummeriert, Achse 1 bis Achse 4.

Eine Antriebseinheit besteht aus zwei parallel laufenden Präzisionsstahlwellen mit je 8 mm Durchmesser. Diese Wellen werden in gefrästen massiven Aluminiumklemmblöcken fixiert.

Der Schlitten wird durch eingepresste Linearkugellager auf den Präzisionsstahlwellen geführt.

Den Vortrieb übernimmt eine Kugelrollspindel, deren Spindelmutter in den Schlitten eingelassen ist.

Die Kugelrollspindel wird motorseitig von einer Festlagereinheit in Flanschausführung, auf der anderen Seite von einer Loslagereinheit, ebenso in Flanschausführung, gelagert.

Präzisions-Kugelgewindetrieb BNK 0801-3 Durchmesser: 8 mm; Steigung: 1 mm

(Abb. 6-2)

Achse 1 bis Achse 3 bestehen aus identischen Kugelrollspindeln mit 145 mm Länge. Die Antriebseinheit der Achse 4 erfordert einen größeren Arbeitsbereich und wird deshalb mit einer Kugelrollspindel der Länge von 175 mm ausgerüstet.

		Länge Gewindespindel					
Baugroße haa	max. Hub						
BNK0801-3G0+115LC3V	-	4	L <sub>2</sub>	L <sub>3</sub>			
BNK0801-3G0+115LC5Y	40	66	76	115			
BNK0801-3G2+115LC7Y							
BNK0801-3G0+145LC3Y		96	106				
BNK0801-3G0+145LC5Y	70			145			
BNK0801-3G2+145LC7Y							
BNK0801-3G0+175LC3Y			136 186				
BNK0801-3G0+175LC5Y	100	126		175			
BNK0801-3G2+175LC7Y							
BNK0801-3G0+225LC3Y							
BNK0801-3G0+225LC5Y	150	176		225			
BNK0801-3G2+225LC7Y							

(Abb. 6-3)

Die Präzisionsstahlwellen stehen über die Festlagereinheit hinaus und dienen somit als Befestigung für die Motorhalterung. Die Motorhalterung wird im Gegensatz zu der Los- und Festlagereinheit nicht mit der Seitenwand verschraubt, um eventuelle Vibrationen im Antriebsstrang nicht direkt auf das Gehäuse zu übertragen.





(Abb. 6-4,5)

Das Drehmoment des DC – Antriebsmotor mit Edelmetallkommutierung wird über ein hochwertiges Planetengetriebe in Stahlausführung mit Hilfe einer drehsteifen Federstegkupplung aus Aluminium an die Kugelrollspindel übertragen.

Mit der Festlagereinheit auf der einen Seite und der Loslagereinheit auf der anderen Seite ist eine mehrfach gefräste Aluminiumschiene verschraubt. Über die gesamte Länge dieser Schiene sind zwei Langlöcher eingearbeitet, die eine verstellbare Befestigungsmöglichkeit für

zwei Grenztaster als Endlagenüberwachung und einem Grenztaster als Referenzpunktschalter bietet.

Nach Möglichkeit wurden sämtliche Schrauben in Innensechskant-Ausführung gewählt, um hohe Anzugsmomente zu erreichen und um den Werkzeugeinsatz im Servicefall so gering wie möglich zu halten.



(Abb. 6-6)

Die Antriebseinheiten sind mit den jeweiligen Seitenverkleidungen verschraubt und können somit nach dem Lösen der elektrischen Verbindungen, die zu diesem Zweck alle steckbar sind, mit der Seitenverkleidung demontiert werden. So bildet jede Achse in Verbindung mit der entsprechenden Seitenwand ein Modul, das zu Servicezwecken ausgebaut, neu justiert oder auch nur überprüft werden kann.

Weitere Detailinformationen können Sie den Skizzen (Kapitel 6.1.5.1) entnehmen.

### 6.1.3 Dimensionierung der mechanischen Komponenten

### 6.1.3.1 Dimensionierung des Gehäuses

Die äußerlichen baulichen Abmaße änderten sich zu der Messvorrichtung nur geringfügig, um keine Nachteile in der Mobilität des Systems zu erlangen.

Die Rahmenprofile wurden im Querschnitt 30 x 30 mm gewählt, um den höchsten Gewichts- / Stabilitätsfaktor zu erreichen.

Die Mehrschichtplatten haben eine Stärke von 10 mm, dies ist mehr als ausreichend, um den Rahmen zu stabilisieren und eine außerordentliche Grundlage, um die Antriebseinheiten präzise justieren zu können.

### 6.1.3.2 Dimensionierung der Antriebseinheit

Die Kugelrollspindeln wurden mit einer Steigung von 1 mm/Umdrehung ausgewählt. Die geplante Schlittenverfahrgeschwindigkeit beträgt 5 mm/s. Die abgangsseitige Drehzahl des Getriebes muß somit 300 Umdrehungen/min betragen.

Bei einem Motor mit einer Leerlaufdrehzahl von 7800 Umdr./min ergibt dies eine erforderliche Getriebeuntersetzung von 26:1. Ausgewählt wurde eine Getriebeübersetzung von 23:1, da ein Getriebe mit der berechneten Untersetzung nicht im Produktkatalog der Fa. Faulhaber zu finden ist.

Auf einen anderen Hersteller wird aus Rücksicht auf das Prinzip der Modulbauweise des Messautomaten verzichtet.

Die erforderliche Kraft ergibt sich aus diversen Gewichten und Anstellwinkeln, in denen Massen bewegt werden, mit 98 Newton. Daraus läßt sich unter Berücksichtigung von Reibungsverlusten und Wirkungsgraden ein an der Spindel erforderliches Drehmoment von 21,1 mNm berechnen.

Durch die Getriebeuntersetzung von 23:1 und einem Wirkungsgrad des Getriebes von 0,75 berechnet sich das erforderliche Drehmoment des Antriebsmotor auf 1,22 mNm.

Hier wurde eine vierfache Sicherheit eingerechnet und ein DC-Motor mit Edelmetallkommutierung mit einem Nenndrehmoment von 5 mNm gewählt.

Das kugelgelagerte Planetengetriebe ist mit 0,5 N belastbar. Der Gehäusewerkstoff des Planetengetriebes ist Stahl, der Zahnräderwerkstoff ist Metall.

Eine Federstegkupplung aus Aluminium bildet die Verbindung zwischen Planetengetriebe und Kugelrollspindel.

Die Berechnung der Komponenten erfolgte auf Dimensionierungsmethoden der Fa. Oriental Motor, Fa. SKF und Fa. THK, welche auf die Verhältnisse am Messautomaten angepaßt wurden.

Umdrehungen / Motor	:	7800
abgangsseitige Umdrehungszahl des Getriebes	:	$N_{G}$
Getriebeuntersetzung	:	i
Gesamtbelastung der Kugelrollspindel	:	F
erforderliches Drehmoment an der Spindel	:	$T_L$
erforderliches Drehmoment am Motor	:	T <sub>M</sub>
externe Kräfte	:	FA
Masse des Schlittens und der zu tragenden Last	:	$m_1$
Koeffizient Reibungsverluste an der Spindel	:	μ
Wirkungsgrad des Getriebes	:	ηG

Berechnung der abgangsseitigen Umdrehungszahl des Getriebes :

$$NG := \frac{V \cdot 60}{PB}$$
$$NG := 300 \frac{U}{min}$$

Berechnung der Getriebeuntersetzung :

$$i := \frac{7800 \frac{U}{min}}{NG}$$

i = 26, gewählt wurde die Getriebeuntersetzung 23:1

Berechnung der Gesamtbelastung der Kugelrollspindel :

$$F := FA + m1 \cdot g \cdot (sin(\alpha) + \mu \cdot cos(\alpha))$$
$$F = 98 N$$

Berechnung des erforderlichen Drehmomentes an der Spindel :

$$\mathsf{TL} := \frac{98\mathsf{N} \cdot 1\mathsf{mm}}{2\pi \cdot 0.95} \quad \bullet + \frac{0.3 \cdot 98\mathsf{N} \cdot 1\mathsf{mm}}{2\pi}$$

TL = 21.1 mNm

Berechnung des erforderlichen Drehmomentes am Motor :

$$TM := \frac{21.1mNm}{23 \cdot \eta G}$$
$$TM = 1.22 \text{ mNm}$$

Um bei der Antriebseinheit ohne zusätzliche mechanische Bremse oder Klemmung auszukommen, wird auf Verfahrgeschwindigkeit verzichtet und deshalb ein Getriebe mit hoher Untersetzung gewählt.

Die Spindelsteigung wird ebenso extrem flach gehalten, um der Notwendigkeit eines drehmomentstärkeren Antriebmotors zu entgehen.

Das Getriebe in Verbindung mit der flachen Spindelsteigung sorgt für genügend Widerstand, um genaues Positionieren zu ermöglichen und ein Absacken der Einheit im Stillstand zu verhindern.

### 6.1.4 Sonstige mechanische Komponenten

### 6.1.4.1 Mikrofonhalterung

Die Achse 4 bildet die Einheit um die Mikrofone in ihrer Höhe zu verstellen. Diese Achse hat auch den längsten Arbeitsbereich. Auf den Schlitten wird nicht wie bei den Achsen 1 bis 3 ein stabiler, aus Vollmaterial gefräster Aluminiumwinkel justiert, sondern eine Präzisionsstahlwelle.

Die Welle wird aus Gründen der Stabilität und um Verwindungen im Lagerbereich des Schlittens zu vermeiden, durch ein separates Gleitlager geführt.

Die Antriebseinheit 4 wird durch diese Maßnahme nicht den Hebelkräften, die eventuell durch unsachgemäße Handhabung des Messautomaten entstehen könnten, ausgesetzt.

Die ebenso gefräste Mikrofonhalterung wird stirnseitig mit der Präzisionsstahlwelle verschraubt.

Weitere Informationen können Sie Kapitel 6.1.5.2 entnehmen.

### 6.1.5 Technische Zeichnungen

### 6.1.5.1 Technische Zeichnungen Antriebseinheit

Im Folgenden die technischen Zeichnungen zu den Antriebseinheiten. Die Zeichnungen sind hier nicht maßstabsgetreu dargestellt und dienen nur der Übersicht.

Die Zeichnungen werden auf der beiliegenden Dokumentations – CDR im CAD – Dateiformat \*.mcd, \*.dxf und \*.dwg für beispielsweise VectorWorks zur Verfügung gestellt.

Selbstverständlich können die Grafikdateien auch von anderen CAD - Zeichenprogrammen importiert werden.

Um die Grafikdateien hier darzustellen zu können, wurden sie mit Adobe Illustrator bzw. Corel Draw bearbeitet.



#### Übersicht Antriebseinheit





(Abb. 6-8)







(Abb. 6-10)







(Abb. 6-12)



## 6.1.5.2 Technische Zeichnung Mikrofonhalterung





## 6.2 Teil 2 : Elektrik

### 6.2.1 Spannungsversorgung

### 6.2.1.1 Spannungsversorgung Messautomat



(Abb. 6-15)

Die Spannungsversorgung erfolgt über die Adapterplatte im Gehäuse des Messautomaten (Abb. 6-15). Die Netzeinspeisung von wahlweise 240 V~ / 50 Hz oder 110 V~ / 60 Hz ist steckbar ausgeführt.



Bei Betrieb an 110 V~ Netzen muss an internen Netzteilen ebenso eine Anpassung durch Auswahl der entsprechenden Versorgungsspannung vorgenommen werden.

Die Anschlussleitungen sind in vertauschsicherer Ausführung gestaltet. Die Netzanschlussbuchse ist an der Adapterplatte mit der Gravur 110-230V~ gekennzeichnet.

### 6.2.1.2 Spannungsversorgung Hauptplatine

Die Spannungsversorgung der Hauptplatine erfolgt über ein Netzteil, das über die Adapterplatte in der Front des Gehäuses an die Netzversorgung angeschlossen ist. Der Anschluß an die Hauptplatine erfolgt mittels eines verpolungssicheren Steckers. Die Versorgungsspannung der Hauptplatine beträgt 12 - 15 V=.

Die Anschlussbezeichnungen können dem Stromlaufplan entnommen werden.

### 6.2.1.3 Spannungsversorgung des Netzteils der Mikrofone

Die Spannungsversorgung des Netzteils der Mikrofone erfolgt ebenso über die Adapterplatte in der Front des Gehäuses. Die Verteilung der Spannungsversorgungen und deren Absicherungen erfolgen intern.

Detailinformationen können dem Stromlaufplan entnommen werden.

### 6.2.1.4 Spannungsversorgung der HMI

Die Spannungsversorgung der HMI erfolgt über das gleiche Netzteil, wie die Spannungsversorgung der Hauptplatine. Das Netzteil ist über die Adapterplatte in der Front des Gehäuses mit der Netzeinspeisung verbunden. Die Verteilung der Spannungsversorgungen und deren Absicherungen erfolgen intern.

Der Anschluß an die HMI erfolgt über die Schraubklemmen J2. Die Versorgungsspannung der HMI an J2 darf zwischen 9 und 35 V= betragen.



Eine Verpolung der Versorgungsspannung der HMI an den Schraubklemmen J2, kann zu sofortiger Zerstörung der HMI führen.

Detailinformationen können dem Stromlaufplan entnommen werden.

### 6.2.2 Hauptplatine

Die Hauptplatine wird aus zwei Teilen zusammengeführt :

- der Basisplatine
- der Prozessorplatine

Die Prozessorplatine wird dabei über die Steckleisten X1A, X1B und X1C auf der Basisplatine fixiert. Auf genaues Einsetzen der Prozessorplatine muß dringend geachtet werden. Das Einsetzen der Platine darf nicht unter Spannung erfolgen.

### 6.2.2.1 Teil 1 : Basisplatine



(Abb. 6-16)

Die Basisplatine (Abb. 6-16) ist in mehrere Bereiche unterteilt:

- Spannungsversorgung
- Logikteil
- Leistungsteil
- Anschlussmöglichkeiten zur Peripherie
- serielle Schnittstellen
- Debugbereich
- reservierter Bereich f
  ür die Prozessorplatine

Sämtliche Komponenten im Layout der Platine wurden in SMD-Ausführung gehalten, dadurch wurde ermöglicht, eine kompakte Einheit, die Steuerung, Logik- und Leistungsteil vereint, zu generieren.

Detailinformationen zur Basisplatine erhalten Sie in Kapitel 7.

### 6.2.2.2 Teil 2 : Prozessorplatine

Prozessorplatine Oberseite



(Abb. 6-17)

Prozessorplatine Unterseite



(Abb. 6-18)

Die Prozessorplatine ist ein miniModul-509 der Firma Phytec. Zu beziehen ist dieses Modul über die Phytec Messtechnik GmbH.

Das Modul ist ESD-empfindlich und darf nur an ESD geschützten Arbeitsplätzen von geschultem Fachpersonal gehandhabt werden.

Das miniModul-509 repräsentiert einen funktionellen, universellen Single Board Computer (SBC) im Scheckkartenformat. Die Größe der Prozessorplatine beschränkt sich auf 85 x 55 mm.

Der SBC basiert auf einem Mikrocontroller C509 von Infineon. Der C509 Controller kann mit maximal 16 Mhz getaktet werden und erreicht damit die Rechenleistung eines mit 32 Mhz getakteten 8032. Daraus ergibt sich eine Zykluszeit von 375 ns.

Die controllerspezifischen Eigenschaften entnehmen Sie bitte der Dokumentations-CDR.

Alle Verbindungen zur Basisplatine werden über Stift-/Buchsenleisten durch einfaches Aufstecken der Prozessorplatine auf die Basisplatine hergestellt. Dieser Vorgang wurde bereits in Kapitel 6.2.2 beschrieben.



Mit der Prozessorplatine wird die eigentliche Steuerung des Messautomaten verwirklicht.

Über die im Standartmaß von 2.54 mm ausgeführten Stift-/Buchsenleisten werden über die bereits in der Basisplatine anstehenden und aufbereiteten Signale, Informationen eingelesen.

Diese Informationen werden auf der Prozessorplatine ausgewertet, weiterverarbeitet und entsprechende Signale über die Stift-/Buchsenleiste an die Basisplatine zurückgegeben.

### Die Lage der Pins an den Stiftleisten kann der Abbildung 6-19 entnommen werden.



(Abb. 6-19)

Diese Ausgangssignale werden über die Schnittstellen oder über die Anschlüsse X4 bis X10 der Basisplatine, die in Kapitel 7 näher beschrieben werden, an die Peripherie ausgegeben.

Nachfolgend zusammengefasst, die wichtigsten Eigenschaften und Features der Prozessorplatine miniModul-509 :

- SBC im Format 85 x 55 mm durch SMD-Technik
- Störsicher durch Multilayerplatine
- Infineon Controller C509 im QFP100-Gehäuse, befehlskompatibel zur INTEL 8051-Prozessor Familie.
- Versorgungsspannung 5V über Netzteil auf der Basisplatine
- Ports, Daten- und Adressleitungen über Stiftleisten abgreifbar
- Die Speicher-Standartkonfiguration des miniModuls bietet 32 kByte externen SRAM und 128 kByte externen EEPROM als Speicher f
  ür Anwenderquellcode.
- Drei freie chip-select Signale sind verfügbar, um externe Ein-/Ausgänge anzubinden bzw. nachzurüsten.
- Wie bereits aus vorangestellten Kapiteln ersichtlich, verfügt das miniModul-509 über zwei RS232–Schnittstellen, die zur Kommunikation mit der HMI bzw. zur Kommunikation mit dem Hasherprogramm, wahlweise OLLICONTROL (OC) genutzt werden.
- Das Modul ist ausgelegt f
  ür den Industrie-Standarttemperaturbereich von 0 bis 70° C.

Weitere Informationen über die Prozessorplatine miniModul-509 können Sie der Phytec-Dokumentation oder der beigelegten Dokumentations-CDR entnehmen.

In Kapitel 8 erhalten Sie spezifische Detailinformationen der Prozessorplatine in Verbindung mit dem Messautomat.

### 6.2.3 Spindelantrieb

Der Spindelantrieb besteht aus mehreren Komponenten:

- DC-Motor
- Inkrementalgeber
- Getriebe
- Kupplung

In den folgenden Kapiteln wird erhalten Sie weitere Informationen zu den einzelnen Komponenten des Antriebes der Spindel.

### 6.2.3.1 DC-Motor

Der Antriebsmotor des Spindeltriebes ist ein bürstenloser DC-Motor mit Edelmetallkommutierung der Fa. Faulhaber des Typs 2224 – 0012 SR.

Die wichtigsten Daten können der folgenden Tabelle entnommen werden.

S	erie 2224 SR								
		2224 U	003 SR	006 SR	012 SR	018 SR	024 SR	036 SR	
1	Nennspannung	UN	3	6	12	18	24	36	Volt
2	Anschlusswiderstand	R	0,56	1,94	8,71	17,50	36,30	91,40	Ω
3	Abgabeleistung	P2 max.	3,92	4,55	4,05	4,54	3,88	3,46	W
- 4	Wirkungsgrad	η max.	80	82	82	82	81	80	%
- 5	Leerlaufdrehzahl	no	8 100	8 200	7 800	8 100	7 800	7 800	rpm
6	Leerlaufstrom (bei Wellen ø 2,0 mm)	lo	0,066	0,029	0,014	0,010	0,007	0,005	A
7	Anhaltemoment	MH	18,5	21,2	19,8	21,4	19,0	16,9	mNm
8	Reibungsdrehmoment	MR	0,23	0,2	0,2	0,21	0,2	0,22	mNm
9	Drehzahlkonstante	kn	2 730	1 380	657	454	328	219	rpm/V
10	Generator-Spannungskonstante	ke	0,366	0,725	1,520	2,200	3,040	4,560	mV/rpm
11	Drehmomentkonstante	k <sub>M</sub>	3,49	6,92	14,50	21,00	29,10	43,50	mNm/A
12	Stromkonstante	kı	0,286	0,144	0,069	0,048	0,034	0,023	A/mNm
1.7			120	2.07	201			165	
13	Steigung der n-M-Kennlinie	Δn/ΔIVI	438	387	394	379	411	462	rpm/mNm
14	Anschlussinduktivität	L	11	45	200	450	800	1 800	μН
15	Mechanische Anlaufzeitkonstante	τπ	11	11	11	11	11	11	ms
16	Rotortragheitsmoment	J	2,4	2,7	2,7	2,8	2,6	2,3	gcm <sup>2</sup>
17	Winkelbeschleunigung	α max.	11	78	74	11	74	/4	·10 <sup>-</sup> rad/s <sup>*</sup>
19	Wärmewiderstände	Date a / Date of	E / 20						KAM
10	The main here is a main the material to a main the main term is a main term term term term term term term term	Tun 1 / Tun 2	5720						N/ W
20	Petriebstemperaturbareich:	τw1/τw2	6,67440						s
20	- Motor		20 .	QE /Sandara	usführung	EE . 12E)			°C
	- Rotor max zulässig		- 50 +	os (sondera	usrunrung –	55 + 125 <i>)</i>			°C
	- Notor, max. Zulassig		+1	25					C
21	Wellenlagerung		Sinterlage	ar	Kugellage	ər	Kugellage	r vorgespannt	
22	Wellenbelastung, max, zulässig:		(standard	)	(Sondera)	usführung)	(Sondera)	usführung)	
	– für Wellendurchmesser		2.0	,	2.0	asramang,	2.0	asramang,	mm
	– radial bei 3000 rpm (3 mm vom Lager)		15		8		8		N
	- axial bei 3000 rpm		0.2		0.8		0.8		N
	- axial im Stillstand		20		10		10		N
23	Wellenspiel:		2.0		10		10		
	- radial	<	0.03		0.015		0.015		mm
	- axial	<	0.2		0.2		0		mm
		-	0,2		0,2		, v		
24	Gehäusematerial		Stahl, sch	warz beschi	chtet				
25	Gewicht		46						a
26	Drehrichtung		rechtsdre	hend auf Ak	otriebswelle (	gesehen			-
Em	pfohlene Werte								
27	Drehzahl bis	he max.	8 00 0	8 000	8 000	8 000	8 000	8 0 0 0	rpm
28	Dauerdrehmoment bis	Me max.	5	5	5	5	5	5	mNm
29	Thermisch zulässiger Dauerstrom	le max.	2,200	1,200	0,570	0,400	0,280	0,180	А
(At	b. 6-20)								

Weitere Informationen zur Dimensionierung des Antriebsmotors entnehmen Sie dem Kapitel 6.1.3.2.
CAD-Zeichnungen und weitere Skizzen zu der kompletten Antriebseinheit finden Sie im Kapitel 6.1.5.1.

Auf den technischen Zeichnungen in Kapitel 6.1.5.1 ist der Antriebsmotor nicht in allen Ansichten dargestellt, deshalb können Sie Details zur Zeichnung diesem Kapitel in Abbildung 6-21 entnehmen.



(Abb. 6-21)

#### 6.2.3.2 Inkrementalgeber

Der Inkrementalgeber ist ein magnetischer Impulsgeber des Typs IE2 – 16 der Fa. Faulhaber. Der Inkrementalgeber liefert 16 Impulse pro Umdrehung und besitzt 2 Ausgänge.

Der Geber wäre somit für eine Erkennung der Drehrichtung geeignet, dies ist aber für diese Anwendung nicht notwendig.

Der Geber ist kombinierbar mit dem DC-Motor des Typs 2224 – 012 SR der Fa. Faulhaber und wird auch in Kombination mit diesem geliefert.

In Kombination mit dem Antriebsmotor trägt der Impulsgeber gerade 1,4 mm am hinteren Lagerschild des Motors auf und ist somit wegen seiner kurzen Bauform ideal für diese Anwendung.

Einige technische Daten können Sie der folgenden Tabelle entnehmen.

Serie IEZ – 16			
		IE2 – 16	
Impulse pro Umdrehung	N	16	
Ausgangssignal, rechteckig		2	Ausgänge
Betriebsspannung	V DD	4 18	V DČ
Nennstromaufnahme, Mittelwert (V DD = 12 V DC)	I DD	typ. 6, max. 12	mA
Ausgangsstrom, max. zulässig	Lour	15	mA
Pulsbreite <sup>2)</sup>	P	180 ± 45	°e
Signal-Phasenverschiebung, Kanal A zu B 2)	Φ	90 ± 45	°e
Signal-Anstiegs-/Abfallzeit, max.(CLOAD = 100 pF)	tr/tf	2,5/0,3	μs
Frequenzbereich <sup>1)</sup> , bis	f	7	kHz
Trägheitsmoment der Impulsscheibe	J	0,11	gcm <sup>2</sup>
Betriebstemperaturbereich		- 25 +85	°С
		1	1

<sup>1)</sup> Drehzahl (rpm) = f (Hz) x 60/N <sup>2)</sup> bei 2 kHz geprüft

(Abb. 6-22)

Durch die Verwendung von Hallsensoren und einem mehrteiligen Magnetring ergeben sich zwei um 90° phasenverschobene Kanäle.

Die Versorgungsspannung für den Impulsgeber und den DC-Kleinstmotor sowie die Ausgangssignale werden über ein Flachbandkabel mit Stecker angeschlossen.

Der nachfolgenden Abbildung können Informationen zu den Ausgangssignalen, dem Schaltdiagramm und der Steckerbelegung (im Original) entnommen werden.



(Abb. 6-23)

Die Steckerbelegung zeigt den Auslieferungszustand der Fa. Faulhaber. Die Steckerbelegung wurde aus Servicegründen der allgemeinen Linie im System angepasst.

Die Pinbelegung ist dem Stromlaufplan zu entnehmen.

#### 6.2.3.3 Getriebe

Als Getriebe wurde ein Planetengetriebe der Serie 20/1 der Fa. Faulhaber gewählt. Das Übersetzungsverhältnis ist 23:1.

Weitere Informationen zur Dimensionierung des Getriebes erhalten Sie in Kapitel 6.1.3.2. Dort wurden die Berechnungen zu den erforderlichen Drehmomenten dokumentiert.

Aus der folgenden Tabelle können einige technische Daten entnommen werden.

				20/1			
Gehäusewerkstoff				Stahl			
Zahnräderwerkstoff				Metall			
Max. empfohlene Eingangso	drehzahl fü	ür:					
– Dauerbetrieb				5000 rpm			
Getriebespiel, unbelastet				≤ 1°			
Abtriebswellenlager				Kugellage	r, vorgespann	t	
Maximal zulässige Wellenbe	lastung:						
- radial (8,5 mm vom Befesti	igungsflar	isch)		≤75 N			
– axial				≤ 20 N			
Maximale Aufpresskraft				≤ 35 N			
Lagerspiel (gemessen am La	ger):						
– radial	-			≤ 0,02 mm	1		
– axial				= 0 mm			
Betriebstemperaturbereich				- 30 + 1	00 °C		
Technische Daten							
				Drehi	moment		
Untersetzungs-	Gewicht	Länge	Länge mit Motor	Dauer-	Kurzzeit-	Drehsinn	Wirkungs-

Untersetzungs- verhältnis	Gewicht ohne	Länge ohne		Län	ge mit Mo	otor		Dauer- betrieb	Kurzzeit- betrieb	Drehsinn der Welle	Wirkungs- grad
(nominal)	Motor	Motor	1727 U	2036 U	2224 U	2230 U	2233 U				
		L2	L1	L1	L1	L1	L1	M max.	M max.		
	g	mm	mm	mm	mm	mm	mm	mNm	mNm		%
3,71:1	28	18,35	45,55	54,35	42,55	48,35	50,95	500	700	=	88
9,7 :1	38	23,45	50,65	59,45	47,65	53,45	56,05	500	700	=	80
14 :1	38	23,45	50,65	59,45	47,65	53,45	56,05	500	700	=	80
23 :1	38	23,45	50,65	59,45	47,65	53,45	56,05	500	700	=	80
43 :1	48	28,55	55,75	64,55	52,75	58,55	61,15	500	700	=	70
66 :1	48	28,55	55,75	64,55	52,75	58,55	61,15	500	700	=	70
86 :1	48	28,55	55,75	64,55	52,75	58,55	61,15	500	700	=	70
134 :1	58	33,65	60,85	69,65	57,85	63,65	66,25	500	700	=	60
159 :1	58	33,65	60,85	69,65	57,85	63,65	66,25	500	700	=	60
246 :1	58	33,65	60,85	69,65	57,85	63,65	66,25	500	700	=	60
415 :1	68	38,75	65,95	74,75	62,95	68,75	71,35	500	700	=	55
592 :1	68	38,75	65,95	74,75	62,95	68,75	71,35	500	700	=	55
989 :1	68	38,75	65,95	74,75	62,95	68,75	71,35	500	700	=	55

(Abb. 6-24)

Das Getriebe wurde zusammen mit der Kupplung und den restlichen Komponenten in dieses Kapitel aufgenommen, da diese Komponenten zusammen eine Einheit bilden.

Der Abbildung 6-25 kann die Bemaßung der Motor-/Getriebeeinheit entnommen werden. Zur Ergänzung können die technischen Zeichnungen aus Kapitel 6.1.5.1 hinzugenommen werden.



# 6.2.3.4 Kupplung

Als Kupplung wird eine torsionssteife Flexbeam - Aluminiumkupplung der Fa. Ruland verwendet. Hierbei handelt es sich um eine Federsteg - Aluminiumkupplung.



(Abb. 6-26)

Näher soll hier nicht auf die Kupplung eingegangen werden. Weitere Informationen erhalten Sie auf der Dokumentations-CDR im entsprechenden Datenblatt zur hier abgebildeten Kupplungen.

#### 6.2.4 Sensorik



Als Sensorik werden an die Steuerung angeschlossene Taster, Schalter, Initiatoren, o. Ä. bezeichnet.

In diesem System werden ausschließlich Referenzpunktschalter und Endlagenschalter verwendet.

Zusätzlich wird mit einem Reedkontakt der Deckel überwacht und mit einem Neigungssensor wird der Arbeitsbereich des Messautomaten kontrolliert.

#### 6.2.4.1 Referenzpunktschalter

Der Referenzpunktschalter ist ebenso wie die Endlagenschalter auf der Endschalterungsschiene montiert.

Der Referenzpunktschalter ist als Schließerkontakt ausgeführt und an der Unterseite der Schiene für die Endschalter montiert.

Der Referenzpunktschalter ist auf Abbildung 6-27 nicht zu sehen.

Die Sensorik einer Antriebsachse ist als Modulbaustein zu sehen und deshalb auch zu Servicezwecken komplett zu wechseln.

Die Anschlüsse der Schalter wurden steckbar ausgeführt.



(Abb. 6-27)

Genaue Informationen zur Verdrahtung können dem Schaltplan zur Hauptplatine oder dem Stromlaufplan zum Gesamtsystem entnommen werden.



Eine Referenzpunktfahrt auf den Referenzpunktschalter ist notwendig, um eine eindeutige mechanische Position zu einem Bezugspunkt herzustellen. Die Positionierungen der Achsen werden in Bezug auf diesen Referenzpunkt hergestellt.

Die Referenzpunktschalter sind als mechanische Schalter mit Rollenhebel ausgeführt und nicht als Initiator. Somit kann auf eine Versorgungsspannung, wie sie zum Betrieb eines Initiators benötigt wird, verzichtet werden.

#### 6.2.4.2 Endlagenschalter

Die Endlagenschalter sind baulich die gleichen Schalter wie die Referenzpunktschalter der Antriebseinheiten.

Es wird jedoch der Öffnerkontakt des Wechslers verwendet. Somit wird für Kabelbruchsicherheit garantiert.

Die Endlagenschalter sind in Abbildung 6-27 dargestellt. Es wird jeweils ein Endlagenschalter für jede Drehrichtung eines Antriebsmotors verwendet. Es wird somit die Endlage in Drehrichtung (+) und die Endlage in Drehrichtung (-) kabelbruchsicher überwacht.

Da die Endlagenschalter ebenso wie die Referenzpunktschalter auf der Endschalterschiene montiert sind, sind sie steckbar ausgeführt und als Modulbaustein bei Bedarf zu wechseln.

Durch die Langlochfräsung, sind die Endlagenschalter ebenso wie die Referenzpunktschalter einstellbar.

Genaue Informationen zur Verdrahtung können dem Schaltplan zur Hauptplatine oder dem Stromlaufplan zum Gesamtsystem entnommen werden.

#### 6.2.4.3 Deckelschalter

Der Deckelschalter ist als Reedkontakt ausgeführt. Der Schalter ist im oberen Rahmen des Gehäuses eingelassen und wird durch einen Magneten im Deckel bedämpft.

Um den Deckel auch verdreht aufsetzen zu können, sind zur Sicherheit zwei Magnete gegenüberliegend im Rahmen des Deckels verarbeitet. Ein Aufsetzen des Deckels bleibt nicht unerkannt.

Eine diagnostizierte Störung wird auf der HMI visualisiert. Weitere Informationen zur Bedienerführung werden ausgegeben.

Der Deckelschalter führt die Bezeichnung SE 1.5 im Stromlaufplan und ist entsprechend an Port 1.5 der Steuerung angeschlossen.

#### 6.2.4.4 Neigungssensor

Ein Neigungssensor wird eingesetzt, um zu überwachen, dass der Messautomat sich in seinem Arbeitsbereich befindet.

Der Neigungssensor ist auf dem unteren Boden des Messautomaten installiert. Die Verschaltung und Verdrahtung des Neigungssensors kann dem Stromlaufplan des Systems entnommen werden.



(Abb. 6-28)

Der Neigungssensor führt die Bezeichnung SE 1.3 im Stromlaufplan und ist entsprechend an Port 1.3 der Steuerung angeschlossen.



Einige technische Daten können der folgenden Tabelle entnommen werden.

CONTACT FORM / STYLE		Normally Closed
SWITCHING VOLTAGE	Max. Vac RMS	60
SWITCHING CURRENT	Max. A	0.25
SWITCHING CAPACITY	Max. VA	3
OPERATING ANGLE	Deg.° Max.	$40^{\circ} \pm 15^{\circ}$
CONTACT RESISTANCE	Max. Ω	6
OPERATING TEMPERATURE	Deg. ⁰C	-20° +70°
STORAGE TEMPERATURE	Deg. ⁰C	-25° +70°
CASE MATERIAL		Polypropylene
CABLE / TERMINATION		2 x round 0.14 <sup>2</sup> PVC
		covered and insulated

(Abb. 6-30)

Weitere technische Daten zum Neigungssensor erfahren Sie auf der Dokumentations-CDR im entsprechenden Datenblatt.

#### 6.2.5 HMI – Human Machine Interface

Als HMI wird ein 5,1" LCD-Touchpanel des Typs EA KIT 160-7 der Fa. Electronic Assembly verwendet.



(Abb. 6-31)

Das Touchpanel hat eine Bildschirmauflösung von 160x128 Pixel. Wie in Abbildung 6-31 dargestellt, wird der Bildschirm in 56 Felder aufgeteilt. Diese 56 Felder können als touch-Felder aktiviert werden.

Einige technische Daten:

- Bildschirm mit Hintergrundbeleuchtung
- Bildschirmauflösung 160 x 128 Pixel
- Touchpanelauflösung 8 x 7 Felder
- Versorgungsspannung 12 V=
- Kommunikation über serielle Schnittstelle RS 232

Weitere technische Daten können Sie dem Datenblatt auf der beiliegenden Dokumentations-CDR entnehmen.



Das LCD-Touchpanel dient der Bedienung des Messautomaten. Informationen zur Bedienerführung werden über die HMI visualisiert.

Die Bedieneinheit enthält neben kompletten Grafikroutinen zur Displayausgabe auch verschiedene Schriften.

Die Programmierung erfolgt über hochsprachenähnliche Grafikbefehle. Das Display muß textbasierend programmiert werden. Eine grafisch orientierte Programmierung ist nicht möglich.

Eine Programmierung kann mit jedem ASCII-Editor realisiert werden. Der Quelltext muss compiliert werden und kann dann anschließend über die serielle Schnittstelle RS 232 in ein EEprom des LCD-Touchpanels übertragen werden.

Mit Bildern oder Grafiken wird ebenso verfahren. Die Bilder werden wie der compilierte Quelltext im EEprom des Display gespeichert und mit Hilfe von Makros über die serielle Schnittstelle RS 232 aufgerufen.

Die Bilder können auch je nach Bedarf über die Schnittstelle direkt auf den Bildschirm geladen werden, hiervon wird aber wegen des relativ geringen Datendurchsatzes der Schnittstelle abgeraten.

In diesem System wurden sämtliche Daten im EEprom des Displays abgelegt und nach Bedarf abgerufen.

Ein Upload der Daten vom EEprom des Displays ist nicht möglich.

Die Baudrate der seriellen Schnittstelle RS 232 ist voreingestellt auf 9600 Baud und lässt sich bei Bedarf ändern. Das Datenformat ist fest auf 8, N, 1 eingestellt.

In diesem System ist die Baudrate unverändert, der dem Auslieferungszustand entsprechend, auf 9600 Baud eingestellt.

	E	Bauc	Iraten
DIP	Scha	alter	Datenformat
1	2	3	8,N,1
ON	ON	ON	1200
OFF	ON	ON	2400
ON	OFF	ON	4800
OFF	OFF	ON	9600
ON	ON	OFF	19200
OFF	ON	OFF	38400
ON	OFF	OFF	57600
OFF	OFF	OFF	115200

(Abb. 6-32)

Über DIP – Schalter 6 lässt sich ein versehentliches Überschreiben der Makros, Bilder und Schriften verhindern.

So	chreibschutz
DIP	Schreibschutz
	für
6	EEPROM
ON	Ein
	keine Makroprog. mögl.
	Aus
OFF	Makroprog. möglich

(Abb. 6-33)

Die Versorgungsspannung wird an den Schraubklemmen J2 angeschlossen.



Versorgungsspannung nicht verpolen !

Eine noch so kurzzeitige Verpolung kann zur sofortigen Zerstörung des LCD-Displays führen.

Weitere Informationen zur HMI erhalten Sie in den Kapiteln zur Bedienung des Messautomaten und zur Programmierung des Displays.

# 7 Detail Basisplatine

# 7.1 Spannungsversorgung



(Abb. 7-1)

Der Bereich der Spannungsversorgung befindet sich im oberen linken Teil der Basisplatine. Herzstück der Spannungsversorgung ist ein Festspannungsregler 7805. Dieser ist in seiner Bauform so gewählt, dass er ohne Kühlkörper auskommt. Eine ausreichende Kühlfläche ist auf der Basisplatine so gestaltet, dass Wärme ungehindert abgeführt werden kann.

Der Grundsatz, die Komponenten in ihrer Bauform so zu wählen, dass sie ohne entsprechende Kühlkörper auskommen, wurde im kompletten Projekt verwirklicht.

Eine Glättung bzw. Siebung der Spannung wurde durch entsprechende Beschaltung des 7805 erreicht.

Für den Leistungsteil steht eine Spannung von 12 V= bei 6 A zur Verfügung. Für den Logikteil stehen 5 V= zur Verfügung.

Durch eine LED auf der Basisplatine wird die korrekte Spannungsversorgung des Logikteils angezeigt. Diese liegt an mit VCC gekennzeichneten Punkten.

Peripherie wie Grenztaster, wird über die mit VDD gekennzeichneter Spannung versorgt. Eine entsprechende Absicherung der Spannungen ist über die mit F1 bis F3 gekennzeichneten Sicherungen auf der Basisplatine im Bereich der Spannungsversorgung verwirklicht.

# 7.2 Logikteil



Der Logikteil befindet sich im linken zentralen Bereich der Basisplatine. Der Bereich Logik besteht aus 6 SMD-Bausteinen der Reihe 74LS. Grundsätzlich wurde versucht, durch Verwendung von Bausteinen der Reihe 74LS (Low-Power-Schottky) oder 74ALS (Advanced Low-Power-Schottky), die Belastung des Netzteils so gering wie möglich zu halten.

Durch dieses Konzept kann auch die Wärmeentwicklung eingedämmt werden und somit auf externe Kühlung bzw. Lüftung verzichtet werden.

Die Bausteine des Typs 14M enthalten invertierende Schmitt-Trigger. Diese sind erforderlich, um eindeutige Pegel an den Eingängen (DI-Ports) von der Peripherie wie Endlagenschalter oder Referenzpunktschalter zu erhalten.



Die Eingänge der Referenzpunktschalter wurden zweimal invertiert, da diese Schalter als Schließerkontakt ausgeführt sind. Die Endlagenschalter sind dagegen als Öffnerkontakt ausgeführt, um Kabelbruchsicherheit zu garantieren und werden deshalb nur einmal invertiert.

Der Neigungssensor und die Schließüberwachung des Gehäuses am Deckel sind ebenso über invertierende Schmitt-Trigger an die digitalen Eingänge der Steuerung geführt. Auch hier ist es erforderlich, eindeutige Pegel an den DI-Ports zu erhalten.

Der Logikteil besteht aus weiteren drei Bausteinen der Reihe 74ALS. Diese ODER-Bausteine des Typs 32M werden dazu verwendet, um eventuelle Störungen, die durch betätigte Endlagenschalter hervorgerufen werden, an einem separatem DI-Port, der Steuerung zu melden.

Die Steuerung liest dann die DI-Ports aller Endlagenschalter ein und wertet diese dann explizit aus. Durch diesen Vorgang müssen die Endlagenüberwachungen nicht zyklisch eingelesen werden. Es wird somit Zykluszeit bei der Abarbeitung des Programmablaufes eingespart.

Durch diese Maßnahme kann sofort bei anstehender Störung entsprechend reagiert werden und eventuell eingeschaltete Antriebseinheiten mit sofortiger Wirkung stillgesetzt werden um mechanische Schäden an der Antriebseinheit, insbesondere Beschädigungen der Spindelmutter des Gewindetriebes, zu verhindern.





Durch dieses Prinzip wird eine Ereignissteuerung des Programmablaufes verwirklicht. Diese Interruptsteuerung des Programmablaufes vermischt sich mit der prinzipiellen

Verknüpfungssteuerung des Ablaufes. Diese Mischform zur Steuerung eines Programmablaufes wird im Kapitel *Software* verdeutlicht.

Weitere ODER-Bausteine werden verwendet, um die Überlastausgänge der Leistungsstufen für die Ansteuerung der Antriebsmotoren zu überwachen.

Die Leistungsstufen, über die Sie im Abschnitt *Leistungsteil* mehr Informationen erhalten, verfügen über sogenannte OCD-Ausgänge. Diese OCD-Ausgänge können durch entsprechende Beschaltung des Bausteines dazu genutzt werden, um eine Überlast des angeschlossenen Antriebsmotor durch eine erhöhte Stromaufnahme festzustellen und anzuzeigen.

Die Schaltschwelle dieses Signals ist durch die Beschaltung des Bausteines festzulegen. Diese Beschaltung können Sie dem Schaltplan entnehmen.

Ein weiterer Baustein des Logikteiles ist ein nicht invertierender 4-Kanal Multiplexer mit enable-Eingang, ein SMD-Baustein im SO16-Gehäuse des Typs 74ALS153M. Der Baustein dient der Feedbackumschaltung der einzelnen Inkrementalgeber der Antriebseinheiten von Achse 1 bis Achse 3. Die Impulse des Inkrementalgebers der Achse 4 werden nicht über diesen Baustein geführt.

Durch entsprechende Ansteuerung des Bausteines wird der Geberdatenkanal der im Moment angesteuerten Antriebseinheit auf einen DI-Port geschaltet.





Es kann durch diese Art der Ausführung jeweils eine Achse der Achsen 1 bis 3 und die Achse 4 in Betrieb sein, wenn eine Lageüberwachung zur Positionierung der Einheit erforderlich ist.

Im Servicefall gibt es auch Ausnahmen, bei welchen auch mehr als zwei Achsen in Betrieb gehen können. Allerdings ist dann keine absolut prozesssichere Lageüberwachung garantiert.

# 7.3 Leistungsteil



(Abb. 7-6)

Der Leistungsteil befindet sich auf der linken Seite der Basisplatine und besteht hauptsächlich aus zwei Motorsteuer-ICs und deren Beschaltung. Bei den Motorsteuer-ICs handelt es sich um zwei SMD-Bausteine im SO36-Gehäuse des Typs L6206PD des Herstellers ST-Microelectronics.

Von Ausgängen (DO-Ports) der Steuerung angesprochen, haben diese ICs die Aufgabe, die Motoren der Antriebseinheiten ein- und auszuschalten.

Die Motorsteuer-ICs arbeiten nach dem Prinzip einer H-Brücke. Jeder Baustein enthält zwei H-Bücken, es können also zwei Motoren pro IC gesteuert werden.



Um die Verlustwärme ableiten zu können, sind diese Bausteine mit einer Metallfläche an ihrer Unterseite versehen. Diese Metallfläche ist mit einer beidseitigen, durchkontaktierten Kühlfläche auf der Basisplatine verbunden.

Durch dieses Prinzip wird eine Beschädigung der Bauteile durch thermische Überlastung ohne Verwendung von Kühlkörpern verhindert.

Folgende Skizzen verdeutlichen dies :





Neben den Anschlüssen für die Motoren sind in den Bausteinen enable-Eingänge integriert um die ICs freizugeben bzw. zu sperren.

Wie bereits schon im Abschnitt *Logikteil* beschrieben, verfügen die Bausteine über sogenannte OCD-Ausgänge. Diese Ausgänge bieten Diagnosemöglichkeiten :

- Die OCD-Ausgänge können dazu genutzt werden, um eine Überlast des angeschlossenen Antriebsmotor durch eine erhöhte Stromaufnahme festzustellen und anzuzeigen. Die Schaltschwelle dieses Signals ist durch die Beschaltung des Bausteines festzulegen.
- Durch eine weitere Verschaltung dieser Ausgänge und Weiterführung auf einen DI-Port der Steuerung, kann sofort auf ein Diagnosesignal reagiert werden und entsprechende Vorgänge abgebrochen bzw. gestartet werden.
- Die Signale werden in der Steuerung weiterverarbeitet und über die HMI eine Meldung zur Bedienerführung ausgegeben.

Um bei den Motorsteuer-ICs ebenso wie bei allen anderen Komponenten der Basisplatine auf Kühlkörper verzichten zu können, wird die entstehende Wärme auf eine Kühlfläche der Basisplatine abgeführt. Diese Kühlfläche ist ausreichend groß, um Schäden durch thermische Überlastung zu verhindern.

Die Strombegrenzung der Antriebe wurde durch Beschaltung der ICs mit 18k Widerständen auf 1,2 A festgelegt.

## 7.4 Anschlussmöglichkeiten zur Peripherie



(Abb. 7-10)

Im unteren Bereich der Basisplatine sind die Anschlussmöglichkeiten zur Peripherie. Diese Leiterplattenbuchsen sind in SMD-Ausführung gestaltet. Die Steckplätze X4, X5, X6 und X7

sind für die Achsen 1 bis 4 reserviert. Über diese Steckverbindungen werden die Achsantriebe versorgt. Aufgelegt sind jeweils die Leistung für Drehrichtung plus, Leistung für Drehrichtung minus, Masse, die Spannungsversorgung für die Inkrementalgeber und das Feedback der Inkrementalgeber.

Über den Steckverbinder X8 werden die Signale der Referenzpunktschalter und der Endlagenschalter der Achsen 1 bis 4 aufgelegt. Die Spannungsversorgung wird pro Achse einmal aufgelegt und jeweils an den Antriebseinheiten entsprechend der Schalter verteilt.

Die Buchse X9 dient dem Anschluss des Neigungssensors und des Gehäuseüberwachungsschalters im Deckel des Messautomaten.

An der Leiterplattenbuchsen X10 kann weitere Peripherie wie Schalter, Taster, etc. angeschlossen werden. Diese Buchse ist bereits intern mit der Steuerung verschaltet und über invertierende Schmitt-Trigger Bausteine geführt um eindeutige Signalpegel von der Peripherie zu erhalten. Diese Eingänge sind auf bisher freie DI-Ports der Steuerung gelegt und dienen der optionalen Erweiterung der Basisplatine.

Über die Leiterplattenbuchse X11 wird die Verbindung zur Adapterplatine an der Frontplatte im Gehäuse des Messautomaten hergestellt. Dort befindet sich ein weiterer Reset-Taster, um die Steuerung rückzusetzen.

Um Softwareupdates der Steuerung benutzerfreundlich durchführen zu können, wurde dort ebenso ein Boot-Taster installiert. Mit diesen Tastern lässt sich die Steuerung in den Boot-Strap-Mode setzen, der erforderlich ist, um ein update zu übertragen.

Eine V24-Schnittstelle ist auf der Adapterplatine enthalten, welche auch in Abbildung 6-9 zu erkennen ist.

## 7.5 Serielle Schnittstellen



(Abb. 7-11)

Die seriellen Schnittstellen S0 und S1 befinden sich im rechten seitlichen Bereich der Basisplatine. Die RS232-Schnittstelle S0 ist für den Anschluss eines kompatiblen PCs reserviert. Über diese Schnittstelle kann der Messautomat wie in Abbildung 3-1 dargestellt, in den Messaufbau integriert werden.

Im Servicefall kann über die Schnittstelle S0 der Messautomat über die Software OLLICONTROL (OC) gesteuert werden. Diese Schnittstelle ist ebenso auf die Frontplatte im Gehäuse des Messautomaten über die Adapterplatine nach außen geführt.

Wie auch auf der Basisplatine vorhanden, sind über die Adapterplatine zwei Taster nach außen geführt. Die Funktion des Reset- und des Boot-Tasters wurde bereits im Abschnitt *Anschlussmöglichkeiten zur Peripherie* beschrieben.

Die RS232-Schnittstelle S1 ist für den Anschluss der HMI reserviert. Das LCD-Touchpanel kommuniziert über diese Schnittstelle mit der Steuerung.

Um eine Verwechslung der beiden Schnittstellen zu vermeiden, sind sie auf der Basisplatine gekennzeichnet. Zusätzlich wurde darauf geachtet, dass auf der Basisplatine eine Schnittstelle als Buchse, die andere Schnittstelle als Stecker ausgeführt ist.

## 7.6 Debugbereich

	ан 1. а. т	Dect.			TRANK.	-
						1
		Summe <sup>1.4</sup>		- 3	-	
	Suma fo	Rename Parties				
		1	C			12
1.1	) 	1		8	4	5445
· Strain				17.8	ΞĐ	
inter R inter R	inter printer ()	heterotababababababababababababababababababab		() e de la		-
Abb. 7-12	2)	ALCOURSE AND				

Unterhalb der Prozessorplatine befindet sich der Debugbereich. Dort sind zu Simulations- und Testzwecken 8 LEDs angebracht. Diese LEDs sind an nicht für die Automation benötigte Ports des Mikrocontrollers angeschlossen und können für Debugzwecke benutzt werden.

Die Pinbelegung des Mikrocontrollers ist in Kapitel 8 ersichtlich. Dort können Sie auch die Bezeichnungen für die verwendeten Portpins der Debug-LEDs ersehen.

## 7.7 Reservierter Bereich



(Abb. 7-13)

Dieser Bereich ist für die Prozessorplatine reserviert. Aus EMV-technischen Gründen wurde dieser Bereich für andere Bauelemente nicht genutzt und wurde somit ausschließlich für die Prozessorplatine freigehalten.

Wie bei der gesamten Basisplatine wurde auch hier darauf geachtet, die Führung der Leiterbahnen unter EMV-technischen Aspekten zu gestalten.

#### Schaltpläne der Basisplatine 7.8

Die Schaltpläne und das gesamte Layout der Basisplatine wurden mit der CAD-Software Eagle, Version 4.11, erstellt.



(Abb. 7-14)











(Abb. 7-17)

# 7.9 Layout Basisplatine



Eine Gesamtansicht des Platinenlayouts. Es wird darauf verzichtet, jeden Layer der Platine hier abzubilden. Die vollständigen Ansichten der verschiedenen Layer sind auf der Dokumentations-CDR enthalten und können u. A. mit der CAD-Software Eagle, Version 4.11, eingesehen werden.

# 7.10 Bestückungsdruck Basisplatine



# 8 Details Prozessorplatine

# 8.1 Allgemein

Details, die Sie in diesem Kapitel vermissen werden, können Sie Kapitel 6.2.2.2 entnehmen. Dort wurden bereits einige Informationen, die auch Details betreffen, behandelt.

In diesem Kapitel erhalten Sie Informationen über spezifische Eigenschaften bzw. Zusammenhänge der Prozessorplatine in Verbindung mit dem Messautomat.

# 8.2 Blockschaltbild Prozessorplatine



Blockschaltbild des miniModuls-509

# 8.3 Pinbelegung Mikrocontroller

Portpin	bitadr.	Funktion	altern. Funktion	Pinnr	. Phy	tec		Funktion	Beschaltung
P1.0	ia	In/Out	/INT 3 input	X1A	15b	30	30		, , , , , , , , , , , , , , , , , , ,
P1.1	ja	In/Out	INT 4 input	X1A	15a	29	29	Interrupt Endschalterschalter	Pulldown
P1.2	ja	In/Out	INT 5 input	X1A	14b	28	28	Interrupt OCD Motorsteuer-ICs	Pullup
P1.3	, ja	In/Out	INT 6 input	X1A	14a	27	27	Neigungssensor	Pulldown
P1.4	ja	In/Out	/INT 2 input	X1A	13b	26	26		
P1.5	ja	In/Out	In/Out	X1A	13a	25	25	Deckelschalter	Pulldown
P1.6	, ja	In/Out	CLKOUT	X1A	12b	24	24		1
P1.7	; ja	In/Out	Timer2 extern input	X1A	12a	23	23		
P3.0	, ia	In/Out	Serial0 RxD 0	X1A	8a	15	15	Serielle Schnittstelle 0 RxD	
P3.1	ia	In/Out	Serial0 TxD 0	X1A	8b	16	16	Serielle Schnittstelle 0 TxD	
P3.2	ia	In/Out	/INT 0 input	X1A	9a	17	17		
P3.3	ia	In/Out	/INT 1 input	X1A	9b	18	18		1
P3.4	ia	In/Out	Counter 0 input	X1A	10a	19	19	Inkrementalgeber Achse 1-3	
P3.5	ia	In/Out	Counter 1 input	X1A	10b	20	20	Inkrementalgeber Achse 4	
P3.6	ja	In/Out	/WR	X1A	11a	21	21		
P3.7	; ja	In/Out	/RD	X1A	11b	22	22		
P4.0	ja	In/Out	CM0	X1A	2a	3	3	Antrieb Achse 1 Richtung (-)	
P4.1	, ia	In/Out	CM1	X1A	2b	4	4	Antrieb Achse 1 Richtung (+)	1
P4.2	ja	In/Out	CM2	X1A	3a	5	5	Antrieb Achse 2 Richtung (-)	
P4.3	, ja	In/Out	CM3	X1A	3b	6	6	Antrieb Achse 2 Richtung (+)	
P4.4	, ja	In/Out	CM4	X1A	4a	7	7	Antrieb Achse 3 Richtung (-)	
P4.5	ja	In/Out	CM5	X1A	4b	8	8	Antrieb Achse 3 Richtung (+)	
P4.6	ja	In/Out	CM6	X1A	5a	9	9	Antrieb Achse 4 Richtung (-)	
P4.7	ja	In/Out	CM7	X1A	5b	10	10	Antrieb Achse 4 Richtung (+)	
P5.0	ja	In/Out	CCM0	X1B	26b	52	116	enable PWR Motoren - enable mit 0	1
P5.1	ja	In/Out	CCM1	X1B	26a	51	115	enable Multiplexer - enable mit 0	1
P5.2	ja	In/Out	CCM2	X1B	25b	50	114	Multiplexer select Geberdatenkanal	1
P5.3	ja	In/Out	ССМЗ	X1B	25a	49	113	Multiplexer select Geberdatenkanal	1
P5.4	ja	In/Out	CCM4	X1B	24b	48	112	DebugLED 0	Pulldown
P5.5	ia	In/Out	CCM5	X1B	24a	47	111	DebugLED 1	Pulldown
P5.6	ia	In/Out	CCM6	X1B	23b	46	110	DebugLED 2	Pulldown
P5.7	ja	In/Out	CCM7	X1B	23a	45	109	DebugLED 3	Pulldown
P6.0	, nein	In/Out		X1B	28a	55	119	, ,	
P6.1	nein	In/Out	Serial1 RxD 1	X1B	28b	56	120	Serielle Schnittstelle 1 RxD	
P6.2	nein	In/Out	Serial1 TxD 1	X1B	29a	57	121	Serielle Schnittstelle 1 TxD	
P6.3	nein	In/Out		X1B	29b	58	122		
P6.4	nein	In/Out		X1B	30a	59	123	DebugLED 4	Pulldown
P6.5	nein	In/Out		X1B	30b	60	124	DebugLED 5	Pulldown
P6.6	nein	In/Out		X1B	31a	61	125	DebugLED 6	Pulldown
P6.7	nein	In/Out		X1B	31b	62	126	DebugLED 7	Pulldown
P7 0	nein	In	AnIn0	X1C	5b	10	138	Endlage- Achse 1	Pulldown
P7 1	nein	In	AnIn1	X1C	6b	12	140	Endlage+ Achse 1	Pulldown
P7.2	nein	In	AnIn2	X1C	7b	14	142	Endlage-, Achse 2	Pulldown
P7.3	nein	In	AnIn3	X1C	8b	16	144	Endlage+. Achse 2	Pulldown
P7.4	nein	In	AnIn4	X1C	9b	18	146	Endlage-, Achse 3	Pulldown
P7.5	nein	In	AnIn5	X1C	10b	20	148	Endlage+, Achse 3	Pulldown
P7.6	nein	In	AnIn6	X1C	11b	22	150	Endlage-, Achse 4	Pulldown
P7.7	nein	In	AnIn7	X1C	12b	24	152	Endlage+, Achse 4	Pulldown
P8 0	nein	In	AnIn8	X1C	1b	2	130		1
P8 1	nein	In	Anin9	X1C	2b	4	132		
P8.2	nein	In	AnIn10	X1C	3b	6	134		
P8.3	nein	In	AnIn11	X1C	4b	8	136		
P8.4	nein	In	AnIn12	X1C	2a	3	131		
P8.5	nein	In	AnIn13	X1C	4a	7	135		
P8.6	nein	In	AnIn14	X1C	6a	11	139		
P9.0	nein	In/Out	CC10	X1B	22b	44	108	Referenzpunktschalter Achse 1	Pulldown
P9.1	nein	In/Out	CC11	X1B	22a	43	107	Referenzpunktschalter Achse 2	Pulldown
P9.2	nein	In/Out	CC12	X1B	21b	42	106	Referenzpunktschalter Achse 3	Pulldown
P9.3	nein	In/Out	CC13	X1B	21a	41	105	Referenzpunktschalter Achse 4	Pulldown
P9.4	nein	In/Out	CC14	X1B	20b	40	104	mit Stecker X10 / 1 kontaktiert	Pulldown
P9.5	nein	In/Out	CC15	X1B	20a	39	103	mit Stecker X10 / 2 kontaktiert	Pulldown
P9.6	nein	In/Out	CC16	X1B	19b	38	102	mit Stecker X10 / 3 kontaktiert	Pulldown
P9.7	nein	In/Out	CC17	X1B	19a	37	101	mit Stecker X10 / 4 kontaktiert	Pulldown
	1	1	Serial0 RxD 0	X1A	23a	45	45	Serielle Schnittstelle 0 RxD	1
		1	Serial0 TxD 0	X1A	24a	47	47	Serielle Schnittstelle 0 TxD	
		İ	Serial1 RxD 1	X1A	28a	55	55	Serielle Schnittstelle 1 RxD	
			Serial1 TxD 1	X1A	23b	46	46	Serielle Schnittstelle 1 TxD	
		AGND	AGND	X1C	1a	1	129		
		RESET	/Reset	X1A	31a	61	61	RESET	
		BOOT / D0	BOOT 537	X1B	5b	10	74	воот	1
		VCC	VCC	X1B	1a	1	65	5V Power Supply for C509	
		VCC	VCC	X1B	1b	2	66	5V Power Supply for C509	
		GND	GND	X1B	32a	63	127	GND Power Supply for C509	
		GND	GND	X1B	32b	64	128	GND Power Supply for C509	1

(Abb. 8-2)

# 9 Programmierung des Mikrocontrollers C509

## 9.1 Allgemeines

Mit dem Mikrocontroller C509 von Infineon wurde die Steuerung des Messautomaten verwirklicht. Der Mikrocontroller ist auf der Prozessorplatine integriert.

Als Programmiersprache wurde die Sprache C gewählt, die Entwicklungsumgebung von Altium.

About EDE	×
TASKING Embedded Development Environment (EDE) Build: 105 Senialit: 095894	
Selected Toolchain: TASKING Tools for 8051 v7.0 r5	al TASKING
Copyright 1997-2002 Altium BV This software is protected by copyright law and international treaties.	Embedded saftware development
http://www.tarkins.com	nuin Annum.

(Abb. 9-1)

Tasking Embedded Development Environment, EDE, ist ein leistungsstarkes Tool, das zur Entwicklung des C-Programmes dient.

Das C-Programm des Mikrocontrollers stellt den Kern des Messautomaten dar. Hier werden die Schnittstellen des Controllers verwaltet, der Signalaustausch mit der Peripherie gesteuert.

Als Compiler dient der 8051 Cross Compiler von Tasking.

Um das generierte project.hex file in den Controller zu übertragen, wurde das flashtool 98 von Phytec verwendet. Über eine RS232-Verbindung zur seriellen Schnittstelle S0 des Controllers werden die Daten in den EEPROM-Speicher des Controllers geladen.





Hierzu muß der Controller in den Boot-Strap-Mode gesetzt werden. Dies erreicht man über das gleichzeitige Betätigen des Reset- und des Boottasters. Dabei muß der Boottaster länger als der Resettaster betätigt werden.

Die beiden Taster sind ebenso wie die serielle Schnittstelle S0 über die Adapterplatine auf die Frontplatte des Messautomaten herausgeführt.

Genauere Bedienungshinweise zum "flashen" des EEPROMS finden Sie in den folgenden Kapiteln.

## 9.2 Tasking

Nach der Installation der Entwicklungsumgebung EDE 8051auf einem kompatiblen PC kann EDE 8051 gestartet werden.

Informationen zur Installation von EDE 8051 erhalten Sie aus dem Handbuch zu EDE 8051 oder sind über Tasking zu beziehen.

Nach dem Start der Entwicklungsumgebung ist zuerst ein neuer Arbeitsbereich, ein ProjectSpace, zu erstellen.

In einem ProjectSpace können verschiedene Projekte abgelegt oder bearbeitet werden. Das momentan zu bearbeitende Projekt ist als "current project" zu deklarieren. Ein Projekt kann mehrere Dateien beinhalten.

Das EDE 8051-Projekt des Messautomaten beinhaltet folgende Dateitypen :

- Dateien mit C-Quellcode : \*.c
- Headerdateien : \*.h
- Controllerspezifische Dateien : \*.sfr
- Umgebungsspezifische Dateien: \*.asm
- Die nach dem compilieren generierte \*.hex Datei befindet sich im Stammverzeichnis des ProjectSpace

Die Dateien können über die Eigenschaften des Projektes verwaltet werden. Dort werden nicht mehr benötigte Dateien aus dem Projekt ausgegliedert oder neue Dateien dem EDE 8051 Projekt hinzugefügt.

Ebenso können dort Verzeichnisse eingestellt werden, in welche Dateien geschrieben werden oder in welchen auf Dateien zugegriffen werden.

Zu den Grundeinstellungen gehört auch die Einstellung der Projektoptionen. Diese Projektoptionen können auch von anderen Projekten, auch außerhalb des aktiven ProjektSpace, übernommen werden.

051 Project Options [OLLI_4.PJT]		x
	Processor Processor Selection Memory Startup Code Bypasses Dyptions string: Options string: Selected SFR file: regsc509.sfr On chip RAM size: 256	
	OK Cancel Defaults Help	

(Abb. 9-3)

In den Projektoptionen müssen Einstellungen zu verwendeter Hardware oder aber auch beispielsweise Compilereinstellungen vorgenommen werden.



Durch die Einstellung der Projektoptionen wird die Software an die Hardware angepasst.

Ein Auszug der Einstellungen :

- Einstellungen zum Prozessor
  - Prozessortyp
  - Größe des Speichers
- Einstellungen des C Compilers
  - Preprozessoreinstellungen
  - Codegenerierung
  - Optimierung
  - Ausgabe von Fehlern und Warnungen
  - Speichermodell
- Einstellungen zum Assembler
  - Angaben zu listfiles
  - Registerbänke
  - Einstellungen zum Linker
  - Ausgabeformat
  - Ausgabe von Fehlern und Warnungen
- Einstellungen zu Cross View Pro
  - Initialisierung
  - Verschiedene Einstellungen zum Debugger
- Flasher
  - Übertragungsrate
  - Flashertyp

Die gesamten Einstellungen werden in einer Datei abgelegt und können so jederzeit gegen eine andere Optionseinstellung ausgetauscht werden.

Die Oberfläche von EDE 8051 kann individuell an verschiedene Bedürfnisse angepasst werden. Schrift- und Hintergundfarbe, Schriftgröße und sämtliche Einstellungen, bezüglich der Darstellung, können wie bei Windows-Arbeitsoberflächen üblich, eingestellt werden.

## 9.3 Ablaufdiagramme C-Projekt

Die folgenden Seiten sollen einen Überblick über das C-Projekt verschaffen. Hier wird nicht der Zusammenhang der einzelnen C-Dateien zueinander dargestellt, sondern der Ablauf der Automation im Gesamten.

Auf eine Darstellung aller Details wird bewusst verzichtet. Die Ablaufdiagramme wurden so gestaltet, dass eine Übersicht über die Automation gegeben werden kann.

Das C-Programm wurde so gestaltet, dass ein zyklischer Ablauf gegeben ist. Die Hauptroutine ist der Ausgangspunkt der Ablaufdiagramme. Diese Hauproutine ist Koordinator des

gesamten Ablaufes. Von hier werden Funktionen des Messautomaten gesteuert, Unterprogramme aufgerufen und Abläufe organisiert.

Die einzelnen Schritte im Ablaufdiagramm stellen nicht die wirklich programmierten C-Funktionen dar, sondern wie schon im ersten Abschnitt dieses Kapitels erwähnt, sinngemäß die Abläufe im Programm. So können z. T. verschiedene C-Funktionen über mehrere Dateien zusammengefasst sein.

Solche Schritte werden hier als Modul bezeichnet. Jedes Modul erhält eine Modulnummer. Das Lupensymbol an einem Modul macht deutlich, dass es zu diesem Modul eine noch genauere Darstellung gibt. Dieses Prinzip ist über mehrere Ebenen gültig.

Die Querverweise sind logisch aufgebaut. Die Blattnummer des Querverweises ergibt sich aus der Modulnummer des entsprechenden Schrittes. Zur besseren Orientierung sind Querverweise auf externe Module farblich gestaltet.

Die Modulnummern sind ebenso logisch aufgebaut. Die Module einer tieferen Ebene erhalten eine Modulnummer, die sich auf das Ursprungsmodul der höheren Ebene bezieht.

Das Prinzip dieser Nummerierung finden Sie nicht nur hier, dieser Grundsatz wurde an allen neuen Komponenten des Messautomaten umgesetzt, wie Sie beispielsweise auch aus den Stromlaufplänen des Messautomaten entnehmen können.

Durch diese Gliederung lässt sich der Programmaufbau verständlich darstellen.

Im jeweiligen dargestellten Schritt erhalten Sie eine Kurzbeschreibung der Funktion des Moduls.

Diese Informationen können Sie nochmals zusammengefasst der Legende auf Blatt 0.1 des Ablaufdiagrammes entnehmen.

Auf Blatt 0.2 des Ablaufdiagrammes können Sie eine Übersicht des prinzipiellen Aufbaus des Programms erkennen. Die Struktur des Programmes entspricht einer Ablaufsteuerung mit Ereignisroutinen.



















(Abb. 9-8)

Ablaufdiagramm Automatisierung
serielle Schnittstellen S0 und S1
Modul 1 Blatt 10.0

(Abb. 9-9)











(Abb. 9-13)





(Abb. 9-14)










(Abb. 9-17)



(Abb. 9-18)



(Abb. 9-19)

























Ablaufdiagramm Automatisierung Fehlergenerierung bei unerlaubtem BA-Wechse Modul 11 Blatt 110.0

(Abb. 9-29)





























































(Abb. 9-47)












# 9.4 Details zur Programmierung

## 9.4.1 Allgemeines

In den folgenden Abschnitten werden einige Details des Programmablaufes dargestellt und explizit verdeutlicht. Auf die Erklärung jedes Details zur Programmierung wird jedoch verzichtet.

Auf der Dokumentations-CDR finden Sie den gesamten Quellcode des C-Projektes. Durch ausreichende Kommentierung können hier weitere Detailinformationen gewonnen werden.

Die Erklärungen in den weiteren Abschnitten sind nicht nach ihrem Umfang oder ihrer Dominanz sortiert, sondern bezugnehmend auf die Gliederung der Programmablaufdiagramme in Kapitel 9.3.

Die Programmauszüge werden den Modulnummern der Programmablaufpläne in Kapitel 9.3 zugeordnet.

## 9.4.2 Modul 1: Initialisierung der seriellen Schnittstellen

In Modul 1 werden die seriellen Schnittstellen S0 und S1 initialisiert. Der folgende Programmauszug stellt die Initialisierung der seriellen Schnittstelle S0 dar.

	//Initialisierung der Seriellen Schnittstelle 0
	void seriell_S0_init (void)
	SMO = 0;
140	SM1 = 1; //im SFR SOCON(98h) sind die Bits 6 und 7 für die ModeEinstellung
	//zuständig. SMO = 0; SM1 =1 für Mode 1 oder: SOCON = 0xCOh
	//Receiver Enable Bit im SOCON Byte, ermöglicht seriellen Empfang, wenn
145	//gesetzt. Wird nur durch Software beeinflußt BD = 1;
	//im SFR ADCON0(D8h) BD=1 (7Bit) um den Baudratengenerator anzuschalten:
	//Nach Reset ist diese Bit auf "0"
150	//Diese Bit ist direkt ansprechbar
	PCON = (PCON & 0x7F);
	//dmit kann die Baudrate verdonnelt werden
	//wenn Bit SMOD: "1" ist, dann wird Baudrate verdoppelt.
155	//Nach Reset ist dieses Bit und das Byte auf "O"
	//Diese Bit ist nicht direkt ansprechbar
	PRSC = (PRSC   0x40); //im GEP DRSC(PAb) ist SOD das 6Bit damit wird die Baudrate verdoppelt
	//Nach Reset ist diese Bit auf "l"
160	//Nach Reset ist dieses Byte auf "1101 0101b"
	//Diese Bit ist nicht direkt ansprechbar.
	SORELH = 0x03;
	//damit werden die 2 oberen MSBits vom 10Bit Zahlregister eingestellt //diese sind das 0 und 1 Bit im SORELH SFR
165	<pre>//nach Reset ist dieses Byte auf "00000 0011b" = "0x03h" eingestellt.</pre>
	//fur 9600Baud mub diese Byte auf "xxxx xxllb" = "0x03h" //eingestellt sein
	SOREL = OxE6;
	//damit werden die 8 unteren LSBits vom 10Bit Zählregister eingestellt.
170	//nach Reset ist dieses Byte auf "1101 1001b" = "0xD9h" eingestellt.
	//fur 9600Baud mus diese Byte auf "lilo Ullob" = "0xE6h" eingesteilt.
	//[1024 - ( (16Wh2) / (64*Bdrate) ) ]= "998d"
	//lt. SiemensBuch ist die Formel:
175	//[(2hochSMOD x 16hoch6) / (64 x (1024-SOREL) x 2hochSOP)]

Durch ausführliche Kommentierung der einzelnen Befehlszeilen des Quellcodes wird eine weitere Erklärung überflüssig.

Die Initialisierung der seriellen Schnittstellen S0 und S1 erfolgt in der Datei *seriell\_olli\_seriell.c.* Der Aufruf der entsprechenden Funktionen erfolgt, wie Sie auch aus dem Programmablaufplan entnehmen können, vor dem zyklisch ablaufenden Teil des Programmes.

Die Steuerung der Funktionsaufrufe geschieht hauptsächlich aus der Datei *main.c.* Diese Datei ist der Koordinator der zyklischen Abarbeitung der verschiedenen Programmteile.

Wie die Initialisierung der seriellen Schnittstelle S0 ist auch die Initialisierung der seriellen Schnittstelle S1 durch Kommentierung der einzelnen Befehlszeilen des Quellcodes nachvollziehbar.

435	//Initialisierung der Seriellen Schnittstelle 1 wold seriell Sl init (wold)
	S1CON = 0xD0;
	//ist gesetzt auf "1101 0000b" oder "0xD0h"
440	//im SFR SICON sind folgende Bits:
	//Nach Reset 1st diese Byte auf "0100 00000" oder "0x40n"
	//bit 1 (RII): RX(receiver)Interrupt IIag
	/bit 2 (DDP1): A tag manaluar bit
445	//bit 3 (TBR1) + 9 tes fransmitter bit
	//bit 4 (REN1): wenn 0 dann ist kein Serieller Empfang möglich
	//bit 5 (SM21): Mutiprocessor enable (bleibt immer auf "0")
	<pre>//bit 6 (S1P): wenn 1(nach Reset)prescaler is active</pre>
	// (für Baudraten Generator)
450	//bit 7 (SM): Mode select bit:0=9bit Uart; 1=8bit Uart
	//Diese Bits in diesem Byte sind nicht direkt ansprechbar !!
	SIRELH = UxU3;
	//damit werden die 2 oberen maaits vom tobit Zahiregister eingestellt
455	//nese sing das o und i bit in sinche set $(/nach Reset ist dieses Rute auf "00000011b" = "0x03b" eingestellt.$
100	//für 9600Baud muß diese Byte auf "xxxxxxlb" ="0x03h" eingestellt sein
	SIRELL = 0xCC;
	//damit werden die 8 unteren LSBits vom 10Bit Zählregister eingestellt.
	//nach Reset ist dieses Byte auf "00000000b" = "0x00h" eingestellt.
460	//für 9600Baud muß diese Byte auf "1100 1100b" ="0xCC" eingestellt sein
	//entspricht "972d"
	//Die Formel zu Berechung der 10Bits (SIREL)ist:
	//[ 1 / [BGFate^32] / IonOcno] ] + 1024 //lt SigmapBuch ist die Formal:
4.65	//I bhoch / /32 x ( $1024$ -SIREL) x 2 hoch SIR)1
	) }

## 9.4.3 Modul 2: Initialisierung der Ports

Die Initialisierung der Ports geschieht hauptsächlich in der Funktion *main { }* der Datei *main.c.* DO-Ports (digitale Ausgangsports) werden mit einer logischen 0 initialisiert, DI-Ports (digitale Eingangsports) werden mit einer logischen 1 initialisiert.

```
// Ausgänge rücksetzen
        // Ausgang Motoren auf LOW
P4=0x00;
145
        // Bit 0,1 setzen; Port 5
// disable Multiplexer, disable Motorenansteuerung
        P5=0x03;
        // Port 7 zum Einlesen vorbereiten
// Eingangsbyte Achsen Endlagenschalter
150
        P7=0xff:
        // Port 9 zum Einlesen vorbereiten
155
         // Referenzpunktschalter P9.0 - P9.3
        P9=0xff;
        // Port1.3 + Pl.5 zum Einlesen vorbereiten
Pl=0x18;
160
        // Ausgänge für "Test-LEDs" rücksetzen
P6 = (P6 & 0x0f);
        // Aufruf Interruptinitialisierung
165
        interrupt_init();
        // Überwachung auf angefahrene Endlage ist mit INT4, (in C Nr.11) realisiert
170
        // Überwachung auf Motorsteuer-IC Überlast ist mit INT5 (in C 12) realisiert
        175
180
        // Aufruf Motorenansteuerung enable
        mot_pwr_enable();
```

Nach der Initialisierung der Ports erfolgen verschiedene Funktionsaufrufe zur Initialisierung diverser Interrupts und der Umdrehungszähler.

Details hierzu können Sie Kapitel 9.4.4 und Kapitel 9.4.5 entnehmen.

## 9.4.4 Modul 3: Initialisierung der Interrupts

In Modul 3 ist die Initialisierung der Interrupts realisiert. Ein Auszug der Datei *intrrpt\_init.c* stellt dies dar.

```
// Interruptinitialisierung
void interrupt_init (void)
        // Interruptnr. 1 (Port 3.4, Countereingang Achse 1-3)
// Überlauf Counter Geberinkremente
30
        ET0=1:
        // Interrupt-Request-Bit ist TF0 und wird von Timer0 gesetzt
35
        TF0=0;
        // Interruptnr. 3 (C), Port 3.5, Countereingang Achse 4
ET1=1;
40
        TF1=0;
        // Interrupt 4, Interruptnr. 11, Port 1.1, Endlage einer Achse angefahren
        IEX4=0:
        EX4=1;
45
           Interrupt 5, Interruptnr. 12, Port 1.2, MotorsteuerIC - Overload/Überlast
        // Inte
IEX5=0;
50
        EX5=1;
```

Die Nummern der Interrupts sind diejenigen, die unter Verwendung der Programmiersprache C unter EDE 8051 verwendet werden müssen.

Folgende Interrupts werden verwendet:

## Interruptnummer 1

- Countereingang der Achsen 1-3 an Port 3.4
- Überlauf des Counters der Geberinkremente

#### **Interruptnummer 3**

Countereingang der Achse 4

#### Interruptnummer 4

Auswertung Endlagenstörung der Antriebsachsen

#### **Interruptnummer 5**

Überlast der Motorsteuer-ICs

Im Folgenden die Initialisierung der Interrupts der seriellen Schnittstellen S0 und S1.

Der Initialisierung inbegriffen, ist die Festlegung der Prioritäten. Diese werden über die SFR (Special Function Register) IP0 und IP1 festgelegt.

Im Anschluss erfolgt die generelle Freigabe aller Interrupts über das Setzen von EAL.

80	// // Prioritäter IP0=0x0A; // IP1=0x9B; //	1 einstellen 7 00001010 7 10011011		
85	//			
	// Intrptnr.	Funktion	Priorität	Bit in IPO/IP1
	// 16	serielle Schnittstelle	2	IP1.0/IP0.0
	// 1	Impulszählung Geber	3	IP1.1/IP0.1
	// 3	Impulszählung Geber	3	IP1.3/IP0.3
90	// 11	Endlagenüberwachung	3	″
	// 4	serielle Schnittstelle	2	IP1.4/IP0.4
	// 12	Motoren Überlast	2	"
	// 13	Deckelüberwachung	0	IP1.5/IP0.5
	//			
95	// um P9 einle	esen zu können muss IP1.7 mi	t 1 beschrieb	en werden
	// alle Intern	upts enable		
	EAL=1;	-		
}				

Detailliertere Informationen können Sie dem Kommentar des Quellcodes entnehmen.

## 9.4.5 Modul 4: Initialisierung der Umdrehungszähler

Die Lageregelung der Antriebseinheiten wird mit Inkrementalgebern realisiert. Diese Inkrementalgeber sind motorseitig montiert und liefern pro Umdrehung 16 Impulse. Jeder Impuls löst einen Interrupt aus. Achsen 1-3 arbeiten mit dem Interrupt Nummer 1, Achse 4 arbeitet mit Interrupt Nummer 3.

Aus der Auflösung des Gebers von 16 Impulsen pro Umdrehung folgt, dass 16 Impulse eine Motorumdrehung ergeben müssen. Somit wird alle 16 Impulse der Umdrehungszähler der entsprechenden Achse inkrementiert.

Aus der Getriebeübersetzung von 23:1 ergibt sich, dass alle 23 Motorumdrehungen der Umdrehungszähler der entsprechenden Spindel inkrementiert wird.

Diese Umdrehungszähler werden jedes Mal initialisiert, wenn die Antriebseinheit auf Referenzpunkt fährt. Die Grundstellung einer Achse ist ebenso Referenzpunktstellung.

Mit diesem Verfahren werden Positionierungenauigkeiten durch z.B. Umkehrlose oder mechanischen Verschleiß kompensiert.

Theoretische Positionierungenauigkeiten würden auch bei Drehrichtungsumkehr einer Achse entstehen, wenn die Drehrichtungsumkehr bei einer nicht vollendeter Motorumdrehung eingeleitet würde. Diese Abweichung würde aber weniger als 0,04 mm Weg am Schlitten der Antriebseinheit betragen. Um aber auch dies auszuschliessen, findet eine Counterinitialisierung statt.

25	// // Counterinitialis void counter_init {	ierung void)
30	TR0 = 1; TR1 = 1;	// Timer 0 einschalten in SFR->TCON // Timer 1 einschalten in SFR->TCON
35	$TMOD = 0 \times 66;$	// 0110 0110b in TMOD // Timer 0 und Timer 1 in Counterbetrieb und Mode 2
55	THO = 0xFO; TH1 = 0xFO;	<pre>// Timer 0: reload-Register setzen (auf Wert 240d) // Timer 1: reload-Register setzen (auf Wert 240d)</pre>
40	TLO = 0xFO;	// Timer 0: // Zählregister einmalig auf Startwert setzen (240d)
45	TL1 = 0xF0;	// Timer 1: // Zählregister einmalig auf Startwert setzen (240d)
50	//	ormationen: 6 Impulse/Umdrehung, deshalb 255-16=239. uls ist eine Umdrehung vollständig -> 239+1=240 hler Motor +1 nterruptauslösung pro Motorumdrehung rrupt Umdrehungszähler aktualisieren
55	}	

Nach Initialisierung der Variablen für die Aktualwerte der Umdrehungszähler in der Datei *umdrehungszaehler.c* wird die Funktion *counter\_init { }* in der Datei *counter\_init.c* aufgerufen.

In dieser Funktion werden Timer 0 und Timer 1 eingeschaltet durch Setzen von TR0 und TR1 im SFR TCON. Timer 0 und Timer 1 arbeiten im Counterbetrieb Mode 2.

Die Zählregister von Timer 0 und Timer 1 werden mit dem Startwert 240d geladen. Nach 16 Impulsen des Inkrementalgebers ist eine Motorumdrehung vollständig und es findet ein Überlauf des Counters statt. Durch diesen Counterüberlauf wird ein Interrupt ausgelöst, welcher die Aktualwerte der entsprechenden Umdrehungszähler inkrementiert.

105	//
	<pre>// Umdrehungszähler Motor inkrementieren // jeden 16. Impuls wird Umdrehungszähler Motor akt. m umdr al++;</pre>
110	m_umdr_hilfsm_al++;
115	<pre>// Umdrehungszähler Spindel inkrementieren // alle GEAR N Motorumdrehungen wird Umdrehungszähler Spindel aktualisiert. // Hilfsmerker Umdrehungszähler Motor wird gelöscht. if (m umdr hilfsm al == GEAR N 23 )</pre>
	<pre>s_umdr_al++; m_umdr_hilfsm_al=0;</pre>
120	}

#### Weitere Informationen

Nicht zu diesem Modul gehörend, aber der Vollständigkeit und dem besseren Verständnis wegen wird hier noch verdeutlicht, wie die Umsetzung zum Erlangen der Aktualwerte der Umdrehungszähler für die Spindeln realisiert ist.

Nach jedem Inkrementieren der Motorumdrehungszähler wird überprüft, ob eine komplette Spindelumdrehung erreicht wurde. Das Übersetzungsverhältnis des Getriebes ist als DEFINE-Wert in der Headerdatei *umdrehungszaehler.h* hinterlegt.

Eine Änderung der Getriebeübersetzung bringt somit keinen großen Programmieraufwand mit sich. Es muß lediglich der DEFINE-Wert in *umdrehungszaehler.h* an die entsprechende Getriebeübersetzung angepasst werden.

## 9.4.6 Modul 5: Motoren Leistung ein

In diesem Modul wird aus der Datei *main.c* die Funktion *mot\_pwr\_enable { }* in der Datei *mot\_outp.c* aufgerufen.

Hier werden die Motorsteuer-ICs generell eingeschaltet. Zu beachten ist allerdings, dass ein enable-Signal mit einer logischen 0 erreicht wird.

#### 9.4.7 Modul 6: LCD rücksetzen

Dieses Modul wird aus der Hauptroutine angestoßen. Als Hauptroutine wird hier eine while (1) - Schleife bezeichnet.

Zuerst wird überprüft, ob die Hauptroutine nach dem Einschalten der Netzversorgung oder dem Auslösen eines Neustarts des Mikrocontrollers das erste Mal durchlaufen wird. Dies wird mit Hilfe eines flags "first\_turn\_lcd" realisiert.

Als Sicherheit werden weitere Bedingungen mit diesem flag verknüpft.

```
// Hauptroutine
while(1)
(
    // -----
190     // nach Reset der Steuerung Display zurücksetzen
    // Makro 1 aufrufen
    if((!first_turn_lcd) || ((!Pl_3) && (!m_all_ax_bb) && (err_neig_ntact == 1)))
    {
        befehl_s1_in[1] = 0;
        transmit_S1_vorbereitung('N',0x01);
        first_turn_lcd=1;
    }
}
```

Ist diese Bedingung erfüllt, wird über die serielle Schnittstelle S1 das Normalmakro 1 des LCD-Displays aufgerufen. Dies hat einen Neustart der HMI zur Folge.

Weitere Informationen zur Programmierung der HMI erhalten Sie in den entsprechenden Kapiteln. Zum besseren Verständnis sollten diese Kapitel eventuell parallel zu Kapitel 9.4 eingesehen werden. Dort erhalten Sie u. A. Informationen zur Programmierung mit Makros der HMI.

## 9.4.8 Modul 7: Überprüfung Deckelschalter

Die Überprüfung des Deckelschalters wird bei jedem Zyklus der Hauptroutine durchgeführt. Über den Aufruf der Funktion *check\_hut {}* wird diese Überprüfung gestartet.

Die Funktion *check\_hut {}* finden Sie in der Datei *input.c.* 

Im oberen Rahmen des Messautomaten finden Sie den Reedkontakt mit der Bezeichnung SE1.5. Die Bezeichnung ist normgerecht gewählt und verweist auf den Eingang 1.5 der Steuerung.

In diesem Fall entspricht der Eingang 1.5 der Steuerung, dem Portpin 1.5 des Mikrocontrollers. Dieser Portpin ist ein digitaler Eingang DI.

Im Deckel des Messautomaten sind zwei Magnete eingearbeitet, die auch ein versehentlich falsches Aufsetzen des Deckels auf den Automaten erkennen.

Eine entsprechende Visualisierung erfolgt über die HMI.

Ist der Eingang 1.5 des Mikrocontrollers auf logisch 0, wird ein Fehlerbit gesetzt. Über dieses Fehlerbit wird der Aufruf zur Fehlerdiagnose und den entsprechenden Reaktionen auf den Fehler realisiert.



Die Fehlerauswertung findet in der Datei *error\_1.c* statt. Die entsprechenden Reaktionen auf die analysierten Fehler werden dort ebenso ausgelöst.

Ist in diesem Fall die Fehlernummer 15 aktiv, wird bei einer Anforderung zur Quittierung der Störung, die Fehlernummer zurückgesetzt, die Anforderung zur Quittierung wird zurückgesetzt und der Leistungsteil der Hauptplatine zur Ansteuerung der Antriebe wird wieder aktiviert.

Detailliertere Informationen zur Fehlerdiagnose erhalten Sie in den folgenden Kapiteln.

## 9.4.9 Modul 8: Überprüfung Neigungssensor

Die Überprüfung des Neigungssensors wird bei jedem Zyklus der Hauptroutine durchgeführt. Über den Aufruf der Funktion *check\_neigung { }* wird diese Überprüfung gestartet.

Die Funktion check\_neigung { } finden Sie in der Datei input.c.



Der Neigungssensor ist mit SE 1.3 bezeichnet. Er liegt somit an Portpin 1.3 des Mikrocontrollers.

Ist der DI 1.3 auf logisch 1, wird ein Fehlerbit gesetzt und die Diagnose in der Datei *error\_1.c* gestartet.

In der Funktion *check\_neigung { }* werden zusätzlich noch mehrere flags wie "err\_neig\_ntact" (Fehler Neigungssensor nicht aktiv) oder "lage\_nio" (Lage des Messautomaten nicht in Ordnung) verwaltet.

Die Reaktion auf das Ansprechen des Neigungssensors wird in der Datei error\_1.c ausgelöst.

## 9.4.10 Modul 9: Referenzpunktschalter einlesen

Das Einlesen der Referenzpunktschalter wird gestartet über den Aufruf der Funktion *check\_refp* {} in der Datei *main.c.* 

Das Einlesen der Referenzpunktschalter erfolgt zyklisch bei jedem Durchlauf der Hauptroutine.

Die Funktion *check\_refp* {} finden Sie in der Datei *input\_1.c.* 

Die Referenzpunktschalter liegen, wie Sie auch dem Pinbelegungsplan in Kapitel 8.2 oder dem Schaltplan der Hauptplatine bzw. dem Stromlaufplan des Messautomaten entnehmen können, an den Portpins 9.0 bis 9.3.

Der Referenzpunktschalter der Achse 1 ist mit SE 9.0, der Referenzpunktschalter der Achse 2 mit SE 9.1, der Referenzpunktschalter an Achse 3 mit SE 9.2 und der Referenzpunktschalter für Achse 4 ist mit SE 9.3 bezeichnet.

Da Port 9 des Mikrocontrollers nicht bitadressierbar ist, gestaltet sich das Auswerten der einzelnen Referenzpunktschalter etwas umfangreicher.

Zuerst wird Port 9 eingelesen und das PAE (Prozessabbild der Eingänge) in eine Variable geschrieben.

Das Bit des Referenzpunktschalters der Achse 1, DI 9.0, wird ausmaskiert und das Bitmuster in eine temporäre Variable geschrieben.

Ist das Bitmuster der temporären Variablen 0000 0001, wird der logische Zustand 1 des Referenzpunktschalters der Achse 1 in das array refp[] geschrieben.

Bei jedem anderem Bitmuster der temporären Variablen, wird im array refp[] der logische Zustand 0 des Referenzpunktschalters der Achse 1 gespeichert.

Das array refp[] wird wie folgt genutzt :

- der Zustand des Referenzpunktschalters der Achse 1 wird abgelegt in refp[0].
- der Zustand des Referenzpunktschalters der Achse 2 wird abgelegt in refp[1].
- der Zustand des Referenzpunktschalters der Achse 3 wird abgelegt in refp[2].
- der Zustand des Referenzpunktschalters der Achse 4 wird abgelegt in refp[3].

Nach dem speichern des PAE und der Aktualisierung des Zustandes von DI 9.0 in refp[0], wird die Aktualisierung des Zustandes für den Referenzpunktschalter der Achse 2 in refp[1] vorgenommen.

```
// Refp Achse 2 ausmaskieren
tmp=(al234_refp & 0x02);

80 // Zustand Refpschalter Achse 2 in Speicher schreiben
if(tmp == 0x02)
{
    refp[1]=1;
}
85 else
{
    refp[1]=0;
}
```

Anschließend wird mit dem aktualisieren der Schaltzustände der Referenzpunktschalter von Achse 3 und 4 verfahren, wie bereits bei den Achsen 1 und 2 geschehen.

```
// Refp Achse 3 ausmaskieren
tmp=(al234_refp & 0x04);
// Zustand Refpschalter Achse 3 in Speicher schreiben
if (tmp == 0x04)
{
    refp[2]=1;
}
else
100 {
    refp[2]=0;
}
105 // Refp Achse 4 ausmaskieren
tmp=(al234_refp & 0x08);
// Zustand Refpschalter Achse 4 in Speicher schreiben
if (tmp == 0x08)
110 {
    refp[3]=1;
}
else
{
    refp[3]=0;
}
```

#### 9.4.11 Modul 10: Referenzpunktfahrt

In der Hauptroutine in *main.c* wird überprüft, ob eine Referenzpunktfahrt einzuleiten ist. Ist der Merker "m\_all\_ax\_bb" (alle Achsen betriebsbereit) im Zustand logisch 0, wird eine Referenzpunktfahrt eingeleitet.

Um zu verhindern, dass der Messautomat während des Aufbauens zur Messung durch Verbinden mit der Versorgungsspannung eine Referenzpunktfahrt durchführt, ist diese Anforderung zusätzlich mit der Bedingung verbunden, dass der Neigungssensor den Messautomat in Arbeitsstellung lokalisiert.

```
// -----
// Wenn noch nicht alle Achsen betriebsbereit sind, Referenzpunktfahrt
// anfordern in grundstellung_l.c
if((!m_all_ax_bb) && (!lage_nio))
215 {
    // Aufruf Referenzpunktfahrt
    referieren_haupt();
}
```

Ist diese Bedingung erfüllt, wird der Aufruf zur Referenzpunktfahrt in grundstellung\_1.c ausgelöst.

In der Funktion *referieren\_haupt { }* wird zuerst überprüft, welche Achsen eine Referenzpunktfahrt durchzuführen haben.



Eine Referenzpunktfahrt ist notwendig, um eine eindeutige mechanische Position zu einem Bezugspunkt herzustellen. Die Positionierungen der Achsen werden in Bezug auf diesen Referenzpunkt hergestellt.

Für die Referenzpunktfahrt ist außerdem entscheidend, ob die Achse, bzw. der Schlitten der Antriebseinheit bereits den entsprechenden Referenzpunktschalter bedämpft hat.

Entsprechend dieser Überprüfung wird eine von zwei möglichen Arten zum Justieren der Antriebseinheit gewählt.

## Möglichkeit 1:

Der Schlitten der Antriebseinheit hat den Referenzpunktschalter bereits bedämpft.

Justieren 1 wird eingeleitet.

- Der Schlitten wird um den DEFINE Wert S\_U\_SOLL in grundstellung\_1.h (zum Zeitpunkt des Schreibens der Dokumentation entspricht S\_U\_SOLL dem Wert 5) Spindelumdrehungen in Richtung (+) verfahren.
- Nach dem Erreichen dieser Position läuft eine kurze Wartezeit ab, bevor die Richtungsumkehr eingeleitet wird. Diese Wartezeit ist notwendig, um die Überlastüberwachung durch die Motorsteuer-ICs nicht ansprechen zu lassen. Die Wartezeit entspricht in etwa 0,2 s.
- Der Schlitten der Antriebseinheit wird nun in Richtung (-) bis zum Erreichen des Referenzpunktschalters bewegt.
- Beim Erreichen des Referenzpunktschalters wird die Achse als betriebsbereit deklariert.

## Möglichkeit 2:

Der Schlitten der Antriebseinheit hat den Referenzpunktschalter zum Zeitpunkt der Anforderung der Referenzpunktfahrt nicht bedämpft.

Justieren 2 wird eingeleitet.

- Der Schlitten der Antriebseinheit wird in Richtung (-) bis zum Erreichen des Referenzpunktschalters bewegt.
- Beim Erreichen des Referenzpunktschalters wird die Achse als betriebsbereit deklariert

Beim Referieren wird grundsätzlich eine Reihenfolge eingehalten, in der die Achsen referiert werden.

Reihenfolge der Achsen beim Referieren:

- Achse 1
- Achse 2
- Achse 3
- Achse 4

Um diese Reihenfolge einhalten zu können, sind mehrere Bedingungen abzufragen. Die Funktion *referieren\_haupt {}* stellt somit eine Sprungtabelle dar.

```
// Achse 2
            if(((refp[1] == true)
                  (lterp[1] -- the)
&&(!refriert[1])
&&(!flag_ref2_a2)
&&(refriert[0] == true))
||(flag_ref1_a2 == true))
130
           {referieren_1_a2(s_umdr_a2);}
135
           if((!referiert[1])
            &&(!flag_ref1_a2)
&&(referiert[0]))
{referieren_2_a2();}
140 //-----
      // Achse 3
            if(((refp[2] == true)
               &&(!referiert[2])
           145
150
            if((!referiert[2])
          % (referierc[1])
% (referierc[1] a3)
% (referierc[1] == true)
% (referierc[1] == true))
(referiercn_2_a3();)
155
      // Achse 4
          160
                   && (referiert[0] == true)
&& (referiert[1] == true)
&& (referiert[2] == true))
165
                   (flag_ref1_a4))
           {referieren_1_a4(s_umdr_a4);}
           if((!referiert[3])
                  (!referiert[3])
&& (!flag_refl_a4)
&& (referiert[0] == true)
&& (referiert[1] == true)
&& (referiert[2] == true))
for (erger 2 = cf(1))
170
           {referieren_2_a4();}
175
```

Aus dem Programmauszug ist ersichtlich, dass eine Achse nur die Freigabe zur Referenzpunktfahrt erhält, wenn die Achsen, die in der Sprungtabelle vorrangig sind, bereits als referiert deklariert sind.

Um die Achsen als referiert zu deklarieren, werden folgende Einträge in das eindimensionale array *referiert* [] notwendig:

- logisch 1 in referiert [0] : Achse 1 ist referiert.
- logisch 1 in referiert [1] : Achse 2 ist referiert.
- logisch 1 in referiert [2] : Achse 3 ist referiert.
- logisch 1 in referiert [3] : Achse 4 ist referiert.

Ein Programmauszug zur Justage 1 der Antriebseinheit 1. Nachfolgend die einzelnen Schritte des Ablaufes.

185	/ oid referieren 1 al (signed char s umdr al)
190	<pre>// Wenn Motor Al R+ noch nicht läuft, dann einschalten // Geberdatenkanal wird autom. über mot_al_plus() ausgewählt if ((!m_mot_al_plus) &amp;&amp; (!m_mot_al_minus)) {</pre>
	<pre>// Motor Al Richtung+ ein mot_al_plus(); }</pre>
195	<pre>// Wenn Spindelumdrehungen - Sollwert, dann Motor stop, // anschließend Motor Al R- if (s_umdr_al S_U_SOLL) (</pre>
200	<pre>mot_al_stp(); if (change_pm_a1 -= 0) (</pre>
205	<pre>change pm_al=1; ax_walt();</pre>
21.0	<pre>// Motor Al Richtung- ein mot_al_minus(); }</pre>
220	<pre>// Wenn Referenzpunktschalter angefahren ist, dann Bit für Achse "ist // referiert" setzen und Motor ausschalten if ((refp[0] 1) &amp;&amp; (m_mot_al_minus))</pre>
215	<pre>// Achse 1 ist referiert referiert[0]=1;</pre>
220	<pre>// Flag "referieren_1_Achse_1_läuft" rücksetzen flag_ref1_al=0; // Achse_1_ist_betriebsbereit_setzen</pre>
	// Motor Al ausschalten
225	<pre>mot_al_stp(); // alle Zähler rücksetzen</pre>
230	<pre>umdr_zaehler_init_al(); } else (</pre>
	<pre>// Flag "referieren_1_Achse_1_läuft" setzen flag_ref1_a1=1; )</pre>
235	

- Wenn der Antrieb noch nicht in Bewegung ist, dann wird der Antriebsmotor in Richtung (+) eingeschaltet.
- Der Istwert des Spindelumdrehungszählers wird mit dem vorgegebenen Sollwert verglichen.
- Ist der Sollwert erreicht, wird der Motor gestoppt.
- Überprüfung, ob Wartezeit zur Drehrichtungsumkehr bereits angefordert wurde.
- Nach Ablauf der Wartezeit wird der Motor in Richtung (-) eingeschaltet.
- Ist der Motor in Richtung (-) eingeschaltet und der Referenzpunktschalter wird erreicht, dann wird die Achse als referiert deklariert.
- Flag : Justage 1 Achse 1 aktiv, wird zurückgesetzt
- Achse 1 wird betriebsbereit gesetzt
- Motor Achse 1 wird gestoppt
- Umdrehungszähler der Achse 1 wird initialisiert

Wie dem nachfolgenden Programmauszug zu entnehmen ist, ist die Justage 2 weniger umfangreich als das Verfahren Justage 1.

Auf eine Erklärung der einzelnen Schritte wird verzichtet, diese sind ähnlich dem Verfahren der Justage 1 und können deshalb der Erklärung zu Justage 1 entnommen werden.

Eine ausreichende Kommentierung der einzelnen Befehlszeilen wurde ebenso vorgenommen.

Diese Programmblöcke wiederholen sich natürlich für Achse 2, Achse 3 und Achse 4 entsprechend.

```
// Merkerbit für "alle Achsen sind referiert" setzen + rücksetzen
// Merkerbit für "alle Achsen betriebsbereit" setzen + rücksetzen
485
      void all_ax_bb_ref(void)
            if ({referiert[0]
                 && (referiert[1]
&& (referiert[2]
490
                  &&(referiert[3]))
                  m_all_ax_ref=1;
                 change_pm_al=0;
change_pm_a2=0;
change_pm_a3=0;
change_pm_a4=0;
495
                ((m_all_ax_ref)
&&(ax_bb[0])
&&(ax_bb[1])
&&(ax_bb[2])
            i.f
500
                  &&(ax bb[3]))
505
                 m_all_ax_bb=1;
                 // Bild Betriebsartenwahl aufrufen
transmit_S1_vorbereitung('N',0x03);
510
        / Wartezeit für Richtungsumkehr der Spindel bei referieren_1 zur
     // Strombegrenzung
515
       void ax_wait(void)
            for(ax_wait_time=0;ax_wait_time<=20000;ax_wait_time++)</pre>
520 }
```

Nach dem Referieren werden Merker aktualisiert, die zur weiteren Verwendung in anderen Programmabschnitten zur Verfügung stehen.

In der Funktion *ax\_wait { }* die erforderliche Wartezeit zur Strombegrenzung bei Drehrichtungsumkehr der Spindel.

## 9.4.12 Modul 11: Fehlergenerierung

Das System der Fehlergenerierung beruht auf drei Schritten:

- Das Erkennen des Fehlers
- Das Erzeugen einer Fehlernummer
- Die auszulösende Reaktion auf den anstehenden Fehler

Grundsätzlich ist dieses Modul zur Fehlergenerierung einfach gestaltet. Die einzelnen Funktionen, die entweder zyklisch oder durch Interrupts aufgerufen werden, überprüfen durch einlesen von Ports oder durch auslesen von Werten in Variablen, den Messautomat auf Fehlerzustände.

Ein Programmauszug zur Generierung der Fehlernummer 1 und Fehlernummer 24:



Fehlernummer 1 wird generiert, wenn der Messautomat in Betriebsart Automatik oder Betriebsart Teilautomatik in unzulässigen Neigungswinkeln verfährt. Der zulässige Arbeitsbereich wird durch einen Neigungssensor überprüft.

Verlässt der Messautomat seinen zulässigen Arbeitsbereich, so wird in die Variable "err\_number" der Wert 1, also die Fehlernummer 1 eingetragen.

Die Variable "err\_number", die nun die Fehlernummer enthält, wird an die Funktion *error\_break {}* übergeben.

Dieses Verfahren wird für die Generierung aller Fehler im Projekt angewendet. Kernstück dieser Fehlergenerierung ist die Datei *error\_1.c.* 

Nach Übergabe der Fehlernummer an die Funktion *error\_break { }* wird die Reaktion auf die entsprechende Fehlernummer vorbereitet und anschließend ausgeführt.

Die Reaktion auf die ausgewertete Fehlernummer ist entsprechend dem Fehler angepasst. So wird als Reaktion auf einen erkannten Fehler nur eine Meldung auf der HMI ausgegeben oder es werden weitere Funktionen, die zur Reaktion auf die generierte Fehlernummer benötigt werden, aufgerufen.

-	
	<pre>// Paarfehler, Achse 4 if (number == 5) </pre>
425	<pre>// LCD-Bild aufrufen // Makro 23 bl='N'; b2=0x17; transmit S1 vorbereitung(b1,b2);</pre>
430	}
135	<pre>//</pre>
400	// Interrupt 4 (Endlagenüberwachung) ausschalten EX4=0;
440	// Ausgänge Motoren zurücksetzen mot_al_stp();
	// Motoransteuerungen disable mot_pwr_disable{};
445	<pre>// Variable initialisieren init_var();</pre>
450	<pre>// Ausgänge initialisieren init_outp();</pre>
400	<pre>// LCD-Bild aufrufen // Makro 28 bl='N'; b2=0x1C;</pre>
455	<pre>transmit_S1_vorbereitung(b1,b2);</pre>

Im Programmauszug wird die Reaktion auf die Fehlernummer 5 und Fehlernummer 6 dargestellt.

Fehlernummer 5 wurde erzeugt durch einen Paarfehler der Endlagenüberwachungen von Achse 4 und löst als Reaktion einen Aufruf des Makros 23 der HMI aus.

Fehlernummer 6 wurde erzeugt durch einen erkannten Endlagenfehler an Achse 1. Der Schlitten der Antriebseinheit 1 bedämpfte den Endlagenschalter Richtung (-) der Achse 1.

- Als Reaktion auf diesen erkannten Fehler wird die Überwachung zuerst deaktiviert. Dieser Vorgang lässt ein späteres Freifahren der Achse zu.
- Der Antriebsmotor der Achse 1 wird stillgesetzt.
- Alle Achsbewegungen werden durch Abschalten des Leistungsteils verriegelt.
- Entsprechende Variablen werden initialisiert, um flags von aktiven Positionierbewegungen rückzusetzen.
- Ausgänge werden initialisiert, um selbsttätiges Anlaufen eines Antriebes bei Wiederkehr der Freigabe des Leistungsteils zu verhindern.
- Eine Meldung zur Bedienerführung wird auf der HMI durch Aufruf des Makros 28 angefordert.

Bei allen erzeugten Fehlern wird ebenso dem Tool OLLICONTROL (OC) die Freigabe zur Bedienung des Messautomaten entzogen.

Dies wird selbstverständlich auf OC als zu quittierende Meldung angezeigt.

```
// Bei Störung Freigabe von Olli-Control entziehen
795 if(err_number!=0)
{
    error_activ=1;
    if((error_activ == 1) && (first_turn_err == 0))
    {
        olli_control_disable();
        first_turn_err=1;
    }
}
```

Nach Beseitigung und Quittierung des Fehlers über die HMI des Messautomaten wird OC wieder, unter Berücksichtigung weiterer Bedingungen, die Freigabe zum Bedienen des Messautomaten erteilt.

```
985 // tool Olli-Control Freigabe erteilen
void olli_control_enable(void)
{
    transmit_S0_vorbereitung(8);
}
990
// -----
// tool Olli-Control Freigabe entziehen
void olli_control_disable(void)
{
    transmit_S0_vorbereitung(7);
}
```

Für die Aktivierung bzw. die Deaktivierung der Freigabe wird ein entsprechender Code über die serielle Schnittstelle S0 zu OC gesendet.

## 9.4.13 Modul 12: Betriebsart Automatik

In der Hauptroutine in *main.c* wird überprüft, ob die Betriebsart Automatik angefordert wurde.

```
// Betriebsart Automatik angewählt
// Springe nur ein, wenn alle Achsen referiert und in Grundstellung
// sind.
if(m_al_ax_ref == 1)
{
// BA Auto angewählt
if(m_ba_auto == 1)
{
// Aufruf Betriebsart Automatik in ba_auto.c
ba_auto();
}
```

Die Betriebsart Automatik wird erst aktiv, wenn alle Achsen referiert und somit in Betriebsbereitschaft sind.

Der eigentliche Automatikablauf wird in der Datei ba\_auto.c verwaltet.

In der Betriebsart Automatik werden die 5 Messpositionen automatisch nacheinander angefahren. Das Anfordern der jeweiligen nächsten Messposition wird von OC bzw. dem Hasheraccelerator übernommen. Details hierzu erhalten Sie im Kapitel Kommunikation.

Das Positionieren des Messautomaten auf die jeweils nächste Messposition erfolgt über das Fahren auf die Grundstellung.

Folgende Positionen werden in der Betriebsart Automatik nacheinander angefahren:

- Grundstellung
- Position 1
- Grundstellung
- Position 2
- Grundstellung
- Position 3
- Grundstellung
- Position 4
- Grundstellung
- Position 5

#### Grundstellung

Nach jedem Erreichen der Grundstellung initialisieren sich die verschiedenen Umdrehungszähler der entsprechenden Achsen. Durch diese Maßnahme wird die größtmöglichste Positioniergenauigkeit erreicht.

Zum Erreichen einer Messposition verfahren maximal 2 Achsen gleichzeitig. Im folgenden Programmauszug wird die Positionierung des Messautomaten zu Messposition 4 dargestellt. Um diese Messposition zu erreichen, muss lediglich Achse 2 in Richtung (+) auf die geforderte Position bewegt werden.

```
// Position 4: Achse 2 in R+
if((move_order_pos == 4) && (!p6_fahrt_aktiv))
              if(pos_erreicht_p4 == 0)
320
                   if((!m_mot_a2_plus) && (!m_mot_a2_minus))
                       p4_fahrt_aktiv=1;
325
                       // Motor A2 Richtung+ ein
mot_a2_plus();
                        // Meldung LCD ausgeben
                       ba_a_msg_p4_aktiv();
330
                   // Wenn Spindelumdrehungen = Sollwert, dann Motor stop
                   if (s_umdr_a2 == S_U_SOLL_A2)
                       mot_a2_stp();
p4 fahrt aktiv=0;
335
                        move_order_pos=0;
transmit_S0_vorbereitung(4);
                        // LCD-Meldung aufrufen
340
                       ba_a_msg_p4();
                   if(m_mot_a2_plus == 0)
345
                       pos_erreicht_p4=1;
350
```

Die angeforderte Position wird in der Variablen "move\_order\_pos" hinterlegt. Die Bewegung zur Positionierung auf die geforderte Messposition Nummer 4 wird erst gestartet, wenn die Grundstellungsfahrt nicht mehr aktiv ist.

Ist Position 4 noch nicht erreicht, der Antriebsmotor der Achse 2 noch nicht eingeschaltet, wird zuerst das flag "p4\_fahrt\_aktiv" gesetzt. Das Setzen dieses flags bedeutet, dass die Fahrt auf Position 4 nun aktiv ist.

Anschließend wird der Antriebsmotor der Achse 2 in Drehrichtung (+) eingeschaltet. Zusätzlich wird eine Meldung zur Bedienerführung auf der HMI abgesetzt.

Der Istwert des Spindelumdrehungszählers wird mit dem Sollwert verglichen. Ist der Sollwert erreicht, wird der Antriebsmotor der Achse 2 stillgesetzt und das flag, welches signalisiert, dass die Fahrt auf Position 4 aktiv ist, zurückgesetzt.

Der aktive Fahrbefehl, der in der Variablen "move\_order\_pos" hinterlegt ist, wird zurückgesetzt.

Anschließend erfolgt eine Bestätigung an OC bzw. den Hasheraccelerator, dass die angeforderte Position erreicht ist. Hierzu wird ein Code über die serielle Schnittstelle S0 an den entsprechenden Adressaten gesendet.

Es erfolgt ebenso nach dem Erreichen der Messposition 4 eine Meldung zur Bedienerführung auf der HMI. Hierzu wird die Funktion *ba\_a\_msg\_p4 { }* (Betriebsart Automatik message Position 4) aufgerufen.

Dies Funktion bewirkt einen Makroaufruf in der HMI, welcher zur Folge hat, dass ein entsprechendes Bild auf dem LCD-Touchpanel angezeigt wird.

Unterschiedlich zum vorherigen Programmauszug stellt sich der Folgende dar. Zur Fahrt auf Messposition 2 werden zwei Achsen bewegt. Prinzipiell ist der Ablauf der Gleiche wie zur Positionierung mit einer Achse.

```
// Position 2: Achse 1 in R+, Achse 4 Mitte
if((move_order_pos == 2) && (!p6_fahrt_aktiv))
             if(pos_erreicht_p2 == 0)
                 p2_fahrt_aktiv=1;
                      // Motor A4, A1 Richtung+ ein
mot_a4_plus();
mot_a1_plus();
230
                      // Meldung LCD ausgeben
ba_a_msg_p2_aktiv();
                  // Wenn Spindelumdrehungen = Sollwert, dann Motor stop
if (s_umdr_a4 == S_U_SOLL_A4_M)
                      mot_a4_stp();
240
                  if (s_umdr_a1 == S_U_SOLL_A1)
                      mot_a1_stp();
245
                      move_order_pos=0;
transmit_S0_vorbereitung(2);
                       // LCD-Meldung aufrufen
                      ba_a_msg_p2();
                  if((m_mot_a4_plus == 0) && (m_mot_a1_plus == 0))
260
                      pos_erreicht_p2=1;
```

Im Beispiel wird Messposition 2 angefahren über die Bewegung durch Achse 1 in Richtung (+) und die Bewegung der Achse 4 ebenso in Richtung (+).

Die Verriegelungen gestalten sich ähnlich dem Ablauf durch Positionierung über eine Achse, deshalb wird hier nicht mehr detailliert darauf eingegangen. Es ist lediglich darauf zu achten, dass Ein-/Ausschaltbedingungen nun für beide zu bewegende Achsen notwendig sind.

Das flag, das die aktive Fahrt zur Messposition signalisiert, wird von der Achse zurückgesetzt, die als letzte Achse ihre Sollposition erreicht. Somit wird gewährleistet, dass entsprechende Meldungen zur Bedienerführung an die HMI erst abgesetzt werden, wenn der Messautomat richtig positioniert ist.

Eine Akustikmessung zu einem zu frühen Zeitpunkt und somit einer fehlerhaften Akustikmessung wird somit verhindert.

## 9.4.14 Modul 13: Betriebsart Teilautomatik

Wie für die Betriebsart Automatik wird für die Betriebsart Teilautomatik in der Hauptroutine in *main.c* überprüft, ob die Betriebsart Teilautomatik angefordert wurde.

Wie auch in die Betriebsart Automatik darf in die Betriebsart Teilautomatik nur gewechselt werden, wenn alle Achsen des Messautomaten referiert und somit betriebsbereit sind. Eine entsprechende Bedienerführung wird auf der HMI ausgegeben.

Wird die Betriebsart Teilautomatik von OC bzw. dem Hasheraccelerator angefordert, findet die Koordination des Ablaufes ebenso wie bei der Betriebsart Automatik in der Funktion *ba\_auto {}* der Datei *ba\_auto.c* statt. Es wird somit das Modul 12 mit in Anspruch genommen.

Findet die Anforderung der Betriebsart Teilautomatik durch die HMI statt, wird die Funktion ba\_teilauto { } in Datei ba\_teilauto.c verwendet.

```
// Position 1: Olli Mitte, Achse 4 Mitte
if((neu_anzufahrende_pos == 1) && (!ta_p6_fahrt_aktiv))
                if(pos erreicht pl == 0)
185
                    if ((!m_mot_a4_plus)
                         && (!m_mot_a4_minus)
&& (!m_mot_a1_minus)
&& (!m_mot_a2_minus)
190
                         &&(!m_mot_a3_minus))
                         ta_pl_fahrt_aktiv=1;
                         // Motor A4 Richtung+ ein
195
                         mot_a4_plus();
                         // Meldung LCD ausgeben
ba_ta_msg_pl_aktiv();
200
                    // Wenn Spindelumdrehungen = Sollwert, dann Motor stop
                    if (s_umdr_a4 == S_U_SOLL_A4_M)
                         mot_a4_stp();
ta_p1_fahrt_aktiv=0;
205
                         neu_anzufahrende_pos=0;
                         // Meldung auf LCD : Position 1 erreicht
                         msg_pl();
210
                    if(m_mot_a4_plus == 0)
                         pos_erreicht_p1=1;
215
```

Der Quellcode ist prinzipiell gleich dem Quellcode der Betriebsart Automatik. Dies wird deutlich im Programmauszug zur Bewegung des Messautomaten zur Position 1.

Unterscheidungen ergeben sich hauptsächlich in der Namensgebung der Variablen. Dies ist eine Folge der Kommunikation, die in weiteren Kapiteln später detailliert dargestellt wird.

Meldungen zur Bedienerführung werden ebenso auf der HMI wie in der Betriebsart Automatik ausgegeben.

Eine Anforderung zum Positionswechsel erfolgt nicht mehr automatisch nach dem Erreichen der vorherigen Position. In der Betriebsart Teilautomatik wurde eine Möglichkeit geschaffen, die verschiedenen Messpositionen in beliebiger Reihenfolge anzufahren.

Ein Anfahren der Messpositionen erfolgt jedoch immer, wie auch in Betriebsart Automatik, über die Grundstellung.

Die Anforderung zum Positionswechsel erfolgt entweder über OC oder den Hasheraccelerator, außerdem kann ein Positionswechsel über die HMI angefordert werden. Informationen hierzu erhalten Sie in den Kapiteln zur HMI.

Ebenso wie in der Betriebsart Automatik wird auf die Rückmeldungen an die entsprechenden Adressaten beim Erreichen einer angeforderten Position nicht verzichtet.

Eine Fehlergenerierung und Diagnose wie bereits im Kapitel zu Modul 11 beschrieben, findet ebenso statt.

#### 9.4.15 Modul 14: Betriebsart Hand

Wie auch bei anderen Betriebsarten, wird in der Hauptroutine in *main.c* überprüft, ob eine Anforderung der Betriebsart Hand ansteht.

Die einzelnen Fahrbefehle, die in der Betriebsart Hand möglich sind, sind in der Funktion ba\_hand {} der Datei ba\_hand.c hinterlegt.

Die Betriebsart Hand ist ausschließlich von der HMI anzufordern. Die Freigaben zur Bedienung des Messautomaten durch OC bzw. den Hasheraccelerator werden dadurch gesperrt.

Die Betriebsart Hand dient dazu, einzelne Achsen des Messautomaten schrittweise zu verfahren. Die Schrittgröße wird über einen DEFINE-Wert in der Headerdatei ba\_hand.h eingestellt. Eine Schrittgröße von 2 Spindelumdrehungen, dies entspricht einem linearen Verfahrweg von 2 mm, wird hier für sinnvoll gehalten.

	//
55	// Ablauf Tippen in Betriebsart Hand void ba_hand (void)
	<pre>// Fahrbefehl 1, Codierung in Kommunikation_Auswertung.c if(ba_h_fahrbef == 1) </pre>
60	<pre>if(pos_erreicht_tipp_al_plus == 0) </pre>
	<pre>if((!m_mot_al_plus) &amp;&amp; (!m_mot_al_minus)) {</pre>
65	<pre>mot_al_plus(); }</pre>
	<pre>// Wenn Spindelumdrehungen = Sollwert, dann Motor stop if (s_umdr_al == S_U_SOLL_A1_TIPP)</pre>
70	<pre>mot_al_stp(); umdr_zaehler_init_al();</pre>
75	<pre>if (m_mot_al_plus == 0) {     pos_erreicht_tipp_al_plus=1;     ba_b_fahrbef=0;</pre>
	)
80	}

Im Programmauszug wird das Verfahren der Achse 1 in Drehrichtung (+) dargestellt. Die Fahrbefehle werden in der Variablen "ba\_h\_fahrbef" gespeichert. Detaillierte Informationen hierzu in den Kapiteln zur Kommunikation.

- Der Verfahrbefehl Nummer 1 signalisiert die Anforderung der Bewegung der Achse 1 in Drehrichtung (+).
- Ist die angeforderte Position noch nicht erreicht und der Antriebsmotor nicht angesteuert, wird der Antrieb in Drehrichtung (+) bewegt.
- Der Istwert des Spindelumdrehungszählers wird mit dem Sollwert in "S\_U\_SOLL\_A1\_TIPP" (Spindelumdrehungen Sollwert Achse 1 Tippbetrieb) verglichen.
- Ist der Sollwert erreicht, wird der Antrieb gestoppt und die Umdrehungszähler der Achse 1 werden initialisiert.
- Ist die angeforderte Position erreicht, wird ein dies in einem flag gespeichert und der aktuelle Fahrbefehl wird gelöscht.

Die Fahrbefehle werden in der Datei *Kommunikation\_Auswertung.c* decodiert. Um alle 4 Achsen in jeweils Drehrichtung (+) und Drehrichtung (-) verfahren zu können, stehen 8 Fahrbefehle zur Verfügung.

Je nach ausgelesenem Fahrbefehl, wird die entsprechende Achse in die geforderte Drehrichtung bewegt.

In der Betriebsart Hand steht wie auch in der Betriebsart Automatik und Betriebsart Teilautomatik die Bedienerführung der HMI zur Verfügung.

Eine Diagnose findet ebenso über die Datei *error\_1.c* statt.

#### 9.4.16 Modul 15: Achse freifahren

Das Freifahren der Achsen von einer Endlagenposition ist in der Datei *achse\_freifahren.c* realisiert.

Der Aufruf der Funktion ax\_free {} findet in der Hauptroutine in main.c statt.

```
// Anforderung von LCD
if(achse_freifahren == 1)
{
    // Aufruf Achse von Endlage freifahren in achse_freifahren.c
    ax_free();
}
```

Das Freifahren wurde programmiert, um eine Achse, die eventuell durch ein defektes Wegmesssystem, auf eine Arbeitsbereichsendlage gefahren ist, automatisiert von dieser Endlage wieder freizufahren.

Ebenso ist es möglich, in der Betriebsart Hand, die Achsen auf die Endlagen zu fahren. Um eine mechanische Beschädigung durch Fehlbedienung auszuschließen, wurde das Freifahren der Achse automatisiert.

Die Funktion *ax\_free { }* wird aufgerufen, nachdem ein entsprechender Fehler in der Datei *error\_1.c* diagnostiziert wurde.

Die im Modul 11 generierte Fehlernummer wird in der Funktion ax\_free { } aus der Variablen "err\_number" ausgelesen.

Die Betriebsarten Automatik und Teilautomatik werden gesperrt. Die Bedienung durch OC und den Hasheraccelerator ist nicht mehr möglich.

Der Entzug der Freigabe zur Bedienung wird entsprechend durch ein Meldungsfenster in OC angezeigt. Die Meldung ist quittierpflichtig.

Eine Bedienung des Messautomaten ist ausschließlich durch die HMI möglich. Detailinformationen hierzu im Kapitel zur HMI.

Der Ablauf des Freifahrens:

- Die Fehlernummer wird ausgelesen.
- ist der entsprechende Antriebsmotor noch nicht eingeschaltet, wird zuerst der Leistungsteil der Hauptplatine wieder freigegeben. Dieser wurde als Reaktion auf den diagnostizierten Fehler ausgeschaltet.
- Die Umdrehungszähler der entsprechenden Achse werden initialisiert.
- Der Antriebsmotor wird in die entsprechende Drehrichtung bewegt.
- Der Istwert des Spindelumdrehungszählers wird mit dem Sollwert verglichen.
- Ist der Sollwert erreicht, wird der Antriebsmotor gestoppt.
- Die Umdrehungszähler werden erneut initialisiert.
- Die Anforderung zum Freifahren der Achse wird zurückgesetzt.
- Die Überwachung der Endlagen wird wieder aktiviert. Dazu wird die Freigabe des Interrupts Nummer 4 wieder gesetzt. Die Überwachung wurde als Reaktion auf den diagnostizierten Fehler deaktiviert.
- Die Fehlernummer wird zur
  ückgesetzt.

Als mögliche Fehlernummer sind die Nummern 6 bis 13 reserviert.

Folgender Programmauszug macht deutlich, dass auch ein automatisiertes Freifahren der Achsen in Drehrichtung (+) möglich ist. Hierzu muss eine Antriebsachse die negative Endlage angefahren haben.

130	<pre>// von Endlage-, Achse 2 freifahren if(err_number == 8)</pre>
	if(!pos_erreicht_ax_free_a2_plus)
135	<pre>if((!m_mot_a2_plus) &amp;&amp; (!m_mot_a2_minus)) {     // Achse 2 freifahren in Richtung +     mot_pwr_enable();     umdr zaehler init a2();</pre>
140	<pre>mot_a2_plus();</pre>
145	<pre>// Wenn Spindelumdrehungen = Sollwert, dann Motor stop if (s_umdr_a2 == S_U_SOLL_AX_FREE) {     mot_a2_stp();     umdr_zaehler_init_a2();     achse_freifahren=0;</pre>
150	<pre>// enable INT4 EX4=1; }</pre>
155	<pre>if(m_mot_a2_plus == 0) {     pos_erreicht_ax_free_a2_plus=1;     err_number=0; }</pre>
160	} }

Grundsätzlich funktioniert das Freifahren der Achsen in Drehrichtung (+) auf gleiche Art und Weise wie das Freifahren der Achsen in Drehrichtung (-).

Auf eine detaillierte Beschreibung hierzu wird deshalb verzichtet.

## 9.4.17 Modul 16 + 17: Freigabecode an OC senden

Bei Störungen wird dem Tool OLLICONTROL bzw. Hasheraccelerator die Freigabe zur Bedienung des Messautomaten entzogen.

Um den Messautomat wieder über externe Software bedienen zu können, muss dieser Software die Freigabe zur Bedienung wieder erteilt werden.

300	//
	// Nachdem Fehler angestanden ist, Olli-Control wieder Freigabe senden
	// Störung wurde quittiert und Olli ist in Grundstellung
	if ((err number == 0)
	&& (error activ == 1)
305	&&(refp[0] == 1)
	&&(refp[1] == 1)
	&&(refp[2] == 1)
	&&(refp[3] == 1))
310	<pre>olli_control_enable();</pre>
	error_activ=0;
	first_turn_err=0;
	)

Dies darf erst geschehen, wenn keine Störung mehr diagnostiziert wird, alle vorherigen Störungen beseitigt und quittiert sind, und der Messautomat sich in Grundstellung befindet.

Über den Aufruf der Funktion *olli\_control\_enable { }* wird ein Freigabecode an den entsprechenden Adressaten gesendet.

```
OLLI – eine Holzkiste lernt laufen
```



Die Anbindung ist über die serielle Schnittstelle S0 realisiert.

Modul 16 und Modul 17 können hier zusammengefasst dargestellt werden, sie unterscheiden sich in der Funktion nicht voneinander.

Modul 16 dient der Erteilung der Freigabe an externe Software zur Bedienung des Messautomaten nach dem Start und der Initialisierung des Systems.

Modul 17 dient der Wiedererteilung der Freigabe an externe Software nach aufgetretenen Störungen im System.

## 9.4.18 Modul 18: Störung quittieren

Nach dem Grundsatz des Aufteilens des C-Projekts in Module, musste hier ein Modul deklariert werden, um explizit die Fehlernummer 1 zu quittieren, da dies in der Hauptroutine in *main.c* stattfindet.

Eine separate Behandlung dieses Fehlers findet aber nicht statt. Es wird hiermit auf das Kapitel zu Modul 20 verwiesen.

Eine Quittierung von quittierpflichtigen Störungen findet nur durch die Bedienung der HMI statt. Dazu finden Sie Hinweise in den Kapiteln zur HMI.

## 9.4.19 Modul 19: Fahrbefehle in Betriebsart Automatik auslesen

Die Datei move\_order.c stellt den Fahrbefehlsmanager der Betriebsart Automatik dar. In der Funktion check\_move\_order { } wird überprüft, ob ein neuer Fahrbefehl ansteht. Anschließend wird der Fahrbefehl ausgelesen.

```
void check_move_order(void)
35 (
        // Überprüfen, ob neuer Fahrbefehl vorhanden ist
if(m_new_move_order == true)
             // Fahrbefehle auslesen und abspeichern
40
             // Fahrbefehl für mechanische Position 1 auslesen
if (((befehl_s0_in[1] == 0) && (befehl_s0_in[2] == 1))
                   ((befehl_s0_in[1] == 1) && (befehl_s0_in[2] == 1))
45
                   ((befehl s0 in[1] == 2) && (befehl s0 in[2] == 1))
                   ((befehl s0 in[1] == 3) && (befehl s0 in[2] == 1)))
50
                       move_order_pos=1;
             // Fahrbefehl für mechanische Position 2 auslesen
if (((befehl_s0_in[1] == 0) && (befehl_s0_in[2] == 2))
55
                    (befehl_s0_in[1] == 1) && (befehl_s0_in[2] == 2))
                   ({befehl_s0_in[1] == 2) && (befehl_s0_in[2] == 2)}
60
                   ({befehl_s0_in[1] == 3) && (befehl_s0_in[2] == 2)})
                       move_order_pos=2;
65
             // Fahrbefehl für mechanische Position 3 auslesen
if (((befehl_s0_in[1] == 0) && (befehl_s0_in[2] == 3))
                   ({befehl_s0_in[1] == 1) && (befehl_s0_in[2] == 3)}
                    (befehl_s0_in[1] == 2) && (befehl_s0_in[2] == 3))
                   ((befehl s0 in[1] == 3) && (befehl s0 in[2] == 3)))
                       move order pos=3;
```

Der Programmauszug stellt das Auslesen der Fahrbefehle 1 bis 3 dar. Insgesamt werden 6 Fahrbefehle deklariert.

Es wird jeweils für eine mechanische Messposition des Messautomaten ein Fahrbefehl hinterlegt. Dieser wird abgelegt in der Variablen "move\_order\_pos".

Die Fahrbefehle für den Automatikbetrieb werden über die serielle Schnittstelle S0 empfangen und im eindimensionalen array "befehl\_s0\_in []" abgelegt.

Im Befehlsstring steht die für die Messposition relevante Information an zweiter und dritter Stelle, somit wird hierfür folgende Stelle im array ausgelesen:

- befehl\_s0\_in [1]
- befehl\_s0\_in [2]

Detaillierte Informationen zu den Befehlscodierungen erhalten Sie im Kapitel Kommunikation.

Verwirrend erscheint hier, dass z. B. der Fahrbefehl Nummer 2 über 4 Möglichkeiten im array "befehl\_s0\_in []" zustande kommt. Dies ist aber eine Folge davon, dass der Messautomat auf 4 verschiedenen Sitzpositionen im Kraftfahrzeug die Messungen durchführt.

Die Messposition 2 ist immer ein zur Fahrzeugmitte geneigter Messautomat. Unterschieden werden folgende Sitzpositionen im Kraftfahrzeug :

- Fahrersitz
- Beifahrersitz
- hinter Beifahrersitz
- hinter Fahrersitz

Somit wäre nicht nur eine Spiegelung der Positionen notwendig, sondern auch eine Unterscheidung zwischen den Sitzen vorne und den Sitzplätzen hinten.

Um eindeutig unterscheiden zu können, wurde für jede Messposition auf dem entsprechenden Sitzplatz ein Befehl der externen Software festgelegt, der hier entsprechend decodiert und den mechanischen Positionen zugeordnet wird.

#### 9.4.20 Modul 20: Befehle der HMI auslesen

Die Befehle der HMI werden in der Datei kommunikation\_auswertung.c decodiert und ausgewertet.

Der Dialog findet über die serielle Schnittstelle S1 statt.

Die ankommenden Befehle werden im Befehlspuffer "befehl\_s1\_in []" gespeichert. Die relevanten Informationen stehen im array an dritter und vierter Stelle, deshalb werden grundsätzlich Informationen an folgender Stelle ausgelesen:

- befehl\_s1\_in [2]
- befehl\_s1\_in [3]

Die Befehle werden nach der Decodierung weiter zur Verwendung aufbereitet, d. h. Verriegelungen werden eingefügt, die Verschaltung mit weiteren Bedingungen wird ebenso in dieser Funktion realisiert.

100	<pre>// BA Teilautomatik setzen if((befehl_s1_in[2] == 0) &amp;&amp; (befehl_s1_in[3] == 2))</pre>
105	<pre>// nur bei Grundstellung umschalten auf BA TeilAuto zulassen, sonst // Fehlerbit setzen if ((refp[0] == 1)         &amp;&amp;(refp[2] == 1)         &amp;&amp;(refp[2] == 1)         &amp;&amp;(refp[3] == 1))         &amp;&amp;(refp[3] == 1))</pre>
110	<pre>// Bei Bedienung über das LCD-Display Olli_Control Freigabe entziehen if(m_ba_teilauto == 0) {</pre>
115	<pre>olli_control_disable(); }</pre>
115	<pre>m_ba_hand=0; m_ba_auto=0; m_ba_teilauto=1;</pre>
120	<pre>/ else {     // Fehlerbit "keine Grundstellung" setzen     m_grdstllg_nio=1;</pre>
125	}

Im Programmauszug das Beispiel zur Umschaltung der Betriebsart. Die Betriebsart Teilautomatik soll gesetzt werden.

- Das array "befehl\_s1\_in []" wird ausgelesen.
- Ergibt die Decodierung die dezimalen Werte 0 und 2 ist die Anforderung zum Betriebsartenwechsel auf die Betriebsart Teilautomatik gegeben.

- Als weitere Bedingung zum Setzen der Betriebsart muss der Messautomat in Grundstellung sein.
- Wird der Betriebsartenwechsel über die HMI angefordert, muss OC die Freigabe zur Bedienung des Messautomaten entzogen werden.
- Die Betriebsarten Automatik und Hand werden zurückgesetzt.
- Wurde der Betriebsartenwechsel angefordert, obwohl der Messautomat nicht in Grundstellung war, wird ein Fehlerbit gesetzt. Über dieses Fehlerbit wird die Bedienerführung auf der HMI aktualisiert.

Ähnlich wird auch verfahren, um die Fahrbefehle im Teilautomatikbetrieb auszulesen. Im Programmauszug werden hier lediglich 4 Fahrbefehle dargestellt. Insgesamt stehen in diesem Bereich 6 Fahrbefehle zur Verfügung.

Ein neuer Fahrbefehl wird nur ausgelesen, wenn der letzte Fahrbefehl bereits abgearbeitet wurde. Ansonsten wird der Fahrbefehl ignoriert.

Der Fahrbefehl wird aus dem Befehlspuffer "befehl\_s1\_in []" ausgelesen und anschließend in die Variable "neu\_anzufahrende\_pos" zur weiteren Verarbeitung geschrieben.

Mit der Anforderung der Grundstellungsfahrt wird ebenso verfahren.

	if((befehl_s1_in[2] == 1) && (befehl_s1_in[3] == 1))
185	<pre>(     // Grundstellung nur anfordern, wenn mindestens eine Achse nicht     // auf Referenzpunktschalter steht     if (!(refp[0])</pre>
190	<pre>  (!refp[2])   (!refp[3])) { // Position 6 entspricht Position Grundstellung neu anzufahrende pos=6;</pre>
195	<pre>m_ba_hand=0; m_ba_auto=0; m_ba_teilauto=1; } else</pre>
200	<pre>{     error_grdstllg_ok(); }</pre>

Die Grundstellungsfahrt wird allerdings nur angefordert, wenn sich mindestens eine Achse nicht in Grundstellung befindet.

Eine entsprechende Bedienerführung wird über die HMI ausgegeben.

Zur Grundstellungsfahrt wird die Betriebsart automatisch in Teilautomatik gewechselt und die Betriebsarten Automatik und Hand zurückgesetzt.

Mit dem Erreichen der Grundstellung werden entsprechende Umdrehungszähler der Achsen initialisiert.

Auf gleiche Art und Weise werden die Befehle zur Betriebsart Hand decodiert und weiter verarbeitet.

```
// BA Hand
        if((befehl_sl_in[2] == 1) && (befehl_sl_in[3] == 6))
            if(m_ba_hand == 1)
210
                ba h fahrbef=1;
                pos_erreicht_tipp_al_plus=0;
        if((befehl_s1_in[2] == 1) && (befehl_s1_in[3] == 7))
            if (m ba hand == 1)
                ba h fahrbef=2;
                pos_erreicht_tipp_al_minus=0;
        if((befehl s1 in[2] == 1) && (befehl s1 in[3] == 8))
            if(m_ba_hand == 1)
                ba_h_fahrbef=3;
                pos_erreicht_tipp_a2_plus=0;
230
        if((befehl_s1_in[2] == 1) && (befehl_s1_in[3] == 9))
235
            if(m_ba_hand == 1)
                ba_h_fahrbef=4;
pos_erreicht_tipp_a2_minus=0;
240
```

Der Befehlspuffer wird ausgelesen. Entspricht die ausgelesene Information dem entsprechenden Befehl, wird dieser gesetzt.

Acht Fahrbefehle stehen insgesamt in der Betriebsart Hand zur Verfügung. Das ist ausreichend, um die 4 Achsen in jeweils beide Drehrichtungen verfahren zu können.

Im Modul 20 werden außerdem noch andere Befehle ausgelesen. Die Anforderung zum Freifahren der Achsen nach einer Endlagenfahrt wird ebenso hier decodiert wie die Anforderung der Störungsquittierung seitens der HMI.

## 9.4.21 Modul 30: Umdrehungszähler der Achsen 1 – 3 aktualisieren

In Modul 30 wird durch einen Funktionsaufruf durch einen Interrupt gesprungen. Interrupt 1 wird ausgelöst durch einen Überlauf von Counter 0.

Mit diesem Interrupt wird die Impulszählung der Inkrementalgeber von Achse 1 bis 3 realisiert. Nach dem Einsprung in die ISR (Interrupt Service Routine) wird überprüft, welche der 3 Achsen im Moment aktiv ist. Entsprechend der aktiven Achse wird die Funktion zur Aktualisierung der Umdrehungszähler der jeweiligen Achse aufgerufen. In diesen Funktionen wie *count\_umdr\_a1 { }* werden die Umdrehungszähler anschließend inkrementiert.

Hierbei handelt es sich um einen Umdrehungszähler für die Motorumdrehungen, einem Hilfsumdrehungszähler und des Spindelumdrehungszählers.

Im Programmauszug ist die ISR des Interrupts 1 dargestellt. Diese Funktion ist in der Datei *intrrpt\_functions.c* programmiert.

Im nachfolgenden Programmauszug ist die Inkrementierung der Umdrehungszähler für Achse 1 und Achse 2 dargestellt.

```
// Umdrehungszähler Achse 1 aktualisieren
// wird aufgerufen durch Interrupt 1 aus Überlauf Counter 0
void count_umdr_al (void)
105
                       // Umdrehungszähler Motor inkrementieren
// jeden 16. Impuls wird Umdrehungszähler Motor akt.
m_umdr_al++;
m_umdr_hilfsm_al++;
110
                       // Umdrehungszähler Spindel inkrementieren
// alle GEAR_N Motorumdrehungen wird Umdrehungszähler Spindel aktualisiert.
// Hilfsmerker Umdrehungszähler Motor wird gelöscht.
if (m_umdr_hilfsm_al == GEAR_N_23 )
115
                               s_umdr_al++;
m_umdr_hilfsm_al=0;
120 }
// Umdrehungszähler Achse 2 aktualisieren
// wird aufgerufen durch Interrupt 1 aus Überlauf Counter 0
125 void count_umdr_a2 (void)
                        // Umdrehungszähler Motor inkrementieren
                       // jeden 16. Impuls wird Umdrehungszähler Motor akt.
m_umdr_a2++;
m_umdr_hilfsm_a2++;
                        // Umdrehungszähler Spindel inkrementieren
// alle GEAR_N Motorumdrehungen wird Umdrehungszähler Spindel aktualisiert.
// Hilfsmerker Umdrehungszähler Motor wird gelöscht.
                        if (m umdr hilfsm a2 == GEAR N 23 )
135
                               s_umdr_a2++;
m_umdr_hilfsm_a2=0;
140 }
```

Nach dem Aufruf der Funktion wird der Umdrehungszähler des Motors inkrementiert. Der Hilfsumdrehungszähler wird ebenso inkrementiert.

Anschließend wird überprüft, ob eine vollständige Spindelumdrehung vollzogen wurde. Dazu wird der Istwert des Motorumdrehungszählers mit dem DEFINE-Wert "GEAR\_N\_23" verglichen. In diesem Wert ist die Getriebeübersetzung hinterlegt.

"GEAR\_N\_23" wird in der Headerdatei *umdrehungszähler.h* deklariert.

Ist eine Spindelumdrehung vollständig, wird der Umdrehungszähler der Spindel inkrementiert und der Hilfsumdrehungszähler des Motors zurückgesetzt.

Mit diesem Verfahren wird Positionierung der einzelnen Achsen im System realisiert.

#### 9.4.22 Modul 31: Umdrehungszähler der Achse 4 aktualisieren

Das Aktualisieren des Umdrehungszählers der Achse 4 wird ebenso wie das Aktualisieren der Umdrehungszähler für Achse 1 bis 3 realisiert.

Hierzu wird der Interrupt 3 durch einen Überlauf des Counters 1 angefordert. Dieser ist für die Zählung der Impulse des Inkrementalgebers der Achse 4 zuständig.

```
// Interrupt aus Überlauf Counter 1, Impulszählung Achse 4
_interrupt(3) void intrpt_3 (void)
[
70 count_umdr_a4();
TF1=0;
```

Informationen zur Initialisierung der Counter erhalten Sie in der Dokumentation zu Modul 4.

In der ISR zu Interrupt 3 wird lediglich die Funktion *count\_umdr\_a4 { }* zur Aktualisierung der Umdrehungszähler der Achse 4 aufgerufen.

Ein Programmauszug explizit zur Inkrementierung der Umdrehungszähler der Achse 4 wird hier nicht separat dargestellt.

Informationen hierzu können der Dokumentation zu Modul 30 entnommen werden. Die Unterschiede liegen lediglich in der Namensgebung der Variablen.

#### 9.4.23 Modul 32: Störung durch Endlagenschalter auswerten

Die Endlagenschalter des Systems sind über die Basisplatine hardwaremäßig mit dem Eingang des Interrupts 11 an Port 1.1 verbunden.

Auf diese Art und Weise muss nicht der gesamte Port, an dem die Endlagenschalter angeschlossen sind, zyklisch abgefragt werden.

75 // Fehler: Achse Endlage // Überprüfen, welche Achse auf Endlage gefahren ist \_interrupt(11) void intrpt\_error\_endlage (void) // Funktionsaufruf, Endlagenschalter auswerten error\_endlage(); 80 // Interruptrequest rücksetzen IEX4=0

Mit diesem Verfahren wird die Zykluszeit der Hauptroutine nicht unnötig verlängert. Sobald eine Antriebsachse auf Endlage gefahren ist, wird dies bemerkt und eine entsprechende Reaktion in Verbindung mit Stillsetzung der Antriebe ausgelöst.

Für eine exakte Diagnose wird nun über den Aufruf der Funktion *error\_endlage { }* der Port 7, an welchem die Endlagenschalter angeschlossen sind, ausgewertet.

In der ISR des Interrupts 11 steht allerdings nur der Aufruf der Funktion *error\_endlage { }*. Nach dem Rücksprung von *error\_endlage { }* wird der Interruptrequest zurückgesetzt.

Die Funktion *error\_endlage { }* wird in der Datei *error\_1.c*, in der nahezu die gesamte Fehlerauswertung stattfindet, ausgeführt.

In der Funktion *error\_endlage { }* werden nun nicht nur die Endlagenschalter auf Paarfehler ausgewertet, sondern auch einzeln.

Diese Auswertung wird u. A. auch für die Freifahrfunktion, die in Modul 15 dokumentiert wird, benötigt.

Eine Bedienerführung durch die HMI findet statt. Die Verwaltung der Freigaben für Steuerung des Messautomaten durch externe Software wird ebenso aktualisiert.

```
/// Überprüfe auf Endlage+, Achse 1
         // Bit 7.1 = 1 ?
if (achse_endlage == 0x02)
180
              err number = 7;
              error_break(err_number);
185
         // Fehlernummer: 8
// Überprüfe auf Endlage-, Achse 2
// Bit 7.2 = 1 ?
190
          if (achse_endlage == 0x04)
              err_number = 8;
error_break(err_number);
195
         // Fehlernummer: 9
// Überprüfe auf Endlage+, Achse 2
// Bit 7.3 = 1 ?
          if (achse_endlage == 0x08)
              err_number = 9;
error break(err number);
         205
          if (achse_endlage == 0x10)
              err_number = 10;
error_break(err_number);
210
          // Fehlernummer: 11
         // Überprüfe auf Endlage+, Achse 3
// Bit 7.5 = 1 ?
215
          if (achse_endlage == 0x20)
              err number = 11;
220
              error_break(err_number);
```

Im Programmauszug nur ein kleiner Ausschnitt der Diagnose. Auf die komplette Darstellung der Diagnose der Endlagenüberwachung wird verzichtet.

Port 7 wird eingelesen und in die Variable "achse\_endlage" geschrieben. Anschließend erfolgt eine Überprüfung auf Paarfehler.



Von Paarfehlern spricht man, wenn das unzulässige Ansprechen beider Endlagenschalter einer Achse diagnostiziert wird.

Nach der Überprüfung auf Paarfehler erfolgt die explizite Auswertung der einzelnen Schalter. Ist ein Fehler diagnostiziert, wird eine entsprechende Fehlernummer generiert.

Die Fehlernummer wird an die Funktion *error\_break {}* übergeben und löst dort die für diesen Fehler hinterlegte Reaktion aus.

Über die HMI kann dieser Fehler nun visualisiert, beseitigt und quittiert werden. Endlagenfehler sind stets quittierpflichtig.

## 9.4.24 Modul 33: Leistungsteil Überlast

Die OCD-Ausgänge der Motorsteuer-ICs sind über die Basisplatine hardwaremäßig an den Port 1.2 des Mikrocontrollers angeschlossen.

Diese Signale wurden zusammengeführt an den Port angeschlossen und lösen bei Überlast der ICs einen Interrupt aus. Hierfür wird der Interrupt Nummer 12 verwendet.

In der ISR (Interrupt Service Routine) des Interrupts Nummer 12 folgt ein Aufruf der Funktion *"error\_overload {}*", welche in der Datei *error\_1.c* abgearbeitet wird.

Eine Visualisierung der Störung über die HMI ist selbstverständlich. Die Störung ist quittierpflichtig.

Eine entsprechende Reaktion auf die generierte Fehlernummer in error\_1.c wird ausgelöst. Diese Reaktion ist mit dem Stillsetzen der Antriebe verbunden, um mechanische Beschädigungen der Antriebsachsen und Schäden an der Elektronik durch thermische Überlastung zu verhindern.

Auf die Darstellung eines Programmauszuges wird verzichtet.

Für diesen Fehler, Überlast Leistungsteil, wird die Fehlernummer 14 generiert.

## 9.4.25 Modul 34: Deckelschalter

Der Deckelschalter wird überprüft, durch den Anschluss eines Reedkontaktes an Port 1.5. Die Auswertung dieser Störung erfolgt in der Funktion *error\_without\_hut {}* in der Datei *error\_1.c.* 

```
255 // -----
// Fehlernummer: 15
// Uberprifen, ob Olli seinen Hut genommen hat (der Deckel abgenommen wurde)
void error_without_hut (void)
{
260 err_number = 15;
error_break(err_number);
}
```

Durch Übergabe der generierten Fehlernummer 15 an die Funktion *error\_break { }* wird eine entsprechende Reaktion auf die Fehlernummer ausgelöst.

Es erfolgt u. A. eine Visualisierung der Störung und weitere Bedienerführung auf der HMI.

Die Störung ist quittierpflichtig.

## 9.4.26 Serielle Schnittstellen

Die Programmierung der seriellen Schnittstelle S0 und S1 unterscheidet sich nur geringfügig voneinander, daher wird in den folgenden Abschnitten nur die Programmierung der seriellen Schnittstelle S0 dokumentiert.

Die Initialisierung der beiden seriellen Schnittstellen wurde bereits im Kapitel 9.4.2 zu Modul 1 behandelt.

Die beiden Schnittstellen sind nach dem Prinzip der Interruptsteuerung programmiert.

Jeder Interrupt-Request in der Datei *seriell\_olli\_seriell.c* wird in der ISR *interrupt\_seriell\_S0 {}* abgearbeitet. In dieser Funktion wird unterschieden, ob es sich um eine Sende- oder Empfangsanforderung handelt. Je nach Anforderung wird in die entsprechende Funktion weitergesprungen.

#### Das Empfangen über S0 :

Beim Empfangen wird die Funktion *seriell\_S0\_int\_receive { }* aufgerufen. In dieser Funktion wird ein Ringpuffer verwaltet, in dem die neu empfangenen Zeichen zwischengespeichert werden.

Sobald dieser Puffer 4 Bytes Daten gespeichert hat, werden diese an die Funktion *interpreter\_S0\_receive { }* weitergegeben. In dieser Funktion wird der 4 Byte lange Befehlsframe auf Plausibilität geprüft.

Wenn durch diese Überprüfung kein Fehler diagnostiziert wird, wird das Bit *m\_new\_move\_order* gesetzt.

Dem Absender wird als Bestätigung zum erfolgreichen Empfang des Befehlsframes ein "B00x" gesandt.

Über diese Befehle erhalten Sie mehr Informationen im Kapitel 10.4.4 – Sicherheit bei der Datenübertragung.

Mit dem global gültigen Bit *m\_new\_move\_order* wird in der Hauproutine der Datei *main.c* angezeigt, dass ein neuer Befehl zur Abarbeitung bereit steht.

Wenn bei der Plausibilitätsprüfung jedoch ein Fehler diagnostiziert wird, dann wird das dem Absender mitgeteilt. Dazu wird dem Absender der Befehlsframe "B09x" gesandt.

Auch an dieser Stelle nochmals der Hinweis auf Kapitel 10.4.4.

## Das Senden über S0 :

Bevor das Versenden eines Befehls stattfinden kann, wird in der Funktion transmit\_S0\_vorbereitung { } der Befehlsframe komplettiert.
Um der Hauptroutine anzuzeigen, dass ein Befehlsframe zum absenden bereit ist, wird das Bit *m\_frame\_s0\_fertiggesendet* auf "false" gesetzt.

```
370 //transmit_S0_vorbereitung
    //Der Aufrüf dieser Funktion erfolgt aus der Funktion: "interpreter_S0_receive"
    //und "main". Der Wert der an die Funktion übergeben wird, ist der Inhalt vom
    //Code Byte2. Ein Daten-Frame besteht aus 4 Byte.
    void transmit_S0_vorbereitung(unsigned char wert)
375 (
        seriell_S0_clear_out_buffer();
        if ((wert == 0)|(wert == 7)|(wert == 8)|(wert == 9))
        (
            s0_outbuf[0] = 'B';
        s0_outbuf[1] = '0';
        s0_outbuf[2] = wert + '0';
        s0_outbuf[2] = wert + '0';
        s0_outbuf[3] = paritaetsberechnung(s0_outbuf[1],s0_outbuf[2]);
        m_frame_s0_fertiggesendet = false;
            index_s0_outbuf[1] = befehl s0_in[1] + '0';
        s0_outbuf[3] = paritaetsberechnung(s0_outbuf[1],s0_outbuf[2]);
        m_frame_s0_fertiggesendet = false;
            index_s0_outbuf[3] = paritaetsberechnung(s0_outbuf[1],s0_outbuf[2]);
        m_frame_s0_fertiggesendet = false;
        index_s0_outbuf[3] = paritaetsberechnung(s0_outbuf[1],s0_outbuf[2]);
        m_frame_s0_fertiggesendet = false;
        index_s0_outbuf[3] = paritaetsberechnung(s0_outbuf[1],s0_outbuf[2]);
        m_frame_s0_fertiggesendet = false;
        index_s0_outbuf[4] = 0;
        }
        s0_outbuf[5] = paritaetsberechnung(s0_outbuf[1],s0_outbuf[5]);
        m_frame_s0_fertiggesendet = false;
        index_s0_outbuf[5] = 0;
        }
        s0_outbuf[5] = 0;
        s0_o
```

Jetzt kann mit dem eigentlichen Versand des Befehls begonnen werden. Dazu wird die Funktion *seriell\_S0\_int\_transmit { }* aufgerufen. In dieser Funktion wird überprüft, ob der Befehlsframe bereits vollständig versandt wurde.

Kommt diese Überprüfung zu einem negativen Ergebnis, bleibt das Bit *m\_frame\_s0\_fertiggesendet* auf "false". Solange dieses Bit auf "false" steht, wird über die Hauptroutine die Funktion *seriell\_S0\_int\_transmit {}* zyklisch aufgerufen.

Mehr Informationen zum Thema Befehlframes und ihre Bedeutung erhalten Sie im Kapitel 10 – Kommunikation.

# 10 Kommunikation

## 10.1 Allgemein

Der Mikrocontroller der Prozessorplatine verfügt über zwei serielle Schnittstellen. Beide Schnittstellen sind RS 232 – Schnittstellen.

- Serielle RS 232 Schnittstelle S0
- Serielle RS 232 Schnittstelle S1

Über die serielle Schnittstelle S0 ist die Anbindung von externer Software wie OLLICONTROL oder das Matlab-Programm Hasheraccelerator an den Mikrocontroller realisiert.

Über die serielle Schnittstelle S1 wurde die Verbindung des Mikrocontrollers mit dem LCD-Touchpanel als HMI projektiert.

## 10.2 Kommunikation: HMI zu Mikrocontroller

Über das Betätigen der Softkeys des LCD-Panels werden codierte Befehle über die Schnittstelle S1 zum Mikrocontroller gesendet.

Die Befehle setzen sich aus einem ASCII-String zusammen, der 5 Byte lang ist. Eine Übersicht der Befehlsstrings, die in der HMI abgelegt sind, zeigt Abbildung 10-1.

ode Byte 0 mpfängeradr. > Olli	Code Byte 1 Tastennummer Ziffer 1	Code Byte 2 Tastennummer Ziffer 2	Code Byte 3 Paritätskennzeichnung Addition Byte 1 + Byte 2 Anzahl "1" gerade -> x Anzahl "1" gerade -> x	Bildname Display	Tastenbezeichnung	Tastennummer	Kommentar
attdruck -> Befehl v	renvendet		Anzani i ungerade -> y				
				Detricheerteneuwehl	A uto	01	
	0	1	y	Betriebsartenauswahl	Auto	01	BA Auto auswanien BA Teileute euswählen
	0	2	<u>y</u>	Betriebsartenauswahl	Hand	02	BA Hand auswählen
A	0	3	*	Betriebeardenauswahl	naliu	03	DA nanu auswanien
-			2	Automotik	Automotile Ereinehe	04	Automotikableuf Ereinaba
	0	5	×	Automatik	Automatik Preigabe	05	RA TA Desition 1 enfetrer
A	0		×	Teilautomatik	Position 1	00	BA TA Position 1 anianien
A .	0	1	y	Tellautomatik	Position 2	07	BA TA Position 2 antanren
A .	0	8	У	Tellautomatik	Position 3	08	BA TA Position 3 antahren
A .	0	9	×	Tellautomatik	Position 4	09	BA TA Position 4 antahren
A	1	0	У	Tellautomatik	Position 5	10	BA TA Position 5 antanren
<u>A</u>	1	1	У	Teilautomatik	Grundstellung	11	BA TA Grundstellung anfahren
A	1	2	x	Handbetrieb	Achse 1	12	Bild BA Hand Achse 1 auswählen
A	1	3	у	Handbetrieb	Achse 2	13	Bild BA Hand Achse 2 auswählen
A	1	4	x	Handbetrieb	Achse 3	14	Bild BA Hand Achse 3 auswählen
A	1	5	x	Handbetrieb	Achse 4	15	Bild BA Hand Achse 4 auswählen
A	1	6	У	Handbetrieb Achse 1	Richtung +	16	BA Hand A1 in Richtung + fahren
А	1	7	У	Handbetrieb Achse 1	Richtung -	17	BA Hand A1 in Richtung - fahren
А	1	8	x	Handbetrieb Achse 2	Richtung +	18	BA Hand A2 in Richtung + fahren
А	1	9	x	Handbetrieb Achse 2	Richtung -	19	BA Hand A2 in Richtung - fahren
A	2	0	У	Handbetrieb Achse 3	Richtung +	20	BA Hand A3 in Richtung + fahren
A	2	1	x	Handbetrieb Achse 3	Richtung -	21	BA Hand A3 in Richtung - fahren
A	2	2	v	Handbetrieb Achse 4	Richtung +	22	BA Hand A4 in Richtung + fahren
A	2	3	x	Handbetrieb Achse 4	Richtung -	23	BA Hand A4 in Richtung - fahren
A	2	4	x	Stoerung 1	Stoerung guittieren	24	Störung guittieren
А	2	5	v	Stoerung 2	Achse freifahren	25	Achse von Endlage fahren

(Abb. 10-1)

Die Übersicht der Befehlsstrings finden Sie auch auf der Dokumentations – CDR als Datei im pdf-Format.

Jedem Softkey der HMI ist ein Befehlsstring zugeordnet. Wird der Softkey betätigt, läuft in der HMI ein Makro ab, das den 4 Byte langen Befehlsstring über die serielle Schnittstelle S1 zum Mikrocontroller sendet.

Als Besonderheit sendet das LCD-Display als erstes Byte immer den Wert 0x04. Dieses Byte wird dem CodeByte 0 vorangestellt, so dass sich immer ein 5 Byte langer Befehl ergibt.

Die Software des Mikrocontrollers decodiert den empfangenen Befehl und löst eine entsprechende Reaktion aus.

Als eigentliches Informationsbyte dient CodeByte 1 und CodeByte 2. Zur Bedeutung der einzelnen CodeBytes im Befehlsstring lesen Sie Kapitel 10.2.1.

Zur Programmierung der HMI lesen Sie Kapitel 11 – Details zur Programmierung der HMI. Dort erhalten Sie auch Informationen zur Programmierung mit Makros.

## 10.2.1 Aufbau eines Befehls

Wie Sie Kapitel 10.2 entnehmen können, besteht ein Befehl, der von der HMI zum Mikrocontroller gesendet wird, aus 5 Byte.

### Befehlsbyte START:

Das LCD-Touchpanel sendet als Startbyte zur Datenübertragung immer den Wert 0x04. Dieses Startbyte muß nicht explizit programmiert werden.

Im entsprechenden Makro zur Versendung der Befehlsstrings werden nur folgende 4 Bytes programmiert. Der Befehl wird im Makro als 4 Byte Befehl deklariert. Wie auch in Kapitel 10.2 hier der Verweis auf Kapitel 11 – Details zur Programmierung mit Makros.

**Befehlsbyte 1** (entspricht in Abbildung 10-1 CodeByte 0) :

Ein Großbuchstabe wird als Empfängeradresse verwendet.

- Empfängeradresse A entspricht dem Mikrocontroller
- Empfängeradresse B entspricht OLLICONTROL / Hasheraccelerator

Die Empfängeradresse B wird hier nur der Vollständigkeit wegen aufgeführt. Ein Befehl seitens der HMI mit der Empfängeradresse ist nicht möglich. Diese Adresse wird nur in der Kommunikation über die serielle Schnittstelle S0 verwendet.

**Befehlsbyte 2** (entspricht in Abbildung 10-1 CodeByte 1) :

Die Softkeys werden nach ihrer Programmierung durchnummeriert. Die HMI wurde projektiert mit Auslegung auf 99 Softkeys.

 Befehlsbyte 2 entspricht der Zehnerstelle des programmierten Softkeys, also der ersten Ziffer der Nummer des Softkeys.

**Befehlsbyte 3** (entspricht in Abbildung 10-1 CodeByte 2) :

 Befehlsbyte 3 entspricht der Einerstelle des programmierten Softkeys, also der zweiten Ziffer der Nummer des Softkeys.

**Befehlsbyte 4** (entspricht in Abbildung 10-1 CodeByte 3) :

Dieses Byte wird zur Überprüfung auf korrekt empfangenen Befehl mitgesendet. Die Software des Mikrocontrollers führt eine Paritätsprüfung durch. Folgende Festlegungen wurden zur Paritätsprüfung getroffen:

- Als Befehlsbyte 4 wird ein "x" gesendet, wenn die Anzahl der Einser nach Addition von Befehlsbyte 2 und Befehlsbyte 3, gerade ist.
- Als Befehlsbyte 4 wird ein "y" gesendet, wenn die Anzahl der Einser nach Addition von Befehlsbyte 2 und Befehlsbyte 3, ungerade ist.

Die Paritätsprüfung findet im Mikrocontroller statt. Wird ein empfangener Befehl als falsch erkannt, löst die Software des Mikrocontrollers eine entsprechende Reaktion aus. Mehr Informationen hierzu erhalten Sie im Kapitel 9 – Programmierung des Mikrocontrollers.

Die Berechnungen zur Paritätsprüfung wurden seitens der HMI bei der Projektierung durchgeführt und bereits als Befehlsstring im EEprom der HMI abgelegt.

## 10.3 Kommunikation: Mikrocontroller zu HMI

Die Oberflächen der HMI zur Bedienung des Messautomaten wurden in Form von Makros im EEprom der HMI gespeichert.

Der Aufruf einer Oberfläche geschieht nun durch das Starten eines Makros. Die Oberflächen sind als sogenannte Normalmakros in der HMI abrufbar.

Lesen Sie hierzu Kapitel 11 – Details zur Programmierung der HMI.

### 10.3.1 Aufbau eines Befehls

Zum Aufruf eines Makros in der HMI wird ein 5 Byte langer Befehlsstring benötigt. Der Befehl zum Aufruf eines Normalmakros setzt sich wie folgt zusammen:

### Befehlsbyte 0 :

 Als Startzeichen wird immer der Wert 0x1B gesendet. Dieser Wert entspricht dem "esc" bzw. # - Zeichen, mit dem grundsätzlich ein Befehl der textbasierenden Programmiersprache, in der das LCD – Touchpanel zu programmieren ist, eingeleitet wird.

### Befehlsbyte 1 :

• Als Befehlsbyte 1 wird ein ,M' gesendet. ,M' steht für den Aufruf eines Makros.

### Befehlsbyte 2 :

 Als Befehlsbyte 2 wird ein ,N' gesendet. ,N' steht in Kombination mit dem Befehlsbyte 1 für den Aufruf eines Normalmakros.

### Befehlsbyte 3 :

- Als Befehlsbyte 3 wird die Nummer des aufzurufenden Makros gesendet. Befehlsbyte 3 entspricht der Nummer, mit welcher das Makro im EEprom des LCD-Displays abgelegt wurde.
- Befehlsbyte 3 ist eine Zahl zwischen 0x00 und 0xFF. 256 Normalmakros können im Speicher des EEproms abgelegt werden.

### Befehlsbyte 4 :

Als Abschlusszeichen wird im Befehlsbyte 4 immer der Wert 0x0D gesendet.

## 10.4 Kommunikation: Externe Software und Mikrocontroller

### 10.4.1 Allgemein

Die Kommunikation zwischen externer Software wie OLLICONTROL oder dem Matlabprogramm Hasheraccelerator mit dem Mikrocontroller findet über die serielle Schnittstelle S0 statt.

Der Befehlsframe ist nach dem gleichen Prinzip wie der Frame zur seriellen Schnittstelle S1 aufgebaut. Der Unterschied besteht darin, dass bei der Schnittstelle S0 ein Befehlsframe 4 Byte lang ist.

## 10.4.2 Übersicht der Befehle

Die Übersicht der möglichen Befehle, die von einer externen Software wie OC oder dem Hasheraccelerator gesendet werden, ist in Abbildung 10-2 dargestellt.

	Olli								Achse	en fahre	en		mecha	nische	Positi	on		
	in Fahrt	richtung	Seitenne	eigung														
Fahrersitz	vome	hinten	links	mittig	rechts	oben	mittig	unten	A 1	A 2	A 3	A 4	P1	P 2	P 3	Ρ4	Ρ5	Grd/P6

1	AUTY		х		x			х					х	х					
2	A02y		х			х		х		х			х		х				
3	A03x		х	х				х				х	х			х			
4	A04y	х			х				х		х						х		
5	A05x		х		х		х						х					×	
6	A06x		х		х				х				х						х

Beifahrersitz

2     A12x     x	1	A11y		х		х			х					х	х					
3     A13y     x     x     x     x     x     x       4     A14x     x     x     x     x     x     x     x	2	A12x		х			х		х		х			х		х				
4 A14x x x x x x x	3	A13y		х	х				х				x	х			x			
	4	A14x	х							х		х						х		
	5	A15x		х		х		х						х					х	
6 A16y X X X X X X X X X	6	A16y		х		х				х				х						х

hinter Fahrersitz

1	A21x		х		Х			х					х	Х					$\square$
2	A22y		х			х		х		х			х		х				
3	A23x		х	х				x				х	х			x			
4	A24x	х							х		х						х		
5	A25y		х		х		х						х					х	
6	A26y		х		х				х				х						х

hinter Beifahrersitz

1	A31y		х		х			х					х	х					
2	A32x		х			х		х		х			х		х				
3	A33x		х	х				х				х	х			х			
4	A34y	х							х		х						х		
5	A35y		х		х		х						х					х	
6	A36x		х		х				х				х						х

(Abb. 10-2)

Welche mechanische Position der Messautomat in Abhängigkeit der Befehle anfährt und welche Achsen dazu Verfahrbewegungen auszuführen haben, ist in Abbildung 10-2 ebenso ersichtlich.

## 10.4.3 Aufbau eines Befehls

Wie bereits in Kapitel 10.4.1 darauf hingewiesen, ist ein Befehlframe 4 Byte lang. Der folgende Abschnitt informiert über die Bedeutung der einzelnen Bytes im Frame. **Befehlsbyte 1 :** 

Ein Großbuchstabe wird als Empfängeradresse verwendet.

- Empfängeradresse A entspricht dem Mikrocontroller
- Empfängeradresse B entspricht OLLICONTROL / Hasheraccelerator

Die Empfängeradresse B ist in der Übersicht der Befehle in Abbildung 10-2 nicht ersichtlich. Mehr Informationen dazu erhalten Sie in Kapitel 10.4.4 – Sicherheit bei der Datenübertragung.

### Befehlsbyte 2 :

Im Befehlsbyte 2 ist der Sitzplatz im Kraftfahrzeug hinterlegt, auf dem der Messautomat positioniert ist.

- Wert "0" entspricht dem Fahrersitz
- Wert "1" entspricht dem Beifahrersitz
- Wert "2" entspricht dem Sitz hinter dem Fahrersitz
- Wert "3" entspricht dem Sitz hinter dem Beifahrersitz

### Befehlsbyte 3 :

Im Befehlsbyte 3 ist die angeforderte Position hinterlegt, die der Messautomat anfahren soll. 6 Positionen sind zur Auswahl möglich.

- Wert "1" entspricht der mechanischen Position 1
- Wert "2" entspricht der mechanischen Position 2
- Wert "3" entspricht der mechanischen Position 3
- Wert "4" entspricht der mechanischen Position 4
- Wert "5" entspricht der mechanischen Position 5
- Wert "6" entspricht der mechanischen Position 6, der Grundstellung

### Befehlsbyte 4 :

Dieses Byte wird zur Überprüfung auf korrekt empfangenen Befehl mitgesendet. Durch dieses Befehlsbyte findet eine Paritätsprüfung statt. Folgende Festlegungen wurden zur Paritätsprüfung getroffen:

- Als Befehlsbyte 4 wird ein "x" gesendet, wenn die Anzahl der Einser nach Addition von Befehlsbyte 2 und Befehlsbyte 3, gerade ist.
- Als Befehlsbyte 4 wird ein "y" gesendet, wenn die Anzahl der Einser nach Addition von Befehlsbyte 2 und Befehlsbyte 3, ungerade ist.

Informationen zu den Folgen eines diagnostizierten Datenübertragungsfehler erhalten Sie in Kapitel 10.4.4 – Sicherheit bei der Datenübertragung.

Im Kapitel 9.4.26 erhalten Sie Detailinformationen zur Programmierung der seriellen Schnittstellen S0 und S1.

### 10.4.4 Sicherheit bei der Datenübertragung

Befehls	sframe			
Byte 1	Byte 2	Byte 3	Byte 4	Funktion
В	0	0	х	Mikrocontroller an ext. SW: "Nachricht verstanden"
В	0	7	у	Mikrocontroller an ext. SW: "Freigabe entzogen"
В	0	8	у	Mikrocontroller an ext. SW: "Freigabe erteilt"
В	0	9	Х	Mikrocontroller an ext. SW: "Nachricht nicht verstanden"

(Abb. 10-3)

Jeder Befehl, der an den Mikrocontroller gesendet wurde, wird auf seine Plausibilität und einer Paritätsprüfung unterzogen.

Die Tabelle in Abbildung 10-3 gibt Aufschluss darüber, mit welchen Befehlen auf eine Bewegungsanforderung geantwortet wird.

Zwei Möglichkeiten stehen zur Auswahl:

- Der Mikrocontroller hat den Befehl verstanden und bestätigt das mit dem Befehlsframe "B00x".
- Der Mikrocontroller hat den Befehl nicht verstanden und meldet das dem Absender mit dem Befehlsframe "B09x".

Der Aufbau dieses Befehlsframe ist gleich dem Aufbau des dargestellten Frames in Kapitel 10.4.2.

### Befehlsbyte 1 :

Ein Großbuchstabe wird als Empfängeradresse verwendet.

Empfängeradresse B entspricht OLLICONTROL / Hasheraccelerator

### Befehlsbyte 2 :

• Erste Stelle des Informationsteils.

### Befehlsbyte 3 :

Zweite Stelle des Informationsteils.

#### Befehlsbyte 4 :

Dieses Byte wird zur Überprüfung auf korrekt empfangenen Befehl mitgesendet. Durch dieses Befehlsbyte findet eine Paritätsprüfung statt. Folgende Festlegungen wurden zur Paritätsprüfung getroffen:

• Als Befehlsbyte 4 wird ein "x" gesendet, wenn die Anzahl der Einser nach Addition von Befehlsbyte 2 und Befehlsbyte 3, gerade ist.

Als Befehlsbyte 4 wird ein "y" gesendet, wenn die Anzahl der Einser nach Addition von Befehlsbyte 2 und Befehlsbyte 3, ungerade ist.

### 10.4.5 Sicherheit beim Prozess

In Kapitel 12.3 erhalten Sie Informationen zur Bedienung des Messautomaten durch externe Software. Dort wird darauf hingewiesen, dass die HMI die dominante Schnittstelle zur Bedienung des Automaten ist.

Bei diagnostizierten Fehlern wird externer Software die Freigabe zur Bedienung des Messautomaten entzogen. Ist die Störung erfolgreich beseitigt und quittiert, wird die Freigabe zur Bedienung wieder erteilt.

Dieser Vorgang des Entzugs und der Erteilung der Freigabe dient der Prozesssicherheit des Systems und gelingt mit folgenden zwei Befehlen:

- "B07y" entspricht "Freigabe entzogen"
- "B08y" entspricht "Freigabe erteilt"

Es wird darauf verzichtet, den Befehlsframe hier nochmals darzustellen. Dieser Befehlsframe entspricht dem in Kapitel 10.4.4 dargestelltem Frame.

# 11 Details zur Programmierung der HMI

## 11.1 Allgemeines

Erste Informationen zum LCD-Touchpanel erhalten Sie in Kapitel 6.2.5. Auf die Darstellung technischer Daten wird hier deshalb verzichtet.

Das Display ist mit hochsprachenähnlichen Grafikbefehlen zu programmieren. Die Programmierung erfolgt textbasierend, eine grafische Programmierung ist nicht möglich.

Zum Programmieren des Displays ist ein ASCII - Editor ausreichend. Dieses Projekt wurde mit dem Editor UltraEdit 32 programmiert.

Die Programmierung erfolgt ausschließlich durch Programmierung von Makros. Drei Arten von Makros werden hierzu unterschieden :

- Normalmakros
- Touchmakros
- Portmakros

Portmakros wurden in diesem Projekt nicht verwendet.

## 11.2 Ablaufdiagramme der HMI

Auf den folgenden Seiten werden die Zusammenhänge der einzelnen Makros in Ablaufdiagrammen dargestellt.

Das Prinzip ist bereits aus den Ablaufdiagrammen zum C-Projekt bekannt und wird hier nicht weiter erläutert.





Kurzbeschreibung der Makros

Kurzbeschreibung dei	Makros :
Normalmakros	Charles line laitelining and
Makro 1	Intro. Starthildschirm
Makro 2:	Intro, Sanduhr
Makro 3:	Aufruf Betriebsartenwahl
Makro 4:	Aufruf Betriebsartenwahl nach Störung
Makro 5:	Betriebsartenwahl nach Initialisierung des Displays
Makro 6:	Neuinitialisierung des Displays durch "restart-Taste"
Makro 10:	Grundmaske 1 Störbild
Makro 11:	Grundmaske 2 Störbild
Makro 12:	Grundmaske 3 Störbild
Makro 20:	Störung Paarfehler Endschalter Achse 1
Makro 21: Makro 22:	Storung Paarfehler Endschalter Achse 2
Makro 22:	Storung Paarfehler Endschalter Achse 3
Makro 24	Störung Endlage + angefahren, Achse 1
Makro 25:	Störung Endlage + angefahren. Achse 2
Makro 26:	Störung Endlage + angefahren, Achse 3
Makro 27:	Störung Endlage + angefahren, Achse 4
Makro 28:	Störung Endlage - angefahren, Achse 1
Makro 29:	Störung Endlage - angefahren, Achse 2
Makro 30:	Störung Endlage - angefahren, Achse 3
Makro 31:	Storung Endlage - angefahren, Achse 4
Makro 32:	Storung Überlast, Achse 2
Makro 34	Storung Überlast, Achse 2 Störung Überlast, Achse 3
Makro 35	Störung Überlast, Achse 4
Makro 36:	Störung keine Grundstellung
Makro 37:	Meldung Grundstellung schon vorhanden
Makro 38:	Meldung BA TeilAuto Position 1 erreicht
Makro 39:	Meldung BA TeilAuto Position 2 erreicht
Makro 40:	Meldung BA TeilAuto Position 3 erreicht
Makro 41:	Meldung BA TeilAuto Position 4 erreicht
Makro 42:	Meldung BA TeilAuto Position 5 erreicht
Makro 43:	Meldung BA umschalten nur bei inaktiven Achsen möglich
Makro 44: Makro 45:	Warpung Olli aufrichten
Makro 46	Störung Deckel nicht abgenommen
Makro 50:	Meldung BA Auto. P1 erreicht
Makro 51:	Meldung BA Auto, P2 erreicht
Makro 52:	Meldung BA Auto, P3 erreicht
Makro 53:	Meldung BA Auto, P4 erreicht
Makro 54:	Meldung BA Auto, P5 erreicht
Makro 55:	Meldung BA Auto, P1-Fahrt aktiv
Makro 56:	Meldung BA Auto, P2-Fahrt aktiv
Makro 58	Meldung BA Auto, P3-Panit aktiv
Makro 59:	Meldung BA Auto, P5-Fahrt aktiv
Makro 60	Meldung BA Auto P6-Fahrt aktiv
Makro 61;	Meldung BA TeilAuto, P1-Fahrt aktiv
Makro 62:	Meldung BA TeilAuto, P2-Fahrt aktiv
Makro 63:	Meldung BA TeilAuto, P3-Fahrt aktiv
Makro 64:	Meldung BA TeilAuto, P4-Fahrt aktiv
Makro 65:	Meldung BA TeilAuto, P5-Fahrt aktiv
Makro 66:	Meldung BA TeilAuto, P6-Fahrt aktiv
Touchmakros	
Touchmakro 1:	Hilfsmakro zu Aufruf Makro2, Betriebsartenwahl
Touchmakro 2:	Automatikablauf Oberfläche 1
Touchmakro 3:	Teilautomatik Oberfläche
Touchmakro 4:	Handbetrieb Oberfläche 1
Touchmakro 5:	Teilautomatik Taste Position 4 anfahren
Touchmakro 6:	Teilautomatik Taste Position 1 anfahren
Touchmakro 7:	Teilautomatik Taste Position 3 anfahren
Touchmakro 9	Teilautomatik Taste Position 2 anfahren
Touchmakro 10:	Teilautomatik Taste Position 0. Grundstellung anfahren
Touchmakro 11:	Handbetrieb Oberfläche 2 - Achse 1
Touchmakro 12:	Handbetrieb Oberfläche 3 - Achse 2
Touchmakro 13:	Handbetrieb Oberfläche 4 - Achse 3
Touchmakro 14:	Handbetrieb Oberfläche 5 - Achse 4
Touchmakro 15:	Handbetrieb Taste Richtung +, Achse 1
Touchmakro 16:	Handbetrieb Taste Richtung -, Achse 1
Touchmakro 17:	Hilfe zu Teilautomatikoberfläche 1
Touchmakro 18:	Hilfe zu Handbetrieboberfläche 1
Touchmakro 19:	Hilfe zu Teilautomatikoherfläche 2
Touchmakro 21	Handbetrieb Taste Richtung + Achse 2
Touchmakro 22	Handbetrieb Taste Richtung - Achse 2
Touchmakro 23:	Handbetrieb Taste Richtung +, Achse 3
Touchmakro 24:	Handbetrieb Taste Richtung Achse 3
Touchmakro 25:	Handbetrieb Taste Richtung +, Achse 4
Touchmakro 26:	Handbetrieb Taste Richtung -, Achse 4
Touchmakro 27:	Störung quittieren
Touchmakro 28:	serielle Schnittstelle an Olli, Grdstilg fahren
Touchmakro 29:	Autrur reset/restart, Neuinitialisierung
Touchmakro 30:	Achse nach Endlagenfahrt freifahren
Touchmakro 50:	serielle Schnittstelle an Olli, BA Auto Freigabe aktiv
Touchmakro 52	serielle Schnittstelle an Olli, BA Hand aktiv

LCD - Display Ablaufdiagramm der Makros Blatt 1.0.1

(Abb. 11-2)





















## 11.3 Details zum Quelltext

Zum besseren Verständnis des Quelltextes, in Abbildung 11-8 und Abbildung 11-9 zuerst eine Befehlsübersicht.

## 11.3.1 Befehlstabelle

					lle für EA KIT160-7												
Befehl	Cod	es			/ Befehle für de				Anmerkung								
						Be	fehle	e für o	den Terminal Betrieb								
Formfeed FF (dez: 12)	۸L								Bildschirm wird gelöscht und der Cursor nach Pos. (1,1) gesetzt								
Carriage Return CR(13)	^M								Cursor ganz nach links zum Zeilenanfang								
Linefeed LF (dez:10)	č								Cursor 1 Zeile tiefer, falls Cursor in letzter Zeile dann auf 1. Zeile setzen								
Cursor On / Off	ESC	Q	С	n1					n1=0: Cursor ist unsichtbar; n1=1: Cursor blinkt (invers 6/10s);								
Cursor positionieren	ESC	0	n1	n2					n1=Spalte; n2=Zeile; Ursprung links oben ist (1,1)								
Terminal Font einstellen	ESC	F	т	n1					n1=1: Font Nr. n1 (116) für Terminal Betrieb einstellen								
							Bef	ehle :	zur Textausgabe								
Text-Modus	ESC	L	n1	mst					Modus n1: 1=setzen; 2=löschen; 3=invers; 4=Replace; 5=invers Replace; mst: Muster Nr. 07 verwenden;								
Font einstellen	ESC	F	n1	n2	n3				Font mit der Nummer n1 (116) einstellen; n2=X- n3=Y-Zommfaktor (1x8x);								
7-i-hh-H-			L						Eine Zeichenkette () an x1,y1 ausgeben. 'NUL' (\$00)=Zeichenkettenende;								
zeichenkette horizontal ausgeben	ESC	z	Z R	x1	y1	/1 Text NUL N Y			Menrere Zeilen werde durch das Zeichen (* 1570, dez: 124) getrennt; 121= Linkbindig an x1; 221= Zentriert an x1; 181= Rechtsbündig an x1; y1 ist immer die Oberkannte der Zeichenkette								
Zeichenkette			0			<b>T</b>			Eine Zeichenkette () um 90° gedreht an x1,y1 ausgeben; 'NUL' (\$00)=Ende;								
90° gedreht (vertikal)	ESC	z	М	×1	y1	1 Text NUL N			Menrere Zeilen werde durch das Zeichen ' ' (\$7C, dez: 124) getrennt; 'O':= Oben-Bündig an y1; 'M':= Mittig an v1: 'U':= Unten-Bündig an v1:								
ausgeben			U						x1 ist immer die Rechte Kannte der Zeichenkette								
Zeichen definieren	ESC	E	n1		d	daten r Befehle			n1=Zeichen Nr.; daten=Anzahl Bytes je nach akt. Font								
							Be	efehle	e zum Zeichnen								
Grafik-Modus	ESC	v	n1						Zeichenmodus einstellen für die Befehle: 'Punkt setzen', 'Gerade zeichnen', 'Rechteck', 'Rundeck' und 'Bereich mit Füllmuster' ni: 1 =setzen; 2=löschen; 3=invers; 4=Replace; 5=Invers Replace;								
Punkt setzen	ESC	Р	x1	y1					Ein Pixel an die Koordinaten x1, y1 setzen								
Gerade zeichnen	ESC	G	x1	y1	x2	y2			Eine Gerade von x1,y1 nach x2,y2 zeichnen								
Gerade weiter zeichnen	ESC	w	x1	y1		E			Eine Gerade vom letzten Endpunkt bis x1, y1 zeichnen								
								Rech	teck Befehle								
Rechteck zeichnen	R x1 y1 x2 y2 Ei			Ein Rechteck (Rahmen) von x1,y1 nach x2,y2 zeichnen													
Rundeck zeichnen			Ν	x1	y1	x2	y2		Ein Rechteck mit runden Ecken von x1,y1 nach x2,y2 zeichnen								
Bereich löschen			L	x1	y1	x2	y2		Einen Bereich von x1,y1 nach x2,y2 löschen (alle Pixel aus)								
Bereich invertieren	ESC	R	Т	×1	y1	x2	y2		Einen Bereich von x1,y1 nach x2,y2 invertieren (alle Pixel umkehren)								
Bereich füllen			s	×1	y1	x2	y2		Einen Bereich von x1,y1 nach x2,y2 füllen (alle Pixel ein)								
Bereich m. Füllmuster			м	×1	y1	x2	y2	mst	Einen Bereich von x1,y1 nach x2,y2 mit Muster mst (07) zeichnen								
Box zeichnen			0	x1	y1	x2	y2	mst	Ein Rechteck mit Füllmuster mst (07) zeichnen; (immer Replace)								
Rundbox zeichnen			J	×1	y1	x2	y2	mst	Ein Rundeck mit Füllmuster mst (07) zeichnen; (immer Replace)								
			-				в	rtmap	Bilder Befehle								
Bild aus EEPROM Bild Iadan		U	E	×1	y1	nr	aton		internes Bild mit der nr (0255) aus dem EEPROM nach X1,y1 laden								
Did laden	ESC		L	XI	y 1		aten .		Ein blid nach x1,y1 laden, daten des blides siehe blidaufbau Es wird ein Bild angefordert. Zuerst werden die Breite und Höhe in Pixel und								
Hardcopy senden		н	×1	y1	×2	y2	- h l -	046-1	dann die eigentlichen Bilddaten über RS232 gesendet.								
Display läsaban					nspia	y-be	enie	(wine	Diselauiskelt lässkas (alle Bivel aus)								
Display loschen			L						Displayinnait loschen (alle Pixel aus)								
Display fillen			•						Displayinnait inventeren (alle Pixel ain)								
Display ausschatten			۵ ۵						Displayimat tulen (ale Fixer ein) Displayimat tulen (ale Fixer ein)								
Display einscheiten	ESC		F						Displayinhak wird displated of beith aber erhaken, bereine weiternin nogien								
Display emocrantem			-						Inhalt des Clipboards wird dargestellt. Displayausgaben sind nicht mehr								
Display Clipboard			С						sichtbar								
Disp. Normaldarstellung			Ν						Aktuelles Bild wird dargestellt (Normalbetrieb). Alle Ausgaben wieder sichtbar								
Display Reset			R						Der Displaykontroller wird per Befehl rückgesetzt und neu initialisiert								
								Ma	kro Befehle								
Makro ausführen			Ν	n1					Das (Normal-)Makro mit der Nummer n1 aufrufen (max. 7 Ebenen)								
i ouch Makro ausführen			т	n1					Das Touch-Makro mit der Nummer n1 aufrufen (max. 7 Ebenen)								
Port Makro ausführen	ESC	м	Р	n1				Das Port-Makro mit der Nummer n1 aufrufen (max. 7 Ebenen)									
autom. Makro zyklisch			Α	n1	n2	n3			Makros n1n2 automatisch zyklisch abarbeiten; n3=Pause in 1/10s								
autom. Makro pingpong			J	n1	n2	n3			Makros autom. von n1n2n1 (PingPong) abarbeiten; n3=Pause in 1/10s								
Abb 11 0)																	

(Abb. 11-8)

<b></b>								Dara	ranh	Pofe	ы		
	-		в					Barg	rapn	Bere		e	Deserants and Links) Diastia) Others) History withday 'or'
			I K			I 1							(1, 16) definieren x1 x1 x2 x2 sind das umschließende
Bargraph definieren		_	ō	nr	×1	y1	×2	y2	aw	ew	ľ	mst	Rechteck des Bargraphs, aw,ew sind die Werte für 0% und
	ESC	в	U			1	I .			I			100%. mst=Muster (07)
Bargraph zeichnen	L		nr	wer									Den Bargraph mit der Nummer nr (116) auf den neuen
				Cliv	abaa	rd Bo	foblo	/7wi	echor	nenoi	cł	hor	für Bildbereiche)
Displayinhalt sichern	<b>1</b>		в		JUUa		Territe	1241	Dero	noper tacami	la ha	Die	nlavinhalt wird als Bildheraich ins Clinhoard koniert
Bereich sichern	1		5	<b>v1</b>	v1	v2	v2		Der g	Rildher	ue ei	ich s	von v1. v1 his nach v2. v2 wird ins Clipboard kopiert
Display restaurieren	ESC	С	R	~	,	~	<i>j</i> -		Der F	Sildber	ei.	ch i	m Clinboard wird wieder ins Display zurückkoniert
Bereich konieren	1		к	×1	v1				Der F	Sildber	ei	ich i	m Clipboard wird ins Display pack x1_v1 kopiert
			15		<i>.</i> ,.	T	astat	ur / T	ouch	-Pane	əl	Be	fehle
Touch-Taste	<b>—</b>					Г.		<u></u>	00011	Die T	οι	uch-	Felder f1 bis f2 (gegenüberliegenden Eckfelder), werden zu
mit horizontaler	L		н			I I				einer	Т	ouc	h-Taste mit dem Rückgabewert 'Ret. Code' (=1255)
Beschriftung definieren						Ret.		Text	ыни	zusar	mr a Tr	men Tou	gefasst (Ret.Code=0 Touch-Taste nicht aktiv). uch-Taste (=0 nichte: =1 löschen: =2 mit Rahmen) zeichnen
Touch-Taste mit	1			l''	12	Code	Point		NOL	Text	5	es f	folgt eine Zeichenkette die zentriert mit dem akt. Font in der
vertikaler (90° gedreht)	L		v			I I				Touc	h-	Tas	te plaziert wird, mehrzeilige Texte werden mit dem Zeichen
Beschriftung definieren	1		_							'  (\$	70	3, de	ez: 124) getrennt; Zeichen NUL (\$00) = Zeichenkettenende
Touch-Tasten (P)Reset			P	<b> </b>					Alle	Touch-	11	aste	en werden aufsteigend aktiviert (Felder mit Code 160)
	1		ĸ	├──	<u> </u>				Alle a1=0	louch-	in.	aste	en werden deaktiviert (alle Felder mit Code U)
Touch Tastas Depiction	ESC	т	1	n1					n1=1	: Touc	h-	-Tas	ste wird beim Berühren automatisch invertiert
rouch-rasten Reakton	L		s	n1					n1=0	: kein	S	umn	ner beim Berühren einer (Touch-)Taste
	L		Ŭ						n1=1	: Sum	m	er p	iepst kurz beim Berühren einer (Touch-)Taste
Touch-Taste	L		м	n1					Die T	ouch-	Ta	aste	mit dem zugeordnetem Return-Code n1 wird manuell
Tosto manuell	1		<u> </u>	<u> </u>					Dis	uen			
abfragen	L		w						Die m desei	nomen ndet	18	in g	edruckte (Touch-)Taste wird auf der RS-232/RS-422
Tastan Abérana	1			<u> </u>					Tasta	aturabl	ira	ae	wird n1=0:deaktiviert:
Tasten-Abtrage Fin/Aus	L		Α	n1					n1=1	aktivi	er	t, Ti	astendrücke werden automatisch gesendet;
									n1=2	aktivi	er	t, Ta	astendrücke werden nicht gesendet (mit ESC T W abfragen)
	-						M	enu /	Рор	up Be	ite	ehle	9 In des Falles und auf Allenderstellen Man V – Belles allenes Falles
Menü	L					I I			Ein N Vertii	renu w kales r	nr me	nd ar emű	b der Ecke x1,y1 (Horizotales Menu = linke obere Ecke; = rechte obere Ecke) mit dem akt. Font gezeichnet
Finträgen definieren	L		м			I I	Text		nr:= a	aktueli	l ir	nver	tierter Eintrag (z.B: 1 = 1. Eintrag)
Menü mit vertikalen	1			x1	y1	nr	Text	NUL	Text:	= Zeic	he	enke	ette mit den Menüeinträgen. Die einzelnen Einträge sind durch
(90° gedrehten)	L		v			I I			Zeich Der k	ten ' ' ( lintera	(\$	70,0	dez:124) getrennt z.B. "Eintrag1 Eintrag2 Eintrag3" des Mentis wird automatisch ins Clinboard desichert
Einträgen definieren									ist be	ereits e	eir	n Me	enü definiert, wird dieses automatisch abgebrochen+entfernt.
Menübox invertieren	ESC.	N	1						Die g	esamt	e	Mer	nübox wird invertiert. Sinnvoll für negative Darstellung
nächster Eintrag			N						Der n	achst	e I	Eint	trag wird invertiert oder bleibt am Ende stehen
vorheriger Eintrag	L		Р						Der v	orheri	ge	e Ei	ntrag wird invertiert oder bleibt am Anfang stehen
Menüende / Senden	L		s						Das	Menü v	WÌ,	rd v	om Display entfernt und durch den Clipboardinhalt ersetzt
	1		<u> </u>		<u> </u>				Dec I	Ktuelle Menú s	-	en u	ag wird als Nummer (1n) gesendet (U=kein Menu dargestellt)
Menüende / Makro			м	nr					Für E	Eintrag	1	wir	d Makro 'nr' aufgerufen; für Eintrag 2 Makro nr+1 usw.
Menüende / Abbrechen	1		Α						Das I	Menü \	WÌ	ird v	om Display entfernt und durch den Clipboardinhalt ersetzt
							Kontr	oll-/	Defin	itions	s-l	Bef	fehle
			D	x1	y1	x2	y2		Defin	iert ei	ne	en B	linkbereich x1,y1 bis x2,y2; Blinkfunktion aktivieren
Automatisch	L		Z	n1	<u> </u>				Einst	ellen o	de	r Bl	inkzeit n1= 115 in 1/10s; 0=Blinkfunktion deaktivieren
blinkender Bereich	ESC	Q	м						Inven	s-Mod	us	8 (B	linkbereich wird invertiert); Blinkfunktion aktivieren
(Cursor-Funktion)	L		<u> </u>	mst	-				Clipb	oard-N	MO In	blin	s mst=Muster (U/) des Blockcursors; Blinken aktivieren
			с	n1					n1=0	Blink	fu	inkti	ion deaktivieren; n1=1: Blinkfunktion aktivieren (Invers, 6/10s)
			S	adr					Kit m	it Adre	28	se r	11 aktivieren (n1=255: alle)
Selekt / Deselekt	ESC	к	D	adr					Kit m	it Adre	26	se r	n1 deaktivieren (n1=255: alle)
			A	adr					Neue	Adres	55	e ac	dr zuweisen (z.B. im Power-On Makro)
Warten (Pause)	ESC	х	n1						n1 Ze	ehntel-	s	eku	nden abwarten bevor der nächste Befehl ausgeführt wird.
Summer Ein / Aus	ESC	J	n1	<u> </u>					n1=0	:Sumr	ne	er A	us; n1=1:Summer Ein; n1=2255:für n1 1/10s lang Ein
Bytes senden	ESC	s	anz		0	laten			Es wi	erden	ar	nz (	1255; 0=256) Bytes auf der RS-232/RS-422 gesendet bites /z B Ansteuerung eines externen seriellen Druckers)
			-			<u> </u>			Von d	dem B	aı	uste	in am I2C-Bus mit der Device Adresse adr werden anz
I2C-Bus lesen	ESC	Ľ.	ĸ	adr	anz	I I			(125	55; 0=:	25	56) i	Bytes angefordert und über die RS-232/RS-422 gesendet.
12C-Bus schreiben	ESC	1	w	adr	anz		laten		Auf d	iem 120	C-	Bus	s für den Baustein mit der Device Adresse adr werden
								-	anz (	1255	C, I	0=2	56) Bytes gesendet. daten = anz Bytes
	_							P	ort-Be	etenie	•		
Output-Port schreihen	L		w	1		I I			n1=0	Alle a	8.	Aus	gabe-Ports entsprechend n2 (#8-Bit Binärwert) einstellen e-Port n1 nicksetzen (n2=0): setzen (n2=1): invertieren
super or sentenen	1	l I			<u> </u>				(n2=2	2)	_	990	en en an næreere (nz=0), serzen (nz=1), inverderen
Fingshe-Port lesen	1		Р	<b>n</b> 1					n1=0	: Alle	8	Eing	gabe-Ports als 8-Bit Binärwert einlesen
Cirigaberr of tiesen	ESC	Y	n l						n1=1	8: Ei	ng	gabe	e-Port <n1> einlesen (1=H-Pegel=5V, 0=L-Pegel=0V)</n1>
Port Scan Ein/Aus	1		Α	n1					Der a	automa	ati	che	Scan des Eingabe-Port wird n1=0: deaktiviert; n1=1: aktiviert
Eingabe-Port invers	1		1	n1					Der E	Eingab	e-	Por	t wird n1=0: normal; n1=1: invertiert ausgewertet
Releuchtung Fin/Aus	1		1	<b>p1</b>					CFL/	LED-B	el	euc	htung n1=0: AUS; n1=1: EIN; n1=2: INVERTIEREN;
a streamining contracts	1		1						n1=3	255:	В	elei	uchtung für n1 Zehntel Sek lang einschalten

(Abb. 11-9)

### 11.3.2 Makroprogrammierung

Einzelne oder mehrere Befehlsfolgen können als sogenannte Makros zusammengefasst und im EEprom der HMI fest abgespeichert werden. Diese Makros können dann mit den Befehlen *Makro ausführen* gestartet werden.

Wie bereits in Kapitel 11.1 beschrieben, gibt es drei unterschiedliche Makroarten. Zwei dieser Makroarten werden in diesem Projekt verwendet.

Touchmakro:

- Start bei Berührung eines Touchfeldes.
- 255 Touchmakros können im EEprom der HMI abgelegt werden.

Normalmakro:

- Start durch Befehl "ESC MN nr" über die serielle Schnittstelle.
- Start durch Aufruf von einem anderen Makro aus
- 255 Normalmakros können im EEprom der HMI abgelegt werden
- Normalmakro 0 wird zur Initialisierung der HMI verwendet

Nachfolgend eine Übersicht der verwendeten Makros und ihrer Kurzbeschreibung:

- Makro 0: Startmakro, Initialisierung
- Makro 1: Intro, Startbildschirm
- Makro 2: Intro Sanduhr
- Makro 3: Aufruf Betriebsartenwahl
- Makro 4: Aufruf Betriebsartenwahl nach Störung
- Makro 5: Betriebsartenwahl nach Initialisierung des Displays
- Makro 6: Neuinitialisierung des Displays durch "restart-Taste"
- Makro 10: Grundmaske 1 Störbild
- Makro 11: Grundmaske 2 Störbild
- Makro 12: Grundmaske 3 Störbild
- Makro 12.
   Makro 20:
   Störung Paarfehler Endschalter Achse 1
- Makro 20. Störung Paarfehler Endschalter Achse 2
   Makro 21: Störung Paarfehler Endschalter Achse 2
- Makro 22: Störung Paarfehler Endschalter Achse 3
- Makro 23: Störung Paarfehler Endschalter Achse 4
- Makro 24: Störung Endlage + angefahren, Achse 1
- Makro 25: Störung Endlage + angefahren, Achse 2
- Makro 26: Störung Endlage + angefahren, Achse 3
- Makro 27: Störung Endlage + angefahren, Achse 4
- Makro 28: Störung Endlage angefahren, Achse 1
- Makro 29: Störung Endlage angefahren, Achse 2
- Makro 30: Störung Endlage angefahren, Achse 3
- Makro 31: Störung Endlage angefahren, Achse 4
- Makro 32: Störung Überlast, Achsen 1-4
- Makro 36: Störung keine Grundstellung
- Makro 37: Meldung Grundstellung schon vorhanden
- Makro 38: Meldung BA TeilAuto Position 1 erreicht
- Makro 39: Meldung BA TeilAuto Position 2 erreicht
- Makro 40: Meldung BA TeilAuto Position 3 erreicht
- Makro 41: Meldung BA TeilAuto Position 4 erreicht
- Makro 42: Meldung BA TeilAuto Position 5 erreicht
- Makro 43: Meldung BA umschalten nur bei inaktiven Achsen möglich
- Makro 44: Störung Olli aufrichten

•	Makro 45:	Warnung Olli aufrichten
•	Makro 46:	Störung Deckel nicht abgenommen
•	Makro 50:	Meldung BA Automatik Position 1 erreicht
•	Makro 51:	Meldung BA Automatik Position 2 erreicht
•	Makro 52:	Meldung BA Automatik Position 3 erreicht
•	Makro 53:	Meldung BA Automatik Position 4 erreicht
•	Makro 54:	Meldung BA Automatik Position 5 erreicht
•	Makro 55:	Meldung BA Automatik, P1-Fahrt aktiv
•	Makro 56:	Meldung BA Automatik, P2-Fahrt aktiv
•	Makro 57:	Meldung BA Automatik, P3-Fahrt aktiv
•	Makro 58:	Meldung BA Automatik, P4-Fahrt aktiv
•	Makro 59:	Meldung BA Automatik, P5-Fahrt aktiv
•	Makro 60:	Meldung BA Automatik, P6-Fahrt aktiv
•	Makro 61:	Meldung BA TeilAutomatik, P1-Fahrt aktiv
•	Makro 62:	Meldung BA TeilAutomatik, P2-Fahrt aktiv
•	Makro 63:	Meldung BA TeilAutomatik, P3-Fahrt aktiv
•	Makro 64:	Meldung BA TeilAutomatik, P4-Fahrt aktiv
•	Makro 65:	Meldung BA TeilAutomatik, P5-Fahrt aktiv
•	Makro 66:	Meldung BA TeilAutomatik, P6-Fahrt aktiv
•	Touchmakro 1:	Hilfsmakro zu Aufruf Makro2, Betriebsartenwahl
•	Touchmakro 2:	Automatikablauf Oberfläche 1
•	Touchmakro 3:	Teilautomatik Oberfläche
•	Touchmakro 4:	Handbetrieb Oberfläche 1
•	Touchmakro 5:	Teilautomatik Taste Position 4 anfahren
•	Touchmakro 6:	Teilautomatik Taste Position 1 anfahren
•	Touchmakro 7:	Teilautomatik Taste Position 5 antahren
•	Louchmakro 8:	Leilautomatik Laste Position 3 antahren
•	Touchmakro 9:	Teilautomatik Taste Position 2 anfahren
•	Touchmakro 10:	Tellautomatik Taste Position 0, Grundstellung anfahren
•		Handbetrieb Oberfläche 2 - Achse 1
-		Handbetrieb Oberfläche 4 - Achse 2
-		Handbetrieb Oberfläche 5 Achse 3
		Handbetrieb Ubernache 5 - Achse 4
	Touchinakio 15.	Handbetrieb Taste Richtung +, Achse 1
	Touchinakio 16.	Hanubelheb Tasle Richlung -, Achse T
	Touchinakio 17.	
-	Touchmakro 10:	Automatikablauf Oborfläche 2
-	Touchmakro 20:	Hilfo zu Toiloutomatikoborflächo 2
-	Touchmakro 21:	Handbetrieb Taste Richtung + Achse 2
-	Touchmakro 22:	Handbetrieb Taste Richtung - Achse 2
-	Touchmakro 23:	Handbetrieb Taste Richtung +, Achse 2
-	Touchmakro 24:	Handbetrieb Taste Richtung - Achse 3
	Touchmakro 25:	Handbetrieb Taste Richtung + Achse 4
	Touchmakro 26:	Handbetrieb Taste Richtung - Achse 4
-	Touchmakro 27	Störung guittieren
	Touchmakro 28	serielle Schnittstelle an Olli, Grdstllg fahren
	Touchmakro 29	Aufruf reset/restart Neuinitialisierung
	Touchmakro 30:	Achse nach Endlagenfahrt freifahren
	Touchmakro 50:	serielle Schnittstelle an Olli, BA Auto Freigabe aktiv
	Touchmakro 51:	serielle Schnittstelle an Olli, BA TeilAuto aktiv
•	Touchmakro 52:	serielle Schnittstelle an Olli, BA Hand aktiv

## 11.3.3 Quelltextauszug: Intro

Ist wird darauf verzichtet, den gesamten Quelltext hier darzustellen. Es werden jedoch einige Beispiele dargestellt.

Den gesamten Quelltext des Projektes finden Sie auf der beiliegenden Dokumentations-CDR.

Dieses Bild ist ein Teil des Intros. Wie auch im Ablaufdiagramm der Makros zu erkennen, wird zuerst das Makro 0 zur Initialisierung des Displays durchlaufen.



(Abb. 11-10)

Zum Intro gehören ebenso eine Reihe Bilder, die in einem kurzen zeitlichen Abstand auf den Bildschirm geladen werden, um den Eindruck eines Filmes zu hinterlassen.

Auszug des Quelltextes:

Makro:1

#DL	;Bildschirm löschen
#L 1,1	;Textmodus setzen
#FT 3	;Font Nummer3 einstellen
#QC 0	;Cursor unsichtbar
#QZ 6	;Blinktakt Cursor
#V 1	;Grafikmodus setzen
#F 2,1,1	;Font 2, kein Zoom
#W 0,0	;Cursor positionieren
#KS 255	;Adresse zuweisen
#YW 0,11111111	;alle Outputs rücksetzen
#TR	;alle TouchFelder deaktivieren
#YI 1	;Inputs invertieren
#DL	
#ZZ 80,5,"by     JR  &  L. C. Kulf       AF	PRIL 2004"
#X 30	;Wartezeit in 1/10s
#DL	;Bildschirm löschen

## 11.3.4 Quelltextauszug: Betriebsartenauswahl

Die Betriebsartenauswahl unterscheidet sich in den Bildern. Hier kommt es darauf an, zu welchem Zeitpunkt die Oberfläche der Betriebsartenwahl aufgerufen wird.



Im Beispiel dargestellt ist die Oberfläche der Betriebsartenauswahl, wie sie nach dem Start des Displays zu sehen ist.

Wird die Oberfläche zur Betriebsartenwahl nach einer Störung aufgerufen, wird die Oberfläche durch ein anderes Makro aufgerufen. Dort stehen dann andere Buttons und somit auch andere anzuwählende Funktionen zur Verfügung.

Auszug des Quelltextes:

Makro:3

#DL;Bildschirm löschen#TR;alle TouchFelder deaktivieren#F 4,1,1;Font 4, kein Zoom#X1;Wartezeit#ZZ 80,0 "Betriebsartenwahl";Text ausgeben (Maße in Pixel !)#F 1,1,1;Font 1, kein Zoom#TH 34,43,51,2,"TEIL-|AUTO"#TH 36,45,52,2,"HAND"#TH 38,47,29,2,"DISPLAY|RESTART"

## 11.3.5 Quelltextauszug: Oberfläche Teilautomatik

Die Oberfläche der Betriebsart Teilautomatik ist in Touchmakro 3 programmiert. Durch Berühren der aktiven Touchfelder wird intern ein Wert zurückgegeben, der Returnwert.



(Abb. 11-12)

Mit Hilfe dieses Returnwertes werden weitere Makros aufgerufen. Die Zahl des Rückgabewertes bestimmt die Nummer des aufzurufenden Makros.

Auszug des Quelltextes:

Touchmakro:3

#DL #TR #F 4,1,1 #ZZ 80,0 "Teilautomatik" #F 1,1,1	;Bildschirm löschen ;alle TouchFelder deaktivieren ;Font 4, kein Zoom ;Text ausgeben ;Font 1, kein Zoom
#TH 12,21,5,2,"POSITION 4"	;ret 5
#TH 28,37,6,2,"POSITION 1"	;ret 6
#TH 44,53,7,2,"POSITION 5"	;ret 7
#TH 26,35,8,2,"POSITION 3"	;ret 8
#TH 30,39,9,2,"POSITION 2"	;ret 9

#TH 46,55,10,2,"GRUND- STELLUNG"	;ret 10
#TH 1,9,1,2,"M E N U E"	;ret 1
#TH 8,16,17,2,"H I L F E"	;ret 17

### 11.3.6 Quelltextauszug: Oberfläche Betriebsart Hand

Die im Beispiel gezeigte Oberfläche zur Betriebsart Hand stellt nicht die ganzen Bedienungsmöglichkeiten in dieser Betriebsart dar.

Xmc Jm	Han	dbetr	ieb	HHHH
	ACHSE 1	ACHSE 4 HALS	ACHSE 3	
		ACHSE 2		

(Abb. 11-13)

Dieses Kapitel soll nicht die Oberflächen und ihre Funktionen erklären, sondern die Programmierung des LCD-Touchpanels dem Benutzer näher bringen.

Informationen zum Bedienen des Messautomaten entnehmen Sie bitte den entsprechenden Kapiteln.

Auszug des Quelltextes:

Touchmakro:4

#DL	;Bildschirm löschen
#TR	;alle TouchFelder deaktivieren
#F 4,1,1	;Font 4, kein Zoom
#ZZ 80,0 "Handbetrieb"	;Text ausgeben
#F 1,1,1	;Font 1, kein Zoom
#TH 26,35,11,2,"ACHSE 1"	;ret 11
#TH 44,53,12,2,"ACHSE 2"	;ret 12
#TH 30,39,13,2,"ACHSE 3"	;ret 13
#TH 28,37,14,2,"ACHSE 4 HALS"	;ret 14
#TH 1,9,1,2,"M E N U E"	;ret 1
#TH 8,16,18,2,"H I L F E"	;ret 18

## 11.3.7 Quelltextauszug: Störung Achse 4

In der Übersicht zur Verwendung der programmierten Makros in Kapitel 11.3.2 finden Sie eine sehr große Anzahl an Makros für Meldungen zur Bedienerführung.

Makros zur Visualisierung von diagnostizierten Störungen sind ebenso in großer Anzahl vorhanden.

In diesem Kapitel werden nicht sämtliche Makros zu Meldungen oder Störungen aufgeführt, sondern nur ein kleiner Teil zur Erläuterung der Programmierung.



(Abb. 11-14)

In diesem Makro ist die Verwendung eines in das EEprom des Displays geladenen Bildes im Dateiformat \*.bmp aufzuzeigen.

Auszug des Quelltextes:

;alle TouchFelder deaktivieren
;Font 4, kein Zoom
;Text ausgeben
;Stopschild laden
;ret 30

efahren"

Als Besonderheit ist in diesem Beispiel hervorzuheben, dass zum Zweck der Schonung der Speicherreserven im EEprom des Displays, zwei Bilder übereinander geladen werden.

Diese Technik wurde bei sämtlichen Visualisierungen von Störungen angewendet. Es wird jeweils ein Grundbild als Hintergrund geladen. Die eigentliche Störung, der Störungstext, steht im Vordergrund.

Der Vorteil ist, dass nicht für jedes Meldungs- oder Störungsbild das Bild mit allen enthaltenen Schriften, Grafiken oder Buttons programmiert werden muss.

Die compilierte Projektdatei *projekt.eep* wird somit wesentlich kleiner und benötigt deshalb im Speicher des Displays weniger Resourcen.

## 11.3.8 Quelltextauszug: Meldung Olli aufrichten

Dieses Bild ist ebenso wie das Bild in Kapitel 11.3.7 ein Beispiel für die Überlagerung mehrerer Makros.



Ist diesem Bild soll zusätzlich die Verwendung der seriellen Schnittstelle aufgezeigt werden. Über das Betätigen des Buttons "Störung quittieren" wird ein Makro aufgerufen, das über die serielle Schnittstelle einen definierten Befehl zum Mikrocontroller sendet.

Dieser Befehl setzt sich aus 4 Byte zusammen. Weitere Informationen zu diesen Befehlen, die über die serielle Schnittstelle versendet und empfangen werden, können Sie den Kapiteln zur Kommunikation entnehmen.

Auszug des Quelltextes:

;Störung Olli aufrichten Makro:44 #DL #TR

#F2,1,1 #TH 43,54,27,2,"Stoerung|quittieren" #F3,1,1 #ZZ 80,45,"Olli aufrichten !" #F2,1,1 #ZZ 80,60,"ACHTUNG !" #ZZ 80,68,"Nach Quittierung|werden Bewegungen|fortgesetzt !" #F 4,1,1 #ZZ 80,0, "Stoerung" #UE 5,5,14

;Störung quittieren Touchmakro:27 #S4,"A24x" #MN 4

## 11.4 Compilieren und Laden des Projekts

Um das Projekt in das EEprom des LCD-Displays laden zu können, muss dieses compiliert werden.

Dazu steht das Tool kitcompiler.exe zur Verfügung.

Nach Installation des Tools *kitcompiler.exe* müssen Bilder, die mit in das EEprom des Displays übertragen werden sollen, in das gleiche Verzeichnis wie die im ASCII-Editor erstellte *Projekt.txt* Datei, kopiert werden.

Bevorzugt wurde das root-Verzeichnis einer Partition verwendet, in der das Tool *kitcompiler.exe* installiert wurde.

Das LCD-Display wird mit der seriellen Schnittstelle eines kompatiblen PCs über eine RS 232 – Leitung verbunden.

Die Schnittstelle des Displays ist voreingestellt mit einer Datenübertragungsrate von 9600 Baud und dem Datenformat 8, N, 1.

Informationen, die zu Änderungen der Schnittstellenparameter benötigt werden, können Sie Kapitel 6.2.5 entnehmen.

Das Tool *kitcompiler.exe* wird durch einen Doppelklick auf das Icon gestartet. Geöffnet wird eine dos-Box in der nun der komplette Verzeichnispfad der Projektdateien angegeben wird.



Dateiendungen müssen mit angegeben werden. Wird darauf verzichtet, bricht der Compiler mit einer Fehlermeldung den Vorgang ab.

Das Projekt wird compiliert und eine *project.eep* automatisch erstellt. Ist die Verbindung zwischen Display und PC nicht fehlerhaft, wird die project.eep in das EEprom des Displays übertragen.

Nach erfolgreicher Übertragung der compilierten Datei, wird das Makro 0 des Displays automatisch gestartet und das Display initialisiert.

Die Spannungsversorgung des Displays muss während des gesamten Vorgangs gewährleistet sein.

Weitere Informationen zur Compilierung und Übertragung eines Projekts zum Display erhalten Sie auf der Dokumentations-CDR.

## **11.4** Testen der erstellten Bilder

Leider wird von der Fa. Electronic Assembly kein Tool zum Testen der Bilder nach dem download in das EEprom des Displays angeboten.

Da die Bilder ausschließlich textbasierend programmiert werden, kann somit ein Testen der Bilder nicht erfolgen.

Um diesen Missstand zu beheben, wurde das *STARTTOOL* entwickelt. Es bietet die Möglichkeit, einzelne Makros im LCD-Display über die serielle Schnittstelle aufzurufen.

So können die programmierten Bilder visualisiert und getestet werden. Fehler in den Grafiken werden somit schon im Vorfeld entdeckt und können beseitigt werden.

Ein weiterer großer Vorteil ist die Möglichkeit, Oberflächen zu entwickeln, ohne eine funktionierende Ansteuerung durch einen Mikrocontroller oder ein ähnliches Steuerungssystem zu haben.

In den folgenden Abschnitten wird kurz über das STARTTOOL informiert.

Das *STARTTOOL* muss nicht installiert werden. Über einen Doppelklick auf *starttool.exe* wird das Programm gestartet.

ፍ LCD Display	- IX
Makro - Starttool für EA KIT 160-7 von Electronic Assembly	
tool zum Statten von Makros des LCD-Displays über die serielle Schnittstelle CDM1 mit 9600 Baud.	
NormalMakro auswählen : NormalMakro starten	
TouchMakro auswählen : TouchMakro starten	
PortMakro auswählen : PortMakro starten	
Beenden	
	by jr '03

(Abb. 11-16)

Über die drei Pulldown-Menüs kann das anzuzeigende Makro ausgewählt werden. Selbst Touchmakros und Portmakros können aufgerufen werden.

🖷 LCD Display	
Makro - Starttool für EA KIT 160-7 von Electronic Assembly	
tool zum Statten von Makros des LCD-Displays über die serielle Schnittstelle COM1 mit 9600 Baud.	
NormalMakro auswählen : Normalmakro 6  VormalMakro starten	
TouchMakro auswählen :	
Touchmakro 255 TouchMakro starten	
PortMakro auswählen :	
Portmakro 2 PortMakro starten	
Portmakro 3	
Portmakro 5	
Portmakro 6 Portmakro 7	
Portmakro 8	
Portmakro 9	
	by jr '03

(Abb. 11-17)

Über einen Klick auf den entsprechenden Button, wird der Befehl zur Anforderung des ausgewählten Makros über die serielle Schnittstelle gesendet.

Das Bild wird durch den Aufruf des Makros visualisiert und kann getestet und nach Bedarf abgeändert werden.

Über den Button "Beenden" wird das Tool geschlossen.

Das STARTTOOL erhalten Sie auf der Dokumentations-CDR.

# 12 Bedienung des Messautomaten

## 12.1 Allgemein

Der Messautomat kann auf verschiedene Art und Weise bedient werden. Standartbedienung wird das Matlabprogramm Hasheraccelerator sein. Weiter kann der Messautomat ersatzweise durch das Tool OLLICONTROL bedient werden.

Zu Servicezwecken und in Sonderfällen kann der Messautomat ebenso durch die HMI bedient werden.

Eine Bedienung über die HMI ist im "normalen" Messablauf nicht notwendig. Hierzu wird die HMI nur als Visualisierungsmöglichkeit zur Bedienerführung genutzt.

## 12.2 Bedienung über die HMI

Zur Bedienung über die HMI des Messautomaten stehen auf dem Display verschiedene Oberflächen zur Verfügung.

Diese Oberflächen werden in diesem Kapitel erklärt.

## 12.2.1 Betriebsartenwahl



Diese Oberfläche wird nach dem Hochlauf des Systems angezeigt. Drei Softkeys stehen zur Auswahl :

- TEILAUTOMATIK
- HAND
- DISPLAY RESTART

Softkey TEILAUTOMATIK:

- Über den Softkey TEILAUTOMATIK kann in die Betriebsart Teilautomatik gewechselt werden.
- In die Betriebsart Teilautomatik kann nur gewechselt werden, wenn sich der Messautomat in Grundstellung befindet.

 Wird versucht in die Betriebsart Teilautomatik zu wechseln, ohne dass der Messautomat sich in Grundstellung befindet, wird eine Meldung zur Bedienerführung angezeigt.

### Softkey HAND:

- Über den Softkey HAND kann in die Betriebsart Hand gewechselt werden.
- In die Betriebsart Hand kann jederzeit gewechselt werden.

### Softkey DISPLAYRESTART:

 Über den Softkey DISPLAYRESTART kann das LCD-Display neu gestartet und initialisiert werden.

## 12.2.2 Betriebsartenwahl nach Störung



Diese Oberfläche wird nach dem erfolgreichen Quittieren einer quittierpflichtigen Störung angezeigt. Fünf Softkeys stehen zur Auswahl :

- TEILAUTOMATIK
- HAND
- FREIGABE AUTO
- STÖRUNG QUITTIEREN
- DISPLAY RESTART

### Softkey TEILAUTOMATIK:

• Gleiche Funktion wie in Kapitel 12.2.1

Softkey HAND:

• Gleiche Funktion wie in Kapitel 12.2.1

Softkey DISPLAYRESTART:

• Gleiche Funktion wie in Kapitel 12.2.1

Softkey FREIGABE AUTO:

- Mit diesem Softkey kann die Freigabe zum Wechsel in Betriebsart Automatik erteilt werden.
- Ein Wechsel in die Betriebsart Automatik ist nur zulässig, wenn sich der Messautomat in Grundstellung befindet.
- Wird versucht in die Betriebsart Teilautomatik zu wechseln, ohne dass der Messautomat sich in Grundstellung befindet, wird eine Meldung zur Bedienerführung angezeigt.

Softkey STÖRUNG QUITTIEREN:

- Mit diesem Softkey kann eine noch anstehende Störung quittiert werden.
- Es wird nochmals die Anforderung zur Quittierung einer Störung über die serielle Schnittstelle an den Mikrocontroller gesendet.

### 12.2.3 Betriebsart Teilautomatik



<sup>(</sup>Abb. 12-3)

Über die Oberfläche zur Betriebsart Teilautomatik ist es möglich, den Messautomat auf die einzelnen Messpositionen zu fahren.

Eine bestimmte Reihenfolge zum Anfahren der Positionen muss nicht eingehalten werden. Die zum Anfahren ausgewählte Position wird grundsätzlich über die Grundstellung angefahren.

Zu beachten ist, dass die Nummerierung der Positionen den mechanischen Positionen des Messautomaten entspricht. Die Nummerierung der Messpositionen wird auf den in Fahrtrichtung rechten Sitzen in vertikaler Linie gespiegelt.

Die Grundstellung bleibt selbstverständlich die Selbe.

Auf dieser Oberfläche stehen acht Softkeys zur Auswahl, sechs davon zur Auswahl einzelner Messpositionen, ein Softkey zur Navigation zwischen den Oberflächen und ein Softkey zum Aufruf der Hilfeseiten zur Oberfläche.

Die Funktionen der Softkeys:

- MENÜ
- HILFE
- POSITION 1
- POSITION 2

- POSITION 3
- POSITION 4
- POSITION 5
- GRUNDSTELLUNG

Softkey MENÜ:

 Mit diesem Softkey kann wieder zurück zur Oberfläche der Betriebsartenauswahl gesprungen werden. Er dient zur Navigation zwischen den Oberflächen.

Softkey HILFE:

 Mit diesem Softkey werden die Hilfeseiten zur Oberfläche Teilautomatik aufgerufen. Hier werden Hinweise zu den einzelnen Messpositionen gegeben. (siehe Abbildung 12-).

Softkey POSITION 1:

Mit diesem Softkey wird der Messautomat zur mechanischen Position 1 bewegt.

Softkey POSITION 2:

Mit diesem Softkey wird der Messautomat zur mechanischen Position 2 bewegt.

Softkey POSITION 3:

Mit diesem Softkey wird der Messautomat zur mechanischen Position 3 bewegt.

Softkey POSITION 4:

Mit diesem Softkey wird der Messautomat zur mechanischen Position 4 bewegt.

Softkey POSITION 5:

• Mit diesem Softkey wird der Messautomat zur mechanischen Position 5 bewegt.

Softkey GRUNDSTELLUNG:

- Mit diesem Softkey wird der Messautomat zur Grundstellung bewegt. Grundstellung bedeutet, dass alle vier Schlitten der Antriebsachsen die Referenzpunktschalter betätigt haben.
- Die Grundstellung in der Betriebsart Teilautomatik unterscheidet sich nicht von den Grundstellungen in anderen Betriebsarten.

### 12.2.4 Betriebsart Hand

Für die Betriebsart Hand stehen mehrere Oberflächen zur Verfügung. Diese Oberflächen werden benötigt, um den Messautomaten einrichten zu können.

In dieser Betriebsart sind die mächtigsten Eingriffe in das System möglich.



Der Messautomat sollte nur von geschultem Fachpersonal in dieser Betriebsart bedient werden !

In den nächsten Abschnitten erhalten Sie Informationen zur Bedienung des Messautomaten in der Betriebsart Hand.

In dieser Betriebsart ist es möglich, die Schlitten der Antriebseinheiten auf die Endlagenschalter der Achsen zu fahren. In diesem Fall wird die Bedienungsgewalt dem Benutzer entzogen und der Messautomat schaltet selbsttätig intern auf die Betriebsart Teilautomatik um.

In der Betriebsart Teilautomatik erfolgt nach Bestätigung durch den Benutzer ein Freifahren der Antriebsachsen von den Endlagen.



(Abb. 12-4)

In dieser Oberfläche zur Betriebsart Hand stehen sechs Softkeys zur Bedienung zur Auswahl, ein Softkey davon zur Navigation zwischen den Oberflächen, ein Softkey zum Aufruf der Hilfeseiten zur Oberfläche.

Über die restlichen vier Softkeys können die Achsen angewählt werden, die im Handbetrieb verfahren werden sollen.

Sechs Softkeys stehen zur Auswahl:

- MENÜ
- HILFE
- ACHSE 1
- ACHSE 2
- ACHSE 3
- ACHSE 4 HALS

Softkey MENÜ:

 Mit diesem Softkey kann wieder zurück zur Oberfläche der Betriebsartenauswahl gesprungen werden. Er dient zur Navigation zwischen den Oberflächen.

Softkey HILFE:

 Mit diesem Softkey werden die Hilfeseiten zur Oberfläche Hand aufgerufen. Hier werden Hinweise zu den einzelnen Achsen gegeben. (siehe Abbildung 12-).

Softkey ACHSE 1:

 Mit diesem Softkey wird die Achse 1 zum Verfahren im Handbetrieb angewählt. Die Achse führt durch Betätigen dieses Softkeys noch keine Bewegung aus.

Softkey ACHSE 2:

- Mit diesem Softkey wird die Achse 2 zum Verfahren im Handbetrieb angewählt. Die Achse führt durch Betätigen dieses Softkeys noch keine Bewegung aus.
- Softkey ACHSE 3:
  - Mit diesem Softkey wird die Achse 3 zum Verfahren im Handbetrieb angewählt. Die Achse führt durch Betätigen dieses Softkeys noch keine Bewegung aus.

Softkey ACHSE 4 HALS:

 Mit diesem Softkey wird die Achse 4 zum Verfahren im Handbetrieb angewählt. Die Achse führt durch Betätigen dieses Softkeys noch keine Bewegung aus.

Durch das Betätigen der Softkeys ACHSE 1-4 werden weitere Oberflächen zur Bedienung des Messautomaten im Handbetrieb aufgerufen. In den weiteren Abschnitten werden nicht alle Oberflächen dargestellt.

Handb	etrieb
Ach	ise 1
RICHTUNG	RICHTUNG

In Abbildung 12-5 wird ersatzweise für alle Achsen die weiterführende Oberfläche zur Bedienung der Achsen dargestellt.

Diese Oberfläche steht für alle vier Achsen zur Verfügung. Sie unterscheiden sich nur im Schriftzug "Achse 1". Die Oberfläche zur Bedienung von Achse 2 enthält den Schriftzug "Achse 2", usw.

Auf dieser Oberfläche stehen drei Softkeys zur Verfügung:

- MENÜ
- RICHTUNG +
- RICHTUNG –

Softkey MENÜ:

 Mit diesem Softkey kann wieder zurück zur letzten Oberfläche gesprungen werden. Er dient zur Navigation zwischen den Oberflächen.

Softkey RICHTUNG +:

 Durch das Betätigen dieses Softkeys wird die angewählte Achse um zwei Spindelumdrehungen in die Verfahrrichtung (+) bewegt. Um die Anzahl der zu

<sup>(</sup>Abb. 12-5)

verfahrenden Spindelumdrehungen zu ändern, lesen Sie Kapitel 9.4.15, Details zur Programmierung der Steuerung.

 Zwei Spindelumdrehungen entsprechen einem linearen Verfahrweg von zwei Millimetern.

Softkey RICHTUNG -:

- Durch das Betätigen dieses Softkeys wird die angewählte Achse um zwei Spindelumdrehungen in die Verfahrrichtung (-) bewegt. Um die Anzahl der zu verfahrenden Spindelumdrehungen zu ändern, lesen Sie Kapitel 9.4.15, Details zur Programmierung der Steuerung.
- Zwei Spindelumdrehungen entsprechen einem linearen Verfahrweg von zwei Millimetern.

Zur besseren Übersicht werden nachfolgend die einzelnen Oberflächen zur Bedienung der vier Achsen im Handbetrieb dargestellt. Auf Erklärungen wird jedoch verzichtet.



(Abb. 12-6,7,8)

Zu Erklärungen dieser Oberflächen nehmen Sie bitte die ersten Abschnitte dieses Kapitels zur Hilfe. Diese Hinweise zu den Oberflächen sind neutral gehalten und können somit auf diese Abbildungen übertragen werden.

### 12.2.5 Online-Hilfe zur Betriebsart Teilautomatik

In den Betriebsarten Teilautomatik und Hand steht Ihnen eine Online-Hilfe zur Verfügung. Diese wird jeweils in den Oberflächen zur Betriebsart Teilautomatik und Betriebsart Hand aufgerufen. Dazu steht der Softkey HILFE zur Verfügung.



In Abbildung 12-9 ist der erste Teil der Online-Hilfe zur Betriebsart Teilautomatik dargestellt. In dieser Oberfläche können Informationen zu den einzelnen Positionen eingeholt werden.

In dieser Oberfläche können zwei Softkeys zur Navigation verwendet werden.

- MENÜ
- PLUS

Softkey MENÜ:

 Mit diesem Softkey kann wieder zurück zur letzten Oberfläche gesprungen werden. Er dient zur Navigation zwischen den Oberflächen.

Softkey PLUS:

• Mit diesem Softkey kann auf die zweite Seite der Online-Hilfe zur Betriebsart Teilautomatik gesprungen werden. Er dient zur Navigation zwischen den Oberflächen.



In Abbildung 12-10 ist der zweite Teil der Online-Hilfe zur Betriebsart Teilautomatik dargestellt.

Ein Softkey steht zur Bedienung zur Verfügung:

MENÜ

Softkey MENÜ:

 Mit diesem Softkey kann wieder zurück zur letzten Oberfläche gesprungen werden. Er dient zur Navigation zwischen den Oberflächen.

## 12.2.6 Online-Hilfe zur Betriebsart Hand

Wie auch zur Betriebsart Teilautomatik steht auch in der Betriebsart Hand eine Online-Hilfe zur Verfügung.

mean	Hilfe
Achse	vorwachlen und mit
Taste	Richtung + und
Taste	Richtung - verfahren.
Achse	1: kippen nach rechts
Achse	2: kippen in Fahrtrichtung
	3: kippen nach links
Achse	4: Mikrofon Hub
(Abb. 12	2-11)

Ein Softkey steht zur Bedienung zur Verfügung:

MENÜ

Softkey MENÜ:

 Mit diesem Softkey kann wieder zurück zur letzten Oberfläche gesprungen werden. Er dient zur Navigation zwischen den Oberflächen.

### 12.2.7 Bedienerführung: Störungen

### Störungen durch angefahrene Endlagen

Zur Visualisierung von Störungen steht eine Vielzahl von Oberflächen und Bildern zur Verfügung. Ein kleiner Auszug hiervon wird in den nächsten Abschnitten dargestellt.



(Abb. 12-12)

In Abbildung 12-12 ist eine Störung der Achse 4 dargestellt. Es wurde die Endlagenüberwachung in Verfahrrichtung (-) betätigt.

Diese Diagnose steht für alle vier Achsen zur Verfügung. Selbstverständlich werden nicht nur die Endlagen der negativen Verfahrrichtungen überwacht, sondern auch die Endlagen der positiven Verfahrrichtungen.

Es ergeben sich hieraus acht Oberflächen zur Visualisierung dieser Art von Störungen. Es steht je Oberfläche ein Softkey zur Bedienung zur Verfügung.
ACHSE FREIFAHREN

Softkey ACHSE FREIFAHREN:

- Durch Betätigen des Softkeys ACHSE FREIFAHREN wird die Achse automatisch von der entsprechenden Endlage gefahren.
- Der Bediener hat keinen Einfluss auf die Verfahrrichtung der Achse.
- Die Verfahrrichtung ist entgegengesetzt der Endlagenfahrt und wird vom System selbstständig diagnostiziert.
- Die Störung ist quittierpflichtig.

## Störungen durch Paarfehler

Zur Endlagenüberwachung stehen weitere Oberflächen zur Verfügung. Die Antriebseinheiten werden auch auf Paarfehler überwacht.



Von Paarfehlern spricht man, wenn das unzulässige Ansprechen beider Endlagenschalter einer Achse diagnostiziert wird.

Auf eine Darstellung der vier Oberflächen zur Visualisierung der Störungen durch Paarfehler wird hier bewusst verzichtet.

Die Störungen durch Paarfehler führen zum sofortigen Stillstand der Antriebsachsen und sind quittierpflichtig.

Diese Art Störungen tritt in der Regel nicht auf und spielt deshalb hier eine untergeordnete Rolle.

Durch die Diagnose dieser Störungen werden defekte Endlagenschalter und eine eventuell beschädigte Verdrahtung der Endlagenschalter analysiert.

Die Endlagenschalter sind als Öffnerkontakte ausgeführt und deshalb als kabelbruchsicher zu bezeichnen.

Im Kapitel 6.2.4.2 erhalten Sie mehr Informationen zur Elektrik und zur Beschaltung der Sensorik einer Antriebseinheit.

## Störungen durch Fehlbedienung

Eine fehlerfreie Bedienung kann auch durch noch so exakte Bedienerführung nicht ausgeschlossen werden.

Eine Zerstörung dieses Systems ist durch unbeabsichtigte Fehlbedienung nahezu ausgeschlossen.

Eine fehlerhafte Bedienung wird erkannt und ruft Meldungen zur Bedienerführung hervor.

STOP	rung
keine Grun	ndstellung
Grund- stellung fahren	Stoerung quittieren

(Abb. 12-13)

In Abbildung 12-13 wird die Visualisierung zur Störung "keine Grundstellung" dargestellt. Diese Störung kann hervorgerufen werden, wenn zu unzulässigem Zeitpunkt ein Betriebsartenwechsel angefordert wurde, d. h. ein Betriebsartenwechsel wurde angefordert, obwohl der Messautomat sich nicht in Grundstellung befand.

Zwei Softkeys stehen in der dieser Oberfläche zur Bedienung zur Verfügung.

- GRUNDSTELLUNG FAHREN
- STOERUNG QUITTIEREN

## Softkey STOERUNG QUITTIEREN:

- Die Störmeldung ist quittierpflichtig und kann mit diesem Softkey quittiert werden.
- Die Störung muss zur Quittierung beseitigt werden. Erfolgt keine Beseitigung der Störung, erscheint die Oberfläche erneut.

#### Softkey GRUNDSTELLUNG FAHREN

- Mit diesem Softkey wird der Messautomat intern auf die Betriebsart Teilautomatik umgeschalten.
- Der Messautomat führt selbstständig eine Grundstellungsfahrt durch.
- Das System zur Lageüberwachung wird initialisiert.
- Die Störung ist nun quittierbar.

Der Messautomat kann bei dieser Störung nicht in der Betriebsart Hand auf die Grundstellung gefahren werden.



Eine Grundstellungsfahrt ist immer automatisiert durchzuführen, da eine Initialisierung des Systems zur Lageüberwachung der Antriebsachsen zur korrekten Positionierung der Achsen notwendig ist.

Eine Initialisierung des Systems zur Lageüberwachung der Antriebsachsen wird ausschließlich durch eine automatisierte Grundstellungsfahrt durchgeführt.

## Störung Deckelüberwachung

In Kapitel 6.2.4.3 wurde bereits auf diese Überwachung eingegangen. Nähere Informationen zur Elektrik und zum Service erhalten Sie in Kapitel 6 dieses Handbuches.

Stoerung
STOP
Deckel nicht abgenommen ! Olli ist nicht in Arbeitsstellung ! ACHTUNG ! Nach Quittierung werden Bewegungen fortgesetzt !
Stoerung quittieren
(Abb. 12-14)

Diese Störung tritt auf, wenn nach der Herstellung der Verbindung des Messautomaten mit der Netzversorgung, der Deckel des Gehäuses nicht abgenommen wurde.

Um eine Beschädigung des Messautomaten durch diese Art der fehlerhaften Inbetriebnahme durch den Bediener zu verhindern, werden als sofortige Reaktion alle Achsbewegungen stillgesetzt.

Beim Neustart des Systems wird die Referenzpunktfahrt erst ausgelöst, wenn der Gehäusedeckel des Messautomaten abgenommen wurde.

Diese Störung ist quittierpflichtig.

In dieser Oberfläche steht ein Softkey zur Bedienung zur Verfügung.

STOERUNG QUITTIEREN

Softkey STOERUNG QUITTIEREN

- Die Störmeldung ist quittierpflichtig und kann mit diesem Softkey quittiert werden.
- Die Störung muss zur Quittierung beseitigt werden. Erfolgt keine Beseitigung der Störung, erscheint die Oberfläche erneut.
- Nach Beseitigung der Ursache und Quittierung der Störung wird die zuletzt aktive Bewegung des Messautomaten fortgesetzt.



Nach erfolgreicher Quittierung wird die zuletzt aktive Bewegung des Messautomaten fortgesetzt !

## Störung Neigungssensor

Der Messautomat ist mit einem Neigungssensor versehen. Nähere Informationen zum Neigungssensor erhalten Sie im Kapitel zur Sensorik.



(Abb. 12-15)

Durch den Neigungssensor werden unzulässige Neigungswinkel des Messautomaten überwacht.

Beim Verlassen dieses durch diesen Neigungssensor festgelegten Arbeitsbereich wird diese Störung generiert.

Als Reaktion auf diese Störung werden alle Antriebsachsen mit sofortiger Wirkung gestoppt. Dadurch wird eine mögliche Beschädigung des Messautomaten verhindert.

Auf der Oberfläche zu dieser Störung steht ein Softkey zur Bedienung zur Verfügung.

STOERUNG QUITTIEREN

Softkey STOERUNG QUITTIEREN:

- Die Störmeldung ist quittierpflichtig und kann mit diesem Softkey quittiert werden.
- Die Störung muss zur Quittierung beseitigt werden. Erfolgt keine Beseitigung der Störung, erscheint die Oberfläche erneut.
- Nach Beseitigung der Ursache und Quittierung der Störung wird die zuletzt aktive Bewegung des Messautomaten fortgesetzt.



Nach erfolgreicher Quittierung wird die zuletzt aktive Bewegung des Messautomaten fortgesetzt !

## Störung Überlast

Ist eine oder mehrere Achsen des Messautomaten mechanisch schwergängig oder blockiert, wird dies von den Motorsteuer-ICs des Leistungsteils der Basisplatine diagnostiziert.

Eine Visualisierung der Störung findet durch die HMI statt.

Auf der Oberfläche zu dieser Störung steht ein Softkey zur Bedienung zur Verfügung.

STOERUNG QUITTIEREN

## Softkey STOERUNG QUITTIEREN:

- Die Störmeldung ist quittierpflichtig und kann mit diesem Softkey quittiert werden.
- Die Störung muss zur Quittierung beseitigt werden. Erfolgt keine Beseitigung der Störung, erscheint die Oberfläche erneut.



Das mehrfache Quittieren dieser Störung kann großen mechanischen Schaden hervorrufen ! Die Antriebseinheiten des Messautomaten sind unbedingt einer Sichtprüfung zu unterziehen !

Allgemeiner Hinweis :

Störungen, die hier nicht aufgeführt sind, können den Ablaufdiagrammen zur HMI in Kapitel 11.2 entnommen werden !

## 12.2.8 Bedienerführung: Meldungen

Meldungen zur Bedienerführung stehen in einer sehr großen Anzahl zur Verfügung. Es wird darauf verzichtet, alle Meldungen hier darzustellen.

Achsspezifische Meldungen stehen selbstverständlich zu allen Achsen zur Verfügung. Ersatzweise wird in diesem Kapitel jeweils die Meldung einer Achse dargestellt.



Meldungen sind nicht quittierpflichtig !

Meldungen werden auf Grund eines Vorganges aufgerufen und werden zeitabhängig angezeigt.

## Meldungen beim Start des Systems





(Abb. 12-16,17)

Beim Neustart des Systems werden verschiedene Initialisierungsroutinen durchlaufen, während dieser Zeit informiert die HMI in einem Intro über den Titel des Projekts.

## Meldung: Fahrt auf Position aktiv

Meldung	
Fahrt auf Position 2 aktiv !	

(Abb. 12-18)

Wird in der Betriebsart Automatik oder Teilautomatik eine Positionierung des Messautomaten auf eine definierte Messposition vorgenommen, zeigt die HMI die aktive Fahrt auf die entsprechende Position an.

Diese Meldung steht natürlich nicht nur für die Fahrt auf Position 2 zur Verfügung, sondern für die folgende Positionen:

- Position 1
- Position 2
- Position 3
- Position 4

Position 5

Die Fahrt auf die Grundstellung wird nicht angezeigt. Aus folgenden Gründen wurde darauf verzichtet, die Grundstellungsfahrt explizit anzuzeigen:

- Die Grundstellungsfahrt wird bewusst in Betriebsart Teilautomatik ausgewählt.
- Jedes Positionieren in der Betriebsart Automatik oder Teilautomatik hat immer eine Grundstellungsfahrt voraus.

### Meldung: Position erreicht

Ist die Betriebsart Automatik oder die Betriebsart Teilautomatik angewählt, und wurde eine definierte Position automatisch oder halbautomatisch angefahren, wird dies über eine Meldung wie in Abbildung 12-19 visualisiert.

Meldung	
Positio	n 5 erreicht

(Abb. 12-19)

Diese Meldung steht natürlich nicht nur für das Erreichen der Position 5 zur Verfügung, sondern für die folgende Positionen:

- Position 1
- Position 2
- Position 3
- Position 4
- Position 5

### Meldung: Achse aktiv

Wird versucht, zu einem unzulässigen Zeitpunkt die Betriebsart zu wechseln, wird eine Meldung zur Bedienerführung ausgegeben.

Die Betriebsart darf nur gewechselt werden, wenn sich alle Achsen und somit der Messautomat in Grundstellung befinden.

In die Betriebsart Hand darf auch gewechselt werden, wenn der Messautomat nicht in Grundstellung ist. Allerdings darf zum Zeitpunkt der Anforderung der Betriebsart keine Achse in Bewegung sein.

Wird die Betriebsart Hand unter Missachtung dieser Punkte angefordert, wird die Meldung "Achse aktiv" ausgegeben und die Anforderung abgewiesen.

## Meldung: Messung starten

Wurde dem Messautomaten die Freigabe zur Betriebsart Automatik entzogen und in eine andere Betriebsart gewechselt, so muss bei Wiederanforderung der Betriebsart Automatik, die Freigabe bestätigt werden.



(Abb. 12-20)

Wurde die Freigabe zum Wechsel in die Betriebsart Automatik durch den Bediener bestätigt, wird dies durch diese Meldung visualisiert.

Eine Bedienung des Messautomaten durch das Tool OLLICONTROL oder das Matlab-Programm Hasheraccelerator kann nun erfolgen.

Allgemeiner Hinweis :

Meldungen, die hier nicht aufgeführt sind, können den Ablaufdiagrammen zur HMI in Kapitel 11.2 entnommen werden !

## 12.3 Bedienung über OLLICONTROL

Das Tool OLLICONTROL wurde entwickelt, um den Messautomaten unabhängig von der eigentlichen Software, dem Matlab-Programm Hasheraccelerator, das die Ansteuerung des Automaten im Messaufbau durchführt, entwickeln und später auch zu Test- und Servicezwecken bedienen zu können.

Außerdem eignet sich das Tool OLLICONTROL, um die Leistungsfähigkeit des Messautomaten demonstrieren zu können.

In erster Linie wurde das Tool aber entwickelt, um eine Bedienung des Automaten in Unabhängigkeit eines Messaufbaus zu ermöglichen.

## 12.3.1 Installation

Das Tool OLLICONTROL muss auf der Festplatte eines kompatiblen PCs installiert werden. Durch einen Doppelklick auf die *setup.exe* wird der Install-Shield geöffnet. Folgen Sie den weiteren Anweisungen auf dem Bildschirm.

Das Tool OLLICONTROL ist auf der beiliegenden Dokumentations-CDR enthalten und kann von dieser CD aus installiert werden.

## 12.3.2 Vorbereitung

Als Vorbereitung zur Bedienung muss der Messautomat über die RS 232 – Schnittstelle der Adapterplatine mit der COM – Schnittstelle des PCs verbunden werden.

Optional ist hier ersatzweise eine RS 232 – Funkverbindung im System integriert. Weitere Informationen hierzu erhalten Sie im Kapitel zur Adapterplatine.

## 12.3.3 Bedienung OLLICONTROL

Das Tool OLLICONTROL wird durch einen Doppelklick auf *ollicontrol.exe* gestartet. Optional kann eine Verknüpfung auf den Desktop des Rechners generiert werden.



Während die Anwendung im Hintergrund gestartet wird, ist ein Hinweis, der über die verwendete Version informiert, geöffnet.

Es erfolgt automatisch eine Weiterleitung zur eigentlichen Bedienungsoberfläche, von welcher aus der Messautomat kontrolliert werden kann.

😓 OLLI Control	
Olli Control Switch	AUTOMOTIVE
Info	
Automatic	
Semiautomatic	Service March
Port Settings	
Detter sound through research	Quit

(Abb. 12-22)

Geöffnet wird das Grundbild der Anwendung. Dieses Bild stellt die Betriebsartenauswahl dar. Folgende fünf Buttons stehen zur Auswahl, die Bezeichnung der Buttons ist in englischer Sprache ausgeführt und wurde nach Möglichkeit zum besseren Verständnis übersetzt:

- Info
- Automatik
- Teilautomatik
- Port Settings
- Beenden

Über den Button "Info" wird ein Informationsfenster geöffnet, das die üblichen Informationen wie Versionsnummer der Anwendung, usw. anzeigt. Das Fenster wird über den Button "OK" wieder geschlossen.



Über den Button "Port Settings" wird ein Informationsfenster geöffnet, das die Einstellungen der seriellen Schnittstelle RS 232 anzeigt.

Ein Einstellen der Schnittstellenparameter ist bei dieser Version noch nicht möglich. Diese Option ist in Vorbereitung. Das Fenster wird über den Button "OK" geschlossen.

<b>i</b> Information	
The default Settings are:	
Com Port 1 9600 Baud 8 N 1	
It's not possible to change this settings in this ¥ersion. ¥ersion 1.2.331	
ΟΚ	

(Abb. 12-24)

In den folgenden Abschnitten erhalten Sie Informationen zur Bedienung des Messautomaten durch die Software OLLICONTROL.

Hauptsächlich wird versucht, auf die Bedienung der Software einzugehen.

Alle Fehlermeldungen, die OLLICONTROL generieren kann, werden in den folgenden Abschnitten nicht dargestellt.

Die Fehlermeldungen oder Hinweise zur Bedienerführung sind in der Regel selbsterklärend und bedürfen keiner weiteren Kommentierung.

Sollten Sie aber doch wider Erwarten Probleme mit dem Umgang der Software OLLICONTROL haben, wenden Sie sich bitte an Ihren **BUSSE** - Ansprechpartner.

### **Betriebsart Automatik**

Über den Button "Automatik" im Hauptfenster wird die Oberfläche zur Bedienung des Messautomaten (Abbildung 12-25) in der Betriebsart Automatik aufgerufen.

Hier steht ein Button zum Schließen dieses Fensters zur Verfügung.

≿ Olli Control	
Automatic Olli	Control
Start auto. measurment	Traffic on RS232
Meas- / Seatposition 🔽	T
Driver-Seat Front-Passenger-Seat Back-Left-Passenger-Seat Back-Right-Passenger-Seat	● Spy on Rx ● Spy on Tx
	Close

(Abb. 12-25)

Über ein Pulldown-Menü muss zuerst die Position des Messautomaten im Kraftfahrzeug ausgewählt werden. Zur Verfügung stehen folgende Positionen:

- auf dem Fahrersitz
- auf dem Beifahrersitz
- auf dem Sitz hinter dem Fahrer
- auf dem Sitz hinter dem Beifahrer

In Abhängigkeit von der gewählten Position im Kraftfahrzeug werden anschließend die verschiedenen zu messenden Positionen des Messautomaten angefahren.

In dieser Oberfläche ist ein kleines Fenster integriert, über das der Datenverkehr über die RS 232 – Schnittstelle beobachtet werden kann. Es kann ausgewählt werden, ob die empfangenen Daten visualisiert werden oder ob die gesendeten Daten angezeigt werden.

Das Fenster zum Anzeigen der Daten ist mit "Traffic on RS 232" bezeichnet. Das Anzeigen der gesendeten Daten, "Spy on Tx", ist die Voreinstellung.

Zum Anzeigen der empfangenen Daten muss die checkbox "Spy on Rx" angeklickt werden. Ein Umschalten ist jederzeit möglich.

Zum Start des Automatikablaufes muss der Button "start automatic measurement" gedrückt werden – die automatische Messung wird gestartet.

<b>1</b> Information	
Incorrect Handling	
You must first select the seatposition, where 'Olli' is standing, before you can start the measurment. Please select the seat by choosing one of them, in the 'Meas- / Seatposition' Dropdown-menue !	
ОК	

(Abb. 12-26)

Wird der Button zum Start der Messung gedrückt, ohne dass eine Position im Kraftfahrzeug ausgewählt wurde, wird der Benutzer über seine Fehlbedienung informiert. Diese Meldung über unkorrekte Handhabung des Systems ist quittierpflichtig.

<b>i</b> Information	
'Olli Control' received ' nothing ' ! This is a timeout error.	
'Olli Control' wants to receive a answer within 1 minute maximum. Possible reasons for this error are:	
- connection lost - 'Olli' is switched off - 'Olli' is sleeping You can 'Retry' your measurment with the actual position, or you can 'Cancel' the measurment.	
Retry	

(Abb. 12-27)

Wird der Button zum Start der Messung gedrückt und die Anwendung OLLICONTROL erhält die Quittierung des Befehls nicht innerhalb einer Minute vom Messautomat zurück, wird ein Zeitüberschreitungsfehler ausgegeben.

Dieser Fehler kann mehrere Ursachen haben. In erster Linie ist zu untersuchen, ob die Spannungsversorgung des Messautomaten gewährleistet ist. Anschließend ist zu untersuchen, ob eine korrekte Verbindung zwischen dem PC und dem Messautomaten besteht.

Dieser Fehler ist ebenso quittierpflichtig. Der Vorgang kann entweder über den Button "Cancel" abgebrochen werden oder der Vorgang kann über den Button "Retry" nochmals gestartet werden.

In Abbildung 12-25 ist im zentralen Bereich ein Fadenkreuz sichtbar. Dieses Fadenkreuz dient der Visualisierung der Istposition des Messautomaten. Ist eine Bewegung auf eine Messposition aktiv und wird diese Bewegung mit dem Erreichen der angeforderten Position erfolgreich beendet, so wird die Visualisierung aktualisiert.

Eine erfolgreiche Positionierung wird durch den Benutzer dadurch erkannt, dass die grafische Darstellung des Messautomaten sich auf die entsprechende Position im Fadenkreuz bewegt hat.

In den folgenden Abschnitten wird ein Automatikablauf dargestellt. Die über die serielle Schnittstelle gesendeten Befehle beziehen sich auf die Messungen auf dem Fahrersitz.

≿ Olli Control	a×
Automatic Olli C	ontrol
	AUTONOTIVE
Start auto. measurment	Traffic on RS232
Driver-Seat	
	🔿 Spy on Rx 🛛 💿 Spy on Tx
0 	
Å.	
ا 252 میں ایر \یر	
≈( <u>3</u>	Se la
0	
	Close



In Abbildung 12-28 ist dargestellt, wie der Automatikablauf gestartet wurde und der Befehl A01y, der die Bewegung zur Messposition 1 auf dem Fahrersitz anfordert, bereits ausgeführt wurde.

Die grafische Darstellung des Messautomaten hat sich im Fadenkreuz auf die symbolische Messposition 1 bewegt. Dadurch wird symbolisiert, dass sich der Messautomat nun in der Messposition 1 befindet.

Der Befehl zur Ausführung der nächsten Bewegung wurde bereits über die serielle Schnittstelle gesendet.



(Abb. 12-29)

Das Erreichen der Messposition 2 wurde bestätigt und die symbolische Darstellung wurde aktualisiert.

Der Befehl A03x als Anforderung der Bewegung zur Fahrt auf Messposition 3 wurde über die serielle Schnittstelle gesendet.



(Abb. 12-30)

Die grafische Darstellung des Messautomaten hat sich auf die symbolische Position der 3. Messposition bewegt und zeigt damit das Erreichen dieser Position an.

Der Befehl zum Anfahren der Messposition 4 wurde gesendet.

😓 Olli Control	
Automatic Olli	Control
Start auto. measurment	Traffic on RS232
Driver-Seat	A01y A02y A03x A04y A05x
e	O Spy on Rx O Spy on Tx
	Close

(Abb. 12-31)

OLLICONTROL zeigt an, dass der Automat die Messposition 4 erreicht hat und sendet die Anforderung zur Bewegung auf Messposition 5.

🏃 Olli Control	a ×
Automatic Olli C	ontrol
Automatic on e	4070407107
Start auto. measurment	Traffic on RS232
Driver-Seat	
	🔿 Spy on Rx 💿 Spy on Tx
0 :	
e	·····•
	3
≥ <u>}</u> **	Close

(Abb. 12-32)

Der Messautomat hat das Erreichen der Position 5 bestätigt. OLLICONTROL aktualisiert die Visualisierung und sendet die Anforderung zur Bewegung auf Position 6.

🧞 Olli Control	
Automatic Olli	Control
	Traffic on R\$232
Start auto. measurment	A01y A02y A03x A04y A05x A06x
Driver-Seat 💌	
	🔿 Spy on Rx 💿 Spy on Tx
	• • •
	Close

(Abb. 12-33)

Position 6 entspricht der Grundstellung und wird durch eine Bewegung der grafischen Darstellung des Automaten auf die Startposition symbolisiert.

Im integrierten Fenster "Traffic on RS 232" ist die Befehlsabfolge der gesendeten Kommandos zu erkennen.

## **Betriebsart Teilautomatik**

Wie bereits in den ersten Abschnitten dieses Kapitels erklärt wurde, kann in der Betriebsartenauswahl (Abbildung 12-22) über den Button Teilautomatik die Betriebsart Teilautomatik angefordert werden.

Folgende Oberfläche (Abbildung 12-34) steht dem Bediener zur Steuerung des Messautomaten in der Betriebsart Teilautomatik zur Verfügung.



(Abb. 12-34)

Eine beliebige anzufahrende Position kann durch anklicken der entsprechenden "checkbox" ausgewählrt werden.



(Abb. 12-35)

Nach dem Auswählen einer anzufahrenden Position muss der Start-Button "Start semiautomatic control" gedrückt werden.

Im integrierten Fenster "Traffic on RS 232" kann beobachtet werden, welcher Befehl über die serielle Schnittstelle gesendet wurde.

Während der Fahrt des Messautomaten auf die angeforderte Position werden die Felder im Fenster zur Betriebsart Teilautomatik grau hinterlegt und verhindern somit eine Fehlbedienung durch den Benutzer.



(Abb. 12-36)

Ist die angeforderte Position erreicht, sind die Auswahlfelder im Fenster zur Betriebsart Teilautomatik wieder beschreibbar.

Wie auch in der Betriebsart Automatik symbolisiert die grafische Darstellung des Messautomaten in der Betriebsart Teilautomatik durch seine Positionierung im Fadenkreuz die tatsächlich angefahrene Position des Automaten.



(Abb. 12-37)

Die Aktualisierung der Oberfläche von OLLICONTROL wird durch die Bestätigung des Automaten zum Erreichen der angeforderten Position hervorgerufen.

Ist die angeforderte Position erreicht, kann die nächste anzufahrende Position angefordert werden.

Im Beispiel in Abbildung 12-37 ist nach der Position 1 die Position 2 angefordert worden. Die Bewegung des Messautomaten auf die angeforderte Messposition 2 ist bereits aktiv.

Auch das Erreichen dieser Position wird der Automat bestätigen.

Dieses Verfahren gilt für alle auszuwählenden Messpositionen, die in der Betriebsart Teilautomatik angefordert werden können.

Zu beachten ist, dass zum erneuten Wechsel der Betriebsart, der Messautomat in seine Grundstellung gefahren werden muss.

Hierzu muss die "checkbox Transport" angeklickt und die Bewegung über den Button "Start semiautomatic control" angefordert werden.



(Abb. 12-38)

Wie auch bei den anderen Positionen wird das Erreichen der Grundstellung über die Bewegung der grafischen Darstellung des Messautomaten in OLLICONTROL symbolisiert.

Auch von der Bedienung her, hat die Grundstellung gegenüber den anderen Positionen keine Sonderstellung.

### Ungeplante Ereignisse

Wie zu Anfang des Kapitels bereits erklärt, werden nicht alle Fehlermeldungen, die OLLICONTROL zu generieren in der Lage ist, hier dokumentiert. Folgende Informationen tragen zum Basiswissen über OLLICONTROL bei.

Zwischen OC und dem Messautomat findet ein Datenaustausch statt. Jedoch werden nicht nur Befehle zur Anforderung von Bewegungen gesendet oder das Erreichen einer angeforderten Position bestätigt, sondern es werden auch Daten gesendet, die rein zu Diagnosezwecken dienen.



(Abb. 12-39)

Die HMI ist gegenüber OC dominant, darüber wurde bereits im Kapitel zur HMI informiert. Das hat zur Folge, dass wenn der Messautomat über die HMI bedient wird, OC die Freigabe zur Bedienung des Automaten entzogen wird.

Gleichzeitige Bewegungsanforderungen von unterschiedlichen Orten wird somit verhindert. Sach- und Personenschaden wird durch unterbundene plötzliche Bewegungen des Automaten abgewendet.

🧞 Olli Control	
Automatic Olli (	Control
Start auto. measurment	Traffic on RS232
Driver-Seat	A01y A02y A03x A04y A04y A05x
pinter seat	Spy on Rx Spy on Tx
No permission to control 'Oll	6 I <sup>r</sup> Close

<sup>(</sup>Abb. 12-40)

Der Bediener erhält eine quittierungspflichtige Meldung. Im Hauptfenster der jeweiligen Betriebsart wird eine Information zur Bedienerführung eingeblendet "No permission to control Olli".

i	Information	
	Olli Control received ' B06x ' !	
	Now, you can control 'Olli' again via this programm.	
	'Olli' is in the transport position.	
	If you want to measure the next position, you must press the 'Go on' button	
	If you want to measure the last position , you must press the 'Retry' button	
	Retry Go on	

(Abb. 12-41)

Ist die Software OC wieder zur Bedienung des Messautomaten freigegeben, wird der Benutzer über eine quittierpflichtige Meldung darüber informiert.

Diese Meldung kann von dem dargestellten Informationsfenster in Abbildung 12-41 variieren, da dies zu unterschiedlichen Zeitpunkten geschehen kann und zusätzlich vom Betriebszustand des Messautomaten vor dem Entzug der Freigabe, abhängt.

Im Hauptfenster der entsprechenden Betriebsart ist die Information "No permission to control Olli" wieder ausgeblendet. Die Software OC ist wieder freigegeben, den Messautomaten zu steuern.

Informationen zu weiteren Meldungsfenstern erhalten Sie im Kapitel Kommunikation. Dort sind die Codes, die zum Datenaustausch benutzt werden, hinterlegt.

Die Befehlscodes werden in den Meldungsfenstern mit angezeigt.

## 13 Stromlaufpläne

## 13.1 Allgemein

Die Stromlaufpläne wurden mit EPLAN 5.50.5 erstellt. Sie dienen in diesem Kapitel zur Übersicht über das Gesamtsystem.

Auf der Dokumentations – CDR erhalten Sie die Stromlaufpläne zum Messautomat im Acrobat Reader pdf-Dateiformat und im Grafikdateiformat \*.tif.

Eine Kopie der Stromlaufpläne im EPLAN-Format wurde auf der Dokumentations – CDR nicht abgelegt. Um diese Projektdateien in Originalformat ansehen zu können, müssten Sie EPLAN 5.50.5 installieren. Dies wird jedoch meist aus wirtschaftlichen Gründen abgelehnt.

Um die Dateien dennoch im EPLAN-Format zu erhalten, wenden Sie sich bitte an Ihren **BOSSE**<sup>--</sup> Ansprechpartner.

Die Blätter 5 und 6 des Stromlaufplanes sind Reserveblätter, im Stromlaufplan sind diese Blätter mit Leerblatt gekennzeichnet. Es wird darauf verzichtet, diese Blätter hier darzustellen.

## 13.2 Stromlaufpläne Messautomat



(Abb. 13-1)



(Abb. 13-2)



(Abb. 13-3)



(Abb. 13-4)



(Abb. 13-5)



(Abb. 13-6)



(Abb. 13-7)



(Abb. 13-8)

Die zugehörigen Klemmenpläne sind auf der beiliegenden Dokumentations – CDR enthalten. Die Pläne sind ebenso wie die Stromlaufpläne im Acrobat Reader pdf-Format und im Grafikdateiformat \*.tif enthalten.

Auf eine Darstellung der Klemmenpläne wird in diesem Kapitel bewusst verzichtet.

## 14 Adapterplatine

## 14.1 Allgemein

Die Adapterplatine beinhaltet mehrere Funktionen. Diese Funktionen wurden bereits im Kapitel 6.2 angesprochen.

Die Adapterplatine ist im Gehäuse des Messautomaten installiert. Die Anschlüsse sind über eine Frontplatte nach außen geführt.

Das Kapitel 14 – Adapterplatine wurde der Dokumentation hinzugefügt, um über einige Details der Adapterplatine zu informieren.

## 14.2 Allgemeine Funktionen

- Spannungsversorgung des Messautomaten ist nun in steckbarer Ausführung möglich.
- Die Taster "Boot" und "Reset" der Hauptplatine sind nach außen geführt. Um ein Softwareupdate durchzuführen, muß der Messautomat nicht mehr geöffnet werden.
- Die serielle Schnittstelle S0 ist über die Adapterplatine von außen erreichbar.
- Die Adapterplatine ist mit einem Funkmodul zur Datenübertragung ausgerüstet. Diese Funkverbindung kann alternativ zur konventionellen RS-232 Verbindung genutzt werden.

## 14.3 Frontplatte



(Abb. 14-1)

- Die Frontplatte ist in Aluminiumausführung. Die Steckverbindungen wurden vertauschsicher eingearbeitet.
- Die Beschriftungen sind dauerhaft graviert und gedruckt.
- Die Taster sind gegen versehentliches Betätigen gesichert.

## 14.4 Schaltplan der Adapterplatine



## 14.5 Layout der Adapterplatine



## Anhang

Im Anhang finden Sie Datenblätter zu den wichtigsten Komponenten des Messautomaten. Es wird jedoch davon abgesehen, die Datenblätter aller verbauten Komponenten an dieser Stelle abzudrucken.

Zusätzliche Informationen erhalten Sie auf der beiliegenden Dokumentations – CDR. Auf dieser CD-ROM finden Sie zu allen Komponenten dieses Messautomaten Datenblätter im Adobe Acrobat Reader pdf-Format.

Anhang A :	Auszug Datenblätter - Komponenten der Mechanik
Anhang B :	Auszug Datenblätter - Komponenten der Elektrik
Anhang C :	Auszug Datenblätter - Komponenten der Elektronik
Anhang D :	Stücklisten
Anhang E :	Notizen
Anhang F :	Dokumentations – CDROM

Die Seitennummerierungen in den Dokumenten der Hersteller der Komponenten haben in diesem Anhang keine Bedeutung.

## A 1 – Kugelgewindetrieb

## 

# Präzisions-Kugelgewindetrieb BNK mit bearbeiteten Wellenenden

Der Präzisions-Kugelgewindetrieb mit bearbeiteten Wellenenden Typ BNK ist ein kompakter und einbaufertiger Kugelgewindetrieb mit standardisierten Abmessungen. Für ein einfaches Handling sind die Wellenenden und die Lagereinheiten werkseitig aufeinander abgestimmt.

Die vorgesehene Lagerung ist bei den kleineren Kugelgewindetrieben (BNK0401 und BNK0601) fest-frei und bei den größeren fest-los. Dies und die fertige Endenbearbeitung ermöglichen einen direkten Anschluss des Motors.

					BI	٨K			
Baug	größe	0401	0601	0801	0802	1002	1004	1010	1202
Toleran	izklasse	C3, C5, C7	C5, C7	C3, C5, C7					
Axia	Ispiel	G0 GT G2	G0 GT G2	G0 GT G2					
	20	•							
	40	•	٠	•	•				
	50					•	•		•
	70	•	٠	•	•				
	100		٠	•	٠	٠	•	•	٠
	150			•	٠	٠	٠	٠	•
	200					•		•	٠
	250						٠	٠	•
	300							•	
	350								
Hublange	400								
(mm)	450								
	500								
	550								
	600								
	700								
	800								
	900								
	1000								
	1100								
	1200								
	1400								
	1600								
Festlager Blockausführung	1	EK4	EK5	EK6	EK6	EK8	EK10	EK10	EK 10
Festlager Flanschausführu	ing	FK4	FK5	FK6	FK6	FK8	FK10	FK10	FK10
Loslager Blockausführung	1			EF6	EF6	EF8	EF10	EF10	EF10
Loslager Flanschausführu	ing	-	-	FF6	FF6	FF6	FF10	FF10	FF10
Mutterg	gehause	-	-	-	-	-	MC1004	MC1004	-

## Endenbearbeitung und -lagerung sowie Muttergehäuse für den Präzisions-Kugelgewindetrieb BNK

Axialspiel GO: mit Vorspannung

GT: max. 0,005 mm G2: max. 0,02 mm

Zur Endenlagerung siehe S. 348 ff; zum Muttergehäuse siehe S. 363 ff.

142 元光光

## 

#### Staubschutz und Schmierung

Die Muttern sind bereits mit einer geeigneten Fettfüllung versehen. Als Staubschutz sind in den Muttern ab BNK0802 Labyrinth-Dichtungen integriert. Die Muttern der Baugrößen BNK1510, 1520, 1616, 2020 und 2520 sind mit Endkappen ausgestattet, die aber auch die Funktion einer Labyrinth-Dichtung übernehmen.

Bei bestimmten Betriebsbedingungen, unter denen das Eindringen von Staub und Schmutz in die Mutter wahrscheinlich ist, muss der Kugelgewindetrieb vollständig abgedeckt sein. Dies geschieht üblicherweise mit einem Faltenbalg oder einer Blechabdeckung.



				В	NK				
1205	1402	1404	1408	1510	1520	1616	2010	2020	2520
C3, C5, C7	C3, C5, C7	C3, C5, C7	C5, C7	C5, C7	C5, C7	C5, C7	C5, C7	C5, C7	C5, C7
G0 GT G2	G0 GT G2	G0 GT G2	G0 GT G2	G0 GT G2	G0 GT G2	G0 GT G2	G0 GT G2	G0 GT G2	G0 GT G2
٠	٠		3) 17						
•	٠	٠			2 				
•	٠	٠	•	٠	•	•			
٠	٠	٠	٠	٠	٠	٠			
•			•	•	•	•			
	٠	٠	٠	٠	٠	٠	٠	٠	
			•	•	٠	•			
	-	•	•	•	٠	•	•	•	
		1	•	•	٠	٠			
		-	٠	•	٠	•	•	٠	•
			•	•	•	•			
	2	0- 0-	•	•	٠	•	•	•	•
			٠	•	٠	•	٠	٠	
5		5	1	•	•	•	•	•	•
p		p	p.		p.		•	•	
-							•	•	•
							•	•	-
		-	-		×		-		•
	-			-					•
FK40	EK40	EK40	EK 40	EK40	EK40	EK 40	FKAF	FKAF	FK00
EK10	EK12	EK12	EK12	EK12	EK12	EK12	EK15 EK15	EK15	EK20
FK10 FF10					FK1Z	FK1Z	FK 15	FK 15	F K20
EF10	EF12	EF 12	EF12	EFIZ	EF12	EF12	EF 15	EF ID	EF20
FF IU MC 1207	FFIZ	FFIZ	FE IZ	HEIZ MC1402	FEIZ MC1402	HE12	HE 15	FF15 MC2020	FFZU

143

# 

Präzisions-Kugelgewindetrieb BNK 0801-3

Durchmesser: 8 mm; Steigung: 1 mm



		Ľ	änge Gewindespind	el
Baugröße າຊສ	max. Hub	í.	1	1.2
BNK0801-3G0+115LC3Y			L2	L3
BNK0801-3G0+115LC5Y	40	66	76	115
BNK0801-3G2+115LC7Y				
BNK0801-3G0+145LC3Y			÷	
BNK0801-3G0+145LC5Y	70	96	106	145
BNK0801-3G2+145LC7Y		1.10.10		
BNK0801-3G0+175LC3Y				
BNK0801-3G0+175LC5Y	100	126	136	175
BNK0801-3G2+175LC7Y	]			
BNK0801-3G0+225LC3Y				)
BNK0801-3G0+225LC5Y	150	176	186	225
BNK0801-3G2+225LC7Y				

<sup>1)</sup> Die Baugröße BNK0801 ist auch in korrosionsbeständiger Ausführung erhältlich. Bei Anfragen oder Bestellungen fügen Sie bitte das Kennzeichen M an die Bestellbezeichnung an.

Beispiel: BNK0801-3G0 + 115LC3YM T

—— Kennzeichen für korrosionsbeständige Ausführung

<sup>2)</sup> Bei den Toleranzklassen C3 und C5 ist das Axialspiel GT standardmäßig eingestellt.

<sup>3)</sup> Zur Zusammensetzung der Bestellbezeichnung siehe S. 106.

148 元光代

# 

Daten Kug	elgewindet	rieb			
Steigung [mm]		1			
Kugelmittenkreis [mm]		8,2			
Kerndurchmesser [mm]		7,3			
Steigungsrichtung	rechts				
Anzahl Reihen×Umlauf	3 × 1				
Kennzeichen für Axialspiel	G0	GT	G2		
Axialspiel [mm]	0	0,005 max.	0,02 max.		
dynamische Tragzahl C <sub>a</sub> [kN]		0,64			
statische Tragzahl C <sub>0a</sub> [kN]		1,4			
Vorspannmoment [Nm]	~1,8×10 <sup>-2</sup>	-			
Abstandskugeln	keine				



Ansicht X-X

#### Einheit: mm

С

				Wegabweichung u	und Wegschwankung
Spindel-	Muttern-	Flansch-	Rundlauf	mittlere	
gesamtrundlauf	rundlauf	rechtwinkligkeit	Spindel-Lagersitz	Wegabweichung	Variation
D	Н	1	J		
0,025	0,009	0,008	0,008	±0,008	0,008
0,025	0,012	0,010	0,010	±0,018	0,018
0,035	0,020	0,014	0,014	mittlere Wegabwe	eichung ±0,05/300
0,030	0,009	0,008	0,008	±0,008	0,008
0,035	0,012	0,010	0,010	±0,018	0,018
0,050	0,020	0,014	0,014	mittlere Wegabwe	eichung ±0,05/300
0,030	0,009	0,008	0,008	±0,010	0,008
0,035	0,012	0,010	0,010	±0,020	0,018
0,050	0,020	0,014	0,014	mittlere Wegabwe	eichung ±0,05/300
0,035	0,009	0,008	0,008	±0,010	0,008
0,050	0,012	0,010	0,010	±0,020	0,018
0,065	0,020	0,014	0,014	mittlere Wegabwe	eichung ±0,05/300

149

## A 2 – THK Lager

Festlagereinheit in Flanschausführung Typ FK



Montagemethode A

```
FK10~30
```

Baugröße	Durch- messer Lager- zapfen d <sub>1</sub>	L	Н	F	E	D	A	Lochkreis
FK10	10	27	10	17	29,5	34 -0,009 34 -0,025	52	42
FK 12	12	27	10	17	29,5	36-0009	54	44
FK15	15	32	15	17	36	40-0009	63	50
FK20	20	52	22	30	50	57-0,010	85	70
FK25	25	57	27	30	60	63-000 63-0029	98	80
FK30	30	62	30	32	61	75-000	117	95

354 77748

## 

#### Einheit: mm

в	Montager	nethode A	Montagen	nethode B	Y	v	7	М	T	eingebautes Kugellager
В	-1	1	L <sub>2</sub>	2	~	1	2	101		
42	7,5	5	8,5	6	4,5	8	4	M3	16	7000DFGMP5
44	7,5	5	8,5	6	4,5	8	4	M3	19	7001DFGMP5
52	10	6	12	8	5,5	9,5	6	MЗ	22	7002DFGMP5
68	8	10	12	14	6,6	11	10	M4	30	7204DFGMP5
79	13	10	20	17	9	15	13	M5	35	7205DFGMP5
93	11	12	17	18	11	17,5	15	M6	40	7206DFGMP5

#### Inhalt des Lager-Sets FK10 ~ 30

lfd. Nummer	Teilebezeichnung	Anzahl			
1	1 Blockgehäuse				
2	2 Lagersatz				
3	Gehäusedeckel	1			
4	Distanzring	2			
5	Dichtung	2			
6 Sicherungsmutter		1			
7	Innensechskantschraube (mit Druckstück)	1			

で出民 355

# 

Loslagereinheit in Blockausführung Typ FF





Baugröße	Durch- messer Lager- zapfen d <sub>1</sub>	Ĺ	н	F	D	A
FF6	6	10	6	4	22 -0,007	36
FF 10	8	12	7	5	28 -0,007 -0,020	43
FF 12	10	15	7	8	34 -0,009	52
FF 15	15	17	9	8	40 -0,009 -0,025	63
FF20	20	20	11	9	57 -002	85
FF25	25	24	14	10	63 -0010 -0029	98
FF 30	30	27	18	9	75 -0010	117

360 元光代
# 

### Einheit: mm

Lochkreis	В	x	Ŷ	Z	eingebautes Kugellager	Wellen- sicherungsring
28	28	3,4	6,5	4	606ZZ	C 6
35	35	3,4	6,5	4	608ZZ	C 8
42	42	4,5	8	4	6000ZZ	C10
50	52	5,5	9,5	5,5	6002ZZ	C15
70	68	6,6	11	6,5	6204ZZ	C20
80	79	9	14	8,5	6205ZZ	C25
95	93	11	17,5	11	6206ZZ	C30

#### Inhalt des Lager-Sets FF6 ~ 30

lfd. Nummer	Teilebezeichnung	Anzahl
1	Blockgehäuse	1
2	Lager	1
3	Wellensicherungsring	1

元光光 361

111

# A 3 – Federstegkupplung

# /////// FLEXBEAM<sup>™</sup>-2 KUPPLUNGEN EDESTAHL

EXIBLE KUPPLUNG



b1

А



b2

L

- · Torsionssteifer, höhere Drehmomente übertragbar
- Zwei Sätze von 2 Spiralschnitten für eine hohe Torsionsteife und geringen Aufwickeleffekt
- Kleinere Rückstellkräfte im Vergleich zur Edelstahlserie MF (Seite 9)
- Korrosionsbeständig
- Kein Drehspiel
- Ausgleich radialer, winkliger, axialer und kombinierter Verlagerung
- Klemmausführung dynamisch ausgeglichen

Т	EILENUMME	R			ABME	SSUNG	EN			LEISTUNGSDATEN Mt TORSIONS <u>VERLAGERUNG</u>					
	AUSF. MIT	~ ~		Ø			DIN	DIN	Mt	TORSIONS-	VERL	AGERUN	IG		
KLEMM- AUSF	Stell- Schraube	b1 b2		A (mm)	L (MWC)	L (MWS)	912 <sup>(3)</sup> (MWC)	916 <sup>(3)</sup> (MWS)	stat <sup>(4)</sup> (Nm)	STEIFE <sup>(2)</sup> (Nm/rad)	ANGULAR (Deg)	PARALLEL (mm)	AXIAL (mm)		
MWC15	3,1 MWS15 4,7	3 18 3 4 76 4 5	3 18 4 ,76 5	15	22	20	M2	M3	1,30 1,30 1,30 1,20 1,20	25,7 25,7 25,7 22,7 22,7 22,7	3°	0,20	0,12		
MWC20	MWS20 4,7	4 76 4 5 6	4 ,76 5 6	20	28	20	M3	M3	2,00 2,00 2,00 1,70	58,5 58,5 58,5 44,4	3°	0,20	0,12		
MWC25	6,3 MWS25 9,5 1	6 35 6 8 53 9 10	6 ,35 8 ,53 10	25	30	24	M3	M4	5,10 5,10 5,10 4,60 4,60	98,8 98,8 98,8 69,1 69,1	3°	0,38	0,25		
MWC30	9,5 MW\$30 1 12,7	8 53 9 10 12 70 12	8 ,53 10 12 ,70	30	38	30	M4	M5	10,40 10,40 10,40 10,00 10,00	173,6 173,6 173,6 124,6 124,6	3°	0,38	0,25		

А

Anmerkung 1

- Anmerkung 2
- Anmerkung 3
- Alle Werte sind in mm angegeben, sofern nicht speziell andere Maßeinheiten in den Tabellen angegeben sind. Mt stat gilt bei maximaler Verlagerung Mt dyn = 0,5 Mt stat Mt dyn rev (im Reversierbetrieb) = 0,25 Mt stat. Nähere Angaben zu den verwendeten Schrauben siehe Seite 12. Führen die Wellen in das Innere der Kupplung, muß ein Abstand zwischen diesen vorgesehen
- Anmerkung 4
- muß ein Abstand zwischen diesen vorgesehen werden
- Fhältliche Bohrung **b1** und Bohrung **b2** auswähler Auswahl der Bestellnummer mit Millimeterangaben ergibt sich wie folgt: 15 mm Ø 4 mm Bohrung Klemmausf. MWC15 - 5 -- SS ¥ 5mm Bohrung Edelstah





<i>RULA\II</i> ∕ Flexbeam™-3 Kupf Al	PLUNGEN Luminium	MFC MFS	
<ul> <li>Durch Drehsteifigkeit und Drehspielfreiheit ideal für Servoantriebe etc.</li> <li>Zwei Sätze von 3 Spiralschnitten für hohe Torsionssteife und hohe Drehmomente</li> <li>Größere Torsionssteife und für höhere Drehmomente als Aluminiumserie MW</li> </ul>	MFC	MFS	FLEXIBLE K
<ul> <li>Kein Drehspiel</li> <li>Ausgleich radialer, winkliger, axialer und kombinierter Verlagerung</li> <li>Klemmausführung dynamisch ausge- glichen</li> </ul>		b2 D	UPPLUN

G

T	EILENUMM	IER			A	BMESSU	NGEN			LEIS	TUNGSDA	TEN	
KLEMM- AUSF.	ausf. mit Stell- Schraube	Ø b1	Ø b2	Ø A (mm)	Ø R (MFC)	Ës	DIN 912 <sup>(3)</sup> (MFC)	DIN 916 <sup>(3)</sup> (MFS)	Mt stat <sup>(2)</sup> (Nm)	TORSIONS- STEIFE <sup>2)</sup> (Nm/rad)	ANGULAR (Deg)	RLAGERU PARALLEL (mm)	AXIAL (mm)
MFC20	MFS20	5 6 6,35 8	5 6 6,35 8	20	22,8	30	M3	M4	2,90 2,90 2,70 2,30	43,1 43,1 34,3 28,1	3°	0,20	0,12
MFC25	MFS25	6,35 9,53 10 12 12,70	6 6,35 8 9,53 10 12 12,70	25	30,2	40	M4	M5	4,00 4,00 3,70 3,70 3,70 2,80 2,80	104,2 104,2 66,6 66,6 66,6 36,5 36,5	3°	0,38	0,25
MFC30	MFS30	8 9,53 10 12 12,70 14	8 9,53 10 12 12,70 14	30	34,9	45	M5	M6	7,30 6,30 6,30 5,10 4,70 4,70	168,5 133,3 133,3 83,1 74,4 74,4	3°	0,38	0,25
MFC40	MFS40	10 12 12,70 14 16	10 12 12,70 14 16	40	45,6	55	M6	M6	12,40 12,40 10,70 10,70 10,70	238,7 238,7 146,9 146,9 146,9	3°	0,76	0,38

Anmerkung 1	Alle Werte sind in mm angegeben, sofern nicht speziell andere Maßeinheiten in den Tabellen	Erhältliche Bohrung <b>b1</b> und Bohrung <b>b2</b> auswählen. Auswahl der Bestellnummer mit Millimeterangaben ergibt sich wie folgt
Anmerkung 2	angegeben sind. Mt stat gilt bei maximaler Verlagerung Mt dyn= 0,5 Mt stat Mt dyn rev (im Reversierbetrieh)= 0.25 Mt stat	25mm Ø 6mm Bohrung Klemmaust — MEC25 - 8 - 6 - A
Anmerkung 3	Nähere Angaben zu den verwendeten Schrauben siehe Seite 12.	8mm Bohrung Aluminium
<u>Anmerkung 4</u>	Führen die Wellen in das Innere der Kupplung, muß ein Abstand zwischen diesen vorgesehen werden.	









MFC20	MFS20 6,3	6 6 5 6,35 8 8	20	22,8	30	M3	M4	4,60 4,60 3,60 3,60	83,1 83,1 81,9 81,9	3°	0,20	0,12
MFC25	6,3 6,3 MFS25 9,5 1 1 12,7 12,7	6 6 5 6,35 8 8 3 9,53 0 10 2 12 0 12,70	25	30,2	40	M4	M5	6,10 6,10 5,60 5,60 5,60 3,90 3,90	197,6 197,6 136,4 136,4 136,4 66,6 66,6	3°	0,38	0,25
MFC30	9,5 9,5 MFS30 1 12,70 14	8 8 3 9,53 0 10 2 12 0 12,70 4 14	30	34,9	45	M5	M6	15,50 15,50 15,50 13,50 10,90 10,90	286,5 286,5 286,5 220,4 130,2 130,2	3°	0,38	0,25
MFC40	1 1 MFS40 12,71 1- 1- 1-	0 10 2 12 0 12,70 4 14 6 16	40	45,6	55	M6	M6	23,56 23,56 23,00 23,00 23,00	337,1 337,1 212,2 212,2 212,2 212,2	3°	0,76	0,38

Anmerkung 1



Anmerkung 2

Anmerkung 3

- Anmerkung 4
- angegeben sind. Mt stat gilt bei maximaler Verlagerung Mt dyn= 0,5 Mt stat Mt dyn rev (im Reversierbetrieb)= 0,25 Mt stat. Nähere Angeben zu den verwendeten Schrauben siehe Seite 12. Führen die Wellen in das Innere der Kupplung, muß ein Abstand zwischen diesen vorgesehen werden
- werden.









9

## A 4 – Getriebe

## **FAULHABER**

Planetenge	triebe						ł	0,5 Ni <sup>Kombinierb</sup>	M armit		
								DC-Kleinstm 1727, 2224, Bürstenlose 2036 DC Motor T	notoren: 2230, 2233 n DC-Servor	notoren:	
Serie 20/1								2251		inacionent.	
ehäusewerkstoff ahnräderwerkstoff fax. empfohlene Einge Dauerbetrieb etriebespiel, unbelaste btriebswellenlager faximal zulässige Well radial (8,5 mm vom Be axial faximale Aufpresskraf agerspiel (gemessen ar radial axial etriebstemperaturbere	ingsdrehzahl fü et enbelastung: efestigungsflans t n Lager): eich	r: :ch)						20/1 Stahl Metall 5000 rpm ≤ 1° Kuge llager ≤ 75 N ≤ 20 N ≤ 35 N ≤ 0,02 mm = 0 mm - 30 + 10	, vorgespani 00 °C	nt	
echnische Daten								Drehm	noment		
ntersetZungs- erhältnis iominal) 3,71:1 9,7:1 14 1 23 :1 43 :1 66 :1 86 1 134 :1 159 :1 246 :1 415 :1 592 :1 989 :1 1 526 :1	Gewint ohne Motor 9 28 38 38 38 38 38 48 48 48 48 48 48 58 58 58 68 68 68 68 68 68	Lange ohne Motor L2 mm 18,35 23,45 23,45 23,45 23,45 28,55 28,55 28,55 28,55 33,65 33,65 33,65 33,65 33,65 33,65 33,65 33,75	1727 U L1 mm 45,55 50,65 50,65 55,75 55,75 55,75 60,85 60,85 60,85 65,95 65,95 65,95	Lan 2036 U L1 mm 59,45 59,45 64,55 64,55 64,55 69,65 69,65 69,65 74,75 74,75 74,75 74,75	ge mit M 2224 U L1 mm 42,55 47,65 52,75 52,75 52,75 52,75 57,85 57,85 57,85 62,95 62,95 62,95 62,95	0000 2230 U L1 mm 48,35 53,45 53,45 53,45 53,45 58,55 58,55 58,55 58,55 63,65 63,65 63,65 63,65 63,75 68,75 68,75	2233 U L1 mm 56,05 56,05 61,15 61,15 66,25 66,25 71,35 71,35 71,35	Dater- betrieb M max. mNm 500 500 500 500 500 500 500 500 500 50	Kur22en- betrieb M max. mNm 700 700 700 700 700 700 700 700 700 70	Jrensinn der Welle	Winkungs- grad % 88 80 80 80 80 70 70 70 70 70 60 60 60 60 60 60 60 60 55 55 55 55
Bx 120°	e zu Anschlussfahne estimmt 3x M2,5 3 ter 4	en	<b>020</b> (20 <b>017</b> (17	<sup>236)</sup> ø22	(2233) (2230) (2224)	020 L2	±0,1 ±0,3		0,01 <b>ø6</b> <b>1</b> <b>8</b> <sup>+0</sup> <b>10,7</b> <b>2</b> <b>10</b>	ø4 3,5 -8,05 2 2 5 ±0,3	-0.01 -0.018

20/1

109

Angaben zu Gewährleistung und Lebensdauer sowie weitere technische Erläuterungen siehe "Technische Informationen".

Änderungen vorbehalten. www.faulhaber.com

## **B1 – Antriebsmotor**

### **FAULHABER**

### **DC-Kleinstmotoren**

Edelmetallkommutierung

2224

### 5 mNm

Kombinierbar mit Getriebe: 20/1, 22E, 22/2, 22/5, 22/6, 23/1, 38/3 Impulsgeber: IE2

		000411					100 COLOR OF 11 COLOR		
		2224 0	003 SR	006 SR	012 SR	018 SR	024 SR	036 SR	
1	Nennspannung	UN	3	6	12	18	24	36	Volt
2	Anschlusswiderstand	R	0,56	1,94	8,71	17,50	36,30	91,40	Ω
3	Abgabeleistung	P2 max.	3,92	4,55	4,05	4,54	3,88	3,46	W
4	Wirkungsgrad	η max.	80	82	82	82	81	80	%
				Ter (St.)				10,000	
5	Leerlaufdrehzahl	no	8 100	8 200	7 800	8 100	7 800	7 800	rpm
6	Leerlaufstrom (bei Wellen ø 2,0 mm)	lo	0,066	0,029	0,014	0,010	0,007	0,005	A
7	Anhaltemoment	MH	18.5	21.2	19.8	21.4	19.0	16.9	mNm
8	Reibungsdrehmoment	MR	0,23	0,2	0,2	0,21	0,2	0,22	mNm
9	Drehzahlkonstante	kn	2 730	1 380	657	454	328	219	rpm/V
10	Generator-Spannungskonstante	KE	0.366	0.725	1.520	2.200	3.040	4,560	mV/rpm
11	Drehmomentkonstante	kм	3.49	6.92	14.50	21.00	29.10	43.50	mNm/A
12	Stromkonstante	kı	0.286	0.144	0.069	0.048	0.034	0.023	A/mNm
			- /		-,		-,,	.,	10.000.000000000
13	Steiauna der n-M-Kennlinie	Δη/ΔΜ	438	387	394	379	411	462	rpm/mNm
14	Anschlussinduktivität	L	11	45	200	450	800	1 800	uН
15	Mechanische Anlaufzeitkonstante	Tm	11	11	11	11	11	11	ms
16	Rotorträgheitsmoment	J	2.4	2.7	2.7	2.8	2.6	2.3	acm <sup>2</sup>
17	Winkelbeschleuniauna	CL may	77	78	74	77	74	74	-10 <sup>3</sup> rad/s <sup>2</sup>
				1.2	1.4.4	1.00		1.4.4	
18	Wärmewiderstände	Rth 1 / Rth 2	5/20						KAW
19	Thermische Zeitkonstante	T we1 / T we2	68/440						S
20	Retriebstemperaturbereich:		0,0 / 110						
	- Motor		- 30 +	85(Sondera	usführung –	55 + 125			°C
	- Rotor, max, zulässig		1	25	astantarig	<i></i>			°C
	no conji mali zanabilg								-
21	Wellenlagerung		Sinterlag	er	Kugellag	er	Kugellage	r, vorgespannt	
22	Wellenbelastung, max zulässig.		(standard	D	(Sondera	usführuna)	(Sondera)	usführung)	
1	– für Wellendurchmesser		2.0		2.0		2.0		mm
	- radial bei 3000 rpm (3 mm vom Lager)		1.5		8		8		N
	- axial bei 3000 rpm		0.2		0.8		0.8		N
	- axial im Stillstand		20		10		10		N
23	Wellensniel:		20		10		10		11.
	- radial	-	0.03		0.015		0.015		mm
	- avial	-	0.2		0.2		0		mm
	da lui	-	√, ∠		V,L				
24	Gehäusematerial		Stahl sch	warz heschi	chtet				
25	Gewicht		16	War Lo Goarn	Great				a
26	Drebrichtung		rechtsdre	hend auf Ah	otriebswelle	aesehen			9
20	prominentarig		i acricadi G	nana aan Ak	a coordine	goodinair			
Emt	ofohlene Werte			2.2		2 m	2.12	مول	
27	Drehzahl bis	De max	8 000	8 000	8.000	8 000	8 000	8 000	rom
28	Dauerdrehmoment bis	Me max	5	5	5	5	5	5	mNm
29	Thermisch zulässiger Dauerstrom	le max.	2.200	1.200	0.570	0.400	0.280	0.180	A



### B 2 – Impulsgeber



### Impulsgeber

Magnetische Impulsgeber

Besonderheiten: 16 Impulse pro Umdrehung 2 Ausgänge Digitalausgang

Serie IE2 – 16			
		IE2 – 16	
Impulse pro Umdrehung	N	16	
Ausgangssignal, rechteckig		2	Ausgänge
Betriebsspannung	V <sub>DD</sub>	4 18	V DČ
Nennstromaufnahme, Mittelwert (V <sub>DD</sub> = 12 V DC)	1 <sub>DD</sub>	typ. 6, max. 12	mA
Ausgangsstrom, max. zulässig	l out	15	mA
Pulsbreite <sup>2)</sup>	P	180 ± 45	°e
Signal-Phasenverschiebung, Kanal A zu B <sup>2)</sup>	Φ	90 ± 45	°e
Signal-Anstiegs-/Abfallzeit, max.(C <sub>LOAD</sub> = 100 pF)	tr/tf	2,5 / 0,3	μs
Frequenzbereich <sup>1</sup> , bis	f	7	kHz
Trägheitsmoment der Impulsscheibe	J	0,11	gcm <sup>2</sup>
Betriebstemperaturbereich		- 25 +85	ēC

<sup>1)</sup> Drehzahl (rpm) = f (Hz) x 60/N <sup>2)</sup> bei 2 kHz geprüft

Impulsgeber	Ausgänge	Impulse pro Umdrehung	kombiniert mit DC-Kleinstmotoren
IE2 - 16	2	16	Serie 1516 SR
IE2 – 16	2	16	Serie 1524 SR
IE2 – 16	2	16	Serie 1717 SR
IE2 – 16	2	16	Serie 1724 SR
IE2 – 16	2	16	Serie 2224 SR

Besonderheiten Diese inkrementalen Impulsgeber, in Verbindung mit den FAULHABER DC-Kleinstmotoren, eignen sich für die Überwachung und Regelung von Drehzahl und Drehrichtung sowie für die Positionierung der Antriebswelle.

Der Impulsgeber ist im DC-Kleinstmotor der Serie SR integriert und verlängert diesen um lediglich 1,4 mm!

Durch die Verwendung von Hallsensoren und einem mehrteiligen Magnetring ergeben sich zwei um 90° phasenverschobene Kanäle. Die Versorgungsspannung für den Impulsgeber und den DC-Kleinstmotor sowie die Ausgangssignale werden über ein Flachbandkabel mit Stecker angeschlossen.

Die Daten der DC-Kleinstmotoren und die dazu passenden Getriebe sind aus den entsprechenden Datenblättern zu entnehmen.



## **FAULHABER**



10.04.2004

## **FAULHABER**



### **B** 3 – **HMI**

20.03.2002

# **EA KIT160-7**

EA KIT160-7LWTP

Abmessungen 140x102mm

# **BEDIENEINHEIT MIT FONTS, GRAFIKBEFEHLEN UND MAKROS**



### **TECHNISCHE DATEN**

5,1"

- \* 160x128PIXEL MIT CFL-BELEUCHTUNG BLAU NEGATIV
- \* AUCHMITLANGLEBIGERLED-BELEUCHTUNGWEISS/BLAU
- \* INTEGRIERTESTOUCH PANEL MIT8x7 FELDERN (ENTSPIEGELT, KRATZFEST)
- \* FONT ZOOM VONca. 3mm ÜBERca. 5mm BIS ZUca. 50mm
- \* VERSORGUNGSSPANNUNG 5V/500mA(CFL)/300mA(LW) ODER 9..35V OPTIONAL
- \* RS-232 ODER OPTIONAL RS-422 MIT BAUDRATEN 1200..115200 BD
- \* PIXELGENAUE POSITIONIER UNG BEIALLEN FUNKTIONEN
- \* PROGRAMMIERUNG ÜBERHOCHSPRACHENÄHNLICHEBEFEHLE:
- \* GERADE, PUNKT, BEREICH, UND/ODER/EXOR, BARGRAPH ...
- \* BISZU 256 MAKROS PROGRAMMIERBAR (EEPROM ONBOARD)
- \* TEXTUNDGRAFIK MISCHEN
- \* 4CLIPBOARDFUNKTIONEN, PULL-DOWN MENÜS
- \* 8 DIGITALEEINGÄNGEUND8 DIGTALEAUSGÄNGE
- \* BELEUCHTUNG PERSOFTWARE SCHALTBAR

### **OPTIONEN/ZUBEHÖR**

VERSORGUNG+9.35V=STATT+5V= RS-422 SCHNITTSTELLE STATT RS-232 OPTOKOPPLERONBOARDFÜR8EIN-UND8AUSGÄNGE ALUMINIUM EINBAUBLENDE, MATT-SCHWARZELOXIERT ALUMINIUM EINBAUBLENDE, BLAU ELOXIERT KABEL (1,5m) FÜR ANSCHLUSS AN 9-POL. SUB-D (RS-232 FEMALE) DISKETTEFÜR MAKROPROGRAMMIERUNG (PC-DOSWIN)

EA OPT-9/35V EA OPT-RS4224 EA OPT-OPTO16 EA0FP160-7SW EA 0FP160-7BL EAKV24-9B EA DISK240

### BESTELLBEZEICHNUNG

160x128 DOTS MIT CFL-BELEUCHTUNG, BLAUNEGATIV, TOUCH PANEL EA KIT160-7CTP EA KIT160-7C WIE OBEN, JEDOCH OHNETOUCH PANEL 160x128 DOTS MITWEISSER LED-BEL., BLAU NEGATIV, TOUCH PANEL EA KIT160-7LWTP WIE OBEN, JEDOCH OHNE TOUCH PANEL EA KIT160-7LW



LOCHHAMER SCHLAG 17 D-82166 GRÄFELFING TEL089/8541991 FAX 090/05 1170 to 1

# **ELECTRONIC ASSEMBLY**

### ALLGEMEINES

EA KIT160 ist eine komplett aufgebaute Steuer- und Bedieneinheit mit diversen eingebauten Funktionen. Das kompakt aufgebaute Display bietet zusammen mit dem sehr guten Supertwistkontrast eine sofort einsetzbare Einheit. Die Ansteuerung erfolgt über die Standard Schnittstellen RS-232 oder RS-422. Die Bedieneinheit enthältneben kompletten Grafikroutinen zur Displayausgabe auch verschiendenste Schriften. Die Programmierung erfolgt über hochsprachenähnliche Grafikbefehle; die zeitraubende Programmierung von Zeichensätzen und Grafikroutinen entfällt hier völlig. Die simple Verwendung von Makros und die Eingabemöglichkeit über Touchpanel machen es zu einem richtigen Power Display.

### DISPLAYVARIANTEN

CFL-Beleuchtung EA KIT160-7CTP: blauer Hintergrund mit weiss leuchtender Schrift. Extrem hell und kontraststark. Lebensdauer der Beleuchtung 10.000-20.000 Stunden. Ersatzbeleuchtung unter EA CFL160-7 lieferbar. Stromverbrauch typ. 450 mA.

LED-Beleuchtung EA KIT 160-7LWTP: blauer Hintergrund mit weiss leuchtender Schrift. Guter Kontrast, Lebensdauer 100.000 Stunden. Stromverbrauch typ. 250 mA.

### HARDWARE

Die Bedieneinheit ist für +5V Betriebsspannung ausgelegt. Optional ist eine Versorgung mit 9..35V möglich. Die Datenübertragung erfolgt seriell im RS-232 oder RS-422 Format. Das Übertragungsformat ist fest auf 8 Datenbits, 1 Stopbit, no Parity eingestellt. Die Baudrate kann über DIP-Schalter von 1.200 Baud bis zu 115.200 Baud ausgewählt werden. Handshakeleitungen RTS und CTS stehen zur Verfügung. Datenformat:::

### **TOUCH PANEL**

Die Versionen EA KIT 160-7CTP und -7LWTP sind mit einem integrierten Touch Panel ausgerüstet. Durch Berühren des Displays können hier Eingaben gemacht und Einstellungen per Menü getätigt werden. Die Beschriftung der "Tasten" ist flexibel und auch während der Laufzeit änderbar (verschiedene Sprachen, Icons). Das Zeichnen dereinzelnen "Tasten", sowie das Beschriften oder Zusammenfassen mehrerer Felder wird von der eingebauten Software komplett übernommen.

#### SOFTWARE

Die Programmierung der Bedieneinheit erfolgt über Befehle wie z.B. Zeichne ein Rechteck von (0,0) nach (64,15). Es ist keine zusätzliche Software oder Treiber erforderlich. Zeichenketten lassen sich **pixelgenau** plazieren. Das Mischen von Text und Grafik ist jederzeit möglich. Es können bis zu 16 verschiedene Zeichensätze verwendet werden. Jeder Zeichensatz kann wiederum 2- bis 8-fach gezoomtwerden.

### ZUBEHÖR

#### **Frontpanel zur Montage**

Als Zubehör ist ein Frontpanel aus eloxiertem Aluminium erhältlich. Damit läßt sich die Bedieneinheit ohne sichtbare Schrauben montieren. Das Frontpanel EA 0FP160-7 ist in den Farben schwarz (SW) und blau (BL) lieferbar.

Diskette zur Makroerstellung

Zur Makroprogrammierung ist eine Diskette EA DISK240 erforderlich<sup>\*</sup>). Diese übersetzt die in eine Textdatei eingegebenen Befehle in einen für die Bedieneinheit lesbaren Code und brennt diesen dauerhaft ins EEPROM.

### Kabel für PC

Für die einfache Anbindung an PC's (Makroprogrammierung) liefern wir ein ca. 1,5m langes Kabel mit 9-pol. SUB-D Stecker (female) EA KV24-9B. Einfach an die COM 1 oder COM 2 anstecken und loslegen. Hinweis: Das Kabel ist nicht für die RS-422 Version EA OPT-RS4224 geeignet.

<sup>9</sup> auch im Internet unter http://www.lcd-module.de/deu/disk/disk240.zip

IN 7

Eingang Spalte

.13

# ELECTRONIC ASSEMBLY

### **EXTERNETASTATUR**

Am Steckanschluss J8 kann eine Tastatur (einzelne Tasten bis zur 8x7 Matrix-Tastatur) angeschlossen werden. Die angeschlossenen Tasten werden dabei per Software entprellt. Bitte beachten Sie, daß der Anschluß einer externen Tastatur nur beiden Versionen ohne integriertem Touch Panel möglich ist.

Jede Taste wird zwischen einen Ausgang und einen Eingang geschaltet. Jeder Eingang ist mit einem ca.  $100k\Omega$ Pullup abgeschlossen.

Um Doppeltastendrücke zu erkennen, müssen die Ausgänge voneinander entkoppelt werden. Dies geht am besten mit Schottky-Dioden (z.B. BAT 43).

#### Senden der Tastendrücke

Bei jedem Druck einer Taste wird die dazugehörende Tastennummer (1..56) über die serielle Schnittstelle gesendet oder es wird ein internes Touch Makro, falls

definiert, mit der Tastennummer (1..56) gestartet. Das Loslassen der Taste wird nicht gesendet. Soll auch das Loslassen gesendet werden, so kann das über die Definition des Touch Makros Nr.0 realisiert werden. Der automatische Tastaturscan läßt sich über den Befehl "ESCTA0" deaktivieren. Falls die Handshakleleitung CTS das Senden nicht erlaubt, können Tastendrücke verloren gehen. Die Tastennummer kann folgendermaßen bestimmt werden:

Pin Sym

OUT 7

Ausgang Zeile 7

Applikationsbeispiel 4x4 Matrix

Tastenummer = (Ausgang -1) \* 8 + Eingang

(Ausgang: Zahl zwischen 1 und 7, Eingang: zahl zwischen 1 und 8).

#### TOUCH PANEL(NUR EA KIT160-7xxTP)

Die Versionen EA KIT160-7CTP und -7LWTP werden mit einem integrierten Touch Panel mit 56 Feldern geliefert. Die Bedieneinheit unterstützt dieses Touch Panel mit komfortablen Befehlen. So können z.B. mehrere Touch-Felder zu einer großen Gesamt-Taste zusammengefasst, die Taste gezeichnet und eine Beschriftung der Taste erfolgen. Ebenso kann dieser eben definierten Taste ein Return-Code (1..255) zugewiesen werden. Wird der Return-Code 0 zugewiesen, so ist die Taste deaktiviert und wird bei Betätigung nicht gemeldet.

Beim Berühren der Touch-Tasten können diese automatisch invertiert werden und ein Summer signalisiert die Berührung. Der definierte Return-Code der Taste wird über die serielle Schnittstelle gesendet oder es wird ein internes Touch Makro mit der Nummer des Return-Codes gestartet. Beispiel:

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56

Definieren einer Taste von Feld 11 bis 21, mit dem Return-Code 65=´A´ und dem Text "STOP". Anmerkung: Vor der Definition einzelner Tasten sollten alle Felder durch "ESC T R" deaktiviert sein.





	1.0 00 00 00 00 00 00 00 00 00 00 00 00 0		100 0000	-	10.000		
3) 	Vlatrix - "	Tasta	turan	schluß	<b>J</b> 8		
bol	Funktio	'n	Pin	Symbol	F	unktio	n

7 9	OUT 5 OUT 4	Ausgang Zeile 5 Ausgang Zeile 4	8 10	IN 5 IN 4	Eingang Spalte 5 Eingang Spalte 4
11	OUT 3	Ausgang Zeile 3	12	IN 3	Eingang Spalte 3
13	OUT 2	Ausgang Zeile 2	14	IN 2	Eingang Spalte 2
15	OUT 1	Ausgang Zeile 1	16	IN 1	Eingang Spalte 1

### ELECTRONIC ASSEMBLY Г

Baudraten

### BAUDRATEN

Die Baudrate läßt sich über die linken 3 DIP Schalter einste lieferungszustand sind 9.600 Baud eingestellt (DIP 3 ON). Bitte b daß der interne Datenpuffer lediglich 24 Byte umfaßt. Deshalb sol die Handshakeleitung RTS abgefragt werden (+10V Pegel: D angenommen werden; -10V Pegel: Display ist Busy). Das Datenf eingestellt auf 8 Datenbits, 1 Stopbit, keine Parität. S

### **SCHREIBSCHUTZFÜR MAKROPROGRAMMIERUNG**

Über DIP Schalter 6 läßt sich ein versehentliches Überschreiben der Makros, Bilder und Fontsverhindern.

llen. Im Aus-	DIP	Scha	alter	Datenformat
beachten Sie.	1	2	3	8,N,1
llte unbedingt	ON	ON	ON	1200
aten können	OFF	ON	ON	2400
format ist fest	ON	OFF	ON	4800
chreibschutz	OFF	OFF	ON	9600
Schreibschutz	ON	ON	OFF	19200
tur EEPROM	OFF	ON	OFF	38400
Ein keine Makroprog. mögl.	ON	OFF	OFF	57600
Aus Makroprog. möglich	OFF	OFF	OFF	115200

### RS-232/RS-422 ANSCHLUSS

Standardmäßig wird die Bedieneinheit mit einer RS-232 Schnittstelle ausgeliefert. Die Stiftleiste J3 hat dann die Pinbelegung wie in der Tabelle links abgebildet. J3 ist im Raster 2,54mm ausgeführt. Wird die Bedieneinheit zusammen mit der Option EA OPT-RS4224 bestellt, sind spezielle RS-422 Treiber bestückt. Damit ist die Pinbelegung in der Tabelle rechts gültig.

DIP

6 ON

OFF

An der Lötaugenleiste J5 stehen übrigens die gleichen seriellen Daten mit 5V Pegeln und TTL-Logik zur Verfügung. Diese Pegel sind für den direkten Anschluß an einen µC geeignet. Bei Verwendung dieser Signale müssen allerdings die Lötbrücken LB 5 + LB 6 geöffnet werden!

	RS-2	32 Ar	schluß J3
Pin	Symbol	In/Out	Funktion
1	VDD	1940	+ 5V Versorgung
2	DCD		Brücke nach DTR
3	DSR	1995	Brücke nach DTR
4	TxD	Out	Transmit Data
5	CTS	In	Clear To Send
6	RxD	In	Receive Data
7	RTS	Out	Request To Send
8	DTR	2.00	siehe Pin 2, Pin 3
9	17	1.5	NC
10	GND		0V Masse



......

	RS-422 Anschluß J3											
Pin	Symbol	Funktion										
1	VDD	+ 5V Versorgung										
2	Data In-	Receive Data										
3	Data In+	Receive Data										
4	Data Out-	Transmit Data										
5	Data Out+	Transmit Data										
6	HS In-	Handshake										
7	HSIn+	Handshake										
8	HS Out-	Handshake										
9	HS Out+	Handshake										
10	GND	0V Masse										

8x Eingang

Inur EA OPT-OPTO16

-Ē P D Ē E

P

### VERSORGUNGSSPANNUNG/EAOPT-9/35V

Die Versorgungsspannung von +5V wird über die Schraubklemme J1 eingespeist. Liegt die Version für 9..35V (EA OPT-9/35V) vor, so erfolgt die Stromversorgung über J2. Achtung: Unbedingt auf die richtige Polarität achten! Eine auch noch so kurzzeitige Verpolung kann zur sofortigen Zerstörung des gesamten Displays führen.





8x Ausgang

(nur EA OPT-OPTO

# ELECTRONIC ASSEMBLY

### EIN-UND AUSGÄNGE

Alle EA KIT160-7 werden mit 8 digitalen Ein- und 8 Ausgängen (5V CMOS Pegel, nicht potentialfrei) geliefert. 8 Ausgänge

Jede Leitung kann per Befehl "ESCYW" individuell angesteuertwerden. Pro Leitung kann ein Strom von max. 6mA geschaltet werden. Es ist somit möglich, mit einem Ausgang direkt eine LED (low current) zu schalten. Größere Ströme können mittels externen Transistors verstärkt werden. 8 Eingänge

Die Eingänge können direkt über die serielle Schnittstelle abgefragt und ausgewertet werden ("ESC Y R"). Zusätzlich ist es möglich, bei Änderungen an den Eingängen ein Portmakro automatisch aufzurufen, durch die binäre Kombination von 8 Eingängen sind bis zu 256 Portmakros ansprechbar. Die automatische Portabfrage läßt sich mit dem Befehl "ESC Y A 0" deaktivieren. Anmerkung:

Die Logik ist für langsame Vorgänge ausgelegt; d.h. mehr als 3 Änderungen pro Sekunde können nicht mehr sinnvoll ausgeführt werden. Falls ein Eingang offen ist, so ist dieser High (interner 100 kOhm PullUp).

### EIN-UNDAUSGÄNGE ÜBER OPTOKOPPLER (EA OPT-OPTO16)

Die Ein- und Ausgänge können optional mit Optokopplernausgestattetwerden (EA OPT-OPTO16). Die Ein- und Ausgänge sind dann sowohl von der restlichen Elektronik, als auch untereinander isoliert. Der Anschlußerfolgtüber 16 einzelne Schraublemmen.

An allen 8 Eingängen können direkt Spannungen von 5..35V angelegt werden. Pegel über 4V werden als H-Pegel erkannt, Pegel unter 2V gelten als L-Pegel. Spannungen zwischen 2 und 4V sind undefiniert. Als Ausgang ist jeweils der Kollektor (+) und Emitter (-)

eines NPN-Transistors an den Schraubklemmen herausgeführt. Jeder Ausgang kann 10mA schalten. Hinweis: Der Minuspol jeder Schraubklemme kann durch Schließen der Lötbrücken LBI1..8 bzw. LBO1..8 zusammengeschalten werden. Zusätzlich können diese Lötbrücken auf die Systemmasse

GND gelegt werden (0Ω Brücke RGND einlöten). Anmerkung: Die Optokoppler invertieren die Eingangslogik (alle Eingänge offen: Portmakro 255). Hier empfielt es sich (z.B. im Power-On-Makro) mit dem Befehl "ESC Y I 1" die Eingänge invertiert auszuwerten (d.h. alle Eingänge offen: Portmakro 0).

### GRUNDEINSTELLUNGEN

Nach dem Einschalten bzw. nach einem manuell ausgelösten Resetwerden die nebenstehenden Register auf einen bestimmten Wertvoreingestellt.

Beachten Sie bitte, daß alle Einstellungen durch Erstellen eines Power-On-Makros (Normal-Makro Nr.0) überschrieben werden können.

# EA KIT160-7









Grundeinstellungen										
Register	Befehl	nach Power-On / Reset								
Text-Modus	ESC L	setzen,schwarz								
Terminal Font	ESC FT	Font 3, kein Zoom								
Cursor	ESC QC	ein								
Blinkzeit	ESC QZ	0,6 sek.								
Selbst definierte Zch	ESC E	undefiniert								
Grafik-Modus	ESC V	setzen								
Grafik Font	ESC F	Font 3, kein Zoom								
Last xy	ESC W	(0;0)								
Bargraph 116	ESC B	undefiniert								
Clipboard	ESC C	leer								
Selekt/Deselekt	ESC K	selektiert								
Ausgänge OUT18	ESC Y	L-Pegel								

10 04 2004

# EA KIT160-7

## **ELECTRONIC ASSEMBLY**

### **MAKRO PROGRAMMIERUNG**

Einzelne oder mehrere Befehlsfolgen können als sog. Makros zusammengefasst und im EEPROM fest abgespeichert werden. Diese können dann mit den Befehlen *Makro ausführen* gestartet werden. Es gibt 3 verschiedene Makrotypen:

#### Touch Makro (1..255)

Start bei Berührung eines Touchfeldes (nur bei Versionen mit Touch Panel TP) oder bei Betätigung einer ext. angeschlossenen Taste/Matrixtastatur oder per Befehl 'ESC MT nr'. Das Touch Makro Nr.0 hat eine Sonderstellung: Beim Loslassen einer x-beliebigen Taste wird das Touch Makro Nr.0 gestartet. Port Makro (0..255)

Start bei Anlegen/Ändernung einer Spannung an den Eingängen IN 1..8 oder per Befehl 'ESC MP nr'. Normal Makro (1..255)

Start per Befehl 'ESC MN nr' über die serielle Schnittstelle oder von einem anderen Makro aus. Es können auch mehrere hintereinander liegende Makros automatisch zyklisch aufgerufen werden (Movie, sich drehende Sanduhr, mehrseitiger Hilfetext).

#### Power-On-Makro

Das Normal Makro Nr.0 hat eine Sonderstellung: es wird automatisch nach dem Einschalten ausgeführt. Hier kann man zB. den Cursor abschalten und einen Startbildschirm definieren.

Achtung: Wird im Power-On-Makro eine Endlosschleife programmiert, ist das Display nicht mehr ansprechbar. In diesen Fall hilft nur noch (ab REV. B): DIP Schalter 5 auf ON, Power off, Power on und dann DIP 5 wieder auf off. Jetzt müssen die Fonts und Makros wieder neu eingespielt werden.

### 256 BILDER FEST ABGELEGT

Um Übertragungszeiten der seriellen Schnittstelle zu verkürzen, oder auch um Speicherplatz im Prozessorsystem zu sparen, können bis zu 256 Bilder im internen EEPROM abgelegt werden. Der Aufruf erfolgt über den Befehl "ESC U E" über die serielle Schnittstelle oder aus einem Touch-/Port-/ Normal-Makro heraus. Verwendet werden können alle Bilder im Windows BMP Format. Die Erstellung und Bearbeitung erfolgt über Standardsoftware wie z.B. Windows Paint oder Photoshop.

### **ERSTELLEN INDIVIDUELLER MAKROS**

Um nun Ihre speziellen Makros erstellen zu können, benötigen Sie folgende Hilfsmittel:

- die Diskette EA DISK240°); sie enthält einen Compiler, Beispiele und Fonts
- einen PC mit serieller Schnittstelle COM1 oder COM2, mit ca. 500kB Platz auf der Festplatte
- einen Texteditor wie z.B. WordPad, Norton Editor o.ä.

Um eine Befehlsfolge als Makro zu definieren, werden alle Befehle auf dem PC in eine Datei z.B. DEMO.KMC geschrieben. Hier bestimmen Sie welche Zeichensätze eingebunden werden und in welchen Makros welche Befehlsfolgen stehen sollen. Sind die Makros definiert, startet man das Programm C:>KITCOMP DEMO.KMC. Dieses erzeugt eine EEPROM-Datei DEMO.EEP, welche dann automatisch mit der eingetragenen Baudrate in das Display-EEPROM gebrannt wird. Dieser Vorgang dauert nur wenige Sekunden und sofort danach können die selbstdefinierten Makros genutzt werden. Eine ausführliche Beschreibung zur Programmierung der Makros finden Sie zusammen mit Beispielen auf der Diskette EA DISK240\*) unter dem Namen DOKU.DOC (für WORD) bzw. DOKU.TXT (DOS).

; Makro Demo	
KIT160-7	/ KIT festlegen
COM2; 115200	; KIT ist an COM2 angeschlossen, ; Übertragung mit 115,200 Baud
· · · · · · · · · · · · · · · · · · ·	
;Konstanten definieren	
AUS = 0	
EIN = 1	
FONT4x6 = 1	
FONT5x6 = 2	
FONTEX8 = 3	
FONT8x8 = 4	
FONT8x16= 5	
;	
;Fonts einbinden	
Font: FONT4x6, 32, 95	INTERN4x6
Font: FONT5x6, 32,158	INTERN5x6
Font: FONT6x8, 32,158	INTERN6x8
Font: FONT8x8, 32,158	INTERN8x8
Font: FONT8x16, 32,158	INTERN8x16
;	
Makro: 0	; Power-On/Reset Makro
#QC EIN	; Cursor sichtbar
#FT FONT8x16	; Terminalfont einstellen
#UL 0, 20, <ea2. bmp=""></ea2.>	/ ELECTRONIC ASSEMBLY LOGO
, 10 O O	5 D

\* auch im Internet unter http://www.lcd-module.de/disk/disk240.zip

6

# ELECTRONIC ASSEMBLY

### **INTEGRIERTEFONTS**

In jeder Grafikeinheit sind standardmäßig 5 Zeichensätze integriert. Jeder Zeichensatz kann in 1- bis 8-facher Höhe verwendet werden. Unabhängig davon läßt sich auch die Breite verdoppeln bis verachtfachen.

	+ Lower Upper	50 (0)	<b>\$1</b> (1)	\$2 (2)	\$3 (3)	<b>\$4</b> (4)	\$5 (5)	<b>\$6</b> (6)	<b>%</b>	\$8 (8)	\$9 (9)	\$A (10)	<b>\$B</b> (11)	<b>\$C</b> (12)	<b>\$D</b> (13)	<b>SE</b> (14)	SF (15)
ichensatz	\$20 (dez: 32)		1	-	*	5	8	\$	•	¢	X	×	٠	e	-		7
et werden.	\$30 (dez: 48)	٥	1	5	з	ч	5	6	7	8	9	i	ţ,	×	=	>	?
die Breile	\$40 (dez: 64)	0	Ĥ	B	¢	D	E	F	G	н	I	Ļ	K	L	H	n	ø
Font 1: 4x6	\$50 (dez: 80)	P	٩	R	s	τ	U	Ų	H	х	Y	z	C	5	3	^	-

\$0 \$1 (0) (1) \$9 \$A \$B (9) (10)(11) \$2 \$3 (2) (3) \$4 \$5 \$6 \$7 (4) (5) (6) (7) \$8 (8) Uppe \$20 (dez: 32) п # \$  $\mathbf{z}$ 8. Ĩ ¢ > \* \$30 (dez: 48) Й 1 2 3 4 5 6 7 8 9 = > 2 < \$40 (dez: 64) e A в С D Е F G Н I J К L Μ Ν 0 \$50 (dez: 80) Т ~ P Q R s U U W X Y Ζ 3 Г \$60 (dez: 96) b d f h i J k 1 а c e 9 m n 0 \$70 (dez: 112) p ~ r t u v ł > ۵ 9 s ω x э z { \$80 (dez: 128) Ç ü é ā ¥. a á ç ē ĕ ï î ì Ä è Å \$90 (dez: 144) E Æ Æ 8 ö. ò. ũ ù, ÿ ö ü ¢ £ ¥ ß

Nr.	Zeichen höhe	Zeilen x Zeichen	Größe in Pixel	ASCII- Bereich	Frei def. ASCII- Codes	Bemerkung
1	2,2 mm	21 x 60	4 x 6	32 - 95	121	Microschrift
2	2,2 mm	21 x 48	5x6	32 - 158	121	Minischrift
3	3,1 mm	16 x 40	6 x 8	32 - 158	116	Normalschrift
4	3,1 mm	16 x 30	8 x 8	32 - 158	116	Fettschrift
5	6,3 mm	8 x 30	8 x 16	32 - 158	18	Großschrift

Zusätzlich können, je nach Font, bis zu 21 eigene Zeichen definiert werden die solange erhalten bleiben, bis die Versorgungsspannung abgeschaltet wird. (Siehe Befehl ESC E).

Jedes Zeichen kann **pixelgenau** plaziert werden. Text und Grafik kann beliebig gemischt dargestellt werden. Auch mehrere verschiedene Schriftgrößen lassen sich gemeinsam darstellen.

Font 3: 6x8

Jeder Text läßt sich linksbündig, rechtsbündig und zentriert ausgeben. Auch eine 90° Drehung (vertikaler Einbau des Displays) ist möglich.

Die Makroprogrammierung erlaubt die Einbindung von weiteren 11 Fonts, sowie die komplette Umgestaltung der einzelnen Zeichen. Durch einen Fonteditor auf der Diskette EA DISKFONT6963 können alle nur erdenklichen Schriften mit bis zu 16x16 Pixeln Größe erstellt und einprogrammiert werden.

+ Lower	\$0 (0)	\$1 (1)	\$2 (2)	\$3 (3)	\$4 (4)	\$5 (5)	\$6 (6)	\$7 (7)	\$8 (8)	\$9 (9)	\$A (10)	\$B (11)	\$C (12)	\$D (13)	\$E (14)	\$F (15)
\$20 (dez: 32)		ļ	11	#	\$	X	&	I	(	)	¥	ŧ	,	-		1
\$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	1	;	<	=	>	?
\$40 (dez: 64)	0	A	B	C	D	Ε	F	G	H	I	J	K	L	M	N	0
\$50 (dez: 80)	P	Q	R	S	T	U	Ų	W	X	Y	Z	[	١	]	٨	1
\$60 (dez: 96)	١	a	b	C	d	e	f	g	h	i	j	k	1	M	Π	0
\$70 (dez: 112)	p	q	Г	s	t	u	Ų	W	X	y	z	{	Ι	}	~	Δ
\$80 (dez: 128)	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
\$90 (dez: 144)	É	æ	Æ	ô	ö	ò	û	ù	ÿ	ö	Ü	¢	£	¥	ß	

Font 5: 8x16 \$9

Durch Überlagerung (EXOR) an

Pos.1,1 enstandene "Outline Schrift"

#### **TIP:SCHRIFTEFFEKTE**

Orginalschrift 8x16 mit ZOOM 3

an Position 0,0 mit Muster Schwarz

Mit dem Befehl ESCL TEXT-Modus (Verknüpfung, Muster) können bei grossen Schriften interessante Effekte durch Überlagerung (mehrmaliges versetztes Schreiben eines Wortes) erzielt werden.

nen bei gerung erden. Nochmalige Überlagerung (EXOR) der "Outline Schrift" an Pos.2,2. führt zu einer "Outline Schrift mit Füllung"



Überlagerung (**ODER**) mit Muster 50% Grau der "Outline Schrfit" an **Pos.0,0**. führt zu einer "Schrift mit Musterfüllung"

# **ELECTRONIC ASSEMBLY**

					E	Befe	hlst	abe	lle für EA KIT160-7
Befehl	Cod	es							Anmerkung
						Be	fehle	für	den Terminal Betrieb
Formfeed FF (dez:12)	۸L								Bildschirm wird gelöscht und der Cursor nach Pos. (1,1) gesetzt
Carriage Return CR(13)	۸M								Cursor ganz nach links zum Zeilenanfang
Linefeed LF (dez:10)	۸J				22				Cursor 1 Zeile tiefer, falls Cursor in letzter Zeile dann auf 1. Zeile setzen
Cursor On / Off	ESC	Q	С	n1					n1=0: Cursor ist unsichtbar; n1=1: Cursor blinkt (invers 6/10s);
Cursor positionieren	ESC	0	n1	n2					n1=Spalte; n2=Zeile; Ursprung links oben ist (1,1)
Terminal Font einstellen	ESC	F	Т	n1					n1=1: Font Nr. n1 (116) für Terminal Betrieb einstellen
		_					Bef	'ehl e	zur Textausgabe
Text-Modus	ESC	L	n1	mst					Modus n1: 1=setzen; 2=löschen; 3=invers; 4=Replace; 5=Invers Replace; mod: Muster Nr. 0. 7. verwenden:
Font einstellen	ESC	F	n1	n2	n3				Font mit der Nummer n1 (1.,16) einstellen; n2=X-n3=Y-Zommfaktor (1x.,8x);
Zeichenkette horizontal ausgeben	ESC	Z	L Z R	x1	y1	Text	NUL		Eine Zeichenkette () an x1,y1 ausgeben. 'NUL' (\$00)=Zeichenkettenende; Mehrere Zeilen werde durch das Zeichen 'l' (\$7C, dez. 124) getrennt; 'L'= Linkbündig an x1; 'Z'= Zentriert an x1; 'R'= Rechtsbündig an x1; y1 ist immer die Oberkannte der Zeichenkette
Zeichenkette 90° gedreht (vertikal) ausgeben	ESC	Z	O M U	x1	y1	Text	NUL		Eine Zeichenkette () um 90° gedreht an x1,y1 ausgeben; 'NUL' (\$00)=Ende Mehrere Zeilen werde durch das Zeichen ' ' (\$7C, dez. 124) getrennt; '0'= Oben-Bündig an y1; ''M'= Mittig an y1; 'U'= Unten-Bündig an y1; x1 ist immer die Rechte Kannte der Zeichenkette
Zeichen definieren	ESC	Е	n1		d	aten .	a.		n1=Zeichen Nr.; daten=Anzahl Bytes je nach akt. Font
	-						Be	efehl	e zum Zeichnen
Grafik-Modus	ESC	V	n1						∠erchenmodus einstellen für die Betehle: 'Punkt setzen', 'Gerade zeichnen', 'Rechteck', 'Rundeck' und 'Bereich mit Füllmuster' n1: 1=setzen: 2=löschen: 3=invers: 4=Reolace: 5=Invers Reolace:
Punkt setzen	ESC	Ρ	x1	y1					Ein Pixel an die Koordinaten x1, y1 setzen
Gerade zeichnen	ESC	G	x1	y1	x2	y2			Eine Gerade von x1,y1 nach x2,y2 zeichnen
Gerade weiter	ESC	W	v1	v1					Eine Gerade vom letzten Endnunkt his v1. v1. zeichnen
zeichnen	200		1.00	1.1	20			-	
	-				line 1			Recr	iteck Berenie
Rechteck zeichnen	-		R	X1	<u> </u>	×2	y2	-	Ein Rechteck (Rahmen) von x1,y1 nach x2,y2 zeichnen Ein Rechteck wirk wirden Enlise was ut ut stand v2, v2 zeichnen
Rundeck zeichnen	-			XI	<u> </u>	×2	¥2	-	Ein Rechteck mit runden Ecken von XI (y) nach X2 (y2 zeichnen Eines Dessiskung (d. d. sack (2 v2 lässkag (d.l.s. Divelaus)
Bereich invortieren	-		L.	XI u1	y1 1	×2	y2	-	Einen Bereich von x1,y1 hach x2,y2 loschen (alle Pixel aus)
Bereich füllen	ESC	R		×1	y   01	×2	y∠ √2		Einen Bereich von x1,y1 hach x2,y2 fillen (alle Pixel on)
Bereich m Füllmuster	-		M	×1	y 1	×2	32 V2	met	Einen Bereich von x1 y1 nach x2 y2 mit Muster met (0, 7) zeichnen
Boy zeichnen				×1	y 1	×2	94 V2	met	Ein Berehterk mit Füllmuster met (0, 7) zeichnen: (immer Benlace)
Bux Zeichnen	-		L.	×1	y1	×2	32 V2	mst	Ein Rundeck mit Füllmuster mst (0, 7) zeichnen; (immer Replace)
Rundbox zeichnen		1	0	AT	y ,	AL	Bi	itm ar	Bilder Befehle
Bild aus EEPROM			E	×1	v1	nr			internes Bild mit der nr (D. 255) aus dem EEPROM nach x1 x1 laden
Bild laden	-	U		x1	y1	d	aten	110	Ein Bild nach x1,y1 laden; daten des Bildes siehe Bildaufbau
Hardcony condon	- ESC	ц	- 		10	5			Es wird ein Bild angefordert. Zuerst werden die Breite und Höhe in Pixel und
папасору зепиен		su.	- A1	71	^2 	<sup>32</sup>		A.V	dann die eigentlichen Bilddaten über RS232 gesendet.
Disulary least			<b>.</b>		vispla	y-Bef	enle	(wirk	ung aur das gesamte Display)
Display lusurien									urspraymmant loschen (alle Pixel aus) Diselauishelt isvertieren (elle Divel umbeteret)
Display invertieren Display füllen	-		0						Displayinnart invertieren (alle Pixel umkenren) Displayinhalt füllen (alle Rivel ein)
Display fuller! Display ausschalton	-								proprayinnan ruiteri (alle Fixer etti) Dienlavinhalt wird unsichthar blaibt aber orbelten. Refeble weiterbie mäelich
Display ausschalten	Fer	D.	F						Displayinnar wird disionsar beint aber ernanen, bereine weiternin möglich. Displayinhalt wird wieder sichthar
	- 280	U							Inhalt des Clipboards wird dargestellt. Displavausgaben sind nicht mehr
Display Clipboard									sichtbar
Disp. Normaldarstellung			N						Aktuelles Bild wird dargestellt (Normalbetrieb). Alle Ausgaben wieder sichtbar
Display Reset			R						Der Displaykontroller wird per Befehl rücknesetzt und neu initialisiert
	-		I .,	L				Ma	kro Befehle
Makro ausführen	Т		Ň	n1					Das (Normal-)Makro mit der Nummer n1 aufrufen (max. 7 Ebenen)
Touch Makro ausführen			т	n1					DasTouch-Makro mit der Nummer n1 aufrufen (max. 7 Ebenen)
Port Makro ausführen	ESC	М	P	n1					Das Port-Makro mit der Nummer n1 aufrufen (max. 7 Ebenen)
autom. Makro zyklisch	-		A	n1	n2	n3			Makros n1n2 automatisch zyklisch abarbeiten; n3=Pause in 1/10s
autom. Makro				-1		.2			Molecon autom upp p1 p2 p1 (Ding Denni) - teste iter
pinapona			ا <sup>ر</sup> ا	<u>ut</u>	<sup>11</sup> 2	113			makros autom, von millinzinn (Emgeong) abarbeiten, no-Pause IN 1/10s

# **ELECTRONIC ASSEMBLY**

								Bar	araph I	Befel	hle	
Bargraph definieren	ESC	в	R L O U	nr	x1	y1	x2	у2	aw	ew	mst	Bargraph nach L(inks), R(echts), O(ben), U(nten) mit der ´nr´ (116) definieren. x1,y1,x2,y2 sind das umschließende Rechteck des Bargraphs. aw,ew sind die Werte für 0% und 100%. mst=Muster (07) Den Bargraph mit der Nummer nr (116) auf den neuen
Bargraph zeichnen		4	nr	wer								Benutzer-'wert' setzen
Barris	-	-	1 -	Cli	pboa	rd Be	fehle	e (Zwi	is chen	spei	cher	für Bildbereiche)
Displayinnalt sichern	<u>.</u>		B		1	~	1	1	Der ge	esamt	te Dis	playinnait wird als Bildbereich ins Clipboard kopiert
Dicelou rectourieme	- ESC	С	5	XI	y i	X2	- y2	_	Der Bi	Idber	eich i	m Clinboord wird wieder ine Dienley zwröckkeniert
Bereich konieren	0		K	v1	v1				Der Di	ildhar	eich i	m Clinhoard wird ins Display back v1 v1 koniert
Derefer Köpreren			IN .	AT	y 1	а Т	astat	tur / 1	ouch-	Pane	el Be	fehle
Touch-Taste mit horizontaler Beschriftung definieren			н	f1	67	Ret.	Form	Text	NUL	Die T einer zusar 'Form	ouch- Touc mmen n':Tou	Felder f1 bis f2 (gegenüberliegenden Eckfelder), werden zu h-Taste mit dem Rückgabewert "Ret. Code" (=1.255) gefasst (Ret.Code=0 Touch-Taste nicht aktiv). Lor Taste (=0 nichts =1 löschen:=2 mit Rahmen) zeichnen
Touch-Taste mit vertikaler (90° gedreht) Beschriftung definieren	-		V			Code				Text Touch <u>1' (\$7</u>	':est h-Tas 7C,de	folgt eine Zeichen kette die zentriert mit dem akt. Font in der te plaziert wird, mehrzeilige Texte werden mit dem Zeichen zz. 124) getrennt; Zeichen NUL (\$00) = Zeichenketten en de
Touch-Tasten (P)Reset			P	-					Alle T	ouch-	Taste	n werden aufsteigend aktivient (Felder mit Code F.,60)
5000+710*	en al constantes	10000		200					n1=0:	kein	invert	ieren beim Berühren der Touch-Taste
Touch-Tasten Reaktion	ESC	Т	s	n1 n1					n1=1: n1=0:	Touc kein Sum	ch-Tas Sumr	ste wird beim Berühren automatisch invertiert ner beim Berühren einer (Touch-)Taste inert kurz heim Berühren einer (Touch-)Taste
Touch-Taste Invertieren	1		м	n1					Die To Inverti	uch-" ert	Taste	mit dem zugeordnetem Return-Code n1 wird manuell
Taste manuell abfragen			w						Die m gesen	omen det	ntan g	edrückte (Touch-)Taste wird auf der RS-232/RS-422
Tasten-Abfrage Ein/Aus			A	n1					1 astat n1=1:: n1=2::	urabf aktivi aktivi	rrage ert, T ert, T	wird n 1⊐∪:ceaktiviert; astendrücke werden automatisch gesendet; astendrücke werden nicht gesendet (mit ESC T W abfragen)
							M	lenü .	Popu	рBe	fehle	
Menü mit horizoltalen Einträgen definieren			н	- 11	v1	D.C.	Text	NU	Ein M Vertik nr:= a Toxt:=	enü w ales r ktuell Zoic	vird al menü linver	b der Ecke x1,y1 (Horizotales Menü = linke obere Ecke; = rechte obere Ecke) mit dem akt. Font gezeichnet. tierter Eintrag (z.B: 1 = 1. Eintrag) the mit den Monienträgen Die einzelnen Einträge eind durch
Menü mit vertikalen (90° gedrehten) Einträgen definieren	6°		v	A.S.	31	005	80	NOL	Zeiche Der Hi Ist ber	en ' ' ( interg eits e	(\$7C, grund ein M	dez:124) getrennt z.B. "Eintrag1 Eintrag2 Eintrag3" des Menüs wird automatisch ins Clipboard gesichert. enü definiert, wird dieses automatisch abgebrochen+entfernt.
Menübox invertieren	ESC	N	1			2		<u>, 1</u>	Die ge	samt	e Mer	nübox wird invertiert. Sinnvoll für negative Darstellung
nächster Eintrag		9943.)	N						Der nå	ichste	e Eint	rag wird invertiert oder bleibt am Ende stehen
/orheriger Eintrag			Р						Dervo	orheri	ige Ei	ntrag wird invertiert oder bleibt am Anfang stehen
Menüende / Senden			S						Das M derak	lenü v tuelle	wird v e Einti	om Display entfernt und durch den Clipboardinhalt ersetzt rag wird als Nummer (1n) gesendet (D=kein Menü dargestellt)
Menüende / Makro	-		М	nr					Das M Für Ei	lenü v ntrag	wird v 1 wir	om Display entfernt und durch den Clipboardinhalt ersetzt d Makro 'nr' aufgerufen; für Eintrag 2 Makro nr+1 usw.
Menuenue / Abbrechen	0.2		A	a			Konte	roll_ (	Das N	tions	wira v	om Display entrernt und durch den Clippoardinnalt ersetzt
-	1	<u> </u>	D	×1	×1	2	v2	1	Defini	ert ei	nen F	enie Ninkhereich x1 y1 his x2 y2: Blinkfunktion aktivieren
0			Z	n1	1	1.2	1.04	-	Einste	llen o	der Bl	inkzeit n1= 115 in 1/10s; 0=Blinkfunktion deaktivieren
hatender Bereich	ESC	0	м						Invers	-Mod	us (B	linkbereich wird invertiert); Blinkfunktion aktivieren
Cursor-Funktion)	200		181	mst					Clipbo	ard-N	Modus	s mst=Muster (07) des Blockcursors; Blinken aktivieren
			С	n1					Autom	atisc Blink	n blin funkti	kender Bereich als Cursor für den Terminal Betrieb
		-	S	adr				_	Kit mit	Adre	esse r	11 aktivieren (n1=255: alle)
Selekt / Deselekt	ESC	Κ	D	adr					Kit mit	Adre	esse r	11 deaktivieren (n1=255: alle)
			A	adr					Neue.	Adres	sse ai	dr zuweisen (z.B. im Power-On Makro)
/varten (Pause) Summer Ein / Aus	ESC	X	n1						n1 Zel	nntel- Sum	Seku	nden abwarten bevor der nächste Betehl ausgeführt wird.
Summer Ein / Aus	ESU	U. 	10	-	, s	10000000			Es we	ournin rden	any (	1 255: 0=256) Bytes auf der RS-232/RS-422 resendet
Bytes senden	ESC	S	anz		-	aten	80 C		daten	=	anz B	ytes (z.B. Ansteuerung eines externen seriellen Druckers)
2C-Bus lesen	ESC	1	R	adr	anz				Von di (125: Auf die	em B: 5; 0=) em 12(	auste 256)   C-Bus	in am i∠u-bus mit der Device Adresse adr werden an z Bytes angefordert und über die RS-232/RS-422 gesendet. sfür den Baustein mit der Device Adresse adr werden
2C-Bus schreiben	ESC	<u>,</u> §	W	adr	anz		daten	ue:	anz (1	255	5; 0=2	56) Bytes gesendet. daten = anz Bytes
								P	ort-Be	fehle	•	
Output-Port schreiben			W	n1	n2				n1=0: n1=1 (n2=2)	Alle 8 8: Au	8 Aus usgab	gabe-Ports entsprechend n2 (=8-Bit Binärwert) einstellen e-Port n1 rücksetzen (n2=0); setzen (n2=1); invertieren
Eingabe-Port lesen			R	n1					n1=0:	Alle 8 8: Fil	8 Eing ngabe	gabe-Ports als 8-Bit Binärwert einlesen 3-Port <n1> einlesen (1=H-Pegel=5V, 0=L-Pegel=0V)</n1>
Port Scan Ein/Aus	ESC	Y	A	n1					Der al	utoma	atiche	Scan des Eingabe-Port wird n1=0: deaktiviert: n1=1: aktiviert
Eingabe-Port invers			I	n1					Der Ei	ingab	e-Por	t wird n1=0; normal; n1=1; invertiert ausgewertet
Beleuchtung Ein/Aus			Ē	01					CFL/L	ED-B	eleuc	htung n1=0: AUS; n1=1: EIN; n1=2: INVERTIEREN;
Londontung LIMAUS			Ľ	1 4 10					n1=3	255:	Belei	uchtung für n1 Zehntel Sek., lang einschalten

9

# EA KIT160-7



## B 4 – Netzteil

### EOS Z-Series <u>external power</u> adapters

## Power-Z

Single Output 60W, 65W, and 70W universal AC-DC power supplies

### Big power, little package.

The EOS "Power-Z" AC-DC universal power adapter gives you up to 70 watts of continuous power in a package so small it could only come from EOS. Delivering nearly 8 watts per cubic inch, the "Power-Z" has over twice the density of any other high wattage power adapter on the market. That puts the "Power-Z" in a class by itself. And it means competitive advantage and more value for you and your customers.

Measuring just 5.24" x 2.28" x 1.15", the "Power-Z" is slightly larger than a deck of cards and weighs just 9 ounces—making it far lighter and much smaller than any competitive adapter available. In fact, the Power-Z has set the industry standard for a 70 watt external power supply.

With full worldwide safety standards and a CE mark, the "Power-Z" series is the power adapter of choice for demanding applications like mobile and laptop computers, point of sale terminals, thermal printers, data networking, telecommunications, medical, bar code readers, Internet servers and a host of industrial products. Millions of EOS "Power-Z" AC-DC adapters are in use worldwide, powering the industry's leading electronics products.



#### FEATURES:

- Very High Efficiency: above 90%
- High Power Density: nearly 8 watts/inch<sup>3</sup>
- Ultra Miniature Size: 5.24" x 2.28" x 1.15"
- Lightweight: only 9 ounces
- Power Factor Correction in HD Models
- · Universal input: 90-264 VAC
- FCC & CISPR Level "B" EMI filter
- No Minimum Load Required
- Fully Regulated Output
- Over Current Protection (OCP)
- 100% Burn-In
- High MTBF
- · Meets Worldwide Safety Standards
- · CE marked
- APPLICATIONS:
- Mobile & Laptop Computers
- · Point of Sale terminals and computers
- · Thermal printers
- · Bar Code Readers
- · Data Networking products
- · Telecommunications products
- Internet servers
- Industrial equipment
- · Medical equipment

Contact us at: 1.800.293.9998 or www.eoscorp.com

EOS Z-Se	eries	Po	wer-Z		
SPECIFICA	TIONS				
INPUT					
AC Input	90 to 135 VAC and 170 to 264 VAC	OUTPUT VOLTAGE/CUP	RENT RATING CHART		
Efficiency	86-90% typical, measured at Full load on main and 230 VAC input voltage	Model Output Number Voltage	Current Power		
Input Frequency	47–63 Hz	ZVC60NT12E12.0 VDC	5.00 60W 65W		
		ZVC60NT15E15.0 VDC	4.00 60W 65W		
Output Power	60W, 65W, 70W	ZVC65NT19E19.0 VDC	3.40 65₩ 70₩		
DC Output	12V, 15V, 19V, 24V	ZVC65NT24E24.0 VDC	2.70 65W 70W		
Hold-Up Time	10msec at full rated load, 130VAC/230VAC	ZVC70NS19E19.0 VDC	3.50 65W 80W		
Line Regulation	1%, over operating range	ZVC70NS24E24.0 VDC	2.90 70₩ 80₩		
Load Regulation	3%, with minimum to full load, with standard cord				
Overload Protection	Yes	a 81			
Ripple and Noise	500mV peak-peak maximum				
Power-Up Initialization Period	4 seconds Max at 120VAC				
ENVIRONMENTAL					
Operating Temperature	0 to 40°C				
Storage Temperature	-40°C to +85°C				
Cooling	Convection		FOS Corporation		
		Eng	1070 Flynn Road		
EMI AND SAFETY	AND TALL OF AN ADDRESS IN THE		Camarillo, California 9301:		
CE Mark	Full compliance with LVD and EMC directives	. Highest Density	Tel: 805.484.9998		
Safety Standards	Meets Worldwide Safety Standards; IEC950, EN60950, UL1950 Class 2, SELV	Smallest Size Most Efficient	Toll free: 1.800.293.9998 Fax: 805.484.5854		
AGENCY APPROVALS EMI/RFI	UL, c-UL, VDE, NORDICS, MITI EN55022 Class B, FCC Part 2 &15 Class B, CISPR22 Class B		E-mail: info@eoscorp.com Web: www.eoscorp.com		
MTBF	Over 200,000 hours @ 25°C ambient condition				
MECHANICAL	Sectable below	Consult your local representative requirements or modifications to are subject to change without no	e below or the factory for custom standard products. Specifications tice		
Weight	9 ounces includes DC cord	-			
Ac Input Connector	Dual pin JEC 320 approved C8 type mates with	1			
	interchangeable cords for world wide operations.				
DC Output Connector	Right angle, hollow coaxial plug, 0.10" (2.54 mm) inside diameter, 0.21" (5.49 mm) outside diameter. Center				
DC Cord Length	40 inches				
		l			
			-		
			<u> </u>		
	ſ	у—ос олент			
LQJ I	l	Positive -	7		
	u				
	2402				
®  #	[	CASE LENGTH	E DIMENSIONS (L) WIDTH (W) HEIGHT (H) in/mm in/mm		
LEN I	l	NT 4.505/11/ NS 5.24/133	1.43 2.384/60.55 1.15/29.21 10 2.28/57.10 1.15/29.21		

Contact us at: 1.800.293.9998 or www.eoscorp.com  $^{\rm 06/00~RO2}$ 

## C – Elektronik

Auf den Druck der Datenblätter der Komponenten zur Elektronik des Messautomaten wird komplett verzichtet.

Alle Datenblätter der verbauten Komponenten erhalten Sie auf der beiliegenden Dokumentations – CDR.

Die Datenblätter sind im Acrobat Reader pdf-Format auf der CD-ROM hinterlegt.

# D 1 – Stückliste Basisplatine

### Basisplatine Rev. 1.6

### Platinendatei main\_final 6 letzte Änderung: 05.03.2004

Menge	Wert	Bauteil Beschreibung	Bauteilnr	Lieferant	Bestellnummer	pdf-Datei
1		AMP MINIMATE-N-Lock 2pol	X1	Famell	361-5182	X_AMP_Buchse
1		2pol AAT Man Mane M - Los Les episente das 20 G ad 5 A IC Fest spannungsregier могазалого гот лок	IC1	R&S Comp.	444-9291	IC_MC7800-D
1		7005 5V11A SMD Regulate na ZX 1/401 - 11251 IC Festspannungsregienski ragerivsz prvok	Das er der Gessene III IC 1	R&S Comp.	263-9795	
		7205 5VIIA SMD Regulates	1500	PRS Come	229 2210	LED Gram
		MOLED shiel 0325 in its		Nexo Comp.	230-2218	CED_Ostant
1		LED SMD 3400 LED Snich Date in Date	LELO	R&S Comp.	444-4864	
8		LED SMD 3MD LED anter 0000 in the	LED17	Famell	322-5112	LED_SMD_0603
2		Taster SMD von Phytec	BOOT, RESET	Phytec/Mentor/Conrad	KT007 & KT002	SVV_Mentor
1		Leiterplattenbuchse Micro-Match + pai swo	х9	R&S Comp.	366-1721	X_Micro-Match_SMD_Buchse
6		Leiterplattenbuchse Micro-Match and swo	X4, X5, X6, X7, X10, X11	R&S Comp.	366-1737	X_Micro-Match_SMD_Buchse
1		Beilennistenbuchse von AMP MacAtech in 3MD Leiterplattenbuch se Micro-Match i Bei 3MD	X8	R&S Comp.	366-1787	X_Micro-Match_SMD_Buchse
1		'Opel Leverplanen fundion von JAHT Mico March in SMD SUBD 9pol geschimmt	X2	Reichelt	BMV/Buchse D9W	X_SUB-D_9pol
1		Spol 3x0 D Buchan geocherne Verson SUB D Spol geschirmt	X3	Reichelt	日小小 Stift 0900	X SUB-D 9pol
1	0.250.4	Spol 346 D Buchas geochims Vesion Sichen ungshalter mit Sichen ung SMID	R	R&S Comp	210.6069	Fuce SMD
	0230A	Scherdingsfaller mit Scherding SMD		ness comp.	218-0000	
1	0.500A	Sicherungshalter mit Sicherung SMD Seleungsfaler mit Seleung n 340	F1	R&S Comp.	219-6074	Fuse_SMD
1	5A	Sicherungshalter mit Sicherung SMD Seleungsfaler mi Seleung n 340	F2	R&S Comp.	219-6131	Fuse_SMD
1	0R	Widerstand SMD 0603	R60	R&S Comp.	213-1982	R_SMD_0603
1	0R	Widerstand SMD 0805	R3	R&S Comp.	223-0146	R_SMD_0805
1	1A	Diode SMD MRA_4003_T3_SMB	D1	R&S Comp.	419-1368	D_MRA4003SMD
4	2K2	Dede in 340 1 / 100 / 2014 Anton 4000 in 340 Californi Wilderstand SMD 0603	R21, R22, R31, R32	R&S Comp.	213-2317	R_SMD_0603
4	2u2	Induktivität SMD 3225M 3,3X2,5	L1, L2, L3, L4	Reichelt	LQH3N 2,2u	L_SMD_3225
6	202	0,3 Chm 47an 47,244 Diah Induktiyatitir SMD 0805	10 111 112 113 114 115	B&S Como	308-8586	L W/URTH 0603 0805 1206
	4705	0,50 km 50m k 2,7 uH in 13155 geschem Konnden meters StatD 12018	C17 C19	Des Comp	405 0499	C SMD 1208 TAL
2	4,707	pete		Reso comp.	400-8466	C_300D_1200_170
2	4,7uF	Kondensator SMD	C17, C18	R&S Comp.	405-9933	C_SMD_THU
1	447	Widerstand SMD 0805	R1, R4	R&S Comp.	223-0528	R_SMD_0805
4	5,6K≻max5A	Widerstand SMD 0603 Anona av 10. Sm 11 haterfolm for Schulumfelt	R23, R24, R33, R34	R&S Comp.	568-213-2373 156-213-2430	R_SMD_0603
1	6A	Diode SMD High Current	02	BOSE	<u>.</u>	
4	10k	Widerstand SMD 0603	R25, R26, R35, R36	R&S Comp.	213-2418	R_SMD_0603
2	10nF	Kondensator SMD 0603	C20, C30	R&S Comp	264-4595	C_SMD_0603_0805
2	10uF	Kondensator SMD 1210	C11, C14	R&S Comp	405-2255 adva a das 401-0012 galo	C_SMD_1210_TPS
2	15V	wav⊯en Diode Zener BZG03C D0214 SMD	021.022	R&S Comp	447-2724	DZ SMD
	1517	- Jame Dade 340 SYbs 157 m 00714 Caldum Diade Zener BZC020, D0214 SMD	1771 1772	PRS Come	245 8252	- DZ 9MD
1	1017	Jame Dode 3MD SYds (BV m D07) 4 Cations		Res comp	240-0203	
4	22p+	Kondensator SMD 0805	042, 043, 052, 053	Hess Comp	237-0804	C_SMD_MUHata
6	74 LS 14M	IC 74 LS 14M SMD SO14 hvale 3dmil/igge Enginge	IC5, IC6, IC7, IC11, IC12, IC13	Famell	527-361	IC_DM74LS14
3	74 ALS 32M	IC74 ALS 32M SMD SO14	IC8, IC9, IC10	Famell	632-510	IC_DM74ALS32
1	74 ALS 153M	IC74 ALS 153M SMD S016	IC4	Bürklin	42 \$ 8412	IC_DM74ALS153
2	100 R	Widerstand SMD 0603	R20, R30	R&S Comp.	213-2143	R_SMD_0603
2	100nF	Kondensator SMD 0603	C22, C32	R&S Comp	220-7922	C_SMD_MuRata
18	100nF	Kondensator SMD 0805	د 4 روغ روغ روغ روغ روغ رونې رونې رونغ رونغ رونغ رونغ روغ، روغ، روغ، روغې روغې روغې روغې	R&S Comp	237-7100	C_SMD_MuRata
2	100uF	Kondensator SMD 0810	C23, C33	R&S Comp	367-9615	C SMD FC Panasonic
2	220nF	™/peta Kondensator SMD 0603	(21, 03)	B&S Come	220-7950	C. SMD MuRata
Ē	470.0		~	Dao 0	000 0070	D_014D_0005
	470R	Viderstand SMID D806	rz	Ress Comp.	223-0376	R_SMD_0805
1	10K	Vilderstand SMD 0603	K36, K36, R26, R26	H&S Comp.	213-2418	R_SMD_0603
1	680 R	Widerstand SMD CAY16 530 Ofm 4 Suck R a in enem Section 1708	RN1, RN2	R&S Comp.	241-9664	R_Cat Cay
1	2k3	Widerstand SMD CAY16	RN3, RN4, RN5, RN8, RN9	R&S Comp.	241-9692	R_Cat Cay
-1	10k	Widerstand SMD CAY16	RN6, RN7	R&S Comp.	241-9743	R_CatCay
2	BA\.99	Diode SMD BAV99 Doppeldiode	D20, D30	R&S Comp	287-263	D_BA\49_SMD
2	L6206 PD	sorzazaan. IC Motorsteuer L6206 PD SMD	IC2, IC3	Bürklin	41 S 1076	IC_L6206_ST
1	MM_C509_2	34 Maaninaanaa Lazarro (rover 3038) Buchsenleite für Phyteo C509 Intineon	Sockelste Rei Standart2x12 & 2x32	Phytec	VS019 & VS013	nicht natwendig
1	- C509	Micro Controller C509 auf Platine		Phytec	MM-103-C509	IC_C509_Infineon,IC_MiniModul_C509_Phytec

# D 2 – Stückliste Adapterplatine

### Adapterplatine Rev. 1.5, Rev. 1.6

letzte Änderung : 07.01.2004

Menge	Wert	Bauteil Beschreibung	Bauteilnummer	Lieferant	Bestellnummer	pdf-Datei
1		LED SMD	LEDO	R&S Comp.	238-2219	LED_Osram
1		SMD LED SMD	LED 0603	R&S Comp.	454-4762	nicht verfügbar
1		SMOLED SMD 0005 m verse LED SMD	LED 0603	Farnell	322-5148	LED_SMD_0603_amber
1		3MD LED Andeh 0005 m amber LED SMD	LED 0603	Farnell	322-9531	LED_SMD_0603_rot
1		SMD (20 SMD	LED 0603	Farnell	322-5112	LED_SMD_0603_blau
1		Stiftleiste einreihig	JP1, JP2	R&S Comp.	173-2809	X_Stiftleiste_einreihig
1		Stiftleiste zweireihig	JP4	R&S Comp.	173-2764	X_Stiftleiste_zweireihig
2		Taster 9mm hoch	BOOT, RESET	R&S Comp.	378-6482	SW_2
1		Zpelen færer mildenofen Leiterplattenbuchse Reichett Spol	X2	Reichelt	PS25/3G broderws	X_PS25
1		Spol.Ms Senden Leiternlattenbuchse Micro-Match ums van	X1	Farnell	148-635 148-799 149-445 149-093 378-4496	
		'7pollesuphumbuchasvon JAF MicoMach in 3MD				
1	4k7	Widerstand SMD 0805	R1, R2	R&S Comp.	223-0528	R_SMD_0805
1	470	Widerstand SMD 0805	R3	R&S Comp.	223-0376	R_SMD_0805
6	242	Induktivität SMD 0805	L1, L2	R&S Comp.	308-8586	L_Würth_0603_0805_1206
4	1uF	0,500 m 50m A 2,20H in 0305 generation Kondensator SMD 1206	C11, C12, C13, C14	R&S Comp.	405-9848	C_SMD_1206_TAJ
2	100 nF	Kondensator SMD 0805	C1, C4, C5, C6	R&S Comp	237-7 100	C_SMD_MuR <i>a</i> ta
2	22pF	Kondensator SMD 0805	C2, C3	R&S Comp	237-6804	C_SMD_MuRata
1	680 R	Widerstand SMD CAY16	RN1	R&S Comp.	241-9664	R_Cat_Cay
1	ER400TRS	83800 m + 3uck Rann enem Bausen 1708 SRD Chip mituC	IC1			IC_ER400 TRS
1	MAX232ECWE	RS232 Treiber in SMD	IC2	Farnell	305-0075	IC_MAX232_SMD
1		LWL Leiter flexibel 80mm	Power	Farnell	3245305	LED_Flex_LWL
1		BNC Buchse gerade	X3	Farnell	365-0558	X_BNC_gerade
1		BNC Buchse 90 Grad gewinkelt 50 Ohm	X3	R&S Comp.	3045501	
1		BNC Buchse gerade Frontplatteneinbau	X3	Farnell	365-0510	X_BNC_front
1		BNC Buchse 90 Grad Frontplatteneinbau	X3	Farnell	251-094	X_BNC_90front
1		2 Fach Umschafter gerade	SW 1	R&S Comp.	334246	SW_1
1		Jumper Brücken rot		R&S Comp.	251-8698	
1		Jumper Brücken schwarz lang		R&S Comp.	251-8575	
1		Jumper Brücken schwarz kurz		R&S Comp.	251-8682	

# D 3 – Stückliste Sonstiges

Menge	Bauteil Beschreibung	Bauteilnummer	Lieferant	Bestellnummer	pdf-Datei
1	Schaltnetzteil extern 12V5A		R&S Comp .	377-7141	Netzteil
3	Neigungsschalter		R&S Comp.	361-5043	Neigungsschalter
1	0 main ya Sehah waka 455 adim Kusa afgata sa AMP MINIMATE-N-Lock 2 pol Stecker		R&S Comp.	302-0215	AMP_Stecker
1	2pd AMT Man MaseA4 act. Secteopritius 5A AMP MINIMATE-N-Lock 2pol Stecker		Famell	133-620	AMP_Stecker
1	Zpd AAT New Max-PH- ack Seckegeldaux 5.4 AMP MINI M ATE: N-1 ock Crimpkontakte		B&S Come	302-0467	- AMP Steckerpins
	2pd XMT Mini MaeH4 ad. Complemente Suffree 5.4	27 17 4141	Famol	122 706	AMB Steelerping
	Zpol ANT Min Market and Comptonia Suchas SA	22-10 7003	Famel	270 4661	Hor _ otecher pins
			Famel	370-4030	
	Micro Match Kabel 4 pol	25Umm:43ru; 15Umm:4368	Fameli	378-4370	
4	Micro Match Kabel 6 pol	250mm:4400; 150mm:4393	Famell	378-4400	11.02 22 34 20.033 83
4	MicroMatch Flachbandkabel Stecker 4 pol		Famell	149-032	MicroMatch_Kabelstecker
4	MicroMatch Flachbandkabel Stecker 6 pol		Famell	149-068	MicroMatch_Kabelstecker
4	MicroMatch Flachbandkabel Stecker 16 pol		Famell	149-147	MicroMatch_Kabelstecker
5	Sub-D 9pol Flachban dkabelstecker		R&S Comp.	409-3208	AMP_SubD_Flachbandkabel_Stecker
5	Sub-D 9pol Flachban dkabelbu chse		R&S Comp.	409-3242	AMP_SubD_Flachbandkabel_Buchse
5	Flaichbandkabelstecker zur Festmontage		Famell	525-091	3M_Flachbandkabel
5	Flachbandkabelbuchse zum Schneidklemmen ohne Zugentlastung		Famell	468-010	3M_Flachbandkabel
10	™anipau Verriegelungen für Flachbandkabelstecker		Famell	413-9136	3M_Flachbandkabel
10	nan pan Bolzen für Flachbandkabelstecker Verriegelungen		Famell		3M_Flachbandkabel
4	տարյաս Motor DC mit Getriebe + Drehzahlgeber	DC Karamata 7274.38, Garata 704.757, Inputgeba 67-18	Faulhaber		Motor2224-0 12SR
1	/ autoine OC Menamoni 2224-399 Motor DC mit Getriebe + Drehzahlgeber	DC Kieromoto 2724.38, Genetie 201,143, impubgetie 62-16	Faulhaber		Impulsgeberi E2-16 Getriebe 20_1
12	Pariote CC Nenanon 724-38 Micro Schalter für Achsenmechanik		Reichelt	MAR 1006.0701	MAR1006 Mar
			Destroit Assembly		
1	Touchpanel		Heatronic Assembly	8.4. NIT 1691 LIPTE OF 9-164 (JPT-OFTOIA, KY3-96-	LCDkit160-7
1	Neutrik 220V Enbaubuchse		R&S Comp.	246-8284	Neutrik_Connector
1	Neutrik 220V Stecker		R&S Comp.	246-8313	Neutrik_Connector
1	Zugentlastung Lerno rund 10pol 28	Stecker + Buckse tilling RSZ32 / DAURS Kabel	R&S Comp .	178-8969	Lemo_hstructions
1	Enbaubuchse Lerno rund 10p ol 28	Stecker + Biolise 11 pr RS232 / DANS Kabel	R&S Comp.	178-9029	Lemo_Buchse
1	Stecker Lerno rund 10 pol 2B	Stecker + Bitclise 11 pr RS232 / DANIS Kabel	R&S Comp.	178-8896	Lemo_Stecker
1	9 pol Sub D Einbaubuchse	RS232 Kabel Verbindung Olli⇔PC	R&S Comp.	472-843	Sub.D_9pol
1	vegenaa Gehäuse für 9 pol Sub D Einbaubuchse	R\$232 Kabel Verbindung Olli⇔PC	R&S Comp.	160-0768	Kappe_SubD
1	am tehumpian Multicore Kabel grün	Microphonkabel 4:2:00.14 se ir nāx beimeta weise	Famell	135-240	Cable_green
1	Multicore Kabel blau	Microphonkabel 4x2x0.14 se ir në x bel mete wekse	Famell	798-459	Cable_blue
2	Mini XLR Kabel-Stecker		R&S Comp.	453-662	Switchgraft_Connector
1	Deckelkontaktschalter				
1	Festlager	uoderes Lagerám Motor/Melle) fűr die Soln dei	тнк	FK6	THK Lager
	losinger	kinterer i zanr Wrdh Roladel	THE	CC6	THK lager
		meles taget in de Opnoer	D40.0	041 1500	
2	Prazisionswelle	amm Ditchmesser Prazisonswelle kuumm lang 347 Lukianavao mohineni	Res Comp.	341-1032	SKF_Welle_LJM
2	Lineartager / Wellenlager	Lager das av i der Präzisiosweite lävit	SKF	LBBR 8-2LS/HV6	SKF_LBBR
2	Linearlager / Wellenlager	Lager In dem der Halt lävit	R&S Comp.	734-9100 pdm 155/718-71341/8	SKF_LBBR
1	Spindel mit Spindelmutter 146mm lang	Kage gewindette b	ТНК	BNK0801-3G2+1451C7Y	THK_Kugelgewindtrieb
1	Spindel mit Spindelmutter 175mm lang	Kige gewindette b	ТНК	BNK0801-3G2+1751C7Y	THK_Kugelgewindtrieb
1	Kupplung 4mm><4mm	Verblading zwitchen Getriebe & Gewindetteb	R&S Comp	814-562	Kupplung_Flexbeam
1	Wünfelverbinder für Rahmensystem	zer Verbindeng von 3 Profilen	R&S Comp	390-1928	Rexroth_Verbinder
1	Strebenprofil mit 8erNut Bosch Rexroth	30:30mm 2000mm big	R&S Comp	389-9780	Rexroth_Profile2
1	Abdeolprofil für Strebenprofil		R&S Comp	418-0841	Rexroth_Profile
1	Gehäuse für Main_Board				
1	Flachbandkabelhalter		R&S Comp	609-506	Flachbandkabelhalter
2	Butterfly	Flay 105/5	hter Technik I.T		
1	Griff	KGSM 15D	Inter Technik I.T		
1	Distanzbolzen für Sub-D Buchse lang	M3 <> M3	R&S Comp .	105-8173	nicht notwendig
1	~ Distanzbolzen für Sub-D Buchse kur⊅	M3 <> 4/40	R&S Comp	259-5544	nicht notwendia
1	Read Sensor	M6 28mm lang	R&S Come	280.7067	Reed Sensor
	Manat fir Bood Son or	8mm Dumbmagerer 20mm Inne	RAS Comp.	200-1301	Read Sensor
		omm boronnesser, zomm lang	nao comp.	2020-1200	ress sensor
	2.5mm Enbaubuchse	tur ⊨asy ∠weitgehäuse	H&S Comp.	286-8741	
	Sicherungshalter 5Fach		R&S Comp.	250-6314	
	Madenschraube M6 mit Kugel		R&S Comp.	257-8145	

### E – Notizen

## **F** – Dokumentation auf CDROM

Die beiliegende CDROM in ein CDROM – oder ein DVD-Laufwerk eines kompatiblen PCs einlegen. Die CDROM startet von selbst und muss nicht installiert werden.