1 Kurzbeschreibung

1.1 Zusammenfassung

Der Tastaturkontroller GCK-972-RS232 ist ein in moderner Flash-Mikrocontroller-Technologie realisierter, über „Teach-In frei programmierbarer Tastaturkontroller mit folgenden hervorzuhebenden Eigenschaften:

- Maximal 128 Tasten in einer 8 x 16 Tastenmatrix werden im n-key rollover mode erkannt und ausgegeben. (Sonderversion mit speziellem Layout kann auch 16x16 Tasten bearbeiten)
- Durch eine Taste „Block" umschaltbar, können alle Tasten im sog. „Blockmode" mit zwei voneinander unabhängigen Tastenbedeutungen belegt werden
- Anschluß einer „fertigen" Tastenmatrix erfolgt über ein variabel gestaltestes Steckerfeld.
- Ausgabe aller ASCII Zeichen (Hex-Format)
- Programmierung aller Tasten mit Strings (maximal 30 Zeichen)
- 4 LEDs können über integrierte Treiber angeschlossen werden (vom Host steuerbar)
- Über Jumper abschaltbarer Signalgeber (Tastenclick)
- Die Anpassung der Tastenzuordnung an beliebige Tasten-Matrix ist mit Teach-In Verfahren in 2 Varianten möglich:
- von einem Terminal mit RS232 Schnittstelle (PC mit Terminalprogramm über COM-Schnitt stelle).
- automatisch von einem an der seriellen Schnittstelle angeschlossenen, bereits programmier ten Master-Controller im Download-Mode.
- Wake-Up-Logik zum Aufwecken über jede beliebige Taste kann optional bestückt werden.
- Die Funktionssparameter (Tastatursperre, Widerholfrequenz, usw.)können vom Host verän dert bzw. zurückgesetzt werden.
- Die Hardware für eine integrierte SPI-Schnittstelle kann für den Anschluß von weiteren Kom ponenten (Barcode-Leser Modul , Magnetkartenleser-Modul, LCD) optional bestückt wer den. Hard- und Software dazu wird kundenspezifisch erstellt.

- Die gleiche Hardware wird auch mit einem Programm geliefert, das über die AT/PS/2-Schnittstelle und integrierte PS/2 Weiche eine frei programmierbare PC-kompatible Tastatur realisiert (GCK-972-PS/2)

1.1.1 Betriebsparameter:

- Versorgungsspannung: +5V DC (Pin 10 am seriellen Schnittstellenstecker)
- Stromaufnahme: 12 mA (typ.)
- Stromsparmodus: 2 mA (Standby)
- Lagertemperatur:-15 … +80°C
- Betriebstemperatur: 0 … +70°C

Inhaltsverzeichnis

Inhalt

Deckblatt: High Lights. 1
1.1 Zusammenfassung 2
1.1.1 Betriebsparameter: 2
2 Technische Beschreibung 6
2.1 Funktion und Anwendung der Tastatur 6
2.2 Hardware Konfiguration 7
2.3 Software Konfiguration 7
2.3.1 Programm-Funktionen 7
2.3.2 Anwenderprogrammierung 7
2.4 Technische Spezifikationen des Controllers 8
2.4.1 System: 8
2.4.2 Tastenanschlüsse: 8
2.4.3 Signale nach aussen: 8
2.4.4 Schnittstellen: 8
2.4.6 Umweltparameter: 8
2.4.7 Abmessungen: 8
2.5 Systemübersicht GCK-972 Controllerserie 9
2.6 Blockschaltbild 10
2.7 Steckerbelegungen: 11
2.7.1 J1: SPI Schnittstelle (optional bestückt) 11
2.7.2 J2: Serielle Schnittstelle: 11
2.7.3 J3: AT/PS/2 Schnittstelle zum PC: 11
2.7.5 J5: Externe zusätzliche PC-Tastatur (Weiche): 12
2.7.7 J7: JUMPER für Stromsparmode Disable 12
2.7.8 J9: JUMPER für Piep Ein/Aus 12
2.7.9 J10: Externe zusätzliche PC-Tastatur an der Weiche 12
Tabelle Steckerbelegung J6 13
Abb.: Steckeranordnung und mechanische Abmasse 14
2.8.1 Betätigung einer Taste 15
2.8.1.1 Zeichentasten 15
2.8.1.2 Steuertasten 15
2.8.1.3 BLOCK-Umschalt-Taste 15
2.8.2 Finden der Tastencodierung 15
2.8.2.1 Erkennen der Tastenposition 15
2.8.2.2 Tastenpositionsnummern 15
2.8.2.4 Anzeige der BLOCK-Ebene mit BLOCK-LED 16
2.8.2.6 Anmerkung: Programmierung der Blocktaste als Toggle-Taste 16
2.8.2.7 Ausgabe der Codierung über die serielle Schnittstelle 16
2.8.3 Aufbau der Codetabellen im seriellen EEPROM-Speicher 17
2.8.3.1 Parameterablage und Kennung 17
2.8.3.2 Adressen Parameterablage und Kennung im EEPROM 17
2.8.3.3 Zuordnung der Parameter und Kennung zu den einzelnen Bytes im EEPROM 17
2.8.3.4 Bedeutung der Codes in den Codetabellen 17
2.8.3.5 Was ist eine ASCII-Hex-Codierung ? 18

2.8.3.6 Adressbereiche der Codetabellen im EEPROM 18
2.8.3.7 Stringablage 18
2.8.3.8 Addressbereiche der Strings im EEPROM 18
2.8.4 Full-N-Key-Rollover 19
2.8.5 Typematic (Auto-Repeat) 19
2.8.6.3 Typmatic Mode 20
2.8.6 Piepser (Click) 20
2.8.7 Tastenspeicher 20
2.8.8 Power On Reset 20
2.8.10 Stromsparmodus 21
2.8.11 Tastatursperre 21
2.8.12 Anschluß der Tastatur über Steckverbinder J6 21
2.8.13 Serielle Datenschnittstelle 21
2.8.13.1 Schnittstellenbelegung 21
2.8.13.2 Datenformat 21
2.8.14 Stromzuführung: 22
2.9 Steuerkommandos vom Host 22
2.9.1 Zusammenfassung der Control-Befehle: 22
2.9.3 Steuerungskommandos zur Änderung der Parameter: 23
2.10 Leuchtdioden 24
3 Programmierung der Tastatur 25
3.1 Aufbau Hardware für das Teach-In-Verfahren 25
3.1.1 Allgemeine Information: 25
3.1.2 Anschluss des Terminals 25
3.1.2.1 Kommunikationsparameter 25
3.1.3 Stromversorgung für den Controller 25
3.1.5 Anschluß der Tastaturmatrix 26
3.2 Manuelles Teach-In Verfahren 26
3.2.1 Bedienungsablauf, Kurzfassung: 26
3.2.2 Ausführlicher Bedienungsablauf bei Teach-In: 26
Blockbild: Manuelles Teach-In und Down Load über PC 27
Blockbild Teach-In durch Down Load von Master auf Slave 28
3.2.2.1 Festlegung von SHIFT, CONTROL und BLOCK 29
3.2.2.2 Besonderheit beim manuellen Teach-In: Automatisches Eintragen 29
3.2.2.4 Einzeltaste programmieren: 30
3.2.2.5 Strings programmieren: 30
3.2.6 Beenden des einzelnen Programmiervorganges: 32
3.2.6.1 Abschließen der einzelnen Programmierung 32
3.2.6.2 Abbrechen während des Programmiervorganges: 32
3.3 Teach-In durch Down-Load-Verfahren 32
3.3.1 Editieren und Duplizieren der EEPROM-Programmierung 32
3.3.1.1 Down-Load aus dem PC 32
3.3.1.2 Erzeugung der EEPROM-Tabellen in einer Dump-Datei 32
3.3.1.3 Down-Load aus einer Mastertastatur zum Slave kopieren 33
3.3.1.4 Vorgehensweise 33
Tabelle Bestückung der Controller der Serie GCK-972 34
3.3.2 Beispiel: Listing einer Dump Datei 35
3.4 Kundenspezifische Applikationen 37
3.4.1 Sonderbestückungen 37
3.4.2 Kundenspezifische Layouts 37

3.4.3 Integration des Controllers in eine Tastaturplatine 37
3.4.4 Erweiterungen des Systems, spezielle Eigenschaften 37
3.4.5 Batteriebetrieb 37
3.4.6 Programmierung bereits in der Fertigung 37
3.4.7 Sonderprogrammierungen 38
3.4.8 Beispiel: Codeschloss 38
3.5 Hinweis auf andere Produkte 38
3.5.1 GCK-972-PS/2 38
4 Anhang 39
4.1 Lieferformen des Controllers GCK-972-RS232 39
4.2 Starterkit GCK-972-RS232-START 39
4.2.1 Lieferumfang Starterkit 40
4.2.2 Bei Aufbau des Starterkits bitte beachten 40
4.2.2.1 Stromversorgung 40
4.2.2.2 Experimentiertastatur 40
4.2.2.3 Konsole 40
4.2.2.4 Programmiertastatur (Terminal) 40

2 Technische Beschreibung

2.1 FUNKTION UND ANWENDUNG DER TASTATUR

In einer Matrix von 8 Reihen und 16 Spalten sind bis zu 128 Tasten (Vom System her sind für ein Sonderlayout auch 16x16 = 256 Tasten möglich) anschließbar. Es können praktisch alle gängigen Tasten, die einen Kontaktübergangs-Widerstand kleiner als 200 Ohm haben, eingesetzt werden (z.B. Folien-, Kontakt-, Gummimattentasten).
Die Tastatur wird in den Reihen über positive Impulse gescannt. Ist eine Taste geschlossen, so wird die zugehörige Spalte HIGH, und dies wird am Spalteneingang (C) des Kontrollers erkannt. Um Schattentasten zu vermeiden, kann zu jeder Taste ein Diode (von der Spalte zur Reihe leitend) in Reihe geschaltet werden. Dann ist auch der N-Key-Rollower-Mode störungsfrei möglich.
Jede der Tasten ist vom Anwender selbst in EEPROM-Tabellen in zwei Ebenen (eine zusätzliche sog. „BLOCK-Ebene") frei mit ASCII-Zeichen (Hex-Zahlen) belegbar. Die Zuordnung der zu erzeugenden Zeichen zu den Tasten ist ohne jede Einschränkung frei programmierbar. Alle Tasten mit besonderer Bedeutung, wie beispielsweise SHIFT, NUM-LOCK, RETURN, BLOCK u.s.w. sind ebenfalls frei in der Matrix zu belegen. Die Tastenzuordnung bei beliebigen Tasten-Matrixen ist mit einem Teach-In Verfahren in 2 Varianten möglich.
Zum einen wird über die serielle Schnittstelle ein ASCII-Terminal (Hex-Format) angeschlossen. Über einen Teach-In-Algorithmus wird dann der Controller mit Hilfe dieses Terminals programmiert.
Zum anderen wird ein bereits programmierter Controller ebenfalls über die serielle Schnittstellee an den zu programmierenden Controller angeschlossen. Der zu programmierende Controller holt sich dann in einem Down-Load-Algorithmus aus dem Mastercontroller die spezielle Programmierung und kopiert diese in sein EEPROM.

Über die Serielle-Schnittstelle erfolgt die Ausgabe aller ASCII-Zeichen (Hex-Zahlen).
Jede Taste kann auch mit Tastenstrings belegt werden. Dabei ist die Ablage von 30 Zeichen je Taste möglich.

Ein akustischer Signalgeber als Tastenclick ist bereits eingebaut. Ein zusätzlicher kann auch extern angeschlossen werden. Der interne Tastenclick ist über einen Jumper zu- und abschaltbar.
Die Treiberelektronik mit Vorwiderständen für 4 Leuchtdioden zur Anzeige der Funktionen CAPS-LOCK, NUM-LOCK und BLOCK ist ebenfalls auf dem Controller realisiert.
Das Tastaturprogramm hat „Full-N-Key-Rollover" sowie „Typematic"-Eigenschaften.
Optional vorgesehen ist die Hardware-Bestückung für eine integrierte SPI-Schnittstelle. Hier können mit kundenspezifischer Sonderprogrammierung unterstützt, Peripheriegerätemodule wie Mausfunktionen, Barcode-Lesermodule, Kartenlesermodule, ID-Gerätemodule sowie Displays angeschlossen werden. Über den Mikroprozessor des Tastaturkontrollers und die SPI-Schnittstelle liesse sich auch ein Interface ansteuern, das zusätzliche Hardwarefunktionen zur Verfügung stellt.
Die serielle Schnittstelle kann mit einem IR-Modul auch zu einer Infrarot-Schnittstelle für portablen Betrieb mit Akku ausgebaut werden. Dazu wird ein Stromspar-Modus von der Hardware und der Software (Power-Down und Wake-Up Logik können bestückt werden) optional unterstützt.
Eine Sperrmöglichkeit für die Tastatur über einen von der Tastenmatrix unabhängigen Schlüsselschalter ist ebenfalls vorgesehen.

Fragen Sie uns nach Sonderapplikationen. Es wurden bereits ein mit einer Tastatur ausgestattetes Codeschloß und ein über die RS232 reportierender Protokollmonitor zur einfachen Beobachtung des Datenverkehrs auf einer AT/PS/2-Schnittstelle realisiert.

2.2 HARDWARE KONFIGURATION

Wesentliche Funktionsteile der Kontrollerhardware sind:

* Single-Chip-Mikroprozessor mit 32 kByte Flash-EPROM und 1kByte RAM
* externes 2 kByte EEPROM für die frei programmierbaren Tastenzuordnungen und Stringbelegungen
* 1 Steckverbinder für die Tastaturmatrix bis 8x16 Tasten, je nach Matrixgröße bestückbar, standardmässig 99 PINs, dreireihiger Pfostestecker in 2,54er Raster
* Schlüsselschalter unabhängig von der Tastenmatrix
* 1 Steckverbinder zum Anschluss einer Konsole mit einer Taste (Programmierauswahltaste) und drei LEDs (Signalgabe beim Programmieren).
* akustischer Signalgeber (Piep) intern oder extern anschließbar
* 4 TTL-Ausgänge über Treiber (Für LEDs, Belastbarkeit max. 40 mA gegen GND)
* 1 Steckverbinder für AT bzw. PS/2-PC-Systemschnittstelle, die aus CLOCK- und DATA-Leitung sowie 5V Stromversorgungsleitungen besteht. Diese Schnittstelle wird vom Controller GCK-972-RS232-STD nicht unterstützt.
* 1 Steckverbinder 6polig Mini-DIN (alternativ 5polig Pfostenstecker) für Anschluss einer externen PC-Tastatur über die integrierte Weiche. Diese Option ist im GCK-972-RS232-STD nicht vorhanden.
* 1 Steckverbinder 10polig für die serielle Schnittstelle RS232 (als Option, auch nur als TTL), 9 PINS gehen 1:1 auf SUD-D PC-Pinbelegung der COM Schnittstellen, 10. PIN ist für +5V Stromversorgung herausheführt.
* 1 Steckverbinder 12polig für die Erweiterungsschnittstelle (SPI) als Option
* Integrierte Power-On-Reset-Schaltung
* Integrierter Watchdog
* Power-Down , Wake up über Tastendruck und Schnittstellen (als Option).

2.3 SOFTWARE KONFIGURATION

2.3.1 Programm-Funktionen

Das strukturierte Mikroprozessor-Programm löst sowohl die hardwaremäßige Steuerung des Controllers, kontrolliert die Datenausgabe und verwaltet die Parametereigenschaften.
Das Encoder-Programm fragt zyklisch die Stellung aller Tasten ab.
Das Tastaturprogramm hat „N-Key-Rollover"-Funktion und „Auto-Repeat"-Eigenschaft.
Die programmierung einer Block Taste (Anordnung frei in der Tastenmatrix wählbar) ermöglicht eine zweite Ebene für alle 128 Tasten.

Sonderapplikationen z.B. unter Einbeziehung der AT/PS/2-Schnittstelle mit Weiche oder der SPI-Schnittstellen sind möglich.

2.3.2 Anwenderprogrammierung

Über 2 alternative Teach-In Methoden können beliebige ASCII-Codierungen (Hex-Zahlen) und Strings (bis zu 30 Zeichen) den einzelnen Tasten zugeordnet werden.

2.4 TECHNISCHE SPEZIFIKATIONEN DES CONTROLLERS

2.4.1 System:

* Single Chip Mikroprozessor mit 32 kByte Flash-EEPROM, 1kByte RAM, 2kByte serielles EE-PROM (kann bei preissensitiven Anwendungen und fester Tastenprogrammierung auch herausgelassen werden), Watchdog, Power-On-RESET

2.4.2 Tastenanschlüsse:

* Tastenmatrix: 8x16 für max 128 Tasten
* Erforderlicher Kontaktübergangswiderstand: RK < 200Ω Normale Tasten, Folientasten, mit und ohne Knackfrosch (Schaltdom) und Gummitastaturen lassen sich anschliessen
* Entkopplungsdioden zur Vermeidung von Schattentasten ermöglichen N-Key-Rollower.
* Block Taste (Anordnung frei in der Tastenmatrix wählbar) ermöglicht zweite Ebene für alle 128 Tasten
* Schlüsselschalter separat
* RESET-Taste on Board
* Programmiermodeauswahltaste über Konsolenanschluß

2.4.3 Signale nach aussen:

* Akustischer Geber (Tastenklick, Piep) on Board oder aussen anschließbar (über Tastaturmatrixstecker)
* 4 LEDs (über Tastaturmatrixstecker oder Programmierkonsolenstecker)

2.4.4 Schnittstellen:

* Serielle RS232 Schnittstelle, V.24 (optional: nur mit TTL-Pegel),
  zu PC- COM-Schnittstellen kompatibel.
  Datenformat: 9600 Baud, No Parity, 1 Stopp-Bit
* AT/PS/2 PC-Anschluss und Weiche: Optinal bestückbar.
  Standardmässig durch Software nicht unterstützt:, zusätzlich integrierte Weiche für Anschluss einer externe PC-Tastatur über mit Mini-DIN Stecker.
* SPI- Schnittstelle (GeBE Konfiguration, standardmässig durch Software nicht unterstützt)

2.4.5 Stromversorgung:

+5 V ± 5 %, ca. 12 mA (2 mA im Stromsparmodus)
Über PS/2 Schnittstelle oder seriellen Schnitstellenstecker.

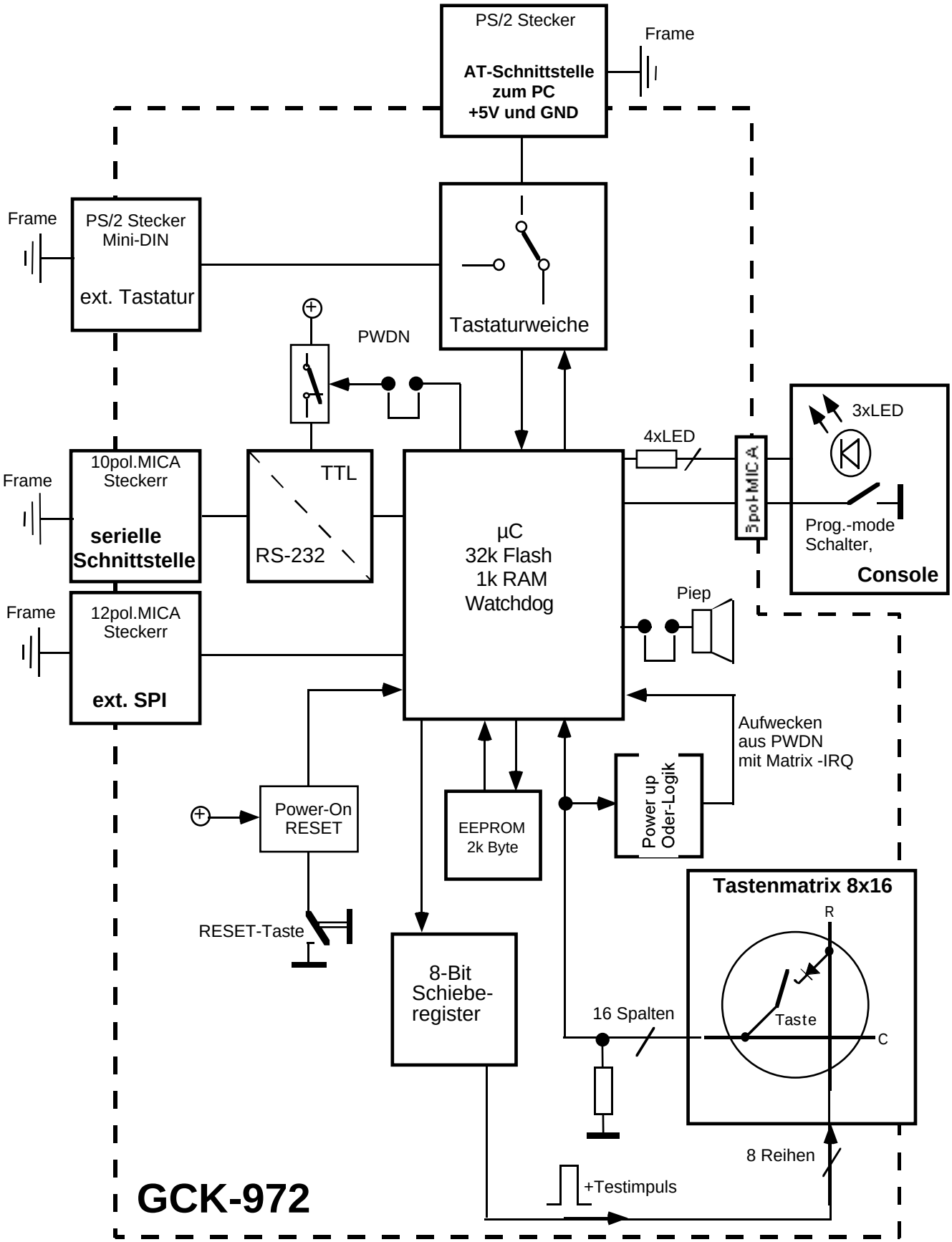
2.4.6 Umweltparameter:

* Lagertemperatur: -15 … +80°C
* Betriebstemperatur: 0 … +70°C

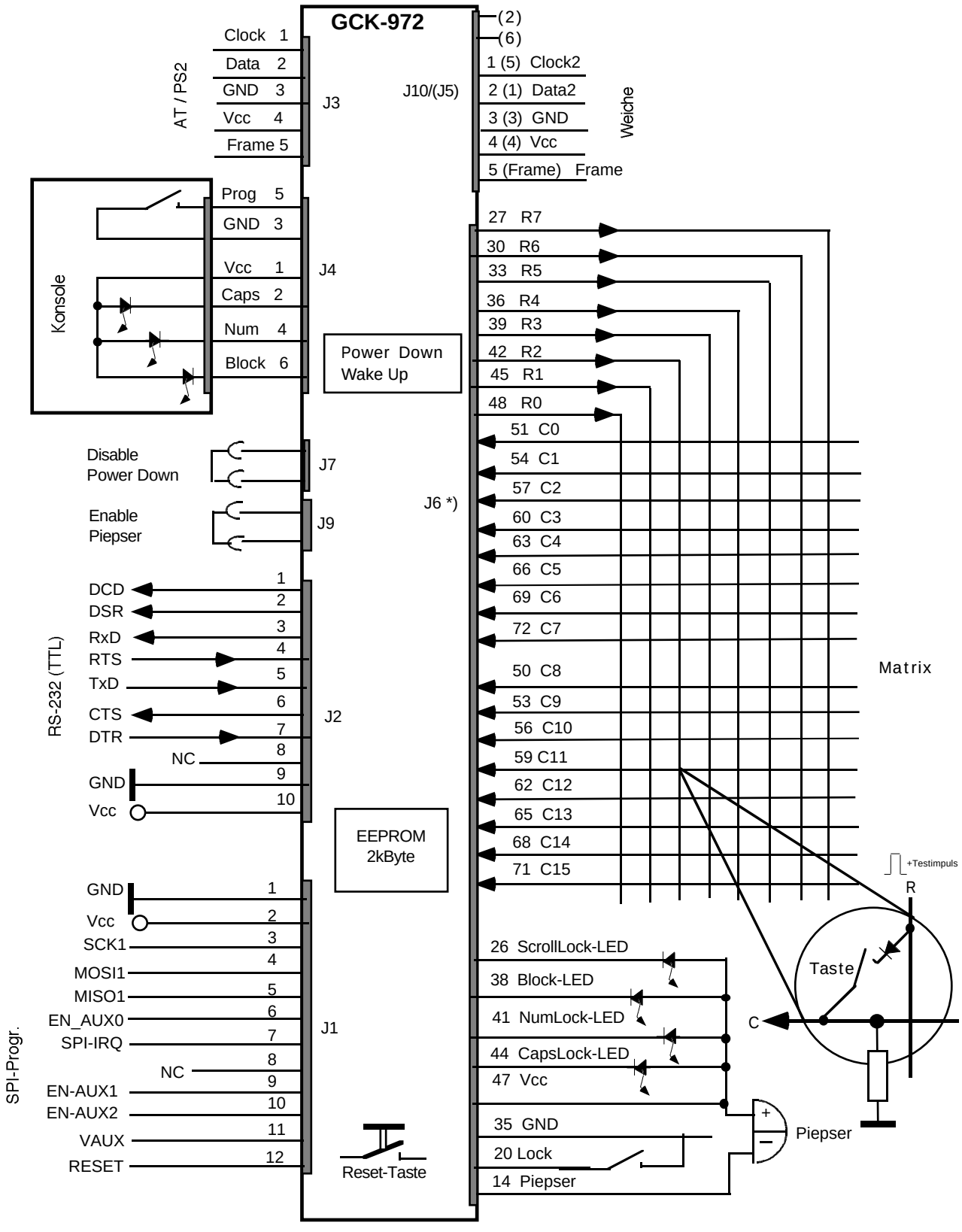
2.4.7 Abmessungen:

* Grösse: (Länge x Breite x Höhe in mm): 98 x 60 x 19
  (bei aufgesetztem Schnittstellenstecker)
* Gewicht: ca. 60 g (voll bestückt).

2.5 Systemübersicht GCK-972 Controllerserie



2.6 Blockschaltbild



*) J6 in Belegung 36-polig

2.7 STECKERBELEGUNGEN:

2.7.1 J1: SPI Schnittstelle (optional bestückt)

12polige MICA Steckverbindung (Stift)

PIN	Bezeichnung	In/Out	Bemerkung
1	GND		
2	Vcc		+5V
3	SCK1		
4	MOSI1		
5	MIOS1		
6	EN-AUX0		
7	SPI-IRQ		
8	POWER_UP		
9	EN-AUX1		
10	EN-AUX2		
11	VAUX		
12	RESET		

2.7.2 J2: Serielle Schnittstelle:

RS232 (V.24) Pegel, ±10V, alternativ mit TTL-Pegeln.
10poliger MICA Steckverbinder (Stift). Es werden nur die Leitungen TxD und RxD benützt. Aussenden der Daten erfolgt ohne Handshake über RxD. Empfangen von Daten (z.B. bei Down Load zur Programnmierung des EEPROMs erfolgt über Datenhandshake mit Xon/Xoff Protokoll.

PIN	Bezeichnung	In/Out	Bemerkung
1	DCD	O	standrdmässig nicht benützt
2	DSR	O	standardmässig nicht benützt
3	RxD	O	Daten seriell senden
4	RTS	I	standardmässig nicht benützt
5	TxD	I	Daten seriell empfangen
6	CTS	O	standardmässig nicht benützt
7	DTR	I	standardmässig nicht benützt
8	NC		
9	GND	I/O	
10	Vcc	I	+5V Stromversorgung

11

2.7.3 J3: AT/PS/2 Schnittstelle zum PC:

Nur optional bestückt. AT/PS/2-compatibel
5poliger Pfostenstecker

PIN	Bezeichnung	In/Out	Bemerkung
1	Clock	I/O	
2	DATA	I/O	
3	GND		GND Stromversorgung
4	Vcc	In	+5V Stromversorgung
5	Frame		

2.7.4 J4: Konsole:

Programmiermodeauswahltaste und 3 LEDs
6poliger MICA /Stift)

PIN	Bezeichnung	In/Out	Bemerkung
1	Vcc	O	+5V
2	CAPS	O	LED1
3	GND		0V
4	NUM	In	LED2
5	PROG_IN	In	Taste
6	BLOCK	O	LED3

2.7.5 J5: Externe zusätzliche PC-Tastatur (Weiche)

Optionale Bestückung,
alternative Bestückung zu J10 6polige MINI-DIN-Buchse

PIN	Bezeichnung	In/Out	Bemerkung
1	DATA	I/O	
2	NC		
3	GND		
4	Vcc	In	+5V
5	CLOCK	I/O	Taste
6	NC	O	LED3

2.7.7 J7: JUMPER für Stromsparmode Disable

2poliger Pfostenstecker Raster 2,54 für Kurzschlussstecker

PIN	Bezeichnung	In/Out	Bemerkung
1	Funktion	I/O	Jumper gesteckt: Stromsparen aktiv
2	GND		

2.7.8 J9: JUMPER für Piep Ein/Aus

2poliger Pfostenstecker Raster 2,54 für Kurzschlussstecker

PIN	Bezeichnung	In/Out	Bemerkung
1	Piepser	I/O	Jumper gesteckt: Piepser eingeschaltet
2	GND		

2.7.9 J10: Externe zusätzliche PC-Tastatur an der Weiche

Optionale Bestückung, alternative Bestückung zu J5, 5poliger Pfostenstecker

PIN	Bezeichnung	In/Out	Bemerkung
1	CLOCK	I/O	
2	DATA	I/O	
3	GND		
4	Vcc	In	+5V
5	Frame		

				Standard-Belegung			PCB, Stecker:			Matrix Steck.			PC-Folie			Tastatur			Matrix::			Matrix 4X6		
	Pin	Signal	Pin	Signal	Pi	Signal	2-reihig 1)			1-zeilig 2)			einseitig 3)			2reihig 4)			RSRS 5)			klein 6)		
Nr.	1		2		3		1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
1	1	R7	2	-DIMmen *)	3	R15 *)						1	1			1	2							
2	4	R6	5	+DIMmen *)	6	R14 *)						2	2			3	4							
3	7	R5	8	Beleuchtung ein/aus *)	9	R13 *)						3	3			5	6							
4	10	R4	11	Option Schalter	12	R12 *)						4	4			7	8							
5	13	R3	14	Piepser	15	R11 *)		1	2			5	5			9	10							
6	16	R2	17	mittlere Maustaste *)	18	R10 *)		3	4			6	6			11	12							
7	19	R1	20	Schlüsselschalter	21	R9 *)		5	6			7	7			13	14							
8	22	R0	23	Sonder Key ein/aus *)	24	R8 *)		7	8			8	8			15	16							
9	25	C0	26	SCROLL-Lock-LED	27	R7		9	10			9				17	18							
10	28	C1	29	linke Maustaste *)	30	R6		11	12			10		9		19	20							
11	31	C2	32	rechte Maustaste *)	33	R5		13	14			11		10		21	22							
12	34	C3	35	GND	36	R4		15	16			12		11		23	24							
13	37	C4	38	BLOCK-LED	39	R3		17	18			13		12		25	26							1
14	40	C5	41	NUM-Lock-LED	42	R2		19	20			14		13		27	28							2
15	43	C6	44	CAPS-Lock-LED	45	R1		21	22			15		14		29	30							3
16	46	C7	47	Vcc	48	R0		23	24			16		15		31	32							4
17	49	R8 *)	50	C8	51	C0		25	26			17				33	34		1	2				5
18	52	R9 *)	53	C9	54	C1		27	28			18			16	35	36		3	4				6
19	55	R10 *)	56	C10	57	C2		29	30			19			17	37	38		5	6				7
20	58	R11 *)	59	C11	60	C3		31	32			20			18	39	40		7	8				8
21	61	R12 *)	62	C12	63	C4		33	34			21			19	41	42		9	10				9
22	64	R13 *)	65	C13	66	C5		35	36			22			20	43	44		11	12				10
23	67	R14 *)	68	C14	69	C6		37	38			23			21	45	46		13	14				
24	70	R15 *)	71	C15	72	C7		39	40			24			22	47	48		15	16				
25	73	C0	74	R0	75	C8						25			23	49	50		17	18				
26	76	C1	77	R1	78	C9						26			24				19	20				
27	79	C2	80	R2	81	C10						27			25				21	22				
28	82	C3	83	R3	84	C11						28			26				23	24				
29	85	C4	86	R4	87	C12						29			27				25	26				
30	88	C5	89	R5	90	C13						30			28				27	28				
31	91	C6	92	R6	93	C14						31			29				29	30				
32	94	C7	95	R7	96	C15						32			30				31	32				
33	97	C0	98	R0	99	C0									31				33	34				

Mögliche Polzahlen bei 2-reihigem Flachbandkabel: 10, 14, 16, 20, 26, 34, 40, 50, 60, 64

1) Beispiel: Tastatur 40poligem, 2reihigen Pfostenstecker, Tasturmatrix (bis 12x16); Sonderfunktionen, Piepser, LEDs, Ro - R11, C0 - C15.

2) Beispiel: Nur Tastenmatrix (bis 16x16), 32poliger, 1reihiger Pfostenstecker, C0-C15, R0-R15

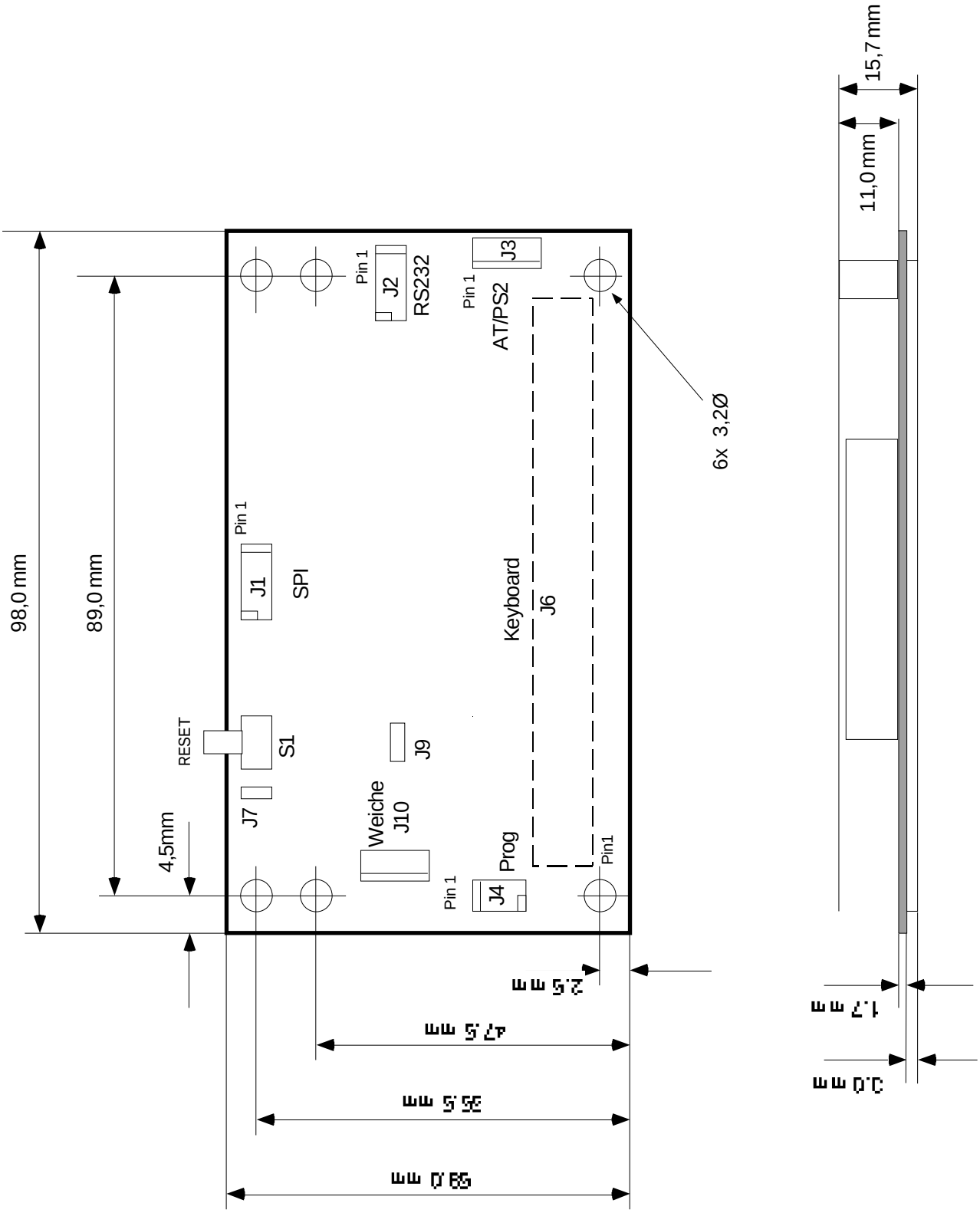
3) Beispiel: Folie mit drei Anschlußfahnen, Tastenmatrix (bis 8x16), R0-R7, Linke/rechte Maustaste; LEDs, Power; C0-C15,

4) Beispiel: Tastatur mit 50poligem, 2reihigem Pfostenstecker; alle Funktionen.

5) Beispiel: Tastatur mit 34poligem, 2reihigem Pfostenstecker, nur Tastaturmatrix, Reihen und Spalten im Wechsel

6) Beispiel: Tastatur mit 10poligem, 1reihigem Pfostenstecker, nur Tastaturmatrix bis 4x6

Abb.: Steckeranordnung und mechanische Abmasse



2.8 BESCHREIBUNG DER KONTROLLER-FUNKTIONEN

2.8.1 Betätigung einer Taste

An den Controller ist über J6 eine uncodierte, mechanische Tastaturmatrix direkt anschließbar. An den Knotenpunkten der Matrix sind Dioden erforderlich, um die Controllereigenschaft Full-N-Key-Rollover zu ermöglichen. In einer 8 x 16 Matrix können standardmäßig maximal 128 Tasten angeordnet werden.
Es gibt folgende Tastenfunktionen :
* Zeichentasten (A, B, &, ?, RETURN .... usw.)
* Steuertasten (SHIFT, CONTROL, ... usw.)
* BLOCK-Umschalt-Taste (Besondere Funktion, erlaubt unabhängige Tastenbelegung in zweiter (Block) Ebene.

2.8.1.1 Zeichentasten

Das der entsprechenden Taste zugeordnete ASCII-Zeichen wird bei Tastendruck über die serielle Schnittstelle ohne auf Handshake zu achten direkt an den Host ausgegeben. Wird die Taste länger gedrückt, so wird der zugehörige Code (Nur Einzelzeichen) nach einer Verzögerungszeit im Rythmus der Wiederholzeit immer wieder Ausgegeben (Autorepeat).

2.8.1.2 Steuertasten

Für die Funktionen UNSHIFT, CTRL, SHIFT und CONTROL-SHIFT sind 4 verschiedene Speicherebenen zur Programmierung der den Tasten jeweils zuzuordnenden Hex-Zeichen vorgesehen. Die Umschaltung in diese weiteren Tastaturebenen (Codetabellen), erfolgt dabei im Gegensatz zu der Organisation einer PC-Tastatur, schon direkt auf dem Controller und nicht erst im Tastaturtreiber der Host-Software.

2.8.1.3 BLOCK-Umschalt-Taste

Die BLOCK-Umschalt-Taste dient zur Umschaltung der beiden Tastatu-rebenen (Code-Tabellen) im Controller. Die BLOCK-Taste verdoppelt die Ausgabemöglichkeiten der einzelnen. So können schliesslich jeder Taste bis zu 8 verschiedene ASCII-Zeichen (Hex-Zahlen) und Strings zugeordnet werden. Damit kann eine hohe Anzahl von Zeichen von der Tastatur generiert werden selbst dann, wenn nur wenige Tasten physisch vorhanden sind.

2.8.2 Finden der Tastencodierung

2.8.2.1 Erkennen der Tastenposition

Durch Scannen der Tasten wird auf Tastenbetätigung abgefragt. Ein Zähler von 0 bis 127 (128 Tasten) wird dabei nach jeder Abfrage um 1 erhöht. Wird eine Taste als gedrückt erkannt, bleibt der Zähler auf dem gerade aktuellen Stand stehen. Der Zählerstand sagt nun aus, welche Taste betätigt wird (bzw. wurde). Diese Tastennummer bildet nun die Referenz für die Taste, die gedrückt wurde.

2.8.2.2 Tastenpositionsnummern

Die Tastenpositionsnummern KN sind den Knotenpunkten der Tastenmatrix wie folgt zugeordnet:
Die einzelnen Adressen sind repräsentativ für die Tastenmatrixknotenpunkte. Sie sind nach folgender Tabelle sinngemäss in allen 8 Codetabellen gleich angeordnet.

	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15
R0	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
R1	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
R2	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
R3	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
R4	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
R5	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
R6	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F
R7	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F

2.8.2.3 Verzweigen mit SHIFT, CONTROL und BLOCK

In der Tastatuebene können an beliebiger Stelle die Steuetasten SHIFT, CONTROL und BLOCK angelegt werden. Die dafür erforderlichen Codes werden beim Teach-In-Verfahren automatisch in allen Codetabellen an der selben Stelle eingetragen. Während SHIFT und CONTROL nur solange wirken, wie die zugehörige Taste gedrückt bleibt, kann die BLOCK-Taste alternativ mit zwei verschiedenen Funktionsweisen belegt werden: Einmal wirkt die Taste nur beim Drücken (On Push:= Code XX(Hex)) in der zweiten Version arbeitet die Taste als Toggle-Taste, jedes Drücken schaltet den Mode um.

2.8.2.4 Anzeige der BLOCK-Ebene mit BLOCK-LED

Zur Unterstützung wird mit der BLOCK-LED der Zustand der Blocktaste angezeigt:
Blockebene ist ausgewählt:= BLOCK-LED leuchtet.

2.8.2.5 Codierung für Steuertasten SHIFT, CONTROL und BLOCK

Die Codierung der Steuertasten wird nicht nach aussen gesendet, sie bewirkt lediglich intern die Umschaltung in die einzelnen Codetabellen.

CAPS (toggle function):	FB(Hex)
CONTROL (on push):	FC(Hex)
SHIFT (on push):	FD(Hex)
BLOCK (toggle function):	FE(Hex)
BLOCK (on push):	FF(Hex)

2.8.2.6 Anmerkung: Programmierung der Blocktaste als Toggle-Taste

Die BLOCK-Tastenfunktion kann über eine Toggelfunktion wirken, das heißt, es erfolgt bei -Tastendruck jeweils eine Umschaltung in oder aus der BLOCK-Ebene und die Taste BLOCK braucht nicht ständig gedrückt gehalten zu werden, um ein Zeichen aus der BLOCK-Ebene zu generieren. Dann ist die Anzeige des Zustandes über die LED Block sinnvoll.
Bitte beachten Sie beim Editieren der Codetabellen auf dem PC, daß die Codierung der BLOCK-Umschalttaste in allen Grund- und BLOCK-Ebenen in den gleichen Tastenpositionen eingetragen sein muß, um das Hin- und Rückschaltung in/aus der BLOCK-Ebene in allen 8 Ebenen zu ermöglichen.

2.8.2.7 Ausgabe der Codierung über die serielle Schnittstelle

Abhängig von den momentanen Zuständen der Tasten SHIFT, CONTROL und BLOCK verzweigt das Programm nach Auffindung der Tastenpositionsnummer in eine der 8 Tastaturebenen und holt aus der zugeordneten, programmierten Tabelle, die sich entweder fest im Flash-Speicher des Mikroprozessors oder im umprogrammierbaren seriellen EEPROM befindet, die zugehörigen Zeichen (oder auch Strings). Diese werden dann unmittelbar in den Ausgabespeicher geschrieben und aus diesem ohne Beachtung eines Handshaksignals über die serielle

RS232-Schnittstelle mit 9.600 Baud ausgegeben.

2.8.3 Aufbau der Codetabellen im seriellen EEPROM-Speicher

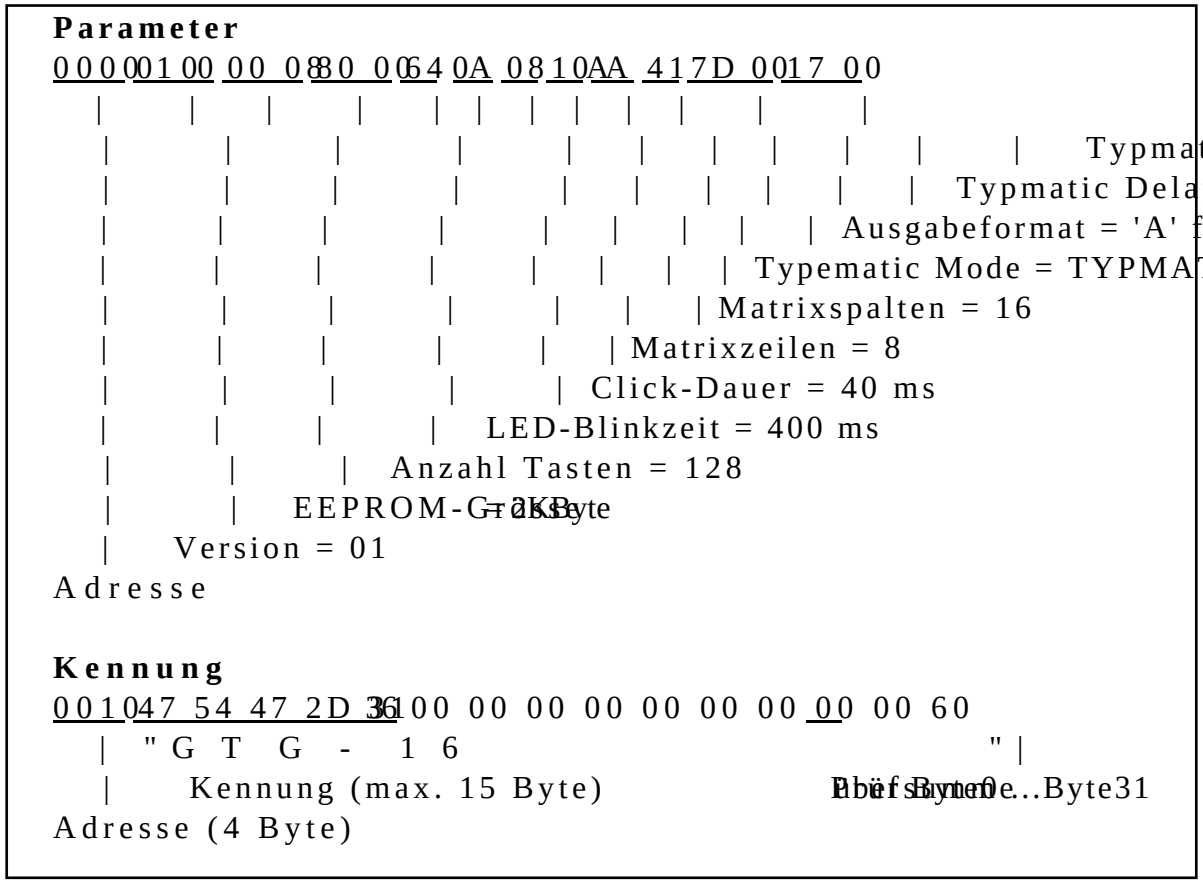
2.8.3.1 Parameterablage und Kennung

Die ersten 32 Byte in dem EEPROM dienen der Ablage von änderbaren Parametern und einer frei eingebbaren Kennung, die in 15 Byte beliebig genutzt werden kann. Das letzte Byte auf der Adresse 001F beinhaltet eine Prüfsumme über den Inhalt dieses ersten Feldes von 31 Byte.

2.8.3.2 Adressen Parameterablage und Kennung im EEPROM

Parameter und Kennungsstring	0000 (Hex) - 001F (Hex)	32 Byte

2.8.3.3 Zuordnung der Parameter und Kennung zu den einzelnen Bytes im EEPROM



2.8.3.4 Bedeutung der Codes in den Codetabellen

Befindet sich im einzelnen Speicherplatz eine 00 (Hex), so ist diesem Tastaurknotenpunkt keine Codierung in der entsprechenden Ebene zugeordnet. Das kann entweder dazu führen, das beim Betätigen der Taste keine Codierung ausgegeben wird, oder aber an dieser Stelle befindet sich keine Taste, so dass ein Code auch nicht ausgegeben werden würde, weil dieser Knotenpunkt bei nicht vorhandener Taste natürlich auch nicht ausgewählt werden kann. Ausgegeben werden diese einzeln codierten Zeichen als 8 bit-Zeichen mit dem obersten Bit auf 0 gesetzt, so dass vom Host diese als 7-bit-ASCII-Zeichen interpretiert werden können. Allerdings kann am der Controller auch einen ASCII-Hex-Mode ausgewählt werden, in dem jeweils 4 Bit des Bytes einem Hex-Wert zugeordnet werden (Werte: {1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E,F } d.h. Zeichen zwischen 01(Hex) bis 7F(Hex) werden jeweils in zwei Byte verschlüsselt, die den Hex-Wert des Codes in zwei ASCII-Zeichen darstellen.

2.8.3.5 Was ist eine ASCII-Hex-Codierung ?
Hexwerte: 4-Bit-Folge entsprechen Zahlen 0100: = 4 und 0101:=5
ASCII-Codierung: 69 (Dezimal):="E"
Ein Byte in 2 x 4 Bitauflösung: 0100 0101:= 69 (Dezimal):= 45(Hex):= ASCII-Codierung für "E"
ASCII-Codierung für "4":=34(Hex); ASCII-Codierung für "5":=35(Hex)
ASCII-Hex-Codierung für 45 (Hex):= "4" "5" in zwei hintereinander kommenden ASCII-Zeichen und das entspricht zwei Bytes:= 34(Hex) 35(Hex) mit der Bitfolgen: 0011 0100 0011 0101.

2.8.3.6 Adressbereiche der Codetabellen im EEPROM
Die Ablage der Codetabellen im EEPROM ist jeweils in den Speicherzellen mit folgenden Adressbereichen:

Codetabelle E1 UNSHIFT	0080 (Hex) - 00FF (Hex)	128 BYTE
Codetabelle E2 SHIFT	0100 (Hex) - 017F (Hex)	128 BYTE
Codetabelle E3 CONTROL	0180 (Hex) - 01FF (Hex)	128 BYTE
Codetabelle E4 SHIFT+CNTRL	0200 (Hex) - 027F (Hex)	128 BYTE
Codetabelle E5 BLOCK+UNSHIFT	0280 (Hex) - 02FF (Hex)	128 BYTE
Codetabelle E6 BLOCK+SHIFT	0300 (Hex) - 037F (Hex)	128 BYTE
Codetabelle E7 BLOCK+CNTRL	0380 (Hex) - 03FF (Hex)	128 BYTE
Codetabelle E8 BLOCK+SHIFT+CNTRL	0400 (Hex) - 047F (Hex)	128 BYTE

2.8.3.7 Stringablage
Findet das Programm in der Codetabelle ein Byte mit dem obersten Bit gesetzt, so bedeutet dies, dass für diese Taste (auch in den verschiedenen Ebenen) einen String abgelegt ist.
Strings befinden sich in der Stringtabelle und werden dort wie folgt zugeordnet:
Das in der Codetabelle vorgefundene Byte hat in den unteren 7 Bit die Nummer des zugeordneten Strings programmiert. Somit sind maximal 127 Strings programmierbar.
Der erste String beginnt an der Adresse 0480(Hex) im EEPROM. Das dort befindliche Byte beinhaltet die Länge des Strings, d.h. die Anzahl der Bytes, die jetzt folgen und den Inhalt des Strings bilden. In diesen Bytes lässt sich übrigens auch das 7. Bit auf 1 setzen, so dass in den Strings auch Zeichen aus dem auf 255 Zeichen erweiterten ASCII-Zeichensatz abgelegt und von dort ausgesendet werden können.
Die Länge eines Strings ist auf 30 Byte begrenzt. Das unmittelbar folgende Byte beinhaltet als Zahl wiederum die Länge des folgenden Strings usw. Sucht also der Controller z.B. den 5.String, so liesst das Programm jeweils an den entsprechenden Adressen die Länge des nächsten Strings und kann so der Stringfolge folgend sich von String zu String vorzählend zum gesuchten String hangeln. Dadurch wird der Speicherplatz für die Stringablage (ca. 1kByte) optimal dynamisch genutzt.
Allerdings: Werden Fehler bei der Eingabe gemacht, so kann der String nicht mehr gelöscht werden. Wird also der gleichen Tastenposition ein korregierter String zugeordnet, so wird dieser auf noch freiem Speicherplatz abgelegt. Ein sogenanntes Packing wird nicht durchgeführt, so dass bei häufigem Wechseln der Stringinhalte letztlich der Speicher vollläuft. Dann hilft, soll eine weitere Korrektur erfolgen, nur noch die Löschung des gesamten Speichers (siehe Pkt. 3.2.4 Sonderfunktion init)

2.8.3.8 Addressbereiche der Strings im EEPROM

Stringtabelle	STRINGS	0480 (Hex) - 07FF (Hex)

2.8.3.9 Anordnung der Daten im EEPROM
Das folgende Bild zeigt, wie die Daten den Tastennummern zugeordnet werden.

```
Daten
0080 1B 31 32 33 34 35 36 37 38 39 30 00 FD 00 00 5E
   |  |           |                          |           |
   |  |           |              KeyNr 11 (R0,C11)= keine  Codierung
   |  |   KeyNr 3 (R0,C3)="3"                          |
   |  KeyNr 0 (R0,C0) = 'ESC'                    KeyNr15 (R0,C15) = '^'
   |
Adresse (4 Byte):  0080...UNSHIFT
                   0100...SHIFT
                   0180...CONTROL
                   0200...SHIFT+CONTROL
                   0280...BLOCK+UNSHIFT
                   0300...BLOCK+SHIFT
                   0380...BLOCK+CONTROL
                   0400...BLOCK+SHIFT+CONTROL
                   0480...Beginn der Strings
                   07FF...Ende der Strings
```

2.8.3.10 Sonderprogrammierung des FLASH-EEPROMS auf dem Controllerchip.
Wird die freie Programmierung nicht genutz und soll der Controller in grösseren Mengen für eine feste Tastaturanordnung verwendet werden, so kann die Scanncodetabelle auch im Flash-EEPROM des µ-Controllerchips abgelegt und auf den Einbau des separaten seriellen EEPROMs verzichtet werden.
Dies ist dann eine kundenspezifische Programmierung und lohnt sich nur bei Serien über 500 Stck. pro Fertigungslos.

2.8.4 Full-N-Key-Rollover

Es ist die Funktion "Full-N-Key-Rollover" programmiert. Bei dieser Funktion erkennt das Programm alle gedrückten Tasten in der Matrix eindeutig, auch wenn mehrere Tasten gleichzeitig betätigt wurden. Die Auto-Repeat-Funktion wirkt dabei aber stets auf die zuletzt gedrückte Taste. Um diese Funktion des Controllers zu unterstützen, ist die Anordnung von Dioden in den Kreuzungspunkten der Tastaturmatrix (zur Vermeidung von Schattentasten, bei jeder Taste) erforderlich. Die Funktionsfähigkeit des Controllers bleibt auch ohne Entkoppeldioden erhalten, die Funktion „Full-N-Key-Rollover" kann dann aber nicht generiert werden und es besteht die Gefahr von Schattentasten.

2.8.5 Typematic (Auto-Repeat)

Das Tastaturprogramm hat die Eigenschaft „Typematic". Wird eine Taste gedrückt und bleibt gedrückt, so wird nach einer als Parameter (Typmatic delay x) festlegbaren Zeit begonnen, die Zeichenausgabe zu wiederholen. Dies geschieht mit einer Geschwindigkeit, welche durch die ebenfalls als Parameter festlegbare Wiederholzeit (Typematic repeat) bestimmt wird.

2.8.5.1 Parameter für Typematic delay
Die Standardeinstellung ist ca. 500ms für die Wartezeit.
Sie wird im EEPROM in das 13. und 14. Byte der Parameterliste eingetragen.

Der dort stehende Zahlenwert (in Hex) wird mit 4 ms multipliziert und ergibt so die Verzögerungszeit, nach der die Wiederholung der Ausgabe des Zeichens nach dem Start des Drük-

kens der Taste begonnen wird.
Im Teach-In-Verfahren stehen dafür 2 feste Werte (250; 500; 750 und 1.000 ms) zur Verfügung.
Programmierung mit Teach-In: siehe Seite 24

Standardeinstellung	007D	(500ms)
Alternativen	003E	(250ms)
	00BB	(750ms)
	00FA	(1000ms)

Bitte beachten: In den Files steht, bezogen auf die aufsteigende Adresse, bei Zweibytewerten das niederwertige Byte vor dem höherwertigen Byte.

2.8.5.2 Parameter für Typematic repeat
Die Standardeinstellung für die Wiederholfrequenz ist ca. 92 ms (10,9 Hz). Der Parameter für diese Zeit ist im EEPROM im 15. und 16. Byte eingetragen.
Der dort stehende Zahlenwert (in Hex) wird mit 4 ms multipliziert und ergibt so die Zeitabstände, mit denen die Wiederholung Zeichenausgabe erfolgt.

Im Teach-In-Verfahren stehen dafür 2 feste Werte (92 ms ; 460 ms) zur Verfügung.
Programmierung mit Teach-In: siehe Seite 24

Standardeinstellung	0017	(92 ms / 10,9 Hz)
Alternative	0073	(460 ms / 2,2 Hz)

Bitte beachten: In den Files steht, bezogen auf die aufsteigende Adresse, bei Zweibytewerten das niederwertige Byte vor dem höherwertigen Byte.

2.8.6.3 Typmatic Mode
Soll die Zeichenausgabe ohne Wiederholungen erfolgen, so muss das 12. Byte den Wert 0000 (Zeichenausgabe nur beim Drücken der Taste) oder 0055 (Zeichenausgabe beim Drükken der Taste , Break-Kode 0F0 + Zeichen beim Loslassen der Taste) erhalten.
Programmierung mit Teach-In: siehe Seite 24

Standardeinstellung	AA	(Typmatic)
Alternativen	00	(nur Make)
	55	(Make / Break)

2.8.6 Piepser (Click)
Durch den Piepser wird eine akustische Rückmeldung für einen Tastendruck (Tasten-Click) möglich. Der Pin „Piepser" geht beim Drücken einer Taste, wenn dort eine Codierung gefunden wird, für die Piepzeit auf LOW. Vorraussetzung ist, dass in der Parameterliste im EEPROM das 8.Byte auf 01(Hex):= Klick aktiv gesetzt ist.
Der interne Piepser kann über den Jumper J9 ein- bzw. ausgeschaltet werden. Über J6/14 und J6/47 (+5V) kann bei abgeschaltetem internen Piepser auch ein externer Piepser (5V=; max. 40 mA) betrieben werden.

2.8.7 Tastenspeicher
Der Controller besitzt einen FIFO-Speicher (First-In-First-Out) für 16 Bytes, in dem die Tastencodes vor der Ausgabe zwischengespeichert werden. Allerdings sendet der Controller die Daten mit 9600 Baud ohne Berücksichtigung irgend eines Handshakesignals hinaus.

2.8.8 Power On Reset
Nachdem die Stromversorgung eingeschaltet ist erzeugt der Controller einen Reset, der ca.

80 ms lang ist. Der Controller wird in der folgenden Initialisierung in einen definierten Anfangszustand überführt.
Der Ausgabepuffer wird gelöscht, der Scan-Zähler wird zurückgestellt, die LEDs werden rückgesetzt, es erfolgt die Festlegung des Protokolles für die Datenschnittstelle.

2.8.9 Watchdog

Der Controllerchip verfügt über einen internen Watchdog, der bei einer Störung im Programm den Controller automatisch in einen RESET-Mode bringt, so dass wie nach einem POWER-ON-RESET die Initialisierung neu durchlaufen wird.

2.8.10 Stromsparmodus

Diese Funktion kann über den Jumper J7 aktiviert werden.
Wenn keine Taste gedrückt ist, so wird der Controller nach Aussenden der letzten Information in den Stop-Mode geschaltet, in dem er nur wenig Strom (< 2 mA) verbraucht. Das Aufwekken erfolgt durch:
a) Datenleitung TxD auf der seriellen-Schnittstelle (V24 Pegel oder TTL)
auf High ziehen (PC-System)
b) Tastendruck auf eine beliebige Taste (nur falls diese Funktion bestückt ist)

2.8.11 Tastatursperre

Über ein Schlüsselschalter-Kontakt kann die Tastatur gesperrt und freigegeben werden. Dieser befindet sich nicht innerhalb der Tastenmatrix sondern am Stecker J8/3. Wird dieser Pin auf LOW gezogen, so erfolgt die Sperrung der Tastatur.

2.8.12 Anschluß der Tastatur über Steckverbinder J6

Wie aus der Tabelle der Steckerbelgung für J6 auf Seite 13 ersichtlich, sind die 99 Anschlußpunkte in der bei GeBE standardisierten 3reihigen Anschlußleiste nicht alle belegt. Trotzdem ist in den Kontrollern mit Standardbestückung die 99polige Stiftleiste eingebaut, die aber bei Sonderbestückungen, das kann sich ab 25 Controllern pro Fertigungs- und Lieferlos lohnen, genau auf die Stiftleiste abgestimmt werden kann, die der letzlich zu bedienenden Anschlussbelegung dient.
Die Standardanordnung der 99 PINs ist so gewählt, dass Tastenmatrixen mit wechselnder Belegung (R; C; R; C...) oder Belegungen mit nur Reihen und dann nur Spalten (R; R; R; ...; C; C; C; .....) in geschlossenen einreihig oder zweireihigen Untermengen der Steckerbelegung des Steckerfeldes J6 erreicht werden können. Beispiele sind bei der Tabelle der Steckerbelgung für J6 angeführt.

2.8.13 Serielle Datenschnittstelle

2.8.13.1 Schnittstellenbelegung

Anschluß des Controllers an den Host erfolgt über den 10poligen MICA Steckerverbinder J2.
Die PIN-Beleging ist in der Tabelle auf Seite 11 ersichtlich.
Die PIN-Belegung ist in 9 Leitungen zu den COM-Schnittstellenstreckern am PC kompatibel, wenn über ein Flachbandkabel die ersten 9 PINs mit einem SUB-D Stecker verbunden werden.

2.8.13.2 Datenformat

Im Controller werden lediglich die Sende- (RxD) und die Empfangsleitung (TxD) verwendet.
Standardmässig ist das Datenformat wie folgt:
9.600 Baud, 8 Bit, no parity, 1 Stop Bit.

2.8.14 Stromzuführung:

An den seriellen COM Schnittstellen hat der PC keine +5V Stromversorgung, so dass der Controller über die separat herauszuführende Leitung PIN10 die Stromversorgung erhält. Andererseits kann die Stromversorgung auch über den PS/2 Schnittstellenstecker J3 (PIN4 und GND auf PIN3) erfolgen.

2.9 STEUERKOMMANDOS VOM HOST

2.9.1 Zusammenfassung der Control-Befehle:

(temporäre Parameter werden nach SW-Reset oder Power-Up auf Defaultwerte aus dem EEPROM zurückgestellt.)

16 (10H)	Tastatursperre / Freigabe (neu gegenüber GCK950)
20 (14H)	Tastenclickdauer / Piepser vom Host gesteuert
21 (15H)	Typmatic-Delay (Wiederholverzögerung für Autorepeatfunktion)
22 (16H)	Typmatic Repeat (Wiederholzeit für Autorepeatfunktion)
23 (17H)	Typmatic-Mode (incl. Break-Code Generierung)
24 (18H)	LED Daten
25 (19H)	LED Blink-Modus
26 (1aH)	LED Blinkzeit
27 (1bH)	reserviert
28 (1cH)	reserviert
29 (1dH)	EEPROM-Dump (neu gegenüber GCK950)
30 (1eH)	SW-Versionsanzeige + Ausgabe
31 (1fH)	Software-Reset

2.9.2 Spezielle Steuerungsfunktionen:

Wird über die serielle Schnittstelle das Steuerungsbyte 29 (1dH) gesendet, so erfolgt die Ausgabe des EEPROM-Speicherinhaltes (Parameter und Tastaturtabellen) über die serielle Schnittstelle. Dabei kann im Terminalprogramm eine Aufzeichnung als Textdatei erfolgen, welche anschließend editiert und dann als neue Master-Tabelle für das automatische Teach-In verwendet werden kann.
Dieses Kommando ist gleichbedeutend mit dem „Sonderkommando d“ beim manuellen Teach-In.
Wird über die serielle Schnittstelle das Steuerungsbyte 30 (1eH) gesendet, so erfolgt die Ausgabe Versionsnummer der Softwar.
Wird das Steuerungsbyte 31 (1fH) gesendet, so führt der Controller einen Softwarereset durch, der alle eingestellten Parameter wieder in den Default Zustand bringt.

2.9.3 Steuerungskommandos zur Änderung der Parameter:

Tastatursperre:
Die Tastatur kann temporär gesperrt bzw. freigegeben werden. Dazu sind die Steuerbytes 16 48 für Tastatursperre aus bzw. 16 49 (10H 30H oder 10H 31H) für Tastatursperreein zu senden.

Tastenclick:
Ein automatischer Tastenklick mit einer voreingestellten Dauer von 40 ms ist auf dem Controller vorhanden, welcher deaktiviert bzw. in der Dauer eingestellt werden kann.
Zur Einstellung der Tastenclickdauer ist über die serielle Schnittstelle das Controllbyte 20 (14H) zu senden und danach die gewünschte Tastenclickdauer in n x 4,1 ms (1 x 4,1 ms bis 254 x 4,1 ms).
Zur Deaktivierung des Tastenclick ist 20 0 oder 20 255 (14H 00H oder 14H ffH) zu senden.
Beispiel:
Einstellen der Tastenclickdauer auf 100 ms: 20 24 (14H 18H)
Mit diesem Kommando kann auch vom Host die akustische Signalisierung gesteuert werden.

Autorepeat-Funktion:
Die integrierte Autorepeat-Funktion der Tastatur kann in ihren Einstellungen verändert b.z.w. deaktiviert werden. Die Standarteinstellung bietet eine Verzögerung von 500 ms und eine Wiederholzeit von 100 ms (10 Hz).
Zur Einstellung der Wiederholverzögerung (Verzögerungszeit in der Taste gehalten werden muß, bis Auto-Repeat beginnt) der Autorepeat-Funktion ist über die serielle Schnittstelle das Controllbyte 21 (15H) zu senden und danach die Wiederholverzögerungszeit in n x 4,1 ms (1 x 4,1 ms bis 254 x 4,1 ms).
Zur Einstellung der Wiederholzeit (Zeit zwischen wiederholten Zeichen bei ständig gehaltener Taste) der Autorepeat-Funktion ist über sie serielle Schnittstelle das Controllbyte 22 (16H) zu senden und danach die Wiederholzeit in n x 4,1 ms (1 x 4,1 ms bis 254 x 4,1 ms).
Beispiele:
Einstellen der Wiederholverzögerung auf 800 ms: 21 195 (15H C3H)
Einstellen der Wiederholzeit auf 300 ms (3.3 Hz): 22 73 (16H 49H)

Zur Deaktivierung der Auto-Repeat-Funktion ist der Typmatic-Mode auf 00H (nur Make-Code) oder 55H (nur Make-Code + Break-Code) einzustellen; d.h. es ist 23 76 / 17H 4cH / 23 < L > bzw. 23 77 / 17H 4dH / 23 < M > zu senden.
Zur Aktivierung der Auto-Repeat-Funktion ist der Typmatic-Mode auf AAH (Make-Code mit Wiederholungen) einzustellen. Dieses erfolgt mit dem Kommando 23 74 / 17H 4aH / 23 <J>.

2.10 LEUCHTDIODEN

An den Controller sind bis zu 4 Leuchtdioden direkt anschließbar.
Diese können über die serielle Schnittstelle einzeln ein- und ausgeschaltet werden oder mit einer einstellbaren Frequenz blinken.
Achtung! Werden LED1 (CAPS-LOCK-LED) bzw. LED 4 (BLOCK-LED) für die Signalisierung des Tastaturzustandes verwendet, so kann es zu Fehlanzeigen kommen, da diese direkt vom Kontroller gesteuert werden.

Ansteuerung der Leuchtdioden:
Die Steuerung der Leuchtdioden erfolgt durch eine 2-Byte Übertragung. Sollen einzelne Leuchtdioden eingeschaltet werden, so lautet das vorangestellte Steuerungsbyte 24 (18H) und im nachfolgenden Byte wird für jedes gesetzte Bit eine LED eingeschaltet. Dabei entspricht Bit 0 (LSB) der LED 1, Bit 1 der LED 2 u.s.w.
Sollen einzelne Leuchtdioden in den Blinkmodus geschaltet werden, so lautet das vorangestellte Steuerungsbyte 25 (19H). Für jedes gesetzte Bit im nachfolgenden Byte wird eine LED in den Blinkmodus geschaltet.
Beispiele:

Alle LED ausschalten:	24 00 (18H 00H)
LED 1 und LED 3 an:	24 10 (18H 0aH) (entspricht 00001010B)

Werden einzelne Leuchtdioden erst eingeschaltet und dann in den Blinkmodus gebracht, so blinken diese gegenphasig zu den LED's welche nur in den Blinkmodus geschaltet werden.
Beispiele:

LED 1 und LED 3 blinken:	25 10 (19H 0aH) (entspricht 00001010B)
Alle LED blinken:	25 255 (1aH ffH)

Zur Änderung der Blinkzeit der blinkenden Leuchtdioden ist das Steuerungsyte 19 (13H) zu senden und danach die Blinkzeit in n x 4,1 ms (1 x 4,1 ms bis 254 x 4,1 ms)
Beispiel:
Blinkzeit der Leuchtdioden auf 1s einstellen: 26 244 (1aH F4H)
Der Defaultwert der Blinkzeit nach einem Reset beträgt 400 ms. Zur Änderung der Defaultwerte siehe: "Teach-In“.

3 Programmierung der Tastatur

3.1 AUFBAU HARDWARE FÜR DAS TEACH-IN-VERFAHREN

3.1.1 Allgemeine Information:

Der Tastatur Controller lässt sich auf eine bereits vorliegende Tastenmatrix anpassen.
Dazu wird für jede Taste ein Code bzw. ein String im EEPROM eingetragen..

Um in einen der beiden Programmieralgothithmen zu gelangen, wird die Progammiermode-auswahltaste (Teach-In Taste auf der kleinen Konsole) beim Einschalten (Power -On-RESET oder RESET nach Drücken der Resettaste) des Controllers gedrückt gehalten. Der Controller durchläuft zunächst die Initialisierung, schaut ob die Teach-In-Taste gedrückt ist. Ist dies der Fall, dann gibt er zunächst einen einfachen, nach einer Sekunde einen doppelten Piepton ab. Je nachdem, wann die Teach-In Taste losgelassen wird, startet der Controller in eines der zwei Teach-In Verfahren:

- Manuelles Teach-In Verfahren
- Auto Load Teach-In (Kopie von externer Master-Tastatur oder vom PC)

3.1.2 Anschluss des Terminals

(>> Blochschaltbild über Teach-In).

Hinweis:
Die Funktion der Lehrer-Tastatur kann vorteilhaft ein PC mit entsprechendem Terminalprogramm
(z.B. Hyperterminal.exe, Telemate.exe usw.) übernehmen.
Vorteil: Es wird die Tastennummer der zu programmierenden Taste und deren bisheriger und neuer Inhalt angezeigt. Dabei besteht zusätzlich die Möglichkeit, den ganzen Programmierablauf zu protokollieren, indem vorher die Option "Textdatei empfangen" aktiviert wurde. Dies gilt vor allem für den Dump des EEPROM-Inhaltes für das spätere, automatische Teach-In (bei Kopieren auf andere Controller) aus einer Textdatei.

Der Anschluß als Terminal erfolgt bei PC über eine seriellen COM-Schnittstelle. Das 9poliges V24-Kabel (Kabel 4) wird auf dem Controllerboard an den MICA Stecker J2 angesteckt.
Achtung: Leitungen TxD und RxD müssen gekreuzt sein!

3.1.2.1 Kommunikationsparameter

Die Parameter für das Datenformat der serielle Übertragung sind:
9.600 Baud, 8 data bit, no parity, 1 stop bit.
Das bedeutet: Es werden asynchron mit einer Baudrate von 9.600 Baud 10 Bit pro Zeichen übertragen. Das Datenformat startet mit einem START-Bit, diesem folgen 8-Daten-Bits, in denen das obere (MSB) Bit = 0 gesetzt ist, d.h. es werden 7-Bit ASCII-Zeichen (128 verschiedenen) verwendet. Das Datenformat enthält kein Parity-Bit (no parity) und wird mit einem Stopbit abgeschlossen.
Im Programmiermode wird, wo nötig, mit X-ON/X-OFF Protokoll als Handshakeverfahren gearbeitet.

3.1.3 Stromversorgung für den Controller

Mit dem Kabel3 wird über den Stecker J3 +5V Betriebsspannung zugeführt (ca. 12 mA).

3.1.4 Bedienkonsole mit Anzeige LEDs

Am Tastatur Controller wird über J4/5 an J4/3 die Progammiermodeauswahltaste (Teach-In Taste) angeschlossen. Wird dazu die Bedienkonsole mit Anzeige LEDs und der Progammiermodeauswahltaste (Teach-In Taste) an den Stecker J4 angeschlossen, so sind auch glei drei LEDs für die Anzeige des Programmierstatus vorhanden. (Ausrüstung im Starterkit).

3.1.5 Anschluß der Tastaturmatrix

Aus dem mehrfach belegten, 99poligen Steckerfeld des Controllers (J6) die Anschlusspins so heraussuchen, dass eine möglichst passende 1:1 Kabelverbindung zustande kommt. In der auf Seite 13 gezeigten Tabelle sind mehrere passende Beispiele aufgeführt. Wichtig ist, dass die Reihen- und Spaltenanschlüsse (Rx und Cy) möglichgst optimal angeordnet sind, damit ein kostengünstiger Anschluss möglich wird.

Im Starterkit GCT-972-RS232-START wird eine Experimentiertastatur mit 16 Tasten mitgeliefert, die 8 Anschlüsse hat, 4 geschlossen nebeneinanderliegende Reihen- und 4 geschlossen nebeneinanderliegende Spaltenleitungen. Diese lassen sich an die Anschlüsse J6/PIN39 bis J6/PIN 60 in der dritten Reihe mit einem einreihigen Pfostenstecker direkt anschliessen.

3.2 MANUELLES TEACH-IN VERFAHREN

26

3.2.1 Bedienungsablauf, Kurzfassung:

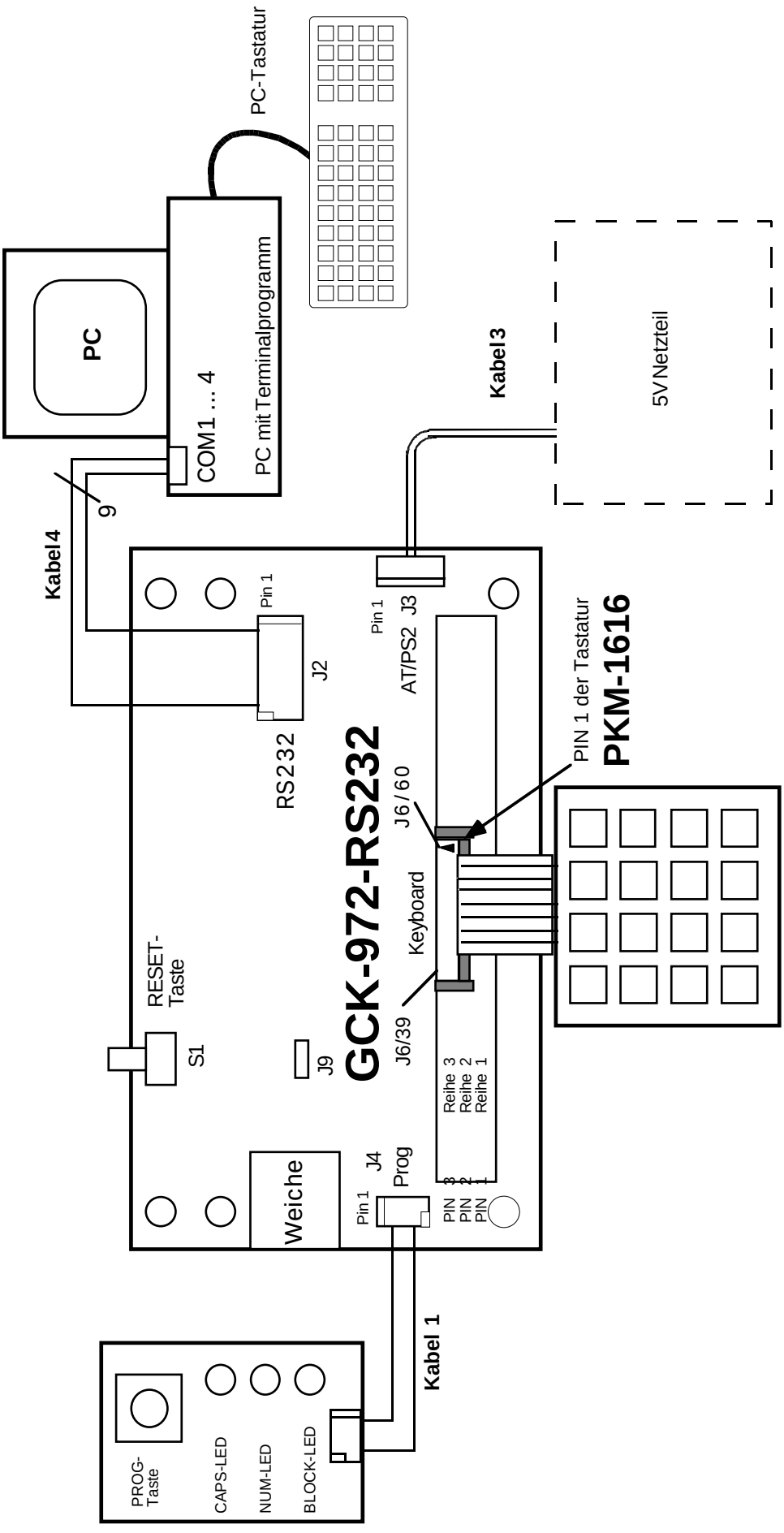
1. Teach-In Taste (Progammiermodeauswahltaste auf der kleinen Konsole) gedrückt halten. Stromversorgung einschalten oder bei bereits eingeschalteter Stromversorgung die RESET-Taste drücken und wieder loslassen.
2. Nach dem ersten Piepton Teach-In Taste loslassen. Dadurch wird Controller in den manuellen Teach-In-Mode geschaltet. >> LED1 (CAPS-LOCK) leuchtet auf.
3. Die gewünschte, zu programmierende Taste an der Tastatur drücken und wieder loslassen. >> LED2 (NUM-LOCK) leuchtet und zeigt damit an, dass jetzt die Eingabe auf der externen Programmiertastatur (dem Terminal) erwartet wird.
4. Eingaben an der externen Tastatur vornehmen.
5. Die zu programmierende Taste nochmals drücken und wieder loslassen.
6. Der Controller quittiert mit einem OK-Piep, falls er die Programmierung akzeptieren konnte oder bei Zurückweisung mit einem Fehler-Piep.
   Wurde keine Eingabe an der externen Tastatur gemacht, wird die zu programmierende Taste aus dem EEPROM gelöscht. Sie hat dann keine Funktion mehr.
7. Um weitere Programmierungen vorzunehmen wird bei Punkt 3. fortgefahren.
8. Ausschalten der Stromversorgung des Controllers oder Drücken der Reset-Taste beendet den Programmiermodus.

3.2.2 Ausführlicher Bedienungsablauf bei Teach-In:

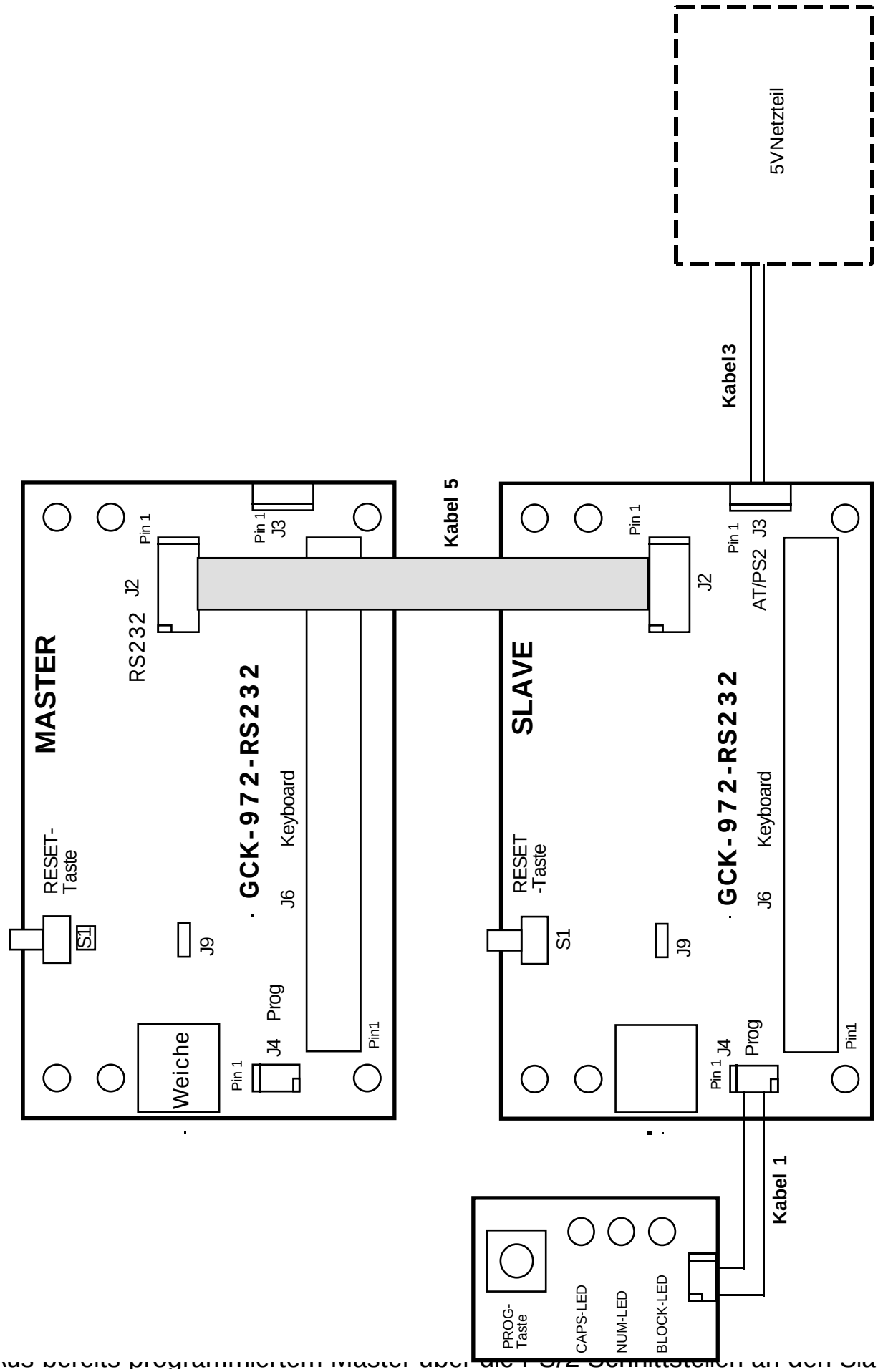
Nach aktivieren von Manual Teach-In (>> Schritte 1. und 2.) prüft der Controller, ob das EEPROM initialisiert ist, wenn nicht wird eine Initialisierung (Tabelleninhalte aller Zellen auf 00(Hex) stellen:= alle Tasten ohne Funktion, Standardinitialisierung in Parameter- und Kennungsstring schreiben, mit Prüfsumme abschliessen) durchgeführt. Damit ist der Programmiermode für manuelles Programmieren im Teach-In-Verfahren eingestellt. LED1 (CAPS-LOCK) leuchtet permanent:

Der Controller erwartet jetzt die Auswahl einer Taste an der zu programmierenden Tastatur

Blockbild: Manuelles Teach-In und Down Load über PC



Blockbild Teach-In durch Down Load von Master auf Slave



Aus bereits programmiertem Master über die PS/2-Schnittstellen an den Slave

durch Drücken (3.) genau dieser Taste. Diese Auswahl wird durch einen kurzen Piepton vom Controller bestätigt. LED1 (CAPS-LOCK) blinkt so lange, wie eine Taste gedrückt ist. Werden mehrere Tasten betätigt, gilt die zuletzt losgelassene Taste als ausgewählt. Diese merkt sich der Controller. Nach der Auswahl darf kein Matrixpunkt mehr geschlossen sein, um im Programmierverfahren fortfahren zu können. Ansonsten blinkt zur Warnung die LED1 (CAPS-LOCK). Ist die Auswahl abgeschlossen, dann leuchtet die LED2 (NUM-LOCK) permanent.

3.2.2.1 Festlegung von SHIFT, CONTROL und BLOCK

Sinnvollerweise beginnt man bei der Programmierung der Funktion einer Tastatur mit der Bestimmung, ob Tasten und welche die Steuertasten SHIFT, CONTROL und BLOCK werden sollen. Die festlegung erfolgt mit dem Eingabealgorithmus für Sonderfunktionen:
Eingaben von Sonderfunktionen werden eingeleitet durch fünfmaliges Betätigen der Eingabetaste (RETURN) (in der folgenden Auflistung steht das Zeichen ø für die einmalige Betätigung der Eingabetaste) an der externen Tastatur - der Controller quittiert nach kompletter Eingabe mit einem dreifach Piepton. Sonderfunktionsaufrufe müssen also nicht quittiert werden.
Danach wird der entsprechende Kennbuchstabe für Sonderfunktion aus folgender Liste eingeben:

ØØØØØg	gross	Die zu programmierende Taste wird SHIFT-Taste (on push) in allen Ebenen
ØØØØØs	strg	Die zu programmierende Taste wird CONTROL-Taste in allen Ebenen
ØØØØØb	block	Die zu programmierende Taste wird Blockmode Taste (on push) in allen Ebenen
ØØØØØt	toggle	Die zu programmierende Taste wird Blockmode Taste (toggle) in allen Ebenen
ØØØØØc	caps	Die zu programmierende Taste wird CAPS-Taste (entspricht SHIFT-LOCK) in allen Ebenen.

3.2.2.2 Besonderheit beim manuellen Teach-In: Automatisches Eintragen

a) Die Sondertasten SHIFT, CONTROL , CAPS und BLOCK werden automatisch in allen Ebenen eingetragen.

b) Während der Programmierung der Ebene E1 (UNSHIFT) werden die Buchstaben A bis Z automatisch in Gross-und Kleinschreibung in den Tabellen E1 und E5 bzw. E2 und E6 eingetragen; die entsprechenden Control-Zeichen in E3 und E7. Alle anderen Zeichen müssen über die Ebenenauswahl E1 bis E8 programmiert werden.

3.2.2.3 Auswahl der Tastenebene

Mit den folgenden Sonderfunktionen wird die Ebene angewählt, in der Tasten programmiert werden sollen

ØØØØØ1	E1	Tasten in Ebene 1 (UNSHIFT) programmieren
ØØØØØ2	E2	Tasten in Ebene 2 (SHIFT) programmieren
ØØØØØ3	E3	Tasten in Ebene 3 (CONTROL) programmieren
ØØØØØ4	E4	Tasten in Ebene 4 (SHIFT+CONTROL) programmieren
ØØØØØ5	E5	Tasten in Ebene 5 (BLOCK) programmieren
ØØØØØ6	E6	Tasten in Ebene 6 (BLOCK+SHIFT) programmieren
ØØØØØ7	E7	Tasten in Ebene 7 (BLOCK+CONTROL) programmieren
ØØØØØ8	E8	Tasten in Ebene 8 (BLOCK+SHIFT+CONTROL) programmieren

3.2.2.4 Einzeltaste programmieren:
Nachdem die Ebene bestimmt ist, in der die Taste programmiert werden soll, wird durch Drücken der ausgewählten Taste auf der zu programmierenden Tastaturmatrix die Programmiermöglichkeit eröffnet, die LED2 (NUM-LOCK) leuchtet und zeigt somit an, dass auf der externer ASCII-Tastatur (Terminal) eine Zeichentaste ausgewählt werden kann. Das Betätigen - (4.) einer einzigen Zeichentaste (A, B, ...,1; 2, §, %, oder .. oder...) an der externen Tastatur (Terminal) wählt den dadurch ausgelösten Code aus. Der durch diese Taste erzeugte Code wird der zu programmierenden Taste durch Eintragen in die entsprechende Stelle der angewählten Codetabelle zugewiesen. Auf der zu programmierenden Tastatur wird die selbe, vorher ausgewählte Taste nochmals gedrückt (5.) und damit der Einzelzyklus für die Programmierung einer Taste abgeschlossen. Der Kontroller quittiert mit einem Piep. Es kann durch Fortfahren bei 3. eine weitere Taste in der bereits angewählten Tastaturebene programmiert werden.

3.2.2.5 Strings programmieren:
Werden nach der Auswahl der zu programmierenden Taste (3.) nacheinander mehrere Tasten auf der externen Programmiertastatur (Terminal) betätigt, dann wird diese Tastensequenz als String aufgenommen. Dieser String wird durch nochmaliges Betätigen der zu programmierenden Taste dieser zugeordnet.

3.2.3 Sonderfunktionen programmieren:

3.2.3.1 Parameter im EEPROM programmieren

Mit den folgenden Befehlen können im manuellen Teach-In-Mode die Parameter und die Kennungen im EEPROM gesetzt werden. Auf der externen Tastatur (Terminal) werden zur Befehlseingabe die folgenden Tastensequenzen eingegeben. Das Zeichen Ø steht für einmalige Betätigung der Eingabetaste (Enter/Return):

ØØØØØh hex	Ausgabeformat ist ASCII-Hex (2 Byte)
ØØØØØa ascii	Ausgabe der Tastencodierung an der Tastaturschnittstelle erfolgt als ASCII-Zeichen (1 Byte pro Tastenzeichen)
ØØØØØo on	Ausgabe mit Click 40 ms (akustische Signalisierung)
ØØØØØn off	Ausgabe ohne Click (akustische Signalisierung)
ØØØØØw delay1	Wartezeit für typematic = 250 ms
ØØØØØx delay2	Wartezeit für typematic = 500 ms
ØØØØØy delay3	Wartezeit für typematic = 750 ms
ØØØØØz delay4	Wartezeit für typematic = 1000 ms
ØØØØØe repeat1	Wiederholzeit für typematic = 5*4*23 ms (2,2 Hz)
ØØØØØf repeat2	Wiederholzeit für typematic = 4*23 ms (10,9 Hz)
ØØØØØj mode1	Ausgabemodus Typmatic
ØØØØØl mode2	Ausgabemodus Make / Break
ØØØØØm mode3	Ausgabemodus nur Make

3.2.4 Initialisierungsbefehle:

Mit den folgenden Befehlen kann eine Neuprogrammierung für das EEPROM vorbereitet werden

ØØØØØi init	EEPROM initialisieren; alle Tasten ohne Funktion, Strings gelöscht
ØØØØØp prog	EEPROM initialisieren auf die Urform, wie sie im FLASH-Speicher des Controllerchips festgelegt ist. Bei der Ausführung dieses Befehls werden alle Daten, so wie sie in einer Tabelle im Flash-Speicher des Controllerchips festgelegt sind, in das EEPROM übertragen. Damit kann eine Grundprogrammierung eingestellt werden.
ØØØØØk kenn	EEPROM-Kennung programmieren. Nach der Eingabe des Kennbuchstabens 'k' können maximal 14 Zeichen als Kennungsstring eingegeben werden. Ein abschließendes 'ENTER' beendet dieses Kommando.

3.2.5 Kommunikationsbefehle:

Mit diesen Befehlen kann der Inhalt des EEPROMs über die serielle Schnittstelle ausgelesen werden. Damit erhält man eine gute Möglichkeit zur Analyse und Auffinden von Fehleingaben.

ØØØØØv vers	Ausgabe der Software-Version und die Version der Initialisierungstabelle
ØØØØØd dump	EEPROM Inhalt über die serielle Schnittstelle ausgeben (ASCII-Hex Fomat),wobei der Kennungsstring mit vorangestelltem Semikolon als Kommentar zuerst gesendet wird.

3.2.6 Beenden des einzelnen Programmiervorganges:

3.2.6.1 Abschließen der einzelnen Programmierung

Nach der Eingabe der gewünschten Codierung über die externe Tastatur (Terminal) wird durch nochmaliges Drücken der zu programmierenden Taste der Programmierzyklus für einen Befehl abgeschlossen. Der Controller programmiert die Taste und quittiert durch ein OK-Piep (zweifach Piepton). Wurden keine Eingaben an der externen Tastatur gemacht, wird die zu programmierende Taste aus dem EEPROM gelöscht. Sie hat dann keine Funktion mehr. Danach leuchtet LED1 (CAPS-LOCK) wieder.
entsprechend verhält sich das Programmierprogramm, wenn Strings eingegeben wurden. Befehle werden dagegen sofort nach Eingabe der Befehlsauswahl ausgeführt, quittiert und abgeschlossen.

3.2.6.2 Abbrechen während des Programmiervorganges:

Wurden bei der Eingabe der gewünschten Programmierung Fehler gemacht, so können diese nicht durch „Backspace" etc. korrigiert werden. Zum Abbruch wird dann auf der zu programmierenden Tastatur eine andere Taste als beim Start dieses Programmierzykel gedrückt. Der Controller zeigt den Abbruch mit Fehler-Piep (langer Piepton) an. DieCodierung der zuvor gewählten Taste bleibt erhalten. Die LED1 (CAPS-LOCK) leuchtet wieder auf und die Programmierung einer Taste kann wieder gestartet werden.

3.3 TEACH-IN DURCH DOWN-LOAD-VERFAHREN

3.3.1 Editieren und Duplizieren der EEPROM-Programmierung

3.3.1.1 Down-Load aus dem PC

Die Funktion der Lehrer-Tastatur (ASCII-Tastatur, die alle 256 verschiedenen Byte ausgeben kann, Serielle V.24 Schnittstelle mit 9600 Baud, 8Datenbits, no parity, 1 Stoppbit, Xon/Xoff Handshake) kann sinnvollerweise ein PC mit entsprechendem Terminalprogramm (z.B. Hyperterminal.exe, Telemate.exe usw.) übernehmen. (Blockbild wie beim Starterkit siehe Seite REF!).
Das Erstellen der EEPROM-Programmierung ist, wie oben gezeigt wurde, in dieser Kombination im manuellen Teach-In-Mode sehr einfach durchzuführen. Ziel ist es jetzt, die erarbeitete EEPROM-Programmierung auf den PC zu übertragen und dort in einer File zu sichern. Werden nämlich weitere gleiche Tastaturen produziert, so müssen diese nicht mühsam im manuellen Tech-In-Mode Taste für Taste programmiert werden, sondern die einmal erstellte Programmieruung wird aus der abgespeicherten File durch Hinunterladen (Down-Load) dieser File in das EEPROM programmiert.

3.3.1.2 Erzeugung der EEPROM-Tabellen in einer Dump-Datei

Es stehen zwei Verfahren zur Verfügung, mit denen der Inhalt des EEPROMs erstellt und umprogrammiert werden kann. Natürlich kann die Datei zunächst vollständig auf dem PC erstellt werden, was aber eine genaue Kenntnis aller Codes und Programmierzusammenhänge (vor allem bei der Erstellung und Einordnung von Strings) voraussetzt. Einfacher ist es zum Erstellen die Teach-In Methode zu verwenden, wie oben (Siehe Seite REF!) gezeigt wird. Zur Kontrolle und zum Editieren einer Datei ist die Bearbeitung auf dem PC vorteilhaft.

z.B. HYPERTERMINAL mit folgender Einstellung:

Übertragung --> Text aufzeichnen --> Datei --> Starten

Um die Programmierung in den PC laden zu können, sendet der PC den Kommunikationsbefehl dump = (ØØØØØd), der den Controller veranlasst, den EEPROM Inhalt (im ASCII-Hex Format) über die serielle Schnittstelle an den PC zu senden. Dort wird der Inhalt in einer Textfile gespeichert und angezeigt. Diese Textfile ist ohne Kommentare, da diese auf dem Controller nicht gespeichert werden. Es kann sich z.B. um eine zuvor im manuellen Teach-In-Mode erstellte EEPROM-Programmierung einer speziellen Tastatur handeln, die jetzt auch auf andere, gleiche Controller GCT-972-RS232 übertragen werden soll. Diese Datei kann natürlich im PC auch abgespeichert werden, so dass sie für das Duplizieren der Controllerfunktion ab sofort zur Verfügung steht, ohne dass jeweils ein Mastercontroller ausgelesen werden müsste.
Diese Datei kann nun editiert und mit Kommentaren (Über Semikolon abgetrennte Zeilen.) versehen werden, die beim Teach-In durch Down-Load aber auf dem Controller wiederum nicht gespeichert sondern beim Programmieren ausgeblendet werden.

z.B. HYPERTERMINAL mit folgender Einstellung:
Übertragung --> Textdatei senden --> Datei --> Starten

3.3.1.3 Down-Load aus einer Mastertastatur zum Slave kopieren

Für die Serienproduktion ist zum Down-Load einer bereits erstellten Programmierung (Master) nicht einmal ein PC notwendig. Eine bereits programmierte Master Tastatur GCK-972-RS232 wird an der seriellen Schnittstelle (J2) über das Kabel 5 (TxD und RxD sind gekreuzt !) an die serielle Schnittstelle (J2) der zu programmierenden Tastatur GCK-972-RS232 (Slave) angeschlossen. Siehe auf Seite REF! das Blockbild: Teach-In durch Down Load von Master auf Slave.
Das Datenformat ist ASCII-Hex. Zu diesem Kopierverfahren wird lediglich noch eine 5V Stromversorgung benötigt, die entweder aus der Tastaturschnittstelle des PCs oder einer externen Stromversorgung (+5V) entnommen werden kann. Der Mastercontroller erhält seine Stromversorgung über das Schnittstellenkabel vom Slave.
Die Programme zum Download befindet sich in der Firmware der beiden Controller.

3.3.1.4 Vorgehensweise

Siehe Blockschaltung:Teach-In durch Down Load von Master auf Slave auf Seite REF!
1.) Einen schon programmierten Controller GCK-972-RS232 (Master) mit Kabel 5 an J2 des zu programmierenden Controllers (Slave) anschließen.

Achtung: Leitungen TXD und RXD müssen gekreuzt sein !

2. Am Controller (Slave) die Teach-In Taste gedrückt halten und einschalten (RESET-Taste betätigen).
3. Sofort nach dem Zweifach-Piep die Teach-In Taste loslassen.

Der Slave-Kontroller sendet darufhin automatisch das Zeichen X-ON, was den Master veranlasst, den kompletten EEPROM-Inhalt zu senden.

4. Der Controller (Slave) quittiert nach ca 20 Sekunden das Ende der Programmierung durch einen OK-Piepton.

Tabelle Bestückung der Controller der Serie GCK-972

Merkmal	GCK972-PS/2-STD	GCK972-PS/2-EVAL 3)	GCK972-RS232-STD	GCK972-RS232-EVAL 3)	Sonderbestückung	Bemerkung
Firmware	PS/2-Schnittstelle	PS/2-Schnittstelle	RS232-Schnittstelle	RS232-Schnittstelle		
PC-Anschluss	5-polige Stiftleiste	5-polige Stiftleiste	nein	5-polige Stiftleiste		
Weiche	ja	ja	nein	ja, nicht benutzt		
Stecker ext. Tastatur	PS2-Buchse	PS2-Buchse	nein	PS2-Buchse	5-pol. Stiftleiste J10	
Matrixanschluss 16*8	99-polig 3-reihig	99-polig 3-reihig	99-polig 3-reihig	99-polig 3-reihig	ja, Untermengen	
EEPROM	ja	ja	ja	ja	ohne spez. Firmware	-> kein Teach-In
Piepser	ja, wählbar über J9	ja, wählbar über J9	ja, wählbar über J9	ja, wählbar über J9		
Powerdown	nein	ja, wählbar über J7	nein	ja, wählbar über J7		
RS232 mit V24 Pegel	nein	ja, nicht benutzt	ja	ja	TTL-Pegel	
SPI-Schnittstelle	nein	ja	nein	ja	ja, Stiftleiste 12-polig	extra Firmware 2)
Schlüsselschalter	ja, über J6	ja, über J6	ja, über J6	ja, über J6		
LED-Treiberschaltung	4x	4x	4x	4x		
Teach-In Anschluss	ja, über Weiche	ja, über Weiche	ja, über RS232	ja, über RS232		
Manual D/E	ja/ja	ja/ja	ja/ja	ja/ja		nicht Lieferumfang
KI (Kurzinformation) D/E	ja/ja	ja/ja	ja/ja	ja/ja		Katalog im Internet

1) Im Konzept vorgesehen, in der Software bereits realisiert

2) Z.B. Display, Kartenleser, Barcode, USB-Hub (alle z.Z. noch nicht existent)

3) GCK-972-PS/2-EVAL und GCK-972-RS232-EVAL Boards sind gleich bestückt. Software verschieden.

3.3.2 Beispiel: Listing einer Dump Datei

Der folgende Text zeigt das Beispiel einer editierten und mit Kommentaren versehenen Datei, wie sie auf einem PC gespeichert und angezeigt wird. Die nicht explizite im EEPROM gespeicherten Daten (Adressen und Kommentare) sind in einfacher Schrift. Daten, die explizite abgespeichert im EEPROM stehen, sind fett geschrieben:

```
;Dump vom 16.10.00
;GTG-16
0000 01 00 00 08 80 00 08 01 08 10 AA 41 7D 00 17 00
0010 47 54 47 2D 39 37 00 00 00 00 00 00 00 00 00 57
;
;UNSHIFT
0080 1B 31 32 33 34 35 36 37 38 39 30 00 FD 00 00 5E
0090 31 32 33 34 35 36 37 38 39 30 7F 34 FC 08 09 71
00A0 77 65 72 74 7A 75 69 6F 70 7C 2B 0D 00 FB 61 73
00B0 64 66 67 68 6A 6B 6C 7D 7E 23 3C 79 00 78 63 76
00C0 62 6E 6D 2C 2E 2D 20 37 38 39 34 35 00 36 31 32
00D0 33 30 2C 0D 00 00 3D 00 00 00 00 00 00 00 00 00
00E0 00 00 00 00 00 00 00 00 00 00 00 00 FC 00 00 00
00F0 00 00 00 00 00 00 00 00 00 00 00 FD FE 00 00 00
;
;SHIFT
0100 1B 31 32 33 34 35 36 37 38 39 30 00 FD 00 00 5E
0110 21 22 A7 24 25 26 2F 28 29 3D 3F 34 FC 08 09 51
0120 57 45 52 54 5A 55 49 4F 50 7C 2B 0D 00 FB 41 53
0130 44 46 47 48 4A 4B 4C 7D 7E 20 3C 59 00 58 43 56
0140 42 4E 4D 3B 3A 5F 20 37 38 39 34 35 00 36 31 32
0150 33 30 2C 0D 00 00 00 00 00 00 00 00 00 00 00 00
0160 00 00 00 00 00 00 00 00 00 00 00 00 FC 00 00 00
0170 00 00 00 00 00 00 00 00 00 00 00 FD FE 00 00 00
;
;CONTROL
0180 1B 81 82 33 34 35 36 37 38 39 30 00 FD 00 00 5E
0190 31 32 33 34 35 36 37 38 39 30 7F 34 FC 08 09 11
01A0 17 05 12 14 1A 15 09 0F 10 7C 2B 0D 00 FB 01 13
01B0 04 06 07 08 0A 0B 0C 7D 7E 23 3C 19 00 18 03 16
01C0 02 0E 0D 2C 2E 2D 20 37 38 39 34 35 00 36 31 32
01D0 33 30 2C 0D 00 00 3D 00 00 00 00 00 00 00 00 00
01E0 00 00 00 00 00 00 00 00 00 00 00 00 FC 00 00 00
01F0 00 00 00 00 00 00 00 00 00 00 00 FD FE 00 00 00
;
;SHIFT+CONTROL
0200 1B 00 00 00 00 00 00 00 00 00 00 00 FD 00 00 00
0210 00 00 00 00 00 00 00 00 00 00 00 00 FC 00 00 00
0220 00 00 00 00 00 00 00 00 00 00 00 00 00 FB 00 00
0230 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0240 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0250 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0260 00 00 00 00 00 00 00 00 00 00 00 00 FC 00 00 00
0270 00 00 00 00 00 00 00 00 00 00 00 FD FE 00 00 00
```

```
;
;BLOCK+UNSHIFT
0280 1B 31 32 33 34 35 36 37 38 39 30 00 FD 00 00 5E
0290 31 32 33 34 35 36 37 38 39 30 7F 34 FC 08 09 71
02A0 77 65 72 74 7A 75 69 6F 70 7C 2B 0D 00 FB 61 73
02B0 64 66 67 68 6A 6B 6C 7D 7E 23 3C 79 00 78 63 76
02C0 62 6E 6D 2C 2E 2D 20 37 38 39 34 35 00 36 2B 0B
02D0 2D 30 0A 0D 00 00 3D 00 00 00 00 00 00 00 00 00
02E0 00 00 00 00 00 00 00 00 00 00 00 00 FC 00 00 00
02F0 00 00 00 00 00 00 00 00 00 00 00 FD FE 00 00 00
;
;BLOCK+SHIFT
0300 1B 31 32 33 34 35 36 37 38 39 30 00 FD 00 00 5E
0310 21 22 A7 24 25 26 2F 28 29 3D 3F 34 FC 08 09 51
0320 57 45 52 54 5A 55 49 4F 50 7C 2B 0D 00 FB 41 41
0330 44 46 47 48 4A 4B 4C 7D 7E 20 3C 59 00 58 43 56
0340 42 4E 4D 3B 3A 5F 20 37 38 39 34 35 00 36 2B 32
0350 2D 30 0A 0D 00 00 00 00 00 00 00 00 00 00 00 00
0360 00 00 00 00 00 00 00 00 00 00 00 00 FC 00 00 00
0370 00 00 00 00 00 00 00 00 00 00 00 FD FE 00 00 00
;
;BLOCK+CONTROL
0380 1B 31 32 33 34 35 36 37 38 39 30 00 FD 00 00 5E
0390 31 32 33 34 35 36 37 38 39 30 7F 34 FC 08 09 11
03A0 17 05 12 14 1A 15 09 0F 10 7C 2B 0D 00 FB 01 13
03B0 04 06 07 08 0A 0B 0C 7D 7E 23 3C 19 00 18 03 16
03C0 02 0E 0D 2C 2E 2D 20 37 38 39 34 35 00 36 2B 32
03D0 2D 30 0A 0D 00 00 3D 00 00 00 00 00 00 00 00 00
03E0 00 00 00 00 00 00 00 00 00 00 00 00 FC 00 00 00
03F0 00 00 00 00 00 00 00 00 00 00 00 FD FE 00 00 00
;
;BLOCK+SHIFT+CONTROL
0400 81 82 00 00 00 00 00 00 00 00 00 FD 00 00 00
0410 00 00 00 00 00 00 00 00 00 00 00 FC 00 00 00
0420 00 00 00 00 00 00 00 00 00 00 00 00 FB 00 00
0430 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0440 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0450 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0460 00 00 00 00 00 00 00 00 00 00 00 00 FC 00 00
0470 00 00 00 00 00 00 00 00 00 00 00 FD FE 00 00 00
;
;Strings (keine abgelegt)
0480 00
;
;Dateiende
FFFF
```

3.4 KUNDENSPEZIFISCHE APPLIKATIONEN

Hard- und Software des Controllersystems GCK-972-RS232 wurde bei GeBE entwickelt und werden auch dort technisch gepflegt.
Das System steht für kundenspezifische Applikationen zur Verfügung.

3.4.1 Sonderbestückungen

Bei Serien ist die Anpassung an Kundenapplikationen durch Sonderbestückungen auf der Standardplatine möglich (auch schon bei kleineren Stückzahlen). Der Anschluss einer speziellen Tastatur mit Tastenmatrix, Piep, LEDs und Schlüsselschalter kann durch Bestückungsvarianten der Steckverbinder gezielt angepasst werden.

3.4.2 Kundenspezifische Layouts

Ein grösserer Schritt ist das kundenspezifische Layout einer Controllerplatine. Das lohnt sich normalerweise allerdings erst bei grösseren Stückzahlen, kann aber zur Anpassung an Gehäuseformen oder zur Einsparung von kostspieligem Verdrahtungsaufwand durchaus angezeigt sein.

3.4.3 Integration des Controllers in eine Tastaturplatine

Sinnvoll ist manchmal die Schaltung des Controllers auf die Hardwareplatine des Anwenders oder direkt in die Tastaturplatine zu integrieren, was durch die verwendete SMD-Technologie auch für Flacheingabetastaturen -- (auf einer Seite die Tastenelemente, auf der Leiterplatte integrierte Schalter mit Metalldom, auf der Rückseite die SMD-Elektronik des Controllers) -- möglich ist und zu einem sehr kompaktem Aufbau führt. GeBE ist in diesen Fällen auch zur Vergabe von Lizenzen für die Benützung des Hardware- und Softwaredesigns bereit.

3.4.4 Erweiterungen des Systems, spezielle Eigenschaften

Der verwendete µ-Controller ist sehr leistungsfähig und kann durchaus auch noch andere Aufgaben, wie z.B. den Anschluß von Kartenlesern, Scannern, Mausfunktionen usw. mit übernehmen. Die vorgesehene SPI-Schnittstelle erlaubt zusätzliche Erweiterungen z.B. den Betrieb einfacher alphanummerischer Displays, von Kleindruckern oder auch sonstiger Steuerungen. Dieser mögliche Weg lohnt sich allerdings nur, wenn Serien dahinter stehen.

3.4.5 Batteriebetrieb

Zu erwähnen sind auch die bereits vorhandenen besonderen Eigenschaften des Systems:
Die integrierte serielle Schnittstelle kann in einem Sonderlayout auch für Infrarotübertragung ausgebaut werden. Mit der bereits vorhandenen Power-Save-Schaltung kann so schnurloser, portabler Betrieb einer batteriebetriebenen IR-Tastatur erfolgen. Die Tastatur würde sich dabei automatisch in den Power-Down-Mode begeben und aus diesem durch Aktivität an den Schnittstellen oder durch Betätigen einer beliebigen Taste aufwecken lassen.

3.4.6 Programmierung bereits in der Fertigung

Die Anpassung an verschiedene Tastatur-Matrixen erfolgt im Controller durch Abspeichern der -Scann-Tabellen im EEPROM (Teach-In manuell oder durch Download (automatisch)). Programmierungen zur Anpassung an kundenspezifische Tastanmatrixen werden bei Serienfertigung bereits in der Produktion von GeBE durchgeführt. Dabei kann, falls später diese Programmierung nicht mehr geändert werden soll, auf die freie Programmierbarkeit verzichtet und das serielle EEPROM eingespart werden. Nur lohnend bei grösseren Serien.

3.4.7 Sonderprogrammierungen

Bei geplanten Anwendungen in größeren Stückzahlen kann es durchaus lohnend sein, spezielle Kontrollereigenschaften programmieren zu lassen.

3.4.8 Beispiel: Codeschloss

Als Bespiel sei hier die Programmierung eines frei programmierbaren Codeschlosses erwähnt, bei dem über die Eingabe an einer 10ner Tastatur 8 verschiedene Erkennungscodes (bis zu 6-stellige Zahlen) eingegeben werden können. Wird eine der Codezahlen eingegeben, so wird der zugehörige Ausgangsport freigeschaltet, ist die Eingabe falsch, so verweigert das Codeschloss die Freigabe. Über eine ID-Kennung kann der Service (auch von einem übergeordneten System über die Schnittstelle) in die Programmierung eingreifen und den einzelnen Ports andere Code-zahlen zuweisen.

3.5 HINWEIS AUF ANDERE PRODUKTE

3.5.1 GCK-972-PS/2

Frei programmierbarer Standardtasturcontroller mit PS/2 Schnittstelle voll PC-Kompatibel..
In der Tastaturcontroller Serie GCK-972 ist auf der gleichen Platine ein frei programmierbarer Tastaturcontroller realisiert, der über eine PS/2 Schnittstelle PC-Kompatible Codes ausgibt. Dieser Controller ist in einem separaten Manual MAN-D-402 oder MAN-E-403 beschrieben.

4 Anhang

4.1 LIEFERFORMEN DES CONTROLLERS GCK-972-RS232

Art. Nr.	Artikelbezeichnung	Eigenschaften	Lieferfähigkeit
	GCK-972-RS232-STD	Tastaturkontroller mit serieller V.24 Schnittstelle freiprogramierbar, Matrix max. 8x16 Tasten in 8 Ebenen (UNSHIFT, SHIFT, CONTROL, CONTROL+ SHIFT, BLOCK+UNSHIFT, BLOCK+SHIFT, BLOCK+CONTROL, BLOCK+CONTROL+SHIFT) variables 99-poliges Matrix-Steckerfeld Strings bis 30 Zeichen Länge, Full-N-Key-Rollover, Schlüssel-schalteranschluß, 4 LED-Treiber, Piepser, Watchdog, Betriebsspannung +5V, ca. 12 mA.	Kleine mengen ab Lager
	GCK-972-RS232-EVAL	Tastaturkontroller, voll bestückt, Schnittstellen: RS232-Seriell-V.24, in EEPROM freiprogramierbar, Tastenmatrix max. 8x16 zwei Ebenen, variables 99-pol. Matrix- Steckerfeld, Ausgabe von ASCII-Zeichen, Strings bis 30 Zeichen, Full-N-Key-Rollover, Schlüsselschalter, 4 LED- Treiber, Piepser, Power-Down, Watchdog, Betriebsspannung +5V, ca.1 2 mA.	Einzeln ab Lager
	GCK-972-RS232-START	Starterkit mit freiprogrammierbaren Tastaturkontroller GCK-972-RS232-EVAL, kompletter Kabelsatz einschließlich Bedienkonsole, Experimentier-Folientastatur 4x4-Matrix , ausführliche Dokumentation	Einzeln ab Lager

4.2 STARTERKIT GCK-972-RS232-START

Um die Möglichkeiten der freien Programmierbarkeit besser einschätzen und üben zu können, hat GeBE für Interessenten ein Starterkit herausgebracht, mit dem der Programmiervorgang (Teach-In) mit einem an die serielle COM-Schnittstelle eines PCs angeschlossenen Controller GCK-972-RS232-EVAL durchgeführt werden kann. Steht ein zweiter Controller zur Verfügung, kann auch der Down-Load Vorgang als Kopieren der Mastertastatur erprobt werden.
Wichtig ist dabei, dass ohne große Beschaffungsaktion alle richtigen Anschlusskabel, eine Testtastatur und eine passende Konsole zum Betrieb zur Verfügung stehen. Es muss lediglich noch für die +5V Stromversorgung gesorgt werden.
Der mitgelieferte Controller GCK-972-RS232-EVAL ist voll bestückt und kann daher auch in der Entwicklung mit entsprechenden Sonderprogrammen verwendet werden. Das Board ist ausserdem in der gleichen Bestückung , allerdings mit einem anderen Programm versehen, als GCK-972-PS/2-EVAL für die Entwicklung von Tastaturen mit einer PC-Kompatiblen Schnittstelle benutzbar.

4.2.1 Lieferumfang Starterkit

Siehe auch Blockschaltbild über Down-Load -Konfiguration Seite REF! oben:

Pos	Art	Bestellbezeichnung	Bemerkung
1	Controller	GCK-972-RS232-EVAL	
2			Standard-PC wird als Terminal seriell über eine COM-Schnittstelle zum Teach-In-Vorgang angeschlossen.
3	Konsole	Bedienkonsole	Platine mit Programmauswahltaster (Teach-In-Taste) und 3 LEDs
4	Kabel 1	GKA-398	6poliges Flachbandkabel MICA zwischen Konsole und Kontroller, 140 mm lang
5	Kabel 3	GKA-396	Kabel, Einzeladern, 2 polig 400mm lang, 5polige Buchseleiste (Panduit) auf 2 Bananenstecker (rot0+5V, schwarz=GND) Kabel für die externe Stromversorgung
6	Kabel 4	GKA-080	Anschlusskabel J2 (9pol. MICA Stecker für seriellen Anschluß PC.
7	Kabel 5	GKA-390	Flachbandkabel, 10 polig, 400mm lang, beidseitig MICA, Kabel gekreuzt, als Verbindung zwischen den beiden seriellen Schnittstellen (jeweils J2) beim Down-Load Master kopieren nach Slave.
8	Experimentiertastatur	PKM-1616, Folientastatur zur Selbstbeschriftung	Folientastur mit 16 Tasten, 4x4 Matrix, Steckerbelegungsform: RRRR CCCC. In Tasche kann Beschriftungsfeld eingelegt werden.

4.2.2 Bei Aufbau des Starterkits bitte beachten

4.2.2.1 Stromversorgung

Für die ersten Versuche lässt sich das Starterkit entsprechend der Abbildung auf Seite REF!

4.2.2.2 Experimentiertastatur

Die mit 16 Tasten ausgerüstete Tastatur kann mit einem eingeschobenen Blatt beschriftet werden. Der 8polige Stecker wird in der dritten Pfostenreihe auf dem Stecker J6 an die PINs J6/39 bis J6/60 angeschlossen (Bitte sorgfältig die Lage der Buchse 1, die mit einem kleinen Pfeil gekennzeichnet ist, beachten. PIN 1 der Buchsenleiste an der Tastatur PKM-1616 wird mit PIN J6/60 verbunden. Damit ist die Tastatur an die Reihen R0, bis R3 und die Spalten C0 bis C3 der Tastaturmatrix angeschlossen.

4.2.2.3 Konsole

Die Konsole trägt die LEDs und den Taster zur Anwahl der Programme (Teach-In-Taste).
Der RESET-Taster befindet sich auf dem Controllerboard.

4.2.2.4 Programmiertastatur (Terminal)

An den 10poligen MICA-Stecker der seriellen Schnittstelle wird mit gekreuzten Datenleitungen ein PC mit Terminalprogramm über dessen COM-Schnittstelle angeschlossen.
Nun kann gestartet werden. Die Betriebsanleitung befindet sich auf Seite 26 , Manuelles Teach-In Verfahren. Dort ist auch ein Blockbild mit dem Aufbau.