

# embedded 5,7" LCD-DISPLAY 320x240 MIT INTELLIGENZ

Stand 08.2011

**WELTNEUHEIT !**



Abmessung:  
138x105x10mm

## TECHNISCHE DATEN

- \* LCD-GRAFIKDISPLAY MIT GRAFIKFUNKTIONEN
- \* 8 EINGEBAUTE FONTS
- \* FONT ZOOM VON ca. 2mm BIS ZU ca. 80mm, in 90° SCHRITTEN DREHBAR
- \* 3 VERSCHIEDENE INTERFACE ONBOARD: RS-232, I<sup>2</sup>C-BUS ODER SPI-BUS
- \* 320x240 PIXEL MIT LED-BELEUCHTUNG BLAU NEGATIV ODER
- \* SCHWARZ-WEISS POSITIV, FSTN-TECHNIK BZW. AMBER
- \* VERSORGUNG +5V @ typ. 50mA / 230mA (OHNE / MIT LED BELEUCHTUNG)
- \* **PIXELGENAU** POSITIONIERUNG BEI ALLEN FUNKTIONEN
- \* GERADE, PUNKT, BEREICH, UND/ODER/EXOR, BARGRAPH...
- \* CLIPBOARD FUNKTIONEN, PULL-DOWN MENÜS
- \* BIS ZU 16 PAGES á 256 BILDER INTERN SPEICHERBAR
- \* BIS ZU 16 PAGES á 768 MAKROS PROGRAMMIERBAR (80kB ONBOARD FLASH)
- \* TEXT UND GRAFIK MISCHEN, BLINKATTRIBUTE: EIN/AUS/INVERS BLINKEN
- \* BELEUCHTUNG PER SOFTWARE REGELBAR
- \* ANALOGESTOUCH PANEL: VARIABLES RASTER
- \* FREI DEFINIERBARE TASTEN UND SCHALTER
- \* POWER-DOWN-MODE (TYP. 150µA) MIT WAKEUP PER TOUCH

## BESTELLBEZEICHNUNG

320x240 DOTS, WEISSE LED-BELEUCHTUNG, BLAU NEGATIV  
WIE VOR, JEDOCH MIT TOUCH PANEL

**EA eDIP320B-8LW**  
**EA eDIP320B-8LWTP**

320x240 DOTS, WEISSE LED-BELEUCHTUNG, POSITIV MODE, FSTN  
WIE VOR, JEDOCH MIT TOUCH PANEL

**EA eDIP320J-8LW**  
**EA eDIP320J-8LWTP**

320x240 DOTS, AMBER LED-BELEUCHTUNG, POSITIV MODE, FSTN  
WIE VOR, JEDOCH MIT TOUCH PANEL

**EA eDIP320J-8LA**  
**EA eDIP320J-8LATP**

EINBLAUBLENDE SCHWARZ, ELOXIERTES ALUMINIUM  
PROGRAMMIER FÜR USB INKL. KABEL, CD FÜR WIN98/ME/2000/XP  
STARTERKIT BLAU, (1x EA eDIP320B-8LWTP + EA9778-1USB)  
STARTERKIT S/W, (1x EA eDIP320J-8LWTP + EA9778-1USB)

**EA 0FP321-8SW**  
**EA 9778-1USB**  
**EA STARTeDIP320B**  
**EA STARTeDIP320J**

**ELECTRONIC  
ASSEMBLY**

making things easy

Documentation of revision				
Date	Type	Old	New	Reason / Description
9.11.2006	V1.0			1st. edition
3.4.2007	V1.1	bug fix: - corrupted character chain - bargraph return code fixed - single picture for touch keys		
9.6.2011	V1.2		changed specification of pull-up resistor (RESET pin)	Changed specification, because of product change notification (SC112002) of ATMEL.

## INHALT

ALLGEMEINES .....	3
ELEKTRISCHE SPEZIFIKATIONEN .....	4
RS-232 .....	5
SPI .....	6
I <sup>2</sup> C .....	7
SOFTWARE PROTOKOLL .....	8 - 9
TOUCH PANEL .....	10
TERMINAL BETRIEB .....	11
ZEICHENSÄTZE .....	12-13
BEFEHLE / FUNKTIONEN INTABELLENFORM .....	14 - 16
RÜCKANTWORTEN DES BEDIENPANELS .....	17
BEFEHLSÜBERGABE / PARAMETER .....	17
TOPVIEW, POWERDOWN .....	18
MAKROPROGRAMMIERUNG .....	19 - 21
MEHRSPRACHIGKEIT, MAKROPAGES .....	21
USB-PROGRAMMIERBOARD .....	22
ABMESSUNGEN .....	23 - 24

## **ALLGEMEINES**

Die EA eDIP-Serie sind die weltweit ersten Displays mit integrierter Intelligenz ! Neben diversen eingebauten Schriften welche pixelgenau verwendet werden können, bieten sie zudem eine ganze Reihe ausgefeilter Grafikfunktionen.

Die Displays sind mit 5V sofort betriebsbereit. Die Ansteuerung erfolgt über eine der 3 eingebauten Schnittstellen RS-232, SPI oder I<sup>2</sup>C. "Programmiert" werden die Displays über hochsprachenähnliche Grafikbefehle; die zeitraubende Programmierung von Zeichensätzen und Grafikroutinen entfällt hier völlig. Die simple Verwendung dieses Displays samt Touchpanel verkürzt die Entwicklungszeit drastisch.

## **HARDWARE**

Das Display ist für +5V Betriebsspannung ausgelegt. Die Datenübertragung erfolgt entweder seriell asynchron im RS-232 Format oder synchron via SPI oder I<sup>2</sup>C Spezifikation. Zur Erhöhung der Datensicherheit wird für alle Übertragungsvarianten ein einfaches Protokoll verwendet.

## **ANALOGESTOUCH PANEL**

Optional sind alle Versionen mit einem integrierten Touch Panel ausgerüstet. Durch Berühren des Displays können hier Eingaben gemacht und Einstellungen per Menü oder Bargraphs getätigt werden. Die Beschriftung der "Tasten" ist flexibel und auch während der Laufzeit änderbar (verschiedene Sprachen, Icons). Das Zeichnen der einzelnen "Tasten", sowie das Beschriften wird von der eingebauten Software komplett übernommen.

## **LED-BELEUCHTUNG; BLAU,WEISS, AMBER**

Alle Displays sind mit einer modernen und stromsparenden LED-Beleuchtung ausgestattet. Die Helligkeit kann per Befehl von 0~100% variiert werden. Während das Schwarz-Weiß-Display (J-LW) und das Amberfarbige (J-LA) auch mit komplett abgeschalteter Beleuchtung noch ablesbar sind, benötigt das blau-weiße Display (B-LW) zum Ablesen in jedem Fall eine minimale Beleuchtung.

Für den Betrieb im direkten Sonnenlicht empfehlen wir die Schwarz-Weiß oder Amber-Version. Für alle anderen Einsatzfälle empfehlen wir die sehr kontraststarke Version in blau-weiß.

Im 24h Betrieb sollte zur Erhöhung der Lebensdauer die Beleuchtung der J-LW und B-LW Typen die Beleuchtung sooft als möglich gedimmt bzw. abgeschaltet werden. Die Version in Amber (J-LA) ist auch bei 100% Beleuchtung für den Dauereinsatz geeignet (MTBF 100,000 Stunden).

## **SOFTWARE**

Die Programmierung erfolgt über Befehle wie z.B. *Zeichne Rechteck von 0,0 nach 64,15*. Es ist keine zusätzliche Software oder Treiber erforderlich. Zeichenketten und Bilder lassen sich **pixelgenau** platzieren. Blinkattribute können beliebig oft vergeben werden. Das Mischen von Text und Grafik ist jederzeit möglich. Es können bis zu 32 Zeichensätze verwendet werden. Jeder Zeichensatz und die Bilder können wiederum 2- bis 8-fach gezoomt und in 90° Schritten gedreht werden. Mit dem größten Zeichensatz lassen sich somit bildschirmfüllende Worte und Zahlen darstellen.

## **ZUBEHÖR**

### Programmer für internes DatenFlash

Das Display wird fertig programmiert mit allen Fonts ausgeliefert. In der Regel ist also der zusätzlich Programmer nicht erforderlich !

Sollen jedoch die internen Zeichensätze geändert oder erweitert werden, oder sollen intern Bilder oder Makros abgelegt werden, brennt der als Zubehör erhältliche USB-Programmer EA 9778-1USB die von Ihnen erstellten Daten/Bilder dauerhaft ins on-board DatenFlash (80kB).

Der Programmer läuft unter Windows und wird an die USB Schnittstelle des PC angeschlossen. Ein Schnittstellenkabel und die Installationssoftware sind im Lieferumfang des Programmers enthalten.

## SPEZIFIKATION UND GRENZWERTE

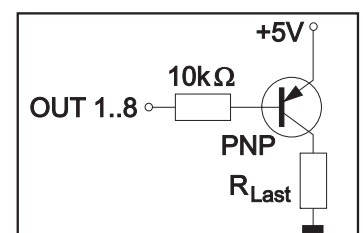
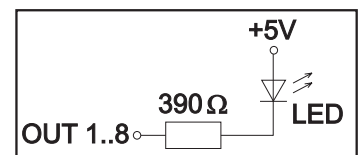
Characteristics					
Value	Condition	min.	typ.	max.	Unit
Operating Temperature		-20		+70	°C
Storage Temperature		-30		+80	°C
Storage Humidity	< 40°C			90	%RH
Operating Voltage		4.5	5.0	5.5	V
Input Low Voltage		-0.5		0.2*VDD	V
Input High Voltage	Pin Reset only	0.9*VDD		VDD+0.5	V
Input High Voltage	except Reset	0.6*VDD		VDD+0.5	V
Input Leakage Current	Pin MOSI only			1	uA
Input Pull-up Resistor		20		50	kOhms
Reset Pull-up Resistor		65		85	kOhms
Output Low Voltage				0.7	V
Output High Voltage		4.0			V
Output Current				20	mA
Power Supply	White Backlight 100%		230		mA
	Amber Backlight 100%		190		mA
	Backlight off		50		mA
	Powerdown (see page 18)	5	150		µA

## AUSGÄNGE

Das EA eDIP320 bietet bis zu 8 Ausgänge, welche z.B. zum Ansteuern von LEDs genutzt werden können. Je nach gewähltem Interface RS232, SPI oder I<sup>2</sup>C werden dazu die nicht benötigten Konfigurationsspins genutzt. Die Konfigurationsspins (Opendrain mit internem Pullup) werden dann als 1=HIGH-Pegel gewertet.

Jeder Ausgang kann per Befehl 'ESC YW n1 n2' individuell angesteuert werden. Strom kann nur bei L-Pegel fließen (Opendrain mit internem Pullup). Jeder Ausgang kann max. 10mA liefern. Es ist somit möglich, mit einem Ausgang direkt eine LED zu schalten. Größere Ströme können durch Verwendung eines externen Transistors geschaltet werden.

Zuordnung Ausgang <-> Pin Nr. je nach Interface						
Ausgang Nr.	RS232/RS422		SPI		I <sup>2</sup> C	
	Pin Nr.	Symbol	Pin Nr.	Symbol	Pin Nr.	Symbol
OUT1	6	BAUD0	10	DORD	6	BA0
OUT2	7	BAUD1	12	OUT2	7	BA1
OUT3	8	BAUD2	13	WUP	8	SA0
OUT4	9	ADR0	14	CPOL	9	SA1
OUT5	13	WUP	15	CPHA	10	SA2
OUT6	14	ADR1	17	DPROT	11	BA2
OUT7	15	ADR2			13	WUP
OUT8	17	DPROT			17	DPROT



## RS-232INTERFACE

Wird das Display wie unten gezeigt beschaltet, so ist das RS-232 Interface ausgewählt. Die Pinbelegung ist in der Tabelle rechts angegeben.

Die Leitungen RxD und TxD führen 5V CMOS-Pegel zur direkten Anbindung an z.B. einen Mikrokontroller.

Wenn "echte" RS-232 Pegel erwünscht sind (z.B. zur Anbindung an einen PC) ist ein externer Pegelwandler wie z.B. MAX232 erforderlich.

### Hinweis:

Die Pins BAUD0..2, ADR0..2, WUP, DPR0T und TEST/SBUF haben einen internen Pull-UP, deshalb ist nur der LO-Pegel (0=GND) aktiv anzulegen. Für Hi-Pegel sind diese Pins offen zu lassen.

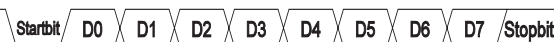
Für RS232 Betrieb (ohne Adressierung) sind die Pins ADR0..ADR2 offen zu lassen.

Am Pin 20 (SBUF) zeigt das Display mit einem low-Pegel, dass im internen Sendepuffer Daten zur Abholung bereit stehen. Diese Leitung kann z.B. mit einem Interrupteingang des Host Systems verbunden werden.

Pinout eDIP320-8: RS-232/RS-422 mode						
Pin	Symbol	In/Out	Function	Pin	Symbol	Function
1	GND	-	Ground Potential for logic (0V)	25	N.C.	not connected
2	VDD	-	Power supply for logic (+5V)	26	N.C.	not connected
3	VADJ	In	Operating voltage for LC driving (input)	27	N.C.	not connected
4	VOUT	Out	Output voltage for LC driving	28	N.C.	not connected
5	RESET	-	L: Reset	29	N.C.	not connected
6	BAUD0	In	Baud Rate 0	30	N.C.	not connected
7	BAUD1	In	Baud Rate 1	31	N.C.	not connected
8	BAUD2	In	Baud Rate 2	32	N.C.	not connected
9	ADR0	In	Address 0 for RS-485	33	N.C.	not connected
10	RxD	In	Receive Data	34	N.C.	not connected
11	TxD	Out	Transmit Data	35	N.C.	not connected
12	EN485	Out	Transmit Enable for RS-485 driver	36	N.C.	not connected
13	WUP	In	L: (Power-On) disable Power-On-Macro L: Wakeup from Powerdownmode	37	N.C.	not connected
14	ADR1	In	Address 1 for RS-485	38	N.C.	not connected
15	ADR2	In	Address 2 for RS-485	39	N.C.	not connected
16	BUZZ	Out	Buzzer output	40	N.C.	not connected
17	DPROT	In	L: Disable Smallprotokoll do not connect for normal operation	41	N.C.	not connected
18	DPWR	Out	L: Normal Operation H: Powerdownmode	42	N.C.	not connected
19	WP	In	L: Writeprotect for DataFlash	43	N.C.	not connected
20	TEST SBUF	IN Out	open-drain with internal pullup 20..50k IN (Power-On) L: Testmode OUT L: data in sendbuffer	44	N.C.	not connected
21	PDI		internal use, do not connect	45	N.C.	not connected
22	PDO		internal use, do not connect	46	N.C.	not connected
23	N.C.		do not connect, reserved	47	N.C.	not connected
24	N.C.		do not connect, reserved	48	N.C.	not connected

## BAUDRATEN

Die Baudrate wird über die Pins 6, 7 und 8 (Baud0..2).eingestellt. Das Datenformat ist fest eingestellt auf 8 Datenbits, 1 Stopbit, keine Parität. Handshakeleitungen RTS/CTS sind nicht erforderlich. Die notwendige Steuerung wird von dem eingebauten Software-Protokoll übernommen (siehe Seiten 8 und 9).



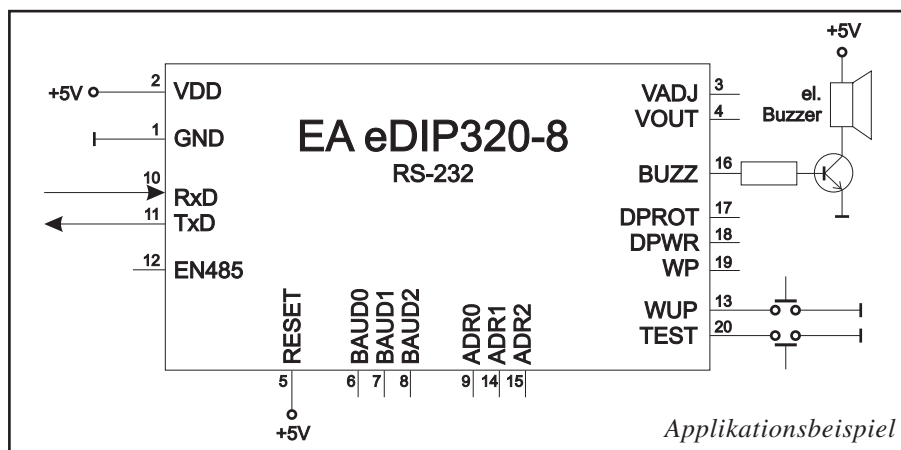
Baudraten			
Baud0	Baud1	Baud2	Datenformat 8,N,1
0	0	0	1200
1	0	0	2400
0	1	0	4800
1	1	0	9600
0	0	1	19200
1	0	1	38400
0	1	1	57600
1	1	1	115200

## RS-485INTERFACE

Mit einem externen Umsetzer (z.B. SN75176) kann das EA eDIP320 an einen 2-Draht RS-485 Bus angeschlossen werden. Somit können grosse Entfernungen bis zu 1200m (Ferndisplay) realisiert werden. Betrieb von mehreren EA eDIP320 an einem RS-485 Bus durch Einstellen von Adressen. Wir empfehlen zur Entwicklung das Board EA 9778-1RS485.

### Adressierung:

- Bis zu acht Hardware-Adressen (0..7) per Pins ADR0..ADR2 einstellbar
- Das eDIP mit Adresse 7 ist nach PowerOn selektiert und Empfangsbereit
- Die eDIPs mit Adresse 0..6 sind nach PowerOn deselektiert
- Bis zu 246 weitere Software-Adressen per Befehl '#KA adr' im PowerOnMakro einstellbar (eDIP extern auf Adresse 0 setzen)



Applikationsbeispiel

## SPI INTERFACE

Wird das Display wie unten gezeigt beschaltet, ist der SPI-Mode aktiviert. Die Datenübertragung erfolgt dann über die serielle synchrone SPI-Schnittstelle. Mit den Pins DORD, CPOL, CPHA werden die Hardwarebedingungen an den Master angepasst.

### Hinweis:

Die Pins DORD, CPOL, CPHA, DPWR und TEST/SBUF haben einen internen Pull-UP, deshalb ist nur der LO-Pegel (0=GND) aktiv anzulegen. Für Hi-Pegel sind diese Pins offen zu lassen.

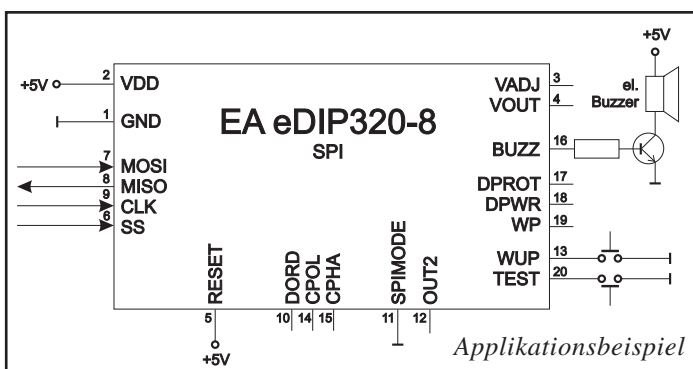
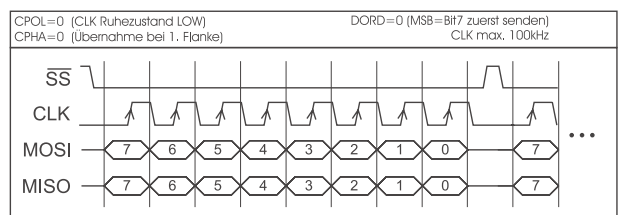
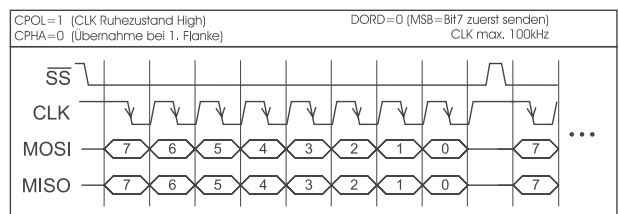
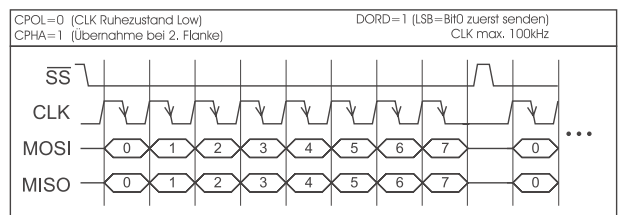
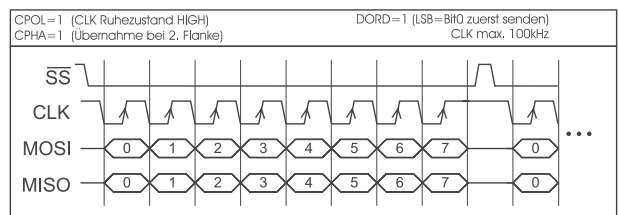
Am Pin 20 (SBUF) zeigt das Display mit einem low-Pegel, dass im internen Sendepuffer Daten zur Abholung bereit stehen. Diese Leitung kann z.B. mit einem Interrupteingang des Host Systems verbunden werden.

Pinout eDIP320-8: SPI mode						
Pin	Symbol	In/Out	Function	Pin	Symbol	Function
1	GND	-	Ground Potential for logic (0V)	25	N.C.	not connected
2	VDD	-	Power supply for logic (+5V)	26	N.C.	not connected
3	VADJ	In	Operating voltage for LC driving (input)	27	N.C.	not connected
4	VOUT	Out	Output voltage for LC driving	28	N.C.	not connected
5	RESET	-	L: Reset	29	N.C.	not connected
6	SS	In	Slave Select	30	N.C.	not connected
7	MOSI	In	Serial In	31	N.C.	not connected
8	MISO	Out	Serial Out	32	N.C.	not connected
9	CLK	In	Shift Clock	33	N.C.	not connected
10	DORD	In	Data Order (0=MSB first; 1=LSB first)	34	N.C.	not connected
11	SPIMODE	In	connect to GND for SPI interface	35	N.C.	not connected
12	OUT2	Out	open-drain with internal pullup 20..50k	36	N.C.	not connected
13	WUP	In	L: (Power-On) disable Power-On-Macro L: Wakeup from Powerdownmode	37	N.C.	not connected
14	CPOL	In	Clock Polarity (0=LO 1=HI when idle)	38	N.C.	not connected
15	CPHA	In	Clock Phase (sampled on 0=1st 1=2nd edge)	39	N.C.	not connected
16	BUZZ	Out	Buzzer output	40	N.C.	not connected
17	DPROT	In	L: Disable Smallprotokoll do not connect for normal operation	41	N.C.	not connected
18	DPWR	Out	L: Normal Operation H: Powerdownmode	42	N.C.	not connected
19	WP	In	L: Writeprotect for DataFlash	43	N.C.	not connected
20	TEST SBUF	IN Out	open-drain with internal pullup 20..50k IN (Power-On) L: Testmode OUT L: data in sendbuffer	44	N.C.	not connected
21	PDI		internal use, do not connect	45	N.C.	not connected
22	PDO		internal use, do not connect	46	N.C.	not connected
23	N.C.		do not connect, reserved	47	N.C.	not connected
24	N.C.		do not connect, reserved	48	N.C.	not connected

## DATENÜBERTRAGUNG SPI

Eine Datenübertragung zum eDIP320 ist bis zu 100 kHz Nonstop möglich. Wenn jedoch zwischen den einzelnen Bytes während der Übertragung Pausen von jeweils min. 100 µs eingehalten werden, kann ein Byte mit bis zu 3 MHz übertragen werden.

Um Daten vom eDIP320 zu Lesen (z.B. das ACK-Byte) muss ein Dummy-Byte (z.B. 0xFF) gesendet werden. Das eDIP320 benötigt eine bestimmte Zeit um die Daten bereit zu stellen; deshalb muss vor jedem zu lesenden Byte mindestens 6µs gewartet werden (keine Aktivität auf der CLK Leitung). Dies gilt auch bei 100kHz Übertragung.



## I<sup>2</sup>C-BUSINTERFACE

Eine Beschaltung des Displays wie unten abgebildet, ermöglicht den direkten Betrieb an einem I<sup>2</sup>C-Bus.

Am Display kann zwischen 8 unterschiedlichen Basisadressen und 8 Slave-Adressen ausgewählt werden.

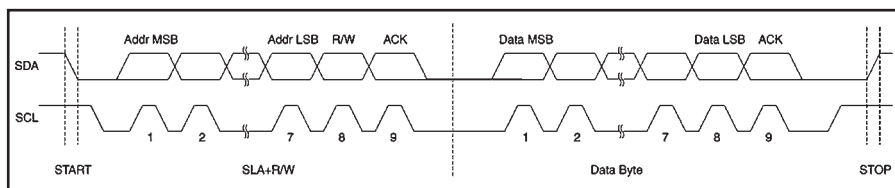
Eine Datenübertragung ist bis zu 100 kHz möglich. Wenn jedoch zwischen den einzelnen Bytes während der Übertragung Pausen von jeweils min. 100 µs eingehalten werden, kann ein Byte mit bis zu 400 kHz übertragen werden.

Pinout eDIP320-8: I2C mode						
Pin	Symbol	In/Out	Function	Pin	Symbol	Function
1	GND	-	Ground Potential for logic (0V)	25	N.C.	not connected
2	VDD	-	Power supply for logic (+5V)	26	N.C.	not connected
3	VADJ	In	Operating voltage for LC driving (input)	27	N.C.	not connected
4	VOUT	Out	Output voltage for LC driving	28	N.C.	not connected
5	RESET	-	L: Reset	29	N.C.	not connected
6	BA0	In	Basic Address 0	30	N.C.	not connected
7	BA1	In	Basic Address 1	31	N.C.	not connected
8	SA0	In	Slave Address 0	32	N.C.	not connected
9	SA1	In	Slave Address 1	33	N.C.	not connected
10	SA2	In	Slave Address 2	34	N.C.	not connected
11	BA2	In	Basic Address 2	35	N.C.	not connected
12	I2CMODE	In	connect to GND for I <sup>2</sup> C interface	36	N.C.	not connected
13	WUP	In	L: (Power-On) disable Power-On-Macro L: Wakeup from Powerdownmode	37	N.C.	not connected
14	SDA	Bidir.	Serial Data Line	38	N.C.	not connected
15	SCL	In	Serial Clock Line	39	N.C.	not connected
16	BUZZ	Out	Buzzer output	40	N.C.	not connected
17	DPROT	In	L: Disable Smallprotokoll do not connect for normal operation	41	N.C.	not connected
18	DPWR	Out	L: Normal Operation H: Powerdownmode	42	N.C.	not connected
19	WP	In	L: Writeprotect for DataFlash	43	N.C.	not connected
20	TEST SBUF	IN Out	open-drain with internal pullup 20..50k IN (Power-On) L: Testmode OUT L: data in sendbuffer	44	N.C.	not connected
21	PDI		internal use, do not connect	45	N.C.	not connected
22	PDO		internal use, do not connect	46	N.C.	not connected
23	N.C.		do not connect, reserved	47	N.C.	not connected
24	N.C.		do not connect, reserved	48	N.C.	not connected

### Hinweis:

Die Pins BA0..2, SA0..2, DPOM, DPROT und TEST/SBUF haben einen internen Pull-Up, deshalb ist nur der LO-Pegel (L=0=GND) aktiv anzulegen. Für Hi-Pegel (H=1) sind diese Pins offen zu lassen.

Am Pin 20 (SBUF) zeigt das Display mit einem LO-Pegel, dass im internen Sendepuffer Daten zur Abholung bereit stehen. Diese Leitung kann z.B. mit einem Interrupteingang des Host Systems verbunden werden.



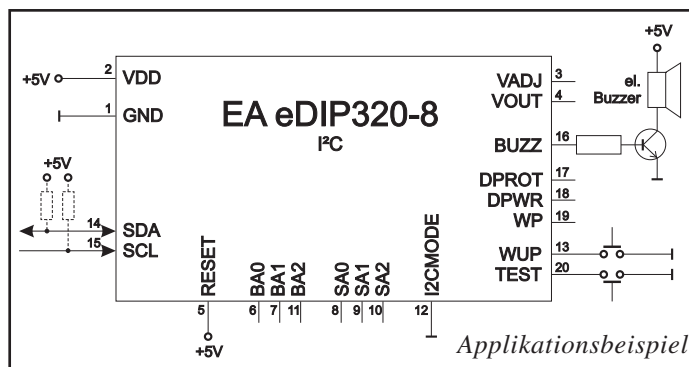
I <sup>2</sup> C - Address												
Pin 11,7,6			Base address		I <sup>2</sup> C address							
BA2	BA1	BA0	address		D7	D6	D5	D4	D3	D2	D1	D0
L	L	L	\$10		0	0	0	1				
L	L	H	\$20		0	0	1	0				
L	H	L	\$30		0	0	1	1				
L	H	H	\$40		0	1	0	0				
H	L	L	\$70		0	1	1	1		S	A	2
H	L	H	\$90		1	0	0	1		S	A	1
H	H	L	\$B0		1	0	1	1		S	A	0
H	H	H	\$D0		1	1	0	1				R

## DATENÜBERTRAGUNG I<sup>2</sup>C-BUS

So funktioniert prinzipiell die Übertragung:

- I<sup>2</sup>C-Start
- Master-Transmit: Display-I<sup>2</sup>C-Adr. (z.B. \$DE), Smallprotokollpaket (Daten) senden
- I<sup>2</sup>C-Stop
- I<sup>2</sup>C-Start
- Master-Read: Display-I<sup>2</sup>C-Adr. (z.B. \$DF), ACK-Byte und evtl. Smallprotokollpaket (Daten) lesen
- I<sup>2</sup>C-Stop

Das Display benötigt eine bestimmte Zeit um die Daten bereit zu stellen; deshalb muss vor jedem zu lesenden Byte mindestens 6µs gewartet werden (keine Aktivität auf der SCL Leitung).



## DATENÜBERTRAGUNGSPROTOKOLL (SMALL PROTOKOLL)

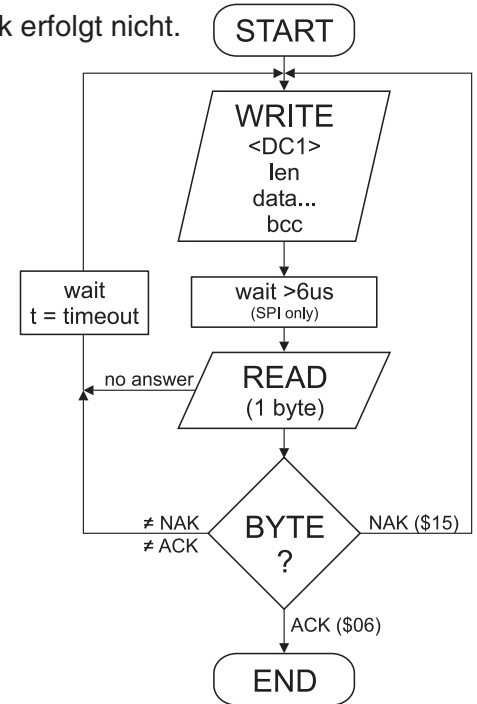
Das Protokoll ist für alle 3 Schnittstellenarten RS-232, SPI und I<sup>2</sup>C identisch aufgebaut. Die Datenübertragung ist jeweils eingebettet in einen festen Rahmen mit Prüfsumme „bcc“. Das EA eDIP320-8 quittiert dieses Paket mit dem Zeichen <ACK> (= \$06) bei erfolgreichem Empfang oder <NAK> (= \$15) bei fehlerhafter Prüfsumme oder Empfangspufferüberlauf. In jedem Fall wird bei <NAK> das komplette Paket verworfen und muss nochmal gesendet werden.

Ein <ACK> bestätigt lediglich die korrekte Übertragung. Ein Syntax-Check erfolgt nicht.

Hinweis: <ACK> muß eingelesen werden.

Empfängt der Hostrechner keine Quittierung, so ist mindestens ein Byte verloren gegangen. In diesem Fall muss die eingestellte Timeoutzeit abgewartet werden, bevor das Paket komplett wiederholt wird.

Die Anzahl (len) der Rohdaten pro Paket kann max. 128 Byte betragen. Befehle die grösser als 128 Byte (z.B. Bild laden ESC UL ...) müssen auf mehrere Pakete aufgeteilt werden. Alle Daten in den Paketen werden nach korrektem Empfang von eDIP wieder zusammengefügt.



## SMALL PROTOKOLL DEAKTIVIEREN

Das Protokoll ist für alle drei Schnittstellen RS-232, I<sup>2</sup>C und SPI identisch. Für Tests kann das Protokoll durch L-Pegel an Pin17(DPROT) abgeschaltet werden. Im normalen Betrieb ist allerdings die Aktivierung des Protokolls unbedingt zu empfehlen. Andernfalls wäre ein möglicher Überlauf des Empfangspuffers nicht zu erkennen.

## DIE PAKETVARIANTEN IN EINZELNEN

### Befehle/Daten zum Display senden

>	<DC1>	len	data...	bcc
<	<ACK>			

<DC1> = 17(dez.) = \$11

<ACK> = 6(dez.) = \$06

len = Anzahl der Nutzdaten in Byte (ohne Prüfsumme, ohne <DC1>)

bcc = 1 Byte = Summe aus allen Bytes inkl. <DC1> und len, Modulo 256

Clear display and draw a line from 0,0 to 319,239

<DC1>	len	ESC	D	L	ESC	G	D	0	0	319	239	bcc	>	
\$11	\$0E	\$1B	\$44	\$4C	\$1B	\$47	\$44	\$00	\$00	\$3F	\$01	\$EF	\$00	\$9F

< <ACK>  
\$06

*Beispiel für ein komplettes Datenpaket*

Eingerahmt von <DC1>, der Anzahl der Daten "len" und der Prüfsumme "bcc" werden die jeweiligen Nutzdaten übertragen. Als Antwort sendet das Display <ACK> zurück.

```

void sendData(unsigned char *buf, unsigned char len)
{
    unsigned char i, bcc;

    SendByte(0x11);           // Send DC1
    bcc = 0x11;

    SendByte(len);           // Send data length
    bcc = bcc + len;

    for(i=0; i < len; i++)   // Send buf
    {
        SendByte(buf[i]);
        bcc = bcc + buf[i];
    }

    SendByte(bcc);           // Send checksum
}
    
```

*C-Beispiel zum Senden eines Datenpaketes*

### Inhalt des Sendepuffers anfordern

>	<DC2>	1	S	bcc
<	<ACK>			
<	<DC1>	len	data...	bcc

<DC2> = 18(dez.) = \$12    1 = 1(dez.) = \$01    S = 83(dez.) = \$53

<ACK> = 6(dez.) = \$06

len = Anzahl der Nutzdaten in Byte (ohne Prüfsumme, ohne <DC1>)

bcc = 1 Byte = Summe aus allen Bytes inkl. <DC1> und len, Modulo 256

Die Befehlsfolge <DC2>, 1, S, bcc entleert den Sendepuffer des Displays. Das Display antwortet zuerst mit der Quittierung <ACK> und beginnt dann alle gesammelten Daten wie z.B. Touchtastendrucke zu senden.



### Pufferinformationen anfordern

>	<DC2>	1	I	bcc	
<	<ACK>				
<	<DC2>	2	send buffer bytes ready	receive buffer bytes free	bcc

<DC2> = 18(dez.) = \$12    I = 1(dez.) = \$01    I = 73(dez.) = \$49

<ACK> = 6(dez.) = \$06

send buffer bytes ready = Anzahl abholbereiter Bytes

receive buffer bytes free = verfügbarer Platz im Empfangspuffer

bcc = 1 Byte = Summe aus allen Bytes inkl. <DC2> Modulo 256

Mit diesem Befehl wird abgefragt, ob Nutzdaten zur Abholung bereit stehen und wie voll der Empfangspuffer des Displays bereits ist.

### Protokolleinstellungen

>	<DC2>	3	D	packet size for send buffer	timeout	bcc
<	<ACK>					

<DC2> = 18(dez.) = \$12    3 = 3(dez.) = \$03    D = 68(dez.) = \$44

packet size for send buffer = 1..128 (Standard: 128)

timeout = 1..255 in 1/100 Sekunden (Standard: 200 = 2 Sekunden)

bcc = 1 Byte = Summe aus allen Bytes inkl. <DC2>, Modulo 256

<ACK> = 6(dez.) = \$06

Hierüber läßt sich die maximale Paketgröße welche das Display senden darf begrenzen. Voreingestellt ist eine Paketgröße mit bis zu 128 Byte Nutzdaten. Weiterhin läßt sich der Timeout in 1/100s einstellen. Der Timeout spricht an, wenn einzelne Bytes verloren gegangen sind. Danach muß das gesamte Paket nochmals übertragen werden.

### Protokollinformationen anfordern

>	<DC2>	1	P	bcc		
<	<ACK>					
<	<DC2>	3	max. packet size	akt. send packet size	akt. timeout	bcc

<DC2> = 18(dez.) = \$12    I = 1(dez.) = \$01    P = 80(dez.) = \$50

<ACK> = 6(dez.) = \$06

max. packet size = maximale Anzahl der Nutzdaten eines Protokollpaketes (eDIP320-8 = 128)

akt. send packet size = eingestellte Paketgröße zum Senden

akt. timeout = eingestellter timeout in 1/100 Sekunden

bcc = 1 Byte = Summe aus allen Bytes inkl. <DC2>, Modulo 256

Mit diesem Befehl werden Protokolleinstellungen abgefragt.

### Letztes Datenpaket wiederholen

>	<DC2>	1	R	bcc
<	<ACK>			
<	<DC1> <DC2>	len	data...	bcc

<DC2> = 18(dez.) = \$12    I = 1(dez.) = \$01    R = 82(dez.) = \$52

<ACK> = 6(dez.) = \$06

<DC1> = 17(dez.) = \$11

len = Anzahl der Nutzdaten in Byte (ohne Prüfsumme, ohne <DC1> bzw. <DC2>)

bcc = 1 Byte = Summe aus allen Bytes inkl. <DC2> und len, Modulo 256

Falls das zuletzt angeforderte Paket eine falsche Prüfsumme enthielt, kann das komplette Paket nochmals angefordert werden. Die Antwort kann dann der Inhalt des Sendepuffers (<DC1>) oder die Puffer-/Protokoll-Information (<DC2>) sein.

### Adressierung nur bei RS232/RS485 Betrieb

>	<DC2>	3	A	select or deselect	adr	bcc
<	<ACK>					

<DC2> = 18(dez.) = \$12    3 = 3(dez.) = \$03    A = 65(dez.) = \$41

select or deselect: 'S' = 83(dez.) = \$53 oder 'D' = 68(dez.) = \$44

adr = 0..255

bcc = 1 Byte = Summe aus allen Bytes inkl. <DC2> und len, Modulo 256

<ACK> = 6(dez.) = \$06

Mit diesem Befehl läßt sich das eDIP mit der Adresse adr Selektieren oder Deselektieren.

## TOUCH PANEL (VERSIONEN EA EDIP320X-8LWTP)

Die Versionen -xxxTP werden mit einem analogen, resistiven Touchpanel geliefert. Bis zu 80 Touchbereiche (Tasten, Schalter, Menüs, Bargrapheingaben), können gleichzeitig definiert werden. Die Felder können pixelgenau definiert werden. Das Display unterstützt die Darstellung mit komfortablen Befehlen (siehe Seite 16). Beim Berühren der Touch-"Tasten" können diese automatisch invertiert werden und ein externer Summer (Pin 16) signalisiert die Berührung. Der zuvor definierte Return-Code der "Taste" wird über die Schnittstelle gesendet oder es wird statt dessen ein internes Touch Makro mit der Nummer des Return-Codes gestartet (siehe Seite 19, *Makroprogrammierung*).

## TOUCHPANELABGLEICH

Das Touchpanel ist bei Auslieferung abgeglichen und sofort einsatzbereit. Durch Alterung und Abnutzung kann es nötig sein, dass das Touchpanel neu abgeglichen werden muss.

### Abgleichprozedur:

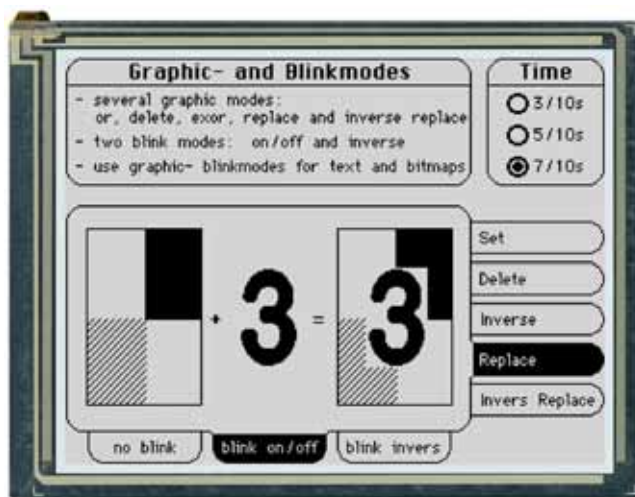
1. Beim Einschalten Touch berühren und gedrückt halten. Nach Erscheinen der Meldung "touch adjustment ?" den Touch wieder loslassen (alternativ den Befehl 'ESC @' senden).
2. Innerhalb 1 Sekunde den Touch nochmals für mindestens 1 Sekunde berühren.
3. Den Anweisungen zum Abgleich folgen (2 Punkte *Linksoben* und *Rechtsunten* betätigen).

## RAHMEN UNDTASTENFORMEN

Mit den Befehlen *Rahmen /Rahmenbox zeichnen* sowie beim Zeichnen von Touchtasten kann ein Rahmentyp eingestellt werden. Es stehen dabei 18 Rahmentypen zur Verfügung (0= keinen Rahmen zeichnen). Die Rahmengröße muß mindestens 16x16 Pixel betragen.

## BITMAPS ALSTASTEN

Ausser den Rahmentypen, die in der Grösse frei skalierbar sind, gibt es noch die Möglichkeit beliebige Bitmaps (jeweils 2 Stück für *nicht-gedrückt* und *gedrückt*) als Touch-Tasten oder -Schalter zu verwenden.



Über die ELECTRONIC ASSEMBLY LCD-Tools<sup>\*)</sup> können eigene Buttons als Bilder eingebunden werden (Compileranweisung "PICTURE"). Ein Button besteht immer aus zwei gleich grossen monochromen Windows-BMPs (ein Bitmap für die normale Darstellung der Touchtaste und ein Bitmap für die gedrückte Touchtaste). Die aktive Fläche der Touchtaste ergibt sich automatisch aus der Grösse der Button-Bitmaps.



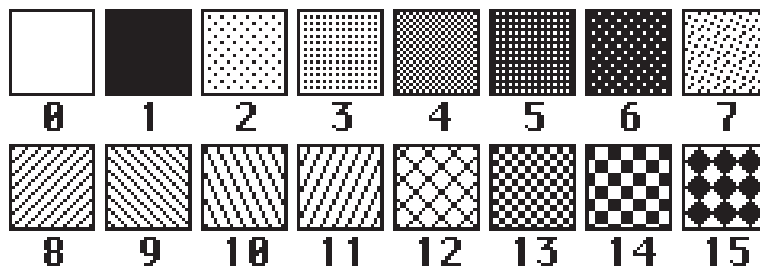
## SCHALTER IN GRUPPEN (RADIO GROUP)

Touch-Schalter ändern ihren Zustand bei jeder Berührung von *EIN* in *AUS* und umgekehrt. Mehrere Touchschalter können zu einer Gruppe zusammengefasst werden (Befehl: 'ESC A R nr'). Wird nun ein Touch-Schalter innerhalb einer Gruppe 'nr' eingeschaltet, dann werden automatisch alle andern Touch-Schalter dieser Gruppe ausgeschaltet. Es ist also automatisch immer nur ein Schalter gesetzt.

<sup>\*)</sup> im Internet unter <http://www.lcd-module.de/deu/touch/touch.htm>

## FÜLLMUSTER

Bei diversen Befehlen kann als Parameter ein Mustertyp eingestellt werden. So können z.B. rechteckige Bereiche und Bargraphs mit unterschiedlichen Mustern gefüllt werden. Dabei stehen 16 interne Füllmuster zur Verfügung.



## TERMINAL-BETRIEB

Nach dem Einschalten blinkt der Cursor in der ersten Zeile und das Display ist empfangsbereit. Alle ankommenden Zeichen werden als ASCII's im Terminal dargestellt (Ausnahme: CR,LF,FF,ESC,'#'). Voraussetzung dafür ist ein funktionierender Portokollrahmen (Seiten 8 und 9) oder ein abgeschaltetes Protokoll.

Der Zeilenvorschub erfolgt automatisch oder durch das Zeichen 'LF'. Ist die letzte Zeile voll, scrollt der Terminalinhalt nach oben. Beim Zeichen 'FF' (Seitenvorschub) wird das Terminal gelöscht.

Das Zeichen '#' wird als Escape-Zeichen benutzt und ist somit nicht direkt im Terminal darstellbar. Soll das Zeichen '#' im Terminal ausgegeben werden, so muß es doppelt gesendet werden '##'. Das Terminal besitzt eine eigene Ebene zur Darstellung und ist somit völlig unabhängig von den Grafikausgaben. Wird z.B. der Grafikbildschirm mit 'ESC DL' gelöscht, so beeinflusst das nicht den Inhalt des Terminalfensters.

Der Terminalfont ist fest im ROM vorhanden und kann auch für Grafikausgaben 'ESC Z...' verwendet werden (FONT nr=0 einstellen).

+ Lower Upper	\$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)		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
\$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$40 (dez: 64)	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
\$50 (dez: 80)	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
\$60 (dez: 96)	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
\$70 (dez: 112)	p	q	r	s	t	u	v	w	x	y	z	{		}	~	Δ
\$80 (dez: 128)	€	ü	é	â	ä	à	ç	ê	ë	è	ï	î	í	ñ	Á	
\$90 (dez: 144)	É	æ	Æ	ö	ö	ò	û	ù	ÿ	ö	ü	ç	£	¥	β	f
\$A0 (dez: 160)	á	í	ó	ú	ñ	ñ	ë	ó	ç	í	í	½	¼	i	«	»
\$B0 (dez: 176)																
\$C0 (dez: 192)																
\$D0 (dez: 208)																
\$E0 (dez: 224)	α	β	Γ	π	Σ	σ	μ	τ	ϕ	θ	η	δ	φ	Φ	Ε	π
\$F0 (dez: 240)	≡	±	≥	≤	ρ	ρ	÷	≈	°	*	.	√	n	z	z	-

Terminal-Font (Font 0): 8x8 monospaced

EA eDIP320-8: Terminalbefehle										nach Reset	
Befehl	Codes		Anmerkung								
Formfeed FF (dez:12)	^L		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)	^J		Cursor 1 Zeile tiefer, falls Cursor in letzter Zeile dann wird gescrollt								
Cursor positionieren	ESC	T	P	s	z					s=Spalte; z=Zeile; Ursprung links oben ist (1,1)	1,1
Cursor On / Off			C	n1					n1=0: Cursor ist unsichtbar; n1=1: Cursor blinkt;	1	
Cursorposition sichern			S					die aktuelle Cursorposition wird gesichert			
Cursorposition restoren			R					die letzte gesicherte Cursorposition wird wieder hergestellt			
Terminal AUS			A					Terminal Anzeige ist ausgeschalten; Ausgaben werden verworfen			
Terminal EIN			E					Terminal Anzeige ist eingeschalten;	Ein		
Version ausgeben			V					Die Versions-Nr. wird im Terminal ausgegeben z.B "EA eDIP320-8 V1.0 Rev.A"			
Fenster definieren	ESC	T	W	s	z	b	h	w	Die Terminal Ausgabe erfolgt nur innerhalb des Fensters ab Spalte s und Zeile z (=linke obere Ecke) mit der Breite b und Höhe h (Angaben in Zeichen) w=Winkel (0=0°; 1=90°; 2=180°; 3=270°) der Terminaldarstellung	1,1 40,30 0	

## INTEGRIERTE UND EXTERNE FONTS

Es sind standardmäßig, außer dem 8x8 Terminalfont (Font-Nr. 0), noch 3 monospaced, 3 proportionale Zeichensätze und 1 grosser Ziffernfont integriert. Die proportionalen Zeichensätze ergeben ein schöneres Schriftbild, gleichzeitig benötigen sie weniger Platz auf dem Bildschirm (z.B. schmales "i" und breites "W"). Jedes Zeichen kann **pixelgenau** platziert werden und in der Höhe und Breite von 1- bis 8-fach vergrößert werden.

Texte lassen sich linksbündig, rechtsbündig und zentriert ausgeben. Eine Drehung in 90° Schritten, z.B. für vertikalen Einbau des Displays, ist möglich. Die Makroprogrammierung erlaubt die Einbindung von weiteren Fonts (max. 31). Es können alle nur erdenklichen Schriften mit einem Texteditor erstellt und über den eDIP320-Compiler<sup>\*)</sup> geladen werden (USB-Programmer EA 9778-1USB notwendig).

+ Lower Upper	\$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)		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
\$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$40 (dez: 64)	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
\$50 (dez: 80)	P	Q	R	S	T	U	U	W	X	Y	Z	[	\	]	^	_
\$60 (dez: 96)	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
\$70 (dez: 112)	p	q	r	s	t	u	v	w	x	y	z	{		}	~	Δ
\$80 (dez: 128)	€	ü	é	â	ä	à	ç	ê	ë	è	ï	î	ï	ñ	ñ	
\$90 (dez: 144)	É	æ	Æ	ô	ö	ò	û	ù	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ
\$A0 (dez: 160)	á	í	ó	ú	ñ	ñ	á	ø	¿	¡	½	¼	i	«	»	
\$B0 (dez: 176)																
\$C0 (dez: 192)																
\$D0 (dez: 208)																
\$E0 (dez: 224)	α	β	Γ	π	Σ	σ	μ	ν	ξ	θ	η	δ	φ	ψ	ε	π
\$F0 (dez: 240)	≡	±	≥	≤	Γ	J	÷	≈	°	•	•	•	•	•	•	•

Font 1: 4x6 monospaced

+ Lower Upper	\$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)		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
\$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$40 (dez: 64)	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
\$50 (dez: 80)	P	Q	R	S	T	U	U	W	X	Y	Z	[	\	]	^	_
\$60 (dez: 96)	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
\$70 (dez: 112)	p	q	r	s	t	u	v	w	x	y	z	{		}	~	Δ
\$80 (dez: 128)	€	ü	é	â	ä	à	ç	ê	ë	è	ï	î	ï	ñ	ñ	
\$90 (dez: 144)	É	æ	Æ	ô	ö	ò	û	ù	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ
\$A0 (dez: 160)	á	í	ó	ú	ñ	ñ	á	ø	¿	¡	½	¼	i	«	»	
\$B0 (dez: 176)																
\$C0 (dez: 192)																
\$D0 (dez: 208)																
\$E0 (dez: 224)	α	β	Γ	π	Σ	σ	μ	ν	ξ	θ	η	δ	φ	ψ	ε	π
\$F0 (dez: 240)	≡	±	≥	≤	Γ	J	÷	≈	°	•	•	•	•	•	•	•

Font 3: 7x12 monospaced

+ Lower Upper	\$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)		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
\$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$40 (dez: 64)	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
\$50 (dez: 80)	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
\$60 (dez: 96)	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
\$70 (dez: 112)	p	q	r	s	t	u	v	w	x	y	z	{		}	~	Δ
\$80 (dez: 128)	€	ü	é	â	ä	à	ç	ê	ë	è	ï	î	ï	ñ	ñ	
\$90 (dez: 144)	É	æ	Æ	ô	ö	ò	û	ù	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ
\$A0 (dez: 160)	á	í	ó	ú	ñ	ñ	á	ø	¿	¡	½	¼	i	«	»	
\$B0 (dez: 176)																
\$C0 (dez: 192)																
\$D0 (dez: 208)																
\$E0 (dez: 224)	α	β	Γ	π	Σ	σ	μ	ν	ξ	θ	η	δ	φ	ψ	ε	π
\$F0 (dez: 240)	≡	±	≥	≤	Γ	J	÷	≈	°	•	•	•	•	•	•	•

Font 2: 6x8 monospaced

+ Lower Upper	\$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)		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
\$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$40 (dez: 64)	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
\$50 (dez: 80)	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
\$60 (dez: 96)	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
\$70 (dez: 112)	p	q	r	s	t	u	v	w	x	y	z	{		}	~	Δ
\$80 (dez: 128)	€	ü	é	â	ä	à	ç	ê	ë	è	ï	î	ï	ñ	ñ	
\$90 (dez: 144)	É	æ	Æ	ô	ö	ò	û	ù	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ
\$A0 (dez: 160)	á	í	ó	ú	ñ	ñ	á	ø	¿	¡	½	¼	i	«	»	
\$B0 (dez: 176)																
\$C0 (dez: 192)																
\$D0 (dez: 208)																
\$E0 (dez: 224)	α	β	Γ	π	Σ	σ	μ	ν	ξ	θ	η	δ	φ	ψ	ε	π
\$F0 (dez: 240)	≡	±	≥	≤	Γ	J	÷	≈	°	•	•	•	•	•	•	•

Font 4: GENEVA10 proportional

+ Lower Upper	\$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)		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	
\$30 (dez: 48)		0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$40 (dez: 64)		@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
\$50 (dez: 80)		P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
\$60 (dez: 96)		`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
\$70 (dez: 112)		p	q	r	s	t	u	v	w	x	y	z	{		}	~	Δ
\$80 (dez: 128)		€	ü	é	â	ä	à	ç	ê	ë	è	ï	î	í	ñ	Å	
\$90 (dez: 144)		É	æ	Æ	ô	ö	ò	û	ù	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ
\$A0 (dez: 160)		á	í	ó	ú	ñ	Ñ	á	ó								
\$B0 (dez: 176)																	
\$C0 (dez: 192)																	
\$D0 (dez: 208)																	
\$E0 (dez: 224)		ß															
\$F0 (dez: 240)									°								

Font 5: CHICAGO14 proportional

+ Lower Upper	\$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)		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	
\$30 (dez: 48)		0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$40 (dez: 64)		@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
\$50 (dez: 80)		P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
\$60 (dez: 96)		`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
\$70 (dez: 112)		p	q	r	s	t	u	v	w	x	y	z	{		}	~	Δ
\$80 (dez: 128)		€	ü	é	â	ä	à	ç	ê	ë	è	ï	î	í	ñ	Å	Å
\$90 (dez: 144)		É	æ	Æ	ô	ö	ò	û	ù	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ
\$A0 (dez: 160)		á	í	ó	ú	ñ	Ñ	á	ó								
\$B0 (dez: 176)																	
\$C0 (dez: 192)																	
\$D0 (dez: 208)																	
\$E0 (dez: 224)		ß															
\$F0 (dez: 240)									°								

Font 6: Swiss30 Bold proportional

+ Lower Upper	\$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)												+	-	.		
\$30 (dez: 48)		0	1	2	3	4	5	6	7	8	9	:				

Font 7: grosse Ziffern BigZif57

## SCHRIFTBILD

Diese Hardcopy zeigt alle im Auslieferungszustand geladenen Schriften.

Die Makroprogrammierung erlaubt das Verändern oder Einbindung von weiteren Fonts. Es können alle nur erdenklichen Schriften (einschl. kyrillisch und chinesisch) mit einem Texteditor erstellt und über den Kitcompiler / LCD-Toolkit<sup>\*)</sup> programmiert werden (Programmer EA 9778-1USB notwendig).



<sup>\*)</sup> im Internet unter <http://www.lcd-module.de/deu/touch/touch.htm>

## ALLE BEFEHLE AUF EINEN BLICK

Die eingebaute Intelligenz erlaubt den Aufbau eines Bildschirms über unten stehende Befehle. Alle Befehle können sowohl über die serielle Schnittstelle (vgl. Seite 17) als auch in selbst-definierten Makros (vgl. Seiten 19/20) verwendet werden.

EA eDIP320-8: Befehlstabelle 1										nach Reset					
Befehl	Codes					Anmerkung									
<b>Befehle zur Ausgabe von Zeichenketten</b>															
Zeichenkette ausgeben L: Linksbündig C: Zentriert R: Rechtsbündig			L	xx1	yy1	Tex ...	NU	Eine Zeichenkette (...) an xx1,yy1 ausgegeben; Zeichenkettenende: 'NUL' (\$00), 'LF' (\$0A) oder 'CR' (\$0D); Mehrere Zeilen werden durch das Zeichen ' ' (\$7C) getrennt; Texte die zwischen zwei '-' (\$7E) Zeichen stehen blinken An/Aus; Texte die zwischen zwei '@' (\$40) Zeichen stehen blinken Invertierend; Das Backslash-Zeichen '\' (\$5C) hebt die Sonderfunktion der Zeichen '~@\' auf; z.B. "name@test.de" => "name@test.de"							
Font einstellen	ESC	Z	F	n1				Font mit der Nummer n1 (0..31) einstellen			0				
Font-Zoomfaktor			Z	n1	n2			n1 = X-Zoomfaktor (1x..8x); n2 = Y-Zoomfaktor (1x..8x)			1,1				
zus. Zeilenabstand			Y	n1				zwischen zwei Zeilen n1 Pixel (0..15) als zusätzlichen Zeilenabstand einfügen							
Text-Winkel			W	n1				Text-Ausgabewinkel: n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°			0				
Text-Verknüpfungsmodus			V	n1				Modus n1: 1=setzen; 2=löschen; 3=iners; 4=Replace; 5=Invers Replace;			4				
Text-Muster			M	n1				Text mit Muster Nr. n1 (0..15) verknüpfen;			1				
Text-Blinkattribut			B	n1				n1: 0=kein blinken; 1=Text blinkt An/Aus; 2=Text blinkt invers			0				
Zeichenkette für Terminal	ESC	Z	T			Text ...		Befehl um eine Zeichenkette aus einem Makro an das Terminal auszugeben							
<b>Geraden und Punkte zeichnen</b>															
Rechteck zeichnen			R	xx1	yy1	xx2	yy2	Vier Geraden als Rechteck von xx1,yy1 nach xx2,yy2 zeichnen							
Gerade zeichnen			D	xx1	yy1	xx2	yy2	Eine Gerade von xx1,yy1 nach xx2,yy2 zeichnen							
Gerade weiter zeichnen			W	xx1	yy1			Eine Gerade vom letzten Endpunkt bis xx1, yy1 zeichnen			0				
Punkt zeichnen			P	xx1	yy1			Ein Punkt an die Koordinaten xx1, yy1 setzen							
Punktgröße / Liniendicke			Z	n1	n2			n1 = X-Punktgröße (1..15); n2 = Y-Punktgröße (1..15);			1,1				
Verknüpfungsmodus			V	n1				Zeichenmodus einstellen n1: 1=setzen; 2=löschen; 3=iners;			1				
Muster			M	n1				Geraden-/Punkt-Muster Nr. n1 (0..15) einstellen			1				
<b>Rechteckige Bereiche verändern / zeichnen</b>															
Bereich löschen			L	xx1	yy1	xx2	yy2	Bereich von xx1,yy1 nach xx2,yy2 löschen (alle Pixel aus)							
Bereich invertieren			I	xx1	yy1	xx2	yy2	Bereich von xx1,yy1 nach xx2,yy2 invertieren (alle Pixel umkehren)							
Bereich füllen			S	xx1	yy1	xx2	yy2	Bereich von xx1,yy1 nach xx2,yy2 füllen (alle Pixel ein)							
Bereich m. Füllmuster	ESC	R	O	xx1	yy1	xx2	yy2	n1	Bereich von xx1,yy1 nach xx2,yy2 mit Muster n1 zeichnen (immer setzen)						
Box zeichnen			M	xx1	yy1	xx2	yy2	n1	Rechteck von xx1,yy1 nach xx2,yy2 mit Muster n1 zeichnen (immer Replace)						
Rahmen zeichnen			R	xx1	yy1	xx2	yy2	n1	Rahmen Typ n1 von xx1,yy1 nach xx2,yy2 zeichnen (immer setzen)						
Rahmenbox zeichnen			T	xx1	yy1	xx2	yy2	n1	Rahmenbox Typ n1 von xx1,yy1 nach xx2,yy2 zeichnen (immer Replace)						
<b>Bitmap Bilder Befehle</b>															
Bild aus Clipboard			C	xx1	yy1			Der akt. Clipboardinhalt wird mit allen Bildattributen nach xx1,yy1 geladen							
internes Bild laden			I	xx1	yy1		nr	internes Bild mit der nr (0..255) aus dem Datenflash nach xx1,yy1 laden							
Bild laden			L	xx1	yy1		BH7 daten ...	Ein Bild nach xx1,yy1 laden; Daten des Bildes siehe Bildaufbau BH7-Format							
Bild-Zoomfaktor			Z	n1	n2			n1 = X-Zoomfaktor (1x..8x); n2 = Y-Zoomfaktor (1x..8x)			1,1				
Bild-Winkel			W	n1				Ausgabewinkel des Bildes: n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°			0				
Bild-Spiegeln			X	n1				n1: 0=Normaldarstellung; 1=Das Bild wird horizontal gespiegelt			0				
Bild-Verknüpfungsmodus			V	n1				Modus n1: 1=setzen; 2=löschen; 3=iners; 4=Replace; 5=Invers Replace;			4				
Bild-Muster			M	n1				Bild mit Muster Nr. n1 (0..15) verknüpfen			1				
Bild-Blinkattribut			B	n1				n1: 0=kein blinken; 1=Bild blinkt An/Aus; 2=Bild blinkt invers; 3=blinkt mit Blinkbild			0				
Hardcopy senden			H	xx1	yy1	xx2	yy2	Nach diesem Befehl wird der Bildausschnitt im BH7-Format gesendet.							
<b>Display-Befehle (Wirkung auf das gesamte Display)</b>															
Display löschen			L					Displayinhalt löschen (alle Pixel aus)							
Display invertieren			I					Displayinhalt invertieren (alle Pixel umkehren)							
Display füllen	ESC	D	S					Displayinhalt füllen (alle Pixel ein)							
Display ausschalten			A					Displayinhalt wird unsichtbar bleibt aber erhalten, Befehle weiterhin möglich							
Display einschalten			E					Displayinhalt wird wieder sichtbar			Ein				
<b>Blinkbereichs-Befehle</b>															
Blinkattribut löschen			L	xx1	yy1	xx2	yy2	Löscht das Blinkattribut von xx1,yy1 bis xx2,yy2							
Invertierend blinken			I	xx1	yy1	xx2	yy2	Definiert einen invertierenden Blinkbereich von xx1,yy1 bis xx2,yy2							
Muster Blinkbereich	ESC	Q	M	xx1	yy1	xx2	yy2	n1	Definiert einen Blinkbereich mit Muster n1 (An/Aus) von xx1,yy1 bis xx2,yy2						
Blinkzeit einstellen			Z	n1				Einstellen der Blinkzeit n1= 1..15 in 1/10s; 0=Blinkfunktion deaktivieren			6				
<b>Bargraph Befehle</b>															
Bargraph definieren			R L O U	n1	xx1	yy1	xx2	yy2	aw	ew	typ	mst	Bar nach L(inks),R(echts),O(ben),U(nten) mit der Nr. n1 (0..255) definieren. xx1,yy1,xx2,yy2 umschließendes Rechteck. aw, ew sind die Werte für 0% und 100%. typ: 0=Balken; 1=Balken im Rechteck; mst=Balkenmuster typ: 2=Strich; 3=Strich im Rechteck; mst=Strichbreite		kein Bar definiert
Bargraph aktualisieren	ESC	B	A	n1	wert			Bar mit der Nummer n1 auf den neuen Benutzer- 'wert' setzen und zeichnen.							
Bargraph neu zeichnen			Z	n1				Den Bargraph mit der Nummer n1 komplett neu zeichnen							
Bargraphwert senden			S	n1				Den aktuellen Wert des Bargraph Nr. n1 senden							
Bargraph löschen			D	n1	n2			Die Definition des Bars mit der Nummer n1 wird ungültig. War der Bargraph als Eingabe mit Touch definiert so wird auch dieses Touchfeld gelöscht. n2=0: Bar weiterhin sichtbar; n2=1: Bar wird gelöscht							

EA eDIP320-8: Befehlstabelle 2										nach	
Befehl	Codes					Anmerkung	Reset				
<b>Clipboard Befehle (Zwischenspeicher für Bildbereiche)</b>											
Displayinhalt sichern	ESC	C	B				Der gesamte Displayinhalt wird als Bildbereich ins Clipboard kopiert				
Bereich sichern			S	xx1	yy1	xx2	yy2	Der Bildbereich von xx1,yy1 bis nach xx2,yy2 wird ins Clipboard kopiert			
Bereich restaurieren			R					Der Bildbereich im Clipboard wird wieder ins Display kopiert			
Bereich kopieren			K	xx1	yy1				Der Bildbereich im Clipboard wird ins Display nach xx1,yy1 kopiert		
<b>Einstellungen für Menübox / Touchmenü</b>											
Menü-Font einstellen	ESC	N	F	n1					Font mit der Nummer n1 (0..31) für Menüdarstellung einstellen	0	
Menüfont-Zoomfaktor			Z	n1	n2					n1 = X-Zoomfaktor (1x..8x); n2 = Y-Zoomfaktor (1x..8x)	1,1
zus. Zeilenabstand			Y	n1					zwischen Menüeinträgen n1 Pixel (0..15) zusätzlichen Zeilenabstand einfügen		
Menü-Winkel			W	n1					Menüdarstellung Winkel: n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°	0	
Touchmenü-Automatik			T	n1					n1=1: Touchmenü öffnet automatisch; n1=0: Touchmenü öffnet nicht automatisch stattdessen wird die Anforderung 'ESC T 0' zum Öffnen an den Hostrechner gesendet, dieser kann dann mit 'ESC N T 2' das Touchmenü öffnen.	1	
<b>Menübox Befehle (Steuerung mit Tasten nicht per Touch)</b>											
Menü definieren und Darstellen	ESC	N	D	xx1	yy1	nr	Text ...	NUL	Ein Menü wird ab der Ecke xx1,yy1 mit dem akt. Menüfont gezeichnet. nr:= aktuell invertierter Eintrag (z.B: 1 = 1. Eintrag) Text:= Zeichenkette mit den Menüeinträgen. Die einzelnen Einträge sind durch Zeichen ' ' (\$7C,dez:124) getrennt z.B. "Eintrag1 Eintrag2 Eintrag3" Der Hintergrund des Menüs wird automatisch gesichert. Ist bereits ein Menü definiert, wird dieses automatisch abgebrochen+entfernt.		
nächster Eintrag			N					Der nächste Eintrag wird invertiert oder bleibt am Ende stehen			
vorheriger Eintrag			P					Der vorherige Eintrag wird invertiert oder bleibt am Anfang stehen			
Menüende / Senden			S					Das Menü wird entfernt und durch den ursprünglichen Hintergrund ersetzt der aktuelle Eintrag wird als Nummer (1..n) gesendet (0=kein Menü dargestellt)			
Menüende / Makro			M	n1					Das Menü wird entfernt und durch den ursprünglichen Hintergrund ersetzt. Für Eintrag 1 wird Menü-Makro n1 aufgerufen, für Eintrag 2 Menü-Makro nr+1 usw.		
Menüende / Abbrechen	A					Das Menü wird entfernt und durch den ursprünglichen Hintergrund ersetzt					
<b>Makro Befehle</b>											
Normal Makro ausführen	ESC	M	N	n1					Das (Normal-)Makro mit der Nummer n1 (0..255) aufrufen (max. 7 Ebenen)		
Touch Makro ausführen			T	n1					Das Touch-Makro mit der Nummer n1 (0..255) aufrufen (max. 7 Ebenen)		
Menü Makro ausführen			M	n1					Das Menü-Makro mit der Nummer n1 (0..255) aufrufen (max. 7 Ebenen)		
Makros sperren			L	typ	n1	n2				Die Makros vom typ = 'N', 'T' oder 'M' (typ = 'A' alle Makrotypen) werden von der Nummer n1 bis n2 gesperrt, d.h. bei Aufruf nicht mehr ausgeführt.	
Makros freigeben			U	typ	n1	n2				Die Makros vom typ = 'N', 'T' oder 'M' (typ = 'A' alle Makrotypen) werden von der Nummer n1 bis n2 freigegeben, d.h. bei Aufruf wieder ausgeführt.	
Makro-/Bildpage auswählen			K	n1					Auswahl einer Page für Makros und Bilder n1=0..15. Ist ein Makro/Bild in der akt. Page 1..15 nicht definiert, dann wird dieses Makro/Bild von Page 0 genommen. z.B. zum Umschalten von Sprachen oder für horizontalen / vertikalen Einbau.		
Makro-/Bildpage sichern			W					die aktuelle Makro-/Bildpage wird gesichert (bei Verwendung in Prozessmakros)			
Makro-/Bildpage restore	R					die letzte gesicherte Makro-/Bildpage wird wieder eingestellt					
<b>automatische (Normal-) Makros</b>											
Makro mit Verzögerung	ESC	M	G	n1	n2					Das (Normal-)Makro mit der Nummer n1 (0..255) in n2/10s aufrufen. Ausführung wird durch Befehle (z.B durch Empfang oder Touchmakros) gestoppt.	
autom. Makros einmal			E	n1	n2	n3				Makros n1..n2 automatisch einmal abarbeiten; n3=Pause in 1/10s. Ausführung wird durch Befehle (z.B durch Empfang oder Touchmakros) gestoppt.	
autom. Makros zyklisch			A	n1	n2	n3				Makros n1..n2 automatisch zyklisch abarbeiten; n3=Pause in 1/10s. Ausführung wird durch Befehle (z.B durch Empfang oder Touchmakros) gestoppt.	
autom. Makros pingpong			J	n1	n2	n3				Makros autom. von n1..n2..n1 (PingPong) abarbeiten; n3=Pause in 1/10s. Ausführung wird z.B. durch Empfang oder Touchmakros gestoppt.	
<b>Makro Prozesse</b>											
Makroprozess definieren	ESC	M	D	nr	typ	n3	n4	zs	Ein Makroprozess mit der Nummer nr (1..8) wird definiert (1=höchste Priorität). Die (Normal-) Makros n3 bis n4 werden nacheinander alle zs/10s ausgeführt. typ: 1=einmal; 2=zyklisch; 3=pingpong n3..n4..n3		
Makroprozess Zeitintervall			Z	nr	zs					Dem Makroprozess mit der Nummer nr (1..8) wird eine neue Zeit zs in 1/10s zugeordnet. Ist die Zeit zs=0 so wird die Ausführung angehalten.	
Makroprozesse anhalten			S	n1					Alle Makroprozesse werden mit n1=0 gestoppt und n1=1 wieder gestartet, um z.B. Einstellungen und Ausgaben über die Schnittstelle ungestört auszuführen	1	
<b>Sonstige-Befehle</b>											
Warten (Pause)	ESC	X	n1					n1 Zehntel-Sekunden abwarten bevor der nächste Befehl ausgeführt wird.			
RS485 Adresse einstellen	ESC	K	A	adr					nur für RS232/RS485 Betrieb und nur bei Hardwareadresse 0 möglich Dem eDIP wird eine neue Adresse adr zugewiesen (im PowerOn-Makro).		
Summer Ein / Aus	ESC	Y	S	n1					Der Summerausgang (PIN16) wird n1=0: AUS; n1=1: EIN; n1=2..255: für n1 Zehntel Sek. lang eingeschaltet	AUS	
Beleuchtung Ein/Aus			L	n1					LED-Beleuchtung n1=0: AUS; n1=1: EIN; n1=2..255: Beleuchtung für n1 Zehntel Sek. lang einschalten.	1	
Beleuchtung Helligkeit			H	n1					Helligkeit der LED-Beleuchtung einstellen n1=0..100%. n1=250 aktuelle Helligkeit als Starthelligkeit speichern; n1=254 LED sofort AUS; n1=255 sofort auf 100%.	100	
Output-Port schreiben			W	n1	n2					n1=0: Alle Ausgabe-Ports entsprechend n2 (=6/8-Bit Binärwert) einstellen. n1=1..6/8: Port n1 rücksetzen (n2=0); setzen (n2=1); invertieren (n2=2);	auf 1
Bytes senden	ESC	S	B	anz	daten ...				Es werden anz (=1..255) Bytes zum Sendepuffer gesendet daten ... = anz Bytes Im Quelltext der Makroprogrammierung darf die Anzahl anz nicht angegeben werden, diese wird vom eDIP-Compiler gezählt und eingetragen.		
Version senden			V					Es wird die Version als String gesendet z.B "EA eDIP320-8 V1.0 Rev.A TP+"			
interne Infos senden			I					Es werden interne Informationen vom eDIP gesendet.			
Power Down	ESC	P	D	n1					Nach diesem Befehl geht das Display in den Power Down mode. n1=0: Aufwachen nur durch Reset; n1=1: Aufwachen durch L-Pegel an WUP-Pin n1=2: Aufwachen durch Touch; n1=3: Aufwachen durch WUP-Pin oder Touch;		

EA eDIP320-8: Befehle für das Touch-Panel											nach		
Befehl	Codes		Anmerkung								Reset		
<b>Touch: Bereiche definieren</b>													
Touch-Taste definieren  (Taste ist gedrückt solange der Touch berührt wird)	ESC	A	T	xx1	yy1	xx2	yy2	down Code	up Code	Text ...	NU	T: Der Bereich von xx1,yy1 nach xx2,yy2 wird als Taste definiert. U: Das Bild Nr. n1 wird nach xx1,yy2 geladen und als Taste definiert. down Code: (1-255) Rückgabe/Touchmakro beim Drücken der Taste. up Code: (1-255) Rückgabe/Touchmakro beim Loslassen der Taste. (down-/up-Code = 0 drücken/loslassen wird nicht gemeldet). Text: Das erste Zeichen bestimmt die Ausrichtung des Textes (C=zentriert L=linksbündig R=rechtsbündig) danach folgt eine Zeichenkette die mit dem akt. Touch-Font in der Taste plaziert wird. Mehrzeilige Texte werden mit dem Zeichen ' ' (\$7C, dez: 124) getrennt; NUL: (\$00) = Zeichenkettenende	
			U	xx1	yy1	n1	down Code	up Code	Text ...	NU			
Touch-Schalter definieren  (Zustand der Schalter toggelt nach jeder Berührung)	ESC	A	K	xx1	yy1	xx2	yy2	down Code	up Code	Text ...	NU	K: Der Bereich von xx1,yy1 nach xx2,yy2 wird als Schalter definiert. J: Das Bild n1 wird nach xx1,yy2 geladen und als Schalter definiert. down Code: (1-255) Rückgabe/Touchmakro beim Einschalten. up Code: (1-255) Rückgabe/Touchmakro beim Ausschalten. (down-/up-Code = 0 Ein-/Ausschalten wird nicht gemeldet). Text: Das erste Zeichen bestimmt die Ausrichtung des Textes (C=zentriert L=linksbündig R=rechtsbündig) danach folgt eine Zeichenkette die mit dem akt. Touch-Font in der Taste plaziert wird. Mehrzeilige Texte werden mit dem Zeichen ' ' (\$7C, dez: 124) getrennt; NUL: (\$00) = Zeichenkettenende	
			J	xx1	yy1	n1	down Code	up Code	Text ...	NU			
Touch-Taste mit Menüfunktion definieren	ESC	A	M	xx1	yy1	xx2	yy2	down Code	up Code	mnu Code	Text ...	NU	Der Bereich xx1,yy1 nach xx2,yy2 wird als Menü-Taste definiert. down Code: (1-255) Rückgabe/Touchmakro beim Drücken. up Code: (1-255) Rückgabe/Touchmakro beim Menü-Abbruch mnu Code: (1-255) Rückgabe/Menuumakro+(EintragsNr-1) nach Auswahl eines Menü-Eintrages. (down-/up-Code=0: Aktivieren/Abbruch wird nicht gemeldet). Text:= Zeichenkette mit den Tastentext und den Menüeinträgen. Das erste Zeichen bestimmt die Richtung in der das Menü aufklappt (R=rechts L=links O=oben U=Unten). Das zweite Zeichen bestimmt die Ausrichtung des Touchtasten-Textes (C=zentriert L=linksbündig R=rechtsbündig). Die Menü-Einträge sind durch Zeichen ' ' (\$7C, dez: 124) getrennt. z.B. "UCTaste Eintrag1 Eintrag2 Eintrag3" Der Tastentext wird mit dem akt. Touchfont und die Menü-Einträge mit dem akt. Menüfont gezeichnet. Der Hintergrund des Menüs wird automatisch gesichert.
Zeichenbereich definieren	ESC	A	D	xx1	yy1	xx2	yy2	n1	Ein Zeichenbereich wird definiert. Innerhalb der Eck-Koodinaten xx1,yy1 und xx2,yy2 kann dann mit der Strichstärke n1 gezeichnet werden.				
Freien Touchbereich def.	ESC	A	H	xx1	yy1	xx2	yy2	Ein frei benutzbarer Touchbereich wird definiert. Touchaktionen (down, up und drag) innerhalb der Eck-Koodinaten xx1,yy1 und xx2,yy2 werden gesendet.					
Bar per Touch einstellbar	ESC	A	B	nr	Der Bargraph mit der Nr. n1 wird zur Eingabe per Touchpanel definiert.								
<b>Touch: Einstellungen</b>													
Touch-Rahmen Form	ESC	A	E	n1	mit n1 wird der Rahmentyp für die Darstellung von Touch-Tasten/Schaltern eingestellt						1		
Touch-Tasten Reaktion			I	n1	automatisches Invertieren beim Berühren der Touch-Taste: n1=0=AUS; n1=1=EIN;						1		
Touch-Taste Invertieren			S	n1	Summer piepst kurz beim Berühren einer Touch-Taste: n1=0=AUS; n1=1=EIN						1		
Touch-Schalter abfragen			N	Code	Die Touch-Taste mit dem zugeordnetem Return-Code wird manuell Invertiert								
Touch-Schalter einstellen			X	Code	Zustand des Schalters (Aus=0; Ein=1) wird in den Sendepuffer gestellt.								
			P	Code	n1	Zustand des Schalters wird per Befehl geändert n1=0=Aus; n1=1=Ein.							
Radiogroup für Schalter			R	nr	Innerhalb einer Gruppe ist immer nur 1 Schalter aktiv, alle anderen werden deaktiviert nr=0: neu definierte Schalter gehören keiner Gruppe an. nr=1..255: neu definierte Schalter gehören der Gruppe mit der Nummer nr an. Bei Schalter in einer Gruppe wird nur der downcode beachtet, der upcode wird ignoriert						0		
Radiogroup abfragen			G	nr	der downcode des aktivierten Schalters aus der Radiogroup mit der Nummer nr wird in den Sendepuffer gestellt.								
Touch-Bereich Löschen			L	Code	n1	Der Touchbereich mit dem Return-Code (Code=0: alle Touchbereiche) wird aus der Touchabfrage entfernt. Mit n1=0 bleibt der Bereich am Display sichtbar, mit n1=1 wird der Bereich gelöscht.							
			V	xx1	yy1	n1	Touchbereich der die Koordinaten xx1,yy1 umschließt aus der Touchabfrage entfernen n1=0: Bereich bleibt sichtbar; n1=1: Bereich löschen						
Barwert automatisch senden	Q	n1	das automatische Senden eines neuen Bargraphwertes per Toucheingabe wird n1=0:deaktiviert; n1=1:neuer Wert wird nach dem Einstellen gesendet; n1=2: jede Änderung wird während des Einstellens gesendet.						1				
Touchabfrage Ein/Aus	A	n1	Touchabfrage wird n1=0:deaktiviert; n1=1:aktiviert;										
Touchabfrage drehen	O	n1	n1=0: normale abfrage; n1=1: Touchabfrage für Topview (Lötbrücken umgestellt)						1				
<b>Touch: Beschriftungs-Font</b>													
Beschriftungs Font	ESC	A	F	nr	Font mit der Nummer nr (0..31) für Touchtastenbeschriftung einstellen						0		
Beschriftungs-Zoomfaktor			Z	n1	n2	n1 = X-Zoomfaktor (1x..8x); n2 = Y-Zoomfaktor (1x..8x)						1,1	
zus. Zeilenabstand			Y	n1	zwischen zwei Textzeilen n1 Pixel (0..15) als zusätzlichen Zeilenabstand einfügen								
Beschriftungs-Winkel			W	n1	Text-Ausgabewinkel: n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°						0		



Antworten des EA eDIP320-8						
Kennung	anz	daten			Anmerkung	
<b>automatische Antworten</b>						
ESC	A	1	code		Antwort vom Analogen Touchpanel wenn eine Taste/Schalter gedrückt wurde. code = down oder up Code der Taste/Schalter. Es wird nur gesendet wenn kein Touch-Makro mit der Nr. code definiert ist !	
ESC	N	1	code		Nach dem Auswählen eines Menüeintrages per Touch wird der ausgewählte Menüeintrag code gesendet. Es wird nur gesendet wenn kein Menü-Makro mit der Nr. code definiert ist !	
ESC	B	2	nr	wert	Nach dem Einstellen eines Bargraph per Touch wird der aktuelle wert des Bars mit der nr gesendet. Barwert Senden muß aktiviert sein siehe Befehl 'ESC A Q n1'.	
ESC	T	0			Falls das automatische Öffnen eines Touchmenüs deaktiviert ist (siehe Befehl 'ESC N T n1'), so wird diese Anforderung an den Hostrechner gesendet. Dieser kann dann das Touchmenü mit dem Befehl 'ESC N T 2' öffnen.	
ESC	H	5	typ	xLO xHI yLO yHI	Bei einem freien Touchbereich-Ereignis wird folgendes gesendet: typ=0 ist Loslassen; typ=1 ist Berühren; typ=2 ist Draggen innerhalb des freien Touchbereiches an den Koordinaten xx1,yy1	
<b>Antworten nur nach Anforderung per Befehl</b>						
ESC	N	1	nr		Nach dem Befehl 'ESC N S' wird der aktuell ausgewählte Menüeintrag gesendet. nr=0: kein Menüeintrag ist ausgewählt.	
ESC	B	2	nr	wert	Nach dem Befehl 'ESC B S n1' wird der aktuelle Wert Bars mit der Nr. nr gesendet.	
ESC	X	2	code	wert	Nach dem Befehl 'ESC A X code' wird der aktuelle Zustand des Touch-Schalters mit dem Return-Code code gesendet. wert = 0 oder 1	
ESC	G	2	nr	code	Nach dem Befehl 'ESC A G nr' wird der code des aktiven Touch-Schalters von der Radiogroup nr gesendet.	
ESC	V	anz	Zeichenkette...		Nach dem Befehl 'ESC S V' wird die Version der eDIP-Firmware als Zeichenkette gesendet. z.B "EA eDIP320-8 V1.0 Rev.A TP+"	
ESC	I	anz	X-Pixel, Y-Pixel, Version, Touchinfo, CRC-ROM, CRC-ROMsoll DF in KB, CRC-DF, CRC-DFsoll, DFanz		anz = 21 Nach dem Befehl 'ESC S I' werden interne Informationen vom eDIP gesendet (16-Bit integer Werte LO- HI-Byte) Version: LO-Byte = Versionsnr. Software; HI-Byte = Hardwarerevisionsbuchstabe Touchinfo: LO-Byte = '- +' X-Richtung erkannt; HI-Byte = '- +' Y-Richtung erkannt DFanz: Anzahl benutzter Bytes im Dateiflash (3 Byte: LO-, MID- HI-Byte)	
<b>Antworten ohne Längenangabe (anz)</b>						
ESC	U	L	xx1	yy1	Bilddaten... (BH7-FORMAT)	Nach dem Befehl 'ESC UH...' wird ein Hardcopy im BH7-Format gesendet. xx1,yy1 = Startkoordinaten des Hardcopys (Linke obere Ecke)

## BEFEHLE ÜBER DIE SERIELLE SCHNITTSTELLE SENDEN

Die Bedieneinheit läßt sich über diverse eingebaute Befehle programmieren. Jeder Befehl beginnt mit ESCAPE gefolgt von einem oder zwei Befehlsbuchstaben und einigen Parametern. Es gibt zwei Möglichkeiten Befehle zu senden:

### 1. ASCII-Modus

- Das Escape-Zeichen entspricht dem Zeichen '#' (hex: \$23, dez: 35).
- Die Befehlsbuchstaben folgen direkt im Anschluss an das '#' Zeichen.
- Die Parameter werden im Klartext (mehrere ASCII Ziffern) mit einem nachfolgenden Trennzeichen (z.B. das Komma ',') gesendet - auch hinter dem letzten Parameter z.B.: **#GD0,0,319,239**,
- Zeichenketten (Texte) werden direkt ohne Anführungsstrichen geschrieben und mit CR (hex: \$0D), oder LF (hex: \$0A) abgeschlossen.

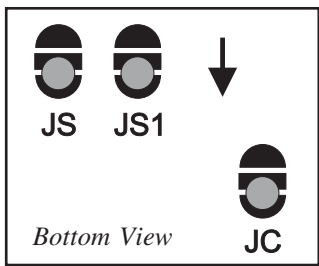
### 2. Binär-Modus

- Das Escape-Zeichen entspricht dem Zeichen ESC (hex: \$1B, dez: 27).
- Die Befehlsbuchstaben werden direkt gesendet.
- Die Koodinaten xx und yy werden als 16-Bit Binärwerte (zuerst das LOW-Byte dann das HIGH-Byte) gesendet.
- Alle anderen Parameter werden als 8-Bit Binärwert (1 Byte) gesendet.
- Zeichenketten (Texte) werden mit CR (hex: \$0D), LF (hex: \$0A) oder NUL (hex: \$00) abgeschlossen.

Im Binär-Modus dürfen keine Trennzeichen z.B. Leerzeichen oder Kommas verwendet werden. Die Befehle benötigen auch **kein Abschlussbyte** wie z.B Carrige Return (außer Zeichenkette: \$00).

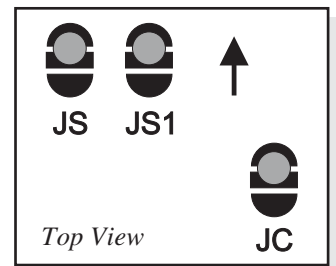
## TOPVIEW DURCH GEDREHTEN EINBAU

Die Vorzugsblickrichtung des EA eDIP320 ist schräg von Unten (BottomView, 6 Uhr). Das eDIP320 kann um **180° gedreht eingebaut** werden um die Blickrichtung von Oben (TopView, 12 Uhr) zu erhalten.

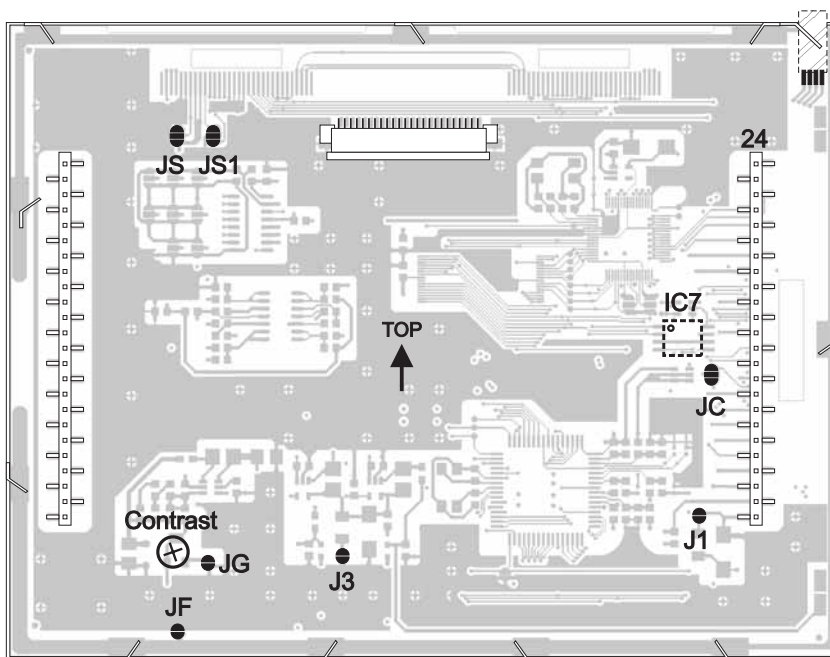


Zur Richtigstellung des Bildinhaltes müssen drei Wechsellötbrücken JS, JS1 und JC umgelötet werden.

**Achtung:** Immer alle Lötbrücken in die gleiche Position setzen, sauber entlöten, Schlüsse zerstören das eDIP320-8.



Wird ein eDIP320-8 mit Touchpanel verwendet, so muss mit dem Befehl 'ESC AO 1' die Touchauswertung ebenfalls umgedreht werden.



## POWER-DOWN-MODE

Um Strom zu sparen (Betrieb mit Akku) kann man mit dem Befehl 'ESC PD n1' (siehe S.15 unten) den Power-down-mode aktivieren. Die LED-Beleuchtung wird dabei ausgeschaltet und der Displayinhalt ist nicht mehr sichtbar, bleibt jedoch erhalten.

Im Power-down-mode inkl. Suppressordioden benötigt das eDIP320 typ. 150µA.

Durch die integrierten Suppressordioden kann der Querstrom aber auch 1000µA und mehr betragen. Die Suppressordioden können durch Öffnen der Lötbrücken J1 und J3 deaktiviert werden, dann wird ein Power-down-strom von typ. 20µA erreicht.

**Achtung:** Bei geöffneten Lötbrücken J1+J3 unbedingt auf die richtige Polarität des Displays VDD,GND (Pin1+2) achten! Eine auch noch so kurzzeitige Verpolung oder Überspannung kann dann zur sofortigen Zerstörung des gesamten Displays führen.

Das eDIP320 kann durch L-Pegel an Pin13 (WUP), durch Berührung des Touches oder durch Ansprechen der I2C Adresse aus dem Power-down-mode aufgeweckt werden.

## MAKRO PROGRAMMIERUNG

Einzelne oder mehrere Befehlsfolgen können als sog. Makros zusammengefasst und im DatenFlash fest abgespeichert werden. Diese können dann mit den Befehlen *Makro ausführend* gestartet werden. Es gibt verschiedene Makrotypen (Compileranweisungen sind grün geschrieben):

### Normal Makro (0..255) *Makro*:

Start per Befehl 'ESC MN xx' über 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). Diese automatischen Makros werden solange abgearbeitet bis ein Befehl über die Schnittstelle empfangen wird, oder ein Touchmakro mit entsprechendem Return-Code ausgelöst wird. Ausserdem werden diese Makros von Makro-Prozessen in definierten Intervallen aufgerufen. Makro-Prozesse werden nicht durch Empfang von Befehlen von der Schnittstelle oder von ausgelösten Touchmakros unterbrochen.

### Touch Makro (1..255) *TouchMakro*:

Start beim Berühren/Loslassen eines Touchfeldes (nur bei Versionen mit Touch Panel TP) oder per Befehl 'ESC MT xx'.

### Menü Makro (1..255) *MenuMakro*:

Start bei Auswahl eines Menüeintrages oder per Befehl 'ESC MM xx'.

### Power-On-Makro *PowerOnMakro*:

Start nach dem Einschalten. Hier kann man zB. den Cursor abschalten und einen Startbildschirm definieren.

### Reset-Makro *ResetMakro*:

Start nach einem externen Reset (L-Pegel an Pin 5).

### Watchdog-Makro *WatchdogMakro*:

Start nach einem Fehlerfall (z.B. Absturz).

### Brown-Out-Makro *BrownOutMakro*:

Start nach einem Spannungseinbruch <3V.

### WakeUpPin-Makro *WakeUpPinMakro*:

Start nach dem Aufwachen aus dem Power-Down-Mode per Pin13 (WUP).

### WakeUpTouch-Makro *WakeUpTouchMakro*:

Start nach dem Aufwachen aus dem Power-Down-Mode per Touch-Berührung (gesamte Touchfläche ist aktiv).

### WakeUpI<sup>2</sup>C-Makro *WakeUpI2CMakro*:

Start aus dem Power-Down-Mode über den I<sup>2</sup>C Bus

**Achtung:** Wird im PowerOn-, Reset-, Watchdog- oder BrownOut-Makro eine Endlosschleife programmiert, ist das Display nicht mehr ansprechbar. In diesen Fall muss die Ausführung des Power-On Makros unterdrückt werden. Das erreicht man durch die Beschaltung von WUP: PowerOff - Pin 13 (WUP) auf GND legen - PowerOn - Pin 13 (WUP) wieder öffnen.

## ERSTELLEN INDIVIDUELLER MAKROS UND BILDER

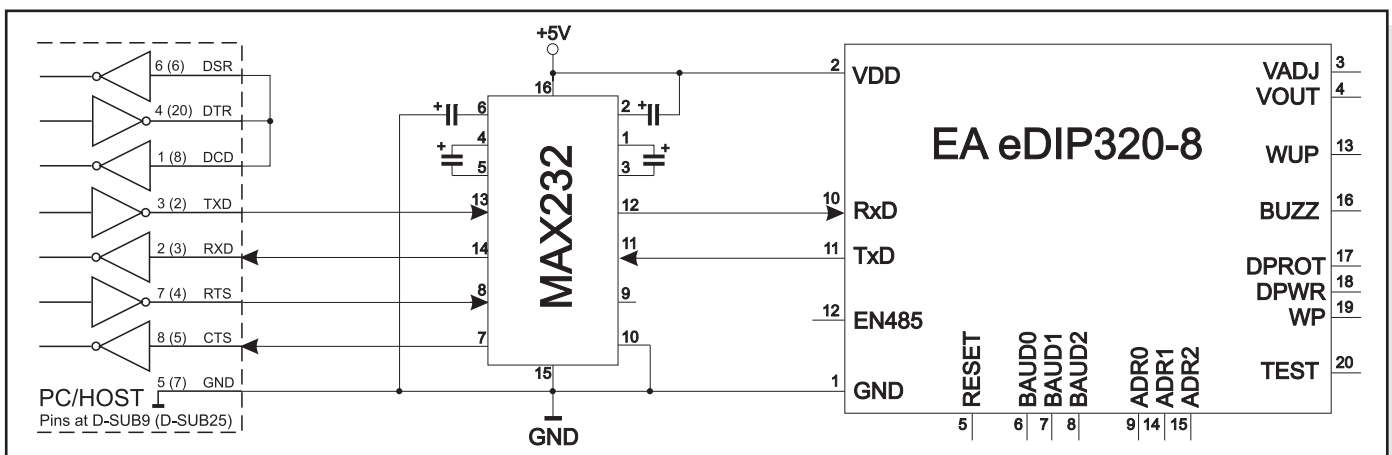
Um nun Ihre speziellen Makros erstellen zu können, benötigen Sie folgende Hilfsmittel:

- um das Display an den PC anschliessen zu können benötigen Sie den als Zubehör erhältlichen USB-Programmer EA 9778-1USB oder einen selbstgebauten Adapter mit Pegelwandler MAX232 (Applikationsbeispiel unten).
- die Software ELECTRONIC ASSMBLY LCD-Tools<sup>\*)</sup>; sie enthält einen Kit-Editor, Kit-Compiler, sowie Beispiele und Fonts (für PC-Win)
- einen PC mit USB oder serieller Schnittstelle COM

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 über den Kit-Editor definiert, startet man über F5 den eDIP320-Compiler. Dieser erzeugt eine Datei DEMO.DF. Ist auch ein Programmer EA 9778-1USB angeschlossen oder das Display über einen MAX232 an den PC angeschlossen, dann wird diese Datei automatisch in das DatenFlash des Displays gebrannt. Der eDIP320-Compiler erkennt das Display mit und ohne eingeschaltetem Small-Protokoll.

Eine ausführliche Beschreibung zur Programmierung der Makros finden Sie zusammen mit Beispielen in der Hilfefunktion der ELECTRONIC ASSEMBLY LCD-Tools<sup>\*)</sup> Software.



Adapter zum Selberbauen für direkten PC-Anschluss

<sup>\*)</sup> im Internet unter <http://www.lcd-module.de/deu/touch/touch.htm>

## BILDER IM DATENFLASH ABGELEGEN

Um die Übertragungszeiten der Schnittstelle zu verkürzen, oder auch um Speicherplatz im Prozessorsystem zu sparen, können bis zu 256 Bilder im internen DatenFlash (80kB) abgelegt werden (Compileranweisung "PICTURE"). Der Aufruf erfolgt über den Befehl "ESC U I" oder aus einem Makro heraus.

Verwendet werden können alle Bilder im Windows BMP-Format (nur monochrome Bilder). Die Erstellung und Bearbeitung erfolgt über Standardsoftware wie z.B. Windows Paint, Photoshop oder über den mitgelieferten BitmapEditor.

Mit der Compileranweisung "PICTURE" können auch zwei gleich grosse monochrome BMPs für Touchtasten, Bildmasken oder Blinkbilder eingebunden werden.

```
PICTURE: 1 <BITMAP1.BMP>
PICTURE: 2 <BITMAP2.BMP>, <MASK2.BMP>
PICTURE: 3 <BITMAP3.BMP>, <BLINK3.BMP>
PICTURE: 4 <TOUCH.BMP>, <TOUCHPRESSED.BMP>
```

## MAKRO PAGES (MEHRSPRACHIGKEIT)

Für die Normal-, Touch- und Menu-Makros, sowie den internen Bildern stehen je 16 komplette Makrosätze zur Verfügung. Somit können z.B. durch einfaches Umschalten der aktiven Makropage (ESC M K n1) bis zu 16 verschiedene Sprachen unterstützt werden.

Wird im Kiteditor ein Makro/Bild definiert so kann nach der Makro-/Bildnummer eine Pagenummer in ecktigen Klammern angegeben werden.

Ist ein Makro/Bild in der aktuellen eingestellten Page [1]..[15] nicht definiert, dann wird automatisch dieses Makro/Bild von Page [0] genommen. Es müssen also nicht alle Makros und Bilder mehrfach abgelegt werden wenn Sie in unterschiedlichen Sprachen gleich sind.

```
PICTURE: 100[0] <SAUSAGE.BMP>
PICTURE: 100[1] <BEER.BMP>
PICTURE: 100[2] <WINE.BMP>

MACRO: 2[0] ; SAME AS "MACRO: 0"
#ZV REPLACE
#ZL 25,0 "DEUTSCH "
#UI 0,20, 100

MACRO: 2[1] ; ENGLISH
#ZV REPLACE
#ZL 25,0 "ENGLISH "
#UI 0,20, 100

MACRO: 2[2] ; FRENCH
#ZV REPLACE
#ZL 25,0 "FRANCAISE"
#UI 0,20, 100
```

## SCHREIBSCHUTZ FÜR MAKROPROGRAMMIERUNG UND FONTS

Ein LO-Pegel am Pin 19 (WP) verhindert ein versehentliches Überschreiben der Makros, Bilder und Fonts im DatenFlash (in jedem Fall empfohlen!).

## SPEICHERERWEITERUNG

Der interne DatenFlash Speicher beträgt 80kB. In der Regel steht dadurch ausreichend Platz für viele Icons und Makros zur Verfügung. Wenn jedoch sehr viele Bilder (vor allem Vollbilder) oder mehrere grosse Zeichensätze abgelegt werden sollen, kann es erforderlich sein Speicher nachzurüsten (max. 8192kB). Möglich ist eine Erweiterung durch direktes Einlöten eines Datenflash aus der Serie AT45DBxxxD-SU auf dem eDIP320 (siehe S.18 IC7).

z.B. AT45DB081D-SU = 1024kB, AT45DB161D-SU = 2048kB oder AT45DB321D-SU = 4096kB.

# BOARD FÜR EA eDIP320-8



## TECHNISCHE DATEN

- \* **EA 9778-1USB**
- \* PROGRAMMIERBOARD FÜR USB
- \* INKL. USB-KABEL
- \* EINFACHSTE ANWENDUNG, KEINE STROMVERSORGUNG NÖTIG
- \* ERFORDERT USBTREIBER, IM LIEFERUMFANG ENTHALTEN
- \* **EA 9778-1RS232**
- \* INTERFACEBOARD RS-232 MIT  $\pm 12V$  PEGELN AN Rx D UND Tx D
- \* INKL. KABEL EA KV24-9B MIT 9-POL. D-SUB STECKER
- \* BENÖTIGT EXTERNE VERSORGUNG +5V/typ. 270mA
- \* OPTIONALVERSORGUNG 9..35VDC STATT 5V (EA OPT-9/35V)
- \* **EA 9778-1RS485**
- \* INTERFACEBOARD FÜR RS-485 2-DRAHTVERBINDUNG
- \* BENÖTIGT EXTERNE VERSORGUNG +5V/typ. 300mA
- \* OPTIONALVERSORGUNG 9..35VDC STATT 5V (EA OPT-9/35V)

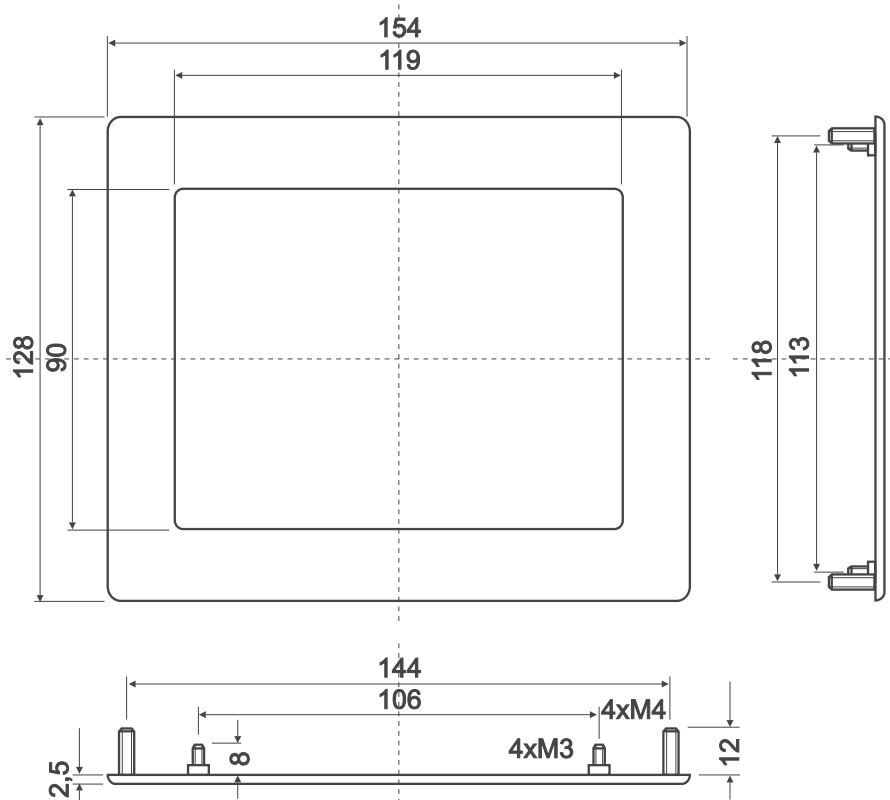
## BESTELLBEZEICHNUNG

STARTERKIT BLAU, (1x EAeDIP320B-8LWTP + EA9778-1USB)  
STARTERKIT S/W, (1x EAeDIP320J-8LWTP + EA9778-1USB)  
PROGRAMMIERBOARD INKL. USB-KABEL UND CD FÜR PC  
RS-232 BOARD MIT  $\pm 12V$  PEGELN AN Rx D UND Tx D  
INTERFACEBOARD FÜR RS-485 2-DRAHTVERBINDUNG  
VERSORGUNG 9..35VDC STATT 5V (NUR 9778-1RS232,-1RS485)

**EA STARTeDIP320B**  
**EA STARTeDIP320J**  
**EA 9778-1USB**  
**EA 9778-1RS232**  
**EA 9778-1RS485**  
**EA OPT-9/35V**

## EINBAUBLENDE EA0FP321-8SW

Als Zubehör liefern wir optional eine schwarz eloxierte Einbaublende aus Aluminium.



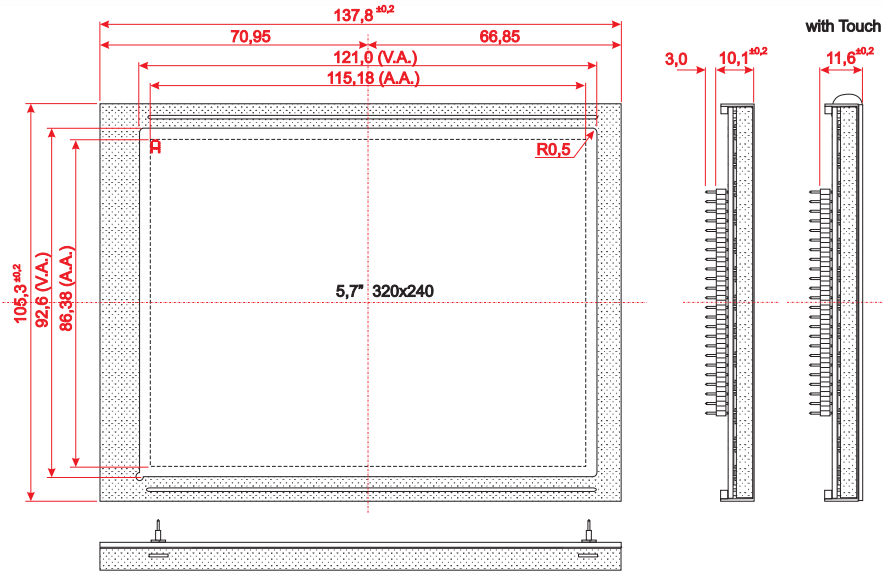
alle Maße in mm

## HINWEISE ZUR HANDHABUNG UND ZUM BETRIEB

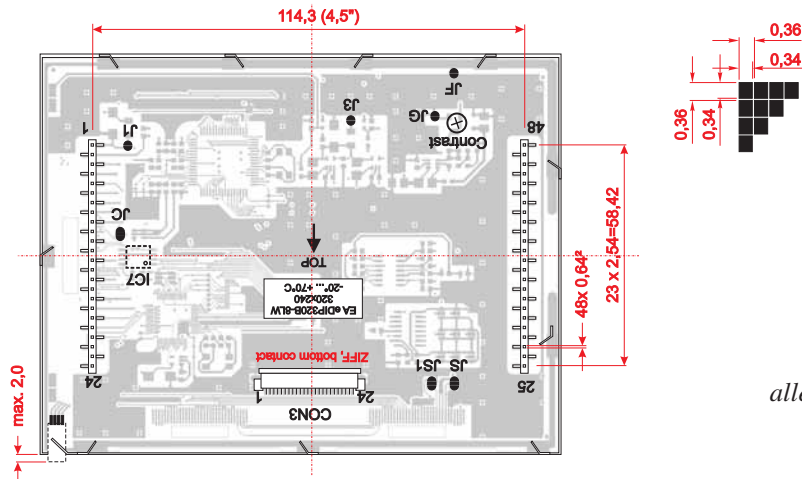
- Zur elektrischen Zerstörung des Moduls kann führen: Verpolung oder Überspannung der Stromversorgung, Überspannung oder Verpolung bzw. statische Entladung an den Eingängen, Kurzschließen der Ausgänge.
- Vor dem Abstecken des Moduls muß unbedingt die Stromversorgung abgeschaltet sein. Ebenso müssen alle Eingänge stromlos sein.
- Das Display und der Touchscreen bestehen aus Kunststoff und dürfen nicht mit harten Gegenständen in Berührung kommen. Die Oberflächen können mit einem weichen Tuch ohne Verwendung von Lösungsmitteln gereinigt werden.
- Das Modul ist ausschließlich für den Betrieb innerhalb von Gebäuden konzipiert. Für den Betrieb im Freien müssen zusätzliche Vorkehrungen getroffen werden. Der maximale Temperaturbereich von  $-20..+70^{\circ}\text{C}$  darf nicht überschritten werden. Bei Einsatz in feuchter Umgebung kann es zu Funktionsstörungen und zum Ausfall des Moduls kommen. Das Display ist vor direkter Sonneneinstrahlung zu schützen.



## ABMESSUNGEN



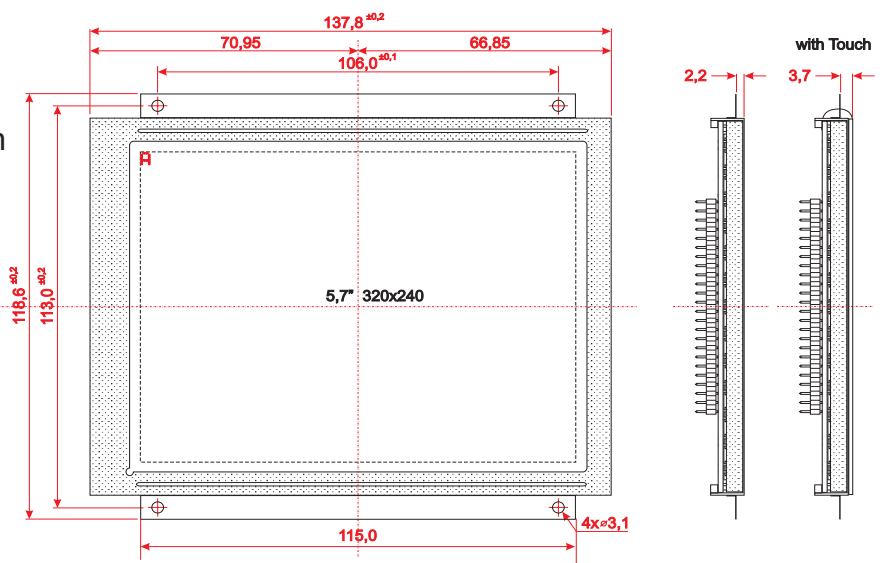
*Hinweis:  
LC-Displays sind generell  
nicht geeignet für Wellen- oder  
Reflowlötung. Temperaturen  
über 90°C können bleibende  
Schäden hinterlassen.*



alle Maße in mm

## ABMESSUNGEN MIT MONTAGELASCHEN

Die Montagelaschen sind im Lieferumfang enthalten.



# ELECTRONIC ASSEMBLY

making things easy