

embedded LCD-DISPLAY 240x128 MIT INTELLIGENZ

Weltneuheit !



EA eDIP240B-7LWTP
Abm. 113x70mm

TECHNISCHE DATEN

- * LCD-GRAFIKDISPLAY MIT DIVERSEN GRAFIKFUNKTIONEN
- * 8 EINGEBAUTE FONTS
- * FONT ZOOM VON ca. 2mm BIS ZU ca. 50mm, auch um 90° GEDREHT
- * 3 VERSCHIEDENE INTERFACE ONBOARD: RS-232, I²C-BUS ODER SPI-BUS
- * 240x128 PIXEL MIT LED-BELEUCHTUNG BLAU NEGATIV ODER SCHWARZ-WEISS POSITIV, FSTN-TECHNIK
- * VERSORGUNG +5V @ typ. 30mA / 190mA (OHNE / MIT LED BELEUCHTUNG)
- * **PIXELGENAU** POSITIONIERUNG BEI ALLEN FUNKTIONEN
- * GERADE, PUNKT, BEREICH, UND/ODER/EXOR, BARGRAPH...
- * CLIPBOARD FUNKTIONEN, PULL-DOWN MENÜS
- * BIS ZU 256 BILDER INTERN SPEICHERBAR
- * BIS ZU 256 MAKROS PROGRAMMIERBAR (32kB EEPROM ONBOARD)
- * TEXT UND GRAFIK MISCHEN, BLINKATTRIBUTE: EIN/AUS/INVERS BLINKEN
- * BELEUCHTUNG PER SOFTWARE SCHALTBAR
- * ANALOGESTOUCH PANEL: VARIABLES RASTER
- * FREI DEFINIERBARE TASTEN UND SCHALTER

BESTELLBEZEICHNUNG

240x128 DOTS, WEISSE LED-BELEUCHTUNG, BLAU NEGATIV
WIEVOR, JEDOCH MIT TOUCH PANEL

EA eDIP240B-7LW
EA eDIP240B-7LWTP

240x128 DOTS, WEISSE LED-BELEUCHTUNG, POSITIV MODE, FSTN
WIEVOR, JEDOCH MIT TOUCH PANEL

EA eDIP240J-7LW
EA eDIP240J-7LWTP

Documentation of revision				
Date	Type	Old	New	Reason / Description
15.2.04	First Issue			Preliminary version

INHALT

ALLGEMEINES	3
SPEZIFIKATION	4
RS-232 BESCHALTUNG	5
SPI BESCHALTUNG	6
I ² C BESCHALTUNG	7
PROTOKOLL	8
TOUCHPANEL	9
ZEICHENSÄTZE	10-11
BEFEHLE INTABELLENFORM	12~14
RÜCKANTWORTEN DES BEDIENPANELS	15
PROGRAMMIERBEISPIEL	16
MAKROPROGRAMMIERUNG	17-18
ABMESSUNGEN	19

ALLGEMEINES

EA eDIP240-7 ist das weltweit erste Display mit integrierter Intelligenz ! Neben diversen eingebauten Schriften welche pixelgenau verwendet werden können, bietet es zudem eine ganze Reihe ausgefeilter Grafikfunktionen.

Das Display ist mit 5V sofort betriebsbereit. Die Ansteuerung erfolgt über eine der 3 eingebauten Schnittstellen RS-232, SPI oder I²C.

Die Programmierung erfolgt 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 I²C Spezifikation. Zur Erhöhung der Datensicherheit wird für alle Übertragungsvarianten ein einfaches Protokoll verwendet.

ANALOGESTOUCHPANEL

Die Versionen EA eDIP240B-7LWTP und EA eDIP240J-7LWTP sind 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, B- UND J-TYPEN

Beide Displays in blau-weiß (B) und schwarz-weiß (J) sind mit einer modernen und stromsparenden LED-Beleuchtung ausgestattet. Während das schwarz-weiß-Display auch mit komplett abgeschalteter Beleuchtung noch ablesbar ist, benötigt das blau-weiße Display dagegen zum Ablesen in jedem Fall eine minimale Beleuchtung. Die Beleuchtung ist per Befehl abschaltbar und die Helligkeit einstellbar.

Für den Betrieb im direkten Sonnenlicht empfehlen wir die schwarz-weiß-Version. Für alle anderen Einsatzfälle entscheidet der Geschmack und das Design über die jeweils bessere Wahl.

Bitte beachten Sie, dass die LED-Beleuchtung einer gewissen Alterung unterliegt, welche mit einer sog. Halbwertszeit (halbe Helligkeit) von 20.000 Stunden definiert ist.

SOFTWARE

Die Programmierung dieses Displays erfolgt über Befehle wie z.B. *Zeichne ein Rechteck von (0,0) nach (64, 15)*. Es ist keine zusätzliche Software oder Treiber erforderlich. Zeichenketten lassen sich **pixelgenau** platzieren. Blinkattribute können beliebig oft vergeben werden - auch für Grafiken. Das Mischen von Text und Grafik ist jederzeit möglich. Es können bis zu 16 verschiedene Zeichensätze verwendet werden. Jeder Zeichensatz kann wiederum 2- bis 4-fach gezoomt werden. Mit dem größten Zeichensatz lassen sich somit bildschirmfüllende Worte und Zahlen darstellen.

ZUBEHÖR

Programmer für internes EEPROM (in Vorbereitung)

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 Programmer EA 9777-1 die von Ihnen erstellten daten dauerhaft ins on-board EEPROM (32kB).

Der Programmer läuft unter Windows und wird an die serielle Schnittstelle des PC (COM) angeschlossen. Ein Netzteil und ein Schnittstellenkabel sind im Lieferumfang 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
Output Low Voltage				0.7	V
Output High Voltage		4.0			V
Output Current				20	mA
Current	Backlight off		30		mA
	Backlight on		180		mA
Lifetime for half brightness of backlight			20,000		hours

ELECTRONIC ASSEMBLY

RS-232/RS-422 INTERFACE

Wird das Display wie unten gezeigt beschaltet, so ist das RS-232/RS-422 Interface ausgewählt. Die Pinbelegung ist in der Tabelle rechts angegeben.

Die Leitungen RxD und TxD führen 5V CMOS-Pegel, wenn "echte" RS-232 Pegel erwünscht sind, ist ein externer Treiber (z.B. ICL232) erforderlich.

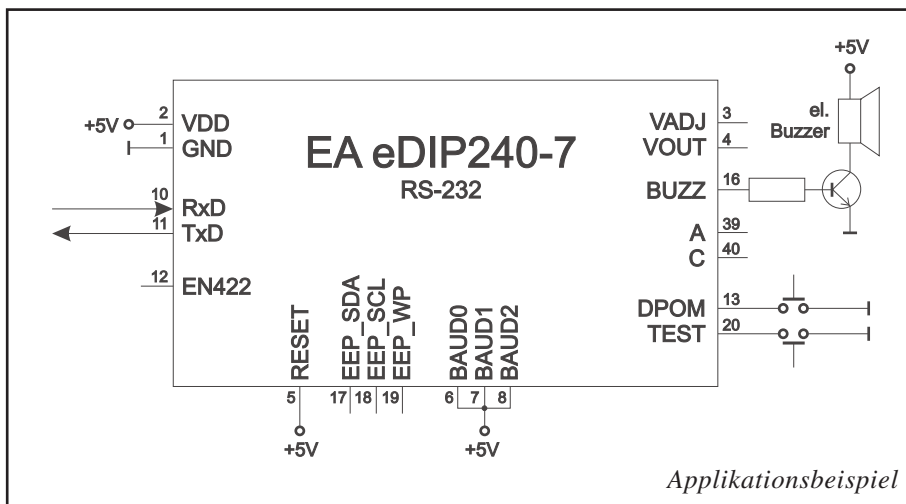
Pinout eDIP240-7						
RS-232 / RS-422 mode						
Pin	Symbol	In/Out	Function	Pin	Symbol	Function
1	GND	-	Ground Potential for logic (0V)	21	N.C.	not connected
2	VDD	-	Power supply for logic (+5V)	22	N.C.	not connected
3	VADJ	In	Operating voltage for LC driving (input)	23	N.C.	not connected
4	VOUT	Out	Output voltage for LC driving	24	N.C.	not connected
5	RESET	-	L: Reset	25	N.C.	not connected
6	BAUD0	In	Baud Rate 0	26	N.C.	not connected
7	BAUD1	In	Baud Rate 1	27	N.C.	not connected
8	BAUD2	In	Baud Rate 2	28	N.C.	not connected
9	N.C.		do not connect, reserved	29	N.C.	not connected
10	RxD	In	Receive Data	30	N.C.	not connected
11	TxD	Out	Transmit Data	31	N.C.	not connected
12	EN422	Out	Enable RS-422 driver	32	N.C.	not connected
13	DPOM	In	L: disable Power-On-Macro do not connect for normal operation	33	N.C.	not connected
14	N.C.		do not connect, reserved	34	N.C.	not connected
15	N.C.		do not connect, reserved	35	N.C.	not connected
16	BUZZ	Out	Buzzer output	36	N.C.	not connected
17	EEP_SDA	Bidir.	Serial Data Line for int. EEPROM	37	N.C.	not connected
18	EEP_SCL	In	Serial Clock Line for int. EEPROM	38	N.C.	not connected
19	EEP_WP	In	H: Write Protect for int. EEPROM	39	A	LED backlight+ / internal connection
20	TEST SBUF	IN Out	open-drain with internal pullup 20..50k IN (Power-On) L: Testmode OUT L: data in sendbuffer	40	C	LED backlight- / internal connection

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.



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



SPI INTERFACE

Wird das Display wie unten gezeigt beschaltet, ist der SPI-Mode aktiviert. Die Datenübertragung erfolgt dann über die serielle synchrone SPI-Schnittstelle.

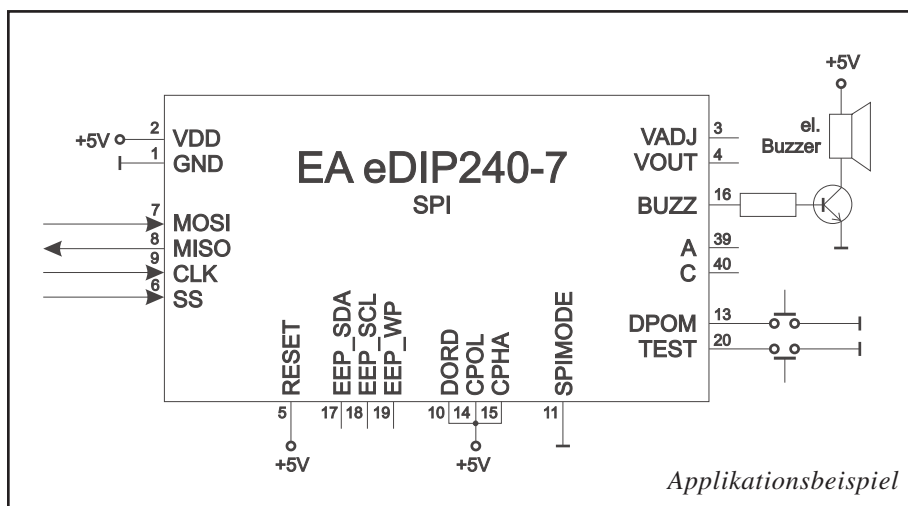
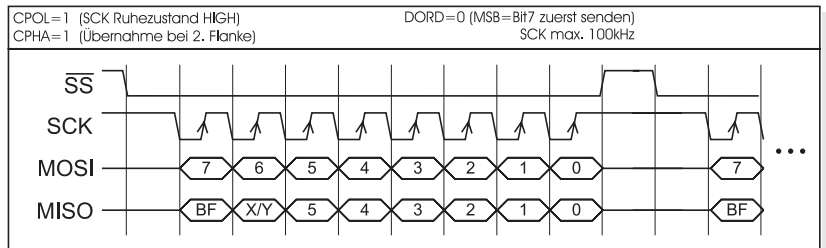
Mit den Eingängen DORD, CPOL und CPHA werden die Hardwarebedingungen an den Master angepasst. Zum Beispiel (vgl. Diagramm unten)

DORD (**D**ata**O**Rder) = 1 = MSB wird zuerst gesendet.

CPOL (**C**lock**P**OLary) = 1 = Ruhezustand von SCK HIGH.

CPHA (**C**lock**P**Hase) = 1 = Übernahme bei steigender (2.) Flanke

Pinout eDIP240-7						
SPI mode						
Pin	Symbol	In/Out	Funktion	Pin	Symbol	Function
1	GND	-	Ground Potential for logic (0V)	21	N.C.	not connected
2	VDD	-	Power supply for logic (+5V)	22	N.C.	not connected
3	VADJ	In	Operating voltage for LC driving (input)	23	N.C.	not connected
4	VOUT	Out	Output voltage for LC driving	24	N.C.	not connected
5	RESET	-	L: Reset	25	N.C.	not connected
6	SS	In	Slave Select	26	N.C.	not connected
7	MOSI	In	Serial In	27	N.C.	not connected
8	MISO	Out	Serial Out	28	N.C.	not connected
9	CLK	In	Shift Clock	29	N.C.	not connected
10	DORD	In	Data Order	30	N.C.	not connected
11	SPIMODE	In	connect to GND for SPI interface	31	N.C.	not connected
12	N.C.	-	do not connect, reserved	32	N.C.	not connected
13	DPOM	In	L: disable Power-On-Macro do not connect for normal operation	33	N.C.	not connected
14	CPOL	In	Clock Polarity	34	N.C.	not connected
15	CPHA	In	Clock Phase	35	N.C.	not connected
16	BUZZ	Out	Buzzer output	36	N.C.	not connected
17	EEP_SDA	Bidir.	Serial Data Line for int. EEPROM	37	N.C.	not connected
18	EEP_SCL	In	Serial Clock Line for int. EEPROM	38	N.C.	not connected
19	EEP_WP	In	H: Write Protect for int. EEPROM	39	A	LED backlight+ / internal connection
20	TEST SBUF	IN Out	open-drain with internal pullup 20..50k IN (Power-On) L: Testmode OUT L: data in sendbuffer	40	C	LED backlight- / internal connection



I²C-BUSINTERFACE

Eine Beschaltung des Displays wie unten ermöglicht den direkten Betrieb an einem I²C-Bus.

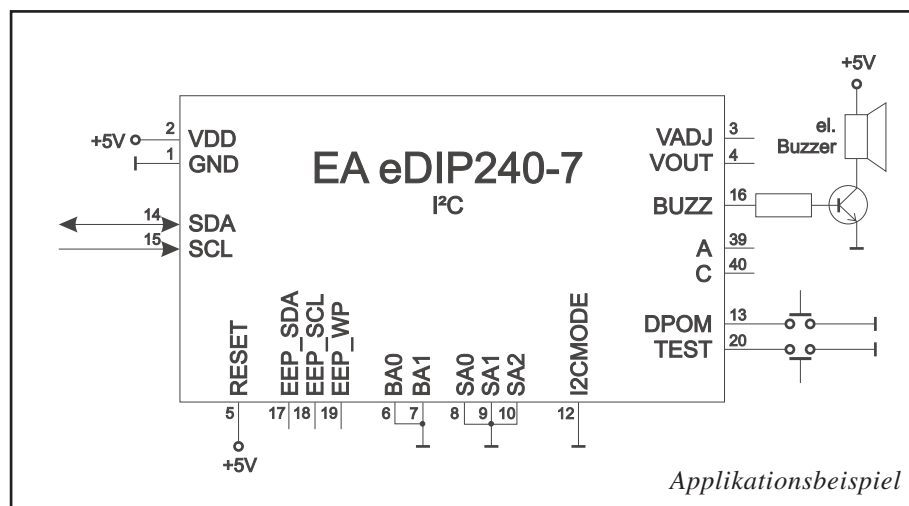
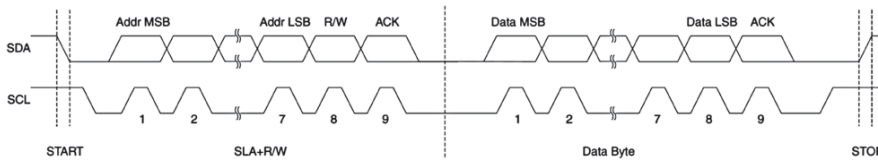
Am Display kann zwischen 4 unterschiedlichen Basisadressen und 8 verschiedenen Slave-Adressen ausgewählt werden.

Eine Datenübertragung ist bis zu 100kHz möglich.

Pinout eDIP240-7						
I ² C-Bus mode						
Pin	Symbol	In/Out	Funktion	Pin	Symbol	Function
1	GND	-	Ground Potential for logic (0V)	21	N.C.	not connected
2	VDD	-	Power supply for logic (+5V)	22	N.C.	not connected
3	VADJ	In	Operating voltage for LC driving (input)	23	N.C.	not connected
4	VOUT	Out	Output voltage for LC driving	24	N.C.	not connected
5	RESET	-	L: Reset	25	N.C.	not connected
6	BA0	In	Basic Adress 0	26	N.C.	not connected
7	BA1	In	Basic Adress 1	27	N.C.	not connected
8	SA0	In	Slave Adress 0	28	N.C.	not connected
9	SA1	In	Slave Adress 1	29	N.C.	not connected
10	SA2	In	Slave Adress 2	30	N.C.	not connected
11	N.C.	-	do not connect, reserved	31	N.C.	not connected
12	I2CMODE	In	connect to GND for I ² C interface	32	N.C.	not connected
13	DPOM	In	L: disable Power-On-Macro do not connect for normal operation	33	N.C.	not connected
14	SDA	Bidir.	Serial Data Line	34	N.C.	not connected
15	SCL	In	Serial Clock Line	35	N.C.	not connected
16	BUZZ	Out	Buzzer output	36	N.C.	not connected
17	EEP_SDA	Bidir.	Serial Data Line for int. EEPROM	37	N.C.	not connected
18	EEP_SCL	In	Serial Clock Line for int. EEPROM	38	N.C.	not connected
19	EEP_WP	In	H: Write Protect for int. EEPROM	39	A	LED backlight+ / internal connection
20	TEST SBUF	IN Out	open-drain with internal pullup 20..50k IN (Power-On) L: Testmode OUT L: data in sendbuffer	40	C	LED backlight- / internal connection

I ² C - Address										
BA1	BA0	Basic Address [HEX]	I ² C-Address [BIN]							
0	0	\$70	0	1	1	1				
0	1	\$90	1	0	0	1	S	S	S	R
1	0	\$B0	1	0	1	1	A	A	A	W
1	1	\$D0	1	1	0	1	2	1	0	

DATENÜBERTRAGUNG



DATENÜBERTRAGUNGSPROTOKOLL (SMALL PROTOKOLL)

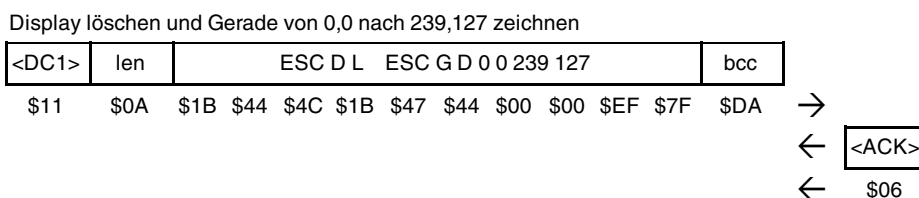
Die Datenübertragung ist eingebettet in einen festen Rahmen mit Prüfsumme (Protokollpaket). Das EA eDIP240-7 quittiert dieses Paket mit dem Zeichen <ACK> (= \$06) bei erfolgreichem Empfang oder <NAK> (= \$15) bei fehlerhafter Prüfsumme (Paket wird verworfen und muss nochmal gesendet werden).

Wird keine Quittierung gesendet, so ist mind. ein Byte verloren gegangen. Wenn nicht innerhalb des eingestellten Timeouts von 2 Sekunden die noch fehlenden Bytes empfangen werden, wird das komplette Paket verworfen und muss nochmals übertragen werden. Die Datenlänge (len) eines Paketes ist auf max. 64 Byte begrenzt.

Das Protokoll ist für alle drei Schnittstellen RS-232, I²C und SPI identisch. Für Tests kann das Protokoll durch Schließen der Lötbrücke J2 (siehe Seite 20) abgeschaltet werden. Im normalen Betrieb ist allerdings die Aktivierung des Protokolls unbedingt zu empfehlen. Andernfalls wäre ein möglicher Überlauf des Empfangspuffers nicht zu erkennen.

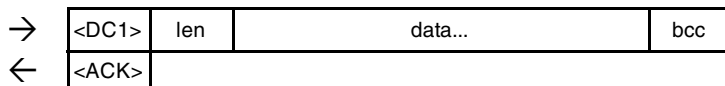
BEISPIEL

Das nachfolgende Beispiel zeigt ein vollständiges Protokollpaket zum Senden von Befehlen



DIE 5 PAKETVARIANTEN IN EINZELNEN

Befehle/Daten zum Display senden



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

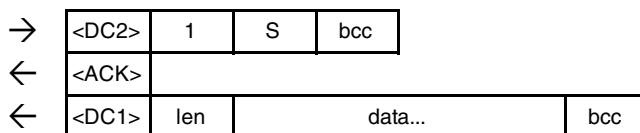
<DC1> = 17(dez.) = \$11

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

bcc = 1 Byte = Summe aus allen Bytes inkl. <DC1> und len, Modulo 8

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

Inhalt des Sendepuffers anfordern



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.

<DC2> = 18(dez.) = \$12

<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 8

Letztes Datenpaket wiederholen

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

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 Pufferinformation (<DC2>) sein.

<DC2> = 18(dez.) = \$12

bcc = 1 Byte = Summe aus allen Bytes inkl. <DC2> und len, Modulo 8

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

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

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

Pufferinformationen anfordern

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

Mit diesem Befehl wird abgefragt, ob Nutzdaten zur Abholung bereit stehen und wie voll der Empfangspuffer des Displays bereits ist. Die Daten selbst werden nur mit dem Befehl "INHALT DES SENDEPuffers ANFORDERN" (siehe oben) übertragen.

<DC2> = 18(dez.) = \$12

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

send buffer bytes ready = Anzahl abholbereiter Bytes

receive buffer bytes free = verfügbarer Platz im Empfangspuffer

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

bcc = 1 Byte = Summe aus allen Bytes inkl. <DC2> und len, Modulo 8

Protokolleinstellungen

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

Hierüber läßt sich die maximale Paketgröße welche das Display senden darf begrenzen. Voreingestellt ist eine Paketgröße mit bis zu 64 Byte Nutzdaten. Weiterhin läßt sich der Timeout (Voreinstellung 2 Sekunden) in 1/100s einstellen.

<DC2> = 18(dez.) = \$12

packet size = 1..64 (Standard: 64)

timeout = 0..255 in 1/100 Sekunden (Standard: 200 = 2 Sekunden)

bcc = 1 Byte = Summe aus allen Bytes inkl. <DC2> und len, Modulo 8

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

TOUCH PANEL (NUR EADIP240x-7LWTP)

Die Versionen EA DIP240B-7LWTP und J-7LWTP werden mit einem analogen resistiven Touchpanel geliefert. Bis zu 60 Touchbereiche (Tasten, Schalter, Menüs, Bargraph eingaben), können gleichzeitig definiert werden. Das Display unterstützt dieses Touch Panel mit komfortablen Befehlen (Seite ???). Beim Berühren der Touch-Tasten können diese automatisch invertiert werden und ein externer Summer (Pin 16) signalisiert die Berührung. Der 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.

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 *Rechtunten* 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).

BITMAPS ALSTASTEN

Ausser den Rahmentypen, die in der Grösse frei skalierbar sind, gibt es noch die Möglichkeit beliebige Bitmaps als Touch-Tasten oder -Schalter zu verwenden.

Über den Kitcompiler 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.

ELECTRONIC ASSEMBLY

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 4-fach vergrößert werden.

Jeder Text läßt sich linksbündig, rechtsbündig und zentriert ausgeben. Auch eine 90° Drehung, z.B. für vertikalen Einbau des Displays, ist möglich.

Die Makroprogrammierung erlaubt die Einbindung von weiteren Fonts (max. 15 Fonts). Es können alle nur erdenklichen Schriften mit einem Texteditor erstellt und über den Kitcompiler einprogrammiert werden (Programmer EA 9777-1 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	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	á	ä	â	ç	ê	ë	è	í	î	ï	ñ	ñ	
\$90 (dez: 144)	É	æ	Æ	ô	ö	ò	û	ü	Û	Ü	φ	£	¥	β	ƒ	
\$A0 (dez: 160)	á	í	ó	ú	ñ	ñ	ã	ø	¿	¡	½	¼	¡	«	»	
\$B0 (dez: 176)																
\$C0 (dez: 192)																
\$D0 (dez: 208)																
\$E0 (dez: 224)	α	β	Γ	Π	Σ	σ	μ	τ	ξ	θ	Ω	ε	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	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	á	ä	â	ç	ê	ë	è	í	î	ï	ñ	ñ	
\$90 (dez: 144)	É	æ	Æ	ô	ö	ò	û	ü	Û	Ü	φ	£	¥	β	ƒ	
\$A0 (dez: 160)	á	í	ó	ú	ñ	ñ	ã	ø	¿	¡	½	¼	¡	«	»	
\$B0 (dez: 176)																
\$C0 (dez: 192)																
\$D0 (dez: 208)																
\$E0 (dez: 224)	α	β	Γ	Π	Σ	σ	μ	τ	ξ	θ	Ω	ε	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	á	ä	â	ç	ê	ë	è	í	î	ï	ñ	ñ	
\$90 (dez: 144)	É	æ	Æ	ô	ö	ò	û	ü	Û	Ü	φ	£	¥	β	ƒ	
\$A0 (dez: 160)	á	í	ó	ú	ñ	ñ	ã	ø	¿	¡	½	¼	¡	«	»	
\$B0 (dez: 176)																
\$C0 (dez: 192)																
\$D0 (dez: 208)																
\$E0 (dez: 224)	α	β	Γ	Π	Σ	σ	μ	τ	ξ	θ	Ω	ε	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	á	ä	â	ç	ê	ë	è	í	î	ï	ñ	ñ	
\$90 (dez: 144)	É	æ	Æ	ô	ö	ò	û	ü	Û	Ü						
\$A0 (dez: 160)	á	í	ó	ú	ñ	ñ	ã	ø								
\$B0 (dez: 176)																
\$C0 (dez: 192)																
\$D0 (dez: 208)																
\$E0 (dez: 224)		β														
\$F0 (dez: 240)									°							

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	ã	ç	ê	ë	è	ï	î	í	Ä	Å
\$90 (dez: 144)	É	æ	Æ	ô	ö	ò	û	ù	ÿ	ö	Ü					
\$A0 (dez: 160)	á	í	ó	ú	ñ	Ñ	ä	ö								
\$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	ã	ç	ê	ë	è	ï	î	í	Ä	Å
\$90 (dez: 144)	É	æ	Æ	ô	ö	ò	û	ù	ÿ	ö	Ü					
\$A0 (dez: 160)	á	í	ó	ú	ñ	Ñ	ä	ö								
\$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

Das Hardcopy eines Bildinhaltes zeigt alle Standard Schriften.

Die Makroprogrammierung erlaubt die Einbindung von weiteren Fonts. Es können alle nur erdenklichen Schriften mit einem Texteditor erstellt und über den Kitcompiler einprogrammiert werden (Programmer EA 9777-1 notwendig).



GRUNDEINSTELLUNGEN

Nach dem Einschalten bzw. nach einem Reset werden einige Funktionen auf einen bestimmten Wert voreingestellt (siehe letzte Spalte 'nach Reset' in der Tabelle). Beachten Sie bitte, daß alle Einstellungen durch Erstellen eines Power-On-Makros überschrieben werden können.

ALLE BEFEHLE AUF EINEN BLICK

EA eDIP240-7: Befehlstabelle 1							nach Reset			
Befehl	Codes			Anmerkung						
Befehle für den Terminal Betrieb										
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 auf 1. Zeile setzen						
Cursor positionieren	ESC	T	P	n1	n2	n1=Spalte; n2=Zeile; Ursprung links oben ist (1,1)	1,1			
Cursor On / Off			C	n1		n1=0: Cursor ist unsichtbar; n1=1: Cursor blinkt;	1			
Terminal AUS	ESC	T	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 eDIP240-7 V1.0 Rev.B"				
Befehle zur Ausgabe von Zeichenketten										
Zeichenkette ausgeben	ESC	Z	L	x1	y1	Text NUL	Eine Zeichenkette (...) an x1,y1 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;			
L: Linksbündig			C							
C: Zentriert			R							
R: Rechtsbündig			F	n1			Font mit der Nummer n1 (0..16) einstellen	0		
Font einstellen			Z	n1	n2		n1 = X-Zoomfaktor (1x..4x); n2 = Y-Zoomfaktor (1x..4x)	1,1		
Font-Zoomfaktor			Y	n1			zwischen zwei Textzeilen n1 Pixel als zusätzlichen Zeilenabstand einfügen			
Font-Zusatzabstand			W	n1			Text-Ausgabewinkel: n1=0: 0°; n1=1: 90°;	0		
Text-Winkel			V	n1			Modus n1: 1=setzen; 2=löschen; 3=iners; 4=Replace; 5=Invers Replace;	4		
Text-Verknüpfungsmodus	B	n1			n1: 0=blinken Aus; 1=Text blinkt An/Aus; 2=Text blinkt Invertierend;	0				
Text-Blinkattribut	ESC	Z	T	Text ...			Befehl um eine Zeichenkette in einem Makro an das Terminal ausgeben zu können			
Zeichenkette für Terminal	Geraden und Punkte zeichnen									
Rechteck zeichnen	ESC	G	R	x1	y1	x2	y2	Vier Geraden als Rechteck von x1,y1 nach x2,y2 zeichnen		
Gerade zeichnen			D	x1	y1	x2	y2	Eine Gerade von x1,y1 nach x2,y2 zeichnen		
Gerade weiter zeichnen			W	x1	y1			Eine Gerade vom letzten Endpunkt bis x1, y1 zeichnen	0	
Punkt zeichnen			P	x1	y1			Ein Punkt an die Koordinaten x1, y1 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	
Rechteckige Bereiche verändern / zeichnen										
Bereich löschen	ESC	R	L	x1	y1	x2	y2	Einen Bereich von x1,y1 nach x2,y2 löschen (alle Pixel aus)		
Bereich invertieren			I	x1	y1	x2	y2	Einen Bereich von x1,y1 nach x2,y2 invertieren (alle Pixel umkehren)		
Bereich füllen			S	x1	y1	x2	y2	Einen Bereich von x1,y1 nach x2,y2 füllen (alle Pixel ein)		
Bereich m. Füllmuster			M	x1	y1	x2	y2	n1	Einen Bereich von x1,y1 nach x2,y2 mit Muster n1 zeichnen (immer setzen)	
Box zeichnen			O	x1	y1	x2	y2	n1	Ein Rechteck von x1,y1 nach x2,y2 mit Muster n1 zeichnen; (immer Replace)	
Rahmen zeichnen			R	x1	y1	x2	y2	n1	Einen Rahmen Typ n1 von x1,y1 nach x2,y2 zeichnen (immer setzen)	
Rahmenbox zeichnen			T	x1	y1	x2	y2	n1	Eine Rahmenbox Typ n1 von x1,y1 nach x2,y2 zeichnen; (immer Replace)	
Bitmap Bilder Befehle										
Bild aus Clipboard	ESC	U	C	x1	y1			Der akt. Clipboardinhalt wird mit allen Bildattributen nach x1,y1 geladen		
internes Bild laden			I	x1	y1	nr		internes Bild mit der nr (0..255) aus dem EEPROM nach x1,y1 laden		
Bild laden			L	x1	y1	BLH	daten ...		Ein Bild nach x1,y1 laden; daten des Bildes siehe Bildaufbau	
Bild-Zoomfaktor			Z	n1	n2			n1 = X-Zoomfaktor (1x..4x); n2 = Y-Zoomfaktor (1x..4x)	1,1	
Bild-Winkel			W	n1				Ausgabewinkel: n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°;	0	
Bild-Verknüpfungsmodus			V	n1				Modus n1: 1=setzen; 2=löschen; 3=iners; 4=Replace; 5=Invers Replace;	4	
Bild-Blinkattribut			B	n1				n1=0 Bild-Attribut blinken Aus; n1=1 Bild blinkt An/Aus; n1=2 Bild blinkt Invertierend;	0	
Hardcopy senden			H	x1	y1	x2	y2		Es wird ein Bild angefordert. Zuerst werden die Breite und Höhe in Pixel und dann die eigentlichen Bilddaten über RS232 gesendet.	
Display-Befehle (Wirkung auf das gesamte Display)										
Display löschen	ESC	D	L			Displayinhalt löschen (alle Pixel aus)				
Display invertieren			I			Displayinhalt invertieren (alle Pixel umkehren)				
Display füllen			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	
Display Clipboard			C			Inhalt des Clipboards wird dargestellt. Displayausgaben sind nicht mehr sichtbar				
Disp. Normaldarstellung			N			Aktuelles Bild wird dargestellt (Normalbetrieb). Alle Ausgaben wieder sichtbar				

EA eDIP240-7: Befehlstabelle 2										nach			
Befehl	Codes		Anmerkung							Reset			
Bargraph Befehle													
Bargraph definieren	ESC B	R L O U	n1	x1	y1	x2	y2	aw	ew	typ	mst	Bargraph nach L(inks), R(rechts), O(ben), U(nten) mit der Nr. n1 definieren. x1,y1,x2,y2 sind das umschließende Rechteck des Bargraphs. aw,ew sind die Werte für 0% und 100%. typ=0: Balken; typ=1: Balken im Rechteck; mst=Balkenmuster; typ=2: Strich; typ=3: Strich im Rechteck; mst= Strichbreite	Kein Bar definiert
Bargraph aktualisieren		A	n1	wert	Bargraph 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 auf der serilen Schnittstelle senden									
Bargraph löschen		D	n1	n2	Die Definition des Bargraph 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								
Clipboard Befehle (Zwischenspeicher für Bildbereiche)													
Displayinhalt sichern	ESC C	B	Der gesamte Displayinhalt wird als Bildbereich ins Clipboard kopiert										
Bereich sichern		S	x1	y1	x2	y2	Der Bildbereich von x1,y1 bis nach x2,y2 wird ins Clipboard kopiert						
Bereich restaurieren		R	Der Bildbereich im Clipboard wird wieder ins Display kopiert										
Bereich kopieren		K	x1	y1	Der Bildbereich im Clipboard wird ins Display nach x1,y1 kopiert								
Einstellungen für Menübox / Touchmenü													
Menü-Font einstellen	ESC N	F	n1	Font mit der Nummer n1 (0..16) für Menüdarstellung einstellen							0		
Menüfont-Zoomfaktor		Z	n1	n2	n1 = X-Zoomfaktor (1x..4x); n2 = Y-Zoomfaktor (1x..4x)							1,1	
zus. Zeilenabstand		Y	n1	zwischen zwei Menüeinträgen n1 Pixel als zusätzlichen Zeilenabstand einfügen									
Menü-Winkel		W	n1	Menüdarstellung Winkel: n1=0: 0°; n1=1: 90°;							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 über RS-232/422 an den Hostrechner gesendet, dieser kann dann mit 'ESC N T 2' das Touchmenü öffnen.							1		
Menübox Befehle (Steuerung mit Tasten nicht per Touch)													
Menü definieren und Darstellen	ESC N	D	x1	y1	nr	Text ...	NUL	Ein Menü wird ab der Ecke x1,y1 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 das 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										
Makro Befehle													
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)									
automatische (Normal-) Makros													
Makro mit Verzögerung	ESC M	G	n1	n2	Das (Normal-)Makro mit der Nummer n1 (0..255) in n2/10s aufrufen. Die 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. Die 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. Die 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. Die Ausführung wird durch Befehle (z.B durch Empfang oder Touchmakros) gestoppt.							
Blinkbereichs-Befehle													
Blinkattribut löschen	ESC Q	L	x1	y1	x2	y2	Löscht das Blinkattribut von x1,y1 bis x2,y2						
Invertierender Blinkbereich		I	x1	y1	x2	y2	Definiert einen invertierenden Blinkbereich von x1,y1 bis x2,y2						
Muster Blinkbereich		M	x1	y1	x2	y2	Definiert einen Blinkbereich mit Muster (An/Aus) von x1,y1 bis x2,y2						
Blinkzeit einstellen		Z	n1	Einstellen der Blinkzeit n1= 1..15 in 1/10s; 0=Blinkfunktion deaktivieren							6		
Sonstige-Befehle													
Warten (Pause)	ESC X	n1	n1 Zehntel-Sekunden abwarten bevor der nächste Befehl ausgeführt wird.										
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%							100		
Bytes senden	ESC S	B	anz	daten ...	Es werden anz (=1..255) Bytes zum Sendepuffer gesendet daten ... = anz Bytes								
Version senden		V	Es wird die Versions-Nr. als String gesendet z.B "EA eDIP240-7 V1.0 Rev.B"										

EA eDIP240-7: Befehle für das Touch-Panel											nach			
Befehl	Codes							Anmerkung			Reset			
Touch: Bereiche definieren														
Touch-Taste definieren (Taste ist gedrückt solange der Touch berührt wird)	ESC	A	T	x1	y1	x2	y2	dow Cod	up Cod	Text ...	NUL	'T': Der Bereich von x1,y1 nach x2,y2 wird als Taste definiert. 'U': Das Bild Nr. n1 wird nach x1,y2 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': es folgt eine Zeichenkette die zentriert mit dem akt. Touch-Font in der Touch-Taste plaziert wird, mehrzeilige Texte werden mit dem Zeichen ' ' (\$7C, dez: 124) getrennt; 'NUL': (\$00) = Zeichenkettenende		
			U	x1	y1	n1	dow Cod	up Cod	Text ...	NUL				
Touch-Schalter definieren (Zustand der Schalter toggelt nach jeder Berührung)	ESC	A	K	x1	y1	x2	y2	dow Cod	up Cod	Text ...	NUL	'K': Der Bereich von x1,y1 nach x2,y2 wird als Schalter definiert. 'J': Das Bild n1 wird nach x1,y2 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': es folgt eine Zeichenkette die zentriert mit dem akt. Touch-Font in der Touch-Taste plaziert wird, mehrzeilige Texte werden mit dem Zeichen ' ' (\$7C, dez: 124) getrennt; 'NUL': (\$00) = Zeichenkettenende		
			J	x1	y1	n1	dow Cod	up Cod	Text ...	NUL				
Touch-Taste mit Menüfunktion definieren	ESC	A	M	x1	y1	x2	y2	dow Cod	up Cod	mnu Cod	Text ...	NUL	Der Bereich x1,y1 nach x2,y2 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/Menu Makro+(EintragsNr-1) nach Auswahl eines Menü-Eintrages. (down-/up-Code = 0 Aktivieren / Abbruch des Menüs wird nicht gemeldet). 'Text':= Zeichenkette mit dem Menü-Tastentext und den Menüeinträgen. Das erste Zeichen bestimmt die Richtung in der das Menü aufklappt (R=rechts L=links O=oben U=Unten). Die einzelnen Einträge sind durch Zeichen ' ' (\$7C, dez: 124) getrennt. z.B. "UTaste 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	x1	y1	x2	y2	n1				Ein Zeichenbereich wird definiert. Innerhalb der Eck-Koodinaten x1,y1 und x2,y2 kann dann mit der Strichstärke n1 gezeichnet werden.		
Freien Touchbereich def.	ESC	A	H	x1	y1	x2	y2				Ein frei benutzbarer Touchbereich wird definiert. Touchaktionen (down, up und drag) innerhalb der Eck-Koodinaten x1,y1 und x2,y2 werden über RS232 gesendet.			
Bar per Touch einstellbar	ESC	A	B	nr							Der Bargraph mit der Nr. n1 wird zur Eingabe per Touchpanel definiert.			
Touch: Einstellungen														
Touch-Rahmen Form			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		
			S	n1							Summer piepst kurz beim Berühren einer Touch-Taste: n1=0=AUS; n1=1=EIN	1		
Touch-Taste Invertieren			N	Cod							Die Touch-Taste mit dem zugeordnetem Return-Code wird manuell Invertiert			
Touch-Schalter abfragen			X	Cod							Zustand des Schalters (Aus=0; Ein=1) wird über die serielle Schnittstelle gesendet			
Touch-Schalter einstellen			P	Cod	n1						Zustand des Schalters wird per Befehl geändert n1=0=Aus; n1=1=Ein.			
Radiogroup für Schalter	ESC	A	R	n1							Innerhalb einer Gruppe ist immer nur 1 Schalter aktiv, alle anderen werden deaktiviert n1=0: neu definierte Schalter gehören keiner Gruppe an. n1=1..255: neu definierte Schalter gehören der Gruppe mit der Nummer n1 an.	0		
Touch-Bereich Löschen			L	Cod	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 mit Display-Hintergrundfarbe gelöscht.			
			V	x1	y1	n1					Touchbereich der die Koordinaten x1,y1 umschließt aus der Touchabfrage entfernen n1=0: Bereich bleibt sichtbar; n1=1: Bereich mit Display-Hintergrundfarbe löschen			
Barwert senden Ein/Aus			Q	n1							das automatische Senden eines neuen Bargraphwertes per Toucheingabe wird n1=0:deaktiviert; n1=1:aktiviert;	1		
Touch-Abfrage Ein/Aus			A	n1							Touchabfrage wird n1=0:deaktiviert; n1=1:aktiviert;	1		
Touch: Beschriftungs-Font														
Beschriftungs Font			F	n1							Font mit der Nummer n1 (0..16) für Touchtastenbeschriftung einstellen	0		
Beschriftungs-Zoomfaktor			Z	n1	n2						n1 = X-Zoomfaktor (1x..4x); n2 = Y-Zoomfaktor (1x..4x)	1,1		
zus. Zeilenabstand	ESC	A	Y	n1							zwischen zwei Textzeilen n1 Pixel als zusätzlichen Zeilenabstand einfügen			
Beschriftungs-Winkel			W	n1							Text-Ausgabewinkel: n1=0: 0°; n1=1: 90°;	0		

Antworten des EA eDIP240-7 über die serielle Schnittstelle						
Kennung	anz	daten			Anmerkung	
automatische Antworten						
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 Menücode 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 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 n1'), so wird diese Anforderung an den Hostrechner gesendet. Dieser kann dann das Touchmenü mit dem Befehl 'ESC N T 2' öffnen.	
ESC	H	3	typ	x1 y1	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 x1,y1	
Antworten nur nach Anforderung per Befehl						
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' wird der aktuelle Zustand des Touch-Schalters mit dem Return-Code code gesendet. wert = 0 oder 1	
ESC	V	anz	Zeichenkette...		Nach dem Befehl 'ESC S V' wird die Version der eDIP-Firmware als Zeichenkette gesendet.	
Antworten ohne Längenangabe (anz)						
ESC	U	L	x1	y1	BLH-Bilddaten...	Nach dem Befehl 'ESC UH...' wird ein Hardcopy gesendet. x1,y1 = Startkoordinaten des Hardcopies (Linke obere Ecke) BLH-Bilddaten: 2 Byte: breite, höhe (in Pixel) + anzahl Bytes Bilddaten anzahl = ((breite+7)/8)*höhe

Hinweis:

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.

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,'#'). 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 und der Cursor nach links oben positioniert.

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 (die Terminalebene wird mit 'FF' gelöscht).

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

+ Lower	\$0 (0)	\$1 (1)	\$2 (2)	\$3 (3)	\$4 (4)	\$5 (5)	\$6 (6)	\$7 (7)	\$8 (8)	\$9 (9)	\$A (10)	\$B (11)	\$C (12)	\$D (13)	\$E (14)	\$F (15)
Upper		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
\$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	û	ä	à	á	ç	ê	ë	è	í	î	ï	ä	å
\$90 (dez: 144)	é	æ	œ	ö	ö	ò	ü	ù	ü	ö	ü	ç	£	¥	β	f
\$A0 (dez: 160)	á	í	ó	ú	ñ	ñ	ë	ó	ç	í	í	½	¼	í	«	»
\$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	ε	π	
\$F0 (dez: 240)	≡	±	≥	≤	ρ	ρ	÷	≈	°	*	.	√	n	z	ε	-

Terminal-Font (Font 0): 8x8 monospaced

Technische Änderungen vorbehalten. Wir übernehmen keine Haftung für Druckfehler.

ELECTRONIC ASSEMBLY**BEFEHLSÜBERGABE/PARAMETER**

Das eDIP240-7 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.
- Zeichenketten (Texte) werden direkt ohne Anführungsstrichen geschrieben und mit CR (hex: \$0D), oder LF (hex: \$0A) abgeschlossen.

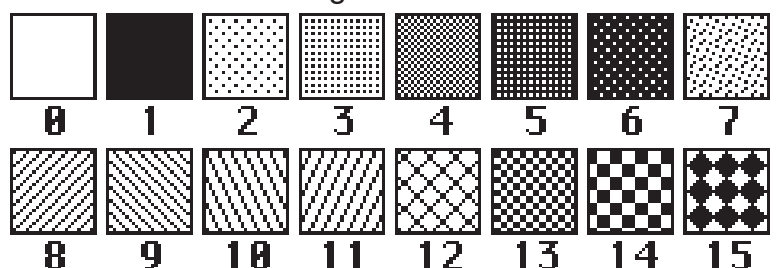
2. Binär-Modus

- Das Escape-Zeichen entspricht dem Zeichen ESC (hex: \$1B, dez: 27).
- Die Befehlsbuchstaben werden direkt gesendet.
- Die Koodinaten x, y und 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).

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.



MAKRO PROGRAMMIERUNG

Einzelne oder mehrere Befehlsfolgen können als sog. Makros zusammengefasst und im EEPROM fest abgespeichert werden. Diese können dann mit den Befehlen *Makro ausführung* gestartet werden. Es gibt verschiedene Makrotypen:

Normal Makro (0..255)

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 RS-232 empfangen wird, oder ein anderes Makro mit entsprechendem Return-Code ausgelöst wird (z.B. Touch-, Port-, Matrixmakro).

Touch Makro (1..255)

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

Menü Makro (1..255)

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

Power-On-Makro

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

Reset-Makro

Start nach einem externen Reset oder nach einem Spannungseinbruch unter 4,7V (VDD-VSS).

Watchdog-Makro

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

Brown-Out-Makro

Start nach einem Spannungseinbruch <4V.

Achtung: Wird im Power-On-, Reset- oder Watchdog-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 DPOM:

PowerOff - Pin 13 (DPOM) auf GND legen - PowerOn -Pin 13 wieder öffnen.

SCHREIBSCHUTZ FÜR MAKROPROGRAMMIERUNG

Ein VDD-Pegel am Pin 19 (EEP_WP) verhindert ein versehentliches Überschreiben der Makros, Bilder und Fonts im EEPROM.

ELECTRONIC ASSEMBLY

BILDER IM EEPROM ABGELEGT

Um die Übertragungszeiten der Schnittstelle zu verkürzen, oder auch um Speicherplatz im Prozessorsystem zu sparen, können bis zu 256 Bilder im internen EEPROM abgelegt werden. Der Aufruf erfolgt über den Befehl "ESC U 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 oder Photoshop (nur schwarz/weiss = 1 Bit).

ERSTELLEN INDIVIDUELLER MAKROS UND BILDER

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

- den als Zubehör erhältlichen Programmierer EA 9777-1
- die Software ELECTRONIC ASSMBLY LCD-Tools^{*)}; sie enthält einen Editor, Compiler, Simulator, sowie Beispiele und Fonts (für PC-Win)
- einen PC mit serieller Schnittstelle COM1 oder COM2

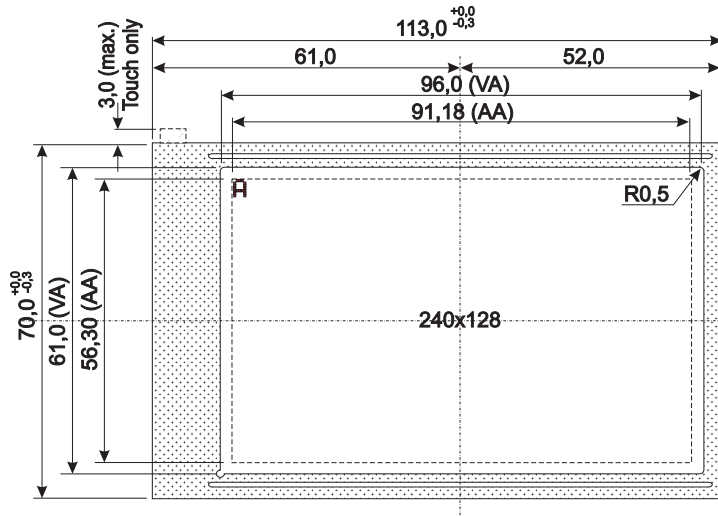
Um eine Befehlsfolge als Makro zu definieren, werden alle Befehle auf dem PC in eine Datei z.B. DEMO.KMC geschrieben. Hier bestimmen Sie, welche Zeichensätze eingebunden werden und in welchen Makros welche Befehlsfolgen stehen sollen.

Sind die Makros definiert, startet man den Compiler. Dieser erzeugt eine Datei DEMO.EEP, welcher das Ergebnis in einem Simulatorfenster (virtuelles Display) sofort anzeigt. Ist auch ein Programmierer EA 9777-1 angeschlossen, dann wird diese Datei automatisch in das EEPROM des Displays gebrannt. Dieser Vorgang dauert nur wenige Sekunden und sofort danach können die selbstdefinierten Makros und Bilder auch im Display genutzt werden. Eine ausführliche Beschreibung zur Programmierung der Makros finden Sie zusammen mit Beispielen in der Hilfefunktion der LCD-Tool Software.

^{*)} auch im Internet unter <http://www.lcd-module.de/deu/touch/touch.htm>

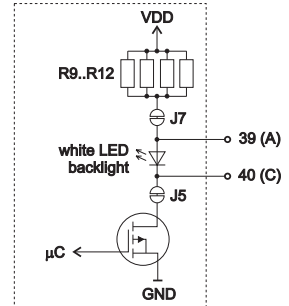
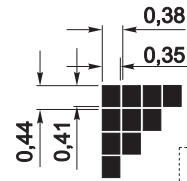
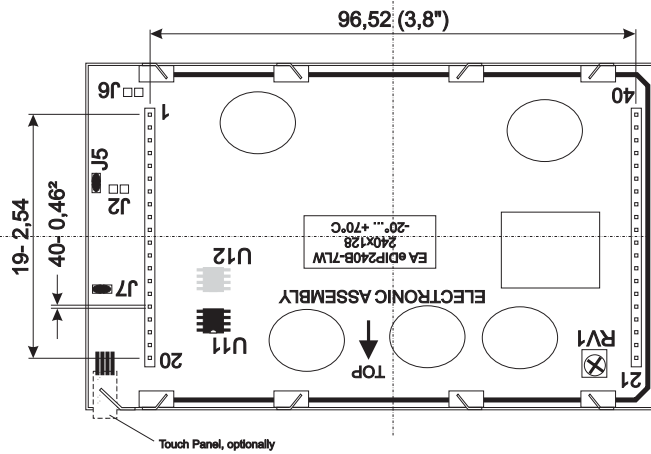
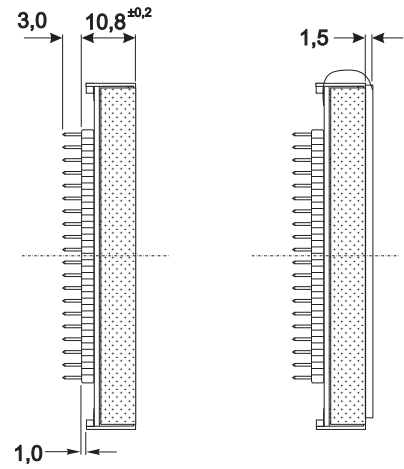
EA eDIP240-7

ABMESSUNGEN



ohne Touchpanel

mit Touchpanel



alle Maße in mm