embedded 4,3" TFT-DISPLAY 480x272 MIT INTELLIGENZ

Stand 6.2011

WELTNEUHEIT !

Abmessung: 107x71x12mm

TECHNISCHE DATEN

* TFT-GRAFIKDISPLAY MIT GRAFIKFUNKTIONEN
* 480x272 PIXEL, 16-BIT COLOR (65.536 FARBEN) MIT LED-BELEUCHTUNG
* GEDREHTER EINBAU 272x480 PIXEL (ORIENTIERUNG PER BEFEHL UMSCHALTBAR)
* 4MB ONBOARD FLASH FÜR FONTS, BILDER, ANIMATIONEN UND MAKROS
* VERSORGUNG +5V / 180mA
* 8 VORDEFINIERTE FONTS, INDIVIDUELL ANPASSBAR
* FONT ZOOM VON ca. 2mm BIS zu ca. 80mm, in 90° SCHRITTEN DREHBAR
* 3 VERSCHIEDENE INTERFACE ONBOARD: RS-232, I²C-BUS ODER SPI-BUS
* PIXELGENAUE POSITIONIERUNG BEI ALLEN FUNKTIONEN
* GERADE, PUNKT, BEREICH, BARGRAPH...
* BILDER UND ANIMATIONEN
* TEXT UND GRAFIK MISCHEN
* MEHRSPRACHIKEIT DURCH MAKROPAGES
* BELEUCHTUNG PER SOFTWARE REGELBAR
* ANALOGES TOUCH PANEL: VARIABLES RASTER
* FREI DEFINIERBARE TASTEN UND SCHALTER
* 8 DIGITALE EIN- UND 8 DIGITALE AUSGÄNGE
* ZWEI ANALOGEINGÄNGE KOMFORTABEL PROGRAMMIERBAR

BESTELLBEZEICHNUNG

TFT 480x272 DOTS, WEISSE LED-BELEUCHTUNG	**EA eDIPTFT43-A**
WIE VOR, JEDOCH MIT TOUCH PANEL	**EA eDIPTFT43-ATP**
EINBAUBLENDE SCHWARZ, ELOXIERTES ALUMINIUM	**EA 0FP481-43SW**
PROGRAMMER FÜR USB INKL. KABEL, CD FÜR WIN98/ME/2000/XP	**EA 9777-1USB**
BUCHSENLEISTE 1x20, 4.5mm HOCH (1 STÜCK)	**EA B254-20**
STARTERKIT: 1x EA eDIPTFT43-ATP INKL. TOUCH + EA9777-1USB	**EA STARTeDIPTFT4**

ELECTRONIC ASSEMBLY
making things easy

Zeppelinstr. 19 · D-82205 Gilching · Tel. 08105-778090 · Fax 08105-778099 · www.lcd-module.de · info@lcd-module.de

Technische Änderung vorbehalten.
Wir übernehmen keine Haftung für
Druckfehler und Applikationsbeispiele.

Documentation of revision				
Date	**Type**	**Old**	**New**	**Reason / Description**
July, 22nd. 2008	1.0			1st. Edition
March 2009	1.1		- additional command: ESC YD, ESC VM, ESC YX - bug fix	
June 2009	1.2		- additional command: ESC ZB - bug fix	
November 2009	1.3		- bug fix "bargraph" and "clear touch"	
May 2011	1.4		- additional functions: Instruments, extended Ports	

INHALT

ALLGEMEINES 3
RS-232, USB 4
SPI 6
I²C 7
ANALOG / DIGITAL EIN- UND AUSGÄNGE 8
MATRIXTASTATUR 9
SOFTWARE PROTOKOLL ..... 10 - 11
TERMINAL BETRIEB, BEFEHLSÜBERGABE 12
BEFEHLE / FUNKTIONEN IN TABELLENFORM ..... 13 - 17
TOUCH PANEL 18
RÜCKANTWORTEN DES BEDIENPANELS 19
ZEICHENSÄTZE 20-22
DARSTELLBARE FARBEN,FÜLLMUSTER 22
RAHMEN, TASTENFORMEN, BARGRAPH, FÜLLMUSTER 23-24
DREH- UND ZEIGERINSTRUMENTE 22
PROGRAMMIERUNG: FONTS, BILDER, ANIMATIONEN 26
BITMAPS ALS TASTEN 27
MAKROS, MEHRSPRACHIGKEIT, MAKROPAGES 28-29
ELEKTRISCHE SPEZIFIKATIONEN 30
ABMESSUNGEN, EINBAUBLENDE 31-32

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 I²C. "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 I²C Spezifikation. Zur Erhöhung der Datensicherheit wird für alle Übertragungsvarianten ein einfaches Protokoll verwendet.

ANALOGES TOUCH PANEL

Optional gibt es eine Version mit integrierten Touch Panel. 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

Die Displays sind mit einer modernen und stromsparenden LED-Beleuchtung ausgestattet. Die Helligkeit kann per Befehl von 0~100% variiert werden.
Im 24h Betrieb wie auch bei erhöhter Umgebungstemperatur sollte zur Verlängerung der Lebensdauer die Beleuchtung sooft als möglich gedimmt bzw. abgeschaltet werden.

SOFTWARE

Die Programmierung erfolgt über Befehle wie z.B. *Zeichne Rechteck von 0,0 nach 479,271*. Es ist keine zusätzliche Software oder Treiber erforderlich. Zeichenketten und Bilder lassen sich **pixelgenau** platzieren. Das Mischen von Text und Grafik ist jederzeit möglich. Es können mehrere Zeichensätze verwendet werden. Jeder Zeichensatz und die Bilder/Animationen 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/ Animationen oder Makros abgelegt werden, brennt der als Zubehör erhältliche USB-Programmer EA 9777-1USB die von Ihnen erstellten Daten/Bilder dauerhaft ins on-board DatenFlash (4MB).
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.

Technische Änderung vorbehalten.
Wir übernehmen keine Haftung für
Druckfehler und Applikationsbeispiele.

RS-232 INTERFACE

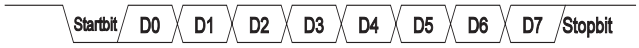
Wird das Display wie unten gezeigt beschaltet, so ist das RS-232 Interface ausgewählt. Die Pinbelegung ist in der Tabelle rechts angegeben.
Die Leitungen RxD und TxD führen 5V CMOS-Pegel zur direkten Anbindung an z.B. einen Mikrokontoller.

Pinout eDIPTFT43-A: RS-232/RS-485 mode

Pin	Symbol	In/Out	Function	Pin	Symbol	In/Out	Function
1	GND		Ground Potential for logic (0V)	21	GND		Ground (0V)
2	VDD		Power supply for logic (+5V)	22	VDD		Power supply (+5V)
3	NC		do not connect	23	AIN1	In	analogue input 0..5V
4	NC		do not connect	24	AIN2		DC impedance 1MOhm
5	RESET	In	L: Reset	25	OUT1 / MO8	Out	8 digital outputs maximum current: IOL = IOH = 10mA alternativ up to 8 matrix keyboard output lines (reduces the digital output lines, see chapter external keyboard)
6	BAUD0	In	Baud Rate 0	26	OUT2 / MO7		
7	BAUD1	In	Baud Rate 1	27	OUT3 / MO6		
8	BAUD2	In	Baud Rate 2	28	OUT4 / MO5		
9	ADR0	In	Address 0 for RS-485	29	OUT5 / MO4		
10	RxD	In	Receive Data	30	OUT6 / MO3		
11	TxD	Out	Transmit Data	31	OUT7 / MO2		
12	EN485	Out	Transmit Enable for RS-485 driver	32	OUT8 / MO1		
13	DPOM	In	L: disable PowerOnMacro do not connect for normal operation	33	IN1 / MI8	In	8 digital inputs open-drain with internal pullup 20..50k alternativ up to 8 matrix keyboard input lines (reduces the digital input lines, see chapter external keyboard)
14	ADR1	In	Address 1 for RS-485	34	IN2 / MI7		
15	ADR2	In	Address 2 for RS-485	35	IN3 / MI6		
16	BUZZ	Out	Buzzer output	36	IN4 / MI5		
17	DPROT	In	L: Disable Smallprotokoll do not connect for normal operation	37	IN5 / MI4		
18	DNC	Out	L: internal, do not connect	38	IN6 / MI3		
19	WP	In	L: Writeprotect for DataFlash	39	IN7 / MI2		
20	TEST SBUF	IN Out	open-drain with internal pullup 20..50k IN (Power-On) L: Testmode OUT L: data in sendbuffer	40	IN8 / MI1		

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.



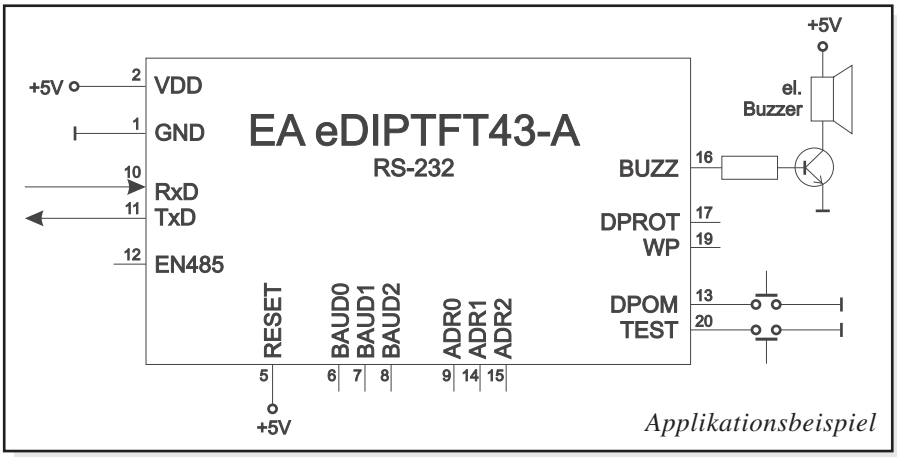
Handshakeleitungen RTS/CTS sind nicht erforderlich. Die notwendige Steuerung wird von dem eingebauten Software-Protokoll übernommen.

Baudraten

Baud0	Baud1	Baud2	Datenformat 8,N,1
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
0	0	0	230400

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)

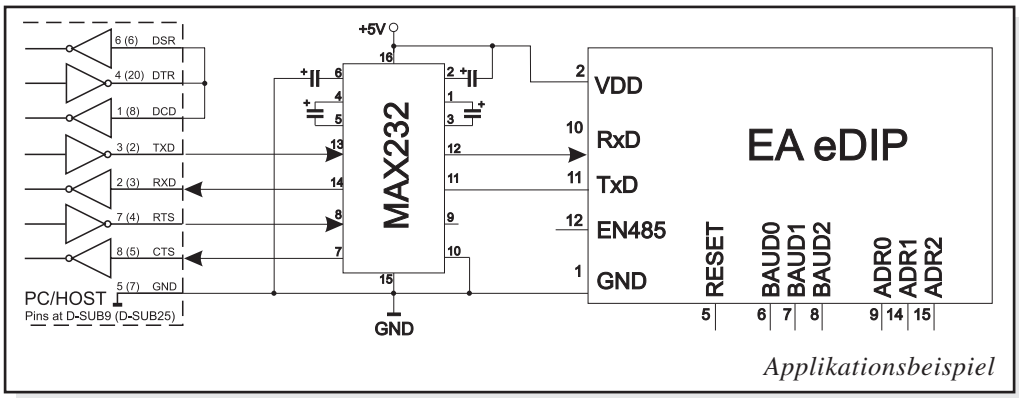


Applikationsbeispiel

Hinweis:
Die Pins BAUD0..2, ADR0..2, DPOM, DPROT 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.

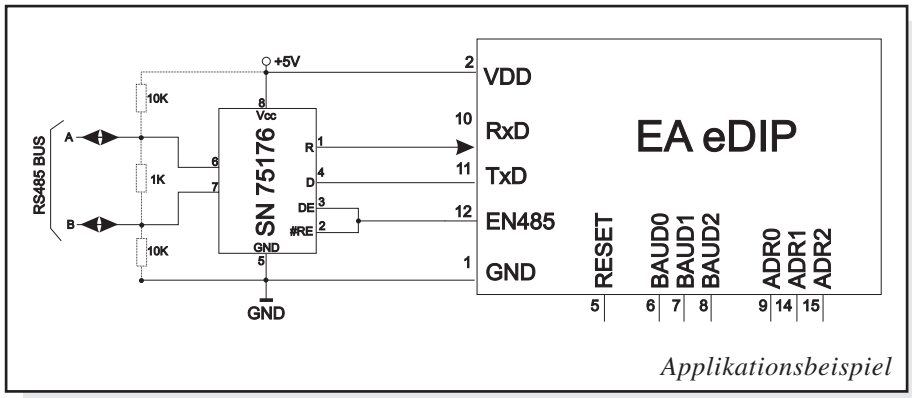
APPLIKATIONSBEISPIEL „ECHTES“ RS-232 INTERFACE

Das eDIP ist für den direkten Anschluss an eine RS-232 Schnittstelle mit 5V Pegeln geeignet.
Steht jedoch nur eine Schnittstelle mit ±12V Pegeln, so ist ein externer Pegelwandler erforderlich.



APPLIKATIONSBEISPIEL: RS-485 INTERFACE

Mit einem externen Umsetzer (z.B. SN75176) kann das eDIP an einen 2-Draht RS-485 Bus angeschlossen werden. Somit können grosse Entfernungen bis zu 1200m (Ferndisplay) realisiert werden. Betrieb von mehreren EA eDIPs an einem RS-485 Bus durch Einstellen von Adressen.

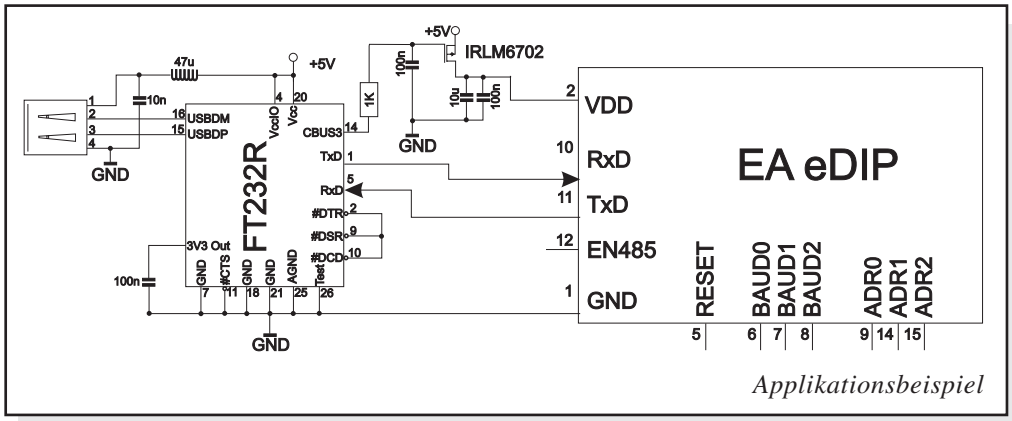


Adressierung:

- *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: USB ANSCHLUSS

Mit einem externen Umsetzer (z.B. FTZ232R) von FTDI kann das eDIP an einen USB-Bus angeschlossen werden. Virtuelle-COM-Port Treiber gibt es für viele Betriebssyteme auf der FTDI Homepage*http://www.ftdichip.com/drivers/vcp.htm*.



Technische Änderung vorbehalten.
Wir übernehmen keine Haftung für
Druckfehler und Applikationsbeispiele.

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.

Pinout eDIPTFT43-A: SPI mode

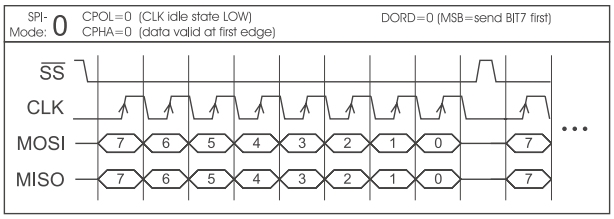
Pin	Symbol	In/Out	Function	Pin	Symbol	In/Out	Function
1	GND		Ground Potential for logic (0V)	21	GND		Ground (0V)
2	VDD		Power supply for logic (+5V)	22	VDD		Power supply (+5V)
3	NC		do not connect	23	AIN1	In	analogue input 0..5V
4	NC		do not connect	24	AIN2		DC impedance 1MOhm
5	RESET	In	L: Reset	25	OUT1 / MO8	Out	8 digital outputs maximum current: IOL = IOH = 10mA alternativ up to 8 matrix keyboard output lines (reduces the digital output lines, see chapter external keyboard)
6	SS	In	Slave Select	26	OUT2 / MO7		
7	MOSI	In	Serial In	27	OUT3 / MO6		
8	MISO	Out	Serial Out	28	OUT4 / MO5		
9	CLK	In	Shift Clock	29	OUT5 / MO4		
10	DORD	In	Data Order (0=MSB first; 1=LSB first)	30	OUT6 / MO3		
11	SPIMO	In	connect to GND for SPI interface	31	OUT7 / MO2		
12	NC		do not connect	32	OUT8 / MO1		
13	DPOM	In	L: disable PowerOnMacro do not connect for normal operation	33	IN1 / MI8	In	8 digital inputs open-drain with internal pullup 20..50k alternativ up to 8 matrix keyboard input lines (reduces the digital input lines, see chapter external keyboard)
14	CPOL	In	Clock Polarity (0=LO 1=HI when idle)	34	IN2 / MI7		
15	CPHA	In	Clock Phase sample 0=1st;1=2nd edge	35	IN3 / MI6		
16	BUZZ	Out	Buzzer output	36	IN4 / MI5		
17	DPROT	In	L: Disable Smallprotokoll do not connect for normal operation	37	IN5 / MI4		
18	DNC	Out	L: internal, do not connect	38	IN6 / MI3		
19	WP	In	L: Writeprotect for DataFlash	39	IN7 / MI2		
20	TEST SBUF	IN Out	open-drain with internal pullup 20..50k IN (Power-On) L: Testmode OUT L: data in sendbuffer	40	IN8 / MI1		

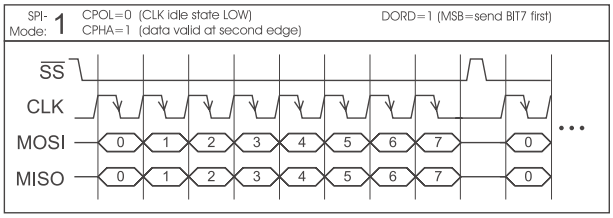
Hinweis:
Die Pins DORD, CPOL, CPHA, DPOM und TEST/SBUF haben einen internen Pull-UP, deshalb ist nur der LO-Pegel (0=GND) aktiv anzulegen. Für Hi-Pegel sind diese Pins offen zu lassen.
Am Pin 20 (SBUF) zeigt das Display mit einem low-Pegel, dass im internen Sendepuffer Daten zur Abholung bereit stehen. Diese Leitung kann z.B. mit einem Interrupteingang des Host Systems verbunden werden.

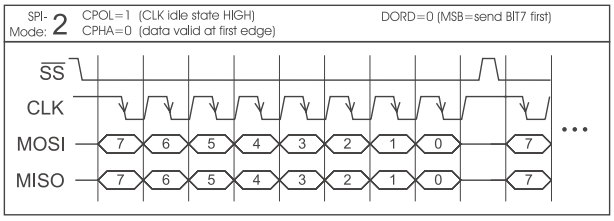
DATENÜBERTRAGUNG SPI

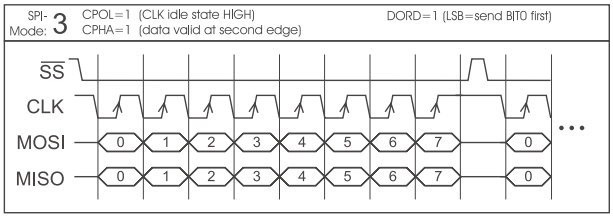
Eine Datenübertragung zum eDIP ist bis zu 200 kHz Nonstop möglich. Wenn jedoch zwischen den einzelnen Bytes während der Übertragung Pausen von jeweils min. 100 µs eingehalten werden, kann ein Byte mit bis zu 3 MHz übertragen werden.

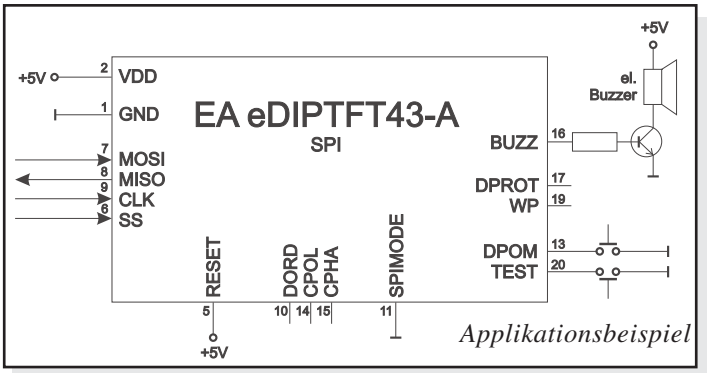
Um Daten vom eDIP zu Lesen (z.B. das ACK-Byte) muss ein Dummy-Byte (z.B. 0xFF) gesendet werden. Das eDIP benötigt eine bestimmte Zeit um die Daten bereit zu stellen; deshalb muss vor jedem zu lesenden Byte mindestens 6µs gewartet werden (keine Aktivität auf der CLK Leitung).











I²C-BUS INTERFACE

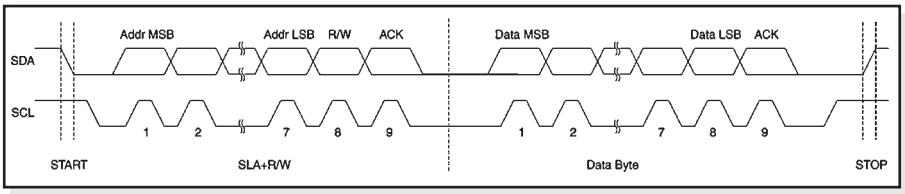
Eine Beschaltung des Displays wie unten abgebildet, ermöglicht den direkten Betrieb an einem I²C-Bus.
Am Display kann zwischen 8 unterschiedlichen Basisadressen und 8 Slave-Adressen ausgewählt werden.

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 eDIPTFT43-A: I2C mode

Pin	Symbol	In/Out	Function	Pin	Symbol	In/Out	Function
1	GND		Ground Potential for logic (0V)	21	GND		Ground (0V)
2	VDD		Power supply for logic (+5V)	22	VDD		Power supply (+5V)
3	NC		do not connect	23	AIN1	In	analogue input 0..5V
4	NC		do not connect	24	AIN2		DC impedance 1MOhm
5	RESET	In	L: Reset	25	OUT1 / MO8	Out	8 digital outputs maximum current: IOL = IOH = 10mA
6	BA0	In	Basic Address 0	26	OUT2 / MO7		
7	BA1	In	Basic Address 1	27	OUT3 / MO6		
8	SA0	In	Slave Address 0	28	OUT4 / MO5		alternativ up to 8 matrix keyboard output lines (reduces the digital output lines, see chapter external keyboard)
9	SA1	In	Slave Address 1	29	OUT5 / MO4		
10	SA2	In	Slave Address 2	30	OUT6 / MO3		
11	BA2	In	Basic Address 2	31	OUT7 / MO2		
12	I2CMO	In	connect to GND for I²C interface	32	OUT8 / MO1		
13	DPOM	In	L: disable PowerOnMacro do not connect for normal operation	33	IN1 / MI8	In	8 digital inputs open-drain with internal pullup 20..50k
14	SDA	Bidir.	Serial Data Line	34	IN2 / MI7		
15	SCL	In	Serial Clock Line	35	IN3 / MI6		
16	BUZZ	Out	Buzzer output	36	IN4 / MI5		
17	DPROT	In	L: Disable Smallprotokoll do not connect for normal operation	37	IN5 / MI4		alternativ up to 8 matrix keyboard input lines (reduces the digital input lines, see chapter external keyboard)
18	DNC	Out	L: internal, do not connect	38	IN6 / MI3		
19	WP	In	L: Writeprotect for DataFlash	39	IN7 / MI2		
20	TEST SBUF	IN Out	open-drain with internal pullup 20..50k IN (Power-On) L: Testmode OUT L: data in sendbuffer	40	IN8 / MI1		

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.



I²C - Address

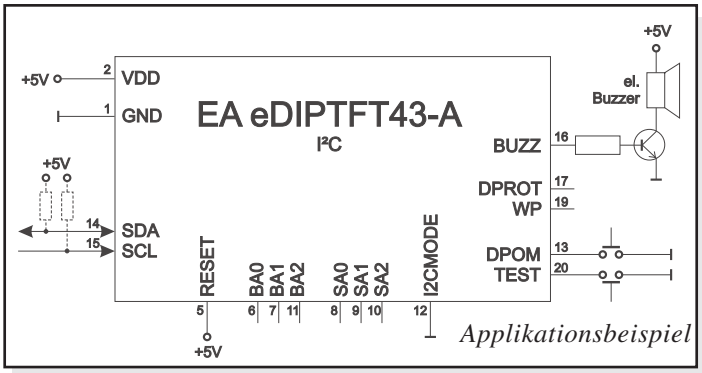
Pin 11,7,6 BA2	BA1	BA0	Base address	I²C address D7	D6	D5	D4	D3	D2	D1	D0
L	L	L	$10	0	0	0	1	SA2	SA1	SA0	RW
L	L	H	$20	0	0	1	0				
L	H	L	$30	0	0	1	1				
L	H	H	$40	0	1	0	0				
H	L	L	$70	0	1	1	1				
H	L	H	$90	1	0	0	1				
H	H	L	$B0	1	0	1	1				
H	H	H	$D0	1	1	0	1				

alle Pins offen: Schreiben $DE
Lesen $DF

DATENÜBERTRAGUNG I²C-BUS

So funktioniert prinzipiell die Übertragung:
- I²C-Start
- Master-Transmit: Display-I²C-Adr. (z.B. $DE), Smallprotokollpaket (Daten) senden
- I²C-Stop
- I²C-Start
- Master-Read: Display-I²C-Adr. (z.B. $DF), ACK-Byte und evtl. Smallprotokollpaket (Daten) lesen
- I²C-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).



Technische Änderung vorbehalten.
Wir übernehmen keine Haftung für
Druckfehler und Applikationsbeispiele.

ANALOGEINGÄNGE AIN1 UND AIN2 (PIN 23+24)

Zur Spannungsmessung stehen 2 Analogeingänge mit einer Eingangsempfindlichkeit von 0..VDD zur Verfügung. Jeder Eingang hat einen Bezug zu GND und einen Eingangswiderstand von ca. 1MΩ. Die Auflösung beträgt 10 Bit, was einem 3st. DVM entspricht. Die Grundgenauigkeit nach Abgleich liegt bei ca. 0,5%. Bitte beachten Sie, dass nur positive Spannungen angeschlossen werden dürfen!

Abgleich

Die Eingänge sind nicht abgeglichen. Eine Abgleichprozedur kann wie folgt aussehen:

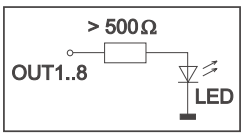
1.) Anlegen einer definierten Spannung im Bereich von 2V - VDD (Beispiel: 3,0V, AIN1)

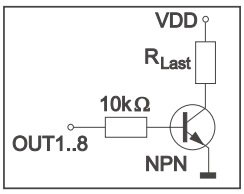
2.) Befehl zum Analogabgleich senden (siehe Seite 16). Im Beispiel: "ESC V@ 1 3000".

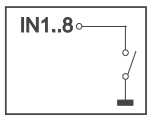
Die Messungen können gezielt angefordert oder auch direkt auf dem Display dargestellt werden (als Ziffernfolge, Bargraph oder in einem Zeigerinstrument in unterschiedlichsten Größen und Farben). Die direkte Darstellung der Messwerte erfolgt am einfachsten über ein Prozessmakro oder eines der Analogmakros (z.B. Ausführung bei jeder Änderung des Analogwertes an AIN1 bzw. AIN2, oder Ausführung bei Über- bzw. Unterschreiten eines Limits).

Für die direkte Darstellung am Display sind die Eingänge individuell skalierbar. Die Skalierung erfolgt über eine Definition an 2 Spannungswerten (Wert1=Anzeige1;Wert2=Anzeige2). Der Anzeigeumfang beträgt maximal 0 bis +/-9999,9. Lesen Sie dazu die Tabelle auf der Seite 16.

EIN- UND AUSGÄNGE







Das eDIP hat 8 digitale Ein- und 8 Ausgänge (CMOS Pegel, nicht potentialfrei).

8 Ausgänge (Pin 25-32)

Jeder Ausgang kann per Befehl "ESC Y W" individuell angesteuert werden. Pro Leitung kann ein Strom von max. 10mA geschaltet werden. Es ist somit möglich, mit einem Ausgang direkt eine LED (low current) zu schalten. Größere Ströme können mittels externen Transistors verstärkt werden.

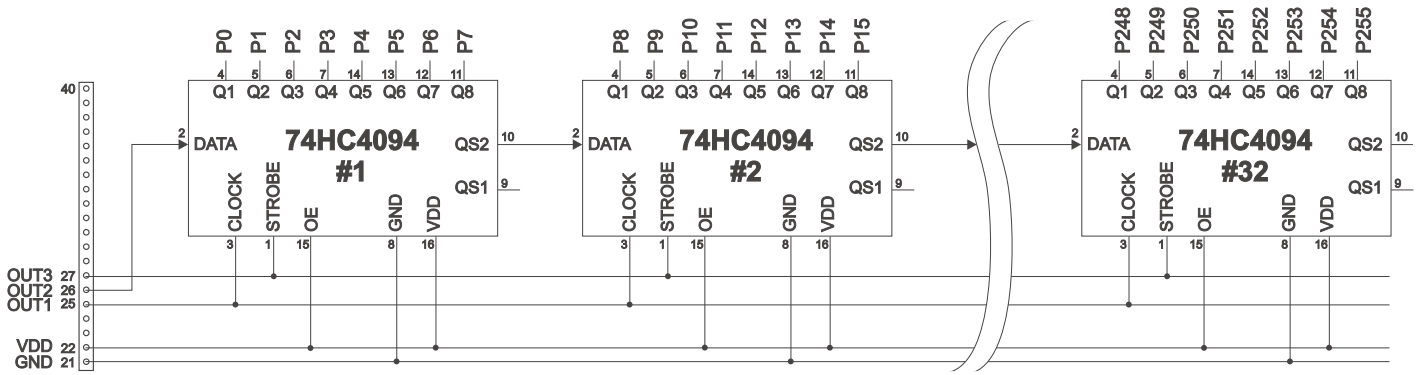
8 Eingänge (Pin 33-40)

Jeder Eingang hat einen ca. 20..50kΩ Pullup, somit ist es möglich Taster und Schalter direkt nach GND anzuschliessen. Die Eingänge können mit dem Befehl "ESC Y R" abgefragt und ausgewertet werden. Zusätzlich ist es möglich, bei Änderungen an den Eingängen ein Bit- / Portmakro automatisch aufzurufen (siehe Seite 28). Die automatische Portabfrage läßt sich mit dem Befehl "ESC Y A 1" aktivieren. Bei jeder Änderung des Eingangports werden zuerst die Bitmakros und dann das Portmakro ausgeführt. Ist kein Makro definiert so wird der neue Portzustand in den Sendepuffer gestellt (siehe auch Seite 19: Antworten/Rückmeldungen).

Anmerkung: Die Logik ist für langsame Vorgänge ausgelegt; d.h. mehr als 3 Änderungen pro Sekunde können nicht mehr sinnvoll ausgeführt werden.

NOCH MEHR AUSGÄNGE (PORTERWEITERUNG, AB V1.4)

Es können 1 bis 32 Bausteine vom Typ 74HC4094 an das eDIP (OUT1...OUT3) angeschlossen werden, damit sind 8 bis 256 weitere Ausgänge möglich. Mit dem Befehl "ESC Y E n1 n2 n3" (siehe Seite 17) können diese Ports komfortabel angesteuert werden.



EXTERNE MATRIX-TASTATUR

An den Ein- und Ausgängen kann eine Matrix-Tastatur (einzelne Tasten bis zur 8x8 Matrix) angeschlossen werden. Mit dem Befehl 'ESC Y M n1 n2 n3' werden die Anzahl der verwendeten Ein- und Ausgänge der Ports (n1,n2=1..8) definiert und die Tastenentprellung (n3=0..15 in 10ms Schritten) festgelegt. Bitte beachten Sie, dass bei Anschluß einer externen Tastatur die digitalen Eingänge um die Anzahl n1, und die Ausgänge um die Anzahl n2 reduziert werden.
Jede Taste wird i.d.R. zwischen einen Ausgang und einen Eingang geschaltet. Jeder Eingang ist mit einem ca. 20..50kΩ Pullup abgeschlossen. Um Doppeltastendrücke zu erkennen, müssen die Ausgänge voneinander entkoppelt werden. Dies geht am besten mit Schottky-Dioden (z.B. BAT 46).

Senden der Tastendrücke

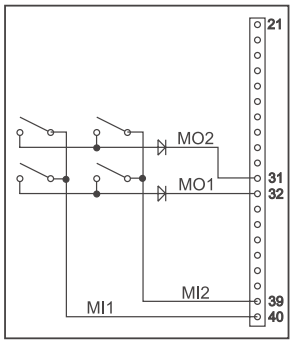
Bei jedem Druck einer Taste (Tastennummer 1..64) wird das dazugehörende Matrix-Makro ausgeführt, oder wenn kein Makro definiert ist, die Tastennummer mit Kennbuchstaben in den Sendepuffer gestellt. Das Loslassen der Taste wird nicht gesendet. Soll auch das Loslassen gesendet werden, so kann das über die Definition des Matrix Makros Nr.0 realisiert werden.
(siehe auch Seite 19: Antworten/Rückmeldungen)

Bestimmung der Tastennummer:

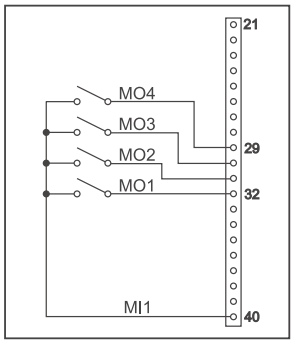
TastenNr = (AusgangNr - 1) * AnzahlEingänge + EingangNr (Ausgang = MOx, Eingang = MIx).

Beispiele:

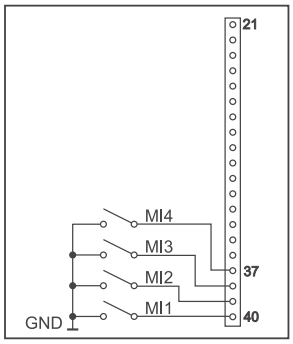
- Beispiel 1: Mit dem Befehl 'ESC Y M 2 2 ..' werden die 4 Tasten als 2x2 Matrix definiert. Die Tasten werden an 2 Eingänge (MI1, MI2) und 2 Ausgänge (MO1, MO2) angeschlossen. Die Ausgänge sind hier mit Dioden voneinander entkoppelt um Doppeltastendrücke erkennen zu können. Es stehen weiterhin 6 Eingänge und 6 Ausgänge zur Verfügung.
- Beispiel 2: Mit dem Befehl 'ESC Y M 1 4 ..' werden die 4 Tasten als 1x4 Matrix definiert. Die Tasten werden an 4 Ausgänge (MO1..MO4) angeschlossen und über den Eingang MI1 eingelesen. Es stehen weiterhin 7 Eingänge und 4 Ausgänge zur Verfügung.
- Beispiel 3: Wird nur ein Ausgang benutzt (4x1 Matrix), so können die Tasten auch gegen Masse geschalten werden und direkt an den Eingänge eingelesen werden (= 4x0 Matrix). Mit dem Befehl 'ESC Y M 4 0 ..' werden die 4 Tasten an den 4 Eingängen (MI1..MI4) definiert. Es stehen weiterhin 4 Eingänge und alle 8 Ausgänge zur Verfügung.
- Beispiel 3: Mit dem Befehl 'ESC Y M 4 4 ..' werden die 16 Tasten als 4x4 Matrix definiert. Die Tasten werden an 4 Eingänge (MI1..MI4) und 4 Ausgänge (MO1..MO4) angeschlossen. Die Ausgänge sind hier mit Dioden voneinander entkoppelt um Doppeltastendrücke erkennen zu können. Es stehen weiterhin 4 Eingänge und 4 Ausgänge zur Verfügung.



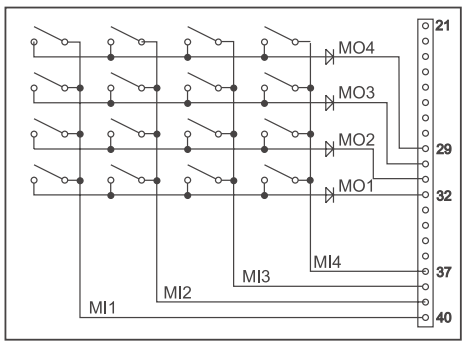
2x2 Matrix



1x4 Matrix



4x0 Matrix

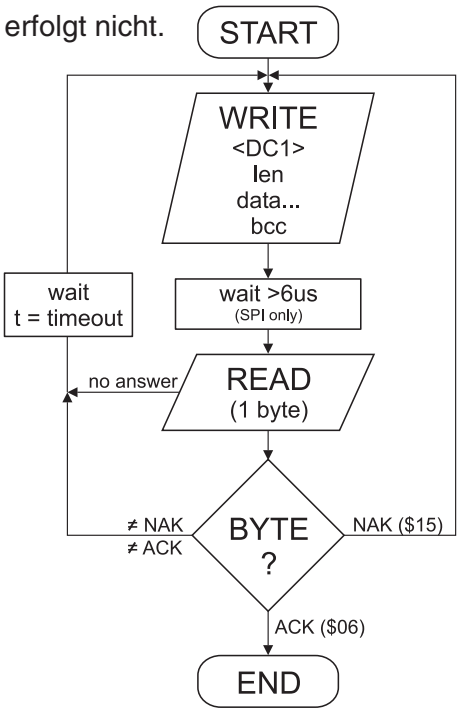


4x4 Matrix

Technische Änderung vorbehalten.
Wir übernehmen keine Haftung für
Druckfehler und Applikationsbeispiele.

DATENÜBERTRAGUNGSPROTOKOLL (SMALL PROTOKOLL)

Das Protokoll ist für alle 3 Schnittstellenarten RS-232, SPI und I²C identisch aufgebaut. Die Datenübertragung ist jeweils eingebettet in einen festen Rahmen mit Prüfsumme „bcc“. Das EA eDIPTFT43-A 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. 255 Byte betragen. Befehle die grösser als 255 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 PROTOLKOLL DEAKTIVIEREN

Das Protokoll ist für alle drei Schnittstellen RS-232, I²C 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 479,271

>	<DC1>	len	ESC	D	L	ESC	G	D	0	0	479	271	bcc			
	$11	$0E	$1B	$44	$4C	$1B	$47	$44	$00	$00	$00	$00	$DF	$01	$0FE	$01	$60
<	<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. Touchtastendrücke zu senden.

Pufferinformationen anfordern

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

<DC2> = 18(dez.) = $12 1 = 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 1 = 1(dez.) = $01 P = 80(dez.) = $50
<ACK> = 6(dez.) = $06
max. packet size = maximale Anzahl der Nutzdaten eines Protokollpaketes (eDIPTFT43-A = 255)
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 Protokoll-einstellungen abgefragt.

Letztes Datenpaket wiederholen

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

<DC2> = 18(dez.) = $12 1 = 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 komlette 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 läst sich das eDIP mit der Adresse adr Selektieren oder Deselektieren.

Technische Änderung vorbehalten.
Wir übernehmen keine Haftung für
Druckfehler und Applikationsbeispiele.

TERMINAL-BETRIEB

Das Display enthält eine integrierte Terminalfunktion. Nach dem Einschalten blinkt ein 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 Protokollrahmen oder ein abgeschaltetes Protokoll (siehe Seite 10+11).
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 '##'.
Die Grösse des benutzbaren Terminalfensters kann frei definiert werden.

+ 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)
$00 (dez: 0)	NUL	↑	↓	→	←	✓	♪	♥	←	↦	LF	↧	FF	CR	SO	SI
$10 (dez: 16)	0	1	2	3	4	5	6	7	8	9	⊙	ESC	↑	↓	→	←
$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	{	|	}	~	△
$80 (dez: 128)	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
$90 (dez: 144)	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	¢	£	¥	ß	ƒ
$A0 (dez: 160)	á	í	ó	ú	ñ	Ñ	ª	º	¿	⌐	¬	½	¼	¡	«	»
$B0 (dez: 176)	⠿	⠿	▒	╎	┤	╡	╢	╖	╕	╣	╗	╝	╜	╛	┐	
$C0 (dez: 192)	└	┴	┬	├	─	┼	╞	╟	╚	╔	╩	╦	╠	═	╬	╧
$D0 (dez: 208)	╨	╤	╥	╙	╘	╒	╓	╫	╪	┘	┌	█	▄	▌	▐	▀
$E0 (dez: 224)	α	β	Γ	π	Σ	σ	µ	τ	Φ	Θ	Ω	δ	∞	φ	ε	∩
$F0 (dez: 240)	≡	±	≥	≤	⌠	⌡	÷	≈	°	•	·	√	ⁿ	²	³	¯

Terminal-Font 2: 8x16

Achtung: Mit Grafikbefehlen kann der Inhalt des Terminalfensters überschrieben werden z.B. Löschen des Grafikbildschirms mit 'ESC DL'. 'ESC DL' löscht aber nicht das Terminalfenster.

BEFEHLE ÜBER DIE SERIELLE SCHNITTSTELLE SENDEN

Das eDIP 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.: **#GD0,0,479,271,**
- 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).

ALLE BEFEHLE AUF EINEN BLICK

Die eingebaute Intelligenz erlaubt den Aufbau eines Bildschirmes über unten stehende Befehle. Alle Befehle können sowohl über die serielle Schnittstelle (vgl. Seite 12) als auch in selbst-definierten Makros (vgl. Seiten 26/28) verwendet werden.

EA eDIPTFT43-A: Terminalbefehle											nach
Befehl	**Codes**									**Anmerkung**	**Reset**
Farbe einstellen	ESC	F	T	vf	hf					Farbe für den Terminal Betrieb einstellen. vf= Schriftfarbe; hf= Hintergrundfarbe	8,1
Fenster definieren	ESC	T	W	n1	s	z	b	h		Die Terminalausgabe erfolgt mit Font n1: 1=8x8; 2=8x16 innerhalb des Fensters ab Spalte s und Zeile z mit der Breite b und Höhe h (Angaben in Zeichen) Display Organisation 480x272: s=1..60; z=1..34/17; 272x480: s=1..34; z=1..60/30	8x16 1,1 60,17
Formfeed FF (dez:12) Carriage Return CR (13) Linefeed LF (dez:10)	^L ^M ^J									Bildschirm wird gelöscht (Hintergrundfarbe) und der Cursor nach Pos.1,1 gesetzt Cursor ganz nach links zum Zeilenanfang 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 restore			R							die letzte gesicherte Cursorposition wird wieder hergestellt	
Terminal AUS			A							Terminal ist ausgeschalten; Ausgaben werden verworfen	
Terminal EIN			E							Terminal ist eingeschalten; Ausgaben werden wieder angezeigt	Ein
Version anzeigen	ESC	T	V							Die Version wird im Terminal ausgegeben z.B. "EA eDIPTFT43-A V1.0 Rev.A"	
Projektname anzeigen			J							Der Makro-Projektname wird im Terminal ausgegeben z.B. "init / delivery state"	
Interface anzeigen			Q							Die eingestellte Schnittstelle wird im Terminal ausgegeben z.B. "RS232, 115200 baud, ADR: $07"	
Informationen anzeigen	ESC	T	I							Das Terminal wird initialisiert und gelöscht, die Software Version, Hardware Revision, der Makro-Projektname und die CRC-Checksummen werden im Terminal ausgegeben.	

EA eDIPTFT43-A: Grafikfunktionen											nach
Befehl	**Codes**									**Anmerkung**	**Reset**
Display-Befehle (Wirkung auf das gesamte Display)											
Displayfarben einstellen	ESC	F	D	vf	hf					Farbe für Display und Bereiche: vf=Vordergrundfarbe; hf=Hintergrundfarbe	8,1
Display Orientierung	ESC	D	O	n1						n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270° (0°+180°=480x272; 90°+270°=272x480)	0°
Display löschen			L							Displayinhalt löschen (mit Hintergrundfarbe füllen)	
Display füllen			S							Displayinhalt füllen (mit Vordergrundfarbe)	
Display mit Farbe füllen			F	n1						Displayinhalt mit Farbe n1=1..32 füllen	
Display invertieren			I							Displayinhalt invertieren (alle Pixel umkehren)	
Befehle zur Ausgabe von Zeichenketten											
Textfarben einstellen	ESC	F	Z	vf	hf					Farbe 1..32 (0=Transparent) für Zeichenketten einstellen: vf=Schrift; hf=Hintergrund	8,0
Font einstellen	ESC	Z	F	n1						Font mit der Nummer n1 einstellen	5
Font-Zoomfaktor			Z	n1	n2					n1 = X-Zoomfaktor (1x..8x); n2 = Y-Zoomfaktor (1x..8x)	1,1
Zeichenbreite / höhe			Y	n1	n2					n1=0..15: zusätzliche Breite Links/Rechts; n2=0..15: zusätzliche Höhe Oben/Unten	0, 0
Leerzeichenbreite			J	n1						Leerzeichenbreite: n1=0 aus Zeichensatz; n1=1 wie Ziffer; n1>=2 Breite in Pixel	0
Text-Winkel			W	n1						Text-Ausgabewinkel: n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°	0
Zeichenkette ausgeben L: Linksbündig C: Zentriert R: Rechtsbündig	ESC	Z	L C R	xx1	yy1	Text ...		NUL		Eine Zeichenkette (...) an xx1,yy1 ausgegeben; Zeichenkettenende: 'NUL' ($00), 'LF' ($0A) oder 'CR' ($0D); Mehrere Zeilen werden durch das Zeichen '|' ($7C) getrennt; Das Backslash-Zeichen '\' ($5C) hebt die Sonderfunkion der Zeichen '|\' auf;	
Zeichenkette innerhalb eines Bereiches ausgeben (ab V1.2)	ESC	Z	B	xx1	yy1	xx2	yy2	n1	Text ...	NUL	Eine Zeichenkette (...) innerhalb xx1,yy1 bis xx2,yy2 an der Position n1=1..9 ausgegeben; Der Bereich wird mit der Hintergrundfarbe gefüllt; n1=1: Oben Links; n1=2: Oben Zentriert; n1=2 Oben Rechts n1=4: Mitte Links; n1=5: Mitte Zentriert; n1=6 Mitte Rechts n1=7: Unten Links; n1=8: Unten Zentriert; n1=9 Unten Rechts	
Zeichenkette für Terminal	ESC	Z	T	Text ...						Befehl um eine Zeichenkette aus einem Makro an das Terminal auszugeben	
Geraden und Punkte zeichnen											
Geradenfarbe einstellen	ESC	F	G	vf	hf					Farbe vf=1..32 für Punkt/Geraden/Rechtecke einstellen; hf=Muster Hintergrundfarbe	8,1
Rechteck zeichnen	ESC	G	R	xx1	yy1	xx2	yy2			Vier Geraden als Rechteck von xx1,yy1 bis xx2,yy2 zeichnen	
Gerade zeichnen			D	xx1	yy1	xx2	yy2			Eine Gerade von xx1,yy1 bis xx2,yy2 zeichnen	
Gerade weiter zeichnen			W	xx1	yy1					Eine Gerade vom letzten Endpunkt bis xx1, yy1 zeichnen	
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
Punkt-/Geraden-Muster			M	n1						n1=1..255 Muster für Punkte/Geraden/Rechtecke einstellen; n1=0 kein Muster	0
Rechteckige Bereiche verändern / zeichnen											
Bereich löschen	ESC	R	L	xx1	yy1	xx2	yy2			Bereich von xx1,yy1 bis xx2,yy2 löschen (mit Displayhintergrundfarbe füllen)	
Bereich füllen			S	xx1	yy1	xx2	yy2			Bereich von xx1,yy1 bis xx2,yy2 mit Displayvordergrundfarbe füllen	
Bereich mit Farbe füllen			F	xx1	yy1	xx2	yy2	n1		Bereich von xx1,yy1 bis xx2,yy2 mit Farbe n1=1..32 füllen	
Bereich invertieren			I	xx1	yy1	xx2	yy2			Bereich von xx1,yy1 bis xx2,yy2 invertieren	
Bereich kopieren			C	xx1	yy1	xx2	yy2	xx3	yy3	Bereich von xx1,yy1 bis xx2,yy2 nach xx3,yy3 kopieren	
Musterfarben einstellen	ESC	F	M	vf	hf					Farbe 1..32 (0=Transp.) für monochrome Muster: vf=Vordergrund; hf=Hintergrund	8,1
Bereich mit Füllmuster		R	M	xx1	yy1	xx2	yy2	n1		Bereich von xx1,yy1 bis xx2,yy2 mit Muster n1 zeichnen	
Box zeichnen			O	xx1	yy1	xx2	yy2	n1		Rechteck von xx1,yy1 bis xx2,yy2 mit Muster n1 zeichnen	
Rahmenfarben einstellen	ESC	F	R	f1	f2	f3				Farben für Rahmen: f1=Rahmen aussen; f2=Rahmen innen; f3=Füllung	8,1,1
Rahmentyp einstellen	ESC	R	E	n1	n2					Rahmentyp n1=1..255; Rahmenwinkel: n2=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°	1, 0
Rahmen zeichnen			R	xx1	yy1	xx2	yy2			Rahmen von xx1,yy1 bis xx2,yy2 zeichnen	

Technische Änderung vorbehalten.
Wir übernehmen keine Haftung für
Druckfehler und Applikationsbeispiele.

EA eDIPTFT43-A: Befehle für Bitmaps / Animationen

Befehl	Codes									Anmerkung	nach Reset
Bitmap Bilder Befehle											
Monochrombild Farben	ESC	F	U	vf	hf					Bildfarbe für monchrome Bilder vf = Vordergrundfarbe; hf = Hintergrundfarbe	1,8
Bild-Zoomfaktor	ESC	U	Z	n1	n2					n1 = X-Zoomfaktor (1x..8x); n2 = Y-Zoomfaktor (1x..8x)	1,1
Bild-Winkel	ESC	U	W	n1						Ausgabewinkel des Bildes: n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°	0
Bild-Spiegeln	ESC	U	X	n1						n1=0: Normaldarstellung; N1=1: Das Bild wird horizontal gespiegelt	0
Farbbild Transparenz	ESC	U	T	n1						n1=0: keine Transparenz; Bild mit allen Farben Rechteckig darstellen n1=1: die Farbe der linken oberen Ecke wird als Transparentfarbe verwendet n1=2: falls vorhanden, definierte Transparentfarbe im Bild (.GIF.TGA.G16) verwenden n1=3: Transparentfarbe im Bild durch aktuelle Hintergrundfarbe ersetzen	2
internes Bild anzeigen	ESC	U	I	xx1	yy1	nr				Internes Bild mit der nr (0..255) aus dem Datenflash nach xx1,yy1 platzieren	
Bild über serielle Schnittst.	ESC	U	L	xx1	yy1	G16 daten ...				Ein Bild an xx1,yy1 platzieren; Daten des Bildes siehe Bildaufbau G16-Format	
Hardcopy senden	ESC	U	H	xx1	yy1	xx2	yy2			Nach diesem Befehl wird der Bildausschitt im G16-Format gesendet (landet im Sendepuffer)	
Animierte Bitmap Bilder											
Monochrombild Farben	ESC	F	W	vf	hf					Bildfarbe für monchrome Animationen vf = Vordergrundfarbe; hf = Hintergrundfarbe;	1,8
Animation-Zoomfaktor	ESC	W	Z	n1	n2					n1 = X-Zoomfaktor (1x..8x); n2 = Y-Zoomfaktor (1x..8x)	1,1
Animation-Winkel	ESC	W	W	n1						Ausgabewinkel der Animationsbilder: n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°	0
Animation-Spiegeln	ESC	W	X	n1						n1=0: Normaldarstellung; n1=1: Das Animationsbild wird horizontal gespiegelt	0
Transparenz für Farbanimationen	ESC	W	T	n1						n1=0: keine Transparenz; Animation mit allen Farben Rechteckig darstellen; n1=1: die Farbe der linken oberen Ecke wird als Transparentfarbe verwendet n1=2: falls vorhanden, die Transparentfarbe in der Animation (.GIF .G16) verwenden n1=3: Transparentfarbe in der Animation durch aktuelle Hintergrundfarbe ersetzen	2
Einzelbild laden	ESC	W	I	xx1	yy1	n1	n2			vom Animationsbild n1=0..255 das Unterbild n2 nach xx1,yy1 laden	
Animation definieren	ESC	W	D	nr	xx1	yy1	n2	typ	zeit	Ein Animationsprozess mit der Nummer nr=1..4 wird an der Position xx1,yy1 (=linke obere Ecke) mit dem Animationsbild n2=0..255 definiert. typ: 1=einmal; 2=zyklisch; 3=pingpong; 4=einmal rückwärts; 5=zyklisch rückwärts; typ: 6=pingpong rückwärts; 7=manuell (Befehle ESC W N|P|F|M verwenden) zeit: 0=Stop; 1..254=fixes Zeitraster in 1/10s; 255=Zeiten aus Animationsbild	
Animatiostyp ändern	ESC	W	Y	nr	typ					Dem Animationsprozess nr=1..4 einen neuen Typ typ=1..7 zuweisen	
Animationszeit ändern	ESC	W	C	nr	time					Dem Animationsprozess nr=1..4 eine neue Zeit time=0..255 in 1/10s zuweisen	
nächstes Animationsbild	ESC	W	N	nr						Das nächste Unterbild von dem Animationsprozess nr=1..4 anzeigen	
vorheriges Animationsbild	ESC	W	P	nr						Das vorherige Unterbild von dem Animationsprozess nr=1..4 anzeigen	
Animationsbild anzeigen	ESC	W	F	nr	n2					Das Unterbild n2 von dem Animationsprozess nr=1..4 anzeigen	
animiere bis Bildnr	ESC	W	M	nr	n2					Animiere vom aktuellen Unterbild bis zu Unterbild n2 von der Animation nr=1..4	
Animation löschen	ESC	W	L	nr						Animation nr=1..4 wird gestoppt und das Bild wird mit Displayhintergrund gelöscht	

EA eDIPTFT43-A: Befehle für Bargraphs

Befehl	Codes								Anmerkung				nach Reset
Bargraph Befehle													
Bargraph Farben	ESC	F	B	vf	hf	rf			Farben für Bargraph: vf = Vordergrund; hf = Hintergrund; rf = Rahmenfarbe				8,1,8
Bargraph Muster	ESC	B	M	n1					Muster für Bargraph n1=1..255; n1=0 kein Muster (gültig für typ=0..3)				0
Bargraph Rahmen			E	n1					Rahmen für Bargraph n1=1..255; einstellen (gültig für typ=4..7)				1
Bargraph Strichbreite			B	n1					Strichbreite für Bargraph n1=1..255; n1=0 automatisch (gültig für typ=2,3,6,7)				0
Bargraph definieren	ESC	B	R L O U	n1	xx1	yy1	xx2	yy2	aw	ew	typ	Bar nach L(inks),R(echts),O(ben),U(nten) als Nr. n1=1..20 definieren xx1,yy1,xx2,yy2 umschließendes Rechteck. aw, ew (0..254) sind die Werte für 0% und 100%. typ: 0=Balkenmuster; 1=Balkenmuster im Rechteck; typ: 2=Strichmuster; 3=Strichmuster im Rechteck; typ: 4=Balkenrahmen; 5=Balkenrahmen im Rechteck; typ: 6=Strichrahmen; 7=Strichrahmen im Rechteck;	kein Bar defi-niert
Bargraph aktualisieren	ESC	B	A	n1	wert				Bar mit der Nummer n1 auf den neuen Benutzer-´wert´ setzen und zeichnen.				
Bargraph neu zeichnen			N	n1					Den Bargraph mit der Nummer n1 komplett neu zeichnen				
Bargraphwert senden			S	n1					Den aktuellen Wert des Bargraph Nr. n1 senden (landet im Sendepuffer)				
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				
Benutzerwerte - Formatierte Zlffernausgabe													
Benutzerwert Farbe	ESC	F	X	vf	hf				Farbe für Bargraph Benutzerwert einstellen. vf=Schriftfarbe; hf=Hintergrundfarbe				8,1
Benutzerwert Font	ESC	B	F	n1					Font für Bargraph Benutzerwert mit der Nummer n1 einstellen				5
Benutzerwert Zoom			Z	n1	n2				Zoomfaktor für Bargraph Benutzerwert. n1=X-Zoom 1x..8x; n2=Y-Zoom 1x..8x				1,1
Benutzerwert Breite/Höhe			Y	n1	n2				n1=0..15: zusätzliche Zeichenbreite Links/Rechts; n2=0..15: zusätzliche Zeichenhöhe Oben/Unten; für Bargraph Benutzerwert;				0, 0
Benutzerwert Winkel			W	n1					Bargraph Benutzerwert Schriftwinkel: n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°				0°
Benutzerwert / Skalierung für Bar definieren	ESC	B	X	n1	xx1	yy1	Format String	NUL	Benutzerwert für Bargraph nr=1..20 definieren. Ausgabe rechtsbündig an xx1,yy1; Jeweils 2 Barwerten (bw1,bw2 =0..254) wird je ein Benutzerwert max. Anzeigenumfang 4 1/2 Stellen 19999 + Dezimalpunkt('.' oder ',') + evtl. Vorzeichen '-' zugeordnet. Format String: "bw1=Benutzerwert1;bw2=Benutzerwert2". 'NUL' ($00)=Stringende Beispiel: Anzeige soll bei 0 "-123.4" und bei 100 "567.8" sein Format String: "0=-123.4;100=567.8"				

EA eDIPTFT43-A: Befehle für Instrumente (ab V1.4)

Befehl	Codes								Anmerkung				nach Reset
Instrumente Befehle													
Instrument definieren	ESC	I	P	n1	xx1	yy1	n2	n3	aw	ew		Instrument Nr. n1=1..4 an xx1,yy1 definieren (=linke obere Ecke); Instrumentenbild n2=0..255 verwenden. Ausgabewinkel n3=0: 0°; n3=1: 90°; n3=2: 180°; n3=3: 270°; aw, ew (0..254) sind die Werte für 0% und 100%.	kein Instru-ment defi-niert
Instrument aktualisieren	ESC	I	A	n1	wert				Instrument mit der Nummer n1 auf den neuen Benutzer-´wert´ setzen und zeichnen.				
Instrument neu zeichnen			N	n1					Den Instrument mit der Nummer n1 komplett neu zeichnen				
Instrument senden			S	n1					Den aktuellen Wert des Instrument Nr. n1 senden (landet im Sendepuffer)				
Instrument löschen			D	n1	n2				Die Definition des Instrument mit der Nummer n1 wird ungültig. War das Instrument als Eingabe mit Touch definiert, so wird auch dieses Touchfeld gelöscht. n2=0: Instrument weiterhin sichtbar; n2=1: Instrument wird gelöscht				
Benutzerwerte - Formatierte Ziffernausgabe													
Benutzerwert Farbe	ESC	F	I	vf	hf				Farbe für Instrument Benutzerwert einstellen. vf=Schriftfarbe; hf=Hintergrundfarbe				8,1
Benutzerwert Font	ESC	I	F	n1					Font für Instrument Benutzerwert mit der Nummer n1 einstellen				5
Benutzerwert Zoom			Z	n1	n2				Zoomfaktor für Instrument Benutzerwert. n1=X-Zoom 1x..8x; n2=Y-Zoom 1x..8x				1,1
Benutzerwert Breite/Höhe			Y	n1	n2				n1=0..15: zusätzliche Zeichenbreite Links/Rechts; n2=0..15: zusätzliche Zeichenhöhe Oben/Unten; für Instrument Benutzerwert;				0, 0
Benutzerwert Winkel			W	n1					Instrument Benutzerwert Schriftwinkel: n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°				0°
Benutzerwert / Skalierung für Instrument definieren	ESC	I	X	n1	xx1	yy1	Format String	NUL	Benutzerwert für Instrument nr=1..4 definieren. Ausgabe rechtsbündig an xx1,yy1; Jeweils 2 Instrumentwerten (iw1,iw2 =0..254) wird je ein Benutzerwert max. Anzeigenumfang 4 1/2 Stellen 19999 + Dezimalpunkt('.' oder ',') + evtl. Vorzeichen '-' zugeordnet. Format String: "iw1=Benutzerwert1;iw2=Benutzerwert2". 'NUL' ($00)=Stringende Beispiel: Anzeige soll bei 0 "-123.4" und bei 100 "567.8" sein Format String: "0=-123.4;100=567.8"				

Technische Änderung vorbehalten.
Wir übernehmen keine Haftung für
Druckfehler und Applikationsbeispiele.

EA eDIPTFT43-A: Befehle für Makros										nach Reset
Befehl	**Codes**								**Anmerkung**	
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)	
Port Makro ausführen			P	n1					Das Port-Makro mit der Nummer n1 (0..255) aufrufen (max. 7 Ebenen)	
Bit Makro ausführen			B	n1					Das Bit-Makro mit der Nummer n1 (0..255) aufrufen (max. 7 Ebenen)	
Matrix Makro ausführen			X	n1					Das Matrix-Makro mit der Nummer n1 (0..255) aufrufen (max. 7 Ebenen)	
Prozess Makro ausführen			C	n1					Das Prozess-Makro mit der Nummer n1 (0..255) aufrufen (max. 7 Ebenen)	
Analog Makro ausführen			V	n1					Das Analog-Makro mit der Nummer n1 (0..255) aufrufen (max. 7 Ebenen)	
Makros sperren	ESC	M	L	typ	n1	n2			Die Makros vom typ = 'N','T','P','B','X','C' oder 'V' (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','P','B','X','C' oder 'V' (typ = 'A' alle Makrotypen) werden von der Nummer n1 bis n2 freigegeben, d.h. bei Aufruf wieder ausgeführt.	
Makro-/Bildpage auswählen	ESC	M	K	n1					Auswahl einer Page für Makros und Bilder n1=0..15. Ist ein Makro/Bild in der akt. Page 1..15 definiert, dann wird dieses Makro/Bild von Page 0 genommen. z.B. zum Umschalten von Sprachen oder für horizontalen / vertikalen Einbau.	nicht
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	
automatische (Normal-) Makros										
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 einmalig			E	n1	n2	n3			Makros n1..n2 automatisch eimal 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.	
Makro Prozesse										
Makroprozess definieren	ESC	M	D	nr	typ	n3	n4	zs	Ein Makroprozess läuft im Hintergrund und im Gegensatz zum autom. Normal-Makro nicht durch Eingaben oder serielle Daten unterbrochen. Bis zu 4 Makroprozesse können mit der Nummer nr (1..4) definiert werden (1=höchste Priorität). Die (Prozess-) Makros n3 bis n4 werden nacheinander alle zs/10s ausgeführt. typ: 1=einmalig; 2=zyklisch; 3=pingpong n3..n4..n3	
Makroprozess Zeitintervall			Z	nr	zs				Dem Makroprozess mit der Nummer nr (1..4) 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 und Animationen werden mit n1=0 gestoppt und n1=1 gestartet, um z.B. Einstellungen und Ausgaben über die Schnittstelle ungestört auszuführen.	1

EA eDIPTFT43-A: Befehle für die Analogeingänge										nach Reset
Befehl	**Codes**								**Anmerkung**	
Befehle für Analogeingänge										
Analogabgleich	ESC	V	@	nr	xx1				Der Abgleich für die Analogkanäle erfolgt folgendermassen: 1.) Definierte Spannung (2V..VDD) an AIN1 (Kanal1) oder AIN2 (Kanal2) anlegen. 2.) Befehl unter Angabe des Kanals nr=1..2 und xx1=Spannugswert (16-Bit) in [mV] ausführen; z.B. 3.0V an AIN1; Befehl: '#V@1,3000;'	nicht kali-briert
Analog-Abfrage Ein/Aus	ESC	V	A	n1					Der automatische Scan der Analogkanäle wird n1=0: deaktiviert; n1=1: aktiviert	0
Analogwert senden			D	nr					Es wird der Wert in [mV] vom Analogkanal nr=1..2 gesendet (landet im Sendepuffer)	
Bereiche / Grenzen für Analog-Makros			K	nr	n1	n2	n3		Zwei Grenzen für Analogkanal nr=1..2 einstellen. Mit Hilfe dieser Grenzen können bei Über- und/oder Unterschreiten diverse Analog-Makros automatisch gestarted werden. n1=untere Grenze in [mV/20]; n2=obere Grenze in [mV/20]; n3=Hyterese in [mV].	0
Analog-Makros umdefinieren	ESC	V	M	n1	n2				Der Analogmakrofunktion n1=0..19 die Analogmakronummer n2=0..255 zuweisen	
Bargraph für Analogeingang	ESC	V	B	nr	n2				Dem Analogkanal nr=1..2 wird der Bargraph mit der Nummer n2=1..20 zugewiesen Bei der Bargraphdefinition sind die Anfangs- und Endwerte in [mV/20] anzugeben	
Instrument für Analogeingang (ab V1.4)			+	nr	n2				Dem Analogkanal nr=1..2 wird das Instrument mit der Nummer n2=1..4 zugewiesen Bei der Instrumentdefinition sind die Anfangs- und Endwerte in [mV/20] anzugeben	
Bar/Instrument aktualisieren			R	nr					Alle definierten Bargraphen und Instrumente für Analogkanal nr=1..2 aktualisieren	
Benutzerwerte - Formatierte Ziffernausgabe										
Benutzerwert Farbe	ESC	F	V	nr	vf	hf			Farbe für Analogkanal nr=1..2 einstellen. vf=Schriftfarbe; hf=Hintergrundfarbe	8,1
Benutzerwert Font		V	F	nr	n1				Font für Analogkanal nr=1..2 mit der Nummer n1 einstellen	5
Benutzerwert Zoom			Z	nr	n1	n2			Zoomfaktor für Analogkanal nr=1..2 einstellen. n1=X-Zoom 1x..8x; n2=Y-Zoom 1x..8x	1,1
Benutzerwert Breite/Höhe			Y	nr	n1	n2			n1=0..15: zusätzliche Zeichenbreite Links/Rechts; n2=0..15: zusätzliche Zeichenhöhe Oben/Unten; für Kanal nr=1..2;	0, 0
Benutzerwert Winkel			W	nr	n1				Analogkanal nr=1..2 Schriftwinkel: n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°;	0
Benutzerwerte / Skalierung einstellen	ESC	V	E	nr	Format String...			NUL	Benutzerwerte für Analogkanal nr=1..2 einstellen. Jeweils 2 Analogwerten (0..5000mV) wird ein Benutzerwert max. Anzeigenumfang 4 1/2 Stellen 19999 + Dezimalpunkt('.' oder ',') + evtl. Vorzeichen '-' zugeordnet. Format String: "mV1=Benutzerwert1;mV2=Benutzerwert2"; 'NUL' ($00)=Stringende Beispiel: Anzeige soll bei 2000 mV "-123.45" und bei 1000mV "0.00" sein Format String: "2000=-123.45;1000=0"	0 =0.00 5000 =5.00
Benutzerwert senden	ESC	V	S	nr					aktuellen Benutzerwert für Analogkanal nr=1..2 senden (landet im Sendepuffer)	
Benutzerwert an Terminal			T	nr					aktuellen Benutzerwert für Analogkanal nr=1..2 zum Terminal ausgeben	
Benutzerwert anzeigen			G	nr	xx1	yy1			aktuellen Benutzerwert für Analogkanal nr=1..2 rechtsbündig an xx1,yy1 ausgeben	

EA eDIPTFT43-A: Allgemeine Befehle										nach Reset
Befehl	**Codes**								**Anmerkung**	
Hintergrundbeleuchtung										
Beleuchtung Helligkeit	ESC	Y	H	n1					Helligkeit der LED-Beleuchtung auf n1=0..100% einstellen	100
Helligkeit erhöhen			N						Helligkeit der LED-Beleuchtung um einen Schritt erhöhen	
Helligkeit verringern			P						Helligkeit der LED-Beleuchtung um einen Schritt verringern	
Änderungszeit einstellen			Z	n1					n1=0..31: Zeit zum Ändern der LED-Helligkeit von 0..100% in 1/10s	5
Beleuchtung Ein/Aus			L	n1					Beleuchtung n1=0: AUS; n1=1: EIN; n1=2..255: für n1/10s lang einschalten	1
Helligkeit über Bargraph			B	n1					Die Helligkeit der Beleuchtung wird mit Bargraph n1=1..20 gekoppelt. d.h wird der Bar per Befehl oder Touch eingestellt, ändert sich die Helligkeit entsprechend.	
Helligkeit über Instrument			+	n1					Die Helligkeit der Beleuchtung wird mit Instrument n1=1..4 gekoppelt. d.h wird das Instrument per Befehl oder Touch eingestellt, ändert sich die Helligkeit entsprechend. (ab V. 1.4)	
Parameter speichern			@						Die aktuelle LED-Helligkeit und Änderungszeit als Startwert im EEPROM speichern	
Ein- Ausgangs Port										
Ausgabe-Port schreiben	ESC	Y	W	n1	n2				n1=0: Alle 8 Ausgabe-Ports entsprechend n2 (=8-Bit Binärwert) einstellen n1=1..8: Ausgabe-Port n1 rücksetzen (n2=0); setzen (n2=1); invertieren (n2=2)	Ports 1-8=0
Eingabe-Port lesen			R	n1					n1=0: Alle 8 Eingabe-Ports als 8-Bit Binärwert einlesen (landet im Sendepuffer) n1=1..8: Eingabe-Port n1 einlesen (1=H-Pegel=5V, 0=L-Pegel=0V)	
Port Scan Ein/Aus			A	n1					Der automatiche Scan des Eingabe-Port wird n1=0: deaktiviert; n1=1: aktiviert	1
Eingabe-Port invers			I	n1					Der Eingabe-Port wird n1=0: normal; n1=1: invertiert ausgewertet	0
Matrix-Tastatur			M	n1	n2	n3			Festlegung einer externen Matrix-Tastatur an den Ein- und Ausgängen n1=Anzahl Eingänge (1..8); n2=Anzahl Ausgänge (0..8); n3= Entprellung (0..7)	0
Bit-Makros für Eingänge umdefinieren	ESC	Y	D	n1	n2	n3			Eingang n1=1..8 wird bei fallender Flanke n2=0 das Bitmakro n3=0..255 zugewiesen Eingang n1=1..8 wird bei steigender Flanke n2=1 das Bitmakro n3=0..255 zugewiesen	
Matrix-Makros für Tasten umdefinieren			X	n1	n2				Der Tastennummer n1=1..64 das Matirxmakro n2=0..255 zugewiesen Beim Loslassen der Taste n1=0 wird das Matirxmakro n2=0..255 aufgerufen	
zusätzliche Ausgänge ab V1.4 (nur mit Porterweiterung)	ESC	Y	E	n1	n2	n3			zusätzliche Ausgänge des 74HC4094 (siehe Porterweiterung S.8) von Port n1=0..255 bis Port n2=0..255 einstellen; n3=0: rücksetzen; n3=1: setzen; n3=2: invertieren;	
Sonstige-Befehle										
Farbe neu definieren	ESC	F	P	n1	R5	G6	B5		Der Farbe n1=1..32 der neue RGB-Werte zuweisen (R5:Bit7..3; G6:Bit7..2; B5:Bit7..3)	
Warten (Pause)	ESC	X	n1						n1/10s 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					Summerausgang (PIN16) wird n1=0:AUS;n1=1:EIN;n1=2..255;für n1/10s eingeschaltet	AUS
Bytes senden	ESC	S	B	anz	daten ...				Es werden anz (=1..255) Bytes zum Sendepuffer gesendet; im Quelltext der Makroprogrammierung darf die Anzahl anz nicht angegeben werden, diese wird vom eDIPTFT-Compiler automatisch eingetragen.	
Version senden			V						Version wird als String gesendet z.B."EA eDIPTFT32-A V1.0 Rev.A TP+" (Sendepuffer)	
Projektname senden			J						Es wird der Makro-Projektname als String gesendet z.B. "init / delivery" (Sendepuffer)	
Interne Infos senden			I						Es werden interne Informationen vom eDIP gesendet (landen im Sendepuffer)	

Technische Änderung vorbehalten.
Wir übernehmen keine Haftung für
Druckfehler und Applikationsbeispiele.

TOUCH PANEL

Die Version EA eDIPTFT43-ATP wird mit einem analogen, resitiven Touchpanel geliefert. Bis zu 60 Touchbereiche (Tasten, Schalter, Bargrapheingaben...), können gleichzeitig und pixelgenau definiert werden. Das eDIP unterstützt die Darstellung mit komfortablen Befehlen. 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.

EA eDIPTFT43-A: Befehle für das Touch-Panel												nach Reset
Befehl	**Codes**										**Anmerkung**	
Voreinstellungen												
Touch-Rahmen Farbe	ESC	F	E	n1	n2	n3	s1	s2	s3		Farben (0..32) für den Rahmen von Tasten/Schaltern einstellen. n=Normal; s=Selektiert; 1=Rahmen aussen; 2=Rahmen Innen; 3=Füllung	8,1,2 8,1,7
Touch-Rahmen Form		A	E	n1	n2						n1=Rahmen Nr. für ESC AT AK; n2=Rahmenwinkel 0=0°; 1=90°; 2=180°; 3=270°	1,0
Touch-Button Farbe	ESC	F	C	nv	nh	sv	sh				Farben (0..32) für monochrome Touchbuttons einstellen. n=Normal; s=Selektiert; v=Vordergrundfarbe; h=Hintergrundfarbe	8,1 8,1
Touch-Button		A	C	n1	n2	n3	n4				n1=Button Nr. für ESC AU AJ; n2=Buttonwinkel; n3=X-Zoom 1..8; n4=Y-Zoom 1..8	1,0,1,1
Radiogroup für Schalter	ESC	A	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
Voreinstellungen Beschriftungs-Font												
Beschriftungs Farbe	ESC	F	A	nf	sf						Farbe für die Touchtastenbeschriftung einstellen. nf=normale sf=selektierte Schriftfarbe	8,1
Beschriftungs Font	ESC	A	F	nr							Font mit der Nummer nr für Touchtastenbeschriftung einstellen	5
Beschriftungs-Zoomfaktor			Z	n1	n2						n1 = X-Zoomfaktor (1x..8x); n2 = Y-Zoomfaktor (1x..8x)	1,1
Zeichenbreite / höhe			Y	n1	n2						n1=0..15: zusätzliche Breite Links/Rechts; n2=0..15: zusätzliche Höhe Oben/Unten	0,0
Beschriftungs-Winkel			W	n1							Text-Ausgabewinkel: n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°	0
Offset für selektierten Text			O	n1	n2						n1=X-Offset; n2=Y-Offset; n1,n2=0..7 +8 für negative Richtung	0, 0
Touchbereiche definieren												
Touch-Taste definieren (Taste ist gedrückt solange der Touch berührt wird)	ESC	A	T	xx1	yy1	xx2	yy2	down Code	up Code	Text ...	NUL	'T': aktueller Rahmen wird von xx1,yy1 bis xx2,yy2 als Taste definiert 'K': aktueller Rahmen wird von xx1,yy1 bis xx2,yy2 als Schalter definiert 'U': aktueller Button wird an xx1,yy2 geladen und als Taste definiert 'J': aktueller Button wird an xx1,yy2 geladen und als Schalter definiert 'down Code': (1-255) Rückgabe / Touchmakro beim Drücken. 'up Code': (1-255) Rückgabe / Touchmakro beim Loslassen. (down- / up-Code = 0 Drücken / Loslassen wird nicht gemeldet). 'Text': Es folgt optional die Zeichenkette für die Beschriftung. Mehrzeilige Texte werden mit dem Zeichen '|' ($7C, dez:124) getrennt; Optional kann nach dem Zeichen '~' ($7E, dez:126) ein Text für die selektierte Darstellung angeben werden. z.B. "LED|EIN~LED|AUS" Ist das erste Zeichen ein 'C', 'L', oder 'R' wird damit die Ausrichtung des Textes eingestellt (C=zentriert=default; L=linksbündig; R=rechtsbündig). 'NUL':($00) = Zeichenketten/Touchtastenende	
			U	xx1	yy1	down Code	up Code	Text ...	NUL			
Touch-Schalter definieren (Zustand der Schalter toggelt nach jeder Berührung)	ESC	A	K	xx1	yy1	xx2	yy2	down Code	up Code	Text ...	NUL	
			J	xx1	yy1	down Code	up Code	Text ...	NUL			
Zeichenbereich definieren	ESC	A	D	xx1	yy1	xx2	yy2	n1	vf		Ein Zeichenbereich wird definiert. Innerhalb der Eck-Koodinaten xx1,yy1 und xx2,yy2 kann dann mit der Strichstärke n1 und Farbe vf 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.	
Instrument per Touch einst. (ab V1.4)	ESC	A	+	nr							Das Instrument mit der Nr. n1 wird zur Eingabe per Touchpanel definiert.	
Globale Einstellungen												
Touchabfrage Ein/Aus	ESC	A	A	n1							Touchrabfrage wird n1=0:deaktiviert; n1=1:aktiviert;	1
Touch-Tasten Reaktion	ESC	A	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
Barwert / Instrumentwert automatisch senden	ESC	A	Q	n1							das automatischen Senden eines neuen Bargraph-/Instrumentenwertes 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
sonstige Funktionen												
Touch-Taste Invertieren	ESC	A	N	Code							Die Touch-Taste mit dem zugeordnetem Return-Code wird manuell Invertiert	
Touch-Schalter einstellen			P	Code	n1						Zustand des Schalters wird per Befehl geändert n1=0=Aus; n1=1=Ein.	
Touch-Schalter abfragen			X	Code							Zustand des Schalters (Aus=0; Ein=1) wird in den Sendepuffer gestellt.	
Radiogroup abfragen			G	nr							der downcode des aktiven Schalters der Radiogroup nr wird in den Sendepuffer gestellt	
Touch-Bereich Löschen	ESC	A	L	Code	n1						Der Touchbereich mit dem Return-Code (Code=0: alle) wird aus der Abfrage 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	

TOUCHPANELABGLEICH

Der Touch ist bei Auslieferung abgeglichen und einsatzbereit. Durch Alterung und Abnutzung kann es nötig sein, dass das Touchpanel mit folgender Prozedur neu abgeglichen werden muss:

1a. Den Befehl 'ESC A@' senden oder

1b. Beim Einschalten Touch berühren und gedrückt halten.
Nach Erscheinen der Meldung *"touch adjustment ?"* den Touch wieder loslassen.
Innerhalb 1 Sekunde den Touch nochmals für mindestens 1 Sekunde berühren.

2. Den Anweisungen zum Abgleich folgen (2 Punkte *Linksoben* und *Rechtsunten* betätigen).

ANTWORTEN / RÜCKMELDUNGEN

Alle Antworten des eDIPs werden zuerst einmal in einen Sendepuffer gestellt. Über das Small-Protokoll werden diese dann vom Host angefordert (siehe Seite 10). Dies kann per „Polling“ geschehen, oder altenativ dazu zeigt der Pin 20 „SBUF“ mit einem LO-Pegel an, dass Daten zur Abholung bereit stehen.

Antworten des EA eDIPTFT-43A								
Kennung		**anz**	**daten**					**Anmerkung**
Selbstständige Antworten (landen im Sendepuffer)								
ESC	**A**	1	code					Antwort vom Analogen Touchpanel wenn eine Taste/Schalter gedrückt wurde. code = down oder up Code der Taste/S Es wird nur gesendet wenn kein Touch-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	**P**	1	wert					Nach Änderung des Eingangs-Port wird der neue 8-Bit Wert gesendet. Der Port-Scan muß aktiviert sein siehe Befehl A n1'. Es wird nur gesendet wenn kein Port-Makro mit der Nr. wert definiert ist !
ESC	**M**	1	nr					Nach Erkennen eines Tastendruckes der externen Matrix-Tastatur wird die neu gedrückte Tastennummer nr gesendet wird nur gesendet wenn kein Matrix-Makro mit der Nr. nr definiert ist !
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 D innerhalb des freien Touchbereiches an den Koordinaten xx1,yy1
Antworten nur nach Anforderung per Befehl (landen im Sendepuffer)								
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 gesende = 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	**Y**	2	nr	wert				Nach dem Befehl 'ESC Y R' wird der angeforderte Eingangs-Port gesendet. nr=0: wert ist ein 8-Bit Binärwert aller 8 Eingänge. nr=1..8: wert ist 0 oder 1 je nach Zustand des Eingans nr
ESC	**D**	3	nr	LO-wert	HI-wert			Nach dem Befehl 'ESC V D nr' wird der aktuelle Analogwert vom Analogeingang nr=1 oder 2 gesendet. (wert = 0..5000 mV)
ESC	**W**	anz	nr	Zeichenkette Benutzerwert...				Nach dem Befehl 'ESC V S nr' wird der aktuelle Analogwert vom Analogeingang nr=1 oder 2 als formatierter Benutzer gesendet (Stringlänge = anz-1).
ESC	**V**	anz	Zeichenkette Version...					Nach dem Befehl 'ESC S V' wird die Version der eDIP-Firmware als Zeichenkette gesendet. z.B. "EA eDIPTFT43-A V1.0 Rev.A TP+"
ESC	**J**	anz	Zeichenkette Projektname...					Nach dem Befehl 'ESC S J' wird der Makro-Projektname als Zeichenkette gesendet. z.B. "init / delivery state"
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 = Hardwarerevisonsbuchstabe Touchinfo: LO-Byte = '-|+' X-Richtung erkannt; HI-Byte = '-|+' Y-Richtung erkannt DFanz: Anzahl benutzter Bytes im Dateflash (3 Byte: LO-, MID- HI-Byte)
Antworten ohne Längenangabe (anz)								
ESC	**U**	L	xx1	yy1	Bilddaten... (G16-FORMAT)			Nach dem Befehl 'ESC UH....' wird ein Hardcopy im G16-Format gesendet. xx1,yy1 = Startkoordinaten des Hardcopy (Linke obere Ecke), die Längenangabe ist im G16-Format enthalten

Technische Änderung vorbehalten.
Wir übernehmen keine Haftung für
Druckfehler und Applikationsbeispiele.

VORGELADENE FONTS

Es sind standardmäßig 3 monospaced, 3 proportionale Zeichensätze und 2 grosse Ziffernfonts 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 ist möglich.
Die Makroprogrammierung erlaubt die Einbindung von weiteren Fonts. Es können alle nur erdenklichen Schriften aus True-Type Fonts gerastert und über den eDIPTFT-Compiler*) geladen werden (USB-Programmer EA 9777-1USB notwendig).

Upper \ + Lower	$0 (0)	$1 (1)	$2 (2)	$3 (3)	$4 (4)	$5 (5)	$6 (6)	$7 (7)	$8 (8)	$9 (9)	$A (10)	$B (11)	$C (12)	$D (13)	$E (14)	$F (15)
$20 (dez: 32)		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
$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	{	|	}	~	⌂
$80 (dez: 128)	€	ü			ä										ß	
$90 (dez: 144)					ä					ß	ü				ƒ	

Font 1: 4x6 monospaced

Upper \ + Lower	$0 (0)	$1 (1)	$2 (2)	$3 (3)	$4 (4)	$5 (5)	$6 (6)	$7 (7)	$8 (8)	$9 (9)	$A (10)	$B (11)	$C (12)	$D (13)	$E (14)	$F (15)
$20 (dez: 32)		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
$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	{	|	}	~	⌂
$80 (dez: 128)	€	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
$90 (dez: 144)	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	¢	£	¥	ß	ƒ
$A0 (dez: 160)	á	í	ó	ú	ñ	Ñ	ª	º	¿	⌐	¬	½	¼	¡	«	»
$B0 (dez: 176)																
$C0 (dez: 192)																
$D0 (dez: 208)																
$E0 (dez: 224)	α	ß	Γ	π	Σ	σ	µ	τ	Φ	Θ	Ω	δ	φ	ϕ	Ε	∩
$F0 (dez: 240)	≡	±	≥	≤	⌠	⌡	÷	≈	°	•	·	√	ⁿ	²	³	¯

Font 2: 6x8 monospaced

Upper \ + Lower	$0 (0)	$1 (1)	$2 (2)	$3 (3)	$4 (4)	$5 (5)	$6 (6)	$7 (7)	$8 (8)	$9 (9)	$A (10)	$B (11)	$C (12)	$D (13)	$E (14)	$F (15)
$20 (dez: 32)		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
$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	{	|	}	~	⌂
$80 (dez: 128)	€	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
$90 (dez: 144)	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü					
$A0 (dez: 160)	á	í	ó	ú	ñ	Ñ	ª	º								
$B0 (dez: 176)																
$C0 (dez: 192)																
$D0 (dez: 208)																
$E0 (dez: 224)	α	ß	Γ	π	Σ	σ	µ	τ	Φ	Θ	Ω	δ	φ	ϕ	Ε	∩
$F0 (dez: 240)	≡	±	≥	≤	⌠	⌡	÷	≈	°	•	·	√	ⁿ	²	³	¯

Font 3: 7x12 monospaced

Upper \ + Lower	$0 (0)	$1 (1)	$2 (2)	$3 (3)	$4 (4)	$5 (5)	$6 (6)	$7 (7)	$8 (8)	$9 (9)	$A (10)	$B (11)	$C (12)	$D (13)	$E (14)	$F (15)
$20 (dez: 32)		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
$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	å	ç	ê	ë	è	ï	î	ì	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	ª	º								
$B0 (dez: 176)																
$C0 (dez: 192)																
$D0 (dez: 208)																
$E0 (dez: 224)		ß														
$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	{	|	}	~	△
$80 (dez: 128)	€	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
$90 (dez: 144)	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü					
$A0 (dez: 160)	á	í	ó	ú	ñ	Ñ	ª	º								
$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	{	|	}	~	△
$80 (dez: 128)	€	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
$90 (dez: 144)	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü					
$A0 (dez: 160)	á	í	ó	ú	ñ	Ñ	ª	º								
$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 BigZif50

+ 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 8: grosse Ziffern BigZif100

Diese Schriften sind im Auslieferungszustand integriert

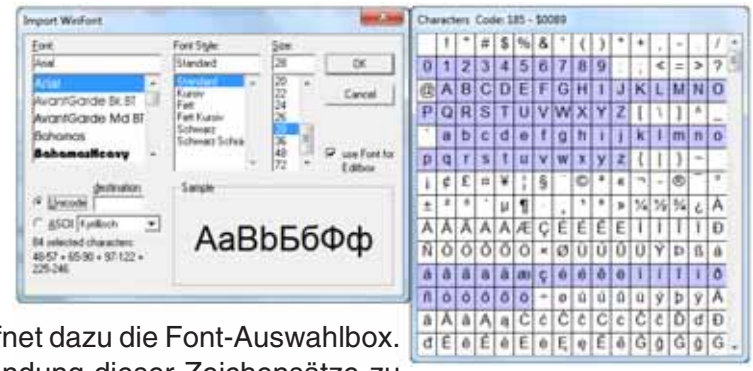
LADBARE ZEICHENSÄTZE

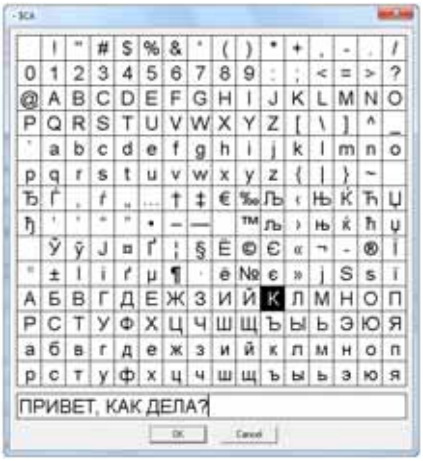
Bis zu 256 Fonts á 16 Pages können im internen DatenFlash abgelegt werden.

Technische Änderung vorbehalten.
Wir übernehmen keine Haftung für
Druckfehler und Applikationsbeispiele.

COMPILERANWEISUNG "WinFont:"

Damit ist es möglich, TrueType-Fonts in verschiedenen Größen zu rastern und einzubinden. Sie können entweder den kompletten Zeichensatz (ASCII) einbinden oder Sie wählen aus dem gesamten Unicode-Zeichensatz bestimmte Zeichen aus. Ein Doppelclick im KitEditor auf den Fontnamen öffnet dazu die Font-Auswahlbox. Um die Verwendung dieser Zeichensätze zu vereinfachen gibt es die kompfortable Möglichkeit einer Zeichen-Auswahlbox. Wird im KitEditor ein String ausgegeben (z.B. #ZL 5,5, "Hallo") kann mit einem Doppelclick auf den String diese geöffnet werden. Es können nun die gewünschten Zeichen ausgewählt werden. Dies ist vor allem bei kyrillischen, asiatischen oder Symbolschriftarten zu empfehelen. Der KitEditor setzt darauf hin automatischen den richtigen ASCII-Code ein. Alternativ zu den Anführungsstrichen können geschweifte Klemmern genutzt werden (z.B. +ZL5,5, {48616C6C6F}).





COMPILERANWEISUNG "Font:"

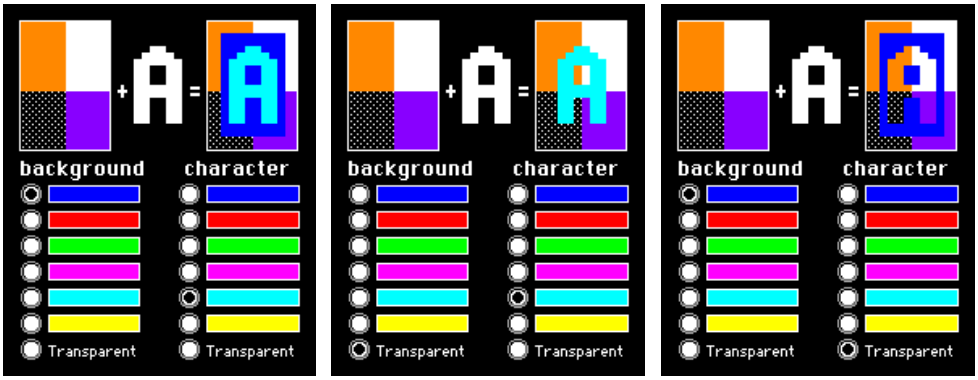
Verwendet werden können folgende Font-Formate:
- FXT: Textfont von eDIP240/eDIP320 und KIT-Serie
- G16: internes eDIPTFT-Format (damit sind auch bunte Zeichensätze möglich)

65.536 DARSTELLBARE FARBEN

Beim EA eDIPTFT32-A sind 65.536 Farben darstellbar. Damit können beliebige Farbilder/Icons und Animationen angezeigt werden.

Color	R	G	B
1	0	0	0
2	0	0	255
3	255	0	0
4	0	255	0
5	255	0	255
6	0	255	255
7	255	255	0
8	255	255	255
9	111	111	111
10	255	143	0
11	143	0	255
12	255	0	143
13	0	255	143
14	143	255	0
15	0	143	255
16	175	175	175

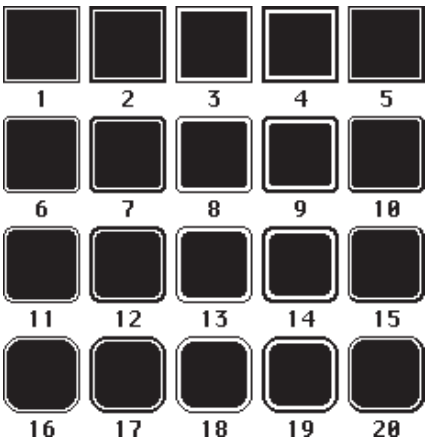
Für den einfachen Zugriff auf Farben für Zeichenfunktionen existiert eine Arbeitsfarbpalette mit 32 Einträgen (16 Farben sind nach PowerOn/Reset vordefiniert). Diese Farbpalette kann beliebig umdefiniert werden (Befehl: ESC FP nr R G B) ohne bereits getätigte Ausgaben zu verändern.
Bei der Einstellung der Farbe für Zeichenbefehle wird eine Farbnummer zwischen 1 und 32 angegeben. Wird die Dummy Farbnummer 255 angegeben so wird die aktuell eingestellte Farbe nicht verändert. Dies ist z.B. bei Einstellung der Vorder- und Hintergrundfarbe nützlich, um nur eine Farbe verändern zu können.
Die Farbnummer 0=Transparent hat dabei eine Sonderstellung, wird Transparent als Hintergrund für z.B. Zeichenketten eingestellt so werden die Buchstaben ohne Hintergrund gezeichnet d.h. der vohandene Hintergrund bleibt erhalten.



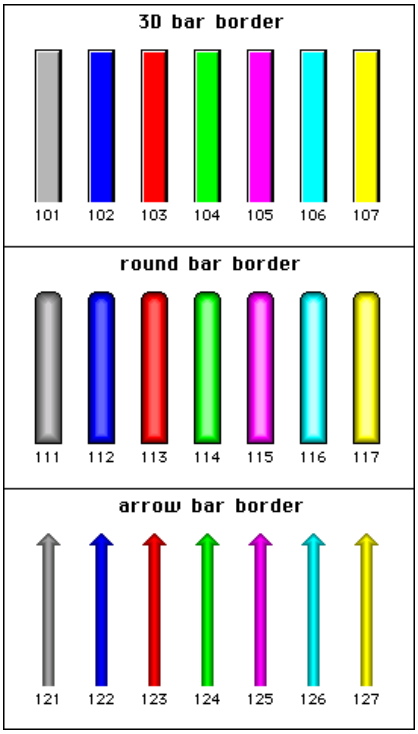
Beispiele zur Darstellung des Zeichens 'A' auf einen vorhandenen Hintergrund

RAHMEN, TASTENFORMEN UND BARGRAPH

Mit dem Befehl *Rahmen zeichnen* sowie beim Zeichnen von Touchtasten und Bargraphs kann ein Rahmentyp eingestellt werden.



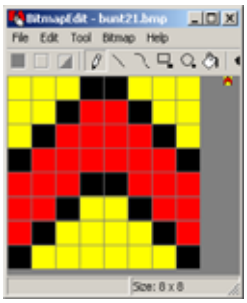
Es stehen dabei 20 vorgeladene Rahmentypen (nr: 1..20) und drei spezielle Rahmen für Bargraphs in verschiedenen Farben zur Verfügung (nr:101..107, 111..117 und 121..127).
Diese Rahmen können in beliebiger Grösse durch Angabe der Rechteck-Koordinaten gezeichnet werden.
Die Rahmen 1..20 sind aus 3 Teilen aufgebaut: Der äußere und der innere Rahmen sowie die Füllung.
Bei diesen Rahmen kann jedem dieser Teilbereiche kann eine eigene Farbe zugewiesen werden.



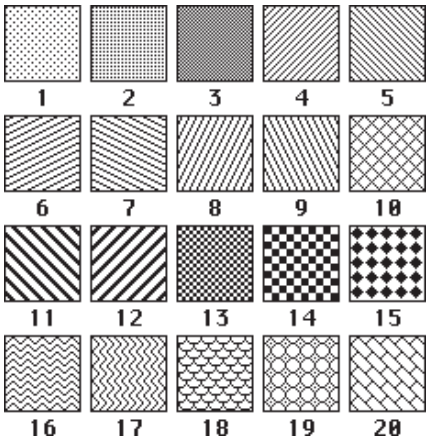
FÜLLMUSTER

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 20 vordefinierte Füllmuster zur Verfügung.

Eigene Muster:



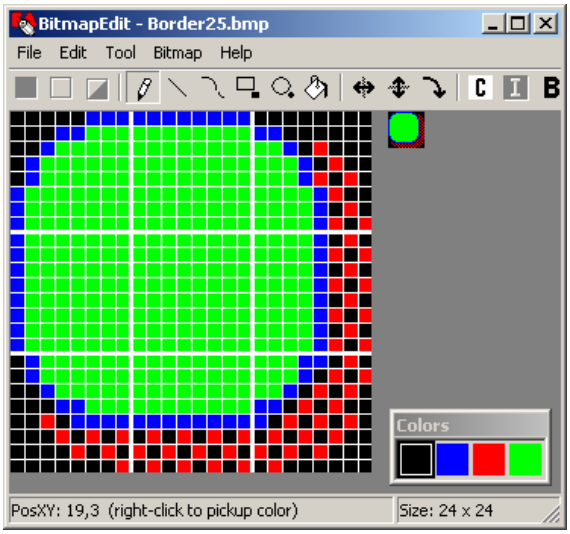
Über die LCD-Tools ist es möglich eigene Muster (8x8 Pixel große Bitmaps) einzubinden (Compileranweisung "`Pattern:`").
Bei monochromen Muster (wie die 20 vorgeladenen Füllmuster) kann die Vorder- und Hintergrundfarbe (inkl. Transparenz) frei eingestellt werden.
Es können auch mehrfarbige Füllmuster eingebunden werden, die in der Farbe nachträglich nicht veränderbar sind. Mit den LCD-Tools sind einige Muster im Verzeichnis *'Pattern'* installiert worden.



Technische Änderung vorbehalten.
Wir übernehmen keine Haftung für
Druckfehler und Applikationsbeispiele.

EIGENE RAHMEN

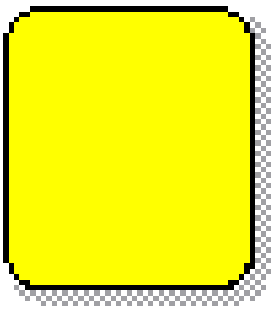
Über die LCD-Tools ist es möglich eigene Rahmentypen einzubinden (Compileranweisung "`Border:`"). Jeder dieser zusätzlichen Rahmen besteht aus einem 24x24 Pixel großem Bitmap.



(Aufbau: 9 Segmente mit je 8x8 Pixel: 4xEcken, 4xMittelstücke, 1xFüllung).

Die Skalierbarkeit wird durch Wiederholung der 8x8 Pixel grossen Mittelstücke/Füllung erreicht.

Bei 4-farbigen Bitmaps (wie die vorgeladenen Rahmentypen 1..20) können 3 Rahmenfarben jederzeit eingestellt werden. Der erste Paletteneintrag ist die Transparenzfarbe und wird im eDIP nicht benutzt.



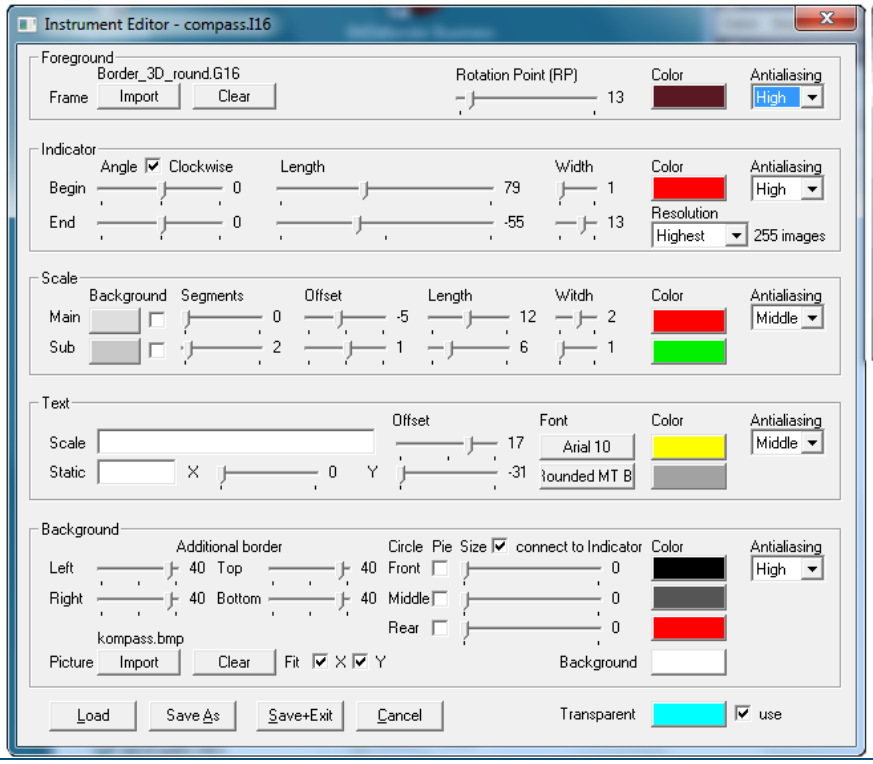
border25: 50x56 Pixel Grösse

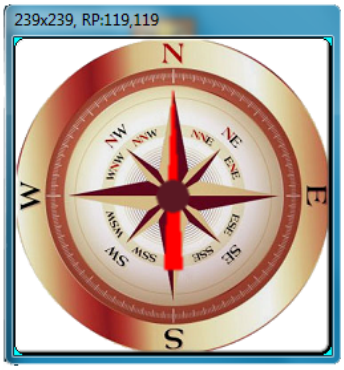
Es können auch mehrfarbige Rahmen (wie z.B die Bargraphtypen 101..127) eingebunden werden, die in der Farbe nachträglich nicht veränderbar sind. Mit den LCD-Tools sind einige Beispielrahmen im Verzeichnis *'Bitmaps\Color\Border'* installiert worden.

DREH- UND ZEIGERINSTRUMENTE (ab V1.4)

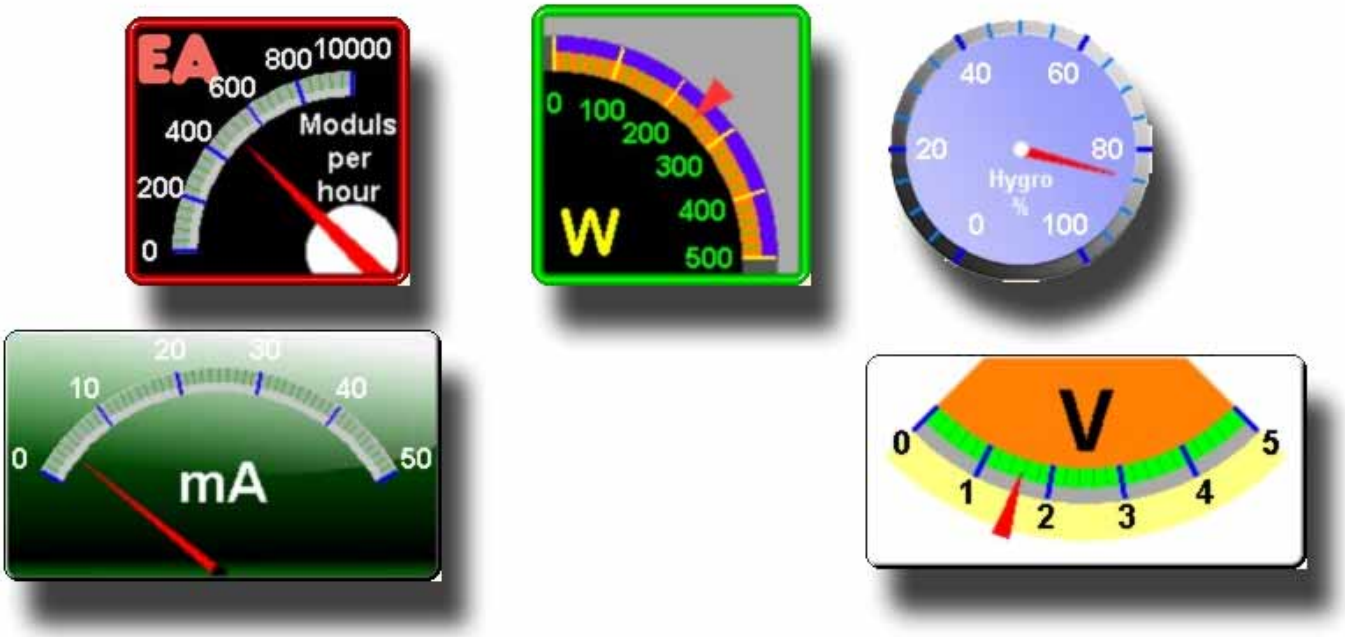
Über die LCD-Tools ist es möglich Instrumente einzubinden (Compileranweisung `Instrument: 4, <instrument.i16>`). Ein Doppelclick im KitEditor auf das Instrumentenfile öffnet dazu die Instrument Einstellbox.

Die Instrumente werden mit komfortablen Funkionen unterstützt ('ESC I..'). Den Instrumenten kann z.B. ein Analogkanal zugewiesen werden. Zudem ist es möglich die Instrumente für die Toucheingabe zu konfigurieren.





Einige Beispiele für Instrumente:

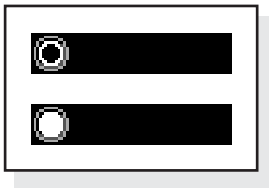


BUTTONS ALS TOUCHTASTEN

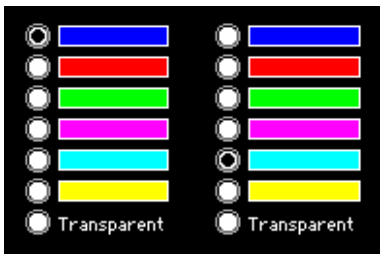
Ausser den Rahmentypen, die in der Grösse frei skalierbar sind, gibt es noch die Möglichkeit beliebige Bitmaps als Touch-Tasten oder -Schalter zu verwenden (Compileranweisung "`Button:`").
Ein Button besteht aus einem oder zwei gleich grossen Bildern. Bei zwei Bildern wird das erste Bild für normale Darstellung und das zweite Bild für die gedrückte Darstellung der Touchtaste verwendet.
Die aktive Fläche der Touchtaste ergibt sich automatisch aus der Grösse der Bitmaps.
Mit den LCD-Tools sind einige Beispieltasten im Verzeichnis *'Button'* installiert worden.

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.



RadioBlack75x15_0.bmp
RadioBlack75x15_1.bmp



zwei Radio-Gruppen mit Touch-Schaltern

Technische Änderung vorbehalten.
Wir übernehmen keine Haftung für
Druckfehler und Applikationsbeispiele.

ERSTELLEN INDIVIDUELLER FONTS, MAKROS UND BILDER

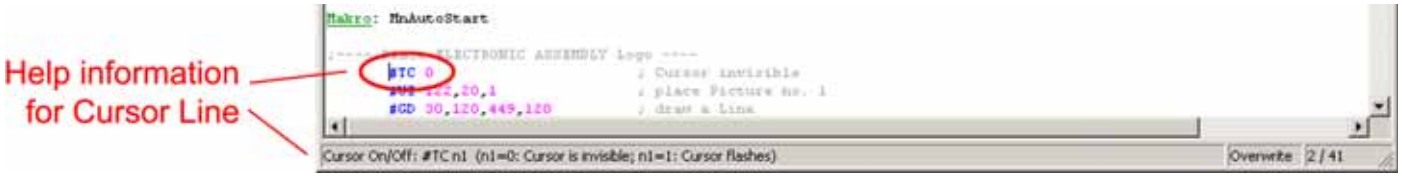
Um nun Ihre speziellen Fonts, Makros und Bilder erstellen zu können, benötigen Sie folgende Hilfsmittel:

- um das Display an den PC anschliessen zu können benötigen Sie den als Zubehör erhältlichen USB-Programmer EA 9777-2USB oder einen selbstgebauten Adapter mit Pegelwandler MAX232 (Applikationsbeispiel siehe S. 5).
- die Software ELECTRONIC ASSMBLY LCD-Tools*); sie enthält einen Kit-Editor, Bitmap-Editor und eDIPTFT-Compiler, sowie Fonts, Bilder, Rahmen, Muster und Beispiele (für PC-Win)
- einen PC mit USB oder serieller Schnittstelle COMx

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/Bilder eingebunden werden und in welchen Makros welche Befehlsfolgen stehen sollen.
Sind die Makros über den Kit-Editor definiert, startet man über F5 den eDIPTFT-Compiler. Dieser erzeugt eine Datei DEMO.DF, ist ein Programmer EA 9777-2USB angeschlossen, oder das Display über einen MAX232 an den PC angeschlossen, dann wird diese Datei in das DatenFlash des Displays gebrannt.
Sie können die vom eDIPTFTcompiler erstellte Datei *.df auch unter einem beliebigen System zum eDIP senden. Dazu übertragen Sie den Inhalt der *.df Datei 1:1 (mit Smallprotokoll in Paketen) zum eDIP. In dieser Datei sind alle Programmierbefehle enthalten.

HILFE IM KIT-EDITOR (ELECTRONIC ASSEMBLY LCD TOOLS)

In der Statuszeile am unteren Rand des Editorfensters werden für den aktuellen Befehl mögliche Parameter kurz erläutert. Der Cursor muss dazu in der entsprechenden Zeile stehen.
Für mehr Informationen drücken Sie F1.



*) im Internet unter http://www.lcd-module.de/deu/dip/edip.htm

BILDER

Um die Übertragungszeiten der Schnittstelle zu verkürzen, oder auch um Speicherplatz im Prozessorsystem zu sparen, können bis zu 256 Bilder á 16 Pages im internen DatenFlash abgelegt werden (Compileranweisung "`Picture:`").
Verwendet werden können folgende Bild-Formate:
- BMP: Windows Bitmap mit 1-, 4-, 8-, 16-, 24-, 32-BIT Farbtiefe inkl. RLE-Codierung.
- GIF: Graphics Interchange Format Inkl. Unterstützung der Transparenz
- JPG: JPEG Compressed Images
- TGA: TARGA Images mit 8-, 16-, 24-, 32-BIT Farbtiefe inkl. RLE-Codierung und Transparenz.
- G16: internes eDIPTFT-Format, Inkl. Beachtung der Transparenz

Die Bilder werden immer im internen G16 Bildformat, komprimiert abgepeichert (spart Speicherplatz).
Zu grosse Bilder werden proportional verkleinert (Compileranweisung "`MaxSize:`").
Zudem kann die Farbtiefe umgerechnet werden (Compileranweisung "`MaxColorDepth:`").
Der Aufruf der Bilder erfolgt über den Befehl "ESC U I" über die Schnittstelle oder aus einem Makro heraus. Bei monochromen Bildern kann die Vorder- und Hintergrundfarbe (inkl. Transparenz) frei eingestellt werden.

ANIMATIONEN

Bis zu 256 Animationen á 16 Pages können im internen DatenFlash abgelegt werden. (Compileranweisung "`Animation:`").
Verwendet werden können folgende Bild-Formate:
- GIF: animiertes GIF (nur identische Transparenzbereiche, Transparenz ist abschaltbar).
- G16: internes animiertes eDIPTFT-Format
- mehrere Einzel-Bitmaps (BMP, GIF, JPG, TGA, G16) z.B. 2 Bitmaps als Blinkfunktion

Bis zu 4 Animationen können gleichzeitig definiert werden. Die Animationen laufen dann automatisch ab, sie können aber auch manuell beeinflusst werden.
Bei monochromen Animationen kann die Vorder- und Hintergrundfarbe frei eingestellt werden.

FÜLLMUSTER

Es können bis zu 255 Füllmuster á 16 Pages im internen DatenFlash abgelegt werden (Compileranweisung "`Pattern:`").
Jedes 8x8 Pixel grosse Bild (BMP, GIF, JPG, TGA, G16) kann als Füllmuster importiert werden.
Bei monochromen Füllmusterrn kann die Vorder- und Hintergrundfarbe (inkl. Transparenz) frei eingestellt werden.

RAHMEN / BARGRAPH

Es können bis zu 255 Rahmen á 16 Pages im internen DatenFlash abgelegt werden (Compileranweisung "`Border:`").
Jedes 24x24 Pixel grosse Bild (BMP, GIF, JPG, TGA, G16) kann als Rahmen importiert werden.
Bei 4-farbigen Bitmaps können die Rahmenfarben jederzeit eingestellt werden. Der erste Paletteneintrag ist die Transparenzfarbe und wird im eDIP nicht benutzt.
Bei GIF, TGA und G16 Bilder wird die definierte Transparenzfarbe beachtet.
Für Touchtasten kann optional ein zweiter Rahmen (gedrückte Taste/Schalter) angegeben werden.

BILDER ALS TOUCHTASTEN (BUTTONS)

Es können bis zu 256 Touchtasten/Buttons á 16 Pages im internen DatenFlash abgelegt werden (Compileranweisung "`Button:`").
Ein Button besteht aus einem oder zwei gleich grossen Bildern (BMP, GIF, JPG, TGA, G16).
Bei zwei Bildern wird das erste Bild für normale Darstellung und das zweite Bild für die gedrückte Darstellung der Touchtaste verwendet.
Bei GIF, TGA und G16 Bilder wird die definierte Transparenzfarbe beachtet.

Technische Änderung vorbehalten.
Wir übernehmen keine Haftung für
Druckfehler und Applikationsbeispiele.

MAKROS

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

Normal Makro `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 Touch-, Port-, Matrixmakro mit entsprechendem Return-Code ausgelöst wird.

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

Bit Makro `BitMakro:`

Start bei Anlegen/Änderung einer Spannung an einzelnen Eingängen IN 1..8 (Bitweise) oder per Befehl 'ESC MB xx'. Die Bit-Makros 1..8 reagieren auf fallende Flanke, Bit-Makros 9..16 auf die steigende Flanke der Eingänge 1..8. Ab Firmware V1.1 kann mit dem Befehl 'ESC YD n1 n2 n3' die Zuordung der Eingänge zu den Bitmakros umdefiniert werden (siehe Seite 17).

Port Makro `PortMakro:`

Start bei Anlegen/Änderung einer Spannung an den 8 Eingängen IN 1..8 (binär kombiniert) oder per Befehl 'ESC MP xx'.

Matrix Makro `MatrixMakro:`

Matrix-Makro 1..64: Start beim Drücken einer Taste oder per Befehl 'ESC MX xx'. Matrix-Makro 0: Start beim Loslassen wenn keine Taste mehr gedrückt ist oder per Befehl. Ab Firmware V1.1 kann mit dem Befehl 'ESC YX n1 n2' die Zuordung der Tasten zu den Matrixmakros umdefiniert werden (siehe Seite 17).

Analog Makro `AnalogMakro:`

automatischer Start bei Änderung des Anlogwertes AIN1 oder AIN2 oder per Befehl 'ESC MV xx'. Siehe Tabelle nebenan: Ab Firmware V1.1 kann mit dem Befehl 'ESC VM n1 n2' die Zuordung der Analogmakros umdefiniert werden (siehe S. 15).

Analog Makros

Makro Nr. AIN1	Makro Nr. AIN2	Start des Makros bei
0	10	jeder Änderung des Analogwertes
1	11	fallendem Analogwert
2	12	steigendem Analogwert
3	13	kleiner unterer Grenzwert
4	14	größer unterer Grenzwert
5	15	kleiner oberer Grenzwert
6	16	größer oberer Grenzwert
7	17	Ausserhalb beider Grenzwerte
8	18	Innerhalb beider Grenzwerte
9	19	kleiner als anderer Analogkanal

Prozess Makro `ProzessMakro:`

automatischer Start in bestimmten Zeitintervallen (0,1s bis 25,5s) oder per Befehl 'ESC MC xx'. Bis zu 4 unabhängige Prozesse können mit dem Befehl 'ESC MD ..'. definiert werden. Prozess-Makros werden nicht durch andere Befehle unterbrochen.

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 <4,6V.

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 DPOM:
-PowerOff - Pin13 (DPOM) auf GND legen
-PowerOn - Pin13 (DPOM) wieder öffnen.

SCHREIBSCHUTZ FÜR MAKROPROGRAMMIERUNG

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

MAKRO PAGES (MEHRSPRACHIGKEIT)

Für die Fonts / Bilder und Makros 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]  <BIER.BMP>
PICTURE: 100[1]  <BEER.BMP>
PICTURE: 100[2]  <BIRRA.BMP>

MACRO: 2[0]                    ; SAME AS "MACRO: 2"
        #FZ 3,1
        #ZL 25,0 "DEUTSCH  "
        #UI 0,20, 100

MACRO: 2[1]                    ; ENGLISH
        #FZ 3,1
        #ZL 25,0 "ENGLISH  "
        #UI 0,20, 100

MACRO: 2[2]                    ; ITALIAN
        #FZ 3,1
        #ZL 25,0 "ITALIAN  "
        #UI 0,20, 100
```

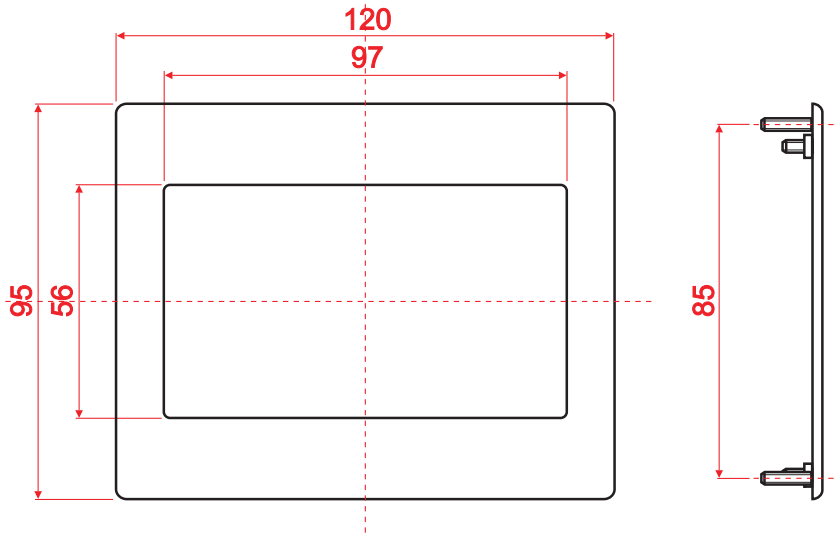
Technische Änderung vorbehalten.
Wir übernehmen keine Haftung für
Druckfehler und Applikationsbeispiele.

SPEZIFIKATION UND GRENZWERTE

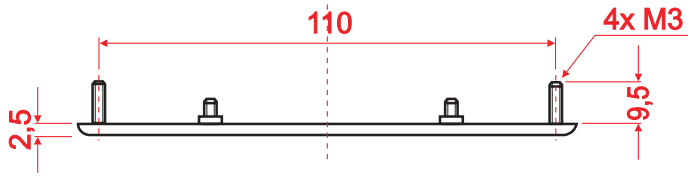
Characteristics					
Value	**Condition**	**min.**	**typ.**	**max.**	**Unit**
Operating Temperature		-20		+70	°C
Storage Temperature		-30		+80	°C
Storage Humidity	< 40°C			90	%RH
Operating Voltage		4.9	5.0	5.1	V
Input Low Voltage		-0.5		0.3*VDD	V
Input High Voltage	Pin Reset only	0.9*VDD		VDD+0.5	V
Input 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		4.2			V
Brightness (white)	w./o. Touch		500		cd/m²
	with Touch		410		cd/m²
Output Current				20	mA
Power Supply	Backlight 100%		180		mA
	Backlight off		80		mA

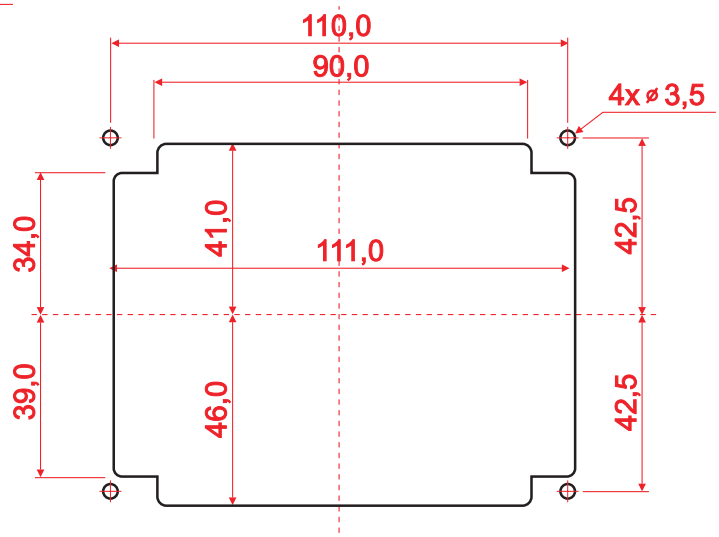
EINBAUBLENDE EA 0FP481-43SW

Als Zubehör liefern wir optional eine schwarz eloxierte Einbaublende aus Aluminium.
Die Montagelaschen sind im Lieferumfang vom EA eDIPTFT43-A enthalten.



alle Maße in mm





Frontplattenauschnitt

HINWEISE ZUR HANDHABUNG UND ZUM BETRIEB

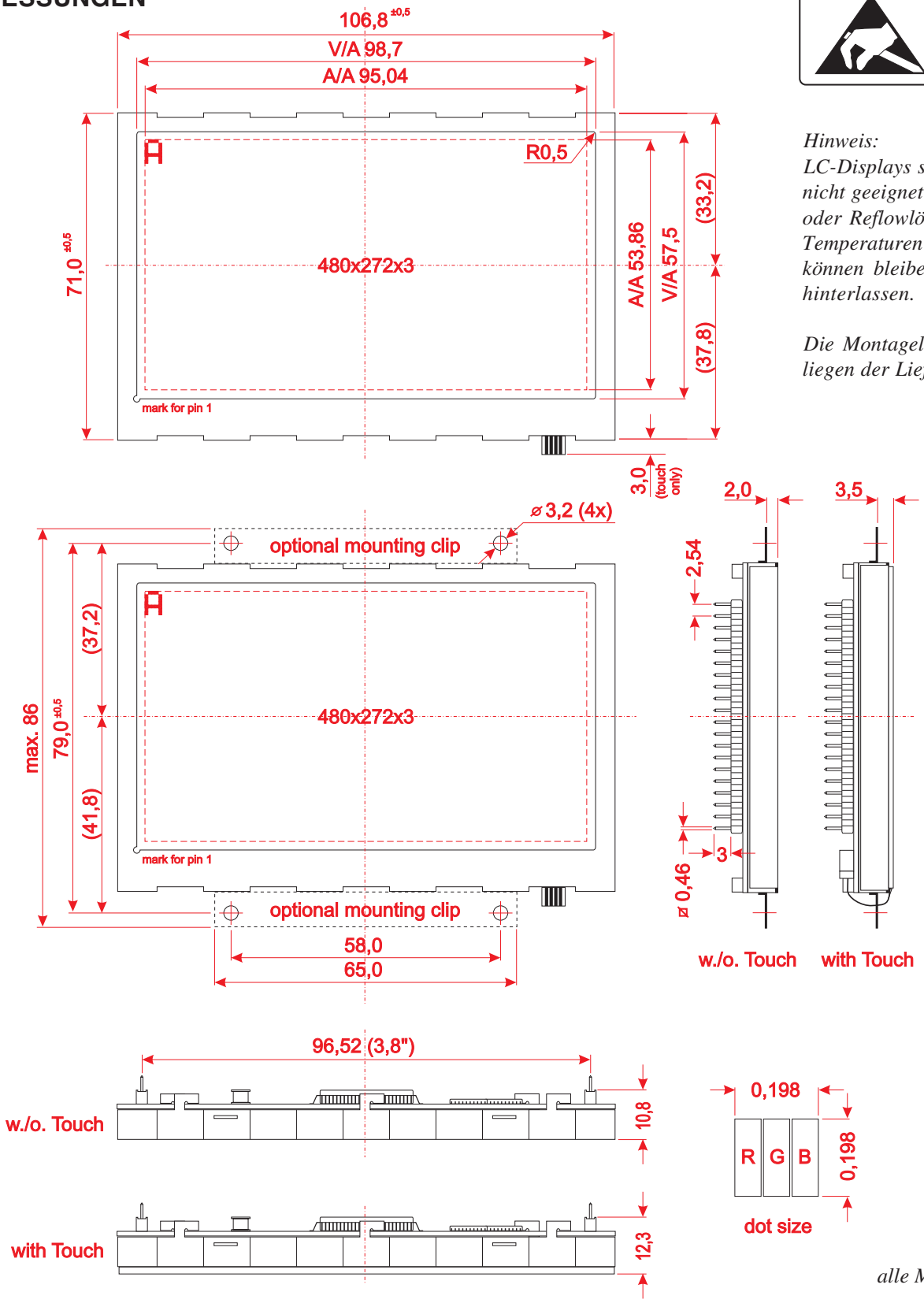
- Zur elektrischen Zerstörungs des Moduls kann führen: Verpolung oder Überspannung der Stromversorgung, Überspannung oder Verpolung bzw. statische Entladung an den Eingängen, Kurzschließen der Ausgänge.
- Vor dem Abstecken des Moduls muß unbedingt die Stromversorgung abgeschaltet sein. Ebenso müssen 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°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.

Technische Änderung vorbehalten.
Wir übernehmen keine Haftung für
Druckfehler und Applikationsbeispiele.

ABMESSUNGEN

Hinweis:
LC-Displays sind generell nicht geeignet für Wellen- oder Reflowlötung. Temperaturen über 80°C können bleibende Schäden hinterlassen.

Die Montagelaschen liegen der Lieferung bei.



alle Maße in mm