embedded 7.0" TFT-DISPLAY 800x480 BUILT-IN INTELLIGENCE

WORLD NEW!



Dimension: 170x112x13,9mm

TECHNICAL DATA

- * TFT-GRAPHIC DISPLAY WITH BUILT-IN GRAPHIC FUNCTIONS
- * 800x480 DOTS, 16-BIT COLOR (65.536 COLORS) WITH LED-BACKLIGHT
- * 4MB ON BOARD FLASH FOR FONTS, PICTURES, ANIMATIONS AND MACROS
- * SIMPLE SUPPLY POWER +5V / 700mA
- * 8 PRE-DEFINED FONTS, CAN BE EXPANDED
- * TRUETYPE-FONTS IMPORTABLE, TURNABLE IN 90° STEPS
- * 3 DIFFERENT INTERFACES ON BOARD: RS-232, I2C-BUS OR SPI-BUS
- * DRAW LINE, PLACE A DOT, AREA, BARGRAPH... ACCURATE TO THE PIXEL
- * ROTARY AND POINTER INSTRUMENTS
- * MIX TEXT, GRAPHIC, PICTURES AND ANIMATIONS
- * LUXOURIOUS KEYBOARD FUNCTIONS
- * MULTI-LINGUAL WITH MACRO PAGES AND STRINGTABLES
- * BACKLIGHT BRIGHTNESS BY SOFTWARE
- * ANALOGUE TOUCH PANEL: VARIABLE GRID
- * FREE DEFINABLE KEY AND SWITCH
- * 8 DIGITAL IN- AND 8 DIGITAL OUTPUTS
- * 2 ANALOGUE INPUTS, COMFORTABLE TO USE
- * 80 DIFFERENT SMALL JINGLES AS AUDIO FEEDBACK

ORDERING CODES

DISPLAYS

TFT 800x480 DOTS, WHITE LED BACKLIGHT AS ABOVE, BUT WITH TOUCH PANEL STARTERKIT

INCLUDES EA eDIPTFT70-ATP AND EVALUATION BOARD WITH USB FOR DIRECT CONNECTION TO PC AND INTERFACE BOARDS FOR CONNECTION WITH YOUR HOST SYSTEM ADDTIOTNAL PARTS

MOUNTING BEZEL (ALUMINIUM), BLACK ANODIZED SOCKET 1x24, 7.1mm HIGH (1 piece)

EA eDIPTFT70-A
EA eDIPTFT70-ATP

EA EVALeDIPTFT70

EA 0FP801-70SW EA B-24



Page 2

Documentation of revision											
Date	Туре	Old	New	Reason / Description							
May, 2011	0.1			preliminary version							

CONTENTS

GENERAL	3
RS-232	4
RS-485, USB	5
SPI	6
I ² C	7
ANALOGUE / DIGITAL IN- AND OUTPUT	8
MATRIX KEYPAD	9
SOFTWARE PROTOCOLL	10 - 11
TERMINAL MODE, COMMAND TRANSFER	12
COMMANDS / FUCTIONS IN TABULAR FORMAT	13 - 21
TOUCH PANEL	18 -1 9
RESPONSES OF THE CONTROL PANEL	21
CHARACTER SET	22 - 23
COLORS	24
FRAMES, KEY STYLE, BARGRAPH AND PATTERN	25
INSTRUMENTS	26 - 27
PROGRAMMING FONTS, PICTURES, ANIMATIONS	28
BITMAPS AS BUTTONS	29
MACROS, MULTI-LINGUAL, MACRO PAGES	30 - 31
SOUNDS	32
KEYBOARD	33
ELECTRICAL CHARACTERISTICS	34
DIMENSION, MOUNTING PANEL	35 - 36



GENERAL

The EA eDIP series of displays are the world's first displays with integrated intelligence. In addition to a variety of integrated fonts that can be used with pixel accuracy, they offer a whole range of sophisticated graphics functions.

They are controlled via one of the 3 integrated interfaces: RS-232, SPI or I²C. The displays are "programmed" by means of high-level language-type graphics commands. There is no longer any need for the time-consuming programming of character sets and graphics routines. The ease of use of this display with its touch panel dramatically reduces development times.

HARDWARE

The display is designed to work at an operating voltage of +5V.

Data transfer is either serial and asynchronous in RS-232 format or synchronous via the SPI or I²C specification. To improve data security, a simple protocol is used for all types of transfer.

ANALOGUE TOUCH PANEL

All versions are also available with an integrated touch panel: You can make entries and menu or bar graph settings by touching the display. The labeling of the "keys" is flexible and can also be changed during runtime (different languages, icons). The drawing of the individual "keys" and the labeling is handled by the integrated software.

LED ILLUMINATION

All displays are equipped with modern, energy-saving LED illumination. Brightness can be varied 0~100% by command.

In 24-hour operation, the illumination should be dimmed or switched off as often as possible to increase their lifetime.

SOFTWARE

This display is programmed by means of commands, such as $Draw \ a \ rectangle \ from (0,0) \ to (799,479)$. No additional software or drivers are required. Strings and images can be placed with **pixel accuracy**. Text and graphics can be combined at any time. Different character sets can be used at same time. Each character set and the images can be zoomed from 2 to 8 times and rotated in 90° steps. With the largest character set, the words and numbers displayed will fill the screen.

ACCESSORIES

Evaluation-Board (EA EVALeDIPTFT70) to write to internal data flash memory

The display is shipped fully programmed and with all fonts. The additional Evaluation-Board is thus generally not required.

However, if the internal character sets have to be changed or extended, or if images or macros have to be stored internally, the Evaluation-Board EA 9777-3USB, which is available as an accessory, will burn the data/images you have created into the on-board data flash memory (4 MB) permanently.

The Evaluation-Board runs under Windows and is connected to the PC's USB interface. It is shipped with an interface cable and the installation software. The Evaluation-Board is equipped with serveral LEDs, pushbottons and potentiometer to test all peripherial modes of the eDIP.

Interface-Expansion for Evaluation-Board EA 9777-2PE (included in the Starter Kit):

With the expansion EA 9777-2PE for the Evaluation-Board all interfaces of the display are made available with the help from small adapter boards: RS-232, RS-485, SPI, I²C, RS-232 (CMOS level). Further information you will find in the datasheet of the Evalution-Board.



Page 4

RS-232 INTERFACE

If the display is wired as shown below, the RS-232 interface is selected. The pin assignment is specified in the table on the right.

The RxD and TxD lines lead CMOS level (5V) for direct connection to a microcontroller for example.

If "genuine" RS-232 levels are required (e.g. for connection to a PC), an external level converter (e.g. MAX232) is required.

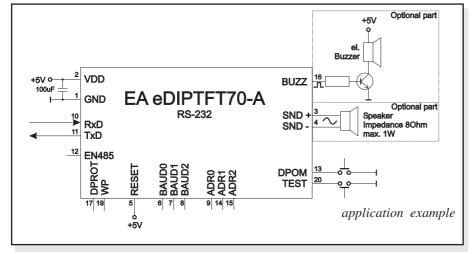
			Pinout eDIPTFT70-A: RS	S-2	32/	RS-485 mod	е	
Pin	Symbol	In/Out	Function		Pin	Symbol	In/Out	Function
1	GND		Ground Potential for logic (0V)		25			
2	VDD		Power supply for logic (+5V)		26	NC NC		do not connect, reserved
3	SND+		Speaker LS1 (Impedance 8 Ohm)		27	NC NC		do not connect, reserved
4	SND-		Speaker LS2		28	<u>-</u> '		
5	RESET	In	L: Reset		29	GND		Ground (=Pin 1)
6	BAUD0	In	Baud Rate 0		30	VDD		Power supply (=Pin 2)
7	BAUD1	In	Baud Rate 1		31	AIN1	In	analogue input 0VDD
8	BAUD2	In	Baud Rate 2		32	AIN2	111	DC impedance 1MOhm
9	ADR0	In	Address 0 for RS-485		33	OUT1 / MO8		8 digital outputs
10	RxD	In	Receive Data		34	OUT2 / MO7		maximum current:
11	TxD	Out	Transmit Data		35	OUT3 / MO6		IOL = IOH = 10mA
12	EN485	Out	Transmit Enable for RS-485 driver		36	OUT4 / MO5		
13	DPOM	In	L: disable PowerOnMacro do not connect for normal operation		37	OUT5 / MO4	Out	alternativ up to 8 matrix keyboard output lines
14	ADR1	In	Address 1 for RS-485		38	OUT6 / MO3		(reduces the digital output
15	ADR2	In	Address 2 for RS-485		39	OUT7 / MO2		lines, see chapter external
16	BUZZ	Out	Buzzer output		40	OUT8 / MO1		keyboard)
17	DPROT	In	L: Disable Smallprotokoll do not connect for normal operation		41	IN1 / MI8		
18	DNC	Out	L: internal, do not connect		42	IN2 / MI7		8 digital inputs open-drain with internal
19	WP	In	L: Writeprotect for DataFlash		43	IN3 / MI6		pullup 2050k
20	TEST SBUF	IN Out	open-drain with internal pullup 2050k IN (Power-On) L: Testmode OUT L: data in sendbuffer		44	IN4 / MI5	In	alternativ up to 8 matrix keyboard input lines
21 22 23 24	N.C.		do not connect, reserved		45 46 47 48	IN5 / MI4		(reduces the digital input lines, see chapter external keyboard)

BAUD RATES

The baud rate is set by means of pins 6, 7 and 8 (baud 0 to 2). The data format is set permanently to 8 data bits, 1 stop bit, no parity.

RTS/CTS handshake lines are not required. The required control is taken over by the integrated software protocol (see pages 10 and 11).

	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								



Note:

The pins BAUD 0 to 2, ADR 0 to 2, DPOM, DPROT and TEST/SBUF have an internal pullup, which is why only the LO level (0=GND) is to be actively applied. These pins must be left open for a Hi level. For RS232 operation (without addressing) the pins ADR 0 to ADR 2 must be left open.

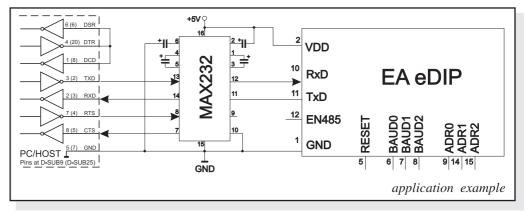
On pin 20 (SBUF) the display indicates with a low level that data is ready to be retrieved from the internal send buffer. The line can be connected to an interrupt input of the host system, for example.



APPLICATION EXAMPLE "REAL" RS-232 INTERFACE

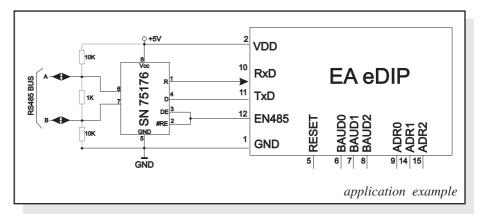
The eDIP fits for direct connection to a RS-232 interface with CMOS level (5V).

If you have an interface with ±12V level, an external levelshifter is needed



APPLICATION EXAMPLE: RS-485 INTERFACE

With an external converter (e.g. SN75176), the EA eDIP can be connected to a 2-wire RS-485 bus. Large distances of up to 1200 m can thus be implemented (remote display). Several EA eDIP displays can be operated on a single RS-485 bus by setting addresses.

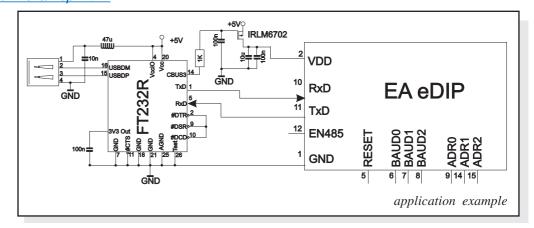


Addressing:

- Up to eight hardware addresses (0 to 7) can be set by means of Pins ADR0..ADR2
- The eDIP with the address 7 is selected and ready to receive after power-on.
- The eDIPS with the addresses 0 to 6 are deselcted after power-on
- Up to 246 further software addresses can be set by means of the '#KA adr' command in the power-on macro (set eDIP externally to address 0)

APPLICATION EXAMPLE: USB INTERFACE

With an external converter (e.g. FT232R) from FTDI the eDIP can be connected to an USB-Bus. Virtual-COM-Port drivers are available for different Systems on the FTDI Homepage: http://www.ftdichip.com/drivers/vcp.htm.





Page 6

SPI INTERFACE

If the display is wired as shown below, SPI mode is activated.

The data is then transferred via the serial, synchronous SPI interface.

The transfer parameter will be set via the pins DORD, CPOL and CPHA.

7	V	'n	1	0	٠
<u>_</u>	<u> </u>	$\underline{\mathbf{v}}$	ι	C	٠

The pins DORD, CPOL, CPHA, DPOM, DPROT and TEST/SBUF have an internal pullup, which is why only the LO level (0=GND) is to be actively applied. These pins must be left open for a Hilayel

			Pinout eDIPTFT70)-A	: SF	l mode				
Pin	Symbol	In/Out	Function		Pin	Symbol	In/Out	Function		
1	GND		Ground Potential for logic (0V)		25					
2	VDD		Power supply for logic (+5V)		26	N.C.		do not connect, reserved		
3	SND+		Speaker LS1 (Impedance 8 Ohm)		27	N.C.		do not connect, reserved		
4	SND-		Speaker LS2		28					
5	RESET	In	L: Reset		29	GND		Ground (=Pin 1)		
6	SS	In	Slave Select		30	VDD		Power supply (=Pin 2)		
7	MOSI	In	Serial In		31	AIN1	In	analogue input 0VDD		
8	MISO		Serial Out		32	AIN2	""	DC impedance 1MOhm		
9	CLK	In	Shift Clock		33	OUT1 / MO8		8 digital outputs		
10	DORD	In	Data Order (0=MSB first; 1=LSB first)		34	OUT2 / MO7		maximum current:		
11	SPIMOD	In	connect to GND for SPI interface		35	OUT3 / MO6		IOL = IOH = 10mA		
12	NC		do not connect		36	OUT4 / MO5				
13	DPOM	In	L: disable PowerOnMacro do not connect for normal operation		37	OUT5 / MO4	Out	alternativ up to 8 matrix keyboard output lines		
14	CPOL	In	Clock Polarity (0=LO 1=HI when idle)		38	OUT6 / MO3		(reduces the digital output		
15	CPHA	ln	Clock Phase sample 0=1st;1=2nd edge		39	OUT7 / MO2		lines, see chapter external keyboard)		
16	BUZZ	Out	Buzzer output		40	OUT8 / MO1		keyboard)		
17	DPROT	In	L: Disable Smallprotokoll do not connect for normal operation		41	IN1 / MI8		8 digital inputs		
18	DNC	Out	L: internal, do not connect		42	IN2 / MI7		open-drain with internal		
19	WP	In	L: Writeprotect for DataFlash		43	IN3 / MI6		pullup 2050k		
20	TEST SBUF	IN Out	open-drain with internal pullup 2050k IN (Power-On) L: Testmode OUT L: data in sendbuffer		44	IN4 / MI5		alternativ up to 8 matrix keyboard input lines (reduces the digital input		
21					45	IN5 / MI4		lines, see chapter external		
22	N.C.		do not connect recoved		46	IN6 / MI3		keyboard)		
23	IN.C.		do not connect, reserved		47	IN7 / MI2				
24				L	48	IN8 / MI1				

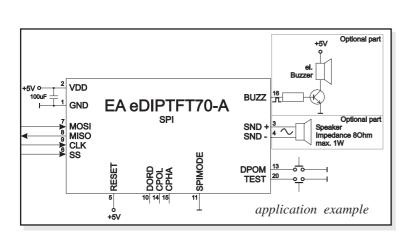
On pin 20 (SBUF) the display indicates with a low level that data is ready to be retrieved from the internal send buffer. The line can be connected to an interrupt input of the host system, for example.

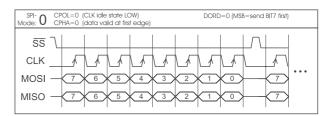
DATA TRANSFER SPI

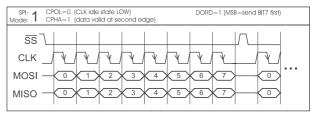
Write operation: a clock rate up to 200 kHz is allowed without any stop. Together with a pause of 100 μ s between every data byte a clock rate up to 3 MHz can be reached.

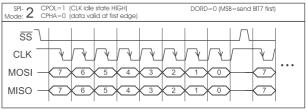
Read operation: to read data (e.g. the "ACK" byte) a dummy byte (e.g. 0xFF) need to be sent.

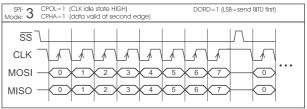
Note that the EA eDIP for internal operation does need a short time before providing the data; therefore a short pause of min. 6µs (no activity of CLK line) is needed for each byte.













I²C-BUS INTERFACE

If the display is wired as shown below, it can be operated directly on an I²C bus.

8 different base addresses and 8 slave addresses can be selected on the display.

Data transfer is possible at up to 100 kHz. However, if pauses of at least 100 µs are maintained between the individual bytes during transfer, a

byte can be transferred at up to 400

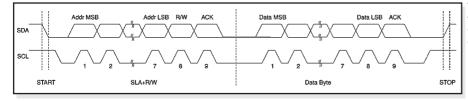
			Pinout eDIPTFT70-	A:	I2C	mode		
Pin	Symbol	In/Out	Function		Pin	Symbol	In/Out	Function
1	GND		Ground Potential for logic (0V)		25			
2	VDD		Power supply for logic (+5V)		26	N.C.		do not connect recorded
3	SND+		Speaker LS1 (Impedance 8 Ohm)		27	N.C.		do not connect, reserved
4	SND-		Speaker LS2		28			
5	RESET	In	L: Reset		29	GND	_	Ground (=Pin 1)
6	BA0	In	Basic Address 0		30	VDD		Power supply (=Pin 2)
7	BA1	In	Basic Address 1		31	AIN1	In	analogue input 0VDD DC impedance 1MOhm
8	SA0	In	Slave Address 0		32	AIN2		
9	SA1	In	Slave Address 1		33	OUT1 / MO8		8 digital outputs
10	SA2	In	Slave Address 2		34	OUT2 / MO7	_	maximum current:
11	BA2	In	Basic Address 2		35	OUT3 / MO6	_	IOL = IOH = 10mA
12	I2CMOD	In	connect to GND for I ² C interface		36	OUT4 / MO5		
13	DPOM		L: disable PowerOnMacro do not connect for normal operation		37	OUT5 / MO4	Out	alternativ up to 8 matrix keyboard output lines
14	SDA	Bidir.	Serial Data Line		38	OUT6 / MO3	_	(reduces the digital output
15	SCL	In	Serial Clock Line		39	OUT7 / MO2	_	lines, see chapter external keyboard)
16	BUZZ	Out	Buzzer output		40	OUT8 / MO1		keyboard)
17	DPROT		L: Disable Smallprotokoll do not connect for normal operation		41	IN1 / MI8	_	O divital in a sta
18	DNC	Out	L: internal, do not connect		42	IN2 / MI7		8 digital inputs open-drain with internal
19	WP	In	L: Writeprotect for DataFlash		43	IN3 / MI6		pullup 2050k
20	TEST SBUF	Out	open-drain with internal pullup 2050k IN (Power-On) L: Testmode OUT L: data in sendbuffer		44	IN4 / MI5	In	alternativ up to 8 matrix keyboard input lines
21					45 46	IN5 / MI4 IN6 / MI3	-	(reduces the digital input lines, see chapter external
22	N.C.		do not connect, reserved		47	IN7 / MI2	-	keyboard)
23 24					48	IN7 / MI2	-	

Note:

kHz.

The pins BA0..2, SA0..2, DPOM, DPROT and TEST/SBUF have an internal pullup, which is why only the LO level (0=GND) is to be actively applied. These pins must be left open for a Hi level.

On pin 20 (SBUF) the display indicates with a low level that data is ready to be retrieved from the internal send buffer. The line can be connected to an interrupt input of the host system, for example.



	I ² C - Address												
Pir	า 11,7	' ,6	Base				l²(ad	dre	ss			
BA2	BA1	BA0	address		D7	D6	D5	D4	D3	D2	D1	D0	
L	L	L	\$10		0	0	0	1					
L	L	Н	\$20		0	0	1	0					
L	Н	L	\$30		0	0	1	1	_	_			
L	Н	Н	\$40		0	1	0	0	S	S	S	R	
Η	١	L	\$70		0	1	1	1	A 2	A 1	A 0	W	
Н	L	Н	\$90		1	0	0	1	_	l	J		
Η	Н	L	\$B0		1	0	1	1					
Н	Η	Н	\$D0		1	1	0	1					

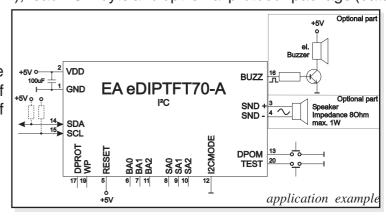
all pins open: Write \$DE Read \$DF

DATA TRANSFER I2C INTERFACE

principle I2C-bus transfer:

- I2C-Start
- Master-Transmit: EA eDIP-I²C-address (e.g. \$DE), send smallprotocol package (data)
- I²C-Stop
- I2C-Start
- Master-Read: EA eDIP-I²C-Address (e.g. \$DF), read ACK-byte and opt. smallprotocoll package (data)
- I2C-Stop

Read operation: for internal operation the EA eDIP does need a short time before providing the data; therefore a short pause of min. 6µs is needed for each byte (no activity of SCL line).





Page 8

ANALOGUE INPUT AIN1 AND AIN2 (PIN 31+32)

For analogue measurement 2 inputs with a range of 0..+5V are available. Each input is grounded (GND) and DC impedance is $1M\Omega$. Please make sure that only positive voltages will be supplied there. Internal resolution is 10 Bit, equal to a 3-digit DVM modul. Linearity (after adjustment) is around 0.5%.

Adjustment

Analogue inputs are not calibrated when shipped out. A procedure for adjustment may be like that:

- 1.) Put a well known voltage within a range of 2V-VDD to analogue input (example: 3,0V, AIN1)
- 2.) Run command for calibration (see page 16). Example: "ESC V @ 1 3000".

Each input query can be done via serial interface or directly shown on display (as digits or bargraph in various colors and sizes).

Best way for direct visualisation are Process-macros or one of Analogue-macros (e.g. starting at every voltage change, or above/below a limit).

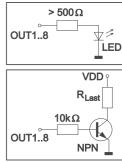
Both input lines are scaleable from 0 to ± 9999.9 . Scaling will be done via definition at 2 voltages "value1=string1;value2=string2" (see table on page 16).

DIGITAL INPUT AND OUTPUT

The EA eDIP is featured with 8 digital input and 8 digital output lines (CMOS level, grounded).

8 outputs (Pin 33-40)

Each line can be controlled individually using the "ESC Y W" command. A maximum current of 10mA can be switched per line. This give the opportunity to drive a low power LED in direct way. To source higher current please use an external transistor.



8 inputs (Pin 41-48)

Each input provides an internal 20..50 k Ω pull-up resistor, so it is possible to connect a key or switch directly between input and GND. The inputs can be queried and evaluated directly via the serial interface ("ESC Y R").



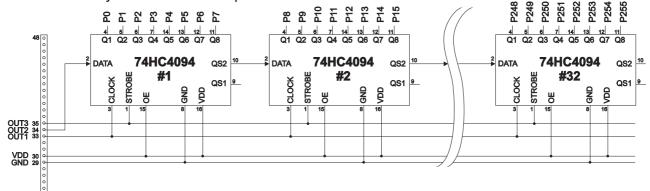
In addition to that every port change may start an individual port - or bit-macro (see p. 30).

The command "ESC Y A 1" activates automatic port query. Every alteration of inputs firstly calls bit macros and afterwards port macros. If there is no definied macro, the new status is transferred into the send buffer (refer to p. 21).

Note: The logic circuitry is designed for slow operations; in other words, more than 3 changes per second cannot be easily executed.

EXTENDED OUTPUTS

It is possible to connect 1 to 32 chips like 74HC4094 to the eDIP (OUT1...OUT3), this is why it is attainable to have 8 to 256 additional outputs. The command "ESC Y E n1 n2 n3" (see p. 17) provides a comfortable way to control the outputs.





EXTERNAL KEYBOARD

A keyboard (anything from individual keys to a 8x8 matrix keyboard) can be connected to the I/O- ports. The command 'ESC Y M n1 n2 n3' defines the count of input lines (n1=1..8) and output lines (n2=1..8). n3 set debounce function with 10ms steps (n3=0..15). Please note that count of digital input and output lines will be reduced while connecting an external keyboard at the same port.

Each key is connected with 1 output and 1 input. All inputs are terminated with a 20..50k Ω pull-up resistor. For double-keytroke function decoupling of outputs is necessary. For that please use schottky diodes (e.g. BAT 46).

Transmitting the keystrokes

At each keystroke, the associated key number (1..64) is transmitted or if a corresponding Matrix-Macro is defined, Matrix-Macro will be started. The release of the key is not transmitted. If the release of the key is to be transmitted as well, this can be done by defining Matrix-Macro no. 0. (see page 21: Responses of EA eDIP)

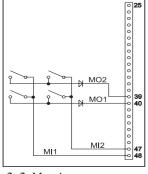
Calculating of key numbers:

Key_number = (output-1) * count_of_inputs + input (output = MOx, input = MIx).

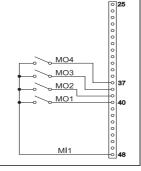
Examples

- 2x2 matrix: Command 'ESC Y M 2 2 ..' defines the 2x2 matrix. Keypad will need input lines MI1, MI2 and output lines MO1, MO2. Output lines are decoupled by diodes; this is for double keystrokes necessary. 6 input and 6 output lines remain free for other requirements.
- 1x4 matrix: Command 'ESC Y M 1 4 ..' defines the 1x4 matrix. Keypad will need output lines MO1..MO4 and a single input line MI1. With that connection 7 input and 4 output lines remain free for other requirements.
- 4x0 matrix: Using one single output only (physically 4x1 Matrix), all keys can switch to GND. So no output line is necessary and comman 'ESC Y M 40..' defines 4 input lines onyl. With that connection 4 input and 8 output lines remain free for other requirements.
- 4x4 matrix: Command 'ESC Y M 4 4 ..' defines the 4x4 matrix. Keypad will need input lines MI1..MI4 and output lines MO1..MO4. Output lines are decoupled by diodes; this is for double keystrokes necessary. 4 input and 4 output lines remain free for other requirements.

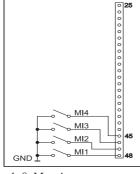
.



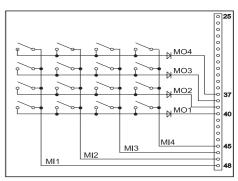
2x2 Matrix



1x4 Matrix



4x0 Matrix



4x4 Matrix



START

WRITE

<DC1>

len data...

bcc

wait >6us (SPI only)

READ

(1 byte)

BYTE

END

ACK (\$06)

NAK (\$15)

EAeDIPTFT70-A

Page 10

DATA TRANSFER PROTOCOL (SMALL PROTOCOL)

The protocol has an identical structure for all 3 interface types: RS-232, SPI and I²C. Each data transfer is embedded in a fixed frame with a checksum (protocol package). The EA eDIPTFT32-A acknowledges this package with the character <ACK> (=\$06) on successful receipt or <NAK> (=\$15) in the event of an incorrect checksum or receive buffer overflow. In the case of <NAK>,

the entire package is rejected and must be sent again.

Receiving the <ACK> byte means only that the protocol package is ok, there is no syntax check for the command.

Note: it is neccessary to read the <ACK> byte in any case. If the host computer does not receive an acknowledgment, at least one byte is lost. In this case, the set timeout has to elapse before the package is sent again. The raw data volume per package is limited to 255 bytes (len <= 255). Commands longer than 255 bytes (e.g. Load image ESC UL...) must be divided up between a number of packages. All data in the packages are compiled again after being correctly received by the EA eDIP.

DEACTIVATING THE SMALL PROTOCOL

For tests the protocol can be switched off with an L level at pin 17 = DPROT. In normal operation, however, you are urgently advised to activate the protocol. If you do not, any overflow of the receive buffer will not be detected.

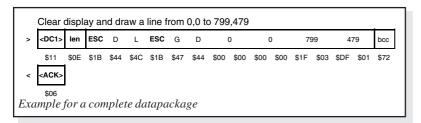
BUILDING THE SMALL PROTOCOL PACKAGES

Command/data to the display



< DC1 > = 17(dez.) = \$11< ACK > = 6(dez.) = \$06

len = count of user data (without <DCI>, without checksum bcc) bcc = 1 byte = sum of all bytes incl. <DCI> and len, modulo 256



The user data is transferred framed by <DC1>, the number of bytes (len) and the checksum (bcc). The display responds with <ACK>.

wait

t = timeout

no answer

≠ NAK

≠ ACK

Request for content of send buffer



< DC2 > = 18(dez.) = \$12 I = I(dez.) = \$01 S = 83(dez.) = \$53 < ACK > = 6(dez.) = \$06

len = count of user data (without <DC1>, without checksum bcc) bcc = 1 byte = sum of all bytes incl. <DC1> and len, modulo 256 The command sequence <DC2>, 1, S, bcc empties the display's send buffer. The display replies with the acknowledgement <ACK> and the begins to send all the collected data such as touch keystrokes.



Request for buffer information

>	<dc2></dc2>	1	I	bcc	
<	<ack></ack>				•
	D.00		send	buffer	receive but

$$< DC2 > = 18(dez.) = $12$$
 $1 = 1(dez.) = 01 $I = 73(dez.) = 49 $< ACK > = 6(dez.) = 06

bytes free

bytes ready

send buffer bytes ready = count of bytes stored in send buffer receive buffer bytes free = count of bytes for free receive buffer bcc = 1 byte = sum of all bytes incl. DC2>, modulo 256

This command queries whether user data is ready to be picked up an how full the display's receive buffer is.

Protocol settings

<DC2>

>	<dc2></dc2>	3	D	packet size for send buffer	timeout	bcc
<	<ack></ack>					

<DC2> = 18(dec.) = \$12 3 = 3(dez.) = \$03 D = 68(dez.) = \$44 packet size for send buffer = 1..128 (standard: 128) timeout = 1..255 in 1/100 seconds (standard: 200 = 2 seconds) bcc = 1 byte = sum of all bytes incl. <DC2>, modulo 256 <ACK> = 6(dec.) = \$06

This is how the maximum package size that can be sent by the display can be limited. The default setting is a package size with up to 128 bytes of user data. The timeout can be set in increments of 1/100 seconds. The timeout is activated when individual bytes get lost. The entire package then has to be sent again.

Request for protocol settings

>	<dc2></dc2>	1	Р	bcc				command ings.	is	used	to	query	protocol
<	<ack></ack>						Setti	irigs.					
<	<dc2></dc2>	3	ma packe	-	akt. send packet size	akt. timeout	bcc						

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

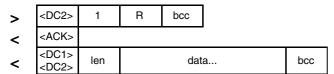
< ACK > = 6(dez.) = \$06

max. packet size = count of maximum user data for 1 package (eDIPTFT70-A = 255)

akt. send packet size = current package size for send akt. timeout = current timeout in 1/100 seconds

bcc = 1 byte = sum of all bytes incl. <DC2>, modulo 256

Repeat the last package

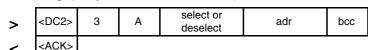


$$< DC2> = 18(dez.) = \$12$$
 $1 = 1(dez.) = \$01$ $R = 82(dez.) = \$52$ $< ACK> = 6(dez.) = \$06$ $< DC1> = 17(dez.) = \$11$

len = count of user data in byte (without ckecksum, without < DC1 > or < DC2 >)bcc = 1 byte = sum of all bytes incl. < DC2 > and len, modulo 256

If the most recently requested package contains an incorrect checksum, the entire package can be requested again. The reply can then be the contents of the send buffer (<DC1>) or the buffer/protocol information (<DC2>).

Adressing (only for RS232/RS485)



 $\langle DC2 \rangle = 18(dez.) = \12 3 = 3(dez.) = \$03

A = 65(dez.) = \$41

 $select\ or\ deselect:\ 'S' = \$53\ or\ 'D' = \$44$

adr = 0..255

bcc = 1 byte = sum of all bytes incl. <DC2> and adr, modulo 256

 $<\!\!ACK\!\!> = 6(dec.) = \06

This command can be used to select or deselect the eDIP with the address adr.



Page 12

TERMINAL MODE

When you switch the unit on, the cursor flashes in the first line, indicating that the display is ready for operation. All the incoming characters are displayed in ASCII format on the terminal (exception: CR,LF,FF,ESC,'#'). The prerequisite for this is a working protocol frame or a deactivated protocol (see pages 10 and 11).

Line breaks are automatic or can be executed by means of the 'LF' character. If the last line is full, the contents of the terminal scroll upward. The 'FF' character (page feed) deletes the terminal. The character '#' is used as an escape character and thus cannot be displayed directly on the terminal. If the character '#' is to be output on the terminal, it must be transmitted twice: '##'. The size of the terminal-window can be set by command 'ESC TW'.

Attention: The terminal has its own level for displaying and is thus entirely independent of the graphic

\$8 \$9 \$A \$B \$C \$D \$E \$F (8) (9) (10) (11) (12) (13) (14) (15) \$4 \$5 \$6 \$7 (4) (5) (6) (7) \$1 \$2 \$3 (1) (2) (3) Φ **V** J. \$00 (dez: 0) 슌 Φ ♦ F Ţ 1 + H 0 8 ŋ 1 3 ч 5 8 8 9 Θ E_S 1 \$10 (dez: 16) ш \$ % & ı \$20 (dez: 32) Ļ # () + 7 5 2 3 4 6 7 8 9 ? 0 1 ; < C F Ι \$40 (dez: 64) 0 A В D Ε G Н J Κ 0 Q R S Τ U Ų М X Y Z E Λ \$50 (dez: 80) f Ь ď h i j k e g \$60 (dez: 96) { t \$70 (dez: 112) Γ s u ٧ М X y z p q é â ä à å ê ë \$80 (dez: 128) C ü Ç è É Æ ö \$90 (dez: 144) æ ô ö ò û ù ÿ Ü ¢ ó ñ Ñ \$A0 (dez: 160) á í ú ₫ 0 ż 1 \$B0 (dez: 176) ١ ‡ \parallel # П 1 Т ŀ + ŀ Ŀ Ήľ _ ŧ \$C0 (dez: 192) т lī 丽 Ш F # J ŧ F \$D0 (dez: 208) Ŧ π П Γ $\beta \mid \Gamma$ Σ τ δ E0 (dez: 224) α $\pi ^{\prime }$ σ Щ δ Θ Ω Φ Ε ۰ \$F0 (dez: 240) ± ≥ չ | Ր | J | ÷ z n 2 3

Terminal-Font 2: 8x16

outputs. If the graphics screen is deleted with 'ESC DL', for example, that does not affect the contents of the terminal window.

USING THE SERIAL INTERFACE

The operating unit can be programmed by means of various integrated commands. Each command begins with ESCAPE followed by one or two command letters and then parameters.

There are two ways to transmit commands:

1. ASCII mode

- The ESC character corresponds to the character '#' (hex: \$23, dec: 35).
- The command letters come directly after the '#' character.
- The parameters are transmitted as plain text (several ASCII characters) followed by a separating character (such as a comma ',') also after the last parameter e.g.: **#GD0,0,799,479**,
- Strings (text) are written directly without quotation marks and concluded with CR (hex: \$0D) or LF (hex: \$0A).

2. Binär mode

- The escape character corresponds to the character ESC (hex: \$1B, dec: 27).
- The command letters are transmitted directly.
- The coordinates xx and yy are transmitted as 16-bit binary values (first the LOW byte and then the HIGH byte).
- All the other parameters are transmitted as 8-bit binary values (1 byte).
- Strings (text) are concluded with CR (hex: \$0D) or LF (hex: \$0A) or NUL (hex: \$00).

No separating characters, such as spaces or commas, may be used in binary mode.

The commands require **no final byte**, such as a carriage return (apart from the string \$00).



ALL COMMANDS AT A GLANCE

The built-in intelligence allows an easy creation of your individual screen content. Below mentioned commands can be used either directly via the serial interface (see page 12) or together with the selfdefinable macro (see page 28).

	•		•			الط	DTE	T70	0-A: Terminal commands	after	
Command	Cod	les			EA (ווטט	<u> </u>		Remarks	reset	
Set terminal color	ESC	F	Т	fg	bg				Preset color for terminal mode: fg= foreground color; bg= background color	8,1	
			The terminal output is executed with font n1: 1=8x8; 2=8x16 only within the window from column C and line L (=upper-left corner) with a width of W and a height of H (specifications in characters). W=1100 and Height H=60/30 (in characters)	8x16 1,1 60,17							
Form feed FF (dec:12)	^L								The contents of the screen are deleted and the cursor is placed at pos. (1,1)		
Carriage return CR (13)	^M								Cursor to the beginning of the line on the extreme left		
Line feed LF (dec:10)	^J								Cursor 1 line lower, if cursor in last line then scroll		
Position cursor			Р	С	L				C=column; L=line; origin upper-left corner (1,1)	1,1	
Cursor on/off			С	n1					n1=0: Cursor is invisible; n1=1: Cursor flashes;	1	
Save cursor position	ESC	۱.	s						The current cursor position is saved	İ	
Restore cursor position	LSC	'	R						The last saved cursor position is restored		
Terminal off			Α						Terminal display is switched off; outputs are rejected		
Terminal on			Е						Terminal display is switched on;	on	
Output version			٧						The version no. is output in the terminal e.g. "EA eDIPTFT70-A V1.0 Rev.A"		
Output projectname	ESC	Т	J						The macrofile-projectname is output in the terminal e.g. "init / delivery state"		
Output interface			Q						The used interface is output in the terminal e.g "RS232,115200 baud,ADR: \$07"		
Output informationen	ESC	Т	I						The terminal is initialisized and cleared; the software version, hardware revision, macrofile-projectname and CRC-checksum are output in the terminal		

					EΑ	<u>eD</u>	IPT	FT7	<u>′0-A: G</u>	<u>apt</u>	nic commands	after
Command	Cod	les							Remark	s		reset
					Dis	splay	com	mar	nds (effec	t on t	the entire display)	
Set display color	ESC	F	D	fg	bg				Defines co	lor 1.	.32 for display and areas: fg=foreground color; bg=background color	8,1
Delete display	ESC		L						Delete dis	olay c	ontents (all pixels to background color)	
Fill display	ESC	D	S						Fill display	conte	ents (all pixels to foreground color)	
Fill display with color	ESC	ם ו	F	n1					Fill comple	te dis	splay content with color n1=132	
Invert display	ESC		ı						Invert disp	lay co	ontent	
							C	omi	nands for	out	outting strings	
Set text color	ESC	F	Z	fg	bg				Color 13	2 (0=t	ransparent) for string and character: fg=text color; bg=background color	8,0
Set font			F	n1					Set font w	th the	number n1	3
Font zoom factor			Z	n1	n2				n1 = X-zoo	m fac	ctor (1x to 8x); $n2 = Y$ -zoom factor (1x to 8x)	1,1
Additional width/height	ESC	Z	Υ	n1	n2				n1=015:	additio	onal width left/right; n2=015: additional height top/bottom	0, 0
Spacewidth			J	n1					n1=0: use	space	ewidth from font; n1=1: same witdh as a number; n1>=2 width in dot	0
Text angle			W	n1					Text outpu	t ang	le: n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°	0
Output string			L						A string (.) is o	utput to xx1,yy1	
L: left justified	ESC	z	С	xx1	V0/4	te	xt	NUL			UL' (\$00), 'LF' (\$0A) or 'CR' (\$0D)	
C: centered	ESC		-	XXI	yy1			NUL			e separated by the character ' ' (\$7C, pipe)	
R: right justified			R						the charac	ter '\'	(\$5C, backslash) cancles the special function of ' ' and '\'	
0.44											Output a string () inside area from xx1,yy1 to xx2,yy2 at position n1=19;	
Output string	ESC	۱.	_						text	l	the area will be filled with background color;	
in an area	ESC	Z	В	xx1	yy1	XX2	yy2	n1		NUL	n1=1: Top Left; n1=2: Top Center; n1=3: Top Right n1=4: Middle Left; n1=5: Middle Center; n1=6: Middle Right	
(since V1.2)											n1=7: Bottom Left; n1=8: Bottom Center; n1=9: Bottom Right	
String for terminal	ESC	Z	Т			text			Command	for o	utputting a string from a macro to the terminal	
		<u> </u>	1	1				Dra			es and points	1
Set color for lines	ESC	F	G	fg	bg			Diu			transparent): fg = color for line; bg = pattern background	8.1
Point size/line thickness			z	n1	n2						e (1 to 15); n2 = Y-point size (1 to 15);	-,
Pattern			М	n1							/point pattern no. n1=1255; 0=do not use pattern	
Draw point			Р	xx1	yy1						pordinates xx1, yy1	
Draw straight line			D	xx1	yy1	xx2	yy2	ĺ	Draw strai	ght lin	ne from xx1,yy1 to xx2,yy2	
Draw rectangle			R	xx1	yy1	xx2	•			_	ht lines as a rectangle from xx1,yy1 to xx2,yy2	
Set start point	ESC	G	S	xx1	yy1		<i>J J</i> =				point at coordinates xx1, yy1 for commands 'GW', 'GX' and 'GY'	
Continue straight line			w	xx1	yy1						line from last end point to xx1, yy1	1,1
					1	.					n fix x-steps (xs=1127 or 129255 for neg. steps) and variable amount (ya=1	
Draw X-Graph			Х	xs	ya	yy1			of y-values			Ĺ
Draw Y-Graph			Υ	ys	xa	xx1					n fix y-steps (ys=1127 or 129255 for neg. steps) and variable amount (xa=1	.255)
Втан т агарт		l		,,	7.00	70.		_	of x-values			
				I	1			Cha			angular areas	1
Delete area		l	L	xx1	yy1	xx2	уу2				from xx1,yy1 to xx2,yy2 (fill with background color)	1
Fill area		l _	S	xx1	yy1	xx2	yy2	<u> </u>			n xx1,yy1 to xx2,yy2 (fill with foreground color)	<u> </u>
Fill area with color	ESC	R	F	xx1	yy1	xx2	yy2	n1			n xx1,yy1 to xx2,yy2 with color n1=132	ļ
Invert area	_		ı	xx1	yy1	xx2	уу2				om xx1,yy1 to xx2,yy2	ļ
Copy area	_	_	С	xx1	yy1	xx2	yy2	ххЗ			rea from xx1,yy1 to xx2,yy2 to new position xx3,yy3	
Patterncolor		F	М	fg	bg	<u> </u>	L.,				ransp.) for monochrome pattern: fg=foreground; bg=background color	8,1
Area with fill pattern	ESC	R	М	xx1	yy1	xx2	yy2	n1			om xx1,yy1 to xx2,yy2 with pattern n1	<u> </u>
Draw box			0	xx1	yy1	xx2	yy2	n1			e xx1,yy1 to xx2,yy2 and fill with pattern n1	
Set color for border		F	R	c1	c2	c3					rder segments: c1=frame outside; c2=frame inside; c3=filling	8,1,1
Set border type	ESC	R	Е	n1	n2	<u> </u>					n1=1255; border angle: n2=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°	1, 0
Draw border box			R	xx1	vv1	xx2	yy2	1	Draw a bo	rder h	pox from xx1,yy1 to xx2,yy2	1



			Е	Ае	DIP.	ΓFT	70-/	4: B	itmap / Animation commands	after
Command	Cod	les							Remarks	reset
								Е	Bitmap image commands	
Set bitmap colors	ESC	F	U	fg	bg				painting color for monchrome bitmaps fg=foreground color; bg=background color	1,8
Image zoom factor			Z	n1	n2				n1 = X-zoom factor (1x to 8x); n2 = Y-zoom factor (1x to 8x)	1,1
lmage angle			W	n1					output angle of the image: n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°	0
Mirror Image		١	Х	n1					n1=0: normal display; n1=1: the image is mirrored horizontally	0
Transparency for color bitmaps	ESC	U	т	n1					n1=0: no transparency; show picture with all colors rectangular n1=1: color of the first dot at top left side will be defined as transparent (like a mask) n1=2: if defined - use transparent color from bitmap-file (.GIF .TGA .PNG .G16) n1=3: replace transparent color from bitmap-file with actually background color	2
Load internal image	ESC	U	I	xx1	yy1	nr			Load internal image with the no (0 to 255) from the data flash memory to xx1,yy1	
Load image	ESC	١	L	xx1	yy1	G.	l6 dat	a	Load an image to xx1,yy1; see image structure (G16 format) for image data	
RLE compression	ESC	U	R						the next hardcopy ('ESC U H xx1,yy1,xx2,yy2) will be send with RLE compression	
Send hardcopy	ESC	U	Н	xx1	yy1	xx2	yy2		After this command, the image extract is sent (to sendbuffer) in G16 format	
								An	imation image commands	
Set animation colors	ESC	F	W	fg	bg				color for monchrome animation images fg=foreground color; bg=background color	1,8
Animation zoom factor			Z	n1	n2				n1 = X-zoom factor (1x to 8x); n2 = Y-zoom factor (1x to 8x)	1,1
Animation angle			W	n1					output angle of the animation image n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°	0
Mirror animation			Х	n1					n1=0: normal display; n1=1: the animation image is mirrored horizontally	0
Transparency for color animation	ESC	W	т	n1					n1=0: no transparency; show animation with all colors rectangular n1=1: color of the first dot at top left side will be defined as transparent (like a mask) n1=2: if defined - use transparent color from animation-file (.GIF .G16) n1=3: replace transparent color from animation-file with actually background color	2
Load single image	ESC	W	ı	xx1	yy1	n1	n2		Load from animation n1=0255 the single image n2 to xx1,yy1	
Define animationprocess	ESC	w	D	no	xx1	yy1	n2	type	Define an animationprocess no=14 at position xx1,yy1 (=left top edge) with animation image n2=0255. time type: 1=run once; 2=cyclically; 3=pingpong; 4=once backwards; 5=cyclic backwards 6=pingpong backwards; 7=manually (use command ESC W N P F M) time: 0=stop; 1254=time in in 1/10 sec; 255=use time from animation-file	
Change animation type			Υ	no	type				Assign a new type=17 to animationprocess no=14	
Change animation time			С	no	time				Assign a new time=0255 to animationprocess no=14	
Next animation image			N	no					Show the next image from animationprocess no=14	
Previous animation image	ESC	w	Р	no					Show the previous image from animationprocess no=14	
Show animation image			F	no	n2				Show image n2 from animationprocess no=14	
Run to animation image			M	no	n2				Run animationprocess no=14 from actually image to image n2	
Stop animationprocess			L	no					Stop animationprocess no=14 and clear last image with actually background color	

					ΕA	\ eD	IPT	FT7	0: C	Clipboard commands	after
Command	Cod	les								narks	reset
						CI	ipbo	ard c	omn	ands (Buffer for display area)	
Save display contents		С	В						The	entire content of the display is copied to the clipboard as an image area	
Save area	ESC	L	S	xx1	yy1	xx2	yy2		The i	mage area from xx1, yy1 to xx2, yy2 is copied to the clipboard	
Restore area	ESC	_	R						The i	mage area on the clipboard is copied back to the display	omnt.
Copy area	Ī	С	K	xx1	yy1				The i	mage area in the clipboard is copied back to xx1, yy1 of the display	empty
								L	oad I	pitmap to clipboard	
Load image to clipboard			L	xx1	yy1	G1	6 data	a	Load	the image data (G16-format) into clipboard to xx1, yy1	
Internal image to clipboard			1	xx1	yy1	n1			Draw	rinternal image from data-flash with n1=0255 into clipboard to xx1, yy1	
Animation sub image to clipboard	ESC	С	F	xx1	yy1	n1	n2			rinternal animation sub image from data-flash with number n1=0255, n2=framenumber into pard to xx1, yy1	0
Clipboardimage autoupdate			U	n1						: Auto update OFF; n1=1: Auto update ON; after loading image into clipboard ('CL', 'CI, 'CF ontent is copied to the screen	⁾ 1
								Exte	nded	clipboard layer usage	
Copy to clipboard layer			Р	xx1	yy1	xx2	yy2	ххЗ	уу3	Area from xx1,yy1 to xx2,yy2 on the screen is copied into the clipboard layer to xx3,yy3	
Copy from clipboard layer			С	xx1	yy1	xx2	yy2	ххЗ	ууЗ	Clipboard area from xx1,yy1 to xx2,yy2 is copied into the screen to xx3,yy3	
Copy area on clipboard layer	ESC	С	A	xx1	yy1	xx2	уу2	ххЗ	ууЗ	The area from x1,y1 to x2,y2 on the clipboard layer is copied to new upper left corner x3,y3	3
Clipboard transparent color			Т	n1		•				n1=0: no transparency; 132: transparent color for clipboard, only for 'ESC CC'	0
Animation through clipboard			w	n1	n2					n1=14: animation nr; n2=0: draw directly to screen, n2=1 first draw to clipboard layer and then copy to screen	1-4 0



				E	ΕΑ ε	DIP	TFT	70-	A: I	nstı	rument commands	after
Command	Cod	es							Ren	nark	S	reset
									Defin	e, us	e instruments	
Definine instrument	ESC	I	Р	n1	xx1	yy1	n2	n3	aw	ew	Define instrument n1=14 and xx1,yy1 (=left top edge); Use intrument image n2=0255 Output angle n3=0: 0°; n3=1: 90°; n3=2: 180°; n3=3: 270°; aw, ew (0254) are start and endvalue (0% and 100%).	nothing definied
Update instrument			Α	n1	val				Upda	ate ins	trument with new value and redraw	
Redraw instrument			N	n1					Redr	aw en	tire instument n1=14	
Send instrument value	ESC	1	S	n1					Send	actua	al instrument value n1=14 to send buffer	
Delete instrument			D	n1	n2				the to	ouch a	ion of the instrumets gets invalid. If the instrument was adjustable by touch, trea will be deleted, too. ument stays visible; n2=1: Instrument is deleted completely	
						U	ser v	alues	s - fo	rmatt	ted string output	
User value color	ESC	F	I	vf	hf				Set c	olor 1	32 for instrument user value fg=foreground; bg=background color	8,1
User value font			F	n1					Set for	ont nr	for instrument user value	5
User value zoom			Z	n1	n2				Set z	oom f	actor for instrument user value: n1=X-Zoom 1x8x; n2=Y-Zoom 1x8x	1,1
User value additional hight/width	ESC	I	Υ	n1	n2						additional width left/right; additional height top/bottom for instrument user value;	0, 0
User value angle			W	n1					Set v	vriting	angle for instrument user value: n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°	0°
User values / scaling	ESC	ı	x	n1	xx1	yy1	For mat Str ing	NUL	Form Assig + ded e.g. d	nat Str gn two cimal p display	r value for instrument no=14. Output is right justified to x1,y1 ing: "iv1=uservalue1;iv2=uservalue2" instrument values (iv1,iv2 =0254) to user defined values max. range: 4 1/2 digits 199 opint ('.' or ',') + sign y "-123.4" for iv1=0 and "567.8" for iv2=100 ing: "0=-123.4;100=567.8"	999

					EΑ	eDI	PTF	T70)-A:	Ва	argra	aph commands	after
Command	Cod	es							Ren				reset
									Bar	gra	ph co	mmands	
Set color for bargraph	ESC	F	В	fg	bg	fc			Colo	rs: f	g = fore	eground; bg = background; fc = color for frame	8,1,8
Bargraph pattern			M	n1					Patte	ern fo	or barg	raph n1=1255; n1=0 no pattern/solid (valid for type=03)	0
Bargraph border	ESC	В	Е	n1					Bord	er fo	r barg	raph n1=1255 (valid for type=47)	1
Bargraph linewidth			В	n1					Line	width	for ba	argraph n1=1255; n1=0 automatic (valid for type=2,3,6,7)	0
Define bargraph	ESC	В	R L O U	no	xx1	уу1	xx2	уу2	sv	ev	type	Define bargraph no=120 to L(eft), R(ight), O(up), U(down) xx1,yy1,xx2,yy2 rectangle enclosing the bar graph. sv, ev are the values for 0% and 100%. type: 0=pattern bar; 1=pattern bar in rectangle; type: 2=pattern line; 3=pattern line in rectangle; type: 4=border bar; 5=border bar in rectangle; type: 6=border line; 7=border line in rectangle;	no bar defined
Update bargraph			Α	no	val				Set a	and o	lraw th	e bargraph with the number no=120 to the new value val	
Draw bargraph			N	no		-			Entir	ely r	edraw	the bargraph with the number no=120	
Send bargraph value	ESC	В	s	no					Send	the	currer	nt value of bargraph number no=120 to sendbuffer	
		_										f the bar graph with the number no=120 becomes invalid.	
Delete bargraph			D	no	n2							was defined as input with touch, this touch field will also be deleted.	
			<u> </u>	<u> </u>	<u> </u>		11					n remains visible; n2=1: Bar graph is deleted	
User value color	ESC	-	· ·	٠.	T	1	Use	r vaii				ext output	
	ESC	F	X	fg	bg				_			rgraph user value; fg=foreground, bg=background color	8,1 5
User value font				n1		1			1			pargraph user value	
User value zoom		В	Z	n1	n2	1			1			r for bargraph user value; n1=X-Zoom 1x8x; n2=Y-Zoom 1x8x	1,1
User value additional width/height	ESC	В	Y	n1	n2							onal width left/right; onal height top/bottom for bargraph user value;	0, 0
User value angle			W	n1					Set v	vritir	g angl	e for bargraph user value; n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°;	0°
User values / scaling	ESC	В	x	no	xx1	yy1	For mat Str ing	NUL	Form Assiq max. e.g. q	at S gn tv ran displ	tring: ' o bar ge: 4 1 ay "-12	ue for bargraph no=120. Output is always right justified to xx1,yy1; bv1=uservalue1;bv2=uservalue2". 'NUL' (\$00) = termination; values (bv1,bv2 =0254) to user defined values /2 digits 1999 + decimal point ('.' oder ',') + sign 23.4" for bar value bv1=0 and "567.8" for bar value bv2=100 0=-123.4;100=567.8"	



					EΑ	eD	IPT	FT	70-A: Macro commands	after
Command	Cod	les							Remarks	reset
	•								Macro commands	•
Run macro			N	no					Call the (normal) macro with the number no (max. 7 levels)	
Run touch macros	1		Т	no					Call the touch macro with the number no (max. 7 levels)	
Run menu macro	1		М						Call the menu macro with the number no (max. 7 levels)	
Run port macro	ESC	м	Р	no					Call the port macro with the number no (max. 7 levels)	
Run bit macro	ESC	IVI	В	no					Call the bit macro with the number no (max. 7 levels)	
Run matrix macro	1		Х	no					Call the matrix macro with the number (max. 7 levels)	
Run process macro	1		С	no					Call the process macro with the number (max. 7 levels)	
Run analogue macro	1		٧	no					Call the analogue macro with the number no (max. 7 levels)	
Disable macros	ESC	м	L	type	n1	n2			Macros of the type'N', T' , P' , B' , X' , C' , 'M' or 'V' (type 'A' = all macro types) are disabled from number n1 to n2; i.e. no longer run when called.	the
Enable macros	LSC	IVI	U	type	n1	n2			Macros of the type 'N','T','P','B','X','C', 'M' or 'V' (type 'A' = all macro types) are enabled fror number n1 to n2; i.e. run again when called.	h
Select macro/image page	ESC	М	к	n1					A page is selected for macros and images n1=0 to 15. if a macro/image is not defined in the current page 1 to 15, this macro/image is taken from page 0 (e.g. to switch languages or for horizontal/vertical installation).	
Save macro/image page	1		W						the current macro/image page is saved (when used in process macros)	
Restore macro/imagepage			R						the last saved macro/image page is restored	
								Αι	utomatic (normal-) macro	
Macro with delay			G	n1	n2				Call the (normal) macro with the number n1 in n2/10s. Execution is stopped by commands receipt or touch macros).	e.g.
Autom. macros once only	ESC	м	E	n1	n2	n3			Automatically run macros n1 to n2 once only; n3=pause in 1/10s. Execution is stopped by commands (e.g. receipt or touch macros).	
Autom. macros cyclical	ESC	IVI	Α	n1	n2	n3			Automatically run macros n1 to n2 cyclically; n3=pause in 1/10s. Execution is stopped by commands (e.g. receipt or touch macros).	
Autom. macros ping pong			J	n1	n2	n3			Automatically run macros n1 to n2 to n1 (ping pong); n3=pause in 1/10s. Execution is stopp for example, by receipt or touch macros.	ed,
									Macro processes	
Define macro process			D	no	type	n3	n4		A macro process with the number no (1 to 4) is defined (1=highest priority). The process macros n3 to n4 are run successively every zs/10s. type: 1=once only; 2=cyclical; 3=ping pong n3 to n4 to n3	
Macro process interval	ESC	М	z	no	zs				a new time zs in 1/10s is assigned to the macro process with the number no (1 to 4). if the time zs=0, execution is stopped.	
Stop macro processes			s	n1					All macro processes and animations are stopped with n1=0 and restarted with n1=1 in orde for example, to execute settings and outputs via the interface undisturbed	r, 1

		E	A el	DIP.	ΓFT	70- <i>P</i>	۱: A	nale	ogue input AIN1, AIN2 commands	after
Command	Cod	les							Remarks	reset
								Com	nmands for analogue inputs	
Calibration	ESC	v	@	ch	xx1				Calibration procedure is as follows: 1.) Apply defined voltage (35V) to AIN1 (channel1) or AIN2 (channel2) 2.) Run this command with channel information ch=12 and xx1=voltage value [mV] (16-Bit) e.g. 4.0V on AIN1; Command: '#V@1,4000;'	not calib rated
Enable/disable AIN scan			Α	n1					n1=0 disables input scan for AIN1 and AIN2; n1=1 enable input scan	0
Send analog value			D	ch					Voltage in [mV] will be sent (to sendbuffer) for channel ch=12	
Limit for analog macro	ESC	V	К	ch	n1	n2	n3		Sets two limits for channel ch=12. n1=lower limit [mV/20]; n2=upper limit [mV/20]; n3=hysteresis [mV] Related to this limits serveral analogmacros can be started automatically.	0
Redefine analoguemacro (since V1.1)	ESC	٧	М	n1	n2				Assign analoguemacrofunction n1=019 with analoguemacro number n2=0255.	
Bargraph for AIN1/AIN2			В	ch	no				Assigns bargraph no=120 to analogue input ch=12 (it is possible to assign more than one bargraph to an anlogue input). Define start- endvalues (sv, ev) for bargraph in [mV/20] (see comand 'ESC B RLOU')	
Instrument for AIN1/AIN2	ESC	V	+	ch	no				Assigns instrument no=14 to analogue input ch=12 Define start- endvalues (sv, ev) for bargraph in [mV/20]	
Redraw bargraph			R	ch					Redraw all bar graphs defined for channel ch=12	
								User	values - Format text output	
User value color		F	٧	ch	fg	bg			Set color for string output of channel ch=12; fg= foreground, bg= background color	8,1
User value Font			F	ch	n1				Set font n1 for channel ch=12	5
User value zoom	ESC		Z	ch	n1	n2			Set zoom factor for channel ch=12; n1=X-Zoom 1x8x; n2=Y-Zoom 1x8x	1,1
User value additional width/height		٧	Υ	ch	n1	n2			n1=015: additional width left/right; n2=015: additional height top/bottom for channel ch=12;	0, 0
User value angle			W	ch	n1				Set writing angle for channel ch=12; n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°;	0
User values / scaling	ESC	v	E	ch	forn	nat str	ring	NUL	Set user value for channel ch=12. Format String: "mV1=uservalue1;mV2=uservalue2". 'NUL' (\$00) = termination Assign two voltages (0.5000mV) to user defined values max. range: 4 1/2 digits 19999 + decimal point ('.' oder ',') + sign e.g. display for 2000 mV input should be "-123.45" and "0.00" for 1000mV Format String: "2000=-123.45;1000=0"	0 =0.00 5000 =5.00
Send user value			S	ch					This will send current voltage as formated string for channel ch=12 to sendbuffer	
Display on terminal	ESC	٧	Т	ch					Show formated string of channel ch=12 on termial window	
Display user value	Ī		G	ch	xx1	yy1			Show formated string of channel ch=12 at coordinate xx1,yy1	



					EA eDII	PTFT70-	A: Backlight commands	after
Command	Cod	les				F	Remarks	reset
Illumination brightness			Н	n1		9	Set brightness of the LED illumination n1=0 to 100%.	[100]
Increase brightness			N			I	ncrease brightness of the LED illumination (one step=1%)	
Decrease brightness			Р				Decrease brightness of the LED illumination (one step=1%)	
Brightness changetime			Z	n1		1	Fime n1=031 in 1/10sec for changing brightness from 0 to 100%	[5]
Illumination on/off	ESC	Υ	L	n1		L	LED n1=0: OFF; n1=1: ON; n1=2 to 255: LED switched ON for n1/10sec	1
Assign bar with backlight			В	no		ļ	Assign bar no=120 for changing brightness of the backlight	0
Asign instrument with backlight			+	n1		E	Brightness is connected to instrument 14.	0
Save parameter			@			9	Save the actual brightness and changetime for poweron to EEPROM	

					EA	eD	PTFT	70-A: I/O port commands	after
Command	Cod	les						Remarks	reset
Write output port			w	n1	n2			n1=0: Set all 8 output ports in accordance with n2 (=8-bit binary value) n1=18: Reset output port n1 (n2=0); set (n2=1); invert (n2=2)	Ports 1-8=0
Read input port			R	n1				n1=0: Read all 8 input ports as 8-bit binary value (to sendbuffer) n1=18: Read input port <n1> (1=H-level=VDD, 0=L-level=GND)</n1>	
Port scan on/off	ESC	Υ	Α	n1				The automatic scan of the input port is n1=0: deactivated; n1=1: activated	1
Invert input port			ı	n1				The input port is n1=0: normal; n1=1: evaluated inverted	0
Matrix keyboard			М	n1	n2	n3		Specifies an external matrix keyboard at the inputs and outputs. n1=number of inputs (18); n2=number of outputs (08); n3=debouncing (07)	0
Redefine input bitmacro		.,	D	n1	n2	n3		input port n1=18 is assigned by falling edge n2=0 to new BitMacro number n3=0255 input port n1=18 is assigned by rising edge n2=1 to new BitMacro number n3=0255	
Redefine matrixmacro for keys	ESC	Υ	х	n1	n2			Assign keynumber n1=165 with matrixmacro number n2=0255 After release the key n1=0 run matrixmacro number n2=0255	
Extended outputs	ESC	Υ	E	n1	n2	n3		Set the outputs of the external 74HC4094 from port n1=0255 to port n2=0255; n3=0: low; n3high; n3=2: invert;	3=1:

					E/	\ eD	IPTF	T70-A: Sound commands	after
Command	Cod	es						Remarks	reset
Tone buzzer on/off	ESC	Υ	S	n1				The tone output (Pin 16) becomes n1=0: OFF; n1=1: ON; n1=2255: ON for n1/10s	AUS
Sound settings	ESC	Υ	v	n1	n2	n3	n4	n1=07: set volume for a soundqueue and tonscale notes n2=0127: wait n2x10ms between the sounds n3=115: repeat counter; n3=0: loop n4=02: Mode; 0=ignbore touchsounds, 1=interrupted, 2=stops by touchsounds	7,10,1,1
Play soundqueue	ESC	Υ	Q	dat	a	NUL		play soundqueue; sound nr 180 (max. 90 Sounds); #YQ 0 stops playing;	
Notenliste	ESC	Υ	т	"Not	estr"	NUL		Allowed characters in "notestr" (max 90 notes): CDEFGAH = notes from one-line octave; cdefgah = notes from two-line octave; P= pause; 18: time divisor for following notes	=
Play touchsound	ESC	Υ	-	n1	n2			play sound number n1=180; n2=07: volume; a played soundqueue will be affected ('#YV')	

						EΑ	\ eDl	PTF	T70-A: Other commands	after
Command	Code	es							Remarks	reset
String table code	ESC	s	Т	n1					n1=0: no use of internal strings n1>0: after code n1 appears following codes are internal string numbers	0
Define color	ESC	F	Р	no	R5	G6	B5		Set a new RGB value for color no. n1=132 (R5:Bit73; G6:Bit72; B5:Bit73)	
Wait (pause)	ESC	Х	n1						Wait n1 tenths of a second before the next command is executed.	
Set RS485 address	ESC	κ	Α			adr			For RS232/RS485 operation only and only possible when Hardware address is 0. The eDIP is assigned a new address adr (in the Power-On macro).	
Send bytes			В	num		dat	ta		num (=1 to 255) bytes are sent to the sendbuffer data = num Bytes. In the source text of the macro programming, the number num must not be specified. This is counted by the ediptftcompiler and entered.	
Send version	ESC	S	٧						The version is sent as a string to sendbuffer e.g. "EA eDIPTFT32-A V1.1 Rev.A TP+"	
Send projectname			J						The macro-projectname is sent as a string to the sendbuffer e.g. "init / delivery state"	
Send internal infos			Ī						Internal information about the edip is sent to the sendbuffer.	



Page 18

			Е	A e	DIP.	TFT	70-	A: C	om	ma	nds	r the touch panel	after
Command	Cod	les							Ren			-	reset
										Touc	h pre	ts	-
Touch bordercolors	ESC	F	Е	n1	n2	n3	s1	s2	s3			(032) for touch borders (ESC AT AK). elected; 1=frame outside; 2=frame inside; 3=filling	8,1,2 8,1,7
Touch borderform		Α	Е	n1	n2				n1=1	255	borde	mber; n1=0 no border; n2=angle 0=0°; 1=90°; 2=180°; 3=270°	1,0
Touch button colors	ESC	F	С	nf	nb	sf	sb) for monochrome touch buttons (ESC AU AJ). d; f=foreground; b=background	8,1 8,1
Touch button number		Α	С	n1	n2	n3	n4		n1=0	255	buttor	mber; n2=button angle; n3=X-Zoom 18; n4=Y-Zoom 18	1,0,1,1
Radio group for switches	ESC	A	R	n1					n1=1 Only of a s	to 25 1 swi switch	5: nev tch in in a g	switches do not belong to a group. defined switches belong to the group with the number n1. roup is active at one time; all the others are deactivated. In the case p, only the down code is applicable, the up code is ignored.	0
											ont p		ľ
Font color	ESC	F	Α	nf	sf				_			ing. nf=normal fontcolor; sf= fontcolor for selection	8,1
Label font	4		F	n1	<u> </u>	<u> </u>						nber n1 for touch key label	5
Label zoom factor		١	Z	n1	n2							(1x to 8x); n2 = Y-zoom factor (1x to 8x)	1,1
Additional width/height	ESC	Α	Υ	n1	n2							width left/right; n2=015: additional height top/bottom	0,0
Label angle	_		W	n1								n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°	0
Offset for selected label			0	n1	n2							ffset; n1,n2=07 (add +8 for negative direction)	0, 0
									De	fine	toucl	eas The area from xx1,yy1 to xx2,yy2 is defined as a key	
Define touch key	ESC	A	T	xx1	yy1 yy1	xx2	ир	dow Cod	up Cod	text 	NUL	The area from xx1,yy1 to xx2,yy2 is defined as a switch The actual button is loaded to xx1,yy2 and defined as a key The actual button is loaded to xx1,yy2 and defined as a switch wn code': (1-255) return/touchmacro when key pressed. code': (1-255) return/touchmacro when key released. wn/up code = 0 press/release not reported). 't' this is a string that is placed in the key with the current touch font. first character determines the alignment of the text (C=centered, L=	
Define touch switch (status of the switch	ESC	Α	к	xx1	yy1	xx2		dow Cod	up Cod	text 	NUL	ified, R=right justified). Multiline texts are separated with the charact C, dec: 124); ional: after the character '~' (\$7E, dec: 126) you can write a 2nd text	'
toggles after each contact)	ESC	A	J	xx1	yy1	dow Cod	up Cod	text 	NUL			ected touch key/switch e.g. "LED on~LED off" ': (\$00) = end of string	
Define touch key with menu function	ESC	A	М	xx1	уу1	xx2	уу2	dow	up code	mnu code	text 	The area from xx1,yy1 to xx2,yy2 is defined as a menu key. 'dowr code':(1-255) Return/touch macro when pressed. 'up Code':(1-255) Return/touch macro when menu canceled 'mnu Code':(1-255) Return/touch macro when menu canceled 'mnu Code':(1-255) Ret macro+(item no. 1) after selection of a menu item. (down/up code activation/cancellation is not reported.) 'text':= string with the key t menu items. the first character determines the direction in which the opens (R=right, L=left, O=up, U=down). The second character det the alignment of the touch key text (C=centered, L=left justified, R justified). The menu items are separated by the character ' (\$TC, (e.g. "uckey item1 item2 item3". The key text is written with the cu font and the menu items are written with the current menu font. The background of the menu is saved automatically.) urn/menu = 0: ext and the le menu ermines =right dec:124) crent touch
Define drawing area	ESC	Α	D	xx1	yy1	xx2	уу2	n1	fg			a is defined. You can then draw with a line width of n1 and color fg w rdinates xx1,yy1 and xx2,yy2.	thin
Define free touch area	ESC	Α	н	xx1	yy1	xx2	уу2			corne	er coo	e touch area is defined. Touch actions (down, up and drag) within the ates xx1,yy1 and xx2,yy2 are sent.	
Set bar by touch	ESC	Α	В	n1						The I	oargra	with number n1 is defined for input by touch panel.	
Set instrument by touch	ESC	Α	+	n1	L					The i	nstrur	t with number n1 is defined for input by touch panel.	

TOUCH PANEL

The Version EA eDIPTFT70-ATP is shipped with an analog, resistive touch panel. Up to 100 touch areas (keys, switches, menus, bar graph inputs) can be defined simultaneously. The fields can be defined with pixel accuracy. The display supports user-friendly commands. When the touch "keys" are touched, they can be automatically inverted and an external tone can sound (pin 16), indicating they have been touched. The predefined return code of the "key" is transmitted via the interface, or an internal touch macro with the number of the return code is started instead.



			Е	A e	DIP.	ΓFT70-A	: Commands for the touch panel	after
Command	Cod	les					Remarks	reset
							Global settings	•
Touch query on/off	ESC	Α	Α	n1			Touch query is deactivated (n1=0) or activated (n1=1)	1
Tarrell Iran management	ESC	А	ı	n1			Automatic inversion when touch key touched: n1=0=OFF; n1=1=ON;	1
Touch key response	ESC	А	s	n1			Tone sounds briefly when a touch key is touched: n1=0=OFF; n1=1=ON	1
Touch key vibration response	ESC	А	*	n1			n1=0: vibration off n1=13: vibration intensity 1=low, 2= mid, 3=high	
Touch key sound response ESC A - n1 n2 n3 Send bar value on/off ESC A Q n1					n2	n3	Set touch sound: n1=0: all types n1=6: free area (#AH) n1=1: key (#AT/#AU) n1=7: make menu(#AM) n1=2: switch (#AJ/#AK) n1=8: menu box(change menu entry) n1=3: bar (#AB) n1=9: keyboard (#KS) n1=4: instrument (#A+) n1=10: edit box (#ET) n1=5: all types draw (#AD) n2=180: sound number (0=no sound; 255=no change) n3=07: sound volume (255 no change) Automatic transmission of a new bar graph / instrument value by touch input is n1=0: deactivated n1=1: is placed in the sendbuffer once at the end of input n1=2: changes are placed continious in the sendbuffer during input	1,19,7 2,50,7 3,23,6 4,24,4 5,23,6 6,23,6 7,43,7 8,14,6 9,9,5 10,51,6 d;
Invert touch key			N	code			Other functions The touch key with the assigned return code is inverted manually	
Set touch switch	-		P	code	-		The status of the switch is changed by means of a command (n1=0=off; n1=1=on)	
Query touch switch	ESC	Α	X	code			The status of the switch with the return code (off=0; on=1) is placed in the sendbuffer	
Query radio group	1		G	n1			down code of the activated switch from the radio group n1 is placed in the sendbuffer	
Delete touch area	ESC	А	L	code	n1		The touch area with the return code (code=0: all touch areas) is removed from the touch query. n1=0 the area remains visible on the display; n1=1, the area is deleted.	
			v	xx1	yy1	n1	remove the touch area that includes the coordinates xx1,yy1 from the touch query. n1=0: area remains visible; n1=1: Delete area	

TOUICH ADJUSTMENT

The touch panel is perfectly adjusted and immediately ready for operation on delivery. As a result of aging and wear, it may become necessary to readjust the touch panel:

- 1a. Send Command 'ESC A@' or
- 1b. Touch the touch panel at power-on and keep it depressed. After the message "touch adjustment?" appears, release the touch panel. Touch the touch panel again within a second for at least a second.
- 2. Follow the instructions for adjustment (press the 2 points upper left and lower right corner).

					-	EΑ	eDII	PTF	T70: menu commands	after			
Command	Coc	les							Remarks	reset			
							Se	etting	s for menu box / touch menu	_			
Set menu colors	ESC	F	N	fg	bg	fc			Colors 132; fg=for characters; bg=for background; fc=for frame	8,1,8			
Set menu font			F	n1					Set font with the number n1 (0 to 31) for menu display	5			
Menu font zoom factor			z	n1	n2				n1 = X-zoom factor (1x8x); n2 = Y-zoom factor(1x8x)	1,1			
Additional character width/height	ESC	N	Υ	The state of the s									
Menu angle			w	n1					Menu display angle n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°	0			
Touch menu automation			т	n1									
						Mer	nu co	mma	nds (control with keys not by touch)				
Define and display menu			D	xx1	yy1	nr	Text 	NUL	A menu is drawn at corner xx1,yy1 with the current menu font.no=currently inverted entry (e.g.: first entry). "text"=string with menu items, the different items are separated by the character ' ' (\$7C,dec:124) (e.g. "item1 item2 item3"). The background of the menu is saved automatically.lf menu is already defined, it is automatically canceled+deleted	i = a			
next item			N		•			•	The next item is inverted or remains at the end				
previous item	ESC	N	Р						The previous item is inverted or remains at the beginning				
End of menu / send			s						The menu is removed and replaced with the original background. The current item is send as a number (1 to n) (0=no menu displayed)				
End of menu / macro			М	n1					The menu is removed and replaced with the original background. Menu macro n1 is called from item1, menu macro nr+1 for item2				
End of menu / cancel			Α						The menu is removed and replaced with the original background.				



					E	le ا)IP	TFT'	70:	Edi	t bo	ox commands	after
Command	Cod	les							Rer	nerk	S		reset
										_	_	edit box	1.000.
Edit box frame colors	500	F	Q	n1	n2	n3						colors 132 (0=transparent) for edit box: de; n2=frame inside; n3=filling	8,1,1
Edit field text colors	ESC	F	н	fd	bd	fa	ba					(132) for edit field: nd; b=font background; d=deactivated; a=activated edit box	8,1 1,8
Offset for edit field			0	left	top	right	bot		Set o	offset	from	editbox frame to edit field	0,0,0,
Set font	ESC	E	F	n1					Set f	ont n	umbe	er n1=0255	5
Set frame	ESC	_	Ε	n1					n1=1	255	: fran	ne number for editbox; n1=0: no frame	1
Save edit box background			Н	n1					n1=1	: sav	e edit	t box background into clipboard layer; n1=0: background is not saved	0
									0	efin	e edi	it box	
			L	-						def.		L: left; C: centerR: right assigned; Define edit box number no=115 from xx1,yy1 to xx2,yy2 and draw	keine Edit-
Define and show edit box	ESC	E	C R	no_	n2	xx1	yy1	xx2	yy2	Text 	NUL	n2=max. text length; n2=0: no limit default Text=userdefined default text; 'NUL' (\$00): End of string	box def.
Delete edit box			D	no	n2			•	n2=0): area	a rem	no=115 (no=0: all edit boxes) nains visible e or restore from clipboard layer (see 'ESC E H')	
Set edit box by touch	ESC	E	т	no	n2	n3						p=115 can be activated by touch panel. uchmacro; n3=deactivate touchmacro (n2/n3=0 touchmacro is not used)	
Set password character			Р	no	code							=115 input mode: aracter are shown; code=1255: use wildcard for each character	0
									Į	Jsing	edi	t box	
Activate edit box			Α	no					Activ	ates	edit b	ox with no=115 for editing; no=0: deactivates actually selected edit box	
Get edit box string			G	no								edit box no=115 into sendbuffer dit box strings into sendbuffer	
Send character to edit box	ESC	E	В	code					8: B <i>i</i> 10: 0 12: 0	ACKS CANC CLEAR	PACI EL edi	racter (code) to active edit box; special codes: E deletes last character ding, set to default text t field string into sendbuffer, copy as default text	
Send string to edit box			s	Te	ı xt	NUL						o active edit box; 'NUL' (\$00): end of string	

				ΕA	eD	IPT	FT7	'0: T	ouch Keyboard commands	after
Command	Cod	les							Remarks	reset
									Frame settings	-
Keyboard frame colors		_	Κ						Set frame colors 132 (0=transparent) for keys (special keys code <32) n=normal;	812817
Special key frame colors	ESC	F	s	n1	n2	n3	s1	s2	s=selected; 1=frame outside; 2=frame inside; 3=filling	819816
Keyboard frame	ESC	К	Е	n1	n2				Set frame for keyboard frames: n1=1255 normal keys (Code >= 32) n2=1255 special keys (Code < 32); n1/n2=0: draw no frame	1
									Keyboard label settings	
Keyboard label color	ESC	_	F						Ser the textcolor 132 for keys.	8,1
Secial key label color	ESC	F	Υ	- n1	s2				(special keys Code<32) n1=normal textcolor; s2=textcolor for selection	1,8
Keyboard label font			F	n1	n2	s1	s2		Set font (1255) for keyboard label: n=normal keys; s=special keys; 1=single letters; 2=strings	6,5
Key label string	ESC	K	L	code	Te	xt	NUL		code=keycode; text= alternative labelfor key (e.g. for special keys) code=0: Clear alternative label for all keys	0
Label offset for selection			0	n1	n2				n1=X-offset; n2=Y-offset; n1,n2=07 +8 for negative direction	0, 0
									Keyboard layout	
Define keyboard	ESC	к	В	no	code	estr	NUL		define keyboard with number no=14 definieren; "codestr" = string with keycodes, several lines are seperated by ' '; 'NUL' (\$00): end of string use backslash '\' (\$5C) for special keycodes: 11,\(2\),\(3\),4: show keyboard no. 1, 2, 3, 4 15: SHIFT use keyboard 2 for one key autofallback to no. 1 16: CAPSLOCKtoggle between keyboard no. 1 and no. 2 18: send code 8 = BACKSPACE for edit box 18: send code 10 = CANCEL for edit box 19: send code 12 = CLEAR for edit box 10: send code 13 = SEND for edit box 10: send code 13 = SEND for edit box 10: placeholder for a key with is not used and drawn	no key- board de- fined
Keyboard position	ESC	к	Р	xx1	yy1	xx2	уу2	gap	xx1,yy1 bis xx2,yy2: rectangle enclosing the keyboard gap: space between the keys	
									Using keyboard	
Show keyboard			s	no	n2				Show keyboard no=14 n2=0: put pressed keys into sendbuffer. n2=115: activate edit box and send pressed keys to it	
delete keyboard	ESC	к	D	n1					remove actually shown keyboard (touch areas) from screen: n1=0: keyboard stays visible n1=1: delete area or restore from clipboard (see command 'ESC KH')	
Keyboard background			н	n1	n2				n1=1: save background into clipboard layer; n1=0: background is not saved Set background color n2=132 (0=transparent)	0, 0
Key matrixmacro			М	code	n2				code=keycode (code=0: all keycodes) Set MatrixMacro for all codes n2 = 1255 MatrixMacro number if key is pressed (n2=0: run no macro)	0



RESPONSE OF THE EA EDIPTFT70-A VIA SERIAL INTERFACE

The table below contains all response codes. Some response data will come automatically some others on request. In addition to that with command 'ESC SB ...' user is able to transmit individual data packages. All reponses are placed into the sendbuffer. With the smallprotocol command 'Request for content of send buffer' (see page 10) the host can read out the sendbuffer. This can be done per polling, alternativly pin 20 'SBUF' shows with LO-signal that data is ready to transmit.

							R	esponses of the EA eDIPTFT70-A
lc	ł	num			data			Remarks
							auto	matic responses (placed into sendbuffer)
ESC	Α	1	code					Response from the analog touch panel when a key/switch is pressed. code = down or up code of the key/swit transmitted if no touch macro with the number code is defined!
ESC	В	2	no	value				When a bargraph is set by touch, the current value of the bar no is transmitted. Transmission of the bar balue activated (see the 'ESC A Q n1' command).
ESC	F	2	no	value				When a instrument is set by touch, the current value of the instrument no is transmitted. Transmission of the intstrument value must be activated (see the 'ESC A Q n1' command).
ESC	K	1	code					Response of the touch keyboard; code = keycode. It's only transmitted if there is no MatrixMacro defined and box.
ESC	Р	1	value					After the input port is changed, the new 8-bit value is transmitted. The automatic port scan must be activated. A n1' command. It is only transmitted when there is no corresponding port/bit macro defined!
ESC	М	1	no					When a keystroke of the external matrix keyboard is detected, the newly pressed key number no is transmitte transmitted if no corresponding matrix macro is defined!
ESC	Н	5	type	xLO	xHI	yLO	yHI	The following is transmitted in the case of a free touch area event: type=0 is release; type=1 is touch; type=2 free touch area at the coordinates xx1, yy1
					Respo	onse o	only v	when requested by command (placed into sendbuffer)
ESC	В	2	no	value				After the 'ESC B S n1' command, the current value of the bar with the number no is transmitted.
ESC	F	2	no	value				After the 'ESC I S n1' command, the current value of the instrument with the number no is transmitted.
ESC	Х	2	code	value				After the 'ESC A X' command, the current status (value=0 or 1) of the touch switch code is transmitted.
ESC	G	2	no	code				After the 'ESC A G nR' command, the code of the active touch switch in the radio group no is sent.
ESC	E	num	no	edit bo	x string			After the 'ESC E G nr' command or sending the code '13' to the active edit box ('ESC E B code'), the content nr= number of the edit box (string length=num-1)
ESC	Υ	2	no	value				After the 'ESC Y R' command, the requested input port is transmitted. no=0: value is an 8-bit binary value of a no=18: value is 0 or 1 depending on the status of the input no
ESC	D	3	ch	LOval	HIval			After the 'ESC V D ch' command, the requested voltage of channel ch=12 will be sent (value = 05000mV)
ESC	w	num	ch	sca	aled AS	CII strin	<u> </u>	After the 'ESC V S ch' command, the requested voltage of channel ch=12 will be set as scaled ASCII charactering = num-1).
ESC	V	num		vers	ion stri	ng		After the 'ESC S V' command, the version of the edip firmware is transmitted as a string e.g. "EA eDIPTFT70-A V1.0 Rev.A TP+"
ESC	J	num		project	name s	tring		After the 'ESC S J' command, the macro-projectname is transmitted. e.g. "init / delivery state"
ESC	ı	21	CF	RC-ROI	M, CRC OF in KE	on, Tou C-ROMs B, Fsoll, DF	oll	after the 'ESC S I' command, internal information is sent by eDIP (16-Bit integer values LO-HI Byte) Version: LO-Byte = version number Software; HI-Byte = Hardware revison letter touch Touchinfo: LO-Byte = '- +' X direction detected; HI-Byte = '- +' Y direction detected DFIen: number of user bytes in data flash memory (3 Bytes: LO-, MID- HI-Byte)
							Resi	conses without length specification (num)
ESC	U	L	xx1	yy1		age data 6-FORM	a	after the 'ESC UH' command, a hard copy is sent in G16-format. xx1,yy1 = Start coordinates of the hard copy (upper left corner)



Page 22

PRELOADED FONTS

As standard, there are 3 monospaced, 3 proportional character sets and 2 large digit fonts integrated. The proportional character sets (which have a narrow "I" and a wide "W", for example) look better and take up less space on the screen. Each character can be placed with **pixel accuracy**, and its height and width can be increased by a factor of 1 to 8. A text can be output left justified, right justified or centered. Rotation in 90° steps is possible. Macro programming permits further fonts to be integrated.

All kinds of fonts can be converted from True-Type Fonts by using using the LCD toolkit/eDIPTFTcompiler⁹ (the USB Evaluation Board EA 9777-3USB is required).

") Web under http://www.lcd-module.com/products/edip.htm

+ 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)		·	"		5	8	8		C)	¥	+		-		,
\$30 (dez: 48)	0	1	2	3	4	5	6	7	В	9	:		<	=	>	?
\$40 (dez: 64)	0	A	В	c	D	E	F	G	н	I	J	К	L	н	n	0
\$50 (dez: 80)	P	a	R	s	т	и	U	н	×	γ	z	ι	,	1		-
\$60 (dez: 96)		a	ь	-	d	e	f	9	h	i	j	k	ı	н	n	
\$70 (dez: 112)	Р	9	r	,	t	u	v		×	9	ı	•	ı	>	"	۵
\$80 (dez: 128)	E	ü			ä										ă	
\$90 (dez: 144)					ä					8	ü				β	

Font 1: 4x6 monospaced

+ Lower Upper	\$0 (0)	\$1 (1)	\$2 (2)	\$3 (3)	\$4 (4)	\$5 (5)	\$6 (6)	\$7 (7)	\$8 (8)	\$9 (9)	\$A (10)	\$B (11)	\$C (12)	\$D (13)	\$E (14)	\$F (15)
\$20 (dez: 32)		į	**	#	\$	z	8.	,	C)	*	+	,	-		7
\$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$40 (dez: 64)	6	A	В	С	D	Е	F	G	Н	I	J	K	L	н	N	0
\$50 (dez: 80)	Р	Q	R	s	Т	U	V	н	X	Y	z	ſ	٨	1	^	_
\$60 (dez: 96)	•	a	ь	С	d	е	f	9	h	i	j	k	ι	m	n	0
\$70 (dez: 112)	Р	q	r	s	ŧ	u	Ų	н	x	y	z	{	1	}		۵
\$80 (dez: 128)	ε	ü	é	â	ä	à	å	Ç	ê	ë	è	ï	ì	ì	Ä	A
\$90 (dez: 144)	É	æ	Æ	ô	ö	ò	û	ù	ÿ	ö	Ü	¢	£	¥	В	f
\$A0 (dez: 160)	á	í	ó	ú	ñ	Ñ	<u>a</u>	0	ż	-	,	½	1/4	i	*	**
\$B0 (dez: 176)																
\$C0 (dez: 192)																
\$D0 (dez: 208)																
\$E0 (dez: 224)	α	ß	Γ	π	Σ	σ	μ	۳	Φ	θ	Ω	8	ø	ф	ε	n
\$F0 (dez: 240)	=	±	Σ	۷.	ſ	J	÷	n	0	•	•	1	n	2	3	-

Font 3: 7x12 monospaced

gramm	IIIQ	J P	211	ш	5 11	וווו	IEI	IU	HIG	5 K	טכ	e II	ILE	gı	ale	tu.
+ 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)		ļ	=	#	\$	z	8.	,	()	*	+	,	-		/
\$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:	j	<	=	>	?
\$40 (dez: 64)	e	A	В	С	D	Ε	F	Œ	Η	I	J	К	L	М	N	0
\$50 (dez: 80)	Р	Q	R	s	Т	U	٥	3	х	Υ	z	С	\	ם	^	-
\$60 (dez: 96)	•	а	b	С	d	e	f	g	h	i	j	k	1	m	n	0
\$70 (dez: 112)	р	9	r	s	t	u	٧	W	×	9	z	{	1)	~	۵
\$80 (dez: 128)	€	ü	œ,	ā	ä	,a	á	ç	ΙŒ	ë	ė	ï	î	ì	Ä	À
\$90 (dez: 144)	É	æ	Æ	8	ö	,	a	ù	ij	ö	ü	¢	£	¥	β	f
\$A0 (dez: 160)	á	í	0	ű	ñ	Ñ	<u>a</u>	01	٤	-	7	lį	lsį	i	«	»
\$B0 (dez: 176)																
\$C0 (dez: 192)																
\$D0 (dez: 208)																
\$E0 (dez: 224)	α	β	г	π	Σ	σ	Д	т	Φ	θ	Ω	8	ø	ø	E	Π
\$F0 (dez: 240)	=	±	2	<u> </u>	٢	J	÷	25	0	•		1	n	2	3	-

Font 2: 6x8 monospaced

+ Lower Upper	\$0 (0)	\$1 (1)	\$2 (2)	\$3 (3)	\$4 (4)	\$5 (5)	\$6 (6)	\$7 (7)	\$8 (8)	\$9 (9)	\$A (10)	\$B (11)	\$C (12)	\$D (13)	\$E (14)	\$F (15)
\$20 (dez: 32)		į		#	\$	%	&		()	×	+	,	-		7
\$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:	j	<	=	>	?
\$40 (dez: 64)	@	Α	В	С	D	Е	F	G	Н	ı	J	К	L	М	N	0
\$50 (dez: 80)	Р	Q	R	s	т	U	٧	W	Х	Υ	Z	[٨]	٨	_
\$60 (dez: 96)	,	a	Ь	С	d	e	f	g	h	i	j	k	1	m	n	0
\$70 (dez: 112)	Р	q	r	s	t	u	٧	W	×	y	z	{	ı	}	~	Δ
\$80 (dez: 128)	€	ü	é	â	ä	à	oa	ç	ê	ë	è	ï	î	ì	Ä	Ã
\$90 (dez: 144)	É	æ	Æ	ô	ö	ò	û	ù	ij	Ö	Ü					
\$A0 (dez: 160)	va	í	ó	ú	ñ	Ñ	a	0								
\$B0 (dez: 176)																
\$C0 (dez: 192)																
\$D0 (dez: 208)																
\$E0 (dez: 224)		В														
\$F0 (dez: 240)									٥							

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)		į		#	\$	%	8	'	()	*	+	,	-		7
\$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$40 (dez: 64)	0	A	В	С	D	E	F	G	Н	ı	J	к	L	м	N	0
\$50 (dez: 80)	Р	Q	R	s	Т	U	U	ш	н	γ	z	I	١	1	^	_
\$60 (dez: 96)	1	а	b	С	d	е	f	g	h	i	j	k	I	m	n	0
\$70 (dez: 112)	p	q	r	s	t	u	υ	w	н	y	z	{	ı	}	~	Δ
\$80 (dez: 128)	€	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Â
\$90 (dez: 144)	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü					
\$A0 (dez: 160)	á	í	ó	ú	ñ	Ñ	<u>a</u>	0								
\$B0 (dez: 176)																
\$C0 (dez: 192)																
\$D0 (dez: 208)																
\$E0 (dez: 224)		ß														
\$F0 (dez: 240)									۰							

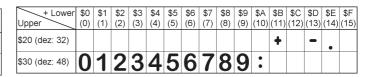
Font 5: CHICAGO14 proportional

+ 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)		İ	11	#	\$	%	&	,	()	*	+	,	_		1
\$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$40 (dez: 64)	@	A	В	C	D	E	F	G	H	I	J	K	L	M	N	C
\$50 (dez: 80)	P	Q	R	S	T	U	٧	W	X	Y	Z		1]	^	
\$60 (dez: 96)	6	a	b	C	d	е	f	g	h	i	j	k	I	m	n	O
\$70 (dez: 112)	p	q	r	S	t	u	٧	W	X	y	Z	{	1	}	2	Δ
\$80 (dez: 128)	€	ü	é	â	ä	à	å	Ç	ê	ë	è	ï	î	ì	Ä	مْ
\$90 (dez: 144)	É	æ	Æ	ô	Ö	Ò	û	ù	ÿ	Ö	Ü					
\$A0 (dez: 160)	á	ĺ	Ó	Ú	ñ	~	<u>a</u>									
\$B0 (dez: 176)																
\$C0 (dez: 192)																
\$D0 (dez: 208)																
\$E0 (dez: 224)		β														
F0 (dez: 240)									0							

Font 6: Swiss30 Bold proportional

+ Lower	\$0	\$1	\$2	\$3	\$4	\$5	\$6	\$7	\$8	\$9	\$A	\$B	\$C	\$D	\$E	\$F
Upper	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
\$20 (dez: 32)												+		-		
\$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:					

Font 7: big numbers BigZif50



Font 8: big numbers BigZif100



This hard copy shows all the fonts with which the product is shipped

ADDITIONAL FONTS

Up to 256 fonts á 16 pages can be loaded into the internal DataFlash.



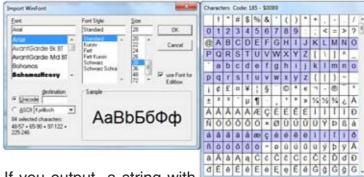
Page 24

COMPILER OPTION "WinFont:"

It is possible to raster TrueType-Fonts in different sizes witch can be used. A double-click to the fontname within the KitEditor opens the font selection box.



To simplify the use of fonts, there is the possibilty



of a edit box. If you output a string with

KitEditor (e.g. #ZL 5,5, "Hello"), you can perform a double click on the string to open it. Now you can select the desired characters. This is mainly recommended using cyrillic, asian or symbol fonts.

In that way, the KitEditor automatically places the right ASCII-Code. Alternativly you can use instead of the quotation mark curly brackets (e.g. +ZL 5,5, {48656C6C6F}).

COMPILER OPTION "Font:"

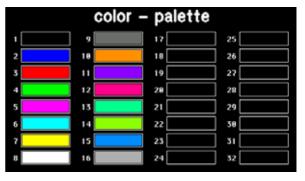
Following font formats can be used:

- FXT: Textfont as used by eDIP240/320 and KIT series
- G16: internal eDIPTFT format (with this format it is possible to user color fonts)

65,536 COLORS

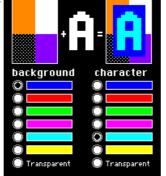
EA eDIPTFT32-A is able to work with 65,536 colors for true-color pictures/icons and animations. For an easy use there exists a color palette with 32 entrys (16 colors are

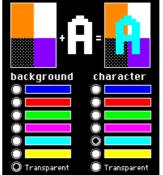
predefined after PowerOn). This color palette can be redefined at any time without changing the content of the display (command: ESC FP no R G B). To use a color for text and graphic functions you set only a

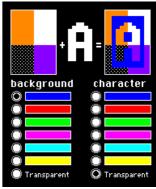


number between 1..32. The dummy color number 255 means that the actually color is not changed e.g you want only to change the foreground- and not the background color. The color number 0=transparent is special and can be used for background of character e.g. that means that for placing a character no rectangular field will be deleted around the character itself. The sensless combination

transparent background and transparent foreground is used to invert all dots (=complementary). Two times inverted will end same as action was started (original drawing is restored).





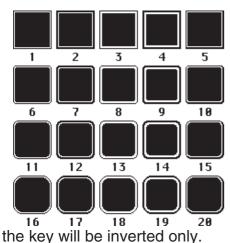


show some examples to show the letter 'A' onto a fixed background

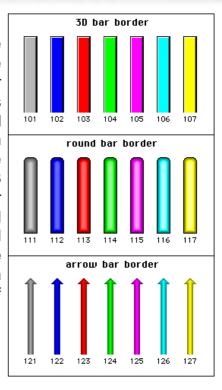


BORDERS, KEY STYLES AND BARGRAPH

The eDIP is shipped with 20 predefined border (no:1..20) for the commands draw box frame and draw touchkeys. There are also three



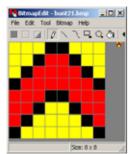
special borders in various colors for using with bargraph commands (no:101..107, 111..117 and 121..127). All of them can be used in various sizes via coordinates. The frames 1..20 are split into 3 segments: the outer frame, inner frame and filling. Each segment will get an individual colour for normal and for selected state. This will give the user the opportunity, when touching a field, the individual part of



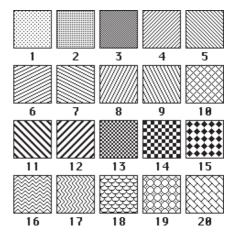
FILL PATTERNS

A pattern type can be set as a parameter with various commands. In this way, for example, rectangular areas and bar graphs can be filled with different patterns. The eDIP is shipped with 20 predefined fill patterns.

Define own pattern:



With the LCD-tools it is possible to define new pattern (=bitmaps with exactly 8x8 dots). (Compileroption "Pattern:"). The foreground- and backgound color can be set for monochrome pattern (as the 20 preloaded pattern) incl. transparency. It is also possible to define full colored pattern. With the LCD-tools some sample pattern has been installed (see folder 'Pattern').





Page 26

DEFINE OWN BORDER

PosXY: 19,3 (right-click to pickup color)

🕏 BitmapEdit - Border25.bmp

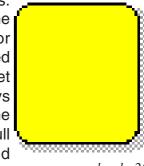
File Edit Tool Bitmap Help

With the LCD-tools it is possible to define new borders (Compileroption "Border:"). Each of these new

border is a bitmap with exactly 24x24 dots (9 segments with 8x8 dots: 4x edge, 4x middle part, 1x filling).

Scaling for bigger touchkeys/frames will be done by

repetition of these 8x8 dot segments. This makes it necessary to keep the 8x8 size in every case. If 4-color bitmaps are used (as the preloaded border 1..20) the color can be set individually (the first color is always transparent and is not used by the eDIP). It is also possible to define full colored border (as the preloaded border for bargraphs 101...127). With the LCD-tools some sample border



border25: 50x56 Dot size

has been installed (see folder 'Bitmaps\Color\Border').

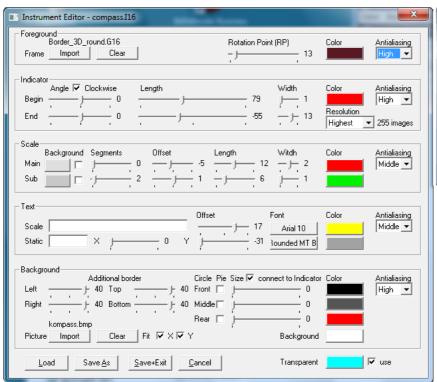
ROTARY AND POINTER INSTRUMENTS

Size: 24 x 24

□ □ | Ø \ \ \ □ □ Ø | ♦ ♦ 🗘 | C 🗓 B

With the help of the LCD-Tools it is possible to include instruments (Compiler command: Instrument: 4, <instrument.i16>). Performing a double click in the KitEditor on the instrument file opens the instrument editor.

The instruments are supported by convinient commands ('ESC I..'). For example the instruments are connectable to an analog input. In addition they are configurable by touch.







Einige Beispiele für Instrumente:









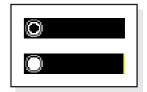


BUTTONS AS KEYS

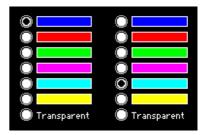
Apart from the border types, which are infinitely scalable, it is also possible to use bitmaps as touch keys or touch switches (Compilerotion "Button:"). A button always consists of two Bitmaps of equal size (one bitmap to display the touch key in its normal state and another for when it is pressed). The active area of the touch key automatically results from the size of the button bitmaps.

SWITCHES IN GROUPS (RADIO GROUPS)

Touch switches change their status from ON to OFF or vice versa each time they are touched. Several touch switches can be included in a group ('ESC A R n1' command). If a touch switch in the group 'n1' is switched on, all the other touch switches in this group are automatically switched off. Only one switch is ever on (see table on page 18).



RadioBlack75x15_0.bmp RadioBlack75x15_1.bmp



 $zwei\ Radio-Gruppen\ mit\ Touch-Schaltern$



Page 28

CREATING INDIVIDUAL MACROS AND IMAGES

To create your own fonts, images, animations and macros you need the following:

- To connect the display to the PC, you need the EA 9777-3USB USB evaluation board, which is available as an accessory, or a self-built adapter with a MAX232 level converter (see the application example on page 5).
- ELECTRONIC ASSEMBLY LCD-Tools*), which contains a kiteditor, bitmapeditor, ediptftcompiler, fonts, images, border, pattern and examples (for Windows PCs)
- A PC with an USB or serial COM interface

To define a sequence of commands as a macro, all the commands are written to a file on the PC (e.g. DEMO.KMC). You specify which character sets are to be integrated and which command sequences are to be in which macros. If the macros are defined using the kit editor, you start the eDIPTFT compiler using F5. This creates a file called DEMO.DF. If an EA 9777-3USB evaluation board is also connected or the display is connected to the PC via a MAX232, this file is automatically burned in the display's data flash memory. You can send the created macrofile *.DF with any other system to the EA eDIPTFT70-A. All programming commands are inside this file, so you only need to send the content of the *.df file (via RS232, SPI or I2C with smallprotocol in packets) to the EA eDIPTFT70-A.

KIT-EDITOR HELP (ELECTRONIC ASSEMBLY LCD TOOLS)

At bottom from the KitEditor window in the statusline you can see a short description for the current command and the parameters. For more information press F1.



") im Internet unterhttp://www.lcd-module.de/deu/dip/edip.htm



IMAGES

To save transfer time via serial interface, it is possible to store up to 256 bitmaps á 16 pages into internal dataflash (Compileroption "Picture:"). Following image file-formats can be used:

- BMP: Windows Bitmap with 1-, 4-, 8-, 16-, 24-, 32-BIT colordepth incl. RLE.
- GIF: Graphics Interchange Format incl. optionally transparency
- JPG: JPEG Compressed Images
- TGA: TARGA Images with 8-, 16-, 24-, 32-BIT colordepth incl. RLE and transparency.
- PNG: Portable Network Graphics incl. colour map, gray scale and transparency
- G16: internal eDIPTFT format, incl. RLE and transparency

All pictures are converted into internal G16 format with RLE encoding (saves memory). Too big pictures are resized proportional (Compileroption "MaxSize:"). It is also possible to reduce the colordepth (Compileroption "MaxColorDepth:"). One color can be defined as transparent (Compileroption "MakeTransparent:") The internal pictures can be used with the command "ESC U I" via serial interface or from a macro. The foreground- and backgound color can be set for monochrome pictures incl. transparency.

ANIMATIONS

It is possible to store up to 256 animations á 16 pages into internal dataflash. (Compileroption "Animation:"). Following image file-formats can be used:

- GIF: animated GIF (only identically transparent areas, transparency can be switched off).
- G16: internal animated eDIPTFT format
- two or more single bitmaps (BMP, GIF, JPG, TGA, PNG, G16) e.g. two bitmaps for blinking Note that max. 4 animations (animationprocesses) can run at the same time. The animations are selfrunning pictures, but you can use the animations manually too. The foreground- and backgound color can be set for monochrome animations.

PATTERN

Patterns are used to fill a box, a bargraph or to draw a line. It is possible to store up to 256 pattern á 16 pages into internal dataflash (Compileroption "Pattern:").

Each bitmap (BMP, GIF, JPG, TGA, PNG, G16) with a size of 8x8 dots can be imported as a pattern. The foreground- and backgound color can be set for monochrome pattern incl. transparency.

BORDER / BARGRAPH

A border can be scaled and is used for rectangles, bargraphs and touch keys/switches. It is possible to store up to 256 border á 16 pages into internal dataflash (Compileroption "Border:")

Each bitmap (BMP, GIF, JPG, TGA, PNG, G16) with a size of 24x24 dots can be imported as a border. The transparency of GIF, TGA, PNG and G16 bitmaps is used by the EA eDIP. It is possible to change the colors for 4-color borders, the first color ist always transparent and is not used by the EA eDIP. When used for a touch key/switch a second border can be loaded witch will be used if the touch key/ switch is pressed.

IMAGES AS TOUCHKEYS (BUTTONS)

It is possible to store up to 256 touchkeys/buttons á 16 pages into internal dataflash. (Compileroption "Button:").

A button consists of one or two images with the same size (BMP, GIF, JPG, TGA, G16). The transparency of GIF,TGA, PNG and G16 bitmaps is used by the EA eDIP and should be identical. The first bitmap is used when the touch key/switch is released and the second bitmap is used if the touch key/ switch is pressed.



Page 30

MACROS

Single or multiple command sequences can be grouped together in macros and stored in the data flash memory. You can then start them by using the Run macro commands. There are different types of macro (compiler directive marked in green letters):

Normal macro Macro:

These are started by means of an 'ESC MN xx' command via the serial interface or from another macro. A series of macros occurring one after the other can be called cyclically (movie, hourglass, multi-page help text). These automatic macros continue to be processed until either a command is received via the interface or a touch macro with a corresponding return code is activated.

Touch macro Touch Macro:

Started when you touch/release a touch field (only in versions with a touch panel - TP) or issue an 'ESC MT xx' command.

Menu macro (1 to 255) MenuMakro:

Started when you choose a menu item or issue an 'ESC MM xx' command.

Bit macro BitMacro:

will be started by a single line IN 1..8 (bit) will change or by command 'ESC MB xx'. Bit-Macro 1..8 are good for falling edge and Bit Macro 9..16 are good for rising edge at input 1..8. It is possible to change the assignment between Bitmacro and intput with command 'ESC YD n1 n2 n3' (see page 17).

Port macro PortMacro:

These are started when voltage (binary) is applied to IN 1..8 or by command 'ESC MP xx'.

Matrix macro MatrixMacro:

Matrix Macro 1..64: start when keypressed or by command 'ESC MX xx'. Matrix Macro 0: start after release of key or by command. It is possible to change the assignment between keynumber and Matrixmacro with command 'ESC YX n1 n2 n3' (see page 17).

Analogue macro AnalogMacro:

will start whenever voltage changes or limit exceeds or by command 'ESC MV xx'. See table at the right:

It is possible to change the assignment between analoguemacrofunction and Analoguemacronumber with command 'ESC VM n1 n2' (see page 15).

Process Makro ProcessMacro:

automatic start at fixed periode (0.1s up to 25s) or by command 'ESC MC xx'. Up to 4 individual process may be defined by command 'ESC MD ..'. These Process Makro will never be stopped by other commands or activities.

Analog Makros					
Makro Nr.		Start des Makres hei			
AIN1	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			



Power-on-macro PowerOnMacro:

Started after power-on. You can switch off the cursor and define an opening screen, for example.

Reset-macro ResetMacro:

Started after an external reset (low level at pin 5).

Watchdog-macro WatchdogMacro:

Started after a fault/error (e.g. failure).

Brown-out-macro BrownOutMacro:

Started after a voltage drop <4.6V (typ.).

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.

WRITE PROTECTION FOR MACRO PROGRAMMING AND FONTS

A LO level at pin 19 (WP) prevents the macros, images and fonts in the data flash memory from being overwritten inadvertently (so it is highly recommended!).

MACRO PAGES (MULTILINGUAL CAPABILITY)

There are 16 complete macro sets available as well as the internal images and fonts. By simply switching the active macro page (ESC M K n1), for example, up to 16 different languages can thus be supported.

If a macro/picture is defined in the kit editor, a page number can be specified in square brackets after the macro/picture number. If a macro/image is not defined in the currently set page [1] to [15], this macro/picture is automatically taken from page [0]. Thus, not all macros and images have to be stored separately for each language when they are identical in each language.

```
PICTURE: 100[0] <BIER.BMP>
PICTURE: 100[1] <BEER.BMP>
PICTURE: 100[2] <BIRRA.BMP>
; make sure to select MacroPage first (#MK n1)
                       ; SAME AS "MACRO: 2"
MACRO: 2[0]
       #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
                         ; ITALIAN
MACRO: 2[2]
        #FZ 3,1
        #ZL 25,0 "ITALIAN "
        #UI 0,20, 100
```

If there is only the need of handling strings differently, stringtables might be used. Depending on the active macropage (ESC M K n1) the desired string is called. Please check the following example:

```
STRINGCODE=$01

STRING: 100[0] "HALLO WELT "; store strings
STRING: 100[1] "HELLO WORLD "; into display's
STRING: 100[2] "CIAO A TUTTI "; memory

; make sure to select MacroPage first (#MK n1)

MACRO: 1

#ST StringCode

#ZL 10,5, StringCode, 100
```



SOUNDS

The EA eDIPTFT70-A(TP) has a sound-module. There are two groups of sounds, notes and jingles. Notes:

The command #YT "notenstr" plays notes, to play individual startmelodies or alerts.

Tone (C'-H')	f (Hz)	Tone (C"-H")	f (Hz)
С	261.6	С	523.3
#C (C sharp/ D flat)	277.2	#c (c sharp/ d flat)	554.4
D	293.7	d	587.3
#D (D sharp/ E flat)	311.1	#d (d sharp/ e flat)	622.3
E	329.6	е	659.3
F	349.2	f	698.5
#F (F sharp/ G flat)	370.0	#f (f sharp/ g flat)	740.0
G	392.0	g	784.0
#G (G sharp/ A flat)	415.3	#g (g sharp/ a flat)	830.6
А	440.0	а	880.0
#A (A sharp/B flat)	466.2	#a (a sharp/b flat)	932.3
Н	493.9	h	987.8

Jingles:

Above all, this group is applicable to support touch functions acoustically. The command #A- n1, n2, n3 applies to every different touch function another jingle with volume (e.g.: #A- 1, 10, 7 = touch button is assigned with jingle 10 volume 7). The jingles are orgnized in subgroubs for an easy selection.

Sound-Number	Jingle-Group	Example
1 - 25	toneless klick	Button, Bar, Instrument
26 - 49	tone klick	Button, Menu, Editbox
50-59	double klick	Button, Switch, Menu
60-80	melody	Alarm, Warning, Info

Defai	ılt	Val	ues:

Touchtype	Sound-Number	Volume	
1=Key	19	7	
2=Switch	50	7	
3=Bargraph	23	6	
4=Instrument	24	4	
5=Draw	23	6	
6=Free area	23	6	
7=Make menu	43	7	
8=Menu Box	14	6	
9=Keyboard	9	5	
10=Edit Box	51	6	

HARDWARE

A Speaker can be directly connected to the pins 3+4. The integrated amplifier reaches 1W in the maximum, driving 8 Ohm speakers.

For print mounting the following type is a possibilty: *) Spitzenwert, falls ein Sound abgespielt wird. "EKULIT AP-30P"



KEYBOARD AND EDIT-BOXES

The EA eDIPTFT70-ATP is able to handle the input of strings itself. You can show a keyboard, which displays in an edit box the user's input data. The example shown in the figure below illustrates the situation. Not till pressing return the data is transferred to the send buffer and transmitted to the host system.



```
- Place Editbox -
                     ;select font
#EF SWISS30B
#FQ BLUE, WHITE, BLUE; set frame colors
#FH WHITE, BLUE, BLUE, YELLOW ; set text colors
                     ; define frame
#EE 20
                      ; define offset between frame and text
#EO 10,10,10,10
#EL 1, 0, 30,140,770,210, "" ; define left justified editbox
#EA 1
                     ; activate editbox

    Place Keyboard -

#FK BLUE, WHITE, BLUE, YELLOW, BLUE, YELLOW
                                          ; keyboard frame colors
#FS RED, WHITE, RED, YELLOW, BLUE, YELLOW
                                          ; special keys for keyboard frame colors
#FF WHITE, BLUE
                                          ; keyboard text color
#FY BLACK, BLUE
                                          ; special key text color
#KF SWISS30B, CHICAGO14, SWISS30B, CHICAGO14; select fonts for keyboard
#KE 6,6
                                         ; frame
#KL 6, "CAPS"
                                          ; define labels for special keys
#KL 8, "<-"
#KL 12, "CLEAR"
#KL 13, "Return"
#KL 5, "Shift"
#KL ' ', "Space"
; define two keyboards, for capital and small letters '\' means special command
; '|' means next line of keyboard
#KB 1, "1234567890ß\8\C|\6qwertzuiopü+\D\D|\5asdfghjklöä#\N|<yxcvbnm,.-|
#KB 2, "°!"'"§$%&/()=?\8\C|\6QWERTZUIOPÜ*\D\D|\5ASDFGHJKLÖÄ\N'"'|<YXCVBNM;:_|
\#KP 0,240,800,480,3 ; set keyboard position
#KS 1,1
                       ; show keyboard
```



Page 34

SPEZIFICATION AND CHARACTERISTICS

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
B : 1 : (1 ::)	w./o. Touch		480		cd/m²
Brightness (white)	with Touch		400		cd/m²
Output Current				20	mA
Dower Cumby	Backlight 100%		690	1000*	mA
Power Supply	Backlight off		200	500*	mA

*) Peak, if a sound is played with full volume.

NOTES ON HANDLING AND OPERATION

- The module can be destroyed by polarity reversal or overvoltage of the power supply; overvoltage, reverse polarity or static discharge at the inputs; or short-circuiting of the outputs.
- It is essential that the power supply is switched off before the module is disconnected. All inputs must also be deenergized.
- The display and touch screen are made of plastic and must not come into contact with hard objects. The surfaces can be cleaned using a soft cloth without solvents.
- The module is designed exclusively for use in buildings. Additional measures have to be taken if it is to be used outdoors. The maximum temperature range of -20 to +70°C must not be exceeded. If used in a damp environment, the module may malfunction or fail. The display must be protected from direct sunshine.





MOUNTING BEZEL EA 0FP801-70SW

As accessory we deliver an optional black anodized mounting bezel. The mounting clips are included in the supplied EA eDIPTFT70-A(TP).

