OPERATING UNIT

160x104 with touch panel

EA eDIP160W-7LWTP





Dimension: 81.5x67.5x13.6mm

TECHNICAL DATA

EA eDIP160B-7LWTP

- * LCD-GRAPHIC DISPLAY WITH A RANGE OF GRAPHIC FUNCTIONS
- * 3 DIFFERENT INTERFACES ONBOARD: RS-232, I²C-BUS OR SPI-BUS
- * 160x104OR104X160DOTSWITHLEDBACKLIGHT
- * WHITE LED-BACKLIGHT BLUE NEGATIVE OR
- * BLACK&WHITE POSITIVE, FSTN-TECHNOLOGY
- * 8 BUILT-IN FONTS
- * FONT ZOOM FROM 2MMTO ABOUT 80MM, TURNABLE IN 90° STEPS
- * POWER SUPPLY WIDE RANGE +3.3V / 190mA/12mA ... +5V / 125mA / 20mA (WITH/ WITHOUT BACKLIGHT)
- * POWER-DOWN-MODE 25 µA, WITHWAKEUPVIATOUCH OR I2C
- * POSITIONING ACCURATETOTHE PIXELWITH ALL FUNCTIONS
- * LINE, DOT, AREA, AND/OR/EXOR, BARGRAPH...
- * CLIPBOARD FUNCTIONEN, PULL-DOWN MENÜ
- * UPTO 256 PICTURES INTERNALY STORED
- * UPTO 256 MACROS PROGRAMMABLE (64kB EEPROM ONBOARD)
- * MIXTEXT AND GRAPHIC, FLASHING ATTRIBUTE: ON/OFF/INVERT
- * BACKLIGHT BRIGHTNESS PER SOFTWARE
- * ANALOGUETOUCH PANEL: VARIABLE GRID
- * FREE DEFINABLE KEY AND SWITCH

ORDERING CODES

DISPLAYS

160x104 DOTS, WHITE LED-BACKLIGHT, BLUENEGATIVE EA eDIP160B-7LW AS ABOVE, BUTWITHTOUCH PANEL 160x104 DOTS, WHITE LED-BACKLIGHT, POSITIVE MODE, FSTN AS ABOVE, BUTWITHTOUCH PANEL

STARTERKIT

INCLUDES EAeDIP160B-7LWPAND EVALUATION BOARDWITH USB FOR DIRECT CONNECTIONTO PC AND INTERFACE BOARDS FOR

CONNECTIONWITHYOUR HOST SYSTEM AS ABOVE, BUTWITH EAeDIP160J-7LWTP

ADDTIOTNAL PARTS

MOUNTING BEZEL (ALUMINIUM), BLACK ANODIZED

SOCKET 1x20, 4.5mm HIGH (1 piece)

EA eDIP160B-7LWTP EA eDIP160J-7LW EA eDIP160J-7LWTP

EA EVALeDIP128B EA EVALeDIP128J

EA 0FP161-7SW EAB254-20



			Documentation of revision	
Date	Туре	Old	New	Reason / Description
October, 2010	0.1			preliminary Version
August, 2011	1.0			first release

CONTENTS

GENERAL	3
RS-232	4
RS-485, USB	5
SPI	6
I ² C	7
IN- AND OUTPUTS	8
ROTATED MOUNTING	9
POWER-DOWN-MODE	9
SOFTWARE PROTOCOL	10 - 11
TERMINAL MODE, FILL PATTERN	12
COMMANDS/FUNCTIONS INTABULAR FORMAT	13 - 17
TOUCHPANEL, KEYSTYLE	16 - 17
RESPONSES OF THE OPERATING PANEL	18
CHARACTER SETS	19 - 20
FLASH-AND GRAYSCALE MODE	21
MACRO PROGRAMMING	24 - 25
ELECTRICAL CHARACTERISTICS	26
MOUNTING BEZEL	27
DIMENSION	28



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.

The displays are ready for operation immediately with an operating voltage range of +3.3V..+5V. 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 range of +3.3V..+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, B- ANDW-TYPES

All displays in blue-and-white (B) and black-and-white (W) are equipped with a modern, low power consumption LED backlight. Whereas the black&white can still be read even when the backlight is switched off completely, the blue-white display requires a minimum level of illumination to be legible. The backlight can be switched off with a software command and the brightness can be adjusted. We recommend the black&white version for use in direct sunlight. For all other applications, we recommend the high-contrast, blue-white version. Note that the white LED backlight is subject to aging. That means switching off or dimming backlight is a must for 24-hour-applications.

SOFTWARE

This display is programmed by means of commands, such as draw a rectangle from (0,0) to (64,15). 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 (Programmer) for 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 9777-2USB, which is available as an accessory, will burn the data/images you have created into the on-board EEPROM (64 kB) 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 (included in the Starter-Kit):

Wtih 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.



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 (VDD) to a microcontroller, for example, for direct connection.

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

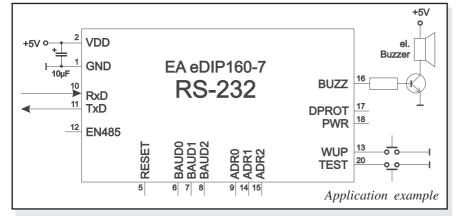
			Pinout eDIP160-7: RS	3-2	32/R	S-485 mode)	
Pin	Symbol	In/Out	Function		Pin	Symbol	In/Out	Function
1	GND		Ground Potential for logic (0V)		21	GND		Ground (0V)
2	VDD		Power supply for logic (+3,3V5V)		22	VDD		Power supply (+3,35V)
3	NC		do not connect		23	NC		do not connect
4	NC		do not connect		24	NC	1	do not connect
5	RESET	In	L: Reset		25	IN8 / OUT1		
6	BAUD0	In	Baud Rate 0		26	IN7 / OUT2		8 digital inputs
7	BAUD1	In	Baud Rate 1		27	IN6 / OUT3		(internal 20k50k pullup
8	BAUD2	In	Baud Rate 2		28	IN5 / OUT4		- la ali
9	ADR0	In	Address 0 for RS-485		29	IN4 / OUT5		alternativ up to 8 digital outputs
10	RxD	In	Receive Data		30	IN3 / OUT6		maximum current:
11	TxD	Out	Transmit Data		31	IN2 / OUT7		IOL = IOH = 10mA
12	EN485	Out	Transmit Enable for RS-485 driver		32	IN1 / OUT8		
13	WUP	In	L: (Power-On) disable Power-On-Macro L: Wakeup from Powerdownmode		33	NC		
14	ADR1	In	Address 1 for RS-485		34	NC		
15	ADR2	In	Address 2 for RS-485		35	NC		
16	BUZZ	Out	H: Buzzer output (L: Buzzer off)		36	NC	1	
17	DPROT	In	L: Disable Smallprotokoll do not connect for normal operation		37	NC		do not connect
18	PWR	Out	L: Normal Operation H: Powerdownmode		38	NC		
19	NC	-	do not connect		39	NC	1	
20	TEST SBUF		open-drain with internal pullup 20k50k IN (Power-On) L: Testmode OUT L: data in sendbuffer		40	NC		

BAUDRATES

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).

	Baud rates									
Baud0	Baud1	Baud2	Data format 8,N,1							
0	0	0	1200							
1	0	0	2400							
0	1	0	4800							
1	1	0	9600							
0	0	1	19200							
1	0	1	38400							
0	1	1	57600							
1	1	1	115200							



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

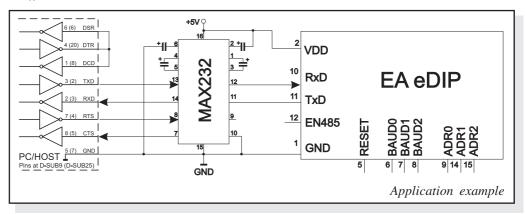
with a low level that data is ready to be retrieved from the internal send buffer. The line can be connected to an interrupt input of the host system, for example.



APPLICATION EXAMPLE "REAL" RS-232 INTERFACE

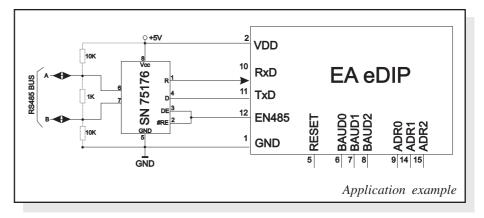
The eDIP fits for direct connection to a RS-232 interface with CMOS level (VDD).

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.

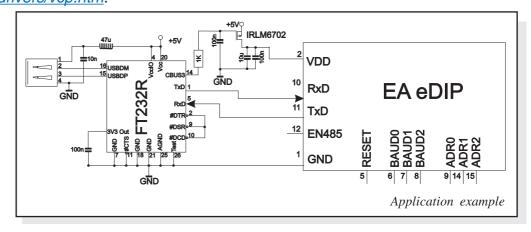


Adressing:

- 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.





SPIINTERFACE

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.

			Pinout eDIP160-	-7:	SPI	mode	
Pin	Symbol	In/Out	Function		Pin	Symbol	In/Out Function
1	GND		Ground Potential for logic (0V)		21	GND	Ground (0V)
2	VDD		Power supply for logic (+3,3V5V)	1	22	VDD	Power supply (+3,35V)
3	NC		do not connect	1	23	NC	do not connect
4	NC		do not connect	1	24	NC	do not connect
5	RESET	In	L: Reset		25	IN8 / OUT1	
6	SS	In	Slave Select		26	IN7 / OUT2	8 digital inputs
7	MOSI	In	Serial In		27	IN6 / OUT3	(internal 20k50k pullup)
8	MISO	Out	Serial Out		28	IN5 / OUT4	
9	CLK	In	Shift Clock		29	IN4 / OUT5	alternativ up to 8 digital outputs
10	DORD	In	Data Order (0=MSB first; 1=LSB first)	1	30	IN3 / OUT6	maximum current:
11	SPIMOD	In	connect to GND for SPI interface	1	31	IN2 / OUT7	IOL = IOH = 10mA
12	NC		do not connect		32	IN1 / OUT8	
13	WUP	In	L: (Power-On) disable Power-On-Macro L: Wakeup from Powerdownmode		33	NC	do not connect
14	CPOL	In	Clock Polarity (0=LO 1=HI when idle)		34	NC	do not connect
15	CPHA	In	Clock Phase sample 0=1st;1=2nd edge	1	35	NC	do not connect
16	BUZZ	Out	H: Buzzer output (L: Buzzer off)	1	36	NC	do not connect
17	DPROT	In	L: Disable Smallprotokoll do not connect for normal operation		37	NC	do not connect
18	PWR	Out	L: Normal Operation H: Powerdownmode		38	NC	do not connect
19	NC	•	do not connect		39	NC	do not connect
20	TEST SBUF	Out	open-drain with internal pullup 20k50k IN (Power-On) L: Testmode OUT L: data in sendbuffer		40	NC	do not connect

Note:

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 hi level.

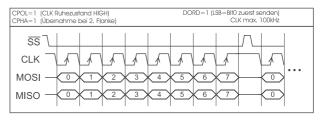
On pin 20 (SBUF) the display indicates with a low level that data is ready to be retrieved from the internal send buffer. The line can be connected to an interrupt input of the host system, for example.

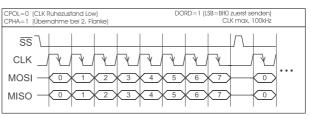
DATATRANSFERSPI

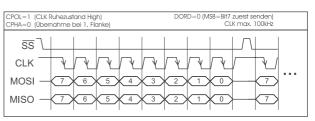
<u>Write operation:</u> a clock rate up to 100 kHz is allowed without any stop. Together with a pause of 100 μ s between every data byte a clock rate up to 3 MHz can be reached.

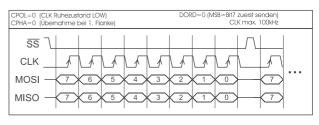
Read operation: to read data (e.g. the "ACK" byte) a dummy byte (e.g. 0xFF) need to be sent.

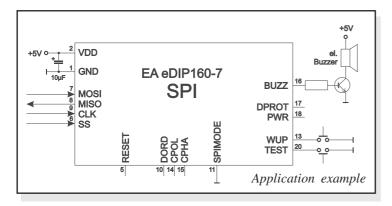
Note that the EA eDIP for internal operation does need a short time before providing the data; therefore a short pause of min. $6\mu s$ (no activity of CLK line) is needed for each byte.













I²C-BUSINTERFACE

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.

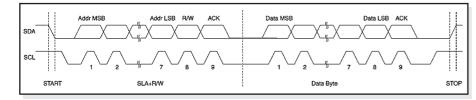
Data transfer is possible at up to 100 kHz. However, if pauses of at least 100 µs are maintained between the individual bytes during transfer, a byte can be transferred at up to 400 kHz.

			Pinout eDIP160-	7: I2C	mode		
Pin	Symbol	In/Out	Function	Pin	Symbol	In/Out	Function
1	GND		Ground Potential for logic (0V)	21	GND		Ground (0V)
2	VDD		Power supply for logic (+5V)	22	VDD		Power supply (+3,35V)
3	NC		do not connect	23	NC		do not connect
4	NC		do not connect	24	NC		do not connect
5	RESET	In	L: Reset	25	IN8 / OUT1		
6	BA0	In	Basic Address 0	26	IN7 / OUT2		8 digital inputs
7	BA1	In	Basic Address 1	27	IN6 / OUT3		(internal 20k50k pullup)
8	SA0	In	Slave Address 0	28	IN5 / OUT4		altamative un ta O dinital
9	SA1	In	Slave Address 1	29	IN4 / OUT5		alternativ up to 8 digital outputs
10	SA2	In	Slave Address 2	30	IN3 / OUT6		maximum current:
11	BA2	In	Basic Address 2	31	IN2 / OUT7		IOL = IOH = 10mA
12	I2CMOD	In	connect to GND for I ² C interface	32	IN1 / OUT8		
13	WUP	In	L: (Power-On) disable Power-On-Macro L: Wakeup from Powerdownmode	33	NC		
14	SDA	Bidir.	Serial Data Line	34	NC		
15	SCL	In	Serial Clock Line	35	NC		
16	BUZZ	Out	H: Buzzer output (L: Buzzer off)	36	NC		
17	DPROT	In	L: Disable Smallprotokoll do not connect for normal operation	37	NC		do not connect
18	PWR	Out	L: Normal Operation H: Powerdownmode	38	NC		
19	NC		do not connect	39	NC		
20	TEST SBUF	IN Out	open-drain with internal pullup 20k50k IN (Power-On) L: Testmode OUT L: data in sendbuffer	40	NC		

Note:

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 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	- A	ddı	res	s					
Pi	n 11,7	7,6	Base				 2(C ad	dres	ss		
BA2	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
Н	L	L	\$70		0	1	1	1	A 2	A 1	A 0	W
Н	L	Н	\$90		1	0	0	1	_	'	U	
Н	Н	L	\$B0		1	0	1	1				
Н	Н	Н	\$D0		1	1	0	1				

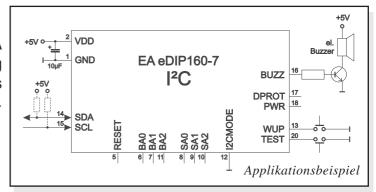
all pins open: Write \$DE Read \$DF

DATATRANSFER 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).





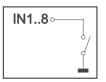
Page 8

IN-AND OUTPUTS

The eDIP160-7 has 8 digital in- or outputs (CMOS level, grounded). They can be redefined freely.

Inputs

As status on delivery, all ports are defined as inputs. Each input provides an internal $20..50 \, k\Omega$ 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.

24). 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. 18).

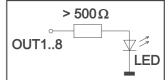
Note: The logic circuitry is designed for slow operations; in other words, more than 3 changes per second cannot be easily executed.

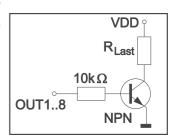
Outputs

The command "ESC Y M number" redefines one or several inputs as outputs. In this case the more significant inputs are used as outputs.

'ESC Y M 3' switches IN8, IN7 and IN6 as outputs OUT1, OUT2 and OUT3 for example.

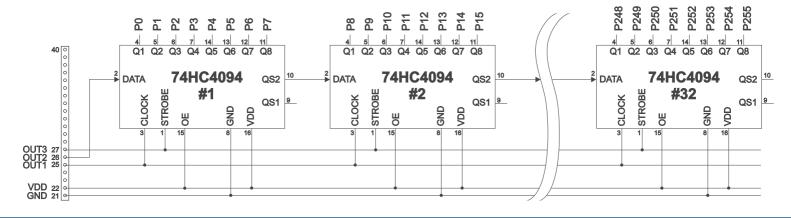
Each line can be controlled individually using the "ESC Y W" command. A maximum current of 10mA can be switched per line. This give the opportunity to drive a low power LED in direct way. To source higher current please use an external transistor.





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. 16) provides a comfortable way to control the outputs.





TOPVIEW ANDTWISTED MOUNTING

The prefered view of the eDIP160 is bottom view, (6 o'clock).

The eDIP can be mounted turned around 180° to gain a top view display (12 o'clock). To set the viewing direction you have to run (e.g. in PowerOnMacro) the command 'ESC DO 2' (refer to p. 13). In addition it is possible to mount the display turned with 90° or 270° to gain a portrait mode display with 104x160 pixels.

0°: 'ESC DO 0'



90°: 'ESC DO 1'



180°: 'ESC DO 2'



270°: 'ESC DO 3'



POWER DOWN MODE

To save energy (battery operation), you can activate one of three power-down modes by means of the command 'ESC PD n1' (see page 15 below).

Mode 0 (25µA): The LED illumination is switched off, and the contents of the display become invisible although they are still there. In power-down mode including suppressor diodes, the eDIP160 requires up to 1000 µA (delivery state). The suppressor diodes can be deactivated by removing the two 0Ω resistors. Then powerdown current of typically 25 μA is reached. They are labeled with R_{pd} .

Important: When deactivating the suppressor diodes, it is essential that the polarity of the display is correct all the time: GND, VDD (pin 1 + 2). Even very brief polarity reversal or overvoltage can damage the display immediately and irreparably.

Mode 1 (1mA): The LED illumination is switched off, the contents of the display stay visible. Current consumption is reducing to 1mA. This power down mode is mainly usable with the versions EA eDIP160W with positive display, because they are readable without backlight.

Mode 2 (2mA): The LED illumination stays on and the display content is readable. The current consumption reduces to 2-3mA plus adjusted LED current. Therefore you can use the eDIP in dark surroundings and dimmed illumination under e.g. 10mA.

The eDIP160 can be woken up from power down mode with a low level on pin 13 (WUP), or the adressing via I²C.

In addition the eDIP160 can be woke up by using the touchpanel (independed from position). After wake up, special WakeUpMacros can be used (refer to p. 24).



DATATRANSFER PROTOCOL(SMALL PROTOCOL)

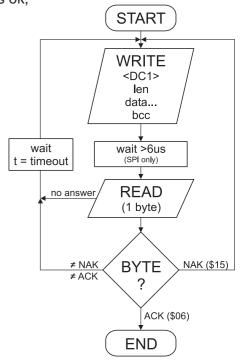
The protocol has an identical structure for all 3 interface types: RS-232, SPI and I²C. Each data transfer is embedded in a fixed frame with a checksum (protocol package). The EA eDIP160-7 acknowledges this package with the character <ACK> (=\$06) on successful receipt or <NAK> (=\$15) in the event of 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 elapsed 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 split up between a number of packages. All data in the packages are compiled again after being correctly received by the EA eDIP.

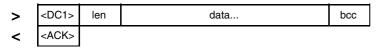
DEACTIVATINGTHE SMALL PROTOCOL

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.



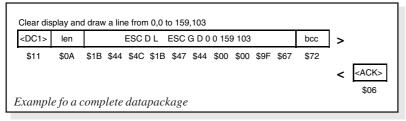
BUILDINGTHE SMALL PROTOCOL PACKAGES

Command/data to the display



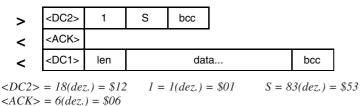
< DC1 > = 17(dez.) = \$11< ACK > = 6(dez.) = \$06

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



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

Request for content of send buffer



len = count of user data (without <DCI>, without checksum bcc) bcc = 1 byte = sum of all bytes incl. <DCI> 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 begins to send all the collected data such as touch keystrokes.



Request for buffer information

>	<dc2></dc2>	1	I	bcc

< <ACK>
<DC2> 2 send buffer bytes ready bytes free bcc

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

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 and how full the display's receive buffer is.

Protocol settings

< 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.

This command is used to query protocol

settings.

Request for protocol settings

>	<dc2></dc2>	1	Р	bcc

< <ACK>

< L	V/ (O/ (>					
<	<dc2></dc2>	3	max. packet size	akt. send packet size	akt. timeout	bcc

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

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

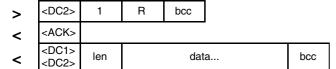
max. packet size = count of maximum user data for 1 package (eDIP160-7 = 255)

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



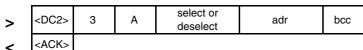
$$< DC2 > = 18(dez.) = $12$$
 $I = I(dez.) = 01 $R = 82(dez.) = 52 $< ACK > = 6(dez.) = 06

< 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

Adressing (only for RS232/RS485)



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

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>).

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



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 (pages 10 and 11) or a deactivated protocol.

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: '##'.

+ Lower Upper	\$0 (0)	\$1 (1)	\$2 (2)	\$3 (3)	\$4 (4)	\$5 (5)	\$6 (6)	\$7 (7)	\$8	\$9 (9)	\$A (10)	\$B (11)	\$C (12)	\$D (13)	\$E (14)	\$F (15)
\$00 (dez: 0)	0	企	Φ	♦	Ф	/	1	#	+	H	LF	Ŧ	FF	$c_{\rm R}$	So	SI
\$10 (dez: 16)	0	:	2	3	Ч	5	5	7	8	9	0	Es	ተ	ψ	÷	+
\$20 (dez: 32)		į	••	#	\$	×.	&	•	C)	¥	+	,	-		/
\$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$40 (dez: 64)	œ	A	В	C	D	E	F	G	Н	Ι	J	K	L	M	N	0
\$50 (dez: 80)	P	Q	R	S	T	u	Ų	М	Х	Y	Z	Г	N]	^	_
\$60 (dez: 96)	~	а	Ь	C	d	e	f	g	h	i	j	k	1	M	n	o
\$70 (dez: 112)	P	q	r	5	t	u	V	W	x	y	z	€	I	>	~	Δ
\$80 (dez: 128)	ç	ü	é	3	ä	à	÷	ç	ê	ë	è	ï	î	ì	Ä	À
\$90 (dez: 144)	É	æ	Æ	ô	ö	ò	û	ù	ÿ	ö	ü	¢	£	¥	β	f
\$A0 (dez: 160)	á	í	ó	ú	ñ	ñ	a	ō	ż	г	٦	1/2	14	i	≪	≫
\$B0 (dez: 176)	::	:::	***		4	7	11	П	7	1	Ш	╗	4	ш	3	7
\$C0 (dez: 192)	L	_	T	F	_	+	F	Iŀ	Ľ	F	4	77	IÈ	=	#	=
\$D0 (dez: 208)	щ	=	π	Ц	E	F	П	#	#	_	г					
\$E0 (dez: 224)	CX	β	Г	Π	Σ	σ	Щ	Ţ	δ	Θ	Ω	6	ø	Φ	E	N
\$F0 (dez: 240)	=	±	<u>></u>	<	ſ	J	÷	\approx	•	-		√	n	2	3	_

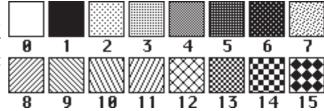
Terminal-Font (Font 0): 8x8 monospaced

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

If the graphics screen is deleted with 'ESC DL', for example, that does not affect the contents of the terminal window. The terminal font is fixed in the ROM and can also be used for graphic outputs 'ESC Z...' (set FONT nr=0).

FILL PATTERN

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. There are 16 internal fill patterns available.



USINGTHE 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 follow 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.: #GD0,0,159,103,
- Strings (text) are written directly without quotation marks and concluded with CR (hex: \$0D) or LF (hex: \$0A).

2. Binary 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.

						Т	erminal commands	After
Command	Cod	es					Remarks	reset
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 ـ	S				The current cursor position is saved	
Restore cursor position	ESC	'	R	3			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	F00		٧				The version no. is output in the terminal (e.g. "EA eDIP160-7 V1.0 Rev.A")	
Output project name	ESC	۱.	J				The macro project name is output to the terminal (e.g. "init / delivery state")	
Output information	ESC		ı				The terminal is initialized and deleted; software version, hardware revision, the macro project name and the CRC-checksum is ouput to the terminal	

				Dis	play commands (effect the entire display)		after
Command	Cod	es			Remarks		reset
Set display orientation			0	n1	n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°; (0°+180°=160x104; 90°+	270°=104x160)	0°
Set display contrast	ESC	D	K	n1	n1=040: Set display contrast to n1 (default = 20) n1='+': increase contrast; n1='-': decrease contrast		20
Set grayscale mode			G	n1	n1=0: grayscale mode ON; flashing is possible n1=1: grayscale mode OFF; flahing not possible		0
Delete display			L		Delete display contents (all pixels off)		
Invert display			-		Invert display contents (invert all pixels)		
Fill display			S		Fill display contents (all pixels on)		
Switch display off	ESC	D	Α		Display contents become invisible but are retained, commands are still	possible	
Switch display on	150	U	Е		Display contents become visible again		οN
Show clipboard			С		Show content of clipboard; Standard display outputs are no longer visil	ole	
Show normal display content			N		Normal operation, standard display outputs are visible		

	Clipboard commands (Buffer for display area)											
Command	Cod	es							Remarks	reset		
Save display contents			В						The entire contents of the display are copied to the clipboard as an image area			
Save area	ESC		S	x1	y1	x2	y2		The image area from x1,y1 to x2,y2 is copied to the clipboard			
Restore area	ESC	٦	R						The image area on the clipboard is copied back to the display			
Copy area			K	x1	y1				The image area on the clipboard is copied to x1,y1 in the display			

							Stra	igl	nt lines and points	after
Command	Cod	les							Remarks	reset
									Settings	
Point size / line thickness	ESC	G	Z	n1	n2				n1 = x-point size (115); n2 = y-point size (115);	1,1
Link mode	ESC	G	٧	n1					Set drawing mode n1: 1=set; 2=delete; 3=inverse	1
Blink attribute	ESC	G	В	4					n1:0=no blink; 1=on/off; 2=blink inverted; 3=off/on (phase shifted)	0
Gray-scale	ESC	G	В	n1					n1:0=black solid line; 1=dark gray; 3=light gray (refer to command ESC DG)	U
									Draw lines and points	
Draw point			Р	x1	y1				Set a point at coordinates x1, y1	
Draw straight line	ESC		D	x1	у1	x2	y2		Draw a straight line from x1,y1 to x2,y2	
Continue straight line	ESC	G	W	x1	y1				Draw a straight line from the last end point to x1,y1	0, 0
Draw rectangle			R	x1	y1	x2	y2		Draw four straight lines as a rectange from x1,y1 to x2,y2	

Change / draw rectangular areas												
Command	Coc	les							Remarks	reset		
Delete area			L	x1	y1	x2	y2		Delete an area from x1,y1 to x2,y2 (all pixels off)			
Invert area			-	x1	y1	x2	y2		Invert an area from x1,y1 to x2,y2 (invert all pixels)			
Fill area			S	x1	y1	x2	y2		Fill an area from x1,y1 to x2,y2 (all pixels on)			
Area with fill pattern	ESC	R	М	x1	y1	x2	y2	n1	Fill an area from x1,y1 to x2,y2 with pattern n1 (always set)			
Draw box			0	x1	y1	x2	y2	n1	Draw rectangle from x1,y1 to x2,y2 with pattern n1 (always replace)			
Draw frame			R	x1	у1	x2	у2	n1	Draw frame of type n1 from x1,y1 to x2,y2 (always set)			
Draw frame box			Т	x1	y1	x2	y2	n1	Draw frame box of type n1 from x1,y1 to x2,y2 (always replace)			



EAeDIP160-7

Page 14

							Т	ext commands	after
Befehl	Cod	es						Remarks	reset
								Settings	
Set font			F	n1				Set font with the number n1 = 015	0
Set font zoom factor			Z	n1	n2			n1 = x-zoom factor (1x4x); $n2 = y$ -zoom factor (1x4x)	1,1
Additonal line spacing	ESC	7	Υ	n1				Insert n1 = 015 dots between two lines as additional spacing	0
Spacwidth	ESC		J	n1				Spacewidth: n1=0 use from font; n1=1 same width as number; n1>=2 width in dot	0
Text angle			w	n1				Text angle: n1=0: 0°; n1=1: 90°;	0
Text link mode			٧	n1				Mode n1: 1=set; 2=delete; 3=inverse; 4=replace; 5=inverse replace	4
Text flashing attribute	ESC	7	В	n1				n1:0=no flashing; 1=on/off; 2=flash inversly; 3=off/on (phase shifted)	
Gray scale	ESC	2	В	nı				n1:0=black; 1=dark gray; 3=light gray (refer to command: ESC DG)	0
								Output strings	
Output string			L					A string is output to x1,y1; string termination is: 'NUL' (\$00), 'LF' (\$0A) or 'CR' (\$0D); several lines are seperated by the character (\$7C);	
L: left justified C: centered	ESC	z	С	x1	у1	Text 	NUL	Text between two '~' (\$7E): characters flashes on/off; Text between two '&' (\$26): characters flashes phase shifted; Text between two '@' (\$40): Zeichen stehen blinken Invertierend;	
R: right justified			R					The character '\' (\$5C, backslash) cancels the special funtion of characters ' ~@\'; e.g. "name\@test.de" => "name@test.de"	
String for terminal	ESC	Z	Т		-	Text		Command to output a string (text) from a macro to the terminal	

							Bitr	nap commands	after
Command	Cod	les						Remarks	reset
								Settings	•
Image zoom factor			Z	n1	n2			n1 = x-zoom factor(1x4x); $n2 = y$ -zoom factor (1x4x)	1,1
lmage angle	ESC	U	W	n1				lmage angle: n1=0: 0°; n1=1: 90°	0
Image link mode			٧	n1				Mode n1: 1=set; 2=delete; 3=inverse; 4=replace; 5=inverse replace;	4
Image flashing attribute	ESC	7	1	1				n1:0=no flashing; 1=on/off; 2=flash inverted; 3=off/on (phase shifted)	
Image gray scale	ESC	Z	В	n1				n1:0=black; 1=dark gray; 3=light gray (refer to command ESC DG)	0
								Output	
Image from clipboard			С	x1	у1			The current contents of the clipboard are loaded to x1,y1 with all the image attributes	
Load internal image	ESC	U	- 1	x1	y1	nr		Load internal image with nr (0255) from EEPROM to x1,y1	
Load image			L	x1	у1	BLH	data	Load an image to x1,y1; data = image in BLH-format	
								Hardcopy	
Send hardcopy	ESC	U	н	x1	у1	x2	y2	An image area x1,y1 to x2,y2 is put into the sendbuffer. The image is send in BLH-format	

							В	arg	rapl	n comn	nan	ds	after
Commands	Cod	es							Ren	narks			reset
Definition													
Define bargraph	ESC	В	R L O U	n1	x1	y1	x2	y2	aw	ew type	pat	Define bargraph with number n1 to l(eft), r(ight), o(up), u(down). x1,y1,x2, are the surrounding rectangle of the bar. aw,ew are the values fo 0% and 100%. type=0:pattern bar; type=1:pattern bar in rectangle; pat=bar pattern type=2:line bar; type=3:line bar in rectangle; pat=line width	y2 no bar defined
Delete bargraph	ESC	В	D	n1	n2				touch	field will a	lso be	raph n1 becomes invalid. If the bargraph was defined as an input by touch, e deleted ins visible; n2=1: Bargraph is deleted	, the
										Use			
Update bargraph			Α	n1	val				Set a	nd draw the	e bar	graph n1 to new val(ue).	
Redraw bargraph	ESC	В	Z	n1					Entire	ely redraw t	he ba	argraph n1.	
Send bargraph value			s	n1					Send	the curren	t valu	e of the bargraph number n1	

							F	las	hing commands	after
Command	Cod	es							Remarks	reset
									Settings	
Set flashing time	ESC	ø	Z	n1					Set flashing time to n1= 115 in 1/10s; 0=flashing deactivated	6
									Flashing areas	
Delete flashing attribute			L	x1	у1	x2	y2		Delete the flashing attribute from x1,y1 to x2,y2. Do not use this command for phase shifted areas! (Copies the area from grafiklayer to blinklayer)	
Flashing inversely	ESC	Q	I	x1	у1	x2	у2		Define an inverted flashing area from x1,y1 to x2,y2. (Copies the inverted area from grafiklayer to blinklayer)	
Flashing area pattern			М	x1	у1	x2	у2	n1	Define a flashing area (on/off) with pattern n1 frim x1,y1 to x2,y2 (Draw the pattern into blinklayer)	
									Phase shifted areas	
Restore phase shifted area			R	x1	у1	x2	у2		Delete the phase shifted flashing area from x1,y1 to x2,y2. Do not use this command for other flashing attributes! (Copues the area from blinklayer to graphiklayer)	
Inverted phase shifted area	ESC	Q	Е	x1	у1	x2	y2		Define a ohase shifted inverted flashing area frim x1,y1 to x2,y2. (Copies the inverted are from blinklayer to graphiklayer)	
Phase shifted flashing pattern			Р	x1	у1	x2	y2	n1	Define a phase shifted flashing area (off/on) with pattern n1 from x1,y1 to x2,y2. (Draw the pattern into grafiklayer)	



								Me	nu commands	after
Command	Cod	les							Remarks	reset
	•						S	ettir	gs for menu box/touch menu	•
Set menu font			F	n1					Set font with the number n1 (0 to 15) for menu display	0
Menu font zoom factor			Z	n1	n2				n1 = x-zoom factor $(1x4x)$; $n2 = y$ -zoom factor $(1x4x)$	1,1
Additional line spacing			Υ	n1					Insert n1 (015) dots between two menu items as additional spacing	
Menu angle	ESC	N	W	n1					Menu display angle: n1=0: 0°; n1=1: 90°;	0
Touch menu automation			т	n1					n1=1: Touchmenu opens autmatically; n1=0: Touchmenu doesn't open automatically, instead the rquest 'ESC T0' is sent to the host, which can then open the touch menu with 'ESC NT2'	1
						N	lenu	box	commands (control not by touch)	
Define and display menu			D	x1	y1	nr	Text 	NUL	A menu is drawn at the corner x1,y1 witch the current font of menu nr:= currently inverted entry (e.g. 1 = first entry) Text:= string with menu items. The different items are seperated by the character " ' (\$7C,dez:124) e.g. "Item1 Item2 Item3" The background of the menu is automatically saved into the clipboard (previous contend will be overwritten). If a menu is already defined, it is automatically canceled and deleted	
Next item	ESC	N	N						The next item is inverted or remains at the end	
Previous item			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 sent 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 for item 1, menu macro n1+1 for item 2 and so on	
End of menu/cancel			Α		•				The menu is removed and replaced with the original background	

								Ма	cro commands	after			
Command	Cod	les							Remarks				
									Call macros				
Run normal macro			N	n1					Call the (normal) macro with the number n1 (max. 7 levels)				
Run touch macro			T	n1					Call the touch macro with the number n1 (max. 7 levels)				
Run menu macro	ESC	M	М	n1					Call the menu macro with the number n1 (max. 7 levels)				
Run port macro			Р	n1					Call the port macro with the number n1 (max. 7 levels)				
Run bit macro			В	n1					Call the bit macro with the number n1 (max. 7 levels)				
								aı	itomatic (normal-) macros				
Macro with delay			G	n1	n2				Call the (normal) macro with the number n1 in n2/10s. Execution is stopped by commands (e.g. receipt or touch macros)				
Automatic 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)				
Automatic macros cyclically	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)				
Automatic macros pin pong			J	n1	n2	n3			Automatically run macros n1 to n2 to n1 (ping pong); n3 = pause in 1/10s. Execution is stopped commands (e.g. receipt or touch macros)	by			
									Macro processes				
Define macro process			D	no	type	n3	n4	zs	A macro process with the number no (1 to 4) is defined (1=highest priority). The macros n3 to n4 are run successuvely every zs/10s. Type: 1=once only; 2=cyclical; 3=ping pong n3 to n4 to n3				
Macro process interval	ESC	M	z	no	zs				A new time zs/10s is assigned to the macro process no (1 to4). If the time zs is set to 0, the execution is stopped.				
Stop macro processes			s	n1		•		•	All macro processes are stopped with n1=0 and restarted with n1=1 in order, for example, to execute settings and outputs via the interface undisturbed.	1			

					G	General commands	after
Command	Cod	es				Remarks	reset
						Backlight	
Illumination brightness			Н	n1		Set brightness of the LED illumination to n1=0%100%	100
Brightness changetime	ESC	γ	Z	n1		Time n1=031 in 1/10s for changing brightness from 0% to 100%	5
Illumination on/off	ESC	Y	L	n1		LED illumination n1=0: off; n1=1: on; n1=2 to 255: The illumination is switched on for n1/10s.	1
Save parameter			@		•	Save actual brightness and changetime parameter for power on to EEPROM	
						Send commands	
Send bytes			В	len	data	len (=1 to 255) bytes are sent to the sendbuffer data = data to send. In the source text of the macro programming, the number len must not be specified. This is counted by the ediptft-compiler and entered.	
Send version	ESC	S	٧			The verion is sent as a sting to sendbuffer, e.g. "EA eDIP160-7 V1.0 Rev.A TP+"	
Send projectname			J			The macro project name is sent as a sting to sendbuffer, e.g. "init / delivery"	
Send internal infos			ı			Internal information about the eDIP is sent to the sendbuffer	
						Other commands	
Wait (pause)	ESC	Х	n1			Wait n1/10s before next command is executed	
Set RS485 address	ESC	κ	A	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).	
Tone on/off	ESC	Υ	s	n1		The tone output (pin 16) becomes n1=0:OFF; n1=1:ON; n1=2 to 255:ON for n1/10s	OFF
Power down	ESC	Р	D	n1	n2	After this command, the display goes into power-down mode n1=02 (see page 9). n2=0: no wake-up by touch; n2=1: wake-up by touch possible	



							I/O-Ports	after
Command	Cod	es					Remarks	reset
							Input ports	
Read input port			R	n1			n1=0: Read all input ports as binary value (to sendbuffer) n1=18: Read input port n1 (1=H-level=VDD, 0=L-level=GND)	
Port scan on/off		.,	Α	n1			The automatic scan of the input port is n1=0 deactivated, n1=1 activated	1
Invert input port	ESC	Y	ı	n1			The input port is n1=0 is evaluated normal, n1=evaluated inverted	0
Redefine input bitmacro			D	n1	n2	n3	Input port n1=18 is assigned by falling edge n2=0 to BitMacro n3=0255 Input port n1=18 is assigned by rising edge n2=1 to BitMacro n3=0255	
							Output ports	
Define output port	ESC	٧	М	n1			n1=0: All 8 I/O-Ports are inputs IN1IN8 (=default after Power-On / Reset) n1=18: n1 I/O-lines will be set to output (beginninge at OUT1 upwards)	0
Write output port	ESC	ĭ	w	n1	n2		n1=0: Set all defined output ports in accordance with n2 (=binary value) n1=18: Reset output port n1 (n2=0); set (n2=1); invert (n2=2)	
	•						Port expansion with 74HC4094	•
Write extended ports	ESC	Υ	Е	n1	n2	n3	Set the outputs of the external 74HC4094 (refer to page 8) from port n1=0255 to port n2=0255; n3=0: low; n3=1: high; n3=2: invert	

TOUCH PANEL(ONLY EAeDIP160x-7xxTP)

The -xxxTP versions are shipped with an analog, resistive touch panel. Up to 40 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 (see page 17). 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 (see page 22, Macro programming).

FRAMES AND KEY SHAPES

A frame type can be set by using the Draw frame or Draw frame box command or by drawing touch keys. 18 frame types are available (0 = do not draw a frame). The frame size must be at least 16x16 pixels.

BITMAPS AS KEYS

Apart from the frame types, which are infinitely scalable, it is also possible to use bitmaps (2 each, for not printed and printed) as touch keys or touch switches. You can use ELECTRONIC ASSEMBLY LCD-Tools*) to integrate your own buttons as images ("PICTURE" compiler statement). A button always consists of two monochrome Windows BMPs of equal size (one bitmap to display the touch key in its normal state and one 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 GROUP)

Touch switches (radio buttons) 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 AR nr' command). If a touch switch in the group 'nr' is switched on, all the other touch switches in this group are automatically switched off. Only one switch is ever on.









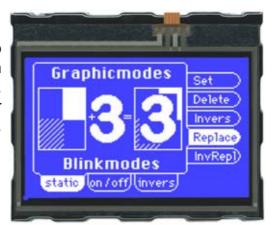














						Co	mm					panel		after
Command	Cod	les							Ren	nark	s			reset
										Se	etting			_
Touch frame form	ESC	Α	Е	n1							•	splay of touch keys/switches is		1
Radio group for switches	ESC	A	R	nr					defin- group	ed sw with	itches	pelong to a group. nr=1 to 255:	e others are deactivated. nr=0: newly newly defined switches belong to the roup, only the down code is applicable.	0
									То	uch:	Lab			
Label font			F	nr								n1 (0 to 31) for touch key labe		0
Label zoom factor	ESC	Α	Z	n1	n2							(1x to 8x); $n2 = Y zoom factor (1x to$		1,1
Add. line spacing			Y	n1							,	between two lines of text as ac	, ,	
Label angle			W	n1						•		0°; n1=1: 90°; n1=2: 180°; n1=	3: 270°	0
	1			1			1		ouch	bere	iche	-	defined as a key. 'U': Image no. n1 is	1
Define touch key (key remains depressed as long as there is contact)	ESC	A	т	x1	y1	x2	у2	dow Cod	up Cod	Text 	NUL	o xx1,yy2 and defined as a key then key pressed. 'up code': (1 d. (down/up code = 0 press/rele er determines the alignment of	r. 'down code':(1-255) Return/touch -255) Return/touch macro when key ease not reported). 'text': the first the text (C=centered, L=left justified,	
,			U	x1	y1	n1	dow Cod	up Cod	Text 	NUL			string that is placed in the key with the eparated with the character ' ' (\$7C,	
Define touch switch (status of the switch toggles after each contact)	ESC	Α	К	x1	y1 y1	x2 n1	y2 dow Cod	dow Cod up Cod	up Cod Text	Text 	NUL	d to xx1,yy2 and defined as a souch macro when switched on witched off. (down/up code = 0 or determines the alignment of justified). this is followed by a so	s defined as a switch. 'J': Image no. n1 witch. 'down code': (1-255) ''up code': (1-255) Return/touch macro on/off not reported). 'text': the first the text (C=centered, L=left justified, string that is placed in the key with the apparated with the character (\$7C,	
Define touch key with menu function	ESC	А	М	x1	y1	x2	у2	dow Cod	up Cod	mnu Cod	Text 	de':(1-255) Return/touch macreturn/touch macro when menu eturn/menu macro+(item no. 1) own/up code = 0: activation/cath the key text and the menu ite ection in which the menu open cond character determines the =centered, L=left justified, R=riparated by the character ' ' (\$7 ckey)item1 item2 item3". The kint and the menu items are writt ckground of the menu is savec	after selection of a menu item. ncellation is not reported.) 'text':= string ems. the first character determines the s (R=right, L=left, O=up, U=down). The alignment of the touch key text ght justified). The menu items are C,dec:124) (e.g. ey text is written with the current touch en with the current menu font. The lautomatically.	
Define drawing area	ESC	Α	D	x1	у1	x2	у2	n1				 You can then draw with a line xx2,yy2. 	e width of n1 within the corner	
Define free touch area	ESC	Α	н	x1	у1	x2	y2					a is defined. Touch actions (do xx2,yy2 are sent.	wn, up and drag) within the corner	
Set bar by touch	ESC	Α	В	nr								o. n1 is defined for input by tou	ch panel.	1
									G	iloba	l set			•
Touch query on/off	ESC	Α	Α	n1							_	ed (n1=0) or activated (n1=1);		1
Touch key response	ESC	A	S	n1 n1								n touch key touched: n1=0=OF n a touch key is touched: n1=0		1
Send bar value automatically	ESC	A	Q	n1								ion of a new bar graph value b setting (n1=1); each change is	y touch input is deactivated (n1=0); sent during setting (N1=2).	1
				_					0	ther	comr			
Invert touch key	4		N	Cod								ssigned return code is inverted	·	1
Set touch switch	ESC		P	Cod	n1							s changed by means of a com	·	
Query touch switch	ESC	Α	Х	Cod								off=0; on=1) is placed in the se		
Radiogroup abfragen			G	nr					Send	epuffe	er ges		roup mit der Nummer nr wird in den	
Delete touch area	ESC	A	L	Cod	n1				Wher	n n1=0	0, the	nains visible on the display; wh		
			v	x1	y1	n1						hat includes the coordinates xx elete area	1,yy1 from the touch query. n1=0: area	

ADJUSTTOUCHPANEL

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.

Adjustment procedure:

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



Page 18

RESPONSES OFTHE EA EDIP160-7VIA 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, alternatively pin 20 'SBUF' shows with Low-level that data is ready to transmit.

								Responses of the eDIP
lo	t	num			data			Remarks
							a	utomatic 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/switch. It is only 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 must be activated (see the 'ESC A Q n1' command).
ESC	N	1	code					After a menu item is selcted by touch, the selected menu item code is transmitted. It is only transmitted if no touch macro is defined with the number code.
ESC	Т	0						If automatic opening of a touch menu is disabled (see 'ESC NT n1'), this request is sent to the host computer. The host can then open the touch menu with the 'ESC N T 2' command.
ESC	P	1	value					After the input port is changed, the new 8-bit value is transmitted. The automatic port scan must be activated. See the 'ESC Y A n1' command. It is only transmitted when there is no corresponding port/bit macro 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 is drag within the free touch area at the coordinates xx1, yy1
					Res	spon	se onl	y when requested by command (placed into sendbuffer)
ESC	N	1	no					After the 'ESC N S' command, the currently selected menu item is transmitted. noo=0, no menu item is selected
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	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	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 all 8 inputs no=18: value is 0 or 1 depending on the status of the input no
ESC	٧	num		vers	sion string			After the 'ESC S V' command, the version of the edip firmware is transmitted as a string e.g. "EA eDIP160-7 V1.0 Rev.A TP+"
ESC	J	num		project	tname stri	ng		After the 'ESC S J' command, the macro-projectname is transmitted. e.g. "init / delivery state"
ESC	ı	21	C	RC-RO	s, Version, M, CRC-F DF in KB, CRC-DFso	ROMso	II	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 DFIen: number of user bytes in data flash memory (3 Bytes: LO-, MID- HI-Byte)
					_		R	esponses without length specification (num)
ESC	U	L	xx1	уу1	image data (G16-F			after the 'ESC UH' command, a hard copy is sent in BLH Format. xx1,yy1 = Start coordinates of the hard copy (upper corner) BLH-Data: 2 Byte: Width, height (in Pixel)+ amount of bytes of image data amount = ((width+7)/8*height
ESC	U	G	xx1	yy1		ge data FORM		In gray-scale mode, a second time 'amount of image data' is send for flashing mode



PRELOADED FONTS

Apart from the 8x8 terminal font (font no. 8), 3 additional monospaced fonts, 3 proportional fonts and 1 large numeric font are integrated as standard. The proportional fonts result in a more attractive appearance, and at the same time require less space on screen (e.g. the "i" is narrow and the "W" is wide). Each character can be positioned with pixel accuracy and the width and height can be scaled. Each text can be output left justified,

right justified or centered. 90° rotation is also possible. Macro programming permits additional fonts to be integrated (up to 15). This is done using the LCD-Tools*) (EA 9777-2USB).

+ 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		C)	×			-		~
\$30 (dez: 48)	0	i	2	3	4	5	6	7	В	9	:	i	<	=	>	?
\$40 (dez: 64)	0	A	В	c	D	E	F	G	н	ı	J	ĸ	L	н	n	0
\$50 (dez: 80)	P	a	R	s	т	Ш	V	н	×	γ	z	ι	ν.	1	^	-
\$60 (dez: 96)			ь		d	e	f	9	h	i	j	k	ι	н	n	
\$70 (dez: 112)	Р	q	,	,	t	u	v		×	9		•	ı	>	,,	٥
\$80 (dez: 128)	E	ü			ä										ä	
\$90 (dez: 144)					ä					ŏ	ü				β	

Font 1: 4x6 monospaced

				E												
+ 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	17	#	\$	z	8.		()	*	+	,	-		/
\$30 (dez: 48)	9	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$40 (dez: 64)	6	A	В	С	D	Е	F	G	Н	I	J	ĸ	L	н	N	0
\$50 (dez: 80)	Р	Q	R	s	Т	U	V	н	X	Y	z	ſ	١	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	v	н	x	y	z	{	:	}		Δ
\$80 (dez: 128)	e	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	¢	£	¥	В	f
\$A0 (dez: 160)	á	í	ó	ú	ñ	Ñ	<u>a</u>	0	į	-	,	½	У,	i	*	»
\$B0 (dez: 176)																
\$C0 (dez: 192)																
\$D0 (dez: 208)																
\$E0 (dez: 224)	α	ß	Г	π	Σ	σ	μ	ď	Φ	θ	Ω	ε	ø	ф	ε	n
\$F0 (dez: 240)	=	±	Σ	٤	ſ	J	÷	ä	0	•		1	n	2	3	-

Font 3: 7x12 monospaced

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

Font 7: Big characters BigZif57

+ 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)		!	=	#	\$	×	8.	,	(>	*	+	,	-		/
\$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:	j	<	=	>	?
\$40 (dez: 64)	0	A	В	С	D	Е	F	G	Н	I	J	К	L	М	N	0
\$50 (dez: 80)	Р	Q	R	s	Т	U	V	W	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	v	W	×	Э	z	{	1)	~	۵
\$80 (dez: 128)	ε	ü	é	ā	ä	à	á	ç	ē	ë	ė	ï	î	ì	Ä	À
\$90 (dez: 144)	É	æ	Æ	8	ö	ò	ũ	ü	ij	ö	ü	¢	£	¥	β	f
\$A0 (dez: 160)	á	i	0	ű	ñ	Ñ	<u>a</u>	01	٤	٦	Г	lá	lá	i	«	»
\$B0 (dez: 176)																
\$C0 (dez: 192)																
\$D0 (dez: 208)																
\$E0 (dez: 224)	α	β	г	π	Σ	σ	Д	т	Φ	θ	Ω	δ	٥	ø	Ε	n
\$F0 (dez: 240)	=	±	2	<u><</u>	Г	J	÷	22	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)		!		#	\$	%	&	•	()	×	+	,	-		7
\$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$40 (dez: 64)	@	Α	В	С	D	E	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]		_
\$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)	ε	ü	é	â	ä	à	0.00	ç	ê	:e	è	ï	î	ì	Ä	Ã
\$90 (dez: 144)	É	æ	Æ	ô	ö	ò	û	ù	ij	ö	Ü					
\$A0 (dez: 160)	á	í	ó	ú	ñ	Ñ	a	ō								
\$B0 (dez: 176)																
\$C0 (dez: 192)																
\$D0 (dez: 208)																
\$E0 (dez: 224)		В														
\$F0 (dez: 240)									0							

Font 4: GENEVA10 proportional



EAeDIP160-7

Page 20

+ Lower		l .		Ι.	Ι.		Ι.					Ι.				l .
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)		!		#	\$	%	8	1	()	*	+	,	-		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	Y	z	I	١	1	^	_
\$60 (dez: 96)	1	a	b	С	d	е	f	g	h	i	j	k	ı	m	n	o
\$70 (dez: 112)	p	q	r	s	t	u	υ	ш	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>	ō								
\$B0 (dez: 176)																
\$C0 (dez: 192)																
\$D0 (dez: 208)																
\$E0 (dez: 224)		ß														
\$F0 (dez: 240)									۰							

\$8 (8) \$A \$B \$C \$D \$E \$F (10) (11) (12) (13) (14) (15) \$3 \$4 (3) (4) \$20 (dez: 32) 0|1|2|3|4|5|6|7|8|9 @ABCDEFGH \$40 (dez: 64) PQRSTUVWX abcdefg \$60 (dez: 96) pqrstuvwxyz \$70 (dez: 112) ‱ (dez. 128) €ü é â ä à å ç ê ë è ï î \$90 (dez: 144) ÉæÆôöòûùÿöÜ \$AO (dez: 160) á í Ó Ú ÑÑao \$B0 (dez: 176) \$C0 (dez: 192) \$D0 (dez: 208) В \$E0 (dez: 224) \$F0 (dez: 240)

Font 6: Swiss30 Bold proportional

ADDITIONAL FONTS

Compile statement "WinFont:"

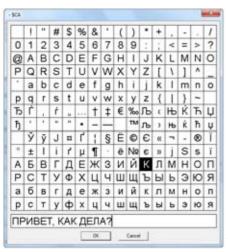
It is possible to raster TrueType-Fonts in different sizes whitch can be used. A doubleclick to the fontname within the KitEditor opens the font selection box. To simplify the use of fonts, there is the possibilty of an edit box. If you output a string with KitEditor (e.g. #ZL 5,5, "Hello"), you can 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}).

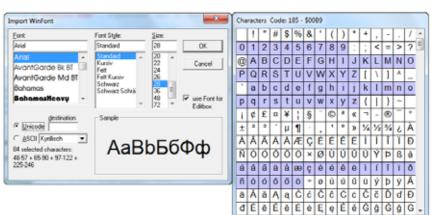
Compiler option "Font:"

Following font formats can be used:

- FXT: Textfont as used by eDIP240/320 and KIT series



Edit Box



Import WinFonts



integrated fonts in delivery state



DISPLAY BLINK MODE

After power on or the command 'ESC DG 0' the eDIP160 is in blink mode. Two picture contents are alternatly shown in an adjustable period.

Blink attributs are set by the commands 'ESC ZB, UB, GB n1':

n1=0: no blink n1=1: On/Off blink

n1=2: blink inverted

n1=3: Off/On blink (phase shifted)

Between strings ('ESC ZL,ZC,ZR. ..), flashing can be activated locally: Strings between two '~' (\$7E) mean blink on/off.

Strings between two '&' (\$26) mean blink off/on phase shifted.

Strings between two '@' (\$40) mean blink inverted.

In addition you can assign or delete postly an rectangle area a blink mode, by using the command 'ESC Q...'



DISPLAY GRAYSCALE

After the command **'ESC DG 1'** the eDIP160 is in grayscale mode. Flashing is no longer possible, instead the blink mode is used for two grayscales.

Gray colours are generated with the help of the flashing modes 'ESC ZB, UB, GB n1':

n1=0: Black

n1=1: Dark gray

n1=3: light gray

Within strings (ESC ZL, ZC, ZR...), gray characters can be used locally:

Text between two '~' (\$7E) are dark gray.

Text between two '&' (\$26) are light gray.

In addition rectangular areas can be filled gray or black postly, with the use of commands of flashing areas ('ESC Q...').



```
Macro: MnAutoStart
   #TA
                                  : Terminal off
   #AR 1
                                   ; define radiogroup
   #AE 13
                                  ; define frame
                                  ; define touchlabelfont
   #AF CHICAGO14
       20,30, 70,50, 1,0, "BLINK"; place button "blink"
   #AK
   #AK 100,30,150,50, 2,0, "GRAY" ; place button "gray"
   #ZF SWISS30B
                                  ; select textfont
   #ZC 45,35, "~OnOff~|&OffOn&"; place blink text
   #ZC 125,35, "~dark~|&light&" ; place gray text
TouchMacro: 1 ; Flashing
              ; grayscalemode off -> flashing possible
TouchMacro: 2 ; Grayscale modus
               ; grayscalemode on -> flashing impossible
                                   Example code of the hardcopys
```



MACRO PROGRAMMING

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'.

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 under 3.0V (typ.).

Wake-up-pin-macro WakeupPinMacro:

Started after wake up from power-down-mode with pin 13 (WUP).

Wake-up touch-Macro WakeupTouchMacro:

Started after wake up from power-down-mode with touch (the whole touch area is active).

Wake-up I2C-Macro WakeupI2CMacro:

Started after wake up from power-down-mode with the I²C bus.

Important: If a continuous loop is programmed in a power-on, reset, watchdog or brown-out macro, the display can no longer be addressed. In this case, the execution of the power-on macro must be suppressed. You do this by wiring DPOM:

- PowerOff connect pin 13 (DPOM) to GND
- PowerOn open pin 13 (DPOM) again.



STORING IMAGES IN THE DATA FLASH MEMORY

To reduce the transmission times of the interface or to save storage space in the processor system, up to 256 images can be stored in the internal EEPROM with the "PICTURE" compiler directive. They can be called using the "ESC U I" command or from within a macro.

All images in the Windows BMP format (monochrome images only) can be used. They can be created and edited using widely available software such as Windows Paint or Photoshop or the bitmap editor shipped with the product.

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 9777-2USB 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 eDIP compiler using F5. This creates a file called DEMO.EEP. If an EA 9777-2USB 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 memory.

You can send the created macrofile *.EEP with any other system to the EA eDIP160-7. 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 eDIP160-7.

KIT-EDITOR HELP (ELECTRONIC ASSEMBLY LCDTOOLS)

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 unter<u>http://www.lcd-module.de/deu/dip/edip.htm</u>



Page 24

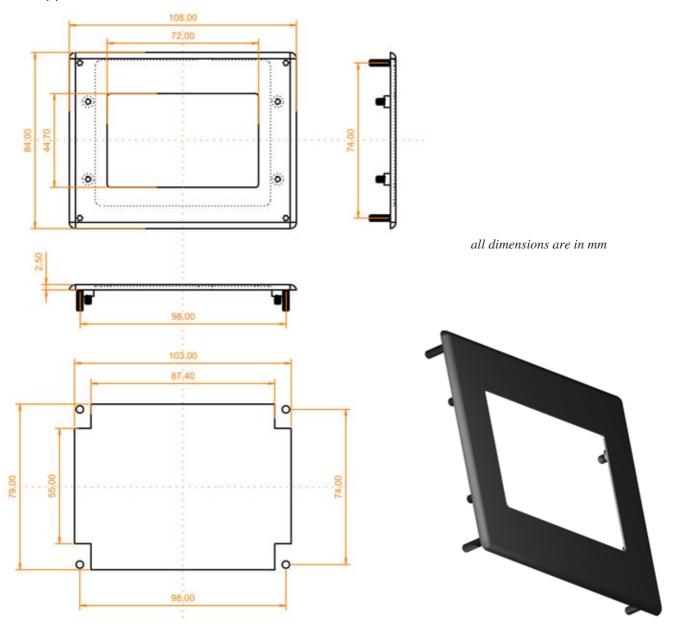
SPECIFICATION AND ELECTRICAL CHARACTERISTICS

	Chara	cteristics			
Value	Condition	min.	typ.	max.	Unit
Operating Temperature		-20		+70	°C
Storage Temperature		-30		+80	°C
Storage Humidity	< 40°C			90	%RH
Operating Voltage		3.2		5.2	V
Input Low Voltage		-0.5		0.2*VDD	V
Input High Voltage	Pin Reset only	0.9*VDD		VDD+0.5	V
Input High Voltage	except Reset	0.6*VDD		VDD+0.5	V
Input Leakage Current				1	uA
Input Pull-up Resistor		20		50	kOhms
Output Low Voltage				0.7	V
Output High Voltage	VDD = 3,3V VDD = 5V	2.5 4.2			V
Output Current				20	mA
Current Backlight on	VDD = 3,3V VDD = 5V		190 125		mA
Current Backlight off	VDD = 3,3V VDD = 5V		10 15		mA
Power Down	Mode 0		25		μΑ



MOUNTING BEZEL EA 0FP161-7SW

As accessory we deliver an optional black anodized mounting bezel. The mounting clips are included in the supplied EA eDIP160-7.



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.



EAeDIP160-7

Page 26

ELECTRONIC ASSEMBLY reserves the right to change specifications without prior notice. Printing and typographical errors reserved.

NOTES



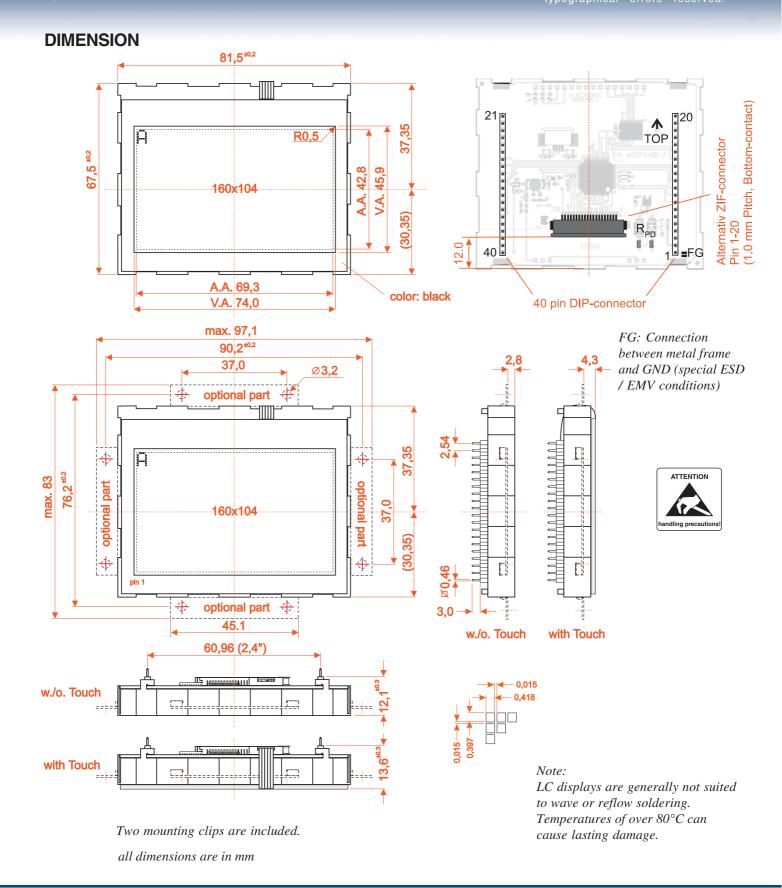
ELECTRONIC ASSEMBLY reserves the right to change specifications without prior notice. Printing and typographical errors reserved.

EAeDIP160-7

Page 27

NOTES





ELECTRONIC ASSEMBLY

making things easy