EA uniTFTm Series

www.lcd-module.com/unitft

FEATURES

- 7.0" 1024x600 dots, 10.1" and 5" will follow soon
- AACS-Display (all angle color stability, optimized backlight and IPS-Panel with widest viewing angle)
- Superbright LED backlight over 800 cd/m²
- Brilliant IPS panel >160° viewing angle
- Object-oriented screen layout
- Change object during run-time: size, shape, color, content
- Animate and move objects, alpha-blending
- Fonts: ASCII and Unicode
- Single supply 3.3 V or directly through USB
- Serial Interfaces: USB, RS232, SPI, I²C
- 8 digital, freely definable I/Os built in, expandable up to 136, 4 analog inputs
- RTC Timer
- Flash-memory as storage for pictures, fonts, menus and log-files
- Internal functions for calculation as well as programmability
- Tone feedback build-in

ORDERING CODES

7.0" IPS 1024 x 600 pixels, PCAP and white LED backlight	**EA uniTFTm070-ATC**
7.0" IPS as above but with 2 red MicroMatch connectors (uniTFT070-A compatible)	**EA uniTFTm070-BTC**

ACCESSORIES

ZIF-connector 40 positions 0.5 mm pitch connector for FPC-cable	**EA WF050-40S**
FPC-cable 40 positions 0.5 mm pitch 102 mm long	**EA KF050-40**
USB-Kabel Typ A -> USB C around 1 m	**EA KUSB-C**
MicroMatch THT, 26 position mating connector for soldering (2 pieces needed)	**EA B2B127M-26T**
MicroMatch ribbon cable, 26 position mating connector crimp (2 pieces needed)	**EA B2B127M-26Q**
Cable (25 cm) with crimped connectors	**EA KM-126**

Content

GENERAL

EA uniTFTm are a high-quality all-in-one implementation of the display, microcontroller unit, and touch screen. It is all users need to directly control their application and expedite development, prototyping, and deployment of their HMI/GUI. Design the HMI/GUI using the easy-to-use drag-and-drop uniTFTdesigner graphics development software

The EA uniTFTm series provides sophisticated graphical functions and intuitive menu control with its built-in instruction set. Thanks to the integrated instruction set and the Windows design software uniTFTDesigner, not only electronics specialists, but also experts in the field of design and user guidance are able to create the entire HMI.

The display modules are immediately ready for operation with 3.3 V, controlled via the built-in serial interfaces RS232, SPI, I²C or USB. The modules can be operated directly through the USB, too.
Object-oriented "programming", the wide set of commands, and the integrated but extensible Unicode fonts make "time-to-market" a breeze.

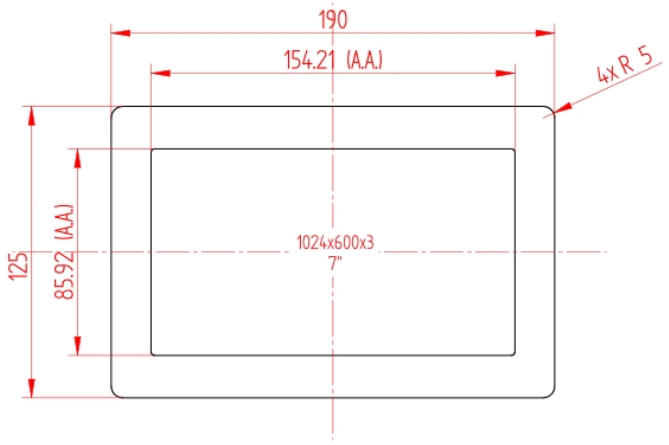
The EA uniTFT series, which forms the high-end market with larger modules, comes up with a very similar command set:
Currently are 3 different sizes available: 5" with 800x480 dots, 7" with 1024x600 dots and 10.1" with 1280x800 dots.

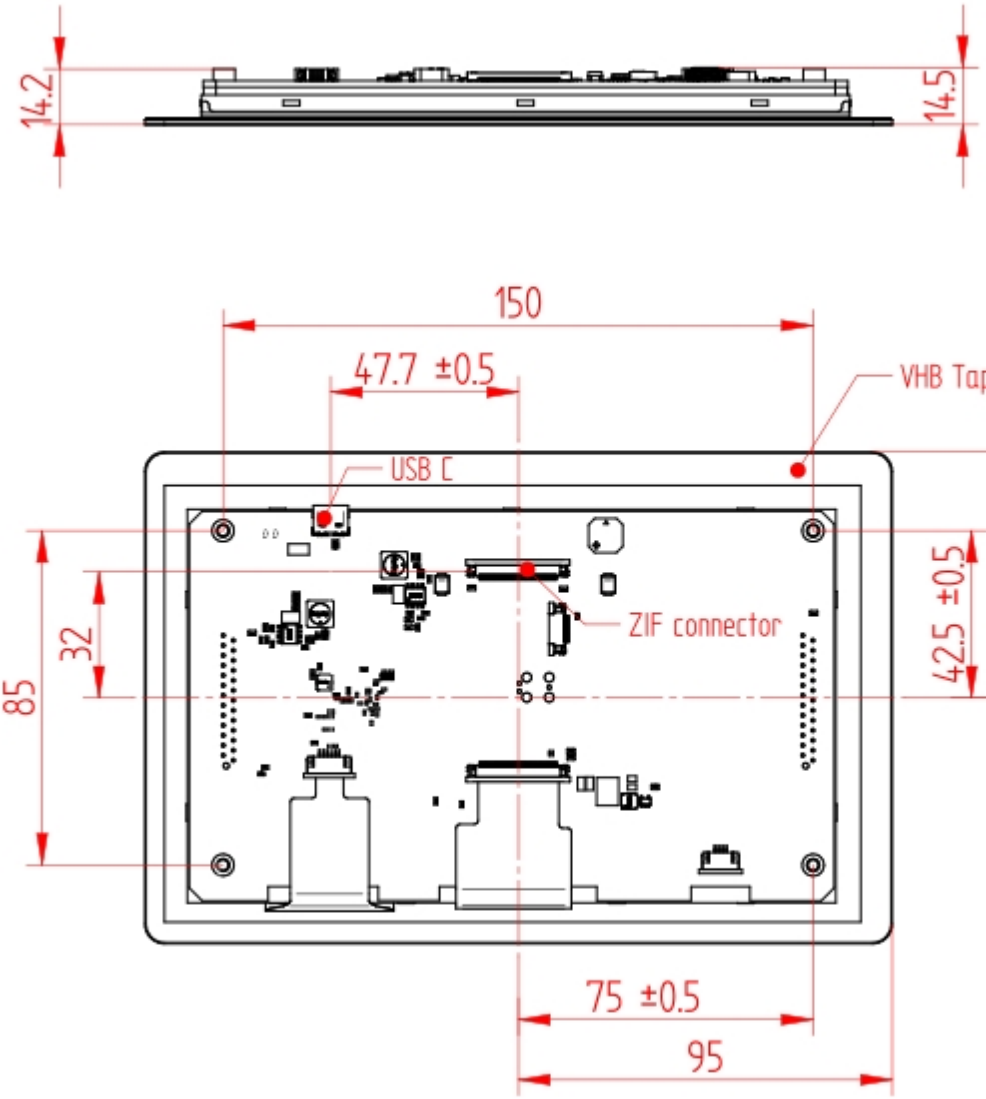
Advantage	Standard TFT	EA uniTFTm
Quality	Consumer	Non-consumer
Brightness	250cd	1000cd (typ.)
Viewing angle	Limited +/-50° (typ.)	Up to 340°
Color	TN with Gray inversion effect	IPS: no color shift
Touch	resistive	PCAP incl. controller
Interface	8-/16 Bit data bus	I²C or SPI or USB
Availability	Minimum order quantity	Ex stock
Longevity	1 year or more	Minimum of 8 years
Support	None unless high quantity	Bundled with product purchase
ce / EMC	None	Tested and certified
Software	Place dot by dot to create character or touch buttons	- High level commands included - Graphics development software f.o.c.

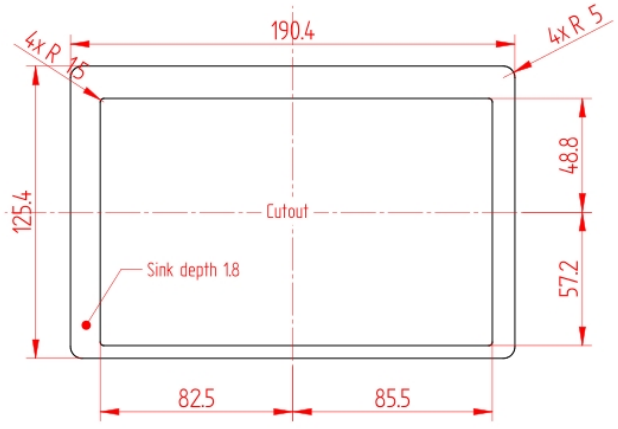
To get an easy introduction to the use of the commands we have a collection of short explanatory videos with the uniTFT Academy online.
You are welcome to subscribe to our Youtube channel.


uniTFT
academy
YouTube

Dimension EA uniTFTm070

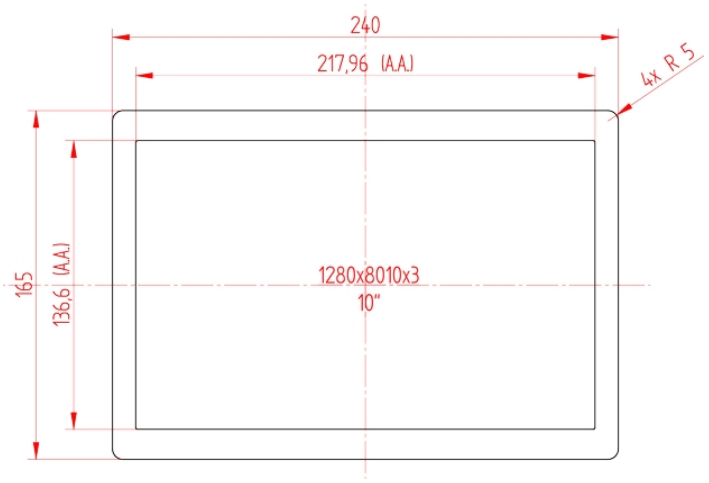
EA uniTFTm070-ATC	EA uniTFTm070-BTC
	

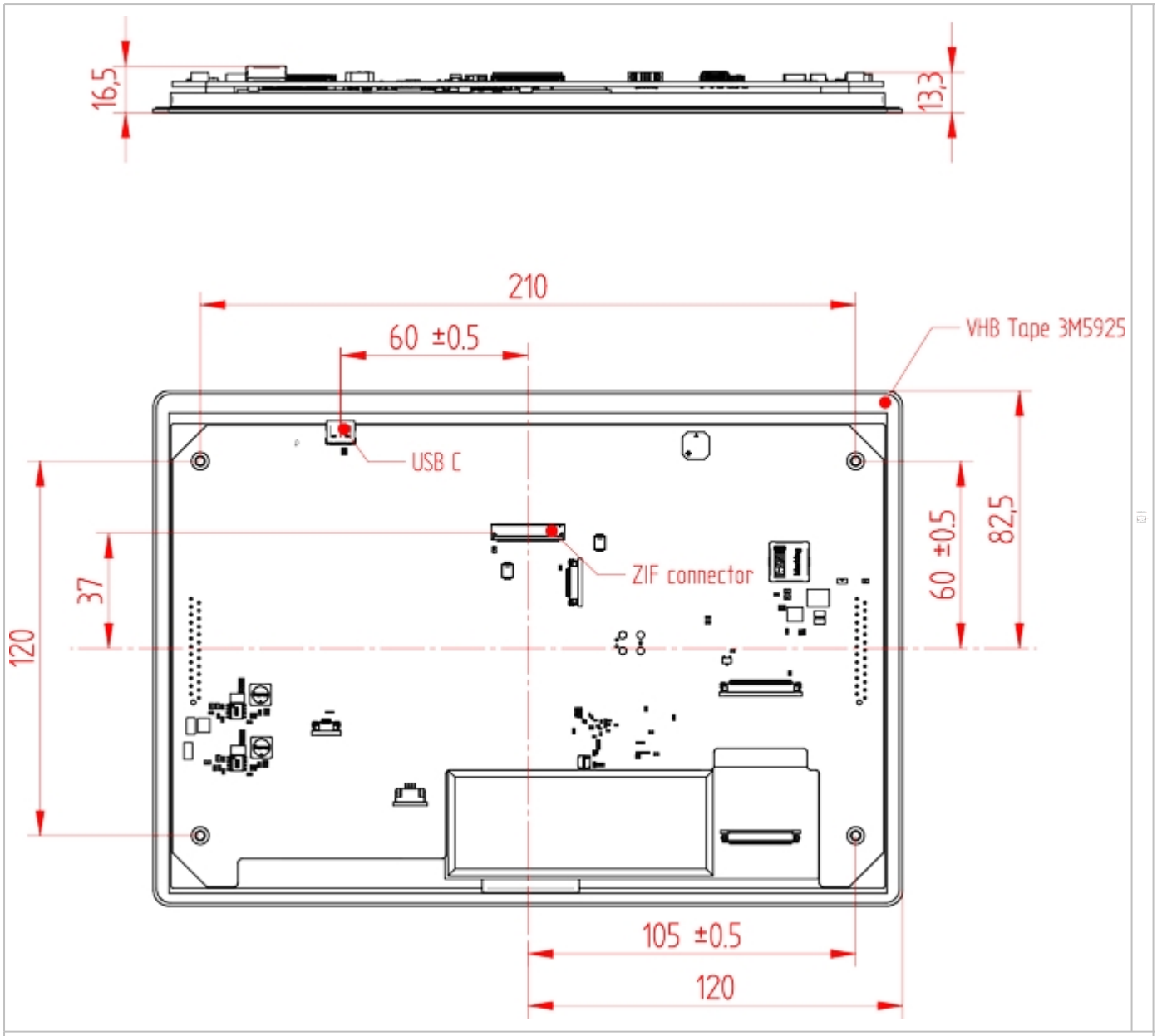

14.2
14.5
150
47.7 ±0.5
VHB Tape 3M5925
USB C
ZIF connector
85
32
42.5 ±0.5
62.5
75 ±0.5
95

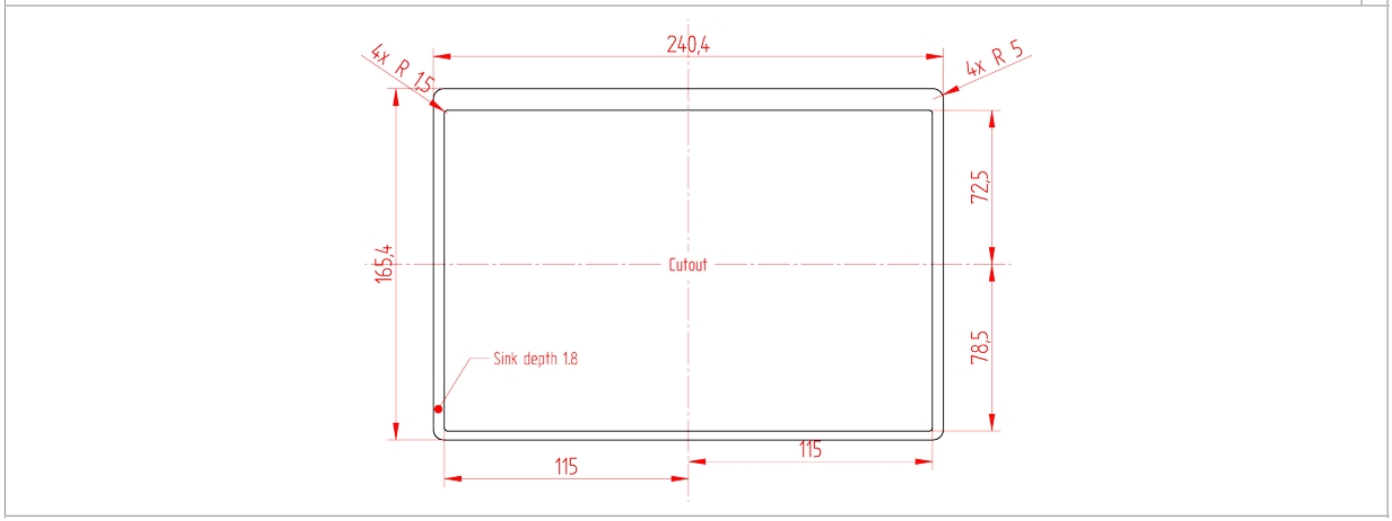

190.4
4x R 5
4x R 15
48.8
125.4
Cutout
Sink depth 1.8
57.2
82.5
85.5

Dimension EA uniTFTm101

EA uniTFTm101-ATC	EA uniTFTm101-BTC



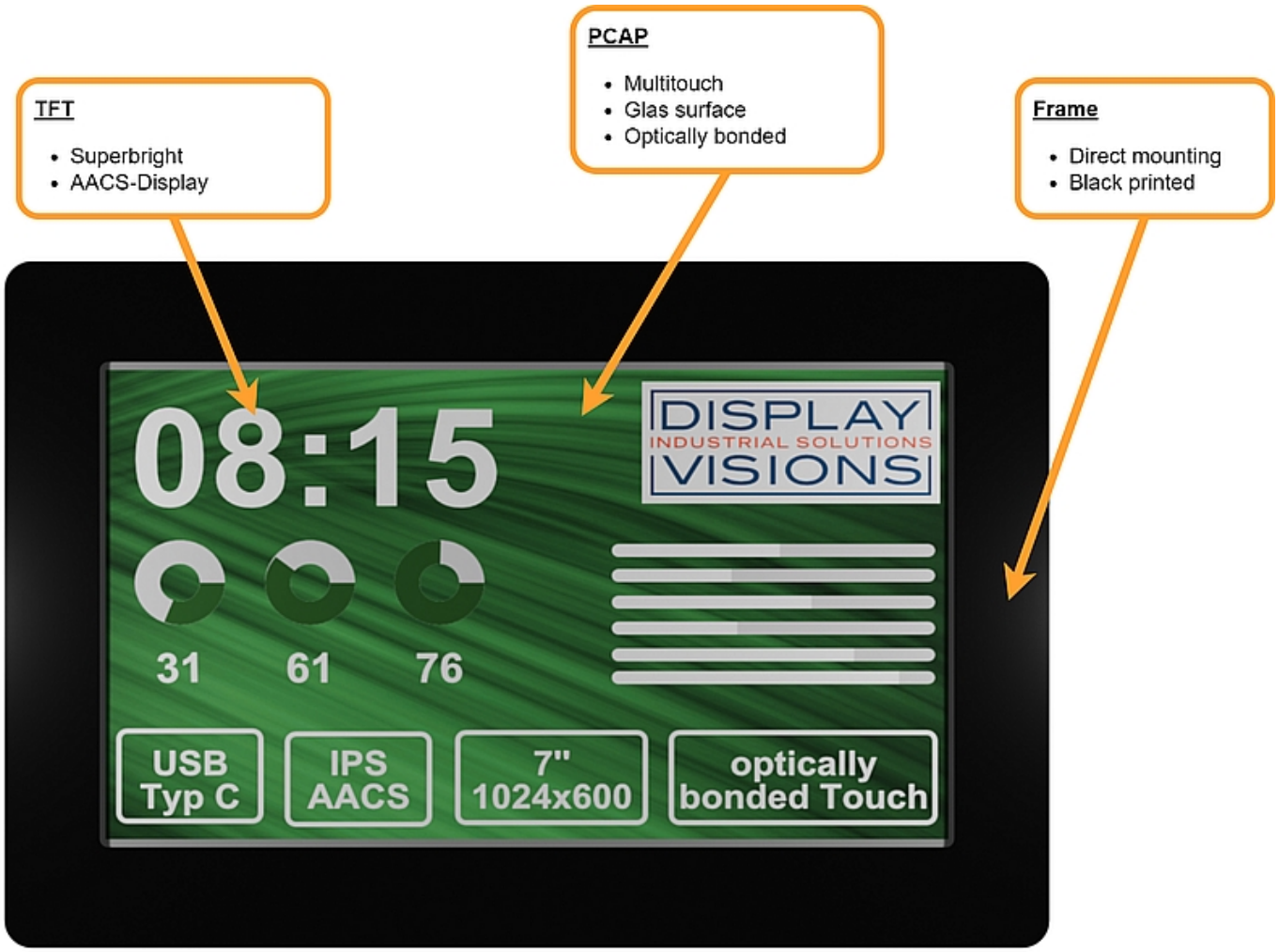

16,5
13,3
210
60 ±0.5
VHB Tape 3M5925
USB C
ZIF connector
37
120
60 ±0.5
82,5
105 ±0.5
120


4x R 15
240,4
4x R 5
72,5
165,4
Cutout
78,5
Sink depth 1.8
115
115

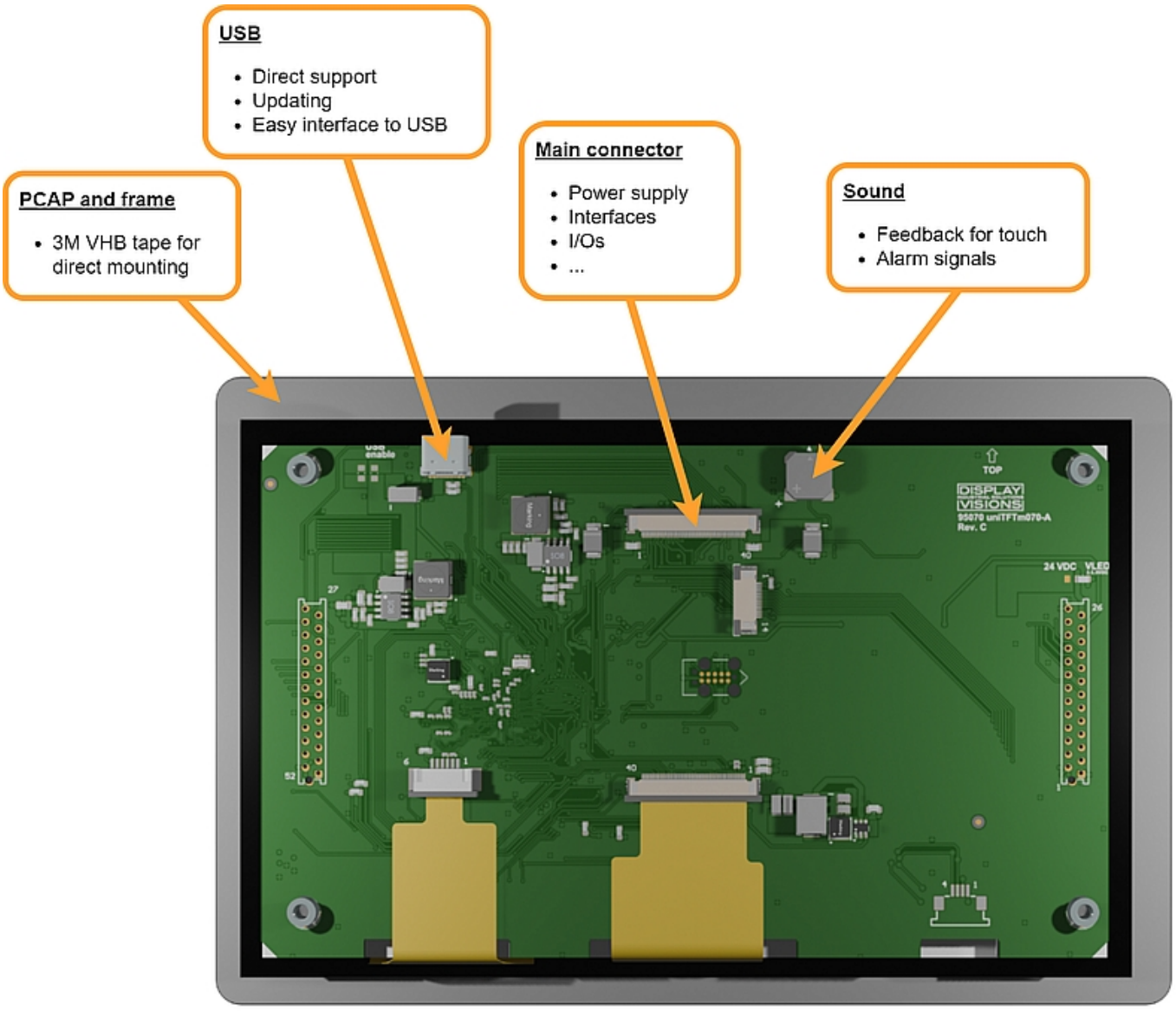
HARDWARE

The EA uniTFTm-Series consists of a TFT/IPS-Display with LED backlight, driven by an integrated driving circuit, which is dimmable using software commands. In 24/7 operation the backlight can be dimmed automatically to increase the LED life-time and save energy. On the other hand brightness may rise to 150% for excellent readability even at direct sunlight. But it reduces life time of backlight.
The module is designed to work with 3.3 VDC. Serial data transfer is possible through RS232, SPI, I²C or direct via USB protocol.
For simple control tasks, the module has 8 freely usable I/Os (expandable up to 136), 4 analogue inputs, one PWM output and 3 serial interfaces (RS232, SPI and I²C).
The modules do have an integrated capacitive touch panel. By touching the display you can enter data and make adjustments via menu or bar graph. The labelling, size and shape of the "keys" is flexible and can also be changed during runtime (different languages, icons). The drawing of the individual "keys", as well as the labelling is completely taken over by the built-in software. The capacitive touchpanel has an robust glass surface that can also be operated with thin gloves.

Front view (example EA uniTFTm070-ATC)



Rear view (example EA uniTFTm070-ATC)


USB
• Direct support
• Updating
• Easy interface to USB
PCAP and frame
• 3M VHB tape for direct mounting
Main connector
• Power supply
• Interfaces
• I/Os
• ...
Sound
• Feedback for touch
• Alarm signals

Pin assignment ZIFF

Pin assignment for ZIF connector. It's an FPC connector with 40 positions and 0.5 mm pitch. Bottom contact.

Pin	Symbol	I/O	Description	
1	GND		Ground 0 V	
2	VDD		Power Supply 3.3 V	Power supply or 3.3V output at USB operation (max. 20
3	$\overline{RES}$	I	$\overline{Reset}$	internal Pull-Up: (10..75 kΩ)
4	$\overline{CS}$	I	SPI: Chip Select	internal Pull-Up: (1 MΩ)
5	MOSI	I	SPI: MOSI	
6	MISO	O	SPI: MISO	
7	CLK	I	SPI: CLK	internal Pull-Up: (1 MΩ)
8	RxD	I	RS232: Receive Data	internal Pull-Up: (1 MΩ)
9	TxD	O	RS232: Transmit Data	
10	DE	O	RS485: Transmit Enable	
11	SDA	I/O	I²C: Serial Data	internal Pull-Up: (10 kΩ); Pull-Up resistors can be chan
12	SCL	I	I²C: Serial Clock	internal Pull-Up: (10 kΩ´
13	$\overline{SBUF}$ $\overline{TESTMODE}$	I	Low: Data available in send buffer PowerOn Low: Test mode	internal Pull-Up: (10 kΩ)
14	$\overline{DPROT}$	I	High: Small-/Shortprotokoll active Low: deactivated	internal Pull-Up: (10 kΩ)
15	A/D 0	I	Analog Input 0	internal Pull-Down: (1 MΩ)
16	A/D 1	I	Analog Input 1	
17	A/D 2	I	Analog Input 2	
18	A/D 3	I	Analog Input 3	
19	I/O 0.0	I/O	I/O 0.0 (Bit 0)	internal Pull-Up: (1 MΩ), Reset-state: Tri-State, default:
20	I/O 0.1	I/O	I/O 0.1 (Bit 1)	
21	I/O 0.2	I/O	I/O 0.2 (Bit 2)	
22	I/O 0.3	I/O	I/O 0.3 (Bit 3)	
23	I/O 0.4	I/O	I/O 0.4 (Bit 4)	
24	I/O 0.5	I/O	I/O 0.5 (Bit 5)	
25	I/O 0.6	I/O	I/O 0.6 (Bit 6)	
26	I/O 0.7	I/O	I/O 0.7 (Bit 7)	
27	PWM	O	PWM-Output	
28	DNC	---	Do not connect	Reserved for future use
29	DNC	---	Do not connect	
30	DNC	---	Do not connect	
31	DNC	---	Do not connect	
32	VDD BAT			Battery VDD for RTC
33	+5-28V=		Power Supply	Power supply input +5..28V=
34	BUZZER	O	Sound	PWM output for external speaker

The following connectors are only assembled in the version uniTFTmxxx-BTC.

Primary connector MicroMatch (-BTC only)

Pin	Symbol	I/O	Description	
1	GND		Ground 0 V	
2	VDD		Power Supply 3,3 V	
3	$\overline{\text{RES}}$	I	$\overline{\text{Reset}}$	intern Pull-Up: (1,3 kΩ)
4	$\overline{\text{CS}}$	I	SPI: Chip Select	intern Pull-Up: (1 MΩ)
5	MOSI	I	SPI: MOSI	intern Pull-Up: (1 MΩ)
6	MISO	O	SPI: MISO	
7	CLK	I	SPI: CLK	intern Pull-Up: (1 MΩ)
8	RxD	I	RS232: Receive Data	intern Pull-Up: (1 MΩ) (Slave)
9	TxD	O	RS232: Transmit Data	(Slave)
10	DE	O	RS485: Transmit Enable	
11	SDA	I/O	I²C: Serial Data	internally Pull-Up: (1 MΩ) → with I²C mode an external
12	SCL	I	I²C: Serial Clock	internally Pull-Up: (1 MΩ) → with I²C mode an external
13	A/D 0	I	Analogue input 0	
14	NC	---	Not connected	
15	A/D 1	I	Analogue input 1	
16	NC	---		
17	I/O 0.0	I/O	I/O 0.0 (Bit 0)	
18	NC	---		
19	I/O 0.1	I/O	I/O 0.1 (Bit 1)	
20	$\overline{\text{SBUF}}$ $\overline{\text{TESTMODE}}$	O I	Low: Data in send buffer PowerOn Low: Testmode active	intern Pull-Up: (10 kΩ)
21	PWM	O	PWM-output	
22	$\overline{\text{DPROT}}$	I	High: Small-/Shortprotokoll active Low: deactivated	intern Pull-Up: (1 MΩ)
23	VDD BAT			
24	NC	---		
25	VLED		Power Supply Backlight	Power Supply Backlight 3.3V - 9V
26	GND		Ground 0 V	

Secondary connector MicroMatch (-BTC only)

Pin	Symbol	I/O	Description	
27	I/O SDA	I/O	I²C data to portexpander	Portexpander MCP23017-E can be connected here
28	I/O SCL	O	I²C clock to portexpander	If not used please leave open
29	I/O $\overline{INT}$	I	Interrupt portexpander	If not used please leave open
30	NC	---		
31	NC	---		
32	NC	---		
33	NC	---		
34	NC	---		
35	I/O 0.7	I/O	I/O 0.7 (Bit 7)	
36	I/O 0.6	I/O	I/O 0.6 (Bit 6)	
37	I/O 0.5	I/O	I/O 0.5 (Bit 5)	
38	I/O 0.4	I/O	I/O 0.4 (Bit 4)	
39	I/O 0.3	I/O	I/O 0.3 (Bit 3)	
40	I/O 0.2	I/O	I/O 0.2 (Bit 2)	
41	A/D 3	I	Analogue input 3	
42	A/D 2	I	Analogue input 2	
43	NC	---		
44	NC	---		
45	NC	---		
46	NC	---		
47	NC	---		
48	NC	---		
49	NC	---		
50	NC	---		
51	VDD		Power Supply 3,3 V	
52	GND		Ground 0 V	

POWER SUPPLY

The modules can be powered in different ways. Separating display and backlight supply may be an advantage to bring down current at 3.3V power supply.

	ZIFF connector			USB-C
Supply Voltage	Single supply 3.3 V	Singe supply 5V - 28V	Splitted supply 3.3V for display 3.3~7.5V for backlight	Power over USB

Current consumption is mentioned in chapter Electrical Characteristics.

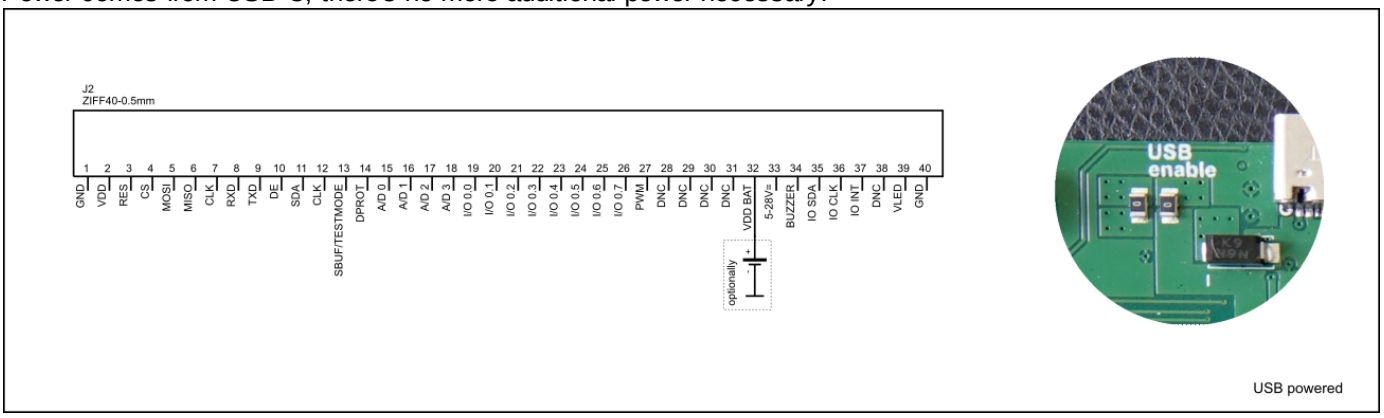
Attention:
In order to avoid fault currents, the "USB Power Enable" solder bridge must be set correctly. The solder bridge is closed by default. The internal voltage regulator is now active and generates 3.3 V from the connected USB supply. If an additional 3.3 V is now supplied externally, fault currents occur.
Attention:
Note that for all power supply options inputs and outputs (RES, AD, I/O. RxD...) are 3.3V level.

APPLICATION EXAMPLES

There are several power supply options available. Pinout for ZIFF connector and red MicroMatch is different.
Please note that some solder link / resistor (0 R) need to be modified.

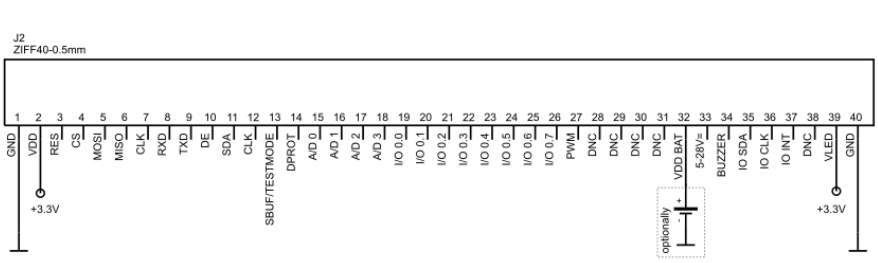
1. USB powered (factory set)

Power comes from USB-C, there's no more additional power necessary.



2. ZIFF: Single supply 3.3V

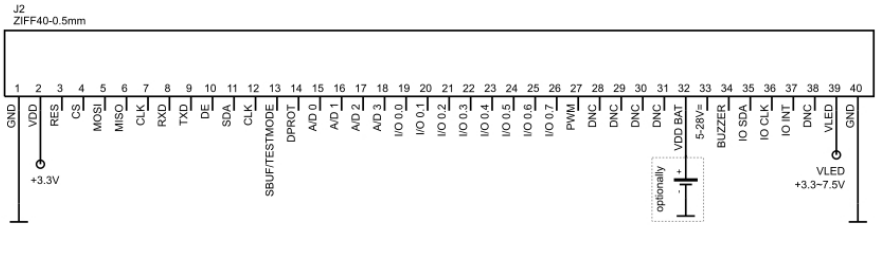
Driving the display with single supply 3.3V.



ZIFF connertor - single supply

3. ZIFF: Display and backlight separately

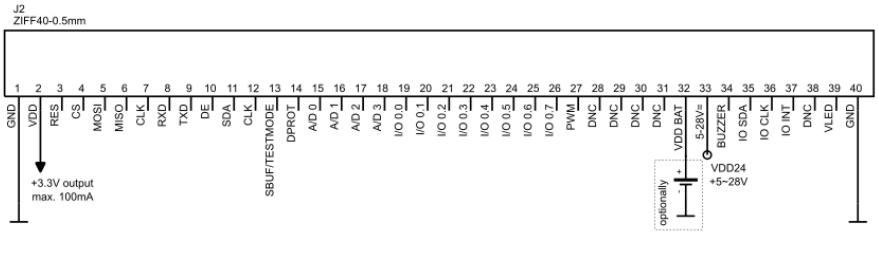
Driving the display with 3.3V and LED backlight with higher voltage up to 7.5V.



ZIFF connertor - separate LED power

4. ZIFF: Display Supply Voltage 5~28V

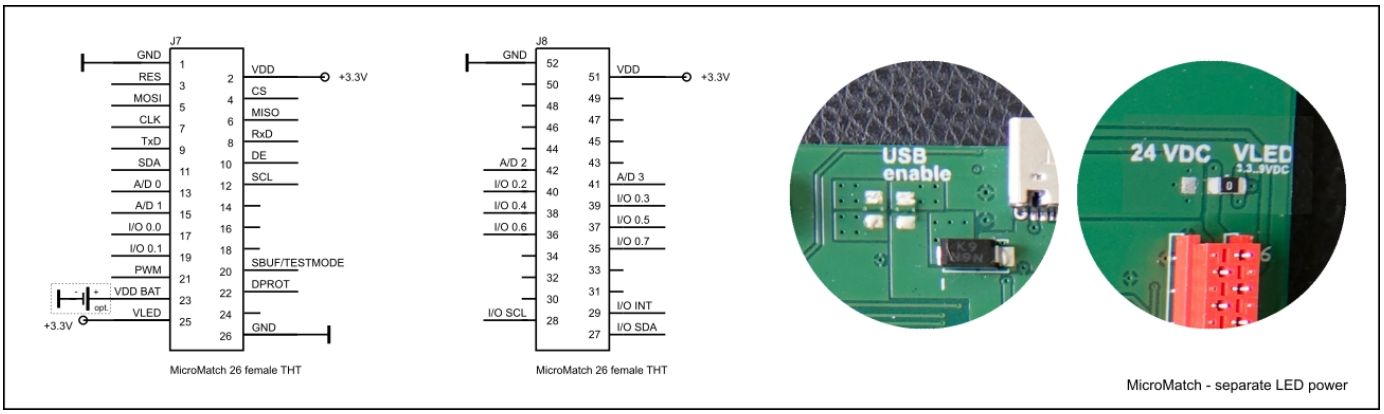
Driving the display and backlight with 5..28V=.



ZIFF connertor - 5~28V

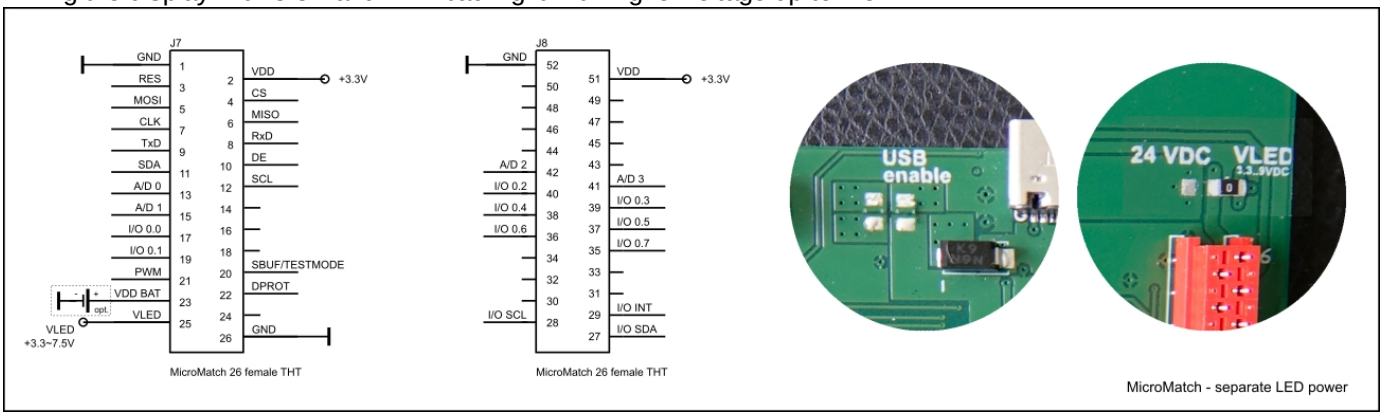
5. MicroMatch: Single supply 3.3V (uniTFTm070-BTC only)

Driving the display with single supply 3.3V.



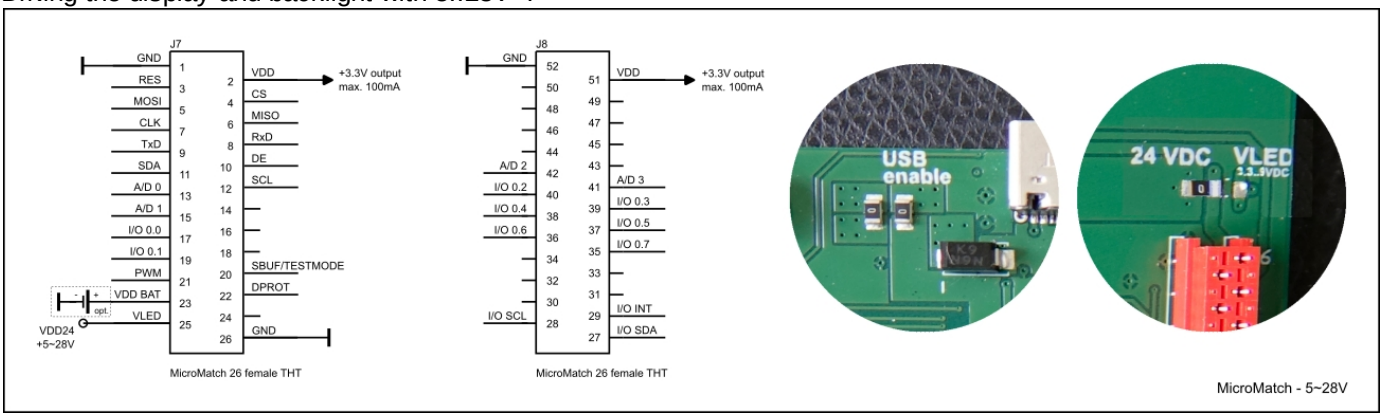
6. MicroMatch: Display and backlight separately (uniTFTm070-BTC only)

Driving the display with 3.3V and LED backlight with higher voltage up to 7.5V.



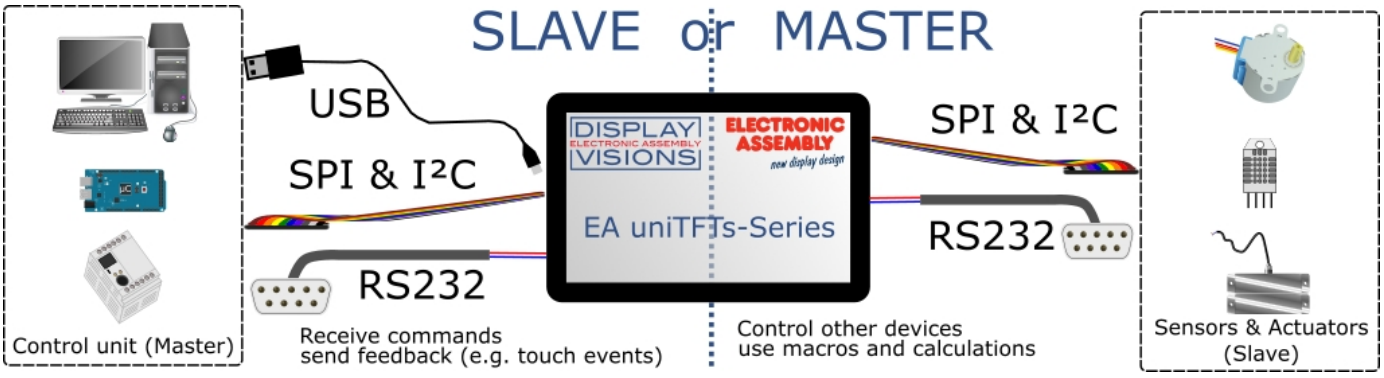
7. MicroMatch: Display Supply Voltage 5~28V (uniTFTm070-BTC only)

Driving the display and backlight with 5..28V=.



Serial interfaces

The module provides 4 serial interfaces, including RS232, SPI, I²C and USB. In addition to the USB interface, the other interfaces can change your behaviour:
They can either be used to connect to an external host, i.e. to a higher-level controller, or used as a master interface.
By default, all interfaces are parameterized as slaves and accept the commands.
Parameterized as a master interface, it enables the control of external sensors and actuators. The display module behaves here as a master.
As already described, the interfaces behave as slave interfaces by default and accept commands. However, as soon as a master interface command (#H ...)is executed, the interface gets master functions.



RS232

RS232 is a standard for a serial interface.

The EA uniTFT provides one RS232 interface, that can be operated as slave (default) or as master: As slave interface it is used to communicate with the display. All data sent to the display are interpreted as a command (with and w./o. Small-/Short-Protocol). If you would like to send and receive any data via RS232 to other devices then you have to use it as master. Those are handled via #H commands.

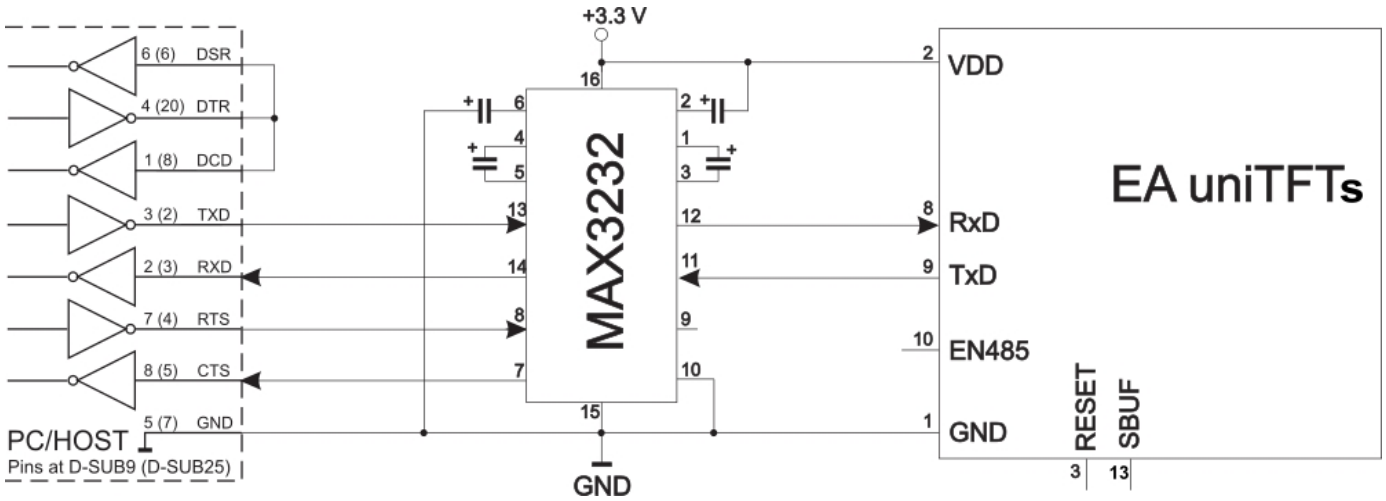
The transmission is serially asynchronous. Thus the data is converted into a bit stream and transmitted. There is no clock signal, so transmitter and receiver need to work with the same data rate (so-called baud rate). RS232 is a voltage interface, such that data is transmitted using changing voltage levels. In the PC world and industrial controls, levels of + 12V and - 12V are defined as standard. With boards or micro-controllers levels of 0V and VDD (in the case of EA uniTFTm-Series 3.3 V) are common. To adjust the signal levels, there are some possibilities in the form of level shifters (e.g., ICL232, MAX202). RS232 consists of "listening" and "talking" lines that are crossed between the two parties.
In the EA uniTFTm-Series, the data format is fixed to 8-N-1. Baud rate is pre set to 115200 baud:

The EA uniTFTm-Series works with the following baud rates:

Baud	Error	Baud	Error
9600	+0.04	115200	+0.64
19200	-0.08	230400	-0-80
38400	+0.16	460800	+2.08
57600	-0.08	921600	-3.68

The parameters (baud rate) are set using command #XCR (higher-level control unit), and the master interface is set with the command #HRP. Those definition can be done in start.emc e.g.
You may check the current settings in Boot menu.

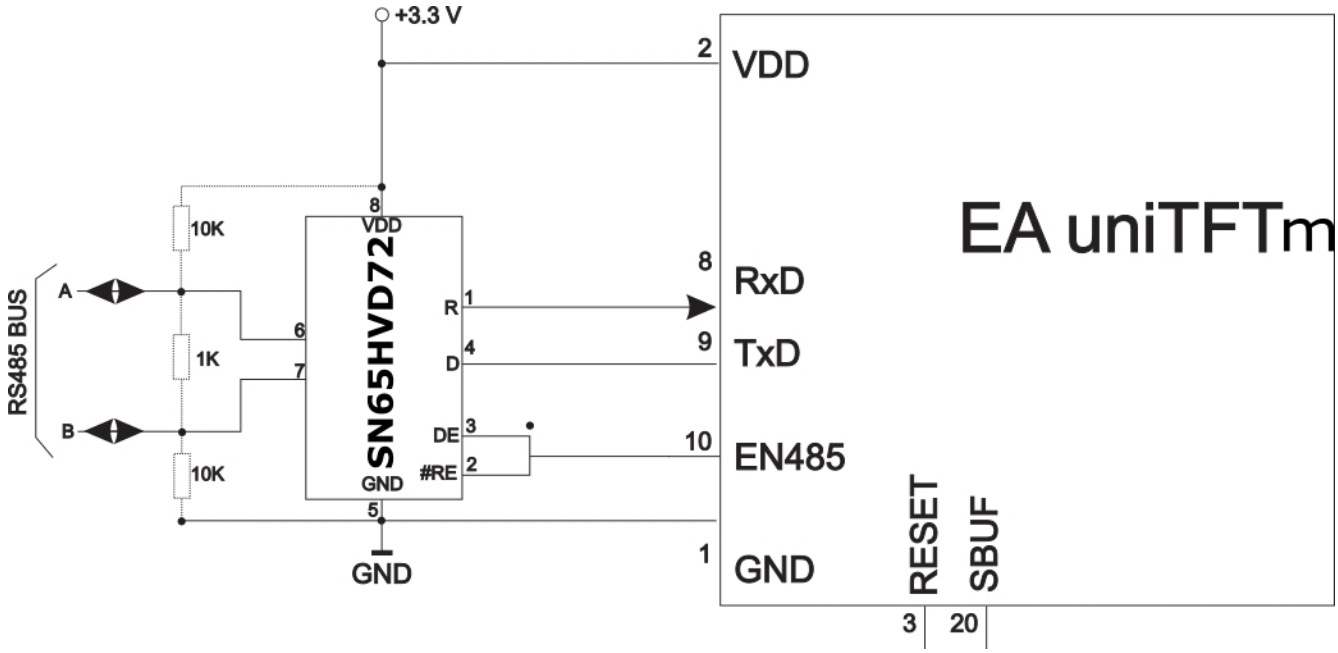
Application notes



RS232 V24 - Interface to a PC (EA uniTFTm)

RS485 / RS422 interface

With this simple external IC a communication to any RS-485 and RS-422 can be done.



RS485 - Interface to a PLC (EA uniTFT)

SPI

The **S**erial **P**eripheral **I**nterface is a bus system for serial synchronous data transfer.

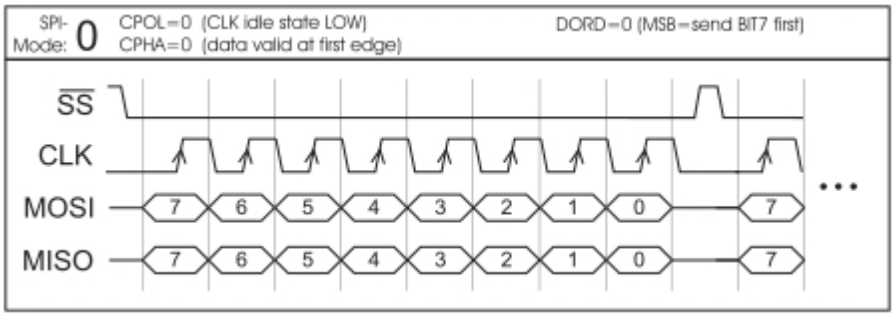
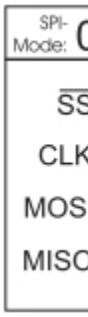
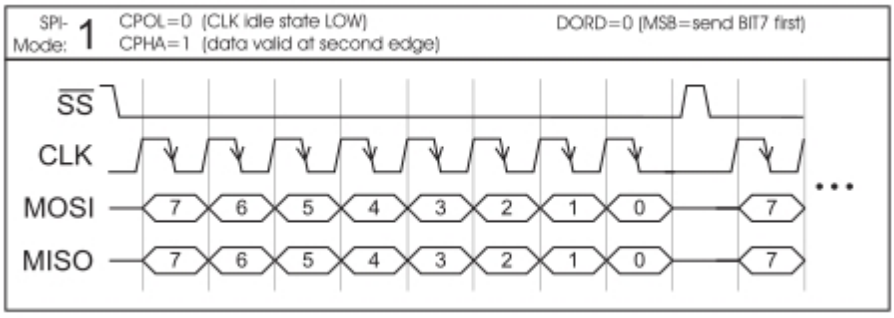
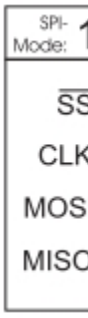
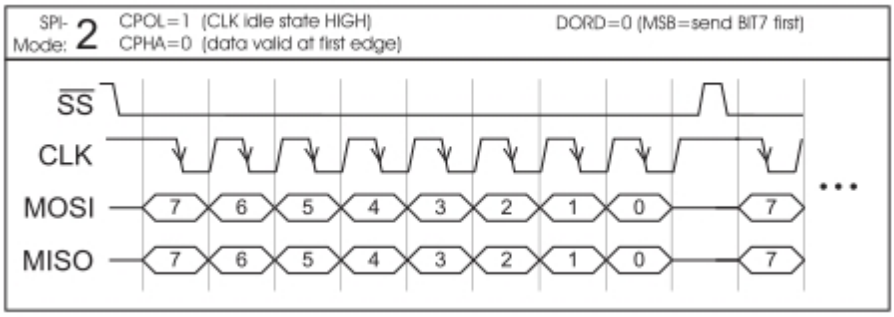
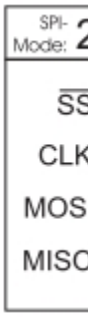
The EA uniTFT provides a SPI interface: AS default the interface has Slave functionality and is used to communicate with the display. All data sent to the display are interpreted as a command (with and w./o. Small-/Short-Protocol).

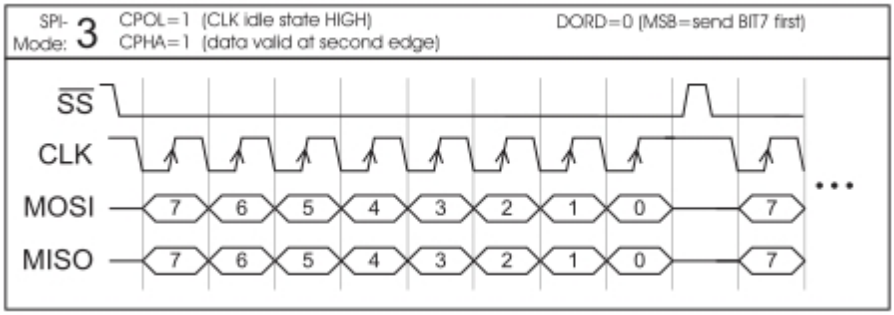
Would you like to send and receive any data via SPI to other devices like temperature sensor, then you have to use the Master interface. Those are handled via #H commands.

The SPI is working with 4 lines:

- MOSI (**M**aster **O**ut → **S**lave **I**n) or SDO (Serial Data Out) or DO
- MISO (**M**aster **I**n ← **S**lave **O**ut) or SDI (Serial Data In) or DI
- SCK (**S**erial **C**lock) - Shift clock
- SS (**S**lave **S**elect → Addressing) or CS (Chip Select)

SPI works with a bidirectional transmission principle, meaning that data is exchanged between the connected devices at the same time. The communication is controlled by the master using the SCK line.
The protocol for data transfer is not defined in SPI, therefore there are different configuration possibilities, which are defined by the parameters Clock Polarity, Clock Phase and Data Order. The default setting is SPI mode 3 with DORD = 0. The commands #XCS and #HSP (master interface)) set the mode 0..3. Alternatively the command can be stored directly into the boot file <start.emc>.

Mode	CPOL	CPHA	DORD (0) - MSB First	DORD (1)
0	0	0	*1 	
1	0	1		
2	1	0	*1 	

3	1	1	

You may check the current settings in Boot menu.
The maximum clock frequency is 1 MHz. The module needs some time to prepare data for transfer. That means a wait cycle (no activity on the SCK-line) of at least **50 µs** is required before reading data.

***1 ATTENTION:**
As a slave in mode 0 and mode 2, the EA uniTFTm requires a negation of the SS signal after each byte. Permanent activation is not permitted.

I²C

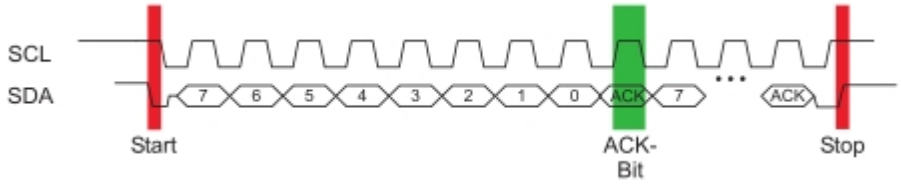
I²C stands for **I**nter-**I**ntegrated **C**ircuit and is a serial data-bus developed by Phillips.

The EA uniTFT provides one I²C interface: As default the interface is parametrized as Slave and used to communicate with the display. All data sent to the display are interpreted as a command (with and w./o. Small-/Short-Protocol). Would you like to send and receive any data via I²C to any other device like temperature sensor, then you have to use the Master functionality. Those are handled via #H commands. That means the slave functionality is deactivated.

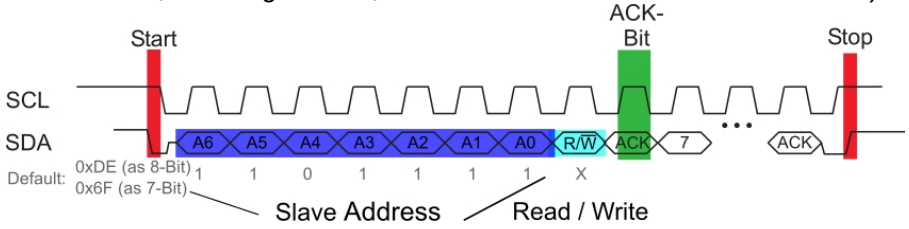
The bus is a Master-Slave implementation and needs 2 signal lines:

- SCL (**S**erial **C**lock **L**ine)
- SDA (**S**erial **D**ata **L**ine)

The electrical specification defines that both lines are terminated with a pull-up resistor at VDD, because all devices connected to the bus have open collector outputs. The bus clock is always given by the master, which controls the entire communication:



After the start condition, the slave address follows. In this case, bit 0 is the so-called R/W bit and determines whether the slave should be read (1) or data is transmitted (0). The data exchange takes place until the master executes the stop condition. More detailed information can be found in the I²C specification. The default I²C bus address is 0xDE (as 8-Bit address, including R/W bit, as 7-Bit address without R/W bit it's 0x6F) when writing to the slave unit.



The command #XCI and #HIP (master interface) can change the I²C write address to any other address. Alternatively the command can be written directly into the boot file <start.emc>.

You may check the current settings in Boot menu.
The maximum frequency in slave mode is 400 kHz, the master interface is capable up to 1 MHz.The module needs some time to prepare data for transfer. That means a wait cycle (no activity on the SCL-line) of at least **50 µs** is required before reading data.

USB

The **U**niversal **S**erial **B**us is a serial bus system for interfacing a PC with other peripherals. It's based on differential data transfer. The bus topology is a strict master-slave communication (Exception: On the Go devices). In the case of EA uniTFTm-Series the PC/Master needs to coordinate the communication. The module has a CDC (Communications Device Class) and is found by Windows PC's as a virtual COM-Port:

Description	Value
Device Class	2
USB Vendor ID	0x2DA9
USB Product ID	0x2454
Device description	EA uniTFT

To program the module, adjust settings or to perform initial tests, we recommend using the USB interface. It's easy to connect, the transfer rate is fast and no interface parameters need to be specified. The driver for Windows can be downloaded on our web-page: http://www.lcd-module.de/fileadmin/downloads/EA_CDCdriver_V5_3.zip

Attention:
A protocol has to be used in USB CDC mode. It's impossible to use the USB interface without a protocol, which means DPROT must not be set to GND (ZIFF pin 14, EA uniTFTmXXX-BTC primary connector pin 22). The high-speed connection of USB leads to buffer overflow, which are prevented by the protocol.

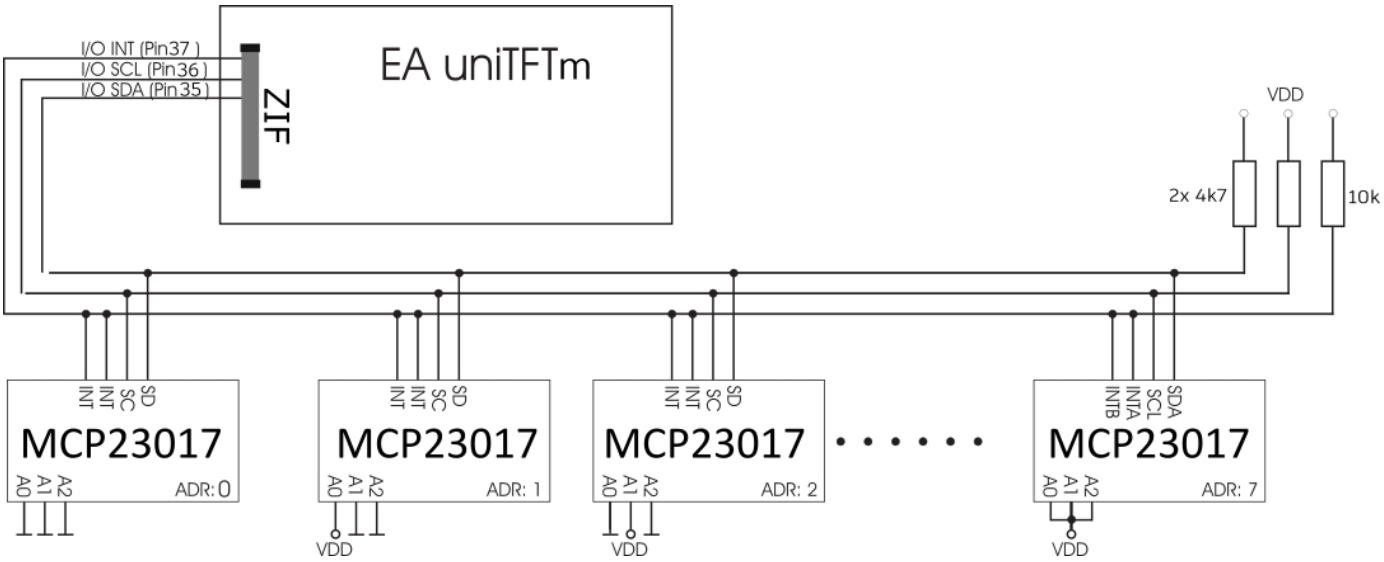
Touchpanel

The modules all do have a optically bonded capcitive touchpanel (PCAP), which is used for mounting, too. By touching the display you can enter data and adjust settings via menus or bar graphs. The labelling of the "keys" is flexible and can also be changed during runtime (different languages, icons). The drawing of the individual "keys" as well as the labelling is completely handled by the built-in software.

I/O - digital in- and outputs

The module has 8 digital I/Os (CMOS level, non-floating). The input range is 0... 3.3 V. All 8 I/Os have a weak pull-up at 1 MΩ and are set as inputs after reset.
Remark: The logic is not designed for time-critical operations; i.e. it is not a real-time operating system.

By using one or more external (max. 8) MCP23017-E (16 I/Os per IC), the total number of I/Os can be expanded up to 136. Therefore the port-expanders are connected to pins 35-37 (see application example). Because interface is I²C bus, 2 pull-up resistors need to be added (4k7):



The maximum power of the MCP23017-E is 700 mW in total. The maximum current load for a single pin is 25 mA, which makes it possible to directly operate a low current LED. If a higher load is required, the I/O current must be amplified with suitable circuitry, e.g. through an external transistor. For more details see Electrical characteristics

The overview of the software commands for the I/Os can be found under the chapter 'I/O Port'.

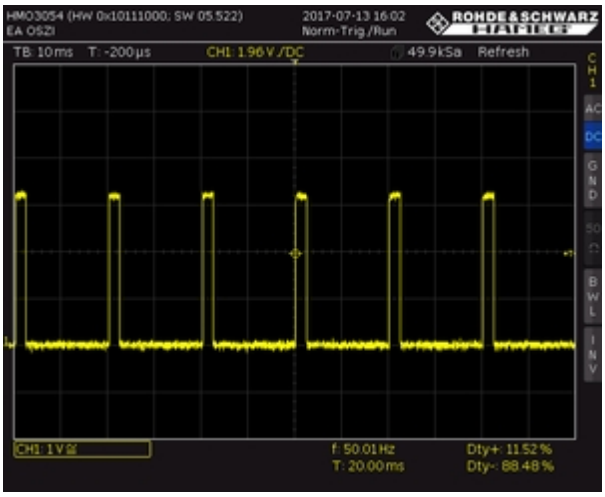
Analogue input

The module uses 4 analogue inputs with a resolution of 12 bit and an input range of 0 V...3.0 V. The input range can be arbitrarily expanded with the help of external voltage dividers or amplifiers. Every single input is referenced to GND and has an input resistance of about 12.5 KΩ. The absolute accuracy is 11 bits, as reference 3.0 is used.
This enables the display to measure analogue voltages, e.g. to display or save the values for further processing. The exceeding or undershooting of a threshold can also be used to trigger an alarm.

The overview of the software commands for the analogue inputs can be found under the chapter 'Analogue Input'

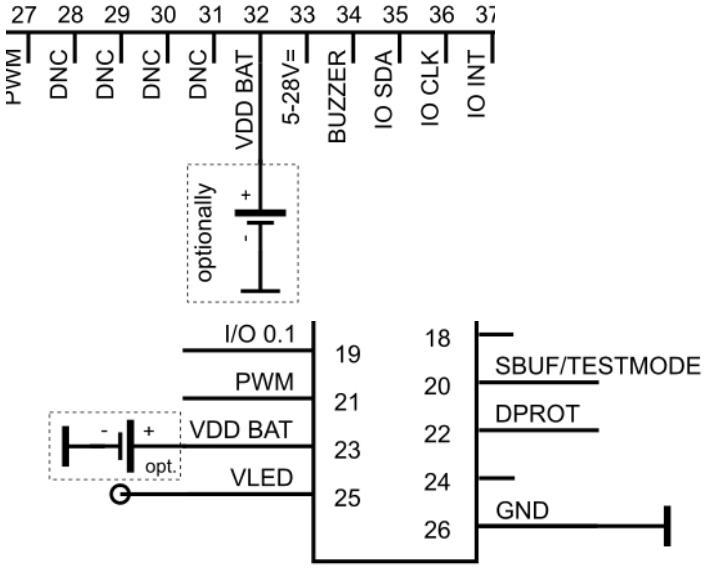
PWM output

The module has the option of controlling external components via a PWM signal (pulse width modulation). At constant frequency (adjustable from 20 Hz to 1.5 MHz #HFO), the duty cycle of a rectangular pulse is changed. Modulation changes the ratio between the on- and off-time and thus the characteristics of the output signal. In this way, electromechanical components such as motor or lamp can be driven or even a quasi-analogue voltage can be generated. The variation of the duty cycles supports a low engine speed/voltage with a short start-up time or a high motor speed/voltage with a long start-up time. The output levels are at 0V and VDD.



Timer / RTC

The EA uniTFTm-Series has a built-in a RTC. In addition to providing time-stamps for log files, the time and date can be displayed on the screen. Depending on the location it may be necessary to set the device to local time (#WTD).
To keep clock running even when power supply (3.3V) is switched off, there's a need to install an externally battery: ZIFF connector pin 32 or primary connector MicroMatch pin 23.



Due to component tolerances and temperature fluctuations, deviations of up to 0.02% are possible. The deviation can be reduced by repeatedly adjusting the time (#WTD) after a while.

Memory

The module has a built-in flash memory. The size is 60 MByte.
This memory is used to store all data, whether generated at runtime, e.g. log files, or pre-loaded as project data, such as macro files, pictures, animations and icons.

Attention:
Flash memories have limited erase / write cycles due to their design. The memory module used in the uniTFTm can typically safely execute 100,000 cycles. In order to write data, a block of memory may have to be erased, typically 30 ms are required for erasing, but it can take up to 400 ms also! This must be taken into account in the macro sequence when write file commands are executed.
Data are stored even after power-off.

Electrical specification EA uniTFTm070-ATC

Value	Condition	min.	typ.	max.	Unit
Single supply current VDD+VLED @3.3 V	Backlight 0%		240		mA
	Backlight 100%		890		mA
	Backlight 150%*)		970		mA
Separate power VLED only @5V	Backlight 100%		340		mA
	Backlight 150%*)		560		mA
Supply current USB (5 V)	Backlight 0%		187		mA
	Backlight 100%		622		mA
	Backlight 150%*)		1000		mA
Single supply current @24 V	Backlight 0%		43		mA
	Backlight 100%		135		mA
	Backlight 150%*)		202		mA
Brightness after PCAP	Backlight 100%		780		cd/m²
	Backlight 150%*)		1108		cd/m²

*) *not recommended for continuous operation*

Electrical specification EA uniTFTm101-ATC

Value	Condition	min.	typ.	max.	Unit
Single supply current VDD+VLED @3.3 V	Backlight 0%				mA
	Backlight 100%				mA
	Backlight 150%*)				mA
Separate power VLED only @5V	Backlight 100%				mA
	Backlight 150%*)				mA
Supply current USB (5 V)	Backlight 0%				mA
	Backlight 100%				mA
	Backlight 150%*)				mA
Single supply current @24 V	Backlight 0%				mA
	Backlight 100%				mA
	Backlight 150%*)				mA

Brightness after PCAP	Backlight 100%				cd/m²
	Backlight 150%*)				cd/m²

*) *not recommended for continuous operation*

Electrical specification (general)

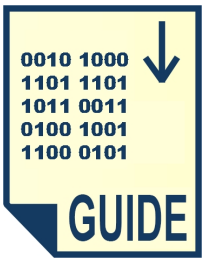
Value	Condition	min.	typ.	max.	Unit
Operating temperature		-20		70	°C
Storage temperature		-30		80	°C
Storage humidity	@ 60°C			90	% RH
Operating voltage	VDD	3.1	3.3	3.5	V
RTC backup battery voltage	VDD BAT	2.4		3.6	V
Operating voltage	VLED	3.3		7.5	V
RTC current	@VDD BAT=3.3V		18		µA
Input low voltage (except USB,I/O)		0		0.3*VDD	V
Input high voltage (except USB,I/O)		0.7*VDD		VDD	V
Output low voltage (except USB,I/O)		-	-	0.15	V
Output high voltage (except USB, I/O)		VDD-0.15	-	-	V
Output current I/O			2		mA
I²C-bus pull-up				10	kΩ

SOFTWARE - OPERATION OF EA UNITFTM-SERIES

The presentation on the display is based on the given commands. The commands can either be transmitted at runtime via one of the serial interfaces or combined on the internal memory in so-called macros and stored permanently. With the help of the commands, graphic objects are created. These objects have different properties, like color, position and built-in actions. These properties can be changed at any time, for example a string or the position of a touch-sensitive button can be changed.
All conceivable objects can be arbitrarily placed, moved and deleted. Windows font sets are stored directly in the display's memory. Thanks to automatic ASCII and Unicode switching, a wide variety of systems are supported flexibly, Chinese characters included. Elegant effects like fading in or out are already integrated. Style sheets can be used to create consistent designs. Images JPEG, PNG and many more (also transparent) can be integrated.

Command Set

There are many Graphic Commands built-in to create a nice screen. With individual macros and logical functions you create an intelligent control unit. All software commands and it's description can be found in separate manual:



Building a screen with the WYSIWYG editor

The most quick and simple way to create screen content is by use of uniTFTDesigner. uniTFTDesigner is a simple to use WYSIWYG tool for Windows. You build your screen by drag-and-drop and editing properties. It includes project management and a documentation tool.
There's also simulator built-in that makes it simple and fast to test your project directly.

Here you can download uniTFTDesigner for free.

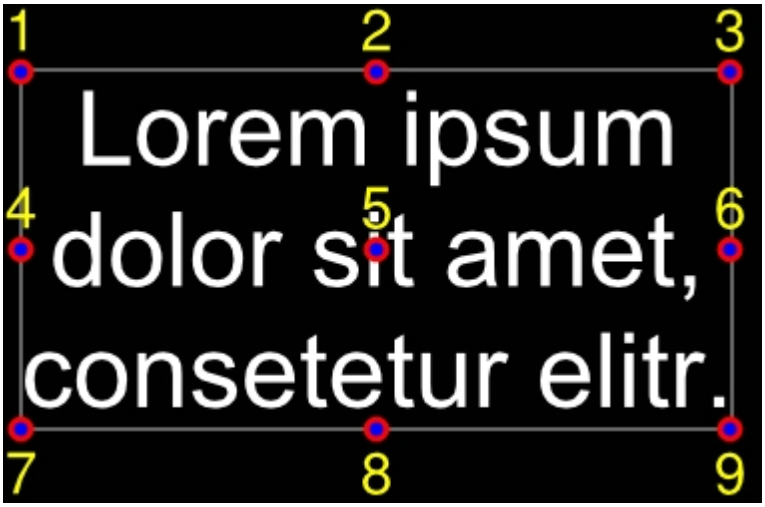
Objects

Every picture, text element and button is a so-called object. Each object got its own, individual object ID, which makes it uniquely identifiable. The object ID can be used to change the properties of an object at any time (size, position...). You can use 0 as ID for creating simple graphical objects. These objects are rendered directly to the background and aren't editable and manipulable any more. If you assign an already existing object ID to a new object, the previous object will be overwritten.
Commands for object management can be found here.

Object position / Anchor

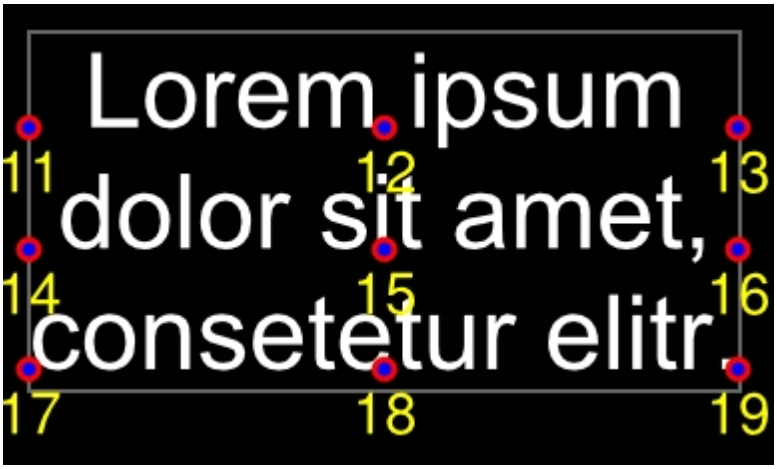
General anchors

The position of an object is based on the coordinates (origin: bottom left edge) related to the object anchor. Each object has 9 fixed anchors. Transformation on the object (e.g. rotation or shear) will be applied to the active anchor.



Strings and anchors

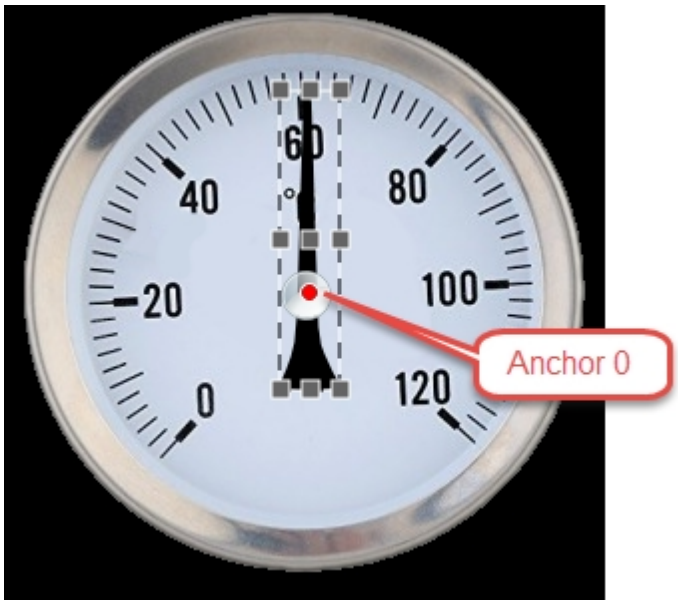
Strings have additional 9 anchors used to align objects (e.g. an underscore line) to the text base line.



Special case: Anchor 0

Each object has additionally a freely definable anchor. For circles, ellipses, and stars, the object anchor 0 is the

construction point.



Example: The pointer should rotate around the centre of the circle. The pointers 9 standard anchors (shown in dark grey) are not useful in this case because none of the defaults are located in the right position. The anchor 0 can be placed pixel-precise (#OAS) as shown, and this custom location marks the correct rotation point for the pointer object.

Styles / StyleSheets

Styles can be used to create consistent design. There are
- DrawStyles
- TextStyles
- ButtonStyles

Before placing any graphic object or text object, a DrawStyle or a TextStyle need to be defined. A DrawStyle defines the pen type and a fill color and the TextStyle the font and it's size.

DrawStyle:
Color, gradients, pattern and pen for (out)lines are defined in a DrawStyle.
TextStyle:
The appearance of a string is defined in a TextStyle. A TextStyle is based on a DrawStyle for color and some font specification for size, alignment and spacing.
ButtonStyle:
Touch buttons and switches are defined by a ButtonStyle, which consists of a TextStyle for labeling and DrawStyles for background painting.

ColorRamp:
Filling an object can be done with solid color or with some gradient. Those gradient and its colors are defined in ColorRamps and can be used linear or radial.

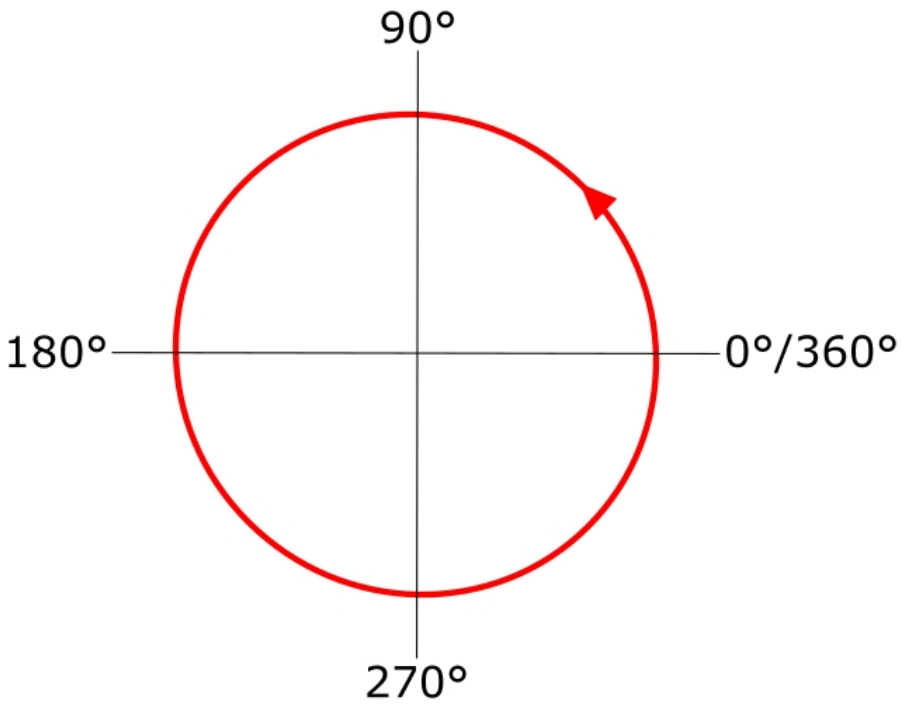
The Windows tools uniTFTDesigner supports StyleSheets that contain a collection of several Draw, Text, and ButtonStyles ans also ColorRamps.

The commands related to styles and colorramps can be found here.

Coordinate system and angle

The coordinate system refers directly to the display resolution of the module with the origin 0|0 placed in the lower left corner of the display. For example the EA uniTFTm070-A has a drawing field of 1024 x 600 dots. Valid coordinates for this display are 0..319 and 0..239 hence.

Angles are given in the mathematical sense of rotation (counter-clockwise). 0° is horizontally right. Besides instruments rotation is available in 90° steps:

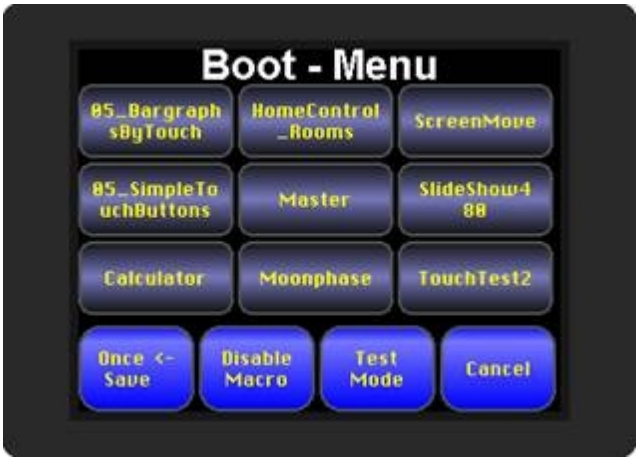


Multi language - String files

In an increasingly interdependent world of international assignments, supporting multiple languages is a must. The EA uniTFTm-Series with its unicode support is part of the solution. Without unicode it's basically impossible to work with Chinese characters e.g..
The second part of supporting internationalization are string files: these text files provide a database of strings to be displayed. In macro files, strings are referenced by an index, then at runtime this index is replaced with the corresponding text taken from the string file.
Further details can be found by looking at the command description under #VFL or the examples.

Boot menu

Multiple projects can be stored on the integrated memory. The project which is started automatically is defined using the "start.emc" file. To load a different project, the start file need to be updated, or on touch enabled panels, a project can be selected via the boot menu:
When switching on the device (or after hardware reset), wipe over the touch panel several times in short interval.



To avoid mis-use by the operator, the boot menu can be deactivated. For this purpose, an empty file named "bootmenu.off" must be placed in the root directory of the memory. This can be done using mass storage mode and Windows Explorer to transfer the file, or directly via uniTFTm commands:
#FWO</bootmenu.off>
#FWC

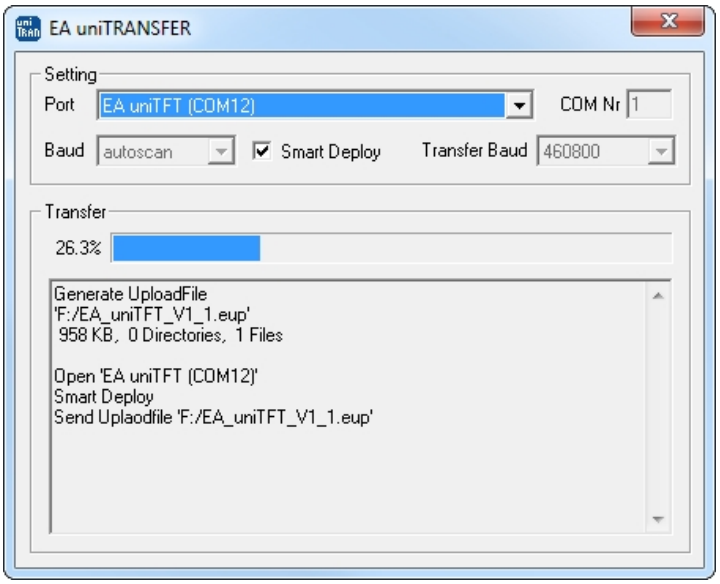
In addition to project selection, the boot menu offers the option to start test mode, or to display information about the module. It provides the version, protocol status, baudrate, SPI mode, I²C bus address etc

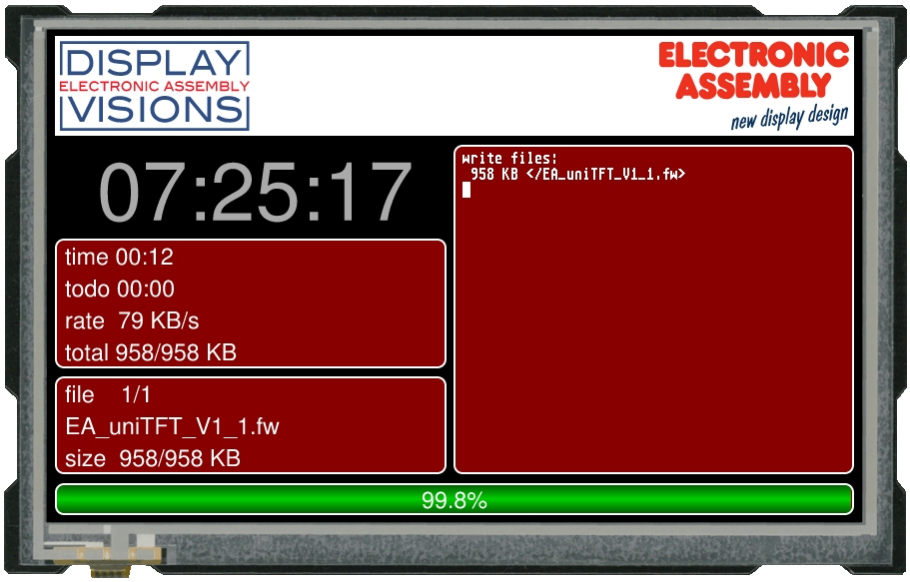
Firmware-update

To use the latest features of the EA uniTFTm-Series, it might be necessary to update the internal firmware of the module.

Firmware-update via serial interface and Windows PC:

- Save the firmware file (e.g. EA_uniTFTm_V1_1.fw) to your local drive
- Connect the EA uniTFT with your PC
- Start uniTRANSFER.exe (found in the Simulator_and_Tools folder of the uniTFTDesigner installation) and select the correct serial interface to the EA uniTFTm.
- Drag'n'Drop the firmware file to the EA uniTRANSFER window.





- After transferring the data, a manually reset needs to be performed, then the firmware will be loaded automatically after restart. **Attention: Please do not switch off the module while updating.**

Firmware update via serial interface

The firmware file also can be transferred to EA uniTFT with any system. To do this, transfer the contents of the * .fw file 1:1 (with protocol in packets) to the EA uniTFT. The transfer progress will become visible on the display module. After successful transfer, a data check will be done automatically. If the data is correct, the update starts automatically.

Attention: Please do not switch off the module while updating.

Filetypes

To use image, font or sound files, they must be converted. This is done automatically by using the uniTFTDesigner software by including the files there.
The display works with the following file types:

File type / Extension	Description
*.evg	vectorized picture
*.epa	animated picture
*.epg	bitmap picture
*.esd	sound file
*.evf	vektorized font
*.epf	bitmap font
*.epi	bitmap gauge
*.emc	macro
*.txt	string file
*.efl	simulation of internally FLASH

Other file formats can be stored in the internal memory, but cannot be used.

Limitations

Here you will find the maximum number of each:

Objects	32,768
ColorRamps	100
Drawstyles	100
Dash Pattern	-
Array	200
Register	200
Stringregister	200
String length	200
Stringconstant files simultanously	8
Stringconstant definitions	1,000
StringBox lines	500
StringBox character	32,768
Process macros	10
Macro marker	100
Menü Items	200
Count of edges (star, n-edge)	40
Filename length incl. path	255

In addition, the internal memory (RAM and FLASH) might already limit beforehand.

DIFFERENCES TO UNITFT-SERIES

Checklist for replacing an EA uniTFT-module with an EA uniTFTm-module.

1. Serial master interfaces

The EA uniTFT-series provides RS232, SPI and I²C as dedicated master interfaces - totally separated from the slave interfaces. In contrast the EA uniTFTm-Series provides the serial interfaces as shared ones, i.e. RS232 can be used as Slave or Master.

2. Inputs and outputs

The EA uniTFT-series has 16 I/Os integrated, optionally expandable to 125. The EA uniTFTm-Series provides 8 I/Os, optionally expandable to 136.

3. Sound output

The EA uniTFT-series has the possibility to play short jingles from converted mp3 or wav files. An external speaker is needed. In contrast the EA uniTFTm-Series has the possibility to play notes (5 , 6 octave). The speaker is integrated.

4. Video input

The EA uniTFT-series provides a video input for PAL, Secam and NTSC. The EA uniTFTm-Series does not support any external video signal.

5. Flash memory (SD-Card)

The EA uniTFT-series uses a micro SD-Card to store project data, like pictures, screens, sounds and even data logging functions. That means the memory is replaceable and can be used with USB as MassStorage device. In contrast the EA uniTFTm-Series has an integrated flash memory of 60 MByte to store project data. To read back logged data to PC, you need a windows application to request and store the data. You can use the tool uniExplorer free of charge.

6. Analog inputs

The EA uniTFT-series can measure the whole supply range from GND to VDD. To improve measurements the EA uniTFTm-Series uses an internally 3.0V generator to be independent from fluctuating and load-dependent supply voltage. But the range is reduced to 3.0 V. The ADC resolution is the same. The input resistance is reduced from 1 MΩ to 12.5 KΩ.

7. RTC

The EA unitTFT-Series has a coin cell to retain the time in case of voltage drop or switch off. The EA uniTFTm-Series provides an input pin to connect an external battery.

8. Graphical appearance, vectorized graphic engine

The main difference is the vector support of the EA uniTF-Series. Vectorized graphic and fonts support lossless zoom and scales. That's why graphical functions like rotation, shearing and scaling are limited or have slightly different optical appearances. For the EA uniTFTm-Series, we recommend to store the needed font sizes (pixel fonts) to the internal flash.

9. Limitations (registers, styles, string lengths)

The EA uniTFT has more internal RAM memory. The limitations are different concerning registers and other constants like styles. Please find the limits for the EA uniTFT-series here. The limits for the EA uniTFTm-Series is here.

Helpfile

Date	Version	Info
13-JUNE-2024	1.0	First release