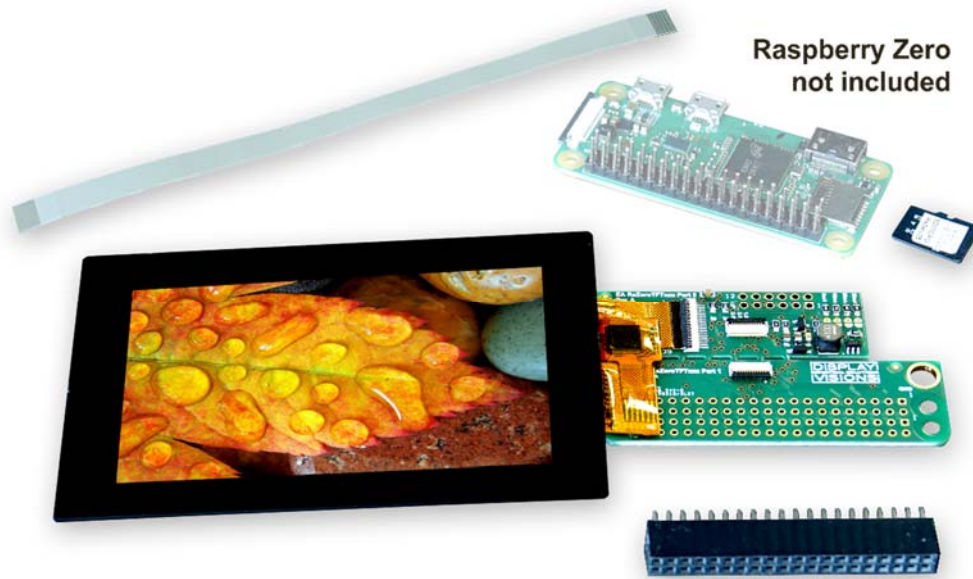


INCL. IPS TFT DISPLAY with optional CAPACITIVE TOUCH



Raspberry Zero
not included

EA RaZeroTFT035-AITC
with PCAP*

FEATURES

- HAT (HARDWARE ATTACHED ON TOP) FOR RASPBERRY PI ZERO
- IPS DISPLAYS FROM 1.5" TO 3.5"
- RESOLUTION FROM 240x240 TO 480x320
- SPI INTERFACE
- OPTIONAL CAPACITIVE TOUCH SCREEN, OPTICALLY BONDED
- I²C INTERFACE: SDA, SCL FOR TOUCH
- PWM LED DRIVER INCLUDED
- HIGH CONTRAST
- WIDE VIEWING ANGLE DUE TO AACS (ALL ANGLE COLOR STABILITY) DISPLAY
- HAT DETACHABLE, FOR FLEXIBLE DISPLAY MOUNTING

ORDERING CODES

- PCB HAT WITH 1.5" IPS DISPLAY, 240x240 dots, 32x35 mm
- PCB HAT WITH 2.0" IPS DISPLAY, PCAP, 43x65 mm
- PCB HAT WITH 2.8" IPS DISPLAY, PCAP, 58x84 mm
- PCB HAT WITH 3.5" IPS DISPLAY, PCAP, 65x100 mm
- HAT ONLY, FULLY ASSEMBLED

EA RaZeroTFT015
EA RaZeroTFT020TC
EA RaZeroTFT028TC
EA RaZeroTFT035TC
EA RaZeroTFT

* Pin socket 2x20 included, Raspberry Pi Zero not included

TABLE OF CONTENTS

Features.....	1
Ordering codes	1
Table of contents	2
Revision	2
General	3
Package Content	3
Hardware	3
Backlight current setting	4
Schematic	5
Software.....	6
Install ready images	6
Usage of DISPLAY VISIONS' ready images.....	8
Integration in existing Raspbian OS.....	9
Display and touch test, sample project	10
Dimension.....	12
HAT	12
HAT split into two parts	12
Display Dimension	12

REVISION

First issue		02/2023
-------------	--	---------

GENERAL

The EA RaZeroTFT is a display shield, suitable for the Raspberry Pi Zero (not included). It is capable of a range of small-sized IPS TFT-Displays by DISPLAY VISIONS. The size range starts with 1.5" and ends with 3.5". Optionally there are optically bonded capacitive touch panels available.

The Raspberry Pi Zero is a tiny computer with quad-core 64-bit ARM Cortex A-53 processor clocked at 1 Ghz and 512 MB of SD-RAM. Wireless LAN is also built in. Because of the tiny factor of 65 x 30 mm, DISPLAY VISIONS decided to create a suiting shield for it's small sized IPS TFT-series. The little Pi Zero is perfect for a range of IoT applications and other projects – like Smart Home.

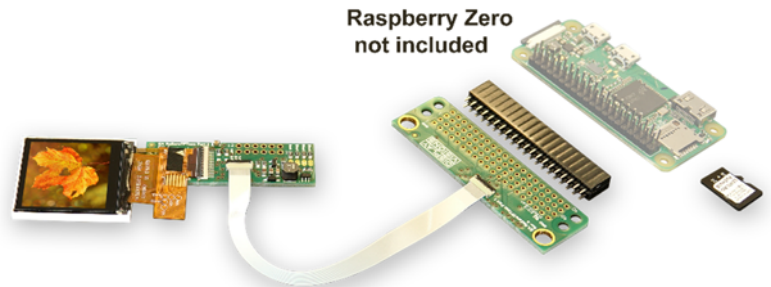
Even though there is a HDMI interface only, that means only large screens are supported directly. You often don't have the space or the need for a big screen. The small displays of the IPS TFT-series can be used to show status or many other information. The easy-to-use touch interface for user input extends the functionality. Thanks to the double IO-header, stacking of multiple shields and HATs is possible.

Another feature is flexible mounting of the display. Just break the PCB into two pieces and use the included FPC cable to have a flexible way of mounting.

PACKAGE CONTENT

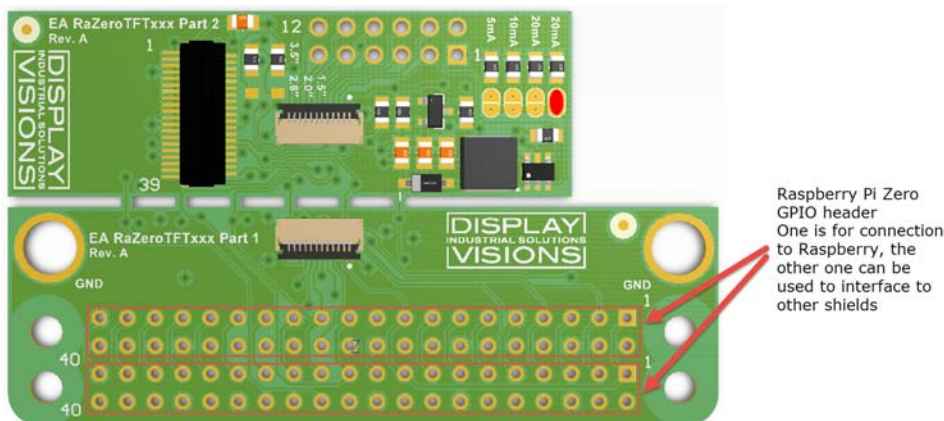
The package includes

- IPS display with or without PCAP
- HAT for direct mounting to Raspberry
- SD card incl. Raspbian, ready to run
- 2x20 socket
- 127mm Flexcable
- Note that Raspberry is not included



HARDWARE

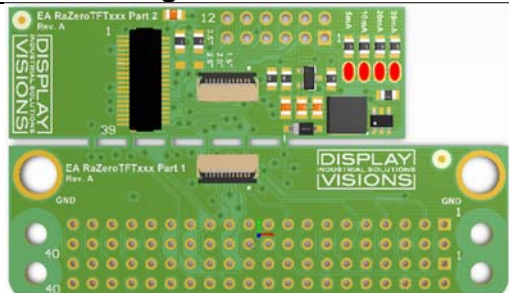
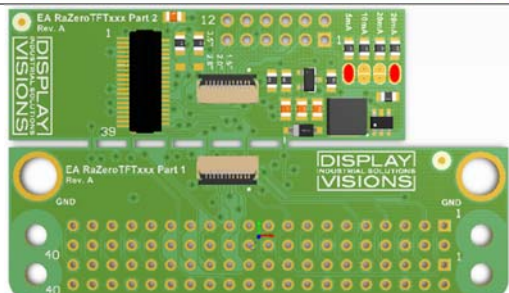
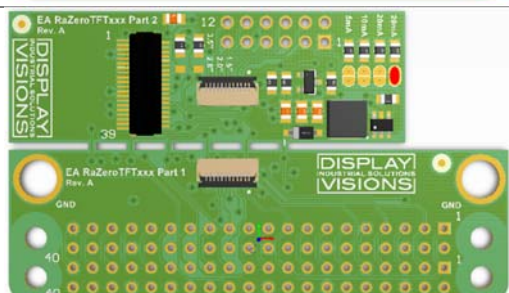
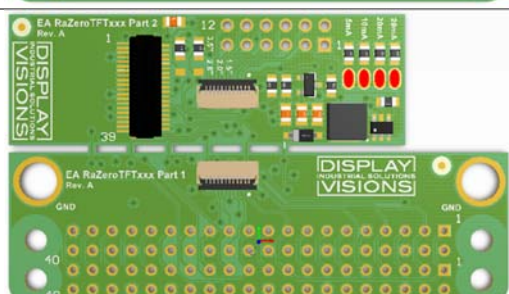
The EA RaZeroTFTxxx is a TFT display with adapter board suitable for Raspberry Pi Zero (not included). There are two equivalent pin headers to connect to Raspberry Pi Zero GPIO's:



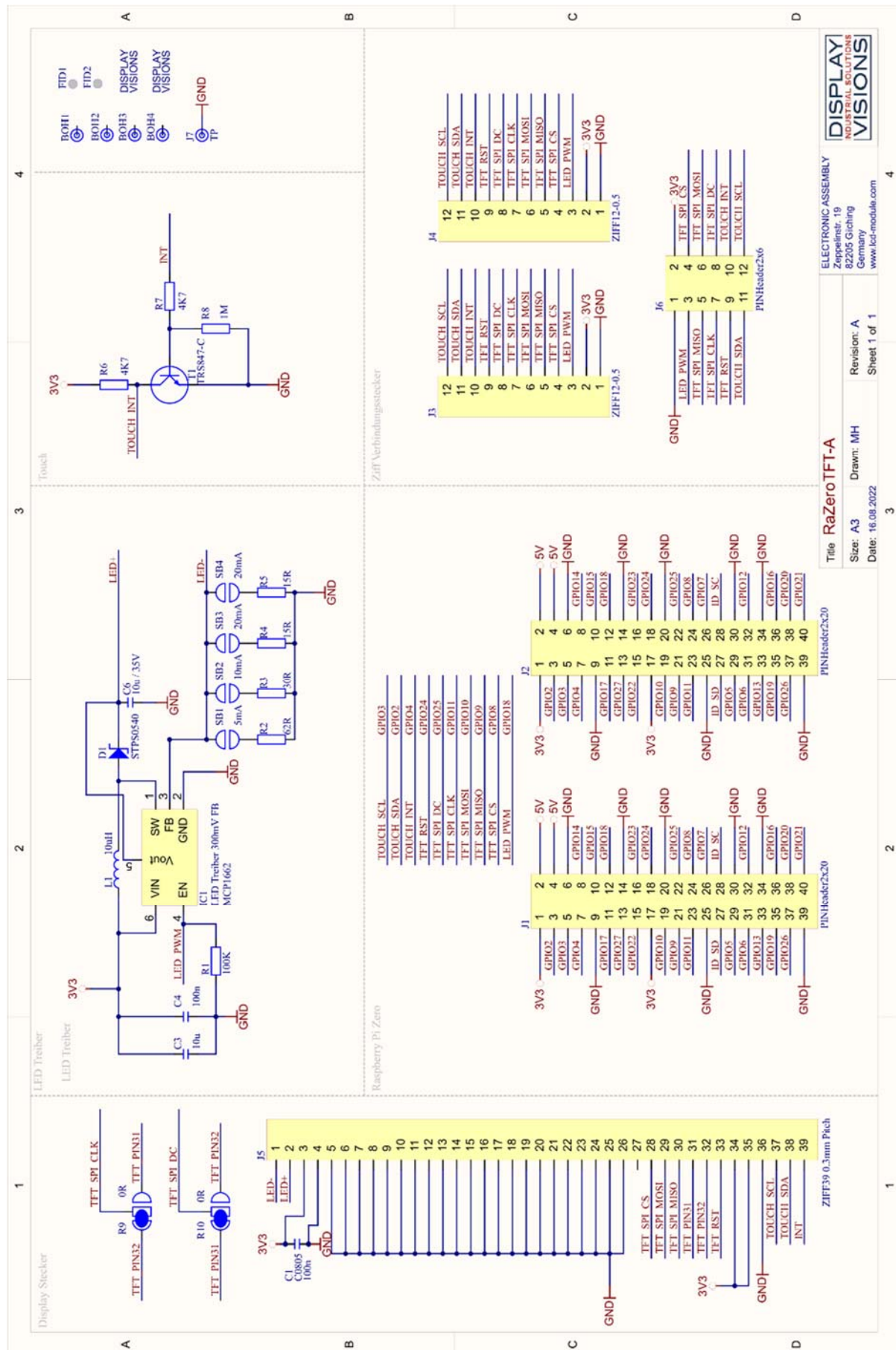
That means you can stack multiple shields / HATS. Make sure that the individual IOs do not interfere with each other. For display shield, GPIO 18 is used for Backlight, GPIO 9,10,11,24,25 is used for display, GPIO 2,3,4 is used for capacitive touch. See schematic for details.

BACKLIGHT CURRENT SETTING

Use solder bridges to set correct current

Connected display	Max. current	Solder bridges
EA TFT015-22AINN	80 mA (always on, PWM not supported)	
EA TFT020-23AIxx	25 mA	
EA TFT028-23AIxx	20 mA	
EA TFT035-34AIxx	60 mA	

SCHEMATIC



SOFTWARE

At the first path there is no software needed. Linux, especially Raspbian brings everything you need. Nevertheless, we provide overlay files and testcode to write content to display as well as reading touch events. Additionally, we provide SD-Card images, where the overlay files are already installed and the display and touch test are started at power on.

INSTALL READY IMAGES

Please download the image corresponding to your display from the following sources:

Part number	Part number display	Download link
EA RaZero015-AIxx	EA TFT015-22AIxx	https://www.lcd-module.de/fileadmin/downloads/development-service/Raspberry/Image_RaZero015.zip
EA RaZero020-AIxx	EA TFT020-23AIxx	https://www.lcd-module.de/fileadmin/downloads/development-service/Raspberry/Image_RaZero020.zip
EA RaZero028-AIxx	EA TFT028-23AIxx	https://www.lcd-module.de/fileadmin/downloads/development-service/Raspberry/Image_RaZero028.zip
EA RaZero035-AIxx	EA TFT035-34AIxx	https://www.lcd-module.de/fileadmin/downloads/development-service/Raspberry/Image_RaZero035.zip

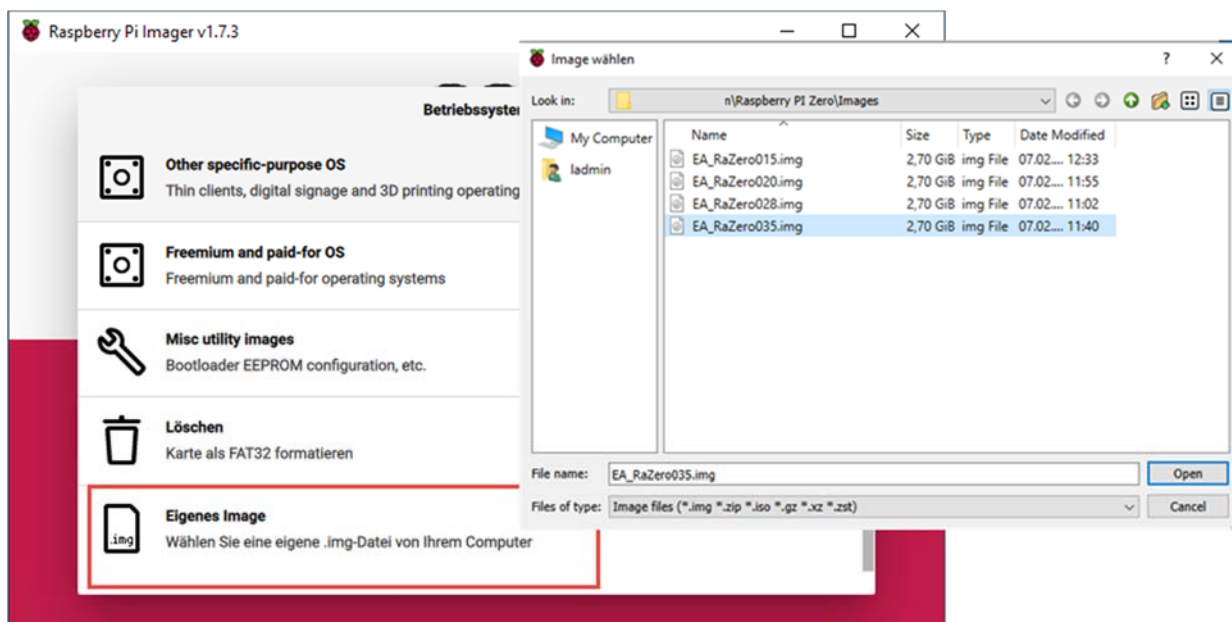
Please store to local disc and unzip.

We recommend using official Raspberry Pi Imager. This is the download link for windows:

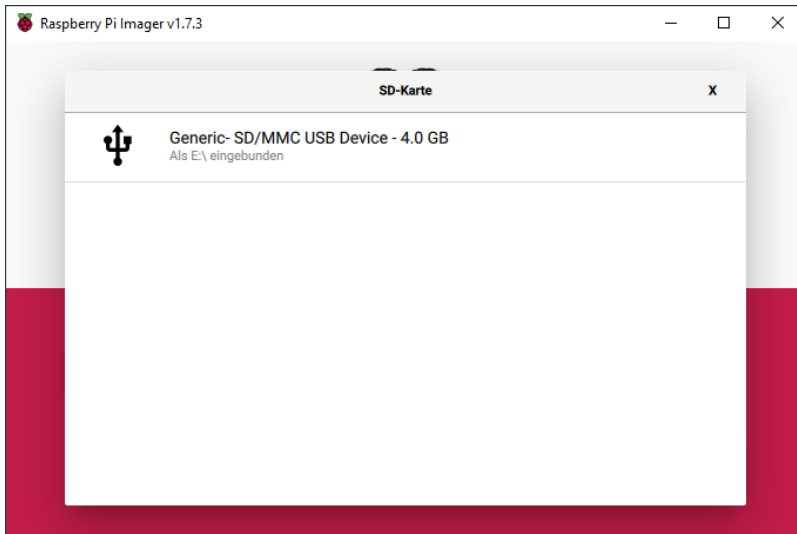
https://downloads.raspberrypi.org/imager/imager_latest.exe



To write image to SD Card, please select operating system ("Choose OS" and scroll down to the bottom (Own OS, image)).



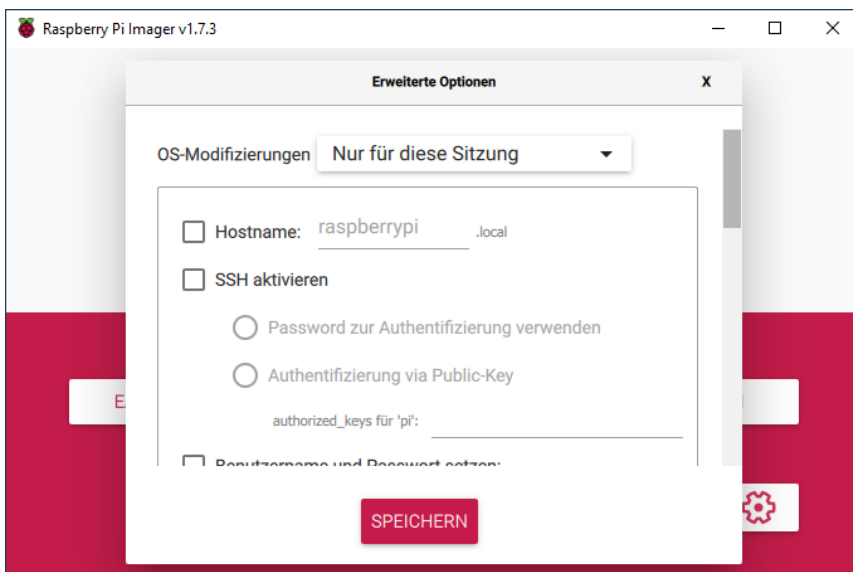
Second step is selecting correct SD-Card.



Third step is setting the options, like username, password and WiFi settings. If you don't change anything, the default is:

Username: 'pi'

Password: 'pi'



Last step is writing. It takes a while. After writing there is a verify and information that you can remove the card. Now you are ready to use it.

USAGE OF DISPLAY VISIONS' READY IMAGES

All Raspberry Pi Zero images start a slide show after system boot (boot time: ~ 1 minute).

You can easily change brightness (not EA TFT015-AINN)

```
pi@raspberrypi:~/DISPLAY_VISIONS/ $ ./Backlight/backlight -f 1000 -b 90
```

Type -h to see all optional parameters (f= fading time in ms, b=brightness in %).

If you have a display with touch, all touch events are directly drawn on the display as yellow dots.

INTEGRATION IN EXISTING RASPBIAN OS

1. Download source files (<https://www.lcd-module.de/fileadmin/downloads/development-service/Raspberry/RaZeroTFT.zip>) and unzip in folder "DISPLAY VISIONS":

```
pi@raspberrypi:~ $ mkdir DISPLAY\ VISIONS
pi@raspberrypi:~ $ cd DISPLAY\ VISIONS/
pi@raspberrypi:~/DISPLAY VISIONS $ wget https://www.lcd-module.de/fileadmin/downloads/development-
service/Raspberry/RaZeroTFT.zip
--2023-01-30 12:19:49-- https://www.lcd-module.de/deu/temp/EA_RaZeroTFT.zip
Resolving www.lcd-module.de (www.lcd-module.de)... 93.90.201.24
Connecting to www.lcd-module.de (www.lcd-module.de)|93.90.201.24|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 178369 (174K) [application/zip]
Saving to: 'EA_RaZeroTFT.zip'

EA_RaZeroTFT.zip           100%[=====>]
174.19K  759KB/s   in 0.2s

2023-02-01 09:50:25 (759 KB/s) - 'EA_RaZeroTFT.zip' saved [178369/178369]

pi@raspberrypi:~/DISPLAY VISIONS $ unzip EA_RaZeroTFT.zip
Archive:  EA_RaZeroTFT.zip
  creating: DisplayTest/
  inflating: DisplayTest/framebuffer.c
  inflating: DisplayTest/framebuffer.h
  inflating: DisplayTest/main.c
  inflating: DisplayTest/stb_image.h
  inflating: DisplayTest/stb_image_resize.h
  inflating: DisplayTest/touch.c
  inflating: DisplayTest/touch.h
  creating: Overlay/
  inflating: Overlay/tft015-22ai.dtbo
  inflating: Overlay/tft015-22ai.dts
  inflating: Overlay/tft020-23ai.dtbo
  inflating: Overlay/tft020-23ai.dts
  inflating: Overlay/tft028-23ai.dtbo
  inflating: Overlay/tft028-23ai.dts
  inflating: Overlay/tft035-34ai.dtbo
  inflating: Overlay/tft035-34ai.dts
  inflating: DisplayTest/DisplayTest
  inflating: Backlight/backlight
  inflating: Backlight/backlight.c
  inflating: Backlight/backlight.h
  inflating: Backlight/pwm
  inflating: Backlight/pwm.c
pi@raspberrypi:~/DISPLAY VISIONS $
```

2. Copy overlay files to system:

```
pi@raspberrypi:~/DISPLAY VISIONS $ cd Overlay/
pi@raspberrypi:~/DISPLAY VISIONS/Overlay $ sudo cp tft*.dtbo /boot/overlays/
pi@raspberrypi:~/DISPLAY VISIONS $ cd ..
```

3. Edit config text to activate SPI (for display), I²C (for touch) and start display overlay file

```
pi@raspberrypi:~/DISPLAY VISIONS $ sudo nano /boot/config.txt
```

Search for the following entries:

```
#uncomment to overclock the arm. 700 MHz is the default.
#arm_freq=800

# Uncomment some or all of these to enable the optional hardware interfaces
dtparam=i2c_arm=on
#dtparam=i2s=on
dtparam=spi=on

# Uncomment this to enable infrared communication.
#dtoverlay=gpio-ir,gpio_pin=17
#dtoverlay=gpio-ir-tx,gpio_pin=18

# Additional overlays and parameters are documented /boot/overlays/README

# Enable audio (loads snd_bcm2835)
dtparam=audio=on

# Automatically load overlays for detected cameras
camera_auto_detect=1

# Automatically load overlays for detected DSI displays
```

```
display_auto_detect=1

# Enable DRM VC4 V3D driver
#dtoverlay=vc4-kms-v3d
dtoverlay=tft035-34ai
#dtoverlay=tft028-23ai
#dtoverlay=tft020-23ai
#dtoverlay=tft015-22ai
max_framebuffers=2

# Disable compensat
```

Example for
EA TFT035-34AITC

Save config.txt (Ctrl+X → [Y]es)

After reboot the display is initialized and active and can be used. Please do not forget to switch backlight on (GPIO18 to high, see schematic for details).

DISPLAY AND TOUCH TEST, SAMPLE PROJECT

In the downloaded files, there are some sample codes. One project fires the backlight with PWM, another project shows a slide show. Touch inputs are directly drawn as pixel on the screen. The sample code is free to use. Please see *.c and *.h files for details.

1. Make projects executable:

```
pi@raspberrypi:~/DISPLAY VISIONS/ $ cd DisplayTest
pi@raspberrypi:~/DISPLAY VISIONS/DisplayTest $ chmod +x DisplayTest
pi@raspberrypi:~/DISPLAY VISIONS/DisplayTest $ cd ..
pi@raspberrypi:~/DISPLAY VISIONS/ $ cd Backlight/
pi@raspberrypi:~/DISPLAY VISIONS/Backlight $ chmod +x pwm
pi@raspberrypi:~/DISPLAY VISIONS/Backlight $ chmod +x backlight
```

2. PWM backlight needs installation of PIGPIO library:

```
pi@raspberrypi:~/DISPLAY VISIONS/Backlight $ sudo apt-get install pigpio
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libpigpio-dev libpigpio1 libpigpio1-armhf libpigpio1-dev libpigpio1-if1 libpigpio1-if2-1 pigpio-tools pigpio
The following NEW packages will be installed:
  libpigpio-dev libpigpio1 libpigpio1-armhf libpigpio1-dev libpigpio1-if1 libpigpio1-if2-1 pigpio pigpio-tools pigpio
0 upgraded, 9 newly installed, 0 to remove and 0 not upgraded.
Need to get 348 kB of archives.
After this operation, 1,029 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.raspberrypi.org/debian bullseye/main armhf libpigpio1 armhf 1.79-1+rpt1 [91.4 kB]
Get:2 http://archive.raspberrypi.org/debian bullseye/main armhf libpigpio-dev armhf 1.79-1+rpt1 [78.5 kB]
Get:3 http://archive.raspberrypi.org/debian bullseye/main armhf libpigpio1-if1 armhf 1.79-1+rpt1 [20.5 kB]
Get:4 http://archive.raspberrypi.org/debian bullseye/main armhf libpigpio1-if2-1 armhf 1.79-1+rpt1 [24.4 kB]
Get:5 http://archive.raspberrypi.org/debian bullseye/main armhf libpigpio1-if-dev armhf 1.79-1+rpt1 [76.9 kB]
Get:6 http://archive.raspberrypi.org/debian bullseye/main armhf pigpio-tools armhf 1.79-1+rpt1 [42.9 kB]
Get:7 http://archive.raspberrypi.org/debian bullseye/main armhf pigpio armhf 1.79-1+rpt1 [9,900 B]
Get:8 http://archive.raspberrypi.org/debian bullseye/main armhf pigpio armhf 1.79-1+rpt1 [3,564 B]
Fetched 348 kB in 1s (380 kB/s)
Selecting previously unselected package libpigpio1.
(Reading database ... 43422 files and directories currently installed.)
Preparing to unpack .../0-libpigpio1_1.79-1+rpt1_armhf.deb ...
Unpacking libpigpio1 (1.79-1+rpt1) ...
Selecting previously unselected package libpigpio-dev.
Preparing to unpack .../1-libpigpio-dev_1.79-1+rpt1_armhf.deb ...
Unpacking libpigpio-dev (1.79-1+rpt1) ...
Selecting previously unselected package libpigpio1-if1.
Preparing to unpack .../2-libpigpio1-if1_1.79-1+rpt1_armhf.deb ...
Unpacking libpigpio1-if1 (1.79-1+rpt1) ...
Selecting previously unselected package libpigpio1-if2-1.
Preparing to unpack .../3-libpigpio1-if2-1_1.79-1+rpt1_armhf.deb ...
Unpacking libpigpio1-if2-1 (1.79-1+rpt1) ...
Selecting previously unselected package libpigpio1-if-dev.
Preparing to unpack .../4-libpigpio1-if-dev_1.79-1+rpt1_armhf.deb ...
Unpacking libpigpio1-if-dev (1.79-1+rpt1) ...
Selecting previously unselected package pigpio-tools.
Preparing to unpack .../5-pigpio-tools_1.79-1+rpt1_armhf.deb ...
Unpacking pigpio-tools (1.79-1+rpt1) ...
Selecting previously unselected package pigpio.
Preparing to unpack .../6-pigpio_1.79-1+rpt1_armhf.deb ...
Unpacking pigpio (1.79-1+rpt1) ...
Selecting previously unselected package pigpio.
Preparing to unpack .../7-pigpio_1.79-1+rpt1_armhf.deb ...
Unpacking pigpio (1.79-1+rpt1) ...
Setting up pigpio-tools (1.79-1+rpt1) ...
Setting up libpigpio1-if1 (1.79-1+rpt1) ...
Setting up libpigpio1-if2-1 (1.79-1+rpt1) ...
Setting up libpigpio1 (1.79-1+rpt1) ...
Setting up pigpio (1.79-1+rpt1) ...
Setting up libpigpio1-if-dev (1.79-1+rpt1) ...
Setting up libpigpio-dev (1.79-1+rpt1) ...
Setting up pigpio (1.79-1+rpt1) ...
Processing triggers for man-db (2.9.4-2) ...
Processing triggers for libc-bin (2.31-13+rpt2+rp1+deb11u4) ...
pi@raspberrypi:~/DISPLAY VISIONS/Backlight $
```

3. Run display and touch test, together with backlight:

```
pi@raspberrypi:~/DISPLAY VISIONS/ $ sudo ./Backlight/pwm &
pi@raspberrypi:~/DISPLAY VISIONS/ $ sudo ./DisplayTest/DisplayTest &
pi@raspberrypi:~/DISPLAY VISIONS/ $ ./Backlight/backlight -f 1000 -b 90
```

If you want to run display test and backlight on boot, you have to install a crontab.

1. Open crontab for root, if you start the first time you have to set editor [1] for nano:

```
pi@raspberrypi:~/DISPLAY VISIONS $ sudo crontab -e
no crontab for root - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano        <---- easiest
 2. /usr/bin/vim.tiny
 3. /bin/ed

Choose 1-3 [1]: 1
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
@reboot (sleep 5; /home/pi/DISPLAY\ VISIONS/DisplayTest/DisplayTest &) >> /home/pi/DISPLAY\
VISIONS/DisplayTest/log.txt 2>&1
@reboot (sleep 5; /home/pi/DISPLAY\ VISIONS/Backlight/pwm &) >> /home/pi/DISPLAY\
VISIONS/Backlight/log.txt 2>&1
```

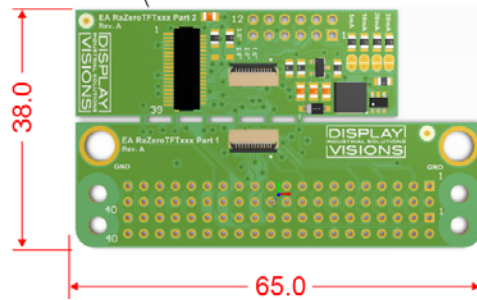
Both processes store their output to files to see errors, like framebuffer not correctly initialized, touch not found. The log files are in the same folder like the programs.

2. Save and close (Ctrl + X → [Y]es). You will see an information, that crontab installs a new crontab.
3. Reboot your system to test. The display is active after booting is finished. That means a Raspbian OS lite (booting to console) on a Raspberry PI Zero needs ~1 minute !

DIMENSION

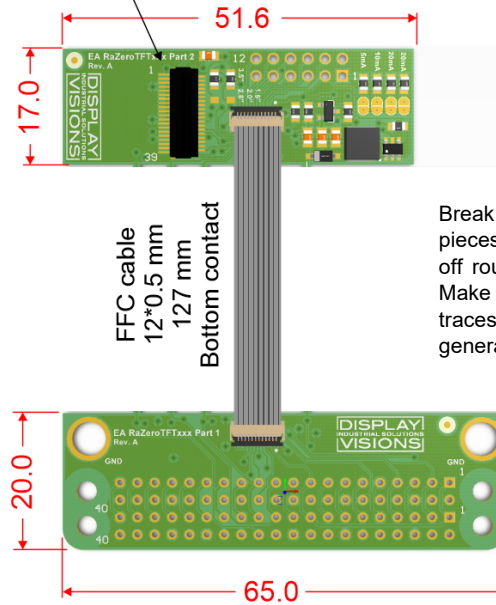
HAT

Insert display here:
Pin 1, bottom contact



HAT SPLIT INTO TWO PARTS

Insert display here:
Pin 1, bottom contact



DISPLAY DIMENSION

- 1.5" EA TFT015-22AI: <https://www.lcd-module.de/fileadmin/eng/pdf/grafik/TFT009-81AI.pdf>
- 2.0" EA TFT020-23AI: <https://www.lcd-module.de/fileadmin/html-seiten/eng/pdf/grafik/TFT020-23AI.pdf>
- 2.8" EA TFT028-23AI: <https://www.lcd-module.de/fileadmin/html-seiten/eng/pdf/grafik/TFT028-23AI.pdf>
- 3.5" EA TFT035-34AI: <https://www.lcd-module.de/fileadmin/html-seiten/eng/pdf/grafik/TFT035-34AI.pdf>