TFT-Display mit Tasmota benutzen

Tasmota bietet viele Vorteile auch in der Benutzung mit TFT-Displays, da man sehr
schnell einen beliebigen Text auf den Display darstellen kann. Ich nutze die Möglichkeit zur Anzeige der aktuellen Wetterdaten.

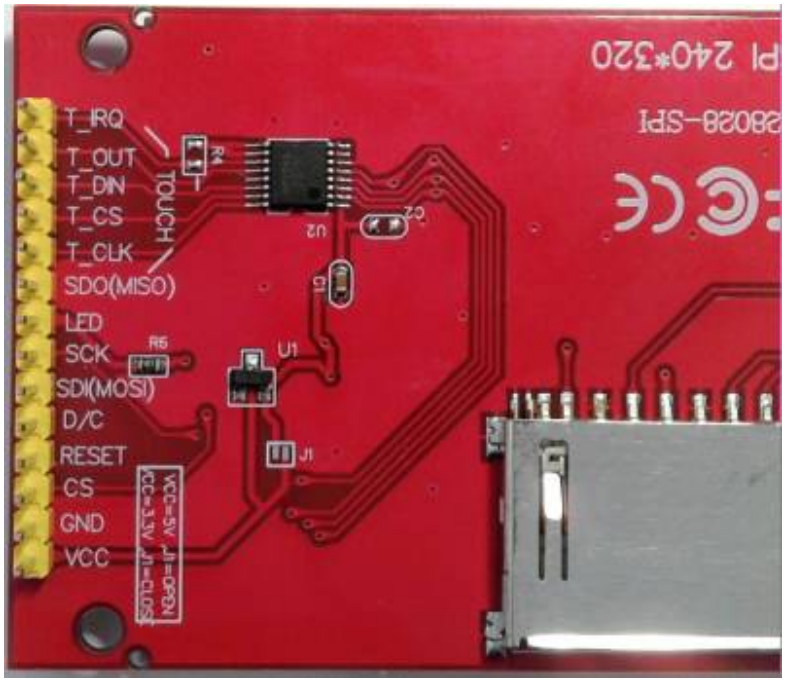
Hardware

Die Hardware ist ein „TFT ILI9341 2,8Zoll“ - ein Hersteller ist auf der Platine nicht zu finden.
Nur eine Kennung: `KMRTM28028-SPI` befindet sich als Text auf der Platine.

Vorderseite

Rückseite

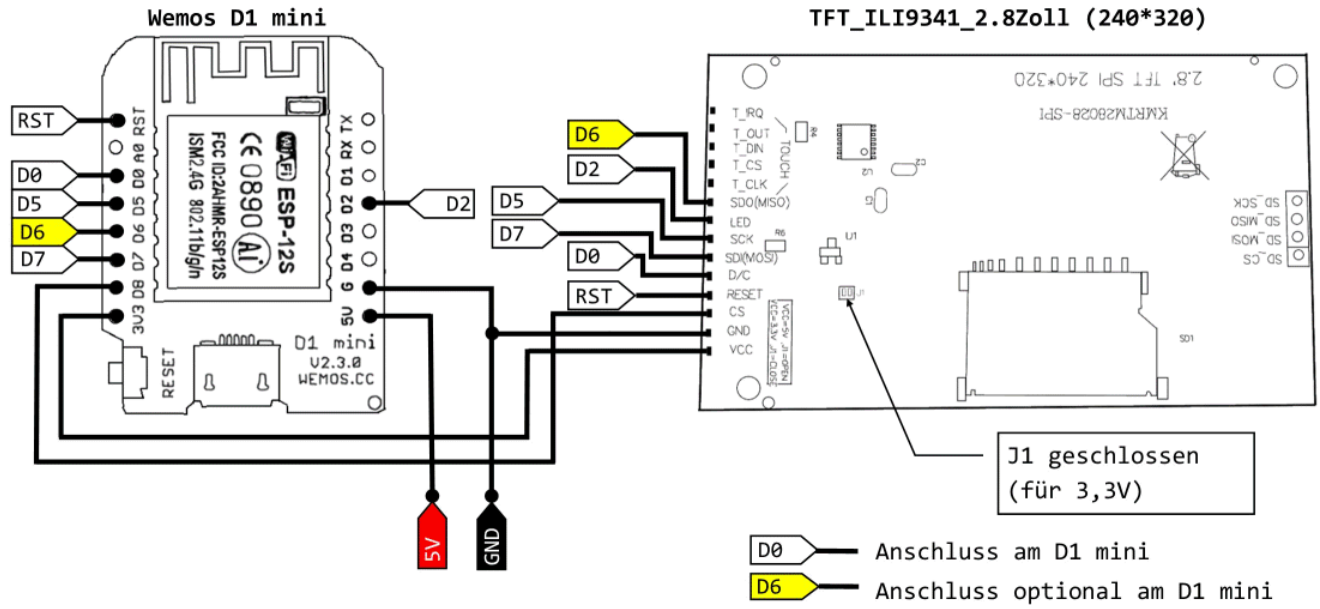
Auf der Anschlussseite (Rückseite) ist noch ein Hinweis zu finden, welche die Funktion der Brücke „J1“ zeigt.
Ist die Brücke offen wird 5V verwendet und im geschlossenen Zustand wird 3,3V verwendet. in meinen Fall habe ich
die Brücke geschlossen, da ich mit 3,3V arbeite. (Logiclevel auch bei 3,3V)



Rückseite
[Klicken zum Vergrößern.]

Schaltung

Zur Schaltung kommt ein „Wemos D1 mini“ (4MB) zum Einsatz.



Schaltplan
[Klicken zum Vergrößern.]

Anschlussbelegung:

D1 Mini	TFT-Display
5V (Versorungsspannung)	—
GND (Versorungsspannung)	GND (Versorungsspannung)
3V3	VCC

D8	CS
RST	RESET
D0	DC
D7	SDI (MOSI)
D5	SCK
D2	LED
D6[1]	SDO (MISO)[1]

Anmerkung 😕 \ [1] - Anschluss ist optional (Habe ich nicht anschlossen.)

Tasmota

Mittels „Tasmota PyFlasher“ wird die neue Firmware auf das Gerät gebracht.
Als Firmware wurde die Datei „`tasmota-display.bin`“ gewählt. Wie man Tasmota (WLAN) einrichtet findet man hier.
Welche Displays in ist in der Dokumentation bei Tastmota zu finden: Hier Es werden unterstützt:

- OLED SSD1306 (I2C)
- MATRIX Display
- **ILI9341** (SPI)
- EPAPER_29 (SPI)
- EPAPER_42 (SPI)
- OLED SH1106 (I2X)
- TFT ILI9488 (SPI)
- OLED SSD135 (SPI)
- TFT RA8876 (SPI)
- 7 segment display (I2C)
- TFT ST7789 (SPI)
- TFT ILI9342 (SPI)
- TFT SD1331 (SPI)
- 7-segment TM1637, TM1638 und MAX7219 Display
- TM1637

Einrichten

Die neue Firmware „Tasmota“ muss auf den Gerät noch eingerichtet werden. Hierzu klickt man auf „Einstellungen“
und dann auf „Gerät einrichten“. Als Gerätetyp ist „Generic (18)“ zu wählen. Siehe: Bild 01
Nach den Neustart des Gerätes sieht die Oberfläche wie im Bild 02 aus.

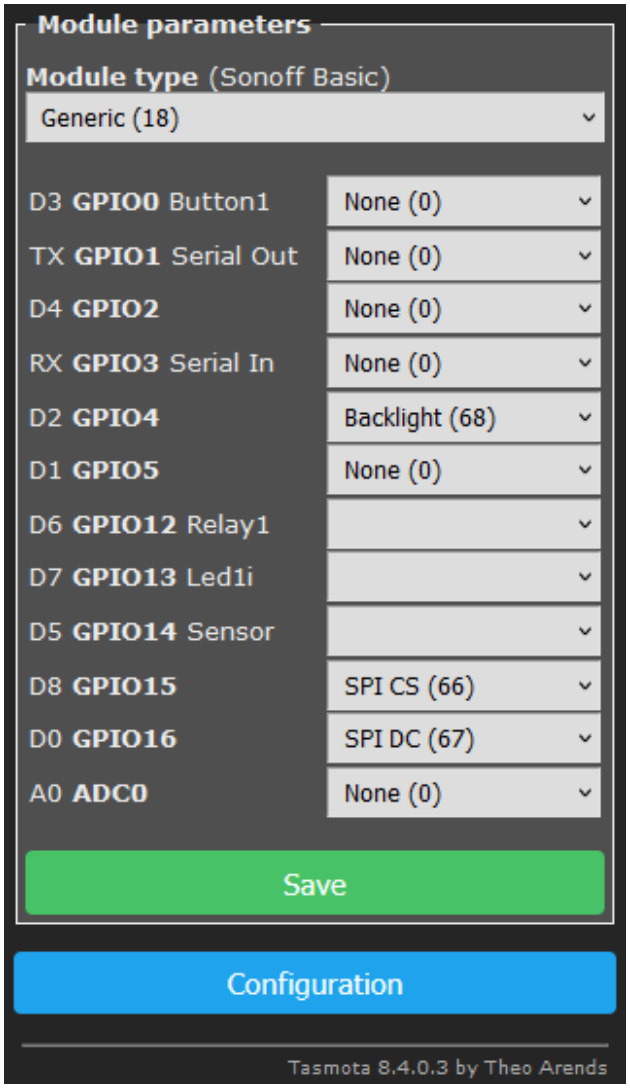


Bild: 01

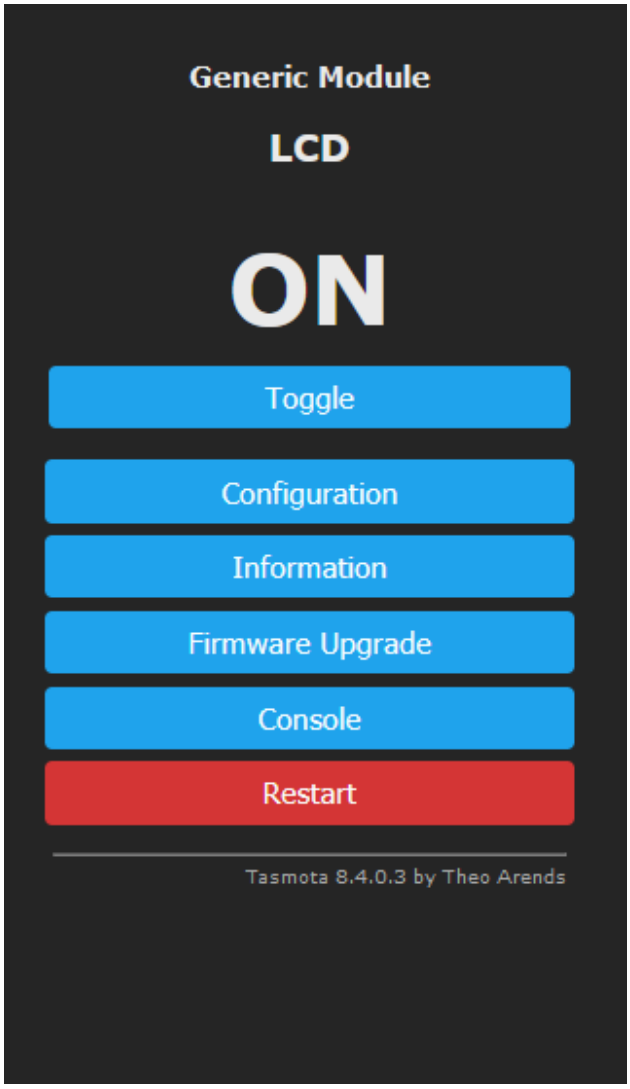


Bild: 02

Über die Schaltfläche „Toggle“ kann man die Hintergrundbeleuchtung des Displays ein- und ausschalten.
Natürlich kann man die Schaltung auch automatisch als Regel (zeitgesteuert) ausführen lassen.

Text darstellen

Neben den normalen Buchstaben und Zahlen können auch Sonderzeichen aus dem „GFXFont“ verwendet werden.
Das hier dargestellte Bild ist als Link aus der Dokumentation von Tasmota:

Link zum Bild.

Siehe auch: Codepage 437

Wetteranzeige

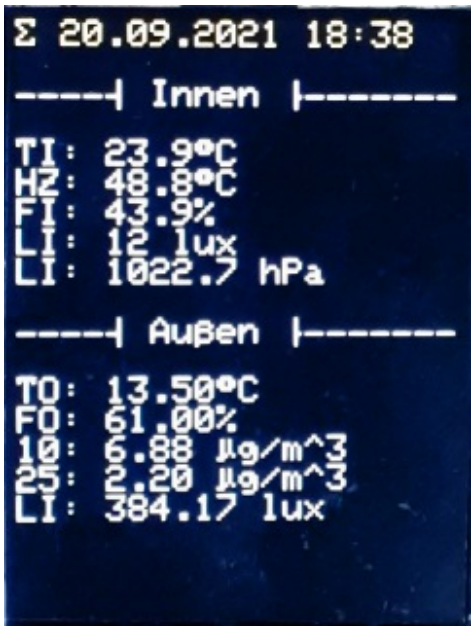


Bild: 03

Hier mal als Beispiel meine Wetteranzeige auf den Display.
Angezeigt wird die Werte für den Innenraum unter „—| Innen |—-“ und
die Werte vom Außenbereich unter „—| Außen |—-“.
Dabei ist auch die Anzeige der Feinstaubbelastung.

Angesteuert wird die Anzeige mit einen Python-Script.

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
# /-------------------------------------------------------------------------
-----\
```

```
#| Aktualisiert am: 02.06.2021  - 15:10 Uhr
|
#\-------------------------------------------------------------------------
-----/

import requests
import json
import datetime, time

class myvars:
  pass

class x1:
  pass

class x2:
  pass

class x3:
  pass

myprog = myvars()
myprog.date = time.strftime("%d.%m.%Y")    # aktuelles Datum
myprog.time = time.strftime("%H:%M:%S")    # aktuelle Zeit
# myprog.unixtime = int(time.time())        # als Unix-Zeitstempel
myprog.unixtime = time.strftime('%a, %d %b %Y %H:%M:%S %z')       # als
Unix-Zeitstempel
myprog.dtcvs = time.strftime("%Y/%m/%d") + ' ' + time.strftime("%H:%M:%S") #
Datum/Uhrzeit für CVS
myprog.altitude = 111                      # Hoehe in Meter

# -------[ Staubmesser - Außen ]-----------------------------------------
-------
sensor1 = x1()  # Staubmesser - Außen

content = requests.get("http://192.168.178.xx/feinstaub/feinstaub.json")
json = json.loads(content.content)
sensor1.temper = json['sensordatavalues'][2]['value']   # Temperatur
sensor1.humi = json['sensordatavalues'][3]['value']     # Luftfeuchte
#sensor1.pm10 = 0
sensor1.pm10 = json['sensordatavalues'][0]['value']     # Partikel PM10
#sensor1.pm25 = 0
sensor1.pm25 = json['sensordatavalues'][1]['value']     # Prttikel PM2.5

# -----[ Messstelle-01 ]-------------------------------------------------
-------

del json
import json

sensor2 = x2()
```

```
content = requests.get("http://192.168.178.xx/cm?cmnd=Status%208")
json = json.loads(content.content)
sensor2.temoere1 = json['StatusSNS']['DS18B20-2']['Temperature']
sensor2.temoere2 = json['StatusSNS']['BME280']['Temperature']
sensor2.temoere3 = json['StatusSNS']['DS18B20-1']['Temperature']
sensor2.humi0 = json['StatusSNS']['BME280']['Humidity']
sensor2.psa = json['StatusSNS']['BME280']['SeaPressure']
sensor2.lux = json['StatusSNS']['BH1750']['Illuminance']

# -----[ Raspi Zero ]-------------------------------------------------------
----

del json
import json

sensor3 = x3()
sensor3.illu = 0

content = requests.get("http://192.168.178.xx/zero/zero.json")
json = json.loads(content.content)
sensor3.illu = json['BH1750'][0]['Illuminance']         # Temperatur
# ---------------------------------------------------------------------------
-----------
# "TFT"

url = 'http://192.168.178.xx/cm?cmnd=DisplayMode%200'
response = requests.put(url, timeout=5)

url = 'http://192.168.178.xx/cm?cmnd=DisplayText%20[z]'   # Anzeige löschen
response = requests.put(url, timeout=5)

# Zeile 1
url = 'http://192.168.178.xx/cm?cmnd=DisplayText%20[C64800s2l1c1]~E4 ' +
myprog.date + ' ' + myprog.time[0:5]
response = requests.put(url, timeout=5)

# Zeile 2
url = 'http://192.168.178.xx/cm?cmnd=DisplayText%20[C0s2l3c1]~C4~C4~C4~C4~B4
' + 'Innen ' + '~C3~C4~C4~C4~C4~C4~C4~C4~C4'
response = requests.put(url, timeout=5)

# Zeile 3
url = 'http://192.168.178.xx/cm?cmnd=DisplayText%20[C0s2l5c1]TI: ' +
str(sensor2.temoere2) + '~F8C'
response = requests.put(url, timeout=5)

# Zeile 4
url = 'http://192.168.178.xx/cm?cmnd=DisplayText%20[C0s2l6c1]HZ: ' +
str(sensor2.temoere3) + '~F8C'
response = requests.put(url, timeout=5)
```

```
# Zeile 5
url = 'http://192.168.178.xx/cm?cmnd=DisplayText%20[s2l7c1]FI: ' +
str(sensor2.humi0) + '~25'
response = requests.put(url, timeout=5)

# Zeile 6
url = 'http://192.168.178.xx/cm?cmnd=DisplayText%20[s2l8c1]LI: ' +
str(sensor2.lux) + ' lux'
response = requests.put(url, timeout=5)

# Zeile 7
url = 'http://192.168.178.xx/cm?cmnd=DisplayText%20[s2l9c1]LI: ' +
str(sensor2.psa) + ' hPa'
response = requests.put(url, timeout=5)

# Zeile 8
url =
'http://192.168.178.xx/cm?cmnd=DisplayText%20[C0s2l11c1]~C4~C4~C4~C4~B4 ' +
'Au~E1en ' + '~C3~C4~C4~C4~C4~C4~C4~C4~C4'
response = requests.put(url, timeout=5)

del requests
import requests

# Zeile 9
url = 'http://192.168.178.xx/cm?cmnd=DisplayText%20[s2l13c1]TO: ' +
str(sensor1.temper) + '~F8C'
response = requests.put(url, timeout=5)

# Zeile 10
url = 'http://192.168.178.xx/cm?cmnd=DisplayText%20[s2l14c1]FO: ' +
str(sensor1.humi) + '~25'
response = requests.put(url, timeout=5)

# Zeile 11
url = 'http://192.168.178.xx/cm?cmnd=DisplayText%20[s2l15c1]10: ' +
str(sensor1.pm10) + ' ~E6g/m~5E3'
response = requests.put(url, timeout=5)

# Zeile 12
url = 'http://192.168.178.xx/cm?cmnd=DisplayText%20[s2l16c1]25: ' +
str(sensor1.pm25) + ' ~E6g/m~5E3'
response = requests.put(url, timeout=5)

# Zeile 13
url = 'http://192.168.178.xx/cm?cmnd=DisplayText%20[s2l17c1]LI: ' +
str(sensor3.illu) + ' lux'
response = requests.put(url, timeout=5)
```

Anmerkung 😕 \ Das Script ist nicht optimiert, natürlich könnte es besser sein - aber es funktioniert.

👍

Links

[01] LCD Wiki: ILI9341 -
[02] Tasmota: Displays -
[03] Tasmota: Project Showcase -