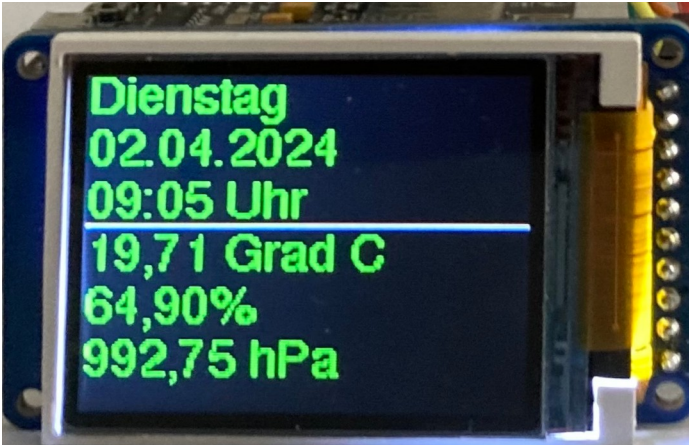


Mit einem ESP32-Wroom und der Bibliothek time.h sollen Datum und Zeit auf einem TFT angezeigt werden.

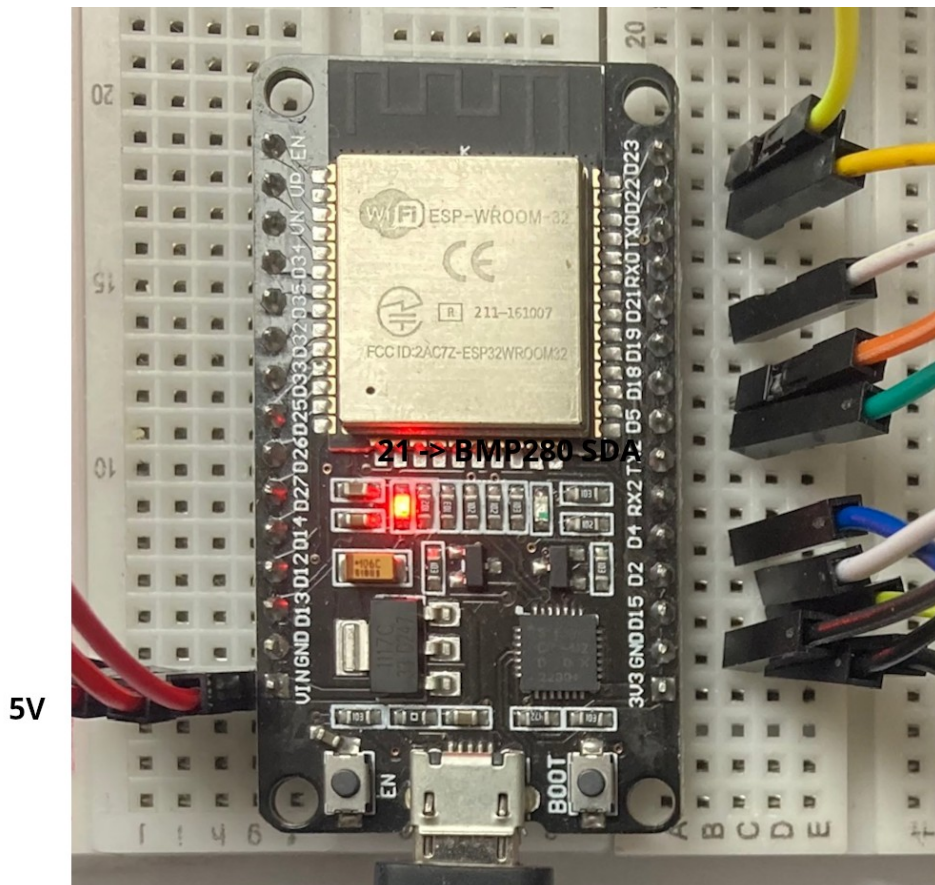
Ein BMP280 steuert Temperatur und Luftdruck bei, der DHT11/DHT22 sorgt für die Anzeige der Luftfeuchtigkeit.

Darstellung auf dem TFT



Zuordnung der Pins

ESP32-Wroom mit 30 Pins

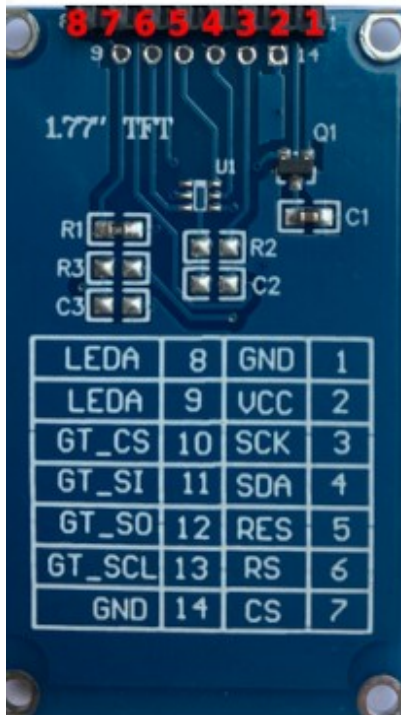


- 23 -> TFT SDO
- 22 -> BMP280 SCL
- 21 -> BMP280 SDA
- 18 -> TFT SCK
- 5 -> TFT TFT_CS
- 4 -> TFT RESET
- 2 -> TFT DC
- 15 -> DHT Daten
- GND

Pinzuordnung alternativer TFTs

1,77 Zoll Pinbelegung

(von rechts nach links)



Pin TFT	ESP32 WROOM
GND	GND
VCC	5V
SCK	18
SDA	23
RES	4
RS	2
CS	5
LEDA	3,3V

TFT Waveshare Pinbelegung (von links nach rechts)

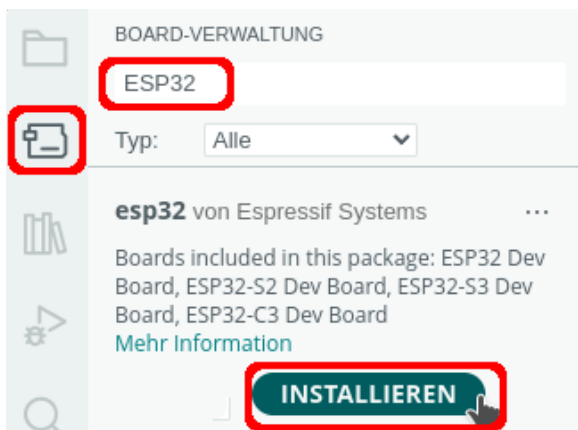
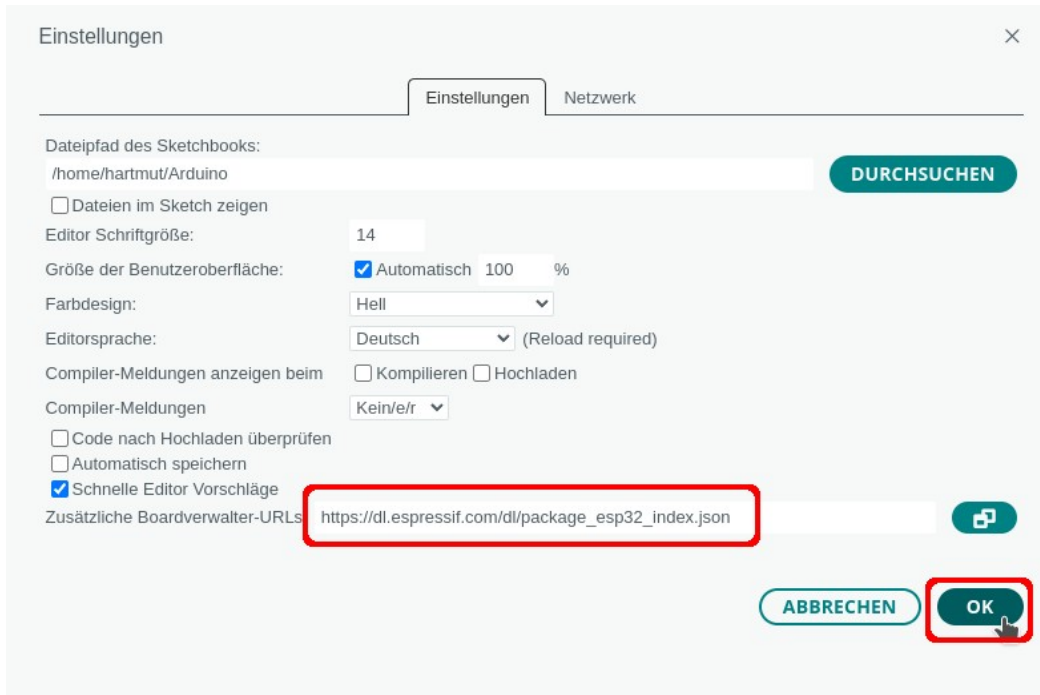


Pin TFT	ESP32 WROOM
VCC	5V
GND	GND
DIN	23
CLK	18
CS	5
DC	2
RST	4
BL	5V

Board installieren:

Trage unter Datei -> Einstellungen eine zusätzliche Boardverwalter-URL ein:

https://espressif.github.io/arduino-esp32/package_esp32_index.json



→ Icon für den Boardverwalter anklicken
oder:

→ Werkzeuge-> Board -> Boardverwalter

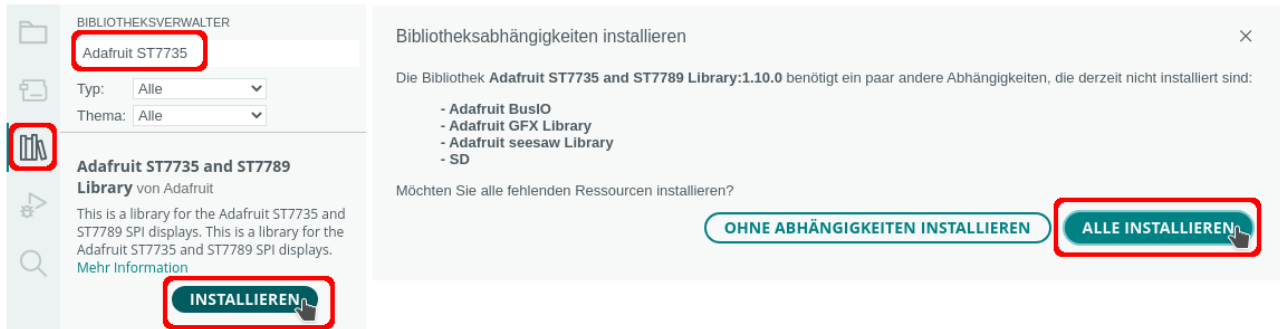
→ nach ESP32 suchen

→ Board installieren



Anschließend wird das Board ausgewählt. Je nach Betriebssystem wird der Port eine andere Bezeichnung haben.

Bibliothek installieren



BIBLIOTHEKSVERWALTER

Adafruit ST7735

Typ: Alle

Thema: Alle

Adafruit ST7735 and ST7789 Library von Adafruit

This is a library for the Adafruit ST7735 and ST7789 SPI displays. This is a library for the Adafruit ST7735 and ST7789 SPI displays.
[Mehr Information](#)

INSTALLIEREN

Bibliotheksabhängigkeiten installieren

Die Bibliothek **Adafruit ST7735 and ST7789 Library:1.10.0** benötigt ein paar andere Abhängigkeiten, die derzeit nicht installiert sind:

- Adafruit BusIO
- Adafruit GFX Library
- Adafruit seesaw Library
- SD

Möchten Sie alle fehlenden Ressourcen installieren?

OHNE ABHÄNGIGKEITEN INSTALLIEREN

ALLE INSTALLIEREN

Benötigte Bibliotheken einbinden und Variablen definieren

```
# include "WiFi.h"
# include "time.h"
# include "Adafruit_GFX.h"
# include "Adafruit_ST7735.h"
# include "Adafruit_BMP280.h"

// Schriftart einbinden
# include "Fonts/FreeSans9pt7b.h"

# include "DHT.h"

int SENSOR_DHT = 15;

// Sensortyp festlegen
// DHT22
# define SensorTyp DHT22

// Sensor einen Namen zuweisen
DHT dht(SENSOR_DHT, SensorTyp);

// Adafruit TFT, WaveShare TFT 1,8 Zoll
# define TFT_CS      5
# define TFT_RST     4
# define TFT_DC      2

Adafruit_BMP280 bmp;
Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS, TFT_DC, TFT_RST);

char Router[] = "Router_SSID";
char Passwort[] = "xxxxxxxx";

// statischeIP = false -> IP-Adresse über DHCP vergeben
// statischeIP = true -> statische IP festlegen
// ip und gateway müssen an das lokale Netz angepasst werden
bool statischeIP = false;
IPAddress ip(192, 168, 1, 200);
```

```

IPAddress gateway(192, 168, 1, 1);
IPAddress subnet(255, 255, 255, 0);

/*
  öffentliche DNS-Server
  -----
  OpenDNS 208, 67, 222, 222 (USA)
  Google 8, 8, 8, 8 (USA)
  Cloudflare 1, 1, 1, 1 (USA)
  DNSWatch 84.200.69.80 (Deutschland)
  Quad9 9, 9, 9, 9 (Schweiz)
  Neustar UltraDNS 56, 154, 70, 3 (USA, gefiltert)
  Deutsche Telekom 217, 5,100,185
  -----
  oder die im Router eingetragene IP
  im Beispiel: 192, 168, 1, 20
*/
IPAddress primaryDNS(192, 168, 1, 20);
IPAddress secondaryDNS(9, 9, 9, 9);

// NTP-Server aus dem Pool
#define Zeitserver "de.pool.ntp.org"

/*
  Liste der Zeitzonen
  https://github.com/nayarsystems/posix_tz_db/blob/master/zones.csv
  Zeitzone CET = Central European Time -1 -> 1 Stunde zurück
  CEST = Central European Summer Time von
  M3 = März, 5.0 = Sonntag 5. Woche, 02 = 2 Uhr
  bis M10 = Oktober, 5.0 = Sonntag 5. Woche 03 = 3 Uhr
*/
#define Zeitzone "CET-1CEST,M3.5.0/02,M10.5.0/03"

// time_t enthält die Anzahl der Sekunden seit dem 1.1.1970 0 Uhr
time_t aktuelleZeit;

/*
  Struktur tm
  tm_hour -> Stunde: 0 bis 23
  tm_min -> Minuten: 0 bis 59
  tm_sec -> Sekunden 0 bis 59
  tm_mday -> Tag 1 bis 31
  tm_wday -> Wochentag (0 = Sonntag, 6 = Samstag)
  tm_mon -> Monat: 0 (Januar) bis 11 (Dezember)
  tm_year -> Jahre seit 1900
  tm_yday -> vergangene Tage seit 1. Januar des Jahres
  tm_isdst -> Wert > 0 = Sommerzeit (dst = daylight saving time)
*/
tm Zeit;
WiFiServer Server(80);
WiFiClient Client;
    
```

Der setup-Teil

```
void setup()
{
  // Zeitzone: Parameter für die zu ermittelnde Zeit
  configTzTime(Zeitzone, Zeitserver);

  Serial.begin(9600);

  // auf serielle Verbindung warten
  while (!Serial);
  delay(1000);

  // WiFi starten
  WiFi.begin(Router, Passwort);

  Serial.println("-----");

  // statische IP vergeben
  if (statischeIP)
  {
    WiFi.config(ip, gateway, subnet, primaryDNS, secondaryDNS);
    Serial.print("Verbunden mit ");
    Serial.println(Router);

    // IP anzeigen
    Serial.print("Statische IP: ");
    Serial.println(ip);
  }

  // IP über DHCP ermitteln
  else
  {
    while (WiFi.status() != WL_CONNECTED)
    {
      delay(200);
      Serial.print(".");
    }
    Serial.println();
    Serial.print("Verbunden mit ");
    Serial.println(Router);
    Serial.print("IP über DHCP: ");
    Serial.println(WiFi.localIP());
  }

  // BMP280 starten
  bmp.begin();

  // DHT starten
  dht.begin();
}
```



```
// schwarzer Hintergrund
tft.initR(INITR_BLACKTAB);

// Rotation anpassen
tft.setRotation(1);
}
```

Der loop-Teil

```
void loop()
{
  // aktuelle Zeit holen
  time(&aktuelleZeit);

  // localtime_r -> Zeit in die lokale Zeitzone setzen
  localtime_r(&aktuelleZeit, &Zeit);

  /*
   TFT-Parameter:
   Schriftart (muss im Kopf eingebunden werden)
   Cursor Spalte, Zeile setzen
   Textfarbe
   Bildschirm mit schwarz füllen
  */
  tft.setFont(&FreeSans9pt7b);
  tft.setCursor(1, 13);
  tft.setTextColor(ST7735_GREEN);
  tft.fillScreen(ST7735_BLACK);

  Serial.println("-----");

  // es kann bis zu 60 Sekunden dauern
  // bis die Zeit ermittelt wird

  // Name des Wochentages 0-6
  switch (Zeit.tm_wday)
  {
    case 0:
      Serial.print("Sonntag");
      tft.print("Sonntag");
      break;
    case 1:
      Serial.print("Montag");
      tft.print("Montag");
      break;
    case 2:
      Serial.print("Dienstag");
      tft.print("Dienstag");
      break;
  }
}
```

```
case 3:
    Serial.print("Mittwoch");
    break;
    tft.print("Mittwoch");
case 4:
    Serial.print("Donnerstag");
    tft.print("Donnerstag");
    break;
case 5:
    Serial.print("Freitag");
    tft.print("Freitag");
    break;
case 6:
    Serial.print("Samstag");
    tft.print("Samstag");
    break;
}

Serial.print(", ");
tft.setCursor(1, 33);

// Datum
if (Zeit.tm_mday < 10)
{
    Serial.print("0");
    tft.print("0");
}
Serial.print(Zeit.tm_mday);
tft.print(Zeit.tm_mday);

Serial.print(".");
tft.print(".");

// Monat: führende 0 ergänzen
if (Zeit.tm_mon < 10)
{
    Serial.print("0");
    tft.print("0");
}

// Zählung beginnt mit 0 -> +1
Serial.print(Zeit.tm_mon + 1);
Serial.print(".");
tft.print(Zeit.tm_mon + 1);
tft.print(".");

// Anzahl Jahre seit 1900
Serial.print(Zeit.tm_year + 1900);
Serial.print(" ");
tft.print(Zeit.tm_year + 1900);
tft.print(" ");
```



```

tft.setCursor(1, 53);

// Stunde: wenn Stunde < 10 -> 0 davor setzen
if (Zeit.tm_hour < 10)
{
    Serial.print("0");
    tft.print("0");
}
Serial.print(Zeit.tm_hour);
Serial.print(":");
tft.print(Zeit.tm_hour);
tft.print(":");

// Minuten
if (Zeit.tm_min < 10)
{
    Serial.print("0");
    tft.print("0");
}
Serial.print(Zeit.tm_min);
Serial.print(":");
tft.print(Zeit.tm_min);
tft.print(" Uhr");

// Sekunden (nur Serieller Monitor)
if (Zeit.tm_sec < 10) Serial.print("0");
Serial.print(Zeit.tm_sec);

Serial.println();
Serial.println("Tage seit dem 1.1. " + String(Zeit.tm_yday));

// Normalzeit/Sommerzeit
if(Zeit.tm_isdst > 0) Serial.println("MESZ = Mitteleuropäische Sommerzeit");
else Serial.println("MEZ = Mitteleuropäische Zeit");

// Temperatur BMP280
String Temperatur = String(bmp.readTemperature());

// alternativ: Temperatur DHT
// String Temperatur = String(dht.readTemperature());

// . durch , ersetzen
Temperatur.replace(".", ",");

/*
    BMP280 Luftdruck messen
    readPressure() liest in Pascal, Ausgabe in hPa (Hekto-Pascal)
    Ergebnis durch 100 teilen
*/
String Luftdruck = String(bmp.readPressure() / 100);
Luftdruck.replace(".", ",");
    
```

```
// Luftfeuchtigkeit DHT lesen
String Luftfeuchtigkeit = String(dht.readHumidity());

// . durch , ersetzen
Luftfeuchtigkeit.replace(".", ",");

// Ausgabe Serieller Monitor
Serial.println("Temperatur: " + Temperatur + "°C");
Serial.println("Luftfeuchtigkeit: " + Luftfeuchtigkeit + "%");
Serial.println("Luftdruck: " + Luftdruck + " hPa");

// Ausgabe TFT
tft.drawFastHLine(1, 57, tft.width(), ST7735_WHITE);
tft.setCursor(1, 73);
tft.print(Temperatur + " Grad C");
tft.setCursor(1, 93);
tft.print(Luftfeuchtigkeit + "%");
tft.setCursor(1, 113);
tft.print(Luftdruck + " hPa");

// Wartezeit bis zur nächsten Messung
delay(5000);
}
```

Hartmut Waller (hartmut-waller.info/arduino-blog) Letzte Änderung: 03.04.24