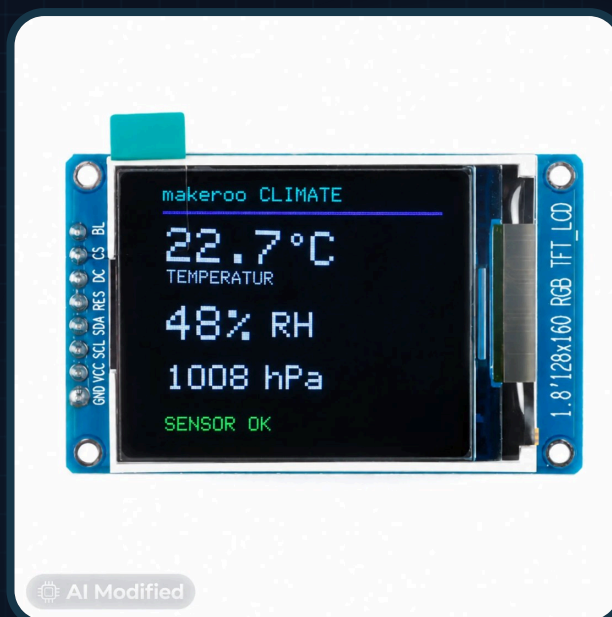


FARBDISPLAY · SPI · ESP32

1,8" TFT ST7735 IPS

128 × 160 Pixel in Farbe auf einem IPS-Panel, schlanke 8-Pin-Leiste. Erstes Farbbild in rund 25 Minuten – mit dem ESP32 DevKit.



AUFLÖSUNG

128 × 160 Pixel

CONTROLLER

ST7735

PANEL

IPS

SCHNITTSTELLE

SPI

01

Verkabeln

8 Leitungen zum ESP32

02

Einrichten

IDE, Board-Paket, Bibliothek

03

Hochladen

Beispiel-Sketch

04

Anzeigen

Farben und Text

01 Auf einen Blick

Kompaktes Farbdisplay ohne SD-Slot: acht Pins für Bild und Hintergrundlicht, ein IPS-Panel, das auch von der Seite farbtreu bleibt. Die Logik arbeitet mit 3,3 V – deshalb nutzen wir den ESP32.

Technische Daten

Display	TFT, IPS, 1,8 Zoll
Auflösung	128 × 160 Pixel
Controller	ST7735
Farbtiefe	16 Bit, 65 536 Farben
Schnittstelle	SPI
Versorgung	VCC 3,3–5 V, Logik 3,3 V
Anschluss	8 Pins, Stiftleiste liegt bei

Lieferumfang

1 ×	1,8" TFT-Displaymodul (ST7735, 8-Pin)
1 ×	Stiftleiste (zum Selbstlöten)



3,3-V-Logik

Am UNO bräuchtest du Pegelwandler in den Datenleitungen. Am ESP32 verdrahtest du direkt.



Helligkeit regeln

BLK steuert das Hintergrundlicht. Der Sketch dimmt es per PWM – offen gelassen leuchtet es dauerhaft.

Pinbelegung

PIN	FUNKTION	SPANNUNG	HINWEIS
● GND	Masse	0 V	Gemeinsame Masse.
● VCC	Versorgung	3,3–5 V DC	An 3V3 des ESP32.
● SCK	SPI Takt	3,3 V	ESP32 GPIO18.
● SDA	SPI Daten (MOSI)	3,3 V	ESP32 GPIO23.
● RES	Reset	3,3 V, aktiv LOW	ESP32 GPIO17.
● DC	Data/Command	3,3 V	ESP32 GPIO16.
● CS	Chip Select	3,3 V, aktiv LOW	ESP32 GPIO5.
● BLK	Hintergrundlicht	3,3 V	ESP32 GPIO4, per PWM dimmbar.

Reihenfolge laut Aufdruck: GND · VCC · SCL · SDA · RES · DC · CS · BL (SCL = SCK, BL = BLK).

Was du zusätzlich brauchst

ESP32 DevKit USB-C

WROOM-32, CH340C – 3,3-V-Logik

[MKR-ESP32-005](#)

Breadboard + Jumper-Set

400 Kontakte, 40 Dupont-Kabel

[MKR-SET-BB400-JUMPER](#)

LötKolben

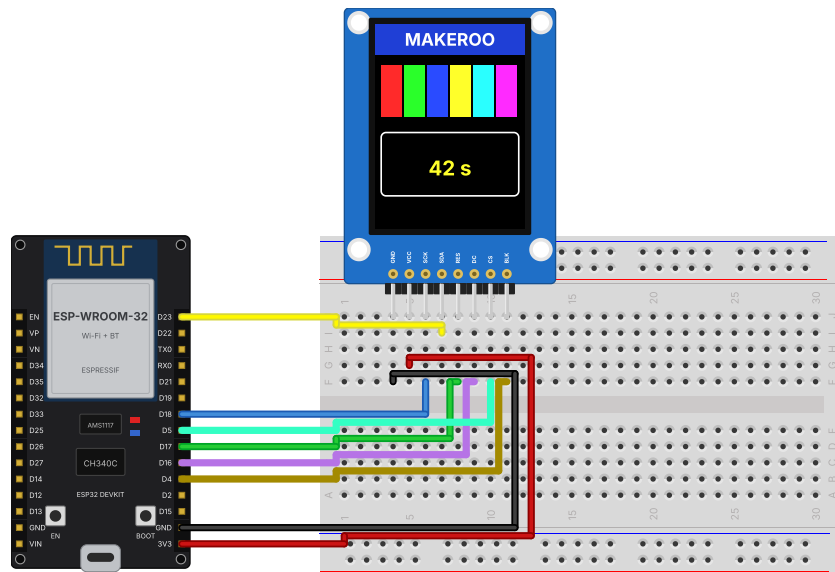
Stiftleiste einlöten (z. B. TOOLCRAFT LS-30E)

[MKR-LS30E-70](#)

02 Verkabeln

Acht Leitungen – Strom, SPI-Bus, drei Steuerleitungen und das Hintergrundlicht.

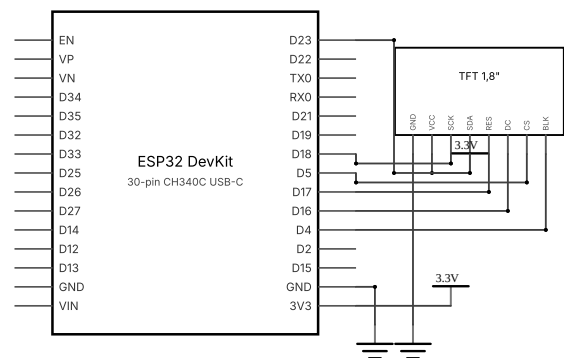
STECKPLATINEN-ANSICHT



Verbindungen

DISPLAY		ESP32 DEVKIT	LEITUNG
● GND	→	GND	Masse
● VCC	→	3V3	Versorgung
● SCK	→	D18	SPI Takt
● SDA	→	D23	SPI MOSI
● RES	→	D17	Reset
● DC	→	D16	Data/Command
● CS	→	D5	Chip Select
● BLK	→	D4	Helligkeit (PWM)

SCHALTPLAN



Verdrahtung erstellt mit Fritzing

i Stiftheiste einlöten

Leiste ins Steckbrett stecken, Display auflegen, acht Pins verlöten.



SDA ist hier SPI

Trotz des Namens ist SDA die SPI-Datenleitung (MOSI) – nicht an den I2C-Pin D21.

03 Einrichten & Hochladen

Schritte 1 bis 3 machst du nur ein einziges Mal – danach ist dein ESP32 für alle weiteren Projekte startklar.

1 Arduino IDE installieren

Lade die aktuelle Arduino IDE 2 von arduino.cc/en/software herunter und installiere sie.

2 ESP32-Board-Paket installieren

Im Boardverwalter (Board-Symbol links) nach **esp32** suchen und **esp32 von Espressif Systems** installieren. Den CH340C-Treiber bringen Windows 10/11 und macOS meist mit. Ausführlich mit Bildern: unser Guide zum ESP32 DevKit.

3 Board und Port wählen

ESP32 per USB-C anschließen. In der Board-Auswahl **ESP32 Dev Module** wählen und den neuen COM-Port (Windows) bzw. `/dev/ttyUSB0` (Linux) / `/dev/cu.wchusbserialL...` (macOS).

4 Bibliothek installieren

Bibliotheksverwalter öffnen (`Strg` + `Umschalt` + `I`), nach **Adafruit ST7735 and ST7789 Library** von Adafruit suchen und installieren. Abhängigkeiten (u. a. *Adafruit GFX Library*) mit **Alle installieren** bestätigen.

5 Sketch hochladen

Neuen Sketch anlegen, den Code von Seite 5 einfügen und mit dem Pfeil-Symbol hochladen (`Strg` + `U`). Unten erscheint „Hochladen abgeschlossen“.

6 Seriellen Monitor öffnen (optional)

`Strg` + `Umschalt` + `M` , unten rechts **115200 Baud**. Dort erscheint „Display bereit.“.



i Schnelltest

Nach dem Upload erscheinen eine blaue Titelleiste, sechs Farbstreifen und die laufende Sekundenzahl in Gelb.

Nachbildung der Anzeige – berechnet aus dem Sketch auf Seite 5, Pixel für Pixel.

04

Der erste Sketch

Titel, Farbstreifen und ein Zähler – dazu die Helligkeit per PWM.

```
MAKER00_TFT18_Start.ino

1 // MAKER00 · 1,8" TFT ST7735 (8-Pin) · Erster Start (ESP32)
2 #include <SPI.h>
3 #include <Adafruit_GFX.h>
4 #include <Adafruit_ST7735.h>
5
6 const int TFT_CS = 5, TFT_DC = 16, TFT_RES = 17; // SDA = 23, SCK = 18
7 const int TFT_BLK = 4; // Hintergrundlicht
8 Adafruit_ST7735 tft(TFT_CS, TFT_DC, TFT_RES);
9 const uint16_t FARBE[] = {ST77XX_RED, ST77XX_GREEN, ST77XX_BLUE,
10 ST77XX_YELLOW, ST77XX_CYAN, ST77XX_MAGENTA};
11
12 void setup() {
13   Serial.begin(115200);
14   ledcAttach(TFT_BLK, 5000, 8); // PWM fuer die Helligkeit
15   ledcWrite(TFT_BLK, 255); // 0 = aus, 255 = voll
16   tft.initR(INITR_BLACKTAB); // 128 x 160 Pixel
17   tft.fillScreen(ST77XX_BLACK);
18   tft.fillRect(0, 0, 128, 24, ST77XX_BLUE); // Titelleiste
19   tft.setTextColor(ST77XX_WHITE);
20   tft.setTextSize(2);
21   tft.setCursor(22, 5);
22   tft.print("MAKER00");
23   for (int i = 0; i < 6; i++) { // sechs Farbstreifen
24     tft.fillRect(4 + i * 20, 32, 18, 40, FARBE[i]);
25   }
26   tft.drawRoundRect(4, 84, 120, 50, 8, ST77XX_WHITE);
27   tft.setTextSize(1);
28   tft.setCursor(12, 92);
29   tft.print("Laufzeit:");
30   Serial.println("Display bereit.");
31 }
32
33 void loop() {
34   tft.setTextColor(ST77XX_YELLOW, ST77XX_BLACK); // Text mit Hintergrund
35   tft.setTextSize(2);
36   tft.setCursor(12, 108);
37   tft.print(millis() / 1000);
38   tft.print(" s ");
39   delay(500);
40 }
```

Zeile 6-7

Pins

CS, DC, Reset und BLK frei wählbar; MOSI und SCK sind der Standard-SPI.

Zeile 14-15

Helligkeit

PWM an BLK: 0 = aus, 255 = volle Helligkeit.

Zeile 16

Display-Typ

BLACKTAB passt zu den meisten 1,8-Zoll-Modulen. Sonst siehe Fehlersuche.

Zeile 23-25

Farbtest

Sechs Grundfarben zeigen, ob die Farbreihenfolge stimmt.

Zeile 34

Text mit Hintergrund

Überschreibt alte Ziffern ohne Flackern.

Zeile 37-38

Zähler

Nur dieser Bereich wird neu geschrieben.

Geprüft: kompiliert für ESP32 Dev Module (Board-Paket esp32 3.3.12) mit Adafruit ST7735 and ST7789 Library 1.12.0 (310.4 kB Flash = 24 %, 23 kB RAM).

05

Wenn es nicht klappt

SYMPTOM	LÖSUNG
Bildschirm bleibt weiß	SPI prüfen: SDA an D23, SCK an D18, CS an D5, DC an D16, RES an D17.
Rot und Blau vertauscht	INITR_GREENTAB statt INITR_BLACKTAB probieren.
Farben invertiert (Schwarz wird weiß)	IPS-Variante: nach <code>initR()</code> die Zeile <code>tft.invertDisplay(true);</code> ergänzen.
Bunte Pixel am Rand	Variante INITR_GREENTAB oder INITR_REDTAB testen.
Display dunkel	BLK an D4? Steht im Sketch <code>ledcWrite(TFT_BLK, 255)?</code>

06

Wie geht's weiter?

Mit ESP32 & ESPHome

Gleiche Verdrahtung wie hier – das Display direkt in ESPHome:

```
spi:
  clk_pin: GPIO18
  mosi_pin: GPIO23
display:
  - platform: st7735
    model: "INITR_BLACKTAB"
    cs_pin: GPIO5
    dc_pin: GPIO16
    reset_pin: GPIO17
    device_width: 128
    device_height: 160
    col_start: 0
    row_start: 0
```

Projektideen

Statusseite

WLAN, Uhrzeit und Sensorwerte in Farbe

Diagramme

Temperaturverlauf als Kurve

Spiel

Snake oder Breakout mit Tasten

Geräte-Front

Im 3D-Druck-Gehäuse verbaut

Passt dazu

BMP280 Luftdruck & Temperatur

Messwerte zum Anzeigen

[MKR-6Y-BMP280-024](#)

ESP32 DevKit USB-C

WROOM-32, CH340C –
3,3-V-Logik

[MKR-ESP32-005](#)

Drehgeber EC11

Werte einstellen per
Drehknopf

[9505453479775](#)

ESP32 Gehäuse aus Acryl

Schützt das Board

[MKR-ESP32-HOUSING-045](#)

Sicherheitshinweise. Elektronikmodul für Ausbildungs-, Entwicklungs- und Experimentierzwecke. Nur mit Kleinspannung betreiben, niemals an Netzspannung. Nicht geeignet für Kinder unter 14 Jahren; Nutzung nur unter Aufsicht. Elektroaltgeräte gehören nicht in den Hausmüll – Rückgabe kostenlos bei kommunalen Sammelstellen.

Verantwortlicher Wirtschaftsakteur (GPSR): Alexander Prez · makeroo · Hinterm Bahnhof 4 · 90513 Zirndorf · support@makeroo.de
Fragen? Schreib uns mit einem Foto deiner Verkabelung. Arduino ist eine Marke der Arduino SA und dient nur der Kompatibilitätsangabe.