

FAQ zum AZ-Touch MOD 2.4“ und 2.8“

In den letzten Tagen, seitdem unser AZ-Touch MOD bei uns im Shop angeboten wird, erreichen uns ständig neue Fragen und Probleme. Auch als Blogautor bleibt man von Misserfolgen nicht verschont und versucht die Blogbeiträge gleich so zu schreiben, dass keine Probleme auftauchen. Im Falle vom AZ-Touch MOD mit dem 2.4“-Display und mittlerweile auch 2.8“-Display sollen in diesem Blogbeitrag die aktuellen Fragen von Kunden aufgegriffen und beantwortet werden.

Die Hardware

Für diesen Blogbeitrag brauchen Sie die Bauteile, welche Sie in Tabelle 1 finden. Beachten Sie, dass Sie nur einen MikroController und ein AZ-Touch MOD brauchen. Im hier gezeigten Fall wird das AZ-Touch MOD mit dem 2.4“-Display mit dem ESP32 NodeMCU Module WLAN WiFi Development Board mit CP2102 verwendet.

Pos	Anzahl	Bauteil	Link
1	1	Touch MOD mit dem 2.4“-Display	https://az-delivery.de
2	1	ESP32 NodeMCU Module WLAN WiFi Development Board	https://az-delivery.de
3	1	Touch MOD mit dem 2.8“-Display (optional)	https://az-delivery.de
4	1	ESP32 D1 Mini NodeMCU WiFi Modul + Bluetooth (optional)	https://az-delivery.de

Tabelle 1: Bauteile für Tests und den CO2-Warner

Bedenken Sie beim Kauf direkt, dass Sie für beide AZ-Touch MOD **separat** den MicroController kaufen müssen! Im Lieferumfang ist nur die Platine, das Gehäuse, das Touch-Display und Kleinmaterial enthalten.

Damit Sie das AZ-Touch MOD benutzen können, müssen Sie die Buchsenleisten noch auf die Platine löten. Wie das genau geht, finden Sie in diesem [Blogartikel](#).

Wie müssen die Bauteile verlötet werden

Achten Sie darauf, dass bei allen Buchsenleisten die Kontakte auf der Unterseite der Platine verlötet werden. Auf der Oberseite befinden sich die Steckkontakte zu den weiteren Bauteilen, siehe Abbildung 1.

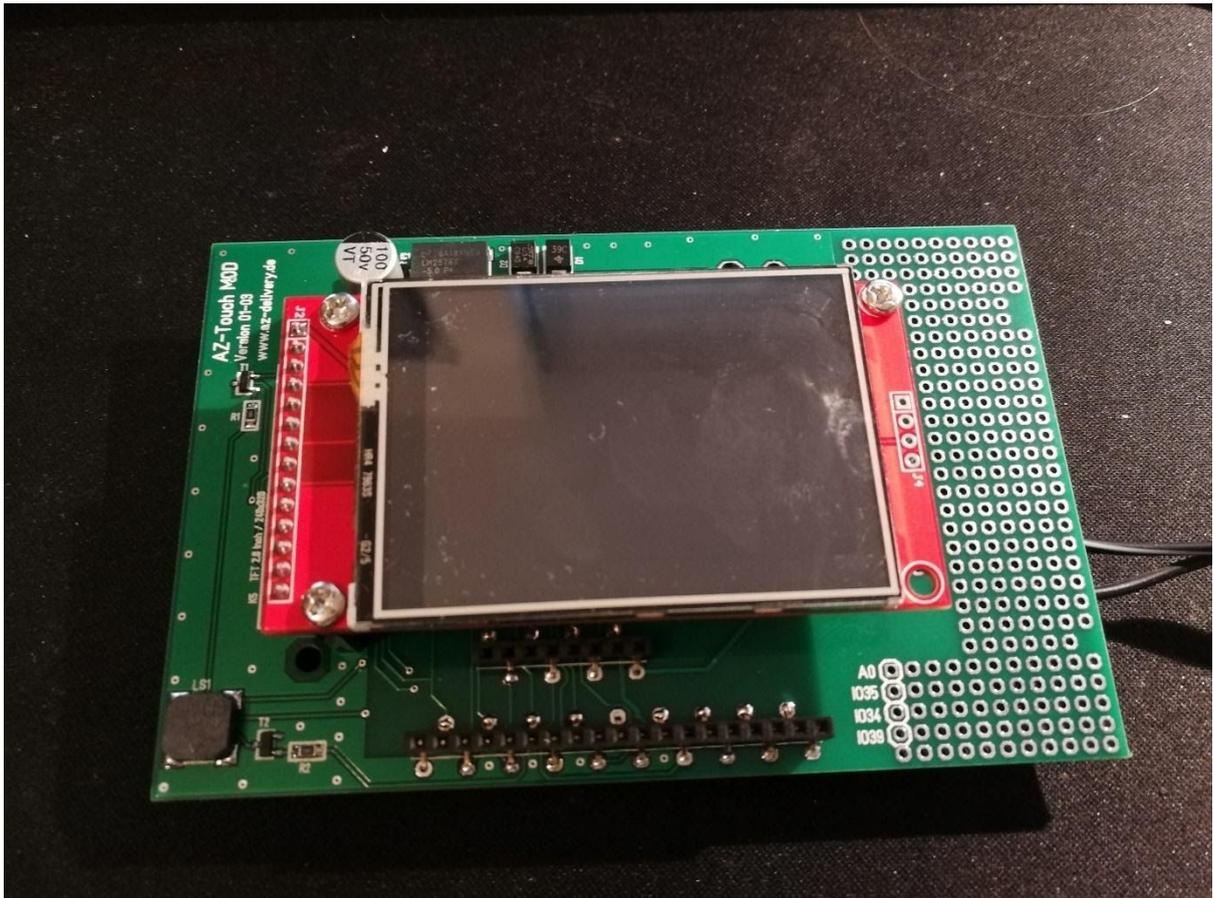


Abbildung 1: Oberseite mit allen verlöteten und verbauten Komponenten

Achten Sie beim Verlöten der Buchsenleiste für das Display auf die Beschriftung, welche Position für welches Touchdisplay gedacht ist.

Die Unterseite sollte wie in Abbildung 2 aussehen. Anders als bei gewöhnlichen Buchsenleisten, wird der Kontakt des Micro Controllers über das Durchstecken auf der Rückseite der Platine erreicht.

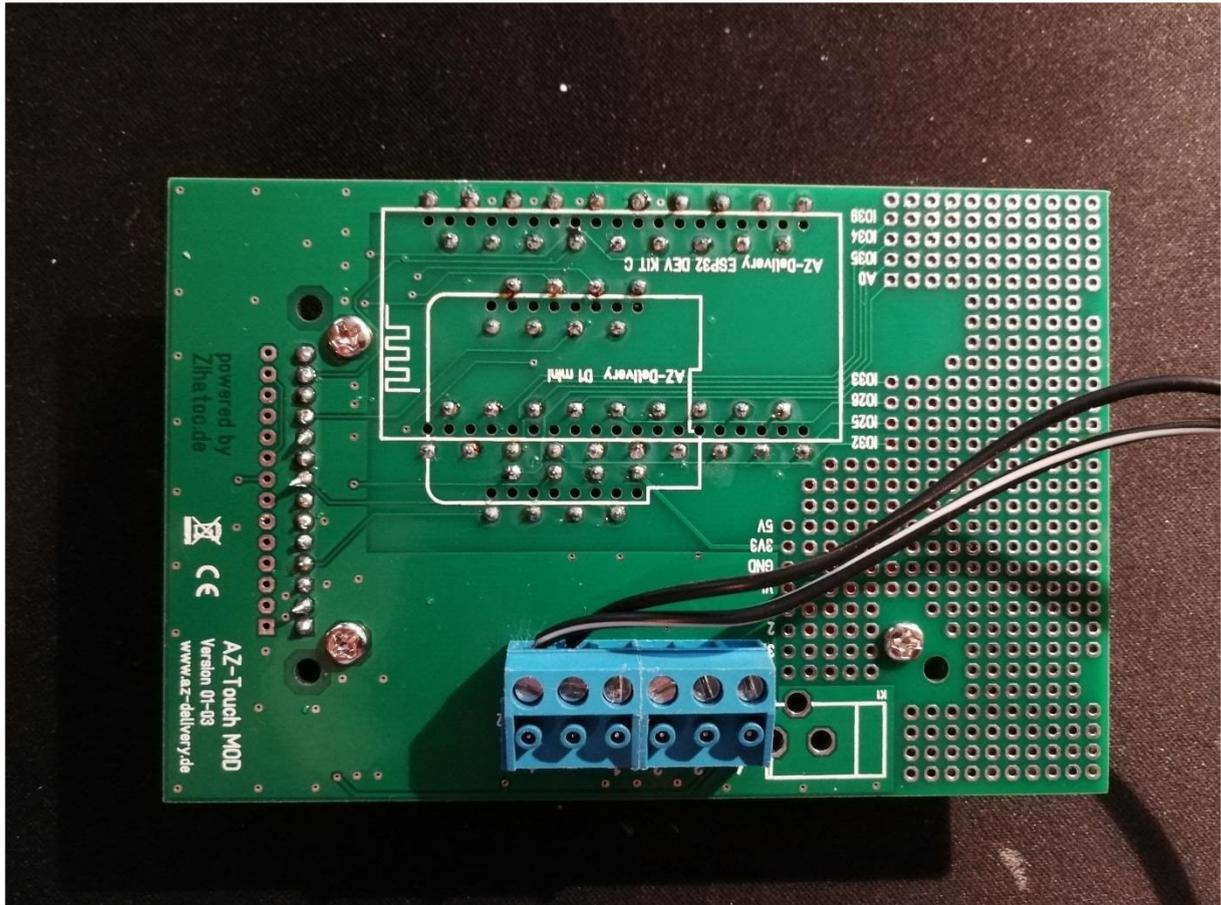


Abbildung 2: Unterseite der fertig verlöteten Platine

Bei der Frage, wie rum das ESP32 NodeMCU Module WLAN WiFi Development Board eingesetzt werden muss, hilft die schematische Zeichnung auf der Rückseite, siehe Abbildung 2 und Abbildung 3.

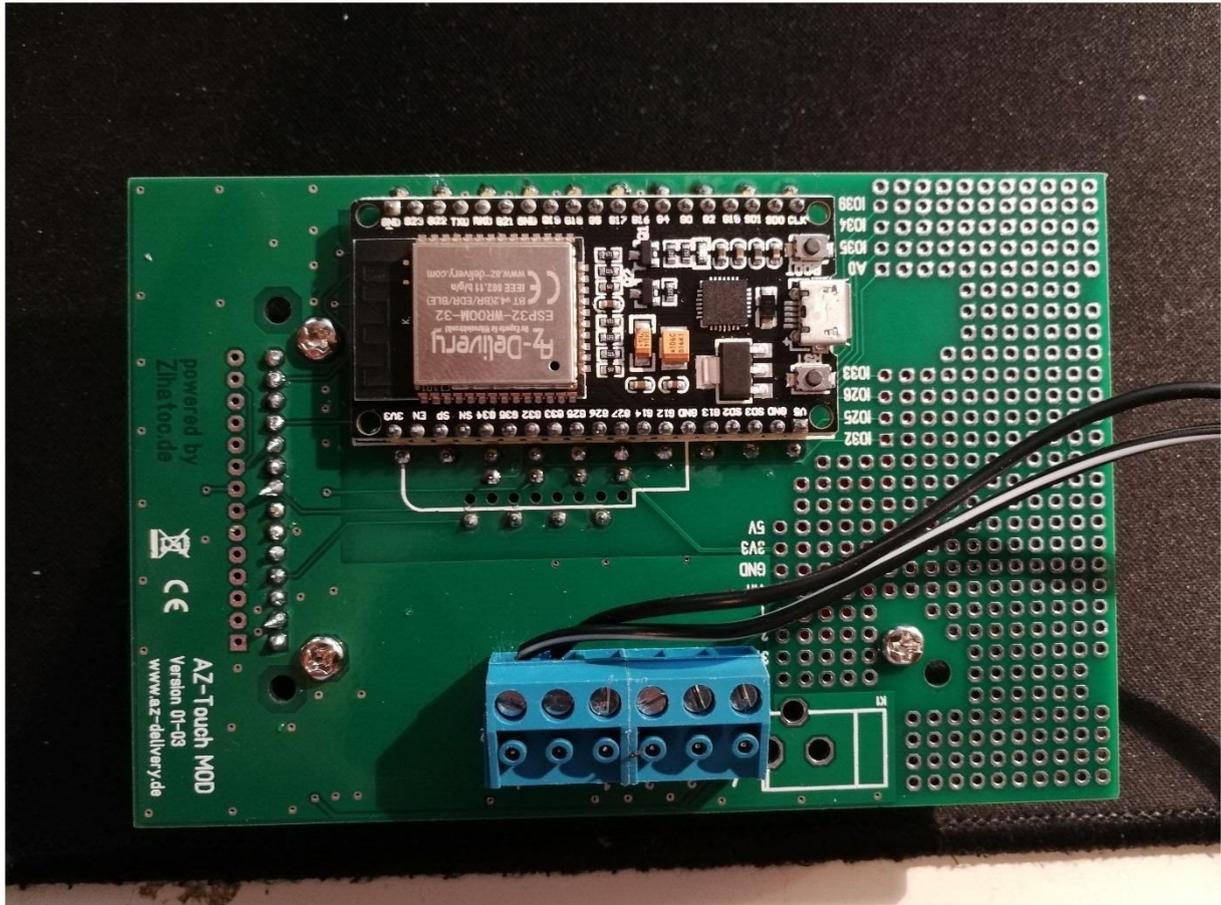


Abbildung 3: Eingesteckter Micro Controller auf der Rückseite der Platine

Was brauchen Sie an Software?

Für die Entwicklung ihrer Ideen und Anzeige auf dem Touchdisplay, brauchen Sie die folgenden Dinge:

- [Arduino IDE](#) , hier am besten die aktuelle Version herunterladen
- Die Bibliothek `TFT_eSPI` mit allen Abhängigkeiten

Wie Sie Bibliotheken installieren, zeigen wir u.a. [hier](#). Damit das Display auch korrekt angesteuert wird, müssen Sie noch die `User_Setup.h` der Bibliothek `TFT_eSPI` modifizieren. Um schnell den Pfad zu den heruntergeladenen Bibliotheken zu finden, gehen Sie in der Arduino IDE auf Datei -> Voreinstellungen und gehen Sie zu dem angegebenen Pfad, siehe Abbildung 4.

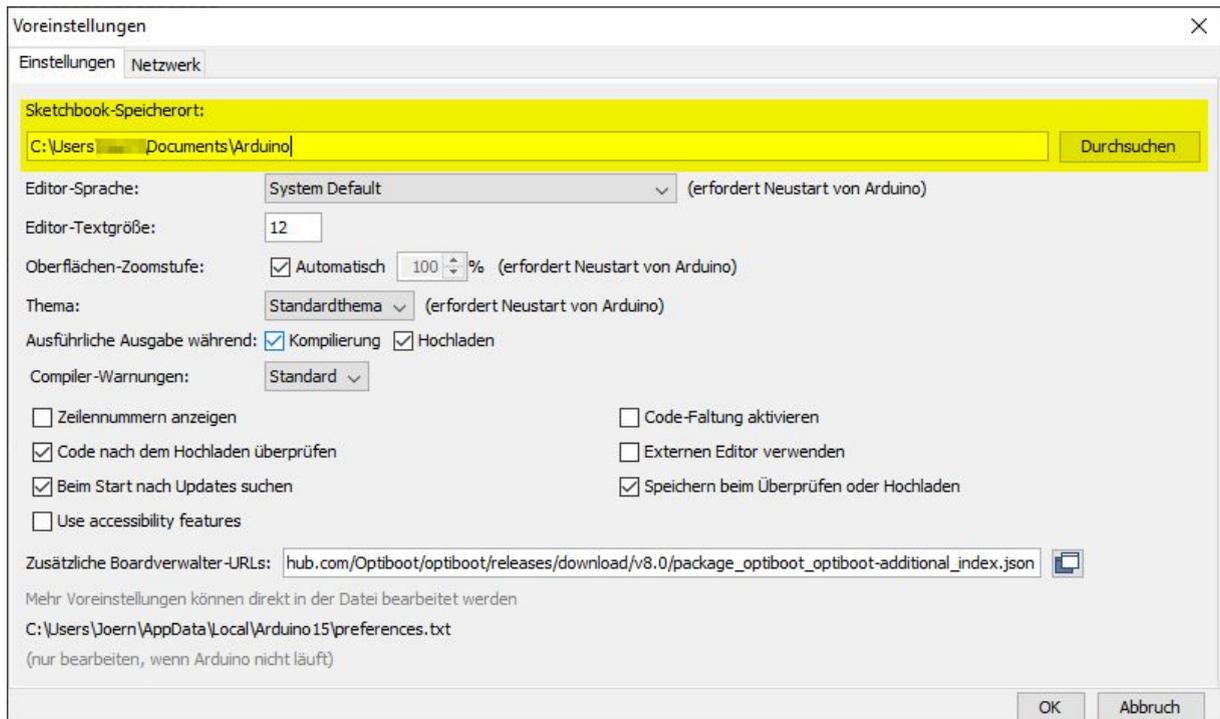


Abbildung 4: Pfad zu Sketches und Bibliotheken

Die Datei **User_Setup.h** finden Sie im Sketchbook-Speicherort im Unterordner **libraries\TFT_eSPI**, siehe Abbildung 5.

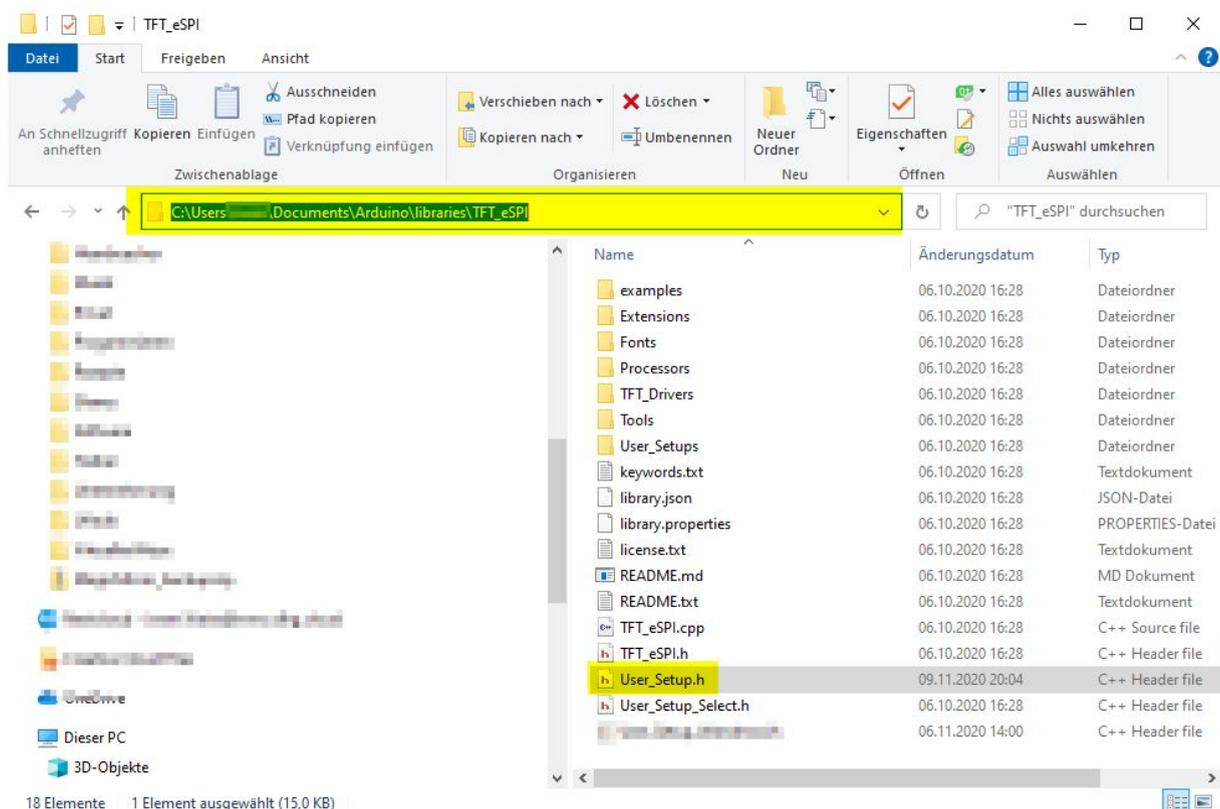


Abbildung 5: User_Setup.h der Bibliothek TFT_eSPI

In dieser Datei müssen Sie nun einige Anpassungen machen, damit das Display später mit dem Micro Controller funktioniert. Dazu müssen Sie zum einen in der Datei User_Setup.h die Displaygröße, siehe Code 1, und die Pins für die Displaykommunikation, siehe Code 2, konfigurieren. Der Schnelligkeit

halber, finden Sie die fertige User_Setup.h-Datei für das 2.4"-Touchdisplay unter https://github.com/M3taKn1ght/Blog-Repo/tree/master/FAQ_AZ-Touch_MOD. Dieselben Einstellungen sollten auch für das 2.8"-Touchdisplay nutzbar sein.

```
// For ST7789, ST7735 and ILI9163 ONLY, define the pixel width and height in portrait orientation
// #define TFT_WIDTH 80
// #define TFT_WIDTH 128
#define TFT_WIDTH 240 // ST7789 240 x 240 and 240 x 320
// #define TFT_HEIGHT 160
// #define TFT_HEIGHT 128
// #define TFT_HEIGHT 240 // ST7789 240 x 240
#define TFT_HEIGHT 320 // ST7789 240 x 320
```

Code 1: Korrekte Displaygröße für Touchdisplay

```
// ##### EDIT THE PIN NUMBERS IN THE LINES FOLLOWING TO SUIT YOUR ESP32 SETUP #####
```

```
// For ESP32 Dev board (only tested with ILI9341 display)
// The hardware SPI can be mapped to any pins
```

```
#define TFT_MISO 19
#define TFT_MOSI 23
#define TFT_SCLK 18
#define TFT_CS 5 // Chip select control pin
#define TFT_DC 4 // Data Command control pin
#define TFT_RST 22 // Reset pin (could connect to RST pin)
// #define TFT_RST -1 // Set TFT_RST to -1 if display RESET is connected to ESP32 board RST
```

```
#define TOUCH_CS 14 // Chip select pin (T_CS) of touch screen
```

Code 2: Pins für Touchdisplay definieren

Anpassung im (Beispiel)Code

Mit den bisher getroffenen Einstellungen ist der naheliegende Schritt, nun ein Beispiel-Sketch der TFT_eSPI-Bibliothek zu verwenden. Anbieten würde sich das Beispiel Touch_calibrate, siehe Abbildung 6.

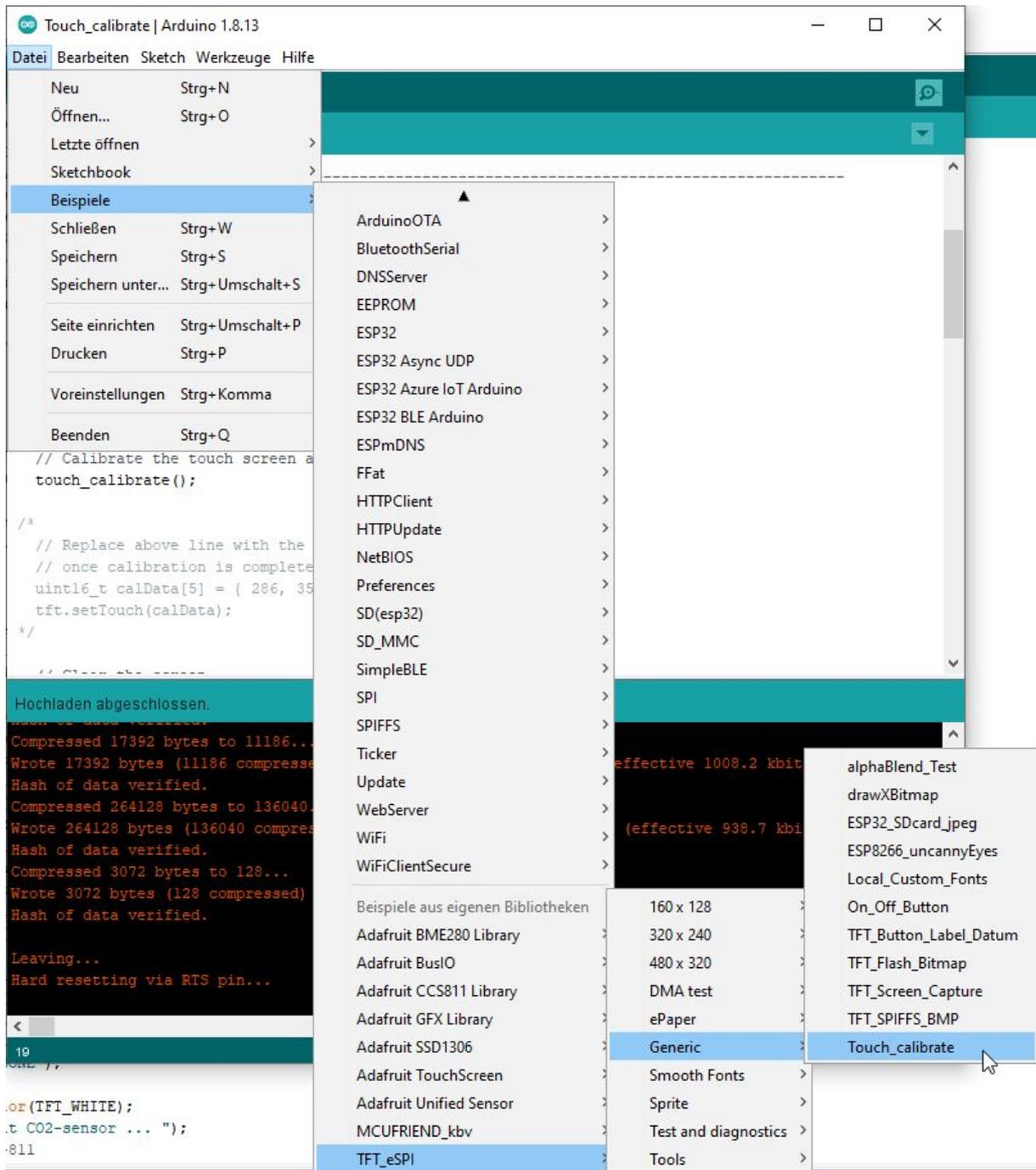


Abbildung 6: Beispiel Touch_calibrate öffnen

Egal ob ESP32 NodeMCU Module oder ESP32 D1 Mini NodeMCU, der Bildschirm wird erst einmal schwarz bleiben, siehe Abbildung 7. In manchen Fällen kann es auch passieren, dass der Bildschirm flackert, was aber erst einmal kein Grund für panische Anrufe beim Kundensupport von AZ-Delivery ist.

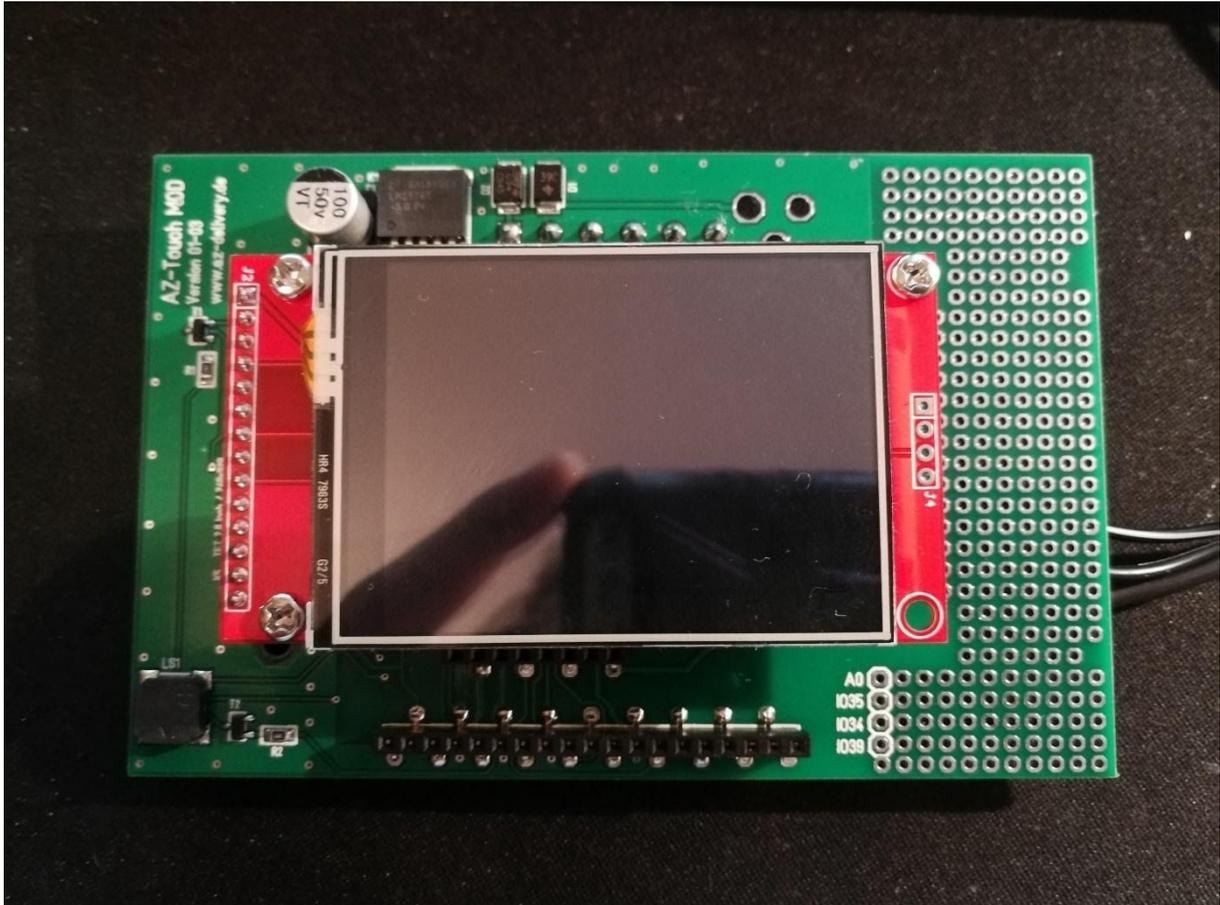


Abbildung 7: Display ohne Anzeige

Tatsächlich wird das Programm auf dem Mikrocontroller ausgeführt, aber der Pin „LED“ vom Display wird nicht mit Spannung versorgt. Fügen Sie der `setup()`-Funktion die beiden Zeilen aus Code 3 hinzu, sodass die `setup()`-Funktion Code 4 entspricht.

```
pinMode(15, OUTPUT);  
digitalWrite(15, LOW);
```

Code 3: Fehlender Codeteil für `setup()`

```
void setup() {  
  // Use serial port  
  Serial.begin(115200);  
  pinMode(15, OUTPUT);  
  digitalWrite(15, LOW);  
  // Initialise the TFT screen  
  tft.init();  
  
  // Set the rotation before we calibrate  
  tft.setRotation(1);  
  
  // Calibrate the touch screen and retrieve the scaling factors  
  touch_calibrate();  
  
  /*  
  // Replace above line with the code sent to Serial Monitor  
  // once calibration is complete, e.g.:  
  */
```

```

uint16_t calData[5] = { 286, 3534, 283, 3600, 6 };
tft.setTouch(calData);
*/

// Clear the screen
tft.fillScreen(TFT_BLACK);
tft.drawCentreString("Touch screen to test!",tft.width()/2, tft.height()/2, 2);
}

```

Code 4: Komplette setup()-Funktion

So wird Ihnen das Display den gewünschten Inhalt anzeigen, siehe Abbildung 8.

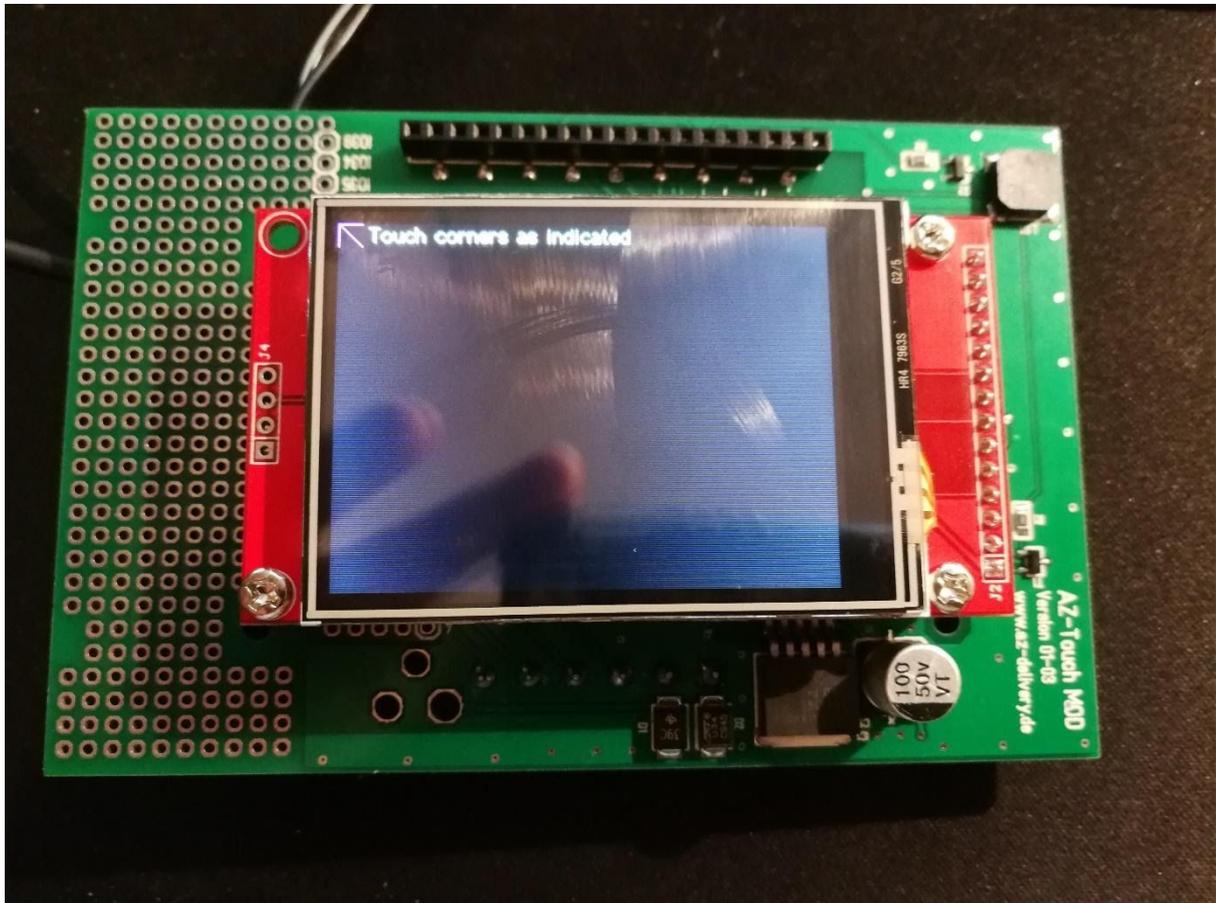


Abbildung 8: Touchdisplay nach hinzufügen der fehlenden Codezeilen

Egal welches Beispiel Sie nun verwenden, die beiden Zeilen aus Code 3 müssen Sie in der setup()-Funktion ergänzen. Somit kriegen Sie immer das gewünschte Ergebnis, siehe Abbildung 9.

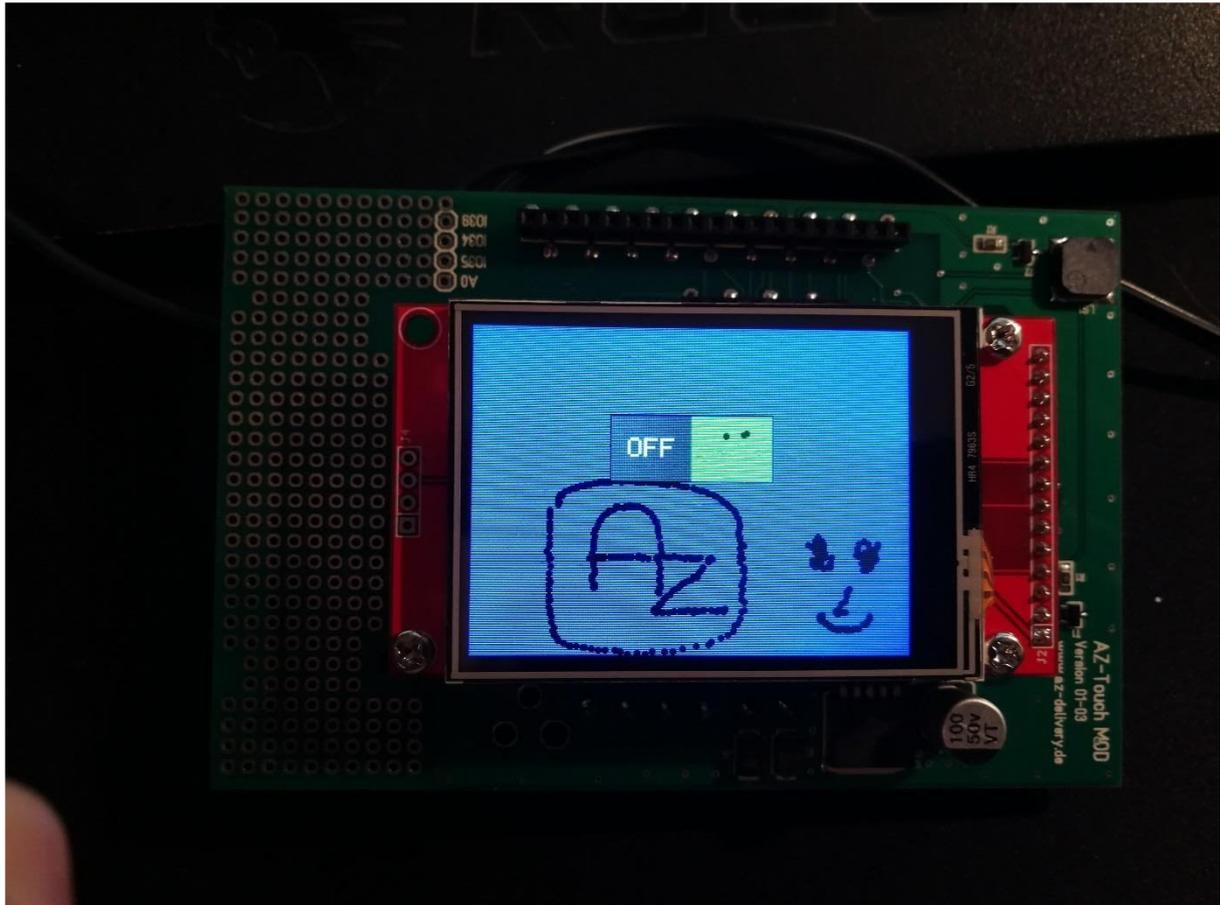


Abbildung 9: Angepasstes Beispiel Btn_on_off

Sollten Sie nun Ihr eigenes Projekt umsetzen, so müssen sie natürlich auch dabei die Codezeilen aus Code 3 verwenden.

Sollten Sie das Beispiel der Wetterstation von der Vorgängerversion umsetzen wollen, so dürfen Sie die Anpassungen aus Code 5 nicht vergessen.

```
#define TOUCH_IRQ 2  
wird dann:  
#define TOUCH_IRQ 27
```

Code 5: Anpassungen Wetterstation mit AZ-Touch Mod

Ich hoffe an dieser Stelle, dass viele Unklarheiten und Probleme beseitigt werden konnten und das AZ-Touch mod nun so funktioniert, wie es auf unserer Produktseite beschrieben ist. Nun können Sie Ihre Projekte umsetzen. Für den Fall, dass Ihnen noch eine Idee fehlt, in naher Zukunft wird es vom gleichen Autor einen Blogbeitrag zur Überwachung und Steuerung eines Aquariums geben.

Weitere Projekte für AZ-Delivery von mir, finden Sie unter <https://github.com/M3taKn1ght/Blog-Repo>.