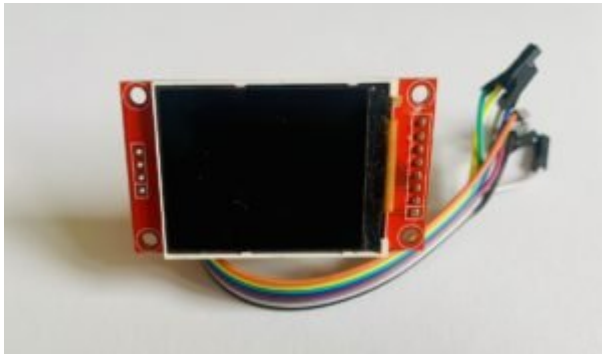


Ein TFT-Display am Arduino anschließen



Wenn dir LCD zu langweilig wird, oder dein Projekt ein Farbdisplay erfordert, wird es Zeit für ein TFT-Display!

Hier lernst du, wie du ein kleines, günstiges SPI-Display richtig mit deinem Arduino verbindest und etwas darauf anzeigen lassen kannst.

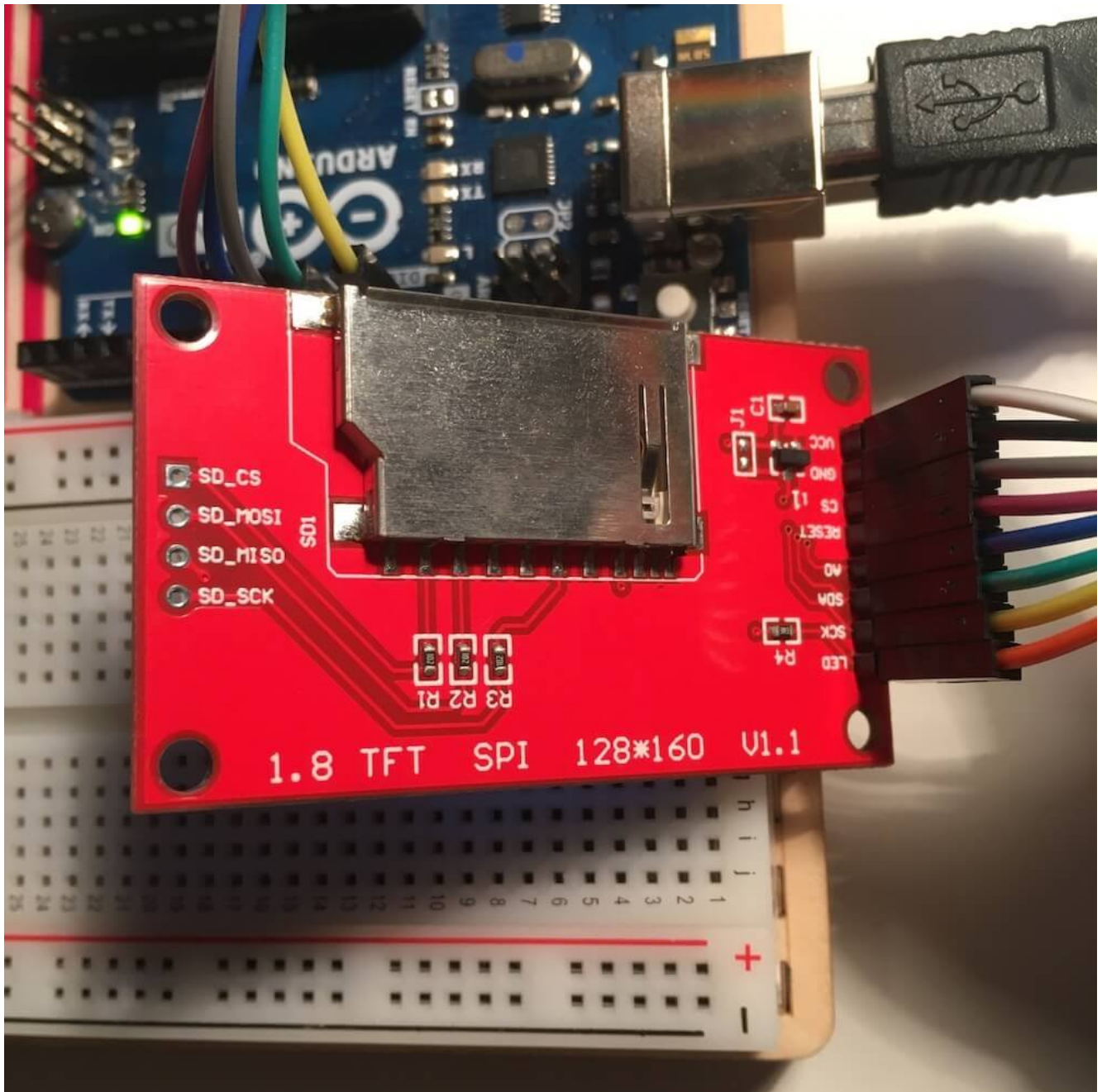
Hinweis: Dieses kleine Tutorial folgt weitestgehend dieser [Anleitung auf randomnerdtutorials.com](http://randomnerdtutorials.com)

Der richtige Anschluss

Das hier verwendete [TFT-Display](#)* hat 8 Pins, die du mit deinem **Arduino Uno** verbinden musst. Folge hierbei dem folgenden Plan:

Pin am TFT-Display	Pin am Arduino
LED	3.3 V
SCK	13
SDA	11
A0 oder DC	9
RESET	8
CS	10
GND	GND
VCC	5 V

Wenn du ein anderes Arduino-Modell verwendest, benötigst du möglicherweise eine andere Belegung der Pins. [In der Dokumentation auf arduino.cc](https://www.arduino.cc/en/documentation) findest du weitere Hinweise.



Rückseite des TFT-Displays

Sobald du das Display richtig angeschlossen hast, kann es auch schon mit dem Programmieren losgehen.

Der Code für dein TFT-Display

Zunächst benötigst du zwei Bibliotheken, die du einbinden musst. Beide sollten bereits in deiner Arduino IDE vorinstalliert sein.

```
include <TFT.h>;  
include <SPI.h>;
```

Die Bibliothek **TFT.h** benötigst du, um auf dem Display schreiben und zeichnen zu können. **SPI.h** hingegen ist für die Kommunikation zwischen Arduino und Display zuständig.

Als nächstes definierst du folgende 3 Pins. Anschließend erstellst du eine Instanz der Bibliothek **TFT** mit dem Namen **TFTscreen**.

```
#define cs 10  
#define dc 9  
#define rst 8
```

```
TFT TFTscreen = TFT(cs, dc, rst);
```

Die Setup-Funktion

Hier rufst du zunächst die Instanz auf, die du gerade erstellt hast.

```
TFTscreen.begin();
```

Danach löschst du das Display, indem du seine Hintergrundfarbe auf Schwarz setzt. Die 3 Zahlen in Klammern repräsentieren die RGB-Werte der Hintergrundfarbe – Rot, Grün, Blau. **(0, 0, 0)** ergibt die Farbe Schwarz. Jeder dieser Zahlen kann einen Wert zwischen 0 und 255 haben – 3x die Null ergibt Schwarz, 3x 255 Weiß. Dazwischen hast du die Möglichkeit von über 16 Millionen Farben. [Lerne hier mehr über RGB](#).

```
TFTscreen.background(0, 0, 0);
```

Zuletzt setzt du die Schriftgröße auf 2. Experimentiere mit

größeren und kleineren Zahlen und beobachte die Zeichen auf deinem Display.

```
TFTscreen.setTextSize(2);
```

Der Loop

Hier wirst du den guten alten Text „Hello, world!“ auf dein Display bringen. Aber nicht nur das, du wirst ihn auch in verschiedenen und zufällig erzeugten Farben anzeigen, um die Möglichkeiten deines Farbdisplays auszunutzen.

Kümmern wir uns zunächst um die zufälligen Farben. Wie du oben gelesen hast, besteht eine RGB-Farbe aus Rot, Grün und Blau. Jede einzelne dieser Farben kann einen Wert zwischen 0 und 255 besitzen. Diesen Wert ermittelst du mit der Funktion **random()** und weist ihn je einer Variablen zu:

```
int redRandom = random(0, 255);  
int greenRandom = random (0, 255);  
int blueRandom = random (0, 255);
```

Anschließend legst du mit der Funktion **TFTscreen.stroke()** die per Zufall ermittelte Farbe des Textes fest und schreibst mit **TFTscreen.text()** in die Mitte des Displays:

```
TFTscreen.stroke(redRandom, greenRandom, blueRandom);  
TFTscreen.text("Hello, World!", 6, 57);
```

Zu guter Letzt baust du noch einen Delay in den Loop ein, mit dem du die Geschwindigkeit des Farbwechsels bestimmst. Also zum Beispiel alle 200 Millisekunden:

```
delay(200);
```

Und das war's! Wenn du den Code jetzt auf deinen Arduino hochlädst, sollte dein TFT-Display „Hello, World!“ in wechselnden Farben anzeigen.

Hier der vollständige Code:

```

/*
Rui Santos
Complete Project Details https://randomnerdtutorials.com
*/

#include <TFT.h>
#include <SPI.h>

#define cs    10
#define dc    9
#define rst   8

TFT TFTscreen = TFT(cs, dc, rst);

void setup() {

    TFTscreen.begin();

    TFTscreen.background(0, 0, 0);

    TFTscreen.setTextSize(2);
}

void loop() {

    int redRandom = random(0, 255);
    int greenRandom = random (0, 255);
    int blueRandom = random (0, 255);

    TFTscreen.stroke(redRandom, greenRandom, blueRandom);

    TFTscreen.text("Hello, World!", 6, 57);

    delay(200);
}

```

Wie geht es weiter?

Weiter Anleitungen zu Bauteilen und vollständige Arduino Projekte findest du in unserem Buch für angehende Maker:



[Vom Anfänger zum Maker: Mit dem Arduino und ESP8266](#)

- Kumbartzki, Frederik (Autor)

[Bei Amazon kaufen](#)

Du kannst nicht nur Text aufs Display bringen, sondern darauf auch zeichnen. Hierfür gibt es in der TFT-Bibliothek bereits Funktionen für Quadrate, Kreise, Rechtecke und so weiter. Auch die große Anzahl an Farben kannst du nutzen – [mit einem Farbsensor kannst du Farben erkennen](#) und sie auf dem TFT-Display anzeigen.

Viele Displays haben auch bereits einen SD-Kartenleser auf der Rückseite. Auf die SD-Karte kannst du zum Beispiel Fotos speichern und sie auf dem Display anzeigen lassen – auch hierfür [findest du in der TFT-Bibliothek](#) bereits eine fertige Funktion, die du verwenden kannst.

Doch lieber LCD? [Lerne hier, wie du ein LCD-Display am Arduino](#)

[anschließt](#).

Fehlen dir noch Bauteile? Dann wirf einen Blick in unsere Übersicht der [besten Arduino Starter Kits](#).