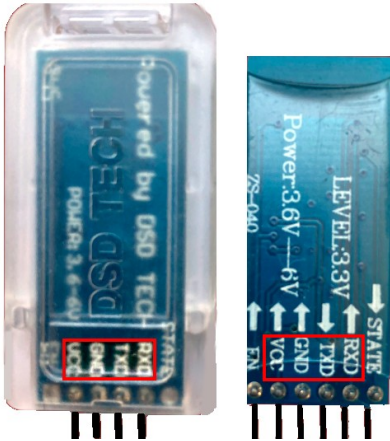


Ziel des Projekts

Eine Ampel soll mit Hilfe eines Bluetooth-Moduls und einer Smartphone-App "per Hand" geschaltet werden.

Die Hardware

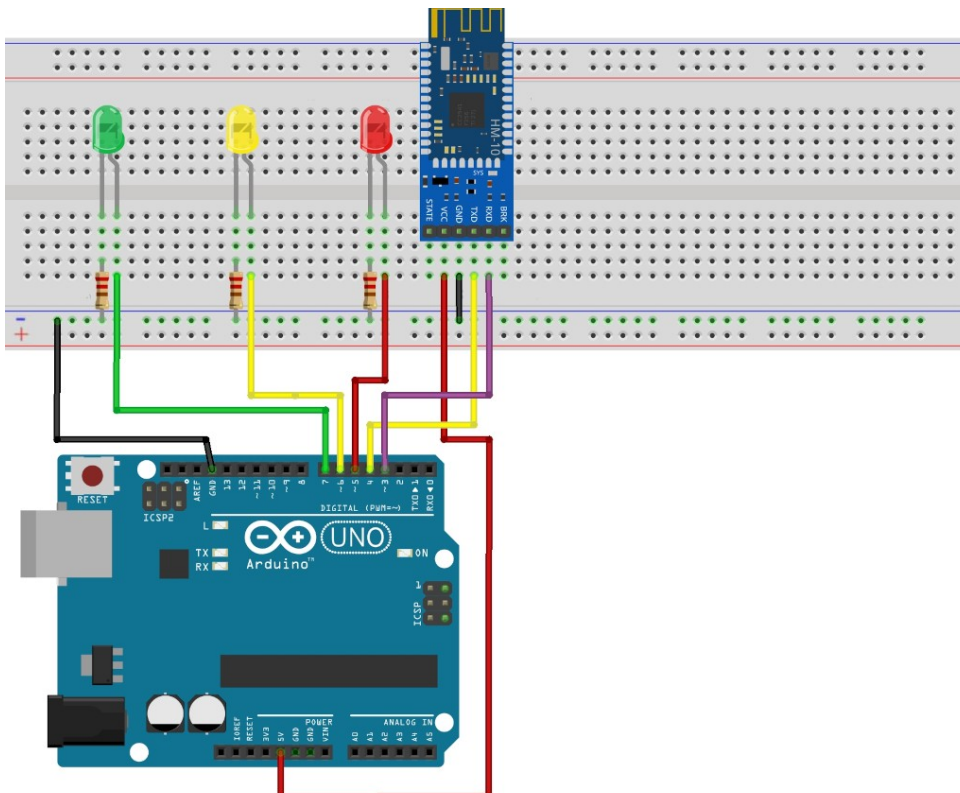


Das Bluetooth-Modul HM-10 arbeitet mit BLE (Bluetooth Low Energy) und kann mit Android-Betriebssystemen und iOS umgehen.

Benötigte Bauteile

- 3 LEDs (rot, gelb, grün)
- 3 Widerstände 220 Ω
- Bluetooth-Modul HM-10
- Leitungsdrähte

Der Schaltplan



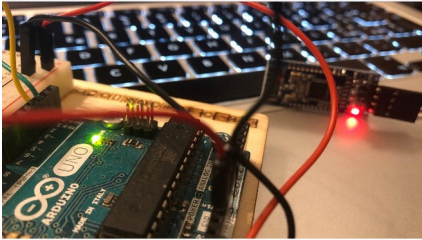


Das Programm

Wenn das Bluetooth-Modul korrekt angeschlossen ist, beginnt die Status-LED zu blinken. Um das Modul nutzen zu können, musst du eine App auf dem Smartphone installieren.

Smartphone-Apps

Arduino Bluetooth Controller



DSD TECH

428AFEC4-0908-483A-9CA4-9B54FFBE62E8

Select a connection mode for your Arduino™

- Switch
send On/Off commands
- Dimmer
use a slider for dimming
- RGB colors controller
fund up your color
- Terminal**
write direct commands

< [Menu]

your command

1 2 3
4 5 6
7 8 9
0 [Back]

Fertig

Tastaturtyp

Standard **Zahlentastatur**

Zeilenvorschub

keiner NL CR CRLF

Befehl nicht löschen ☐

Antworten anzeigen ☒

< [Menu]

rot an!

3

gelb an!

4

gelb aus!

your command

1 2 3
4 5 6
7 8 9
0 [Back]



DSD-Tech Bluetooth

DSD TECH Bluetooth

Scan

Paired Device

DSD TECH

Available devices

DSD TECH(-58dBm) Disconnect

Data

Relay

HEX RX: ☐ ON ☒ OFF

CLEAR

[12:27:55]->rot an!
[12:28:05]->gelb an!
[12:28:15]->gelb aus!
[12:28:24]->rot aus!
[12:28:53]->grün an!

5

SEND

HEX TX: ☐ ON ☒ OFF

1000 ms ☐ Auto Send



Die eigentlich vorgesehenen Pins RX (0) und TX (1) werden beim Hochladen verwendet. Deshalb müsste das Bluetooth-Modul beim Hochladen herausgezogen und anschließend wieder eingesteckt werden. Über die Bibliothek `SoftwareSerial` können die Pins RX und TX anderen digitalen Pins zugeordnet werden. Das Herausziehen und Wiedereinstecken entfällt dann.

Im Kopf des Programms werden die Variablen definiert. Die eingebundene Bibliothek `SoftwareSerial` ist eine Standardbibliothek, sie muss nicht installiert werden.

```
// über Bluetooth vom Smartphone gesendeter Wert
// dieser Wert wird im Programm ausgewertet
char btDaten;

int GRUEN = 7;
int GELB = 6;
int ROT = 5;

# include <SoftwareSerial.h>

// Zuordnung von RX und TX: TX -> 2, RX -> 3
SoftwareSerial BTSerial(2, 3);
```



Der setup-Teil definiert lediglich die pinModes für die LEDs und startet den über SoftwareSerial zugewiesenen Seriellen Monitor BTSerial:

```
void setup()
{
  BTSerial.begin(9600);
  pinMode(ROT, OUTPUT);
  pinMode(GELB, OUTPUT);
  pinMode(GRUEN, OUTPUT);
}
```

Den jeweiligen LEDs wurden in der App Werte zugewiesen, die im Programm mit BTSerial.read() gelesen werden:

Aktion	Taste
rot ein	1
rot aus	2
gelb ein	3
gelb aus	4
grün ein	5
grün aus	6

Im loop-Teil wird den Werten jeweils eine Aktion zugeordnet:

```
void loop()
{
  if (BTSerial.available())
  {
    // vom Smartphone gesendeten Wert lesen
    btDaten = BTSerial.read();

    // rot ein
    if (btDaten == '1')
    {
      digitalWrite(ROT, HIGH);

      // gelesenen Wert in der App anzeigen
      BTSerial.println("rot an!");
    }

    // rot aus
    if (btDaten == '2')
    {
      digitalWrite(ROT, LOW);
      BTSerial.println("rot aus!");
    }
  }
}
```



```
// gelb an
if (btDaten == '3')
{
    digitalWrite(GELB, HIGH);
    BTSerial.println("gelb an!");
}

// gelb aus
if (btDaten == '4')
{
    digitalWrite(GELB, LOW);
    BTSerial.println("gelb aus!");
}

// grün an
if (btDaten == '5')
{
    digitalWrite(GRUEN, HIGH);
    BTSerial.println("gr\u00fcn an!");
}

// grün aus
if (btDaten == '6')
{
    digitalWrite(GRUEN, LOW);
    BTSerial.println("gr\u00fcn aus!");
}
}
```

Hartmut Waller letzte Änderung: 20.11.25