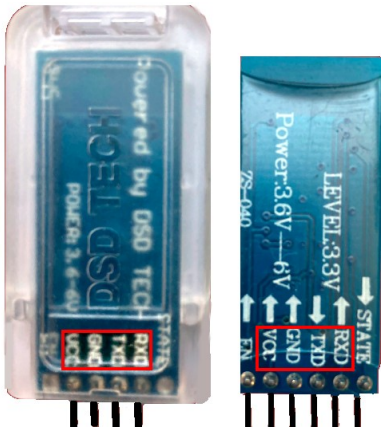
Mit einer Smartphone-App sollen über das Bluetooth-Modul HM-10 LEDs geschaltet werden.

Aktion	Taste
Lauflicht vorwärts	1
Lauflicht rückwärts	2
alle einschalten	3
blinkende LEDs	4
alle ausschalten	8

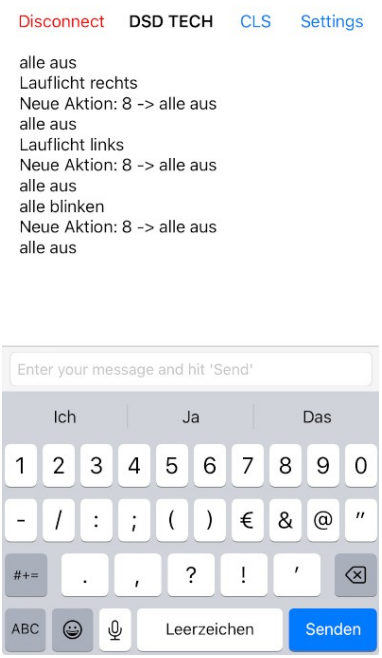
Beispiele für Bluetooth-Module HM-10



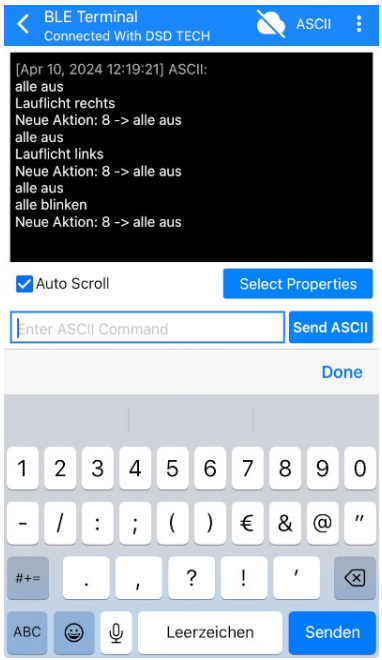
Das Bluetooth-Modul HM-10 arbeitet mit BLE (Bluetooth Low Energy) und kann mit Android-Betriebssystemen und iOS umgehen.

Benötigte Bauteile:

- 5 LEDs
- 3 Widerstände 220 Ω (gelb, rot und grüne LEDs)
- 2 Widerstände 100 Ω (blaue und weiße LEDs)
- Bluetooth-Modul HM-10
- Leitungsdrähte

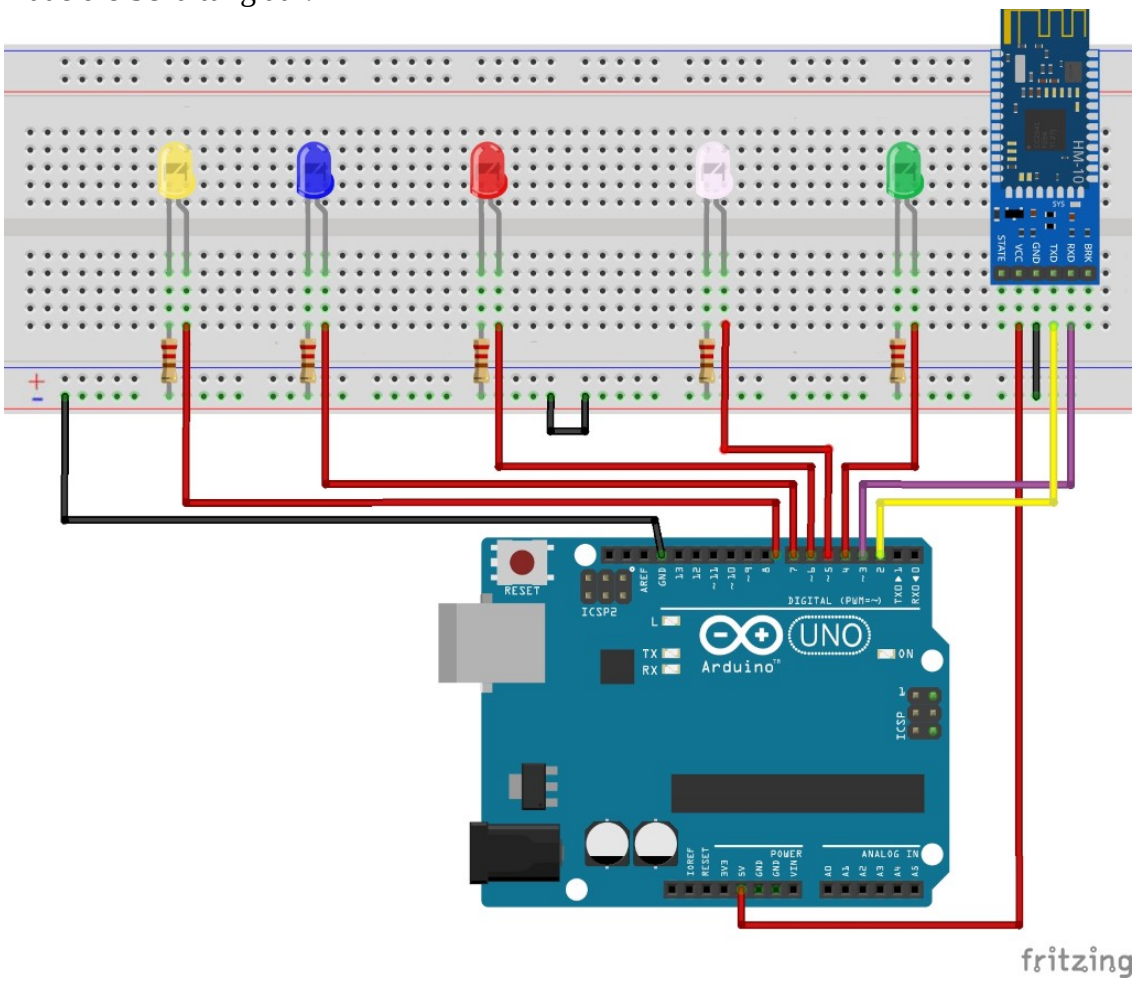


BLESerial tiny



BLE Terminal

Baue die Schaltung auf.



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.

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.

```
# include <SoftwareSerial.h>

// TX -> 2, RX -> 3
SoftwareSerial BTSerial(2, 3);

// Array für die Pins der LEDs
int LED[5] = {4, 5, 6, 7, 8};

// über Bluetooth vom Smartphone gesendeter Wert
char btDaten;
```

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

```
void setup()
{
  BTSerial.begin(9600);
  Serial.begin(9600);

  // pinMode LEDs
  for (int i = 0; i <= 4; i ++)
  {
    pinMode(LED[i], OUTPUT);
  }

  // Menü im Seriellen Monitor
  Serial.println("Men\u00fc:");
  Serial.println("1 -> Lauflicht vor");
  Serial.println("2 -> Lauflicht zur\u00fcck");
  Serial.println("3 -> alle an");
  Serial.println("4 -> blinkende LEDs");
  Serial.println("8 -> alle aus");
}
```

Im loop-Teil liest BTSerial.`read()` den eingegebenen Wert und ordnet ihm die entsprechende Aktion zu.

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

  switch (btDaten)
  {
    case '1':
    {
      BTSerial.println("Lauflicht rechts");
      BTSerial.println("Neue Aktion: 8 -> alle aus");

      // Status true übergeben
      LauflichtRechts(true);
      break;
    }

    case '2':
    {
      BTSerial.println("Lauflicht links");
      BTSerial.println("Neue Aktion: 8 -> alle aus");

      // Status true übergeben
      LauflichtLinks(true);
      break;
    }
```

```
    case '3':
    {
      BTSerial.println("alle an");
      BTSerial.println("Neue Aktion: 8 -> alle aus");

      AlleLEDAus();
      AlleLEDAn();
      break;
    }
    case '4':
    {
      BTSerial.println("alle blinken");
      BTSerial.println("Neue Aktion: 8 -> alle aus");

      // Status true übergeben
      LEDBlinken(true);
      break;
    }

    case '8':
    {
      BTSerial.println("alle aus");
      AlleLEDAus();
      break;
    }
  }
}
```

Die verschiedenen Aktionen werden jeweils in einer Funktion ausgelagert. Beachte die Kommentare.

```
void LauflichtRechts(bool Status)
{
  // Status ist true
  while (Status)
  {
    for (int i = 0; i <= 4; i ++)
    {
      /*
        Eingabe Smartphone abfragen
        wenn Wert 8 (ausschalten):
        -> for-Schleife unterbrechen
        -> Status auf false setzen,
        -> while wird unterbrochen
      */
      btDaten = BTSerial.read();

      if (btDaten == '8')
      {
        Status = !Status;
        BTSerial.println("alle aus");
        AlleLEDAus();
        break;
      }
```

```
      // aktuelle LED i einschalten
      digitalWrite(LED[i], HIGH);
      delay(100);

      // aktuelle LED i ausschalten
      digitalWrite(LED[i], LOW);
    }
  }
}

void LauflichtLinks(bool Status)
{
  // Status ist true
  while (Status)
  {
    for (int i = 4; i >= 0; i --)
    {
      btDaten = BTSerial.read();
      if (btDaten == '8')
      {
        Status = !Status;
        BTSerial.println("alle aus");
        AlleLEDAus();
        break;
      }

      // aktuelle LED i einschalten
      digitalWrite(LED[i], HIGH);
      delay(100);

      // aktuelle LED i ausschalten
      digitalWrite(LED[i], LOW);
    }
  }
}

void AlleLEDAn()
{
  for (int i = 0; i <= 4; i ++)
  {
    // aktuelle LED i einschalten
    digitalWrite(LED[i], HIGH);
  }
}
```

```
void LEDBlinken(bool Status)
{
  // Status ist true
  while (Status)
  {
    for (int i = 0; i <= 4; i ++)
    {
      btDaten = BTSerial.read();
      if (btDaten == '8')
      {
        Status = !Status;
        BTSerial.println("alle aus");
        AlleLEDAus();
        break;
      }

      // aktuelle LED i einschalten
      digitalWrite(LED[i], HIGH);
    }
    delay(100);

    for (int i = 0; i <= 4; i ++)
    {
      btDaten = BTSerial.read();
      if (btDaten == '8')
      {
        Status = !Status;
        BTSerial.println("alle aus");
        AlleLEDAus();
        break;
      }

      // aktuelle LED i ausschalten
      digitalWrite(LED[i], LOW);
    }
    delay(100);
  }
}

void AlleLEDAus()
{
  for (int i = 0; i <= 4; i ++)
  {
    // aktuelle LED i ausschalten
    digitalWrite(LED[i], LOW);
  }
}
```