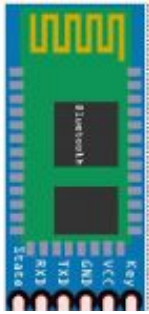


LEDs mit Bluetooth schalten



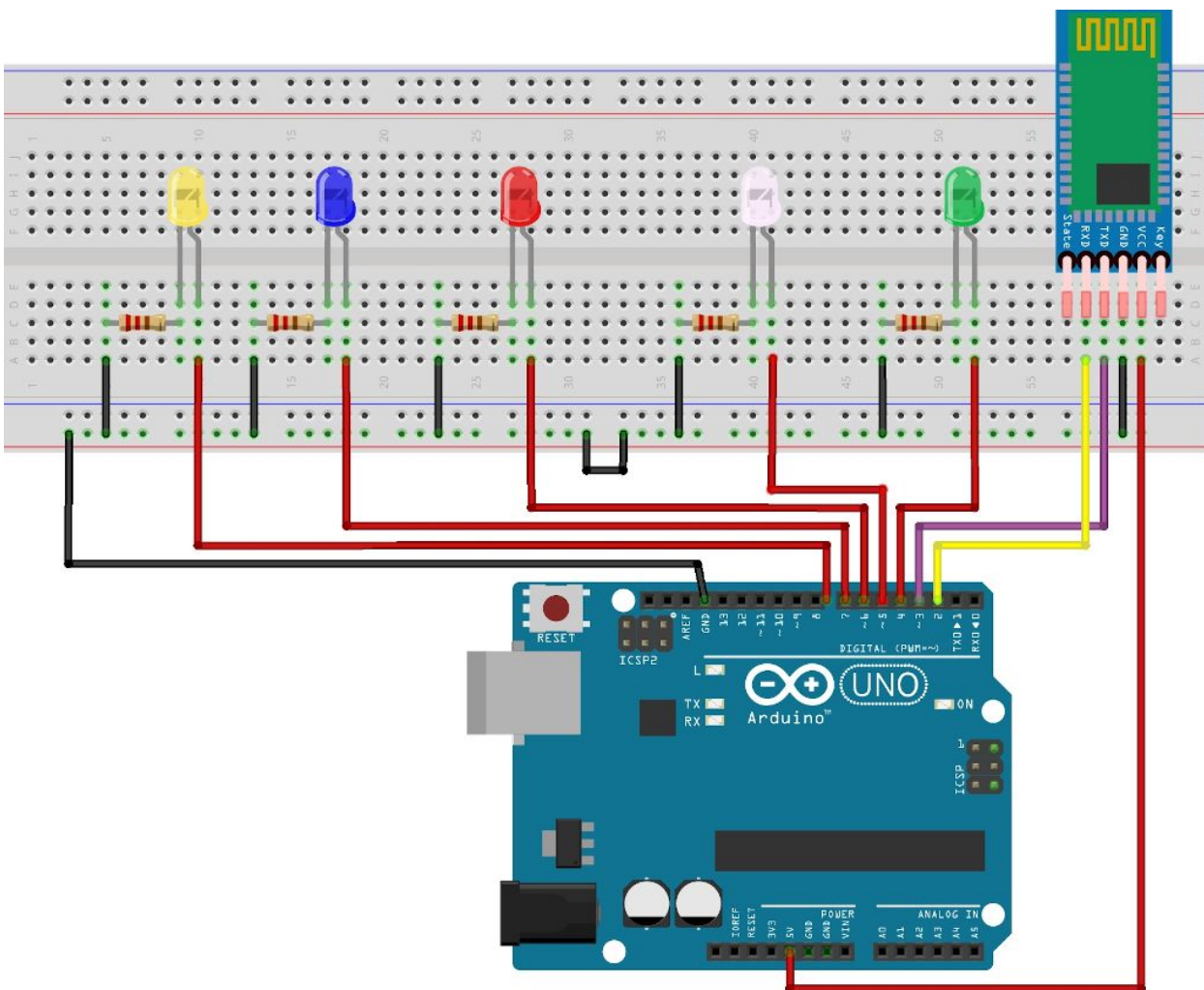
Die Bluetooth-Module HC-05 und HM-10 unterscheiden sich in der Verwendung von Apps für Android und iOS. Die Programmierung ist identisch.

Der HC-05 und funktioniert nur mit Smartphones mit Android-Betriebssystem, der HM-10 kann mit dem Android-Betriebssystem und iOS umgehen. Apple-Geräte verwenden BLE (Bluetooth Low Energy), mit dem der HC-05 nicht arbeiten kann.

Benötigte Bauteile:

- ➔ 5 LEDs
- ➔ 5 Widerstände > 100 Ω
- ➔ Bluetooth-Modul HC05/HM-10
- ➔ Leitungsdrähte

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.

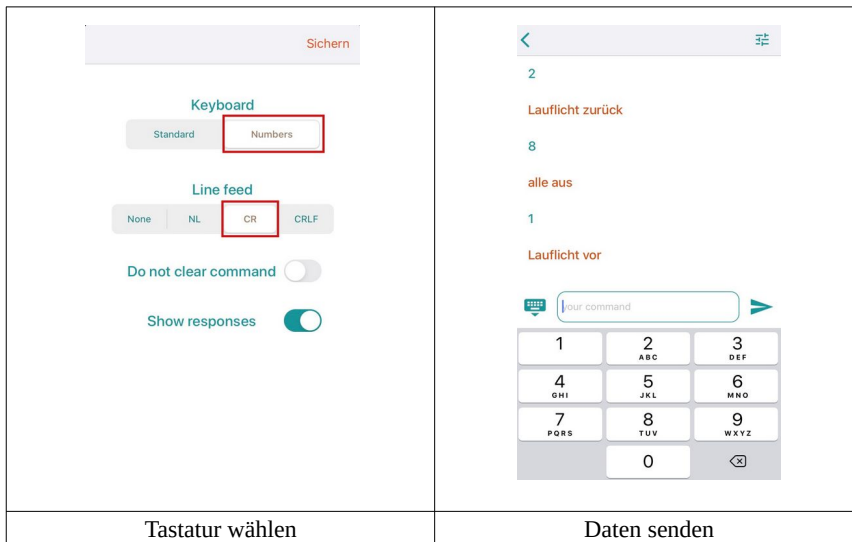
Für Android und iOS verfügbare Apps:

DSD-Tech Bluetooth

<p>Bluetooth-Modul scannen</p>	<p>Daten senden</p>

Arduino Bluetooth Controller

<p>Bluetooth-Modul scannen</p>	<p>Terminal auswählen</p>	<p>Einstellungen</p>



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);

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\u00f6cck");
Serial.println("3 -> alle an");
Serial.println("4 -> blinkende LEDs");
Serial.println("8 -> alle aus");
}
    
```

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

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

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

```

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

    switch (btDaten)
    {
        case '1':
        {
            BTSerial.print("Lauflicht vor");

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

        case '2':
        {
            BTSerial.print("Lauflicht zur\u00f6cck");

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

        case '3':
        {
            BTSerial.print("alle an");
            AlleLEDAus();
            AlleLEDAn();
            break;
        }
    }
}
    
```

```
case '4':
{
  BTSerial.print("alle blinken");

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

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

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

```
void LauflichtVor(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.print("alle aus");
        AlleLEDAus();
        break;
      }

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

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

void LauflichtZurueck(bool Status)
{
    // Status ist true
    while (Status)
    {
        for (int i = 4; i >= 0; i --)
        {
            btDaten = BTSerial.read();
            if (btDaten == '8')
            {
                Status = !Status;
                BTSerial.print("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.print("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.print("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);
    }
}
```