



Ziel des Projekts

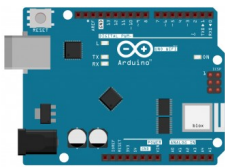
Mit der Bluetooth-Funktion eines Arduinos mit WiFi sollen die einzelnen LEDs über eine Smartphone-App gesteuert werden. Zusätzlich kann eine automatische Ampelschaltung gestartet werden.

Den jeweiligen LEDs wurden in der App Werte zugewiesen:

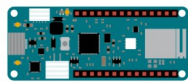
Aktion	Taste
rot ein	10
gelb ein	11
grün ein	12
rot aus	20
gelb aus	21
grün aus	22
automatische Ampelschaltung	30

Die Hardware

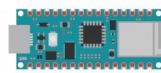
Für diese Anleitung benötigst du einen Arduino mit WiFi.



Arduino WiFi Rev 2



Arduino MKR WiFi 1010



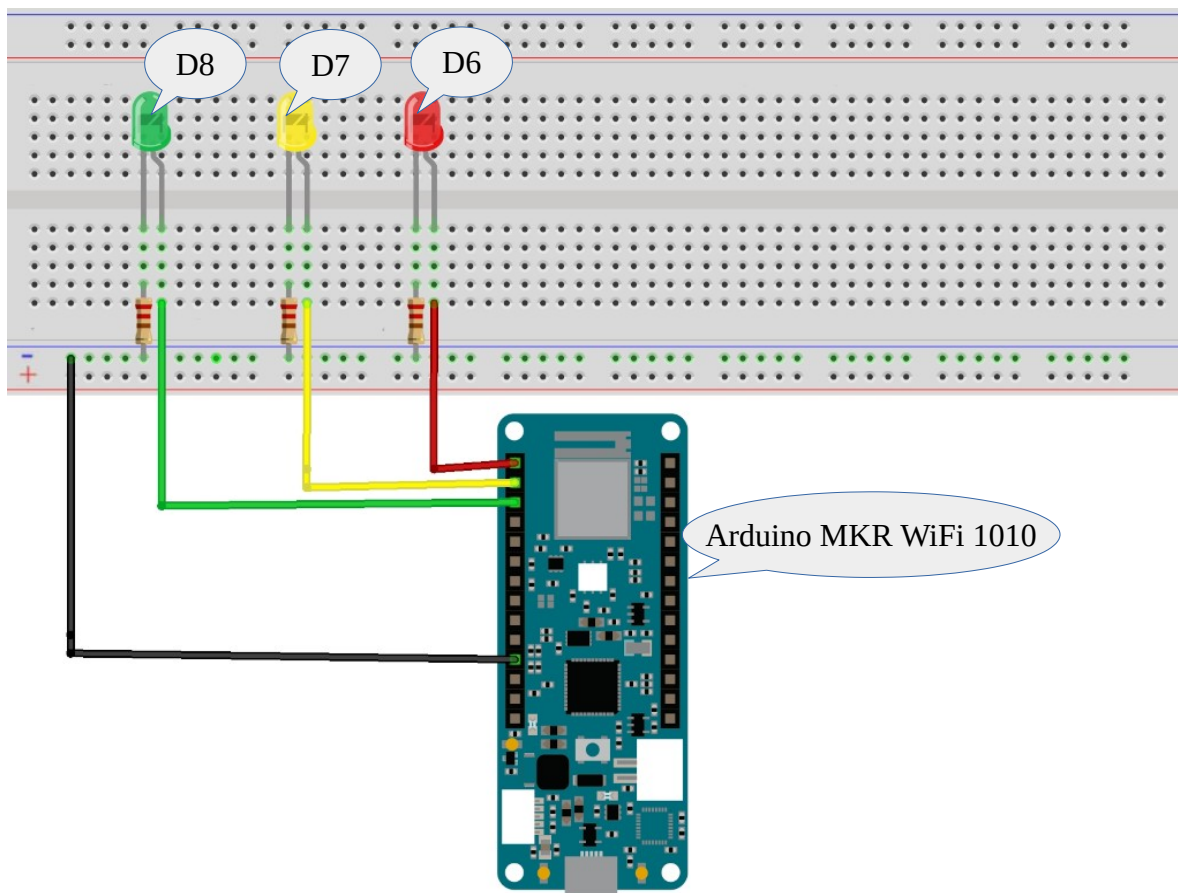
Arduino Nano 33 IoT

Benötigte Bauteile

- ➡ 3 LEDs (rot, gelb, grün)
- ➡ 3 Widerstände 220 Ω
- ➡ Leitungsdrähte



Der Schaltplan



fritzing

Smartphone-Apps

Für Android und iOS verfügbare App:

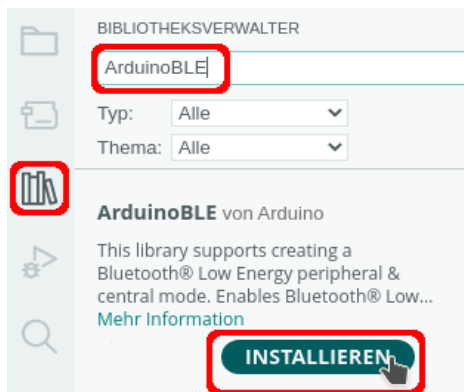
LightBlue

Bluetooth-Modul scannen	UUID auswählen	Wert schreiben



Typ des Werts festlegen	Daten senden

Benötigte Bibliothek



Das Programm

Binde die benötigte Bibliothek ein und definiere die Variablen:

```
#include "ArduinoBLE.h"

/*
  eindeutige UUID bestimmen:
  https://www.guidgenerator.com/online-guid-generator.aspx
  https://www.uuidgenerator.net/
  BLERead | BLEWrite | BLENotify
  -> schreiben, lesen, Info
*/
// Name BLE-Service
BLEService LEDSchalten("19b10000-e8f2-537e-4f6c-d104768a1214");

BLEUnsignedCharCharacteristic Auswahl("19b10000-e8f2-537e-4f6c-d104768a1214", BLERead | BLEWrite | BLENotify);
enum Ampel
{
  ROT = 6,
  GELB,
  GRUEN
};
```



Der setup-Teil:

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

  // pinmode festlegen
  for (int i = ROT; i <= GRUEN; i++)
  {
    pinMode(i, OUTPUT);
  }

  // BLE starten
  if (!BLE.begin()) Serial.println("Bluetooth-Modul konnte nicht gestartet werden!");
  else Serial.println("Bluetooth-Modul erfolgreich gestartet!");

  // Name festlegen (wird in der App angezeigt) und den Service (LEDSchalten) zuweisen
  BLE.setLocalName("LEDS schalten");
  BLE.setAdvertisedService(LEDSchalten);

  // Auswahl als Platzhalter für den in der App gewählten Wert
  LEDSchalten.addCharacteristic(Auswahl);

  // Service LEDSchalten hinzufügen
  BLE.addService(LEDSchalten);

  // Startwert für die Kommunikation schreiben
  Auswahl.writeValue(0);

  // Zuweisung starten
  BLE.advertise();
}
```

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

```
void loop()
{
  // auf die Verbindung zu Geräten warten
  BLEDevice Verbindung = BLE.central();

  // wenn der Arduino mit einem Gerät verbunden ist ...
  if (Verbindung)
  {
    Serial.println("Verbunden ... ");
  }
}
```



```
// solange der Arduino mit einem Gerät verbunden ist ...
while (Verbindung.connected())
{
    /*
    Dezimal -> Hex-Umrechnung
    10 -> A
    11 -> B
    12 -> C
    20 -> 14
    21 -> 15
    22 -> 16
    30 -> 1E
    */
    if (Auswahl.written())
    {
        if (Auswahl.value() == 10)
        {
            Serial.print(Auswahl.value());
            Serial.println(" -> rot ein");
            digitalWrite(ROT, HIGH);
        }

        if (Auswahl.value() == 11)
        {
            Serial.print(Auswahl.value());
            Serial.println(F(" -> gelb ein"));
            digitalWrite(GELB, HIGH);
        }

        if (Auswahl.value() == 12)
        {
            Serial.print(Auswahl.value());
            Serial.println(F(" -> gr\u00fcn ein"));
            digitalWrite(GRUEN, HIGH);
        }

        if (Auswahl.value() == 20)
        {
            Serial.print(Auswahl.value());
            Serial.println(F(" -> rot aus"));
            digitalWrite(ROT, LOW);
        }

        if (Auswahl.value() == 21)
        {
            Serial.print(Auswahl.value());
            Serial.println(F(" -> gelb aus"));
            digitalWrite(GELB, LOW);
        }
    }
}
```



```
if (Auswahl.value() == 22)
{
  Serial.print(Auswahl.value());
  Serial.println(F(" -> gr\u00fcn aus"));
  digitalWrite(GRUEN, LOW);
}

// Ampel schalten
if (Auswahl.value() == 30)
{
  Serial.print(Auswahl.value());
  Serial.println(F(" -> Ampel schalten"));
  digitalWrite(ROT, HIGH);
  delay(5000);
  digitalWrite(GELB, HIGH);
  delay(1000);
  digitalWrite(ROT, LOW);
  digitalWrite(GELB, LOW);
  digitalWrite(GRUEN, HIGH);
  delay(3000);
  digitalWrite(GRUEN, LOW);
  digitalWrite(GELB, HIGH);
  delay(1000);
  digitalWrite(GELB, LOW);
  digitalWrite(ROT, HIGH);
}
}
}
}
```