

Lauflicht mit der Keyes-Fernbedienung



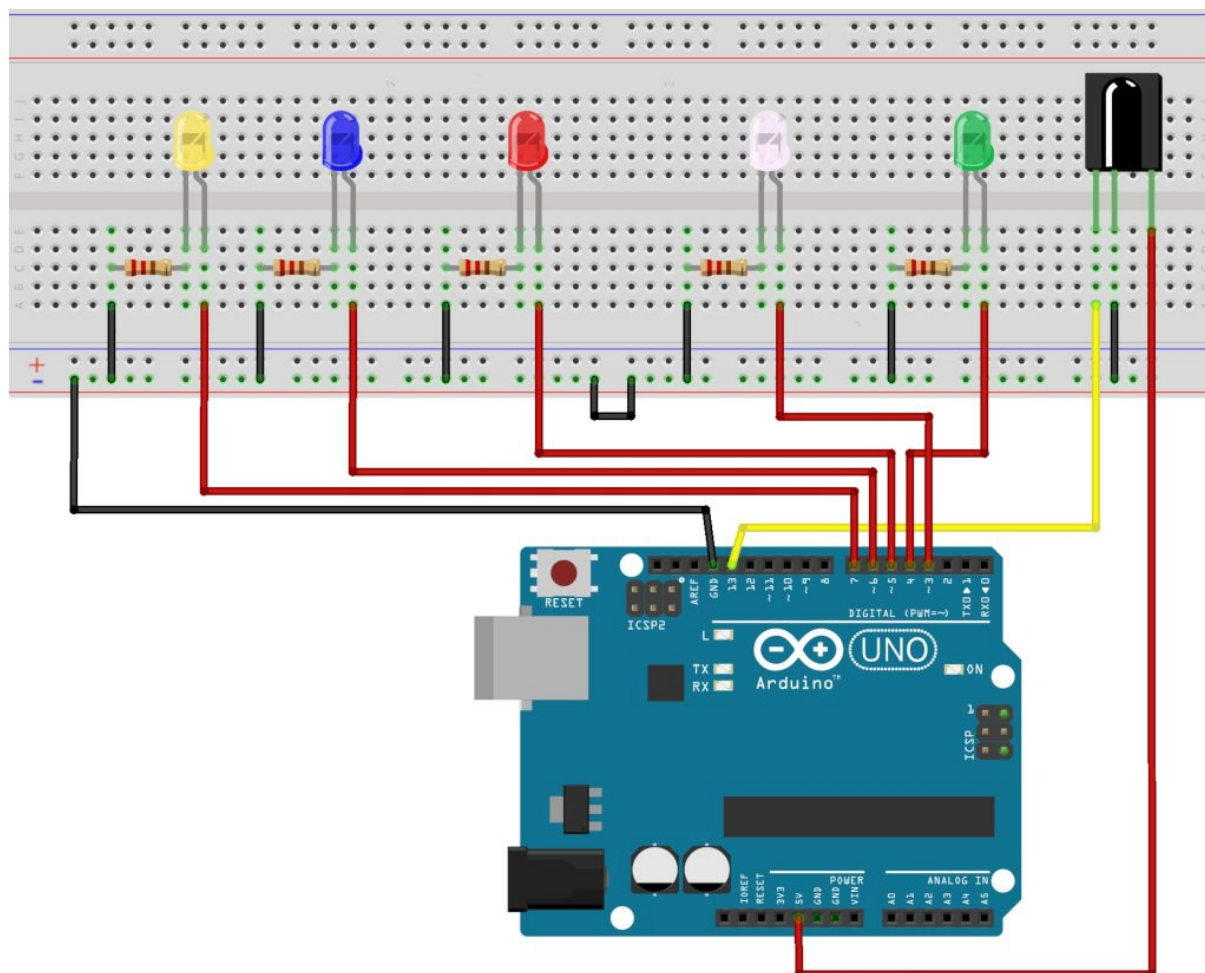
Die Tasten der Fernbedienung sollen verwendet werden um das Lauflicht zu steuern.

Taste 1	→	die LEDs leuchten nacheinander für 100 ms
Taste 2	→	die LEDs leuchten nacheinander für 100 ms vor und zurück
Taste 3	→	LEDs blinken nacheinander
Taste nach oben	→	Anzahl der Wiederholungen erhöhen
Taste nach unten	→	Anzahl der Wiederholungen verringern

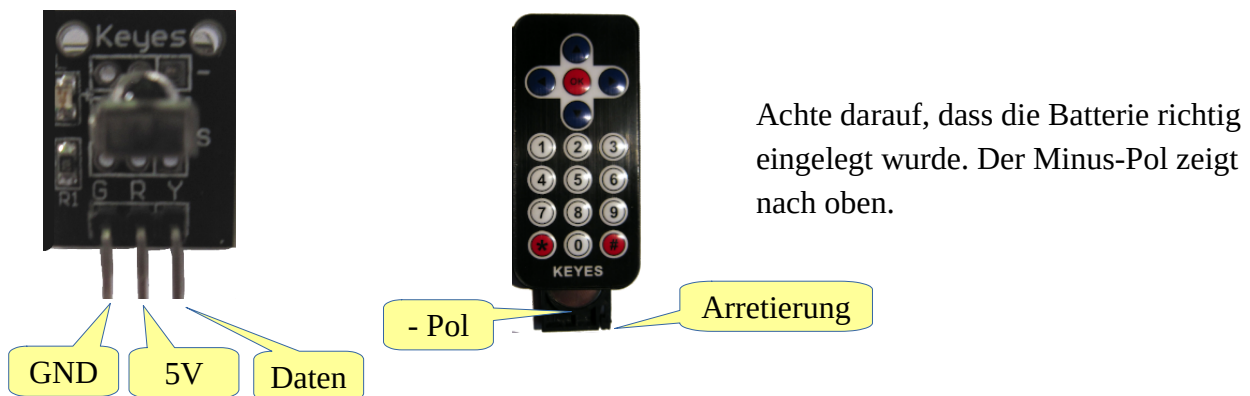
Benötigte Bauteile:

- 5 LEDs
- 3 Widerstände 220 Ω (gelb, rot und grüne LEDs)
- 2 Widerstände 100 Ω (blaue und weiße LEDs)
- Infrarot-Empfänger
- Fernbedienung
- Leitungsdrähte

Baue die Schaltung auf.



Die Pin-Belegung des Infrarot-Empfängers:

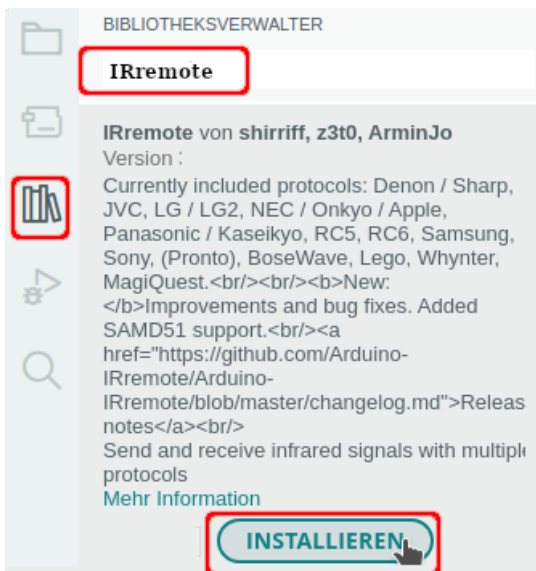


Die Tastencodes beziehen sich auf die **Keyes-Fernbedienung**. Die Tastencodes anderer Fernbedienungen kannst du mit Hilfe des Seriellen Monitors herausfinden. Außerdem werden die Codes nicht immer zuverlässig gelesen, deshalb musst du manchmal die Taste mehrmals drücken. **Die Tastencodes der Keyes Fernbedienung.**

Bevor du starten kannst, musst du eine Bibliothek installieren:

Sketch → Bibliothek einbinden → Bibliotheken verwalten

Suche nach der Bibliothek **IRremote**, bewege die Maus in das entsprechende Feld und ...



... klicke auf **Installieren**.

Die Tastencodes:

oben	links	rechts	unten	OK	1	2	3	4
16736925	16720605	16761405	16754775	16712445	16738455	16750695	16756815	16724175
5	6	7	8	9	*	0	#	
16718055	16743045	16716015	16726215	16734885	16728765	16730805	16732845	

Die Zahlencodes kannst du mit folgendem Programm herausfinden:

Die Tastencodes werden im Seriellen Monitor angezeigt.

```
// benötigte Bibliothek einbinden
# include <IRremote.h>

// der Pin, an dem der Infrarot-Empfänger angeschlossen ist
int EmpfaengerPin = 2;

void setup()
{
    // Seriellen Monitor starten
    Serial.begin(9600);

    // Infrarot-Empfänger starten
    IrReceiver.begin(EmpfaengerPin);
}

void loop()
{
    // decode() -> Daten lesen
    if (IrReceiver.decode())
    {
        // kurzes delay, damit nur ein Tastendruck gelesen wird

        delay(200);

        // resume -> nächsten Wert lesen
        IrReceiver.resume();

        /*
        der Empfänger empfängt zwischendurch Signale,
        die nicht ausgewertet werden können
        es sollen deshalb nur die korrekt erkannten Tasten ausgewertet werden
        die Dezimalwerte der korrekten erkannten Tasten liegen zwischen > 0 und < 95
        es wird abgefragt, ob das empfangene Kommando decodedIRData.command
        zwischen 0 und (&&) 95 liegt
        */
    }
}
```

```
if (IrReceiver.decodedIRData.command > 0 && IrReceiver.decodedIRData.command < 95)
{
    Serial.print("Dezimalwert: ");

    // IrReceiver.decodedIRData.command = Wert der gedrückten Taste
    Serial.print(IrReceiver.decodedIRData.command);
    Serial.print(" -> ");

    // Werte abfragen und anzeigen
    if (IrReceiver.decodedIRData.command == 22) Serial.println("Taste 1");
    if (IrReceiver.decodedIRData.command == 25) Serial.println("Taste 2");
    if (IrReceiver.decodedIRData.command == 13) Serial.println("Taste 3");
    if (IrReceiver.decodedIRData.command == 12) Serial.println("Taste 4");
    if (IrReceiver.decodedIRData.command == 24) Serial.println("Taste 5");
    if (IrReceiver.decodedIRData.command == 94) Serial.println("Taste 6");
    if (IrReceiver.decodedIRData.command == 8) Serial.println("Taste 7");
    if (IrReceiver.decodedIRData.command == 28) Serial.println("Taste 8");
    if (IrReceiver.decodedIRData.command == 90) Serial.println("Taste 9");
    if (IrReceiver.decodedIRData.command == 82) Serial.println("Taste 0");
    if (IrReceiver.decodedIRData.command == 66) Serial.println("Taste *");
    if (IrReceiver.decodedIRData.command == 74) Serial.println("Taste #");
    if (IrReceiver.decodedIRData.command == 68) Serial.println("Pfeil links");
    if (IrReceiver.decodedIRData.command == 67) Serial.println("Pfeil rechts");
    if (IrReceiver.decodedIRData.command == 70) Serial.println("Pfeil oben");
    if (IrReceiver.decodedIRData.command == 21) Serial.println("Pfeil unten");
    if (IrReceiver.decodedIRData.command == 64) Serial.println("OK");
}
}
```

Ausgabe Serieller Monitor ✕

Nachricht (STRG + Enter, um Nachricht an 'Arduino Uno' auf '/c

Sowohl NL als... ▼ 9600 baud ▼

```
Dezimalwert: 70 -> Pfeil oben
Dezimalwert: 68 -> Pfeil links
Dezimalwert: 64 -> OK
Dezimalwert: 67 -> Pfeil rechts
Dezimalwert: 21 -> Pfeil unten
Dezimalwert: 22 -> Taste 1
Dezimalwert: 25 -> Taste 2
Dezimalwert: 13 -> Taste 3
Dezimalwert: 12 -> Taste 4
Dezimalwert: 24 -> Taste 5
Dezimalwert: 94 -> Taste 6
Dezimalwert: 8 -> Taste 7
Dezimalwert: 28 -> Taste 8
Dezimalwert: 90 -> Taste 9
Dezimalwert: 66 -> Taste *
Dezimalwert: 82 -> Taste 0
Dezimalwert: 74 -> Taste #
```

Beispiel:

Durch Drücken der Taste Pfeil nach oben wird eine LED eingeschaltet, Pfeil nach unten schaltet sie wieder aus.

```
# include <IRremote.h>

int EmpfaengerPin = 2;

int LED = 3;

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

  // auf serielle Verbindung warten
  while (!Serial) {;}

  // Empfänger starten
  IrReceiver.begin(EmpfaengerPin);

  pinMode(LED, OUTPUT);
}
```

```
void loop()
{
  // Daten lesen
  if (IrReceiver.decode())
  {
    // oben
    if (IrReceiver.decodedIRData.command == 0x46)
    {
      digitalWrite(LED, HIGH);
    }

    // unten
    else if (IrReceiver.decodedIRData.command == 0x15)
    {
      digitalWrite(LED, LOW);
    }

    delay(200);

    // nächsten Wert lesen
    IrReceiver.resume();
  }
}
```

Binde die benötigte Bibliothek ein und definiere die Variablen.

```
# include <IRremote.h>

// Pin für den Auslöser des Interrupts
int InterruptPin = 2;

// Array mit 5 Elementen und den zugehörigen Ports
int LED[5] = {3, 4, 5, 6, 7};

// Anzahl der LEDs feststellen
int LEDMax = sizeof(LED) / sizeof(LED[0]);

// Variable im Flash-Speicher ablegen
volatile bool Status = true;

int Leuchtdauer = 100;
```

Der setup-Teil. Beachte die Kommentare

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

    // auf serielle Verbindung warten
    while (!Serial) {}

    // Zufallsgenerator starten
    randomSeed(analogRead(0));

    // Empfänger starten

    IrReceiver.begin(InterruptPin);

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

    // Funktion Schalten() dem Interrupt-Pin zuordnen
    attachInterrupt(digitalPinToInterrupt(InterruptPin), Schalten, FALLING);
}
```

Im loop-Teil musst du zunächst die Tastencodes abfragen und anschließend die Tastencodes mit switch auswerten:

```
void loop()
{
    // Daten lesen
    if (IrReceiver.decodedIRData.address == 0)
    {
        if (IrReceiver.decode())
        {
            delay(200);

            Status = true;
            /*
             * solange der Status true ist, wird die jeweilige while-Schleife
             * ausgeführt, ein weiterer Druck auf eine Taste der Fernbedienung löst
             * den Interrupt aus
             * -> Status wird zu false, die while-Schleife wird nicht erneut
             * ausgeführt
             */
            switch (IrReceiver.decodedIRData.command)
            {
                // Taste 1: Lauflicht vor
                case 0x16:
                    while (Status) LauflichtHin();
                    break;
```

```
// Taste 2: LEDs leuchten vor und zurück
case 0x19:
    while (Status) LauflichtHinUndHer();
    break;

// Taste 3: LEDs blinken nacheinander
case 0xD:
    while (Status) LauflichtMitBlinken();
    break;
}

// nächsten Wert lesen
IrReceiver.resume();
}
}
```

Hartmut Waller letzte Änderung: 19.12.22