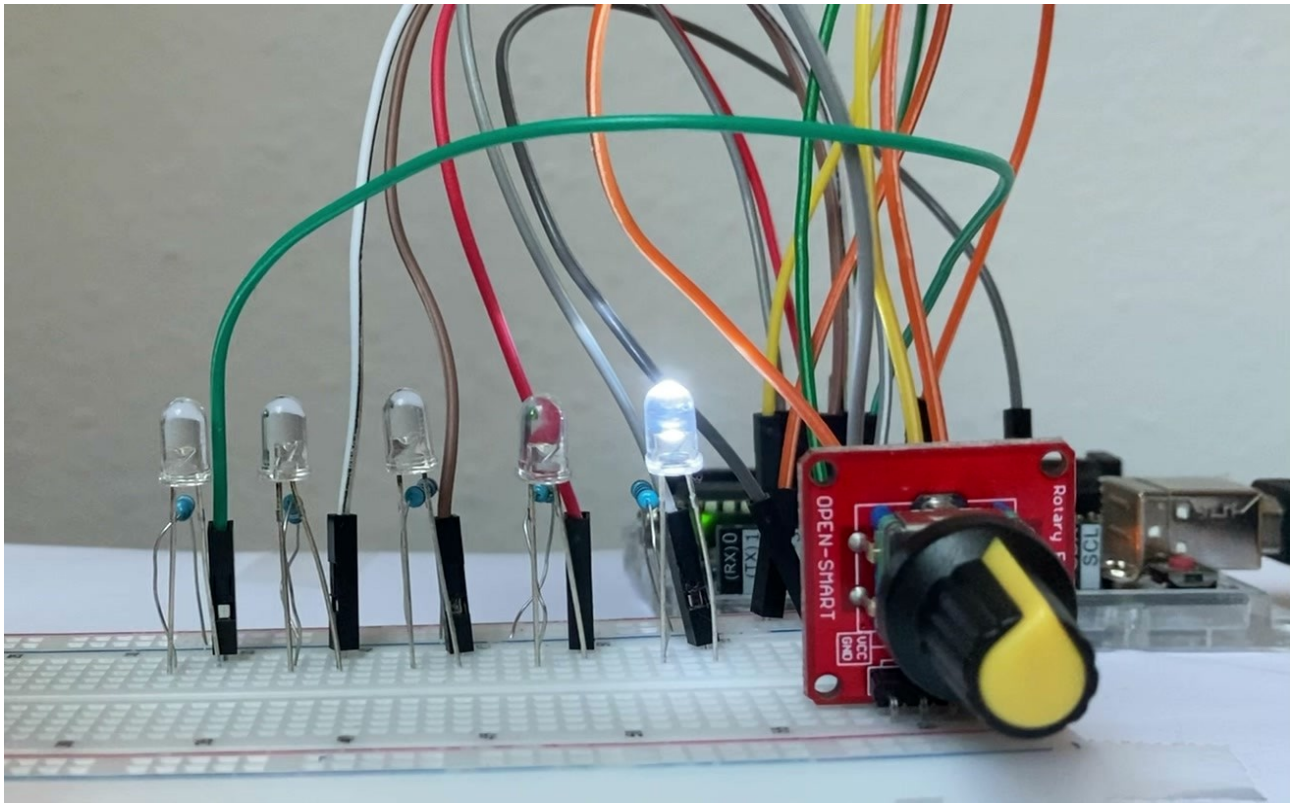
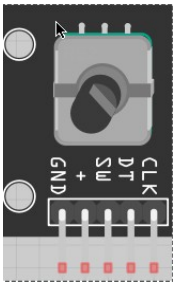
Die Geschwindigkeit eines Lauflichts soll durch einen Drehgeber (Rotary-Encoder) gesteuert werden. Die Drehung vorwärts beschleunigt das Lauflicht, die Drehung rückwärts verlangsamt das Lauflicht. Durch die Kopplung eines Pins an einen Interrupt kann das Lauflicht durch Drehen des Drehgebers unterbrochen und neu gestartet werden.





Eine mechanische Bewegung wird in elektrische Impulse umgesetzt. Hier wird ein 2-Kanal-Drehgeber eingesetzt, der Vorwärts- und Rückwärtsbewegungen erkennen kann.

Der Interrupt soll bei CHANGE ausgelöst werden, weil eine Drehbewegung erkannt wurde:

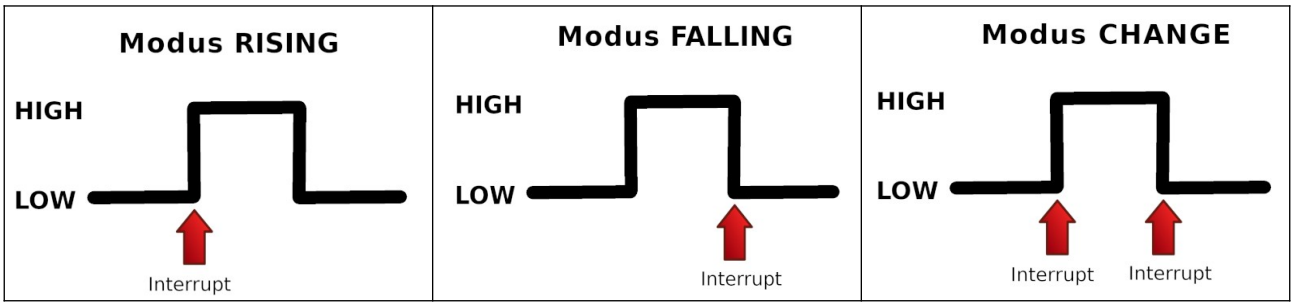
```
  attachInterrupt(digitalPinToInterrupt(EncoderPinA), DrehgeberLesen, CHANGE);
```

Ein Pin des Drehgebers wird dem Interrupt zugeordnet (`attachInterrupt`). Eine Drehbewegung löst den Interrupt aus. Der normale Programmablauf wird unterbrochen und die festgelegte Methode (Interrupt-Service-Routine) wird ausgeführt. Anschließend wird das Programm normal fortgesetzt.

```
attachInterrupt(digitalPinToInterrupt(EncoderPinA), DrehgeberLesen, CHANGE);
```

Die Drehbewegung löst den Interrupt aus, die Interrupt-Service-Routine `EncoderLesen` wird aufgerufen.
Es gibt verschiedene Ereignisse, die den Interrupt auslösen können:

RISING	der Interrupt wird ausgelöst wenn sich der Status von LOW zu HIGH ändert
FALLING	der Interrupt wird ausgelöst wenn sich der Status von HIGH zu LOW ändert
CHANGE	der Interrupt wird ausgelöst wenn sich der Status ändert



Ein Pin des Drehgebers muss zwingend an Pin 2 oder 3 angeschlossen werden.

Zusätzlich muss eine Variable, die im Hauptprogramm und in Interruptroutinen verwendet wird, als **volatile** definiert werden.
Variable werden im SRAM und temporär in internen Registern des Prozessors gespeichert, bearbeitet oder verändert. Es kann vorkommen, dass der Wert der Variablen im SRAM durch eine neue Wertzuweisung für kurze Zeit nicht stimmt, da der letzte Wert zunächst nur in den Prozessorregistern liegt.
Das Schlüsselwort **volatile** weist das Programm an, die Variable immer aktuell im SRAM zu halten. Damit wird garantiert, dass der jeweils aktuelle Wert geladen wird.

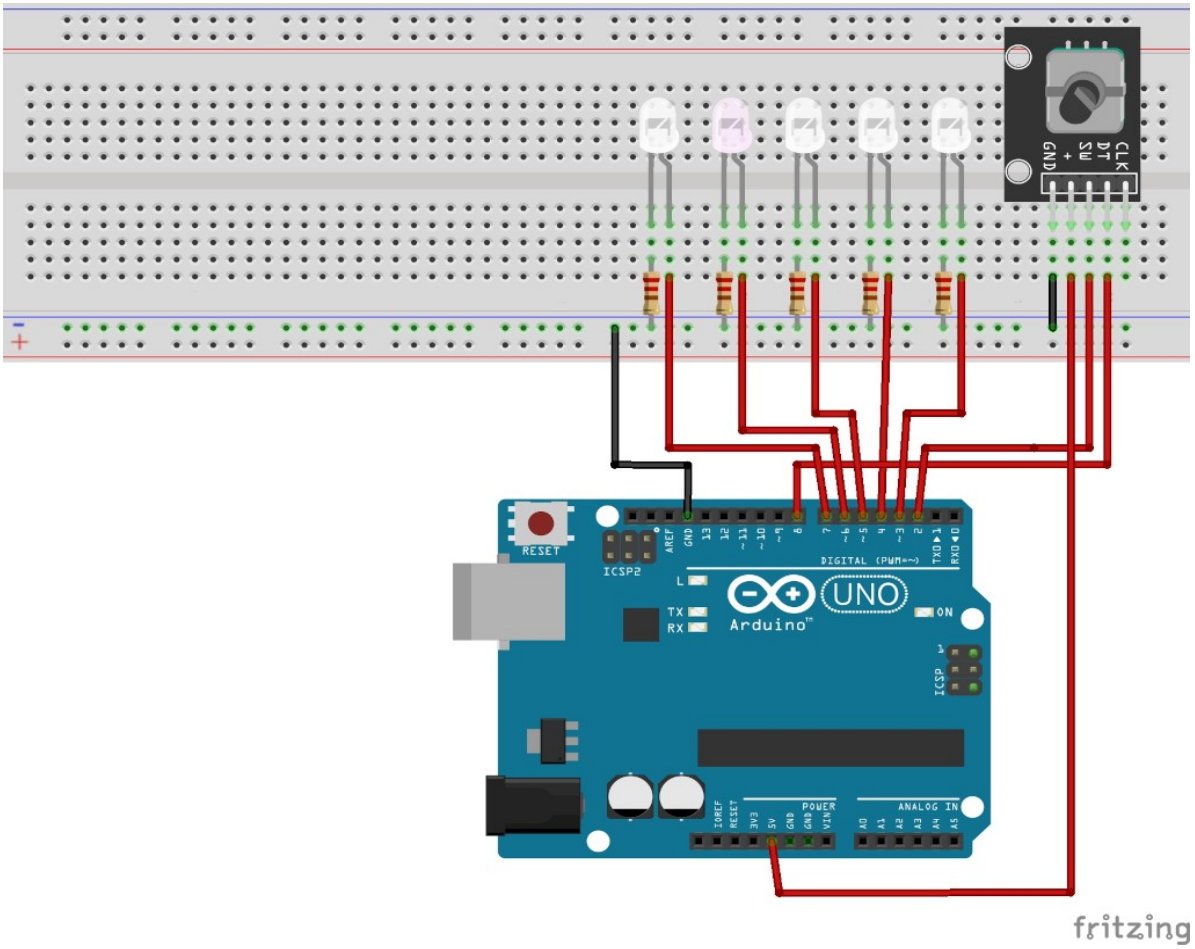
Außerdem sind bei der Programmierung einer Interrupt-Methode einige Regeln zu beachten:

- Der Programmteil muss so kurz wie möglich sein.
- Verwende kein `delay()`.
- Benutze auch kein `Serial.print()`.

Benötigte Bauteile:

- 5 LEDs
- 5 Widerstände 100 Ω
- Drehgeber
- Leitungsdrähte

Baue die Schaltung auf.



Definiere die Variablen:

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

// Anzahl der LEDs feststellen
byte LEDMax = sizeof(LED);

// Interruptpin
# define EncoderPinA 2
# define EncoderPinB 8
// Leuchtdauer beim Start: 100 Millisekunden
volatile unsigned int Leuchtdauer = 100;

// Sprung in Schritten, wenn eine Bewegung erkannt wird
int SprungSchritte = 10;
```

Der setup-Teil:

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

```
  pinMode(EncoderPinA, INPUT_PULLUP);
  pinMode(EncoderPinB, INPUT_PULLUP);

  // Interrupt-Pin EncoderPinA zuordnen
  attachInterrupt(digitalPinToInterrupt(EncoderPinA), DrehgeberLesen, CHANGE);
}
```

Die durch den Interrupt ausgelöste Methode DrehgeberLesen(). Beachte die Kommentare.

```
void DrehgeberLesen()
{
  /*
    wenn beide Drehgeber-Pins gleich sind -> Bewegung vorwärts
    sind sie unterschiedlich -> Bewegung rückwärts.
    Drehung nach links -> Leuchtdauer vergrößern
    Drehung nach rechts -> Leuchtdauer verkleinern
  */
  if (digitalRead(EncoderPinA) == digitalRead(EncoderPinB))
  {
    Leuchtdauer += SprungSchritte;
  }
  else Leuchtdauer -= SprungSchritte;
  if (Leuchtdauer < 100) Leuchtdauer = 100;
}
```

Der loop-Teil startet das Lauflicht.

```
void loop()
{
  // Lauflicht starten
  for (int i = 0; i < LEDMax; i++)
  {
    Lauflicht();
  }
}
```

Die Funktion Lauflicht():

```
void Lauflicht()
{
  for (int i = 0; i < LEDMax; i++)
  {
    // aktuelle LED i einschalten
    digitalWrite(LED[i], HIGH);
    delay(Leuchtdauer);
    digitalWrite(LED[i], LOW);
  }
}
```