

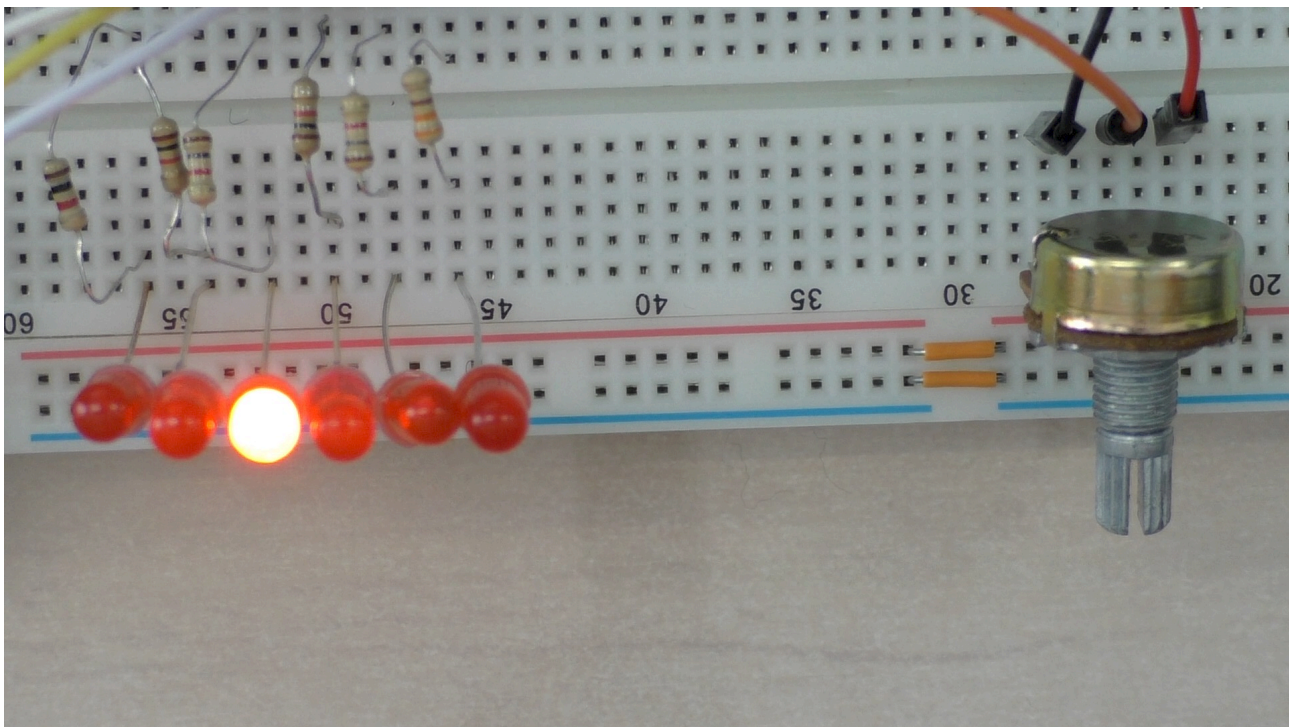
Ein Lauflicht

Aufgabe

Schliesse 6 LED's an den Arduino an und programmiere ein Lauflicht, das vor- und rückwärts läuft. Verwende dazu eine der neu gelernten Schleifenkonstruktionen. Damit das Ganze noch etwas spannender wird, soll die Geschwindigkeit mit einem Potentiometer eingestellt werden können.

Eine weitere Vorgabe solltest du ebenfalls erfüllen:

Wenn die Aufgabenstellung ändert und du plötzlich mit 7 anstatt mit 6 Leuchtdioden arbeiten sollst, muss das Programm durch Änderung einer einzelnen Ziffer angepasst werden können.



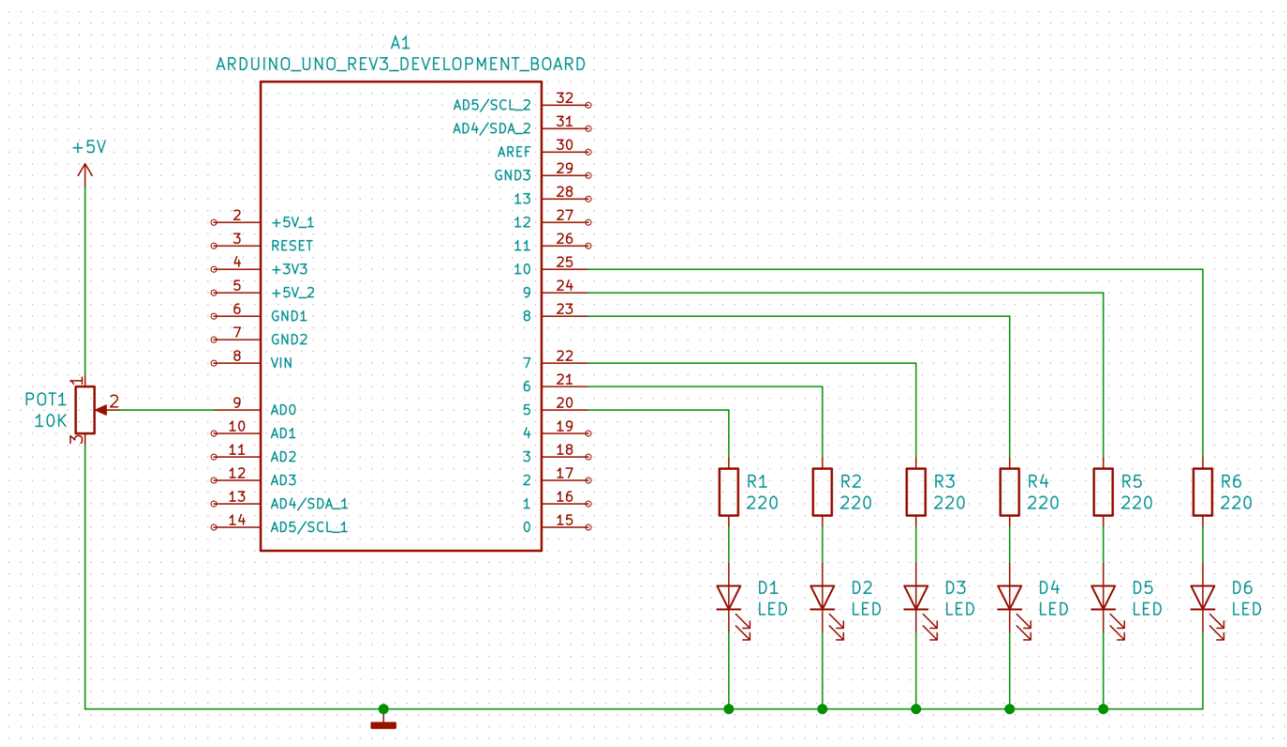
Lösung Verdrahtung

Zuerst müssen wir uns über die Hardware klar werden.

Wie das Potentiometer angeschlossen wird, ist klar. Wir haben das schon oft gemacht.

Auch die LED's sind kein Problem. Allerdings haben wir diesmal 6 LED's und müssen uns überlegen, mit welchen Pins wir sie verbinden. Da wir eine Schleife benützen möchten, ist es sinnvoll, die LED's an aufeinanderfolgende Pins anzuschliessen.

Hier siehst du eine der Möglichkeiten:



Für die Widerstände eignen sich Werte zwischen 220 Ohm und 1 KOhm. Probiere verschiedene Werte aus, bis du eine angenehme Helligkeit erreichst.

Lösung 1: mit FOR - Schleife

Zuerst definieren wir die Konstanten. Das Ganze soll ja einfach auf mehr oder weniger LED's angepasst werden können. Daher geben wir nur den Pin der ersten LED an und führen in einer weiteren Variablen die Anzahl der Leuchtdioden.

```
const int ledPin = 5;
const int potiPin = A0;
const int anzahlLeds = 6;
```

In **setup()** müssen wir die Ausgabepins in den **OUTPUT** - Mode versetzen. Das kann in einer FOR - Schleife durchgeführt werden.

```
void setup() {
  for (int i = 0; i < anzahlLeds; i++) {
    pinMode(ledPin+i,OUTPUT); // Es werden 6 Ausgänge benötigt
  }
}
```

Die Geschwindigkeit wird über einen **delay()** reguliert. Dazu wird ein Wert vom Potentiometer eingelesen und auf einen passenden Delay - Wert umgerechnet. Dazu kann eine Funktion erstellt werden. Der eingelesene Bereich von 0 bis 1023 wird auf 500 bis 5 umgerechnet. Wenn das Potentiometer ganz links steht, lesen wir einen Wert 0 ein. Dann soll die Geschwindigkeit sehr klein sein (Delay-Wert hoch), darum wurde 0 zu 500 gemappt.

```
int geschwindigkeit() {
  // Das Poti wird gelesen und der Wert umgerechnet
  int p = analogRead(potiPin);
  return map(p,0,1023,500,5);
}
```

Die eigentlichen Operationen erfolgen in loop(). Da diese Funktion bereits aus einer Schleife aufgerufen wird, müssen wir nur noch einen Vorgang machen oben und wieder nach unten programmieren.

Zuerst gehen wir nach oben. Wir beginnen mit Pin 5 (=ledPin) und gehen bis Pin 10 (=ledPin+anzahlLeds-1). Die Länge der Pause bekommen wir aus der Funktion **geschwindigkeit()**.

```
void loop() {
  // wir gehen nach oben
  for (int i=ledPin; i<ledPin+anzahlLeds; i++) { // von Pin 5 bis Pin 10
    digitalWrite(i,HIGH); // LED ein
    if (i > ledPin) {
      digitalWrite(i-1,LOW); // vorherige aus
    } else {
      digitalWrite(ledPin+1,LOW); // Beim Einschalten der untersten LED
      // muss die zweitunterste ausgeschaltet werden
    }
    delay(geschwindigkeit());
  }
}
```

Jetzt müssen wir wieder nach unten gehen. Da der oberste und unterste Pin schon behandelt wurde, beginnen wir bei Pin 9 und enden bei Pin 6.

```
// wir gehen nach unten
for (int i=ledPin+anzahlLeds-2; i>ledPin; i--) { // Pin 9 bis Pin 6
  digitalWrite(i,HIGH);
  digitalWrite(i+1,LOW);
  delay(geschwindigkeit());
}
}
```

Lösung 2: mit Berechnung

Wie so oft, gibt es auch hier mehrere Möglichkeiten. Manchmal wird die Lösung, die einem erst auf den zweiten Blick einfällt, sogar einfacher.

Die Konstanten sind wie in Lösung 1 definiert. Zusätzlich werden aber noch drei Variablen eingeführt. **ledNummer** gibt an, welche LED eingeschaltet werden soll und **alteLed** ist die Led, die ausgeschaltet werden muss. **richtung** gibt an, ob wir momentan nach oben oder nach unten gehen. Diesmal arbeiten wir nicht mit Pin-Nummern sondern nummerieren die LED's mit 0 bis 5.

```
const int ledPin = 5;
const int potiPin = A0;
const int anzahlLeds = 6;

// Die LED's sind nummeriert von 0 bis 5!
int alteLedNummer;
int ledNummer; // 0 bis 5
int richtung;  // 1 für auf und -1 für ab
```

In **setup()** setzen wir die Ausgänge wie in Lösung 1 auf **OUTPUT**. Zusätzlich definieren wir hier aber noch einen Startzustand.

```
void setup() {
  for (int i = 0; i < anzahlLeds; i++) {
    pinMode(ledPin+i,OUTPUT); // Es werden 6 Ausgänge benötigt
  }
  // Startzustand
  alteLedNummer = 1; // hier kann eine beliebige Lednummer eingetragen werden,
                    // mit Ausnahme der Startled
  ledNummer = 0;     // Start mit der LED 0
  richtung = 1;     // es geht nach oben
}
```

In **loop()** schalten wir die LED's gemäss dem gerade aktuellen Zustand. Danach berechnen wir den nächsten Zustand. Da **loop()** ja immer wieder aufgerufen wird, müssen wir das nur einmal machen.

```
void loop() {

  // Die LED's werden ein- und ausgeschaltet
  digitalWrite(ledPin+ledNummer,HIGH); // Die aktuelle LED wird eingeschaltet
  digitalWrite(ledPin+alteLedNummer,LOW); // Die alte LED wird ausgeschaltet

  // Die nächste LED wird ermittelt
  alteLedNummer = ledNummer; // Die aktuelle LED wird zur alten LED
  ledNummer += richtung;     // nächste LED bestimmen

  // Neue Richtung bestimmen
  if (ledNummer == anzahlLeds-1) richtung = -1; // oben angekommen, nach unten
  if (ledNummer == 0) richtung = 1;             // unten angekommen, nach oben

  // Die Pause
  delay(map(analogRead(potiPin),0,1023,500,5)); // vom Poti lesen
}
```