

Arduino	Modul 3
	Verzweigungen

MOTIVATION

Im letzten Modul wurde ein Blinklicht erstellt, das immer schneller wurde, bis es nicht mehr wahrgenommen werden konnte. Der Programmcode sah so aus:

```
int Pausenlaenge;

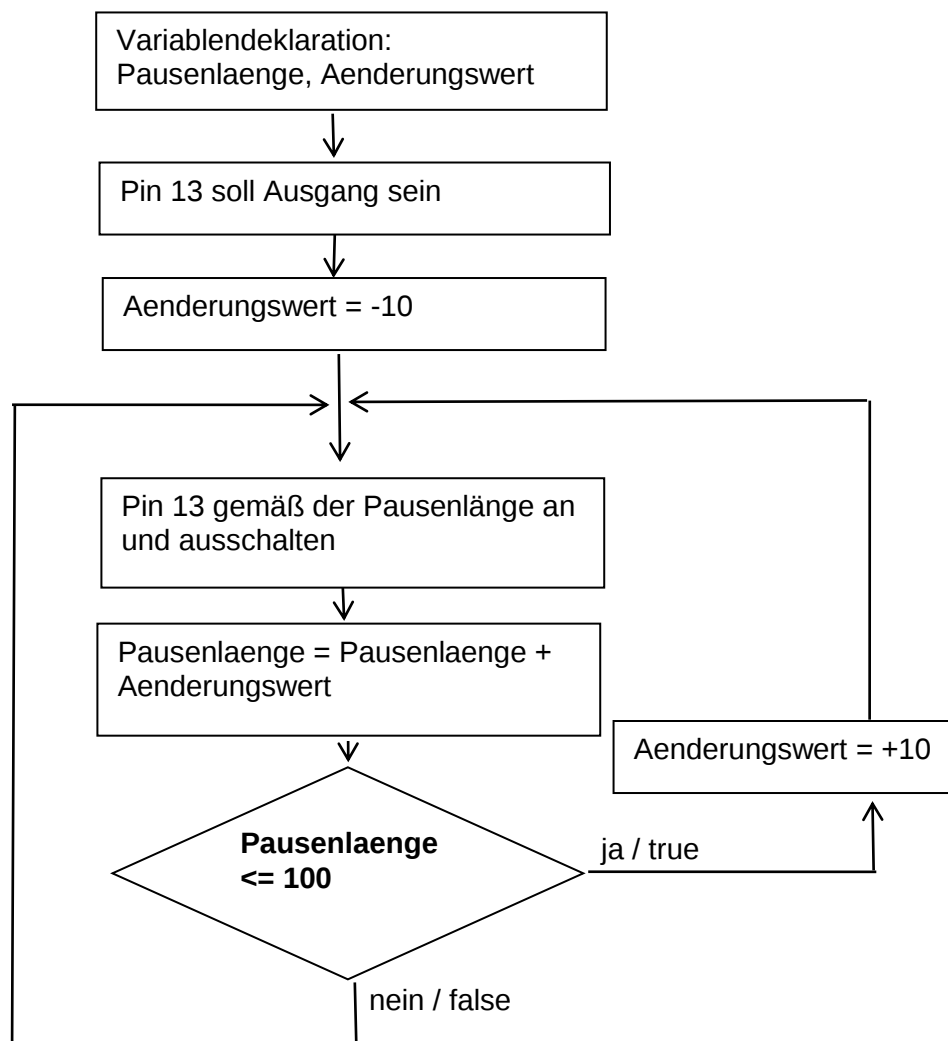
void setup() {
  pinMode(13, OUTPUT);

  Pausenlaenge = 300;
}

void loop() {
  digitalWrite(13, HIGH);
  delay(Pausenlaenge);
  digitalWrite(13, LOW);
  delay(Pausenlaenge);

  Pausenlaenge = Pausenlaenge - 10;
}
```

Das Programm soll nun bei einer Pausenlaenge von 100 anfangen, die Pausenlaenge wieder um 10 zu erhöhen. Als Flussdiagramm sieht das so aus:



Arduino	Modul 3
	Verzweigungen

Damit muss das Programm so erweitert werden:

```
int Pausenlaenge;
int Aenderungswert;

void setup() {
  pinMode(13, OUTPUT);
  Pausenlaenge = 300;
  Aenderungswert = -10;
}

void loop() {
  digitalWrite(13, HIGH);
  delay(Pausenlaenge);
  digitalWrite(13, LOW);
  delay(Pausenlaenge);

  Pausenlaenge = Pausenlaenge + Aenderungswert;

  if (Pausenlaenge < 50) {
    Aenderungswert = +10;
  }
}
```

Setze die Neuerungen um und erweitere das Programm, so dass ab einer Pausenlaenge von 300 wieder in 10er Schritte runtergezählt wird.

ÜBERSICHT

Einseitige Verzweigung

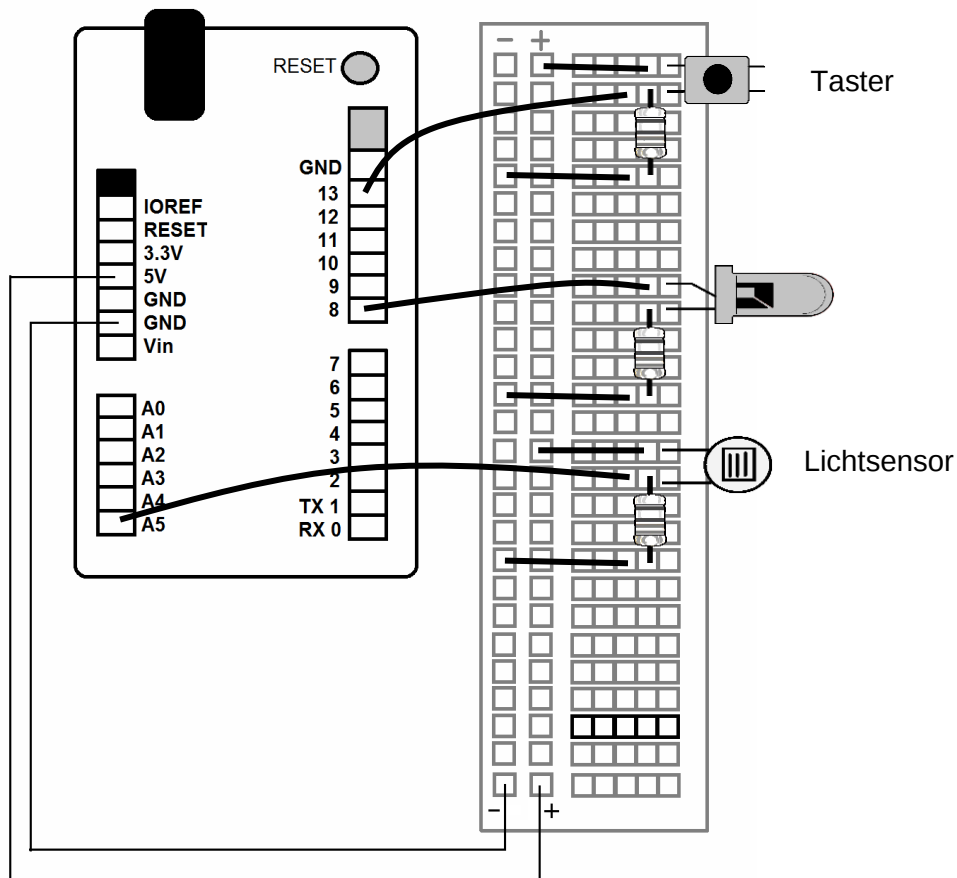
```
if (Pausenlaenge < 100) {
  Aenderungswert = +10;
}
```

Zweiseitige Verzweigung

```
if (a < 100) {
  b = 10;
} else {
  c = 30 ;
}
```

Arduino	Modul 3
	Verzweigungen

Taster und Lichtsensor anschließen



Taster	Lichtsensor
<pre> int Taster = 0; void setup() { pinMode(13, INPUT); pinMode(8, OUTPUT); } void loop () { Taster = digitalRead(13); if (Taster == HIGH){ digitalWrite(8, HIGH); } else { digitalWrite(8, LOW); } delay(100); } </pre>	<pre> int Licht; void setup() { // A ist immer ein Eingang pinMode(8, OUTPUT); } void loop() { Licht = (analogRead(5)); if (Licht < 500) { digitalWrite(8, HIGH); } else { digitalWrite(8, LOW); } delay(100); } </pre>

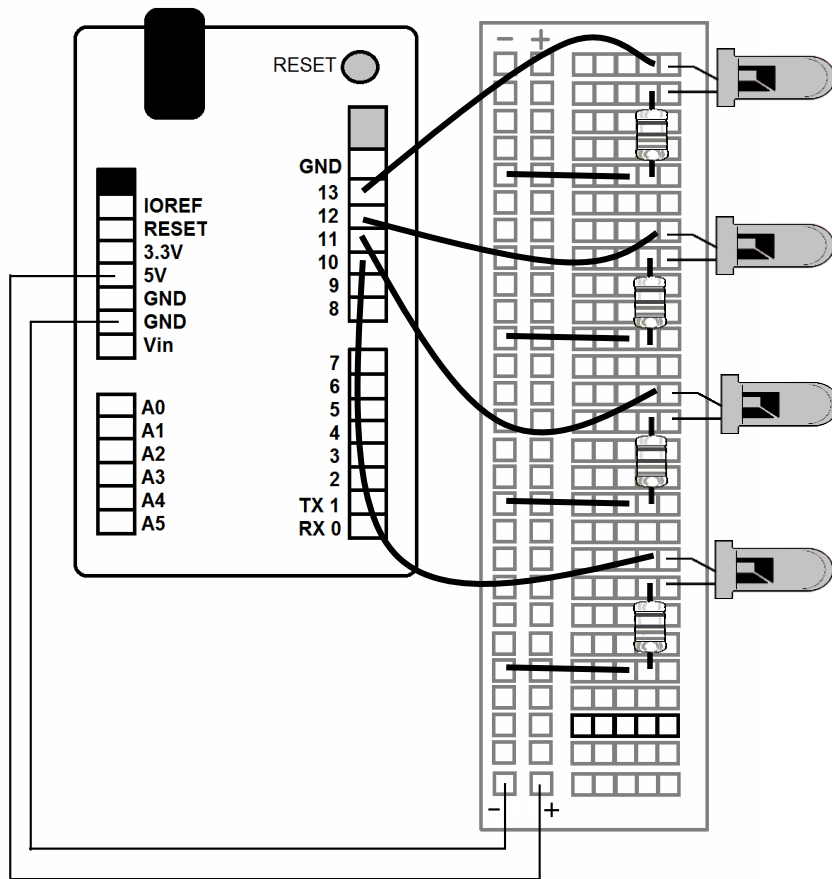
Physikalische Hintergründe im Anhang.

Arduino	Modul 3
	Verzweigungen

TRAINIEREN

Aufgabe 1

Ein Lauflicht mit 4 (oder 5) LEDs soll immer von links nach rechts laufen. Die LEDs sollen **über eine Variable** angesteuert werden. Als Grundlage dient der folgende Aufbau.



- Erstelle zunächst ein Flussdiagramm und
- setze das Lauflicht um.

Aufgabe 2

Erweiterung A: Das Lauflicht soll nun von links nach rechts und wieder nach links laufen.

Aufgabe 3

Erweiterung B: Auch soll das Lauflicht nun während es hin und her läuft schneller und wieder langsamer und wieder schneller werden.

ANWENDEN

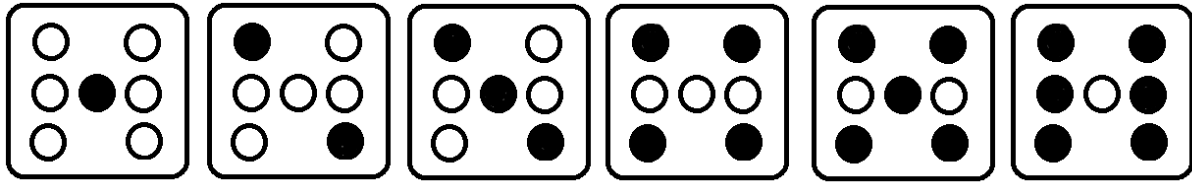
Aufgabe 4

Erweitere deinen Aufbau aus Aufgabe 1 um einen Taster. Das Lauflicht soll nur laufen, wenn der Taster gedrückt ist.

Arduino	Modul 3
	Verzweigungen

Aufgabe 5

a) Ordne 7 LEDs wie die Punkte eines Würfels an und lass alle 3 Sekunden eine Zufallszahl würfeln. Über die Zufallszahl sollen dann die Würfel-LEDs leuchten.



Tipp: „Oder“-Verknüpfung – Bsp: LED oben links an Ausgang 13

```
if ((ZufallsZahl == 2) || (ZufallsZahl == 3) || (ZufallsZahl == 4)
    || (ZufallsZahl == 5) || (ZufallsZahl == 6)) {
    digitalWrite(13, HIGH);
} else {
    digitalWrite(13, LOW);
}
```

b) Schließe einen Taster an, über den der Würfel neu gewürfelt wird.

Aufgabe 5

a) Schließe einen Helligkeitssensor an und lass dir über den seriellen Monitor die gemessenen Werte anzeigen.

b) Decke den Sensor mit der Hand ab und ermittle einen Schwellenwert, ab dem es „Dunkel“ wird.

c) Erweitere deinen Aufbau um eine LED, die bei „Dunkelheit“ (sobald der Schwellenwert unterschritten ist) leuchtet.

VERNETZEN

Aufgabe 6 - Treppenhausbeleuchtung

Fotozelle und Taster mit LED. Wenn die Fotozelle Dunkelheit registriert und gleichzeitig der Taster gedrückt wird, dann soll die LED leuchten.

Aufgabe 7 – LED dimmen

Eine LED soll langsam heller und dann wieder dunkler werden.

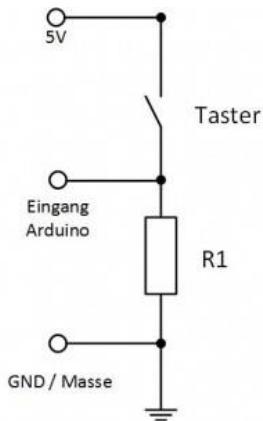
Tipp: Die digitalen Ausgänge können bei einer LED auch Helligkeitswerte zwischen 0 und 255 mit annehmen.

Bsp: `analogWrite(11, 255);` oder `analogWrite(11, 50);` oder `analogWrite(11, 0);`

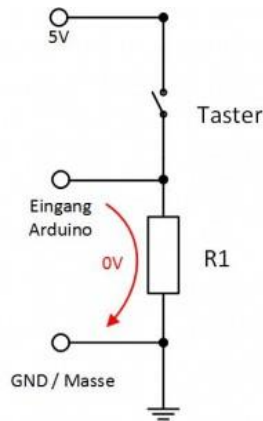
A N H A N G

Physikalischer Hintergrunde - Taster

Grundschaltung



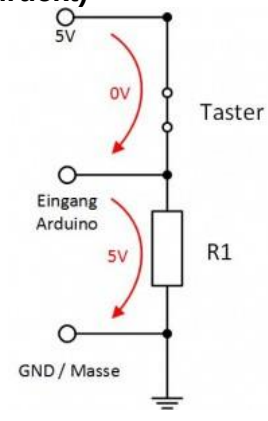
Taster offen (nicht gedrückt)



Der Taster ist nicht betätigt. Über den Widerstand R1 ist der Eingang des Arduinos mit der Masse verbunden.

Der Widerstand R1 „zieht das Potential des Eingangs auf Masse“. Das bedeutet, dass am Eingang des Arduinos eine Spannung von 0V gegenüber Masse anliegt. In diesem Fall wird der Widerstand R1 auch Pull-Down Widerstand genannt, da er das Potential „runter zieht“. Der Eingang ist somit „LOW“ geschaltet.

Taster geschlossen (gedrückt)



Wird der Taster betätigt, liegt auf einmal eine Spannung von 5V gegenüber Masse am Eingang des Arduinos an.

Hier spricht man auch davon, dass der Eingang „HIGH“ geschaltet ist.