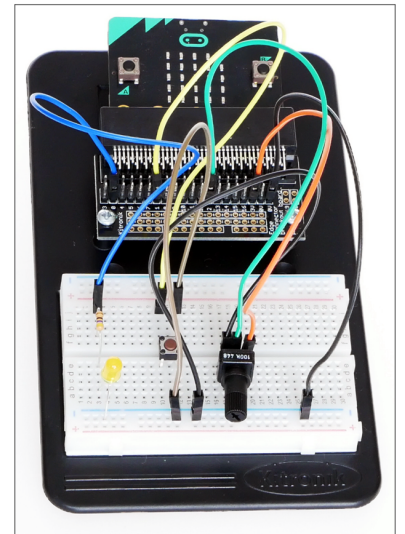


# Dimming LED

**Ziel:** Helligkeitsregelung einer Leuchtdiode

**Inhalte:** Potentiometer (Drehregler)  
LED + Vorwiderstand, Taster  
Verwenden eines ADCs  
Pulsweitenmodulation (PWM)



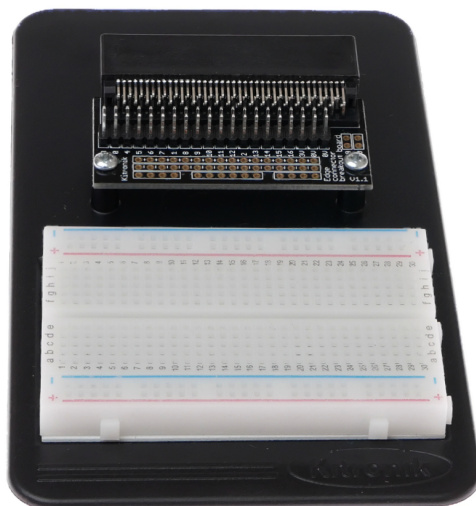
## LESSON 1

### Aufgabenstellung

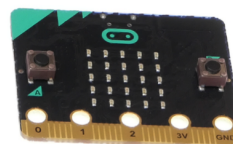
Mit einem Potentiometer (Drehregler) stellst du die Helligkeit einer Leuchtdiode ein. Unabhängig davon kannst du mit dem Taster die LED ein- bzw. ausschalten.

### Zutaten

**Steckboard**



**Microbit**



**Potentiometer  
& Taster**



**LED + Vorwiderstand**



## LESSON 2

### Pulsweitenmodulation (PWM)

Da mit digitalen Pins des Microbits lediglich 3V oder 0V ausgegeben werden können, muss man sich überlegen, wie man beliebige Spannungswerte erzeugen kann, um z.B. eine Leuchtdiode zu dimmen oder die Drehzahl eines Motors zu steuern.

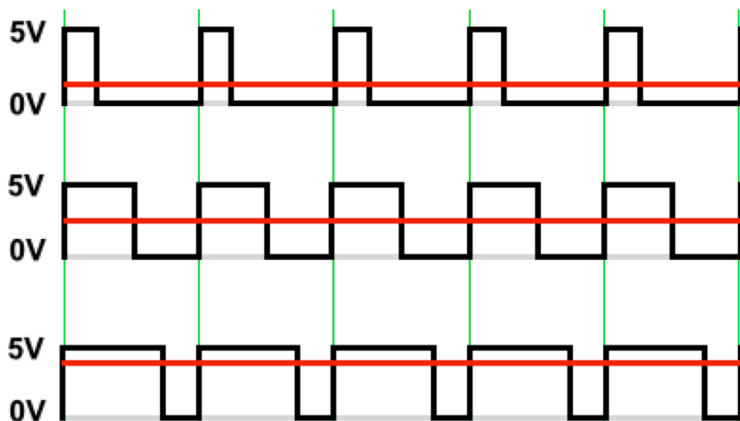
Da kommt die sogenannte **Pulsweitenmodulation (PWM)** ins Spiel.

Dabei wird nicht die Spannung selbst verändert, sondern die Zeitdauer, in der die Spannung abgegeben wird. D.h. die LED oder der Motor wird in schneller Abfolge ein- und ausgeschaltet.

Da das menschliche Auge ein schnelles Aus- und Einschalten einer LED nicht wahrnimmt bzw. die Trägheit eines Motors diesen trotz PWM kontinuierlich laufen lässt, entscheidet lediglich das Verhältnis zwischen Puls (ein) und Pause (aus) die Helligkeit der LED bzw. die Geschwindigkeit des Motors.

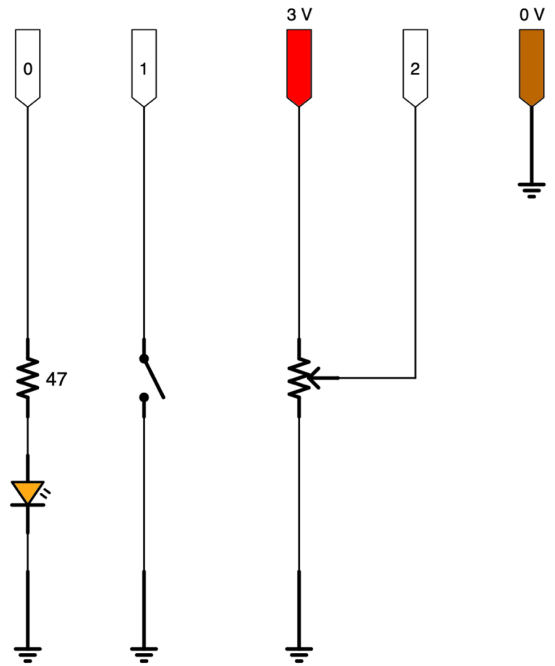
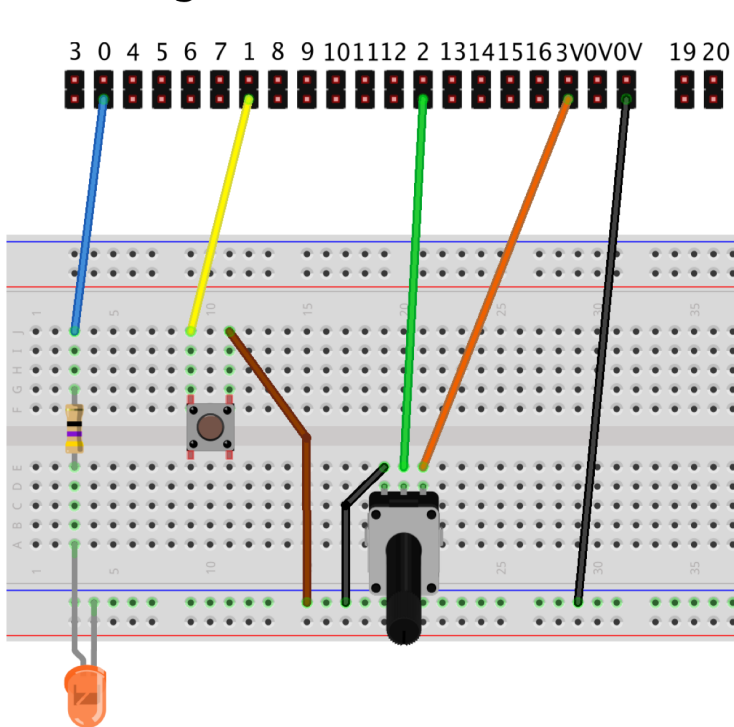
Ist das Puls-Pause Verhältnis 50:50, so dreht sich der Motor mit halber Kraft (LED => Helligkeit 50%).

Ist das Puls-Pause Verhältnis 80:20, so dreht sich der Motor mit 80% der max. Geschwindigkeit (LED => Helligkeit 80%).



# LESSON 3

## Schaltungsaufbau



- Positioniere die Leuchtdiode so wie im Bild ersichtlich. Der lange Pin (Anode) zeigt nach oben zum Vorwiderstand.
- Der kurze Pin (Kathode) wird zum gemeinsamen Gnd (- Pol) geführt.
- Der lange Pin wird über einen sogenannten Vorwiderstand (47 Ohm) zum Ausgangspin des Microbits geführt.  
[Kabel blau - Pin0]
- Positioniere den Taster auf dem Steckbrett so, dass die zwei eng aneinander liegenden Pins oberhalb und unterhalb der Kerbe liegen (siehe Bild). Ein Anschluss des Tasters wird zum Pin1 geführt, der andere Anschluss zur gemeinsamen Masse-Leiste.
- Nun platzierst du den Potentiometer. Den Anschluss ganz links führst du ebenfalls zur Masseleiste, der Anschluss ganz rechts wird über ein **orange Kabel** mit dem **3V Pin** des Microbits verbunden. Der mittlere Anschluss wird über ein **grünes Kabel** zum **Pin2** geführt.
- Zuletzt wird die Masseleiste (Gnd - 0V) mit dem Microbit verbunden.  
[Kabel schwarz - 0V]

# LESSON 4

beim Start

ändere LED\_Zustand auf 0

dauerhaft

wenn LED\_Zustand = 1 dann

schreibe analogen Pin P0 auf analoge Werte von Pin P2

ansonsten

schreibe digitalen Wert von Pin P0 auf 0

wenn Pin P1 gedrückt

wenn LED\_Zustand = 1 dann

ändere LED\_Zustand auf 0

ansonsten

ändere LED\_Zustand auf 1

## Infos zum Blockcode

The code is organized into three main sections:

- beim Start** (blue background):
  - Block: `ändere LED_Zustand auf 0`
- dauerhaft** (blue background):
  - Block: `wenn LED_Zustand = 1 dann`
  - Block: `schreibe analogen Pin P0 auf analoge Werte von Pin P2`
  - Block: `ansonsten`
  - Block: `schreibe digitalen Wert von Pin P0 auf 0`
- wenn Pin P1 gedrückt** (purple background):
  - Block: `wenn LED_Zustand = 1 dann`
  - Block: `ändere LED_Zustand auf 0`
  - Block: `ansonsten`
  - Block: `ändere LED_Zustand auf 1`

Callouts provide the following explanations:

- Der LED\_Zustand gibt an, ob die LED momentan an oder aus ist. Zu Programmstart ist diese aus.
- Wenn die LED an ist ...
- ... wird der vom Poti gelieferte Wert (ADC Pin2) an die LED (PWM Pin0) ausgegeben.
- ... ansonsten wird die LED ausgeschaltet.
- Bei Tastendruck wird geprüft, ob die aktuell an oder aus ist.
- Ist sie an => LED\_Zustand ändert sich auf 0. Ist sie aus => LED\_Zustand ändert sich auf 1.