

Aufgabe: Eine LED soll für 5 Sekunden leuchten, wenn ein Taster betätigt wurde.

Material:

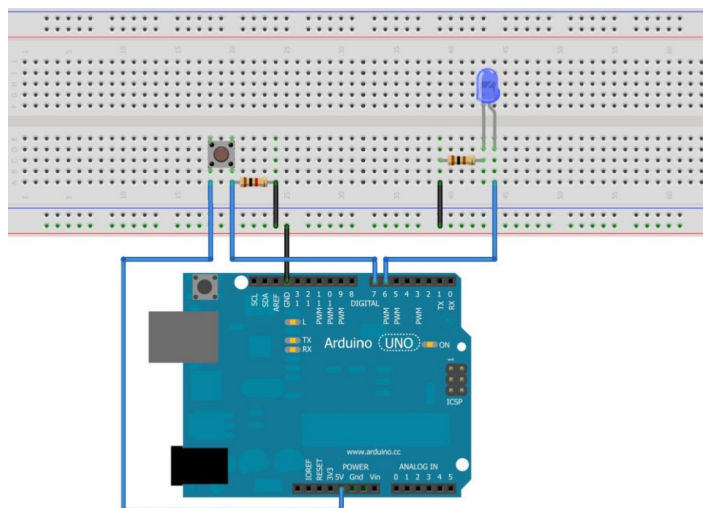
- Arduino-Board
- Ein Widerstand mit 1K Ohm (1000 Ohm)
- Eine blaue LED
- Breadboard
- Ein Taster
- Steckkabel
- Ein Widerstand mit 100 Ohm

In dieser Anleitung lernst du:

- Variablen verwenden für einzelne Pins
- Taster verwenden
- Variablen Verwenden für Messwerte (Taster)
- Pulldown-Widerstand verwenden

Der Mikrocontroller kann an seinen digitalen Pins nicht nur Spannungen ausgeben, sondern auch einlesen. Dies wollen wir in diesem Sketch ausprobieren. Bei dem Aufbau gibt es jedoch eine Besonderheit. Wenn man den Taster einfach nur mit dem Mikrocontroller verbindet, dann liegt an dem Pin des Mikrocontrollers eine Spannung an, sobald der Taster gedrückt wird. Man kann sich das so vorstellen, als würden an dem besagten Pin ganz viele Elektronen herumschwirren. Wenn der Taster dann losgelassen wird, kommen keine neuen Elektronen mehr zu dem Pin am Mikrocontroller hinzu. Die Elektronen, die sich auf dem Pin befinden, verbleiben dort recht lange und entweichen nur ganz langsam über kleine Kriechströme. Der Mikrocontroller „denkt“ dann also, dass der Taster nicht nur kurz gedrückt wird, sondern dass er ganz lange gedrückt wird. Nämlich so lange, bis sich keine Elektronen mehr auf dem Pin aufhalten. Dieses Problem lässt sich dadurch beheben, dass man den Pin über einen Widerstand (ca. 1000 Ohm bzw. 1 KOhm) mit GND verbindet. Die Elektronen können dadurch schnell vom Pin abfließen und der Mikrocontroller „erkennt“, dass der Taster nur kurz „angetastet“ wurde. Da der Widerstand die Spannung an dem Eingangspin immer auf 0V „herunter zieht“, wird er auch als „PULLDOWN-“ Widerstand bezeichnet. ACHTUNG: Wenn man dafür einen zu kleinen Widerstand verwendet, kann beim Drücken des Tasters ein Kurzschluss auf dem Mikrocontrollerboard entstehen.

Aufbau:



Made with  Fritzing.org



Sketch:

```
int LEDblau=6;           //Das Wort „LEDblau“ steht jetzt für den Wert 6.
int taster=7;            //Das Wort „taster“ steht jetzt für den Wert 7.
int tasterstatus=0;      //Das Wort „tasterstatus“ steht jetzt zunächst für
                          //den Wert 0. Später wird unter dieser Variable
                          //gespeichert, ob der Taster gedrückt ist oder
                          //nicht.

void setup()
{
  pinMode(LEDblau, OUTPUT); //Hier beginnt das Setup.
                             //Der Pin mit der LED (Pin 6) ist jetzt ein
                             //Ausgang.
  pinMode(taster, INPUT);   //Der Pin mit dem Taster (Pin 7) ist jetzt ein
                             //Eingang.
}

void loop()
{
  tasterstatus=digitalRead(taster); //Mit dieser Klammer wird der Loop-Teil geöffnet
                                     //Hier wird der Pin7 ausgelesen
                                     //(Befehl:digitalRead). Das Ergebnis wird
                                     //unter der Variable „tasterstatus“ mit dem
                                     //Wert „HIGH“ für 5Volt oder „LOW“ für 0Volt
                                     //gespeichert.

  if (tasterstatus == HIGH) //Verarbeitung: Wenn der taster gedrückt ist (Das
                             //Spannungssignal ist hoch)..
  {
    digitalWrite(LEDblau, HIGH); //Programmabschnitt des IF-Befehls öffnen.
    delay(5000);                 //...dann soll die LED leuchten..
    //...und zwar für für 5 Sekunden (5000
    //Millisekunden).
    digitalWrite(LEDblau, LOW);  //danach soll die LED aus sein.
  }                               //Programmabschnitt des IF-Befehls schließen.
  // ...ansonsten...
  else
  {
    digitalWrite(LEDblau, LOW); //Programmabschnitt des else-Befehls öffnen
    // ...soll die LED aus sein.
  }                               //Programmabschnitt des else-Befehls schließen
  //Mit dieser letzten Klammer wird der Loop-Teil
  //geschlossen.
}
```