|  |  |
| --- | --- |
| **Arbeitsblattsammlung GDC Ostwürttemberg** | |
| Modul | 2.2 |
| Thema | Pulsmesser Programmieren |
| Zielalter | Ab 12 |
| Methodik | Steckanleitung/Programmierung |



**Programmierung mit Arduino und Breadboard**

1. **Benötigte Komponenten:**

Halterung aus dem 3D-Druck

Arduino-Mikrontroller, hier Uno

PC mit Arduino IDE

USB-Verbindungskabel

Pulssensor (www.pulsesensor.com)

Rote LED

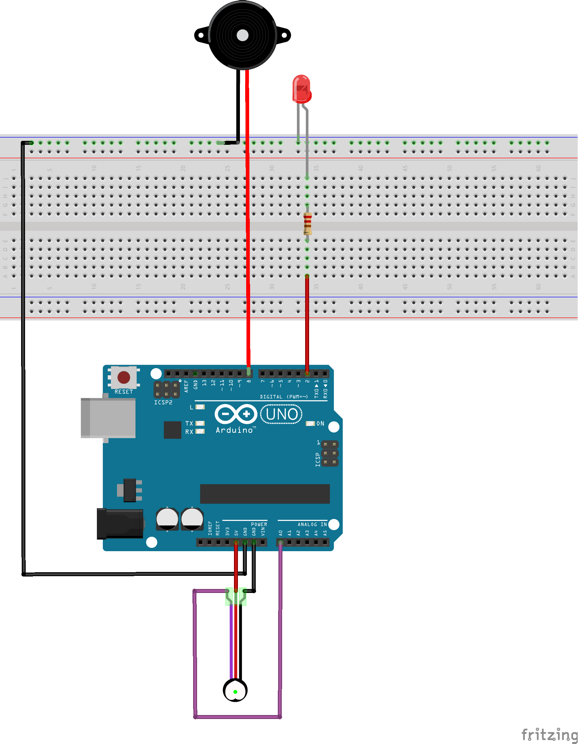
Widerstand 220 Ohm

Summer

Jumper-Wire-Kabel

1. **Steckanleitung:**

Das Datenkabel des Pulssensors wird an den Analog-Pin „A0“ des Arduinos angeschlossen. Die Stromversorgung des Pulssensors erfolgt über die Anschlüsse „5V“ und „GND“. Die LED wird über einen Widerstand an „Pin 2“ geführt; der Summer wird direkt an „Pin 8“ angeschlossen. Eine Masseleitung verbindet LED und Summer mit dem „GND“-Anschluss des Mikrocontrollers.



1. **Programmierhinweise:**

int SensorPin = A0; // Der Analog-Pin A0 wird der Variable “SensorPin” zugeordnet.

int LED = 2; // Die LED an Pin 2 wird der Variable “LED” zugeordnet.

int TON = 8; // Der Summer an Pin 8 wird der Variable „TON“ zugeordnet.

int Sensorwert; // Variable für den Sensorwert.

int Grenzwert = 515; /\* Grenzwert, ab welchem Summer und LED aktiv sein sollen. Dieser Grenzwert muss für jeden Pulssensor individuell festgelegt werden. \*/

void setup()

{

pinMode(LED, OUTPUT); // Pin 2 wird als Ausgang festgelegt.

Serial.begin(9600); /\*Die serielle Verbindung mit der Baudrate 9600 wird gestartet, sodass die Daten am seriellen Monitor dargestellt werden können \*/

}

void loop()

{

Sensorwert = analogRead(SensorPin); /\* Der Sensorwert vom Sensor wird ausgelesen und unter der Variable „Sensorwert“ abgespeichert \*/

Serial.println(Sensorwert); // Der Sensorwert wird über den seriellen Monitor ausgegeben.

if(Sensorwert > Grenzwert) // Wenn der Sensorwert über dem Grenzwert liegt…

{

digitalWrite(LED, HIGH); // … dann soll die LED leuchten …

tone(TON,440); // … und der Summer soll einen Ton mit der Frequenz 440 Hz abspielen.

}

else // Andernfalls …

{

digitalWrite(LED, LOW); // … soll die LED ausgeschaltet bleiben …

noTone(TON); // … und der Summer soll keinen Ton abspielen.

}

delay(50); // Eine kurze Pause von 50ms ermöglicht das bessere Erkennen von Messwerten.

}