

# BPM-Messung mit einem LCD-Display

## Benötigtes Material

- Arduino Uno
- USB-Kabel für den Arduino
- Pulse Sensor
- 4x MW-Kabel
- 2x MM-Kabel
- kleines Breadboard

## Benötigte Software

- Bibliothek PulseSensor Playground von Joel Murphy

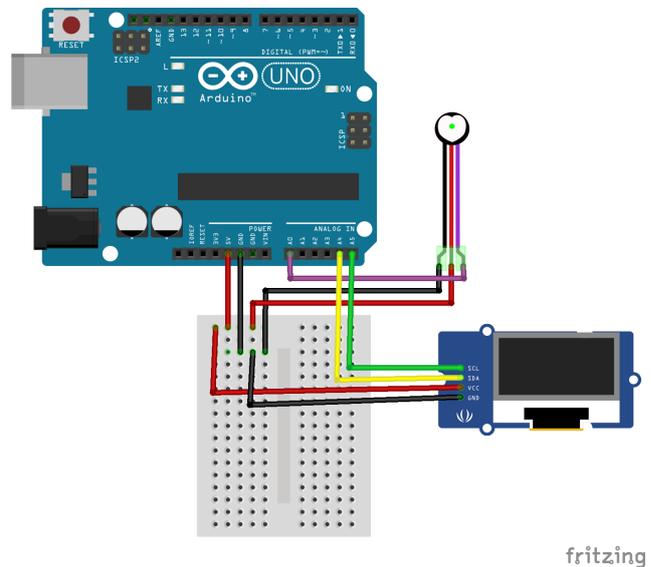
## Beschreibung

Mit dieser Anleitung kann man seinen Herzschlag in BPM, also beats per minute, messen und auf einem LCD-Display anzeigen lassen. Zusätzlich bietet die Bibliothek auch die Möglichkeit die Messungen auf dem seriellen Monitor anzuzeigen. Wähle ihn dafür einfach unter dem Reiter „Werkzeuge“ aus.

## Verkabelung

Pulse Sensor	Arduino	Sonstiges
-	GND	über Breadboard
+	5V	über Breadboard
S	A0	

I <sub>2</sub> C-Display	Arduino	Sonstiges
GND	GND	über Breadboard
VCC	5V	über Breadboard
SCL	A5	
SDS	A4	



Schaltplan

## Code

```
#define USE_ARDUINO_INTERRUPTS true // Interrupts für eine genauere BPM-Mathematik.
#include <PulseSensorPlayground.h> // Binde die PulseSensorPlayground Library ein.
#include <LiquidCrystal_I2C.h> // Binde die Bibliothek für das Display ein.

// Variablen:
const int PULSEWIRE = A0; // Signalleitung an Analog A0.
const int LED_13 = 13; // LED an Port 13 wird verwendet.
int GRENZWERT = 510; // Grenzwert, ab dem die LED an Pin13 später leuchten soll.
int SENSORWERT; // Variable für den Sensorwert im Bereich zwischen 0-1024.

//Objekte:
// Objekt für das LCD-Display namens lcd.
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```
// Ein Objekt des PulseSensorPlayground wird erstellt namens pulseSensor.
PulseSensorPlayground pulseSensor;

void setup() {

  // Pin 13 wird als Ausgang deklariert, um den Puls anzuzeigen.
  pinMode(LED_13, OUTPUT);
  // Starte serielle Schnittstelle zur Kommunikation mit dem Arduino
  Serial.begin(9600);

  // Objekt wird durch zuweisen unserer Variablen konfiguriert.
  pulseSensor.analogInput(PULSEWIRE);
  pulseSensor.blinkOnPulse(LED_13);      // LED blinkt zum Herzschlag.
  pulseSensor.setThreshold(GRENZWERT);

  // Starte das Display.
  lcd.init();
  // Hintergrundbeleuchtung fürs Display einschalten.
  lcd.backlight();

  // Test, ob das Objekt richtig erstellt wurde und ein Signal empfängt.
  if (pulseSensor.begin()) {
    // Anzeige beim Start oder reset des Arduinos.
    Serial.println("We created a pulseSensor Object !");
    lcd.clear();
    lcd.print("BPM:"); // Anzeige in beats per minute.
  }
}

void loop() {

  // Anzeigen des Pulses auf dem Arduino:
  // Sensorwert vom Sensor auslesen und unter der Variablen "sensorwert" abspeichern.
  SENSORWERT = analogRead(PULSEWIRE);
  // Sensorwert über die Serielle Schnittstelle an den Arduino senden.
  Serial.println(SENSORWERT);

  // Holt die Messwerte vom Sensor und speichert diese in myBPM.
  int myBPM = pulseSensor.getBeatsPerMinute();

  if (myBPM > GRENZWERT) {      // Prüfe, ob der myBPM über dem Grenzwert ist.
    digitalWrite(LED_13, HIGH); // Wenn ja, soll die LED zu Deinem Herzschlag leuchten.
  } else {                      // Ansonsten...
    digitalWrite(LED_13, LOW);  // ...ist die LED aus.
  }

  // Ausgabe des BPM auf dem seriellen Monitor:
  if (pulseSensor.sawStartOfBeat()) { // Prüfe, ob ein Herzschlag erkannt wurde.
    Serial.println("A HeartBeat Happened ! "); // Wenn ja, printe Nachricht und...
    Serial.print("BPM: "); // ... printe auf Monitor "BPM: ".
    Serial.println(myBPM); // Gebe Wert von myBPM aus.

    // Gebe die BPM auf dem Display aus.
    lcd.clear();
    lcd.setCursor(0, 0);
  }
}
```

```
    lcd.print("BPM:");  
    lcd.setCursor(0, 1);  
    lcd.print(myBPM);  
}  
  
delay(10000);           // 10sec Pause für bessere Messwerte.  
  
}
```

## Aufgaben

- ① Messe deinen Herzschlag einmal im ruhigen Zustand, wie viele BPM wird dir angezeigt?

Antwort: \_\_\_\_\_

- ② Mache nun 10 Hampelmänner und messe deinen Herzschlag erneut, wie viele BPM werden dir nun angezeigt?

Antwort: \_\_\_\_\_

Das Material und dessen Inhalte sind - sofern nicht anders angegeben - lizenziert unter der Creative Commons Lizenz CC BY-NC-SA 4.0 (für den vollständigen Lizenztext siehe <https://creativecommons.org/licenses/by-sa/4.0/legalcode>)