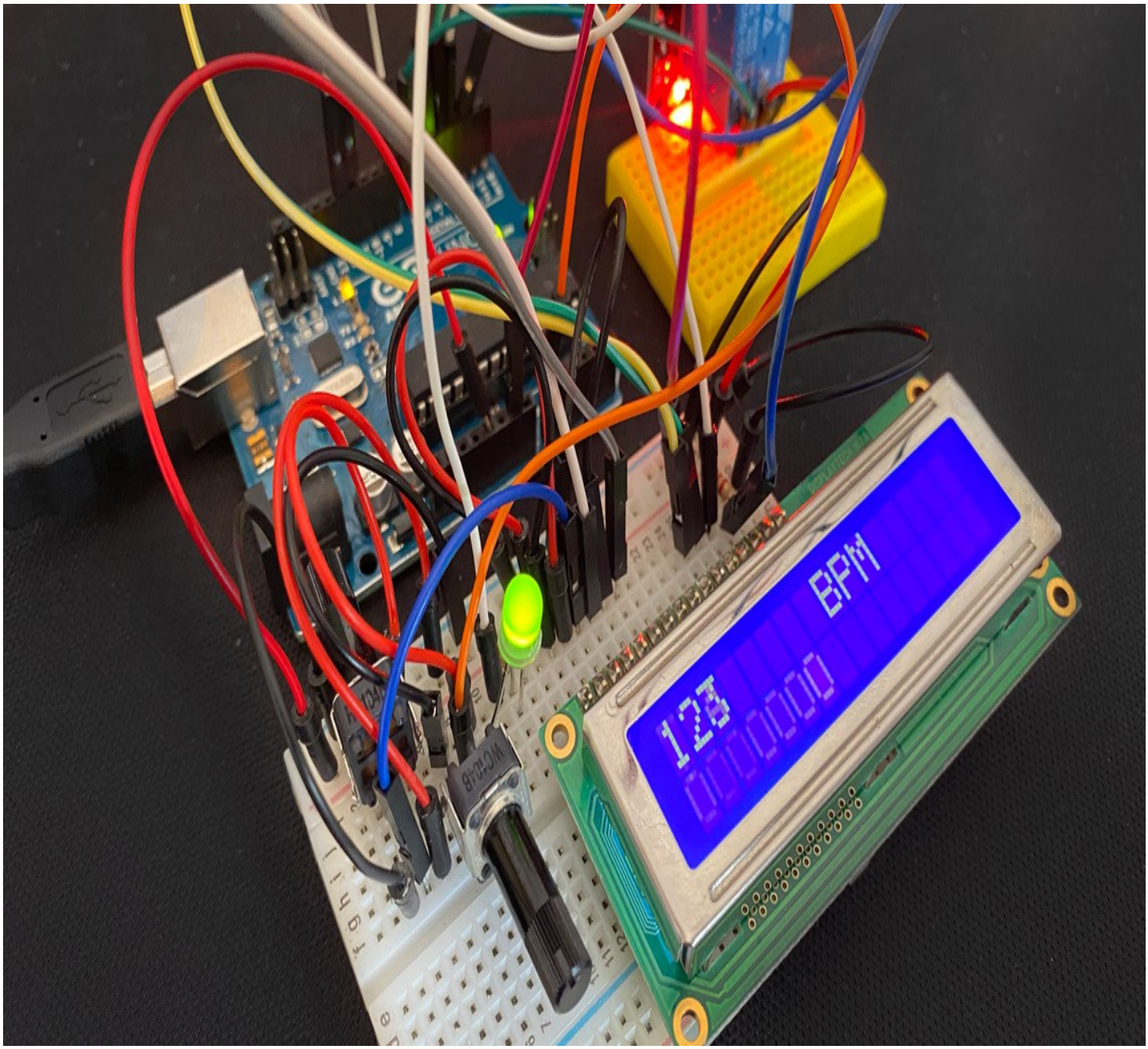
So baust du ein Metronom mit Arduino und LCD



Laut Wikipedia ist Metronom ein mechanisches oder elektronisches Gerät, das durch akustische Impulse in gleichmäßigen Zeitintervallen ein konstantes Tempo vorgibt. Es wird in der Musik verwendet, um den Takt zu halten. Das können wir doch super mit Arduino nachbauen?

Bauteile

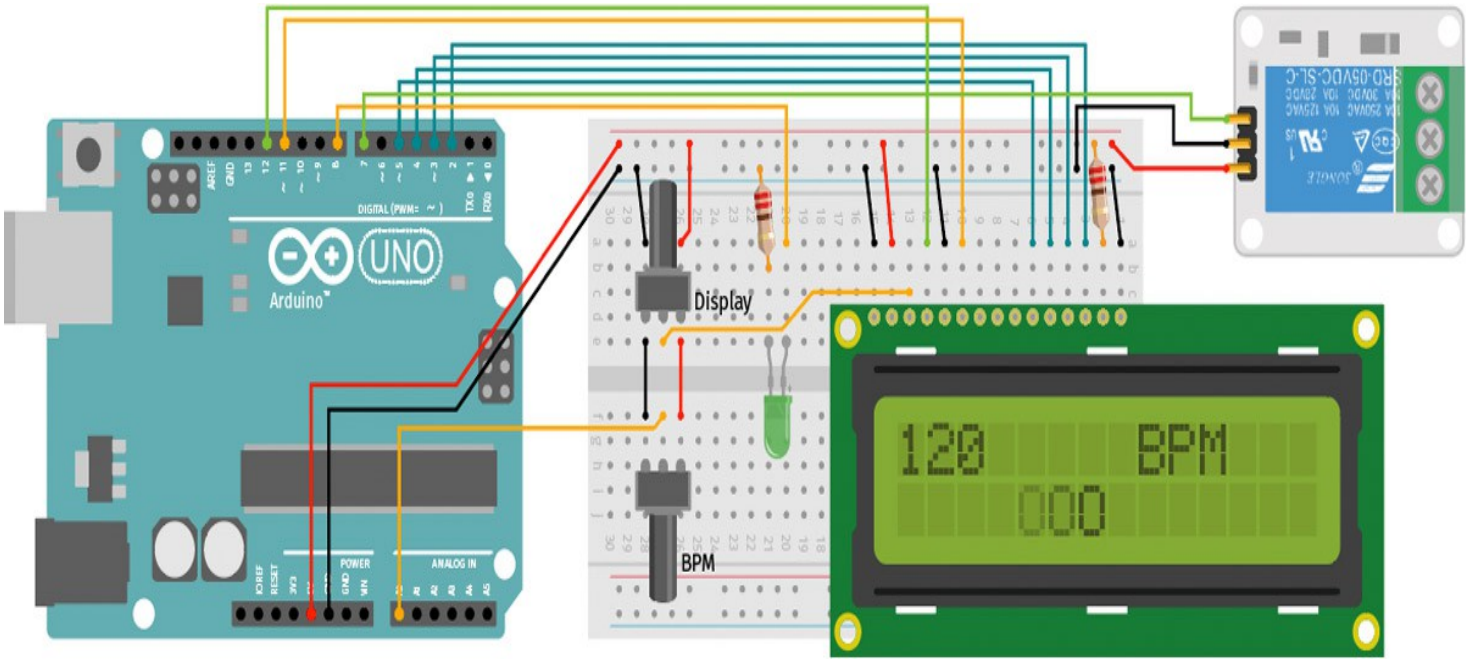
- 1x

  Arduino Set, Breadboard, Kabel, etc.*
- 2x Widerstand 220 Ohm

https://starthardware.org/so-baust-du-ein-metronom-mit-arduino-und-lcd/

- 1x LED Grün
- 2x Potentiometer 100kOhm
- 1x Relais-Platine*
- 1x Zwei-Zeilen-Display 20×2, blau/weiß*

Schaltplan



Der Schaltplan besteht aus einem LCD-Display, einem Potentiometer und einer Relais-Platine. Hier kann ich nur raten, die als fertiges Modul zu kaufen. Es ist super günstig und man spart sich Schaltungen mit verstärkendem Transistor und Diode.

Funktionsweise

Dreht man das Potentiometer nach rechts, wird die BPM-Rate, also die Beats per Minute (oder auch die Geschwindigkeit) erhöht. Das Relais klickt im Takt, die grüne LED blitzt bei jedem Takt auf. Dreht man das Potentiometer ganz nach links, schaltet man das Metronom aus.

Code für das Metronom mit Arduino

```
#include <LiquidCrystal.h>                // Einbinden der LCD Library
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);    // Anlegen des LCD-Objektes
int potPin = A0;                          // Analoger Pin des Potentiometers
int relaisPin = 7;                        // Pin, an dem ein Relais angeschlossen ist
int ledPin = 8;                           // Pin, an dem eine LED angeschlossen ist
int maxBPM = 200;                         // maximale beats per minute
int minBPM = 60;                          // minimale beats per minute
int theBPM = 0;                           // aktuelle beats per minute

int pendulumDirection = 1;                // Richtung, in die das Pendel ausschlägt
int pendulumPosition = 0;                 // Position des Pendels

void setup() {
  Serial.begin(115200);
```

https://starthardware.org/so-baust-du-ein-metronom-mit-arduino-und-lcd/

```
  lcd.begin(16, 2);                         // Start des LCD mit 16 Zeichen und 2 Zeilen
  pinMode(relaisPin, OUTPUT);               // relaisPin als OUTPUT deklariert
  pinMode(ledPin, OUTPUT);                  // ledPin als OUTPUT deklariert
}

void showPendulum(long myBPM){              // Methode für optische Ausgabe des Metronoms
  lcd.setCursor(pendulumPosition,1);        // Verschiebe LCD Cursor auf letzte Position des
Pendels
  lcd.print(" ");                           // Lösche letzte Pendel-Position
  pendulumPosition = pendulumPosition+pendulumDirection;        // Verändere Pendel Position
(+1 oder -1)
  digitalWrite(ledPin, LOW);                                    // schalte Feedback-LED aus
  if ((pendulumPosition>14)||(pendulumPosition<0)){              // falls Pendel ganz rechts
    pendulumDirection*=-1;                                      // ändere Pendelrichtung
    if (pendulumDirection>0) digitalWrite(relaisPin, HIGH);     // falls Pendel-Richtung nach
rechts, schalte das Relais ein
    else digitalWrite(relaisPin, LOW);                          // sonst schalte das Relais
aus
    digitalWrite(ledPin, HIGH);                                 // schalte die LED ein
  }
  lcd.setCursor(pendulumPosition,1);                            // Verschiebe LCD Cursor
  lcd.print("O");                                               // Schreibe Pendel-Zeichen auf
LCD
  delay(int(60/float(float((myBPM)/4)/16)));                    // Berechne Wartezeit in
Abhängigkeit der BPM
}

void loop() {
  if (analogRead(potPin)!=0){                                   // wenn Poti nicht auf 0
    theBPM = map(analogRead(potPin), 1, 1023, minBPM, maxBPM);  // berechne BPM von minBPM bis
maxBPM
    showPendulum(theBPM);                                       // zeige Pendel an
  } else {                                                      // sonst (wenn Poti auf 0)
    theBPM = 0;                                                 // setze BPM auf 0 und zeige
das Pendel nicht an
  }

  lcd.setCursor(8, 0);                                          // Verschiebe LCD Cursor
  lcd.print("BPM");                                             // LCD-Ausgabe

  lcd.setCursor(0, 0);                                          // Verschiebe LCD Cursor
  lcd.print(theBPM);                                            // LCD-Ausgabe
  lcd.print("  ");                                              // Lösche mögliche weitere
Stellen beim Wechsel von 3-stellig auf 2-stellig
}
```