

Eine Stoppuhr programmieren

Stellt euch vor, ihr hättet eine Carrera Bahn und wolltet mit Freunden ein Start-Ziel-Rennen veranstalten. Wer durchfährt mit einem Modellauto die vorgegebene Strecke am schnellsten? Die Zeit vom Start bis zum Ziel muss sehr genau bestimmt werden. Möglicherweise geht es um Millisekunden, damit ihr einen Sieger ermitteln könnt. Ihr beschließt, dieses Vorhaben in einem Arduino-Projekt zu realisieren.

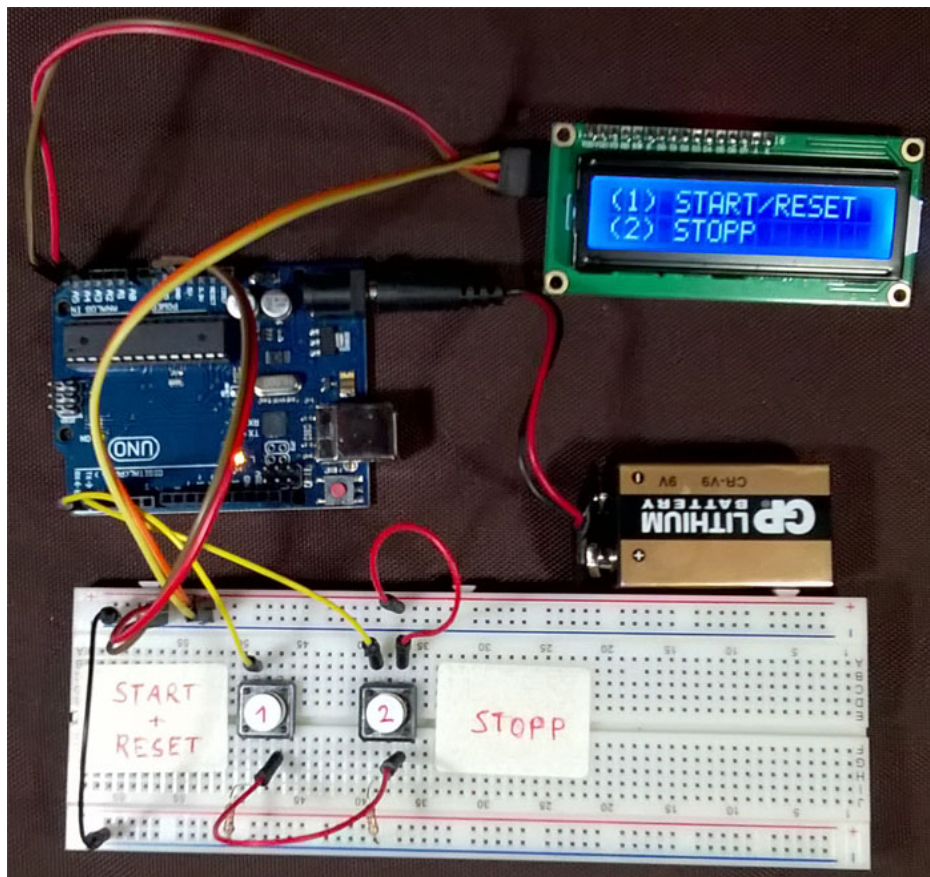
Nach kurzer Bedenkzeit ergeben sich zwei Aufgaben:

1. Bau einer Stoppuhr, die auf die Millisekunde genau messen kann.
2. Entwicklung einer Vorrichtung, die exakt die vom Fahrzeug benötigte Zeit zwischen Start und Ziel messen (die Uhr für jeden Lauf gleich starten und stoppen) kann.

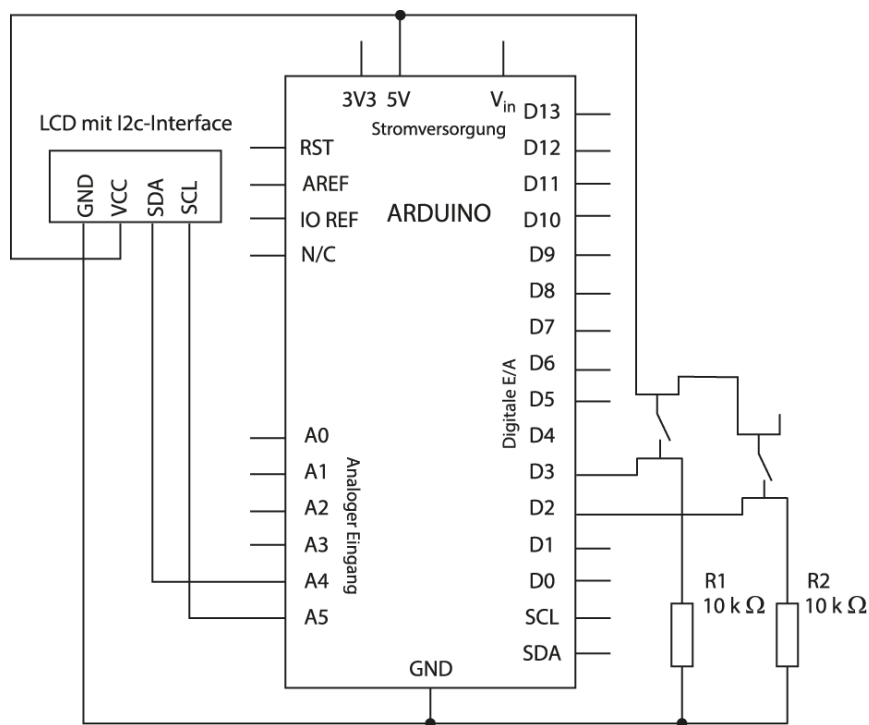
Zunächst geht es um die Lösung der ersten Aufgabe.

Aufgabe 1

Die Stoppuhr soll mit zwei Tastern ausgestattet sein. Der eine dient zum Start und Zurücksetzen, der andere zum Stoppen. Beim Einschalten der Uhr soll auf dem LCD-Display die Funktion der beiden Knöpfe ersichtlich sein (siehe Abbildung). Wird der Startknopf betätigt, soll der Hinweis „GESTARTET ...“ ausgegeben werden. Wird die Messung gestoppt, soll auf dem LCD-Display die Zeit in Sekunden und Millisekunden angezeigt werden. Ein erneutes Drücken des Startknopfes löscht die Zeitangabe und löst gleichzeitig eine weitere Messung aus.



Schaltskizze



Benötigte Materialien

- Computer mit der Arduino-Software
- Arduino UNO
- USB-Kabel
- LCD-Display mit I²C-Interface
- zwei 10 k Ω Widerstände
- eine Steckplatine (Breadboard)
- Steckbrücken male/male
- Steckbrücken male/female

Programmcode

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

unsigned long start, finished, elapsed;

void setup()
{
    lcd.begin();
    lcd.backlight();
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("(1) START/RESET");
    lcd.setCursor(0,1);
    lcd.print("(2) STOPP");

    pinMode(2, INPUT); //Starttaste
    pinMode(3, INPUT); //Stopptaste
}

void displayResult(){
```

```
int s, ms;
unsigned long over;

elapsed = finished - start;
over = elapsed % 3600000;
over = over % 60000;
s = int(over / 1000);
ms = over % 1000;

lcd.print(s);
lcd.print("s ");
lcd.print(ms);
lcd.print("ms");
}

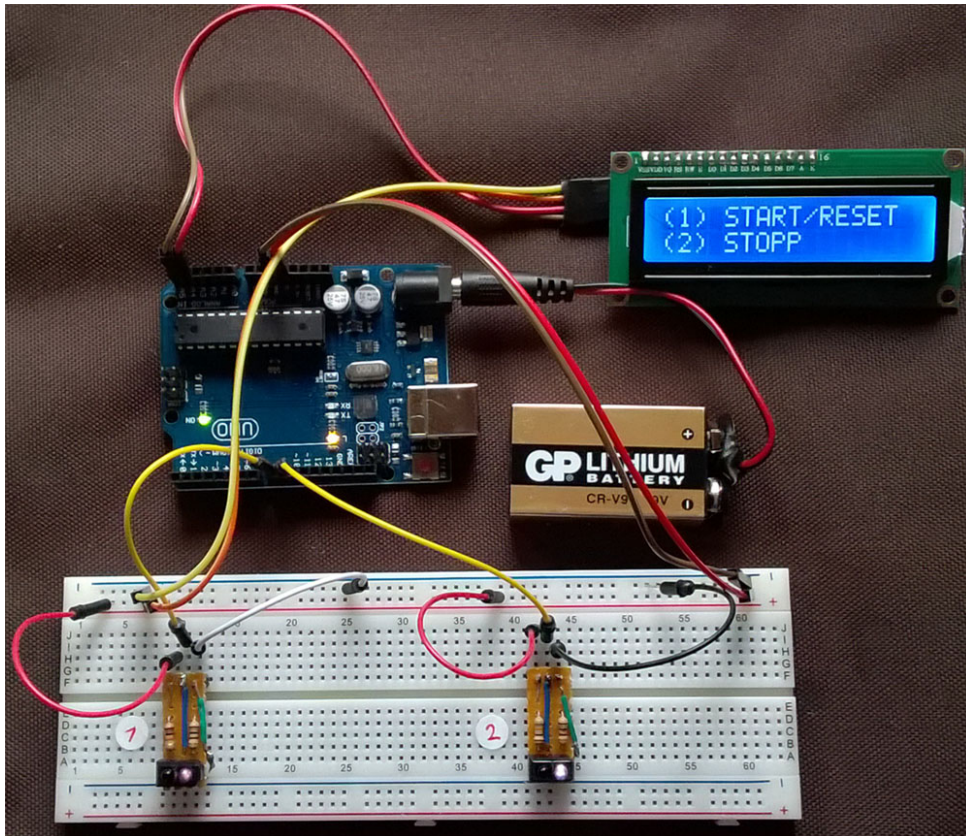
void loop(){

  if (digitalRead(2) == HIGH)
  {
    start = millis();
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("GESTARTET ...");
    lcd.setCursor(0,1);
  }

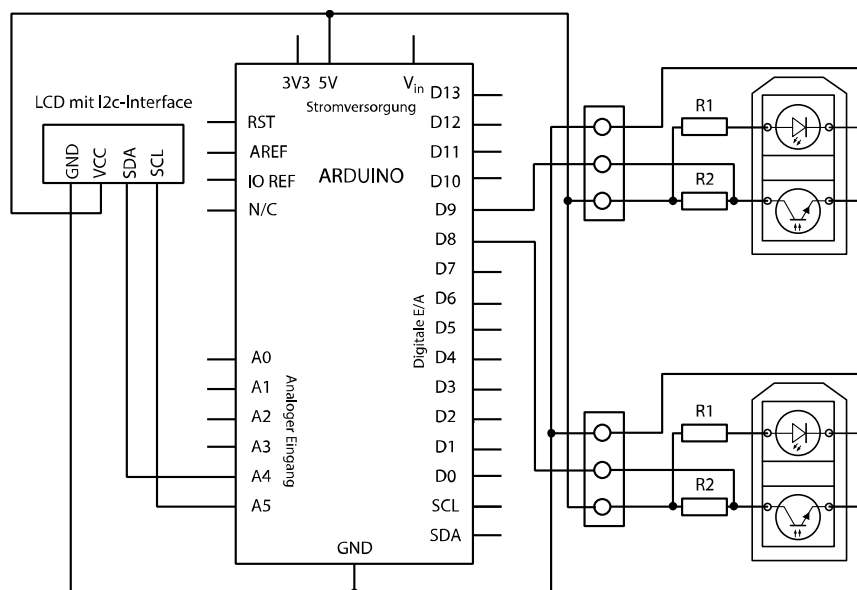
  if (digitalRead(3) == HIGH)
  {
    finished = millis();
    delay(200);
    displayResult();
  }
}
```

Aufgabe 2

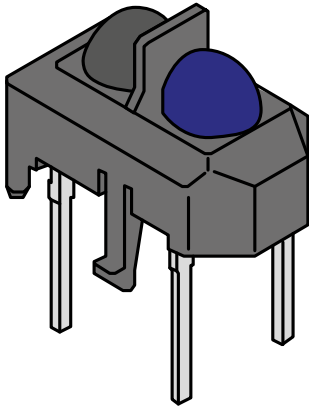
Die beiden Taster aus Aufgabe 1 sollen durch Sensoren ersetzt werden, damit Fehler vermieden werden, die beim Stoppen mit der Hand zwangsläufig auftauchen. Die Anzeigen auf dem LCD-Display bleiben unverändert (siehe Abbildung). Fährt ein Fahrzeug an Sensor 1 vorbei, wird „GESTARTET ...“ auf dem LCD-Display ausgegeben. Beim Passieren von Sensor 2 wird die Zeit in Sekunden und Millisekunden angezeigt. Jede weitere Vorbeifahrt an Sensor 1 löscht die Zeitangabe und löst gleichzeitig eine weitere Messung aus.



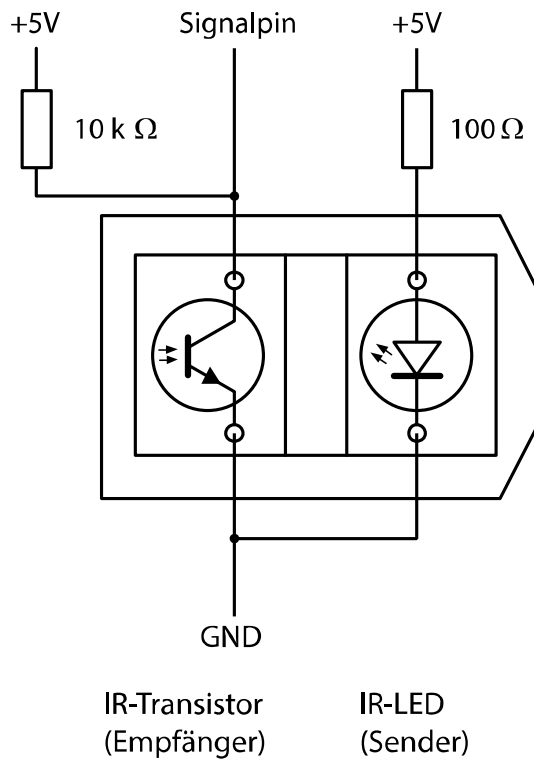
Schaltskizze



Zusätzlich benötigte Materialien



TCRT5000: 3D-Ansicht



TCRT5000: Anschlussplan

Programmcode

```
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

int IRsensor1 = 8;
int IRsensor2 = 9;

unsigned long start, finished, elapsed;

void setup() {

  pinMode(IRsensor1, INPUT);
  pinMode(IRsensor2, INPUT);

  digitalWrite(IRsensor1, HIGH);
  digitalWrite(IRsensor2, HIGH);

  lcd.begin();
  lcd.backlight();
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("(1) START/RESET");
  lcd.setCursor(0, 1);
  lcd.print("(2) STOPP");

}

void displayResult() {
```

```

int s, ms;
unsigned long over;

elapsed = finished - start;
over = elapsed % 3600000;
over = over % 60000;
s = int(over / 1000);
ms = over % 1000;

lcd.print(s);
lcd.print("s ");
lcd.print(ms);
lcd.print("ms");
}

int sensorState = 0;
int brightness = 0;

void loop() {

    sensorState = digitalRead(IRsensor1);

    if (sensorState == LOW) {

        start = millis();
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("GESTARTET ...");
        lcd.setCursor(0, 1);
        delay(1000);

    }

    sensorState = digitalRead(IRsensor2);

    if (sensorState == LOW) {

        finished = millis();
        delay(200);
        displayResult();
        delay(1000);

    }

}

```