

# Projektdokumentation: Zeitmessung

---

## Einführung

---

Dieses Projekt wurde von der gymnasialen Oberstufe des BKT Lüdenscheid (<http://www.bkt-luedenscheid.de>) in Kooperation mit dem Unternehmen di-soric GmbH & Co. KG (<https://www.di-soric.com>) initiiert und wurde von Christian Koch (<https://github.com/KochChristian>), Sebastian Schaffring und Daniel Schäfer (<https://github.com/schaefer-services>) umgesetzt.

Kontakt: [daniel.schaefer.gost61@schueler.bkt-luedenscheid.de](mailto:daniel.schaefer.gost61@schueler.bkt-luedenscheid.de)  
(<mailto:daniel.schaefer.gost61@schueler.bkt-luedenscheid.de>)

## Projektziel

---

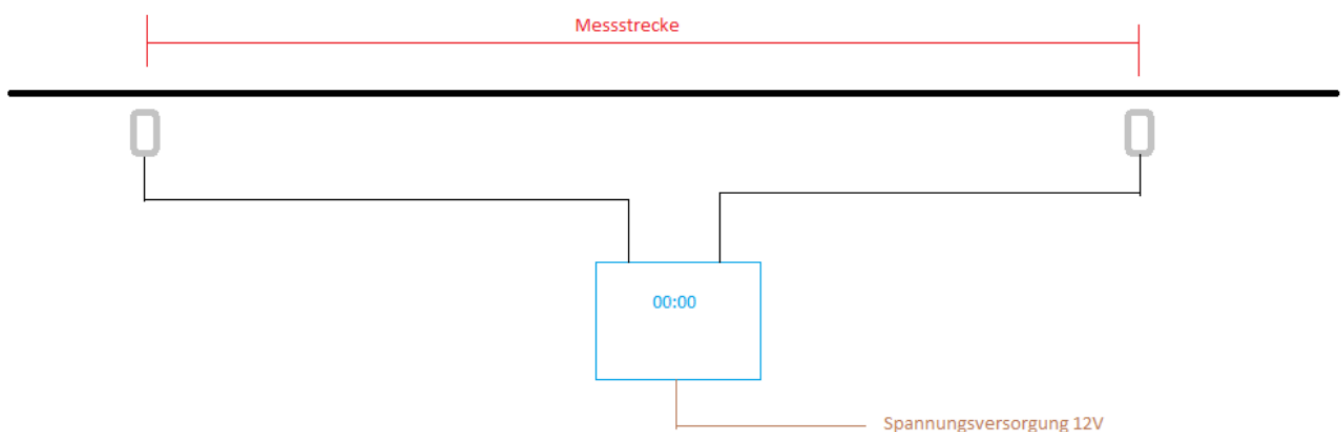
### Grundsätzliche Idee

Am Anfang und am Ende der Messstrecke befindet sich jeweils eine Reflexionslichtschranke (grau).

Am Startpunkt Sensor 1 und am Endpunkt Sensor 2.

Die Sensoren sind jeweils über eine Leitung mit der Steuerungseinheit (blau) verbunden.  
In der Steuerungseinheit befindet sich ein LC-Display zur Anzeige.

Die Spannungsversorgung beträgt 12V DC und erfolgt über ein Netzteil.



### Mechanischer Aufbau

- Die Reflexionslichtschranken und die zugehörigen Reflektoren werden auf einem kipp sicheren Stativ befestigt.
- Die Anschlussleitungen zwischen den Reflexionslichtschranken (Sensoren) und der Steuerungseinheit sind steckbar.

- In der Steuerungseinheit befinden sich folgende Bedienelemente:
  - eine Anschlussbuchse für die Stromversorgung
  - ein LC-Display
  - ein Taster
  - je eine Anschlussbuchse für Sensor 1 und Sensor 2
- Die Leitung für die Spannungsversorgung ist mit einem Stecker mit der Steuerungseinheit verbunden. Am anderen Ende befindet sich ein Stecker für die Steckdose.

## Funktionsablauf

Unmittelbar nach dem Einschalten der Zeitmessenanlage wird überprüft, ob die beiden Sensoren angeschlossen sind.

Erfolgt von einem oder beiden Sensoren keine Rückmeldung, so erscheint im Display eine Fehlermeldung.

Nach Betätigung des Tasters werden die Sensoren erneut auf korrekten Anschluss überprüft.

Sind beide Sensoren korrekt angeschlossen, erscheint im Display der Text „Ready for measurement“. Sowohl der Taster wie auch der Sensor 2 sind inaktiv. Sobald der Lichtstrahl des Sensors 1 unterbrochen wird, startet die Zeitmessung.

Im Display erscheint der Text „Measurement started“.

Sobald der Lichtstrahl des Sensors 1 wieder frei ist, wird nach einer Wartezeit von 2 Sekunden der Sensor 2 aktiviert.

Wird der Lichtstrahl des Sensors 1 innerhalb der Wartezeit erneut unterbrochen, beginnt die Wartezeit von vorn. Sobald der Sensor 2 aktiviert wird, wird der Sensor 1 deaktiviert.

Ist der Sensor 2 aktiv und sein Lichtstrahl wird unterbrochen, wird die Zeitmessung gestoppt. Die gemessene Zeit wird im Display mit einer Genauigkeit von 1/1000s angezeigt. Nach Betätigung des Tasters, beginnt die Prozedur wieder von vorne.

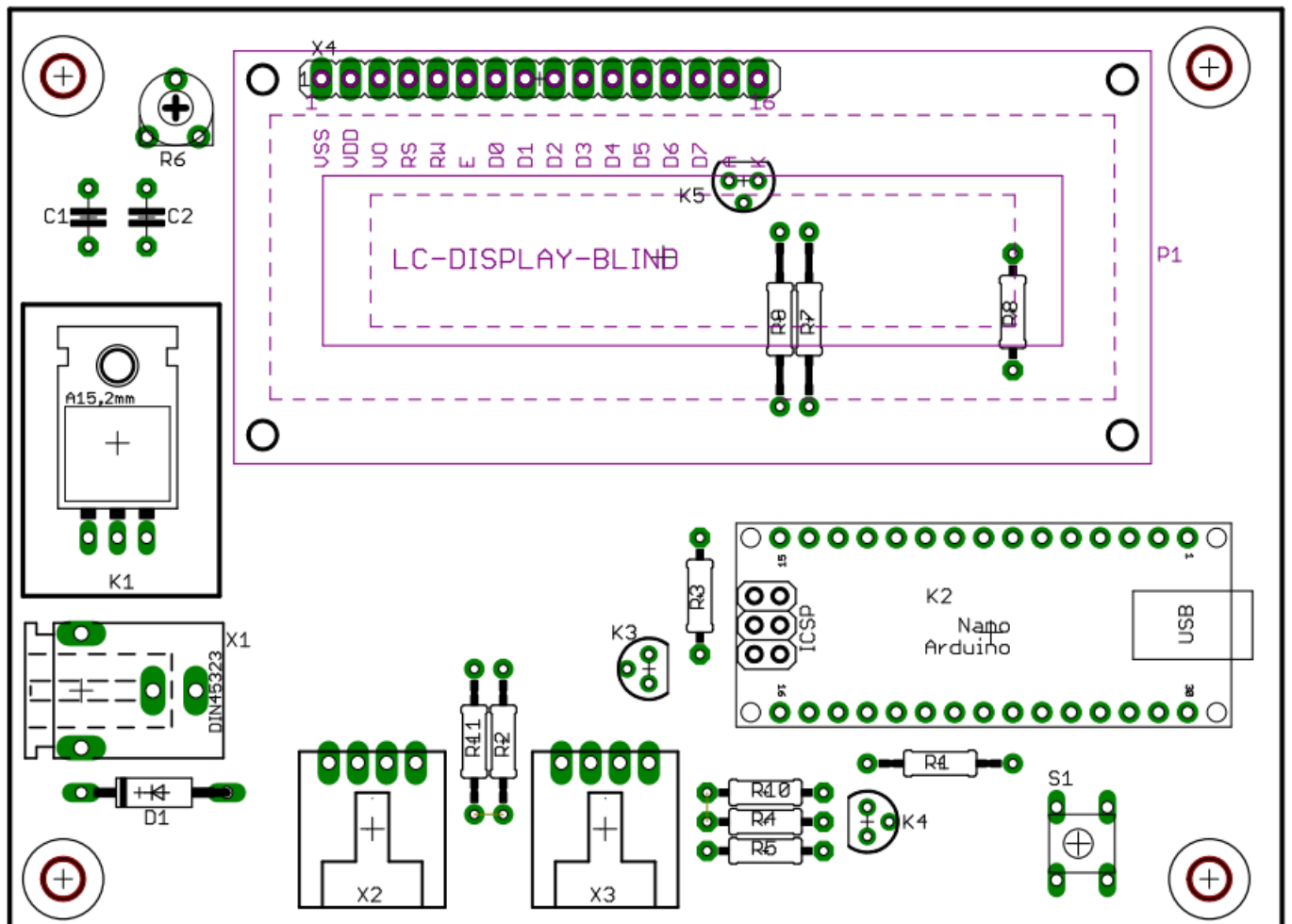
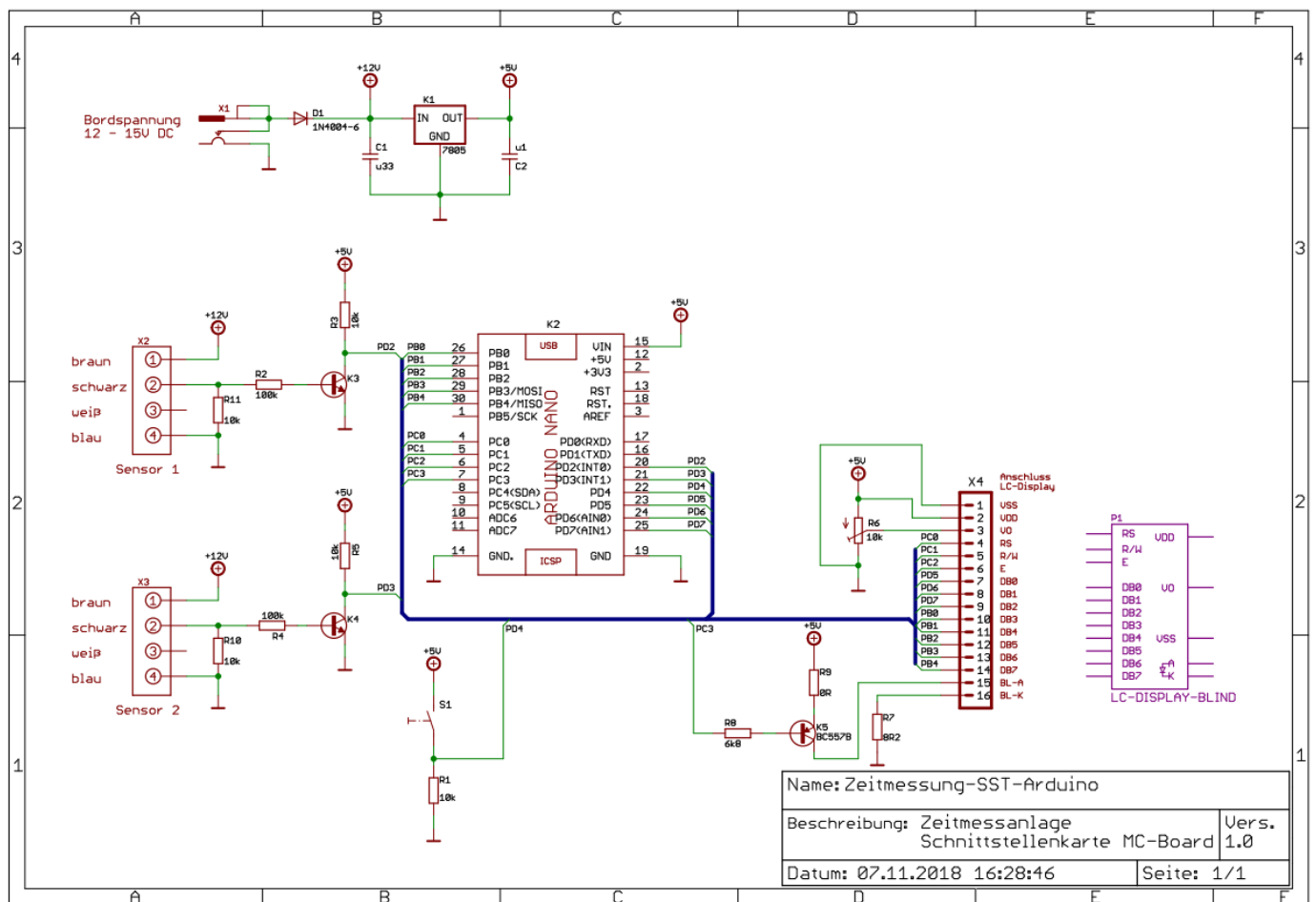
## Projektumsetzung

---

## Hardwareseitige Umsetzung

---

## Schalt- und Bestückungsplan



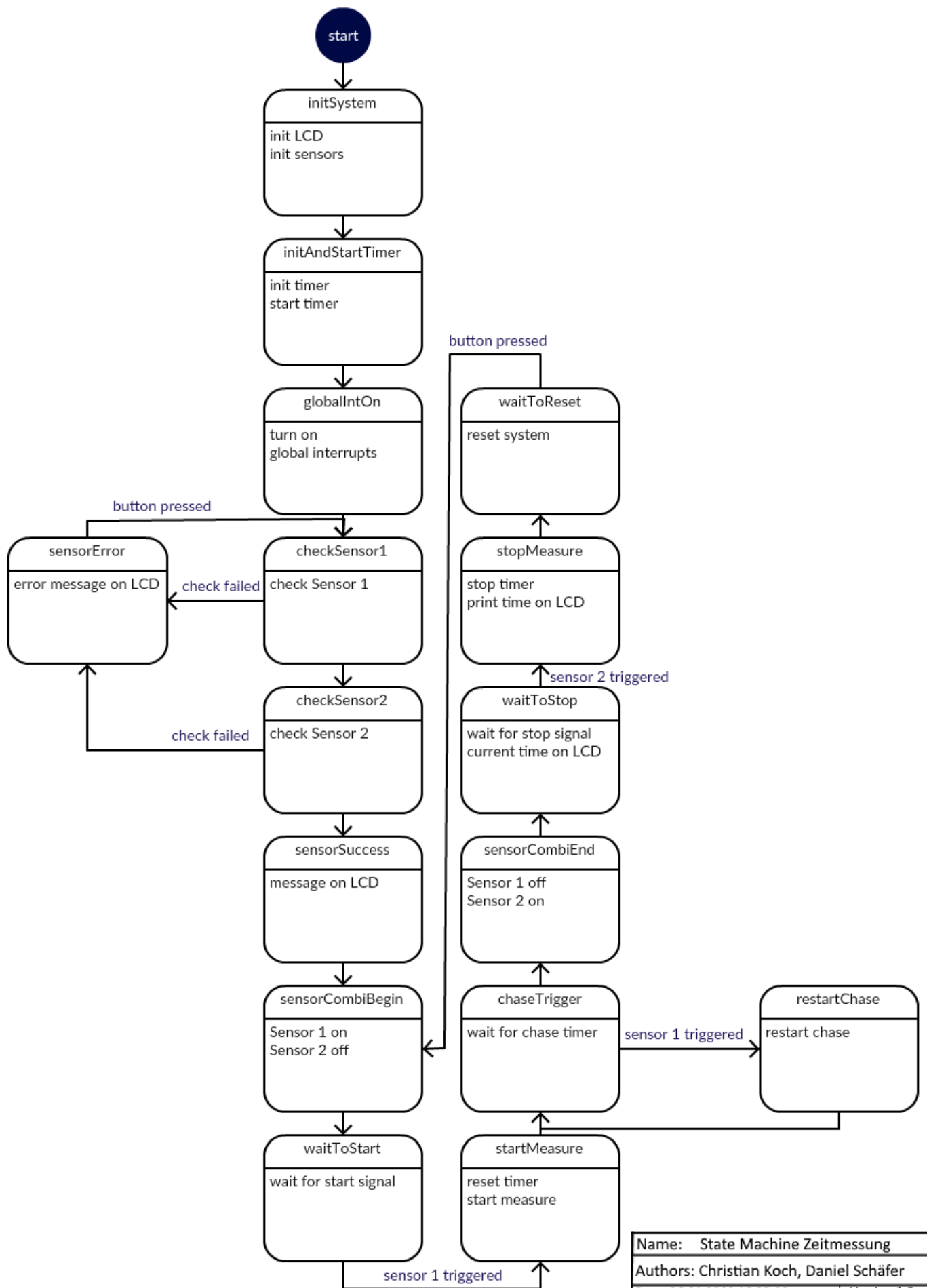
## Eingesetzte Hardware:

- Arduino Nano (<https://store.arduino.cc/arduino-nano>) (Atmel ATmega 328P (<https://www.microchip.com/wwwproducts/en/ATmega328P>)) als Mikrocontroller
- Reflexionslichtschranke di-soric OR 31 K 5000 P3-T4 (<https://www.di-soric.com/de/OR-31-K-5000-P3-T4-37530.html>)
- LCD Display 1602
- 12V Netzteil DVE Switching Adapter
- 2x E257058 Style 20549 4x0,25mm<sup>2</sup> Anschlusskabel
- 3D gedrucktes Gehäuse
- 4 Holzstative á 1,00m
- PCB mit elektronischen Bauteilen

## Softwareseitige Umsetzung

---

### State Machine



|          |                                |
|----------|--------------------------------|
| Name:    | State Machine Zeitmessung      |
| Authors: | Christian Koch, Daniel Schäfer |
| Date:    | 26.01.2019 16:25:00            |
| Version: | 1.0                            |

## Eingesetzte externe Software:

- IDE: Atmel Studio 7 (<https://www.microchip.com/mplab/avr-support/atmel-studio-7>)
- GIT-Repository (<https://github.com/schaeferservices>)
- LCD-Library libLCD162CstaticLib.a (Autor: A. Broxtermann)

Die Software (<https://github.com/schaeferservices/Zeitmessung>) wurde in C geschrieben und modular aufgebaut.

Es existieren diese 4 Module

- main.C ([https://github.com/schaeferservices/Zeitmessung/blob/master/2019-01-14\\_Zeitmessung/main.c](https://github.com/schaeferservices/Zeitmessung/blob/master/2019-01-14_Zeitmessung/main.c)) (Hauptmodul)
- hwconf.h ([https://github.com/schaeferservices/Zeitmessung/blob/master/2019-01-14\\_Zeitmessung/hwconf.h](https://github.com/schaeferservices/Zeitmessung/blob/master/2019-01-14_Zeitmessung/hwconf.h)) (Header zur Hardwarekonfiguration)
- function.C ([https://github.com/schaeferservices/Zeitmessung/blob/master/2019-01-14\\_Zeitmessung/function.c](https://github.com/schaeferservices/Zeitmessung/blob/master/2019-01-14_Zeitmessung/function.c)) (Modul für Funktionen des Hauptprogramms)
- function.h ([https://github.com/schaeferservices/Zeitmessung/blob/master/2019-01-14\\_Zeitmessung/function.h](https://github.com/schaeferservices/Zeitmessung/blob/master/2019-01-14_Zeitmessung/function.h)) (Header zur Einbindung des function.c Moduls)

Die Software (<https://github.com/schaeferservices/Zeitmessung>) wurde mit GIT versionsverwaltet. Das Repository ist öffentlich auf GitHub zu finden.

**Zum Git-Repository** (<https://github.com/schaeferservices/Zeitmessung>)

## Mögliche Benutzereinstellungen

Der Benutzer kann im Header `function.h` ([https://github.com/schaeferservices/Zeitmessung/blob/master/2019-01-14\\_Zeitmessung/function.h](https://github.com/schaeferservices/Zeitmessung/blob/master/2019-01-14_Zeitmessung/function.h)) folgende Einstellungen vornehmen:

Einstellung der Zeit, in der die Sensoren auf Funktion geprüft werden:

```
1  /*****  
2  /* setting for sensor check time in seconds                               */  
3  /*****  
4  #define SENSOR_CHECK_TIME 10
```

Einstellung für die Wartezeit, nach der Sensor 1 deaktiviert und Sensor 2 aktiviert wird:

```
1  /*****  
2  /* setting for chase time in seconds                                     */  
3  /*****  
4  #define CHASE_TIME 2
```

Einstellung für die Zeit des Ausgabeintervalls:

```
1  /*****  
2  /* setting for LCD output time in milliseconds */  
3  /*****  
4  #define OUTPUT_TIME 10
```

Einstellung für die Auslösezeit des Tasters, nachdem der Reset auslöst:

```
1  /*****  
2  /* setting for reset time in seconds */  
3  /*****  
4  #define RESET_TIME 3
```

## Aufbau & Bedienung des Systems

---

### Aufbau

1. Sensoren und Reflektoren jeweils gegenüber aufstellen.
2. Anschließen der beiden Sensoren an die Steuereinheit mithilfe der beiden Kabel.
3. Anschließen der 12V Stromversorgung mithilfe des beiliegenden Netzteils.

### Bedienung

1. Sensorüberprüfung (Hinweise auf dem Display befolgen.)
2. Starten der Messung durch Auslösen des ersten Sensors.
3. Stoppen der Messung durch Auslösen des zweiten Sensors.

#### **Hinweis:**

Beim Start einer Messung leuchtet die Status-LED dauerhaft.

Nach Ablauf der Wartezeit blinkt diese bis die Messung gestoppt wird.

4. Nach dem Stoppen der Messung wird die Zeit auf dem Display ausgegeben. Um eine neue Messung zu starten, muss der Taster einmal kurz gedrückt werden.

#### **Reset des Systems:**

Um das gesamte System zurückzusetzen, muss der Taster 3 Sekunden

(Standardeinstellung siehe Mögliche Benutzereinstellungen) lang gedrückt werden.

Anschließend müssen die Anweisungen auf dem Display zur Sensorüberprüfung befolgt werden. (siehe Punkt 1.)