

Eentwicklung eines CO₂-Messers

Mikrocomputertechnik - Bericht

Studiengang Elektrotechnik

Studienrichtung Fahrzeugelektronik

Duale Hochschule Baden-Württemberg Ravensburg, Campus Friedrichshafen

von

Alexander Herrmann Johannes Ruffer Serkant Soylu

Abgabedatum:	19.04.2020
Bearbeitungszeitraum:	01.10.2019 - 19.04.2020
Matrikelnummer:	9859538 x 1011921 x 9964027
Kurs:	TFE18-2
Gutachter der Dualen Hochschule:	Hans Jürgen Herpel

Eidesstattliche Erklärung

Gemäß Ziffer 1.1.13 der Anlage 1 zu §§ 3, 4 und 5 der Studien- und Prüfungsordnung für die Bachelorstudiengänge im Studienbereich Technik der Dualen Hochschule Baden-Württemberg vom 29.09.2015.

Wir versichern hiermit, dass wir unsere Projektarbeit mit dem Thema:

Entwicklung eines CO2-Messers

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt haben. Wir versichern zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Friedrichshafen, den 5. März 2020

Alexander Herrmann

Johannes Ruffer

Serkant Soylu

AUTOREN

Kurzfassung

Der Bericht wurde von drei Studierenden an der Duale Hochschule Baden-Württemberg (DHBW)-Ravensburg Campus Friedrichshafen im Rahmen der Mikrocomputertechnik Vorlesung eigenständig von der Projektplanung, über die Durchführung, bis hin zum Projektabschluss mit Dokumentation durchgeführt.

Ziel der Arbeit ist es, durch die Praxiserfahrung mit Mikrocomputern, Fähigkeiten und Wissen in diesem Bereich zu erwerben. Durch das Vergleichen von anfänglichen Kosten- und Komplexitätsschätzungen mit den späteren Ergebnissen in der Umsetzung können die Studierenden ein Fazit ziehen, inwiefern diese übereinstimmen. So werden auch Fähigkeiten im Bereich des Projektmanagements weiterentwickelt.

Inhaltsverzeichnis

1. Einleitung	1
2. Anforderungen	3
3. Kosten und Arbeitsplan	5
4. Entwurf	7
4.1. Schaltungslayout	7
4.2. Gehäuse	7
5. Softwareimplementation	9
6. Hardware	11
7. Testing	13
8. Handbuch	15
9. Installationsanleitung	17
10. Fazit	19
Verzeichnis verwendeter Abkürzungen und Formelzeichen	21
Literaturverzeichnis	23
Sachwortverzeichnis	23
Abbildungsverzeichnis	25
Tabellenverzeichnis	27
A. Anhang	29
A.1. Weitere Abbildungen	29

1. Einleitung

Mithilfe eines Arduinos wurde in diesem Projekt ein CO₂-gesteuerter Fensterheber simuliert, welcher dazu dienen soll die Räumlichkeiten bei schlechter Luftqualität automatisch zu lüften. Auch das automatische schließen des Fensters nach Wiederherstellung von guter Luftgüte wird simuliert.

Das Öffnen und Schließen der Fenster wird anhand von einer LED simuliert, welche nach der Überschreitung eines bestimmten Grenzwertes aufleuchtet. Sobald nach ausreichendem Lüften die Luftgüte unter einen Schwellwert gerät, soll die LED wieder aus gehen.

Damit dies möglich ist, wurde die Entwicklung von Software-Code, sowie ein Schaltungslayout und der 3D-Druck des Gehäuses selbständig vorgenommen.

Der folgende Bericht dokumentiert die Vorgehensweisen und Umsetzung von der Projektplanung, über die Implementierung von Hardware und Software, bis hin zu durchgeführten Tests und den Projektabschluss.

2. Anforderungen

Damit die Bewertung des Projektes erfolgreich wird, müssen zu Projektbeginn Anforderungen erarbeitet und festgelegt werden. Diese sind unveränderbar, da das Ergebnis sonst verfälschen würde.

Nummer	Anforderungen	Verifikationsmethode
1	Echtzeitmessung der Luftgüte	Measurement
2	Mindestmessbereich von 300 ppm bis 3000 ppm	Review
3	Visualisierung der Luftgüte mithilfe von LEDs (gut, mittel, schlecht)	Test
4	Ausgabe der Luftgüte mithilfe von LCD-Display	Test
5	Ansteuern eines Fensterscheibenmotors mithilfe einer LED simulieren	Test
6	Bei schlechter Luftgüte: Fenster öffnet sich (LED an)	Test
7	Bei guter Luftgüte: Fenster schließt sich (LED aus)	Test
8	Speichern im CSV-Format	Test, Analysis
9	Externe Abfrage über USB-Schnittstelle	Test
10	Benutzer kann zwischen drei Messprofilen auswählen (Messprofil: Abtastrate)	Test

Tabelle 2.1.: Anforderungen an das Projekt

3. Kosten und Arbeitsplan

4. Entwurf

4.1. Schaltungslayout

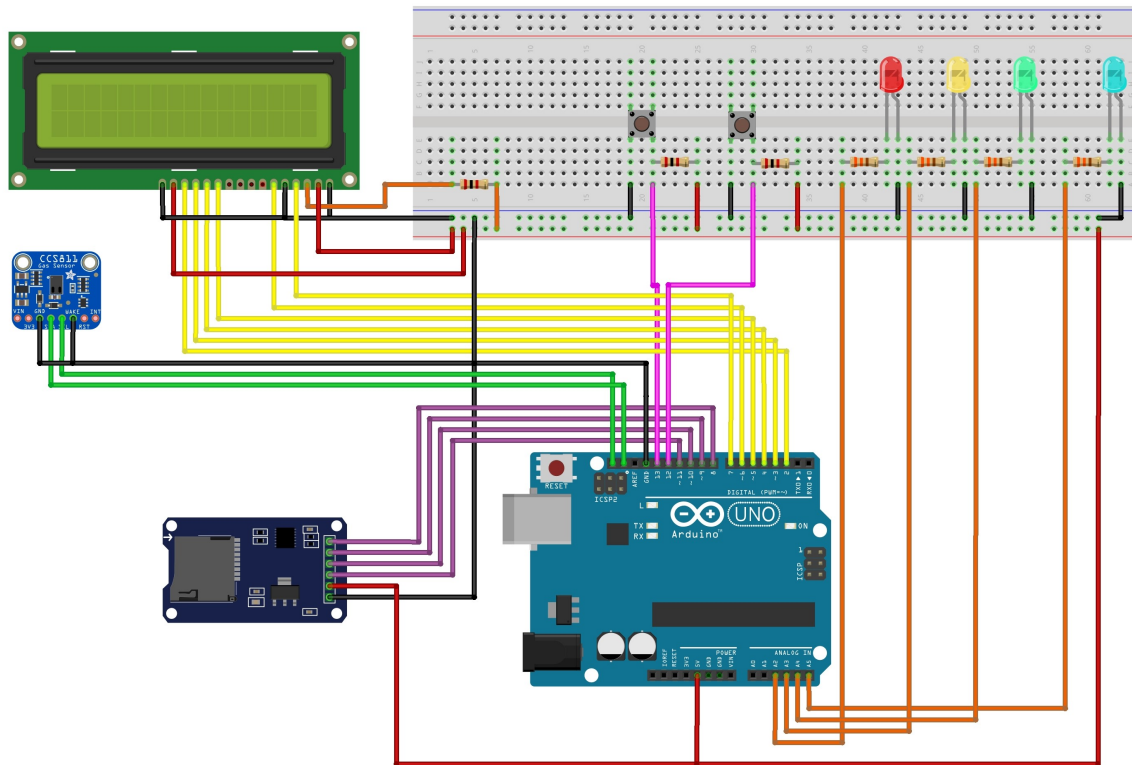


Abbildung 4.1.: Schaltungslayout vom 27.02.2020

4.2. Gehäuse

5. Softwareimplementation

Für die entgültige Struktur der Software waren mehrere Anforderungen ausschlaggebend.

Zunächst wurden drei verschiedene Messprofile definiert und implementiert, sodass der Benutzer zu Beginn zwischen eine Echtzeit-, Stunden- und Tagesmessung wählen kann. In den Messprofilen ist definiert, in welchen Abständen und wie lange Messungen durchgeführt werden sollen. Somit konnten Anforderung Nummer 1 und 10 aus der Tabelle 2.1 gemeinsam umgesetzt werden.

Nach der Wahl des Messprofils muss der Arduino den CO2-Sensor ansteuern und richtig konfigurieren. Es muss getestet werden, ob er funktionstüchtig und bereit ist eine Messung zu starten. Zudem kann der verwendete Sensor nicht nur CO2-Werte, sondern beispielsweise auch Temperaturen messen, sodass der Arduino die richtigen Werte anfordern muss. Damit dies möglich ist, mussten wir die Bibliothek `<Adafruit_CCS811.h>` einbinden.

Zunächst wird im Setup geprüft, ob der Sensor gestartet werden kann. Falls dies nicht der Fall sein sollte, wird eine Fehlermeldung ausgegeben. Nach erfolgreichem Start ist der Arduino angehalten so lange mit dem Programm zu warten, bis der Sensor zurückmeldet, dass er bereit ist, die Messung zu beginnen.

Auch während dem Programmdurchlauf wird bei jeder neuen Messung kontrolliert, ob der CO2-Sensor funktionstüchtig ist. Danach wird mithilfe der oben genannten Bibliothek der CO2-Wert gemessen und an den Arduino weitergegeben.

Nach Einlesen der Daten, vergleicht der Mikrocomputer diese mit den gegebenen Grenzwerten. Je nach Bewertung des gemessenen Werte wird eine der grün/gelb/roten LEDs eingeschaltet. Auch die blaue LED, welche die Ansteuerung des automatisierten Fensterscheibenmotors simulieren soll, wird je nach Messwert an- oder ausgeschaltet.

Zudem werden dem Anwender in Echtzeit die jeweiligen Daten im LCD Display aufgegeben, was durch die Bibliothek `<LiquidCrystal.h>` möglich ist.

Damit der Verlauf der Messung später auf Excel geplottet werden kann, wird auf einer Mikro-SD-Karte der gemessene Wert im .csv-Format als .txt-Datei abgespeichert.

Der Anwender nach nun entscheiden, ob er eine weitere Messung durchführen möchte oder nicht. Bei positiver Eingabe wird das Programm von vorne durchgeführt, während beim Ablehnen einer weiteren Messung der Mikrocomputer in einen sogenannten Sleep-Mode geht.

6. Hardware

@Serkant: Bitte auch noch das CAD-Modell einbinden!

7. Testing

Projekt: CO2-Sensor	Datum:
ID: CO201	Version: 1.0
Titel: Visualisierung der Luftqualität auf Basis von CO2-Grenzen	
Items: void ask(int)	TestKfg: 01
Zielsetzung: Der Test soll zeigen, dass die Software die gemessenen CO2-Werte richtig interpretieren kann.	
Anforderungen: R01	
Erforderliche Inputs zu Testbeginn: CO2 Werte	

Tabelle 7.1.: Test 1

Tester:	Beobachter:
Protokolldatei:	
Status:	Problembericht:

Tabelle 7.2.: Tester 1

Projekt: CO2-Sensor	Datum:
ID: CO202	Version: 1.0
Titel: Auswahl von verschiedenen Messprofilen	
Items: void ask(int)	TestKfg: 01
Zielsetzung: Der Test soll zeigen, dass der Anwender zwischen drei verschiedenen Messprofilen wählen kann.	
Anforderungen: R03	
Erforderliche Inputs zu Testbeginn: Anwender	

Tabelle 7.3.: Test 2

Tester: Alexander Herrmann	Beobachter: Johannes Ruffer
Protokoll: Zunächst wurde der Arduino an den Laptop angeschlossen und somit das aktuelle Programm g	
Status: Erfolgreich	Problembericht: Nicht vorhanden

Tabelle 7.4.: Tester 2

8. Handbuch

9. Installationsanleitung

Nummer	Bauteil	Spezifikation	Anschlusspin Arduino
1	LCD-Display	RS RW E D4 D5 D6 D7 A K	D7 GND D6 D5 D4 D3 D2 5V GND
2	LEDs	Rot Gelb Grün Blau	A3 A2 A1 A0
3	Taster	Up-Button Enter-Button	12 13
4	CCS811	WAK SDA SCL VCC	GND SDA SCL 5V
5	MicroSD-Slot	GND MISO MOSI SCK CS VCC	GND D11 D10 9 8 5V

Tabelle 9.1.: Zuordnung der Pins

10. Fazit

Verzeichnis verwendeter Abkürzungen und Formelzeichen

DHBW Duale Hochschule Baden-Württemberg

Literaturverzeichnis

Abbildungsverzeichnis

4.1. Schaltungslayout vom 27.02.2020	7
--	---

Tabellenverzeichnis

- 2.1. Anforderungen an das Projekt 3
- 7.1. Test 1 13
- 7.2. Tester 1 13
- 7.3. Test 2 13
- 7.4. Tester 2 13
- 9.1. Zuordnung der Pins 17

A. Anhang

A.1. Weitere Abbildungen