



HOCHSCHULE COBURG

Hochschule für angewandte Wissenschaften Coburg
Fakultät Elektrotechnik und Informatik

Studiengang: Master Informatik
Modul: Hardware cyber-physischer Systeme

Technische Dokumentation Smarte Küchenwaage
Betreut durch: Prof. Dr. Matthias Mörz

Eric Gendner

Abgabe des Berichts: 12. August 2022

Inhaltsverzeichnis

| | |
|--|-----------|
| Abbildungsverzeichnis | 3 |
| Tabellenverzeichnis | 4 |
| Codebeispielverzeichnis | 5 |
| Symbolverzeichnis | 6 |
| Abkürzungsverzeichnis | 6 |
| 1 Einleitung | 7 |
| 2 Hardwareaufbau | 7 |
| 2.1 Lösungskonzept | 7 |
| 2.2 Controller | 9 |
| 2.3 Wiegevorrichtung | 10 |
| 2.4 lokale Anzeige | 11 |
| 2.5 Verbinden der Komponenten | 11 |
| 3 Software | 12 |
| 3.1 Arduino | 12 |
| 3.1.1 Wägezelle | 12 |
| 3.1.2 BLE | 13 |
| 3.1.3 Segmentdisplay | 13 |
| 3.2 App | 13 |
| 3.2.1 Framework | 13 |
| 3.2.2 BLE | 13 |
| 3.2.3 Frontend | 13 |
| 4 Zusammenfassung und Ausblick | 13 |
| 4.1 Reflexion | 13 |
| 4.2 Ausblick | 13 |
| 5 Einführung | 13 |
| 6 Entwicklungsumgebung | 14 |
| 6.1 Online Entwicklungsumgebung | 14 |
| 6.2 Offline Entwicklungsumgebung | 14 |
| 7 Funktionalitäten | 14 |

| | | |
|---------------------------------|---|-----------|
| 7.1 | Verzeichnisse | 15 |
| 7.1.1 | Inhalt | 15 |
| 7.2 | Abbildungen | 15 |
| 7.3 | Tabellen | 17 |
| 7.4 | Code | 17 |
| 7.5 | Symbole | 18 |
| 7.6 | Abkürzungsverzeichnis | 18 |
| 7.7 | Glossar | 18 |
| 7.8 | Literaturverzeichnis | 19 |
| Literaturverzeichnis | | 20 |
| Glossar | | 21 |
| A Anhang | | 22 |
| A.1 | Literaturverwaltungsprogramm | 22 |
| A.2 | Titelseite | 22 |
| A.3 | Achtung bei der ehrenwörtlichen Erklärung | 23 |
| Ehrenwörtliche Erklärung | | 24 |

Abbildungsverzeichnis

| | | |
|---------|---|----|
| Abb. 1: | Aufbau der smarten Küchenwaage | 9 |
| Abb. 2: | Entwurf des Fußes und der Wägeplattform der Waage | 10 |
| Abb. 3: | Die 3D gedruckten Teile sind mit M4-Schrauben an der Wägezelle befestigt, hierzu wurde eine Aussparung an der Unterseite der Teile für die Schrauben gelassen. | 11 |
| Abb. 4: | Die Komponenten sind mit dem Arduino verbunden, sollte im Folgenden Ka- pitel 3 auf Pins eingegangen werden, entspricht dies der hier abgebildeten Verkabelung. | 12 |
| Abb. 5: | Logo der Hochschule Coburg | 15 |
| Abb. 6: | Logo der Hochschule Coburg - kleiner dargestellt | 16 |

Tabellenverzeichnis

| | | |
|---------|---------------------------|----|
| Tab. 1: | Beispieltabelle | 17 |
|---------|---------------------------|----|

Codebeispielverzeichnis

| | | |
|---------|------------------------------------|----|
| Code 1: | Beispielcode MATLAB | 17 |
| Code 2: | Wahl der Titelseite | 22 |
| Code 3: | Variablen für Titelseite | 22 |

Abkürzungsverzeichnis

| | |
|---------|---|
| BLE | Bluetooth Low Energy |
| EBA | exemplarische Beispielabkürzung |
| FEIF | Fakultät Elektrotechnik und Informationstechnik |
| IDE | Entwicklungsumgebung (Integrated Development Environment) |
| WYSIWYG | What You See Is What You Get |

1 Einleitung

Die analoge Welt wird immer digitaler, mit dem Einzug der Mikroelektronik wurden analoge Messvorrichtungen durch meist präzisere, digitale Sensoren ersetzt. Dieses Konzept zieht sich durch alle Bereiche des Lebens. Mit dem Aufkommen und der starken Verbreitung von leistungsfähigen Endgeräten, wie Computer, Smartphone oder Tablet und der massiv voranschreitenden Vernetzung verschiedenster Komponenten, hält auch die Digitalisierung Einzug in alltägliche Bereiche und Tätigkeiten wie beispielsweise dem Kochen.

Innerhalb dieser Dokumentation wird die Entwicklung einer smarten Küchenwaage beschrieben. Die smarte Küchenwaage zeichnet mit einem durchgängigem Integrierungskonzept in den normalen Alltag des Nutzers aus. Hierfür muss die Waage einfach bedienbar, die Grundfunktionalitäten einer normalen Küchenwaage und erweiterte, smarte Funktionen aufweisen. Zu den Grundfunktionen einer Waage wird im Folgendem das Anzeigen des aktuellen Gewichtes verstanden. Um die Waage smarten Funktionen zugänglich zu machen, wird eine Schnittstelle benötigt, über diese Schnittstelle kann der Nutzer mit der Waage interagieren und erweiterte Funktionen aufrufen. Die Waage besteht im Grundaufbau aus verschiedenen Hardwarekomponenten, welche in Kapitel 2 vorgestellt werden, an dieser Stelle wird auch der prinzipielle Aufbau der Waage beschrieben. Neben der Hardware, muss die Waage ebenfalls mit der passenden Software ausgestattet sein, welche die Hardwarekomponenten korrekt ansteuert und so den reibungslosen Ablauf garantiert, dies wird in Kapitel 3 dargestellt.

Innerhalb des Kapitels 4 wird der aktuelle Entwicklungsstand der Waage aufgezeigt und weitere smarte Funktionalitäten in Form eines Ausblicks dargestellt.

2 Hardwareaufbau

Im Folgenden werden verschiedene Lösungskonzepte miteinander verglichen um eine smarte Küchenwaage aufzubauen.

2.1 Lösungskonzept

Um eine smarte Küchenwaage mit erweiterten Funktionalitäten aufzubauen bestehen mehrere Möglichkeiten.

1. Eine bestehende smarte Küchenwaage reverse-engineeren und deren Übertragungssignal abfangen und manipulieren um hierauf eine eigene Anwendung aufzubauen

2. eine normale Küchenwaage nutzen, diese mit einem Microcontroller ausstatten und den gemessenen Wert übertragen
3. Aufbauen einer eigenständigen Lösung mithilfe verschiedener Sensoren und Aktoren

Die Hauptaufgabe innerhalb des ersten Szenarios liegt darin zu verstehen, wie die genutzte bestehende smarte Küchenwaage arbeitet, den Messwert aus der Datenübertragung zu extrahieren und anschließend mit diesem eine Anwendung aufzusetzen, welche weitere Features implementieren kann.

Das zweite Szenario ist hingegen schon wesentlich komplexer, hier muss das Signal über bestehende Leiterbahnen abgefangen werden. Eine besondere Herausforderung besteht darin diese zuerst freizulegen und den normalen Messvorgang nicht zu beeinflussen. Anschließend muss eine Schnittstelle aufgebaut werden, um den Nutzer weitere Funktionen zugänglich zu machen.

Mit dem eigenen Aufbau einer smarten Küchenwaage bestehen die meisten Freiheiten, so können die Komponenten beliebig gewählt und ausgetauscht werden, auch muss keine bestehende Lösung adaptiert werden. Jedoch besteht darin auch die besondere Herausforderung, die schiere Größe des Projekts. Hier muss nicht nur der Client aufgebaut werden, auch muss die komplette Waage selbst erstellt und getestet werden. Hierfür müssen bestehende Lösungen evaluiert werden. Diese macht diesen Ansatz jedoch auch so interessant, da hier mit der Hardware direkt gearbeitet werden und so ein tieferes Verständnis über diese und den genutzten Technologien erworben werden kann.

Es ist zu beachten, dass innerhalb aller Szenarien eine Schnittstelle für erweiterte Funktionalitäten benötigt wird, diese kann lokal durch zum Beispiel einem großen Bildschirm oder extern über eine Schnittstelle zu einem mobilem Endgerät bereitgestellt werden. Da davon ausgegangen werden kann, dass jeder Nutzer einer smarten Küchenwaage auch ein Handy oder Tablet besitzt, wird sich für die Lösung einer Softwareschnittstelle entschieden, da diese kostengünstiger implementiert werden kann. Mehr hierzu wird unter Kapitel 3.1.2 beschrieben.

Da mit der Erstellung einer eigenen Küchenwaage die meisten Freiheiten und das größte Lern-Potential ist, wurde sich für diesen Weg entschieden. Der allgemeine Systemaufbau ist in Abbildung 1 dargestellt. Um ein Objekt zu wiegen benötigt es eine zentrale Wiegevorrichtung, welche über einen Mikrocontroller angebunden ist, die lokale Anzeige des Gewichts übernimmt ein separates Bauteil. Über den Mikrocontroller selbst oder einem eigenständigen Bauteil soll die Kommunikation mit einem mobilen Endgerät aufgebaut werden.

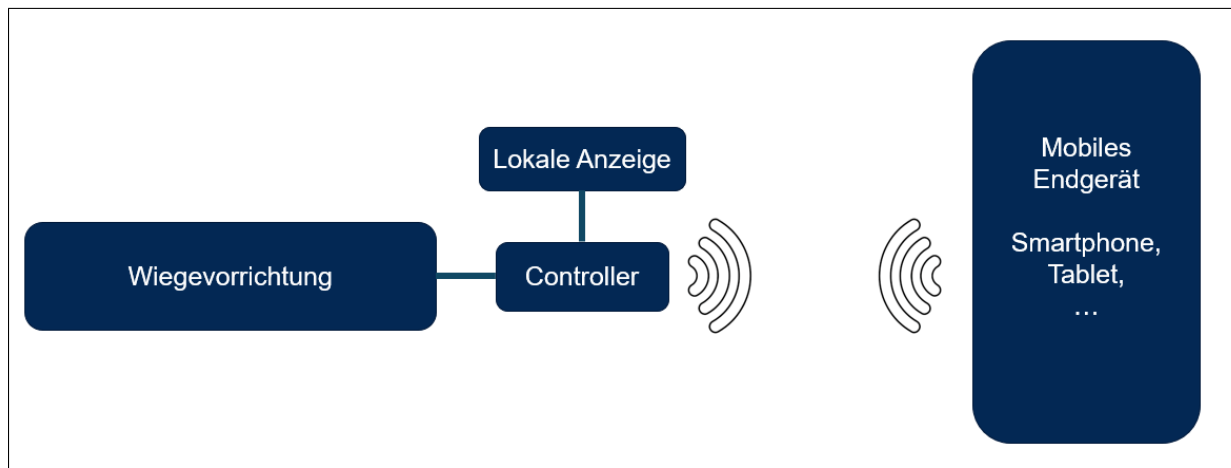


Abb. 1: Aufbau der smarten Küchenwaage

2.2 Controller

Der Controller der smarten Waage sollte möglichst kostengünstig sein und einen geringen Stromverbrauch aufweisen um gegebenenfalls auch mit einer Batterie betrieben werden zu können. Aus diesem Hintergrund wird ebenfalls zur Kommunikation mit dem mobilen Endgerät Bluetooth Low Energy (BLE) verwendet. Bestehende Möglichkeiten für den Controller sind somit die Mikroprozessoren von Arduino oder RaspberryPi. Da die Aufgabe der Prozessoren lediglich in der Messwerterfassung, Anzeige und Übertragung liegt, scheint der RaspberryPi überdimensioniert. In der Familie der Arduino-Mikroprozessoren sind die folgenden Boards mit Bluetooth ausgestattet:

- Nano IOT
- Nano BLE
- Nano BLE Sense
- BT
- Genuino 101
- MKR VIDOR 4000

Ebenso besteht die Möglichkeit ein eigenes Bluetooth-Modul wie das HM10 BLE 4.0 anzuschließen, da nur ein sehr kompakter Bauraum vorhanden ist (siehe Kapitel 2.3), wird sich zur Auswahl auf die zuvor beschriebenen Boards beschränkt, im speziellen auf die kleinen Nano Boards. Da das Messen der Umgebung nicht gebraucht wird, entfällt der BLE Sense, eine finale Größe des Codes ist noch nicht vorhersehbar, weshalb sich für den Nano BLE entschieden wird, da dieser einen größeren Speicher besitzt.

2.3 Wiegevorrichtung

Zur Aufnahme des Gewichts wird eine Wiegevorrichtung benötigt, diese setzt sich aus Sensor zur Messung des Gewichts, Standbein und Plattform für den zu messenden Gegenstand zusammen.

Innerhalb der Wägetechnik werden oft Wäghezellen verwendet, diese bestehen aus einem Federkörper mit Dehnungsstreifen. Über diese kann ein analoges Signal erzeugt werden, welches auf die über der Wägezelle abgeleiteten Masse schließen lässt. Mittels eines Digital Analog Konverters, im speziellen dem HX711, kann ein digitales Signal erzeugt werden und an den Arduino zur Weiterverarbeitung gesendet werden.

Die Wägezelle selbst ähnelt einem Quader, welcher über eine Seite mit einem Fuß und auf der anderen mit der Plattform befestigt wird. Um das Messen komfortabel zu gestalten, wurden zwei baugleiche Teile entworfen und mithilfe von einem 3D-Drucker erzeugt. Die Entwürfe der Teile sind in Abbildung 2 dargestellt.

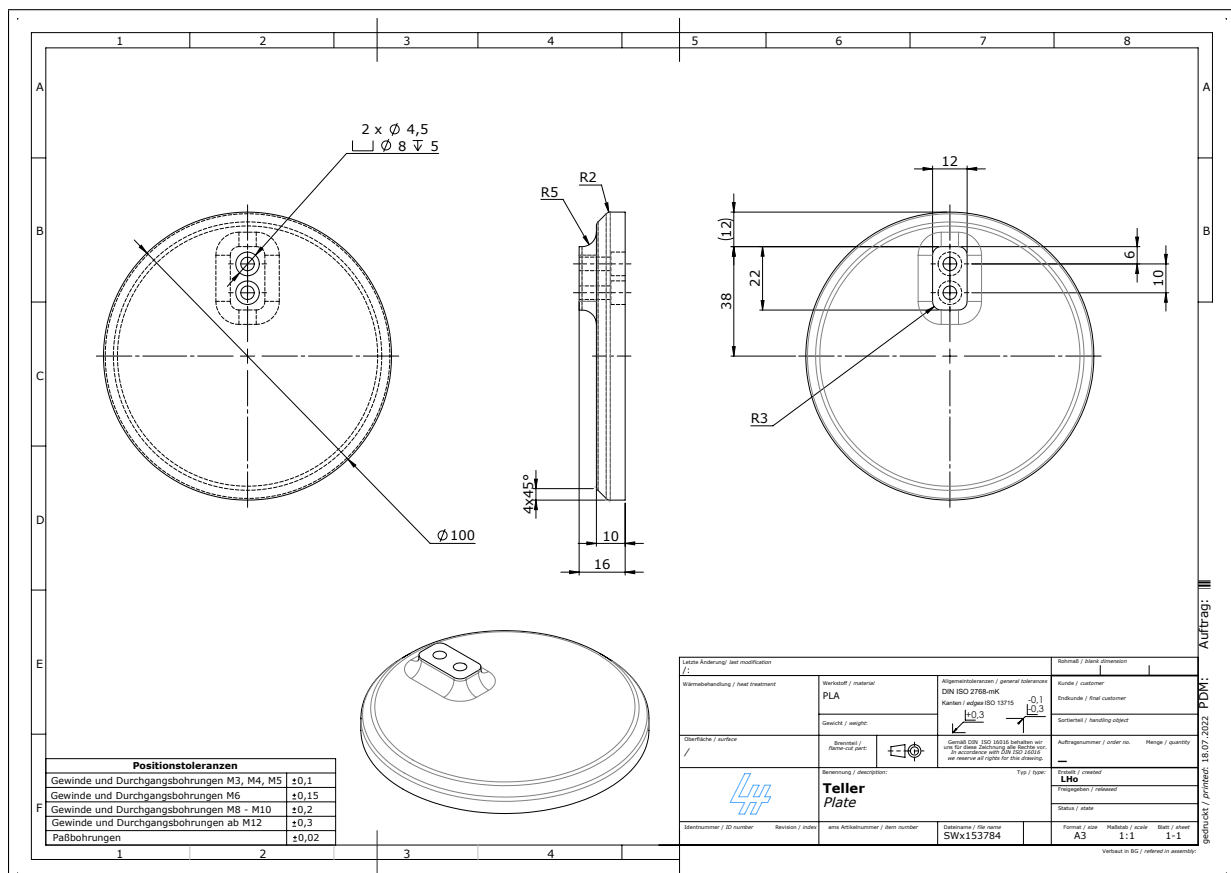


Abb. 2: Entwurf des Fußes und der Wägeplattform der Waage

Die Erhöhung ist nötig, um der Wägezelle zum messen genügend Spiel einzuräumen. Das Bauteil wird zweimal gedruckt und horizontal, sowie vertikal um 180 Grad entgegengesetzt an der Zelle angebracht, zu sehen in Abbildung 3.

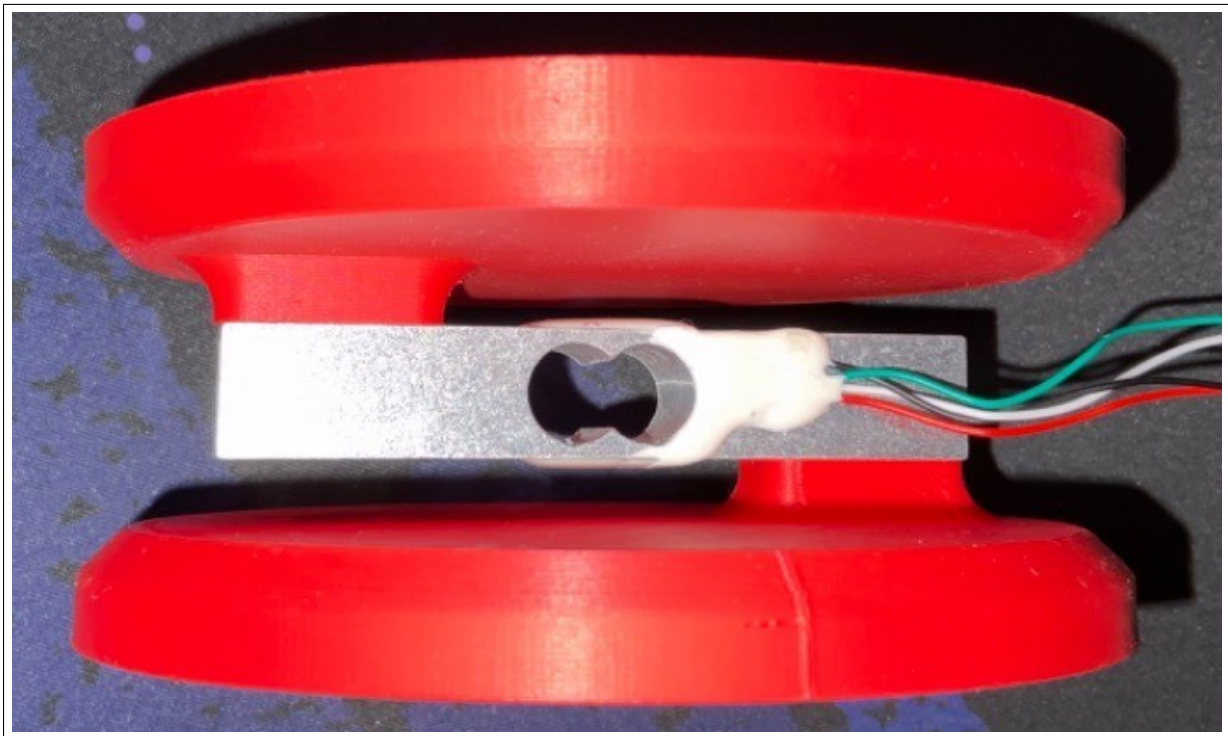


Abb. 3: Die 3D gedruckten Teile sind mit M4-Schrauben an der Wägezelle befestigt, hierzu wurde eine Aussparung an der Unterseite der Teile für die Schrauben gelassen.

2.4 lokale Anzeige

Um die Küchenwaage auch ohne mobilem Endgerät nutzen zu können, wird eine Anzeige des aktuellen Gewichts benötigt, die Wägezelle hat ein Maximalgewicht von 10 kg, da beim Backen oder Kochen die Menge der Zutaten teilweise sehr genau bestimmt werden muss, sollte die Anzeige mindestens 4 Stellen besitzen um bis auf 9999 g grammgenau wägen zu können. Um die Hardwarekosten sowie den Stromverbrauch der Waage möglichst gering zu halten, wird eine 7 Segmentanzeige mit 4 Stellen verbaut.

2.5 Verbinden der Komponenten

Die soeben vorgestellten Komponenten müssen nun mit dem Controller verbunden werden. Für die prototypische Umsetzung wird mithilfe eines Steckbretts gearbeitet, in einem finalen Produkt kann der 3D-Druck jedoch so angepasst werden, dass die Komponenten zwischen Fuß und Messvorrichtung untergebracht und verkleidet werden. Die physische Verbindung der Komponenten ist in Abbildung 4 zu sehen.

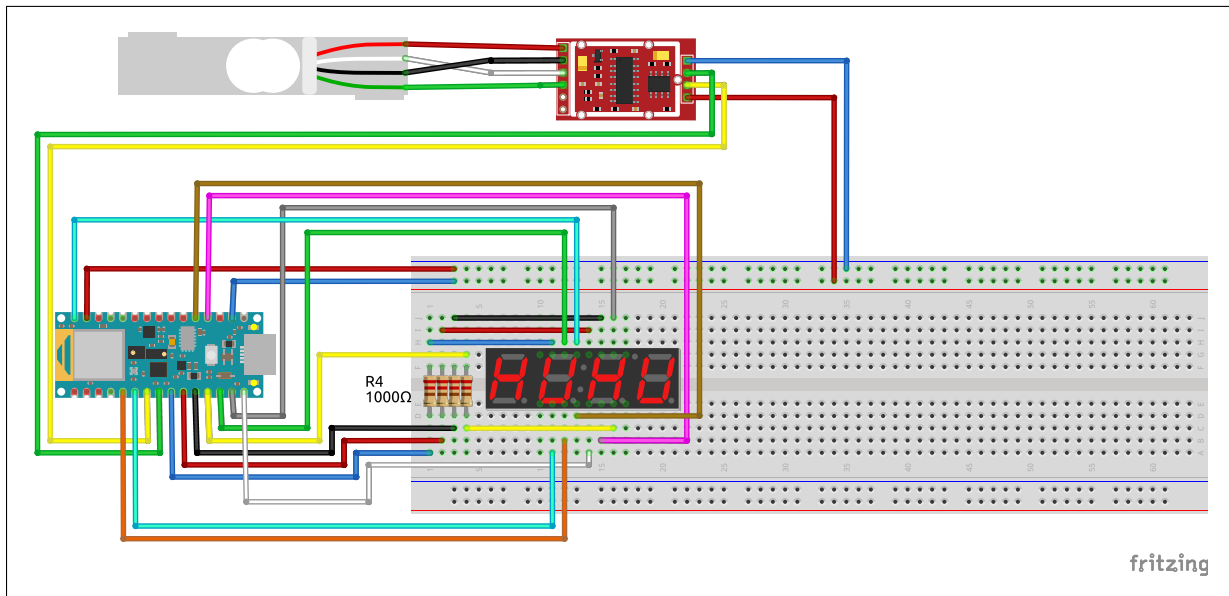


Abb. 4: Die Komponenten sind mit dem Arduino verbunden, sollte im Folgenden Kapitel 3 auf Pins eingegangen werden, entspricht dies der hier abgebildeten Verkabelung.

3 Software

Mit dem Zusammenstellen der Komponenten wird nun die passende Software benötigt, diese teilt sich auf in Software für den Microcontroller und für das mobile Endgerät.

3.1 Arduino

Essentiell für den Erfolg des Projektes ist die Datenerfassung, Datenvorverarbeitung und Datenübertragung vom Microcontroller, da verschiedene Hardwarekomponenten verwendet werden, wird der Programmcode separat beschrieben und anschließend zu einem Sketch zusammengeführt.

3.1.1 Wägezelle

Die Wägezelle ist wie bereits beschrieben mit einem Digital Analog Converter verbunden, dieser wird neben der Spannungsversorgung über 3,3 V und Ground mit zwei digitalen Pins des Arduinos verbunden, wobei einer der Pins zur Regelung des Taktes und die andere zur Signalübertragung genutzt wird. Z

um Auslesen des Sensors wird die Bibliothek HX711.h verwendet. Nach dem initialen tarieren der Waage mit einem bekannten Gewicht, was idealerweise 50% des zulässigen Gesamtgewichts der Waage schwer ist, kann das aktuelle Gewicht mit nur einem Funktionsaufruf bestimmt werden. Um nicht

3.1.2 BLE

3.1.3 Segmentdisplay

3.2 App

3.2.1 Framework

3.2.2 BLE

3.2.3 Frontend

4 Zusammenfassung und Ausblick

4.1 Reflexion

4.2 Ausblick

5 Einführung

Bei diesem Dokument handelt es sich um eine \LaTeX Vorlage für wissenschaftliche Arbeiten in der Fakultät Elektrotechnik und Informationstechnik (FEIF). Es handelt sich hierbei nicht um eine Anleitung wie \LaTeX funktioniert, sondern rein um eine Vorlage die den formalen Richtlinien der FEIF entspricht.

Es werden lediglich Hinweise und Tipps für das Arbeiten mit \LaTeX gegeben, so dass das arbeiten mit dieser Vorlage erleichtert wird. Anleitungen und Tutorials wie \LaTeX grundsätzlich funktioniert und die Syntax aufgebaut sind zu genüge im Internet vorhanden.

Eine gute Quelle hierfür ist die Dokumentation von „Overleaf“¹ oder das „Kochbuch für \LaTeX “². Eine weitere Hilfestellung ist es, die \LaTeX Datei dieses Dokuments zu öffnen um die verwendeten Syntax für die Erstellung zu sehen. Diese steckt voller nützlicher Beispiele wie mit dieser Vorlage Effizient gearbeitet werden kann.

¹<https://www.overleaf.com/learn>

²<https://archiv.dante.de/TeX-Service/TSP/tex/cookbook/cookbook.html>

6 Entwicklungsumgebung

Zunächst lohnt es sich, eine passende Entwicklungsumgebung (Integrated Development Environment) (IDE) zu wählen. Hier gilt es, sich zwischen einer Online oder Offline Entwicklungsumgebung zu entscheiden.

6.1 Online Entwicklungsumgebung

Soll eine Online IDE gewählt werden so empfiehlt sich die Verwendung von „Overleaf“³. Diese läuft rein browserbasiert und ermöglicht das gemeinsame Arbeiten an einem Dokument. Um es zu nutzen muss man sich lediglich mit einer E-Mail-Adresse anmelden.

6.2 Offline Entwicklungsumgebung

Bei einer Offline IDE handelt es sich um ein Programm, was auf dem Rechner installiert werden muss. Auch für die Installation von \LaTeX gibt es im Internet viele Anleitungen. Eine Programmempfehlung wäre das kostenfreie Programm „TEXMAKER“⁴. Hierbei handelt es sich um eine Entwicklungsumgebung, die unter den Betriebssystemen Linux, MacOS und Windows funktioniert. Sie ist flexibel nach den eigenen Wünschen anpassbar, bietet eine Vielzahl von Assistenzfunktionen, eine gute Syntax-Highlighting Funktion und eine integrierte Dokumentenvorschau.

Natürlich können auch andere Programme verwendet werden, auch hier führt eine Internetrecherche zu vielen Ergebnissen.

7 Funktionalitäten

In diesem Kapitel wird auf die Funktionalitäten dieser Vorlage eingegangen. Vielen erscheint \LaTeX Anfangs sehr kompliziert, da es sich dabei um eine Skriptsprache handelt, die nicht wie klassische Textverarbeitungsprogramme auf das „What You See Is What You Get (WYSIWYG)“ Prinzip setzt. So wird hier der Inhalt und die Formatierung erst nach dem übersetzten des Codes sichtbar. Dies erscheint für viele auf den ersten Blick ein Nachteil und Behinderung zu sein, jedoch ist das Gegenteil der Fall. Im Folgenden wird auf nützliche Funktionalitäten eingegangen, die einem nicht jedes Textverarbeitungsprogramm zur Verfügung stellt.

³<https://www.overleaf.com>

⁴<https://www.xmlmath.net/texmaker/>

Im folgenden werden die Funktionalitäten beschrieben, jedoch auch hier der Hinweis: Es gibt keine expliziten Codebeispiele im Text, die verwendeten \LaTeX Befehle sind nur im Quellcode zu sehen.

7.1 Verzeichnisse

Einer der größten Vorteile ist es, dass Verzeichnisse automatisch erstellt werden. Grundvoraussetzung ist hierbei lediglich, dass die richtigen Befehle an den entsprechenden Stellen verwendet werden.

7.1.1 Inhalt

Das Inhaltsverzeichnis wird durch die Verwendung von Überschriften im Text automatisch erstellt. Wird eine neue Überschrift hinzugefügt, taucht diese an der entsprechenden Stelle im Inhaltsverzeichnis auf. Wird eine Überschrift gelöscht, so wird sie auch aus dem Inhaltsverzeichnis entfernt.

Die Gliederungstiefe in diesem Dokument ist dreistufig:

1. Überschrift 1 (section)
 - (a) Überschrift 1.1 (subsection)
 - i. Überschrift 1.1.1 (subsubsection)

7.2 Abbildungen

Auch das Abbildungsverzeichnis wird automatisch erstellt, hierzu muss eine Abbildung mit entsprechender Syntax eingebunden werden. In diesem Beispiel haben wir in Abbildung 5 das Logo der Hochschule Coburg als Bild eingebunden. Ebenfalls wird die Position des Bildes und dessen Größe fest vorgegeben.



Abb. 5: Logo der Hochschule Coburg

In Abbildung 6 ist das identische Bild aus Abbildung 5 dargestellt, lediglich verkleinert. Durch den „caption“ Befehl tauchen beide Abbildungen im Abbildungsverzeichnis mit entsprechendem Verweis auf die Seite im Dokument automatisch auf.



Abb. 6: Logo der Hochschule Coburg - kleiner dargestellt

7.3 Tabellen

Tabellen mit \LaTeX zu erstellen ist zugegebenermaßen etwas umständlich, jedoch stellen viele Programme Assistenzfunktionen für das Erstellen zur Verfügung. Wurde das Arbeiten mit Tabellen in \LaTeX verstanden, so stellt dies kein Problem mehr dar. Tabellen wie Tabelle 1 können so mit Leichtigkeit erstellt werden. Auch diese Tabelle taucht automatisch im Tabellenverzeichnis auf.

| Kopfzeile 1 mit fester Breite | Kopfzeile 2 mit dynamischer Breite | Kopfzeile 3 (dynamisch) |
|--|--|-------------------------|
| In dieser Tabellenzeile entsteht ein automatischer Zeilenumbruch wenn der Rand überschritten wird. | Hier nicht, sie wird dynamisch erweitert. | Ich bin blau. |
| Achtung, unter meiner Zelle fehlt die horizontale Trennlinie | Zellenverbindung, rechtsbündig, ohne rechten vertikalen Rand | |
| Tabellenende | | |

Tab. 1: Beispieltabelle

7.4 Code

Oft gibt es in der Elektrotechnik und Informatik Code zu dokumentieren. Auch hier bietet \LaTeX gute Möglichkeiten. Es werden nahezu alle Programmiersprachen unterstützt, ebenfalls ist ein angepasstes Syntax-Highlighting möglich. Im Codebeispiel 1 ist eine for-Schleife in MATLAB realisiert. Auch dieses Beispiel taucht automatisch im Codebeispielverzeichnis auf.

```
for v = 1.0:-0.2:0.0
    disp(v)
end
```

Code 1: Beispielcode MATLAB

7.5 Symbole

Das Symbolverzeichnis dient der Darstellung von Symbolen, welche in Formeln verwendet werden. Hier ist darauf zu achten, dass die Symbole in der .tex-Datei „Symbolverzeichnis.tex“ im Ordner „Dateien/Verzeichnisse/“ eingetragen werden. Die dort eingetragenen Symbole werden automatisch alphabetisch geordnet und aufgelistet.

Auch hier wieder ein Beispiel:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (7.1)$$

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (7.2)$$

Hier werden die Formeln automatisch nummeriert, ebenfalls kann auf diese Nummerierung eine Referenzierung erfolgen (siehe Formel 7.2).

7.6 Abkürzungsverzeichnis

Abkürzungen wurden schon im Laufe des Dokumentes verwendet. Auch hier gilt die Abkürzungen in der Datei „Abkuerzungsverzeichnis.tex“ im Ordner „Dateien/Verzeichnisse/“ auszufüllen. Das Verzeichnis wird automatisch erstellt und alphabetisch geordnet ausgegeben. Bei erstmaliger Verwendung wird die Abkürzung automatisch ausgeschreiben und steht dahinter in Klammern. Allerdings gibt es verschiedene Möglichkeiten, Abkürzungen auszugeben.

- acs: EBA
 - Auch wenn die Abkürzung neu ist gibt dieser Befehl nur die Abkürzung aus.
- ac: exemplarische Beispielabkürzung (EBA)
- acl: exemplarische Beispielabkürzung
 - Hier wird eine vollständige Ausgabe der Abkürzung erzwungen.

7.7 Glossar

Auch für die Erstellung des Glossars gibt es eine einfache Möglichkeit. Die Begriffe, welche im Glossar auftauchen sollen müssen in der Datei „Glossar.tex“ im Ordner „Dateien/Verzeichnisse/“ eingetragen werden. Dort eingetragene Begriffe können mit den entsprechenden Befehlen

im Text aufgeführt werden. Wurden Begriffe in der Glossar Datei angelegt, allerdings nicht im Dokument verwendet, erscheinen diese bei der Ausgabe nicht im Glossars. Die Ausgabe des Glossar erfolgt alphabetisch geordnet.

Hinweis

Für das erstellen des Glossars kann es notwendig sein, den Befehl „makeglossaries“ auszuführen.

7.8 Literaturverzeichnis

Das Literaturverzeichnis wird mit Hilfe von BibTex erstellt. Hierzu muss die verwendete Literatur in der Datei „Literatur.bib“ im Ordern „Literatur/“ eingetragen werden.

Hier noch Beispiele für das Zitieren mit \LaTeX :

„The engineers started trying to understand how heat would dissipate and current flow would behave across seventy batteries by supergluing them together into groups called bricks.“ [Van16, S. 158]

Eine Faltung zweier Funktionen im Zeitbereich gestaltet sich als kompliziert, eine Vereinfachung bringt es, die Funktionen in den Laplace-Bildbereich zu überführen. Die Faltung im Zeitbereich entspricht einer Multiplikation im Bildbereich. [Pap06, S. 339f]

Literaturverzeichnis

- [Dud] DUDENVERLAG, Bibliographisches Institut G.: *Duden*. <https://www.duden.de>. – (Zugriff: 18.04.2020)
- [Pap06] PAPULA, Lothar: *Mathematische Formelsammlung für Ingenieure und Naturwissenschaftler : mit zahlreichen Rechenbeispielen und einer ausführlichen Integraltafel*. Wiesbaden : Vieweg, 2006. – ISBN 9783834801562
- [Van16] VANCE, Ashlee: *Elon Musk*. Random House UK Ltd, 2016 https://www.ebook.de/de/product/25175854/ashlee_vance_elon_musk.html. – ISBN 0753557525

Glossar

Glossar

„selbstständig oder als Anhang eines bestimmten Textes erscheinendes Wörterverzeichnis“ [Dud]

A Anhang

Der Anhang bekommt automatisch eine eigene Überschrift. Hier nun noch ein Tipp für die Literaturarbeit mit \LaTeX .

A.1 Literaturverwaltungsprogramm

Neben „Citavi“ und „Zotero“ gibt es das Programm „JabRef“⁵ speziell für \LaTeX . Es ist ein Programm um die Literatur sinnvoll und einfach zu verwalten mit dem Vorteil, dass es strukturierte .bib Dateien ausgibt. Auch diese Software ist kostenfrei und für Linux, MacOS und Windows verfügbar.

A.2 Titelseite

In der Präambel ist die Titelseite zu wählen (siehe Code 2). Ausgewählt wird durch auskommentieren der nicht benötigten Datei.

```
% Titelseite auswaehlen
\include{Dateien/Allgemein/Titelseite}
%\include{Dateien/Allgemein/Titelseite_Praxisbericht}
```

Code 2: Wahl der Titelseite

Weiterhin sind alle Variablen mit Inhalt zu füllen (siehe Code 3).

```
%-----%
%           Titelseite           %
%-----%

% in Variablen den Inhalt fuer die Titelseite eintragen

\newcommand{\Studiengang}{<Name des Studiengangs>}
\newcommand{\Dokumentenart}{<Dokumentenart>}
\newcommand{\Dokumententitel}{<Titel der Arbeit>}
\newcommand{\Autorenname}{<Autorenname>}
\newcommand{\Abgabedatum}{<Abgabedatum>}
\newcommand{\Betreuer}{<Prof. Dr. Vorname Nachname>}

% Variablen fuer Titelseite des Praxisbericht
```

⁵<https://www.jabref.org>

```
\newcommand{\UnternehmenName}{<Firma>}
\newcommand{\UnternehmenAbteilung}{<Abteilung>}
\newcommand{\UnternehmenStrasse}{<Strasse>}
\newcommand{\UnternehmenOrt}{<PLZ Ort>}
\newcommand{\UnternehmenBetreuer}{<Name des Betreuers>}
\newcommand{\BetreuerFunktion}{<Funktion des Betreuers>}
\newcommand{\BetreuerTelefon}{<Telefonnummer des Betreuers>}
\newcommand{\BetreuerEmail}{<Mailadresse des Betreuers>}

\newcommand{\DatumBeginn}{<DD.MM.JJJJ>}
\newcommand{\DatumEnde}{<DD.MM.JJJJ>}
\newcommand{\DatumAbgabe}{<DD.MM.JJJJ>}
```

Code 3: Variablen für Titelseite

A.3 Achtung bei der ehrenwörtlichen Erklärung

Ich versichere hiermit, dass ich **meine/n** Dokumentation selbstständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt, sowie nicht an anderer Stelle als Prüfungsarbeit vorgelegt habe.

Ehrenwörtliche Erklärung

Ich versichere hiermit, dass ich meine Dokumentation selbstständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt, sowie nicht an anderer Stelle als Prüfungsarbeit vorgelegt habe.

Ort

Datum

Unterschrift