



STUDIENARBEIT
des Studiengangs Informationstechnik
der Dualen Hochschule Baden-Württemberg Mannheim

THEMA
[Thema]

von
Belana Roman & Noah Wiederhold
[Datum]

Bearbeitungszeitraum: 24.12.2020 - 02.04.2021, 28.06.2021 - 30.09.2021
Matrikelnummer, Kurs: <Matrikelnummer>, TINF20IT1
Ausbildungsbetrieb: Deutsches Zentrum für Luft- und Raumfahrt
Betreuer: <Name des Betreuers>

Erklärung

Wir versichern hiermit, dass unsere Studienarbeit mit dem

THEMA

[Thema]

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.*

* falls beide Fassungen gefordert sind

Mannheim , den [Datum] Mannheim , den [Datum]

Kurzfassung

Abstract

Inhaltsverzeichnis

Abbildungsverzeichnis

Anhangsverzeichnis

1. Einleitung	2
1.1. Motivation	2
1.2. Zielsetzung	2
2. Strukturierte Aufgabenstellung	4
3. Methoden & Verfahren	5
3.1. Software	5
3.2. Hardware	5
4. Durchführung/Bearbeitung/...	6
4.1. Theorie	6
4.1.1. ARDUINO	6
4.1.2. ARDUINO UNO	7
4.1.3. Feuchtigkeitssensoren	9
4.1.4. Temperatursensoren	11
4.1.5. Lichtsensor	11
4.1.6. Arduino IDE	11
4.2. Konzept	11
4.2.1. Zusammenhang Module	11
4.2.2. Messmodul	11
4.2.3. Speichermodule	11
4.2.4. Auswertungsmodul	11
4.2.5. Automatisierungsmodul	11
4.3. Implementierung	11
4.3.1. Zusammenhang Module	11
4.3.2. Messmodul	11
4.3.3. Speichermodule	11
4.3.4. Auswertungsmodul	11

4.3.5. Automatisierungsmodul	11
5. Ergebnis	12
6. Kritische Reflexion	13
7. Ausblick	14
Literatur	15

Abbildungsverzeichnis

4.1. ARDUINOUNOSketch 8

Anhangsverzeichnis

Anhang A: Begriffsdefinitionen	16
A.1. Begriffsbezeichner	16
Anhang B: Abkürzungsverzeichnis	17

1. Einleitung

1.1. Motivation

Innerhalb eines dualen Studiums ist das Wechseln der Standorte zwischen Theorie- und Praxisphasen ein zentraler Bestandteil des Alltages. Der Wechsel findet dabei meist alle 3 Monate statt, und hat je nach Entfernung der beiden Standorte häufig auch einen Wechsel des Wohnortes für diese Zeit zur Folge. Bei der Haltung beziehungsweise Zucht von Pflanzen stehen wir Studenten daher vor einigen Herausforderungen. Entweder die Pflanzen müssen alle drei Monaten zum nächsten Standort transportiert werden, oder es müssen stets neue Pflanzen gezüchtet beziehungsweise gekauft werden. Um solche Probleme anzugehen, gibt es auf dem Markt einige vollautomatische Smart-Garden-Systeme, die den Anbau von Pflanzen in jeder Wohnung ermöglichen sollen und die Pflanzen dabei selbstständig mit Wasser, Licht und Nährstoffen versorgen. Innerhalb solcher Systeme finden allerdings nur wenige Pflanzen Platz und je umfangreicher ein solches System sein soll, desto schneller steigt auch der Preis dafür.

Ziel dieser Arbeit ist daher die Entwicklung einer eigenständigen digitale Überwachung und automatisierten Pflege von Pflanzen im eigenen Zuhause, um eine Möglichkeit zur Erhaltung von Pflanzen über längere Abwesenheiten hinweg zu schaffen.

1.2. Zielsetzung

Innerhalb des zu entwickelnden Systems sollen verschiedenen Parametern bezüglich des Pflanzenwachstums, darunter die Bodenfeuchte der Erde sowie die Temperatur im

1.2. Zielsetzung

Raum überwacht und schließlich für den Anwender geeignet dargestellt werden, sodass eine Überwachung der Pflanzen auch aus der Ferne möglich wird. Auf die gemessenen Parameter soll das Überwachungssystem darüber hinaus auch entsprechend mit automatischen Systemen, wie einem automatischen Bewässerungssystem, reagieren.

Das so entwickelte digitale Gartensystem lässt sich schließlich individuell anpassen und beliebig um weitere Parameter oder weitere Pflanzen erweitern.

2. Strukturierte Aufgabenstellung

(verifizierbar, validierbar)

Aufgabe 1 Darstellung der theoretischen Grundlagen 1.1 arduinouno 1.2 arduino uno
1.3 Sensoren 1.3.1 Feuchtigkeitssensoren ... Aufgabe 2 Entwicklung eines Konzepts 2.1
Zusammenhang Module Schnittstellen Wie sollen Daten ausgetauscht werden 2.2
Modul Arduino 2.3 Modul Web... Aufgabe 3 Implementierung 3.1 Modul ...

3. Methoden & Verfahren

Um die Reproduzierbarkeit dieser Arbeit zu gewährleisten, werden im Folgenden die verwendeten Methoden und Verfahren, speziell die verwendete Hard- und Software vorgestellt.

3.1. Software

(Versionsnummern)

3.2. Hardware

(Produktnummern)

4. Durchführung/Bearbeitung/...

(Reihenfolge basiert auf Reihenfolge die in Aufgabenstellung erwähnt wurde.)

4.1. Theorie

4.1.1. ARDUINO

ARDUINO ist eine Open-Source Plattform für die anfangersfreundliche Entwicklung von Hardwareprojekten. Seit der Gründung 2005 durch Massimo Banzi und David Cuartielles am Ivrea Interaction Design Institute in Italien hat sie sich zu einem beliebten Werkzeug für Hobby-Elektroniker und Ingenieure in Bereichen wie IoT (Internet of Things) und Robotik entwickelt.

Die Plattform-eigene Hardware erstreckt sich über verschiedene Produktserien mit den Namen Nano, Classic und MKR. Nano ist dem Namen entsprechend eine Boardfamilie, die sich durch ein besonders kompaktes Design auszeichnet. Die Classic Reihe ist im Gegensatz dazu vergleichsweise groß konzipiert und bietet Basis Boards mit vielen Ein- und Ausgängen. Die MKR Serie zeichnet sich im Vergleich zu den anderen Produktreihen durch ein hohes Maß an Konnektivität aus, da fast alle Boards von Werk aus unter anderem mit Wi-Fi- und Bluetooth-Chips ausgestattet sind. Einige verfügbare Boards der jeweiligen Reihen sind in Abbildung X zu sehen. Von ARDUINO selbst werden innerhalb dieser Serien vor allem Boards und Shields entwickelt.

Neben der Hardware bietet die ARDUINO-Plattform eine eigene integrierte Programmierungsumgebung (IDE), welche es Anwendern ermöglicht schnell kompatiblen Code

zu schreiben und diesen auf die Boards zu übertragen. Vor allem die Möglichkeit Code nicht nur herkömmlich per Kabel, sondern auch kabellos über WLAN oder Bluetooth zu übertragen vereinfacht und beschleunigt den Entwicklungsprozess sehr. Für die Programmierung mit der Entwicklungsumgebung wird eine C++ ähnliche Sprache verwendet. Zusätzlich können über die Umgebung Code-Bibliotheken in eigene Projekte eingebunden werden, welche gängige Aufgaben übernehmen oder Kompatibilität zu Geräten und Mikrocontrollern von Drittanbietern ermöglichen.

Aufgrund dieser Kompatibilität, der Möglichkeit eine Vielzahl von analogen sowie digitalen Bauteilen zu verwenden und vergleichsweise geringen Anschaffungskosten, sind die Boards zum Beispiel im Bereich der DIY-Bewegung (“Do It Yourself“-Bewegung) sehr beliebt und verbreitet. Im Bereich professioneller Anwendung wird ARDUINO zum Beispiel für Prototyping und Rapid-Iteration verwendet.

4.1.2. ARDUINO UNO

Der ARDUINO UNO ist ein Mikrocontroller-Board der ARDUINO-Plattform aus der Classic-Produktreihe. Zum Zeitpunkt dieser Arbeit existieren 3 Revisionen des Basis-Boards und jeweils verschiedenste um zusätzliche Bauteile erweiterte Versionen. Für die vorliegende Arbeit wurde aus dieser Vielzahl an Boards konkret die Wi-Fi-Version der Revision 2 ausgewählt, da diese bereits mit einem WLAN-Chip ausgestattet ist. Für die Anwendung als Steuereinheit eines smarten Pflanzenzuchtsystems ist dies von Vorteil, da kein Netzkabel oder zusätzliche Hardware für die Übertragung von Messwerten an die auswertende Instanz nötig ist. Die folgenden Ausführungen beziehen sich demzufolge auf diese Version des Boards.

Das Board ist in Abbildung X zu sehen. Die Abbildung 4.1 zeigt zusätzlich dazu den schematischen Aufbau des Boards mit Kennzeichnung wichtiger Bauteile.

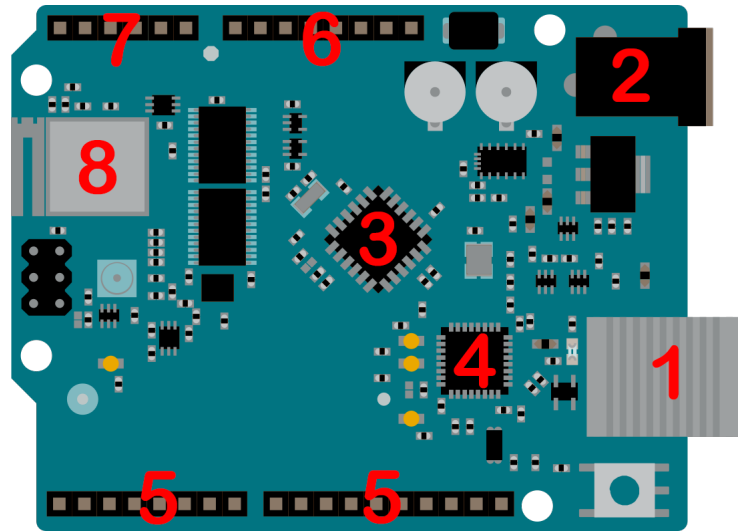


Abbildung 4.1.: Struktur des ARDUINO UNO WIFI REV2 ¹

Das Board besitzt einen Eingang für sogenannte “Barrel Plugs“ (Abbildung X Markierung Y), welcher der Stromversorgung dient. Er ermöglicht es den ARDUINO mit Spannungen zwischen 6 und 20 Volt zu betreiben. Aufgrund des geringen Strombedarfs des Boards kommen dafür normale Netzgeräte sowie Powerbanks infrage.

Die Stromversorgung kann zusätzlich über einen USB Typ-B Port (Abbildung X Markierung Y) erfolgen. Dieser dient jedoch vor allem zum klassischen Laden von Programmcode über Kabel.

Für die USB Verbindung und den ISP Flash besitzt der Arduino einen eigenen Controller (Abbildung X Markierung Y).

Die zentrale Recheneinheit des Boards ist der Mikroprozessor ATmega4809 (Abbildung X Markierung Y *TODO*), welcher mit Taktraten von 16MHz arbeitet. Er besitzt 48 KB Flash Memory, 6144 Bytes SRAM und einen 256 Bytes großen EEPROM.

Die Kommunikation mit externer Hardware erfolgt über digitale (Abbildung X Markierung Y) und analoge (Abbildung X Markierung Y) I/O-Pins. Mit diesen können z.B. Werte von Sensoren gelesen oder auch geschrieben werden. Die Versorgung der

¹ *UNO WiFi Rev2* / *Arduino Documentation* / *Arduino Documentation* [22]

angeschlossenen Hardware erfolgt über separate Power-Pins, die entweder mit 3,3 oder 5 Volt arbeiten.

Die WIFI-Version des Boards besitzt im Vergleich zum Basis-Board zusätzlich ein NINA-W102 WLAN- und Bluetooth-Modul. Über dieses wird die Kommunikation zwischen verschiedenen, räumlich getrennten Geräten und das Hochladen von ARDUINO Sketches “over the air“ ermöglicht.

4.1.3. Feuchtigkeitssensoren

Zur Feuchtigkeitsbestimmung mit Hilfe von Sensoren können 2 grundlegende Verfahren verwendet werden. Die einfachste Methode ist dabei die Messung über die elektrische Leitfähigkeit. Hierbei wird mit Hilfe von zwei Elektroden die Leitfähigkeit der zwischen den Elektrodenstäben befindlichen Erde gemessen und von dieser auf die Bodenfeuchte geschlossen. Dieses Verfahren beruht jedoch auf dem direkten Kontakt von Elektroden mit Feuchtigkeit im Boden, wodurch die Elektroden schnell korrodieren. Weiterhin beeinflussen die im Wasser gelösten Salze und Mineralien die Leitfähigkeit wodurch die Messergebnisse unvorhersehbar beeinflusst werden.

Die Alternative zur Messung mit Elektroden ist die kapazitive Messung. Hierbei wird ein Kondensator mit einem Dielektrikum verwendet. Das hygroskopische Dielektrikum tauscht dabei Wasserdampf aus dem Boden mit dem Dielektrikum aus. Über eine Messung der Kapazität des Kondensators kann anschließend die relative Feuchte des Bodens über den Wasseranteil des Dielektrikum bestimmt werden. Dieses Verfahren hat den Vorteil, dass kein direkter Kontakt zwischen dem Boden und den Platten des Kondensators vorliegt und dadurch, bei korrekter Verwendung, praktisch keine Korrosion auftritt. Weiterhin kann über dieses Prinzip, eine Auswirkung von im Wasser gelösten Salzen auf das Messergebnis verhindert werden.

Auf Grund der Vorteile des kapazitiven Messverfahrens gegenüber dem Messverfahren mit Elektroden wurden im Rahmen dieser Arbeit nur kapazitive Sensoren verwendet.

Konkret wurde die Sensoren 101020614 von Seeed Studio und SEN0193 von DFRobot verwendet.

4.1. Theorie

Beide Sensoren lassen sich entweder mit 3,3 oder 5 Volt betreiben und liefern analoge Werte. Im direkten Vergleich unterscheiden sich die Messwerte beider Sensoren

4.1.4. Temperatursensoren

4.1.5. Lichtsensor

4.1.6. Arduino IDE

4.2. Konzept

4.2.1. Zusammenhang Module

4.2.2. Messmodul

4.2.3. Speichermodul

4.2.4. Auswertungsmodul

4.2.5. Automatisierungsmodul

4.3. Implementierung

4.3.1. Zusammenhang Module

4.3.2. Messmodul

4.3.3. Speichermodul

4.3.4. Auswertungsmodul

4.3.5. Automatisierungsmodul

5. Ergebnis

(Kopie der Aufgabenstellung mit gelöst/gelöst, vielleicht als Tabelle)

6. Kritische Reflexion

(Stellung nehmen zum Ergebnis) (Eigene Meinung zum Ergebnis)

7. Ausblick

(Erweiterungsmöglichkeiten) (Coole weitere Funktionen) (Was kann verbessert werden?)

Literatur

- [22] *UNO WiFi Rev2 / Arduino Documentation / Arduino Documentation*. 23.11.2022.
URL: <https://docs.arduino.cc/hardware/uno-wifi-rev2>.

Anhang A.

Begriffsdefinitionen

A.1. Begriffsbezeichner

Anhang B.

Abkürzungsverzeichnis