

**autom@ted**

Das autonome  
Tomatengewächshaus

**Jahresarbeit**

von

**Friedjof Noweck**

Klasse 10/11 2018/19

# Inhaltsverzeichnis

1	Motivation.....	1
2	Planung und Konstruktion.....	1
2.1	Tomaten.....	2
2.1.1	Aufzucht und Verarbeitung der Tomatenpflanzen.....	2
2.2	Das Gewächshaus.....	3
2.2.1	Planung und Bau.....	3
2.2.2	Herausforderungen beim Bau.....	5
2.3	Die Programmierung.....	6
2.3.1	Kommunikation mit dem Handy.....	7
2.3.2	Stabile Software.....	8
2.3.3	Fehler erkennen und finden.....	8
2.3.4	User Interface [UI].....	9
2.3.5	Die Konfigurationsdatei.....	9
2.3.6	Die Softwarearchitektur.....	9
2.3.6.1	Hauptprogramme.....	10
2.3.6.2	Module.....	11
2.3.7	Das Modul Logging.....	11
2.4	Die Sensoren.....	11
2.5	Der Hubzylinder.....	15
2.6	Elektronik.....	15
2.7	Mein Projekt-Blog im Internet.....	16
2.8	Die Kooperation mit dem Wissenschaftsladen Bonn.....	17
2.9	Kleinere Version meines Gewächshaussystems.....	17
3	Schlussenteil und Fazit.....	18
4	Erklärung.....	18
5	Quellenangaben.....	19
5.1	Pflanzen.....	19
5.2	Inspiration.....	19
5.3	Hardware.....	20
5.4	Software.....	20
6	Weitere Bilder.....	22

## 1 Motivation

Schon seit meiner Kindheit hat mich Technik fasziniert: Zunächst waren es einfache Spielzeuge, die ich zusammen mit meinem Vater konstruierte, dann immer mehr selbst konstruierte Dinge mit Lego-Technik. Der Übergang von Lego-Technik zum Einplatinencomputer „Raspberry Pi“ und der damit einhergehenden Notwendigkeit eine Programmiersprache zu erlernen, war für mich eine kontinuierliche Entwicklung, die ich bis heute genieße.

So war die Idee meiner Jahresarbeit, „die Entwicklung eines autonomen Gewächshauses“, motiviert durch jene Technikbegeisterung, gepaart mit einem für den Menschen praktischen Anwendungszweck. Auf diese Weise konnte ich meine Ziele:

- eine Programmiersprache erlernen,
- technische Dinge konstruieren und
- etwas für die Natur tun

im Rahmen meiner Jahresarbeit gleichzeitig verwirklichen.

Konkret hatte ich die Vision, dass das autonome Gewächshaus verschiedene Daten erfassen, interpretieren und für den Benutzer aufbereiten könnte. Mein Wissen eignete ich mir aus Büchern und aus dem Internet an. Darüber hinaus stand mir mein Vater mit Rat und Tat zur Seite.

Zur Steuerung des Gewächshauses wählte ich den Einplatinencomputer Raspberry Pi 3 und den Arduino Uno aus. Dabei arbeitet der Raspberry Pi als Hauptsteuereinheit. Er speichert alle Daten und wertet sie anschließend aus. Der Arduino Uno dient nur dazu, die Bodenfeuchtigkeitssensoren abzufragen.

Neben dem Gewächshaus baute ich eine Internetseite<sup>1</sup> auf, auf die ich regelmäßig Blogbeiträge hochlud und Fotos von meiner Arbeit veröffentlichte. Gegen Ende der Jahresarbeit erstellte ich auch den Instagram – Account `project_automated`.

## 2 Planung und Konstruktion

Meine Planungen für das Gewächshaus starteten am 7. April 2018 und bereits am 13. April 2018 konnte ich mit dem Bau des Gewächshauses beginnen. Zuerst zeichnete ich einige Entwürfe und entschied mich dann für ein 2 m langes, 2,3 m hohes und 80 cm tiefes Gewächshaus, indem 4 Tomatenpflanzen Platz finden. Die Tomaten-

---

1 Vgl. Quellenverzeichnis unter F. Noweck

pflanzen stehen in variablen Kästen, die mit Erde gefüllt sind und nach der Bepflanzung durch eine Frontklappe entnommen werden können. Dies erleichtert das Befüllen mit neuer Erde. Rechts neben den Kästen ist die Technik in einem wasserdichten Elektrokasten untergebracht und hinter diesem befindet sich der Anschluss für den Wasserschlauch.

### 2.1 Tomaten

Im Frühjahr 2018 habe ich mich dafür entschieden, die Tomatensorten Ernteglück, Rumänische Flaschentomate und Omas Beste zu pflanzen. Dabei habe ich auf die Größe geachtet und zwischen den Sorten Ernteglück und Omas Beste, die sehr groß werden, die kleinere Sorte Rumänische Flaschentomate gepflanzt<sup>2</sup>. Somit konnte ich vier Tomatenpflanzen in meinem Gewächshaus unterbringen.

Ich habe mich auch mit den verschiedenen Krankheiten von Tomaten befasst und überlegt, wie ich diese möglichst vermeiden kann. Trotz einiger Vorkehrungen, wie das Trockenhalten der Pflanzen und einer guten Belüftung, erkrankten die Sorten Ernteglück und Omas Beste an der Dürrefleckenkrankheit<sup>3</sup>. Die Tomatensorte Rumänische Flaschentomate stellte sich als resistenter heraus. Die Dürrefleckenkrankheit befällt von unten die Blätter der Tomatenpflanzen. Es entstehen auf den Blättern Flecken, die bis ins Dunkelbraune gehen. Auch im Jahr 2019 zeigte sich diese Krankheit, obwohl ich das Gewächshaus gereinigt und die Erde erneuert habe. Das könnte daran liegen, dass der Pilz nicht nur in der Erde, sondern auch in Holz überwintern kann. Daher würde ich, wenn ich das nächste Mal ein Gewächshaus baue, es aus Aluminium bauen. Das hätte zudem den Vorteil, dass es leichter wäre.

#### 2.1.1 Aufzucht und Verarbeitung der Tomatenpflanzen

Dieses wie auch letztes Jahr zog ich die Tomaten zuerst in kleinen Töpfen vor, bevor ich sie vereinzelte. Dabei wählte ich die Pflanzen aus, die meiner Ansicht nach die kräftigsten waren. Da ich viele gesät hatte, hatte ich eine große Auswahl<sup>4</sup>.

Im ersten Jahr benutze ich Spiral-Rankhilfen aus Metall, um den Tomaten einen besseren Halt zu geben. Dies veränderte ich im zweiten Jahr und nahm Bänder, wie in großen Gewächshäusern für Tomaten<sup>5</sup>, da die Rankhilfen nicht sehr einfach zu befestigen waren und auch keinen guten Halt in den Pflanzkästen hatten.

---

2 Mehr dazu auch im Blogbeitrag vom 04. und 05.05.2018 unter „Die Tomatenpflanzen“

3 Mehr dazu im Dokument zu „Tomatenkrankheiten und Schädlinge“

4 Mehr dazu auch in den Blogbeiträgen vom 12.05.2018 und folgenden und auf meiner Internetseite unter Tomaten > Die Entwicklung der Tomaten 2018/19

5 Quelle im Dokument „Quellenangaben“ unter „Pflanzen“

Nach der Ernte von 2018 verarbeitete ich die Tomaten weiter. Einmal machte ich aus den Tomaten Tomatensoße. Aber um die Tomaten auch länger haltbar zumachen, trocknete ich sie auch. Einige von den getrockneten Tomaten legte ich in Öl ein.<sup>6</sup>

## 2.2 Das Gewächshaus

### 2.2.1 Planung und Bau

Bei der Planung des Gewächshauses hatte ich zuerst die Idee, ein kleines Gewächshaus zum Vorziehen für drinnen zu bauen, doch dann reizte mich die Idee, ein größeren Gewächshaus für draußen zu konstruieren, da ich dies als größere Herausforderung empfand.<sup>7</sup>

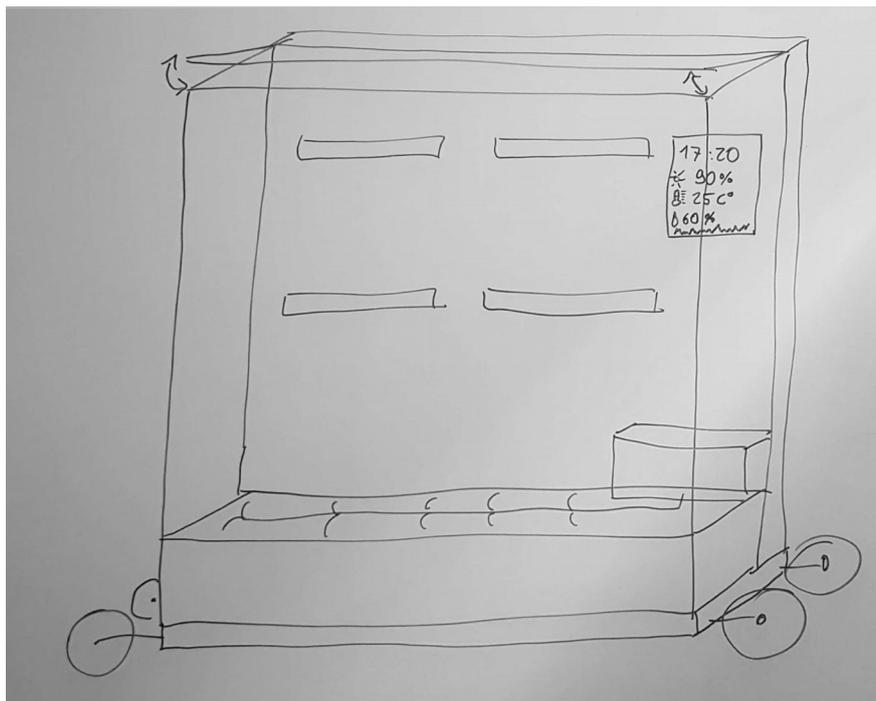


Abbildung 3: Erster Entwurf des Gewächshauses

Am darauffolgenden Wochenende legte ich bereits mit dem Bau des Gewächshauses los. Ich arbeitete sehr motiviert und kam schnell voran. Nach ein paar Wochenenden, an denen ich fast durchgehend daran arbeitete, stand das Gewächshaus. Als Materialien benutzte ich ausschließlich Holz, da sich hiermit relativ einfach bauen lässt. Jedoch wurde das Gewächshaus dadurch sehr schwer und so konnte ich die vorgesehenen Räder, zum einfacheren Verschieben, nicht einbauen. Außerdem dehnt

<sup>6</sup> Mehr dazu auch auf meinem Block unter *Rezepte mit Tomaten* oder im Wochenrückblick vom 04.09.2018 und folgenden

<sup>7</sup> Mehr dazu auch in meinem ersten Blogeintrag

sich Holz stark aus, wenn es wärmer wird und man muss einen gewissen Spielraum mit einplanen.



*Abbildung 4: Das fertige Gewächshaus*



*Abbildung 5: Das Gewächshaus der Schule*

Ich habe für das Gewächshaus eine durchsichtige Folie gewählt, da entsprechend viel Licht einfallen kann und man zudem auch von außen das Wachstum der Pflanzen gut beobachten kann.

Als Dach nahm ich zuerst ein dünnes, durchsichtiges Wellblechdach aus Plexiglas. Doch als sich dieses als zu instabil herausstellte, verbaute ich ein stabileres, schwarzes Dach<sup>8</sup>. Zudem habe ich eine Regenrinne angebracht, sodass Regenwasser gesammelt werden kann.



*Abbildung 6:  
Regenwasserableitung mittels  
Dachrinne*

Bisher bewässert das Gewächshaus die Pflanzen mit Leitungswasser über ein Magnetventil. Das ließe sich in Zukunft modifizieren, sodass die Pflanzen mit dem gesammelten Regenwasser bewässert werden könnten.<sup>9</sup> Dies könnte zu einem besseren Wachstum der Tomaten führen, weil durch Regenwasser mehr Nährstoffe aufgenommen werden können. Um auch in trockenen Zeiten die Tomaten zuverlässig mit Wasser versorgen zu können, würde ich eine Mischform für die Zukunft wählen. Dies könnte so funktionieren, dass wenn kein Regenwasser mehr vorhanden ist, mit Leitungswasser bewässert wird. Dadurch könnte eine permanente Bewässerung gewährleistet werden.

---

<sup>8</sup> Mehr zu den Problemen unter „Herausforderungen beim Bau“

<sup>9</sup> Für den Schulgarten der FWS Bonn habe ich bereits ein funktionierendes Regenwassersystem mit einer Pumpe konstruiert.

### 2.2.2 Herausforderungen beim Bau

Die durchsichtige Folie erwies sich als teilweise brüchig, sodass sie an mehreren Stellen einriss. Dies war einmal der Fall, als ein kräftiger Windstoß die vordere Klappe aus den Scharnieren riss und einmal, als mir die Leiter abrutschte<sup>10</sup>. So musste ich die Folie mit Klebeband reparieren.

Auch das dünne, transparente Dach blieb nicht von Wetterextremen verschont.



*Abbildung 7: Defektes Dach*

Das Dach hatte ich so konstruiert, dass man es relativ einfach abnehmen kann. Leider führte dies dazu, dass das Dach bei einem starkem Wind herunter geweht wurde. Zum Glück waren noch keine Tomaten im Gewächshaus und so hielt sich der Schaden in Grenzen, sodass ich es nach einer kleinen Reparatur wieder anbringen konnte. Ich lernte aus dieser Situation und schraubte das Dach mit Schrauben an dem Unterbau an.

---

<sup>10</sup> Mehr dazu in dem Blogbeitrag vom 15.05.2018.

Im Winter unterschätzte ich die Last des Schnees, der auf dem Dach lag, sodass das Dach in der Mitte einbrach. Daher baute ich im Frühling ein neues, stabileres Dach.



Abbildung 8: Stabileres Dach

Das neu gebaute Dach ist zwar schwarz und nicht durchsichtig wie das Alte, aber hat dadurch den Vorteil, dass sich das Gewächshaus schneller aufwärmt.

### 2.3 Die Programmierung

Für mich wurde schnell klar, dass ich die Programmiersprache Python für mein Projekt einsetzen würde, denn diese bringt sehr viele Werkzeuge mit und arbeitet eng mit dem Raspberry Pi zusammen. Python ist eine interpretierte höhere Programmiersprache und noch verhältnismäßig jung. Sie spaltet sich in die Versionen Python 2 und Python 3 auf. Diese sind teilweise inkompatibel zueinander. Ich benötige beide Versionen, da einige Werkzeuge wie z.B. der Temperatur- und der Bodenfeuchtigkeitssensor nur mit Python 2 angesteuert werden können, aber für MQTT<sup>11</sup> Python 3 verwendet werden muss.

Insgesamt umfasst das gesamte, aktive Projekt der Gewächshaussteuerung über 2.240 Zeilen Programmcode. Um Erfahrung mit der Programmierung zu gewinnen,

---

<sup>11</sup> Mehr dazu unter „Kommunikation mit dem Handy“

habe ich jedoch insgesamt wesentlich mehr Programmcode entwickelt. Anfangs testete ich die Sensoren<sup>12</sup> wie den Temperatursensor, den Luftfeuchtigkeitssensor, den Bodenfeuchtigkeitssensor und noch einige mehr. Anschließend baute ich eine Datenstruktur auf, die ich Anfang 2019 komplett überarbeitet. In der neuen Version der Datenstruktur werden die Daten, die über die Sprache SQL angesprochen werden können<sup>13</sup>, nicht in Dateien gespeichert, sondern in einer Datenbank.

### 2.3.1 Kommunikation mit dem Handy

Der Computer des Gewächshauses kann über das Internet of Things (IoT) Protokoll MQTT kleine Datenmengen in unserem Heimnetzwerk verschicken und empfangen. Dazu wird ein Server (auch Broker genannt) benötigt, an den die Daten gesendet und von ihm abgefragt werden können. In unserem Heimnetzwerk ist ein eigener passwortgeschützter Broker unter anderem für die Gewächshaussteuerung aufgesetzt worden.

Die Daten werden an verschiedene, sogenannte Topics des Brokers gesendet. Zum Beispiel heißt der Topic, an den die Temperaturdaten geschickt und ausgelesen werden können, „automated/temperature“. Dieser Topic kann auf dem Handy über verschiedene Apps abgefragt werden. Die Apps, die ich verwendet habe, heißen *IoT MQTT Panel* und *MQTT Dash*. Diese können Daten vom Broker empfangen aber auch Daten an diesen senden. Auf diese Weise war ich in der Lage, ohne weitere App-Entwicklung eine Kommunikation zwischen dem Handy und dem Gewächshaus herzustellen.

Die Softwarebibliothek, die ich für das Gewächshaus verwende und über die auf den MQTT-Broker zugegriffen werden kann, heißt *paho* und arbeitet mit Python 3. Das Empfangen von Daten ist nicht so einfach, wie das Senden, da im Hintergrund immer ein Python-Programm laufen muss, um die ankommenden Nachrichten schnell und zuverlässig verarbeiten zu können. Dieses Programm wird bei jedem Hochfahren des Raspberry Pi's automatisch durch das Tool *crontab* (mehr unter „Stabile Software“) gestartet.

Nachteile der Vorgehensweise einer indirekten Kommunikation über den MQTT-Broker ist, dass wenn dieser ausfällt das Gewächshaus keine Daten senden und auch keine empfangen kann. Dadurch kann ich auch nicht mehr manuell die Bewässerung oder die Frontklappe bedienen und auch keine aktuellen Daten mehr empfangen. Diese Situation ist aber bislang nur ein mal aufgetreten.

Neben MQTT schickt ein weiteres Programm alle 10 Minuten Daten der Temperatur, der Bodenfeuchtigkeit und der Luftfeuchtigkeit an meine Internetseite zweite

---

12 Mehr zu den Sensoren unter „Die Sensoren“

13 Unter „Module“ findet sich mehr zum Modul, dass die anfallenden Daten in die Datenbank schreibt.

Internetseite<sup>14</sup>. Leider konnte ich diese Daten nicht direkt auf meine Projektseite hochladen lassen, da eine Sicherheitseinstellung des Moduls `wordpress_xmlrpc` in Python die Ansteuerung verweigerte.

### 2.3.2 Stabile Software

Es ist wichtig eine stabile Softwarearchitektur aufzubauen. Die Idee dahinter ist, dass wenn ein Fehler auftaucht nicht der gesamte Ablauf unterbrochen wird, sondern die Steuerung robust weiterläuft. Dazu benutze ich das zu Linux gehörende Tool `crontab`. Mit Hilfe des Tools können Programme zeitgesteuert und automatisiert gestartet werden. Es kann nicht nur Shellprogramme beim Hochfahren des Rechners, sondern auch nach einem vorgegebenen Rhythmus starten.

Ich habe `crontab` so eingerichtet, dass es alle 10 Minuten die Programme der Sensoren startet, die Daten auf die Internetseite überträgt und auch Daten per MQTT versenden kann. Beim Starten des Raspberry Pi wird das Programm zum Empfangen von MQTT-Daten gestartet. Falls es einen Fehler beim Starten eines Programms gibt, wird dies in einer Logdatei<sup>15</sup> festgehalten. Jedoch wirkt sich dieser Fehler nicht auf den weiteren Verlauf von `crontab` aus. `Crontab` kann unter Linux über den Befehl `crontab -e` eingerichtet werden.

Ein Beispiel für einen Eintrag in der Crontab-Datei:

```
# FKN 13.02.2019
#Temperatursensor
*/10 * * * * /home/pi/automated2/shellscripts/writeTemp.sh >>
/home/pi/automated2/shellscripts/writeTempCron.log 2>&1
(...)
```

Mit diesem Eintrag wird alle 10 Minuten das Shellscript `writeTemp.sh` im Verzeichnis `/home/pi/automated2/shellscripts/` ausgeführt und die Logausgabe (mehr unter: „Das Modul Logging“) in die Datei `writeTempCron.log` im selben Verzeichnis umgeleitet.

### 2.3.3 Fehler erkennen und finden

Fehler erkennen und finden ist ein wichtiger Teil des Programmierens, da kein Programm von Anfang an fehlerfrei ist. Ich gebe daher in jedem Hauptprogramm eine Log-Ausgabe aus, um festzuhalten, wann es gestartet ist, welche Ergebnisse heraus gekommen sind und ggf. welcher Fehler aufgetreten ist. Diese Informationen werden in Log-Dateien gespeichert. Diese Fehler kann man durch die Endungen `.log` oder `.logdatei` erkennen. Diese Logdateien können mit einem einfachen Texteditor eingesehen werden. Nach einiger Zeit können die Logdateien allerdings sehr groß werden. Daher sollten sie nach jedem Monat zurückgesetzt werden.

<sup>14</sup> Vgl. Quellenangaben Inspiration > F.Noweck *zweite Internetseite* > `autom@ted live`

<sup>15</sup> Mehr dazu unter „Fehler erkennen und finden“

Es gibt Systeme, die Log-Dateien überwachen und automatisch Fehlermeldungen identifizieren können. Aus Zeitgründen konnte ich mich damit jedoch noch nicht intensiver auseinandersetzen.

Fehlermeldungen können auch per E-Mail geschickt werden. Daher habe ich das Modul logging geschrieben.

### 2.3.4 User Interface [UI]

Anfangs entwickelte ich für die Steuerung auch ein Display mit Buttons, das aktuelle Werte und weiteres anzeigt. Dies war gegen Anfang des Jahres fertig, doch ich schaffte es bisher nicht das Display und die Button draußen in das Gewächshaus einzubauen. Mit diesem Display könnte schneller manuell auf Daten zugegriffen werden. Jedoch baute ich das Display in das kleinere Bewässerungssystem<sup>16</sup> ein.

### 2.3.5 Die Konfigurationsdatei

Die Konfigurationsdatei ist eine sehr wichtige und zentrale Datei in meiner Programmarchitektur. Jedes Programm meiner Gewächshaussteuerung greift direkt oder indirekt auf diese Datei zu, da in dieser Datei alle möglichen wichtigen Werte konfiguriert sind. Sie ist nicht nur sehr umfangreich, sondern auch sehr praktisch, denn so kann man schnell und einfach zum Beispiel: Ausgänge am Raspberry Pi, die Anzahl der Bodenfeuchtigkeitssensoren, die minimale Bodenfeuchtigkeit und noch vieles mehr einstellen. Doch auch Informationen wie die E-Mailadressen der Empfänger des Wochenrückblickes und auch der HTML-Code der E-Mail sind dort zu finden. Die Konfigurationsdatei ist so gebaut, dass sie sowohl von Python 2 als auch von Python 3 benutzt werden kann.

### 2.3.6 Die Softwarearchitektur

Die Softwarearchitektur ist für den Software-Architekten ein zentraler Plan über den Aufbau des Softwareprojekts und hilft dem Programmierer zu verstehen, wie das Gesamtsystem aufgebaut werden soll. In der unten stehenden Grafik ist die Softwarearchitektur meiner Gewächshaussteuerung aufgezeichnet. Es ist dargestellt, welche Programme grundsätzlich existieren, welche Aufgaben sie haben und wie sie miteinander kommunizieren. Ein derartiger Plan ist entscheidend, um den Überblick zu behalten und mögliche Fehlerquellen eingrenzen zu können. Daher erstellt man so einen Plan am Anfang eines jeden größeren Softwareprojekts.

Es gibt unterschiedliche Gruppen von Programmen. Es gibt Hauptprogramme<sup>17</sup>, die ausgeführt werden und welche, die als Module<sup>18</sup> benutzt werden. Außerdem kann

---

<sup>16</sup> Mehr zum „kleineren Bewässerungssystem“ findet sich unter „*Kleinere Version meines Gewächshaus-systems*“

<sup>17</sup> Mehr dazu unter „*Hauptprogramme*“

<sup>18</sup> Mehr dazu unter „*Module*“

man die Hauptprogramme in weitere Kategorien einteilen. Die Hauptprogramme, die die Sensoren über Module abfragen und die ermittelten Werte über das Modul Datenbankverwaltungssystem<sup>19</sup> in der Datenbank speichern.

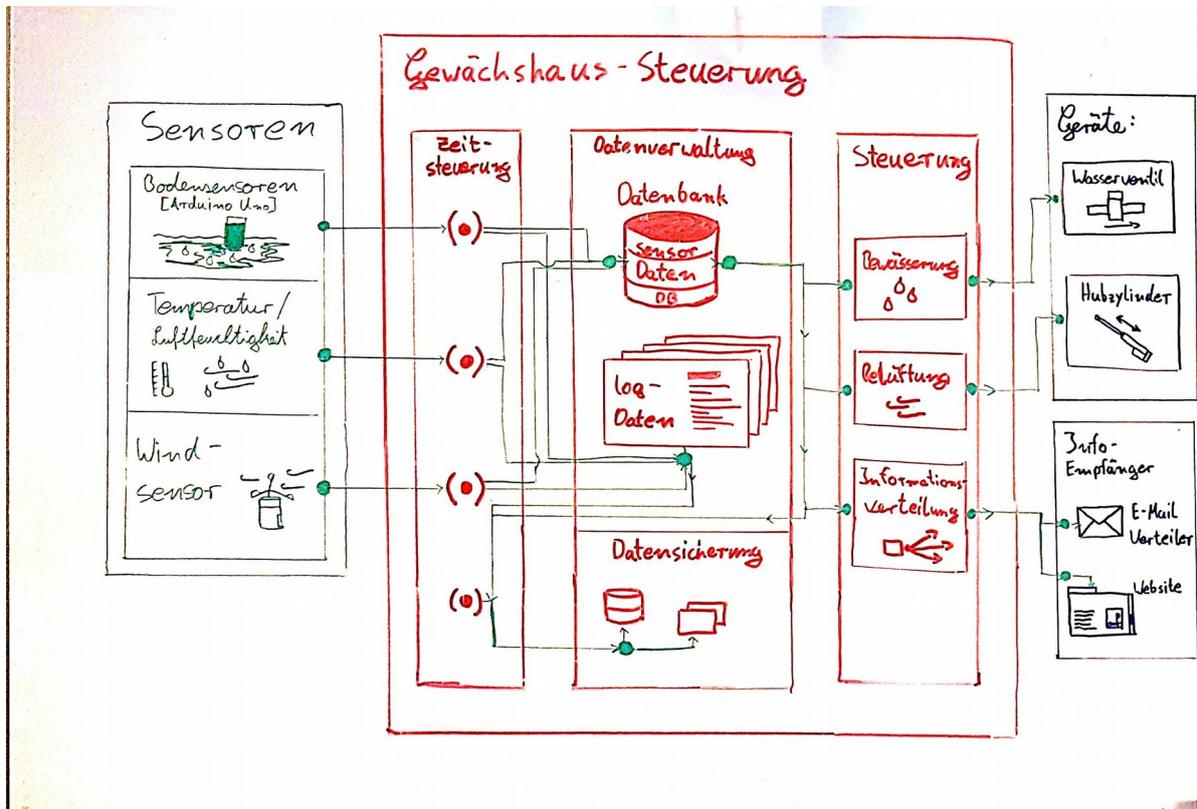


Abbildung 9: Die Datenstruktur meines Projektes

### 2.3.6.1 Hauptprogramme

Hauptprogramme werden direkt von dem Programm crontab<sup>20</sup> gestartet. Sie steuern und koordinieren alles mögliche. „Ein (...) Hauptprogramm ist (...) ein Mensch, der die Werkzeuge zu einem bestimmten Arbeitseinsatz benutzt.“<sup>21</sup>

### 2.3.6.2 Module

„Man kann sich Module wie Werkzeuge vorstellen, die zu einem bestimmten Zweck erstellt sind und alle etwas anderes können. Wie ein Schraubenzieher, der schrauben aber nicht sägen kann. Zum Sägen benötigt man dann ein anderes Modul. Darüber hinaus werden Hauptprogramme benötigt, die die Module kombinieren. Ein derartiges Hauptprogramm ist, um im obigen Vergleich zu bleiben, ein Mensch, der die Werkzeuge zu einem bestimmten Arbeitseinsatz benutzt.“<sup>22</sup>

<sup>19</sup> Mehr dazu unter „Module“

<sup>20</sup> Mehr zu crontab unter „Sichere Software“

<sup>21</sup> Zitiert aus dem Blogbeitrag vom 24.06.2018

<sup>22</sup> Zitiert aus dem Blogbeitrag vom 24.06.2018

Ein zentrales Modul ist das Datenbankverwaltungssystem, welches ich selber geschrieben habe. Anfangs habe ich mit diesem Modul die gemessenen Daten in einfache Textdateien geschrieben. Anfang des Jahres, als ich die Datenstruktur überarbeitete<sup>23</sup>, stellte ich das System auf eine Datenbank um. Man kann sagen, dass es jetzt Python Befehle in SQL-Befehle umwandelt. Dabei kann es in die Datenbank schreiben wie auch lesen. Natürlich gibt es für Python auch derartige Module im Internet zu finden, doch ich wollte ein eigenes entwerfen. Die erzeugten SQL-Befehle werden in der Log-Datei<sup>24</sup> gespeichert.

### 2.3.7 Das Modul Logging

Das Modul Logging ist für alle Ausgaben der Programme zuständig. Alle Programme benutzen es. Das Modul kann über mehrere Wege Ausgaben machen: Einmal ist es in der Lage einfache schriftliche Ausgaben auszugeben, die in Logdateien umgeleitet werden können. Es kann aber auch, z. B. bei gravierenden Fehlern den in der Konfigurationsdatei konfigurierten Benutzer per E-Mail informieren. Außerdem ist es auch für das Versenden des Wochenrückblicks<sup>25</sup> zuständig. Das Logging ist so aufgebaut, dass es von Python 2 wie auch von Python 3 benutzt werden kann.

## 2.4 Die Sensoren

Gegen Anfang des Projektes stellte ich mir die Frage, welche Sensoren ich in das Gewächshaus einbauen könnte. Neben dem Bodenfeuchtigkeitssensor und dem Temperatur - und Luftfeuchtigkeitssensor, die ich als fertige Bauteile einkaufte, konstruierte ich einen Windsensor und einen Regensensor selbst. Der Regensensor ist bei der Gewächshaussteuerung bislang nicht im Einsatz, da der praktische Nutzen zur Steuerung relativ gering ist.

Mein erster Sensor, den ich testete, war der DHT22, der Temperatur und Luftfeuchtigkeit messen kann. In der erste Messreihe zeigte sich dieser als recht zuverlässig. Ich wählte den DHT22 auch aus, weil er vom Raspberry Pi angesteuert werden kann. Die Temperaturdaten dieses Sensors sind die Grundlage für die Belüftung.

---

<sup>23</sup> Mehr dazu im Blogbeitrag vom 06. Januar 2019

<sup>24</sup> Mehr zum Logging findet sich unter „Das Modul Logging“

<sup>25</sup> Wochenrückblick



*Abbildung 10: Der Sensor zum Messen der Temperatur und der Luftfeuchtigkeit*

Bei den Bodenfeuchtigkeitssensoren war problematisch, dass er die Daten als Analoges Signal ausgab und daher nicht direkt mit dem Raspberry Pi ausgewertet werden konnte. Deshalb testete ich zuerst einen Signalwandler (MCP3008), der das Signal des Sensors in eine für den Raspberry Pi lesbare Form umwandeln konnte. Doch dieser zeigte keine zuverlässige Datenermittlung. Das Problem umging ich, indem ich den Raspberry PI mit dem Mikrocontroller Arduino Uno kombinierte. Da der Arduino Uno mit einer anderen Programmiersprache programmiert wird (eine C++ Variante), musste ich einen Weg finden, wie dieser mit dem Raspberry Pi Daten austauschen konnte. Nach einer längeren Recherche fand ich heraus, dass jemand diese Herausforderung bereits gelöst und eine entsprechende Library verwendet hatte (StandardFirmata). Diese konnte ich auf den Arduino Uno hochladen und über ein Skript auf dem Raspberry Pi die an den Arduino Uno angeschlossenen Sensoren abfragen.

Die nächste Herausforderung trat an den Sensoren selbst auf: Nach einer längeren Messreihe fingen diese an zu korrodieren, so dass ich mich kurzer Hand dazu entschied ein robusteres Sensormodell zu verwenden.



*Abbildung 11: Der Sensor zum Messen der Bodenfeuchtigkeit*

Anfangs wollte ich auch einen Windsensor integrieren und suchte nach einem entsprechenden Modell im Internet. Doch alle Sensoren, die ich fand, waren nicht mit dem Raspberry Pi kompatibel und auch sehr teuer. Daher überlegte ich, ob ich nicht selber einen bauen könnte. Diesen konstruierte ich dann schließlich aus einer Flasche, zwei Tischtennisbällen, drei Holzspießen, einer Schraube, einem Magnetsensor, einem Magneten und einem Inlineskate-Rad. Den Magneten befestigte ich an der Schraube in der Mitte, die mit den halben Tischtennisbällen durch den Wind angetrieben wird. Der Magnetsensor ist innen an der halben Flasche angebracht und zählt die Umdrehungen.



*Abbildung 12: Das Anemometer zum Einschätzen des Windes*

Das Modul, das den Windsensor abfragt, ist recht einfach gehalten. In einer in der Konfigurationsdatei festgelegten Zeit wird gemessen, wie oft sich der Magnet an dem Magnetsensor vorbei bewegt. Dies wird zweimal durchgeführt und dann der Durchschnitt gebildet. Diese Messung wird nicht wie bei den anderen Sensoren alle zehn Minuten durchgeführt, sondern alle drei Minuten. Doch es werden die Daten erst dann gespeichert, wenn der ermittelte Wert über einem konfigurierten Schwellenwert liegt. Somit wird verhindert, dass die Datenmenge zu groß wird. Die Daten des Windsensors (Anemometer) werden jedoch nicht nur einfach gespeichert, sondern auch ausgewertet, denn diese Werte sind für die Steuerung die Frontklappe erforderlich. Die Gefahr besteht nämlich, dass bei zu starkem Wind das Gewächshaus Schaden nimmt. Daher wird die Frontklappe bei zu starkem Wind geschlossen.

Der Regensensor besteht aus einer rostfreien Platine und einer Messeinheit. Auf der rostfreien Platine gibt es zwei normalerweise unabhängige Stromkreise, die durch Wassertropfen zusammen geschaltet werden. Die Messplatine misst den Widerstand und sendet diese Daten dann an den Arduino Uno. Auf diese Weise kann ermittelt werden, ob es trocken oder feucht ist.

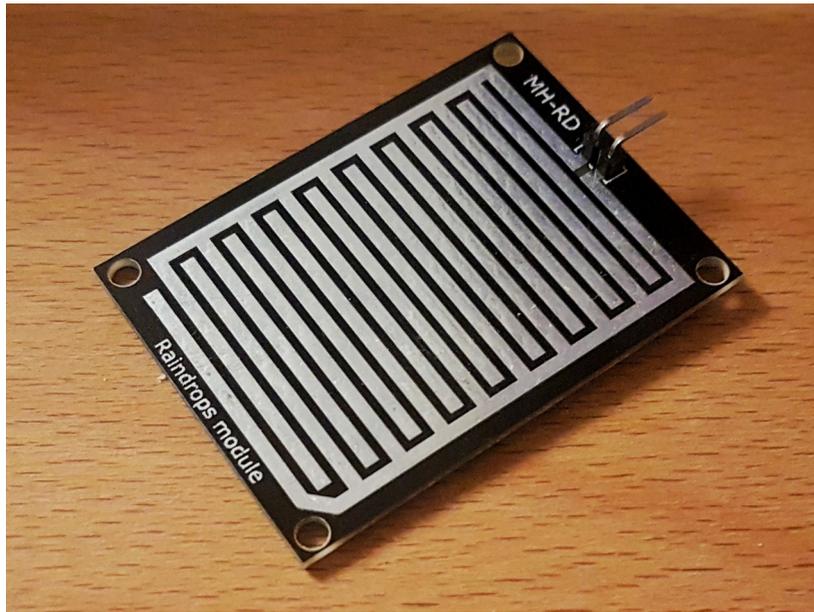


Abbildung 13: Der Regensensor

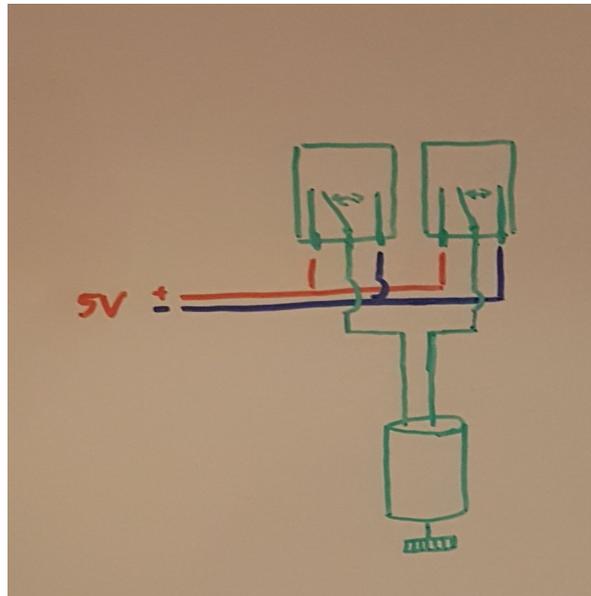
## 2.5 Der Hubzylinder

Mit Hilfe des Hubzylinders kann die Frontklappe des Gewächshauses geöffnet und geschlossen werden. Damit kann die Temperatur und die Belüftung für das Gewächshaus gesteuert werden. Der Hubzylinder wird mit 12 Volt und 3 Ampere betrieben und kann mit 153 kg Hubkraft die Frontklappe hochheben. Die mitgelieferte Steuereinheit kann nur über Funk bedient werden. Damit auch der Raspberry Pi die Steuereinheit und somit den Hubarm bedienen kann, musste ich ein Funkmodul an den Raspberry Pi anschließen und das Funksignal der mitgelieferten Fernbedienung imitieren. Dazu musste ich zunächst die Funksignale auslesen und die gemessenen Werte über das Funkmodul testen. Nachdem dies einwandfrei funktioniert hatte, war ich in der Lage, über ein Programm den Hubzylinder zu steuern. Der Weg zur Steuerung des Hubzylinders per Handy mittels MQTT war dann nur noch ein kleiner Schritt. Die mitgelieferte Fernbedienung für den Hubzylinder kann dann eingesetzt werden, wenn eine Fehlersituation auftritt und die Gewächshaussteuerung nicht funktionieren sollte.

## 2.6 Elektronik

Für die Realisierung des autonomen Gewächshauses benötigte ich grundlegende Kenntnisse im Bereich Elektronik, die ich mir über Internet-Artikel aneignete. Um die platzraubenden Steckbretter zu ersetzen, kaufte ich Platinen, Lötzubehör und ein Lötgerät und arbeitete mich in das Löten ein. Ich entwarf und lötete einige Platinen.

Eine Platine sollte den Hubzylinder steuern. Die Aufgabe der Platine sollte es sein, den Stromfluss umzupolen, da auf diese Weise die Richtung des Hubzylinders (auf bzw. ab) gesteuert werden sollte. Dazu baute ich in die Platine zwei Relais ein. Diese können unabhängig voneinander geschaltet werden und auf diese Weise die Umpolung vornehmen. Die entworfene Platine musste ich jedoch nicht einsetzen, da es mir gelungen war, die Funksteuerung des Hubzylinders zu imitieren. Die grundsätzliche Funktionsweise der Platine konnte ich jedoch erfolgreich testen.



*Abbildung 14: Schaltplan für die Steuerung des Hubzylinders*

Eine andere Platine hatte ich für die Bodenfeuchtigkeitssensoren gedacht und sollte einige Kabel einsparen. Jedoch wurde sie überflüssig, als ich auf rostfreie Sensoren umstellte.

### **2.7 Mein Projekt-Blog im Internet**

Ich bekam die Möglichkeit eine eigene Webseite aufzubauen. Dabei lernte ich das Gestalten von Webseiten und die gesetzlichen Bestimmungen einer eigenen Webseite kennen.

Die Domain und die Internetseite sponserte mir Tobias Fritz von der Firma e-colori. Er half mir auch beim Aufbau des Blocks im Internet.

## 2.8 Die Kooperation mit dem Wissenschaftsladen Bonn

Im Herbst 2018 stieß ich auf eine Mitarbeiterin des Wissenschaftsladens Bonn<sup>26</sup>, die von meinem Projekt sehr angetan war, da der Wissenschaftsladen gerade im Bereich der Stadtbegrünung tätig war. Ich wurde kurz darauf zu einem ersten Treffen eingeladen und bekam den Auftrag ein ähnliches Bewässerungssystem für ein Gewächshaus in meiner Schule zu bauen.

## 2.9 Kleinere Version meines Gewächshaussystems

Durch den Wissenschaftsladen lernte ich den Ingenieur Gerhard Reisinger aus München kennen. In seinem Auftrag entwarf ich eine Minimalversion meines Systems. Dadurch, dass ich nur den Einplatinencomputer Arduino Uno verwendete, ist diese Version kleiner, stromsparender und auch billiger. Jedoch speichert diese Version keine Wetterdaten, kann nicht ins Internet und dadurch auch keine E-Mails verschicken. Die erhobenen Daten können nicht auf eine Website hochgeladen und das System kann auch nicht mit dem Handy bedient werden.

Diese Version verfügt über einen Bodenfeuchtigkeitssensor, ein Füllstandssensor für die Wasserhöhe<sup>27</sup>, einen Transistor, um die Bewässerung zu steuern, ein Display und zwei Buttons. Der Arduino Uno wird mit 12 Volt betrieben. Daher kann an den Transistor zum Beispiel eine Pumpe oder ein Wasserventil mit 12 Volt angeschlossen werden, um die Pflanzen zu bewässern.

Über das Display kann die aktuelle Feuchtigkeit, der Wasserstand im Tank und die Laufzeit des Programms angezeigt werden. Mit den zwei Buttons kann das Menü bedient oder eine manuelle Bewässerung vorgenommen werden. Um möglichst viel Strom zu sparen, fällt das Display, wenn es eine Zeit lang nicht benötigt wurde, in den Schlafmodus. Auch kann durch den Bearbeitungsmodus die minimale Bodenfeuchtigkeit, die aktive Displayzeit nach Benutzung, die Sensorhöhe, die Tanktiefe und die maximale wie auch minimale Wasserhöhe während des Programmlaufs verändert werden.

Dieses System soll in Entwicklungsländern wie Nigeria eingesetzt werden, um die Agrarwirtschaft dort zu verbessern.<sup>28</sup>

---

26 Ein Projektnachweis liegt bei

27 Zum Beispiel für einen Tank als Trockenlaufschutz, wenn eine Pumpe verwendet wird.

28 Eine für das System angefertigte Anleitung liegt bei.

### 3 Schlussteil und Fazit

Durch mein frühes Beginnen konnte ich von Anfang April 2018 bis Ende Juni 2019 einiges erreichen. Durch Blogeinträge, die anfangs wöchentlich und später nach größeren Entwicklungsschritten erschienen, konnte ich meine Arbeit gut und lückenlos dokumentieren. Die Ziele meiner Jahresarbeit habe ich erreicht, aber könnte mir noch einige Erweiterungen vorstellen, wie das User Interface<sup>29</sup> oder einen Regensensor.

Jedoch ist mir auch bewusst geworden, dass ein Gewächshaus unter wirtschaftlichen Aspekten nicht vollständig automatisierbar ist, da weiterhin die Pflanzen gepflegt werden und die Tomaten geerntet werden müssen. Aber es erleichtert den Alltag und funktioniert rund um die Uhr, so dass man auch ohne Sorgen in den Urlaub fahren kann. Zudem wird aufgrund der tatsächlichen Bodenfeuchtigkeit bewässert und nicht durch eine Zeitschaltuhr oder den oberflächlichen Blick eines Menschen.

Nachdem ich mit meiner Jahresarbeit begonnen hatte, fielen mir viele Sachen ein, die für ein Gewächshaus nützlich wären. Doch nach einer Weile musste ich feststellen, dass ich dafür nicht genug Zeit hatte. Außerdem brauchte ich sehr lange, um die Sensoren zu testen und fing mit der richtigen Programmierung und dem Entwerfen einer Datenstruktur erst im Herbst 2018 an.

Die mehr oder weniger zufällig aufgebauten Kontakte zum Wissenschaftsladen konnten meine Arbeit bereichern. Ich habe viele Erfahrungen aus den Kontakten mitgenommen und fühlte mich von den Projektmitgliedern erst genommen und respektiert. Dies war für mich eine schöne und auch für die Zukunft wertvolle Erfahrung.

### 4 Erklärung

Ich versichere hiermit, dass ich die Arbeit selbstständig angefertigt und keine weiteren als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die im Wortlaut oder im wesentlichen Inhalt anderen Werken entnommen wurden, habe ich mit genauer Angabe der Quellen kenntlich gemacht.

Ort, Datum \_\_\_\_\_

Unterschrift \_\_\_\_\_

---

<sup>29</sup> Mehr über das „User Interface“ unter „User Interface [UI]“

## 5 Quellenangaben

### 5.1 Pflanzen

- Autor: Dr. Helmut Tischner  
Titel: Tomaten - Krankheiten, Schädlinge und physiologische Störungen  
Informiert am: 12.08.2018  
Link: <https://www.lfl.bayern.de/ips/kleingarten/019401/index.php>
- Autor: Christel Rupp  
Titel: Feldsalat anbauen und ernten  
Informiert am: 13.10.2018  
Link: <https://www.mein-schoener-garten.de/gartenpraxis/nutzgaerten/feldsalat-anbauen-und-ernten-3166>
- Autor: Einfach genial  
Titel: Das größte Tomatengewächshaus Deutschlands  
Informiert am: 25.09.2018  
Link: <https://www.mdr.de/einfach-genial/eg-deutschlands-groesstes-tomaten-gewaechshaus-wittenberg-100.html>

### 5.2 Inspiration

- Autor: F.Noweck  
Titel: Mein Blog im Internet  
Link: <https://automated.noweck.info>  
  
Titel: Meine zweite Internetseite  
Link: <https://friedjofnoweck.com>
- Autor: Der Kanal Einfach genial  
Titel: Smartes Gewächshaus | Einfach genial | MDR  
Informiert am: 17.10.2018  
Link: <https://www.youtube.com/watch?v=9jl6V-1fzwE>
- Autor: Der Kanal Fast Racers  
Titel: Bewässerungssystem selber bauen !!! (DIY)  
Informiert am: 17.10.2018  
Link: <https://www.youtube.com/watch?v=XKS78DqeMWQ>
- Autor: Der Kanal Einfach genial  
Titel: Gewächshaus für den Balkon | Einfach genial | MDR  
Informiert am: 22.04.2018  
Link: <https://www.youtube.com/watch?v=hSCwDGp58Vs>

### 5.3 Hardware

- Autor: Michael Weigend

Titel: Raspberry Pi für Kids 2. Auflage

Herausgabedatum: 2016

Verlag: mitp Verlags GmbH

- Autor: Der Kanal Hardware Junkie

Titel: Automatische Pflanzenbewässerung "OHNE" Arduino oder Programmieren

Informiert am: 18.10.2018

Link: <https://www.youtube.com/watch?v=6cwqPqOyD7c>

- Autor: Der Kanal MarVtec

Titel: Arduino Bewässerungsanlage selber bauen

Informiert am: 18.10.2018

Link: <https://www.youtube.com/watch?v=JrbkBX2V46U>

### 5.4 Software

- Autor: Thomas Theis

Titel: Einsteigen in Python

Herausgabedatum: 2002

Verlag: Rheinwerk Verlag

- Autor: Felix Stern

Titel: Raspberry Pi: Luftfeuchtigkeit und Temperatur messen

Informiert am: 12.04.2018

Link: <https://tutorials-raspberrypi.de/raspberry-pi-luftfeuchtigkeit-temperatur-messen-dht11-dht22/>

- Autor: Felix Stern

Titel: Arduino und Raspberry Pi miteinander kommunizieren lassen

Informiert am: 04.05.2018

Link: <https://tutorials-raspberrypi.de/arduino-raspberry-pi-miteinander-kommunizieren-lassen/>

- Autor: Felix Stern

Titel: Raspberry Pi LCD Display: 16x2 Zeichen anzeigen (HD44780)

Informiert am: 07.06.2018

Link: <https://tutorials-raspberrypi.de/raspberry-pi-lcd-display-16x2-hd44780/>

- Autor: Felix Stern  
Titel: HD44780 LCD Display per I2C mit dem Raspberry Pi ansteuern  
Informiert am: 08.06.2018  
Link: <https://tutorials-raspberrypi.de/hd44780-lcd-display-per-i2c-mit-dem-raspberrypi-ansteuern/>
  - Autor: Hochschule München, FK 04, Prof. Jürgen Plate und weitere Autoren  
Titel: Raspberry Pi: Einfaches Text-LCD mit I2C-Interface  
Informiert am: 09.06.2018  
Link: <http://www.netzmafia.de/skripten/hardware/RasPi/Projekt-LCD/index.html>
  - Autor: Galileo Press 2008, Peter Kaiser & Johannes Ernesti  
Titel: Galileo <openbook>  
Informiert: 2018/19  
Link: <http://gwise.itwelzel.biz/Books/Openbook%20GalileoComputing/python>
- Anmerkung: Dies ist eine sehr umfassende informative Sammlung. Ich habe mich besonders mit Kapitel 19, der Datenbanksprache SQL, befasst.
- Geschäftsführer: Dipl. Inform. Manuel Dietrich  
Titel: Raspberry Pi – WLAN einrichten (Edimax)  
Informiert am: 01.01. - 05.01.2019  
Link: <https://www.datenreise.de/raspberrypi-wlan-einrichten-edimax/>
  - Autor: Roger Light  
Titel: paho-mqtt 1.4.0  
Informiert am: 17.04.2019  
Link: <https://pypi.org/project/paho-mqtt/>
  - Autor: Felix Stern  
Titel: Raspberry Pi Funksteckdosen (433MHz) steuern – Tutorial  
Informiert am: 25.05.2019  
Link: <https://tutorials-raspberrypi.de/raspberrypi-funksteckdosen-433-mhz-steuern/>
  - Autor: Python Software Foundation (US)  
Titel: DB-API 2.0 interface for SQLite databases  
Informiert am: 29.12.2018-13.01.2019  
Link: <https://docs.python.org/2/library/sqlite3.html>
  - Autor: Dr. Eugen Stall  
Titel: WEATHERMAN ... die perfekte Wetterstation für die Hausautomation  
Informiert am: 17.08.2018  
Link: <https://www.stall.biz/project/weatherman-die-perfekte-wetterstation-fuer-die-hausautomation>

6 Weitere Bilder



Abbildung 15: Die jungen Tomatenpflanzen

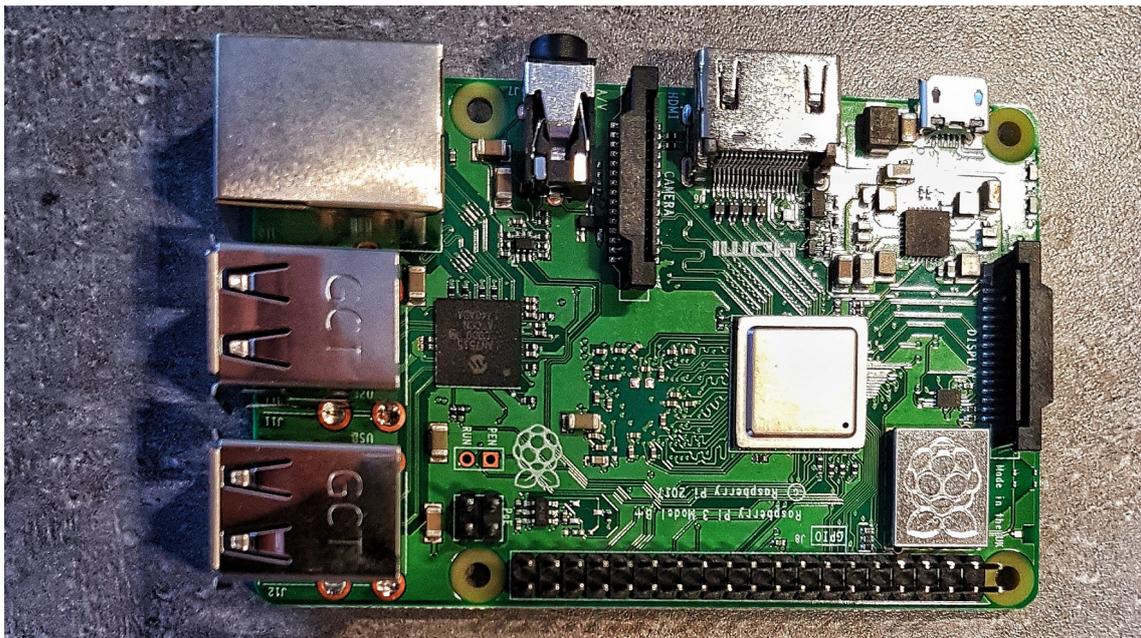


Abbildung 16: Der Computer, der das Gewächshaus steuert



*Abbildung 17: Eine Tomatenrispe aus dem Jahr 2018*



*Abbildung 18: Eingelegte Tomaten*



*Abbildung 19: Ersten Tomaten in 2018*



*Abbildung 20: Rohbau des Gewächshauses*