

Das smarte Feld



Bild vom smarten Feld mit Bodenfeuchtesensor im Vordergrund

Vorgelegt für den Wettbewerb Jugend forscht 2022

Arbeitsfeld: Mathematik/Informatik

von: Tristan Sander

Projektnummer: 92080

Erich Kästner Gymnasium

Kurzfassung des Themas

In Deutschland werden mehr als 50% der Fläche landwirtschaftlich genutzt, um Nahrungsmittel zu erzeugen (Stand 2020 [1]). Die Erzeugung von Lebensmitteln hat folglich einen hohen Stellenwert für die deutsche Wirtschaft.

In diesem Projekt geht es daher darum, mit Hilfe von Sensoren und anderen Bauteilen ein Überwachungssystem für Ökosysteme wie Felder oder Gewächshäuser zu schaffen.

Im Vorfeld des Projekts wurden anhand von Versuchen und Messreihen Werte erhoben, auf deren Basis die Sensoren Abweichungen von der optimalen Versorgung der Pflanzen erkennen und melden können. In diesem Projekt habe ich ein Gerät gebaut und programmiert, das es ermöglicht, diese Daten jederzeit und komfortabel auch aus der Entfernung von mobilen Endgeräten oder Steuerungseinheiten abrufbar zu machen. Dieses System soll dabei helfen, die Landwirtschaft und das Anbauen von Pflanzen im Treibhaus noch effektiver und umweltschonender zu gestalten.

Inhalt

Kurzfassung des Themas	2
Inhalt	2
Einleitung	3
Smarte Landwirtschaft	3
Vision	4
Konstruktion und Programmierung	4
Programmierung der Website (html/php)	9
Konstruktion der Hardware	9
Resultat	10
Reflexion	15
Anhang	16

Einleitung

Als ich auf YouTube unterwegs war, stieß ich auf ein Video des niederländischen YouTube-Kanals von Google (https://www.youtube.com/watch?v=nsPQvZm_rgM). In diesem Video, welches eigentlich als Aprilscherz veröffentlicht worden war, geht es um ein Projekt, das Tulpen helfen soll, mit Menschen zu kommunizieren. Mir war klar, dass dies nicht funktionieren würde - die Idee, dass die Pflanze sagen kann, wann es ihr gut geht und wann nicht, blieb. Da die Pflanze selber nicht sprechen kann, habe ich mir überlegt, eine Pflanze mit Hilfe eines Programms „sprechen“ zu lassen.

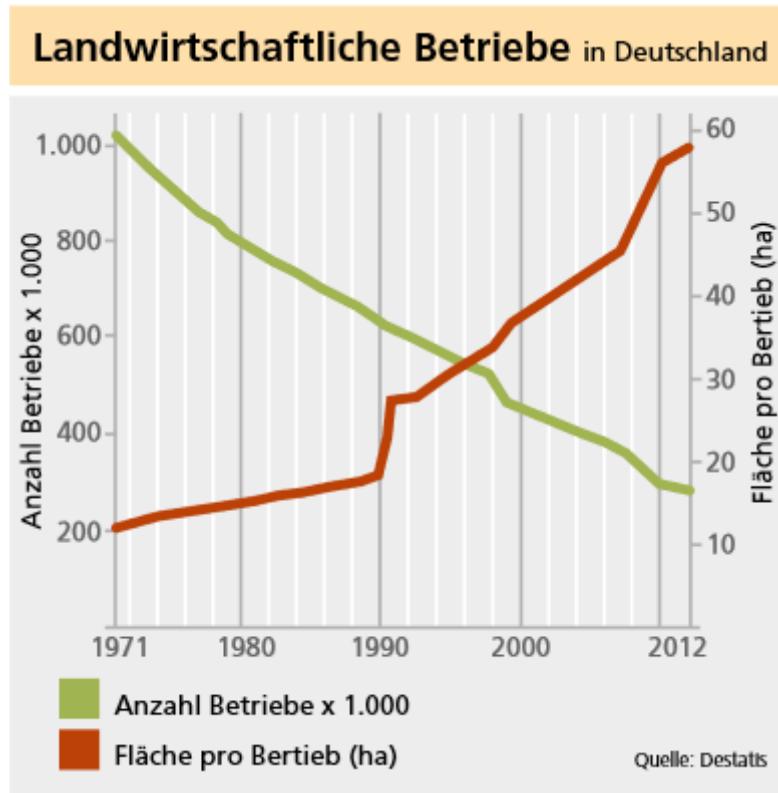
Diese Idee aus dem letzten Jahr habe ich nun weiter entwickelt, vergrößert und verbessert. Nun kann also nicht nur eine einzelne Pflanze beobachtet werden, sondern beispielsweise ein Feld, ein Gewächshaus oder eine Obstbaumplantage. Es gibt bereits zahlreiche Apps und Programme, die landwirtschaftliche Geräte bei der Aussaat und der Ernte überwachen und steuern, aber ich habe keine App gefunden, die die Wachstumsbedingungen überwacht. Dadurch, dass die Sensoren modular verwendbar sind, können sie je nach Bedarf einzeln ausgewählt oder zusammengesteckt werden.

Die Weiterentwicklung meines vorigen Projekts erfolgte, indem ich auf Basis der Programmiersprachen **html** bzw. **php** eine Website programmiert habe, welche ihre Daten von einem Raspberry Pi erhält, auf dem eine modifizierte Variante des Node-red Programms des smarten Topfs läuft. Die Hardware dazu ähnelt der des smarten Topfs, weist dennoch Unterschiede auf (mehr Sensoren, eigenes Gehäuse).

Smarte Landwirtschaft

In den letzten 50 Jahren hat sich die Zahl der Menschen, die ein Bauer ernährt, beinahe verdreifacht [2]. Dabei ist die Zahl der Höfe drastisch gesunken, die Anbauflächen sind jedoch nahezu gleich geblieben. Diese intensive Landwirtschaft hat eine Reihe von Schattenseiten. Die Böden sind an vielen Orten zu stark verdichtet, durch Monokulturen und Pestizide schwindet die Artenvielfalt und der Klimawandel macht Dürren in den Sommermonaten immer wahrscheinlicher. Gleichzeitig wächst die Weltbevölkerung und die Menschen in den Industrieländern werden immer anspruchsvoller, was die Qualität der Lebensmittel angeht. Unter diesen Umständen noch wirtschaftlich Ackerbau zu betreiben, setzt Landwirte unter enormen Druck.

Eine Digitalisierung der Landwirtschaft, das smarte Feld, könnte landwirtschaftlichen Betrieben helfen, sparsam und ökologisch sinnvoll mit den Ressourcen umzugehen und trotzdem keine geringeren Ernten in Kauf nehmen zu müssen.



Vision

Die Idee dieses Projekts besteht darin, eine smarte Steuerungsanlage für ein Ökosystem zu erschaffen. Diese soll mehrere Sensoren, Pflanzen bzw. Felder, Gewächshäuser oder Hochbeete und verschiedene Komponenten (beispielsweise Pumpen oder Lichtinstallationen) miteinander verbinden, so dass sie von einem Punkt koordiniert werden können und alle Daten einsehbar sind. Gleichzeitig sollen dabei auch Empfehlungen zur Optimierung der Versorgung der Pflanzen ausgesprochen werden. Die Steuerung des Systems soll über eine Website gesteuert werden können.

Konstruktion und Programmierung

Node-RED

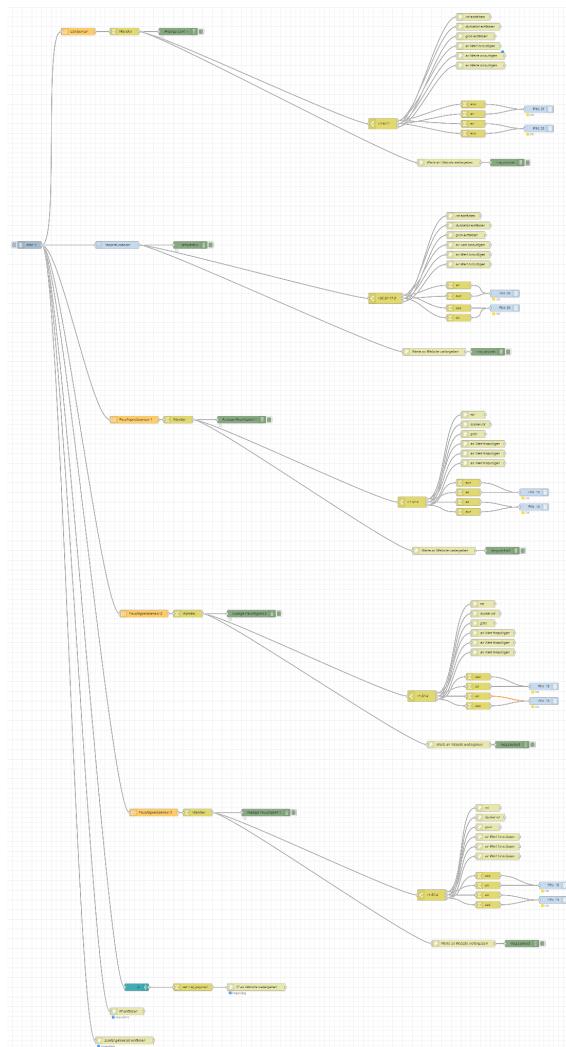
Beim Programmieren der Hardware wurde auf Node-Red gesetzt, da es mit einer einfachen und übersichtlichen Blockprogrammierung überzeugte.

Das Programm in Textform lässt sich unten im Anhang finden. [5]

Begriffserklärung

Die einzelnen Programmteile werden als **Nodes** bezeichnet.

Wenn an manchen Stellen von Tabellen oder Tabellenspalten gesprochen wird, hat dies etwas mit der Programmierung der Website zu tun. In diesen Tabelle werden die Werte zusammen mit einem Namen gespeichert.



Screenshot vom Node-RED Programm (volle Auflösung siehe [5])

Start/inject node:

Diese Node startet das Programm in einem Intervall von 6 Sekunden und sorgt somit dafür, dass die Sensoren Werte messen und weitergeben.

Licht-, Bodenfeuchtesensor/ads1x15 raspi node:

Dieser Block fragt die analogen Sensoren ab (Bodenfeuchte und Lichtintensität).

Temperatursensor/rpi ds18b20 node:

Das oben genannte Node fragt den digitalen Temperatursensor ab.

Wandler/change node:

Dieser wandelt den analogen Wert in einen digitalen um, deswegen wird dieser auch nicht für den Temperatursensor benötigt, da er direkt einen digitalen Wert ausgibt.

Anzeige bspw. Feuchtigkeit 1/ debug node:

Es zeigt zum einen noch mal den Messwert des Sensors an, zum anderen werden von hier aus die Werte von dem Link kopiert und später eingefügt.

>xy/<xy /switch node

Leitet ein Signal weiter, wenn ein bestimmter Zustand zutrifft, aber auch wenn keiner zutrifft, wird ein Signal ausgegeben. Dieses Node beinhaltet ebenfalls die Parameter, die anzeigen, wie es den Pflanzen geht. Hierbei habe ich die alten Ergebnisse des smarten Topfs übernommen. Dies ist eines der wichtigsten Nodes, da dieses alles steuert, was damit zu tun hat, dass sich etwas optisch auf der Website und lokal am Raspberry Pi verändert.→ farbe xy einfärben/ http request (pro/air)

an/aus/ change node

Von diesem Node werden die Lampen angesteuert, dieses Node hingegen wird vom oben genannten switch node gesteuert.

Pin: xy/ gpio out

Wenn dieses Node von einem an/aus change node einen Befehl erhält, geht die Lampe entweder an oder aus. Dies ist theoretisch unnötig, hilft aber beim Testen, wenn die Website noch nicht fertig programmiert ist.

Werte an Website senden/ http request

Dieses Node leitet die Werte an die Website weiter. In ihm steht Folgendes:

`http://tristan.sander.is/kom.php?key=Temperatur&value={{payload}}`

Die markierten Teile haben folgende Funktionen:

- Dieser Teil sagt, welche Website erreicht werden soll und wohin die Daten weitergegeben werden sollen.
- Gibt an, dass nach dem Gleichzeichen der Name des Werts folgt
- Name des Werts
- Gibt an, dass nach dem Gleichzeichen der Wert folgt
- In diesem Teil wird dann der Wert übermittelt, der ausgegeben werden soll. Dieser wird vom msg.payload node an das http request node weitergegeben.

air Werte hinzufügen/http request

Dieses Node muss nach dem switch Node stehen, da jenes für die air Variante keine gemessenen Werte ausgibt, sondern für verschiedene Zustände Text einfügt.

Dies würde im Link beispielsweise so aussehen:

http://tristan.sander.is/kom.php?key=Temperatur_air&value=zu kalt

Die markierten Teile haben folgende Bedeutung:

- Dieser Teil sagt, welche Website erreicht werden soll und wohin die Daten weitergegeben werden sollen.
- Gibt an, dass nach dem Gleichzeichen der Name des Werts folgt
- Name des Werts, der mit _air enden muss
- Gibt an, dass nach dem Gleichzeichen der Wert folgt
- In diesem Teil wird dann der Wert übermittelt, der im Air Modus ausgegeben werden soll.

farbe xy einfärben/ http request

Dieses Node wird, wie die Schalter für die Lampen, vom >xy/<xy /switch node gesteuert.

Wenn dieses Node aktiviert wird, dann wird folgender Link abgerufen:

https://tristan.sander.is/kom.php?key=Temperatur_style&value=color:red

Die markierten Teile haben folgende Funktionen:

- Dieser Teil sagt, welche Website erreicht werden soll und wohin die Daten weitergegeben werden sollen.
- Gibt an, dass nach dem Gleichzeichen der Name des Werts folgt.
- Name des Werts, der mit _style enden muss
- In diesem Teil wird dann der Wert übermittelt, der als Style in HTML verwendet werden soll.

Ip/ip

Dieses Node ermittelt dieIpv4 Adresse.

Ip an Website senden/ ip/zuletzt gelöscht einfärben

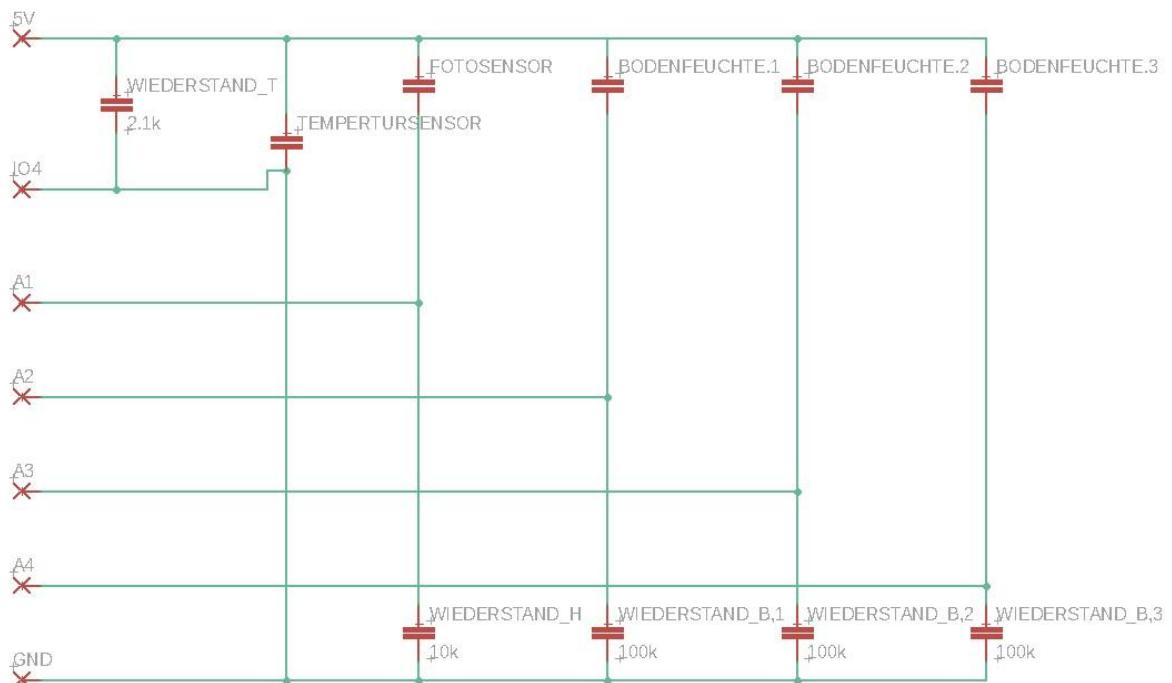
Diese http request Nodes haben gewissermaßen die gleiche Aufgabe, wie die anderen Blöcke, nämlich die Werte an die Website zu senden oder einzufärben. Hier ist also der Wertname beispielsweise Ip.

Programmierung der Website (html/php)

Hier würde eigentlich der Quelltext der Website und eine Erklärung dazu folgen. Da dieser aber zu lang ist und zu einer Überschreitung der maximalen Seitenanzahl dieser Arbeit führen würde, habe ich mich dafür entschieden, ihn erst beim Wettbewerb in der Präsentation zu zeigen und zu erklären. Der Quelltext dazu lässt sich schon im Anhang finden. [5]

Konstruktion der Hardware

Schaltplan



Schaltplan des smarten Felds

Anmerkung: Der Temperatursensor ist eigentlich anders geschaltet. Da aber **Eagle**, das Programm, mit welchem der Schaltplan erstellt wurde, nicht die Möglichkeit bietet, einen

solchen richtig einzulegen, musste er als Kondensator eingezeichnet werden. Die Datei dazu lässt sich im Anhang finden. [5]

Gehäuse

Das Gehäuse des Raspberry Pis und der Sensoren wurde mit Hilfe eines Lasercutters aus einer Acrylglasplatte herausgeschnitten. Anschließend wurden die einzelnen Schichten aufeinander gestapelt. Auf diese Weise entstand ein Gehäuse, in das alle Komponenten perfekt hineinpassen. Die einzelnen Schichten werden mit Gewindestangen in den Ecken zusammengehalten. Die obere Abschlussplatte ist durchsichtig, damit der Lichtsensor funktionieren kann, aber dennoch vor Feuchtigkeit geschützt ist.

Beim Temperatursensor wurde ein Ausschnitt ausgespart und in das entstandene Loch eine Metallplatte gelegt, da Metall gut Wärme leitet und trotzdem dafür sorgt, dass auch dieser Sensor vor Feuchtigkeit geschützt ist. Die Datei dazu lässt sich im Anhang finden. [5]

Verbaute Komponenten

- Raspberry Pi 4 model b
- GPIO extension Board
- Digital-analog Wandler Aufsatz
- Fotowiderstand
- Digitaler Temperatursensor
- 1x 2.1k Ω Widerstand
- 1x 10k Ω Widerstand
- 3x 100k Ω Widerstand
- Kupferplatten

Resultat

Website

Auf der Website <https://tristan.sander.is/kom.php?modus=air> kann der Nutzer nun den Zustand seiner Pflanzen einsehen. Auf der “air” Website wird er informiert, ob die Pflanze eine bestimmte Ressource braucht (beispielsweise ob sie genug Wasser hat), aber auch wenn sie überversorgt ist. Zusätzlich sind die Werte entsprechend dem Zustand eingefärbt. Hierbei steht **rot** für “zu wenig”, **grün** für “alles gut” und **dunkelrot** für einen Überschuss der Ressource.

Wenn der Nutzer mehr Informationen haben möchte, kann er einfach zur “pro” Website wechseln, indem er auf den Button “Pro Modus” drückt. Hier hat er dann Zugriff auf die exakten Daten. Diese passen sich farblich auch, wie im “air” Modus, entsprechend den Zuständen an.

Für den Fall, dass es zu fehlerhaften Anzeigen kommen sollte, gibt es einen weiteren Button: "Bei Anzeige Fehlern drücken". Er sorgt für die Löschung aller bisherigen Daten, wodurch der Fehler behoben werden sollte.

<h3>Das smarte Feld</h3> <p>77.20.124.225</p> <p>Helligkeit alles gut Sektor1 alles gut Sektor2 alles gut Sektor3 alles gut Temperatur alles gut</p> <p><input type="button" value="Pro Modus"/></p> <p><input type="button" value="Bei Anzeige Fehlern Drücken"/></p>	<h3>Das smarte Feld</h3> <p>77.20.124.225</p> <p>Helligkeit 2.4 Sektor1 3.2 Sektor2 3.9 Sektor3 3.4 Temperatur 21.3 lokaleIP 192.168.0.175</p> <p>zuletzt gelesen 2022-01-15T20:17:57+01:00</p> <p><input type="button" value="Air Modus"/></p> <p><input type="button" value="Bei Anzeige Fehlern Drücken"/></p>
<h3>Das smarte Feld</h3> <p>77.20.124.225</p> <p>Helligkeit alles gut Sektor1 alles gut Sektor2 alles gut Sektor3 alles gut Temperatur alles gut</p> <p><input type="button" value="Pro Modus"/></p> <p><input type="button" value="Bei Anzeige Fehlern Drücken"/></p>	<h3>Das smarte Feld</h3> <p>77.20.124.225</p> <p>Helligkeit 1.75 Sektor1 3.2 Sektor2 3.9 Sektor3 3.4 Temperatur 21.3 lokaleIP 192.168.0.175</p> <p>zuletzt gelesen 2022-01-15T20:17:57+01:00</p> <p><input type="button" value="Air Modus"/></p> <p><input type="button" value="Bei Anzeige Fehlern Drücken"/></p>
<h3>Das smarte Feld</h3> <p>77.20.124.225</p> <p>Helligkeit zu viel Licht Sektor1 alles gut Sektor2 alles gut Sektor3 alles gut Temperatur alles gut</p> <p><input type="button" value="Pro Modus"/></p> <p><input type="button" value="Bei Anzeige Fehlern Drücken"/></p>	<h3>Das smarte Feld</h3> <p>77.20.124.225</p> <p>Helligkeit 5.3 Sektor1 3.2 Sektor2 3.9 Sektor3 3.4 Temperatur 21.3 lokaleIP 192.168.0.175</p> <p>zuletzt gelesen 2022-01-15T20:17:57+01:00</p> <p><input type="button" value="Air Modus"/></p> <p><input type="button" value="Bei Anzeige Fehlern Drücken"/></p>

Das smarte Feld

77.20.124.225

Helligkeit alles gut
Sektor1 alles gut
Sektor2 alles gut
Sektor3 alles gut

Temperatur zu warm

Pro Modus

Bei Anzeige Fehlern Drücken

Das smarte Feld

77.20.124.225

Helligkeit 2.4
Sektor1 3.2
Sektor2 3.9
Sektor3 3.4

Temperatur 33.3

lokaleIP 192.168.0.175

zuletzt gelesen 2022-01-15T20:17:57+01:00

Air Modus

Bei Anzeige Fehlern Drücken

Das smarte Feld

77.20.124.225

Helligkeit alles gut
Sektor1 alles gut
Sektor2 alles gut
Sektor3 alles gut

Temperatur zu kalt

Pro Modus

Bei Anzeige Fehlern Drücken

Das smarte Feld

77.20.124.225

Helligkeit 2.4
Sektor1 3.2
Sektor2 3.9
Sektor3 3.4

Temperatur 13.0

lokaleIP 192.168.0.175

zuletzt gelesen 2022-01-15T20:17:57+01:00

Air Modus

Bei Anzeige Fehlern Drücken

Das smarte Feld

77.20.124.225

Helligkeit alles gut
Sektor1 zu viel Wasser
Sektor2 alles gut
Sektor3 alles gut

Temperatur alles gut

Pro Modus

Bei Anzeige Fehlern Drücken

Das smarte Feld

77.20.124.225

Helligkeit 2.4
Sektor1 6.7
Sektor2 3.9
Sektor3 3.4

Temperatur 21.3

lokaleIP 192.168.0.175

zuletzt gelesen 2022-01-15T20:17:57+01:00

Air Modus

Bei Anzeige Fehlern Drücken

Das smarte Feld

77.20.124.225

Helligkeit alles gut
Sektor1 dringend gießen
Sektor2 alles gut
Sektor3 alles gut

Temperatur alles gut

Pro Modus

Bei Anzeige Fehlern Drücken

Das smarte Feld

77.20.124.225

Helligkeit 2.4
Sektor1 6.0
Sektor2 3.9
Sektor3 3.4

Temperatur 21.3

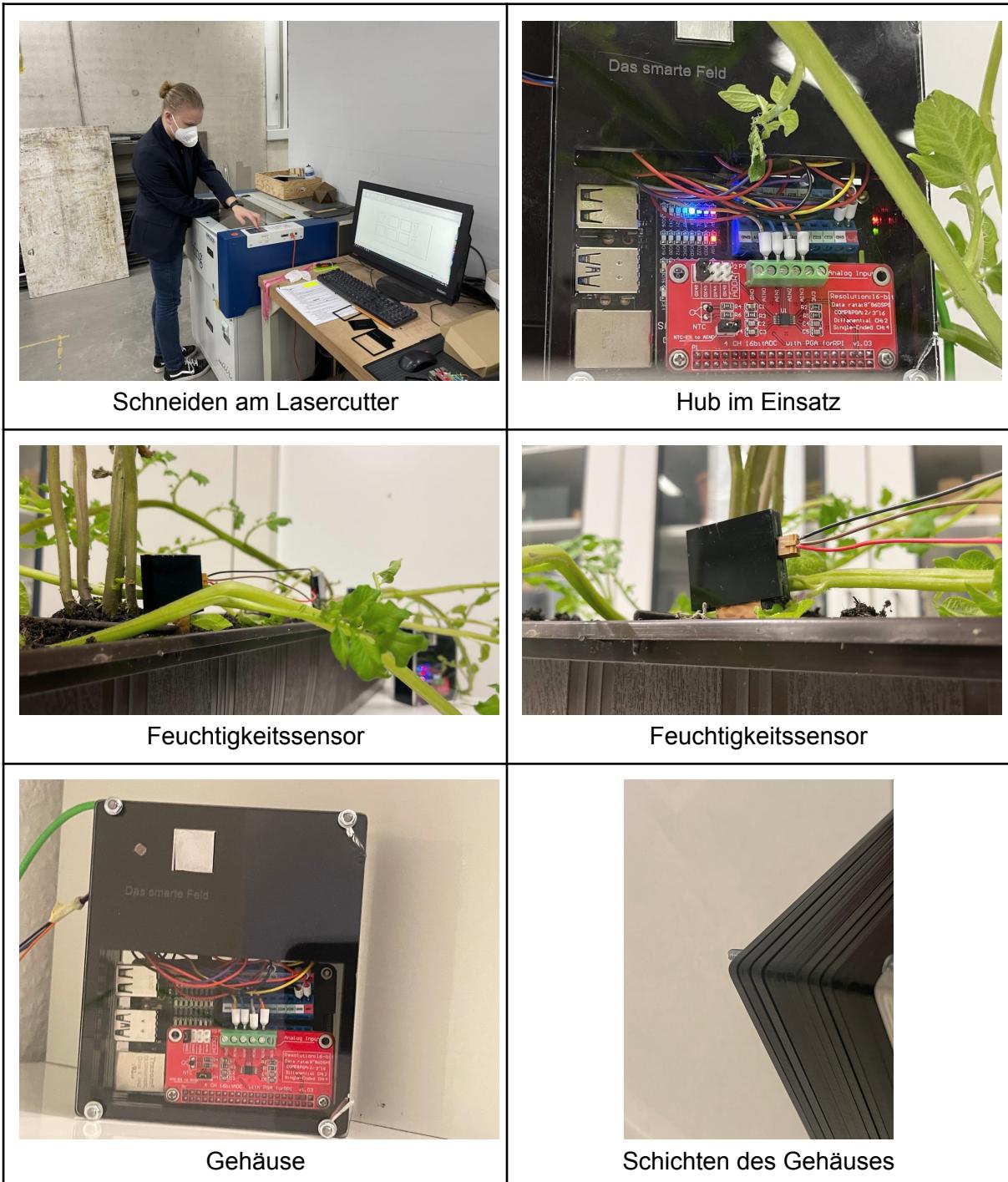
lokaleIP 192.168.0.175

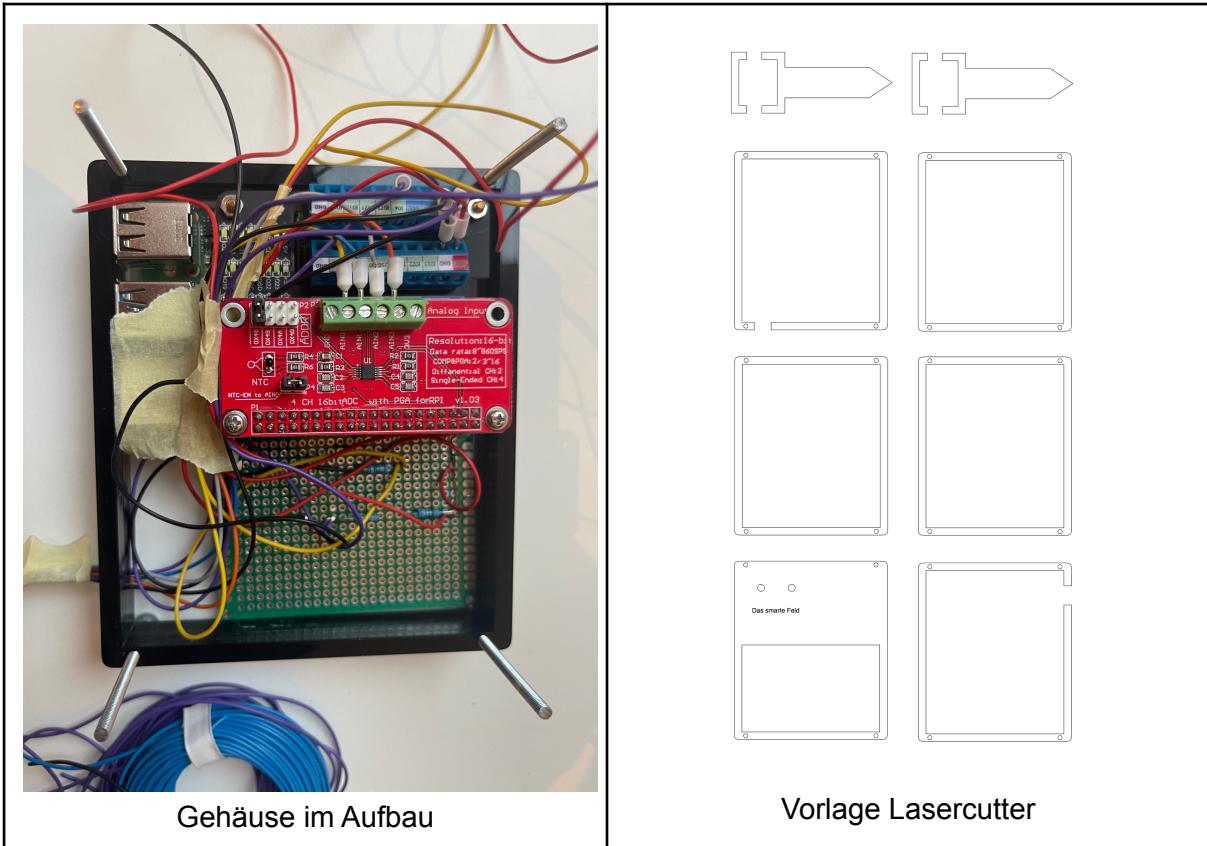
zuletzt gelesen 2022-01-15T20:17:57+01:00

Air Modus

Bei Anzeige Fehlern Drücken

Aussehen der Technik





Einsatzmöglichkeiten und weiterführende Ideen

Durch die Modularität kann der Nutzer entscheiden, welche und wie viele Sensoren verwendet werden sollen. Auf diese Weise kann dieses System an die Bedürfnisse verschiedener Umgebungen angepasst werden.

Bei einem Gewächshaus ergibt es beispielsweise Sinn, Temperatur- und Lichtsensoren zu verwenden, da hier Faktoren wie Helligkeit und Wärme einfacher zu steuern sind. Auf einem Feld hingegen ist es nur schwer möglich, die Temperatur oder Lichtintensität zu steuern, während sich die Wasserzufuhr recht einfach regeln lässt. Deswegen bietet es sich an, dort nur Bodenfeuchtesensoren zu verwenden.

Ein Ziel für das nächste Jahr wäre es, diese Technik so weiterzuentwickeln, dass sie für Obstbäume nutzbar ist. Für einige bei deren Anbau auftretenden Probleme, könnte eine solche Technik hilfreiche Lösungen bieten. Beispielsweise werden die Blütenknospen, um sie vor Frost zu schützen, vor Eintritt der Minusgrade mit Wasser besprüht. Durch das vollständige Einfrieren wird die Knospe vor dem Erfrieren geschützt. [3] Eine Technik, die kontrolliert Wasser direkt an den richtigen Stellen verteilt, könnte helfen. Auch die zur Wasserzufuhr zum Gießen der Bäume genutzten Schlauchanlagen könnten in dieses System integriert werden. Sie würden in der Erde vergraben, da so die Wurzeln das Wasser erreichen können, aber das Wasser nicht an der Erdoberfläche verdunstet. Dies wäre bei einem Feld nicht möglich, da es immer wieder gepflügt werden muss. Die Erde um die Bäume herum muss hingegen nicht bearbeitet werden.

Um beim Obstbau gut die Bodenfeuchte messen zu können, müssten die Bodenfeuchtesensoren überarbeitet werden, da diese in ihrer momentanen Bauweise nicht weit genug in den Boden vorstoßen können.

Eine weitere Idee besteht darin, einen Sauerstoffsensor zu verbauen und einem Programm beizubringen zu lernen, welche Außeneinwirkungen für die Pflanze am besten sind. Somit müssten im vorhinein keine Parameter festgelegt werden, welche Umstände für die Pflanze am vorteilhaftesten sind. Dies müsste aber in einem geschlossenen Raum stattfinden, da sonst der Sauerstoffsensor von außen beeinflusst werden würde.

Reflexion

Es war mein Ziel eine Anlage für ein Ökosystem zu schaffen, die mehrere Sensoren und Pumpen miteinander verbindet. Diese sollten von einer Website aus steuer- und einsehbar sein.

Um dieses Ziel zu erreichen, habe ich eine Website programmiert, welche es ermöglicht, die gemessenen Daten einzusehen. Außerdem gibt es dort nun auch Informationen darüber, was diese Werte bedeuten - ein Problem aus dem letzten Jahr, welches nun behoben ist. Bis zum Wettbewerb sollen noch Pumpen angeschlossen werden, die als Modellversuch zeigen sollen, wie ein solches Bewässerungssystem funktioniert.

Weiterentwicklungen für ein Folgeprojekt wäre, die Sensorik für Obstbäume, die in "Einsatzmöglichkeiten und weiterführende Ideen" beschrieben wurde.

Anhang

Literaturangaben

[1]

<https://www.umweltbundesamt.de/daten/flaeche-boden-land-oekosysteme/flaeche/struktur-der-flaechennutzung#die-wichtigsten-flachennutzungen>

[2]

<https://de.statista.com/statistik/daten/studie/201243/umfrage/anzahl-der-menschen-die-durch-einen-landwirt-ernaehrt-werden/>

[3] <https://zedernhof.de/frost-im-obstanbau/>

[4] [Landwirtschaft im Wandel](#)

[5] <https://github.com/grimlon-cyborg/Mark4>

Unterstützungsleistung

Folgende Personen haben mich bei meinem Projekt unterstützt:

- Frau Hendrika van Waveren und Herr Tobias Peckskamp (Projektbetreuer/in)
- Martin Sander (Programmierer)