Pflanzengießanlage

Hofbauer, Christoph Fischer-Has, Markus Nagl, Matthias Schubäck, Stefan

I. Idee

Auch unter den Studenten gibt es den einen oder anderen mit einem grünen Daumen, der seine WG will durch ein paar Pflanzen aufgewertet hat. Wenn nur nicht das ständige Gießen wäre. Vor jedem längeren Urlaub stellt sich die Frage, was machen mit den ganzen Pflanzen? Den Nachbarn fragen und hoffen er ist da und verlässlich. Die Pflanzen Äbsaufen"lassen und hoffen, dass die sich das was einteilen? Beides keine zufriedenstellende Lösung. Wieso lässt man sich die Pflanze nicht selber gießen?

Aus diesen Überlegungen ist die Idee entstanden, eine Pflanze mit Sensoren auszustatten, die uns über den aktuellen Zustand Informieren. Die wichtigste Information in dieser Hinsicht ist natürlich die Bodenfeuchtigkeit im Topf der Pflanze. Neben dieser können noch weitere Informationsquellen herangezogen werden, die über das Wohlbefinden der Pflanze Aufschluss geben. Hierunter fallen unter anderer der Nährstoffgehalt der Erde, die Temperatur an der Wurzel oder die Luftqualität im Umfeld der Pflanze.

Gießsystem WG

II. Eingrenzung

Im Rahmen des Projekts soll ein Gießsystem für eine alleinstehende Pflanze erstellt werden. Es misst die Bodenfeuchtigkeit der Pflanze gießt und bei bedarf die Pflanze. Das Wasser hierfür wird mithilfe einer Pumpe aus einem bereitgestellten Wassertank entnommen. Das System ist für den Innenraum ausgelegt und soll autonom ohne das Zutun des Menschen die Pflanze gießen. Der notwendige Strom wird über ein Netzteil bereitgestellt. Als Zusatzfeature soll das Gießsystem seine Messwerte Drahtlos an einen Computer übermitteln.

Weitere Ideen wie z.B. eine Autarke Lösung mit Batterien, ein *Mehr-Pflanzenbetrieb* oder ein System für den Garten wurden zwar diskutiert aber wegen dem erhörten Zeitaufwand und begrenzten Budgets nicht weiter verfolgt. Auch wurde auf ein Warnsystem verzichtet, der den Menschen informiert, dass der Wassertank leer wird.

III. Aufbau

Das System lässt sich in drei Bereiche einteilen. Zum ersten der Mechanik mit Hilfe der das Wasser zu der Pflanze kommt. Eine Elektronik die die Energieversorgung regelt und Messwerte aufnimmt. Gesteuert wird das ganze durch die Logik die auf ein Arduino kompatiblem Board programmiert wird.

A. Mechanik

1) Wassertransport und Gefäß: Um den Einsatzbereich so flexibel wie möglich zu halten haben wir uns für ein Pumpensystem entschieden. Dadurch ist die Anordnung der Pflanze relativ zum Wassertank nicht relevant. Der hiermit einhergehen erhöhte Strombedarf ist zu verkraften, da unser System nicht auf Batterien angewiesen ist.

Für den Wassertransport haben wir uns für eine Zahnradpumpe entschieden. Diese setzte sich gegenüber anderen Lösungen vor allem wegen ihrer selbstsaugenden Eigenschaft durch. Selbstsaugend bedeutet, dass eine mit Luft gefüllten Wasserleitung und Pumpe Wasser ansaugen und fördern kann. Als Alternativen wurden Ventile und Kreiselpumpen angedacht. Die Kreiselpumpe konnte trotz dem geringeren Stromverbrauch und geringerem Geräuschpegel nicht durchsetzen. Noch weniger Strom und Lärm verursachen Ventile, die aber auf gespeicherte Energie angewiesen sind. Entweder durch Druck im Wassertank oder durch Erhöhung des Tankes über den Ausfluss. Auf Grund der Wahl der Zahnradpumpe kann ein 4 mm Schlauch zur Förderung des Wassers genutzt werden. Das Gefäß ist frei wählbar deswegen wurde hier ein fünf-Liter-Weinballon gewählt. In den der Schlauch gesteckt wird. Auf der anderen Seite wird das Schlauchende mit einem durchbohrten Kantholz in der Pflanzenerde befestigt.

Die Nachteile der Zahnradpumpe (Stromverbrauch und Lärm) haben wir dadurch minimiert, dass das Gießsystem über ein Netzgerät mit Strom versorgt wird ist der erhöhte Stromverbrauch zwar unschön aber vertretbar. Um den Geräuschpegel zu minimieren wird über einen Helligkeitssensor verhindert, dass in der Nacht gegossen wird. Dies wird im Abschnitt III-B3 näher erläutert.

2) Gehäuse: Das Gehäuse wurde möglichst klein, aber genügend Platz für die Elektronik konzipiert. So muss es genügend Platz bieten um ein LCD-Display, zwei Taster, die Hauptplatine, das Arduino Board, die Vorschaltung für den Bodenfeuchtesensor, die Verkabelung und Anschlüsse für die Sensoren, den Motor und Stromversorgung bieten. Um den Körper der Box zu gestalten, wurde der BoxMaker benutzt und mit Inkscape angepasst. Das gesamte Gehäuse wurde im FabLab Erlangen mit Hilfe des Lasercutters gefertigt.

B. Elektronik

Im folgenden wird der Aufbau der Version 1.0 vorgestellt. Während des Aufbau vor allem aber während der Testphase sind ein paar Probleme aufgetreten die dazu geführt habe, dass eine neue Version 1.1 erstellt werden muss. Diese neue Version ist jedoch aus zeitlichen Gründen noch nicht komplett aufgebaut. Es wurden ein neues Layout erstellt und die Software hierzu bereits so angepasst, dass diese ohne große Änderungen übernommen werden kann. Im folgenden wird

Pumpentyp	*selbstsaugend*	*Lautstärke*	*Stromverbrauch*	*Förderleistung*	*Preis*
Zahnradpumpe	ja	sehr laut	@12V 2,8A	gering (XXXX l/min)	2,95 €
Kreiselpumpe	nein	mittel laut	@12V 600mA	groß (XXXX l/min)	2,95 €
Ventil	nein	leises Klacken	@12V 80 mA	keine eigene	4,95 €

Tabelle I
VERGLEICH ZWISCHEN WASSERPUMPEN UND VENTIL

die Funktionsweise auf Basis der Version 1.0 erläutert. An gegebener Stelle wird auf die Anpassungen eingegangen die bereits umgesetzt wurden bzw. noch in Planung sind.

1) Hauptplatine: In Abbildung XXX ist die Version 1.0 des Layouts und des Schaltplans zu sehen. Die drei Funktionsblöcke

- Senorschaltung
- Kommunikationsschaltung
- Stromversorung

werden im folgenden erläutert.

2) Arduino Nano: Als Kern des Projektes steht ein Arduino Nano Nachbau, dieser sticht durch seine geringe Größe und günstigen Preis aus der Arduino kompatiblen Boards heraus. Er besitzt einen ATmega168 Microcontroller

3) Senorschaltung: Für den Helligkeitssensor wird ein einfacher Photo-widerstand verwendet, der über einen Spannungsteiler an einem der Analogen Pins des Arduino Bords angeschlossen ist. Der Pin verfügt über einen 10Bit AD Konverter und gibt demnach einen Integerwert von 0 - 1023 zurück. Der Bodenfeuchtesensor bestimmt den Wassergehalt des Bodens über einen Widerstand zwischen zwei Zinken einer Messgabel. Je mehr Wasser im Erdreich vorhanden ist, desto kleiner ist der gemessene Widerstand im Boden. Die Schaltung für den Feuchtigkeitssensor ist identisch mit der des Helligkeitssensor. Der Feuchtigkeitssensor hat jedoch noch eine Vorschaltung Abbildung XXX2 über die es möglich ist entweder einen Analogen Wert zu erhalten, oder direkt eine Digitales Signal. Das Digitale Signal liefert einen Null-wert solange ein Grenzwiderstand nicht überschritten wird. Über ein Potentiometer(1) lässt sich diese Grenze Einstellen. Wegen des schlechten Zugangs zum Potentiometer wird der Digitale Output nicht verwendet sondern der Analoge Messwert selbst ausgewertet.

Anpassung zu Version 1.1: Leider zeigte sich das nach nur 48 Stunden Dauermessung die Gabel erhebliche Korrosion erlitten hat (Abbildung XXX3).Die Vorschaltung sieht keine Abschaltung des Messprozesses vor noch eine Umpolung der Gabel. Deswegen muss die gesamte Vorschaltung stromlos geschaltet werden, um das Auflösen des Sensors zu verlangsamen. Hierfür haben wir für die Version 1.1 eine Transistorschaltung für den Sensor eingefügt. Eine Verpollung der Messgabel ist damit leider nicht möglich. Dies kann jedoch relativ einfach gelöst werden, indem der Sensor alle paar Wochen *manuell* umgepolt wird. Hierfür muss lediglich die Messgabel vom Verbindungskabel abgesteckt werden und um $180°$ gedreht wieder verbunden werden.

4) Stromversorgung: Eine Stromversorgung über USB ist nicht möglich, da die Pumpe in Volllast über 2 Ampere benötigt. Um den Strom zu begrenzen haben wir einen $5\ \Omega$ Lastwiderstand in Reihe geschalten. Die Ansteuerung der Pumpe mit dem Mikrocontroller wurde über eine Transistorschaltung Gelöst.

Daher wird die Stromversorung extern über ein 12V Netzteil betrieben. Für die Kommunikation mit einem Computer wurde ein xBee Modul verbaut, welches nur mit 3,3 V betrieben werden darf. Auch die Datenleitungen dürfen nur mit 3,3 V betrieben werden. Hierfür war als ein Pegelwandler notwendig.

5) Kommunikation über XBee: Ziel der Kommunikation zwischen Mikrocontroller und Computer ist es, die Messdaten zu übertragen und die Möglichkeit die Grenzwerte ab wann gegossen werden soll bzw. wie lange gegossen werden auch nachträglich zu ändern. Hierfür sollten die Variablen über eine Serial-read neue beschrieben werden. Genauso soll der Mikrocontroller die Messwerte der aktuellen Feuchtigkeit übermitteln

Die Kommunikation über das XBee Modul hat zu stärkeren Problemen geführt. Mit der Version 1.0 konnte keine Verbindung mit den PC hergestellt werden.

C. Logik

1) Mikrocontroller: Die Aufgaben des Mikrocontrollers sind die folgenden:

- Messung der Feuchtigkeit
- Messung der Helligkeitssensor
- Vergleich der Messwerte mit den festgelegten Grenzen
- Ansteuerung der Pumpensystem
- Ausgabe der wichtigen Variablen auf dem Display über Taster
- Manuelles Gießen über Taster
- Display Hintergrundbeleuchtung abschalten nach 60 Sekunden
- Übertragen/Empfangen von Einstellungen über xBee Verbindung

In der ersten Version wurden die Messungen andauernd durchgeführt. Wie aber bereits oben erläutert hat dies zu einer starken Korrosion am Feuchtigkeitssensor geführt. Aus diesem Grund wurde für die zweiten Version einige Änderungen eingeführt. Anstelle bei jedem Aufruf der Loop() Methode eine Messung durchzuführen wird nur noch alle 4 Stunden eine Messung vorgenommen. Ist das Intervall abgelaufen werden die Sensoren über eine Transistorschaltung angesteuert und ausgelesen. Dadurch sind die Sensoren nur noch einen kurzen Zeitraum unter Spannung, was die Zerstörung des Feuchtigkeitssensor stark reduziert. Da diese Änderung eine neue Platine benötigt hat, haben wir sowohl den Feuchtigkeitssensor als auch den Helligkeitssensor mit einer Transistorschaltung ausgestattet.

2) Websteuerung:

IV. Sparversion

In der der günstigeren Variante der Gießanlage wurde alles, wurde die Konstruktion auf das wesentlichste beschränkt,das Gießen.

A. Aufbau

In dieser Version wird auf das Display und die Kommunikation verzichtet.

B. Logik

Um weiter Stromsparen zu können wurde diese Version nicht mit Arduino, sondern mit C programmiert. Dies ermöglicht die Ausnutzung der Sleep Modi und die Interrupts des ATMega328. Dadurch befindet sich der „Arduino Nano“ hauptsächlich im Schlafmodus und verbraucht deutlich weniger Energie.

V. BOM - Bill of Matirial

VI. Kostenplan

VII. Resüme - Do’s And Dont’s