

# Pflanzengießanlage

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

**Zusammenfassung**—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. Im folgenden wird der Aufbau zweier Lösungen vorgestellt. Die erste Lösung basiert auf dem Arduino-System die einen einfachen Einstieg in Umgang mit Mikrocontroller ermöglicht. Der zweite Ansatz baut auf einen XXXX Microkontroller auf ohne die Verwendung der Arduino Programmierumgebung, welche dadurch die Möglichkeit von Sleep Modi und die Verwendung von Interrupts ermöglicht und somit eine Low-Energy Variante des Gießsystems darstellt.

## I. IDEE

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

Es lassen sich viele Faktoren finden, die einen Einfluss auf das Wohlbefinden einer Pflanze haben. Hierunter fallen unter anderem die Bodenfeuchtigkeit im Topf der Pflanze, die durchschnittliche Sonneneinstrahlung, der Luftqualität sowie der Nährstoffgehalt des Bodens. Diese nicht abschließende Aufzählung zeigt, dass eine Eingrenzung der zu untersuchenden Faktoren notwendig ist um das Projekt in der gegebenen Zeit fertigzustellen. Auch in Hinblick des Einsatzgebietes gibt eine Vielfalt von unterschiedlichen Möglichkeiten wie z.B. Einpflanzenbetrieb oder Mehrpflanzenbetrieb sowie Indoor oder Outdoor. Im Rahmen des Projekts werden demnach folgende Eingrenzung vorgenommen.

- Die Bodenfeuchtigkeit wird abgeleitet aus der Widerstandsänderung einer Messgabel die sich im Erdreich der Pflanze befindet. Die Widerstandsänderung ergibt sich aus der Änderung des Wassergehalts der Erde.
- Das System soll für eine alleinstehende Pflanze aufgebaut werden.
- Das Einsatzgebiet wird im Innenbereich sein. Dadurch fallen Einschränkungen bzgl. Witterungsbeständigkeit weg sowie für die Energieversorgung.
- Das System wird so ausgelegt, dass es über ein Netzteil ständig mit Strom versorgt wird.

- Das System soll über eine Drahtlose Verbindung eingesetzt werden können. Diese Einschränkung gilt nur für die Arduino Variante, da in der Sparversion auf die Kommunikation verzichtet wird. Hier wird die Einstellung direkt im Code vorgenommen.

Weitere Ideen wie z.B. eine Autarke Lösung mit Batteriebetrieb, ein *Mehr-Pflanzenbetrieb* oder ein System für den Garten wurden zwar diskutiert aber wegen dem erhöhten Zeitaufwand und begrenzten Budgets nicht weiter verfolgt. Im folgenden wird Arduino Version ausführlich vorgestellt. Die *Sparversion* wird im Anschluss kurz erläutert, da viele der Funktionen grundsätzlich identisch sind und darauf verzichtet diese erneut auszuformulieren.

## III. ARDUINO VARIANTE

Im Folgenden beschreiben wir das Vorgehen für das Gießsystem basierend auf der Arduino Variante. Es wird demnach auf folgende Bereich eingegangen.

- Zu Beginn beschreiben wir verschiedenen Möglichkeiten des Wassertransports zur Pflanze.
- Im weiteren stellen wir das Vorgehen zur Erstellung des Gehäuses für die Elektronik dar.
- In Abschnitt Elektronik werden wir auf den Aufbau der Platine und die hiermit verbunden Bereiche Sensorschaltung, Stromversorgung und Kommunikation eingehen.
- Anschließend gehen wir auf das Programm ein, mit dem entschieden wird wann der Zeitpunkt erreicht ist um die Pflanze zu gießen sowie.
- Abschließend werden die verwendeten Materialien aufgelistet und
- eine kurzer Kostenplan vorgestellt.

### A. Wassertransport

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 einhergehende erhöhte Strombedarf ist zu verkraften, da unser System nicht auf Batterien angewiesen ist.

Für den Wassertransport haben wir uns für eine Zahnraddpumpe entschieden. Diese setzt 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

Tabelle I  
VERGLEICH WASSERPUMPEN UND VENTIL

Eigenschaft	Zahnradpumpe	Kreiselpumpe	Ventil
Selbstsaugend	ja	nein	nein
Lautstärke	sehr laut	mittel laut	leises Klacken
Stromverbrauch	@12V 2,8A	@12V 0,6A	@12V 80mA
Förderleistung	gering	groß	keine eigene
Preis	2,95 €	2,95 €	4,95 €

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-C1 näher erläutert.

### B. Gehäuse

Das Gehäuse wurde so konzipiert, dass es möglichst klein ist aber genügend Platz für die Elektronik bietet. Im Gehäuse verbaut sind

- das LCD-Display
- zwei Taster zur Steuerung der Displayanzeige und zum manuellen Gießen
- der Photowiderstand zur Messung der Helligkeit (platziert zwischen den Tastern)
- Stromschluss auf der linken Seite
- Ausgänge für die Pumpe und den Feuchtigkeitssensor auf der rechten Seite

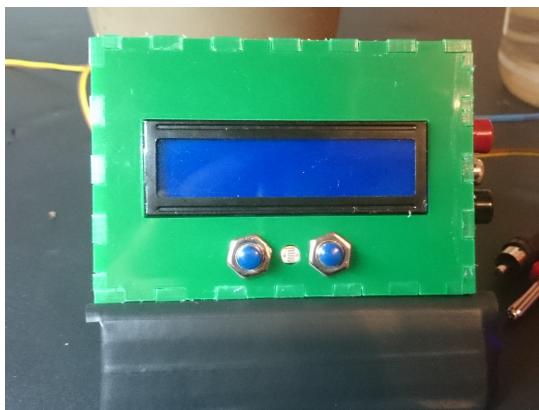


Abbildung 1. Gehäuse Frontansicht

Das Gehäuse wurde in FabLab Erlangen mit einem Lasercutter gefertigt. Für das Design des Gehäuses wurde der BoxMaker<sup>1</sup> verwendet. Das Gehäuse besteht aus grünen 3mm dicken Acrylglass.

<sup>1</sup>LINK BOXMAKER

### C. Elektronik

Im folgenden wird der Aufbau der Version 1.0 der Arduino Variante vorgestellt. Während des Aufbaus, vor allem aber während der Testphase sind ein paar Probleme aufgetreten die dazu geführt habe, dass eine neue Version 1.1 erstellt werden musste. Diese neue Version ist jedoch aus zeitlichen Gründen noch nicht komplett aufgebaut. Ein neues Layout wurden bereits erstellt und die Software ist so angepasst worden, dass diese ohne größere Änderungen übernommen werden kann. Daher wird 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) Sensorik:* Für den Helligkeitssensor wird ein einfacher Photowiderstand verwendet, der über einen Spannungsteiler an einem der Analogen Pins des Arduino Bords angeschlossen ist. Der Pin verfügt über einen 10 Bit AD Konverter und gibt demnach einen Integerwert von 0 - 1023 zurück. Der Bodenfeuchtigkeitssensor bestimmt den Wassergehalt des Bodens über eine Widerstandsmessung zwischen den zwei Zinken einer Messgabel. Je mehr Wasser im Erdreich vorhanden ist, desto kleiner ist der gemessene Widerstand im Boden.

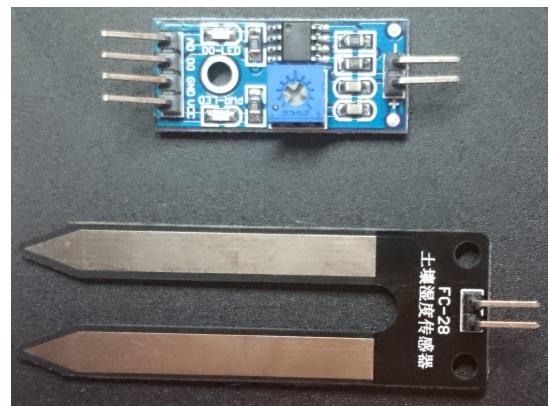


Abbildung 2. Feuchtigkeitssensor mit Vorschaltung

Der Feuchtigkeitssensor benötigt keine weiteren Schaltelemente, da er über eine Vorschaltung verfügt, in der ein Spannungsteiler bereits verbaut ist. Abbildung 2 zeigt den verwendeten Sensor und die Vorschaltung. Es ist möglich sowohl den Analogen Wert, oder ein Digitales Signal auszuwerten. 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 im eingebautem Zustand wird der Digitale Output nicht verwendet sondern der Analoge Messwert selbst ausgewertet und mit einer Variablen im Mikrocontroller abgeglichen.

*Anpassung zu Version 1.1:* Leider zeigte sich, dass nach nur 48 Stunden Dauermessung die Gabel erhebliche Korrosion erlitten hat, Abbildung 3 zeigt dies deutlich. 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. Über diese Schaltung wird die Vorschaltung des Feuchtigkeitssensor stromlos geschaltet. Eine Verpolung 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  $180^\circ$  gedreht wieder verbunden werden. Beim Routing der Platine wurde die Verbindung MANUALGIES2 zu Pin 7 vergessen, weswegen diese nachträglich durch eine Drahtbrücke verbessert wurde.

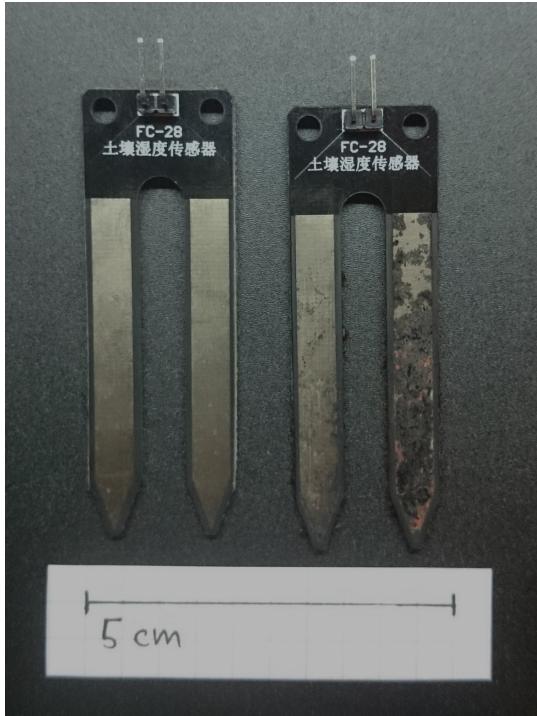


Abbildung 3. Vergleich neuer Sensor und Sensor mit 48h Dauerbetrieb

**2) Kommunikation:** Ziel der Kommunikation ist es, eine Möglichkeit zu haben das Gießsystem im laufenden Betrieb einzustellen. Jede Pflanze braucht unterschiedlich viel Wasser. Genauso hat die Zusammensetzung der Erde einen Einfluss auf den gemessenen Widerstand des Feuchtigkeitssensors. Diese Werte müssen daher für jede Pflanze extra einstellbar sein. Wir haben uns für eine Drahtlose Kommunikation entschieden und haben uns auf den XBee Standard geeinigt. Das XBee Modul wurde direkt, d.h. ohne Verwendung eines *Arduino XBEE-Shields*, mit dem Arduino verbunden. Hierfür ist es notwendig einen Pegelwandlung von 5 V auf 3,3 V vorzunehmen, da die Logik-Pins des XBee Moduls nicht mit 5 V betrieben werden können. Bei dieser Pegelschaltung ist uns in Version 1.0 ein Fehler unterlaufen, weswegen wir keine Verbindung mit einem Computer aufbauen konnten. Der Fehler war, dass für die Pegelwandlung für den Anschluss TX, der Transistor falsch geschaltet wurde. Auf dem Gate des Transistors wurde das falsche Spannungspotential geschalteten. Dieser Fehler wird für die Version 1.1 behoben und noch getestet.

Die zweite Schnittstelle mit dem Menschen wird über ein Display ermöglicht. Hier lässt sich über einen Taster die einzelnen Variablen mit ihren aktuellen Werten Überprüfen.

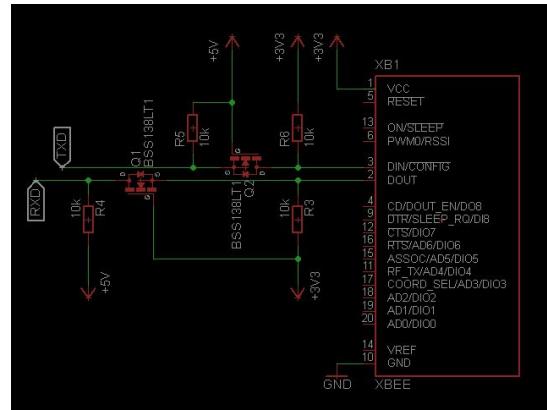


Abbildung 4. Pegelwandler mit Small-Signal-Transistor BSS138W

Bei dem Display handelt es sich um ein 2 Zeiliges LCD-Display mit jeweils 16 Zeichen. Die Kommunikation zwischen Display und Mikrocontroller wird über eine I2C-Schnittstelle bewerkstelligt. Hierfür haben wir die frei verfügbare Liquid-Crystal\_I2C Libary von fdebrabander verwendet.<sup>2</sup>

**3) Stromversorgung:** Eine Stromversorgung über USB ist nicht möglich, da die Pumpe in Vollast 2,8 A benötigt. Um den Strom zu begrenzen haben wir einen  $5\ \Omega$  Lastwiderstand in Reihe geschalten. Die Ansteuerung der Pumpe über dem Mikrocontroller wurde über eine Transistorschaltung gelöst (Ziffer 1 in Abbildung 5). Es ist zu überlegen, den Transistor und den Mikrocontroller über eine Schutzdiode über die Anschlüsse VENTIL-1 und VENTIL-2 vor Überspannung zu schützen, die beim abgeschaltet des gesamten Systems auftreten. Da die Pumpe jedoch nur für einen kurzen Zeitraum (Standardeinstellung 5 sekunden) in Betrieb ist, wurde davon abgesehen.

#### D. Programm

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 Version 1.1 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 wird der Feuchtigkeitssensor über eine Transistorschaltung angeschaltet

<sup>2</sup><https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library>

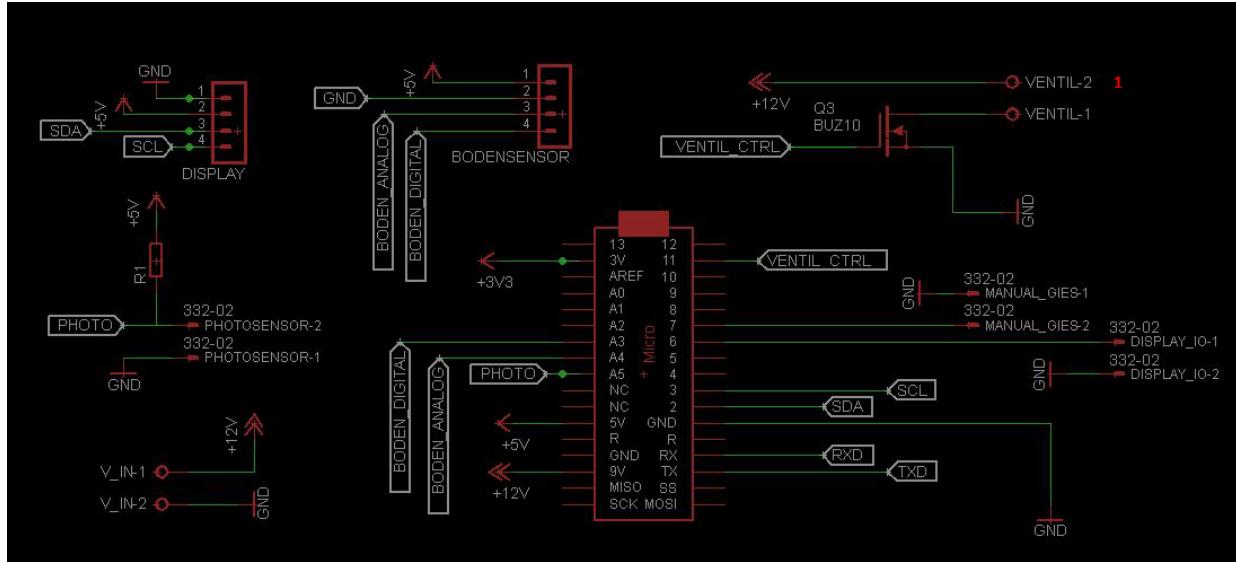


Abbildung 5. Schaltpaln V1.0 mit Arduino Micro

und anschließend ausgelesen. Dadurch ist der Sensoren nur noch für einen kurzen Zeitraum unter Strom, was die Lebensdauer des Feuchtigkeitssensor verlängert.

Da die Erde in jeder Pflanze eine andere Zusammensetzung hat und jede Pflanze unterschiedlich viel Wasser benötigt ist es notwendig die Grenzwerte für die Feuchtigkeit und die Wassermenge einzustellen. Hierfür gibt es ein Cmd-Tool welches über die

#### E. BOM - Bill of Material

#### F. Kostenplan

### IV. SPARVERSION

In 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 Modus und die Interrupts des ATmega328. Dadurch befindet sich der „Arduino Nano“ hauptsächlich im Schlafmodus und verbraucht deutlich weniger Energie.

### V. RESÜME - DO'S AND DONT'S