

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 Top der Pflanze. Neben dieser können noch weitere Informationsquellen herangezogen werden, die über das Wohlbefinden der Pflanze Aufschluss geben. Hierunter fallen unter anderem der Nährstoffgehalt der Erde, die Temperatur an der Wurzel oder die Luftqualität im Umfeld.

In der folgenden Darstellung soll der Entstehungs- und Entscheidungsprozess des ersten Prototyp der "Gießanlage für die WG" näher erläutert werden. Zu Beginn wird eine kurze Ideensammlung aufgestellt, aus der sich der Prototyp hervorhebt. Im weiteren wird der Gesamtaufbau erläutert, der in den Unterkapiteln *Wassertransport und Gefäß*, *Sensorik*, *Gehäuse*, *Programmierung* und *Weboberfläche* weiter detailliert wird. Abschließend wird eine Auflistung aller Materialien und Tools gegeben, die für das Projekt verwendet wurden, sowie eine Kostenaufstellung. Im letzten Abschnitt geben wir ein kurzes Resümee und stellen unsere Do's and Don't's zusammen, die uns während des Projekts aufgefallen sind, damit die selben Fehler nicht nochmal gemacht werden.

II. VARIANTEN

Die Anwendungsfelder lassen sich grob über zwei Achsen aufspannen. Zum einen die Anzahl der zu bewässernden Pflanzen und zum anderen die Art der Bewässerung. Recht schnell haben wir uns für eine Einpflanzenlösung entschieden, da eine Erweiterung auf mehrere Pflanzen grundsätzlich möglich ist, dies jedoch wegen dem erhöhten Aufwand und die notwendigen Anpassungen auf den individuellen Einzelfall den Projektrahmen gesprengt hätten.

Bei einer Einpflanzen-Variante ...

Bei einer Mehrpflanzen-Variante ...

Die Gießart lässt sich grob in Pumpgießen und Schwerkraftgießen einteilen. Beim Pumpgießen wird, wie im Namen bereits enthalten, das Wasser mithilfe einer Pumpe zu der Pflanze transportiert. Dadurch haben wir einen größeren Freiraum für die Positionierung des Wassertanks im Bezug zur Pflanze. Dieser kann, je nach Leistung der Pumpe in einiger Entfernung zur Pflanze verstaubt werden. Beim Schwerkraftpumpen wird die Schwerkraft genutzt, um das Wasser zu verteilen. Hierbei ist jedoch immer darauf zu achten, dass der Wassertank oberhalb der Pflanze angebracht wird. Für diese Variante ist außerdem ein Ventil notwendig, welches auch bei wechselnden Füllständen zuverlässig schließt.

III. AUFBAU

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

A. Mechanik

1) *Wassertransport und Gefäß*: In der Mechanik wurde sich für eine Zahnradpumpe entschieden. Diese setzte sich gegenüber anderen Lösungen vor allem wegen ihrer selbstsaugenden Eigenschaft durch. Selbstsaugend bedeutet, dass selbst trotz einer mit Luft gefüllten Wasserleitung und Pumpe Wasser ansaugen und dann 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 Tanks 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.

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

1) Hauptplatine:

2) *Sensorik*: Sensorisch erfasst werden die Helligkeit und die Bodenfeuchtigkeit. Um eine Pflanze mit Wasser zu versorgen reicht es zu wissen, wie feucht der Boden ist. Aber um Rücksicht auf die Bewohner einer Wohnung zu nehmen, ist zusätzlich ein Helligkeitssensor verbaut, dieser soll dafür sorgen, dass bei Dunkelheit im Raum eine "Nachtruhe" gehalten wird. Der Bodenfeuchtesensor bestimmt den Wassergehalt des Bodens über einen Widerstand zwischen zwei Zinken einer Messgabel. Leider zeigte sich das nach nur 48 Stunden Dauermessung die Gabel erhebliche Korrosion erlitten hat. Außerdem sieht die Vorschaltung 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.

3) Kommunikation über XBee:

4) Stromversorgung:

C. Logik

1) Mikrocontroller:

2) Websteuerung:

D. BOM - Bill of Matirial

E. Kostenplan

F. Resüme - Do's And Dont's

<i>Pumpentyp</i>	<i>selbtsaugend</i>	<i>Lautstärke</i>	<i>Stromverbrauch</i>	<i>Förderleistung</i>
Zahnradpumpe	ja	sehr laut	@ 12V 2,8A	gering (XXXX l/min)
Kreiselpumpe	nein	mittel laut	@ 12V 600mA	groß (XXXX l/min)
Ventil	nein	leises Klacken	@ 12V 80 mA	keine eigene

Tabelle I

VERGLEICH ZWISCHEN WASSERPUMPEN UND VENTIL