



Maturitätsarbeit HS 2021/22

Aquaponik – Entwicklung und Bau einer Aquaponik-Anlage

Maturitätsarbeit
von

Niels Sigg
Klasse 4e

geboren am 22.03.2004

Betreuungsperson: Wolfgang Pils

Winterthur, 3. Januar 2022

Abstract

Eine Aquaponik-Anlage ist ein sich (fast) selbstversorgendes System mit Fischen und Pflanzen. Dabei werden dem System keine Düngemittel oder sonstige Pestizide hinzugefügt und die Tiere leben mit den Pflanzen in einem Gleichgewicht. In dieser Arbeit erzähle ich über den Bau und die Entwicklung einer solchen Anlage in meinem eigenem Zimmer und erläutere die Probleme, Erfolge und natürlich auch die Ernte der Pflanzen. Damit eine solche Anlage zuverlässig funktioniert und es nicht zu einer Überschwemmung kommt, steckt sehr viel Technik und Code dahinter. Neben all den Pumpen, die das Wasser durch alle Stationen der Anlage befördern, braucht es überall und meist sogar mehrere und verschiedene Wasserstands-Sensoren, die die ganze Zeit die aktuellen Wasserstände in der ganzen Anlage im Blick haben. Ich habe während einem halben Jahr ein Ebbe- und Flutsystem und ein Nährstoff-Film-Technik-System gebaut, welche beide von einer selbst gebauten Steuerungseinheit mit einem Arduino UNO angesteuert werden. Dazu gibt es noch ein kleines LCD-Display, welches schematisch den momentanen Zustand der Anlage ausgibt und über welches man verschiedene Programme der Anlage ausführen kann. Weil der Bau der Anlage mehrheitlich im Winter stattfand und es aufgrund der Jahreszeit schwierig war, Nutzpflanzen zu züchten oder erst zu kaufen, mussten andere Alternativen her, aber im Großen und Ganzen wurden die Ziele des Baus einer smarten Aquaponik-Anlage sehr gut erreicht.



Abb. 1 Fertige Aquaponik-Anlage (Stand 17.12.2021)

Inhaltsverzeichnis

Vorwort	3
1 Was ist Aquaponik?	4
2 Planung und Konzept der Anlage	4
2.1 Die Nährstoff Film Technik	5
2.2 Ebbe und Flut	5
3 Bau der Anlage	6
3.1 Aquarium	6
3.1.1 Pumpen	7
3.2 NFT - Nährstoff-Film-Technik	10
3.3 Ebbe- und Flutsystem	11
3.4 Überlaufbecken	12
4 Steuerungseinheit des Wasser-Kreislaufes	13
5 Steuerung	16
5.1 Wasserstände	17
5.1.1 Aquarium	17
5.1.2 NFT-System	18
5.1.3 Ebbe- & Flut	19
5.1.4 Überlaufbecken	20
5.2 Display	20
5.3 Automatik-Programm	21
5.3.1 Notfall-Massnahmen	21
5.3.2 Die Ebbe und die Flut	23
6 Betrieb der Anlage	26
6.1 Probleme	26
6.2 Pflanzen und Fische	29
7 Fazit	33
7.1 Zielerreichung	33
7.2 Kosten	33
7.3 Verbesserungen	34
Literatur und Quellen	36
Abbildungsverzeichnis	38
Anhang	39

Vorwort

Ich interessiere mich bereits seit vielen Jahren für die Aquaristik und habe zuhause zwei Aquarien. Ich stecke sehr viel Zeit in die Gestaltung und Wartung dieser Biotope und habe mich auch ein paar Mal in der Züchtung von Fischen und Garnelen versucht, jedoch ohne Erfolge... Eine weitere Leidenschaft sind handwerkliche, gestalterische und naturwissenschaftliche Projekte. Deswegen war auch von Anfang an klar, dass ich als Maturitätsarbeit ein Produkt herstellen und programmieren will. Auch sollte es noch einen praktischen Nutzen haben.

Leider war ich anfangs sehr ideenlos und mir fiel einfach nichts ein, was mich begeistern würde und auch noch gehaltvoll genug für eine Arbeit ist. Nach sehr vielen Überlegungen und Diskussionen mit meinen Eltern und Freunden, hatte aber Herr Wolfgang Pils die perfekte Idee für mich, nämlich den Bau einer Aquaponik-Anlage. Wie viele andere Leute wusste auch ich zuerst nicht, was das ist, aber ich realisierte sehr schnell, dass diese Arbeit genau das Richtige für mich sein wird, da sie meine zwei grossen Leidenschaften miteinander verbindet. Und auch rückblickend war dies eine der besten und richtigsten Entscheidungen, die ich treffen musste. Ich war nach der definitiven Wahl dieses Projektes voller Tatendrang und Elan, so dass ich eigentlich sofort anfangen wollte zu bauen, planen, schreiben und programmieren.

Ich danke allen, die mir auf diesem Wege geholfen und dazu beigetragen haben, dass das Projekt nicht gescheitert ist. Einen grossen Dank geht an meine Eltern aufgrund ihrer finanziellen Unterstützung und ganz besonders danke ich meinem Vater, der mir mit seinem elektrotechnischem Wissen in vielen Bereichen geholfen hat. Auch danke ich Herrn Wolfgang Pils für seine professionelle und erfahrene Begleitung durch das ganze halbe Jahr.

1 Was ist Aquaponik?

Aquaponik ist eine Verbindung aus den Wörtern Aquarium und Hydroponik. Damit ist ein System gemeint, das aus dem Zusammenspiel von einem Aquarium mit Fischen und Pflanzen in Hydrokultur besteht und ein Gleichgewicht bildet. Der Sinn dahinter ist, dass der von den Fischen und Pflanzen im Aquarium produzierte Dreck und Kot als Dünger für die Nutzpflanzen dient. So sollte es im Idealfall möglich sein, dass die Pflanzen ausserhalb vom Aquarium ohne Düngemittel oder sonstige Pestizide auskommen und wachsen. Das Futter für die Fische ist der einzige Zusatz, der dem System von aussen zugegeben wird und ansonsten versorgt sich alles selbst. Denn während die Fische Nährstoffe für die Pflanzen liefern, filtern und putzen die Pflanzen das Wasser auf physikalischer und biologischer Ebene und das Aquariumwasser besitzt immer die richtigen Wasserwerte. Mit einer Aquaponik-Anlage hat man eine Kombination von Fisch- und Pflanzenzucht.¹

In dieser Arbeit wird versucht, ein Aquarium mit Zierfischen zu einer Aquaponik-Anlage zu bauen und dabei sind nur die Pflanzen zum Verzehr gedacht.

2 Planung und Konzept der Anlage

Die Aquaponik-Anlage soll aus zwei verschiedenen Teilen bestehen, die separat betrieben werden können und die auch für unterschiedliche Pflanzenarten gedacht sind. Natürlich darf der wichtigste Teil, das Aquarium, nicht fehlen, denn aus diesem werden die ganzen Nährstoffe rausgepumpt und verteilt. Das heisst also, es braucht im Aquarium zwei Pumpen, die den Dreck der Fische und Pflanzen in die zwei

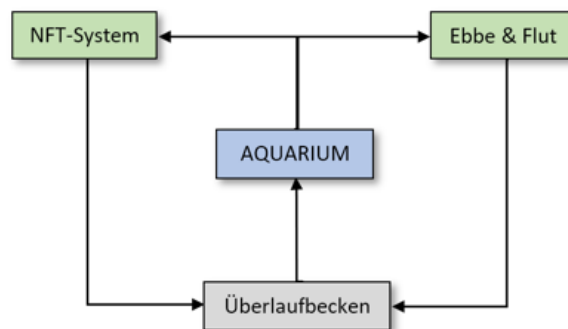


Abb. 2 Erster grober schematischer Aufbau der Aquaponik-Anlage

Systeme pumpen. Aber das Wasser soll danach nicht einfach zurück in das Aquarium fliessen, sondern sich zuerst in einem Sammelbecken oder Überlaufbecken sammeln. Dieses Becken sorgt dann für viel mehr Flexibilität, Puffer

¹Aquaponik im Eigenbau, Thomas Rücker, S. 6

und Spielraum, was das Herumpumpen des Wassers angeht. Für einen ersten groben Überblick sind in der Abbildung 2 die wichtigsten Stationen und Schlauchverbindungen schematisch eingezeichnet.

2.1 Die Nährstoff Film Technik

Dieses System ist besonders für schwachzehrende Pflanzen wie Salate und Kräuter geeignet. Die Idee dabei ist, dass das Wasser wie ein dünner Nährstoff-Film unter den Pflanzen durchfließt und sie diese mit ihren Wurzeln aufnehmen können. In diesem Nährstoff-Film hat es viele Bakterien, aber auch der Sauerstoff gelangt auf diese Weise zu den Wurzeln. Das Wasser soll dabei kontinuierlich am Fließen sein. Diese Methode ist weltweit sehr verbreitet, da sie sehr produktiv ist (braucht nicht viel Wasser). Das Ziel ist, dass nicht überall in diesem Kanal der gleiche Wasserstand herrscht, damit man die Pflanzen tiefer oder weniger tief in das Wasser stellen kann. ²

2.2 Ebbe und Flut

In einem Ebbe- & Flutsystem, wird, wie der Name schon sagt, versucht, eine Ebbe und eine Flut zu simulieren. Dabei gibt es ein eher grösseres Pflanzenbeet, welches mit Blähton gefüllt ist. Es funktioniert eigentlich genau gleich, wie normale Pflanzen in Hydrokultur, nur etwas schneller. In diesem Beet gibt es ganz unten eine ca. 5 cm grosse Nasszone, welche immer mit Wasser gefüllt bleibt. Darauf folgen dann eine wechselhafte Zone und ganz oben 5 cm trockenes Substrat. Der Wasserstand soll in diesem mittleren Teil kontinuierlich wechseln und so die Wurzeln der Pflanzen einerseits mit Wasser (Nährstoffen) versorgen, andererseits kommt so auch genug Sauerstoff zu den Pflanzen und die Wurzeln können nicht faulen. Das Substrat, in welchem sich die Pflanzen befinden, bietet eine grosse Oberfläche für Bakterien und dient als ein Biofilter für das Aquarium. ²

²Aquaponik im Eigenbau, Thomas Rücker, S. 27

3 Bau der Anlage

Die schematisch gezeichnete Anlage mit all den Stationen und Modulen, die verwirklicht werden soll, ist zwar schön und gut, um einen ersten groben Überblick über das ganze Vorgehen zu erhalten, aber wie und was genau gebaut werden soll, ist damit noch nicht klar. Nach vielen Einkäufen und Überlegungen, war dann das gesamte Material beisammen und der eigentliche Bau der Aquaponik-Anlage konnte beginnen.

3.1 Aquarium

Das Aquarium ist das Herzstück des ganzen Kreislaufes, da das Wasser durch jenes hindurch in alle Bereiche der Anlage gepumpt wird. Es werden in diesem Becken selber also zwei Zimmerbrunnenpumpen benötigt, die das von den Lebewesen im Aquarium verschmutzte Wasser in das NFT- und Ebbe- & Flutsystem pumpen. Dann gibt es noch eine Pumpe, die das von den Pflanzen gereinigte Wasser aus dem Überlaufbecken wieder zurück zu den Fischen pumpt. Diese Pumpe sorgt gleichzeitig auch dafür, dass im Aquarium eine gewisse Strömung herrscht, damit der ganze Mulm (Kot und Dreck von den Fischen und Pflanzen) auch gleichmässig auf dem Boden verteilt wird. Eine Strömung ist für viele Fische auch sehr wichtig, da sie dafür sorgt, dass alle Nährstoffe im Wasser verteilt werden.



Abb. 3 Vollständig ausgerüstetes Aquarium mit 2 Zimmerbrunnenpumpen.

3.1.1 Pumpen

Verwendet werden einfache, boden-ansaugende Zimmerbrunnenpumpen in den zwei hinteren Ecken des Aquariums. Diese haben den Vorteil, dass der Mulm so effizient wie möglich nahe am Boden angesogen werden kann. Mit zwei Pumpen ist auch die Wahrscheinlichkeit grösser, dass der Kies des Aquariums etwas gleichmässiger gereinigt wird.

Vom Überlaufbecken in das Aquarium wird das Wasser mit Hilfe eines Aquarium-Aussenfilters gepumpt. Dieser war schon vorhanden und damit ergibt sich der praktische Vorteil, dass das Wasser noch ein letztes Mal gereinigt wird, das Aquariumwasser klar bleibt und die sich im Aquarium befindenden Pumpen nicht zu sehr verstopfen beziehungsweise verschmutzten. Denn wie sich später dann herausstellte, löst sich vom Blähton sehr viel feiner Staub, welcher nicht unbedingt zu den Fischen gelange sollte.



Abb. 4 Dennerle Scaper's Flow Aussenfilter (links), Oase Aquarius Universal 440i Zimmerbrunnenpumpe (rechts)¹

Damit im Aquarium immer der ungefähre Wasserstand bekannt ist, wird dieser auf zwei verschiedene Arten gemessen. Es gibt ganz oben einerseits einen Pegelschalter mit Schwimmkörper, als Sicherung vor einem Überlauf, andererseits aber auch einen Infrarot-Sensor (IR-Sensor), welcher (fast) senkrecht auf die Wasseroberfläche gerichtet ist (wieso er nicht ganz senkrecht ist, wird weiter unten noch erklärt). So können grob drei Wasserstände unterschieden werden: Das Aquarium ist voll, wenn der Schwimmerschalter geöffnet wird, das Aquarium hat zu wenig Wasser, wenn der IR-Sensor ein zu

¹Conrad-Electronics, <https://www.conrad.ch/de/p/oase-aquarius-universal-440i-zimmerbrunnenpumpe-440-l-h-0-75-m-572138.html>, und Aquaristikshop, <https://www.aquaristikshop.com/aquaristik/Dennerle-Scapers-Flow-Anhaengefilter/908039/>, (abgerufen am 14.12.2021)

kleines analoges Signal ausgibt, und alles dazwischen kann in einem „Soll-Wasserstand“ zusammengefasst werden.

Der Infrarot-Sensor gibt, wie bereits gesagt, ein analoges Signal zwischen 0V und 3V aus. Diese Ausgabe ist aber nicht linear zum Abstand oder der Distanz zwischen Wasser und Sensor, sondern durch eine spezielle (unregelmässige) Funktion gegeben (siehe Abb. 6). Aufgrund dieser Ausgaben sind leider nicht alle Abstände durch eine eindeutige Spannung definiert. Es gibt zu jeder Distanz genau eine passende Spannung, doch das Umgekehrte ist nicht eindeutig. Wie in Abbildung 6 zu erkennen, gibt es für jede Spannung zwei mögliche Distanzen. Somit sollte der Sensor ca. 3 Zentimeter zu hoch installiert werden, damit der Abstand zwischen Wasser und Sensor gar nicht kleiner als 3 cm werden kann.

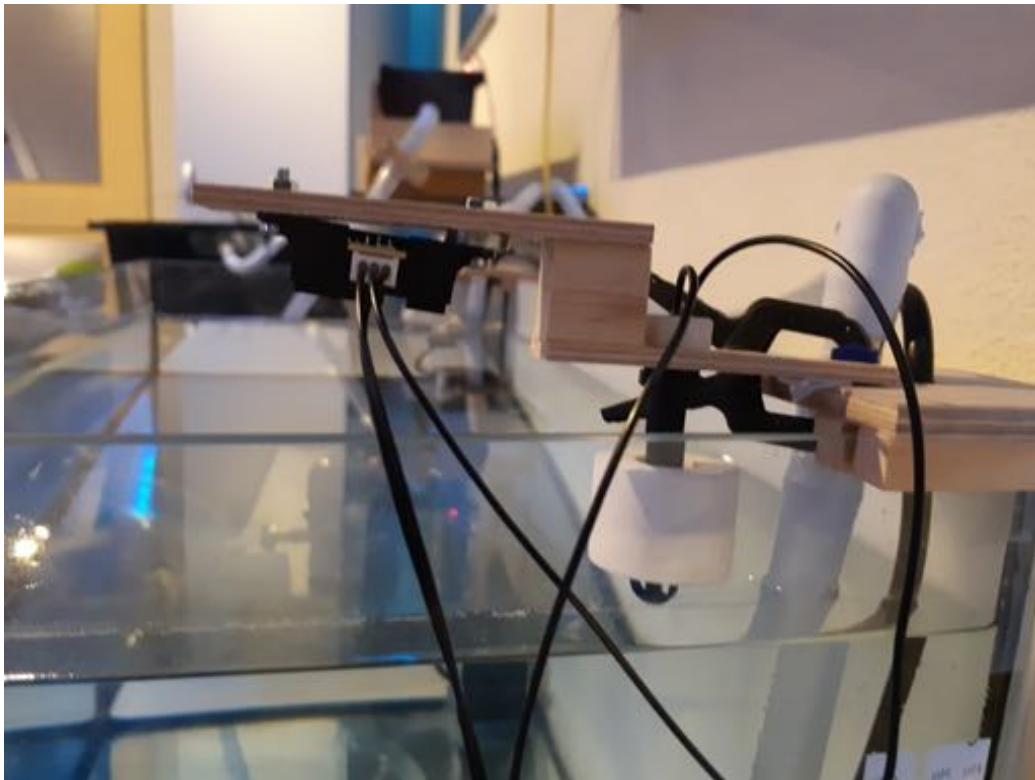


Abb. 5 Infrarot-Sensor und Pegelsensor am Aquarium

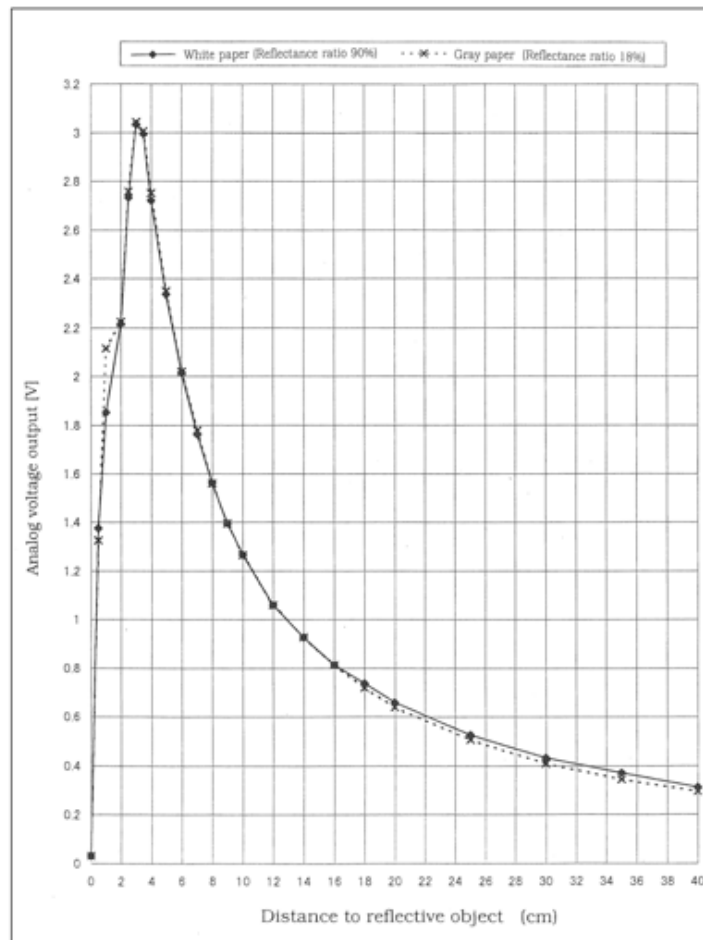


Abb. 6 Sharp IR-Sensor, Analoger Output verglichen mit Distanz ¹

Die Werte in Abb. 6 sind Musterwerte, von einem senkrecht auf eine ebene Fläche gerichteten Sensores. Leider ist das Wasser in einem Aquarium gar nicht eben und zu allem Unglück auch noch durchsichtig. Dies hat den folgenden Nachteil: Die Infrarot-Strahlen aus dem Sensor werden an der Wasseroberfläche fast gar nicht reflektiert und es ist sehr schwierig den wahren Abstand zwischen Sensor und Wasser herauszufinden. Deshalb wurde der Sensor etwas schräg ausgerichtet platziert, damit wenigstens ein Teil des Lichtes reflektiert und wieder empfangen werden kann. Mit all diesen Anpassungen der Befestigung des IR-Sensors, ist es nun problemlos möglich, den Wasserstand zu bestimmen.

¹DatasheetsPDF, S.4, <https://datasheetspdf.com/datasheet/GP2Y0A41SK0F.html>, (abgerufen am 14.12.2021)

3.2 NFT - Nährstoff-Film-Technik

Für das NFT-System wird eine 1m lange Regenrinne mit passenden Deckeln rechts und links verwendet, welche mit einer ca. 1.5° starken Neigung aufgestellt ist. In diese Rinne werden dann kleine Töpfe gestellt, die entweder mit Blähton oder Watte gefüllt sind. Das Wasser wird dann unter den Töpfchen durchfließen und die Pflanzen gelangen so an genug Wasser. Im unteren Teil der Rinne wird das Wasser etwas gestaut und so kann mit der Position der Pflanze bestimmt werden, wie viel Wasser sie bekommen soll. In der Rinne befindet sich ausserdem noch ein analoger Bodenfeuchtigkeits-Sensor der Marke "Maker Factory", welcher aber als einfacher Wasserstands-Sensor verwendet wird. Dieser Sensor dient lediglich dazu, dass das Wasser nie überlaufen kann, denn das Ziel ist es, dass das Wasser in der Regenrinne immer am Fließen ist, und nur im Notfall schaltet die Pumpe ab. Um die Fließgeschwindigkeit des Wassers kontrollieren zu können, kann man einerseits die Leistung der Zimmerbrunnenpumpe vermindern, andererseits gibt es beim Auslaufs-Schlauch einen selbstgemachten "Durchlaufverminderer", mit welchem man einstellen kann, wie schnell das Wasser aus der Rinne fließen darf (Abb. 7). Damit

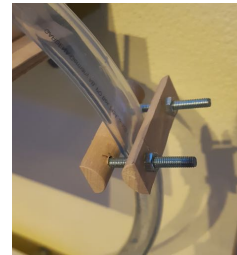


Abb. 7 Wasser-Durchlauf-Verminderer

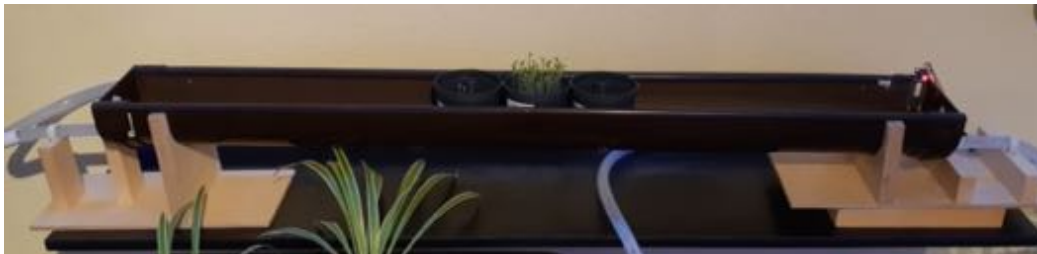


Abb. 8 NFT-System (Regenrinne) mit Wasserstands-Sensor und ersten kleinen Pflanzen

die ganze Regenrinne gut wasserdicht ist, wurde sie links und rechts in den Fugen komplett verleimt und auch die Schlauchanschlüsse sind mit viel Leim fixiert. Des weiteren werden die Schläuche ausserhalb der Rinne nochmals fixiert, damit nicht zu viel Druck (Bewegung) auf die Leimstellen ausgeübt wird.

3.3 Ebbe- und Flutsystem

Das Ziel dieses Systems ist es, den natürlichen Wasserzyklus von Pflanzen in Hydrokultur nachzuahmen. Dort gibt es nämlich eine nasse Zone ganz unten, eine trockene Zone ganz oben und ein sich wechselndes Gebiet dazwischen. Mit einem als Blumenbeet verwendeten „Wäschekorb“ wird versucht, genau so ein Ebbe- & Flutsystem zu bauen. Diese „Wanne“ wurde dafür komplett abgedichtet und fünf Zentimeter über dem Boden der Wanne gibt es ein Loch für den Auslauf. So bleibt ganz automatisch der untere Bereich von fünf Zentimetern immer nass, da das Wasser dort gar nicht mehr ablaufen kann. Ganz oben braucht es natürlich auch ein Loch für den Schlauch, der das Wasser aus dem Aquarium in die Wanne befördert. Überdies braucht es noch zwei Wasserstands-Sensoren in Form von Pegelsensoren, um zu wissen, wann wieder neues Wasser eingefüllt werden kann (also wann die Wanne leer ist) und um zu schauen, dass nichts überläuft. Einer dieser Sensoren befindet sich 10 cm unterhalb vom oberen Rand, der andere sitzt etwas höher als das Auslaufsloch unten (Abb. 9, links).

Wäre das Beet komplett mit Blähton gefüllt, würden die Sensoren nicht mehr funktionieren. Deshalb müssen die Sensoren mit Abdeckungen in Form von zweckentfremdeten Tupperwares geschützt werden (Abb. 9, Mitte & rechts).



Abb. 9 Ebbe- & Flutsystem während Bau von Innen

Wie auch beim NFT-System befindet sich ein Durchlauf-Verminderer beim Auslauf der Wanne, damit die Geschwindigkeit des Abflusses reguliert werden kann. So kann die gewünschte Bewässerungszeit der Pflanzen eingestellt werden.

Und auch hier darf auch eine Befestigung des Schlauches ausserhalb der Wanne nicht fehlen, sonst würde der Schlauch schnell abreißen.



Abb. 10 Fertiges Ebbe- & Flutsystem mit ersten Zimmerpflanzen

3.4 Überlaufbecken

Ein sehr wichtiger und zentraler Teil der Anlage ist das Überlaufbecken oder Sammelbecken des Wassers (Abb. 11). Hier läuft alles (fast saubere) Wasser aus dem NFT- und dem Ebbe- & Flutsystem zurück und wird dann wieder in das Aquarium gepumpt. Beim Rauspumpen wird das Wasser nochmals gefiltert, weil die Filterung mit den Pflanzen nicht immer ganz zuverlässig ist und der Blähton in den ersten Wochen sehr viel feinen Staub ins Wasser absondert. Somit wird gewährleistet, dass bestimmt sauberes Wasser zurück zu den Fischen ins Aquarium fließt. Trotzdem benötigt es auch in diesem Becken Wasserstands-Sensoren, um sicherzustellen, dass es nie überläuft. Andererseits muss aber auch gemessen werden können, ob noch genügend Wasser vorhanden ist, um dieses weiter ins Aquarium zu pumpen. Insgesamt wurden also zwei Pegel-Schwimm-Schalter oben und unten im Überlaufbecken angebracht.



Abb. 11 Überlaufbecken, bereits mit Wasser (und Dreck) gefüllt. Rechts hängt der Aquarium-Aussenfilter.

4 Steuerungseinheit des Wasser-Kreislaufes

Um die Aquaponik-Anlage zu steuern und zu betreiben, ist eine Steuerungseinheit notwendig, mit welcher man die Sensoren einlesen, deren Werte auswerten und schliesslich die Pumpen an- und abstellen kann. Die Steuerung der ganzen Anlage übernimmt ein Arduino UNO mit einem aufsteckbarem LCD und Keypad Shield. Versorgt wird der Arduino mit 5V Gleichstrom, welche direkt ein AC/DC-Wandler liefert, der an den 230V Stromkreis angeschlossen ist. Gesteuert werden müssen bis jetzt insgesamt drei Pumpen. Dazu wird eine 4er Stecker-Leiste verwendet, damit sicher noch genug Platz für allfällige Erweiterungen ist. Als Inputs zählen alle 6 Sensoren, der analoge IR-Sensor, wie auch die digitalen Schwimmsensoren. Die Sensoren, die ein analoges Signal ausgeben, müssen



Abb. 12 Arduino UNO mit LCD Keypad Shield ¹

¹Blogdarobotica: Shield Display, <https://www.blogdarobotica.com/2021/05/12/como-utilizar-o-shield-display-16x02-lcd-com-teclado-no-arduino/>, (abgerufen am 23.10.2021)

ausserdem noch mit einer 5V Spannung versorgt werden. Die Steuerungseinheit (Gehäuse) muss nun also vier 230V Outputs, sechs Sensoren Inputs, eine Stromversorgung und einen 5V Output haben. All diese Sachen werden in ein Gehäuse eingebaut mit den Anschlüssen auf der Rückseite und dem eingelassenen Display vorne.

Wie auf dem Schema (vgl. Abbildung 14) zu sehen, werden die 4 Ausgänge mit Relais angesteuert, welche wiederum mit Transistoren aktiviert werden.

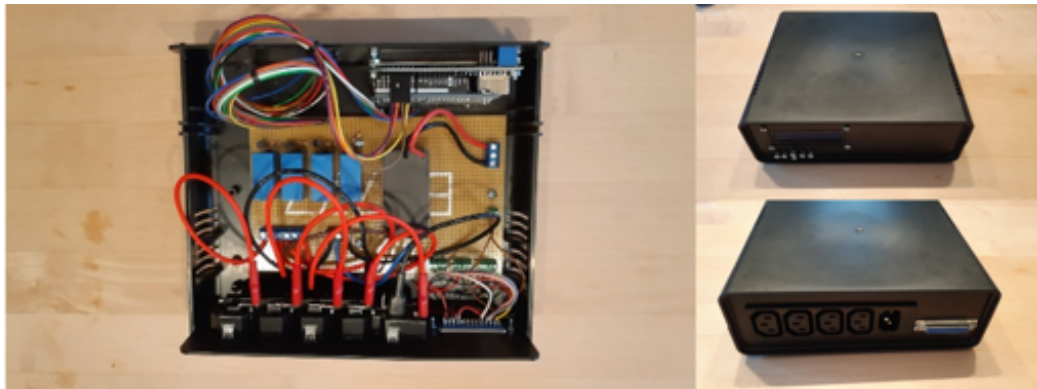


Abb. 13 Fertige Steuerung der Aquaponik-Anlage (links: Innenansicht, rechts: Aussenansicht)

Die Transistoren müssen verwendet werden, da die Output Leistung vom Arduino zu gering wäre, die Magnetspulen in den Relais zu aktivieren. Vor jeder Basis der Transistoren befindet sich noch ein 1k Ohm Widerstand. Der Transistor bildet den Schalter zum Minuspol vom Kollektor zum Emitter. Die Basis des Transistors ist dann mit einem digitalen Pin des Arduino Moduls verbunden. Dann können die Relais vom Arduino her aktiviert werden, sobald die entsprechenden Pins als Output deklariert und aktiviert wurden.

Beim Abschalten der Relais, also beim Deaktivieren des Magnetfeldes der Relais, entsteht jedoch ein kleines Problem: Immer, wenn ein Magnetfeld abgeschaltet wird, entsteht eine kurzzeitige Überspannung ausgehend vom Relais. Dies könnte dann (relativ schnell) andere technische Bauteile beschädigen. Deshalb ist eine sogenannte Freilaufdiode parallel zum geschalteten Relais notwendig. Diese Diode begrenzt die Überspannung, die vom Relais ausgeht, auf ihre Durchlassspannung und schützt somit effektiv andere Bauteile, wie zum Beispiel Transistoren, vor einer Überspannung.³

³Wikipedia, <https://de.wikipedia.org/wiki/Schutzdiode>, (abgerufen am 29.12.2021)

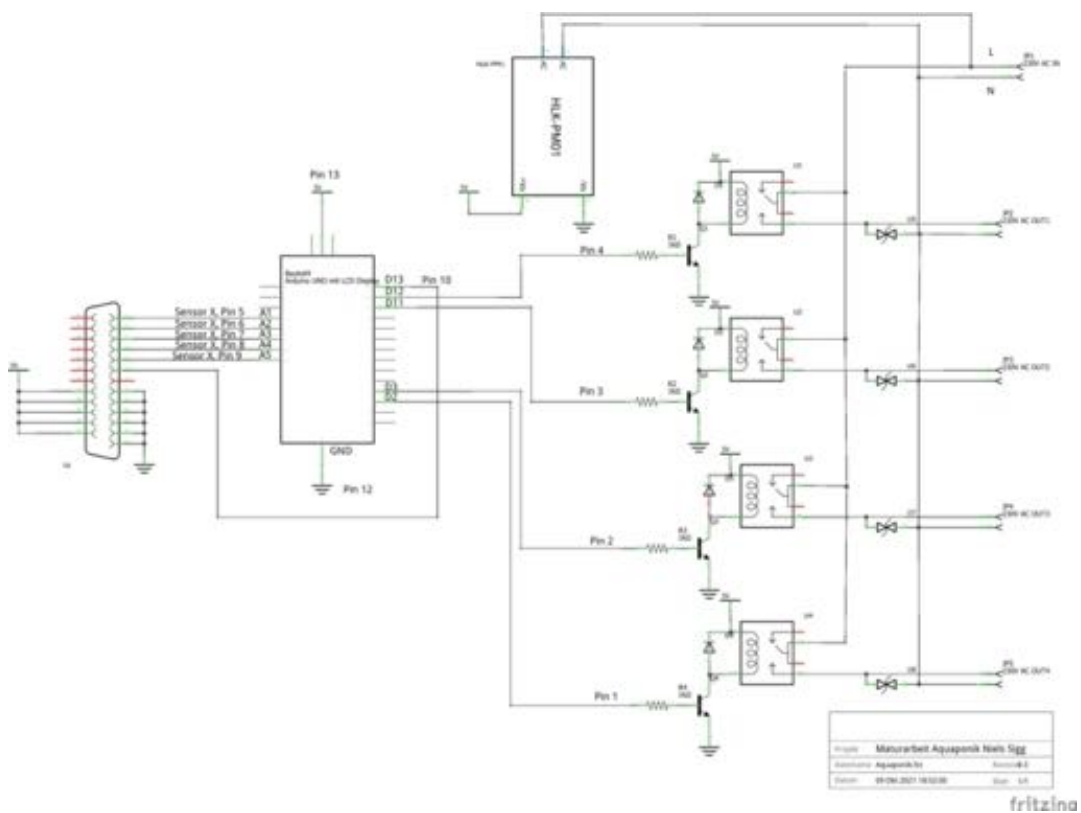


Abb. 14 Elektronisches Schema der Steuerungseinheit (gezeichnet mit fritzing).

Der grosse Vorteil von der Verwendung von Relais ist, dass der 5V und der 230V Stromkreis sicher voneinander getrennt sind und nirgends einen Kontakt zwischen den beiden Spannungen besteht (vgl. Abb. 15). Die einzige Schnittstelle stellt der AC/DC-Wandler dar, da er aus 230V Input Wechselstrom 5V Gleichstrom herstellt und, wie bereits gesagt, alle technischen Bauteile mit einer geeigneten Betriebsspannung versorgt. Man kann mit Hilfe von einem Relais also bequem mit 5V einen 230V Schalter steuern. Beim Bau und der Lötung der Platine muss man aber immer versuchen, diese zwei Stromkreise so gut wie möglich voneinander getrennt zu bauen und bestmöglich zu isolieren, damit keine Kurzschlüsse entstehen können und die restliche Technik wie auch der Mensch geschützt ist.

Es gab jedoch gegen Aussen, also vom 230V Kreislauf her, sehr grosse Störungen, die von den extern geschalteten Verbrauchern, also den Pumpenmotoren, ausgingen. Diese Störungen waren zum Teil so gross, dass ganze Bildschirme kurzzeitig aussetzten. Um dem entgegenzuwirken, musste man zu Varistoren greifen, da die Idee mit den Freilaufdioden im Wechselstrom-

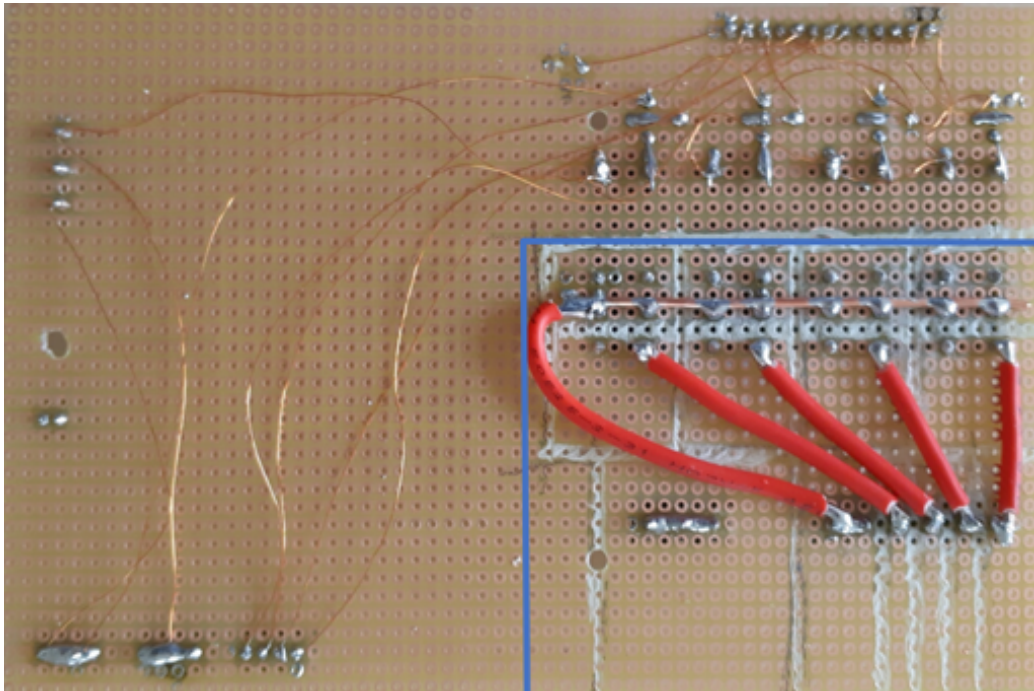


Abb. 15 Rückseite der Platine in der Steuerungseinheit. Blau umrandet: 230V Stromkreis

kreis nicht funktioniert, weil dieser seine Richtung immer wieder ändert. Die Varistoren funktionieren in beide Richtungen, jedoch erst ab sehr hohen Spannungen. Diese spezielle Fähigkeit wurde ausgenutzt und vor jedem 230V Output an die Pumpen gibt es nun einen von der Phase zum Neutral-Leiter verbindenden Varistor.

5 Steuerung

Programmiert wird der Arduino direkt mit der eigenen Entwicklungsumgebung (IDE) von Arduino. Die Programmiersprache dazu ist C++. Die Steuerung der ganzen Anlage soll schlussendlich alle Wasserstände in allen Becken und Teilen der Anlage selbst regulieren können. Dafür müssen laufend alle Sensoren eingelesen und ausgewertet werden. Es gibt einen manuellen und einen automatischen Betrieb der Anlage. Beim manuellen Betrieb kann jede Pumpe einzeln mit einem Knopf aktiviert bzw. deaktiviert werden. Dies ist besonders für die Testphase am Anfang sehr hilfreich, um das Wasser einfach in ein anderes Becken zu pumpen.

Bei beiden Betriebsarten ist auf dem LCD-Display eine kleine schematische Zeichnung der ganzen Anlage zu sehen. Diese zeigt an, welche Pumpe(n) gerade am Laufen ist und wie hoch der Wasserstand in den jeweiligen Bereichen ist.

5.1 Wasserstände

Um die vielen Wasserstände der insgesamt vier Bereiche der Anlage zu beschreiben, wird jedem Teil oder Becken der Aquaponik-Anlage ein Wert von null bis zwei zugeordnet. Diese Werte sind in einer Liste als folgende Wasserstände (WaSt) gespeichert:

```
typedef enum {  
    WaSt_LEER,  
    WaSt_NORMAL,  
    WaSt_MAX  
} WaSt;
```

Mit Hilfe dieser Zuordnung von "allgemeinen Wasserständen", die in allen Bereichen der Anlage gleich sind, ist es nun viel einfacher damit weiter zu arbeiten. Praktisch daran ist ebenfalls noch, dass so auch mehrere Ausgänge von Sensoren in einer Wasserstands-Bezeichnung (in einem Wert) beschrieben werden können. Wichtig zu erwähnen ist jedoch noch, dass nicht immer alle diese vier Wasserstände verwendet werden können/müssen, und sie bedeuten auch in jedem Becken wieder etwas anderes und stehen nicht für einen universalen Wasserstand, der überall genau gleich hoch ist.

5.1.1 Aquarium

Im Aquarium gibt es einen Schwimmsensor und einen Infrarot-Sensor. Da der Pegelsensor ganz oben am Becken sitzt, bestimmt dieser den **WaSt_MAX**, also denjenigen Wasserstand, bei welchem das Aquarium komplett voll ist. Da durch die Verwendung des LCD-Shields die noch freien I/O Anschlüsse des Arduino UNO stark reduziert sind, soll versucht werden, möglichst wenige zu verwenden. Um das möglich zu machen, wurde Folgendes herausgefunden: Wenn man den Pegelsensor in Reihe vor den Minus-Pol des IR-Sensors schaltet, und diesen aktiviert, gibt der Infrarot-Sensor einen relativ hohen analogen Wert aus. Die Begründung dazu fehlt leider, aber es war einfach ein sehr praktischer, wenn nicht ganz legitimer Weg, die zwei Sensoren einfach zu verknüpfen. Nun kann man also bereits anhand eines sehr hohen Wertes (größer als 900) den **WaSt_MAX** bestimmen. Der IR-Sensor ist für zwei

weitere Wasserstände verantwortlich: Ist sein analoger Output (Spannung zwischen 0-1024V) kleiner als 435, wird der Wasserstand vom Aquarium auf **WaSt_LEER** gesetzt. Alles dazwischen ist der **WaSt_Normal**. So weit so gut, aber es gibt noch ein kleines Problem: Nehmen wir an, im Aquarium hat es genau so viel Wasser, dass der IR-Sensor ein Signal der Stärke 435 ausgibt. Hat es aber nur ein paar Wellen oder wird nur etwas Wasser ins Aquarium gepumpt, ist der Wert sofort höher (oder auch schnell wieder tiefer) als diese 435. Das würde bedeuten, dass sehr schnell zwischen den Wasserständen LEER und NORMAL gewechselt wird, was äusserst unpraktisch für ein Programm ist. Die Benutzung einer Hysterese bedeutet hier eine schnelle und effiziente Lösung dieser Problematik, denn mit einer solcher ist es möglich, die Wechsel der Wasserstände zu verzögern oder gar nicht erst (oder noch nicht) eintreten zu lassen. Im Falle des Aquariums kommen damit folgende if-Bedingungen zustande (Wasserstand_AQ ist eine Variabel, die den analogen Input Wert speichert):

```
if (Wasserstand_AQ > 900) {
    WaSt_AQ = WaSt_MAX;
}
else if (Wasserstand_AQ < 435) {
    WaSt_AQ = WaSt_LEER;
}
else if ((Wasserstand_AQ > 470) && (Wasserstand_AQ < 500)) {
    WaSt_AQ = WaSt_NORMAL;
}
```

Das bedeutet, nur wenn der Wert zwischen 470 und 500 liegt, wird der **WaSt_NORMAL** gesetzt. Ansonsten bleibt einfach der vorherige Wert weiter bestehen, denn es gibt jetzt Bereiche, die gar nicht definiert sind (von 435 bis 470 und von 500 bis 900 → Hysterese).

5.1.2 NFT-System

Den Wasserstand in der Regenrinne zu bestimmen ist sehr einfach, denn man muss nur den **WaSt_MAX** kennen. Dafür gibt es ganz oben einen Bodenfeuchtigkeits-Sensor, welcher mit nur sehr wenig Kontakt mit dem Wasser einen sehr hohen analogen Wert ausgibt. Und auch hier muss eine Lücke (zwischen 0 und 550) bei den Bedingungen zwischen den Wasserständen MAX und NORMAL existieren: (Wasserstand_REGEN ist analoger Eingang vom Sensor im NFT)

```
if (Wasserstand_REGEN > 550) {
    WaSt_REGEN = WaSt_MAX;
}
else if (Wasserstand_REGEN < 1) {
    WaSt_REGEN = WaSt_NORMAL;
}
```

5.1.3 Ebbe- & Flut

Anders als beim vorherigen System, müssen hier zwei separate Signale aus zwei Schwimmsensoren verarbeitet werden. Das ist aber sehr einfach, da beide Sensoren ein digitales Signal (also 0 oder 1) ausgeben. So reichen nur zwei boolesche Bedingungen, um zwischen den Wasserständen LEER, NORMAL und MAX zu unterscheiden:

```
if (Wasserstand_WANNE_OBEN) {
    WaSt_WANNE = WaSt_MAX;
}
else if (Wasserstand_WANNE_UNTEN) {
    WaSt_WANNE = WaSt_NORMAL;
}
else {
    WaSt_WANNE = WaSt_LEER;
}
```

- Wenn also der obere Sensor (Wasserstand_WANNE_OBEN) aktiviert wird, ist die Wanne (Ebbe- & Flut) voll → WaSt_MAX
- Wenn der obere Sensor nicht aktiviert ist, der untere aber schon, dann ist das Wasser irgendwo dazwischen → WaSt_NORMAL
- Wenn gar kein Schwimmkörper angehoben ist, ist das Wasser zu tief → WaSt_LEER

Wie auch bei den Sensoren im Aquarium braucht es hier irgendeine Form einer Hysterese, da sich die Schwimm-Pegel-Sensoren im Wasser ständig bewegen und so während einem Übergang in den nächsten Wasserstand die Information schnell schwankt und man ein bisschen Puffer zwischen den verschiedenen Levels haben will. Dafür wird eine einfache Timer-Library⁴ aus

⁴muTimer, <https://www.arduino.cc/reference/en/libraries/mutimer/>, (abgerufen am 28.12.2021)

dem Internet verwendet. Mit dieser kann man eine Ein- oder Ausschaltverzögerung der Sensoren programmieren. Das heisst konkret, dass die Information einer Aktivierung eines Schwimmsensors, erst nach zum Beispiel 20 Sekunden weitergegeben wird und so genug Zeit bleibt, das Becken sicher weit genug zu füllen, dass der Wert nicht mehr schwanken kann.

5.1.4 Überlaufbecken

Da das Überlaufbecken von den Sensoren her gleich aufgebaut ist wie die Wanne des Ebbe- & Flutsystems, funktioniert auch deren Auswertung genau gleich wie oben beschrieben.

5.2 Display

Das Ziel für die Verwendung von einem LCD-Display für die Steuerung der Aquaponik-Anlage ist sehr simpel. Besonders in der Entwicklungsphase von Anlage und Programm ist es sehr praktisch, auf einen Blick den aktuellen Zustand der gesamten Anlage ablesen zu können und auch später ist es einfach schön, visuell noch etwas von der Funktion der Anlage zu sehen.

Das Display ist aus zwei Linien aufgebaut mit je 16 Zeichenfeldern, die wiederum aus 5 x 8 Pixeln bestehen. Auf diesem kleinen Bildschirm sollen alle vier Bereiche der Anlage, deren Wasserstände, die Pumpen und der Vermerk, ob die Anlage im Automatik- oder manuellen Modus ist, angezeigt werden. Dabei werden die laufenden Pumpen mit Pfeilen in die Richtung des fließenden Wassers gezeichnet und die Wasserstände werden mit Hilfe von mehr oder weniger ausgefüllten Felder neben dem entsprechendem System der Anlage angezeigt (siehe Abb. 16). Im Beispiel in Abbildung 16 läuft im linken



Abb. 16 Beispiele vom LCD-Display des Arduino mit verschiedenen Zuständen der Anlage. (W = Ebbe- und Flutsystem (Wanne), AQ = Aquarium, R = NFT-System (Regenrinne), UB = Überlaufbecken)

Bild die Pumpe, die Wasser aus dem Aquarium in das Ebbe- & Flutsystem (Wanne) transportiert und laut dem rechten Bild, ist die Regenrinne (NFT-System) voll mit Wasser. Des Weiteren befindet sich die Anlage links im manuellen, rechts im automatischen Betrieb, welcher noch 17 Minuten

andauert (weitere Informationen dazu siehe weiter unten). Für diese Art der Wasserstands-Anzeige wurden selbst Zeichen für das LCD-Display pixelweise definiert. Zum Beispiel steht folgendes Zeichen für den Wasserstand vom Aquarium im rechten Bild in Abbildung 16:

```
byte Wasserstand_Beispiel[8] ={
    B11111,
    B10001,
    B10001,
    B11111,
    B11111,
    B11111,
    B11111,
    B11111,
    B11111,
};
```

Das alles ist aber nur mit der Library *LiquidCrystal*¹ für das LCD-Display möglich.

5.3 Automatik-Programm

Der automatische Betrieb hat zwei Hauptaufgaben: Kein Becken soll überlaufen, beziehungsweise kein Becken darf zu wenig Wasser haben, damit keine Pumpe trocken läuft, und in regelmässigen Abständen soll das Ebbe- & Flutsystem mit Wasser gefüllt werden.

5.3.1 Notfall-Massnahmen

In jedem Bereich der Anlage ist es sehr wichtig, dass nichts Unerwartetes passiert und die Wasserstände immer bekannt sind. Dafür gibt es mehrere Spezifikationen, welche alles überschreiben können und somit an erster Stelle im Programm stehen:

1. Das NFT-System hat nur einen Sensor und es muss auch nur darauf geachtet werden, dass die Regenrinne nicht überläuft.

```
    if (WaSt_REGEN == WaSt_MAX) {
        filterRechts = false;
    }
```

¹LiquidCrystal, <https://www.arduino.cc/en/Reference/LiquidCrystal>, (abgerufen am 28.12.2021)

Wenn also der Wasserstand in jener zu hoch ist, dann wird die Pumpe im Aquarium (filterRechts) ausgeschaltet.

2. Auch im Aquarium muss darauf geachtet werden, dass dieses nicht überläuft, aber auch nicht zu wenig Wasser hat.

```
if (WaSt_AQ == WaSt_LEER) {
    filterLinks = false;
    filterRechts = false;
}
if (WaSt_AQ == WaSt_MAX) {
    filterAQ = false;
}
```

Wenn aber trotzdem der Wasserstand im Aquarium so stark sinken sollte, müssen die zwei Pumpen, die Wasser aus dem Aquarium befördern, abgeschaltet werden.

3. In der Wanne des Ebbe- & Flutsystems ist für die Notfall-Massnahmen nur der maximale Wasserstand interessant. Für was der untere Sensor gebraucht wird, wird im nächsten Kapitel beschrieben.

```
if (WaSt_WANNE == WaSt_MAX) {
    filterLinks = false;
}
```

Und auch diese wichtige Handlung für das Ebbe- & Flutsystem ist sehr einfach, denn es muss, wenn der Wasserstand zu hoch ist, die Pumpe, welche Wasser in die Wanne pumpt (filterLinks), abgeschaltet werden.

4. Im Überlaufbecken gibt es oben und unten einen Sensor, die den höchsten und niedrigsten Wasserstand markieren, der herrschen darf.

```
if (WaSt_UEBERLAUF == WaSt_MAX) {
    filterRechts = false;
    filterLinks = false;
    filterAQ = true;
}
if (WaSt_UEBERLAUF == WaSt_LEER){
    filterAQ = false;
}
```

Dabei ist noch wichtig zu wissen, dass filterLinks und filterRechts jeweils die Pumpen für das Ebbe- & Flut beziehungsweise für das NFT-System sind, und filterAQ ist die Pumpe, die das Wasser vom Überlaufbecken ins Aquarium pumpt. Wie man im oberen Code-Ausschnitt erkennen kann, muss die Aquarium Pumpe abgeschaltet werden, wenn der Wasserstand im Überlauf zu gering ist, sie muss aber eingeschaltet werden, wenn es zu viel Wasser in diesem Becken hat. Auch die Pumpen für die beiden Pflanzen-Systeme schalten ab, wenn zu viel Wasser im Überlaufbecken ist.

5.3.2 Die Ebbe und die Flut

Die meiste Zeit hat die Anlage lediglich die Aufgabe, Wasser ins NFT-System und wieder zurück ins Aquarium zu pumpen. Dieser Wasserkreislauf soll eigentlich die ganze Zeit ungestört laufen können. Doch in regelmässigen Abständen, soll auch die Wanne des Ebbe- & Flutsystems mit Wasser geflutet werden, was mit ein paar weiteren Aufgaben verbunden ist. Die Abbildung 17 soll diese eher komplexen Abläufe für die Ebbe und die Flut in einer sogenannten Zustandsmaschine veranschaulichen.

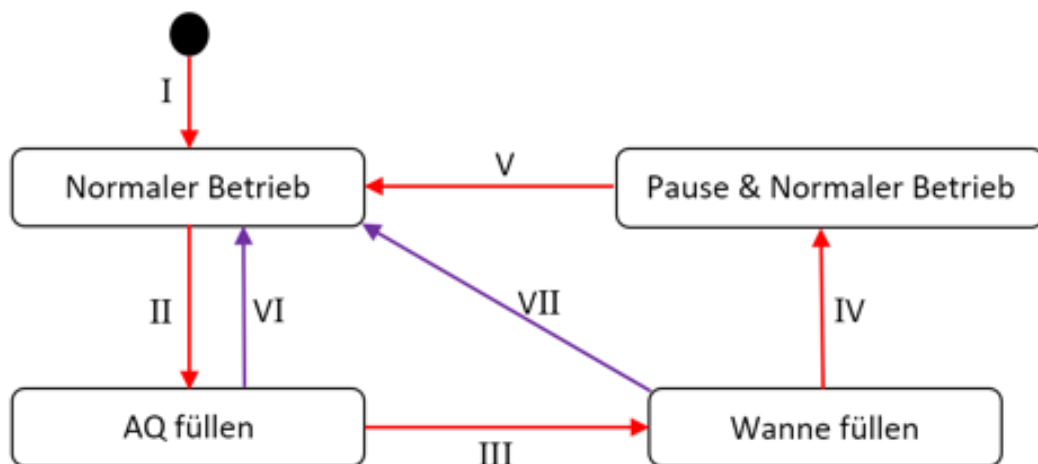


Abb. 17 Zustandsmaschine während dem automatischen Betrieb

- I Gestartet wird das ganze Automatik-Programm mit dem normalen Betrieb. Bei dieser Betriebsart passiert beim Ebbe- & Flutsystem nichts und nur die Regenrinne wird bewässert. Natürlich wird auch wieder Wasser zurück ins Aquarium gepumpt und die Notfall-Massnahmen

gelten immer noch. Zusätzlich wird ein Timer (1) von 45 Minuten gestartet, welcher auf dem LCD-Display ausgegeben wird.

- II** Wenn der Timer (1) vom normalen Betrieb abläuft, geht das Programm in die zweite Phase über, nämlich in das Aquarium füllen. Hier werden alle Pumpen, bis auf die Aquarium-Pumpe, ausgeschaltet und das Aquarium wird komplett mit Wasser gefüllt. Im Hintergrund wird noch ein Timer (2) von 15 Minuten gestartet, welcher später noch wichtig sein wird.
- III** Wenn das Aquarium voll ist, kann die nächste Aktion ausgeführt werden: die Wanne (Ebbe- & Flutsystem) füllen. Das ist relativ selbsterklärend: Die eine Zimmerbrunnenpumpe im Aquarium beginnt die Wanne aufzufüllen, während gleichzeitig die Aquarium-Pumpe für genügend Wasser im Aquarium sorgt. Dieser Vorgang dauert etwa 10 Minuten, da zwischendurch das Befüllen der Wanne abgebrochen wird, weil die Aquarium-Pumpe eine geringere Leistung hat als die Zimmerbrunnen-Pumpe. Der Timer (2), welcher in der vorherigen Phase gestartet wurde, läuft hier immer noch weiter.
- IV** Wenn es aber gelungen ist, die Wanne zu füllen (\rightarrow Flut), fällt die Anlage wieder in den normalen Betrieb (NFT-System wird bewässert) zurück, aber mit einer kleinen Änderung: Bei diesem Betrieb, weiss man nicht, wie lange er dauern wird, da hier gewartet wird, bis sich die Wanne wieder entleert hat. Im Normalfall dauert das Abfließen des Wassers (\rightarrow Ebbe) aus dem Ebbe- & Flutsystem etwa 20 Minuten.
- V** Sobald dann das ganze Wasser wieder abgeflossen ist, also der untere Sensor in der Wanne nicht mehr aktiviert wird, springt das System in den richtigen normalen Betrieb am Anfang zurück und auch der Timer (1) von 45 Minuten wird zurückgesetzt und neu gestartet.

Der gerade beschriebene Ablauf (I-V) ist die gewünschte Reihenfolge und sollte eigentlich immer so funktionieren. Es kann aber auch sein, dass das Aquarium oder die Wanne aus irgendeinem Grund (z.B. wegen zu wenig Wasser, Verstopfung, ...) nicht gefüllt werden können und dann ist es sinnvoll, diese Aktionen abubrechen.

- VI** Der Timer (2) spielt für diese Kontrolle eine wichtige Rolle. Er schaut, dass die Phasen Aquarium und Wanne füllen zusammen nicht länger als 15 Minuten dauern. Wenn das System vergeblich versuchen sollte, das Aquarium bis zum WaSt_MAX zu füllen, springt das Programm

nach 15 Minuten automatisch wieder in den normalen Betrieb zurück und bleibt dort logischerweise wieder 45 Minuten, bevor die Flut erneut versucht wird.

- VII** Genau so wie beim Befüllen des Aquariums, würde der Timer (2) auch beim Füllen der Wanne einschreiten und dieses mit dem normalen Betrieb unterbrechen, wenn das Ebbe- & Flutsystem nicht gefüllt werden kann.

6 Betrieb der Anlage

Die fertige Anlage läuft seit dem 17.12.2021 praktisch fehlerfrei im automatischen Betrieb. Es gab auf dem Weg dahin aber ein paar Probleme, die die Weiterentwicklung verzögerten.

6.1 Probleme

Bei einer Anlage, die eigentlich nur Wasser transportiert, ist es sehr wichtig, dass alles dicht ist. So gab es vor allem am Anfang immer wieder kleinere Überschwemmungen des NFT-Systems, da es sehr schwierig war, die Kanten genau abzudichten. Zuerst wurde es mit Silikon versucht (Abbildung 18), was jedoch am Ausschnitt der Löcher in den Enden scheiterte und ganz zum Schluss wurde alles mit Epoxitharz-Kleber zugeklebt.

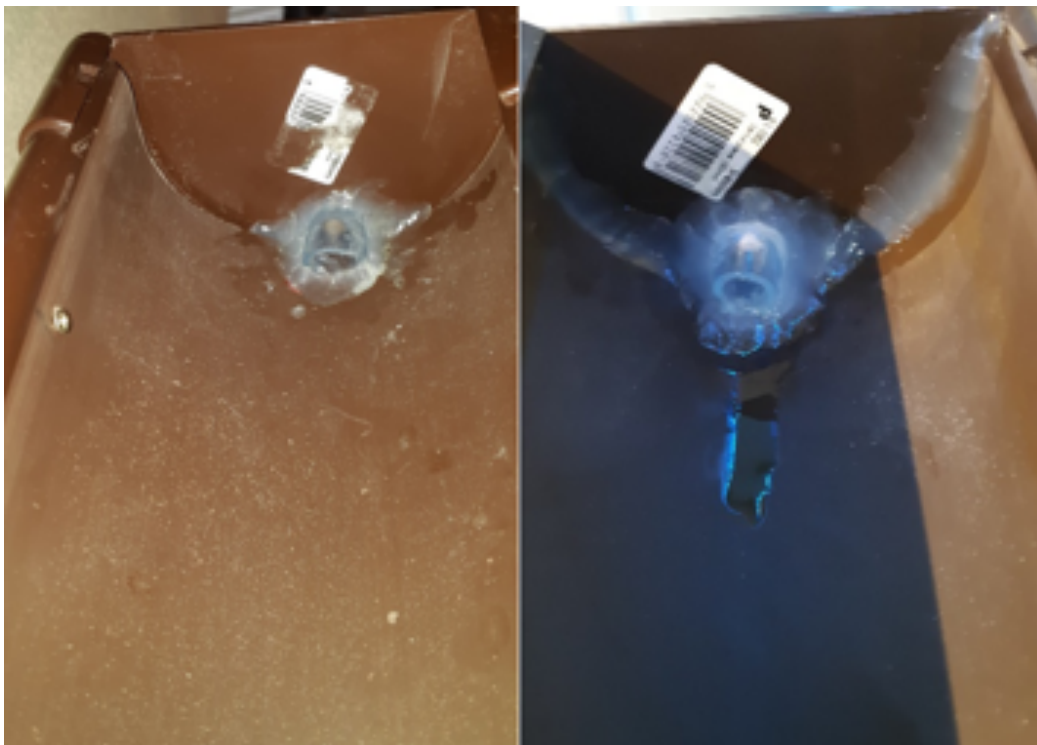


Abb. 18 Abdichtungsversuche der Regenrinne vom NFT-System

Eher unerwartet führten die analogen Wasserstands-Sensoren der Firma Maker Factory zu sehr grossen Problemen, gefolgt von Zeit-Verzögerungen. Diese Sensoren sind kleine Plättchen aus Kunststoff, in welche dünne Metallstreifen eingelassen sind. Diese sind dann abwechselnd an den Plus- und

Minuspol angeschlossen und das Wasser kann dann so den Stromkreislauf schliessen. Die Fläche der Metallstreifen im Wasser bestimmt also, wie viel Strom geleitet wird und der analoge Ausgang ist grösser oder kleiner. Diese Messmethode hängt also auch sehr mit der Leitfähigkeit des Wassers zusammen und ist deshalb in einem biologischen Aquarium ungeeignet, da sich dort die Wasserwerte immer wieder ändern können. Dieses Problem hätte man mit ein bisschen mehr Code umgehen können, aber da kam noch Folgendes dazu:

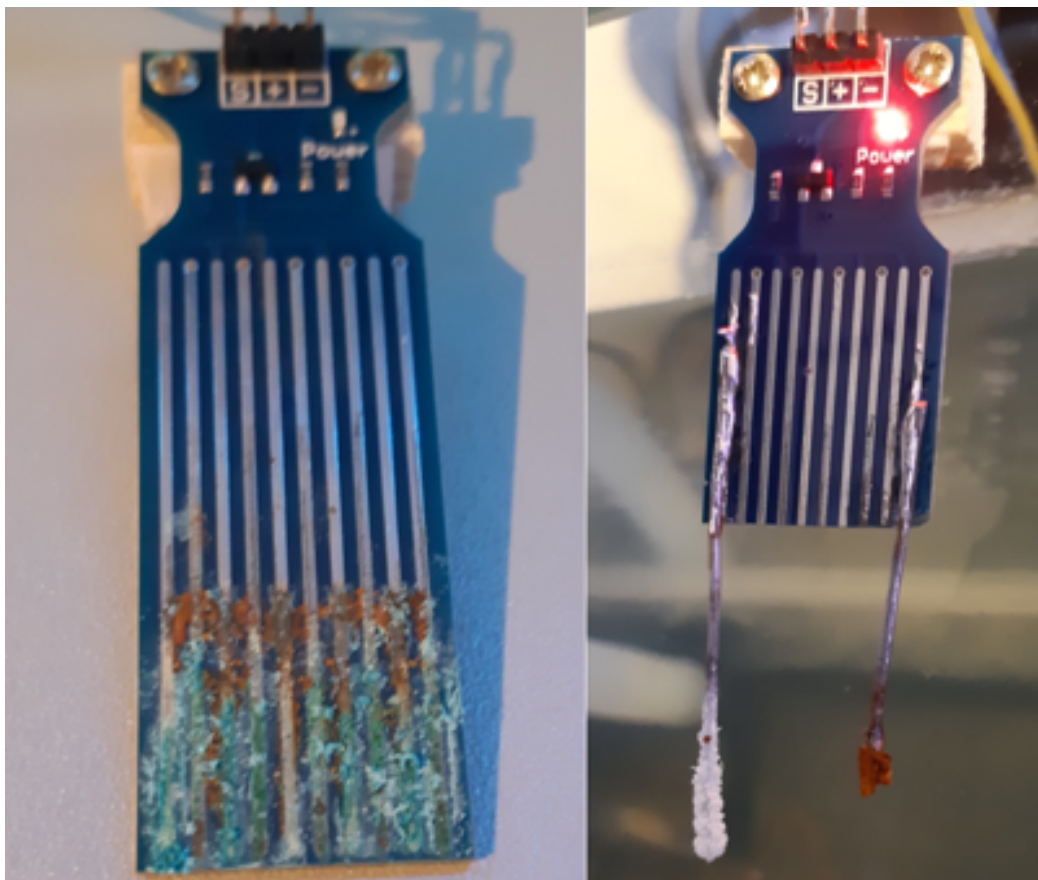


Abb. 19 Wasserstands-Sensoren (Maker Factory) nach 2 Tagen Dauereinsatz. Rechts wurden zwei der Metallstreifen verlängert, damit der Kunststoff das Wasser nicht mehr berührt.

Aus Gründen der Chemie (Galvanisierung) fand an den beiden verschiedenen Metallen eine chemische Oxidation beziehungsweise eine Reduktion statt, die eine der Elektroden schnell auflöste. Auch der Plastik dahinter verfärbte schnell das Wasser blau und bereits nach 2-3 Tagen, konnte man die Sensoren nicht mehr brauchen.

Dies ist sofern unerklärlich, weil diese Sensoren für den Gebrauch in Wasser gedacht sind und das Wasser in der Anlage einen neutralen pH-Wert hat.

Diese Art der Messmethode war eigentlich im Aquarium und in der Wanne (Ebbe- & Flut-System) gedacht. Besonders im Aquarium belastete dieses Problem die weitere Entwicklung der ganzen Anlage sehr, denn wenn der Wasserstand im Aquarium (wichtigster Teil der Anlage) nicht richtig gemessen werden kann, ist der Betrieb der Anlage unmöglich. Schlussendlich wurde dann auf diese Plättchen verzichtet und nur noch zu den sehr zuverlässigen digitalen Schwimmsensoren und einem Infrarot-Sensor gegriffen.

In der Regenrinne (NFT-System) befindet sich ein Bodenfeuchtigkeits-Sensor der selben Marke, welcher sich ebenfalls langsam auflöst. Früher oder später wird also auch dieses Plättchen ersetzt werden müssen.



Abb. 20 Höhenunterschied vom Überlaufbecken (links) und dem Aquarium (rechts)

Zwischen dem Überlaufbecken und dem Aquarium befindet sich eine Höhendifferenz von 30 Zentimetern (Abbildung 20). Aufgrund von diesem Hö-

henunterschied ist es möglich, dass das Wasser automatisch vom Aquarium ins Überlaufbecken zurück fließen kann, wenn der Filter nicht läuft. Dies geht dann so lange, bis der Auslauf oben im Aquarium Luft einsaugt, was dann nachher beim Einschalten der Pumpe zu weiteren Problemen führt. Das Erhöhen der Position des Überlaufbeckens hatte nicht den gewünschten Effekt zur Folge, deshalb musste zu einem Rückschlag-Ventil gegriffen werden, welches sich im Schlauch von AQ-Filter zum Aquarium befindet. Dieses verhindert das Zurücklaufen des Wassers, verlangsamt aber den Wasserfluss enorm, was eine geringere Förderleistung und Strömung bei den Fischen zur Folge hat. Damit der Aquarium-Filter nicht an seine Grenzen kommt, wird er am Einsaugschlauch von einer dritten Zimmerbrunnenpumpe unterstützt.

6.2 Pflanzen und Fische

Aufgrund der ungünstigen Jahreszeit (Winter), konnten keine richtigen Nutzpflanzen gekauft werden. Deshalb wurden im NFT-System selbst verschiedenste Pflanzen angepflanzt, um zu schauen, was alles gedeiht.



Abb. 21 Pflanzen im NFT-Systeme. v.l.n.r.: "Lollo Rossa"-Salat, Basilikum, Buschbohne, Unbekannte Ableger einer Pflanze, Salat (mit Wurzelblock aus Coop), Gartenkresse

- Lollo-Salat: Samen keimten, wurden aber nicht grösser als 5 cm.
- Basilikum: Das ist ein Ableger einer grösseren Pflanze. In der Anlage ist er minimal gewachsen, hat aber ein starkes Wurzelwerk gebildet. Von der selben Pflanze ist auch ein Ableger im Ebbe- & Flutsystem (siehe weiter unten).
- Buschbohnen: Wachsen innerhalb von einer Woche ca. 20 cm hoch.
- Unbekannte Pflanze: Das sind kleine Ableger von den Blättern einer (leider) unbekannt Pflanze. Sie konnten im NFT-System ihre Wurzeln vergrössern.

- Salat mit Wurzeln aus Coop: Diesen Salat konnte man schon gross kaufen, aber die Wurzeln und Erde waren noch vorhanden. Das Umtopfen funktionierte nur bedingt und alle Blätter welkten/gingen nach einer Woche kaputt.
- Gartenkresse: Wuchs sehr schnell und gross

Alle oben genannten Pflanzen sind in den zwei Wochen mehr oder weniger gut gewachsen, aber es gab auch Pflanzen-Samen, die gar nicht keimten. So wuchs zum Beispiel Lauch oder Petersilie gar nicht. Dies liegt nicht unbedingt an der Pflanze oder der Anlage, sondern einfach an der Tatsache, dass die Samen teilweise schon 18 Jahre alt waren.



Abb. 22 Pflanzen im Ebbe- & Flutsystem. Links und hinten: Zimmerpflanzen, Mitte: Basilikum, rechts vorne: Salat aus Coop, rechts hinten: Buschbohne

Im Ebbe- & Flutsystem braucht es bereits grössere Pflanzen mit Wurzelwerk. Ausser Salat aus dem Coop (wie oben erklärt) gab es dafür nichts Passendes zu kaufen. Deshalb wurden einfache Arten von Zimmerpflanzen verwendet, welche dann auch sehr gut in die Tiefe und Höhe wuchsen. Der

Salat ging wie im NFT-system sehr schnell kaputt, was bedeutet, dass diese Salate aus der Gemüseabteilung vom Coop nicht zum weiteren Wachsen geeignet sind.

Wie im NFT-System, wurden in das Ebbe- & Flutsystem eine bereits grosse Buschbohne und der Basilikum eingepflanzt, welche dann auch beide sehr gut gewachsen sind. Ganz zum Schluss wurde über beiden Systemen eine Lampe für künstliches Licht installiert, da sich die Anlage in einer eher dunkleren Ecke des Zimmers befindet.



Abb. 23 Fertig bepflanzte Beete mit künstlichem Licht.

Das Aquarium wurde dann noch mit Kies gefüllt und nicht sehr dicht mit langsam- & schnellwachsenden Pflanzen bepflanzte. Als Fische kamen ein kleiner Schwarm von 15 Esbes Bärblingen und 5 Pandapanzerwelse zum Einsatz. Diese haben sich alle relativ gut eingelebt und es gab bis jetzt noch keine Verluste.

Abb. 24 Fertiges Aquarium (27.12.2021) mit vier verschiedenen Pflanzenarten (oben), 5 Pandapanzerwelse (unten rechts) und 15 Espos Bärblingen (unten links):



7 Fazit

Das Projekt ist am 30.12.2021 offiziell beendet worden und ich bin mit dem Erreichten mehr als zufrieden. Ich lernte nicht nur sehr viel mehr über das Programmieren, sondern konnte mich auch intensiv mit Pflanzen und Tieren auseinander setzen und so bestimmt etwas für das spätere Leben lernen.

7.1 Zielerreichung

Das Endergebnis ist eine funktionsfähige Aquaponik-Anlage mit Fischen und Pflanzen. Alle Bewässerungsmethoden für die Pflanzen und der gesamte Wasserkreislauf funktionieren ohne Probleme, Wasserstände können in allen Bereichen problemlos gemessen werden und es gibt sogar noch ein Display, welches einen Einblick in die Funktionsweise der Anlage gibt. Von der Seite der Pumpen, Pflanzen und Wasserstände ist das Ziel komplett erreicht, einzig schade ist, dass es im Hydrokultur-Teil nicht viele Nutzpflanzen gibt. Trotzdem hat sich in den drei Wochen Betrieb ein biologisches Gleichgewicht eingestellt und den Fischen und Pflanzen scheint es gut zu gehen.

Wäre mehr Zeit zur Verfügung gestanden, hätte ich noch das Ziel gehabt, die Wasserwerte in der ganzen Anlage zu messen, zu kontrollieren und zu regulieren. Nun gibt es „leider“ nur eine Steuerungseinheit für die Pumpen und Wasserstände, aber auch so ist eine komplette Aquaponik-Anlage am Laufen und das mit den Wasserwerten wäre mehr noch ein „nice to have“.

7.2 Kosten

Im Folgenden sind gerundete oder auch geschätzte Preise der benötigten Teile für die Entwicklung der Aquaponik-Anlage aufgelistet, da nicht alles Material noch gekauft werden musste. Auch sind in der folgenden Tabelle nur diejenigen Teile aufgeführt, welche auch wirklich in der fertigen Anlage verwendet werden:

Produkt	Menge	Gesamtbetrag (CHF)
Zimmerbrunnenpumpe	3 Stk.	75
Schwimmerschalter	5 Stk.	60
Kaltgerätstecker	4 Stk.	24
Gehäuse	1 Stk.	10
AC/DC-Wandler	1 Stk.	20
LCD & Keypad Shield	1 Stk.	10
Relais	4 Stk.	4
Rückschlagventil	1 Stk.	8
IR-Sensor	1 Stk.	9
weitere elekt. Bauteile		5
Schläuche	6.5 Meter	20
Regenrinne	1 Meter	10
Becken	2 Stk.	30
Pflanzentöpfe	15 Stk.	15
Blähton	40 Liter	15
Baumaterialien (Holz, Leim,...)		15
Fische	15 Stk.	100
AQ-Pflanzen	4 Stk.	35
AQ-Kies	10 kg	10
AQ-Wasserzusätze		30

Mit Berücksichtigung aller Teile, die ich schon Zuhause hatte (z.B AQ-Filter, Heizung, Beleuchtung, CO2-Anlage...), kommt man auf einen Gesamtbetrag von etwa mindestens 600 Franken. Aber wie oben gesagt, ist dies nur eine grobe Schätzung, da es schwierig ist zu entscheiden und unterscheiden, welche Teile schon vorhanden waren und wie wichtig sie für die Anlage an sich sind.

7.3 Verbesserungen

Zu sagen, diese Anlage sei perfekt, ist wohl etwas übertrieben ausgedrückt, aber sie funktioniert genau so, wie sie soll. Man könnte nun noch weiter gehen, damit die Anlage wirklich den Sinn einer Aquaponik-Anlage im grossen Stil hat, in dem Zuchtfische und Nutzpflanzen verwendet werden. Der Massstab ist aber recht klein für so etwas, also müsste alles vergrössert werden. Sehr schön wäre es überdies auch noch (wie oben beschrieben), eine weitere Steuerung für alle Wasserwerte hinzuzufügen und so die Anlage komplett autonom zu gestalten. Dann wäre es vielleicht das Ziel, dass man die ganzen Pumpen und Sensoren via Handy auslesen und steuern kann und dann eine automatische und grüne Lebensmittelproduktion im eigenen Haus hat. Im

Falle einer Weiterentwicklung müsste die ganze Anlage aber aufgrund des Lärms, den sie produziert, und auch wegen dem Platzbedarf, aus meinem Zimmer verlegt werden.



Abb. 25 Komplette Anlage zum Abgabezeitpunkt der Arbeit (29.12.2021)

Literatur

- [Buch] *Der Aquaponik-Garten – Fische und Pflanzen in einem geschlossenen Kreislauf kultivieren*, Felix van Zyl, tredition Verlag, 2017
- [Buch] *Aquaponik im Eigenbau – Aquakultur und Hydroponik zum Anfassen, Mitmachen und Erleben*, Thomas Rücker, 2021
- [Web] Blogdarobotica: Shield LCD Display, <https://www.blogdarobotica.com/2021/05/12/como-utilizar-o-shield-display-16x02-lcd-com-teclado-no-arduino/> (abgerufen am 23.10.2021).
- [Web] Conrad, <https://www.conrad.ch/de/p/oase-aquarius-universal-440i-zimmerbrunnenpumpe-440-l-h-0-75-m-572138.html>, (abgerufen am 14.12.2021)
- [Web] Aquaristikshop, <https://www.aquaristikshop.com/aquaristik/Dennerle-Scapers-Flow-Anhaengefilter/908039/>, abgerufen am 14.12.2021)
- [Web] DatasheetsPDF, <https://datasheetspdf.com/datasheet/GP2Y0A41SK0F.html>, abgerufen am 14.12.2021
- [Web] Library "muTimer", <https://www.arduino.cc/reference/en/libraries/mutimer/>, abgerufen am 28.12.2021
- [Web] Library "LiquidCrystal", <https://www.arduino.cc/en/Reference/LiquidCrystal>, abgerufen am 28.12.2021
- [Web] Wikipedia, Schutzdiode, <https://de.wikipedia.org/wiki/Schutzdiode>, abgerufen am 29.12.2021

Abbildungsverzeichnis

1	Fertige Aquaponik-Anlage (Stand 17.12.2021)	1
2	Erster grober schematischer Aufbau der Aquaponik-Anlage	4
3	Vollständig ausgerüstetes Aquarium mit 2 Zimmerbrunnenpumpen.	6
4	Dennerle Scaper's Flow Aussenfilter (links), Oase Aquarius Universal 440i Zimmerbrunnenpumpe (rechts) ¹	7
5	Infrarot-Sensor und Pegelsensor am Aquarium	8
6	Sharp IR-Sensor, Analoger Output verglichen mit Distanz ¹	9
7	Wasser-Durchlauf-Verminderer	10
8	NFT-System (Regenrinne) mit Wasserstands-Sensor und ersten kleinen Pflanzen	10
9	Ebbe- & Flutsystem während Bau von Innen	11
10	Fertiges Ebbe- & Flutsystem mit ersten Zimmerpflanzen	12
11	Überlaufbecken, bereits mit Wasser (und Dreck) gefüllt. Rechts hängt der Aquarium-Aussenfilter.	13
12	Arduino UNO mit LCD Keypad Shield ¹	13
13	Fertige Steuerung der Aquaponik-Anlage (links: Innenansicht, rechts: Aussenansicht)	14
14	Elektronisches Schema der Steuerungseinheit (gezeichnet mit fritzing).	15
15	Rückseite der Platine in der Steuerungseinheit. Blau umrandet: 230V Stromkreis	16
16	Beispiele vom LCD-Display des Arduino mit verschiedenen Zuständen der Anlage. (W = Ebbe- und Flutsystem (Wanne), AQ = Aquarium, R = NFT-System (Regenrinne), UB = Überlaufbecken)	20
17	Zustandsmaschine während dem automatischen Betrieb	23
18	Abdichtungsversuche der Regenrinne vom NFT-System	26
19	Wasserstands-Sensoren (Maker Factory) nach 2 Tagen Dauereinsatz. Rechts wurden zwei der Metallstreifen verlängert, damit der Kunststoff das Wasser nicht mehr berührt.	27
20	Höhenunterschied vom Überlaufbecken (links) und dem Aquarium (rechts)	28
21	Pflanzen im NFT-Systeme. v.l.n.r.: "Lollo RossaSalat, Basilikum, Buschbohne, Unbekannte Ableger einer Pflanze, Salat (mit Wurzelblock aus Coop), Gartenkresse	29
22	Pflanzen im Ebbe- & Flutsystem. Links und hinten: Zimmerpflanzen, Mitte: Basilikum, rechts vorne: Salat aus Coop, rechts hinten: Buschbohne	30

23	Fertig bepflanzte Beete mit künstlichem Licht.	31
24	Fertiges Aquarium (27.12.2021) mit vier verschiedenen Pflanzenarten (oben), 5 Pandapanzerwelse (unten rechts) und 15 Espos Bärblingen (unten links):	32
25	Komplette Anlage zum Abgabezeitpunkt der Arbeit (29.12.2021)	35

Anhang

Hauptprogramm

```
1  #include "keyboard.h"
2  #include "lcd.h"
3  #include <muTimer.h> //library für timer Funktionen
4
5  //Ausgänge:
6  // Pin 12 ist erster Stecker von rechts (von oben) gesehen
7  const int PIN_FILTER_RECHTS = 12;
8  // Pin 11 ist 2ter Stecker von rechts (von oben) gesehen
9  const int PIN_FILTER_LINKS = 11;
10 // Pin 3 ist dritter Stecker von rechts (von oben) gesehen
11 const int PIN_FILTER_AQ = 3;
12 // Pin 2 ist der linke Stecker von oben gesehen
13 const int frei = 2;
14 // Pin 2 ist der linke Stecker von oben gesehen
15
16 //Eingänge:
17 //Sensoren werden auf die entsprechenden Eingänge definiert
18 const int PIN_SENSOR_AQUARIUM = A1;
19 const int PIN_SENSOR_REGEN = A2;
20 const int PIN_PEGEL_WANNE_OBEN = A5;
21 const int PIN_PEGEL_WANNE_UNTEN = 13;
22 const int PIN_PEGEL_UEBERL_OBEN = A4;
23 const int PIN_PEGEL_UEBERL_UNTEN = A3;
24
25
26 typedef enum { //Liste mit allen möglichen Wasserständen
27     WaSt_LEER,
28     WaSt_NORMAL,
29     WaSt_MAX
30 } WaSt; //Diese Liste heisst WaSt
31
32 //Zu Beginn werden alle Teile der Anlage auf WaSt_LEER gesetzt
33 WaSt WaSt_AQ = WaSt_LEER;
34 WaSt WaSt_REGEN = WaSt_LEER;
35 WaSt WaSt_WANNE = WaSt_LEER;
36 WaSt WaSt_UEBERLAUF = WaSt_LEER;
37
38 //Bool Variablen für die drei Pumpen der Anlage, um On und Off zu steuern
39 bool filterRechts = false; //Pumpe für Regen-Rinne
40 bool filterLinks = false; //Pumpe für Wanne
41 bool filterAQ = false; //Pumpe für AQ
42
43 //Timer zur Verzögerung von verschiedenen Aktionen oder Sensoren:
44 muTimer timerUeberlaufUnten = muTimer();
45 muTimer timerUeberlaufOben = muTimer();
46 muTimer timerWanneUnten = muTimer();
47 muTimer timerWanneOben = muTimer();
48 muTimer timerAQ = muTimer();
49 muTimer WanneTimer = muTimer();
50 //Delay Verzögerungs-Zeiten:
```

```

51  const uint32_t DELAY_UEBERLAUF_UNTEN = 10000;
52  const uint32_t DELAY_UEBERLAUF_OBEN = 20000;
53  const uint32_t DELAY_WANNE_UNTEN = 20000;
54  const uint32_t DELAY_WANNE_OBEN = 10000;
55  const uint32_t DELAY_AQ_OFF = 20000;
56  const uint32_t DELAY_AQ_ON = 5000;
57  //Zeit im normalen Betrieb (Pausen zwischen dem Füllen)
58  const uint32_t ZEIT_NORMAL_BETRIEB = 1800000;
59  //Maximale Zeit zum Füllen der Wanne und Aquarium
60  const uint32_t TIMEOUT_FUELLEN = 900000;
61
62  //restliche Zeit einer Betriebsart gespeichert
63  int32_t RestZeitBetriebsart = 0;
64  //Boolean, um zwischen den Betriebsarten Auto und Manuell zu wechseln
65  bool Betriebsart_AUTOMATIK = false;
66  //Für Mittelwertbestimmung für IR-Sensor:
67  const float ProzentAnteilMittelwert = 0.05;
68
69  void setup() {
70      //Ausgänge:
71      // Setzt den Digitalpin 12 als Outputpin
72      pinMode(PIN_FILTER_RECHTS, OUTPUT);
73      // Setzt den Digitalpin 11 als Outputpin
74      pinMode (PIN_FILTER_LINKS, OUTPUT);
75      // Setzt den Digitalpin 2 als Outputpin
76      pinMode(frei, OUTPUT);
77      // Setzt den Digitalpin 3 als Outputpin
78      pinMode(PIN_FILTER_AQ, OUTPUT);
79      //Eingänge:
80      //Setzte den Analogpin 1 als Inputpin
81      pinMode(PIN_SENSOR_AQUARIUM, INPUT);
82      //Setzte den Analogpin 2 als Inputpin
83      pinMode(PIN_SENSOR_REGEN, INPUT);
84      //Setzte den Digitalpin 5 als Inputpin mit Pullup-Funktion
85      pinMode(PIN_PEGEL_WANNE_OBEN, INPUT_PULLUP);
86      //Setzte den Digitalpin 13 als Inputpin mit Pullup-Funktion
87      pinMode(PIN_PEGEL_WANNE_UNTEN, INPUT_PULLUP);
88      //Setzte den Analogpin 4 als Inputpin mit Pullup-Funktion
89      pinMode(PIN_PEGEL_UEBERL_OBEN, INPUT_PULLUP);
90      //Setzte den Analogpin 3 als Inputpin mit Pullup-Funktion
91      pinMode(PIN_PEGEL_UEBERL_UNTEN, INPUT_PULLUP);
92
93      setupLCD(); //LCD-Display wird konfiguriert
94  }
95
96  void sensorenAuswerten (){
97      //Sensor-Signale werden in Variablen gespeichert:
98      // Startwert für Mittelwert-Bestimmung:
99      static float Wasserstand_AQ = 0.0;
100     //Mittelwert (mit Prozentanteilen) vom IR-Sensor im AQ ist neuer Wasserstand
101     Wasserstand_AQ = Wasserstand_AQ*(1.0-ProzentAnteilMittelwert)+ProzentAnteilMittelwert*
102     (analogRead(PIN_SENSOR_AQUARIUM));
103     //Wasserstand der Regenrinne aus analogRead vom Sensor wird in Variabel gespeichert
104     int Wasserstand_REGEN = analogRead(PIN_SENSOR_REGEN);
105     /*Wasserstand Überlauf Oben aus digitalRead wird in Variabel gespeichert mit Verzögerung,
106     (Delay) um zu schnelles Wechseln des Wertes zu verhindern*/

```

```

107  bool Wasserstand_UEBERLAUF_OBEN = timerUeberlaufOben.delayOff(digitalRead
108      (PIN_PEGEL_UEBERL_OBEN), DELAY_UEBERLAUF_OBEN);
109
110  /*Wasserstand beim Überlauf Unten aus digitalRead vom Sensor wird in Variabel gespeichert
111      mit Verzögerung (Delay), um zu schnelles Wechseln des Wertes zu verhindern*/
112  bool Wasserstand_UEBERLAUF_UNTEN = timerUeberlaufUnten.delayOff(digitalRead
113      (PIN_PEGEL_UEBERL_UNTEN), DELAY_UEBERLAUF_UNTEN);
114
115  //Wasserstand der Wanne Oben aus analogRead vom Sensor wird in Variabel gespeichert
116  bool Wasserstand_WANNE_OBEN = timerWanneOben.delayOff(digitalRead
117      (PIN_PEGEL_WANNE_OBEN), DELAY_WANNE_OBEN);
118
119  /*Wasserstand der Wanne Unten aus analogRead vom Sensor wird in Variabel gespeichert
120      mit Verzögerung (Delay): */
121  bool Wasserstand_WANNE_UNTEN = timerWanneUnten.delayOff(digitalRead
122      (PIN_PEGEL_WANNE_UNTEN), DELAY_WANNE_UNTEN);
123
124
125  //Der analoge Wert (vom Aquarium) wird mit Hysterese einem Wasserstand zugeordnet:
126  if (Wasserstand_AQ > 900) {
127      WaSt_AQ = WaSt_MAX;
128  }
129  else if (Wasserstand_AQ < 370) {
130      WaSt_AQ = WaSt_LEER;
131  }
132  else if ((Wasserstand_AQ > 400) && (Wasserstand_AQ < 440)) {
133      WaSt_AQ = WaSt_NORMAL;
134  }
135
136  //Die digitalen Werte vom Überlaufbecken werden hier in einen Wasserstand ausgewertet:
137  if (Wasserstand_UEBERLAUF_OBEN) {
138      WaSt_UEBERLAUF = WaSt_MAX;
139  }
140  else if (Wasserstand_UEBERLAUF_UNTEN) {
141      WaSt_UEBERLAUF = WaSt_NORMAL;
142  }
143  else {
144      WaSt_UEBERLAUF = WaSt_LEER;
145  }
146
147  //Der analoge Wert der Regenrinne wird in einen Wasserstand umgewandelt (mit Hysterese)
148  if (Wasserstand_REGEN > 550) {
149      WaSt_REGEN = WaSt_MAX;
150  }
151  else if (Wasserstand_REGEN < 10) {
152      WaSt_REGEN = WaSt_NORMAL;
153  }
154
155  //Die digitalen Wertet aus der Wanne werden in einen Wasserstand ausgewertet
156  if (Wasserstand_WANNE_OBEN) {
157      WaSt_WANNE = WaSt_MAX;
158  }
159  else if (Wasserstand_WANNE_UNTEN) {
160      WaSt_WANNE = WaSt_NORMAL;
161  }

```

```

162     else {
163         WaSt_WANNE = WaSt_LEER;
164     }
165 }
166
167 //Aufgrund der jeweiligen Wasserstände, wird dem LCD ein WaSt-Symbol gedruckt:
168 void WaStANZEIGEN(){
169     lcd.setCursor(1,0);
170     if (WaSt_WANNE == WaSt_LEER) {
171         lcd.write(WaSt0);
172     }
173     else if (WaSt_WANNE == WaSt_NORMAL) {
174         lcd.write(WaSt3);
175     }
176     else if (WaSt_WANNE == WaSt_MAX) {
177         lcd.write(WaSt6);
178     }
179
180     lcd.setCursor(15,0);
181     if (WaSt_REGEN == WaSt_MAX) {
182         lcd.write(WaSt6);
183     }
184     else if (WaSt_REGEN == WaSt_NORMAL) {
185         lcd.write(WaSt3);
186     }
187
188     lcd.setCursor(4,1);
189     if (WaSt_UEBERLAUF == WaSt_MAX) {
190         lcd.write(WaSt6);
191     }
192     else if (WaSt_UEBERLAUF == WaSt_NORMAL) {
193         lcd.write(WaSt3);
194     }
195     else if (WaSt_UEBERLAUF == WaSt_LEER) {
196         lcd.write(WaSt0);
197     }
198
199     lcd.setCursor(9,0);
200     if (WaSt_AQ == WaSt_MAX) {
201         lcd.write(WaSt6);
202     }
203     else if (WaSt_AQ == WaSt_NORMAL) {
204         lcd.write(WaSt4);
205     }
206     else if (WaSt_AQ == WaSt_LEER) {
207         lcd.write(WaSt0);
208     }
209 }
210
211
212
213 void pumpenAutomatik() {
214     //Es gibt verschiedene Phasen des Automatik-Programms:
215     typedef enum {
216         NORMAL_BETRIEB,      //AQ wird laufend befüllt und Wasser läuft durch Regen-Rinne

```

```

217     AQ_FUELLEN,           //AQ wird bis WaSt_MAX aufgefüllt und nichts anderes läuft
218     WA_AQ_FUELLEN,       //AQ wird laufend befüllt und Wanne wird bis WaSt_MAX befüllt
219     WA_Ablassen          //Erst wenn Wanne wieder leer ist, geht es weiter
220 } WanneAUTOMATIK_Typ;
221 //zu Beginn wird auf Phase NORMAL_BETRIEB geschaltet:
222 static WanneAUTOMATIK_Typ WanneAUTOMATIK = NORMAL_BETRIEB;
223 WanneTimer.delayOn(false, ZEIT_NORMAL_BETRIEB); //Zeit zum normalen Betrieb wird gestartet
224 WanneTimer.delayOn(true, ZEIT_NORMAL_BETRIEB);
225
226 filterAQ = true;
227 filterRechts = true;
228 filterLinks = false;
229
230 //In WanneAUTOMATIK wird gespeichert, was in welcher Phase des Automatik-Programms passiert:
231 switch (WanneAUTOMATIK) {
232     case NORMAL_BETRIEB:
233         //Jede halbe Stunde wird WanneAUTOMATIK umgeschaltet von NORMAL_BETRIEB zu AQ_FUELLEN
234         if (WanneTimer.delayOn(true, ZEIT_NORMAL_BETRIEB)) {
235             WanneAUTOMATIK = AQ_FUELLEN;
236             //Zeit Neustart
237             //Zeit beginnt neu zu zählen, wenn es einen Wechsel/Flanke von false auf true gibt
238             WanneTimer.delayOn(false, TIMEOUT_FUELLEN);
239             WanneTimer.delayOn(true, TIMEOUT_FUELLEN);
240         }
241         //Differenz aus vergangener Zeit und Zeit, die diese Phase max. dauern darf:
242         RestZeitBetriebsart = (ZEIT_NORMAL_BETRIEB - WanneTimer.getTimeElapsed())/60000;
243         break;
244
245     case AQ_FUELLEN:
246         filterLinks = false;
247         filterRechts = false;
248         filterAQ = true;
249         if (WaSt_AQ == WaSt_MAX){
250             WanneAUTOMATIK = WA_AQ_FUELLEN;
251         }
252         else if (WanneTimer.delayOn(true, TIMEOUT_FUELLEN)) {
253             WanneAUTOMATIK = NORMAL_BETRIEB;
254             //Zeit Neustart
255             //Zeit zum normalen Betrieb wird neu gestartet
256             WanneTimer.delayOn(false, ZEIT_NORMAL_BETRIEB);
257             WanneTimer.delayOn(true, ZEIT_NORMAL_BETRIEB);
258         }
259         //Differenz aus vergangener Zeit und Zeit, die diese Phase max. dauern darf:
260         RestZeitBetriebsart = (TIMEOUT_FUELLEN - WanneTimer.getTimeElapsed())/60000;
261         break;
262
263     case WA_AQ_FUELLEN:
264         filterLinks = true;
265         filterRechts = false;
266         filterAQ = true;
267         if ((WaSt_WANNE == WaSt_MAX) || WanneTimer.delayOn(true, TIMEOUT_FUELLEN)){
268             WanneAUTOMATIK = WA_Ablassen;
269             //Zeit Neustart
270             //Zeit zum normalen Betrieb wird neu gestartet:
271             WanneTimer.delayOn(false, ZEIT_NORMAL_BETRIEB);

```

```

272     WanneTimer.delayOn(true, ZEIT_NORMAL_BETRIEB);
273 }
274 //Differenz aus vergangener Zeit und Zeit, die diese Phase max. dauern darf:
275 RestZeitBetriebsart = (TIMEOUT_FUELLEN - WanneTimer.getTimeElapsed())/60000;
276 break;
277
278 case WA_Ablassen:
279     RestZeitBetriebsart = -1;
280     if (WaSt_WANNE == WaSt_LEER){
281         WanneAUTOMATIK = NORMAL_BETRIEB;
282         //Zeit Neustart
283         //Zeit zum normalen Betrieb wird neu gestartet:
284         WanneTimer.delayOn(false, ZEIT_NORMAL_BETRIEB);
285         WanneTimer.delayOn(true, ZEIT_NORMAL_BETRIEB);
286     }
287 }
288
289 //Wichtige Bedingung im Kreislauf:
290 if (WaSt_AQ == WaSt_LEER) {
291     filterLinks = false;
292     filterRechts = false;
293 }
294
295
296 //Notfall Massnahmen (diese überschreiben alle vorherigen Einstellungen!):
297 if (WaSt_UEBERLAUF == WaSt_MAX) {
298     filterRechts = false;
299     filterLinks = false;
300     filterAQ = true;
301 }
302 if (WaSt_REGEN == WaSt_MAX) {
303     filterRechts = false;
304 }
305 if (WaSt_UEBERLAUF == WaSt_LEER){
306     filterAQ = false;
307 }
308 if (WaSt_AQ == WaSt_MAX) {
309     filterAQ = false;
310 }
311 if (WaSt_WANNE == WaSt_MAX) {
312     filterLinks = false;
313 }
314 }
315
316 //Auswahl vom pumpenAUTOMATIK() oder Loop() wird ausgeführt:
317 void output() {
318     //Soll der linke Filter an- oder ausgehen?
319     digitalWrite(PIN_FILTER_LINKS, filterLinks);
320     if (filterLinks) { //Soll Display dementsprechend bedruckt werden?
321         lcd.setCursor(3,0);
322         lcd.write(PfLinks);
323         lcd.setCursor(4,0);
324         lcd.print ("---");
325     }
326     else {

```

```

327     lcd.setCursor(3,0);
328     lcd.print ("  ");
329 }
330
331 //Soll der AQ Filter an- oder ausgehen?
332 digitalWrite (PIN_FILTER_AQ, filterAQ);
333 if (filterAQ){ //Soll Display dementsprechend bedruckt werden?
334     lcd.setCursor(5,1);
335     lcd.print ("---");
336     lcd.write(PfLAuf);
337 }
338 else {
339     lcd.setCursor(5,1);
340     lcd.print ("  ");
341 }
342
343 //Soll der rechte Filter an- oder ausgehen?
344 //filterRechts ist entweder true oder false; HIGH oder LOW
345 digitalWrite(PIN_FILTER_RECHTS, filterRechts); /
346 if (filterRechts) {
347     lcd.setCursor(10, 0); //Soll Display dementsprechend bedruckt werden?
348     lcd.print ("--");
349     lcd.write(PfRechts);
350 }
351 else {
352     lcd.setCursor(10, 0);
353     lcd.print ("  ");
354 }
355 //Verbleibende Zeit einer Pahese des Automatik-Programms:
356 lcd.setCursor(9, 1);
357 if (Betriebsart_AUTOMATIK) {
358     if (RestZeitBetriebsart == -1) {
359         lcd.print("??");
360     }
361     else {
362         char strBuffer[4];
363         sprintf(strBuffer, "%21d", RestZeitBetriebsart);
364         lcd.print(strBuffer);
365     }
366 }
367 else {
368     lcd.print ("  ");
369 }
370
371 }
372
373 //Eigentliches Programm, das die einzelnen Funktionen ausführt:
374 void loop() {
375     sensorenAuswerten(); //Funktion sensorenAuswerten() wird aufgerufen
376     WaStANZEIGEN(); //Funktion WaStANZEIGEN() wird aufgerufen
377
378     Tasten aktuelleTaste = TasteLesen();
379     //Welche Taste gedrückt wird, wurde in Funktion TasteLesen() ausgewertet
380     //Hier wird, je nachdem welche Taste gedrückt wird, etwas anderes ausgeführt/verändert:
381     switch (aktuelleTaste) {

```

```

382 //Bei dieser Taste wird zwischen Manuellem und Automatikbetrieb gewechselt:
383 case TASTE_SELECT:
384     Betriebsart_AUTOMATIK = !Betriebsart_AUTOMATIK;
385     lcd.setCursor(12, 1); //aUSGABE AUF lcd, welcher Modus aktiv ist
386     lcd.print(Betriebsart_AUTOMATIK ? "AUTO" : "MAN.");
387     WanneTimer.delayReset();
388     filterRechts = false; //Bei einem Wechsel werden immer alle Pumpen abgeschaltet
389     filterLinks = false;
390     filterAQ = false;
391
392     break;
393 case TASTE_AUF:
394     //nichts definiert
395     break;
396 case TASTE_AB: //Bei dieser Taste wird der AQ Filter umgeschaltet
397     filterAQ = !filterAQ;
398     break;
399 case TASTE_LINKS: //Bei dieser Taste wird der linke Filter umgeschaltet
400     filterLinks = !filterLinks;
401     break;
402 case TASTE_RECHTS: //Bei dieser Taste wird der rechte Filter umgeschaltet
403     filterRechts = !filterRechts;
404     break;
405 case TASTE_KEINE:
406     //Hier gibt es nichts zu tun
407     break;
408 }
409
410 /*Programm soll, wenn eine Taste gedrückt ist,
411 hier solange warten, bis sie wieder losgelassen wird.
412 Sonst wird ein Tastendruck als mehrere hintereinander wahrgenommen.*/
413 if (aktuelleTaste != TASTE_KEINE) {
414     while (TasteLesen() != TASTE_KEINE){
415     }
416 }
417
418 //Funktion pumpenAUTOMATIK() wird aufgerufen, wenn Betriebsart AUTO aktiv ist:
419 if (Betriebsart_AUTOMATIK) {
420
421     pumpenAutomatik();
422
423 }
424
425 /*Funktion output() wird zuletzt aufgerufen, da ja hier erst alle erstellten,
426 abgeänderten oder gespeicherten Werte tatsächlich ausgeführt werden.*/
427 output();
428 }

```

keyboard.cpp

```
1  #include <Arduino.h>
2  #include "keyboard.h"
3
4  /**
5   * Liest die Tasten des LCD Displays ein.
6   * @return Die aktuell gedrückte Taste
7   */
8  Tasten TasteLesen() {
9      int taste = analogRead (A0); //Analoge Eingangstaste der Knöpfe
10     //Serial.print("Taste=");
11     //Serial.println(taste);
12
13     if (taste < 60) { //Rechte Taste
14         return TASTE_RECHTS;
15     }
16     else if (taste < 200) { //Obere Taste
17         return TASTE_AUF;
18     }
19     else if (taste < 350){ //Untere Taste
20         return TASTE_AB;
21     }
22     else if (taste < 550){ //Linke Taste
23         return TASTE_LINKS;
24     }
25     else if (taste < 800){ //"Select" Taste
26         return TASTE_SELECT;
27     }
28     return TASTE_KEINE;
29 }
```

keyboard.h

```
1  #pragma once
2
3  typedef enum { //Alle Tasten, die es gibt in einer Liste gespeichert
4      TASTE_KEINE,
5      TASTE_SELECT,
6      TASTE_AUF,
7      TASTE_AB,
8      TASTE_LINKS,
9      TASTE_RECHTS
10 } Tasten;
11
12
13 /**
14  * Liest die Tasten des LCD Displays ein.
15  * @return Die aktuell gedrückte Taste
16  */
```

```
17
18 extern Tasten TasteLesen();
```

lcd.cpp

```
1  #include <Arduino.h>
2  #include "lcd.h"
3
4  //LCD pin to Arduino
5  const int pin_RS = 8;
6  const int pin_EN = 9;
7  const int pin_d4 = 4;
8  const int pin_d5 = 5;
9  const int pin_d6 = 6;
10 const int pin_d7 = 7;
11 const int pin_BL = 10;
12 LiquidCrystal lcd( pin_RS, pin_EN, pin_d4, pin_d5, pin_d6, pin_d7);
13
14 void setupLCD() {
15     //Eigene Sonderzeichen pixelweise erstellen:
16     byte WasserVoll[8] = {B11111,B11111, B11111, B11111,B11111,B11111,B11111,B11111,
17     };
18
19     byte Wasserdazwischen4[8] ={B11111,B10001,B11111,B11111,B11111,B11111,B11111,B11111,
20     };
21
22     byte Wasserdazwischen3[8] ={B11111,B10001,B10001,B11111,B11111,B11111,B11111,B11111,
23     };
24
25     byte Wasserdazwischen2[8] ={B11111,B10001,B10001,B10001,B11111,B11111,B11111,B11111,
26     };
27
28     byte Wasserdazwischen1[8] ={B11111,B10001,B10001,B10001,B10001,B11111,B11111,B11111,
29     };
30
31     byte Wasserdazwischen0[8] = {B11111,B10001,B10001,B10001,B10001,B10001,B10001,B11111
32     };
33
34     byte WasserLeer[8] = {B11111,B10001,B10001,B10001,B10001,B10001,B10001,B11111,
35     };
36
37     byte PfeilvonLinksAuf[8] = {B01110,B11111,B00100,B11100,B00000,B00000,B00000,B00000
38     };
39
40     //Display konfigurieren & eigene Zeichen erstellen (wird einmal ausgeführt):
41     lcd.begin(16, 2);
42     lcd.clear();
43
44     lcd.createChar(WaSt6, WasserVoll);
45     lcd.createChar(WaSt5, Wasserdazwischen4);
46     lcd.createChar(WaSt4, Wasserdazwischen3);
47     lcd.createChar(WaSt3, Wasserdazwischen2);
```

```

48   lcd.createChar(WaSt2, Wasserdazwischen1);
49   lcd.createChar(WaSt1, Wasserdazwischen0);
50   lcd.createChar(WaSt0, WasserLeer);
51   lcd.createChar(PfLAuf, PfeilvonLinksAuf);
52
53   lcd.setCursor(0,0);
54   lcd.print("W");
55   lcd.setCursor(7, 0);
56   lcd.print("AQ");
57   lcd.setCursor(14, 0);
58   lcd.print("R");
59   lcd.setCursor(2, 1);
60   lcd.print("UB");
61
62   lcd.setCursor(12, 1);
63   lcd.print("MAN.");
64
65 }

```

lcd.h

```

1   #pragma once
2   #include <LiquidCrystal.h>
3
4   //Alle Variablen, die für die eigenen Zeichen für das Display verwendet werden:
5   const int WaSt6 = 6;
6   const int WaSt5 = 5;
7   const int WaSt4 = 4;
8   const int WaSt3 = 3;
9   const int WaSt2 = 2;
10  const int WaSt1 = 1;
11  const int WaSt0 = 0;
12
13  const int PfLinks = 0x7f;
14  const int PfRechts = 0x7e;
15  const int PfLAuf = 7;
16
17
18
19  extern LiquidCrystal lcd;
20
21  extern void setupLCD();

```

Verwendete Libraries

Dies sind die Module, welche nicht selber erstellt sind und aus dem Internet kopiert sind um sie zu verwenden:

1. **muTimer**
Zeitfunktionen: Ein- & Ausschaltverzögerungen
2. **LiquidCrystal**
Ansteuerung für LCD-Display

Datenblätter

Hier sind Ausschnitte der Datenblätter von wichtigen Bauteile, die für die Anlage verwendet wurden:

Technische Daten

Wasserpegelsensor

Betriebsspannung	5 V/DC
Sensortyp	Analog
Abmessungen (ca.)	65 x 22 x 6 mm
Gewicht (ca.)	4 g

Bodenfeuchtesensor

Betriebsspannung	3,3 oder 5 V/DC
Betriebsstrom	<20 mA
Ausgangsspannung	Stromversorgung 5 V: 0 - 3 V/DC
	Stromversorgung 3 V: 0 - 1,7 V
Sensortyp	Analog
Nutzungsdauer (ca.)	1 Jahr
Abmessungen (ca.)	65 x 22 x 6 mm
Gewicht (ca.)	3 g

Entsorgung



Elektronische Geräte sind Wertstoffe und gehören nicht in den Hausmüll.

Entsorgen Sie das Produkt am Ende seiner Lebensdauer gemäß den geltenden gesetzlichen Bestimmungen.

Sie erfüllen damit die gesetzlichen Verpflichtungen und leisten Ihren Beitrag zum Umweltschutz.

Impressum

Dies ist eine Publikation der Conrad Electronic SE, Klaus-Conrad-Str. 1, D-92240 Hirschau (www.conrad.com).

Alle Rechte einschließlich Übersetzung vorbehalten. Reproduktionen jeder Art, z. B. Fotokopie, Mikroverfilmung, oder die Erfassung in elektronischen Datenverarbeitungsanlagen, bedürfen der schriftlichen Genehmigung des Herausgebers. Nachdruck, auch auszugsweise, verboten.

Die Publikation entspricht dem technischen Stand bei Drucklegung.
Copyright 2019 by Conrad Electronic SE.

LIQUID LEVEL SENSOR

VCS Series – Compact Liquid Level Sensor

VCS-02

Design Features

- Overall length less than 60mm
- M8 x 1.25mm fitting
- Internal fit
- 30mm float diameter
- Point Alarm

Product Comments

- High or Low level switching
- Normally Open or Normally Closed by float reversal
- Reed Switch Reliability (UL File E153493)
- Wetted magnet
- Interface option

Typical Applications

- Water high or low Level
- Condensate level alarm
- Chemical high or low level
- Cooling systems
- Coolant level indication



Material, Design and Operation

Mounting:	Top or Bottom Entry
Material:	Polypropylene
Fitting:	Internal Fitting
Contact Form:	Form A - SPST
Contact Material:	Ruthenium
Maximum Switching Voltage:	250VAC
Maximum Switching Current:	1.0
Maximum Power:	15 Watts
Housing Temperature Range °C:	70
Housing Material:	Glass Filled Polypropylene
Housing Colour:	Black
Float Material:	Foamed Polypropylene
Connection type:	Stripped Cables
Cable Type	2 x 22AWG, 16/0.2mm PVC Insulated 0.1M long
Cable Colour:	
Minimum Specific Gravity:	0.64
Operating Temperature Range °C:	70
Medium / Liquid:	Water
Operate Mode:	N/C with float down Internal Fitting

749

Geräteanschlussstecker

Geräteanschlussstecker 70°, 10 A, mit Schutzkontakt und Knickschutzhülse
 Plug connector 70°, 10 A, with earthing contact and strain relief sleeve



Gewicht / Weight 28 g



Beschreibung / Description

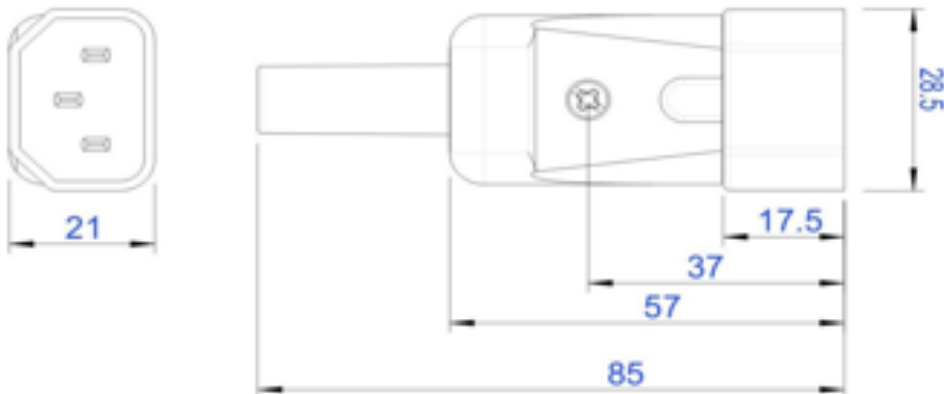
Wiederanschließbare Ausführung für flexible Leitungen, H05RR-F 3G0,75 mm² und 3G1,0 mm², ab 2 m Zuleitungslänge nur 3G1,0 mm² max. Ø 8mm

Rewireable, for supply line H05VV-F 3G0,75 mm² and 3G1,0 mm², for lengths over 2 meters only 3G1,0 mm² max. Ø 8mm

Technische Daten / Specifications

Nennspannung / Rated voltage	250 V~
Nennstrom / Rated current	10 A
Bauart / According to	DIN EN 60320-2-2, Normblatt E
Anschlüsse / Type of terminals	Schraubanschluss / Screw terminals
Kontakte / Material of terminals	CuZn 39, Fe
Kontaktträger / Carrying parts	Polyamid 6 / Polyamide 6
Gehäuse / Cover	Polyamid 6 / Polyamide 6

Abmessungen / Dimensions

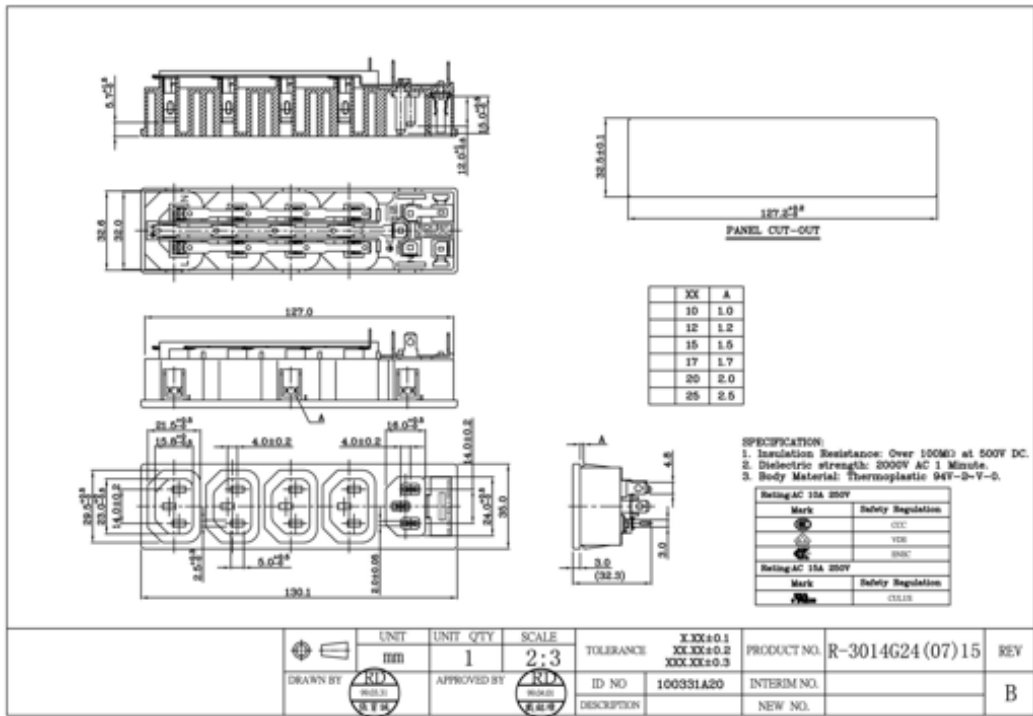


Abisolierung / Wire stripping:

N	L	PE
25 mm	25 mm	40 mm

www.martin-kaiser.de info@martin-kaiser.de

Technische Angaben und Änderungen behalten wir uns vor • Technical data are subject to change



UNIT	UNIT QTY	SCALE	TOLERANCE	PRODUCT NO.	REV
mm	1	2:3	±0.1 ±0.2 ±0.3	R-3014G24(07)15	
DRAWN BY	APPROVED BY	ID NO	DESCRIPTION	INTERM NO.	NEW NO.
		100331A20			
					B



Patent: ZL 2010 2 0563194.3



ISO9001, ISO14001 Certification Enterprises

Main Feature

1. Small size (29.3x12.8x20.4 in mm) produces a switching capacity up to 5A for high density P.C Board mounting technique.
2. The contact form construction is 2A2B2C
3. The Surge Resistance of BPS2 series is 10,000V
4. Sealing Construction (Free from dust and solder flux): BPS2-SS: Plastic Sealed Type.
5. The selection of plastic insulation material is designed for high temperature and provides better chemical solution performance.
6. RoHS Compliant.

Application

Air Conditioning, Fridge, Washing Machine, etc Household Appliances

Contact Rating

- Nominal Load/Resistive Load Cos φ=1)
 - BPMD-LLM 5A at 250VAC
 - BPMD-LLM 5A at 30VDC
- Max. Allowable Current
 - BPMD-LLM 5A
- Max. Allowable Voltage
 - BPMD-LLM AC250V DC30V
- Max. Allowable Power Force
 - BPMD-LLM 1,250VA, 150W
- Contact Material Ag Alloy
- Contact Form DPDT & CPST

Performance (at Initial Value)

- Contact Resistance <50mΩ at 6VDC/1A
- Operate Time 10ms, Max
- Release Time 5ms, Max
- Dielectric Strength:
 - Between Coil & Contact 5,000VAC at 50/60 Hz for one minute
 - Between Contacts 1,000VAC at 50/60 Hz for one minute
- Surge Resistance 10,000V (between Coil & Contact 1.2x50 μs)
- Insulation Resistance 1,000 Mega Ω Min. at 500VDC

- Max. On/Off Switching:
 - Electrical 30 Ops per minute
 - Mechanical 300 Ops per minute
- Temperature Range -40~85°C
- Humidity Range 5%~85% RH
- Coil Temperature Rise 35°C Maximum
- Vibration:
 - Endurance 10 to 55 Hz dual amplitude width 1.5mm
 - Error Operation 10 to 55 Hz dual amplitude width 1.5mm
- Shock:
 - Endurance 981ms/7 Min
 - Error Operation 98.1ms/7 Min
- Life Expectancy:
 - Electrical 10⁷ Operations at Rated Resistive load
 - Mechanical 10⁷ Operations at No load condition
- Weight about 13g

Safety Standard & Its File Number

- UL E332719
- TUV R50179592
- CCC CQC10001046679

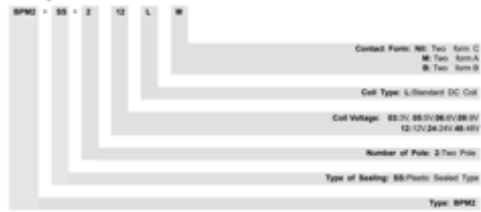


ISO9001, ISO14001 Certification Enterprises

Coil Specification (at 20 °C)

Coil Sensitivity	Nominal Voltage (VDC)	Nominal Current (mA)	Coil Resistance (Ω/1%)	Power Consumption (W)	Pull-in Voltage (VDC)	Drop-Out Voltage (VDC)	Maximum Allowable Voltage (VDC)
BPMD-L BPMD-LM	2	150.0	17	Max 0.54	75% Maximum	15% Minimum	100%
	3	100.7	45				
	5	60.4	87				
	6	50	100				
	12	25.0	210				
	24	12.5	420				

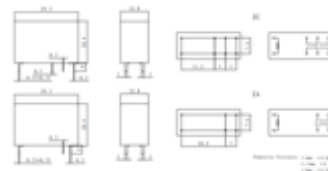
Ordering Information



Classification

Model	Coil Sensitivity	Standard DC Coil
BPMD-SS-2-L-L	2C	5A
BPMD-SS-2-L-LM	2C	5A

Dimension



Absolute Maximum Ratings (V_{CC}=0V, V_{IO}=0V)

Parameter	Symbol	Rating	Unit
Supply voltage	V _{CC}	0.3 to 1.7	V
Output terminal voltage	V _{OL}	0.3 to V _{CC} -0.3	V
Operating temperature	T _{op}	-40 to +85	°C
Storage temperature	T _{stg}	-40 to +75	°C

Electro-optical Characteristics (V_{CC}=0V, V_{IO}=0V)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Average supply current	I _{CC}	1-White (Dist. 1)	-	30	80	mA
Distance measuring	DR	White (Dist. 1)	30	-	80	cm
Output voltage	V _{OH}	1-White (Dist. 1)	0.20	0.4	0.55	V
Output voltage differential	ΔV _{OL}	Output voltage difference between 1-White (Dist. 1) and 2-White (Dist. 1)	0.07	0.5	0.55	V

① L: Distance to reflective object
 Note 1: Using reflective object: White paper (Made by Kodak Co., Ltd) gray card 8.27*white face, reflectance: 90%

Recommended operating conditions

Parameter	Symbol	Rating	Unit
Supply voltage	V _{CC}	4.5 to 5.5	V

Fig. 1 Timing chart

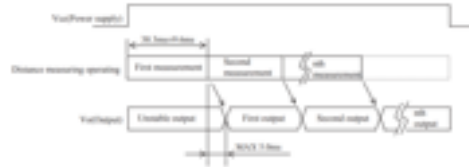
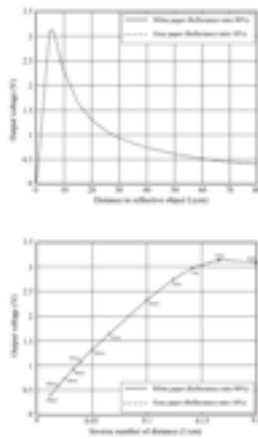


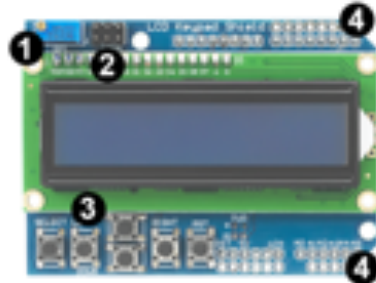
Fig. 2 Example of distance measuring characteristics graph



Notes

- Notes for the output**
 - The face of the detector must be kept clean. When you use the detector, water or oil and so on deteriorate the characteristics of the device. Please consider in actual application.
 - Please do not touch. Wiping dust deteriorates the characteristics of optical system and so on. Please confirm instructions in documents under the actual usage since the product has not been designed against rubbing.
- Notes for the characteristics**
 - In case that an optical filter is set in front of the emitter and detector portion, the optical filter which has the most effective transmission at the emitting wavelength range of 850nm for the product is 90%. There, shall be recommended to use. Both faces of the filter should be covered with adhesive film, in case any case that the characteristics may not be satisfied according to the distance between the protective cover and the product or the thickness of the transparent cover. Please use the product after confirming the operation sufficiently in actual application.
 - In case that there is an object near to emitter side of the sensor between sensor and a detecting object, please use the device after confirming sufficiently that the characteristics of the sensor do not change by the object.
 - When the detector is exposed to the direct light from the sun, lamp etc. long and so on, there are cases that it can not measure the distance correctly. Please consider the design that the detector is not exposed to the direct light from such light source.
 - Distance to a mirror reflector can not be measured accurately.
 - In case of changing the mounting angle of this product, it may measure the distance correctly.
 - In case the reflective object has protrusion (see which material or color etc. are necessary different, in order to decrease deviation of measuring distance, it shall be recommended to set the sensor that the distance of protrusion face and the face between emitter sensor and detector sensor are in parallel.
- Notes for the power supply**
 - In order to optimize power supply line, we recommend to insert a fuse (specification of 100mA or more) between Vcc and GND near this product.
- Notes on handling**
 - There are some possibilities that the internal components in the sensor may be exposed to the excessive mechanical stress. Please be careful not to cause any excessive pressure on the sensor package, and also on the PCB while mounting this product.

Bedienelemente



Anzeige	Beschreibung
1	Kontrasteinstellschraube (geeigneten Schraubendreher verwenden)
2	ICSP
3	Steuertasten (dem A0-Pin zugeordnet)
4	Lötstellen

Pinout / Pin Map

Pin	Funktion
Analog 0	Taste (anwählen, hoch, rechts, herunter und links)
Digital 4	DB4 (LCD)
Digital 5	DB5 (LCD)
Digital 6	DB6 (LCD)
Digital 7	DB7 (LCD)
Digital 8	RS
Digital 9	RW
Digital 10	Steuerung der Hintergrundbeleuchtung

Installation

Sie können das Shield direkt auf der Platine installieren oder auf andere Weise mit der Platine verbinden. Pins und Lötstellen sind verfügbar.

Technische Daten

Betriebsspannung	5 V/DC
Displaytyp	LCD
Auflösung	16 x 2 px
Abmessungen (ca.)	80 x 58 x 20 mm
Gewicht (ca.)	50 g



VDR Metal Oxide Varistors Standard



LINKS TO ADDITIONAL RESOURCES



QUICK REFERENCE DATA		
PARAMETER	VALUE	UNIT
Maximum continuous voltage in operating temperature range:		
RMS	14 to 680	V
DC	18 to 895	V
Maximum non-repetitive transient current $I_{pp}(8 \times 20 \mu s)$	100 to 6500	A
Maximum energy (10/1000 μs)	0.5 to 496	J
Detailed specification	Based on IEC 61051-1 IEC 61051-2 IEC 61051-3-2	
Storage temperature	-40 to +125	°C
Operating temperature	-40 to +85	°C

ORDERING INFORMATION

The varistors are available in a number of packaging options:

- Bulk
- On tape and reel
- On tape in ammpack (fanfold)

The basic ordering code for each option is given in tables titled Varistors on Tape on Reel, Varistors on Tape in Ammpack and Varistors in Bulk. To complete the catalog number and to determine the required operating parameters, see Electrical Data and Ordering Information table.

Note

- Special lead-configuration as inside or outside crimped leads on request.

AGENCY APPROVALS

- cUL certificate
- ULus certificate
- VDE certificate

Note

- Agency approval documents, please see: www.vishay.com/doc?2001&documents

FEATURES

- Low β high purity zinc oxide disc
- Halogen free insulating epoxy coating
- Straight leads and kinked leads
- Straight leads with flange (VDRS05 and VDRS07 only)
- Certified for operation up to 85 °C according to UL 1449 edition 4, VDE/IEC 61051-1/2
- Material categorization: for definitions of compliance please see www.vishay.com/doc?99912



APPLICATION

- Overvoltage and transient voltage protection

DESCRIPTION

The varistors consist of a disc of low- β ZnO ceramic material with two solid copper leads (S20 types only) or copper clad steel wire. The wires have a matte tin plating. They are coated with UL 94 V-0 approved other colored halogen-free epoxy, which provides electrical, mechanical and climatic protection.

MOUNTING

The varistors are suitable for hand-mounting (bulk) or automatic pick and place mounting (tape on reel or fanfold). The parts can be soldered by hand or wave soldering. Pin-in-paste reflow soldering is not recommended. Bending of the leads for different angle placement is not recommended.

Typical soldering

- 235 °C, duration: 5 s (Pb-bearing)
- 245 °C, duration: 5 s (lead (Pb)-free)

Resistance to soldering heat

260 °C, duration: 10 s max.

MARKING

The varistors are marked with the following information:

- Maximum continuous RMS voltage
- Series numbers
 - 592 for VDRS05
 - 593 for VDRS07
 - 594 for VDRS10
 - 595 for VDRS14
 - 596 for VDRS20
- Safety marks on VDRS10-14-20 types
- Manufacturers logo
- Date of manufacture (YYWW)

INFLAMMABILITY

The varistors are passive non-flammable.

The encapsulation is made of flammable resistant epoxy in accordance with UL 94 V-0.

NPN Epitaxial Silicon Transistor
BC546 / BC547 / BC548 / BC549 / BC550

Features

- Switching and Amplifier
- High-Voltage: BC546, $V_{CE(sat)} = 65\text{ V}$
- Low-Noise: BC546, BC550
- Complements BC156, BC157, BC158, BC359, and BC360
- These are Pb-Free Devices



1 - Collector 2 - Base 3 - Emitter

ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Value	Unit
Collector-Base Voltage BC546 BC547 - BC549 BC548 - BC549	V_{CB}	60 60 30	V
Collector-Emitter Voltage BC546 BC547 - BC549 BC548 - BC549	V_{CE}	60 40 30	V
Emitter-Base Voltage BC546 - BC547 BC548 - BC549 / BC550	V_{EB}	6 5	V
Collector Current (DC)	I_C	100	mA
Collector Power Dissipation	P_C	500	mW
Ambient Temperature	T_A	150	°C
Storage Temperature Range	T_{STG}	-65 to +150	°C

Stresses exceeding those listed in the Maximum Ratings table may damage the device. If any of these limits are exceeded, device functionality should not be assumed. Storage may incur and reliability may be affected.

MARKING DIAGRAM



BC546 - Device Code
 1 - 4 or 5
 Y - 0, 1, 6, 8 or 9
 Z - A, B, C
 A - Assembly Location
 Y - Year
 Stock - Stock Stock

ORDERING INFORMATION

See detailed ordering and shipping information on page 4 of this data sheet.

BC546 / BC547 / BC548 / BC549 / BC550

ELECTRICAL CHARACTERISTICS ($T_C = 25^\circ\text{C}$ unless otherwise noted)

Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Units	
I_{CBO}	Collector Cut-off Current	$V_{CB} = 30\text{ V}, I_B = 0$		10	10	nA	
β_{DC}	DC Current Gain	$V_{CE} = 9\text{ V}, I_C = 2\text{ mA}$	110		800		
$V_{CE(sat)}$	Collector-Emitter Saturation Voltage	$I_C = 10\text{ mA}, I_B = 0.5\text{ mA}$		80	250	mV	
		$I_C = 100\text{ mA}, I_B = 5\text{ mA}$		250	400		
$V_{BE(sat)}$	Base-Emitter Saturation Voltage	$I_C = 10\text{ mA}, I_B = 0.5\text{ mA}$		700		mV	
		$I_C = 100\text{ mA}, I_B = 5\text{ mA}$		800			
$V_{BE(ON)}$	Base-Emitter On Voltage	$V_{CE} = 9\text{ V}, I_C = 2\text{ mA}$	580	680	750	mV	
		$V_{CE} = 9\text{ V}, I_C = 10\text{ mA}$			750		
f_T	Current Gain Bandwidth Product	$V_{CE} = 9\text{ V}, I_C = 10\text{ mA}, F = 100\text{ kHz}$		300		MHz	
C_{ob}	Output Capacitance	$V_{CE} = 10\text{ V}, I_C = 0, F = 1\text{ MHz}$		3.0	6.0	pF	
C_{ib}	Input Capacitance	$V_{CE} = 0.5\text{ V}, I_C = 0, F = 1\text{ MHz}$		0		pF	
NF	Noise Figure	BC546 - BC547 - BC548	$V_{CE} = 9\text{ V}, I_C = 200\text{ }\mu\text{A}, F = 1\text{ kHz}, R_N = 2\text{ k}\Omega$		3.0	10.0	dB
		BC548 - BC550			1.2	4.0	
		BC546	$V_{CE} = 9\text{ V}, I_C = 200\text{ }\mu\text{A}, R_N = 2\text{ k}\Omega, F = 30\text{ to }15000\text{ MHz}$		1.4	4.0	
		BC550			1.4	3.0	

Product parameter performance is indicated in the Electrical Characteristics for the listed test conditions, unless otherwise noted. Product performance may not be indicated by the Electrical Characteristics if operated under different conditions.

β_{DC} CLASSIFICATION

Classification	A	B	C
β_{DC}	110 - 200	200 - 400	400 - 800

Small Signal Fast Switching Diodes



FEATURES

- Silicon epitaxial planar diode
- Electrically equivalent diodes: 1N4148 - 1N914
- Material categorization for definitions of compliance please see www.vishay.com/doc/700012



RoHS
compliant
RECYCLED
FREE

APPLICATIONS

- Extreme fast switches

DESIGN SUPPORT TOOLS click logo to get started



MECHANICAL DATA

Case: DO-35 (DO-204AA)

Weight: approx. 105 mg

Cathode band color: black

Packaging codes / options:

TR10K per 13" reel (32 mm tape), 50K/box

TAP10K per ammpack (32 mm tape), 50K/box

PARTS TABLE				
PART	ORDERING CODE	TYPE MARKING	CIRCUIT CONFIGURATION	REMARKS
1N4148	1N4148-TAP-02 1N4148TR	VN148	Single	Tape and reel / ammpack

ABSOLUTE MAXIMUM RATINGS ($T_{amb} = 25\text{ }^{\circ}\text{C}$, unless otherwise specified)				
PARAMETER	TEST CONDITION	SYMBOL	VALUE	UNIT
Repetitive peak reverse voltage		V_{RRM}	150	V
Reverse voltage		V_R	75	V
Peak forward surge current	$t_p = 1\text{ }\mu\text{s}$	I_{FSM}	2	A
Repetitive peak forward current		I_{FRM}	500	mA
Forward continuous current		I_F	300	mA
Average forward current	$V_R = 0$	I_{AVG}	150	mA
Power dissipation	$l = 4\text{ mm}$, $T_c = 45\text{ }^{\circ}\text{C}$	P_{DM}	440	mW
	$l = 4\text{ mm}$, $T_c = 25\text{ }^{\circ}\text{C}$	P_{DM}	500	mW

THERMAL CHARACTERISTICS ($T_{amb} = 25\text{ }^{\circ}\text{C}$, unless otherwise specified)				
PARAMETER	TEST CONDITION	SYMBOL	VALUE	UNIT
Thermal resistance junction to ambient air	$l = 4\text{ mm}$, $T_c = \text{constant}$	$R_{\theta JA}$	350	K/W
Junction temperature		T_j	175	$^{\circ}\text{C}$
Storage temperature range		T_{stg}	-65 to +150	$^{\circ}\text{C}$

Rev. 1.4, 06-20-17 8 Document Number: 81887
 For technical questions within your region: DiodesAmerica@vishay.com, DiodesEurope@vishay.com, DiodesJapan@vishay.com
 THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE. THE PRODUCTS DESCRIBED HEREIN AND THIS DOCUMENT ARE SUBJECT TO SPECIFIC DISCLAIMERS, SET FORTH AT www.vishay.com/doc/781000

ELECTRICAL CHARACTERISTICS ($T_{amb} = 25\text{ }^{\circ}\text{C}$, unless otherwise specified)						
PARAMETER	TEST CONDITION	SYMBOL	MIN.	TYP.	MAX.	UNIT
Forward voltage	$I_F = 10\text{ mA}$	V_F			1	V
	$V_R = 20\text{ V}$	V_F			25	mV
Reverse current	$V_R = 20\text{ V}$, $T_c = 150\text{ }^{\circ}\text{C}$	I_R			50	μA
	$V_R = 75\text{ V}$	I_R			5	μA
		I_R				
Breakdown voltage	$I_R = 100\text{ }\mu\text{A}$, $I_{FT} = 0.01$, $t_p = 0.1\text{ ms}$	V_{BR}	100			V
Diode capacitance	$V_F = 0\text{ V}$, $f = 1\text{ MHz}$, $V_{DR} = 50\text{ mV}$	C_D			4	pF
Rectification efficiency	$V_{DR} = 2\text{ V}$, $f = 100\text{ MHz}$	η_r	45			%
Reverse recovery time	$I_F = I_R = 10\text{ mA}$, $I_R = 1\text{ mA}$	t_r			8	ns
	$I_F = 10\text{ mA}$, $V_R = 6\text{ V}$, $I_R = 0.1 \times I_F$, $R_f = 100\text{ }\Omega$	t_{rr}			4	ns