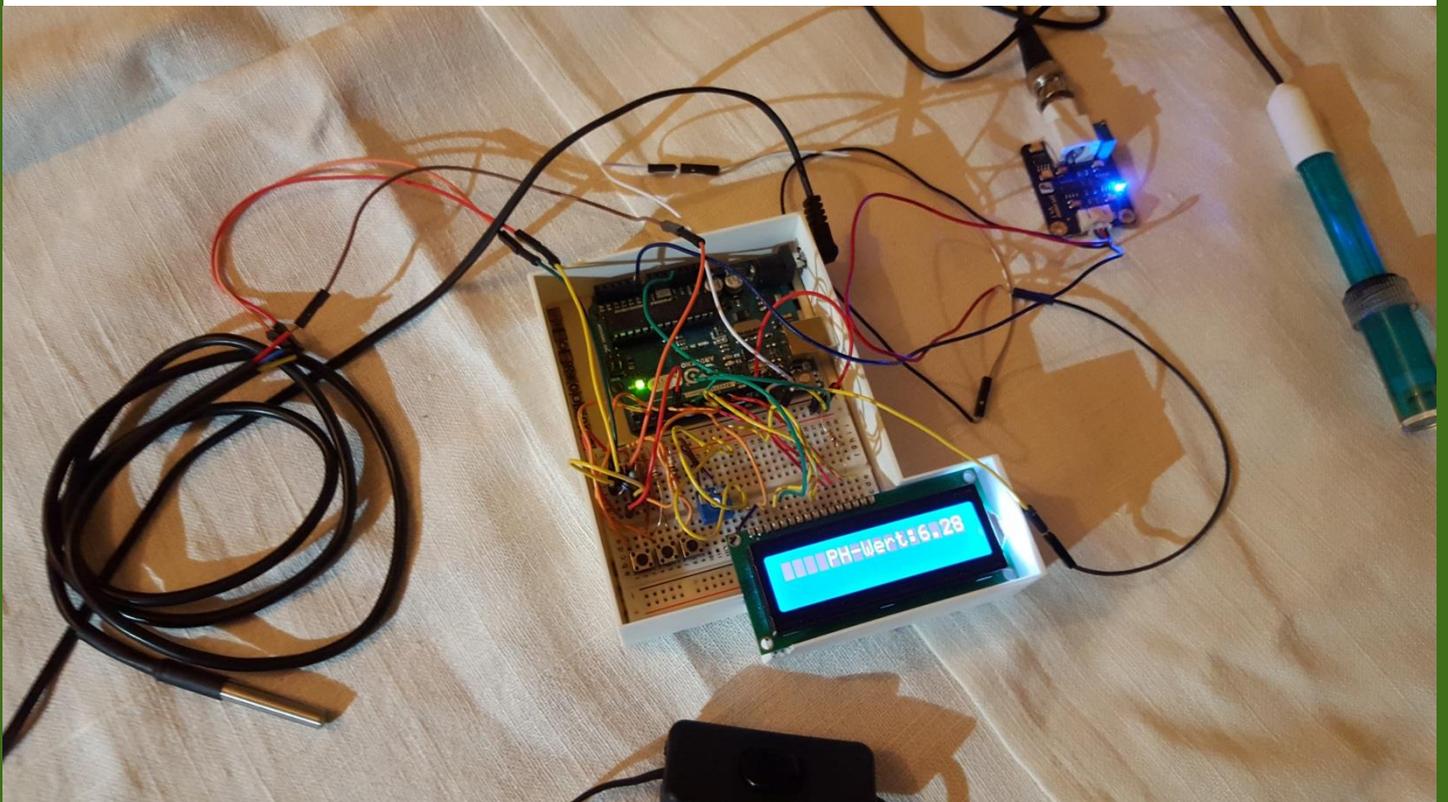


2015/16

MATURAARBEIT

TEMPHERINO



Bau eines digitalen pH- und Temperaturmessgeräts | Näf Noah  
Im Fach Informatik | Betreut von Dr. Rainer Steiger  
Kantonsschule Schaffhausen

# Inhaltsverzeichnis

---

1	Einleitung .....	2
1.1	Thema .....	2
1.2	Struktur der Arbeit .....	2
2	pH-Wert und Temperatur .....	3
2.1	Grundlagen zum pH-Wert .....	3
2.2	Messmethoden für den pH-Wert.....	3
2.3	Messmethoden für die Temperatur.....	4
3	Hardware .....	5
3.1	Arduino Uno R3 .....	5
3.2	Breadboard.....	7
3.3	Liquid Crystal Display .....	8
3.4	Sensoren .....	8
3.4.1	PH-Sensor .....	8
3.4.2	Temperatursensor.....	9
4	Ablauf der praktischen Arbeit .....	10
4.1	Erlernen der Grundlagen zur Hard- und Software .....	10
4.2	Sammlung des Materials .....	11
4.3	Bau .....	11
4.4	Gehäuse .....	12
4.5	Software .....	12
4.6	Ergebnisse und Vergleich mit anderen Geräten.....	15
5	Schlusswort.....	19
6	Danksagung .....	20
7	Quellenverzeichnis.....	21
7.1	Literaturverzeichnis .....	21
7.2	Internetquellenverzeichnis .....	21
7.3	Bildquellenverzeichnis: .....	21
8	Anhang.....	23
8.1	Programmcode: .....	23

# 1 Einleitung

---

## 1.1 Thema

---

### **Motivation**

Mir war schon von Anfang an klar, dass ich eine Arbeit in dem Gebiet Informatik schreiben wollte und habe dann von einer Lehrperson die Idee aufgeschnappt, etwas im Zusammenhang mit dem Arduino zu machen. Damals hatte ich die Idee eines Roboters, welche sich dann im Gespräch mit meiner Betreuungsperson schnell aufgelöst hatte. Da ich schon lange ein grosses Interesse an der Chemie habe war die Idee eines Geräts für das Chemielabor sehr willkommen und daraus entwickelte sich schlussendlich diese Arbeit.

### **Ziel**

Ziel dieser Maturaarbeit ist es ein Gerät zu bauen, welches die pH- und Temperaturmessung im Chemielabor verbessert. Dazu wird ein Mikrocontroller Board namens Arduino Uno R3, welches die Schaltzentrale des Endgeräts sein wird, verwendet. Dieses Board wurde ausgewählt, da es eines der bekanntesten und günstigsten Boards auf dem Markt ist. Als Sensor zur pH-Messung wird eine Silber-Silber-Chlorid Elektrode verwendet. Für die Messung der Temperatur kommt ein wasserdichter Wärmesensor zum Einsatz und zur Darstellung der Temperatur- und pH-Werte wird ein Liquid Crystal Display verwendet. Nach der Verknüpfung und Programmierung der Teile wurde ein Case mittels eines 3D-Druckers angefertigt um das Produkt für das Labor praktisch nutzbar zu machen. Zudem wurden Messungen durchgeführt und mit anderen digitalen Messgeräten für das Chemielabor in den Kategorien Preis und Funktionalität verglichen.

## 1.2 Struktur der Arbeit

---

Diese Arbeit ist in vier Teile aufgeteilt. Nach der Einleitung werden die nötigen Grundlagen zum pH-Wert geliefert und verschiedene Messmethoden für die Temperatur und den pH-Wert vorgestellt. Danach wird die Hardware thematisiert. Dort werden die verwendeten Teile genauer erläutert und ihre Funktion erklärt. Und als letzter grosser Teil wird der genaue Ablauf der praktischen Arbeit geschildert. Im Anhang findet man den Programmiercode des Endprodukts (TempHerino).

## 2 pH-Wert und Temperatur

---

### 2.1 Grundlagen zum pH-Wert <sup>1</sup>

---

Der pH-Wert gibt an wie stark die saure bzw. basische Wirkung einer wässrigen Lösung ist. Es gibt keine Einheit die den pH-Wert angibt und somit ist der eigentliche Wert eine dimensionslose Zahl. Eine Lösung mit dem Wert Sieben ist neutral, höher als Sieben ist basisch und tiefer bedeutet, dass sie sauer ist. Der pH-Wert setzt sich aus dem negativen Zehnerlogarithmus der Wasserstoffionen-Aktivität zusammen.

$$\text{pH} = -\log_{10} a(\text{H}^+)$$

Bei einfachen Berechnungen wird jedoch meistens eine vereinfachte Formel eingesetzt, die eine ziemlich genaue Näherung darstellt. Sie ersetzt die Wasserstoffionen-Aktivität mit der Oxoniumionkonzentration.

$$\text{pH} = -\log_{10} c(\text{H}_3\text{O}^+)$$

Ausserdem ist der pH-Wert temperaturabhängig. Das bedeutet, dass die obigen Formeln nur bei 25°C genau zutreffen. Dieser Umstand ist auch dafür verantwortlich, dass man für genaue Messungen die Messgeräte immer wieder neu kalibrieren muss.

### 2.2 Messmethoden für den pH-Wert

---

#### Potentiometrie

In der Potentiometrie wird eine Pufferlösung in eine Glasmembrankugel gegeben, welche dann in die zu messende Flüssigkeit eingetaucht wird. Die Wasserstoffionen haben die Tendenz sich an Silikatgruppen anzulagern, was zu einer Spannung zwischen der Innen- und der Aussenseite (Potential) der Glaskugel führt. Diese Spannung wird mittels Bezugselektroden gemessen und führt dadurch zum pH-Wert. Diese Methode wurde für den TempHerino verwendet, da man eine elektrisch messbare Spannung erhält, die für den Arduino wahrnehmbar ist.

Glasmembrankugel



Abbildung 1: PH-Elektrode als Beispiel für Potentiometrie

---

<sup>1</sup> <https://de.wikipedia.org/wiki/PH-Wert>

### Indikatorfarbstoffe

Man kann mit gewissen Farbstoffen innerhalb von kleinen Bereichen der pH-Skala einen Farbumschlag herbeiführen. Doch um die ganze Skala zu erfassen braucht man sogenannte Universalindikatoren. Dies sind Farbstoffgemische, welche bei verschiedenen pH-Werten gewisse Farben annehmen. Diese Methode wird meist in Form des sogenannten pH-Papiers verwendet. Auf diesem Papier befindet sich ein solcher Universalindikator. Somit hat das Papier je nach pH-Wert eine andere Farbe.



Abbildung 2: PH-Papier

## 2.3 Messmethoden für die Temperatur<sup>2</sup>

---

### Ausdehnungskoeffizienten von Flüssigkeiten

Es gibt Flüssigkeiten, die sich proportional zur Temperatur ausdehnen. Dies wird bei den klassischen Thermometern in Form von Alkohol oder Quecksilber ausgenutzt.

### Thermochromatische Farben

Bei den sogenannten thermochromatischen Farben wird vom Umstand Gebrauch gemacht, dass es Flüssigkeiten gibt, welche je nach Temperatur ihre Farbe ändern. Das funktioniert sehr ähnlich wie die oben beschriebenen Indikatorfarbstoffe.

### Messung elektrischer Größen

Es gibt zwei Arten die Temperatur anhand von elektrischen Größen zu messen. Entweder macht man vom temperaturabhängigen Widerstand von Leitern Gebrauch oder man verwendet ein Thermoelement, welches temperaturabhängige Spannungen verursachen kann. In dieser Arbeit wurde das Thermoelement im Sensor DS18B20 verwendet, da man wie beim pH-Wert auch eine für den Arduino messbare Spannung erhält.

---

<sup>2</sup> <https://de.wikipedia.org/wiki/Temperatur>

## 3 Hardware

### 3.1 Arduino Uno R3

Der Arduino Uno R3 ist ein Mikrocontroller Board der Physical-Computing-Plattform Arduino (seit März 2015 auch Genuino). Ein Mikrocontroller Board ist eine Halbleiterplatine (bzw. ein Board), welches einen Mikrocontroller darauf montiert hat. Im Falle des Arduino Uno R3 ist das der ATmega328. Ausserdem beinhaltet das Board 6 analoge Eingänge und 14 digitale Ein- bzw. Ausgänge, 3 Ground-Pins, einen 5 Volt- und einen 3.3 Volt-Pin, eine 2.1 mm Strombuchse, einen USB 2.0-Anschluss und 12 ICSP-Pins (welche bei meiner Arbeit nicht wichtig sind)<sup>3</sup>. Die verwendeten Teile werden nachfolgend genauer erklärt.

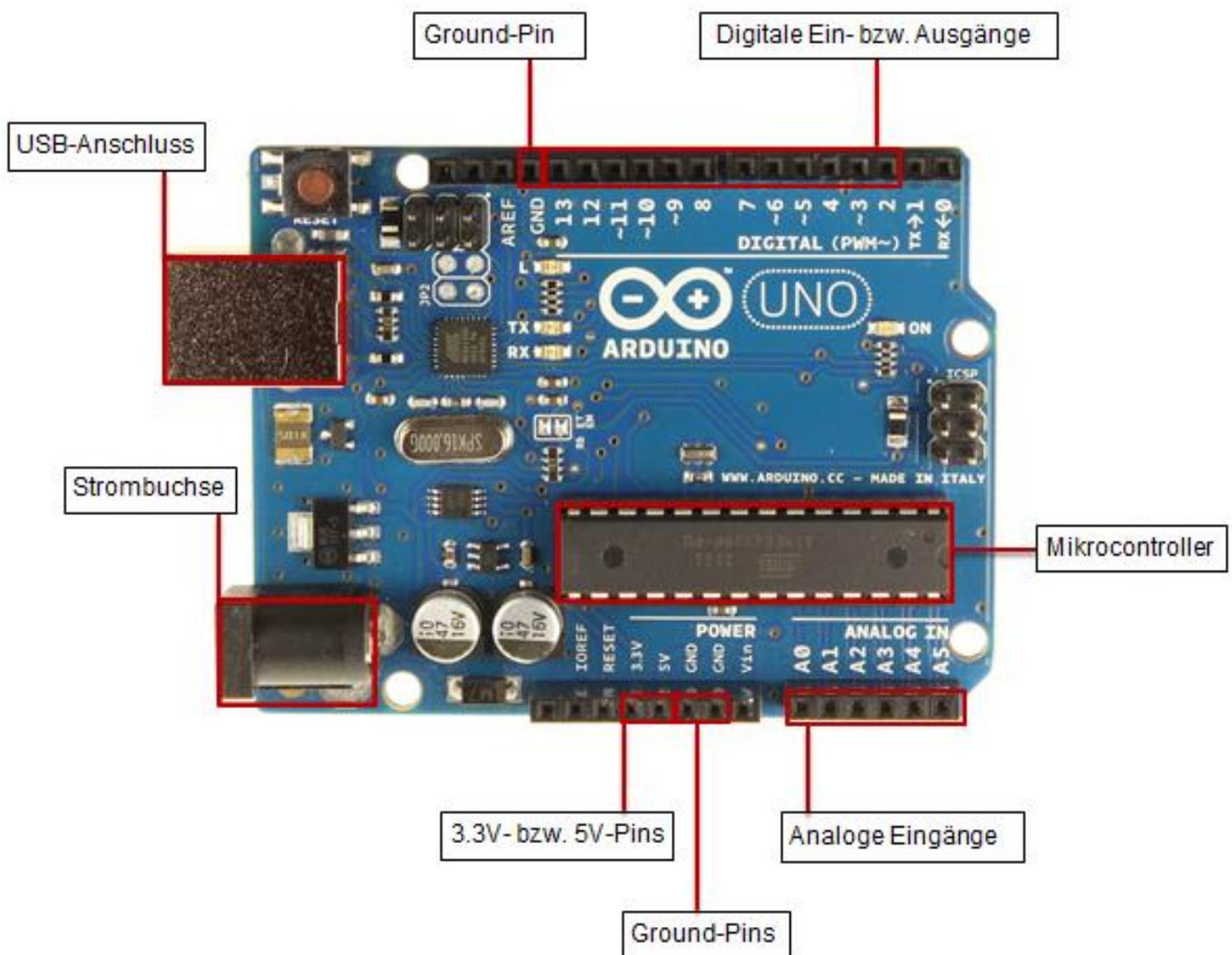


Abbildung 3: Arduino Uno R3

<sup>3</sup> <https://www.arduino.cc/en/Main/arduinoBoardUno>

## Der Mikrocontroller

Das ist der Mikrocontroller des Arduino Uno R3 Boards. Er ist das Zentrum des Arduinos, da alle Prozesse über den Mikrocontroller laufen. Zur Funktion eines Mikrocontrollers:



Abbildung 4: Mikrocontroller ATmega 328

Ein Mikrocontroller ist ein Ein-Chip-Computersystem. Das heisst es ist ein Mikrocomputer, welcher hauptsächlich über einen Chip arbeitet. Er beinhaltet sogar einen 8-Bit Prozessor, einen Arbeitsspeicher (2 Kilobytes) und einen Programmspeicher (32 KB)<sup>4</sup>. Diese sind bedeutend kleiner als bei alltäglichen Computer. Zum Vergleich: Ein durchschnittlicher Computer hat heutzutage ca. 500 Gigabyte Programmspeicher. Der ATmega328 hat mit seinen 32 Kilobytes nicht einmal einen Millionstel davon! Dieser Mikrocontroller ist über eine Halbleiterplatine mit den Peripheriefunktionen verbunden, welche ich im obigen Abschnitt schon aufgelistet habe.

### Die analogen Eingänge und digitalen Ein- bzw. Ausgänge

Die analogen Eingänge und digitalen Ein- bzw. Ausgänge werden zur Anschliessung von weiteren Peripheriegeräten gebraucht, wie zum Beispiel LEDs, Sensoren oder Druckknöpfe. Die digitalen Anschlüsse sind sogenannte I/O Pins, was bedeutet, dass sie sowohl Input (Eingang) als auch Output (Ausgang) sein können. Im Gegensatz dazu sind die analogen Anschlüsse nur Inputs.

Ein weiterer Unterschied der beiden Pin-Arten ist, dass der digitale Ein- bzw. Ausgang nur die Werte 5V und 0V bzw. HIGH und LOW annehmen kann und der analoge Eingang kann alle Werte von 0 bis 5V bzw. 0 bis 1024 annehmen. Deswegen werden Sensoren an den analogen Eingängen und Schalter an den digitalen Eingängen angeschlossen. Bei meiner Arbeit habe ich die digitalen Anschlüsse für die Druckknöpfe und das Liquid Crystal Display verwendet. Die analogen Anschlüsse kamen für den Temperatur- und den pH-Sensor zum Einsatz.

### Ground- und 5 Volt- bzw. 3.3 Volt-Pins

Die 5 Volt-Pins bilden die Stromversorgung und den elektrischen Pluspol für Schaltungen mit Peripheriegeräten. Der Ground-Pin ist die Erdung d.h. der elektrische Minuspol einer solchen Schaltung. Die beiden Pins werden mit einem Breadboard (siehe 3.2) verbunden und dort wird die Schaltung weiter aufgebaut.

### Die 2.1 mm Strombuchse

Diese Strombuchse kann zur Stromversorgung des Arduino gebraucht werden. Entweder man schliesst sie mit einem Netzgerät an die Steckdose an oder sie wird über Batterien mit Strom versorgt. Bei der Auswahl eines Netzgeräts muss beachtet werden, dass die ideale Spannung für den Arduino bei 7-12 Volt liegt.<sup>5</sup>

### Der USB-Anschluss

Der USB-Anschluss eines Arduinos kann zur Datenübertragung und/oder zur Stromversorgung verwendet werden.

<sup>4</sup> <http://www.atmel.com/devices/atmega328.aspx>

<sup>5</sup> <https://www.arduino.cc/en/Main/arduinoBoardUno>

## 3.2 Breadboard

Das Breadboard ist ein zentrales Teil meines Geräts, denn es ist die Verbindung zwischen dem Arduino Uno R3 Board und den Sensoren und Druckknöpfen. Grundsätzlich ist es eine Steckplatte mit verschiedenen Leitern, die es ermöglichen, einfacher komplexe Schaltungen aufzubauen.

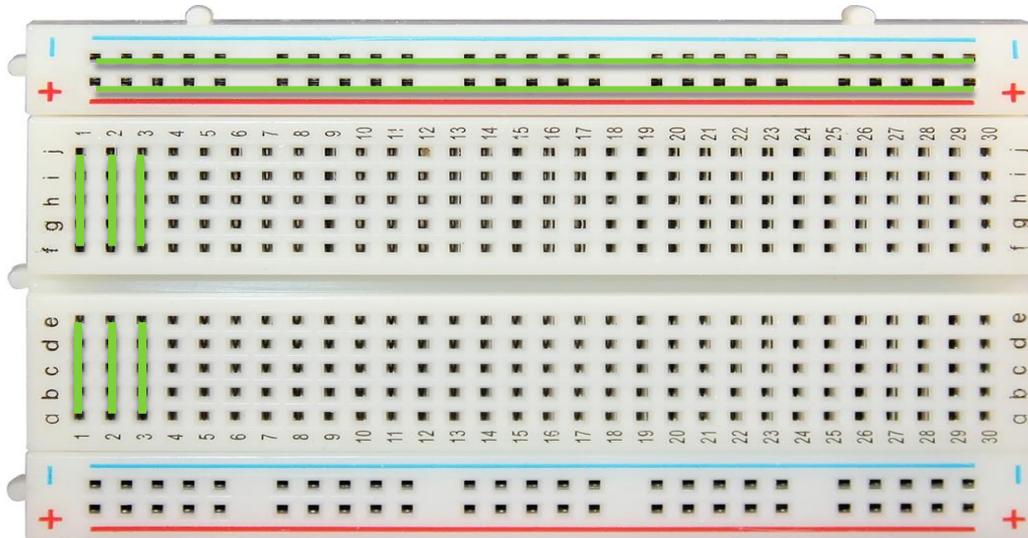


Abbildung 5: Breadboard mit Leiterverbindungen

Diese Leiter sind auf der Abbildung mit den grünen Linien markiert. Das System welches mit a-j in der vertikalen und 1-30 in der horizontalen beschriftet ist, wird gebraucht, um die Peripheriegeräte (Sensoren, Druckknöpfe und Display) anzuschliessen. Die beiden Systeme mit den langen + und - Leitern sind für die Stromversorgung zuständig.

Ich werde die Funktionsweise des Boards an dem Beispiel einer Schaltung eines einfachen Druckknopfs erklären.

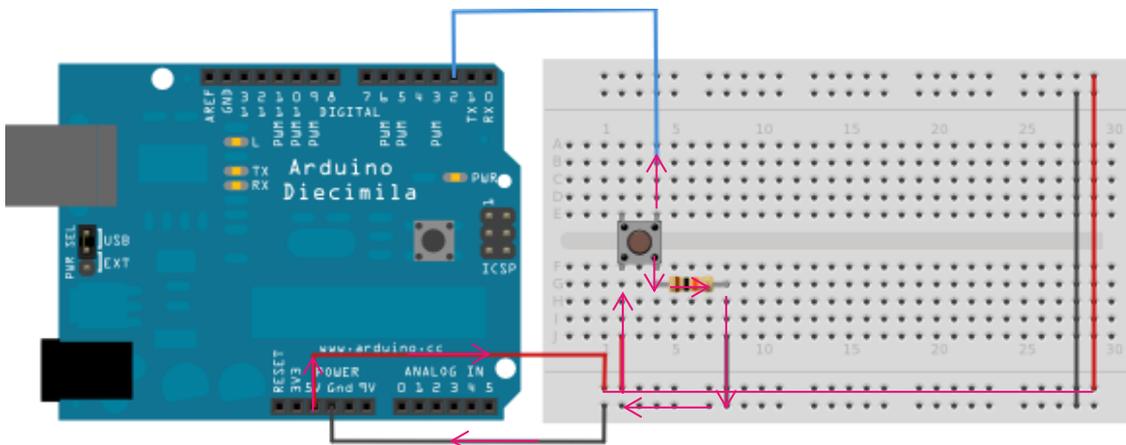


Abbildung 6: Schaltplan eines Druckknopfs

Wir verfolgen den Strom durch die Schaltung. Der Start ist beim 5V Pin auf der unteren Pinleiste. Von dort aus fließt der Strom durch das rote Kabel und den unteren positiven Leiter des Breadboards. Das heisst, dass dieser ganze Leiter mit Strom versorgt wird. Dieser Strom wird dann vom zweiten roten Kabel in die 4. Spalte des anderen Systems weitergeleitet. Dort trifft der Strom auf den Druckknopf. Wenn der Knopf gedrückt wird, kann der Strom durch den Widerstand über die 6. und dann die 10. Spalte in den negativen Leiter des anderen Systems fließen. Ausserdem fließt der Strom auch über die obere 6. Spalte in den Arduino zurück welcher den Strom wahrnehmen kann. Der negative Leiter ist mit dem GND Pin des Arduinos verbunden, welcher dann den Stromkreis schliesst.

## 3.3 Liquid Crystal Display

Das Liquid Crystal Display (LCD) wird genutzt, um Messwerte von Sensoren anzuzeigen. Seine Funktionsweise beruht darauf, dass Flüssigkeitskristalle die Polarisationsrichtung von Licht beeinflussen, wenn eine gewisse elektrische Spannung an sie angelegt wird. Einfach gesagt: Sie können durch Strom ihre Farbe ändern. Somit kann man die einzelnen Pixel beziehungsweise Kristall-bereiche mit Strom versorgen, um sie weiss werden zu lassen und so Buchstaben oder Zahlen anzuzeigen. Man muss nicht jeden Pixel selber ansteuern, sondern der Arduino hat in seiner Programmiersprache eine Bibliothek, welche einem das Anzeigen von Daten erleichtert. Auf den Anschluss des Displays möchte ich an dieser Stelle nicht eingehen, da das bei meiner Arbeit nicht sehr relevant ist und auch sehr viele spezielle Anschlüsse beinhalten würde.

## 3.4 Sensoren

### 3.4.1 pH-Sensor

Der verwendete Sensor ist der SEN0161 von der Firma DFRobot<sup>6</sup> aus Shanghai, da er einer der Wenigen pH-Sensoren für den Arduino und zudem relativ günstig ist. Der Sensor besteht aus einer pH-Elektrode, die über einen BNC-Anschluss (Bayonet Neill Conclleman), der hohe Frequenzen überträgt, mit dem pH Meter V1.0 Board verbunden ist, welches dann an den Arduino angeschlossen wird.

#### Die pH-Elektrode<sup>7</sup>

Die Elektrode des SEN0161 ist eine klassische Einstabmesskette, welche aus einem inneren Rohr und einem äusseren Mantel aufgebaut ist. Der äussere Mantel dient als Referenzelektrode (Silber-Silberchlorid-Elektrode) und besteht aus einem Silberdraht, Silberchlorid und einer Elektrolytlösung (Kaliumchlorid). Auch im inneren Rohr befinden sich Silberdraht, Silberchlorid und Kaliumchloridlösung, wobei dieses zusätzlich noch einen Puffer (Phosphatpuffer) enthält. Das innere Rohr ist durch eine Glasmembran mit der zu messenden Lösung verbunden, der äussere Mantel wiederum mit einem Diaphragma.

Das Potential entsteht dabei wie folgt:

Die Referenzelektrode steht über ein Diaphragma (meist Platinschwamm) in elektrischem Kontakt mit der Messlösung, wobei das Diaphragma Stoffaustausch mit der Lösung aber weitgehend unterbindet, um das Potential der Referenzelektrode nicht durch Fremdionen zu verändern. In dem Messstab befindet sich die Messelektrode in einer auf pH 7 eingestellten Phosphatpufferlösung. Diese steht durch eine sehr dünne Glasmembran in leitender Verbindung mit der Messlösung, an der das zur pH-Messung verwendete Potential entsteht. Die in der Glasmembran befindlichen Natrium- und Lithiumionen sind relativ frei beweglich, für Wasserstoffionen ist die Membran aber undurchlässig. Dennoch können die Wasserstoffionen Gitterplätze an den Sauerstoffanionen der unterkühlten Silikatschmelze einnehmen, da diese bei Kontakt mit der wässrigen Lösung an der Oberfläche aufzuquellen beginnt. Ein niedriger pH-Wert hat zur Folge, dass die Wasserstoffionen die Gitterplätze bevölkern und Natrium- und Lithiumio-

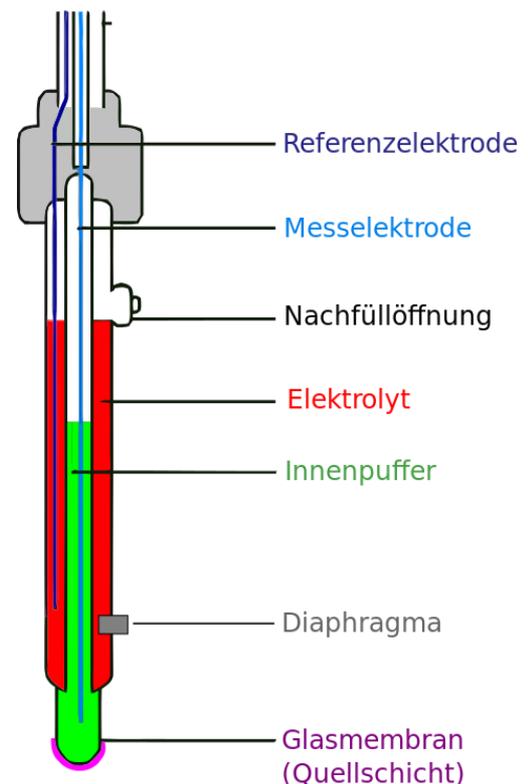


Abbildung 7: Einstabmesskette

<sup>6</sup> <http://www.dfrobot.com/index.php>

<sup>7</sup> <https://de.wikipedia.org/wiki/Glaselektrode>

nen in die Membran „zurückdrängen“. Da diese in der Membran frei beweglich sind, werden sie tendenziell auf die Innenseite der Membran geschoben, die gemessene Potentialdifferenz entsteht. Bei einem hohen pH überwiegt die Wasserstoffionenkonzentration im Inneren des Messstabes, der beschriebene Prozess läuft in anderer Richtung ab, das Potential entsteht mit anderem Vorzeichen. Diese Potentialdifferenz wird als elektrisches Signal an das PH-Meter V1.0 Shield weitergeleitet.

### **Das pH-Meter V1.0 Board**

Das pH-Meter V1.0 Board wandelt das BNC Signal der Elektrode in die drei Komponenten negativer, positiver und analoger Ausgang um. Der negative Ausgang wird mit dem Ground und der positive mit dem 5V-Pin verbunden. Über den analogen Ausgang kommen dann die elektrischen Signale, welche den pH-Wert darstellen. Dieser Ausgang wird mit dem analogen Eingang des Arduinos verbunden.

### **3.4.2 Temperatursensor**

---

Der in dieser Arbeit verwendete Temperatursensor ist der DS18B20 der Firma Maxim Integrated<sup>8</sup>, da er wie der SEN0161 für den Arduino gebaut wurde und für den Gebrauch im Chemielabor geeignet ist weil er mit Metall ummantelt und somit wasser- und säurefest ist.

Der Sensor arbeitet als Halbleiter-Temperatursensor auf der Basis von Halbleitermaterialien. Er erzeugt eine zur gemessenen Temperatur proportionale digital umgesetzte Spannung, die der Arduino dann auslesen und wieder in eine Temperatur umsetzen kann. Ausserdem arbeitet er mit der One-Wire Technologie von Maxim Integrated. Durch diese Technologie wird eine Datenader mit der Stromversorgung zu einem Kabel kombiniert. Diese zeigt ihre Vorteile jedoch nur bei Geräten mit mehreren One-Wire Sensoren und ist deswegen nicht relevant für diese Arbeit.

---

<sup>8</sup> <https://www.maximintegrated.com/en.html>



## 4.2 Sammlung des Materials

---

Das Zusammenstellen des Materials war ein wichtiger nächster Schritt für die Arbeit. Mit dem Arduino Starter Kit (vgl. Abbildung 8) hatte ich schon eine relativ grosse Sammlung an Material. Das Breadboard, der Arduino UNO R3 und das LCD sowie Kabel und Druckknöpfe waren vorhanden. Der pH-Sensor war die grosse Herausforderung. Über Google startete ich die Recherche nach einem geeigneten Sensor für meine Arbeit. Nach vielen Stunden der Evaluation stiess ich auf das Dormant Labs PH Module V2<sup>10</sup>. Dieser Sensor wäre geeignet gewesen, doch er war erst im September 2015 wieder erhältlich und kam deswegen nicht in Frage. Schlussendlich stiess ich auf den SEN0161 von DFrobot<sup>11</sup>, welchen ich dann auch bestellte. Auch bei diesem Produkt gab es verschiedene Probleme, da der Standort von DFrobot in Shanghai war und sie dazu den SEN0161 noch extra fertigestellen mussten, da es ein sehr selten angefordertes Produkt ist. Nach mehreren E-Mails mit der Produktionsleitung war der Umstand jedoch geklärt und drei Wochen später hatte ich den Sensor. Schlussendlich brauchte ich noch den Temperatursensor. Es gibt sehr viele Temperatursensoren für den Arduino auf dem Markt, doch solche die für Flüssigkeiten geeignet sind, sind sehr selten. Der DS18B20 war der einzig wirklich geeignete obwohl er den Nachteil hat, dass er die OneWire-Technologie<sup>12</sup> verwendete, welche für mein Projekt keine Verwendung fand.

## 4.3 Bau

---

Der Bau des Geräts (TempHerino) war eine mehrstufige Arbeit. Zuerst musste ich die Funktion der einzelnen Teile testen und somit alles einmal einzeln anschliessen um zu testen ob es funktioniert. Das hat bei allen Teilen gut funktioniert ausser beim Temperatursensor. Es gab dort mehrere Probleme: Einerseits da die Programmierung mit dem OneWire Protokoll für mich neu war andererseits, dass die Kabel um den Sensor anzuschliessen schlecht verarbeitet waren. Doch mit der intensiven Beschäftigung mit dem Protokoll und dem Ersetzen der Kabel waren diese Probleme gelöst.

Nach den Tests kam die wirkliche Planung. Bei dem Bau eines solchen Geräts ist es sehr wichtig platzsparend zu bauen. Das heisst auf einem kleinen Breadboard alle Geräte unterzubringen. Das stellte sich als relativ schwierig heraus. Ich musste die Struktur mehrere Male verändern bis es zum fertigen Produkt kam. Die letzte Frage beim Bau war die Frage der Stromzufuhr. Dort gibt es verschiedene Möglichkeiten wie Batterien, Solarzellen oder Netzteile. Für den TempHerino war das Netzteil (9V, 500mA) am praktischsten, da es nie aufgebraucht ist (wie Batterien) und immer eine Stromzufuhr liefert nicht wie bei Solarzellen. Doch um eine praktische Stromzufuhr zu erhalten musste noch ein Schalter eingebaut werden. Da der Arduino sein Programm wenn er Strom erhält automatisch ausführt wäre es nicht praktisch den TempHerino direkt an die Steckdose anzuschliessen, denn dadurch müsste man um ihn anzuschalten das Netzteil einstecken und um ihn abzuschalten es wieder herausziehen. Deshalb wurde ein Schalter in das Kabel des Netzteils eingebaut.

---

<sup>10</sup> <https://www.tindie.com/products/rezahussain/dormant-labs-ph-module-v2/>

<sup>11</sup> [http://www.dfrobot.com/index.php?route=product/product&product\\_id=1025#.VhttpL\\_ntlBc](http://www.dfrobot.com/index.php?route=product/product&product_id=1025#.VhttpL_ntlBc)

<sup>12</sup> <http://playground.arduino.cc/Learning/OneWire-DE>

## 4.4 Gehäuse

Für den TempHerino ist ein Gehäuse nötig und für dessen Produktion ist 3D-Druck gut geeignet. Das Modell für den Druck wurde in dem Freeware-Programm FreeCAD<sup>13</sup> gebaut und dann gedruckt. Da dieses Modell sehr genau sein musste wurden vorher genaue Messungen am TempHerino durchgeführt und dann die Masse auf das Modell übertragen. Der Druck des Modells dauerte 17 Stunden und 30 Minuten.

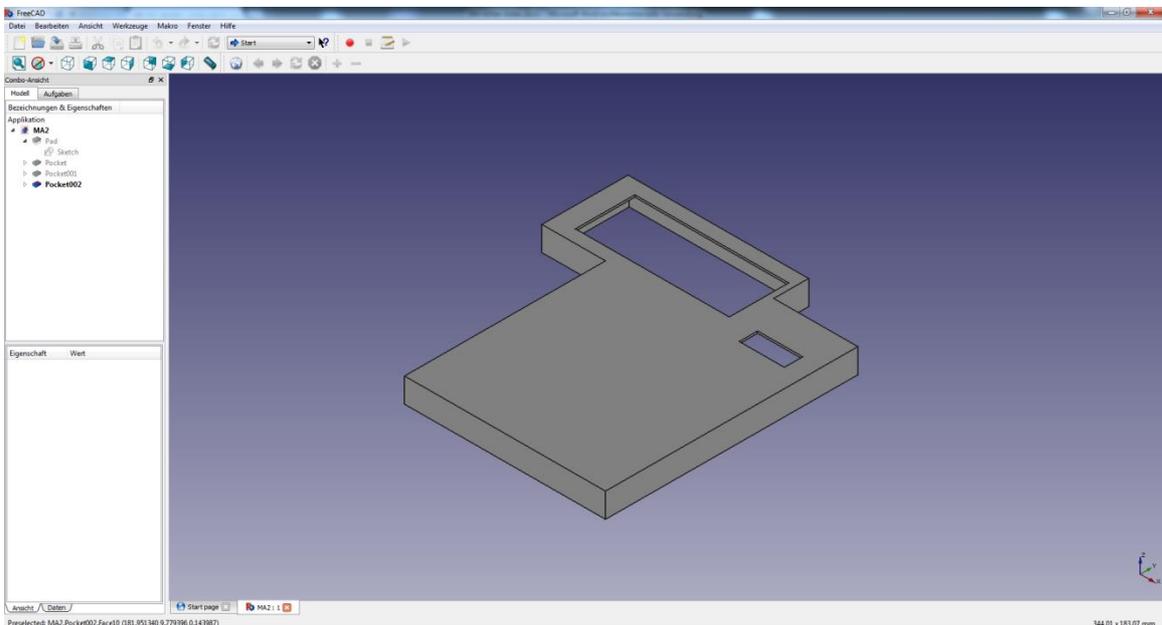


Abbildung 9: Gehäuse in FreeCAD



Abbildung 10: Gehäuse in FreeCAD

<sup>13</sup> <http://www.freecadweb.org/>

## 4.5 Software<sup>14</sup>

---

Arduino liefert eine integrierte Entwicklungsumgebung (IDE). Darin ist ein Code-Editor sowie verschiedene Arduino-Libraries enthalten, die das Programmieren erleichtern. Man kann auch verschiedene Libraries herunterladen und dann in die IDE speichern um sie in einem Programm zu verwenden. Die Programmiersprache basiert auf C und C++.

Für ein funktionstüchtiges Programm sind zwei Methoden vonnöten:

- Setup() – wird beim Start des Arduinos einmal ausgeführt und wird meistens benötigt um Pins den Peripheriegeräten zuzuordnen und diese dann ebenfalls zu starten
- Loop() – wird wiederholt solange das Arduino Board Strom hat

Hier ein Beispiel eines einfachen Programms mit dem man eine LED zum Blinken bringt:

```
int ledPin = 13; // die LED ist an Pin 13 angeschlossen, was in der Vari-
ablen ledPin gespeichert ist
void setup() {
    pinMode(ledPin, OUTPUT); // legt den LED-Pin als Ausgang fest
}
void loop() {
    digitalWrite(ledPin, HIGH); // LED anschalten
    delay(1000); // 1 Sekunde (=1000 ms) warten (delay() braucht den Para-
meter in ms)
    digitalWrite(ledPin, LOW); // LED ausschalten
    delay(1000); // 1 Sekunde warten
}
```

---

<sup>14</sup> [https://de.wikipedia.org/wiki/Arduino\\_\(Plattform\)#Software](https://de.wikipedia.org/wiki/Arduino_(Plattform)#Software)

Nach dieser kurzen Einführung wird jetzt noch eine wichtige Stelle aus dem Endcode (vgl. Anhang 8.1) des TempHerino erklärt. In diesem Auszug des Programmiercodes werden die Messwerte, je nachdem welcher Druckknopf betätigt wurde, auf dem LCD angezeigt. Mit der Variable „pHoderTemp“ wird entschieden ob der pH-Wert oder die Temperatur auf dem LCD angezeigt wird und durch die „clearvariable“ wird verhindert, dass bei jedem Durchlauf des Programms die Anzeige auf dem Display gelöscht wird, sondern nur wenn die Anzeige von pH auf Temperatur wechseln soll.

```

1 temps=digitalRead(4); // temps ist die Variable für den Taster um die Temperatur
anzuzeigen. Mit digitalRead wird gelesen ob der Taster gedrückt wird
2 phs=digitalRead(3); // phs ist wie temps einfach für den pH-Wert
3 if (phs==HIGH){pHoderTemp=1;}; // wenn phs gedrückt wird setze pHoderTemp auf 1
4 if (temps==HIGH){pHoderTemp=0;}; // wenn temps gedrückt wird setze pHoderTemp auf
0
5 if (pHoderTemp==0){
6     if (clearvariable==0){
7         lcd.clear();
8         clearvariable=1;}// wenn die clearvariable=0, wird der gesamte Inhalt
des Displays gelöscht und die clearvariable auf 1 gesetzt, damit es beim nächsten
Mal nicht wieder gelöscht wird
9     lcd.print(" Temperatur:");
10    lcd.setCursor(12,1);
11    lcd.print(celsius);} // auf dem LCD wird die Temperatur angezeigt, falls pHo-
derTemp=0
12 if (pHoderTemp==1){
13     if(clearvariable==1){
14         lcd.clear();
15         clearvariable=0;} // die clearvariable hat dieselbe Funktion wie oben,
doch 1 und 0 ist vertauscht
16    lcd.print("    PH-Wert:");
17    lcd.setCursor(12,1);
18    lcd.print(pHValue,2);} // pH-Wert wird angezeigt, falls pHoderTemp=1

```

In den ersten zwei Zeilen wird mit der Funktion digitalRead() ausgelesen ob einer der Schalter zur Änderung der Anzeige gedrückt wird. Danach wird, je nach Zustand der Schalter, zuerst der vorhergehende Inhalt des Displays durch lcd.clear() gelöscht, und dann mit lcd.print() der richtige Wert mit dem zugehörigen Text angezeigt. Damit jedoch nicht bei jedem Durchlauf des Programms der gesamte Inhalt des Displays gelöscht wird, wurde die clearvariable verwendet. Immer wenn das erste Mal eine neue Anzeige kommen soll wird der Code von Zeile 6-8 bzw. 11-13 ausgeführt (je nach Schalter). Dort wird zuerst der Wert der clearvariable geprüft und wenn sie den richtigen Wert (0 oder 1) hat, das Display geleert. Doch in der Zeile 8 bzw. 15 wird der Wert der clearvariable von 0 auf 1 bzw. von 1 auf 0 geändert, sodass bei der nächsten Änderung der Anzeige der Inhalt des Displays wieder entfernt werden kann und sich die Ausführung dieser drei Zeilen sich nicht wiederholt.

## 4.6 Ergebnisse und Vergleich mit anderen Geräten

Der Vergleich von diesem Gerät mit anderen ist relativ schwierig, da es kein wirklich vergleichbares Produkt auf dem Markt gibt. Ich werde zuerst das Gerät allgemein auf Genauigkeit testen und dann in den Kategorien Preis und Geschwindigkeit mit den von der Kantonsschule Schaffhausen verwendeten Geräten vergleichen.

### Ergebnisse

VOLTAGE (mV)	pH value	VOLTAGE (mV)	pH value
414.12	0.00	-414.12	14.00
354.96	1.00	-354.96	13.00
295.80	2.00	-295.80	12.00
236.64	3.00	-236.64	11.00
177.48	4.00	-177.48	10.00
118.32	5.00	-118.32	9.00
59.16	6.00	-59.16	8.00
0.00	7.00	0.00	7.00

Abbildung 11: pH-Spannungstabelle

Dies ist die Grundeinstellung des pH-Sensors. Das heisst bei dem Spannungswert 354.96 mV hat die Lösung einen Wert von 1. Diese Tabelle wurde dann anhand von Messungen überprüft und dabei kam dieses Diagramm zustande.

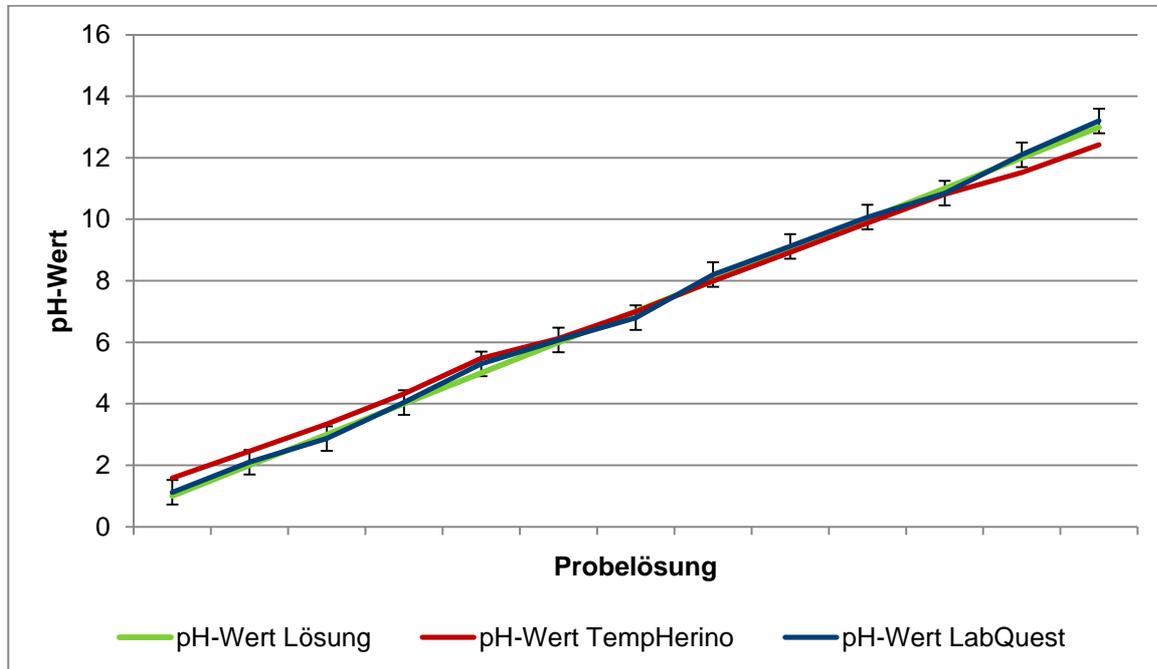


Abbildung 12: pH-Messung

Für diese Grafiken habe ich durch Verdünnung von NaOH und HCl mit destilliertem Wasser saure und basische Lösungen vom pH-Wert 1 bis 13 hergestellt. Danach habe ich die pH-Sonden vom LabQuest von Vernier<sup>15</sup> (siehe Vergleich mit anderen Geräten) und TempHerino gleichzeitig in die Lösung getaucht und dann die Werte notiert. Die Error Bars bei der Messung vom LabQuest stammen daher, dass er sich nicht auf einen festen Wert eingestellt hat. Der Wert veränderte sich innerhalb eines Bereichs von 0.4.

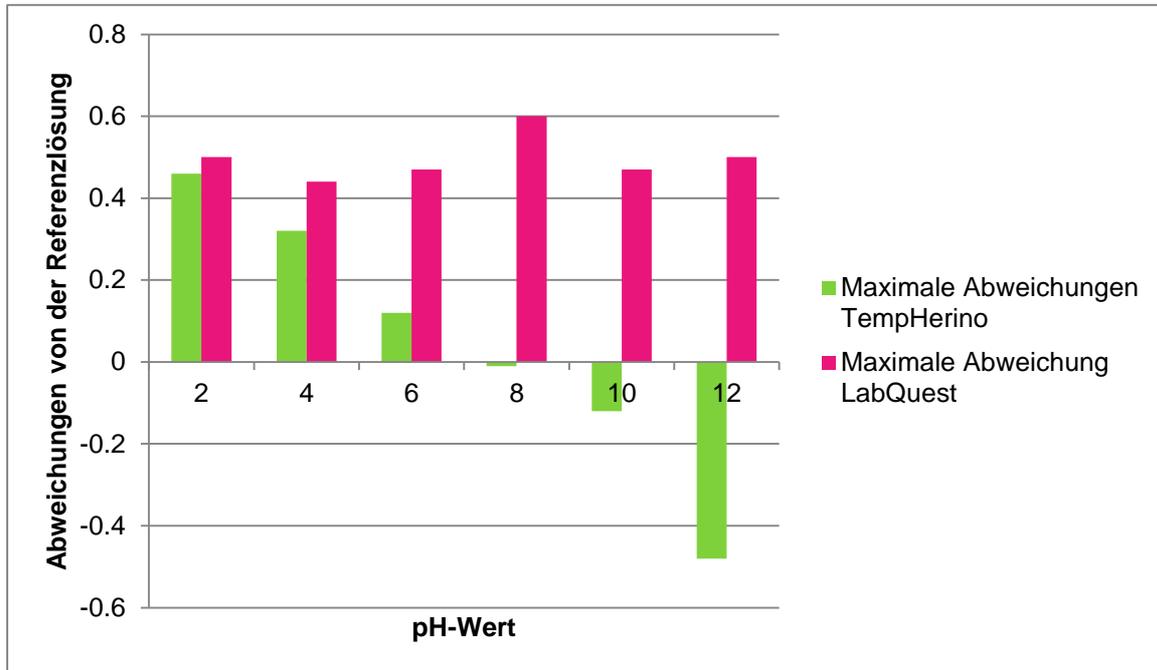


Abbildung 13: Abweichungen pH-Wert

In dieser Grafik werden die Abweichungen von der Referenzlösung genauer gezeigt. Der TempHerino zeigt eine klare Tendenz, dass er je weiter der pH-Wert von 7.5 entfernt ist desto grössere Abweichungen verzeichnet. Im Falle des LabQuest lassen sich keine Systematik in den Abweichungen finden. Die Abweichungen vom TempHerino wurden im Programmiercode (vgl. 8.1 Zeile 92-105) angepasst und so den richtigen Werten der Referenzlösung genähert. Durch diese Näherung hat das pH-Messergebnis vom TempHerino nur eine durchschnittliche Abweichung von 0.02.

Die Temperaturmessung wurde durch einen Vergleich mit einem elektronischen digitalen Thermometer von Amarell Electronics<sup>16</sup> verifiziert und hat keine grösseren Abweichungen gezeigt. Für die untenstehende Grafik wurde ein Behälter Wasser kontinuierlich auf einer Herdplatte erhitzt und bei den Punkten 5°C, 10°C ... 100°C (gemessen vom Thermometer von Amarell) der auf dem TempHerino angezeigte Wert aufgeschrieben.

<sup>15</sup> <http://www.vernier.com/>

<sup>16</sup> <http://www.amarell.de/>

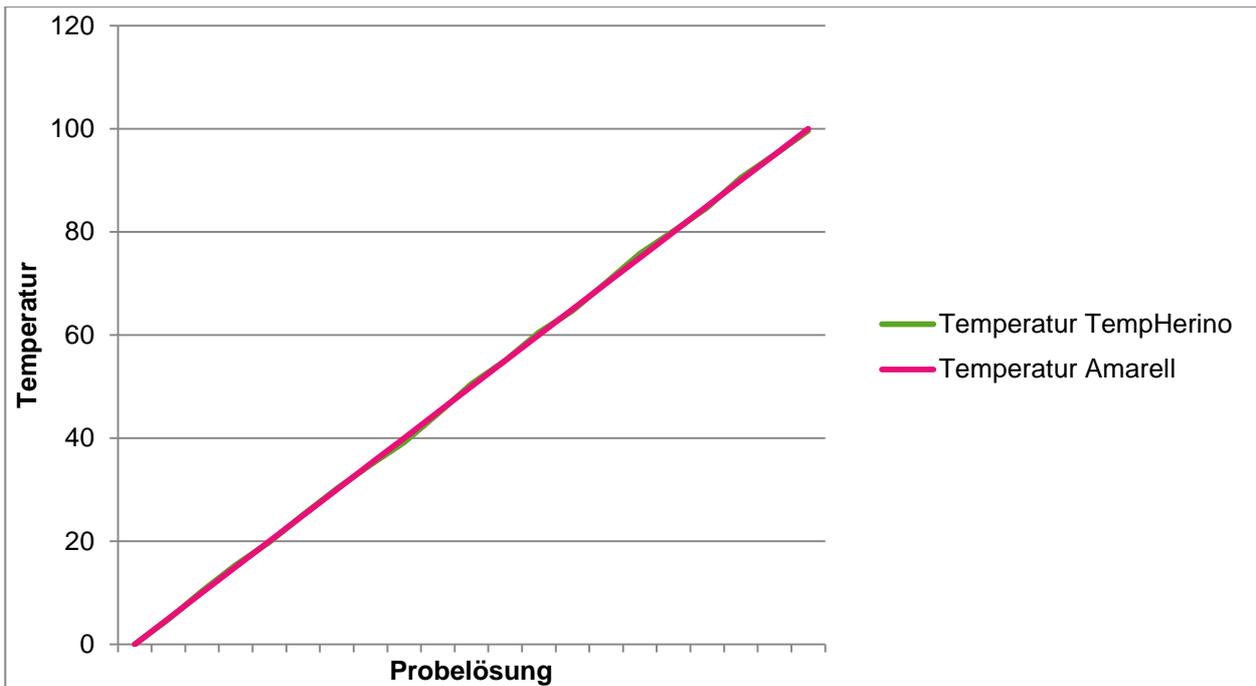


Abbildung 14: Temperaturmessung

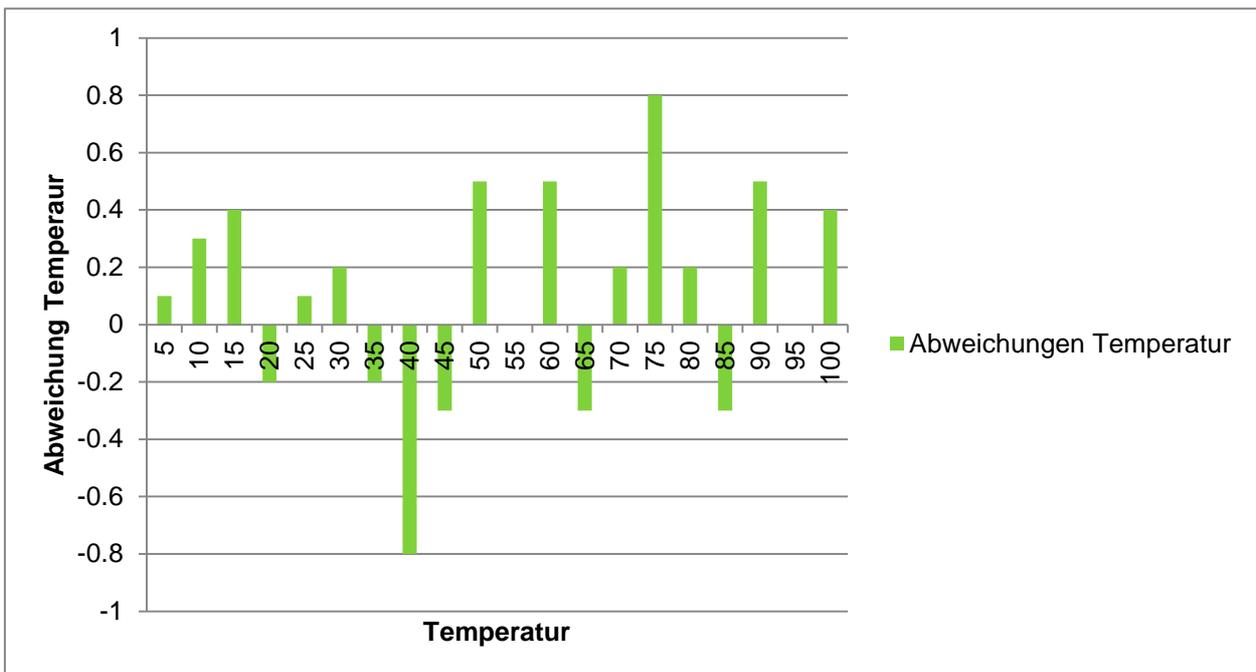


Abbildung 15: Abweichungen Temperatur

Wie man sieht sind die Abweichungen beim TempHerino minimal, sodass eine Näherung wie beim pH-Wert nicht vonnöten war. Der einzige Nachteil beim TempHerino ist, dass er, wenn er einen grösseren Temperaturunterschied messen muss, ca. 5 Sekunden braucht bis er sich richtig eingestellt hat. Dies ist auf die Geschwindigkeit des Sensors zurückzuführen sowie auf die Kombination von zwei Messmethoden in einem Gerät. Dies beeinflusst die Geschwindigkeit der Messung, da beide Werte gleichzeitig gemessen werden. Auch die Geschwindigkeit der pH-Messung wurde verlangsamt.

## Vergleich mit anderen Geräten

In diesem Teil der Arbeit wird der TempHerino mit den Produkten der Firma Vernier verglichen. Der grundsätzliche Unterschied zwischen den zwei Messgeräten ist, dass mein Gerät direkt den pH-Wert und die Temperatur anzeigt und bei den Sensoren von Vernier wird es in einem Programm angezeigt, in welchem man verschiedene Optionen zur Anzeige hat. Man kann dadurch zum Beispiel Titrationskurven zeichnen. Das Gerät besteht aus drei einzelnen Teilen: Der pH-Sonde mit Netzwerkanschluss, dem Temperatursensor ebenfalls mit Netzwerkanschluss und dem LabQuest. Der LabQuest hat eine ähnliche Funktion wie der Arduino in meinem Gerät. Es verwertet die Daten der Sensoren und zeigt sie dann auf einem Display an. Ausserdem könnte man durch den Netzwerkanschluss an den Sensoren diese auch an einen Computer anschliessen.



Abbildung 16: LabQuest 2

Kommen wir zu einem weiteren Vergleich. Zuerst wird die Kategorie Preis behandelt. Die Materialkosten meines Geräts setzen sich aus den folgenden Komponenten zusammen.

1 LCD:	24.5 Fr.	
1 Arduino:	23.86 Fr.	
1 pH-Sensor:	26.46 Fr.	
1 Temperatur-Sensor:	13.90 Fr.	
1 Breadboard:	3.92 Fr.	
25 Jumper wires:	1 Fr.	
3 Taster:	1 Fr.	
1 Potentiometer:	0.30 Fr.	
1 Case:	89.40 Fr.	
<hr/>		
Total:	184,34 Fr.	(Stand November 2015)

Endkosten des Geräts von Vernier:

1 pH-Sensor:	100.43 Fr.	
1 Temperatur-Sensor:	35.20 Fr.	
1 LabQuest:	465 Fr.	
<hr/>		
Total:	600.63 Fr.	(Stand November 2015)

Der Preis des Geräts von Vernier ist signifikant höher. Diese hat hauptsächlich zwei Gründe: Der LabQuest hat eine eigene Software, mit mehreren Funktionen. Bei den Geräten von Vernier sind die Kosten des Personals miteingerechnet. Das Produkt dieser Arbeit besteht aus den günstigsten, doch funktionellen Komponenten auf dem Markt.

Die andere Vergleichskategorie ist die Funktionalität. Die offensichtlichen Unterschiede wurden in den obigen Abschnitten schon thematisiert. Die spezifischen Unterschiede in der Messung sind die Messgeschwindigkeit und die Messgenauigkeit. Bei der Temperatur und dem pH-Wert sind die Sensoren von Vernier in der Messungsgeschwindigkeit klar überlegen. Die Messung mit dem LabQuest in Kombination mit den Sensoren ist mit ca. 2 Sekunden zum Zielwert etwa doppelt so schnell wie mit dem TempHerino. In der Genauigkeit ist mein Gerät jedoch überlegen. Der PH-Sensor von Vernier legt sich nicht auf einen genauen Wert fest. Die Anzeige verändert sich laufend innerhalb des Bereichs 0.4 (vgl. Abbildung 11). Im Gegensatz dazu hat der TempHerino eine maximale Abweichung von 0.02. Dadurch gelangt man zum Fazit, dass mein Gerät eine günstige und verlässliche Methode ist, den pH-Wert und die Temperatur zu messen und anzuzeigen. Jedoch stehen nicht zusätzliche Funktionen wie Kurvendiskussion, sondern nur der genaue Wert zur Verfügung.

## 5 Schlusswort

---

Das Ergebnis dieser Arbeit hat meine Erwartungen vollständig erfüllt, wenn nicht sogar übertroufen. Im ersten Schritt habe ich einen Einblick in die Welt der Elektronik sowie in die Programmierung von Arduino erhalten. Ausserdem entstand aus den gesammelten Erfahrungen ein preiswertes und funktionales Gerät für das Chemielabor der Kantonsschule Schaffhausen. Interessant wäre jetzt die Entwicklung einer zum Gerät passenden Software wie beim im letzten Kapitel beschriebenen LabQuest sowie die Industrialisierung des TempHerino.

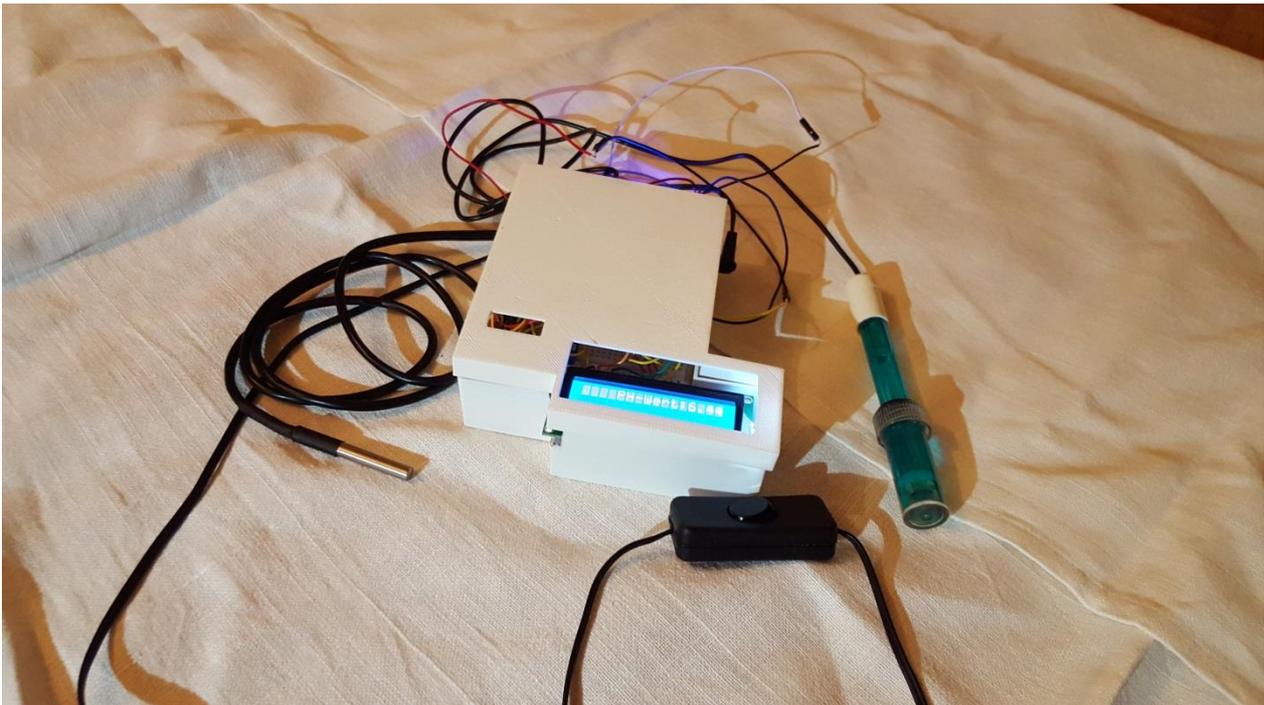


Abbildung 17: Endprodukt (TempHerino)

## 6 Danksagung

---

An dieser Stelle möchte ich allen danken, die zur Entstehung dieser Arbeit beigetragen haben. Zuerst möchte ich meiner Familie danken, die mir während meiner praktischen Arbeit das Wohnzimmer für meine chemischen Messungen und Elektronikbauteile zur Verfügung gestellt haben und mich auch sonst immer unterstützt haben. Dank gilt auch Tony Stamm aus der Firma EzToz, da ich dank ihm ein Gehäuse mit einem 3D-Drucker herstellen konnte. Ausserdem möchte ich meinem Betreuer Herr Steiger für die Beratung und Wegweisung während dem Entstehungsprozess danken.

## 7 Quellenverzeichnis

---

### 7.1 Literaturverzeichnis

---

Arduino Company: Arduino Projects Book, 2012

Knapp, Markus: Roboter bauen mit Arduino, 2014

### 7.2 Internetquellenverzeichnis

---

<http://www.vernier.com/>

[https://de.wikipedia.org/wiki/Arduino\\_\(Plattform\)#Software](https://de.wikipedia.org/wiki/Arduino_(Plattform)#Software)

<http://www.freecadweb.org/>

<https://www.tindie.com/products/rezahussain/dormant-labs-ph-module-v2/>

[http://www.dfrobot.com/index.php?route=product/product&product\\_id=1025#.VhttpL\\_ntlBc](http://www.dfrobot.com/index.php?route=product/product&product_id=1025#.VhttpL_ntlBc)

<http://playground.arduino.cc/Learning/OneWire-DE>

<https://www.arduino.cc/en/Main/ArduinoStarterKit>

<https://www.maximintegrated.com/en.html>

<http://www.dfrobot.com/index.php>

<https://de.wikipedia.org/wiki/Glaselektrode>

<http://www.atmel.com/devices/atmega328.aspx>

<https://www.arduino.cc/en/Main/arduinoBoardUno>

<https://www.arduino.cc/en/Main/arduinoBoardUno>

<https://de.wikipedia.org/wiki/PH-Wert>

<https://de.wikipedia.org/wiki/Temperatur>

Alle Internetquellen Stand 13.11.2015

### 7.3 Bildquellenverzeichnis:

---

Abbildung 1: [http://img.directindustry.de/images\\_di/photo-g/ph-elektrode-glas-kombiniert-15372-3309393.jpg](http://img.directindustry.de/images_di/photo-g/ph-elektrode-glas-kombiniert-15372-3309393.jpg)

Abbildung 2: <http://www.kochunterricht.ch/wp-content/uploads/2015/03/ph-papier.jpg>

Abbildung 3: <http://www.forefront.io/attachments/uno.jpg>

Abbildung 4: <http://www.digitalsmarties.net/products/atmega328>

Abbildung 5: <http://tinyurl.com/npr86dj>

Abbildung 6: <https://www.arduino.cc/en/uploads/Tutorial/button.png>

Abbildung 7: <http://tinyurl.com/p2m9cpq>

Abbildung 8: <https://www.adafruit.com/images/1200x900/1078-00.jpg>

Abbildung 16: [http://www.merlan.ca/productimages/huge/product.labq2\\_hero.001.jpg](http://www.merlan.ca/productimages/huge/product.labq2_hero.001.jpg)

Alle Bildquellen Stand 10.11.2015

# 8 Anhang

---

## 8.1 Programmcode:

---

```
1  #include <OneWire.h>
2  #include <LiquidCrystal.h>
3  OneWire ds(5);
4  LiquidCrystal lcd(6, 7, 8, 9, 10, 11);
5  #define SensorPin A0
6  #define LED 13
7  #define samplingInterval 20
8  #define printInterval 800
9  #define ArrayLenth 40
10 int pHArray[ArrayLenth];
11 int pHArrayIndex=0;
12 int kals=0;
13 int temps=0;
14 int phs=0;
15 int pHoderTemp=0;
16 int clearvariable=0;
17 float Offset=0.00;
18 void setup(void)
19 {
20 pinMode(2,INPUT);
21 pinMode(3,INPUT);
22 pinMode(4,INPUT);
23 pinMode(LED,OUTPUT);
24 }

25 void loop(void)
26 {
27 byte i;
28 byte present = 0;
29 byte type_s;
30 byte data[12];
31 byte addr[8];
32 float celsius;

33 int Kalibrierung=0;
34 int temps=0;
35 int phs=0;
36 static unsigned long samplingTime = millis();
37 static unsigned long printTime = millis();
38 static float pHValue,voltage;
39 if(millis()-samplingTime > samplingInterval)
40 {
41 if ( !ds.search(addr)) {
42 Serial.println("No more addresses.");
43 Serial.println();
44 ds.reset_search();
45 delay(0.001);
46 return;
47 }
48 ds.reset();
49 ds.select(addr);
50 ds.write(0x44, 1);
51 delay(1000);
52 present = ds.reset();
53 ds.select(addr);
54 ds.write(0xBE);
```

```

55 for ( i = 0; i < 9; i++) {
56 data[i] = ds.read();
57 int16_t raw = (data[1] << 8) | data[0];
58 if (type_s) {
59 raw = raw << 3;
60 if (data[7] == 0x10) {
61 raw = (raw & 0xFFFF0) + 12 - data[6];
62 }
63 }
64 else {
65 byte cfg = (data[4] & 0x60);
66 if (cfg == 0x00) raw = raw & ~7;
67 else if (cfg == 0x20) raw = raw & ~3;
68 else if (cfg == 0x40) raw = raw & ~1;
69 }
70 celsius = (float)raw / 16.0 ;

71 pHArray[pHArrayIndex++]=analogRead(SensorPin);
72 if(pHArrayIndex==ArrayLenth)pHArrayIndex=0;
73 voltage = avergearray(pHArray, ArrayLenth)*5.0/1024;
74 pHValue = 3.5*voltage-Offset;
75 Kalibrierung=digitalRead(2);

76 if(Kalibrierung==HIGH){
77 lcd.setCursor(1,1);
78 lcd.print("Kalibrierung. ");
79 delay(500);
80 lcd.setCursor(14,1);
81 lcd.print(".");
82 delay(500);
83 lcd.setCursor(15,1);
84 lcd.print(".");
85 delay(500);
86 Offset=pHValue-7;
87 lcd.clear();
88 samplingTime=millis();
89 }}

90 if(millis() - printTime > printInterval
91 {
92 if(pHValue>13){pHValue+=0.60;};
93 if (13>pHValue&& pHValue>12){pHValue+=0.53;};
94 if (12>pHValue&& pHValue>11){pHValue+=0.335;};
95 if (11>pHValue&& pHValue>10){pHValue+=0.155;};
96 if (10>pHValue&& pHValue>9){pHValue+=0.095;};
97 if (9>pHValue&& pHValue>8){pHValue+=0.04;};
98 if (8>pHValue&& pHValue>7){pHValue+=0.01;};
99 if (7>pHValue&& pHValue>6){pHValue-=0.055;};
100 if (6>pHValue&& pHValue>5){pHValue-=0.15;};
101 if (5>pHValue&& pHValue>4){pHValue-=0.35;};
102 if (4>pHValue&& pHValue>3){pHValue-=0.37;};
103 if (3>pHValue&& pHValue>2){pHValue-=0.40;};
104 if (1>pHValue&& pHValue>2){pHValue-=0.54;};
105 if (pHValue<1){pHValue-=0.60;};
106 temps=digitalRead(4);
107 phs=digitalRead(3);
108 if (pHs==HIGH){pHoderTemp=1;};
109 if (temps==HIGH){pHoderTemp=0;};

110 if (pHoderTemp==0){
111 if (clearvariable==0){

```

```

112 lcd.clear();
113 clearvariable=1;}
114 lcd.print(" Temperatur:");
115 lcd.setCursor(12,1);
116 lcd.print(celsius);}
117 if (pHoderTemp==1){
118 if(clearvariable==1){
119 lcd.clear();
120 clearvariable=0;}
121 lcd.print(" PH-Wert:");
122 lcd.setCursor(12,1);
123 lcd.print(pHValue,2);}
124 digitalWrite(LED,digitalRead(LED)^1);
125 printTime=millis();
126 }
127 }
128 double avergearray(int* arr, int number){
129 int i;
130 int max,min;
131 double avg;
132 long amount=0;
133 if(number<=0){
134 Serial.println("Error number for the array to avraging!/n");
135 return 0;
136 }
137 if(number<5){
138 for(i=0;i<number;i++){
139 amount+=arr[i];
140 }
141 avg = amount/number;
142 return avg;
143 }else{
144 if(arr[0]<arr[1]){
145 min = arr[0];max=arr[1];
146 }
147 else{
148 min=arr[1];max=arr[0];
149 }
150 for(i=2;i<number;i++){
151 if(arr[i]<min){
152 amount+=min; //arr<min
153 min=arr[i];
154 }else {
155 if(arr[i]>max){
156 amount+=max; //arr>max
157 max=arr[i];
158 }else{
159 amount+=arr[i]; //min<=arr<=max
160 }
161 }//if
162 }//for
163 avg = (double)amount/(number-2);
164 }//if
165 return avg;
166 }

```