

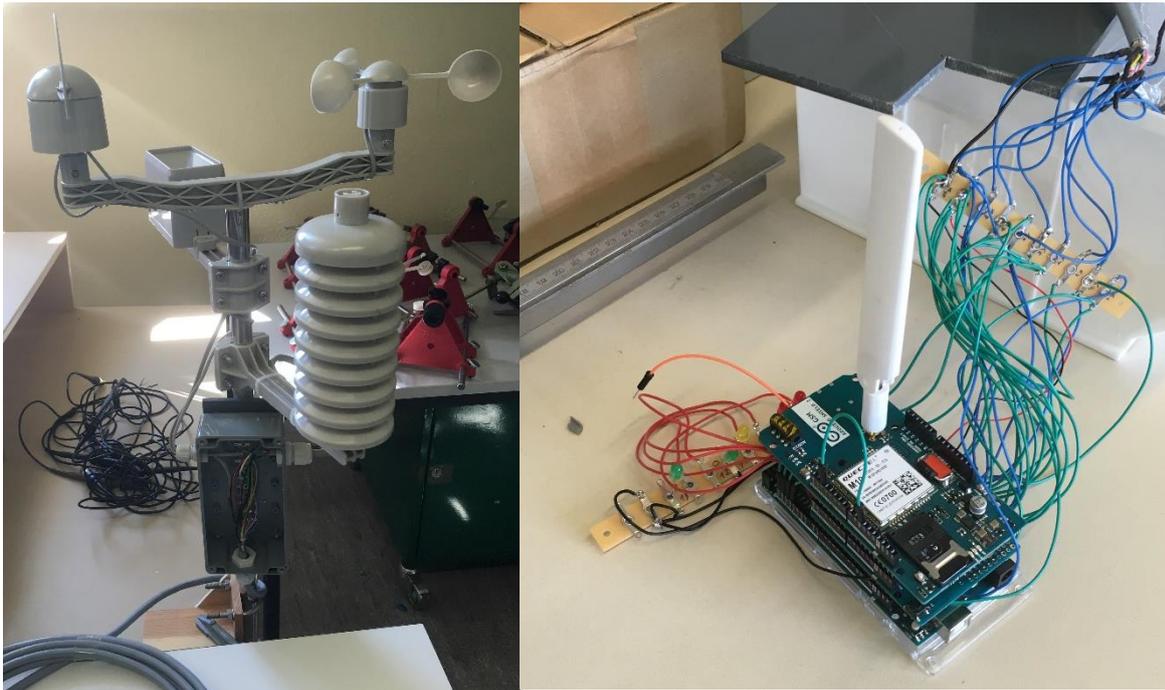
# Entwicklung einer Wetterstation mit Datenupload über Mobilfunk

Von Andreas Jürgens und Philipp Berdesinski

Prof. Dr. Götz Lipphardt

Hochschule Mannheim

In Kooperation mit dem Hector-Seminar



## Entwicklung einer Wetterstation mit Datenupload über Mobilfunk (Bericht)

### Inhalt

1. Ziele und Ideen.....	2
2. Methodik .....	2
3. Aufbau der Wetterstation .....	3
4. Hardware-Komponenten .....	3
4.1 Arduino.....	3
4.2 GSM-Shield.....	4
4.3 Proto-Shield.....	4
4.4 WH1080 .....	4
4.4.1 Regensensor .....	4
4.4.2 Windgeschwindigkeit.....	5
4.4.3 Windrichtung.....	5
4.5 BMP180 .....	5
4.6 SHT21 .....	6
4.7 5TM.....	6
4.8 Pyranometer .....	7
5. Aufbau der Schaltung.....	8
6. Software der Wetterstation.....	10
6.1. Das Setup.....	11
6.2. Die zyklische Routine .....	12
7. Zusammenfassung .....	12
8. Ausblick .....	13
9. Danksagung .....	13
10. Anhang .....	14
11. Quellenverzeichnis .....	15

## 1. Ziele und Ideen

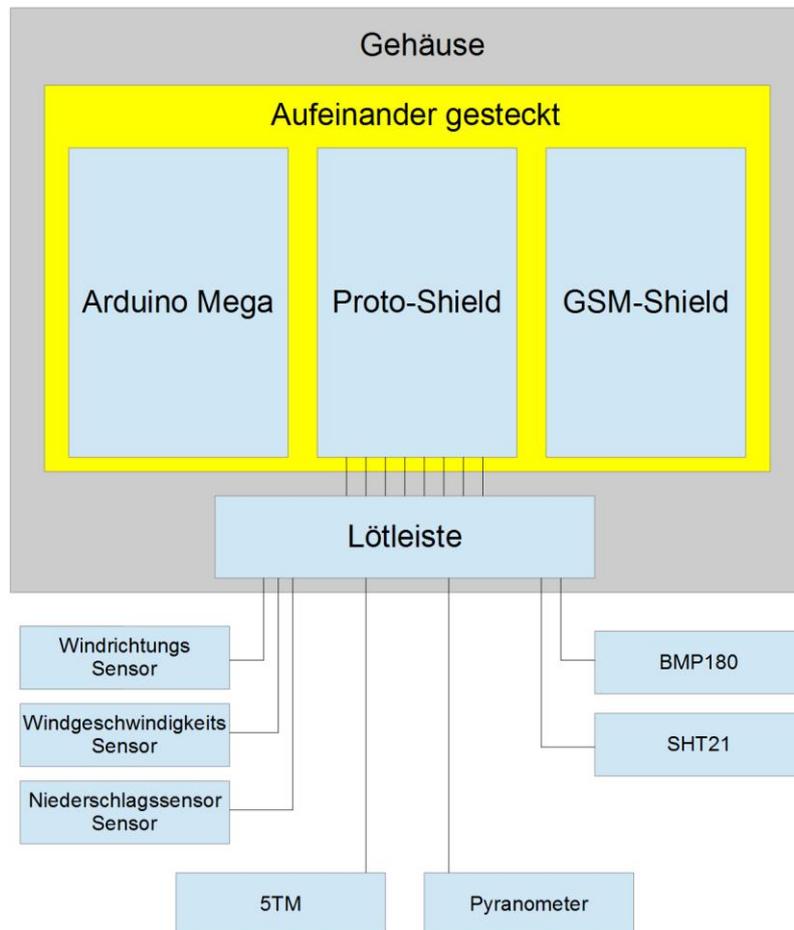
Das eigentliche Ziel unserer Projektgruppe unter der Leitung von Prof. Dr. Götz Lipphardt an der Hochschule Mannheim war der Bau einer autark arbeitenden Wetterstation mit verschiedenen Sensoren, welche nach dem Erhalt der Wetterdaten diese an einen Server senden sollte. Die einzelnen Sensoren sollten hierbei ihre gemessenen Daten an einen Arduino zur Weiterverarbeitung senden, welcher diese dann an den Server übermitteln sollte. Vorgesehene Messwerte waren:

Messwert	Sensor
Temperatur	SHT21,BMP180
Luftdruck	BMP180
Windrichtung	WH1080 Windrichtungssensor
Windgeschwindigkeit	WH1080 Windgeschwindigkeitssensor
Luftfeuchte	SHT21
Regenmenge	WH1080 Niederschlagssensor
Bodenfeuchte	5TM
Bodentemperatur	5TM

## 2. Methodik

Die Arbeit sollte hierbei einmal wöchentlich in einem Drei-Stunden-Treffen erfolgen und sich in mehrere Aufgabengebiete unterteilen: programmieren, sowie löten und aufbauen der einzelnen Schaltungen. Drei Sensoren wurden hierbei aus einem vorher demontierten Wetterstationen Bausatz entnommen. Eine Vielzahl der Sensoren war schon von Anfang an mit dem Arduino kompatibel und sollte im Verlauf des Programmierens mit eingebunden werden. Verschiedene LEDs sollten dabei die richtige Funktion der einzelnen Abläufe darstellen (Serveraustausche der Daten, Messungen usw.). Ein Gehäuse sollte die Elektronik vor Wind und Wetter schützen wobei sich Elektronik und Messinstrumente in verschiedenen, voneinander getrennten Orten, befanden. Die anfängliche Idee einer automatischen Ventilsteuerung aufgrund der gesammelten Wetterdaten wurde durch eine manuelle Bedienung dessen durch den Nutzer ersetzt.

### 3. Aufbau der Wetterstation



Alle Sensoren sind über eine Löt-leiste mit dem Arduino Proto-Shield verbunden. Dieses beinhaltet alle zur Funktion notwendigen Komponenten (Widerstände, den Operationsverstärker, Schmitt-trigger, die Diode und den Kondensator). Die Verbindung zur Löt-leiste verbindet lediglich Sensorabgabe und Stromverbindung mit dem Arduino Proto-Shield.

### 4. Hardware-Komponenten

#### 4.1 Arduino

Der Arduino Mega ist das Herzstück des Projekts. Es stellt die Recheneinheit dar, auf welcher das Programm läuft. Auf ihn können die Arduino-Shields gesteckt werden. Die Programmierung erfolgt über den Hauseigenen Arduino-Editor. Die Programmiersprache ist eine leicht veränderte Variante von C.

## 4.2 GSM-Shield

Das GSM-Shield ist ein Modem, über welches SMS empfangen, gesendet und eine Internetverbindung hergestellt werden können. Es wird über die Umfangreiche GSM-Library bedient, welche auch leicht editiert wurde, um Überschneidende Interrupts zu verhindern.

## 4.3 Proto-Shield

Der Proto-Shield dient dazu, einfach Verbindungen zwischen losen Kabeln und den Pins des Arduino herzustellen. Auch ganze Schaltungen finden auf diesem Gerüst Platz.

## 4.4 WH1080

Der WH1080 Wettermast ist eine Sensoreinheit für kleinere Wetterstationen. Sie enthält einen Windrichtungssensor, einen Windgeschwindigkeitssensor, einen Regensensor und weitere Sensoren, welche leider nicht benutzt werden konnten.

### 4.4.1 Regensensor

Der Regensensor besteht aus einem Trichter und einer Wippe, welche nach einer bestimmten Menge Wasser kippt und einen Kontakt schließt. Da der Kontakt nicht direkt und abrupt schließt, musste eine weitere Schaltung den Schalter

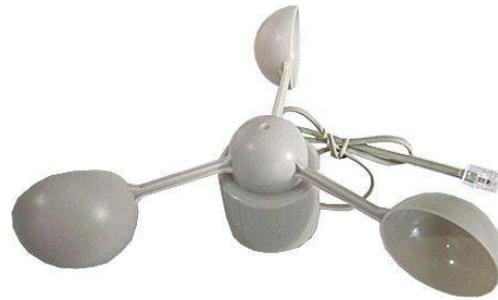


<https://images.ges.cz/images/pictures/0/08105172.jpg>

„debouncen“, was heißt, aus dem gestörten Signal ein einziges, kontinuierliches zu erzeugen. Pro 0.2794mm/m<sup>2</sup> kippt der Sensor ein Mal. Diese Signale zählt der Arduino über einen Interrupt und zählt für das Hochladen immer den Niederschlag der letzten Stunde zusammen. Der Wetterserver akzeptiert den Niederschlag in Inches, was 0,011" pro Signal entspricht.

## 4.4.2 Windgeschwindigkeit

Der Windgeschwindigkeitssensor besteht aus einem Rotierenden Rad aus 3 Bechern, welche den Wind einfangen. Pro Umdrehung des Rades wird ein Puls generiert. Von der Pulszahl pro Sekunde lässt sich die Windgeschwindigkeit errechnen. Alle 1,492 MPH



<https://images.ges.cz/images/pictures/0/08105170.jpg>

wird ein Puls generiert. Auf Anfrage errechnet der Arduino so die Windgeschwindigkeit.

## 4.4.3 Windrichtung

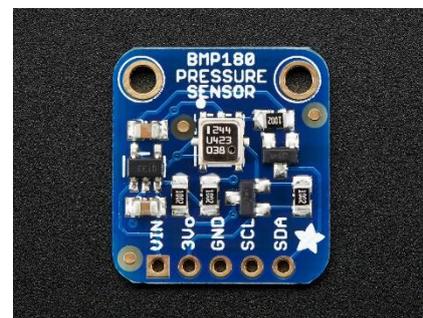
Der Windrichtungsmesser basiert auch auf einem Rad, welches sich durch Finnen immer in den Wind richtet. Ein Magnet auf dem Rad schließt so immer einen Stromkreis in der Basis des Sensors. Aus unterschiedlichen Widerständen in den verschiedenen Stromkreisen kann so die Ausrichtung des Sensors über einen Spannungsteiler errechnet werden.



<http://i.ebayimg.com/images/g/U--oAAOSwd4tT6Oza/s-l300.jpg>

## 4.5 BMP180

Der BMP180 ist ein Sensor zur Messung von Druck und Temperatur. Die Kommunikation zwischen Sensor und Arduino erfolgt über die Adafruit-BMP085-Library. Diese übernimmt das Kommunizieren, was alternativ über das Grundgerüst Wire.h erfolgen könnte, auf dem die Verbindung beruht. Vor dem Hochladen



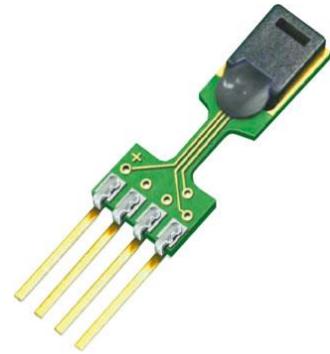
[https://c1.staticflickr.com/6/5769/23346083742\\_b27052dcc9\\_b.jpg](https://c1.staticflickr.com/6/5769/23346083742_b27052dcc9_b.jpg)

werden die Daten in vom Server benötigte Einheiten verrechnet. Die eigentliche Messung von Temperatur und Druck beruht auf den besonderen Eigenschaften von Piezo-Elementen. Dies sind Kristalle, welche unter äußeren Einflüssen Spannung

generieren. So wird der Luftdruck erkannt und die Temperatur über die Ausdehnung eines Materials bei Hitze bzw. das Zusammenziehen selbiges bei Kälte gemessen.

## 4.6 SHT21

Der SHT21 funktioniert ähnlich dem BMP180 über eine Bibliothek. Die SHT2x-Arduino-Library übernimmt hier das Auslesen von Temperatur und Luftfeuchte. Die Temperatur wird hier auch in Grad Fahrenheit umgewandelt während die Luftfeuchte direkt in % angegeben wird. Die Bibliothek wurde minimal geändert um einen unendlichen Loop zu verhindern, welcher



[http://www.tme.eu/html/gfx/ramka\\_2244.jpg](http://www.tme.eu/html/gfx/ramka_2244.jpg)

dann auftreten kann, wenn es Verbindungsprobleme gibt. Die Temperatur wird mit einem Halbleiterelement gemessen, welches durch die Temperatur eine Spannung generiert, welche intern gemessen und verarbeitet wird. Die Luftfeuchte wird über einen Kondensator gemessen. Die Luftfeuchtigkeit beeinflusst dessen Kapazität, was wiederum intern ausgewertet wird.

## 4.7 5TM

Der Sensor 5TM der Firma Decagon ist für das Messen der Bodenfeuchtigkeit und der Bodentemperatur ausgelegt. Das Auslesen der Daten erfolgt hier über die SDISerial-Library, welche geringfügig geändert wurde um Überschneidungen der Interrupts mit der GSM library zu verhindern.



[https://www.decagon.com/media/filer\\_public/58/d6/58d6d188-d99a-4f60-994b-a03d8c61265d/5tm-web.jpg](https://www.decagon.com/media/filer_public/58/d6/58d6d188-d99a-4f60-994b-a03d8c61265d/5tm-web.jpg)

Die Bodentemperatur wird hier ebenfalls zum Hochladen in Grad Fahrenheit umgewandelt. Die Messungen von Temperatur und Bodenfeuchte basieren auf dem gleichen Prinzip wie die Messungen des SHT21.

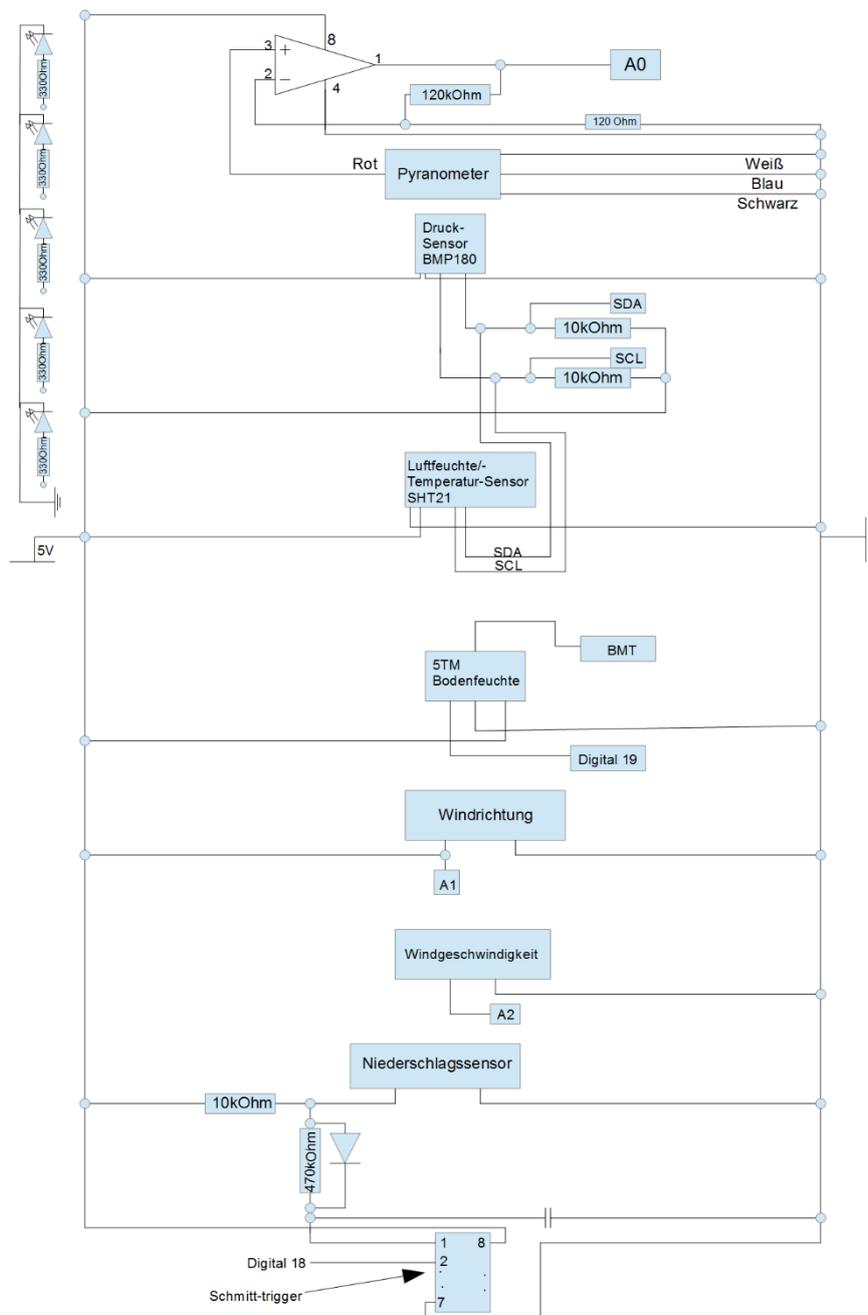
## 4.8 Pyranometer

Das Pyranometer gibt eine Spannung proportional zur Sonneneinstrahlung aus, welche mithilfe eines Operationsverstärkers gemessen werden kann. Die Sonneneinstrahlung wird dabei komplett absorbiert und die Wärme über eine Thermosäule in Spannung umgewandelt. Die gemessene Spannung lässt die Bestimmung der Sonneneinstrahlung zu.



[https://upload.wikimedia.org/wikipedia/commons/5/53/Hukseflux\\_radiometer\\_sr11\\_photo.jpg](https://upload.wikimedia.org/wikipedia/commons/5/53/Hukseflux_radiometer_sr11_photo.jpg)

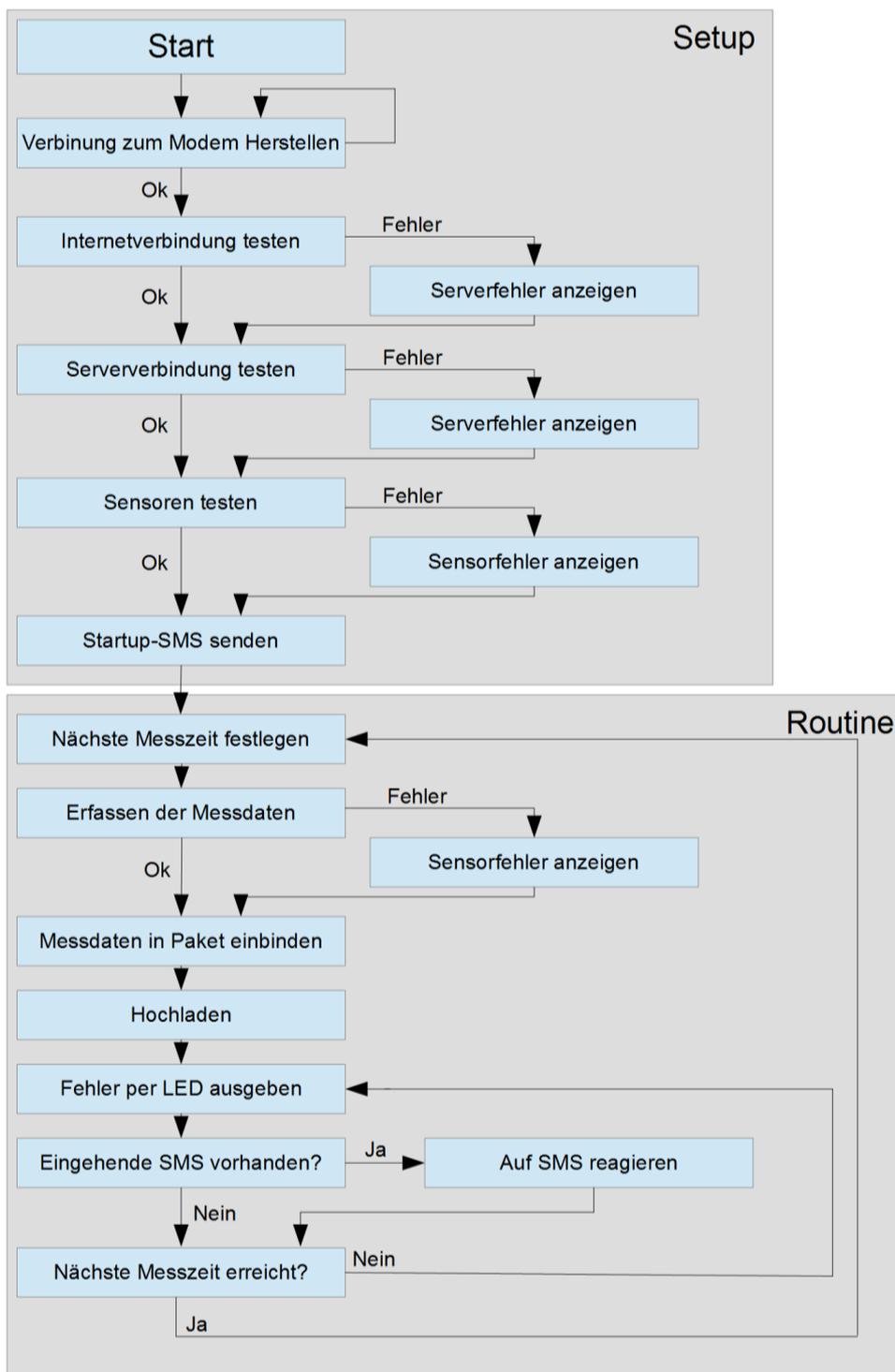
## 5. Aufbau der Schaltung



Die Sensoren: Das Pyranometer, der Luftdrucksensor BMP 180, Luftfeuchte/-Temperatursensor SHT21, Bodenfeuchtesensor 5TM, sowie die vom Wetterstationenpaket entnommenen Sensoren (ein Windrichtungssensor, Windgeschwindigkeitssensor, Regenmengensensor), sollten alle an zentraler Stelle über verschiedene Leisten mit dem Arduino verbunden werden. Die Funktion der Komponenten soll dabei von fünf LEDs gezeigt werden. Die finale Schaltung besteht letztendlich aus sieben verschiedenen größtenteils voneinander abgeschotteten Schaltungen. Das Pyranometer, für die Bestimmung der Sonneneinstrahlungsintensität, ist das CM11 von Kipp und Zonen. Dieses sehr

empfindliche Gerät gibt die gemessenen Werte in wenigen Mikrovolt aus, daher wurde ein Operationsverstärker verwendet um diese Mikrovolt zu verstärken, als nichtinvertierender Verstärker. Der Luftdrucksensor (BMP180) und der Luftfeuchtesensor (SHT21) wurden gemeinsam mit dem Arduino verbunden. Der Bodenfeuchtesensor (5TM) und der Windrichtungssensor werden einzeln mit dem Arduino verbunden, wobei der Windgeschwindigkeitssensor ohne eine Stromverbindung funktionierte (eine Umdrehung zeigt eine Spannungsspitze). Der Regensensor benötigte aufgrund eines Konstruktionsfehlers einen Schmitt-Trigger um seine Signale empfangen zu können Die Sensoren wurden durch Lötleisten vom Arduino getrennt. Alle Sensorkabel wurden zentral in dem Hauptgehäuse der Elektronik verbunden. Die Sensoren wurden auf einem, dem WH1080-Wettermast entnommenen, Geräteständer montiert.

## 6. Software der Wetterstation



Das Programm der Wetterstation auf dem Arduino Mega besteht aus zwei Hauptteilen: Einem Setup und einer zyklischen Routine. Die exakte Funktionsweise jedes Abschnitts ist im Code auf Englisch dokumentiert. Im Folgenden ist die Funktionsweise zusammengefasst.

Das Setup sorgt unter anderem dafür, dass alle Komponenten erkannt und überprüft werden. Dies kann jedoch nicht bei allen Komponenten angewendet werden. Mehr dazu folgt in den entsprechenden Unterkapiteln.

Die Zyklische Routine ist für das Sammeln der Daten zuständig. Sie überprüft die Daten auf ihre Plausibilität und erstellt ein Update-Paket, welches am Ende der Routine an den WeatherUnderground-Server gesendet wird.

Um Fehler in der Übertragung zu vermindern wird der Arduino bei jeder Internetanfrage 5mal versuchen, eine Verbindung herzustellen.

## 6.1. Das Setup

Im Setup werden zuerst die 5 LED-Pins und die 4 Pins zur Ventilsteuerung als Output designed. 3 weitere Pins werden als Input für den Regensensor, den Windrichtungssensor und den Windgeschwindigkeitssensor bestimmt. Hier wird auch die Power-LED aktiviert und die Ventile werden geschlossen, da diese möglicherweise nach einem Absturz noch offen sind.

Ein Interrupt zählt, wie oft der Regensensor aktiviert wird, wobei der derzeitige Zustand des Arduinos keine Rolle spielt. Um Überschneidungen mit der GSM-Bibliothek zu verhindern, wurde diese editiert um gewisse Interrupt-Ports freizugeben.

Nachdem eine Verbindung zum GSM-Modem hergestellt ist, wird die GSM-LED eingeschaltet.

Eine Synchronisation der modeminternen Uhr zeigt auch an, ob eine Internetverbindung besteht. Die Verbindung zum Wetterserver wird mit einer Dummy-Anfrage getestet.

Aufgrund der Messmethoden können nur der BMP180, der SHT21, der 5TM und der Windrichtungssensor auf Funktionalität geprüft werden. Erkennt der Arduino einen der Sensoren als nicht vorhanden oder fehlerhaft, so wird er bis zum nächsten Neustart nicht verwendet, die Error-LED zeigt ihn als fehlerhaft an und die Startup-SMS beinhaltet einen entsprechenden Hinweis.

Die Startup-SMS dient zur Benachrichtigung des Stationsinhabers. Sie sendet beim Start der Wetterstation die Uhrzeit und Informationen über die aktiven Sensoren. Die Sensoren werden hier nicht direkt genannt, da aus Gründen der Nachrichtenlänge und Einfachheitshalber die Messwerte benutzt werden. Diese sind:

Serverstatus(Server), Temperatur(Temp), Windrichtung(Windricht), Luftfeuchtigkeit(Feuchte), Luftdruck(Druck) und Bodenfeuchte(Bodenfeuchte). Die Bezeichnungen, welche in der SMS genannt werden sind dabei in Klammern. Am Ende des Setups wird die OK-LED aktiviert um die Ausführung der Routine anzuzeigen.

## 6.2. Die zyklische Routine

Die Routine beginnt mit dem Festsetzen der nächsten Aktivierung in 15 Minuten. Da die Übertragung der Wetterdaten zum Server über eine http-Anfrage läuft, wird ein Datenpaket in Form eines Strings vorbereitet, an welche die Daten und die entsprechenden Kommandos gehängt werden. Zuerst werden die Wetterstations-ID und das Passwort eingefügt und die Zeit angehängt. Falls die entsprechenden Sensoren vorhanden sind werden nach und nach Daten für Luftfeuchtigkeit, Luftdruck, Temperatur, Bodenfeuchte, Bodentemperatur, Windrichtung, Windgeschwindigkeit und Regenmenge in angloamerikanischen Einheiten an diesen String angehängt. Wie genau die einzelnen Messungen vorgenommen werden und die Konvertierung der Einheiten erfolgt, wird im Kapitel Komponenten abgehandelt. Wenn das Sammeln der Daten abgeschlossen ist, wird die http-Anfrage gestartet. Nach dem Hochladen der Wetterdaten geht der Arduino in eine Warteschleife über, in welcher er kontinuierliche eventuelle Error Codes per LED anzeigt und für das Empfangen von SMS offen ist.

Per SMS können die Daten der letzten Messung abgerufen und einzelne Ventile geöffnet oder geschlossen werden. Der Arduino nimmt hier aber nur Befehle von der Nummer des Stationsinhabers entgegen.

Sollte ein Ventil nach 16 Zyklen noch immer offen sein, so wird es automatisch vom Arduino geschlossen.

## 7. Zusammenfassung

Die Bibliotheken des 5TM Sensors und die GSM-Library mussten angepasst werden, da beide versuchten, alle Interrupts für sich zu sichern. Dies ist jedoch einfach zu beheben, da beide Bibliotheken eigentlich unterschiedliche Interrupt-Pins nutzen.

Ein weiterer Teil der GSM-Library wurde weiterhin geändert um die A&T Kommandos, sehr tiefgreifende Modem-Kommandos, benutzen zu können. Da sich das Pyranometer als defekt herausstellte, wird es in der Endversion nicht benutzt.

Die SHT2x-Bibliothek wurde auch leicht geändert um eine Endlosschleife zu verhindern.

Das Projekt ist trotz des defekten Pyranometers ein Erfolg. Alle Punkte bis auf das Pyranometer wurden erfüllt und die Wetterstation ist komplett einsatzbereit.

## 8.Ausblick

Da die Software für den Arduino eher einfach ist, noch viele Pins am Arduino frei sind und der Code für den Arduino bisher nur ca. 25% des Speichers benutzt, ist einem Ausbau in der Zukunft der Weg geebnet. Auch die Steuerung des Arduinos per SMS ist sehr einfach zu erweitern, was hoffentlich dazu beiträgt, andere zu motivieren, sich eigene Stationen nach diesem Vorbild zu bauen.

## 9.Danksagung

An dieser Stelle möchten wir Prof. Dr. Götz Lipphardt danken, der dieses Projekt sehr gut geleitet hat.

Auch danken wir Frau Krämer für ihre Betreuung des Projekts und natürlich Herrn Hector dafür, dass er das Hector-Seminar und damit Projekte wie dieses möglich macht.

## 10.Anhang

Tbl.1: Pinbelegung:

Operationsverstärker	Analog Pin 0
Windrichtungssensor	Analog Pin 1
Windgeschwindigkeitssensor	Analog Pin 2
Luftfeuchte- und Luftdrucksensor	Digital Pin 21 SCL
Luftfeuchte- und Luftdrucksensor	Digital Pin 20 SDA
5TM	Digital Pin 19
Regenmesssensor	Digital Pin 18
Pyranometer	Operationsverstärker

Tbl.2: Kabelbelegung:

–	Gelb
Windrichtungssensor	Grün
Windgeschwindigkeit	Blau
Luftfeuchte/Luftdruck	Pink
Luftfeuchte/Luftdruck	Lila
5TM	Braun
Regenmesssensor	Weiß
Pyranometer	Grau
5V	Rot
GND	Schwarz

Tbl.3: Lötleistenbelegung:

Operationsverstärker	x1	Analog Pin 0
Windrichtungssensor	x2	Analog Pin 1
Windgeschwindigkeitssensor	x3	Analog Pin 2
Luftfeuchte- und Luftdrucksensor	x4	Digital Pin 21 SCL
Luftfeuchte- und Luftdrucksensor	x5	Digital Pin 20 SDA
5TM	x6	Digital Pin 19
Regenmesssensor	x7	Digital Pin 18
Pyranometer	x8	Operationsverstärker
5V	x9	5V
GND	x10	GND

## 11.Quellenverzeichnis

[http://manuals.decagon.com/Manuals/13441\\_5TM\\_Web.pdf](http://manuals.decagon.com/Manuals/13441_5TM_Web.pdf), Zugriff: 16.06.17

[https://anel-elektronik.de/SITE/produkte/sensor\\_1/Datasheet%20SHT21.pdf](https://anel-elektronik.de/SITE/produkte/sensor_1/Datasheet%20SHT21.pdf)

Zugriff: 16.06.17

[https://ae-bst.resource.bosch.com/media/\\_tech/media/datasheets/BST-BMP180-DS000-121.pdf](https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BMP180-DS000-121.pdf) Zugriff: 16.06.17

[https://de.wikipedia.org/wiki/Piezoelektrischer\\_Sensor](https://de.wikipedia.org/wiki/Piezoelektrischer_Sensor) Zugriff: 16.06.17

[https://de.wikipedia.org/wiki/Kapazitiver\\_Sensor](https://de.wikipedia.org/wiki/Kapazitiver_Sensor) Zugriff: 16.06.17

<https://de.wikipedia.org/wiki/Pyranometer> Zugriff: 16.06.17

<https://de.wikipedia.org/wiki/Thermos%20A4ule> Zugriff: 16.06.17