



Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

Fachbereich Informatik
Department of Computer Science

Seminararbeit

im Bachelor-Studiengang Computer Science

Microcontroller

(Wetterstation)

Von

Mohamad Inalo

Betreuer: Prof. Dr.- Ing. Thomas Breuer

Inhaltsverzeichnis

Abbildungsverzeichnis	ii
Listings	ii
1. Einleitung	1
2. Grundlagen	2
2.1. Hardware	2
2.2. Software Entwicklungsumgebung	3
2.3. Sensoren und Module	4
2.3.1. DHT22	4
2.3.2. MQ135	6
2.3.3. RTC DS1302 (Real Time Clock)	7
2.3.4. Anemometer WH-SP-WS01	8
2.3.5. LCD	9
3. Schaltplan	10
4. Systemtest	11
4.1. DHT22 Test.	11
4.2. RTC DS1302	12
4.3. MQ135 Test	13
4.4. WH-SP-WS01 Test	14
4.5. LCD Test	15
5. Fazit & Ausblick	16
6. Literaturverzeichnis	16

Abbildungsverzeichnis

1. Wetterstation_Mindmap	1
2. Arduino Uno R3 Platine und GPIOs	2
3. Arduino IDE	3
4. Vergleich DHT11 mit DHT 22	5
5. MQ135 Platine und Pins	6
6. RTC DS1302 Platine und Pins	7
7. Anemometer WH-SP-WS01 Schaltplan	8
8. Anemometer WH-SP-WS01 Messungszeiten und Umwandlung in km/h	8
9. LCD und I2C Schnittstelle	9
10. Willkommensnachricht	9
11. Projekt Schaltplan mit Fritzing	10
12. DHT22 Test Umgebung	11
13. DHT22 Änderung Luftfeuchtigkeit Messwert	11
14. DHT22 Temperatur Messwert nach dem Test	11
15. RTC Test	12
16. MQ135 Test	13
17. WH-SP-WS01 Test	14
18. LCD Test	15

Listings

1. Abfrage der Messwerte von DHT22	4
2. Einstellung vom RTC 1302 Modul	7
3. Konfiguration eines LCDs	9

Einleitung

Wetter" ist die Abkürzung für Meteorologie, was wörtlich übersetzt die Messung des Zustands der Atmosphäre bedeutet. Und das ist ziemlich breit gefächert: Luftqualität in Innenräumen, Außenklima, Wettervorhersage, Umweltverschmutzung - all diese Variablen fallen in den Bereich der Meteorologie. Die Wetterstation ist ein Gerät, das diese Elemente mit Sensoren messen kann und Ihnen alle Daten auf einem Bildschirm anzeigt.

In dieser Seminararbeit wird eine Wetterstation vorgestellt, die mithilfe eines Microcontrollers realisiert wurde. Dabei wird auf die Funktionsweise und die Erwartungen von der Station eingegangen, und die benutzten Sensoren mit deren Messwerte erläutert.

Zeile des Projekts sind vor allem Echtzeitdaten und Aktuelle Messwerte für das Wetter zu liefern und diese Werte in bestimmter Art für den Benutzer darzustellen, dabei werden Werte für Temperatur, Luftfeuchtigkeit, Luftqualität und Windgeschwindigkeit erwartet. Es wird außerdem erwartet, dass das System sich flexible in mehreren Umgebungen unter verschiedene Voraussetzungen und Zuständen problemlos verhält und so genaue wie mögliche Daten liefert.

Die folgende Abbildung zeigt ein MindMap für das gesamte Projekt. Wir werden im ersten Abschnitt die verwendeten Hardwarekomponenten vorstellen, in dem wir das Herzstück unseres Projekts der Arduino Microcontroller vorstellen, danach gehen wir schrittweise den Sensor nach dem Anderen. Bei jedem Modul gehen wir kurz auf der Software ein und erläutern, wie man den Sensor startklar bekommt.

Wie man aus dem MindMap entnehmen kann, kann man das ganze Projekt mit einer externen Batterie (7-12) Volt betreiben oder auch mit einer USB-Anschluss. In diesem Projekt reicht uns eine traditionelle Energiequelle die wir später weiterentwickeln in dem wir auf erneubare Energie umsteigen.

Die komplette Software (auch Sketch genannt) finden Sie außerdem auf folgendem GitHub Repository: <https://github.com/Mo-Inalo/Wetterstation.git>

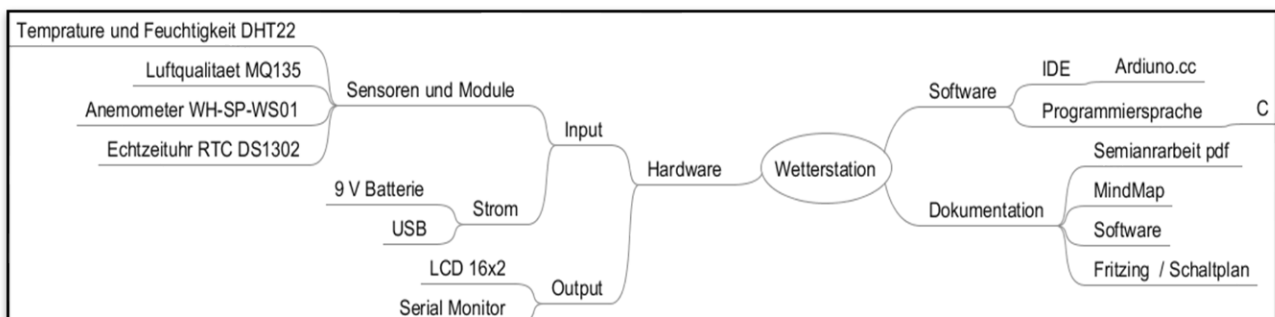


Abbildung 1 Wetterstation_Mindmap

Grundlagen

Hardware

Als Basis des Projekts wird ein Microcontroller der Marke Arduino, Modell UNO R3 verwendet, der sich in verschiedenen Programmiersprachen, unter anderem C, C++ und Python, programmieren lässt. Der Arduino Uno verfügt über 14 Digitale und 6 Analoge GPIOs (General Purpose Input/Output) und kann mit einer Spannung von 6 - 20 Volt in Betrieb genommen werden wobei der Hersteller eine Spannung von 7 - 12 Volt empfiehlt. Zudem wurden in diesem Projekt ein DHT22 Sensor zur Messung von Temperatur und Feuchtigkeit und ein MQ135 Sensor zur Messung von Luftqualität in Betrieb genommen. Außerdem wurde ein Anemometer verwendet um die Windgeschwindigkeit zu messen.

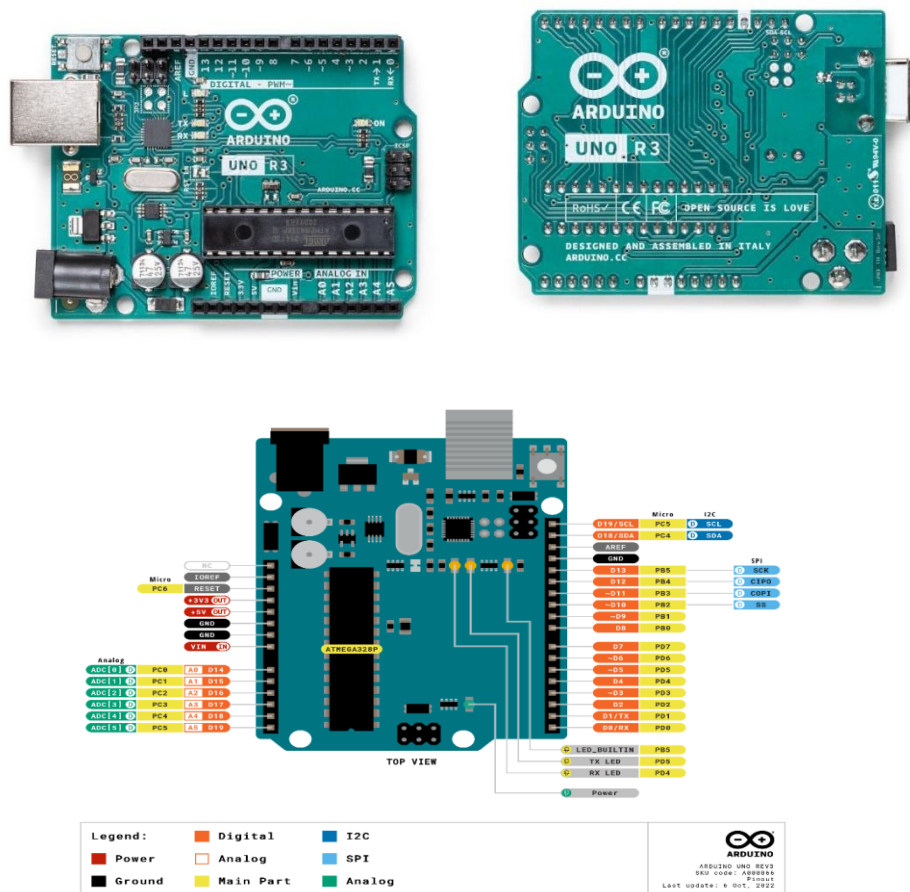


Abbildung 2 Arduino Uno R3 Platine und GPIOs

Software und Entwicklungsumgebung

Neben dem Hardware-Teil, dem Arduino-Board, gehört zum Arduino-Projekt die kostenlose Arduino-Entwicklungsumgebung. Diese Entwicklungsumgebung, in der Praxis auch als IDE bezeichnet, erlaubt das Erstellen, Testen und Hochladen von Arduino-Sketchen auf das Arduino-Board. Die Arduino-Entwicklungsumgebung ist ein Java-Programm. Die Software ist für die drei gängigen Betriebssysteme Windows, Mac OS X und Linux verfügbar. Sie wird laufend erweitert und ist momentan in der Version 2.0.3 verfügbar.

Für die Programmierung eines Arduino wird standardmäßig die Sprache C verwendet. Man kann aber den Microcontroller auch mit C++ oder mit Python programmieren. Allerdings braucht man weitere Installationen und Schritte, um mit Python programmieren zu können.

Als Compiler benutzen wir den mitgelieferten Compiler der Arduino IDE, den man kostenlos von der Arduino Webseite unter <https://www.arduino.cc/en/software> runterladen kann.

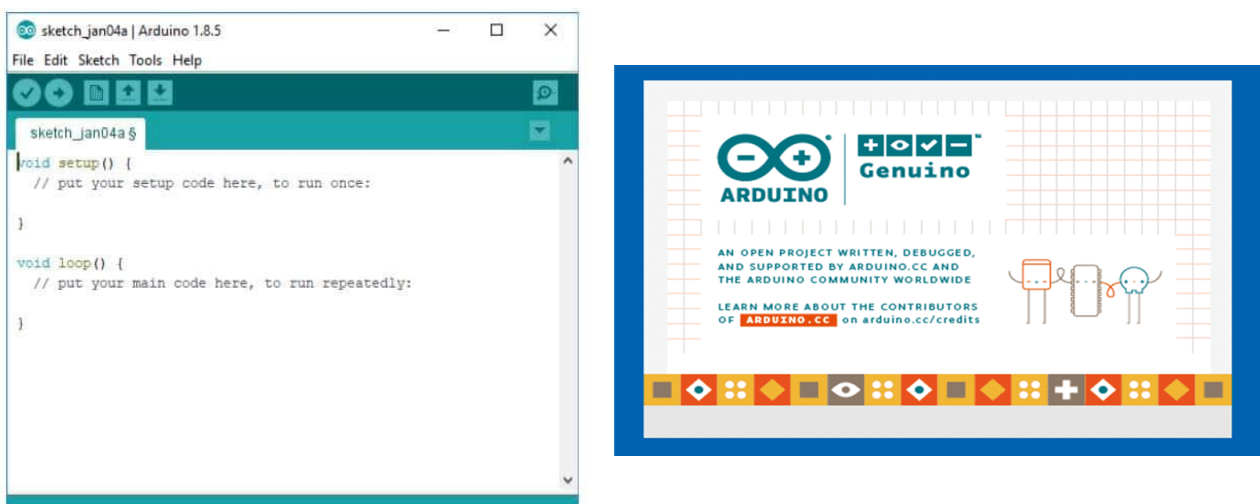


Abbildung 3 Arduino IDE

Die Software verfügt über einen integrierten Editor mit Compiler und mehrere Tools zum Einbinden von externen Bibliotheken. Man kann außerdem die Daten auf dem integrierten seriellen Monitor darstellen oder auch als Graph auf dem Plotter, dafür muss man im Code zuerst eine Verbindung zum seriellen Monitor mit der Anweisung `Serial.begin(9600)` herstellen und mit `Serial.println()` etwas ausgeben.

Für Anfänger bietet das Programm schon eine reiche Auswahl an Codebeispielen, um den Umgang mit dem Microcontroller besser üben zu können. Es können außerdem Microcontroller anderer Marken mit dem IDE konfiguriert und programmiert werden.

Sensoren und Module

DHT22

Der DHT22 Sensor ist ein Temperatur und Feuchtigkeitssensor der ein Temperatur-Messbereich [-40 °C bis + 80 °C] und einer Genauigkeit von ± 0.5 °C hat und Luftfeuchtigkeit von [0 % bis 100 %] und einer Genauigkeit von 2 – 5 % messen kann. Um eine bessere Genauigkeit zu erzielen und statische Fehler zu vermeiden kann man zwei (oder mehrere) Sensoren der Art DHT22 an der Wetterstation anschliessen und ihren durchschnittlichen Wert abfragen.

```
#include "DHT.h"

#define DHT1PIN 12

#define DHT2PIN 13

#define DHTTYPE DHT22

DHT dht1(DHT1PIN, DHTTYPE); //Der Sensor wird ab jetzt mit „dht1“ angesprochen

DHT dht2(DHT2PIN, DHTTYPE); //Der Sensor wird ab jetzt mit „dht2“ angesprochen

dht1.begin(); //DHT22 Sensor starten

dht2.begin();

float hum1 = dht1.readHumidity();

float temp1 = dht1.readTemperature();

float hum2 = dht2.readHumidity();

float temp2 = dht2.readTemperature();

if (isnan(hum1) || isnan(temp1) || isnan(hum2) || isnan(temp2)) {

    Serial.println(F("Lesevorgang DHT Sensor fehlgeschlagen!"));

}
```

Listing 1 Abfrage der Messwerte von DHT22

Die Sensoren bestehen aus einem Feuchtesensor, einem NTC-Temperatursensor (oder Thermistor) und einem Chip auf der Rückseite des Sensors. Zur Messung der Feuchte existieren im Feuchtigkeitssensor zwei Elektroden mit einer feuchtigkeitshaltenden Substrat zwischen beiden. Wenn sich also die Luftfeuchtigkeit ändert, ändert sich die Leitfähigkeit des Substrats oder der Widerstand zwischen diesen Elektroden. Diese Widerstandsänderung wird vom Chip gemessen und verarbeitet, so dass sie von einem Mikrocontroller gelesen werden kann. Zum anderen verwenden diese Sensoren zur Temperaturmessung einen NTC-Temperaturfühler oder Thermistor.

Ein Thermistor ist eigentlich ein veränderlicher Widerstand, der seinen Widerstand bei Änderung der Temperatur ändert. Diese Sensoren werden durch Sintern von Halbleitermaterialien wie Keramik oder Polymeren hergestellt, um größere Widerstandsänderungen bei geringen Temperaturänderungen zu erreichen. Der Begriff "NTC" bedeutet "Negativer Temperaturkoeffizient", was bedeutet, dass der Widerstand mit zunehmender Temperatur abnimmt. Alternative dazu könnte man den kleineren Sensor DHT11 nehmen, der aber einen deutlich geringeren Messbereich besitzt (Abb.4).

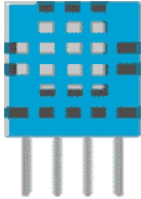
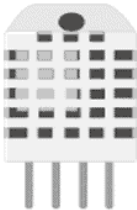
	 DHT11	 DHT22
Operating Voltage	3 to 5V	3 to 5V
Max Operating Current	2.5mA max	2.5mA max
Temperature Range	0-50°C / ± 2°C	-40 to 80°C / ± 0.5°C
Humidity Range	20-80% / 5%	0-100% / 2-5%
Sampling Rate	1 Hz (reading every second)	0.5 Hz (reading every 2 seconds)
Advantage	low cost	More Accurate

Abbildung 4 Vergleich DHT11 mit DHT 22

MQ 135

Der MQ-135 Sensor ist ein Modul, welches die Konzentration verschiedener Schadstoffe wie Benzin, Alkohol, Rauch sowie Verunreinigungen in der Luft erfasst. Der Sensor gibt die erfassten Schadstoffwerte als analogen Wert aus. Diese können im Anschluss im seriellen Monitor oder auf dem I2C Display ausgegeben oder anderweitig verarbeitet werden. Der Sensor verfügt über vier Pins:

- VCC – Pin für die Stromversorgung, anzuschließen an den 5V Pin des Arduinos
- GND – Ground-Pin, anzuschließen an den GND Pin des Arduinos
- A0 – Pin für den analogen Output, anzuschließen an den Pin A0 des Arduinos
- D0 – Pin für den digitalen Output, anzuschließen an den Pin D0 des Arduinos

Zudem können wir den Schwellenwert, welcher über den Pin D0 als „1“ bzw. „0“ ausgegeben wird, mechanisch einstellen. Die Kalibrierung des Schwellenwertes erfolgt durch Drehen des Potentiometers, welches auf der Rückseite des Sensormoduls zu finden ist.

Der Sensor hat eine hohe Empfindlichkeit und schnelle Reaktionszeit, benötigt allerdings einige Minuten bis er genaue Messwerte ausgibt, da der Sensor sich erst aufheizen muss.

Und hier weitere Details zum Sensor:

- Auflösung: 10~1000ppm
- Sensitive Widerstand: $2K\Omega$ - $20K\Omega$ in 100ppm CO
- Genauigkeit: $\geq 3\%$
- Reaktionszeit: $\leq 1s$
- Ansprechzeit nach Einschalten: $\leq 30s$
- Heizstrom: $\leq 180mA$
- Heizspannung: $5.0V \pm 0.2V$ / $1.5 \pm 0.1V$
- Eingangsspannung: 5V
- Heizenergieleistung ca.: 350mW
- Abmessungen: ca. 30mm (Länge) * 20mm (Breite)

Dieser Sensor muss die ersten 24 in einer Referenzumgebung kalibriert werden.

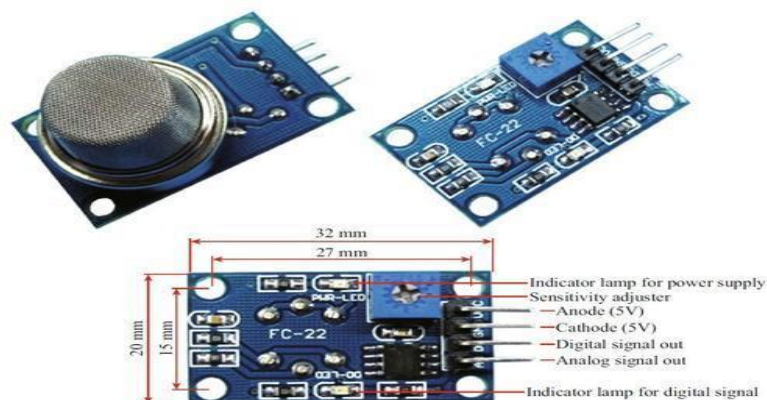


Abbildung 5 MQ135 Platine und Pins

RTC (Real Time Clock)

Eine RTC ist eine Echtzeituhr, die mit einer Batterie betrieben wird und auch bei Stromausfall die Zeit hält. Mit einer Echtzeituhr könnte man lange Zeitreihen verfolgen, auch wenn man den Mikrocontroller neu programmieren oder ihn von USB oder einem Netzstecker trennt.

Die meisten Mikrocontroller, einschließlich des Arduino, verfügen über einen integrierten Zeitnehmer namens `millis()`. Außerdem sind im Chip Zeitgeber eingebaut, die längeren Zeiträume wie Minuten oder Tage nachverfolgen können.



Abbildung 6 RTC DS1302 Platine und Pins

Mit dem folgenden Sketch kann man die Uhrzeit einstellen:

```
#include <virtuabotixRTC.h>

virtuabotixRTC myRTC(9, 10, 11);

void setup() {
    myRTC.setDS1302Time(30, 49, 05, 5, 24, 12, 2022);
}
```

Listing 2 Einstellung vom RTC DS1320 Modul

Anemometer WH-SP-WS01

Ein Anemometer ist ein Gerät welches zur Messung von Windgeschwindigkeit verwendet wird. Es besteht aus 3 oder Mehrere Zweige die den Wind empfangen und sich dementsprechend drehen. In dem Modul drin sind ein Magnet und ein Glasröhrchen verbaut. Das Glasröhrchen wird auch „Reedschalter“ oder „Reedkontakte“ genannt. Der Magnet fährt pro Umdrehung einmal um den Schalter und bei schnellem Wind entsprechend schneller

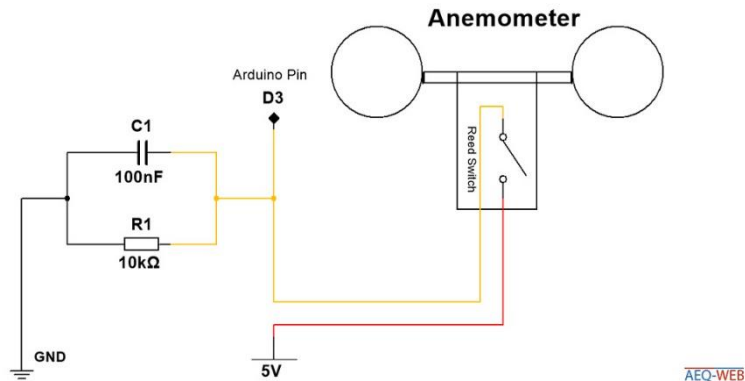


Abbildung 7 Anemometer WH-SP-WS01 Schaltplan

Da der Anemometer ein mechanisches Bauteil besitzt, kann es sein, dass es sich über die Jahre Metallstaub sammelt und somit im schlimmsten Fall bei Abnutzung jedes Mal, wenn der Schalter zu ist, dass es mal kontakt hat und mal nicht bis er vom Magnet angezogen wird. Das Ganze löst man mit einem 100nF Kondensator den man parallel zu einer 10 KΩ widerstand verbindet. Dies ist einfach eine Sicherheitsfunktion, die vermeidet, dass der Anemometer falsche Daten liefert. Die Aufgabe vom Arduino ist hierbei Impulse pro Sekunde zu liefern. Der Hersteller gibt vor, dass ein Impulse 2,4 km/h Windgeschwindigkeit entspricht und somit erhält man mit der Formel und die Anzahl der Impulse pro Sekunde: $6:3 = 2 \cdot 2,4 = 4,8$ km/h die gewünschte Geschwindigkeit in km/h oder in m/s.

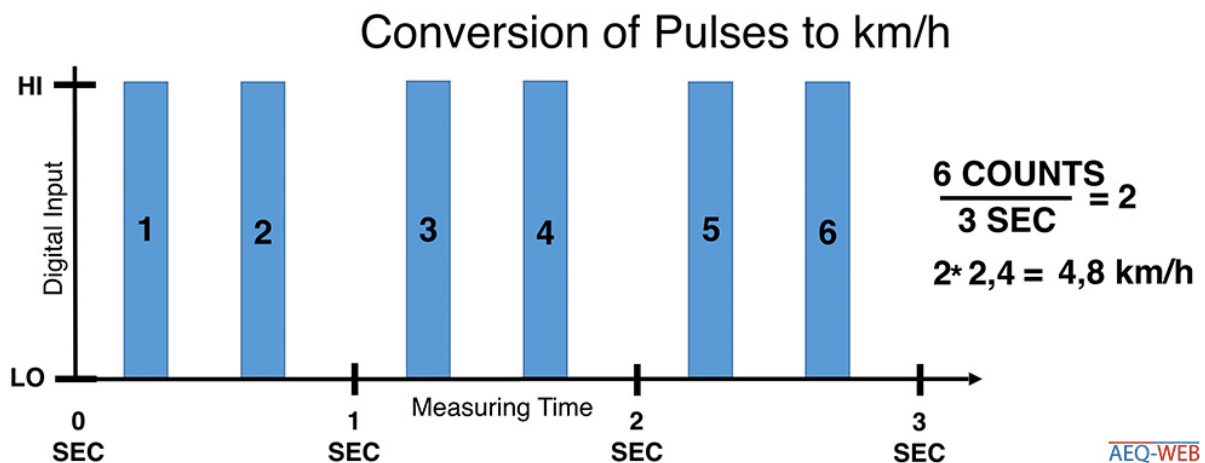


Abbildung 8 Anemometer WH-SP-WS01 Messungszeiten und Umwandlung in km/h

LCD

Zur einfachen Darstellung wurde neben der seriellen Monitore ein LCD in Anspruch genommen, das mithilfe einer I2C Schnittstelle über 4 Pins mit dem Arduino verbunden ist. Die Helligkeit lässt sich mit dem blauen Potentiometer auf der Rückseite des I2C Schnittstelle einstellen. Die Funktion beruht darauf, dass Flüssigkristalle die Polarisationsrichtung von Licht beeinflussen, wenn ein bestimmtes Maß an elektrischer Spannung angelegt wird; sie bestehen aus Segmenten, die unabhängig voneinander ihre Transparenz ändern können. Dazu wird mit elektrischer Spannung in jedem Segment die Ausrichtung der Flüssigkristalle gesteuert. Damit ändert sich die Durchlässigkeit für polarisiertes Licht.



Abbildung 9 LCD und I2C Schnittstelle

```
#include "LiquidCrystal_I2C.h"
LiquidCrystal_I2C lcd(0x27,16,2);
lcd.init();
lcd.backlight();
lcd.setCursor(1,0);
lcd.print("Programmstart");
lcd.setCursor(1,1);
lcd.print("Wetterstation");
```

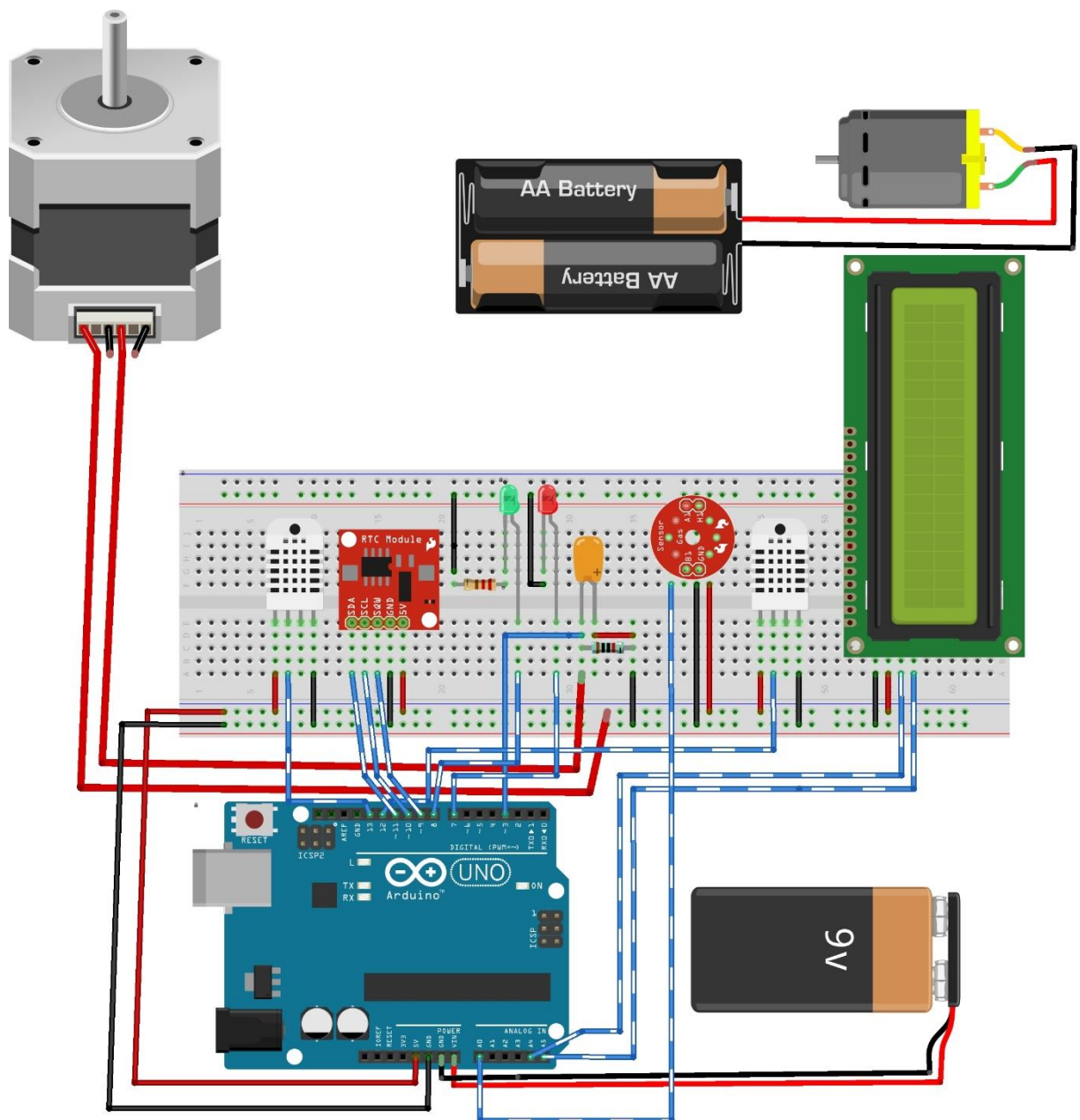
Listing 3 Konfiguration eines LCDs



Abbildung 10 Willkommensnachricht

Der Schaltplan

Mithilfe des Programms Fritzing zur Zeichnung von Schaltplänen, das über die Webseite <https://fritzing.org/download/> zu erwerben ist wurde der Schaltplan hier erzeugt.



fritzing

Abbildung 11 Projekt Schaltplan mit Fritzing

Systemtest

DHT22 Test

Der Sensor hat sich in mehreren Umgebungen als Flexible gestellt. Wie auf der Abbildung zu sehen ist, liegt der Sensor neben einer Flamme unter einem Kunststoff Pecher und man sieht deutlich auf dem Seriellen monitor wie die Temperatur sich erhöht.



Abbildung 12 DHT22 Test Umgebung

Luftfeuchtigkeit1: 62.10%	Temperatur1: 21.10 C	Luftfeuchtigkeit2: 53.30%	Temperatur2: 20.80 C	Luftqualitaet : 50	Wind Speed: 0.00 km/h - 0.00 m/s
Luftfeuchtigkeit1: 68.50%	Temperatur1: 21.20 C	Luftfeuchtigkeit2: 53.20%	Temperatur2: 20.80 C	Luftqualitaet : 49	Wind Speed: 0.00 km/h - 0.00 m/s
Luftfeuchtigkeit1: 75.10%	Temperatur1: 21.30 C	Luftfeuchtigkeit2: 53.20%	Temperatur2: 20.80 C	Luftqualitaet : 50	Wind Speed: 0.00 km/h - 0.00 m/s
Luftfeuchtigkeit1: 78.30%	Temperatur1: 21.30 C	Luftfeuchtigkeit2: 53.20%	Temperatur2: 20.80 C	Luftqualitaet : 49	Wind Speed: 0.00 km/h - 0.00 m/s
Luftfeuchtigkeit1: 76.90%	Temperatur1: 21.30 C	Luftfeuchtigkeit2: 53.10%	Temperatur2: 20.80 C	Luftqualitaet : 50	Wind Speed: 0.00 km/h - 0.00 m/s
Luftfeuchtigkeit1: 64.90%	Temperatur1: 21.40 C	Luftfeuchtigkeit2: 53.00%	Temperatur2: 20.80 C	Luftqualitaet : 50	Wind Speed: 0.00 km/h - 0.00 m/s

Abbildung 13 DHT22 Änderung Luftfeuchtigkeit Messwert

Luftfeuchtigkeit1: 51.80%	Temperatur1: 24.00 C	Luftfeuchtigkeit2: 53.10%	Temperatur2: 20.90 C	Luftqualitaet : 49	Wind Speed: 0.00 km/h - 0.00 m/s
Luftfeuchtigkeit1: 51.70%	Temperatur1: 24.00 C	Luftfeuchtigkeit2: 53.20%	Temperatur2: 20.90 C	Luftqualitaet : 50	Wind Speed: 0.00 km/h - 0.00 m/s
Luftfeuchtigkeit1: 51.80%	Temperatur1: 23.90 C	Luftfeuchtigkeit2: 53.20%	Temperatur2: 20.90 C	Luftqualitaet : 50	Wind Speed: 0.00 km/h - 0.00 m/s
Luftfeuchtigkeit1: 51.80%	Temperatur1: 24.00 C	Luftfeuchtigkeit2: 53.20%	Temperatur2: 20.90 C	Luftqualitaet : 51	Wind Speed: 0.00 km/h - 0.00 m/s
Luftfeuchtigkeit1: 51.90%	Temperatur1: 24.00 C	Luftfeuchtigkeit2: 53.20%	Temperatur2: 20.90 C	Luftqualitaet : 52	Wind Speed: 0.00 km/h - 0.00 m/s

Abbildung 14 DHT22 Änderung Temperatur Messwert

RTC DS1203 Test

Durch einstellen des Moduls (Listing 2) und Nach Ausschalten das System Liefert die Uhrzeit weiterhin wie erwartet das korrekte Datum mit der Uhrzeit.

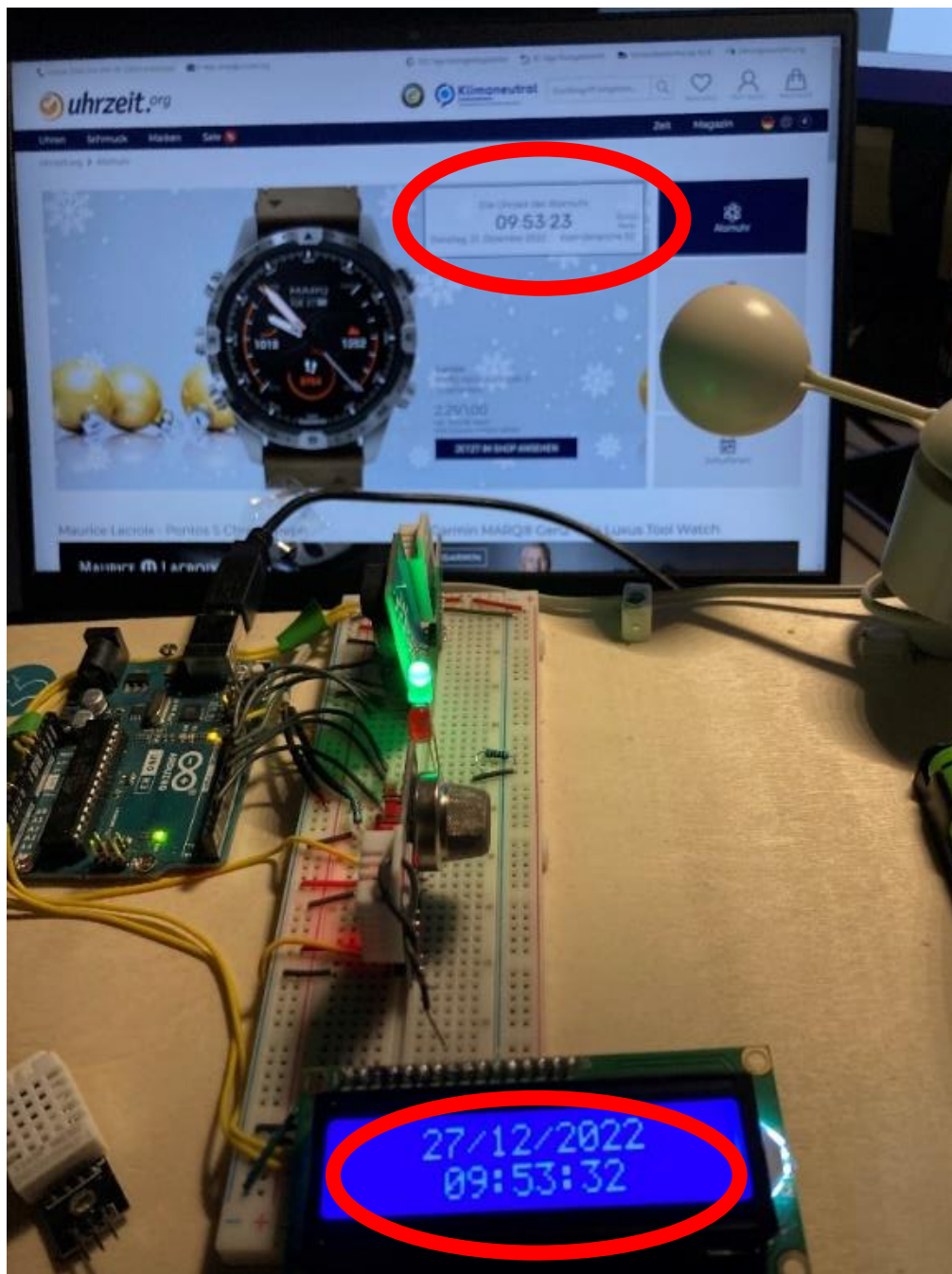
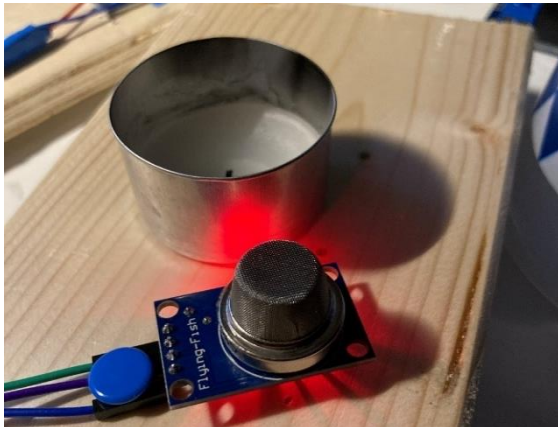


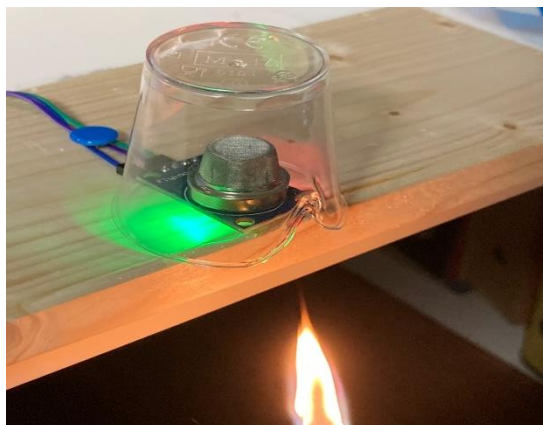
Abbildung 15 1RTC Test

MQ135 Test

Der Sensor hat sich in mehreren Umgebungen als Flexible gestellt. Wie auf der Abbildung zu sehen ist, bekommt der Sensor CO₂ das sich unter dem Kunststoff Pecher sammelt und man sieht deutlich auf dem Seriellen monitor wer Sensorwert sich erhöht.



```
temperatur2: 21.00 C Luftqualitaet : 60 Wind Speed: 0.00 km/h  
temperatur2: 21.00 C Luftqualitaet : 61 Wind Speed: 0.00 km/h  
temperatur2: 21.00 C Luftqualitaet : 60 Wind Speed: 0.00 km/h  
temperatur2: 21.00 C Luftqualitaet : 60 Wind Speed: 0.00 km/h
```



```
temperatur2: 21.10 C Luftqualitaet : 60 Wind Speed: 0.00 km/h - 0.00 m/s  
temperatur2: 21.10 C Luftqualitaet : 60 Wind Speed: 0.00 km/h - 0.00 m/s  
temperatur2: 21.10 C Luftqualitaet : 61 Wind Speed: 0.00 km/h - 0.00 m/s  
temperatur2: 21.10 C Luftqualitaet : 60 Wind Speed: 0.00 km/h - 0.00 m/s  
temperatur2: 21.10 C Luftqualitaet : 60 Wind Speed: 0.00 km/h - 0.00 m/s  
temperatur2: 21.10 C Luftqualitaet : 60 Wind Speed: 0.00 km/h - 0.00 m/s  
temperatur2: 21.10 C Luftqualitaet : 67 Wind Speed: 0.00 km/h - 0.00 m/s  
temperatur2: 21.10 C Luftqualitaet : 82 Wind Speed: 0.00 km/h - 0.00 m/s  
temperatur2: 21.10 C Luftqualitaet : 94 Wind Speed: 0.00 km/h - 0.00 m/s  
temperatur2: 21.10 C Luftqualitaet : 99 Wind Speed: 0.00 km/h - 0.00 m/s  
temperatur2: 21.10 C Luftqualitaet : 107 Wind Speed: 0.00 km/h - 0.00 m/s  
temperatur2: 21.10 C Luftqualitaet : 112 Wind Speed: 0.00 km/h - 0.00 m/s  
temperatur2: 21.10 C Luftqualitaet : 112 Wind Speed: 0.00 km/h - 0.00 m/s  
temperatur2: 21.10 C Luftqualitaet : 111 Wind Speed: 0.00 km/h - 0.00 m/s
```

```
Temperatur2: 21.30 C Luftqualitaet : 62 Wind Speed: 0.00 km/h - 0.00 m/s  
Temperatur2: 21.30 C Luftqualitaet : 62 Wind Speed: 0.00 km/h - 0.00 m/s  
Temperatur2: 21.30 C Luftqualitaet : 62 Wind Speed: 0.00 km/h - 0.00 m/s  
Temperatur2: 21.30 C Luftqualitaet : 62 Wind Speed: 0.00 km/h - 0.00 m/s  
Temperatur2: 21.30 C Luftqualitaet : 62 Wind Speed: 0.00 km/h - 0.00 m/s  
Temperatur2: 21.30 C Luftqualitaet : 62 Wind Speed: 0.00 km/h - 0.00 m/s
```

Abbildung 16 MQ135 Test

WH-SP-WS01 Test

Der Anemometer zeigt nach unseren Beobachtungen angemessene werte, die zwischen 12 -16 km/h liegen.

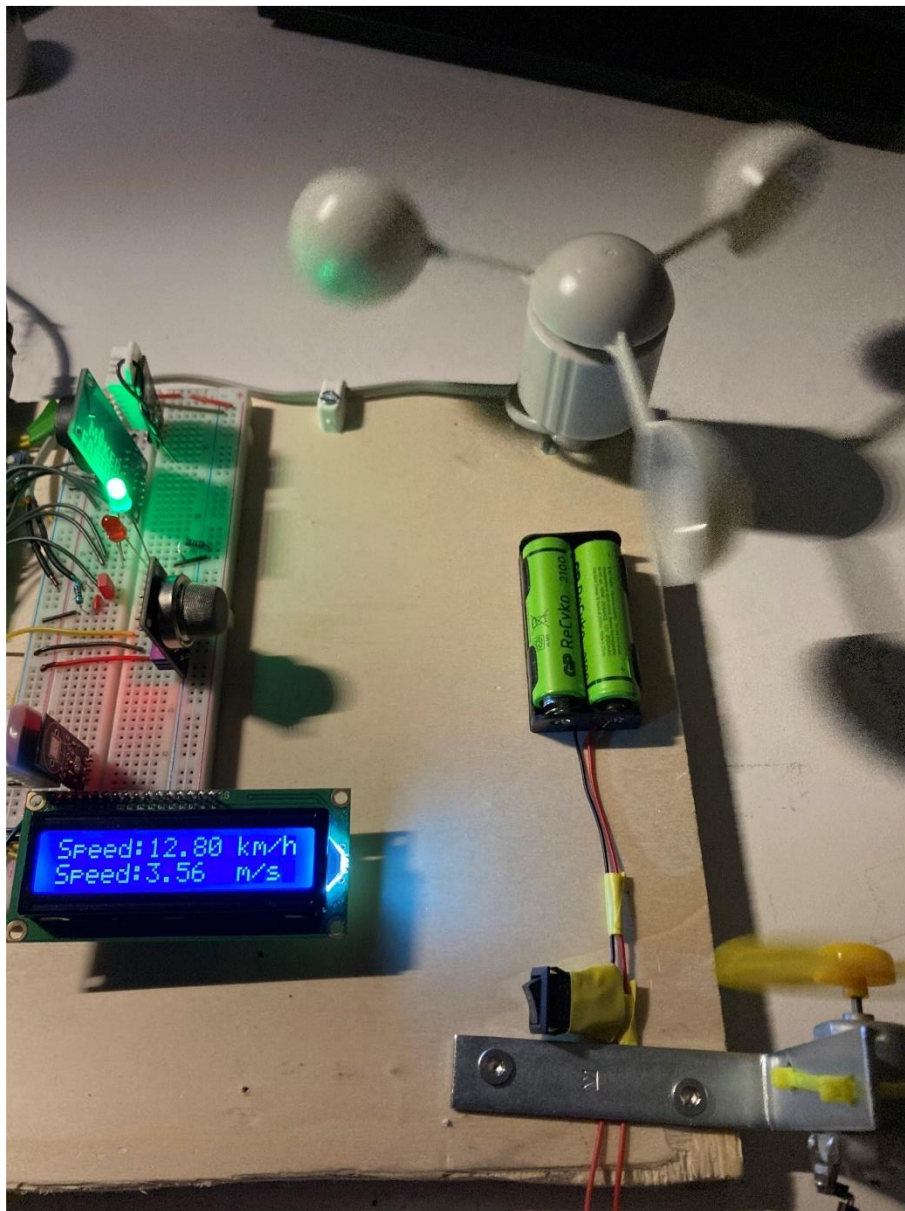


Abbildung 17 WH-SP-WS01 Test

LCD Test

Wie erwartet gibt das LCD alle gewünschte Informationen aus.

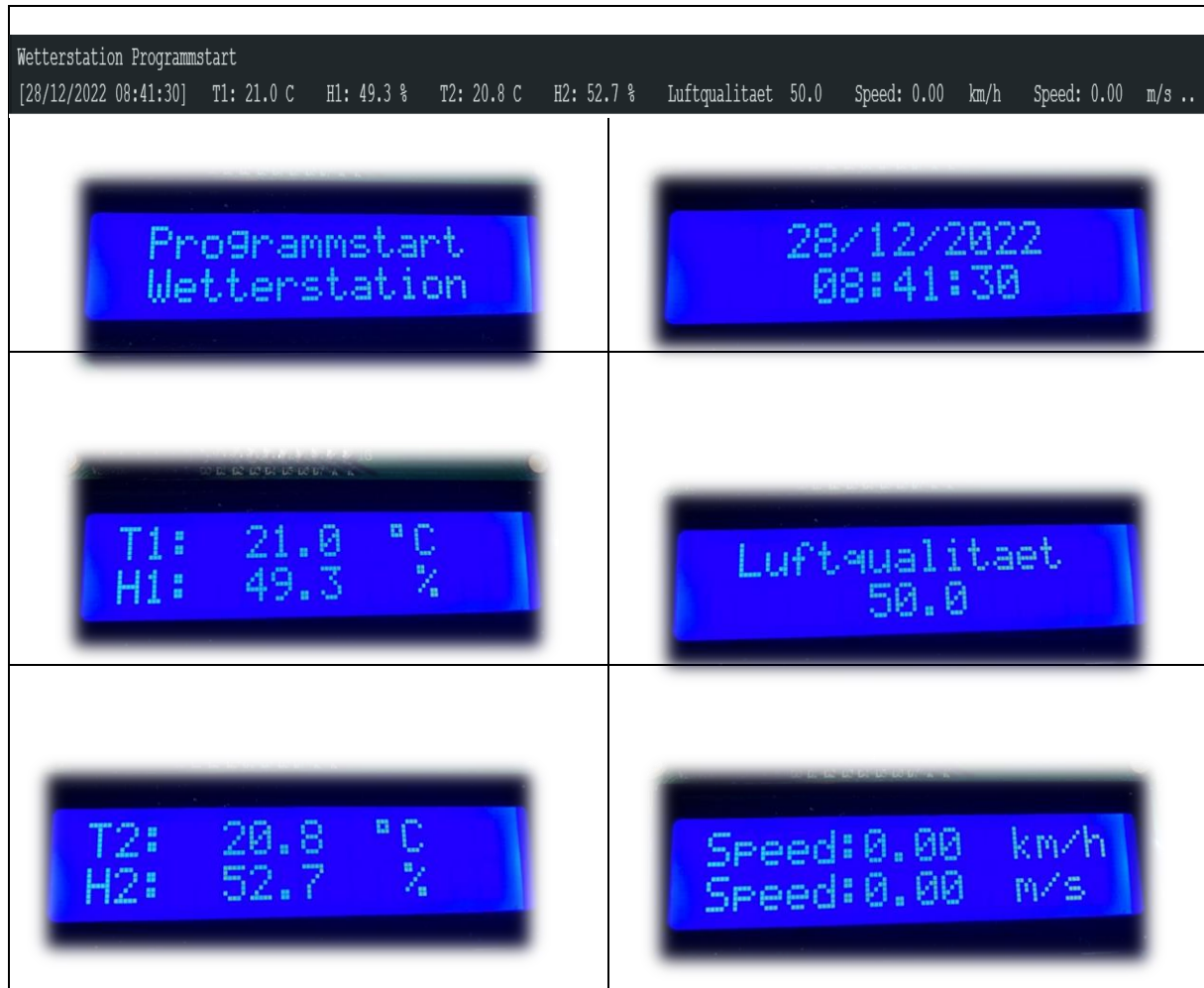


Abbildung 18 LCD Test

Fazit & Ausblick

Zusammenfassend, stellt die Variante die wir in diesem Projekt erstellt haben eine sehr gute Lösung um Echtzeitdaten für das Wetter zu erhalten. Man kann das Projekt mit einem Home-Automation-System verknüpfen in die mehrere Alarmer und Kameras eingebunden werden können. Zum Beispiel im Falle vom Feuer oder Einbruch, da man zurzeit für jeden Zweck ein Sensor erhalten kann. Das ganze System ist auch für das Outdoor machbar in dem auf Strom und W-Lan verzichtet werden kann, da man das ganze Netzwerk mit einem GSM Modul und die Stromversorgung mit erneuerbare Energy realisieren kann. Nicht zuletzt kann man mithilfe von Datenbanken die Daten sammeln und mit der Programmiersprache Python die gesammelten Daten und die erstellten Statistiken und Graphen darstellen und dies an verschiedene Umwelt Organisationen versenden bzw. schenken um bessere Auswertungen für das Klima zu erzielen damit man in der Zukunft gegen Probleme wie Klima Wandel besser vorzeitig planen und vorgehen kann.

Literaturverzeichnis

- <https://www.arduino.cc/>
- <https://www.netatmo.com/de-de/guides/weather/weather-factors/solutions/weather-station>
- <https://nerdyelectronics.com/working-of-dht-sensor-dht11-and-dht22>
- <https://microcontrollerslab.com/ds1302-rtc-chip-pinout-example-arduino-applications/>
- <https://www.aeq-web.com/arduino-anemometer-wind-sensor/>
- <https://fritzing.org/download/>
- <https://www.uhrzeit.org/atomuhr.php>