
BACHELORARBEIT

Herr
Sun, Xin

**Entwicklung und Umsetzung
eines Systems zur Messung
von Temperaturen**

Mittweida, 2022

Fakultät: Ingenieurwissenschaften

BACHELORARBEIT

Entwicklung und Umsetzung eines Systems zur Messung von Temperaturen

Autor:

Herr Sun, Xin

Studiengang:

Elektro- und Informationstechnik

Seminargruppe:

EI18sA-BC

Erstprüfer:

Prof. Dr.-Ing. Michael Kuhl

Zweitprüfer:

M.Sc. Kevin Blümel

Einreichung:

Mittweida, 21.03.2022

Verteidigung/Bewertung:

Mittweida, 2022

Faculty of Engineering

BACHELORTHESIS

Development and implementation of a system for measuring of temperatures

author:

Mr. Sun, Xin

course of studies:

Electro- and Informationtechnology

seminar group:

EI18sA-BC

first examiner:

Prof. Dr.-Ing. Michael Kuhl

second examiner:

M.Sc. Kevin Blümel

submission:

Mittweida, 21.03.2022

defence/ evaluation:

Mittweida, 2022

Bibliografische Beschreibung:

Sun,Xin:

Entwicklung und Umsetzung eines Systems zur Messung von Temperaturen. - 2022.

Development and implementation of a system for measuring of temperatures. - 2022.

Seiten 51, Mittweida, Hochschule Mittweida, Fakultät Ingenieurwissenschaften,
Bachelorarbeit, 2022

Referat:

Ziel der Arbeit ist der prototypische Aufbau eines Systems zur Messung bzw. Langzeitmessung von Temperaturen. Das System sollte automatisch Temperaturen aufgenommen über einen Mikrokontroller bewertet und an einen PC übergeben werden. Die Werte vom System gemessenen können auf dem LCD-Display angezeigt werden.

Inhalt

Inhalt.....	I
Abbildungsverzeichnis.....	III
Tabellenverzeichnis.....	V
Abkürzungsverzeichnis.....	VI
1 Einleitung.....	11
2 Aufgabenstellung.....	12
3 Design-Hintergrund.....	13
3.1 Thermometer.....	3
3.2 Temperatursensoren.....	3
4 Konzepte für den Systementwurf.....	4
4.1 Gesamtkonzept.....	4
4.2 Hardware.....	5
4.2.1 Arduino Mega 2560.....	5
4.2.2 Temperatursensor (BME280)	6
4.2.3 LCD-Display.....	7
4.2.4 Taste.....	8
4.2.5 Summer.....	9
4.2.6 Hardwarekonzept.....	9
4.3 Software.....	10
4.3.1 Softwarekonzept.....	11
4.3.2 Erläuterungen zu einigen Codes.....	12
5 Technischer Aufbau.....	15
6 Experimentelle Ablauf und Ergebnisse.....	19
6.1 Beschreibung des experimentellen Inhalts.....	19
6.2 Experimentelle Vorbereitung.....	19
6.3 Testen der Tasten.....	20
6.4 Summer testen.....	22
6.5 Test der Datenübertragung.....	23
6.6 Versuchsauswertung.....	26
7 Zusammenfassung.....	27

Literaturverzeichnis.....	28
Anlagen.....	30
Anlagen, Verpackung.....	31
Anlagen, Arduino Mega2560 Technische Daten.....	32
Anlagen, Schaltplan.....	33
Anlagen, PCB Board.....	34
Anlagen, PCB Board.....	35
Anlagen, Bauteil: footprint Zuordnungen.....	36
Anlagen, Gerberdatei.....	37
Anlagen, Vollständiger Software-Code.....	38
Selbstständigkeitserklärung.....	42

Abbildungsverzeichnis

Abbildung 1: Flussdiagramm des Gesamtkonzepts

Abbildung 2: Arduino Mega 2560

Abbildung 3: BME280

Abbildung 4: BME280 Technische Daten

Abbildung 5: LCD1602 Vorderseite

Abbildung 6: LCD1602 Rückseite

Abbildung 7: Taste

Abbildung 8: Summer

Abbildung 9: Blockdiagramm des Hardwarekonzepts

Abbildung 10: Flussdiagramm des Softwarekonzepts

Abbildung 11: Auswahl der erforderlichen Bibliotheken

Abbildung 12: Codes für Höchst- und Mindesttemperaturen

Abbildung 13: Codes für Durchschnittswerte

Abbildung 14: Codes für Temperaturschwellen und Summer

Abbildung 15: Schaltplan des Systems

Abbildung 16: 3D-Betrachter(Vorderseite)

Abbildung 17: 3D-Betrachter(Rückseite)

Abbildung 18: Startseite des LCD-Displays

Abbildung 19: Anzeige der Momentantemperatur

Abbildung 20: Anzeige von Maximal- und Minimaltemperatur

Abbildung 21: Anzeige von Durchschnittstemperatur

Abbildung 22: Summer testen

Abbildung 23: Code für PLX-DAQ 1

Abbildung 24: Code für PLX-DAQ 2

Abbildung 25: PLX-DAQ

Abbildung 26: Liniendiagramm der momentanen Temperatur

Abbildung 27: Liniendiagramm der verbleibenden Werte

Abbildung 28: Temperatur in verschiedenen Positionen des Raume

Tabellenverzeichnis

Tabelle 1: Erforderliche Komponenten

Abkürzungsverzeichnis

GND.....	Grund
I ² C.....	Inter-Integrated Circuit
IDE.....	Integrated Development Environment
LCD.....	Liquid Crystal Display
PC.....	Personal Computer
PCB.....	Printed Circuit Board
PLX-DAQ.....	Parallax-Data Acquisition
SCL.....	Serial Clock Line
SDA.....	Serial Data Line
USB.....	Universal Serial Bus
VCC.....	Voltage of circuit

1 Einleitung

Die Temperatur ist eine physikalische Größe, der die Menschen in ihrem Leben ständig ausgesetzt sind. Die Temperatur hat einen wichtigen Einfluss auf das menschliche Leben, Experimente, die industrielle Produktion und vieles mehr. Die Erfassung und Kontrolle der Temperatur wird im täglichen Leben sowie in der industriellen und landwirtschaftlichen Produktion häufig eingesetzt. In der Metallurgie, der Lebensmittelverarbeitung, der Chemie und anderen industriellen Produktionsprozessen wird eine breite Palette von Wärme- und Reaktionsöfen eingesetzt, die alle eine strenge Temperaturkontrolle erfordern. Im täglichen Leben benötigen auch Backofen, Mikrowellenherde, elektrische Warmwasserbereiter, Kühlschränke und andere Geräte eine Temperaturerfassung und -regelung.

Traditionelle Temperaturmesselemente sind RTDs und Thermoelemente. RTDs und Thermoelemente hingegen messen in der Regel Spannung, die dann in eine entsprechende Temperatur umgewandelt wird, was mehr externe Hardwareunterstützung und hohe Produktionskosten erfordert. Traditionell werden als Thermometer in der Regel Quecksilber-, Paraffin- oder Alkoholthermometer verwendet. Diese Thermometer haben in der Regel eng beieinander liegende Skalen, die sich nicht genau unterscheiden lassen und schwer abzulesen sind. Außerdem haben sie eine große Wärmekapazität, brauchen länger, um das thermische Gleichgewicht zu erreichen, und sind sehr unpraktisch in der Anwendung.

Im Vergleich zu herkömmlichen Thermometern haben elektronische Thermometer den Vorteil, dass sie leicht ablesbar sind, einen großen Temperaturmessbereich haben, die Temperatur schnell und genau messen. Die Temperatur wird in digitaler Form angezeigt und wird hauptsächlich dort eingesetzt, wo genaue Temperaturen erforderlich sind, oder in Forschungslabors. In dieser Arbeit geht es daher um die Entwicklung und Implementierung eines Systems zur Messung bzw. Langzeitmessung von Temperaturen. Das System erfordert eine automatische Temperaturmessung und Umrechnung in Celsius für die Anzeige auf dem LCD-Display. Das System kann die Messschwellen frei einstellen, und ein Alarm wird durch einen Summer ausgelöst, wenn die Temperatur die Schwelle überschreitet. Außerdem kann das System Höchst-, Mindest- und Durchschnittstemperaturen aufzeichnen und die auf dem LCD-Display angezeigten Daten mit zwei Tasten umschalten.

2 Aufgabenstellung

Ziel dieser Bachelorarbeit ist die Entwicklung eines Temperaturmesssystems auf der Basis eines integrierten Umgebungssensors und eines Mikrocontrollers. Hierzu sollen automatisch Temperaturen aufgenommen über einen Mikrocontroller bewertet und an einen PC übergeben werden. Die Werte sind als Reihen mit entsprechend sinnvollen Angaben abzuspeichern und auszugeben. Um dieses Ziel zu erreichen, müssen folgende Aufgaben erledigt werden:

- Einarbeitung in die Thematik
- Erarbeitung eines Systemkonzeptes (Hardware, Software, realer Aufbau) unter Nutzung eines Arduino- Mikrocontrollers
- Entwurf von Hardwareschaltungen mit Breadboard und KiCad
- Programmieren mit der Arduino IDE
- Durchführung und Auswertung von Basisversuchen
- Bewertung, Darstellung und Dokumentation der Projektergebnisse

3 Design-Hintergrund

3.1 Thermometer

Ein Thermometer ist ein Instrument zur Messung der Umgebungstemperatur. Mit dem technologischen Fortschritt schreiten auch die Thermometer voran. Thermometer haben sich vom ursprünglichen Glasröhrenthermometer zu den heutigen RTD-Thermometern, Thermoelementthermometern und elektronischen Thermometern entwickelt. Bei elektronischen Thermometern bestimmt die Empfindlichkeit des Sensors die Genauigkeit, den Messbereich, den Kontrollbereich und die Verwendung des Thermometers.

3.2 Temperatursensoren

Die Entwicklung von Temperatursensoren hat drei Stufen durchlaufen: traditionelle diskrete Temperatursensoren, analoge integrierte Temperatursensoren und intelligente Temperatursensoren.

Am weitesten verbreitet sind intelligente Temperatursensoren. Der intelligente Temperatursensor ist in der Lage, Temperaturdaten und zugehörige Temperaturregelgrößen auszugeben, ist an eine Vielzahl von Mikrocontrollern angepasst und kann zur Realisierung von Testfunktionen per Software verwendet werden. Intelligente Temperatursensoren werden heute in einer Vielzahl von Temperaturkontrollsystemen in der industriellen Produktion, elektronischen Thermometern, medizinischen Instrumenten und anderen Anwendungen eingesetzt.

4 Konzepte für den Systementwurf

4.1 Gesamtkonzept

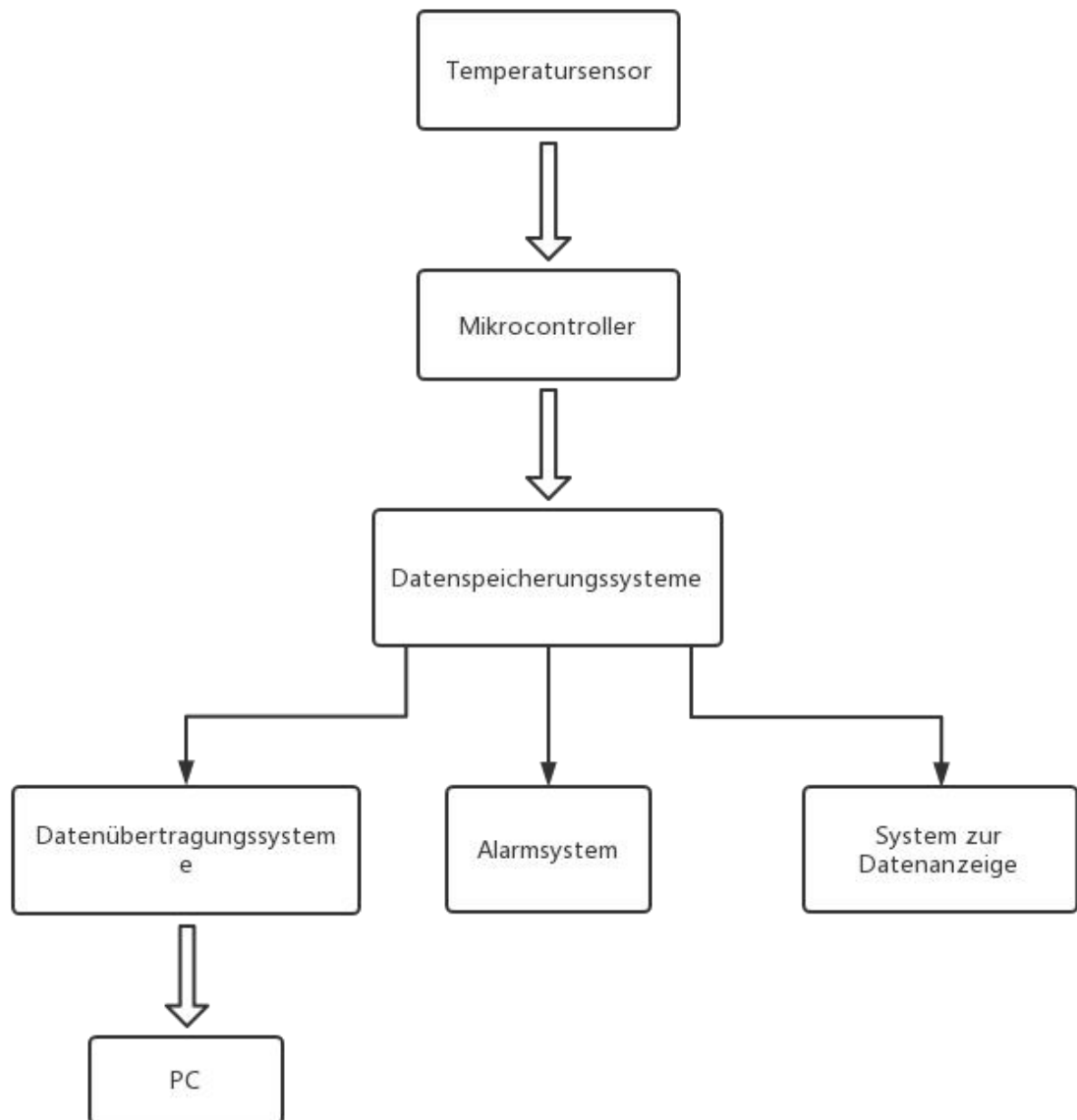


Abbildung 1: Flussdiagramm des Gesamtkonzepts

Die obige Abbildung zeigt ein Flussdiagramm des Gesamtkonzepts des Systems. Der Temperatursensor wird an den Mikrocontroller angeschlossen und beginnt mit der Datenerfassung. Die gesammelten Daten werden im Mikrocontroller gespeichert. Ein externes LCD-Display zeigt die gesammelten Daten an. Es wird ein Temperaturschwellenwert programmiert, der mit den erfassten Daten verglichen wird, und ein Alarmsystem wird aktiviert, wenn der eingestellte Schwellenwert überschritten

wird. Der Mikrocontroller kann auch an einen PC angeschlossen werden und über die serielle Schnittstelle Daten an den PC übertragen. Da es sich um ein komplettes Temperaturmesssystem handelt, sollten auch die Höchst-, Mindest- und Durchschnittstemperaturen so programmiert werden, dass sie berechnet und auf dem LCD-Bildschirm angezeigt werden.

4.2 Hardware

Nachfolgend finden Sie eine Beschreibung der gesamten in dieser Arbeit verwendeten Hardware.

4.2.1 Arduino Mega 2560

Der Mega 2560 ist ein Mikrocontroller-Board auf Basis des ATmega2560. Es verfügt über 54 digitale Eingangs-/Ausgangs-Pins (von denen 15 als PWM-Ausgänge verwendet werden können), 16 analoge Eingänge, 4 UARTs (serielle Hardware-Ports), einen 16-MHz-Quarzoszillator, einen USB-Anschluss, eine Stromversorgungsbuchse, einen ICSP-Header und eine Reset-Taste. Es enthält alles, was zur Unterstützung des Mikrocontrollers benötigt wird.[1] In dieser Arbeit wird die Verwendung des digitalen Pinout-Abschnitts untersucht.



Abbildung 2: Arduino Mega 2560

4.2.2 Temperatursensor (BME280)

Der BME280 ist ein integrierter Umweltsensor, der speziell für mobile Anwendungen entwickelt wurde, bei denen Größe und geringer Stromverbrauch wichtige Designvorgaben sind. Das Gerät kombiniert einzelne hochlineare und hochgenaue Sensoren für Druck, Feuchtigkeit und Temperatur in einem 8-Pin Metalldeckel 2,5x 2,5 x 0,93mm³ LGA-Gehäuse. Der BME280 unterstützt sowohl I2C als auch SPI (3-Draht/4-Draht) digitale serielle Schnittstellen.[2] Zusammenfassend lässt sich sagen, dass der BME280 hochpräzise, vielseitig und klein ist und einfach und schnell als Prototyp für verschiedene Anwendungen, wie z. B. Umweltüberwachung, entwickelt werden kann.

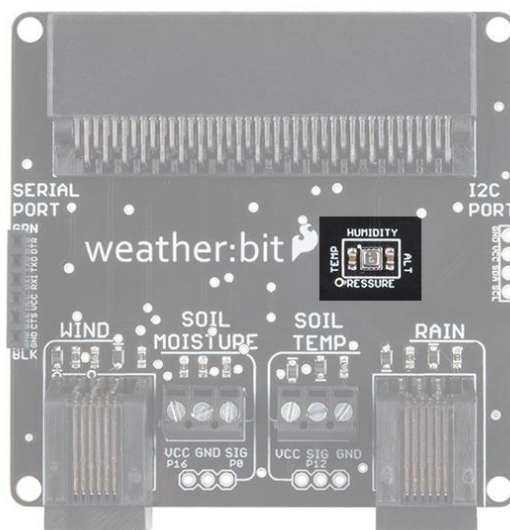


Abbildung 3: BME280[3]

BME280 (preliminary) Technical data	
Package dimensions	8-Pin LGA with metal 2.5 x 2.5 x 0.93 mm ³
Operation range (full accuracy)	Pressure: 300 ... 1100 hPa Temperature: -40 ... +85 °C
Supply voltage V _{DDIO}	1.2 ... 3.6 V
Supply voltage V _{DD}	1.71 ... 3.6 V
Interface	I ² C and SPI
Average current consumption (typ.) (1Hz data refresh rate)	1.8 µA @ 1 Hz (H, T) 2.8 µA @ 1 Hz (P, T) 3.6 µA @ 1 Hz (H, P, T) T = temperature
Average current consumption in sleep mode	0.1 µA
Humidity sensor	
Response time (τ _{63%})	1 s
Accuracy tolerance	±3 % relative humidity
Hysteresis	≤2 % relative humidity
Pressure sensor	
RMS Noise	0.2 Pa (equiv. to 1.7 cm)
Sensitivity Error	±0.25 % (equiv. to 1 m at 400 m height change)
Temperature coefficient offset	±1.5 Pa/K (equiv. to ±12.6 cm at 1 °C temperature change)
RoHS compliant, halogen-free, MSL1	

Abbildung 4: BME280 Technische Daten[4]

In dieser Arbeit wird nur die Verwendung dieses Sensors zur Temperaturmessung untersucht. Der BME280 arbeitet mit einer Spannung von 3,3-5V und einem Temperaturerfassungsbereich von -40° C bis +85° C und einem Fehler von ±0,5° C. Dieses Temperaturmesssystem verbindet den BME280 über eine serielle I2C-Schnittstelle mit einem Mikrocontroller und führt eine Datenerfassung durch.

4.2.3 LCD-Display

Obwohl LCD- und einige andere Bildschirme die Interaktion zwischen Mensch und Maschine erheblich verbessern, haben sie eine gemeinsame Schwäche. Wenn sie an einen Controller angeschlossen sind, werden mehrere IOs des Controllers belegt, der nicht so viele äußere Ports hat. Außerdem schränkt es andere Funktionen des Controllers ein. Deshalb wurde der LCD1602 mit einem I2C-Bus entwickelt, um dieses Problem zu lösen. Der I2C-Bus ist eine Art serieller Bus, der von PHILIPS entwickelt wurde. Es ist ein hochleistungsfähiger serieller Bus, der über eine Busregelung und eine Hoch- oder Niedriggeschwindigkeits-Synchronisationsfunktion verfügt, die von mehreren Hostsystemen benötigt wird. Der I2C-Bus hat nur zwei bidirektionale Signalleitungen, die Serial Data Line (SDA) und die Serial Clock Line (SCL). Das blaue Potentiometer auf dem I2C LCD1602 dient zur Einstellung der Hintergrundbeleuchtung, um die Anzeige auf dem I2C LCD1602 zu erleichtern.[5]

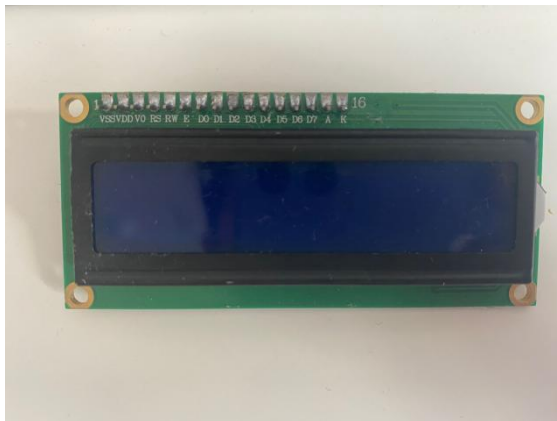


Abbildung 5: LCD1602 Vorderseite

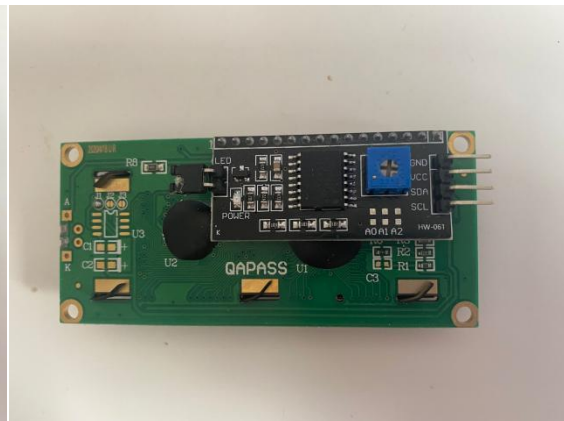


Abbildung 6: LCD1602 Rückseite

Das 1602 LCD ist ein Punktmatrix-LCD-Modul zur Anzeige von Buchstaben, Zahlen, Symbolen usw.

1602 LCD bedeutet, dass das Display 16x2 ist, d.h. es kann zwei Zeilen mit 16 Zeichen pro Zeile LCD-Modul (Anzeige von Zeichen und Zahlen) anzeigen.

4.2.4 Taste



Abbildung 7: Taste

Da bei diesem Temperaturmesssystem mehrere Temperaturwerte auf einem einzigen LCD-Display angezeigt werden müssen, sind zwei Tasten erforderlich, um die Art der auf dem LCD-Display angezeigten Werte zu steuern. Wenn eine Taste gedrückt wird, kann das Signal von den Ausgangspins des Mikrocontrollers ausgegeben werden und der entsprechende Temperaturwert wird auf dem LCD-Display angezeigt.

4.2.5 Summer

Ein Summer ist ein akustisches Signalgerät, das mechanisch, elektromechanisch oder piezoelektrisch (kurz Piezo) sein kann. Typische Verwendungszwecke für Summer und Piepser sind Alarmvorrichtungen, Zeitschaltuhren und die Bestätigung von Benutzereingaben wie Mausklicks oder Tastendrucke.[6]



Abbildung 8: Summer[7]

Bei diesem System wird ein aktiver piezoelektrischer Buzzer verwendet. Die Funktionsweise aktiver Buzzer: Die eingespeiste Gleichstromleistung wird durch das Schwingungssystem verstärkt und abgetastet, um ein akustisches Signal unter der Wirkung eines Resonanzkörpers zu erzeugen.

4.2.6 Hardwarekonzept

Ein Blockdiagramm des Hardwarekonzepts ist unten abgebildet. Die verwendeten Mikrocontroller ist Arduino Mega 2560. Wie in Abbildung 8 unten dargestellt, ist der BME280 über die serielle I2C-Schnittstelle mit dem Mikrocontroller verbunden. Der LCD-Display verbindet seine vier Pins über den seriellen I2C-Port mit den entsprechenden Pins des Mikrocontrollers.

Der längere Pin des Summers ist mit digitalen Pin 5 des Mikrocontrollers verbunden und der kürzere Pin ist mit GND verbunden. Eine Temperaturschwelle kann durch Programmierung festgelegt werden. Wenn die gemessene Temperatur den eingestellten Schwellenwert überschreitet, gibt der Mikrocontroller über digitalen Pin 5 einen hohen Pegel aus, wodurch der Summer einen Alarm auslöst.

Die digitalen Pins 3 und 4 des Mikrocontrollers sind jeweils mit zwei Tasten verbunden. Durch separates Drücken dieser beiden Tasten können die Mindest-, Höchst- und Durchschnittstemperaturen auf dem LCD-Bildschirm angezeigt werden.

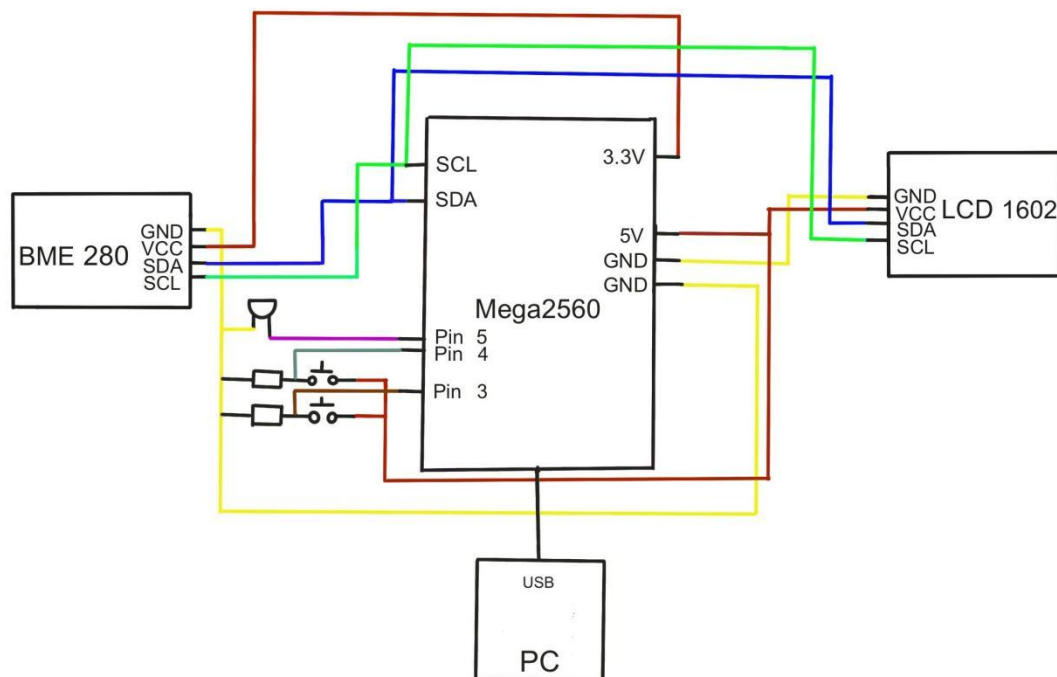


Abbildung 9: Blockdiagramm des Hardwarekonzepts

4.3 Software

Die Programmierung ist für den Aufbau eines vollständigen Temperaturmesssystems unerlässlich. Das System kann so programmiert werden, dass es die Temperaturschwelle festlegt und auch steuert, was auf dem LCD-Display angezeigt wird. Der gesamte Code für dieses Temperaturmesssystem wurde mit der Arduino IDE geschrieben.

4.3.1 Softwarekonzept

Das Flussdiagramm des Softwarekonzepts für dieses Temperaturmesssystem ist unten dargestellt. Der Temperatursensor gibt das Temperatursignal über die serielle I2C-Schnittstelle an den Mikrocontroller weiter. Der Mikrocontroller vergleicht die Eingangsdaten mit einem voreingestellten Temperaturschwellenwert. Wenn die Temperatur den eingestellten Schwellenwert überschreitet, gibt der Mikrocontroller einen hohen Pegel aus, um den Summer zu aktivieren und ihn zu alarmieren. Liegt die Temperatur innerhalb der eingestellten Temperaturschwelle und wird die Taste gedrückt, werden die Mindest-, Höchst- und Durchschnittstemperaturen auf dem LCD-Display angezeigt.

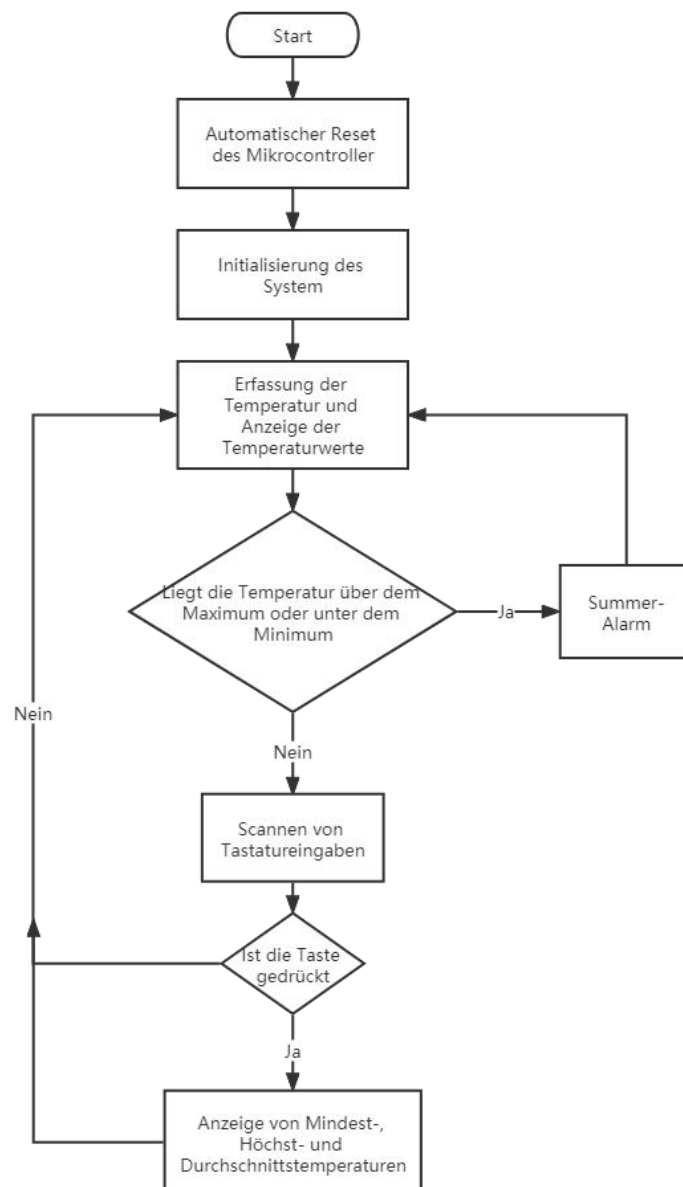


Abbildung 10: Flussdiagramm des Softwarekonzepts

4.3.2 Erläuterungen zu einigen Codes

Der erste Schritt bei der Programmierung mit der Arduino-IDE besteht darin, die Bibliotheken zu importieren, die dieses Programm verwenden wird. Das System erfordert die Verwendung von insgesamt drei Bibliotheken. Die Bibliotheken "Adafruit_Sensor" und "Adafruit_BME280" sind notwendig, um den BME280-Sensor zu verwenden, wie in der Abbildung unten gezeigt. Die Bibliothek "LiquidCrystal_I2C" ist notwendig, um das LCD 1602 zu verwenden. Die Bibliotheken "Adafruit_Sensor" und "LiquidCrystal_I2C" müssen aus dem Internet heruntergeladen und importiert werden, bevor sie verwendet werden können.

```
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
#include <LiquidCrystal_I2C.h>
```

Abbildung 11: Auswahl der erforderlichen Bibliotheken

Der Code für die Berechnung der Höchst- und Mindesttemperaturen ist in der nachstehenden Abbildung dargestellt. In diesem Code werden zwei if-Anweisungen verwendet, um die aktuell abgelesene Temperatur mit der höchsten und der niedrigsten zuvor gemessenen Temperatur zu vergleichen. Wenn die aktuelle Temperatur größer ist als die höchste zuvor gemessene Temperatur, wird die aktuelle Temperatur als höchste Temperatur gespeichert und ausgegeben. Wenn die aktuell gemessene Temperatur niedriger ist als die zuvor gemessene niedrigste Temperatur, wird die aktuelle Temperatur als niedrigste Temperatur gespeichert und ausgegeben. Ansonsten bleiben die Höchst- und Mindesttemperaturen gleich.

```
temperature = bme.readTemperature();
if (temperature > temperatureMax)
{
    temperatureMax = temperature;
}

if (temperature < temperatureMin)
{
    temperatureMin = temperature;
}
for (int thisReading = 0; thisReading < numReadings; thisReading++)
{
    readings[thisReading] = 0;
}
```

Abbildung 12: Codes für Höchst- und Mindesttemperaturen

Der Code zur Berechnung der Durchschnittstemperatur ist in der nachstehenden Abbildung dargestellt. Der erste Schritt besteht darin, den Stichprobenumfang zu bestimmen. Die Durchschnittstemperatur ist dann gleich der Summe der aktuell abgelesenen Temperaturwerte geteilt durch die Anzahl der abgelesenen Temperaturwerte.

```
if (temperature > temperatureMax)
{
    temperatureMax = temperature;
}

if (temperature < temperatureMin)
{
    temperatureMin = temperature;
}
total = total - readings[readIndex];
// 读入当前旋转电位计的数值,
// 并将其存储到数组的最后一位。
readings[readIndex] = temperature;
// 将最新读入的数值加入到总值中
total = total + readings[readIndex];
// 将数组指示索引值加1
readIndex++;
// 判断数组指示索引是否超出数组范围,
// 如果是, 将数组指示索引重置为0
if (readIndex >= numReadings)
{
    readIndex = 0;
}
// 计算平均值
average = total / numReadings;
// 将最终计算的平均值输出到串行调试窗口。
```

Abbildung 13: Codes für Durchschnittswerte

Die folgende Abbildung zeigt den Code für die Einstellung der Temperaturschwelle und des Summeralarms. Der Code ist sehr einfach. Auch hier werden zwei if-Anweisungen verwendet, um die aktuell gemessene Temperatur mit der eingestellten Temperaturschwelle zu vergleichen. Liegt die aktuell abgelesene Temperatur über dem eingestellten Maximalwert oder unter dem eingestellten Minimalwert, aktiviert der Mikrocontroller den Summer, indem er intermittierend einen hohen Pegel an Digitalpin 5 ausgibt und so eine Alarmwirkung erzeugt.

```
if(temperature > 28)
{
    digitalWrite(summer, HIGH);
    delay(50);
    digitalWrite(summer, LOW);
    delay(50);
    digitalWrite(summer, HIGH);
    delay(50);
    digitalWrite(summer, LOW);
}
if(temperature < 15 )
{
    digitalWrite(summer, HIGH);
    delay(50);
    digitalWrite(summer, LOW);
    delay(50);
    digitalWrite(summer, HIGH);
    delay(50);
    digitalWrite(summer, LOW);
}
```

Abbildung 14: Codes für Temperaturschwellen und Summer

5 Technischer Aufbau

Auf der Grundlage des oben gezeigten Blockdiagramms des Hardwarekonzepts (Abbildung 8) wurde ein kompletter Systemschaltplan entworfen (siehe unten). Für den Anschluss des Mikrocontrollers, des Temperatursensors, des LCD-Display und der Schaltungskomponenten wurden eine lötfreie Breadboard und Dupont-Draht verwendet, da man befürchtete, dass der erste Anschluss der Schaltung fehleranfällig sein könnte. Da das lötfreie Breadboard nicht gelötet werden muss, ist es wiederverwendbar. Das macht es einfach, temporäre Prototypen zu erstellen und mit dem Schaltungsdesign zu experimentieren.[8]

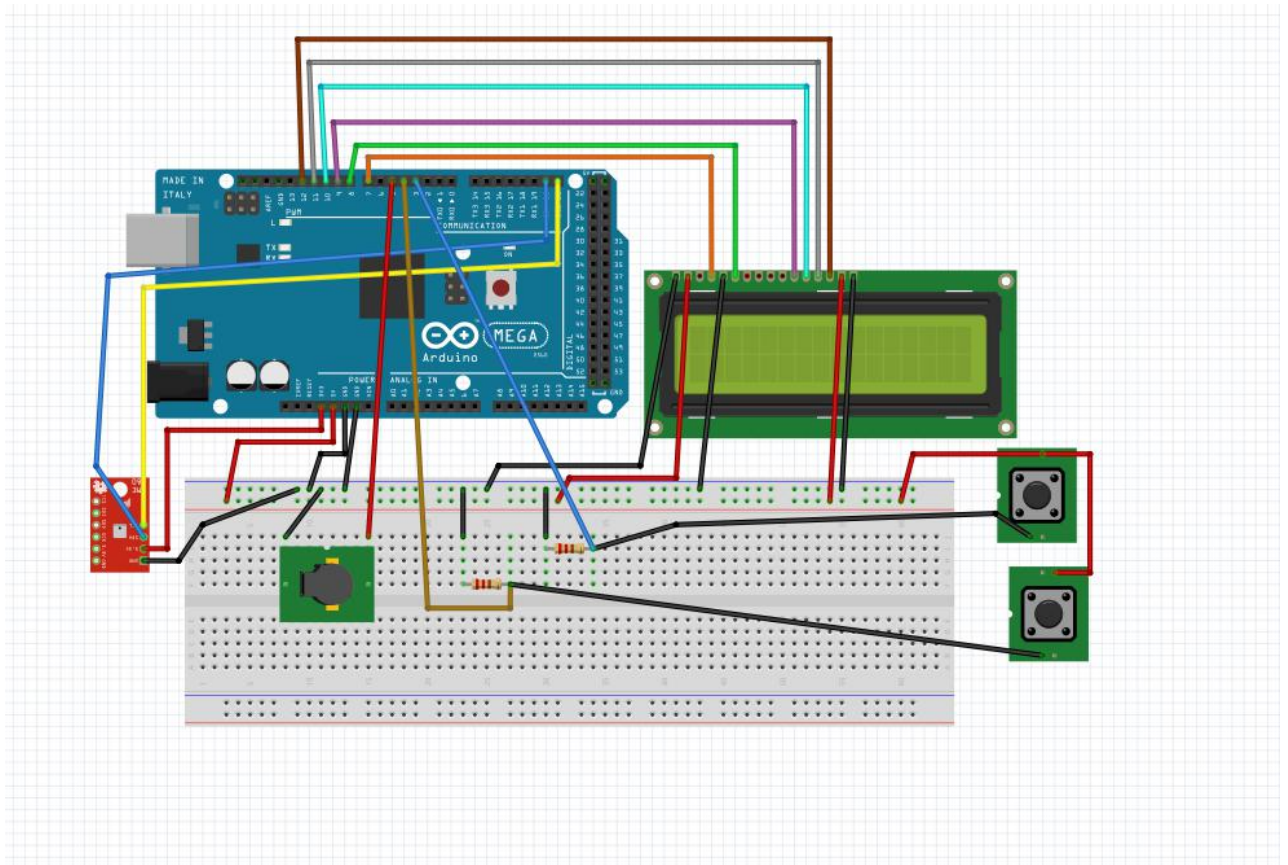


Abbildung 15: Bastelanleitung des Systems

- Erforderliche Komponenten:

Bauteil- Typ	Größe	Anzahl
Widerstand	10 K Ω	2
Summer	/	1
Taste	/	2
Mikrocontroller	/	1
LCD-Display	/	1
Temperatursensor	/	1

Tabelle 1: Erforderliche Komponenten

Durch die Verwendung von Dupont-Drähten und lötfreien Breadboards sind elektrische Komponenten wie Mikrocontroller und Sensoren nicht fest verbunden. Dupont-Drähten lösen sich häufig und fallen während der Prüfung aufgrund von Kollisionen usw. heraus, was dazu führt, dass das System nicht richtig funktioniert. Und wegen der vielen Dupont-Drähte wirkt das ganze System unübersichtlich. Nach der Fertigstellung dieses Temperaturmesssystems muss das gesamte System verpackt werden, was bei einer zu großen Anzahl von Drähten in Bezug auf Wartung und spätere Änderungen unpraktisch sein kann.

Es ist daher notwendig, eine Platine in der gleichen Größe wie der Mikrocontroller zu entwerfen, und die zu verwendenden digitalen Pins werden auf der Leiterplatte in der gleichen Weise positioniert wie die Pins des Mikrocontrollers. So können einige der elektrischen Komponenten direkt auf die Leiterplatte gelötet werden, die dann über die Steckerleiste des Mikrocontrollers direkt mit diesem verbunden wird. Dies reduziert die Verwendung von Dupont-Drähten und verleiht dem gesamten System ein saubereres und klareres Aussehen. Außerdem sind die elektrischen Komponenten sicherer miteinander verbunden, um den ordnungsgemäßen Betrieb des Systems zu gewährleisten.

Die folgende Abbildung zeigt eine 3D-Ansicht der entworfenen Leiterplatte.

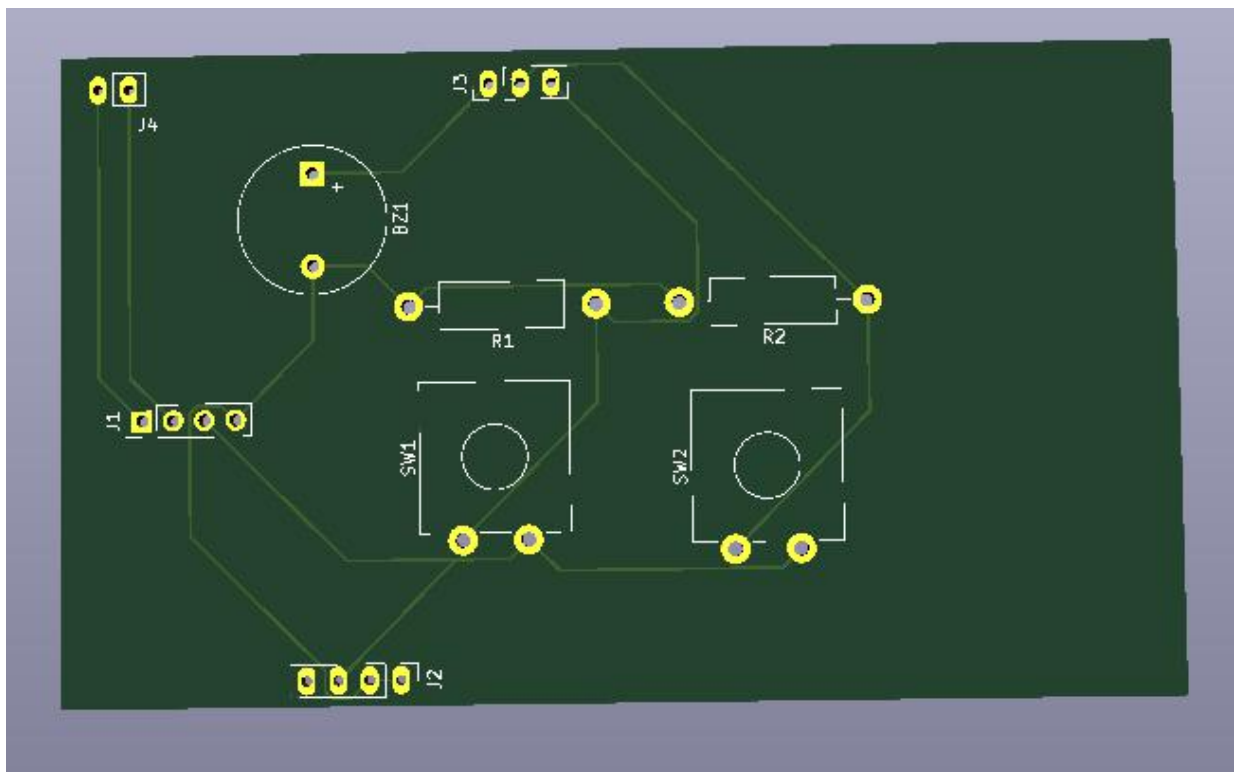


Abbildung 16: 3D-Betrachter(Vorderseite)

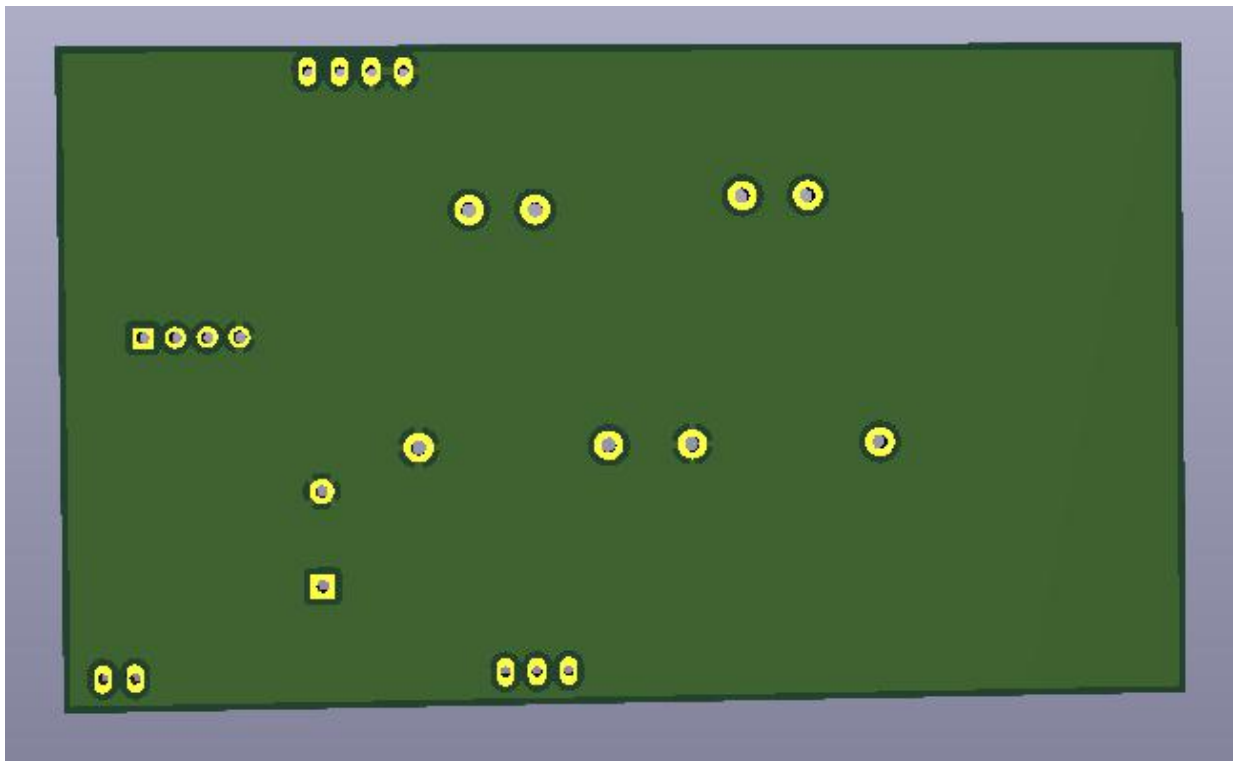


Abbildung 17: 3D-Betrachter(Rückseite)

6 Experimentelle Ablauf und Ergebnisse

6.1 Beschreibung des experimentellen Inhalts

Ziel der Arbeit ist die prototypische Aufbau eines Systems zur Messung bzw. Langzeitmessung von Temperaturen. Das System sollte die momentane Temperatur messen und die Höchst-, Mindest- und Durchschnittstemperaturen berechnen und diese Werte dann auf dem LCD-Display anzeigen sowie einen Summeralarm ausgeben, wenn die momentane Temperatur die eingestellte Temperaturschwelle überschreitet. Das Experiment kann daher in drei Teile gegliedert und nach den folgenden Schritten durchgeführt werden.

6.2 Experimentelle Vorbereitung

Der erste Schritt bei der Vorbereitung des Experiments besteht darin, sicherzustellen, dass die Schaltkreise des Systems vollständig sind und dass die Kabel richtig angeschlossen sind. Das fertige Programm wird dann über die serielle Schnittstelle des PCs auf den Mikrocontroller gebrannt. Nachdem das Programm erfolgreich gebrannt wurde, blinkt die TX-LED weiter und die Startseite des LCD-Displays wird unten angezeigt. Dann ist es an der Zeit, mit Experimenten zu beginnen.

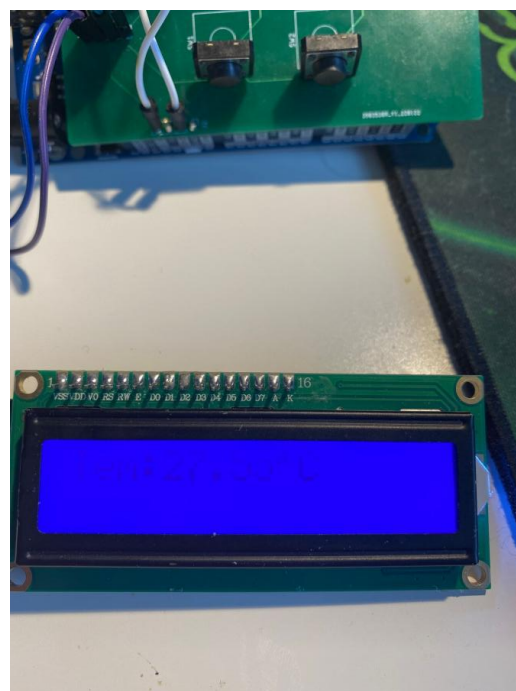


Abbildung 18: Startseite des LCD-Displays

6.3 Testen der Tasten

Zwei Tasten auf der Platine steuern die Momentan-, Maximal-, Minimal- und Durchschnittstemperatur auf dem LCD-Display. Wenn keine der beiden Tasten gedrückt wird, zeigt das LCD-Display die momentane Temperatur an. Wenn die Taste 1 gedrückt wird, zeigt das LCD-Display die Höchst- und Mindesttemperatur an. Wenn die Taste 2 gedrückt wird, zeigt das LCD-Display die Durchschnittstemperatur an. Das LCD-Display zeigt weiterhin die momentane Temperatur an, sobald die Taste losgelassen wird.

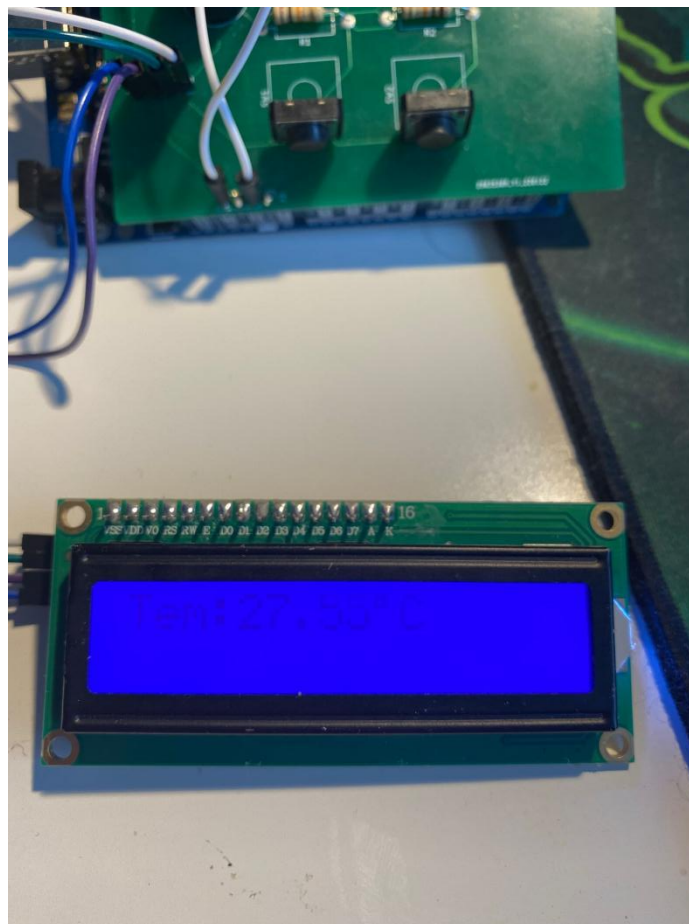


Abbildung 19: Anzeige der Momentantemperatur



Abbildung 20: Anzeige von Maximal- und Minimaltemperatur



Abbildung 21: Anzeige von Durchschnittstemperatur

6.4 Summer testen

Zum Testen des Summers müssen Sie zunächst die Temperaturschwelle im Programm einstellen. Die hier eingestellte Temperaturschwelle liegt bei 15°C - 23°C . Da das Experiment in einem Innenraum durchgeführt wird, ändert sich die Temperatur nicht wesentlich. Die Temperatur wird also über den Schwellenwert angehoben, indem der Finger mit dem Temperatursensor in Kontakt gebracht wird. Wie im Diagramm unten dargestellt, hat der Temperatursensor zu diesem Zeitpunkt $26,63^{\circ}\text{C}$ gemessen, was den eingestellten Temperaturschwellenwert überschritt, so dass der Summer ertönte.

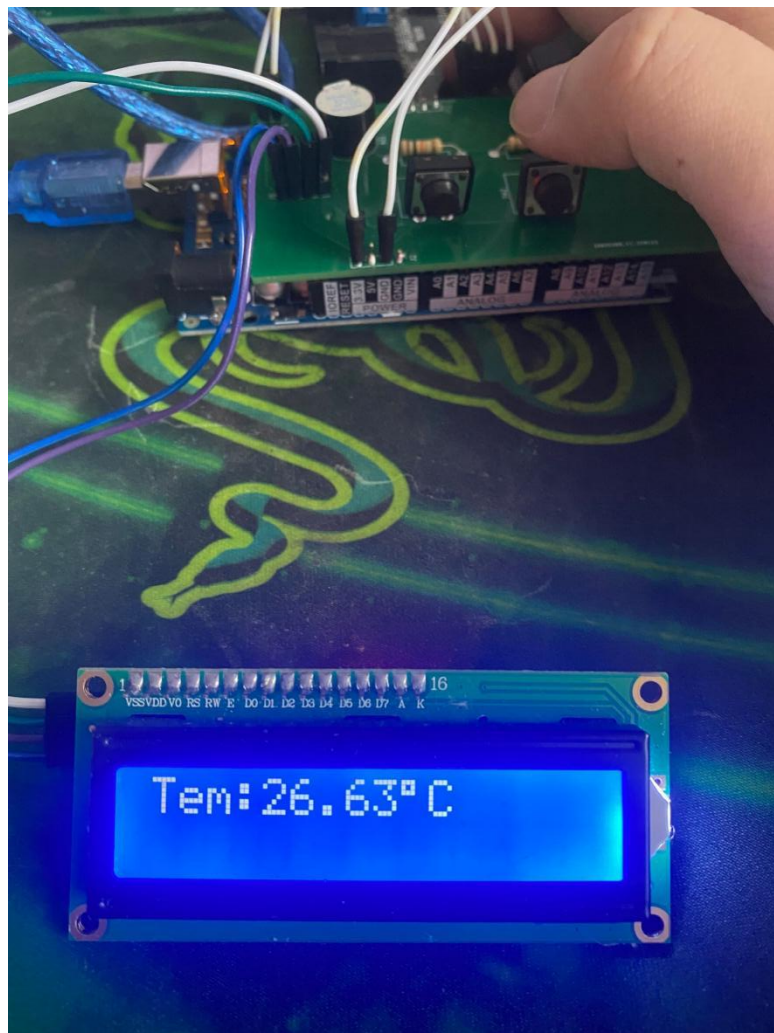


Abbildung 22: Summer testen

6.5 Test der Datenübertragung

Ein vollständiges System muss in der Lage sein, die gemessenen Daten zu analysieren. In dieser Arbeit wurde PLX-DAQ verwendet. PLX-DAQ ist eine Excel-Tabelle mit Makro-Definitionen, die es ermöglicht, Daten direkt aus Arduino zu lesen und zu speichern, wenn die entsprechende Konfiguration vorgenommen wird.

Zunächst muss etwas Code in das Programm eingefügt werden. Der folgende Code muss zu void setup() hinzugefügt werden.

```
Serial.begin(9600);  
Serial.println("CLEARSHEET"); // clears sheet starting at row 1  
Serial.println("LABEL,TIME,Temperature,Tem_Min,Tem_Max,Tem_AVG");
```

Abbildung 23: Code für PLX-DAQ 1[9]

In der ersten Zeile wird die Baudrate für die Übertragung festgelegt. Die zweite Code-Zeile bedeutet, dass die Daten gelöscht werden und die dritte Code-Zeile bedeutet, dass die Tabelle des PLX-DAQ gesetzt wird.

Der folgende Code muss zu void loop() hinzugefügt werden.

```
Serial.print("DATA, TIME,");  
Serial.print( temperature );  
Serial.print(",");  
Serial.print( temperatureMin );  
Serial.print(",");  
Serial.print( temperatureMax );  
Serial.print(",");  
Serial.print( average );
```

Abbildung 24: Code für PLX-DAQ 2[9]

DATA ermöglicht es dem PLX-DAQ, die Daten zu lesen. Die zweite Zeile ist die Variable, die ausgegeben werden soll. Die dritte Zeile dient dazu, dem PLX-DAQ die Möglichkeit zu geben, die Daten in einer anderen Spalte zu speichern, da sie sonst alle in einer Spalte stehen würden.

Sobald das Programm eingerichtet ist, kann der Mikrocontroller an den PC angeschlossen werden. PLX-DAQ läuft auf einem PC. Die PLX-DAQ-Benutzeroberfläche ist unten abgebildet.



Abbildung 25: PLX-DAQ

Nachdem die Baudrate auf 9600 und die Schnittstelle auf 4 (die Nummer des Ports, an dem der Mikrocontroller mit dem PC verbunden ist) eingestellt wurde, kann der Test beginnen. Da sich die Temperatur über einen kurzen Zeitraum nicht dramatisch ändert, wurde das Temperatursmesssystem bei diesem Versuch von außerhalb des Raumes in den Raum gebracht. Da die Temperatur konstant ansteigt, überschneiden sich die Liniendiagramme der Momentantemperatur und der Höchsttemperatur fast genau, so dass die Liniendiagramme der Momentantemperatur getrennt von den Liniendiagrammen der anderen Daten dargestellt werden.

Unten sehen Sie ein Liniendiagramm der Versuchsdaten.

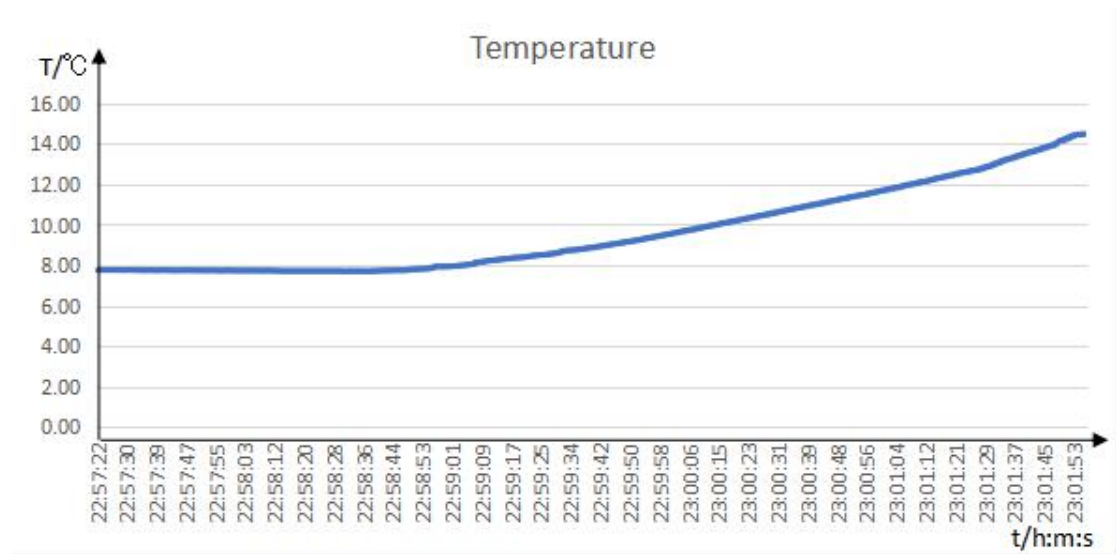


Abbildung 26: Liniendiagramm der momentanen Temperatur

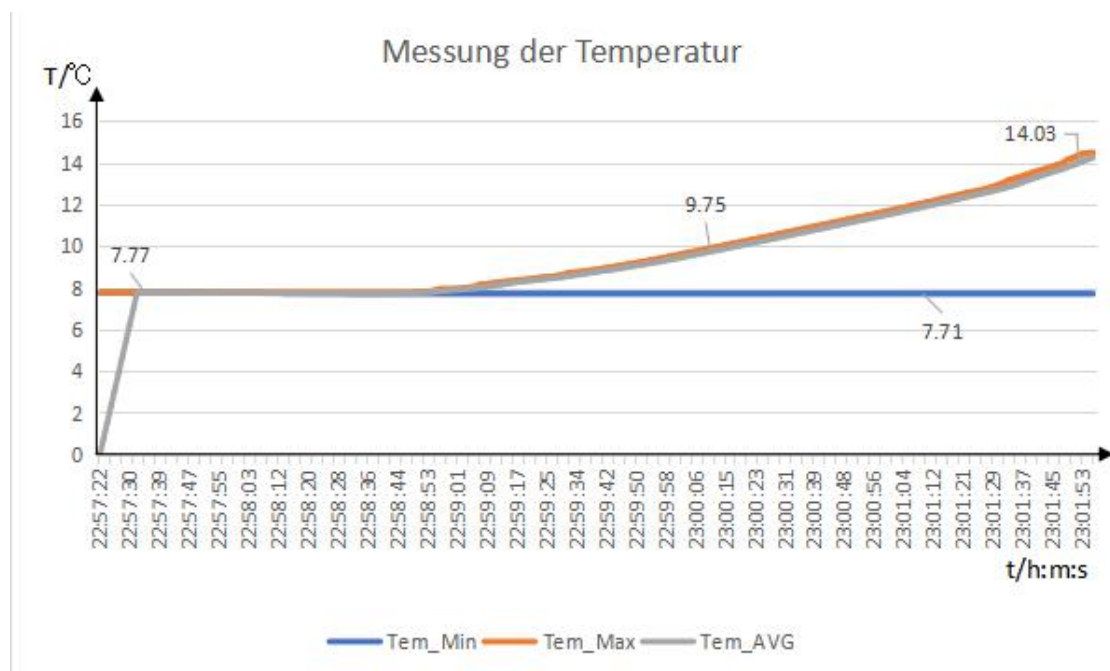


Abbildung 27: Liniendiagramm der verbleibenden Werte

6.6 Versuchsauswertung

Die Momentan-, Höchst-, Mindest- und Durchschnittstemperaturen werden gemessen und auf dem LCD-Display angezeigt. Ein Summer ertönt, wenn die momentane Temperatur die eingestellte Temperaturschwelle überschreitet. Während des Tests wurde jedoch ein Problem festgestellt: Die Temperatur, die gemessen wurde, als sich der Sensor an der Zimmertür befand, war niedriger als die Temperatur, die neben der Raumheizung gemessen wurde. Wenn sich der Sensor langsam dem Heizgerät nähert und dann langsam von ihm wegbewegt, scheint die gemessene Temperatur zu steigen und dann zu fallen. Die Ergebnisse des Experiments sind in dem nachstehenden Liniendiagramm dargestellt.

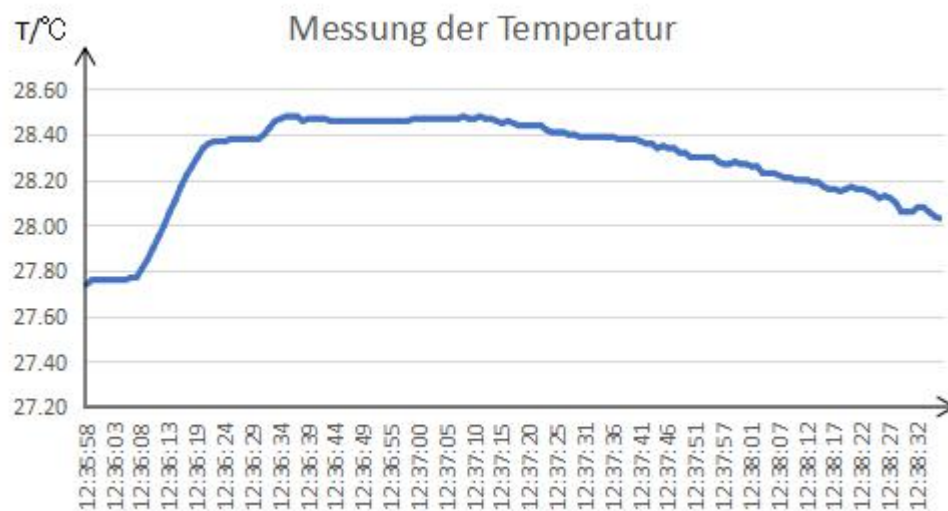


Abbildung 28: Temperatur in verschiedenen Positionen des Raumes

Daraus folgt, dass der Sensor, wenn die Umgebungstemperatur gemessen werden muss, in der Nähe der Mitte der Umgebung und entfernt von wärmeerzeugenden Geräten angebracht werden sollte. Obwohl der daraus resultierende Fehler innerhalb von 1°C liegt, muss dies für einige Standorte mit strengeren Temperaturanforderungen vermieden werden.

7 Zusammenfassung

Nach einer Reihe von Tests kann das Temperaturmesssystem Momentan-, Höchst-, Mindest- und Durchschnittstemperaturen messen. Diese Daten können auf dem LCD-Display angezeigt werden. Das System verfügt über eine Alarmfunktion, die es ermöglicht, einen Alarm über einen Summer auszulösen, wenn die momentane Temperatur einen festgelegten Schwellenwert überschreitet.

Der Entwicklungsprozess für dieses Projekt verlief in fünf Phasen: Verstehen der Funktionsweise der benötigten Komponenten und ihrer Verwendung, Entwurf und Optimierung der Hardwareschaltung, Softwareprogrammierung, Entwurf der Leiterplatte sowie Testen und Einstellen des gesamten Systems. Bei der Durchführung des Projekts traten auch zahlreiche Schwierigkeiten auf. Wie schließt man den Sensor an den Mikrocontroller an und verwendet ihn? Wie kann man die momentane, maximale, minimale und durchschnittliche Temperatur auf einem einzigen LCD-Display anzeigen? Schließlich konnten die Probleme durch die Konsultation zahlreicher Informationen aus dem Internet erfolgreich gelöst werden. Das LCD 1602 kann nur zwei Datenzeilen mit nur 16 Zeichenmodulen pro Zeile anzeigen. Deshalb wurden dem System zwei Schalter hinzugefügt und so programmiert, dass die auf dem LCD-Display angezeigten Daten über die Schalter umgeschaltet werden können. Nach Studien und Untersuchungen sowie einer kontinuierlichen Verbesserung und Optimierung des gesamten Systems konnten die im ursprünglichen Entwurf vorgesehenen Funktionen umgesetzt werden.

Literaturverzeichnis

Internet

[1] Arduino.cc: "Arduino Mega 2560", 2022

URL: <https://www.arduino.cc/en/Main/arduinoBoardMega2560>

[2] bosch-sensortec:"BME 280", Juni 2015

URL: https://www.bosch-sensortec.com/media/boschsensortec/downloads/product_flyer/bst-bme280-fl000.pdf

[3] learn.sparkfun:"Introduction to the BME280", 2022

URL: <https://learn.sparkfun.com/tutorials/microclimate-kit-experiment-guide/experiment-1-reading-the-temperature-humidity-and-pressure>

[4] bosch-sensortec:"BME 280 Technische Daten", Juni 2015

URL: https://www.bosch-sensortec.com/media/boschsensortec/downloads/product_flyer/bst-bme280-fl000.pdf

[5] sunfounder: "I2C LCD1602", 23 Dezember 2019

URL: http://wiki.sunfounder.cc/index.php?title=I%C2%B2C_LCD1602

[6] wikipedia: "Buzzer", 17 November 2021

URL: <https://en.wikipedia.org/wiki/Buzzer>

[7] aliexpress:"5 V Aktive Alarm Summer", 2022

URL: <https://de.aliexpress.com/item/4000595791841.html?gatewayAdapt=glo2deu>

[8] wikipedia: "Breadboard", 25 Februar 2021

URL: <https://en.wikipedia.org/wiki/Breadboard>

[9] Code für PLX-DAQ, 2022

URL: <http://jjpaid.blogspot.com/2017/12/plx-daq-excelarduino.html>

[10] arduino-projekte.info: "Technische Daten", 2022

URL: <https://arduino-projekte.info/arduino-mega-2560/>

Anlagen

Anlagen, Verpackung
Anlagen, Arduino Mega2560 Technische Daten
Anlagen, Schaltplan
Anlagen, PCB Board
Anlagen, Bauteil: footprint Zuordnungen
Anlagen, Gerberdatei
Anlagen, Vollständiger Software-Code

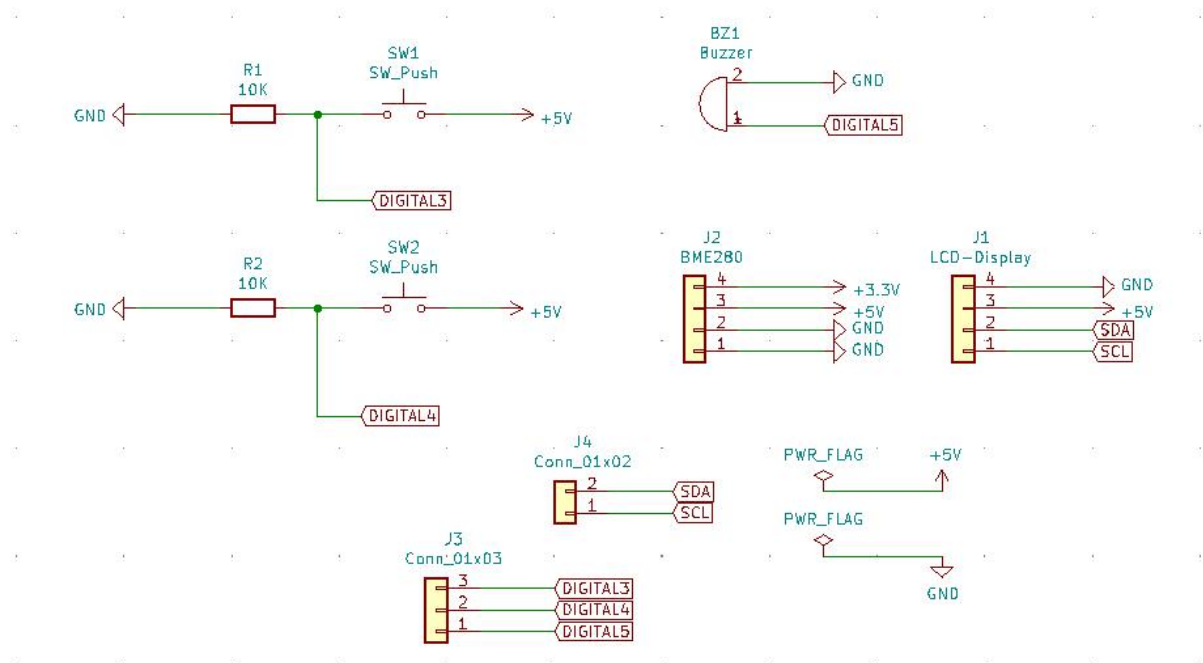
Anlagen, Verpackung



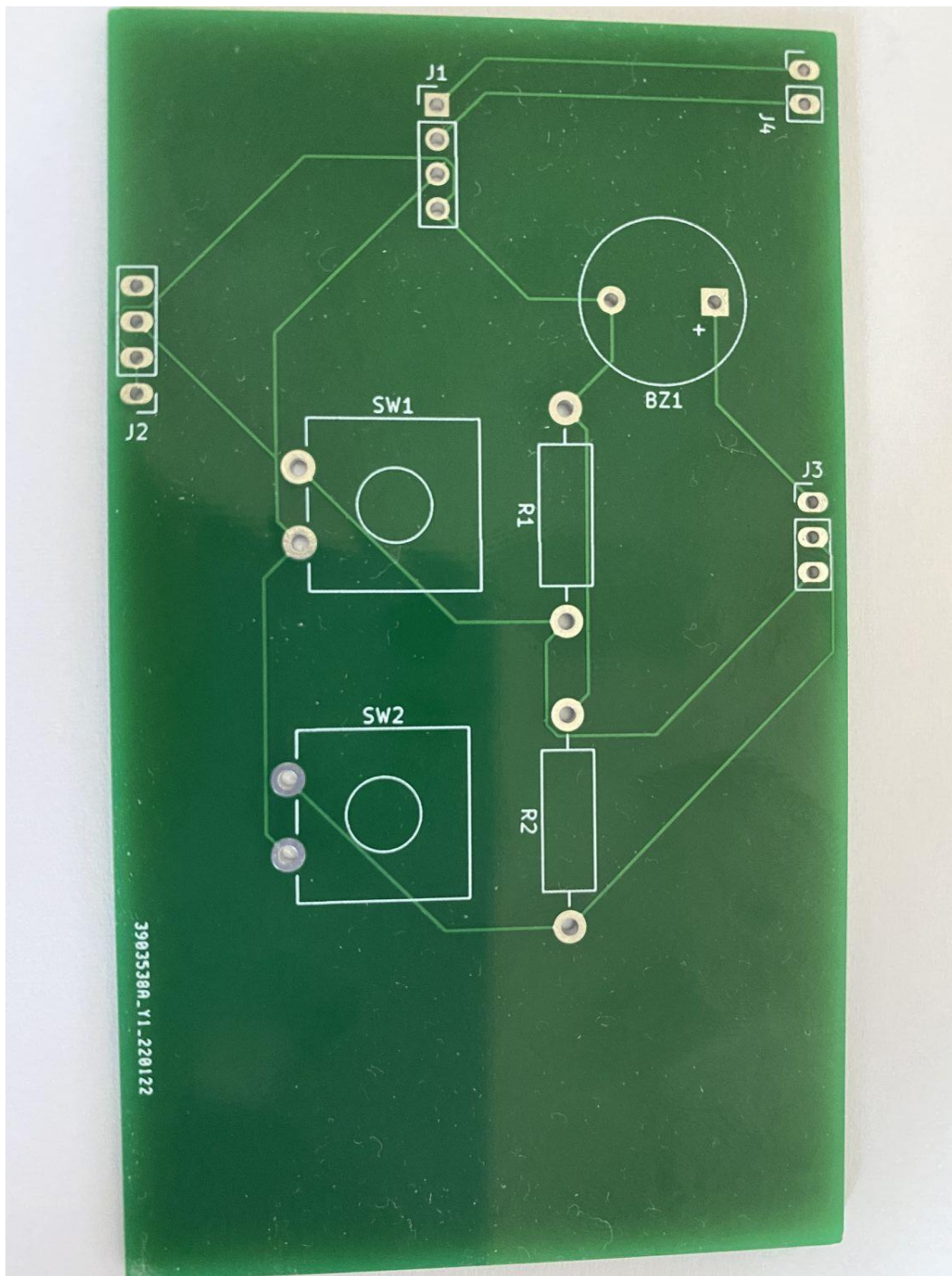
Anlagen, Arduino Mega2560 Technische Daten^[10]

Microcontroller	ATmega2560
Betriebsspannung	5V
Eingangsspannung (empfohlen)	7-12V
Eingangsspannung (Max.)	6-20V
Digital E/A Pins	54 (davon 15 PWM-Ausgang)
PWM Digital E/A Pins	15
Analog Eingangs Pins	16
DC Strom pro E/A Pin	20 mA
DC Strom für 3.3V Pin	50 mA
Flash Memory	256 KB 8 KB werden vom Bootloader verwendet
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
CPU	8-bit
Länge	101.52 mm
Breite	53.3 mm
Gewicht	37 g
Stromverbrauch Normalbetrieb	70,2 mA
Stromverbrauch Schlafmodus	29,4 mA

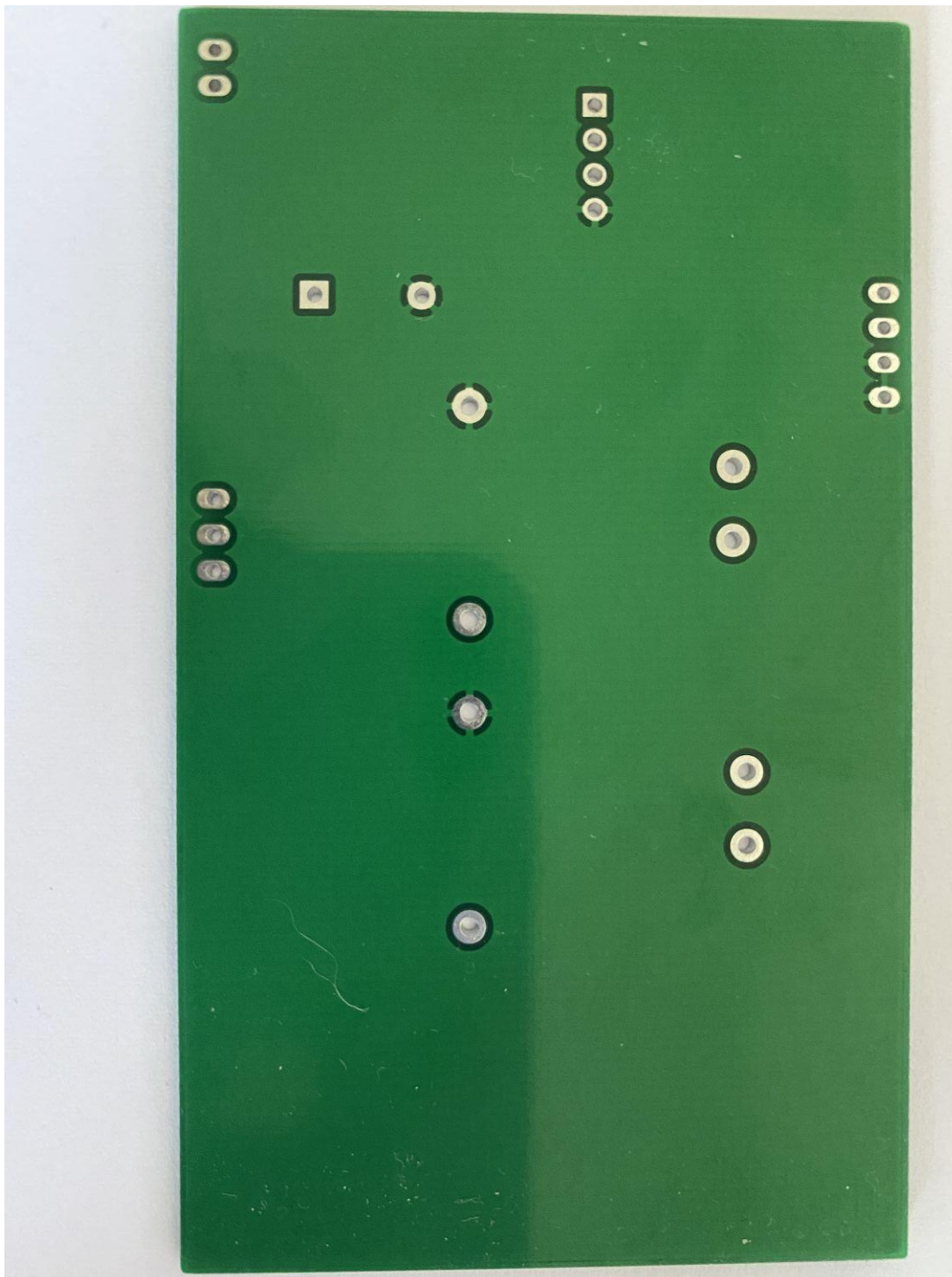
Anlagen, Schaltplan



Anlagen, PCB Board



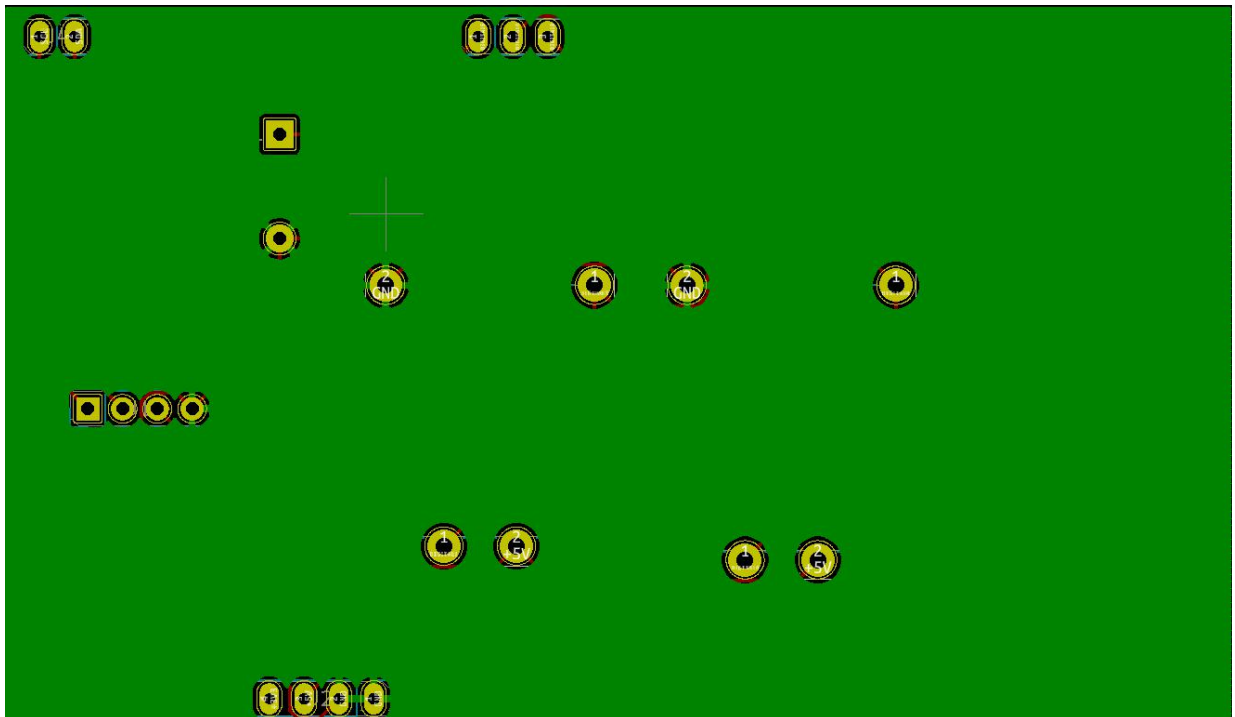
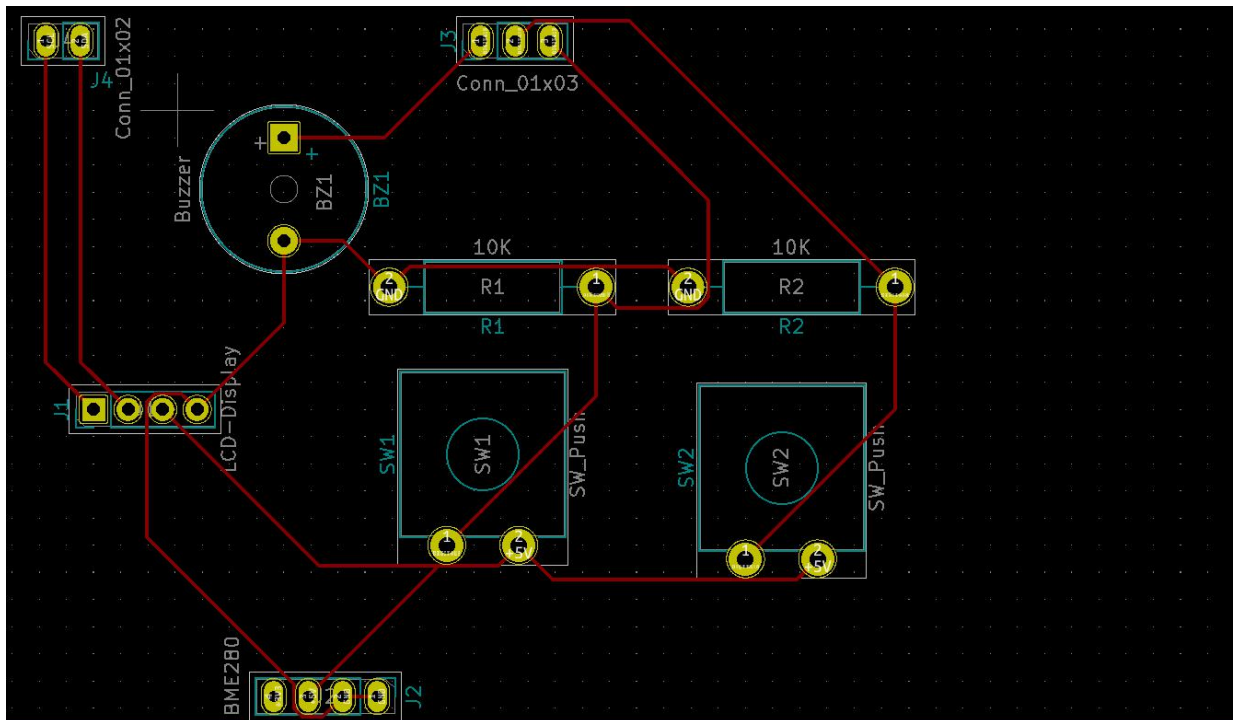
Anlagen, PCB Board



Anlagen, Bauteil: footprint Zuordnungen

1	BZ1 -	Buzzer : Buzzer_Beeper:Buzzer_12x9.5RM7.6
2	J1 -	LCD-Display : Connector_PinHeader_2.54mm:PinHeader_1x04_P2.54mm_Vertical
3	J2 -	BME280 : Connector_PinHeader_2.54mm:PinHeader_1x04_P2.54mm_Vertical
4	J3 -	Conn_01x03 : Connector_PinHeader_2.54mm:PinHeader_1x03_P2.54mm_Vertical
5	J4 -	Conn_01x02 : Connector_PinHeader_2.54mm:PinHeader_1x02_P2.54mm_Vertical
6	R1 -	10K : Resistor_THT:R_Axial_DIN0411_L9.9mm_D3.6mm_P15.24mm_Horizontal
7	R2 -	10K : Resistor_THT:R_Axial_DIN0411_L9.9mm_D3.6mm_P15.24mm_Horizontal
8	SW1 -	SW_Push : Button_Switch_THT:SW_CW_GPTS203211B
9	SW2 -	SW_Push : Button_Switch_THT:SW_CW_GPTS203211B

Anlagen, Gerberdatei



Anlagen, Vollständiger Software-Code

```
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
#include <LiquidCrystal_I2C.h>
#define summer 5

Adafruit_BME280 bme; // I2C
LiquidCrystal_I2C lcd(0x27,16,2);

float temperature=0;
float temperatureMin=30;
float temperatureMax=0;
const int numReadings = 1000;
float readings[numReadings];
int readIndex = 0;
float total = 0;
float average = 0;
const int ButtonPin_1 = 3;
int ButtonState_1 = 0;
const int ButtonPin_2 = 4;
int ButtonState_2 = 0;

void setup(void)
{
  Serial.begin(9600);
  Serial.println("CLEAR SHEET"); // clears sheet starting at row 1
  Serial.println("LABEL,TIME,Temperature,Tem_Min,Tem_Max,Tem_AVG");

  lcd.backlight();
  lcd.init();

  bool status;
  // default settings
  status = bme.begin(0x76); //The I2C address of the sensor I use is 0x76

  pinMode(summer,OUTPUT);

  temperature = bme.readTemperature();
  if (temperature > temperatureMax)
  {
    temperatureMax = temperature;
  }

  if (temperature < temperatureMin)
  {
```

```
    temperatureMin = temperature;
  }
  for (int thisReading = 0; thisReading < numReadings; thisReading++)
  {
    readings[thisReading] = 0;
  }
}

void loop()
{
  Serial.print("DATA, TIME,");
  Serial.print( temperature );
  Serial.print(",");
  Serial.print( temperatureMin );
  Serial.print(",");
  Serial.print( temperatureMax );
  Serial.print(",");
  Serial.print( average );
  Serial.print(",");

  delay(1000);

  temperature = bme.readTemperature();

  if (temperature > temperatureMax)
  {
    temperatureMax = temperature;
  }

  if (temperature < temperatureMin)
  {
    temperatureMin = temperature;
  }
  total = total - readings[readIndex];

  readings[readIndex] = temperature;

  total = total + readings[readIndex];

  readIndex++;

  if (readIndex >= numReadings)
  {
    readIndex = 0;
  }

  average = total / numReadings;
```

```
Serial.println(average);

delay(1);

if(temperature > 23)
{

digitalWrite(summer, HIGH);
  delay(50);
  digitalWrite(summer, LOW);
  delay(50);
  digitalWrite(summer, HIGH);
  delay(50);
  digitalWrite(summer, LOW);

}
if(temperature < 15 )
{
  digitalWrite(summer, HIGH);
  delay(50);
  digitalWrite(summer, LOW);
  delay(50);
  digitalWrite(summer, HIGH);
  delay(50);
  digitalWrite(summer, LOW);
}

//Printing Temperature
String temperatureString = String(temperature,2);
String temperatureMinString = String(temperatureMin,2);
String temperatureMaxString = String(temperatureMax,2);
String averageString = String(average,2);
ButtonState_1 = digitalRead(ButtonPin_1);
if (ButtonState_1 == LOW){
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Tem:");
  lcd.print(temperatureString);
  lcd.print((char)223);
  lcd.print("C ");

}else{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Min:");
  lcd.print(temperatureMinString);
  lcd.print((char)223);
  lcd.print("C ");
  lcd.setCursor(0,1);
```

```
    lcd.print("Max:");  
    lcd.print(temperatureMaxString);  
    lcd.print((char)223);  
    lcd.print("C ");  
  
    }  
  
    ButtonState_2 = digitalRead(ButtonPin_2);  
    if (ButtonState_2 == HIGH){  
  
        lcd.clear();  
        lcd.setCursor(0,0);  
        lcd.print("AVG:");  
        lcd.print(averageString);  
        lcd.print((char)223);  
        lcd.print("C ");  
    }  
}
```

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Mittweida, den 21.März.2022

Xin Sun