



Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

G762 Interdisziplinäres Projekt

Modell Gebäudetechnik

Alexander Puslat (577079)
Maurice Ruchti (577067)

WiSe 22/23

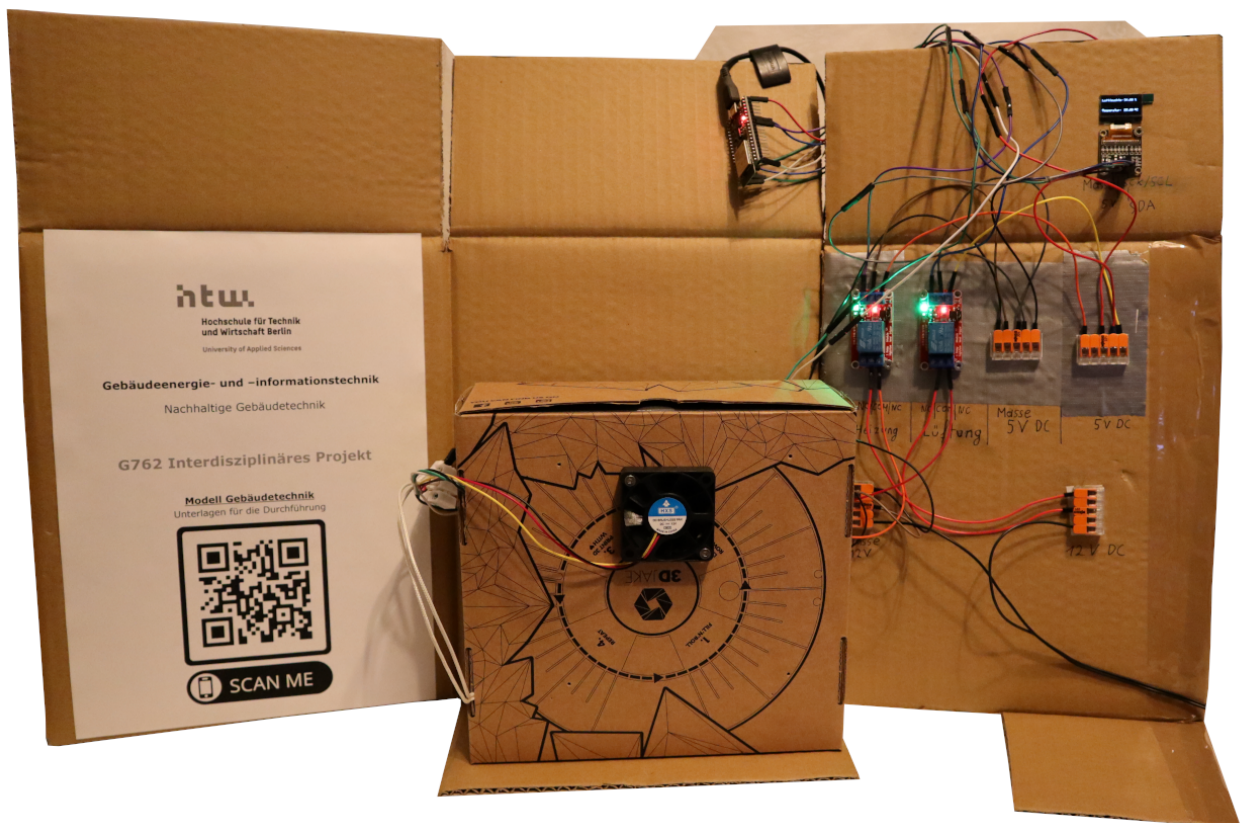
Dozent:
Frau Prof. Dr.-Ing. habil. Müller

Studiengang: GEIT

Gruppe: 2

Berlin, stand: 01.02.2023

Anleitung für die Erstellung eines Modells zur simulierung einer Gebäudeheizung und -Lüftung



Gesamtaufbau des fertigen Projekts; Foto Ruchti 03.01.2023

Inhaltsverzeichnis

Vorwort	3
Bauteile	4
Software	11
Bauanleitung	12
Vorbereitung	12
Verkabelung	12
Code	14
Der Programmiercode	15
Inbetriebnahme	18
Erweiterungen	19
Quellen	21
Abbildungsverzeichnis	21

Vorwort

Die technische Gebäudeausrüstung, zu der unter anderem die Gebäudeheizung und etwaige Lüftungsanlagen zählen, dient zur Erreichung von behaglichen Wohn- bzw. Raumklima sowie der Einsparung von Energie. Gerade letzteres wird in näherer Zeit immer wichtiger. Umwelt- und Klimaschutz durch die Einsparung von CO₂ sind heutzutage unabdingbar.

Dieses Projekt soll mittels eines vereinfachten Gebäudemodells mit technischer Gebäudeausrüstung aufzeigen, mit was wir Studierenden des Studiengangs "Gebäudeenergie- und Informationstechnik" kurz GEIT uns befassen und Interesse wecken.

Das Projekt wird möglichst einfach gehalten und gibt einen Einblick in die Themen der Gebäudeausrüstung, Elektronische Bauteile sowie das Programmieren.

Das fertige Modell besitzt eine Heizung und eine Lüftungsanlage in Form einer 12 Volt Heizmatte und eines 12 Volt Ventilators. Durch einen Luftfeuchtigkeits- und Temperatursensor wird der Zustand in einer kleinen Box, welche als vereinfachtes Modell eines Gebäudes dient, beobachtet.

Eine Schaltung mit Relais und einem Mikrocontroller regelt durch das jeweilige Zu- und Abschalten der Heizmatte und des Ventilators die Luftfeuchtigkeit und die Temperatur in der Box. Lasten der Raumluft werden durch Atemluft simuliert, da diese schnell die Luftfeuchtigkeit auf ein sehr hohes Niveau bringt.

Maxima und Minima für Temperatur und Luftfeuchtigkeit können im Programmiercode des Mikrocontrollers eingestellt werden.

Diese Anleitung gibt Lehrkräften eine Übersicht der verbauten Teile, der Verkabelung sowie der Programmierung, um den Aufbau mit Schülern nachzubauen und in Betrieb zu nehmen.

Das Projekt kann mit Schülern ab der achten Klasse in Zusammenarbeit mit einer sachkundigen Lehrkraft umgesetzt werden.

Bauteile

Microcontroller:

Der Microcontroller ist das Gehirn des Aufbaus. Er verarbeitet Messwerte und wir können in seiner Programmierung Ereignisse hinterlegen, die bei bestimmten Messwerten oder deren Über- oder Unterschreitung eintreffen sollen.

An ihm können Sensoren (Temperatur und Luftfeuchtmessger) und Aktoren (Relais) angeschlossen werden. Zudem hat er Ausgänge für eine 3,3 V und 5V Spannungsversorgung.

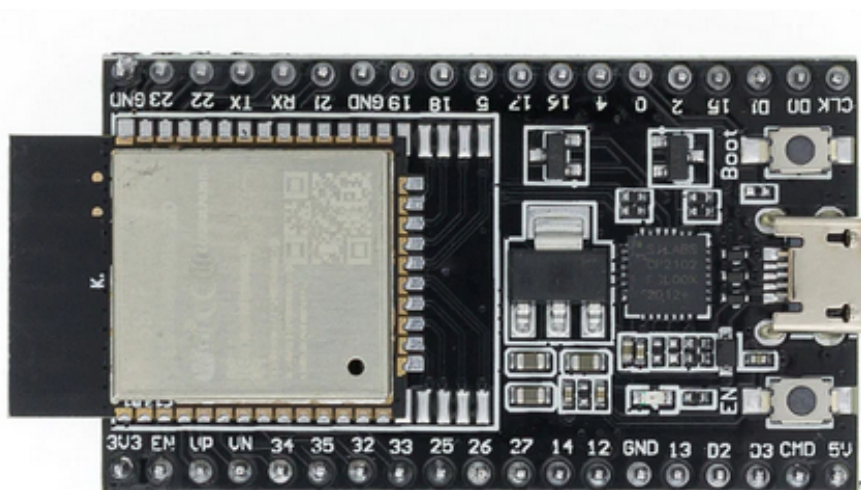
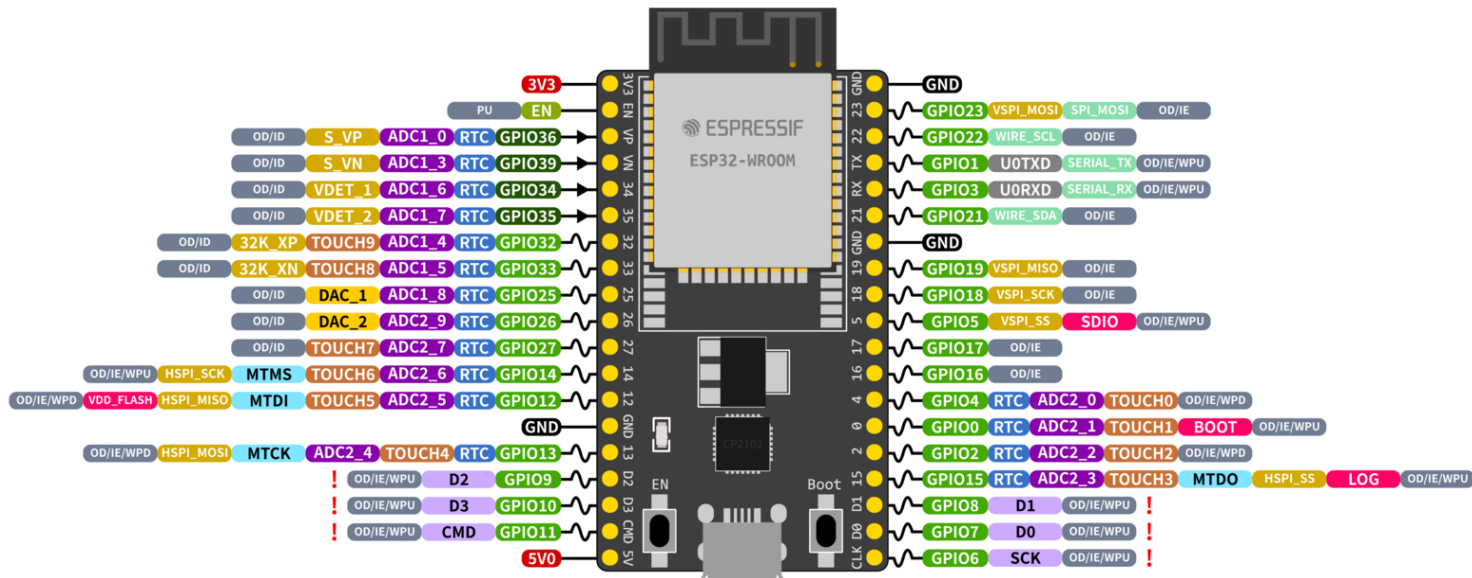


Abbildung 1; ESP 32 Mikrocontroller

Für diesen Aufbau haben wir uns aus kostengründen für das Modell das Modell "ESP 32 WROOM 32D" entschieden. Die Spannungsversorgung und Programmierung erfolgt mittels USB-C Kabel.



ESP32 Specs

32-bit Xtensa® dual-core @240MHz
 Wi-Fi IEEE 802.11 b/g/n 2.4GHz
 Bluetooth 4.2 BR/EDR and BLE
 520 KB SRAM (16 KB for cache)
 448 KB ROM
 34 GPIOs, 4x SPI, 3x UART, 2x I2C,
 2x I2S, RMT, LED PWM, 1 host SD/eMMC/SDIO,
 1 slave SDIO/SPI, TWAI®, 12-bit ADC, Ethernet

- PWM Capable Pin
- GPIOX GPIO Input Only
- GPIOX GPIO Input and Output
- DAC_X Digital-to-Analog Converter
- DEBUG JTAG for Debugging
- FLASH External Flash Memory (SPI)
- ADCX_CH Analog-to-Digital Converter
- TOUCHX Touch Sensor Input Channel
- OTHER Other Related Functions
- SERIAL Serial for Debug/Programming
- ARDUINO Arduino Related Functions
- STRAP Strapping Pin Functions

- RTC RTC Power Domain (VDD3P3_RTC)
- GND Ground
- PWD Power Rails (3V3 and 5V)
- Pin Shared with the Flash Memory
Can't be used as regular GPIO

GPIO STATE

- WPU:** Weak Pull-up (Internal)
- WPD:** Weak Pull-down (Internal)
- PU:** Pull-up (External)
- IE:** Input Enable (After Reset)
- ID:** Input Disabled (After Reset)
- OE:** Output Enable (After Reset)
- OD:** Output Disabled (After Reset)

Abbildung 2; Pinout ESP32-WROOM 32D

Sensor:

Der Sensor hat die Aufgabe, Temperatur und Luftfeuchtigkeit zu messen. Die ermittelten Werte werden vom Mikrocontroller aufgenommen.

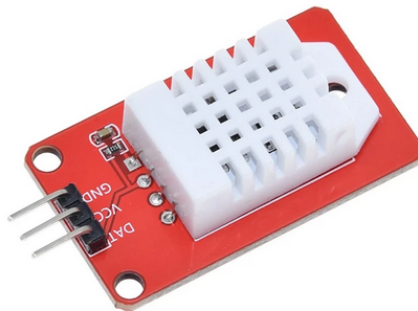


Abbildung 3; DHT22 Sensor

Wir haben uns für das Modell DHT22 entschieden. Dieser kann mit einer Versorgungsspannung von 3,3V bis 6V betrieben werden.

Das verwendete Modell besitzt bereits einen eingebauten Pull-Up-Widerstand.

Die Messgenauigkeit liegt bei $\pm 0.5^\circ\text{C}$ bzw. $\pm 2\%$ Luftfeuchtigkeit (max $\pm 5\%$). [Quelle 1]

Ventilator

Der 12V Ventilator dient als Lüftung für das Modell.

Zur Vereinfachung werden wir ihn mittels Relai nur Ein- oder Ausschalten eine Drehzahlregelung ist generell möglich, siehe hierzu unter dem Punkt "Regelung der Lüftergeschwindigkeit" bei den [Erweiterungen](#).



Abbildung 4; 12V DC Ventilator

Der Ventilator wird mit 12V Gleichspannung betrieben und besitzt die Maße 50 mm × 50 mm.

Er hat 3 Anschlüsse, diese sind Rot = Spannungsversorgung, Schwarz = Masse, Gelb = Drehzahl Tachometer.

In der Grundauführung des Modells wird der Drehzahl Tachometer nicht benötigt.

Relais:

Mit Relais kann die Spannungsversorgung eines Bauteils gezielt über den Mikrocontroller durch An- oder Abschalten gesteuert werden.



Abbildung 5; Relais Modul

Die verwendeten Relais Module verfügen über einen Kanal und werden mit einer Spannung von 5V gesteuert. Ob das Relai bei einem High- oder einem Low-Signal schalten soll, kann am Modul durch eine kleine schwarze Klammer eingestellt werden. Zudem kann der Zustand der Spannungsversorgung über die Anschlüsse "Normally Open" = NO oder Normally Closed = NC bestimmt werden.

Für das Projekt werden zwei Stück dieser Relais benötigt.

Bei der Verkabelung nutzen wir den NC Anschluss für die Spannungsversorgung der Bauteile.

Den Trigger-Clip stecken wir auf High-Trigger, da dies auch im Programmiercode dementsprechend eingerichtet wurde.

Heizmatte:

Die Heizmatte simuliert eine Heizung bzw. einen Heizkörper im Modell.



Abbildung 6; Heizmatte 12V DC

Die Matte wird mit 12V Gleichstrom betrieben, sie besitzt zwei Anschlüsse und kann sich auf bis zu 60°C aufheizen.

Netzteil:

Für die Spannungsversorgung des Ventilators und der Heizmatte wird ein Netzteil benötigt, welches 12V Gleichstrom liefert, da der Mikrocontroller eine solch hohe Spannung nicht liefern kann.



Abbildung 7; Netzteil 12V DC

Box:

Als Modell eines Gebäudes oder Raums kann ein Karton oder eine Kunststoffbox verwendet werden.



Abbildung 8; Karton als Gebäudemodell

In dem Karton wird die Heizmatte und der Sensor zur Temperatur und Luftfeuchtmessung angebracht. Zudem wird er mit einem passenden Loch für den Ventilator versehen und dieser darüber befestigt. Ein zusätzliches kleines Loch mit einem Durchmesser von ca. 2 cm dient zur Zuführung von Atemluft, diese simuliert eine Feuchtelast.

Kabel:

Dupont Kabel dienen zur Verdrahtung der gesamten Schaltung.



Abbildung 9; Dupont Kabel

Display:

Das Display dient zur Anzeige der gemessenen Werte des Sensors. Der Einbau des Displays ist optional, da die Messwerte auch auf dem seriellen Monitor unter der Software Arduino IDE angezeigt werden können.



Abbildung 10; SSD1306 Display

Für die Spannungsversorgung werden 5 V Gleichstrom benötigt. Die Kommunikation mit dem Mikrocontroller erfolgt über einen I2C-Bus, die dafür vorgesehenen Anschlüsse am ESP 32 Mikrocontroller sind Pin 21 für SDA (Serial Data) Verbindung sowie Pin 22 für den SCL bzw. SCK (Serial Clock) Anschluss für die Übertragung des Kommunikationstaktes.

Software

Als Entwicklungsumgebung nutzen wir die kostenfreie Software "Arduino IDE", diese ermöglicht eine einfache Programmierung von Arduino Boards.

Um ein ESP 32 Board nutzen zu können muss über die Boardverwaltung das ESP32 Add-on installiert werden. Anleitung siehe Quelle 2

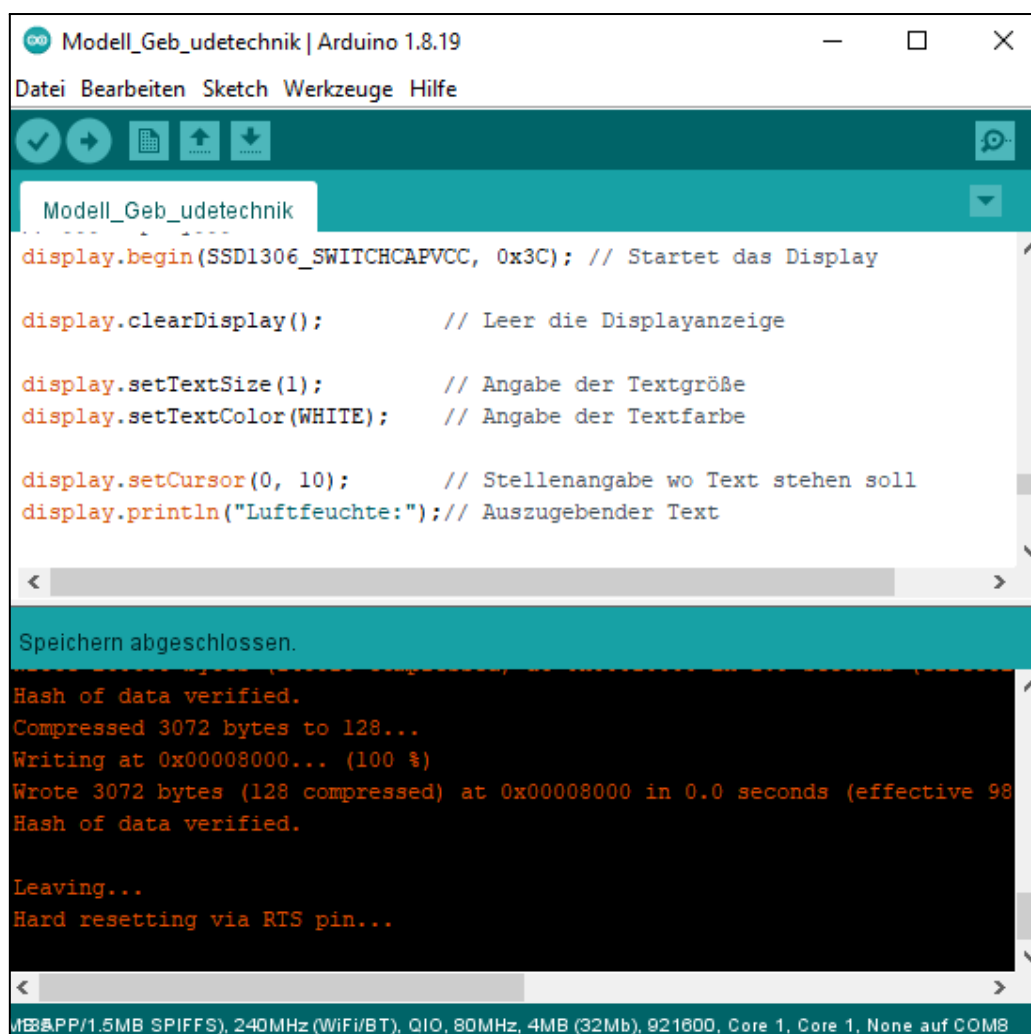


Abbildung 11; Arduino IDE; Beispielbild

Über die Software kann auch über die Werkzeuge der Serielle Monitor angezeigt werden. In ihm können dann die Messwerte des Sensors ohne den Einsatz eines Displays angezeigt werden.

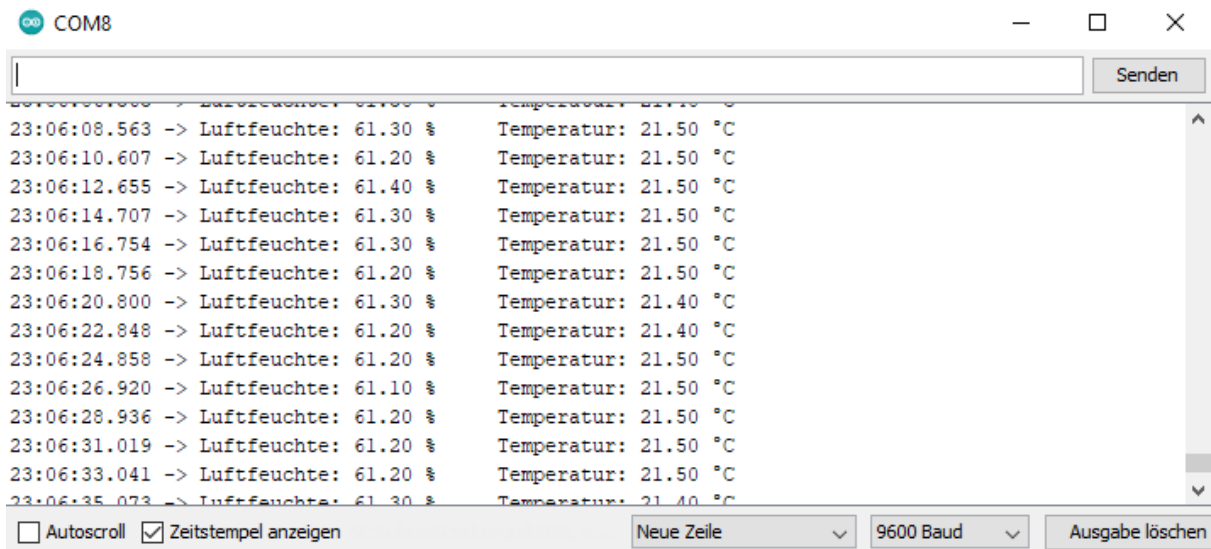


Abbildung 12; Serieller Monitor Arduino IDE

Bauanleitung

Vorbereitung

Zunächst muss die Box vorbereitet werden. In der oberen Hälfte muss ein kreisrundes Loch angebracht werden, welches etwas kleiner ist als der Ventilator. Anschließend kann der Ventilator darüber befestigt werden.

Heizmatte und Sensor werden im Inneren der Box angebracht. Der Sensor sollte möglichst im oberen Viertel der Box befestigt werden.

Anschließend wird die Box noch mit einer weiteren kleinen Öffnung versehen. Diese sollte an der Außenseite, welche dem Ventilator gegenüberliegt, vorgenommen werden.

Verkabelung

Im nächsten Schritt werden die Bauteile miteinander verkabelt, dies erfolgt wie in Abbildung 13 dargestellt.

Zur besseren Handhabung empfiehlt es sich, die Bauteile wie Relais und Kabelverbinder an einem Brett oder einem steifen Karton zu befestigen.

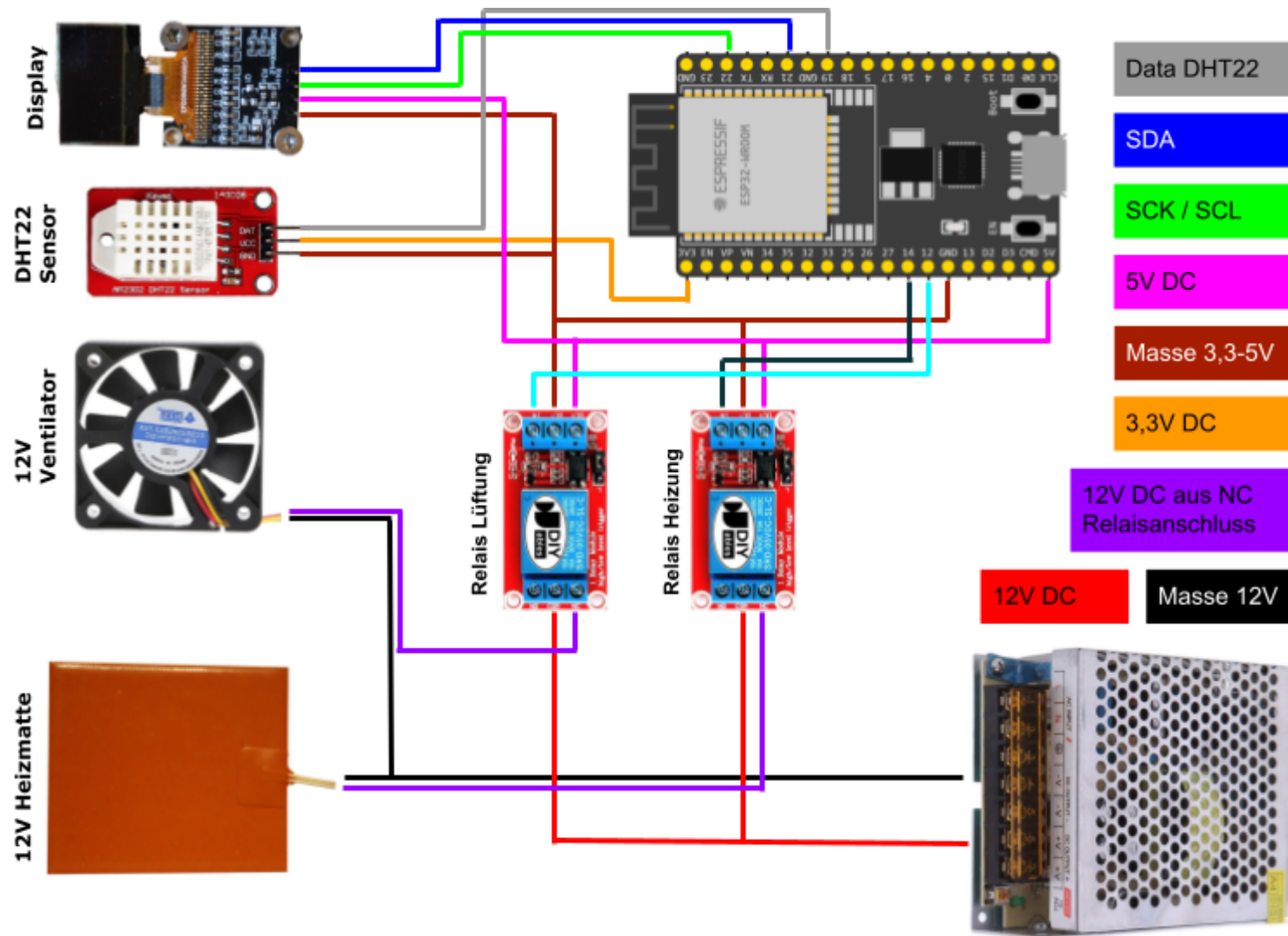


Abbildung 13, Verkabelungsplan

Code

Die Programmiersprache von Arduino ähnelt der von C+ und C++
Der Programmiercode setzt sich aus folgenden vier Bereichen zusammen.

Bibliotheken

In den Bibliotheken sind Unterprogramme enthalten, welche der Sensor und das Display für den Betrieb benötigen.

Definitionen

In den Definitionen werden Eigenschaften der Bauteile hinterlegt, so dass die Unterprogramme der Bibliotheken mit diesen Arbeiten können.
Zudem werden hier Daten-Pins des Arduino als solche definiert.

Void Setup

Unter Void Setup befinden sich Befehle, welche nur einmal installiert werden müssen. Dies sind beispielsweise die Festlegung der Baudrate, welche die Kommunikation der Bauteile festlegt sowie das Installieren von Pins des Mikrocontrollers als Ausgang

Void Loop

Der Void Loop startet eine Endlosschleife der darunter stehenden Befehle, wenn die Befehlskette bis zum Ende durchgeführt wurde, beginnt sie wieder von vorne.

In diesem Bereich befindet sich der eigentliche Code, welcher den Bauteilen vorschreibt, welche Aktionen bei bestimmten Ereignissen ausgeführt werden sollen.

Programmiercode

```
// #####Bibliotheken#####
// ###DHT22 Bibliothek###
#include "DHT.h"

// ###Display Bibliotheken###
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

// #####Definitionen#####
// ###DHT22###
#define DHTPIN 19
// Der Sensor wird an Datenpin 19 des Mikrocontrollers angeschlossen.

#define DHTTYPE DHT22 // Um welchen Sensor handelt es sich.

DHT dht(DHTPIN, DHTTYPE);
// Aus der DHT Bibliothek wird die Angabe des Pins und des Sensortyps
// unter der Variable "dht" installiert.

// ###Display###
#define SCREEN_WIDTH 128 // Definition der Display Breite in Pixel.
#define SCREEN_HEIGHT 64 // Definition der Display Höhe in Pixel.

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
// Erklärung für das SSD1306 Display über I2C Bus zu verbinden,
// I2C Pins sind die Pins SDA & SCL des Microcontrollers.

// #####void setup#####
void setup() {

// ###DHT22###
Serial.begin(9600); // Initialisiert die serielle Kommunikation
// mit der Kommunikationsgeschwindigkeit(Baud).

dht.begin(); // Startet die Messwertaufnahme des Sensors.

// ###Relais Lüfter###
pinMode(12, OUTPUT); // Pin 12 soll ein Ausgang sein, dadurch kann
// das Signal ein- oder ausgeschaltet werden

// ###Relais Heizung###
pinMode(14, OUTPUT); // Pin 14 soll auch ein Ausgang sein
}
```



```

        // #####void loop#####
void loop() {

delay(2000);
// Dies stoppt die Abarbeitung der Endlosschleife für 2000 Millisekunden.
// Das entlastet die Bauteile, da Beispielsweise der Sensor
// so nur alle 2 Sekunden eine Messung machen muss.

float h = dht.readHumidity();
//die Luftfeuchtigkeit auslesen und unter Luftfeuchtigkeit "h" speichern.

float t = dht.readTemperature();
//die Temperatur auslesen und unter Temperatur "t" speichern.

// ###Relais Lüfter###
if(h>63)digitalWrite(12, HIGH);
// Wenn die Luftfeuchtigkeit "h" größer als der eingegebene Wert ist, dann
// wird an Pin 12 (Relais Lüftung) ein Signal ausgegeben und das Relais
// schaltet die 12V DC Spannung durch. Der Lüfter springt an.

if(h<59)digitalWrite(12, LOW);
// Wenn die Luftfeuchtigkeit "h" kleiner als der Eingegebene Wert ist, dann
// wird kein Signal an das Relais geschickt und die 12V DC Spannung
// wieder unterbrochen. Der Lüfter geht aus.

// ###Relais Heizung###
if(t<22.5)digitalWrite(14, HIGH);
// Wie beim Relais Lüfter, jedoch wird hier eine Mindesttemperatur "t"
// eingegeben, bei welcher die Heizung angehen soll.

if(t>22.6)digitalWrite(14, LOW);
// hier wird die Maximaltemperatur "t" eingegeben, bei welcher die Heizung
// wieder abgeschaltet werden soll.

// ###Serieller Monitor###

Serial.print("Luftfeuchte: ");
Serial.print(h);
Serial.print(" %\t");          // "%" und ein Tab Abstand.
Serial.print("Temperatur: ");
Serial.print(t);
Serial.println(" °C ");

// Diese Befehle sorgen dafür, dass die Messwerte im seriellen Monitor
// angezeigt werden. z.B.: "Luftfeuchte: 53.80 % Temperatur: 21.70 °C".

```

```

// ###Display###
display.begin(SSD1306_SWITCHCAPVCC, 0x3C); // Startet das Display

display.clearDisplay(); // Leer die Displayanzeige

display.setTextSize(1); // Angabe der Textgröße
display.setTextColor(WHITE); // Angabe der Textfarbe

display.setCursor(0, 10); // Stellenangabe wo Text stehen soll
display.println("Luftfeuchte:");// Auszugebender Text

display.setCursor(75, 10);
display.println(h);

display.setCursor(110, 10);
display.println("%");

display.setCursor(0,45);
display.println("Temperatur:");

display.setCursor(75,45);
display.println(t);

display.setCursor(110,45);
display.println("\xF7"C");
// "\xF7" ist der Displaycode für das Darstellen von " ° " für °C.

display.display();
// Alle Eingabebefehle sollen auf dem Display angezeigt werden.
}

```

Inbetriebnahme

Nach erfolgreicher Verkabelung und Aufspielen des Programmiercodes auf den Mikrocontroller beginnt der Sensor mit den Messungen. Diese können im Seriellen Monitor und auf dem Display betrachtet werden. An den Relais sollten grüne LEDs aufleuchten, rote LEDs können aufleuchten, müssen dies aber nicht.

Da immer andere Temperaturen und Luftfeuchtigkeiten am Ort des Versuchs vorherrschen können, müssen im nächsten Schritt die Werte für Temperatur und Luftfeuchtigkeit, bei welchen die Relais geschaltet werden, neu programmiert werden.

Dies erfolgt durch Eingabe neuer Werte unter dem Punkt "Void Loop", genauer im Code der "if - Bedingungen" für Relais für Lüfter und Heizung. Hier können jeweils Werte für Minima und Maxima eingetragen werden.

Für die Luftfeuchtigkeit eignen sich folgende Werte:

Maxima	=	vorherrschende Luftfeuchtigkeit + 15
Minima	=	vorherrschende Luftfeuchtigkeit + 5

Für die Temperatur eignen sich folgende Werte:

Maxima	=	vorherrschende Temperatur + 1.2K
Minima	=	Maxima - 0.1K

Wurden diese Änderungen vorgenommen, kann beobachtet werden, ob die Heizmatte eingeschaltet ist. Sichtbar ist das am Relais, da die rote LED ausgeht, sobald der Stromfluss durchgeschaltet wird. Auf dem Display wird auch ein langsamer Anstieg der gemessenen Temperatur festzustellen sein.

Zum Schluss wird noch eine hohe Feuchtigkeitsbelastung der Luft in der Box simuliert. Dies geschieht über hineinpusten in die kleine Öffnung auf der Vorderseite. Die Veränderung der Luftfeuchtigkeit ist schnell auf dem Display sichtbar. Auch hier erlischt die rote LED am Relais, sobald dieses geschaltet wird und der Ventilator beginnt zu arbeiten.

Erweiterungen

Das Display ist bereits eine erste optionale Erweiterung, da dieses nicht zwingend notwendig ist. Die gemessenen Werte des DHT22 Sensors können auch im Seriellen Monitor in der Arduino IDE betrachtet werden.

Dämmung der Box

Eine Dämmung der Box, beispielsweise mit Styroporplatten oder zusätzlichen Kartonschichten dient als Nachstellung einer Dämmung der Gebäudehülle. Hierdurch werden Wärmeverluste durch die Außenflächen verringert.

Als Versuch kann bei verschlossener Ventilatoröffnung der Zeitraum verglichen werden, welchen es für einen bestimmten Temperaturabfall benötigt.

Somit kann der Nutzen einer besser gedämmten Gebäudehülle aufgezeigt werden.

Regelung der Lüftergeschwindigkeit

In der beschriebenen Basisversion des Projekts wird der Lüfter nur ein oder ausgeschaltet.

Der verwendete Lüfter mit 3 Anschlüssen besitzt einen Stromanschluss für variable Versorgungsspannung und lässt sich durch die Modulation von 12 Volt Impulsen in der Drehzahl steuern.

Zusätzlich besitzt er einen Anschluss für ein Tachosignal, an diesem kann die jeweilige Drehzahl ausgelesen werden. Für das Nutzen des Tachosignals muss eine Spannung auf den Anschluss gegeben werden. Bei jeder halben Umdrehung des Lüfters wird ein Puls aus dem "Open Drain" Anschluss gegeben, hierbei wird die Spannung dann für einen kurzen Moment auf Ground Niveau gezogen. Dadurch kann der Mikrocontroller die Drehzahl auslesen.

Für eine Pulsweitenmodulation zur Drehzahlsteuerung wird ein PNP Transistor benötigt, da nur die positive Spannungsschiene moduliert werden sollte. Bei einer Modulation der negativen Spannungsschiene geht das Ground Signal verloren und das Tachosignal kann nicht mehr ausgelesen werden.

Es muss beachtet werden, dass die Lüfterdrehzahl nicht zu gering gewählt wird, da der Ventilator sonst Anlaufschwierigkeiten bekommt. Dies kann durch einen Lüfter mit 4 Anschlüssen umgangen werden, da bei einem solchen für die PWM Modulation nicht die Spannung gesenkt wird. Die Drehzahl wird in einem solchen

durch einen eingebauten Drehzahlregler eigenständig an das eingehende PWM-Signal angepasst.

Durch die Aufnahme einer geregelten Lüfterdrehzahl entspricht das Modell mehr einer realen Lüftungsanlage. Zudem gibt diese Erweiterung einen Einblick in das Thema Pulsweitenmodulation.

Wärmetauscher

Die Erweiterung um einen Wärmetauscher ist wesentlich aufwändiger, jedoch lässt sich durch diese Erweiterung gut aufzeigen, wie viel Energie im Vergleich zum Betrieb ohne eingespart werden kann, da der Ventilator in der Grundversion des Aufbaus die erwärmte Luft aus der Box abträgt.

Mit Hilfe eines 3D-Druckers kann eine einfache Version eines Wärmetauschers kostengünstig gedruckt werden.

Kostenlose Druckvorlagen gibt es beispielsweise auf www.Thingiverse.com.

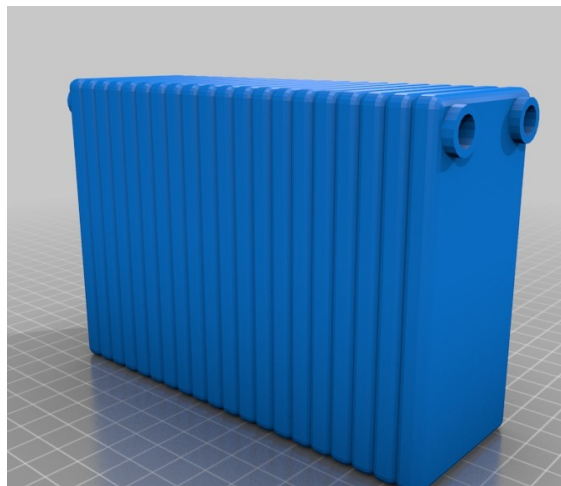


Abbildung 14; Wärmetauscher 3D-Druck

Um den Wärmetauscher nutzen zu können, werden Anschlusschläuche und ein zweiter Ventilator benötigt, so kann ein Ventilator die Abluft und der zweite die Zuluft bewegen.

Quellen

- [1] Produktinformationen DHT22; letzter Zugriff 03.01.2023
<https://www.az-delivery.de/products/dht22>
- [2] Installation ESP32 Add-on Arduino IDE; letzter Zugriff 03.01.2023
<https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/>

Abbildungsverzeichnis

- 1 ESP32-WROOM-32D; letzter Zugriff 02.01.2023
https://de.aliexpress.com/item/32959541446.html?spm=a2g0o.order_list.order_list_main.86.73c15c5ftnWxjS&gatewayAdapt=glo2deu
- 2 Pinout ESP32-WROOM-32D; letzter Zugriff 02.01.2023
https://docs.espressif.com/projects/esp-idf/en/latest/esp32/_images/esp32-devkitC-v4-pinout.png
- 3 Sensor DHT22; letzter Zugriff 02.01.2023
https://de.aliexpress.com/item/1005004651781612.html?spm=a2g0o.order_list.order_list_main.97.3e2f5c5f3SGRAe&gatewayAdapt=glo2deu
- 4 Ventilator 12V; letzter Zugriff 02.01.2023
https://de.aliexpress.com/item/4000848111886.html?spm=a2g0o.order_list.order_list_main.80.73c15c5ftnWxjS&gatewayAdapt=glo2deu
- 5 Relais; letzter Zugriff 02.01.2023
https://de.aliexpress.com/item/1005004417196058.html?spm=a2g0o.order_list.order_list_main.92.73c15c5ftnWxjS&gatewayAdapt=glo2deu

- 6 Heizmatte; letzter Zugriff 02.01.2023
https://de.aliexpress.com/item/4000806764010.html?spm=a2g0o.order_list.order_list_main.105.73c15c5ftnWxjS&gatewayAdapt=glo2deu
- 7 Netzteil; letzter Zugriff 03.01.2023
https://www.amazon.de/Docooler-Spannungswandler-Schalter-Spg-Versorgungsteil-Streifen/dp/B01LLFEMQ0/ref=pd_lpo_3?pd_rd_w=8CoUT&content-id=amzn1.sym.5079f52b-0953-4469-bf63-a91f67470055&pf_rd_p=5079f52b-0953-4469-bf63-a91f67470055&pf_rd_r=3WS3MRG2W85QHZKENAMQ&pd_rd_wq=841xZ&pd_rd_r=1a40b717-5539-4955-b9e2-f160644097c2&pd_rd_i=B01LLFEMQ0&pssc=1
- 8 Karton; Foto Ruchti 02.01.2023
- 9 Dupont-Kabel; letzter Zugriff 03.01.2023
https://de.aliexpress.com/item/1005004073424751.html?spm=a2g0o.order_list.order_list_main.98.73c15c5ftnWxjS&gatewayAdapt=glo2deu
- 10 Display SSD1306; letzter Zugriff 03.01.2023
https://de.aliexpress.com/item/32896971385.html?spm=a2g0o.order_list.order_list_main.112.73c15c5ftnWxjS&gatewayAdapt=glo2deu
- 11 Arduino IDE (Screenshot)
- 12 Serieller Monitor Arduino IDE (Screenshot)
- 13 Verkabelungsplan (Bildquellen selbe wie Bildquellen der Bauteile)
- 14 3D-Druckvorlage Wärmetauscher; letzter Zugriff 03.01.2023
<https://www.thingiverse.com/thing:1917929>