

3D-gedrucktes Lüftungsgerät
mit
Wärmerückgewinnung

Steuerung

Hardware

und

Software

Einleitung	2
Hardware	4
Verwendete Hardware:	4
Anleitung Hardware verkabeln.....	4
Arduino io expansion board	4
Pinbelegung:.....	4
pin assignment:	4
HD44780 Display	5
Pinbelegung.....	5
pin assignment:	5
I2C/TWI Controller	5
Pinbelegung.....	5
pin assignment:	5
Arduino LCD Library herunterladen.....	6
Ermitteln der I2C Adresse des angeschlossenen Displays	6
Noctua Lüfter NO NF-A6X25 PWM	7
Analog 5 button module.....	7
DHT11 Sensor	8
Software	8
Bibliotheken	8
Code	8
Haftungsausschluss	13

Einleitung

Beschrieben wird die Lüftersteuerung und Temperaturanzeige für eine Wohnraumlüftung mit Wärmerückgewinnung. Der Wärmetauscher, das Gehäuse, die Anschlussstutzen, die Kondensatsammler und die Sensorhalter werden mit einem 3D-Drucker hergestellt. Die Lüftersteuerung wird hardwareseitig mit einem Arduino Microcontroller realisiert der auf einem io expansion board gesockelt ist. An das io expansion board werden Sensoren, LC-Display, Tastatur, Lüfter und eine Stromversorgung angeschlossen.

Bedienung



Button 1:

Kurzer Tastendruck -> Lüfterdrehzahl wird um eine Lüfterstufe erhöht

Langer Tastendruck (> 3sec) -> Maximale Lüfterdrehzahl

Button 2:

Kurzer Tastendruck -> Lüfterdrehzahl wird um eine Lüfterstufe reduziert

Langer Tastendruck (> 3sec) -> Lüfter aus

Button 0:

Kurzer Tastendruck -> Zuluft Temperaturen/Luftfeuchte werden angezeigt

Bedeutung der Abkürzungen:

ZuEin -> Zuluft Einlass

ZuAus-> Zuluft Auslass

Button 3:

Kurzer Tastendruck -> Abluft Temperaturen/Luftfeuchte werden angezeigt

Bedeutung der Abkürzungen:

AbEin -> Abluft Einlass

AbAus -> Abluft Auslass

Button 4:

Kurzer Tastendruck -> Lüfterstufe und Raumtemperatur werden angezeigt

(Die Werte werden nicht automatisch aktualisiert)

Hardware

Verwendete Hardware:

- 1x Nano Mega328P V3.0 -- Preis ca. 12 EUR
- 4x DHT11 digitaler Feuchtigkeits- und Temperatursensor (ungenau aber günstig) -- Preis ca. 2 EUR
- Besser 4x DHT22, allerdings muss der Code angepasst werden
- 1x MQ-5 Sensor (H2, LPG, CH4, CO, Alkohol) (ungenau aber günstig) -- Preis ca. 2 EUR
- 1x Arduino Button Modul -- Preis ca. 3 EUR
- 1x LCD Display Modul 1602 HD44780 mit I2C Controller -- Preis ca. 3 EUR
- 1x Erweiterungs-Shield für Arduino Nano -- Preis ca. 3 EUR
- 2x PWM Lüfter 60x60mm 4 Pin -- Preis ca. 17 EUR
- Diverse Jumper Kabel, 1, 3 und 4 adrig
- 1x 12V Netzteil

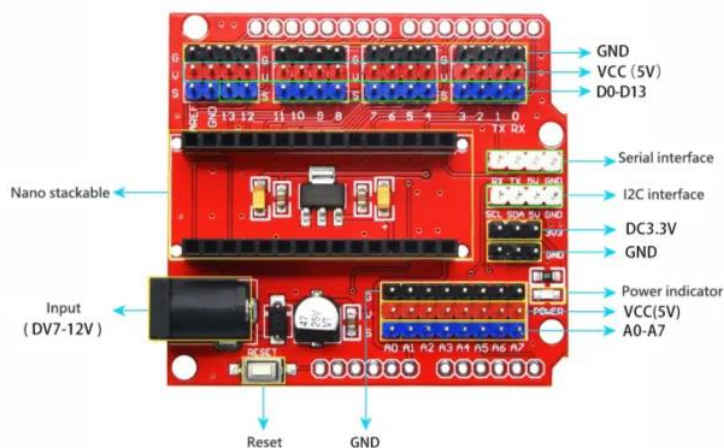
Anleitung Hardware verkabeln auf Youtube

<https://youtu.be/j1OchZ1aGrs>

Arduino io expansion board

Pinbelegung:

pin assignment:



HD44780 Display

Pinbelegung

pin assignment:

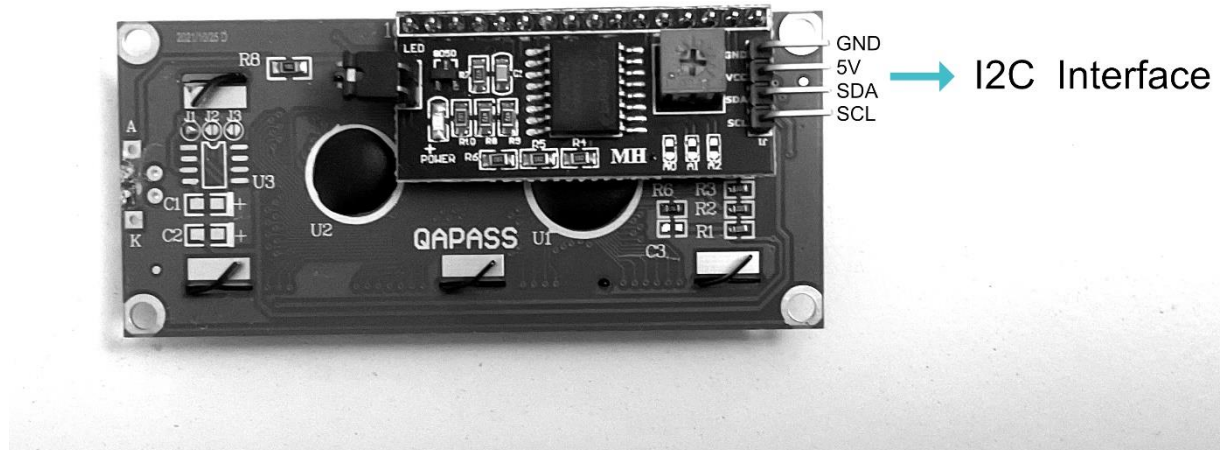
Das Display kann über 4 oder 8 Bit angesteuert werden. Die Belegung der 16 Kontakte sieht wie folgt aus:

1. VSS: GND (Masse)
2. VDD: +5V Versorgungsspannung
3. VO: Kontrastspannung, diese kann zwischen 0V und 5V liegen
4. RS: Register, Auswahl 0:Befehlsregister, 1: Datenregister
5. R/W: 1:Read/0:Write Kontakte
6. E: Enable
7. D0: Datenleitung (Nur 8 Bit Betrieb)
8. D1: Datenleitung (Nur 8 Bit Betrieb)
9. D2: Datenleitung (Nur 8 Bit Betrieb)
10. D3: Datenleitung (Nur 8 Bit Betrieb)
11. D4: Datenleitung (4 oder 8 Bit Betrieb)
12. D5: Datenleitung (4 oder 8 Bit Betrieb)
13. D6: Datenleitung (4 oder 8 Bit Betrieb)
14. D7: Datenleitung (4 oder 8 Bit Betrieb)
15. A: Anode der Hintergrundbeleuchtung
16. K: Kathode der Hintergrundbeleuchtung

I2C/TWI Controller

Pinbelegung

pin assignment:



Der I2C/TWI Controller verfügt neben dem selben Pinning, welches zum anlöten oder anschließen an das HD44780 Display dient, über 4 weitere Pins. Diese Pins sind für den Anschluss an das Endgerät, bspw. ein Arduino gedacht und ermöglichen einen Anschluss mit deutlich weniger Verkabelungsaufwand. Die Pins sind wie folgt belegt:

1. GND: Masse (GND)
2. VCC: +5V Versorgungsspannung
3. SDA: Datenleitung
4. SCL: Taktleitung

Die Pins können mit einem 4-adrigen Jumperkabel direkt im i2c Anschluss des io expansion boards eingesteckt werden weil die Pins auf beiden Boards die gleiche Reihenfolge haben.

Arduino LCD Library herunterladen

Arduino-LiquidCrystal-I2C-library-master

<https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library>

Library installieren

Statt lcd.begin sollte lcd.init verwendet werden weil sonst eine Fehlermeldung angezeigt wird.

Ermitteln der I2C Adresse des angeschlossenen Displays

Sketch herunterladen unter: <https://playground.arduino.cc/Main/I2cScanner/>

In Arduino einfügen, testen, auf arduino hochladen und serial monitor öffnen

Die I2C Adresse wird angezeigt: in meinem Fall **0x27**

Interiorterrier@gmail.com

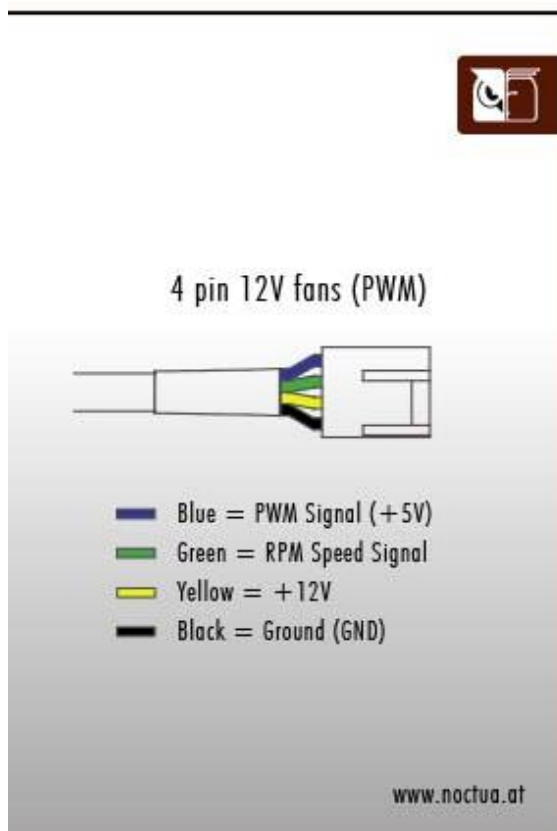
Die Adresse in folgender Zeile im Code einfügen:

```
// Set the LCD address to 0x27 for a 16 chars and 2 line display  
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

Folgende Bibliotheken werden für das Display benötigt:

```
#include <Wire.h>  
#include <LiquidCrystal_I2C.h>
```

Noctua Lüfter NO NF-A6X25 PWM



4 Pin Version:

Masse und +12V vom io expansion board abnehmen (anlöten)

PWM Signal an Pin 9 vom io expansion board

Analog 5 button module

Download Library under:

<https://www.codeproject.com/Tips/709109/ADKeyboard-Library-for-Arduino>

Interiorterrier@gmail.com

DHT11 Sensor

Erklärvideo auf Youtube:

https://www.youtube.com/watch?v=iTF6_w7AJ8

Download Library under:

<https://github.com/adidax/dht11>

Digital Pin 2 (DHT11_0) -> Zuluft Einlass

Digital Pin 3 (DHT11_1) -> Zuluft Auslass

Digital Pin 4 (DHT11_2) -> Abluft Einlass

Digital Pin 5 (DHT11_3) -> Abluft Auslass

Software

Bibliotheken

Folgende Bibliotheken müssen in der Arduino IDE manuell installiert werden.

LCD Bibliothek:

<https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library>

Keyboard Bibliothek:

<https://www.codeproject.com/Tips/709109/ADKeyboard-Library-for-Arduino>

DHT11 Bibliothek:

<https://github.com/adidax/dht11>

Code

```
#include <dht11.h>
```

```
// include the header file
```

```
#include <Arduino.h>
```

```
#include <AdKeyboard.h> // Keyboard Bibliothek
```

```
#include <Wire.h>
```

```
#include <LiquidCrystal_I2C.h> // LCD Bibliothek
```

```
#include <dht11.h> // Sensor Bibliothek
```

```
// Temp/Humidity Input Pin
```

```
#define DHT11PIN0 2
```


Interiorterrier@gmail.com

```
#define DHT11PIN1 3
#define DHT11PIN2 4
#define DHT11PIN3 5

dht11 DHT11_0;
dht11 DHT11_1;
dht11 DHT11_2;
dht11 DHT11_3;

// Set the LCD address to 0x27 for a 16 chars and 2 line display
LiquidCrystal_I2C lcd(0x27, 16, 2);

//create custom character Celsius Symbol
byte customChar[8] = {
    0b01100,
    0b10010,
    0b10010,
    0b01100,
    0b00000,
    0b00000,
    0b00000,
    0b00000
};

// instantiate it on Analog0 of the arduino board
AdKeyboard adKeyboard(0);

// PWM Variablen deklarieren
const int fan_exhaust_pin = 9; // Blue wire on fan(25 kHz)
int count = 0;
unsigned long start_time;
int rpm; // this is needed when you want to read the rpm signal of the pwm-fan
int pwm;
int fanspeed;

void setup() {

//Timing Register set to zero
TCCR1A = 0 ;
TCCR1B = 0;
TCNT1 = 0;

TCCR1A = _BV(COM1A1) | _BV(COM1B1) | _BV(WGM11);
TCCR1B = _BV(WGM13) | _BV(CS10);
ICR1 = 320;
pinMode (fan_exhaust_pin, OUTPUT);
OCR1A = 0;
OCR1B = 0;
Serial.begin(9600);
adKeyboard.setClickCallback(clickHandler);
```

Interiorterrier@gmail.com

```
adKeyboard.setPressCallback(pressHandler);
adKeyboard.setTwoPressCallback(twoPressHandler);

// initialize the LCD
lcd.init();
lcd.backlight();
lcd.createChar(0, customChar); // create a new custom character
lcd.clear();
lcd.print("Lueftung und");
lcd.setCursor(0,1);
lcd.print("Waermetauscher");
delay(4000);
}

void loop() {
  // put your main code here, to run repeatedly:
  adKeyboard.update();
}
void clickHandler(int key) { // Execute when button is pressed
once

  if ((key == 1) && (pwm <= 257)) { //if button 1 is pressed then
increment the fan speed 1/5
  pwm +=64;
  OCR1A = pwm;
  fanspeed = (pwm/64);
  lcd.clear();
  lcd.print("Luefterstufe: ");
  lcd.print(fanspeed);
}
  if ((key == 2) && (pwm >= 63)) { //if button 2 is pressed then
decrease the fan speed 1/5
  pwm -=64;
  OCR1A = pwm;
  fanspeed = (pwm/64);
  lcd.clear();
  lcd.print("Luefterstufe: ");
  lcd.print(fanspeed);
}
  if (key == 0) { //if button 0 is pressed then display supply air
values
  int chk = DHT11_0.read(DHT11PIN0);

  //Sensor 0
  lcd.clear();
  delay(500);
  lcd.print("ZuEin ");
  lcd.print((float)DHT11_0.temperature, 1);
  lcd.write((byte)0); // print the custom char at (2, 0)
  lcd.print("C");
```

Interiorterrier@gmail.com

```
//lcd.setCursor(0,1);
lcd.print(" ");
lcd.print((float)DHT11_0.humidity, 0);
lcd.print("%");

delay(500);

//Sensor 1

int chk1 = DHT11_1.read(DHT11PIN0);

lcd.setCursor(0,1);
lcd.print("ZuAus ");
lcd.print((float)DHT11_1.temperature, 1);
lcd.write((byte)0); // print the custom char at (2, 0)
lcd.print("C");

lcd.print(" ");
lcd.print((float)DHT11_1.humidity, 0);
lcd.print("%");

delay(5000);

}
if (key == 3) { //if button 3 is pressed then display exhaust air
values
//Sensor 2

int chk2 = DHT11_2.read(DHT11PIN0);

lcd.clear();
delay(500);
lcd.print("AbEin ");
lcd.print((float)DHT11_2.temperature, 1);
lcd.write((byte)0); // print the custom char at (2, 0)
lcd.print("C");

lcd.print(" ");
lcd.print((float)DHT11_2.humidity, 0);
lcd.print("%");

delay(500);

//Sensor 3

int chk3 = DHT11_3.read(DHT11PIN0);

lcd.setCursor(0,1);
```

Interiorterrier@gmail.com

```
lcd.print("AbAus ");
lcd.print((float)DHT11_3.temperature, 1);
lcd.write((byte)0); // print the custom char at (2, 0)
lcd.print("C");

lcd.print(" ");
lcd.print((float)DHT11_3.humidity, 0);
lcd.print("%");

delay(5000);
}

if (key == 4) { //if button 3 is pressed then display exhaust air
values
//Allgemeine Informationen anzeigen
lcd.clear();
lcd.print("Luefterstufe: ");
lcd.print(fanspeed);
lcd.setCursor(0,1);
lcd.print("tInnen: ");
int chk2 = DHT11_2.read(DHT11PIN0);
lcd.print((float)DHT11_2.temperature, 1);
lcd.write((byte)0); // print the custom char at (2, 0)
lcd.print("C");

}
}
void pressHandler(int key) { // Execute when button is pressed
longer

if ((key == 1) && (pwm <= 257)) { //if button 1 is pressed then drive
the fan with full speed
pwm = 320;
OCR1A = pwm;
fanspeed = (pwm/64);
lcd.clear();
lcd.print("Luefterstufe: ");
lcd.print(fanspeed);
}
if ((key == 2) && (pwm >= 63)) { //if button 2 is pressed then switch
off the fan
pwm =0;
OCR1A = pwm;
fanspeed = (pwm/64);
lcd.clear();
lcd.print("Luefterstufe: ");
lcd.print(fanspeed);
}
Serial.print("pressHandler: ");
```

Interiorterrier@gmail.com

```
    Serial.println(key);  
}  
void twoPressHandler(int k1, int k2) {  
    Serial.print("twoPressHandler: ");  
    Serial.print(k1);  
    Serial.print(" ");  
    Serial.println(k2);  
}
```

Haftungsausschluss

Der Nachbau des beschriebenen Gerätes erfolgt auf eigene Gefahr. Elektroarbeiten müssen von Fachkräften ausgeführt werden.