

**Fakultät Maschinenbau**

**Labor Informatik**

A2 - PulseChat

vorgelegt von:

Lukas Kalb, 70485979, [l.kalb@ostfalia.de](mailto:l.kalb@ostfalia.de)

Nelly Schrader, 70486749, [n.schrader@ostfalia.de](mailto:n.schrader@ostfalia.de)

Paul Lewetz, 70485694, [p.lewetz@ostfalia.de](mailto:p.lewetz@ostfalia.de)

Inhaltsverzeichnis

[Kurzbeschreibung der Lösung: 3](#_Toc162263593)

[Schaltplan: 4](#_Toc162263594)

[Quellcode: 5](#_Toc162263595)

[Arduino: 5](#_Toc162263596)

[Python: 7](#_Toc162263597)

# **Kurzbeschreibung der Lösung:**

Die Lösung besteht aus zwei Teilen: einem Arduino-Programm zur Erfassung von Sensordaten und deren Übertragung über WLAN sowie einem Python-Programm zur Speicherung der Daten, zur Interaktion mit dem ChatGPT-Chatbot über die OpenAI-Schnittstelle und zur Anzeige der Sensordaten.

Das Arduino-Programm umfasst die folgenden Schritte:

* Initialisierung des Pulssensors.
* Kontinuierliche Erfassung von Pulssensordaten.
* Übertragung der Sensordaten über WLAN an den Computer.

Das Python-Programm umfasst die folgenden Schritte:

* Einrichtung eines „Servers“, der die Sensordaten vom Arduino empfängt.
* Speicherung der empfangenen Sensordaten in einer Datei.
* Einbindung von ChatGPT über die OpenAI-Schnittstelle für die vorgefertigten Benutzerinteraktion.

Zusammen ermöglichen diese Programme die Erfassung von Sensordaten, die Interaktion mit einem Chatbot und die Speicherung der Daten für spätere Analyse oder Verwendung

Um die Effizienz und Kostenkontrolle zu optimieren, haben wir beschlossen, dem Benutzer vordefinierte Antwortmöglichkeiten anzubieten. Dadurch wird sichergestellt, dass nur „sinnvolle“ Anfragen zugelassen werden, was wiederum dazu beiträgt, die Menge der eingehenden Anfragen zu kontrollieren und die Kosten im Zusammenhang zu minimieren.

# **Schaltplan:**

Ein Bild, das Text, Schaltung, Elektronik, Elektrisches Bauelement enthält.

Automatisch generierte Beschreibung

Quelle: <https://microcontrollerslab.com/pulse-sensor-esp8266-nodemcu-tutorial/>

# **Quellcode:**

## Arduino:

#include <ESP8266WiFi.h>  
#include <WiFiUdp.h>  
#include <PulseSensorPlayground.h> // Includes the PulseSensorPlayground Library.   
  
const char\* ssid = "Schnelly";  
const char\* password = "essenkochen23";  
  
WiFiUDP Udp;  
  
const int PulseWire = *0*; // PulseSensor PURPLE WIRE connected to ANALOG PIN 0  
const int LED = LED\_BUILTIN; // The on-board Arduino LED, close to PIN 13.  
int Threshold = *550*; // Determine which Signal to "count as a beat" and which to ignore.  
  
PulseSensorPlayground pulseSensor;   
  
void setup() {  
 connectToWifi();  
 setupPulseSensor();  
 initSocket();  
}  
  
//Main Schleife, misst und sendet Daten  
void loop() {  
 if (pulseSensor.sawStartOfBeat()) {  
 int myBPM = pulseSensor.getBeatsPerMinute();  
 Serial.println(myBPM);  
 sendToPC(myBPM);  
 }  
 delay(*20*);   
}  
  
// -------------  
  
//Soll Verbindung zum Wlan herstellen  
void connectToWifi() {  
 WiFi.begin(ssid, password);  
 waitUntilConnectionIsAvailable();  
}  
  
//Überprüft ob die Verbindung aktuell gegeben ist  
void waitUntilConnectionIsAvailable() {  
 while (WiFi.status() != WL\_CONNECTED) {  
 delay(*500*);  
 }  
}  
  
void setupPulseSensor() {  
 pulseSensor.analogInput(PulseWire);   
 pulseSensor.blinkOnPulse(LED);  
 pulseSensor.setThreshold(Threshold);  
 pulseSensor.begin();  
}  
  
//UDP Begin muss aufgerufen werden, wird jedoch nicht benötigt  
void initSocket() {  
 Udp.begin(*42069*);  
}  
  
void sendToPC(int bpm) {  
 char bpm\_as\_char\_array[*4*];  
 itoa(bpm, bpm\_as\_char\_array, *10*); //Konvertiert int to char[]  
 Udp.beginPacket("192.168.137.1", *12000*); // Hard coded ip und port. Wir wissen die IP-Adresse, weil der Pc immer der Host ist  
 Udp.write(bpm\_as\_char\_array);  
 Udp.endPacket();  
}

## Python:

import socket  
from datetime import datetime  
from openai import OpenAI  
  
#UDP Server Socket  
server\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)  
server\_socket.bind(('', 12000)) #Port 12000 einfach ausgedacht  
  
memory = [] #Speicher der vom ESP empfangenen Messdaten  
  
client = OpenAI()  
messages = [{"role": "system", "content": "Du bist ein Kardiologe und bekommst Messwerte, die die Herzschläge pro Minute eines Patienten angeben. Dies sind Messwerte eines Sensors, die in etwa jede Sekunde neu generiert wurden."}] #Information für die Open AI, damit diese präziser antworten kann  
  
#Hier passiert die Interaktion mit ChatGPT. Input wird weitergegeben und gespeicherte Konversation wird aktualisiert. Gibt die Antwort von ChatGPT wieder.  
def chatbot(input):  
 if input:  
 messages.append({"role": "user", "content": input})  
 chat = client.chat.completions.create(  
 model="gpt-3.5-turbo-0125", messages=messages  
 )  
 reply = chat.choices[0].message.content  
 messages.append({"role": "assistant", "content": reply})  
 return reply  
  
#returnt die gewünschte Datei als String  
def dateioeffnen():  
 while True:  
 try:  
 dateiname = input("Wie heißt die gewünschte Datei?\n")  
 file = open("Messungen/" + dateiname + ".txt", "r")  
 break  
 except:  
 print("Fehler beim öffnen der Datei, probiere es nochmal")  
 datensatz = ""  
 for line in file:  
 datensatz = datensatz + line + " "  
 file.close()  
 return datensatz  
  
def neue\_daten\_ueber\_esp\_empfangen\_und\_speichern():  
 nutzdauer = float(input("Wie viele Sekunden soll gemessen werden?\nSinnvoll sind Zahlen zwischen 20 und 40\n"))  
 now = datetime.now().timestamp()  
 while (datetime.now().timestamp() - now) < nutzdauer: # Nutzer kann Dauer der Messung selber bestimmen  
 message, address = server\_socket.recvfrom(1024)  
 print(message)  
 memory.append(int(message.decode()))  
 print("Hier sind deine Messwerte:\n")  
 print(memory)  
 dateiname = input("Wie möchtest du die Datei nenen?\n")  
 file = open("Messungen/" + dateiname + ".txt", "w")  
 for i in memory:  
 file.write(str(i) + "\n")  
 file.close()  
  
print("Wie kann ich dir helfen?\n Fragen, die ich dir beantworten kann sind:")  
print("1. Neue Werte messen und speichern")  
print("2. Datensatz laden")  
while True:  
 try:  
 anfrage = int(input("Gib die gewünschte Zahl ein\n"))  
 except:  
 print("Bitte 1, oder 2 angeben")  
 continue  
 if anfrage > 0 and anfrage < 3: #Interaktion mit ChatGPT ohne Datensatz ist sinnlos, daher muss zuerst ein Datensatz erstellt/geladen werden  
 if anfrage == 1:  
 neue\_daten\_ueber\_esp\_empfangen\_und\_speichern()  
 elif anfrage == 2:  
 print(chatbot("Hier sind die Messwerte, bitte merke sie dir für später.\n" + dateioeffnen()))  
 break  
  
#Hauptschleife für das Menü  
while True:  
 print("Wie kann ich dir helfen?\n Fragen, die ich dir beantworten kann sind:") #Dient zur sinnvollen und eingeschränkten Kommunikation mit ChatGPT  
 print("1. Neue Werte messen und speichern")  
 print("2. Datensatz laden")  
 print("3. Mittelwert bestimmen")  
 print("4. Datensatz bereinigen")  
 print("5. Vergleich mit anderen Daten")  
 print("6. Max/Min BPM ausgeben")  
 print("7. Ratschlag von einem Arzt einholen")  
 print("8. ChatGPT direkt fragen")  
 print("9. Beenden")  
 try:  
 anfrage = int(input("Gib die gewünschte Zahl ein\n"))  
 except:  
 print("Es muss eine Zahl zwischen 1 und 9 sein")  
 continue  
 if anfrage > 0 and anfrage < 10: #Es können nur die angegebenen Modi ausgewählt werden  
 if anfrage == 1:  
 neue\_daten\_ueber\_esp\_empfangen\_und\_speichern()  
 elif anfrage == 2:  
 print(chatbot("Hier sind die Messwerte, bitte merke sie dir für später.\n" + dateioeffnen()))  
 elif anfrage == 3:  
 print(chatbot("Bitte bestimme den Mittelwert meines Datensatz."))  
 elif anfrage == 4:  
 print(chatbot("Bitte bereinige meinen Datensatz. Die Daten, die du erhalten hast wurden von einem Sensor gemessen, der Fehler aufweisen kann, weshalb es zu unrealistischen Daten kommen kann, die entfernt werden müssen. Bitte arbeite in Zukunft mit den bereinigten Daten weiter."))  
 elif anfrage == 5:  
 print(chatbot("Bitte vergleiche den Durchschnitt des aktuellen Datensatz mit dem durchscnitt der Daten, die ich dir jetzt mitgebe\n" + dateioeffnen()))  
 elif anfrage == 6:  
 print(chatbot("Suche bitte den maximalen und minimalen Wert aus dem Datensatz raus."))  
 elif anfrage == 7:  
 print(chatbot("Bitte analysiere den dir gegebenen Datensatz und gebe einen ärztlichen Ratschlag"))  
 elif anfrage == 8:  
 while True:  
 frage = input("Stell deine Frage.\nBitte beachte, dass maximal 100 Zeichen zulässig sind.\n")  
 if len(frage) <= 100:  
 print(chatbot(frage))  
 break  
 else:  
 print("Deine Frage ist " + str(len(frage)-100) + " Zeichen zu lang.\nBitte formuliere sie erneut.")  
 elif anfrage == 9:  
 exit()  
 else:  
 print("Es muss eine Zahl zwischen 1 und 9 sein")