Hier kommt das Labor-Deckblatt hin

Inhaltsverzeichnis

1 Aufgabenstellung

Das Ziel von dem Projekt war es, eine intelligente Lüftersteuerung mit Web-Application zu entwerfen. Ein Temperatursensor liefert die Temperaturen an ein ESP-Modul, welches die Daten per WLAN weiter sendet an einen RaspberryPi. Der RaspberryPi dient als Server und verwaltet die Daten. Mit Hilfe einer Web-Application werden die gemessenen Werte in einem Diagramm ausgegeben. Werden Temperaturen über einer gewissen Schwelle gemessen, schaltet sich der Lüfter ein. Hat sich die Temperatur normalisiert wird der Lüfter wieder abgeschaltet.

1.1 Realisierung

Das ESP-Modul dient als Client und der RaspberryPi als Broker. Damit das ESP-Modul Daten senden kann muss es eine stabile Verbindung zu einem vorhandenen WLAN-Netzwerk haben. Falls die Verbindung zum WLAN oder zum Broker abbricht schalten sich zwei Warn-LEDs ein. Eine dritte LED geht an, falls es zu heiß wird und der Lüfter eingeschaltet werden muss. Zur sicheren Übertragung der Werte, wird das MQTT-Protokoll (Message Queuing Telemetry Transport) verwendet. Der Pi bekommt die Daten und wertet diese mit Hilfe von Python aus. **RAPHIS PART BILD CLIENT WLAN BROKER DATENBANK**

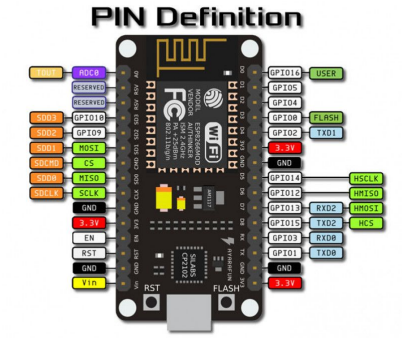
2 ESP-Modul

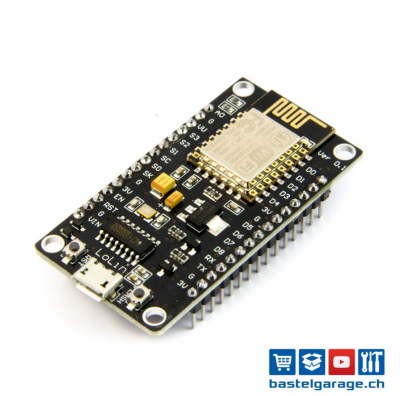
2.1 Übersicht

Modul mit WLAN verbinden, Daten senden und empfangen mit Arduino IDE kompatibel, Warum ESP verwendet,

2.2 Technische Daten

Zur Realisierung wurde das Entwicklerboard *NodeMCU Lua Lolin V3 Module ESP8266* verwendet. Es besitzt ein WLAN-Funkmodul mit einer integrierten 25dBm Antenne. Programmiert wird es über eine Micro-USB Schnittstelle und ist mit der Arduino IDE kompatibel. Die Versorgung des Moduls funktioniert über die Schnittstelle sowie mit einem externen Netzgerät oder einer Batterie. Obwohl das Entwicklerboard im Normalfall 3V3 kompatibel ist, kann es auch mit 5V versorgt werden da es einen internen Spannungswandler besitzt. **https://www.bastelgarage.ch/esp8266-nodemcu-v3-kompatibles-development-board**





2.3 (Verbindung mit) Temperatursensor

serieller Monitor

3 Konfigurationen

3.1 ESP8266 als Client

3.2 Virtual Machine als Broker

3.2.1 MQTT

Publishen und Subcriben

3.3 Verbindungsaufbau

Einfaches senden und empfangen auf terminal

4 Erweiterungen

4.1 Warn-LEDs

Einfaches Blockschal bild/ selbst gezeichnetes auf zettel, noch nicht in altium

4.2 Lüfter

4.3 Display

4.4 Sleep Mode

4.5 Aufbau auf Lochrasterplatte

Altium Schaltung

5 Raspberry Pi

RAPHIS PART

5.1 Übersicht

5.2 Technische Daten

6 Pi als Broker

6.1 MQTT

6.2 Daten empfangen

7 WebOberfläche

7.1 Daten in Diagramm ausgeben

8 ESP2 Daten empfangen

8.1 Ausgabe auf Display

Temp-Sensor drausen sendet Daten zu Pi, der sendet Daten weiter zu zweitem ESP-Modul mit Display wo man Temperatur ablesen kann(Wenn Pi in anderem Raum wie bennötigtes Display)