Arduino RoomControlling

Inhaltsverzeichnis

[1.0 Auftrag 1](#_Toc536712267)

[2.0 Planung 1](#_Toc536712268)

[3.0 Hardware 2](#_Toc536712269)

[3.1 Microcontroller 2](#_Toc536712270)

[3.2 PCB 2](#_Toc536712271)

[3.3 HYT 939 Temperatur und Feuchtesensor 2](#_Toc536712272)

[3.3.1 Technische Daten 2](#_Toc536712273)

[4.0 Software 3](#_Toc536712274)

[4.1 Entwicklungsugebung 3](#_Toc536712275)

[5.0 Web Development 3](#_Toc536712276)

[5.1 API 4](#_Toc536712277)

[5.1.1 Einführung 4](#_Toc536712278)

[5.1.2 Anbieter 4](#_Toc536712279)

[5.2 Datenbank 4](#_Toc536712280)

[5.2.1 Einführung 4](#_Toc536712281)

[5.2.3 Anbieter 4](#_Toc536712282)

[5.2.4 Struktur 4](#_Toc536712283)

[5.3 Grafana 5](#_Toc536712284)

[5.3.1 Einführung 5](#_Toc536712285)

[5.3.2 Anbieter 5](#_Toc536712286)

[5.3.3 Übersicht 6](#_Toc536712287)

[5.3.4 Dashboard 6](#_Toc536712288)

# 1.0 Auftrag

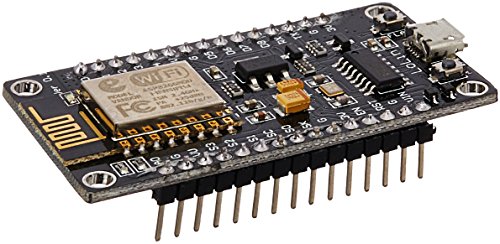
Im Fach Informatik muss ein Arduino Project erstellt werden.

# 2.0 Planung

# 3.0 Hardware

## 3.1 Microcontroller

Als Microcontroller verwende ich einen NodeMCU. Dieser hat direkt ein WLAN Modul integriert und ist sehr kostengünstig.



## 3.2 PCB

Das PCB habe ich im Circuit Studios erstellt. Mit dem PCB wir der Feuchte Sensor mit dem NodeMCU verbinden.

Gefertigt habe ich das PCB bei [www.jlcpcb.com](http://www.jlcpcb.com). Für 2$ habe ich 10 PCBs bekommen, dadurch kann ich das System beliebig erweitern.

## 3.3 HYT 939 Temperatur und Feuchtesensor

Die wichtigste Aufgabe bei der Messung ist der stabile Nachweis eines korrekten Feuchtewerts und Temperaturwerts innerhalb einer angemessenen Ansprechzeit. Für diesen zweck verwende ich ein HYT 939 von der Firma IST AG da dieser sehr genau ist und direkt auch die Temperatur misst.

Der Sensor wir über I2C ausgelesen.

Der Sensor ist in einem TO-46 Gehäuse verbaut.

### 3.3.1 Technische Daten



# 4.0 Software

## 4.1 Entwicklungsugebung

Für die Programmierung des Arduino Codes habe ich nicht die normale Arduino Entwicklungsumgebung verwendet, sondern Atom.io.

<https://atom.io/>

Atom.io nennt sich “A hackable text editor for the 21st Century”. Der Editor ist für sehr viele sprachen geeignet und kann mit diverser Erweiterungen erweitert werden.

Für das Programmieren vom NodeMCU wird die Erweiterung PlattforIO verwendet. Diese stellt alles Wichtige für das Programmieren des Arduino zur Verfügung. Unter anderem einen besseren Compiler, Serial monitor und eine Abdeckung von 588 Bards.

<https://platformio.org/>

# 5.0 Web Development

## 5.1 API

### 5.1.1 Einführung

Die API erlaubt die Kommunikation zwischen der Datenbank und dem NodeMCU. Hierzu dient ein PHP Skript welches mit der Datenbank verbunden ist. Die Temperatur und Feuchtigkeit wir mit der Room ID per PHP GET übergeben. Das script macht dan anschlissend den insert in ie Datenbank.

INSERT INTO environment(room,temperature,humidity) VALUES ('$room','$temperature','$humidity')

### 5.1.2 Anbieter

Die API läuft mit dem PHP Scripts auf meinem Webspace der bei Nitrado gemietet wurde. Diese bieten diverse Grössen des Webspaces an.

## 5.2 Datenbank

### 5.2.1 Einführung

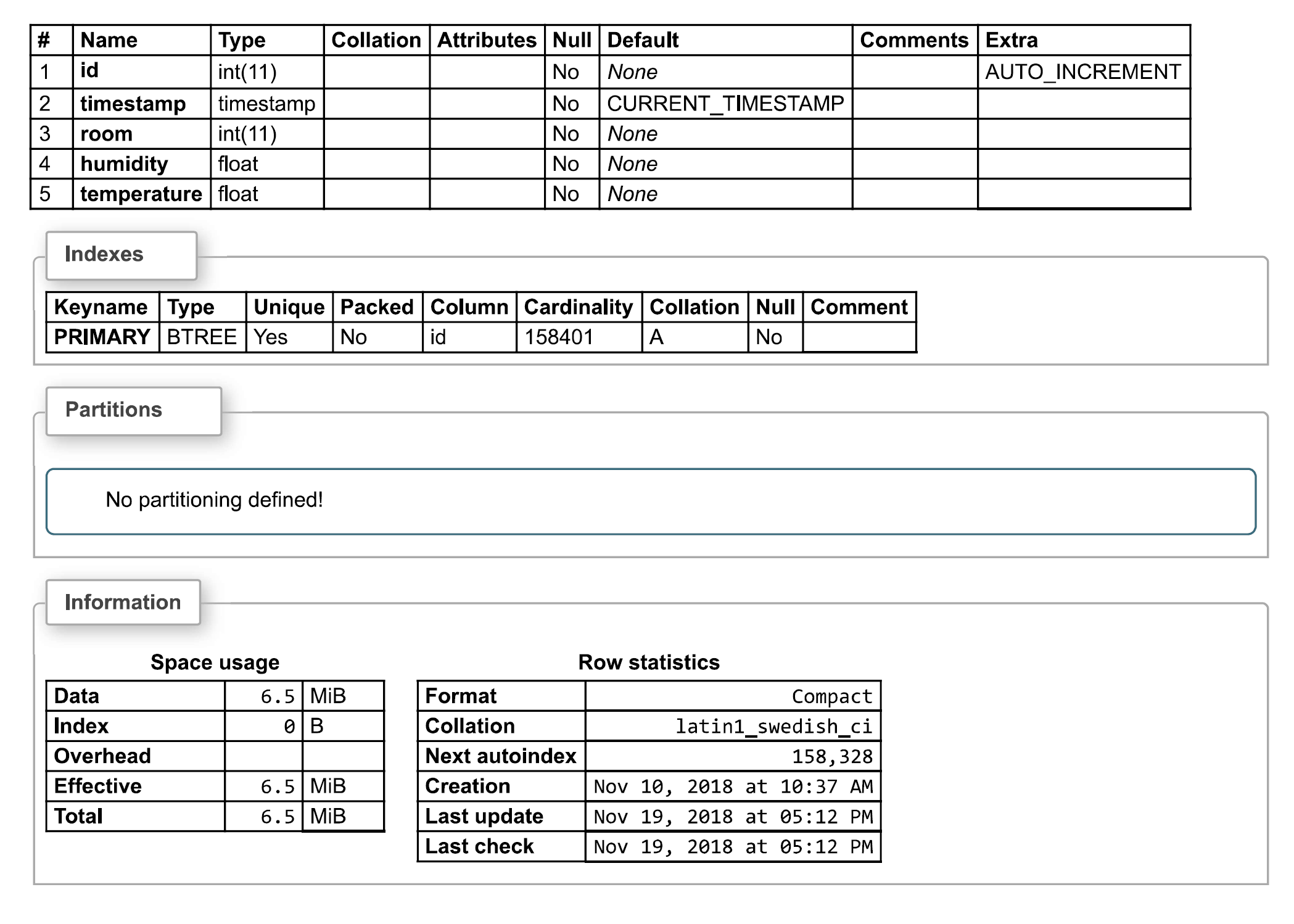
Als Datenbank verwende ich MySQL, als webinterface PHPmyAdmin. Die Datenbank ist dazu da um alle Werte abzuspeichern. Es fallen sehr schnell sehr viele werte an, dadurch ist eine Datenbank die Optimale Lösung für das abspeichern.

### 5.2.3 Anbieter

Die Datenbank lläuft genau wie der webspace bei Nitrado.

### 5.2.4 Struktur

Die Datenbank weist folgende Struktur auf:



## 5.3 Grafana

### 5.3.1 Einführung

Grafana ist ein Open source Monitoring und Analyse System. Es bieted diverse Darstellungsmöglichkeiten und datenquellen an.

### 5.3.2 Anbieter

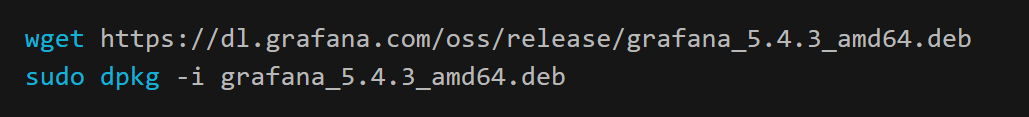
Grafana wird auf einem VPS gehostet bei Hetzner. Der Server verfügt über folgende eigenschaften:

1 vCPU  
2GB RAM  
20 GB HDD

Insgesamt ist 20TB Traffic im Monat inbegriffen, was für meine Zwecke niemals erreicht werden kann.

Auf dem Server ist Ubuntu 18.04 installiert.

Grafana wird über die Konsole instaliert, heirfür braucht man folgende Befehle:



### 5.3.3 Übersicht

### 5.3.4 Dashboard