

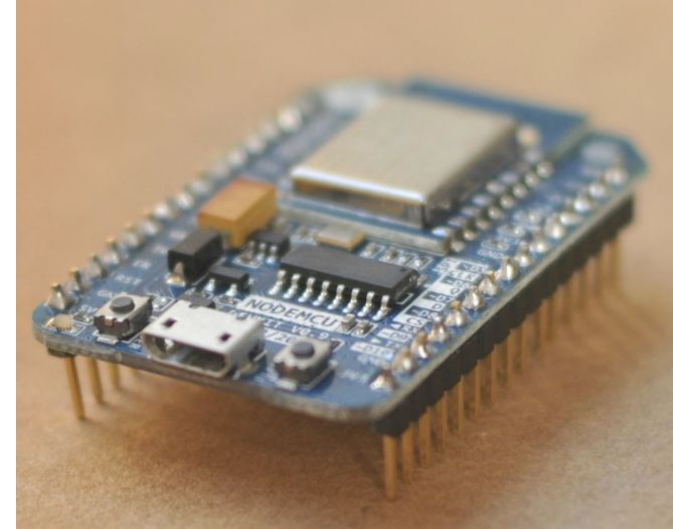
# SmHoEzMa

Tobias Rieß  
Martin Lesser  
Karl-Erik Kley

- Was ist SmHoEzMa?
- NodeMCU
  - Schaltplan & Aufgabe
- Backend/API
  - Docker
  - API NodeMCU
  - API Phillips Hue
  - API Android App
- Android App

- Smarthome Geräte verwalten
- Geräte einstellen und Daten lesen
- Homecoming Funktion

- freies Betriebssystem für ESP8266
- Betriebssystem zum Aufbau von WLAN-basierten Netzwerken
- basiert auf Arduino Hardware
- Programmierung mittels C
- kostengünstig
- sendet periodisch Sensorwerte an die API



**FH** **FACHHOCHSCHULE**  
**ERFURT** UNIVERSITY  
OF APPLIED SCIENCES  
Angewandte  
Informatik

- 
- The diagram shows a NodeMCU V1 ESP12E microcontroller connected to a 74HC4051PW-Q100,11 8-to-1 multiplexer. The NodeMCU's pins are labeled on the left, and the multiplexer's pins are labeled on the right. The circuit includes a 3.3V power supply, ground connections, and two sensors: a thermistor (R1) and a photoresistor (U3). The multiplexer's output is connected to the NodeMCU's A0 pin.
- | NodeMCU Pin | NodeMCU Label | 74HC4051PW-Q100,11 Pin | 74HC4051PW-Q100,11 Label |
|-------------|---------------|------------------------|--------------------------|
| 1           | 3.3V          | 16                     | VCC                      |
| 2           | GND           | 15                     | Y2                       |
| 3           | TX            | 14                     | Y1                       |
| 4           | RX            | 13                     | Y0                       |
| 5           | D8            | 12                     | Y3                       |
| 6           | D7            | 11                     | Y4                       |
| 7           | D6            | 10                     | Y5                       |
| 8           | D5            | 9                      | Y6                       |
| 9           | GND           | 8                      | Y7                       |
| 10          | 3.3V          | 7                      | Y8                       |
| 11          | D4            | 6                      | Y9                       |
| 12          | D3            | 5                      | Y10                      |
| 13          | D2            | 4                      | Y11                      |
| 14          | D1            | 3                      | Y12                      |
| 15          | D0            | 2                      | Y13                      |
- Additional components and connections include:
- R2 (10K)**: Resistor connected between the multiplexer's output and the NodeMCU's A0 pin.
  - R3 (10K)**: Resistor connected between the multiplexer's output and the NodeMCU's A0 pin.
  - R1 (Thermistor)**: Sensor connected to the multiplexer's input.
  - U3 (Photoresistor-5528)**: Sensor connected to the multiplexer's input.

- Erstellung von 3 APIs (NodeMCU, Philips Hue Bridge, App)
- mittels Python umgesetzt
- kümmert sich um alle eintreffenden Anfragen
- wird in Docker ausgeliefert
- Docker:
  - light-weight Container-basierte “VM” für einzelnen Dienst

- über JSON Format ausgeliefert
- NodeMCU
  - verarbeitet Werte von Microcontroller
- Phillips
  - steuert und fragt Hue Bridge an
- SmHoEzMa-APP
  - liefert Werte an App

- Weiterhin Nutzung von Qt
- umsteigen von C++ auf QML für besseren Android Look
  - von Qt entwickelte deklarative Programmiersprache für GUI entwicklung
  - Json ähnlicher Aufbau
  - Funktionen werden in Javascript geschrieben





# Danke für Ihre Aufmerksamkeit



Fork us on git