

Technik mobiler Systeme

Daten von Sensoren ermitteln und mittels ESP8266 Mikrokontroller an Raspberry Pi senden und auswerten.

05.01.2018

Gires Ntchouayang, Sylwester Korpik,
Nikiema Wendpanga Mohamed

Inhalt

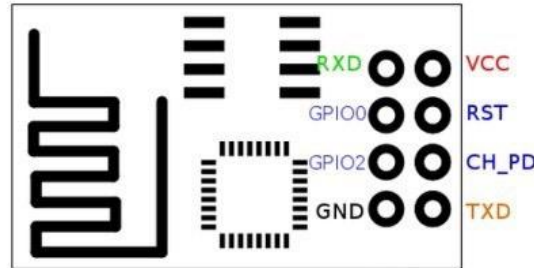
- . Problemstellung
- . ESP8266 01
- . DHT11
- . HC-SR04
- . Verkabelung und Programmierung
- . Visualisierung
- . Demo

Problemstellung

- Sensoren bieten Daten an
- Diese Daten wollen wir mit dem ESP auslesen,
- an einen MQTT-Broker weiterleiten
- Und im Browser live Visualisieren

ESp8266 01

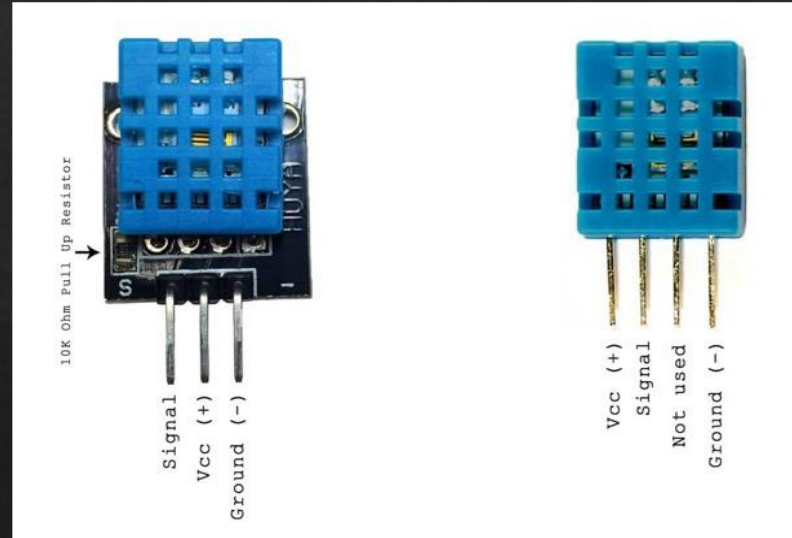
- Programmierbares WLAN-Funkmodul



Label	Signal
VCC	3.3V (3.6V max) supply voltage
GND	Ground
TXD	Transmit Data (3.3V level)
RXD	Receive Data (3.3V level!)
CH_PD	Chip Power down: (LOW = power down active)
GPIO0	General Purpose I / O 0
GPIO2	General Purpose I / O 2
RST	Reset (reset = LOW active)

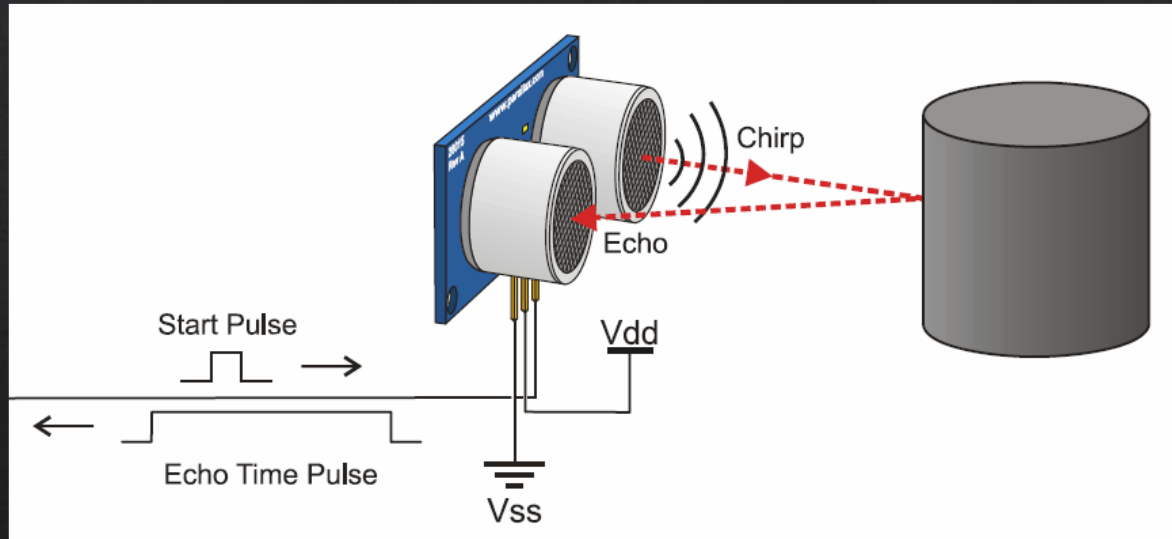
DHT11

- Sensor zum Messen von Temperatur und Luftfeuchte
- Temperatur im Bereich von 0-50° C ($\pm 2^\circ$ C)
- Relative Luftfeuchtigkeit von 20 bis 95% ($\pm 5\%$)



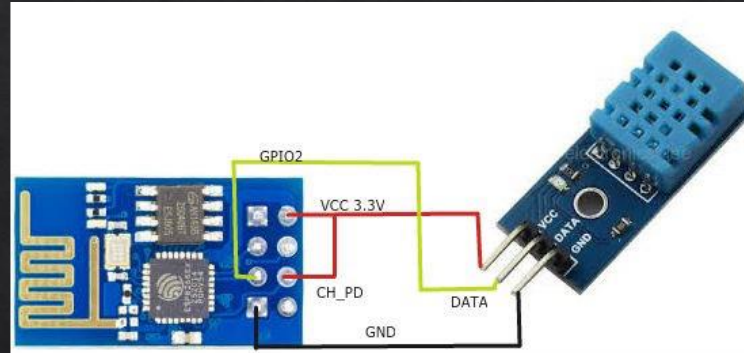
HC-SR04

- Ultraschallsensor HC-SR04 zur Entfernungsmessung bzw. Abstandsbestimmung zwischen 2cm und ca. 3m



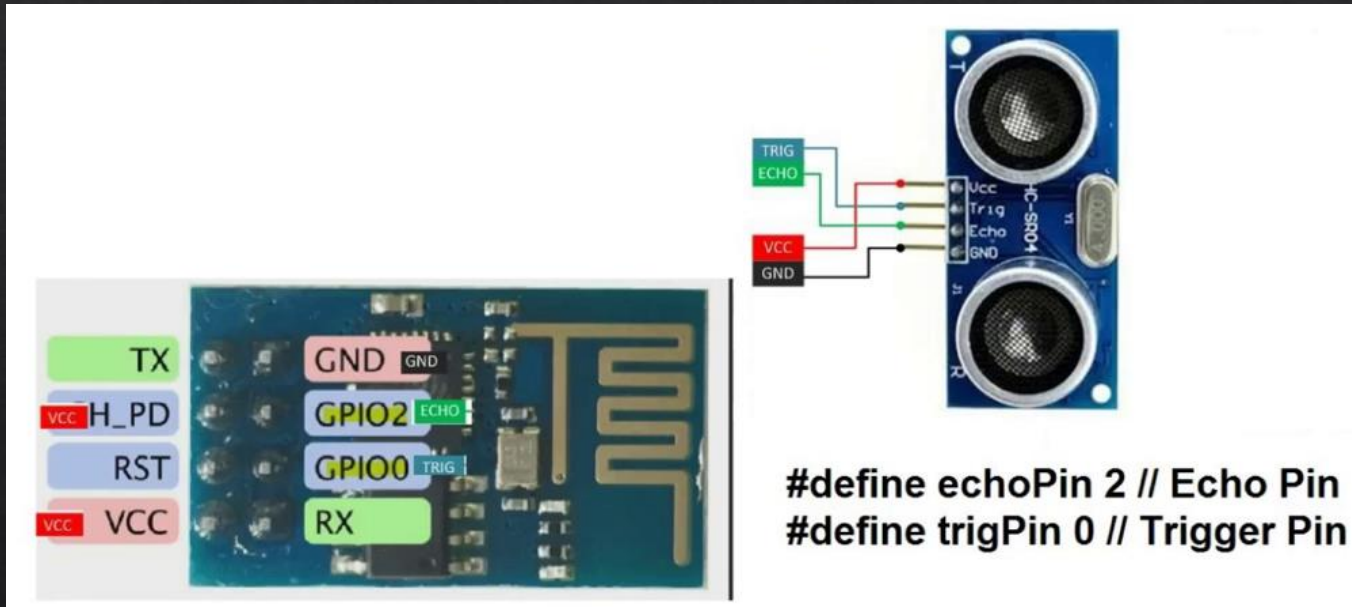
Verkabelung und Programmierung

- ESP8266 01 Verkabelung mit dem DHT Sensor



Verkabelung und Programmierung

- ESP8266 01 Verkabelung mit dem HC-SR04 Sensor



Verkabelung und Programmierung

- Programmierung mit Arduino IDE
- Einstellung für Zusätzliche Boardverwalter
- http://arduino.esp8266.com/stable/package_esp8266com_index.json
- Im Bordverwalter das ESP Packet installieren
- ESP8266 Generic als Board auswählen
- Zusätzliche Bibliotheken anbinden (DHT11)

Verkabelung und Programmierung

```
ultraschall | Arduino 1.8.5

#include <ESP8266WiFi.h>
#include <PubSubClient.h>

#define echoPin 2 // Echo Pin
#define trigPin 0 // Trigger Pin

long duration, distance; // Duration used to calculate distance

//const char* ssid = "XenYou";
//const char* password = "Nngpedson1";
const char* ssid = "DarkMachine";
const char* password = "pedsonpro";

const char* mqtt_server = "192.168.1.31";

WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char buf[50];

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(115200);
  setup_wifi();
}
```

```
dht | Arduino 1.8.5

#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <DHT.h>
//include <DHT11.h>

#define dhtPin 2
#define type DHT11

DHT dht(dhtPin, type);

long duration, distance;

const char* ssid = "DarkMachine";
const char* password = "pedsonpro";
//const char* ssid = "XenYou";
//const char* password = "Nngpedson1";
const char* mqtt_server = "192.168.1.31";

WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char buf1[50];
char buf2[50];

void setup() {
```

Visualisierung

- Node.js Webserver
- Node.js-Module
 - . Express Web Framework
 - . Vis-Bibliothek für die Visualisierung
 - . MQTT für die MQTT-Verbindung
 - . Socket.io für die bidirektionale Verbindung
 - . Body-parser

Visualisierung

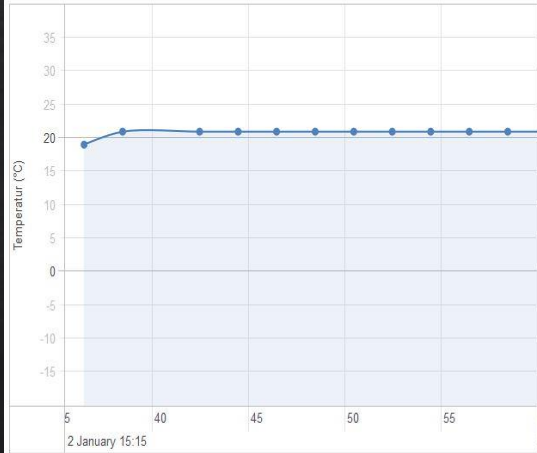
ESP8266 | Sensordaten mit Mqtt visualisieren

Sensorentyp: DHT11, HCSR04

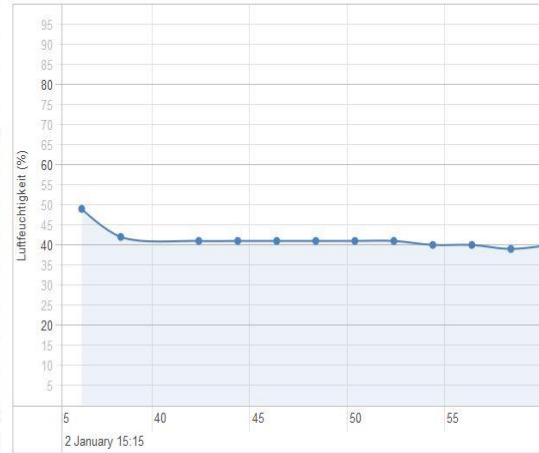
Mqtt topics: sensor/humidity, sensor/distance, sensor/temperature.

Anzeigetyp:

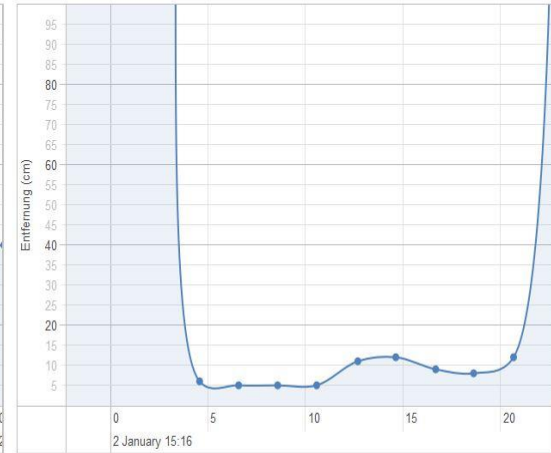
Temperatur: 21 °C



Luftfeuchtigkeit: 40%



Entfernung: 279 cm



DEMO