



Internet of Things im Team

// Spartakiade 2017 //////////////////////////////////////

@andrelanger

@oliverlorenz



Herzlich willkommen

Wer sind wir?

André



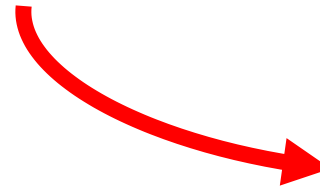
TECHNISCHE UNIVERSITÄT
CHEMNITZ



Oliver



Cybus



Kick-Off



Was haben wir heute vor?

- Kick-Off
- Einführung ins Thema IoT
- Microcontroller + Sensoren
- Development
- Showcase
- Retrospektive



Kurze Vorstellungsrunde

1. Was ist dein Nickname?
2. Was kannst du als Entwickler besonders gut?
3. Was sind deine Erwartungen an den Workshop?



Arbeitsmaterial

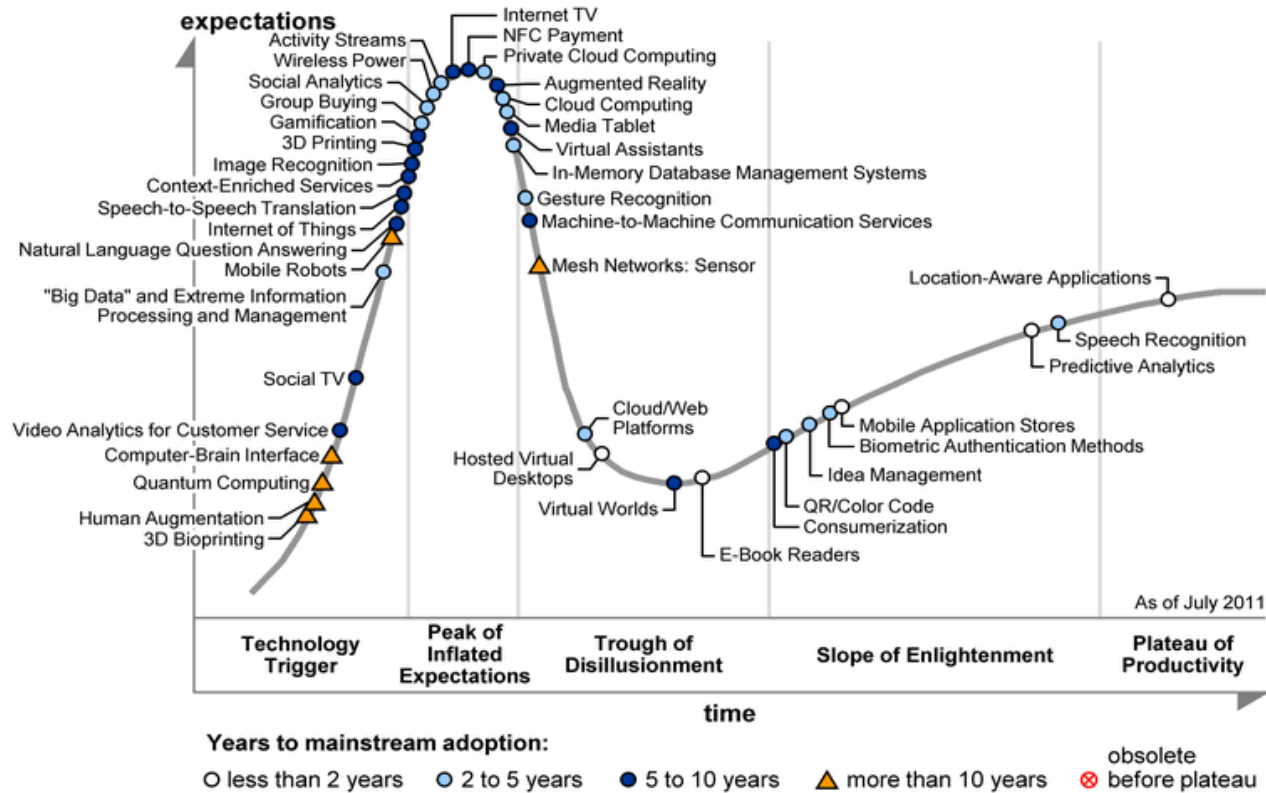
- Virtuelle Maschine
 - Ubuntu mit allerlei Tools
 - Benutzername *iot*, Passwort *iot*
- Lokales WLAN
 - SSID: *IOT-WS*, PW: *Spartakiade2017*
- Knowledge Base
 - <http://192.168.30.10>



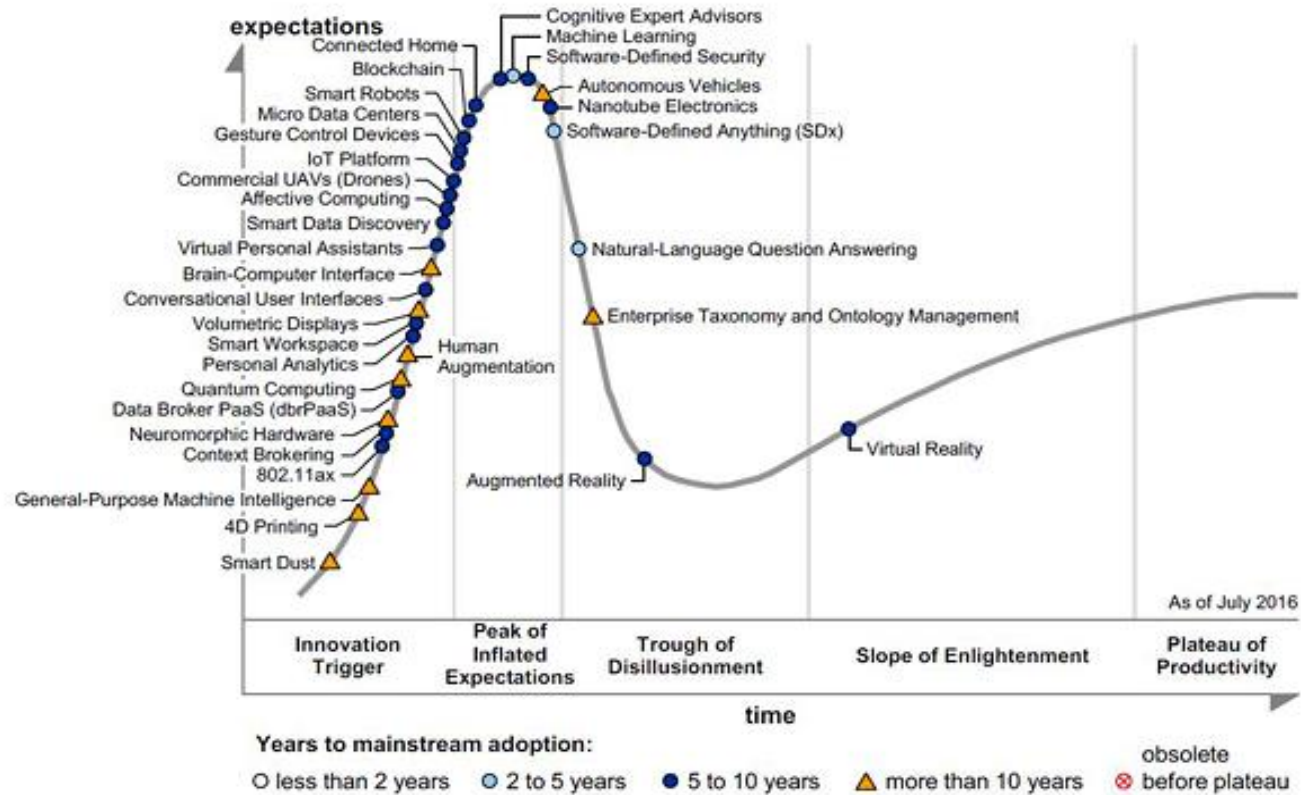
2

Einführung

Gartner Hype Cycle (2011)



Gartner Hype Cycle (2016)

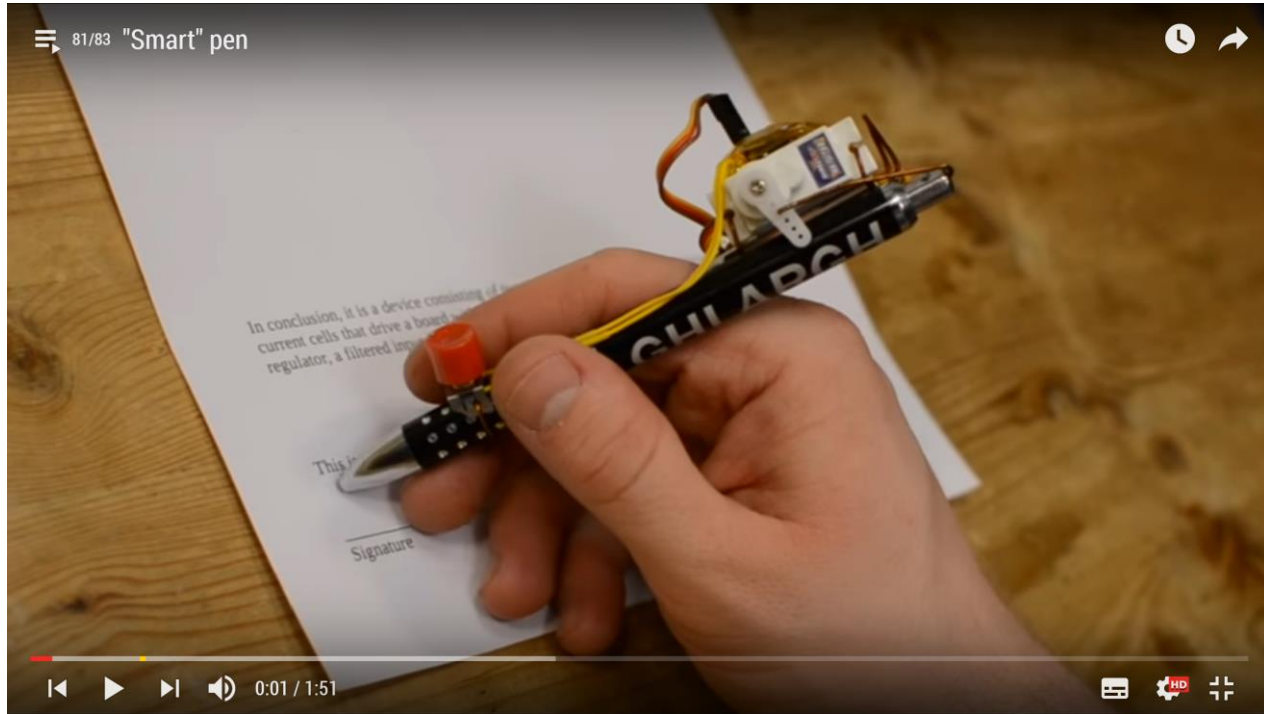


Source: Gartner (July 2016)

Ist das IoT?



Ein Video-Beispiel



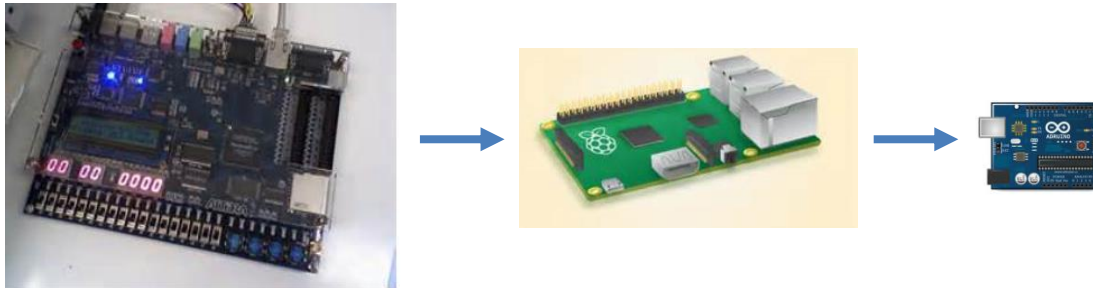
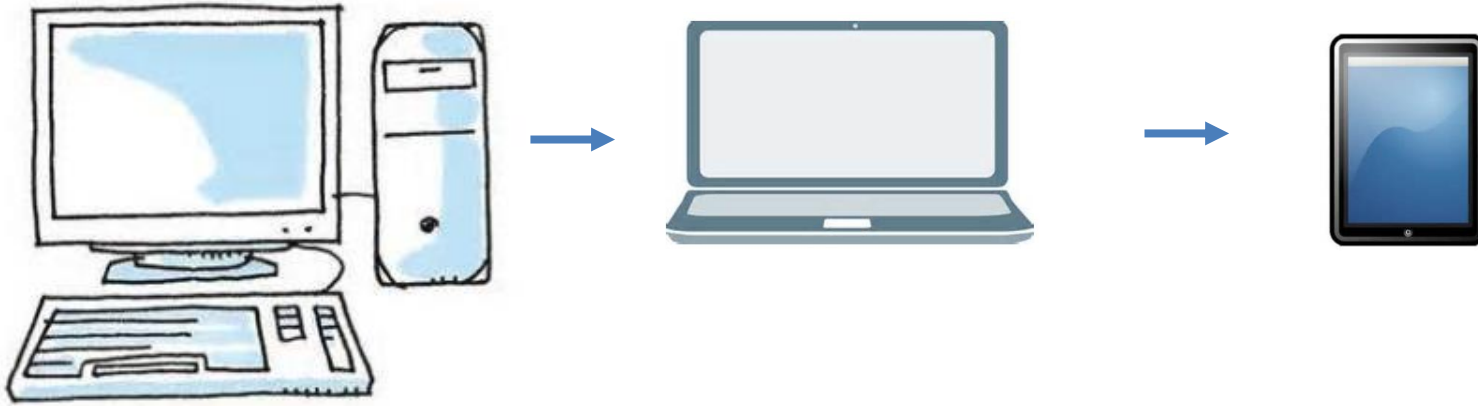


Was assoziiertst du mit IoT?

Anforderungen an IoT

1. „Collect“
 - Datenerfassung über Sensor(en)
2. „Control“
 - Datenverarbeitung über IP-fähiges Gerät
3. „Communicate“
 - Datenweiterleitung und Kommunikation über das Internet

Immer kleiner, immer günstiger, ...

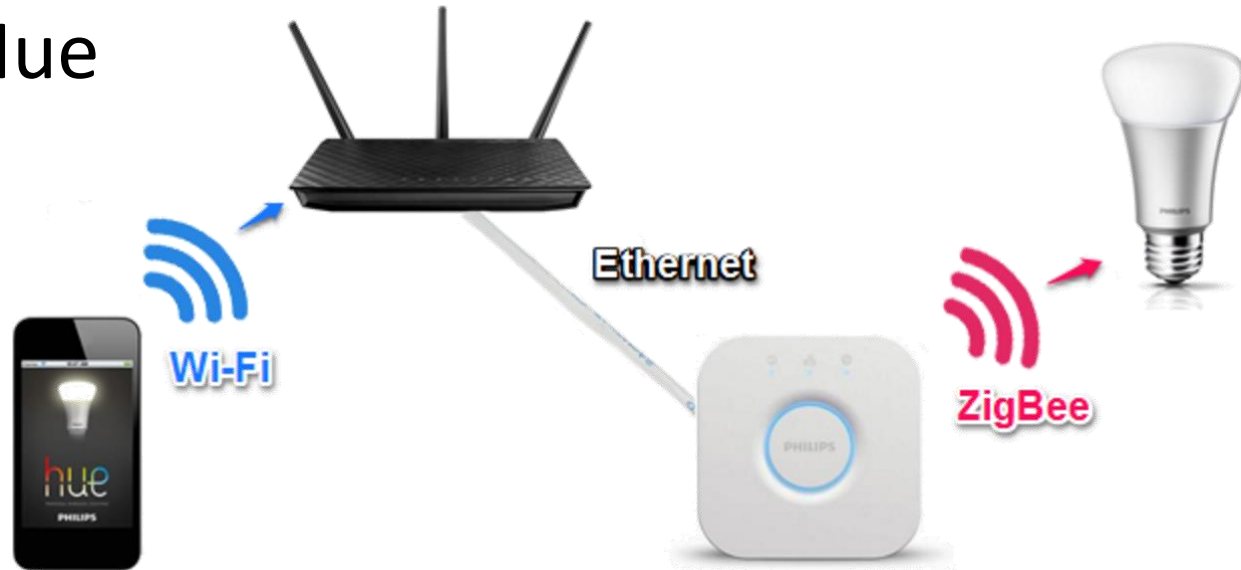




Aufgabe: Bringe eine WLAN-fähige Lampe dazu, alle 5 Sekunden ihre Farbe zu ändern

Erstes Beispiel: Lampensteuerung

Philips Hue



<https://developers.meethue.com/philips-hue-api>



Erstes Beispiel: Lampensteuerung

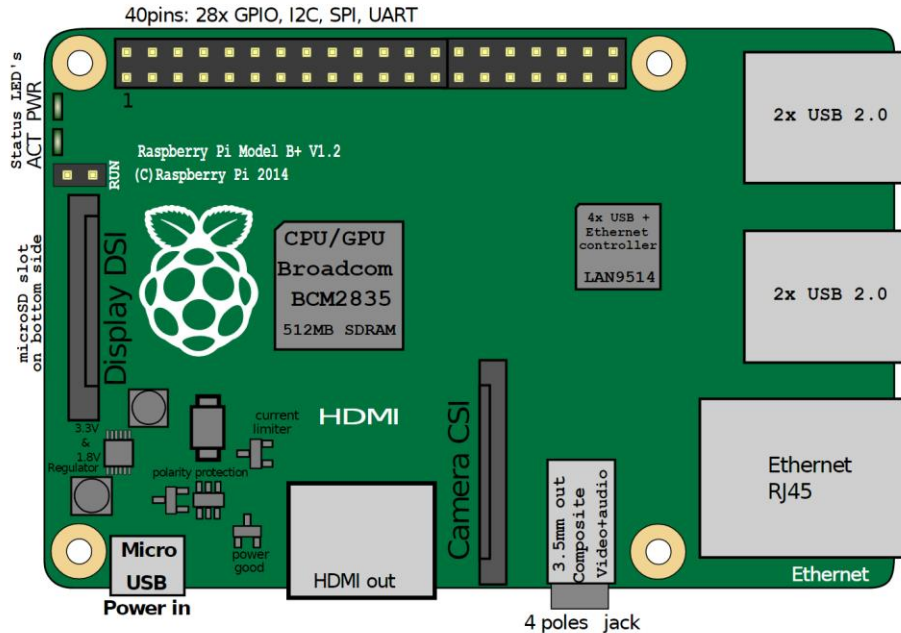
RESTful Webservice

Address	<code>http://192.168.30.110/api/bUGlwMHwOPzXVltpqnCgnpJF1gLVTKIVZ6CAjwu-/lights/1/state</code>
Body	<code>{"on":true, "sat":254, "bri":254,"hue":10000}</code>
Method	PUT

<http://<bridgeip>/debug/clip.html>



Raspberry Pi



- Raspberry Pi 3 Model B
- Raspbian Jessie OS
- Verbindung mittels SSH
- User: **pi**
- PW: **raspberry**

> *ssh pi@192.168.30.xxx*



Codebeispiel (Python)

```
import requests
import json

url =
"http://[bridgeurl]/api/bUGlwMHwOPzXVltpqnCgnpJF1gLVTKIVZ6CAjwu-
/lights/1/state"

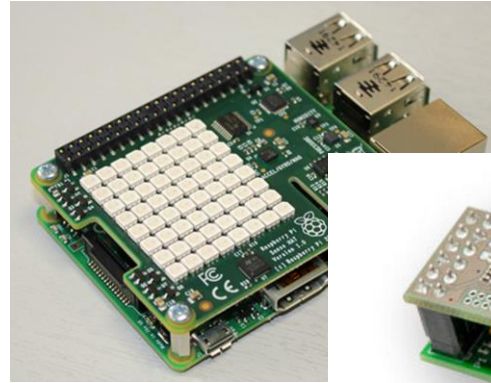
data_on = {"on":True, "sat":254, "bri":254,"hue":5000}
data_off = {"on":False}

r = requests.put(url, json.dumps(data_on), timeout=5)
```



Verschiedene Aufsätze möglich

- Sensor Hats
- ZigBee-Modul
- Andere Erweiterungen





Ist das nicht mit Kanonen auf
Spatzen geschossen?



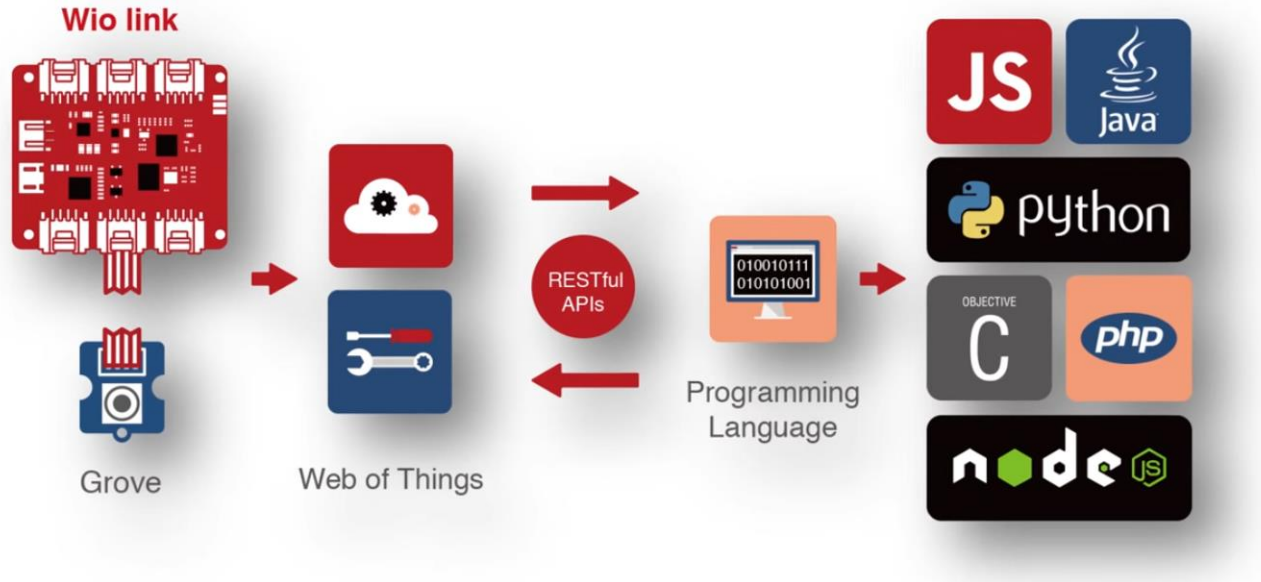
Micro- controller

SeeedStudio Wio Family

- Ehemaliges Kickstarter-Projekt



Hersteller-Video



Grove Sensoren

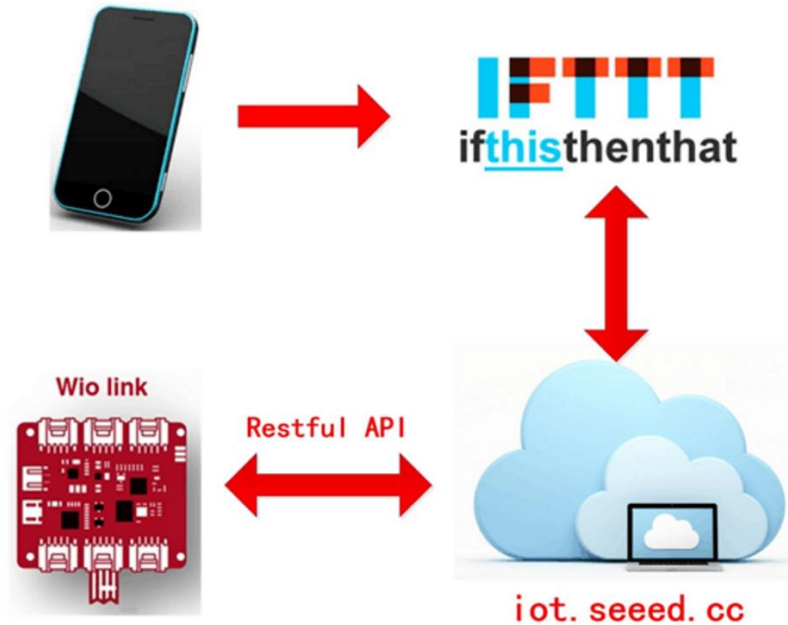


Zugriffsmöglichkeiten

- Via Smartphone App
- Via Cloud-basiertem RESTful Webservice
- Oder direkte Programmierung der Firmware



Via Cloud API




<https://us.wio.seeed.io/v1/node/>



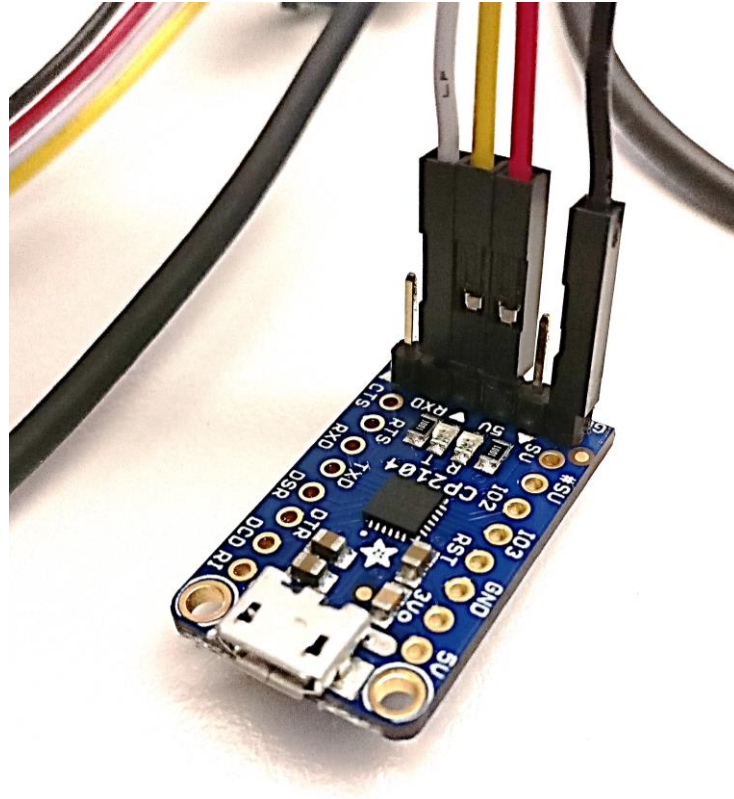
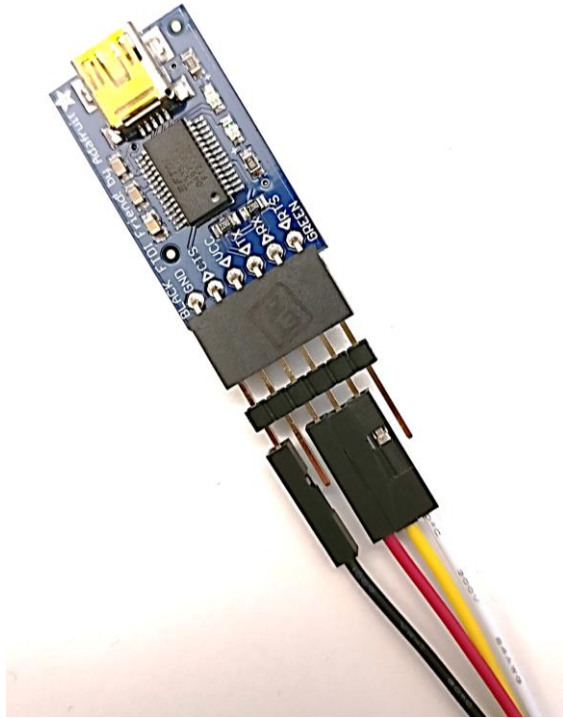
Aufgabe: Bringe eine LED dazu,
aller 5 Sekunden an/aus zu
gehen

Via direkter Programmierung

Wir benötigen dazu:

- Arduino IDE 
- Einen USB-to-Serial-Wandler (FTDI-Adapter)

FTDI Verkabelung



Transfer auf Microcontroller

- Der Wio Node muss zuvor in den Konfigurationsmodus versetzt werden
 - FUNC – Taste für 4 Sekunden drücken
 - Danach zusätzlich kurz RESET-Taste drücken
 - Beide Tasten loslassen
- Korrekten seriellen Port und Baudrate einstellen
- ino-Sketch compilieren und übertragen



Zweites Beispiel: LED-Steuerung

Arduino sieht in einem ino-Sketch
zwei Funktionen vor:

```
void setup() {  
}
```

```
void loop() {  
}
```



Zweites Beispiel: LED-Steuerung

Allgemeine Befehle

```
#define PORT 5
```

```
pinMode(PORT, OUTPUT);
```

```
Serial.begin(115200);
```

```
Serial.println("this is a string");
```

```
digitalWrite(PORT, HIGH);
```

```
delay(3000);
```

```
{  
  "board_name": "Wio Node v1.0",  
  "board_vendor": "seeedstudio",  
  "board_flash_map": 6,  
  "board_flash_spi_speed": 40,  
  "board_flash_spi_mode": "QIO",  
  "board_builtin":  
  {  
    "FUNCTION_KEY": 0,  
    "STATUS_LED": 2,  
    "GROVE_POWER_SWITCH": 15  
  },  
  "interfaces":  
  {  
    "D0": { "type": "GPIO", "pin": 3 },  
    "D1": { "type": "GPIO", "pin": 5 },  
    "A0": { "type": "ANALOG", "pin": 17 },  
    "I2C0": { "type": "I2C", "pinsda": 1, "pinscl": 3 },  
    "I2C1": { "type": "I2C", "pinsda": 4, "pinscl": 5 },  
    "UART0": { "type": "UART", "pintx": 1, "pinrx": 3 }  
  }  
}
```





Was ist daran jetzt IoT?

Letztes Beispiel iotifizieren

- Microcontroller mit WLAN verbinden
- Aktuellen Status weiterkommunizieren, z.B. an einen MessageBroker



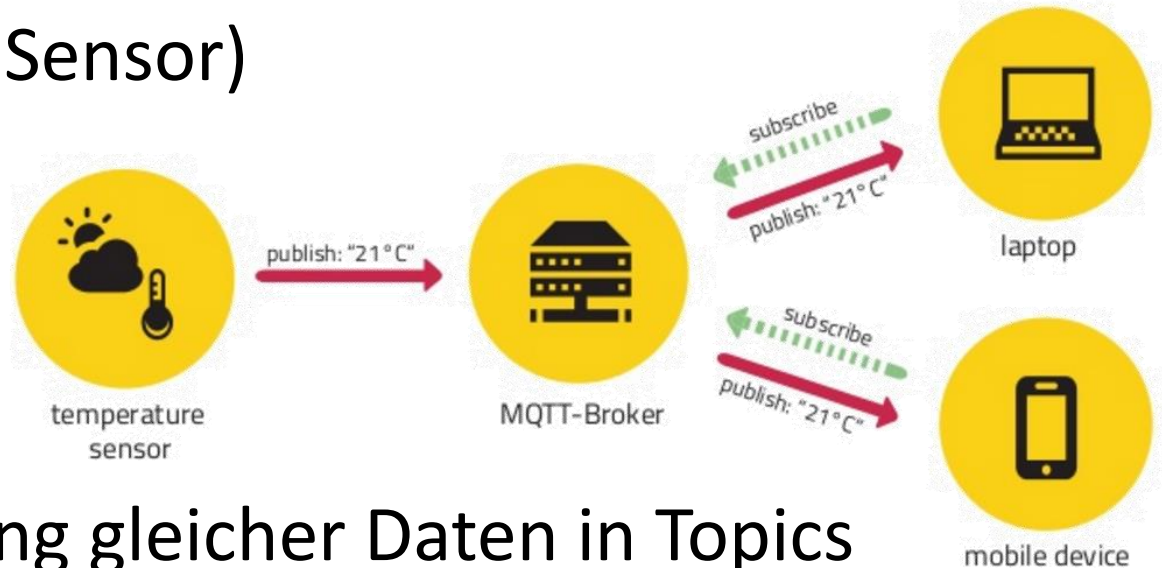
Arduino-Code zur WiFi-Verbindung

```
#include <ESP8266WiFi.h>
const char* ssid = "IOT-WS";
WiFiClient wifiClient;
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println("WiFi connected");
```



Sensoren vs. Aktoren

- Subscriber (Aktor)
- Publisher (Sensor)



- Gruppierung gleicher Daten in Topics

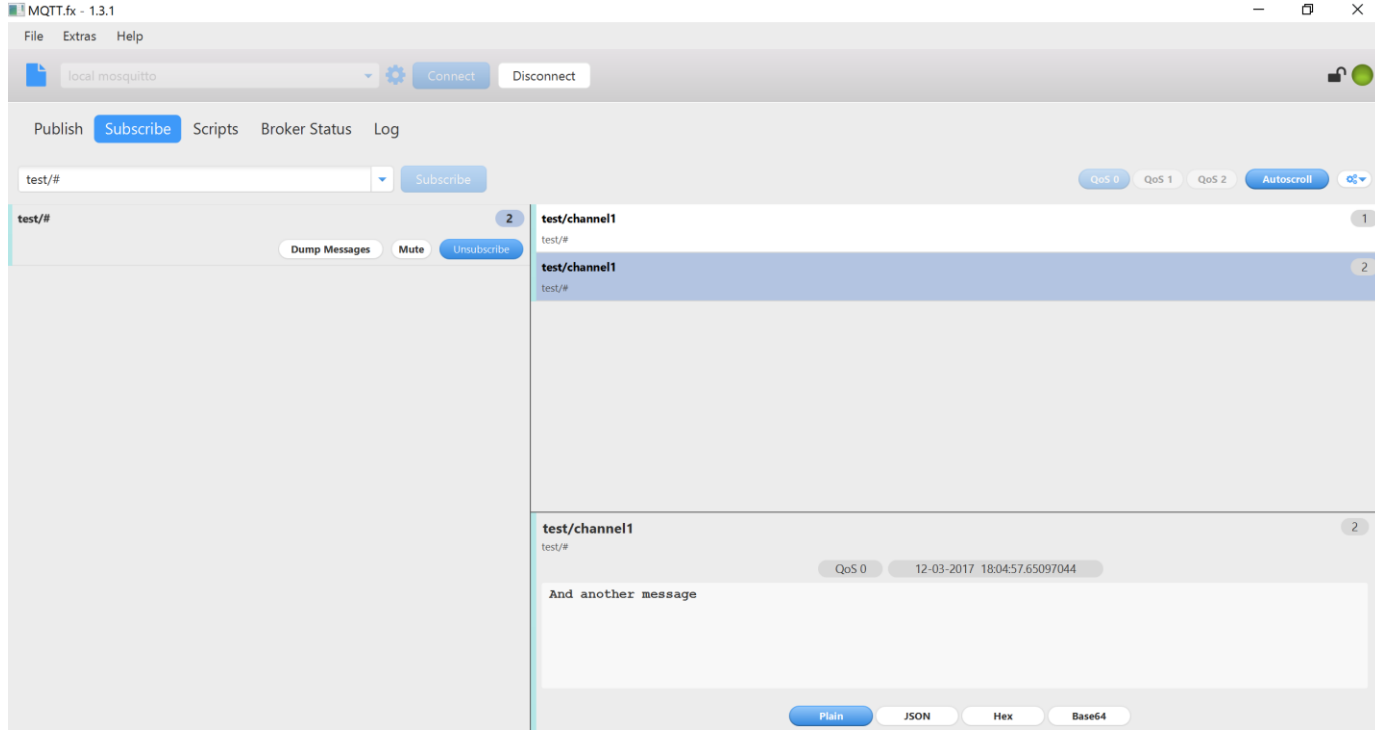


MessageBroker

- Warum Messaging statt einfacher HTTP Requests?
- Was ist ein MessageBroker?
- Was ist MQTT?



Mosquitto MQTT Broker



Besonderheiten

- Authentication & SSL
- Retained Messages
- QoS Levels



Arduino Code zum Pub

```
#include <PubSubClient.h>
const char* mqttServer = "192.168.30.10";
const char* mqttUsername = "spartakus";
const char* mqttPassword;
const int mqttPort = 1883;

String clientName;
String topicName;
while (!mqttClient.connected()) {
    mqttClient.connect((char*) clientName.c_str(), mqttUsername, mqttPassword);
}
mqttClient.loop();
mqttClient.publish(topicName, Message.c_str(), true);
```



Arduino Code zum Sub

[...]

```
int subscriptionTopicLength = TopicName.length() + 1;
```

```
char subscriptionTopicCharArray[subscriptionTopicLength];
```

```
topicName.toCharArray(subscriptionTopicCharArray, subscriptionTopicLength);
```

```
if (mqttClient.subscribe(subscriptionTopicCharArray)) {
```

```
    Serial.print("Subscribed to ");
```

```
    Serial.println(topicName);
```

```
}
```

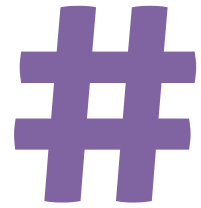




Was nützen uns diese
Daten jetzt?

4 Development





IoT ist nicht nur Hardware,
sondern vor allem Software
und die Verbindung
intelligenter Systeme

Aktionen

- Jede beliebige Software / App kann eingehende Nachrichten vom MQTT MessageBroker abonnieren und darauf aufbauend weitere Aktionen auslösen
- Meist finden sich ähnliche Workflows



NodeRED

- Tool von IBM
- nodeJS Anwendung mit Web GUI
- Ein Werkzeug, um Hardware, APIs und Online-Dienste miteinander zu verbinden („Baukasten“)



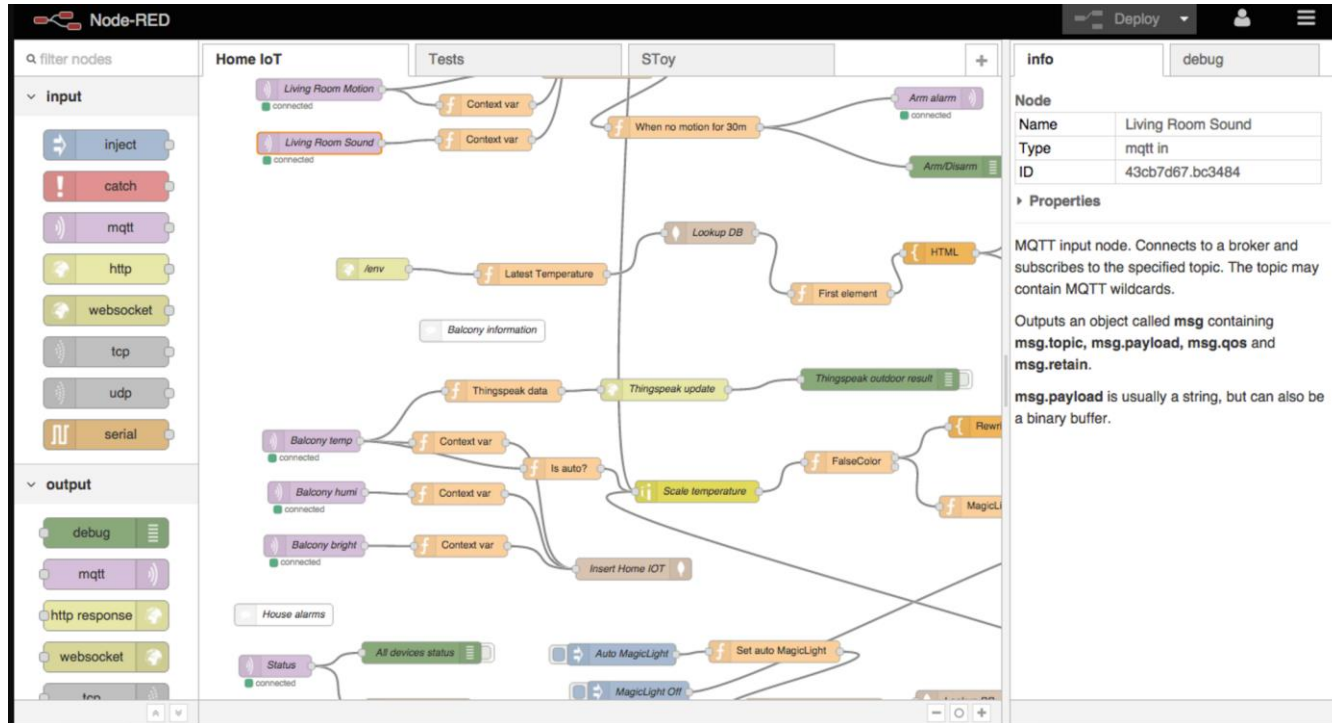
NodeRED

```
$ sudo npm install -g node-red  
$ node-red
```

- Default port 1880
- <http://localhost:1880>



NodeRED





Showcase

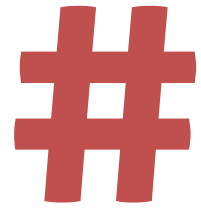
Showcase

- Die restliche Zeit des Workshops wollen wir nutzen, um das Wissen anzuwenden und gemeinsam ein komplexeres Szenario umzusetzen



Nerfgun Battle





Zielsetzung: Heute Abend
17:00 Uhr sollen zwei Teams
mit Nerfguns gegeneinander
antreten können

Wir benötigen dazu

- Spiel-Ideen
- Eine gute Team-Organisation
- Unterschiedliche Verantwortlichkeiten
- Die Integration zu einem Gesamtsystem



Wir bieten Euch

- Hardware-Equipment
- Eine zentrale Server-Instanz
- Ein NodeRED-Gaming-Framework
- Rat bei Problemen



Gaming-Framework



6 Retro

Retrospektive

- Was hast du heute Neues gelernt?
- Bist du mit dem erreichten Stand zufrieden?
- Was hätte besser laufen können?





**Danke für
den spannenden Tag**

@andrelanger

@oliverlorenz

