

IoT, Smart-Home und Web in einen Topf und kräftig umrühren

Klaus Weichinger

snaky.1@gmx.at / Instagram: __wg__

Die Herausforderung von IoT und Smart-Home

sinnvolle Anwendungen
finden

&!

nette Spielereien

Zutaten

- Hardware für die physikalische Anbindung
- Infrastruktur (Kommunikation, Server)
- Übergeordnete Anwendungsschicht für die Logik & Verknüpfung
- Eventuell bestehende Datenquellen nutzen



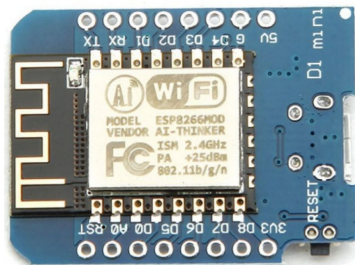
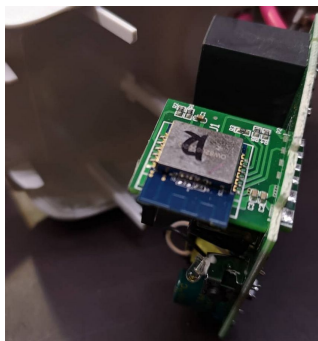
Vorhandene Zutaten

— — —

Hardware

Kleine Mikroprozessoren mit WLAN
einfach zu programmieren (C/C++)

Werden auch im Consumer-Bereich eingesetzt.



Infrastruktur

Internet

WLAN, LoRa-WAN, ...

Server

Smartphones

Browser HTML+CSS+JS

Benötigt: Übergeordnete Anwendungsschicht

— — —

Bekannte Systeme für IoT reduzieren sich im Wesentlichen auf

- Daten sammeln/austeilen
- Anzeigen

z.B. MQTT, TheThingsNetwork

Mehr Funktionalität:

ThingSpeak (Mathworks), AWS, ...
Phoenix Contact - ProfiCloud

(Meine) Anforderung

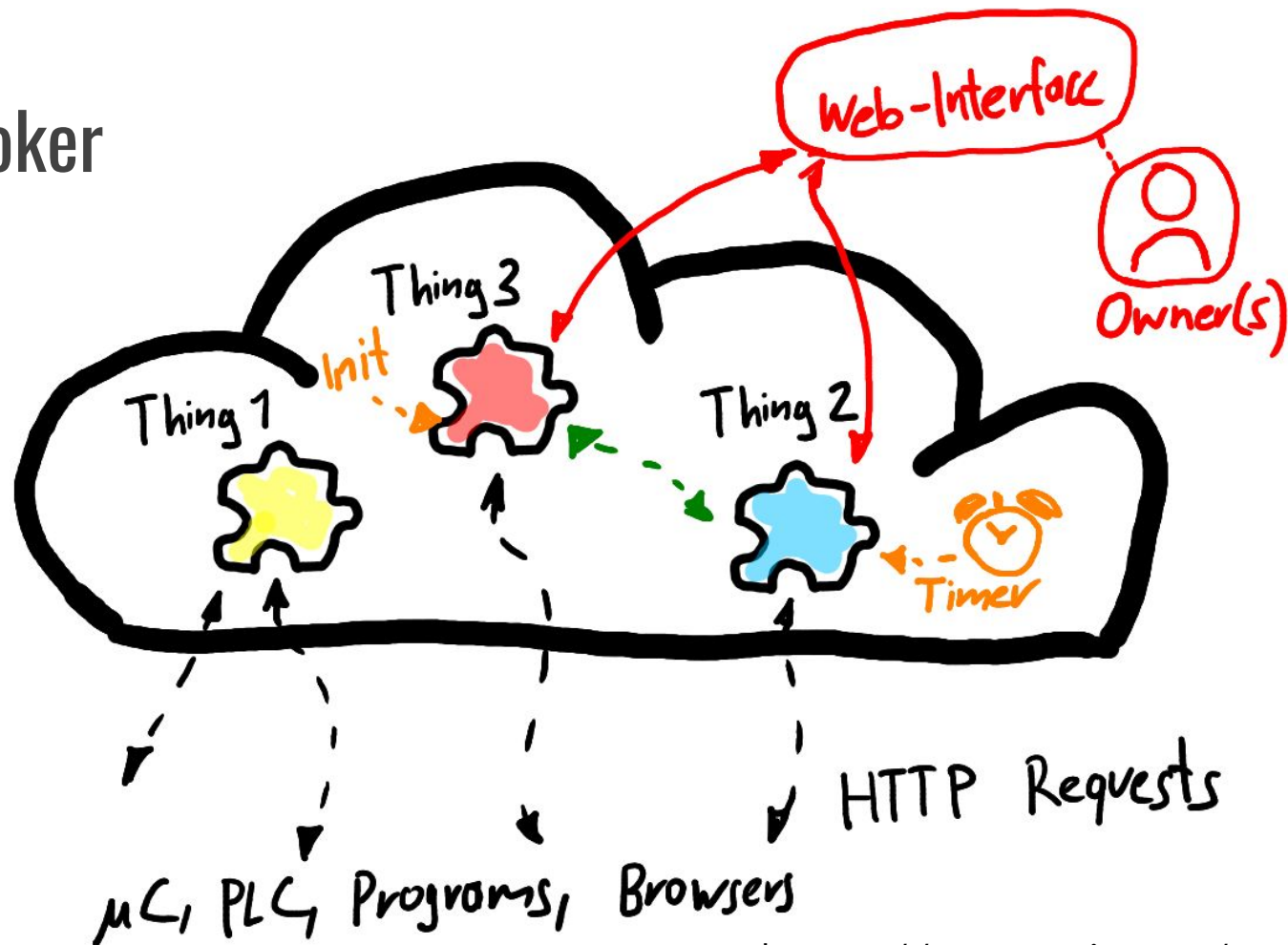
- Frei programmierbarer Micro-Service mit Web basierter IDE
- Interaktion mit anderen Micro-Services
- Persistenter Datenspeicher
- Zyklische Aufgaben
- Zugriffssteuerung API-Token & User-Session
- REST-API (WEB-Sockets, ...)

Aktueller Stand

- Spielwiese mit unterschiedlichen Strategien und Technologien
- Bewährte Design-Pattern müssen sich durch Try/Error und praktischen Anwendungen herauskristallisieren
- Denkweise noch nicht IoT gerichtet



IoT-Broker



Demo IoT Steckdose

Schritt 1:

Neue Firmware für die Dose

Schritt 2:

Microservice

Schritt 3:

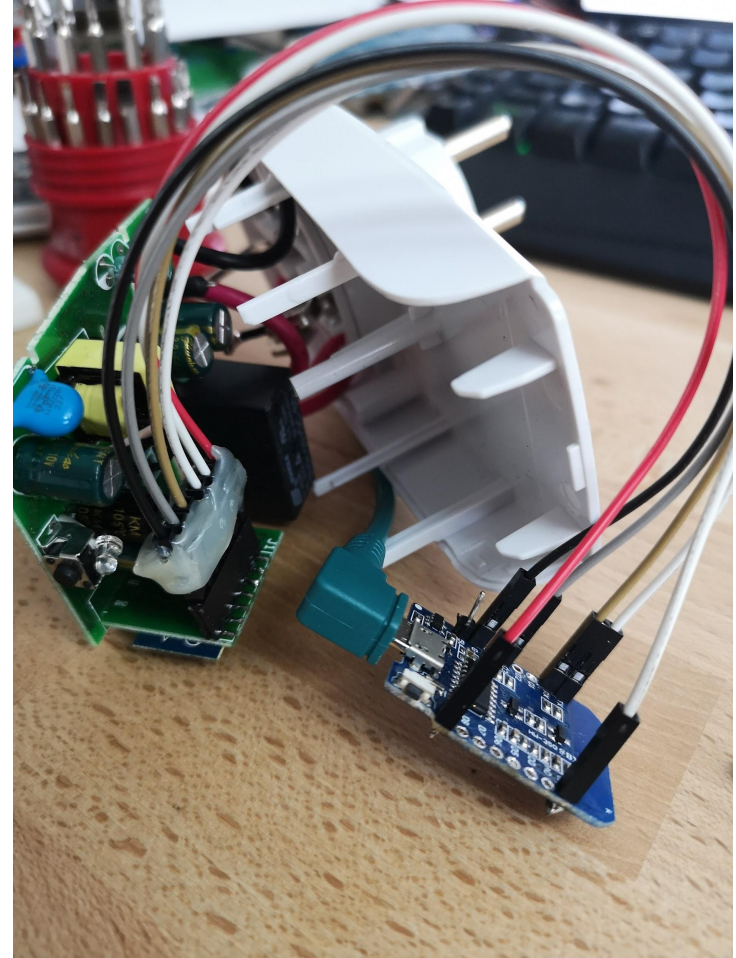
Neue Funktionen und nicht
nur Ein/Aus



Schritt 1: Firmware

Bild: ESP8266 des Steckdose wird mit einer neuen Firmware bespielt.

Live-Coding...



Schritt 2 & 3:

Live Coding

— — —

- Arduino Firmware (links)

- IoT-Broker Thing-Script (rechts)

```
#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>
#include <ESP8266HTTPClient.h>
#include "ArduinoJson.h"

ESP8266WiFiMulti WiFiMulti;

void setup() {
  Serial.begin(115200);
  Serial.println();
  WiFi.mode(WIFI_STA);

  delay(100);
  WiFiMulti.addAP("IoT", "iot-2018");
  WiFiMulti.addAP("Zuhause", "123456789");

  pinMode(D1, OUTPUT); // D1 GPIO5 Relay off/LOW Pulse
  pinMode(D6, OUTPUT);
  digitalWrite(D1, LOW); // D6 GPIO12 Relay on /LOW Pulse
  digitalWrite(D6, HIGH);

  pinMode(D2, OUTPUT); // D2 GPIO4 blue WiFi LED
  digitalWrite(D2, LOW);

  pinMode(D5, INPUT); // D5 GPIO14 Push Button

  Serial.println("Init done");
}

// Request to Broker { button:0, ssid:"Zuhause" }
//
// Response from Broker { relay:0, blueled:0 }

void loop() {
  if (WiFiMulti.run() == WL_CONNECTED)
  {
    HTTPClient http;
    StaticJsonBuffer<400> jsonBuffer;
    jsonBuffer.clear();
    JsonObject& root=jsonBuffer.createObject();
    root["button"]=(digitalRead(D5)?0:1);
    root["ssid"]=WiFi.SSID();

    String value="";
    root.printTo(value); // {"button":1}

    http.begin("http://iotdev.hilvy.ac.at/thing/xxxxxxxxxxxxx/io");
    http.addHeader("Content-Type", "application/json");
    int httpCode=http.POST(value);
    if (httpCode==HTTP_CODE_OK)
    {
      String json=http.getString();
      jsonBuffer.clear();
      JsonObject& res=jsonBuffer.parseObject(json);
      digitalWrite(D2,(res["blueled"]==1)?1:0);
      digitalWrite(D6,(res["relay"]==1)?0:1);
      digitalWrite(D1,(res["relay"]==1)?1:0);
    }
    http.end();
    delay(100);
  }
  else
  {
    Serial.println("Not connected");
    delay(100);
  }
}
```

```
io=ramVar("io",{});
if (method=="init")
{
  io={relay:0, blueled:0, button:0, ssid:"", timestamp:0, online:false};
  timeout=0.1;
}
else if (method=="tick")
{
  //io.relay=(io.relay==0)?1:0;
  timeout-=1;
  oldstate=io.online;
  io.online=((unixtime-io.timestamp)<2000);
  if (oldstate==true && io.online==false)
  {
    webPush("info","Steckdose ist offline!");
  }
}
else if (method=="http")
{
  if (name=="relay")
  {
    if (defined(value))
    {
      if (value==1) io.relay=1;
      if (value=="on") io.relay=1;
      if (value==0) io.relay=0;
      if (value=="off") io.relay=0;
      if (value=="toggle") io.relay=(io.relay==0)?1:0;
    }
    res=io.relay;
  }
  if (name=="button") res=io.button;
  if (name=="io" && defined(value))
  {
    io.timestamp=unixtime;
    // input
    io.button=value.button;
    io.ssid=value.ssid;

    // logic

    // output
    res=io;
  }
  if (name=="status")
  {
    if (io.online)
    {
      res=(io.relay==1)?"EIN":"AUS";
    }
    else
    {
      res="OFFLINE";
    }
  }
  if (name=="gui")
  {
    res = {
      title: "Dev-Treff",
      refreshinterval:300, // content refresh interval in ms
      elements:
      [
        {
          type:"textview",
          label:"Status",
          thinguri:thingname+"/status",
          value:"..."
        },
        {
          type:"buttons",
          label:"Actions",
          list:
          [
            {thinguri:thingname+"/relay?value='toggle'",label:"Toggle"}
          ]
        }
      ]
    };
  }
}
```

Und, ist der Maker in dir geweckt?