

IoT-Lab reloaded

Welcome to IoT-Lab reloaded

IoT-Lab

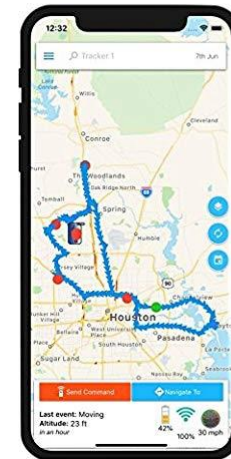
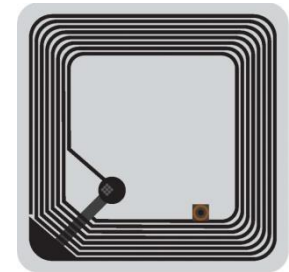
- v1.0 in 2017 – Clever!Lab / Watson Build Challenge
- v2.0 ohne konkretes Projekt
- offenes Lab
 - Mittwochs 17:30 Uhr
 - Einführung oder Vortrag 15 min – jeder darf
 - Eigenes oder Open Source-Projekte vorstellen
 - Gemeinsam an Projekten arbeiten
 - Mikrokontroller programmieren, Back-End stricken

Die Kindertage des IoT

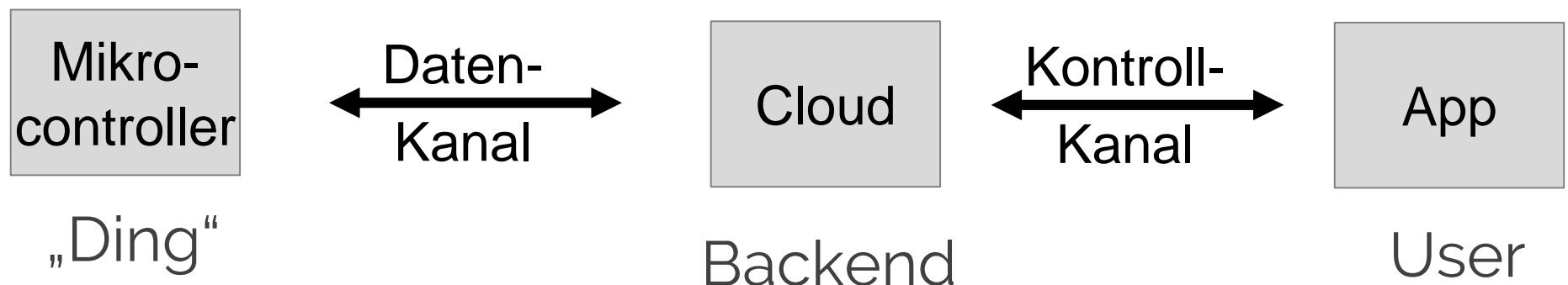
IoT-Lab

Maschinenlesbare Etiketten

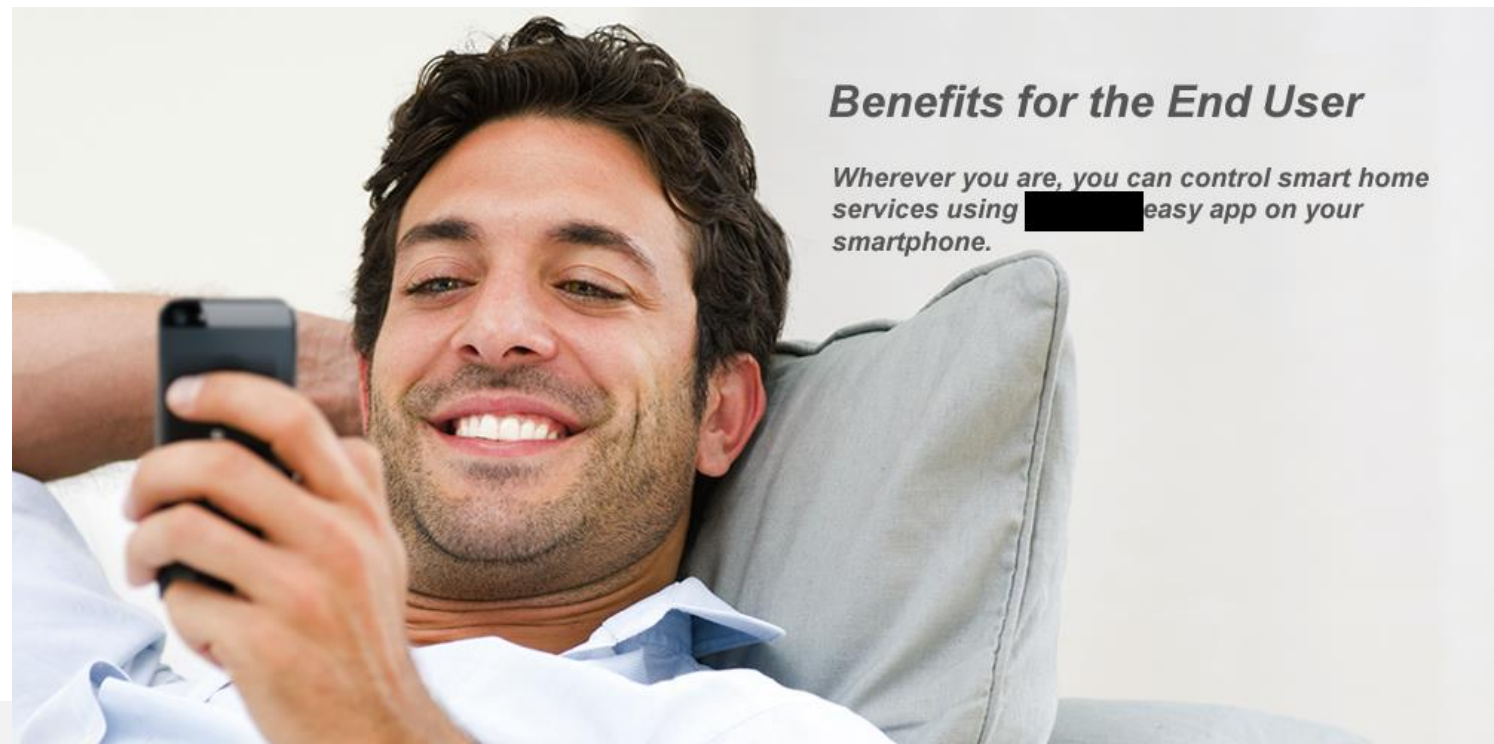
- Am Anfang war der „Tag“
- Passive Tags:
 - Printed Tags: Barcode, QR-Code
 - Electronic Tags: RFID-Tag
- Aktive Tags
 - GPS-Tracker/-Sender
 - SmartPhones ...



- Alltägliche Dinge werden „smart“
- Verbindung zu einem zentral gesteuerten Backend
- Kontrolle beim Endbenutzer, meist über App
- Monitoring/Überwachung, Alarm
- Daten sammeln
- Steuerung von überall
- *any³: Control anything, from anywhere and anytime*



- Einfache Überwachung und Verwaltung der Smart-Devices zu akzeptablen Preisen (*in €*)
- Ersparnisse bei Strom, Heizung, Sprit etc. („Personal Revenue“)

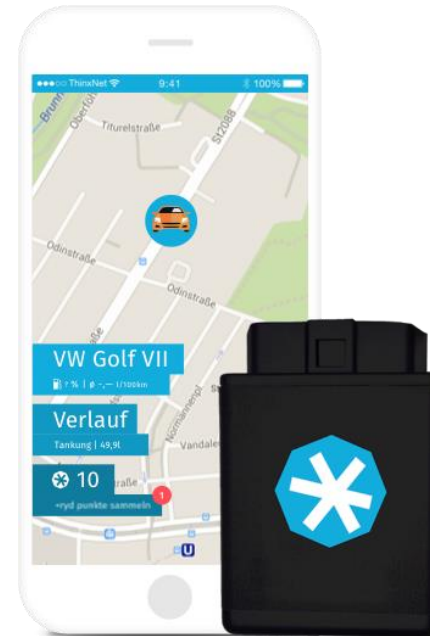


- Kopplung mit anderen Produkten
 - Kommunikationsanbieter \Leftrightarrow Smart-Home
 - DSL-Router \Leftrightarrow Smart-Home
- Sammlung von **Daten** über viele Geräte, User, Länder, Zeitzonen, ...
 - Generierung von Informationen
 - Big Data, Data Science
- Steuerung von
 - Flotten/Fahrzeugen
 - Warenströmen



Anwendungsbeispiele

- Smart-Home
 - Schaltsteckdosen
 - Thermostate
- Smart-Farm
 - Herdenmanagement
 - Bodensensoren
- Smart-Car
 - OBD-Stecker
- Smart-City
- Smart-Fab
- Smart-Everything



IoT-Lab

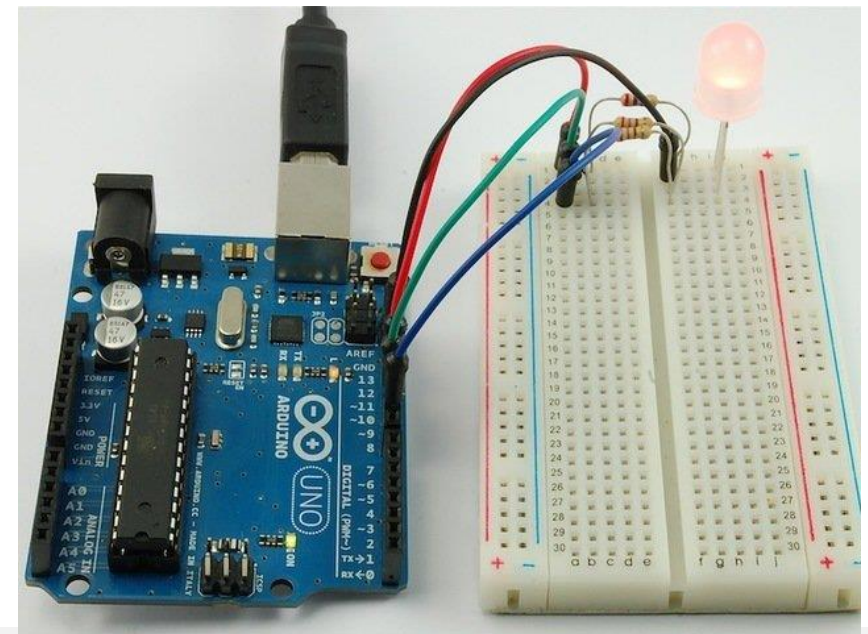
Abhängig von Einsatzgebiet und Stromversorgung:

- Std.-IT: WLAN, Bluetooth
 - High power, high Bandwidth, short range
- Home-Automation: DECT, Zigbee, ISM 433MHz
 - Low power, medium bandwidth, short range
- Low-Energy: LoRaWan, SigFox
 - (Ultra) Low power, very low bandwidth, high range
 - https://media.ccc.de/v/35c3-9491-hunting_the_sigfox_wireless_iot_network_security
- Meist verwendete Protokolle:
 - MQTT(s)
 - HTTP(s)



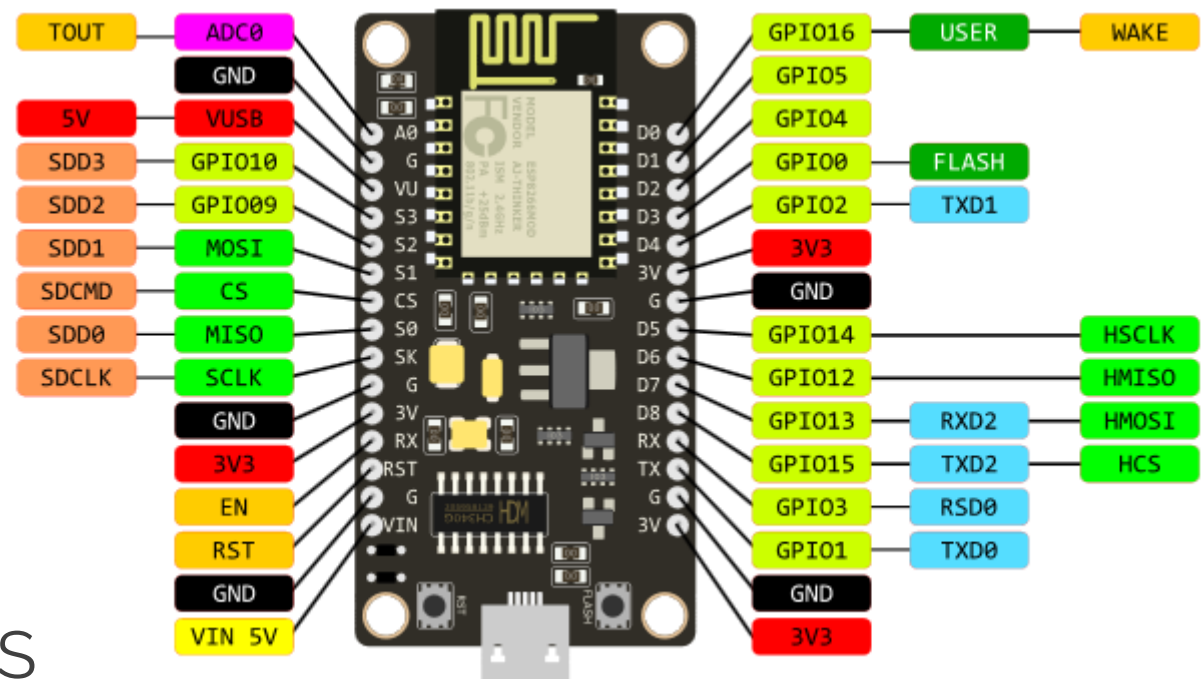
- Stromversorgung: oft nur Batterien
- Kosten: muss günstig sein
- Ressourcen: muss mit wenig MHz, RAM usw. auskommen
- Sensoren: Vielzahl von Protokollen
- Security: ein Thema für sich ...

- Mikrocontroller erfüllen diese Anforderungen
- Preise in den vergangenen Jahren drastisch gefallen
- Parallel: Open Source-Projekte, offene Entwicklungswerkzeuge u.a.
- Keine Lizenz zum Löten notwendig
 - „Breadboards“ ermöglichen einfaches Zusammenstecken und Verdrahten
- Flut von Projekten im Netz
 - Was die anderen können, können wir doch schon lange



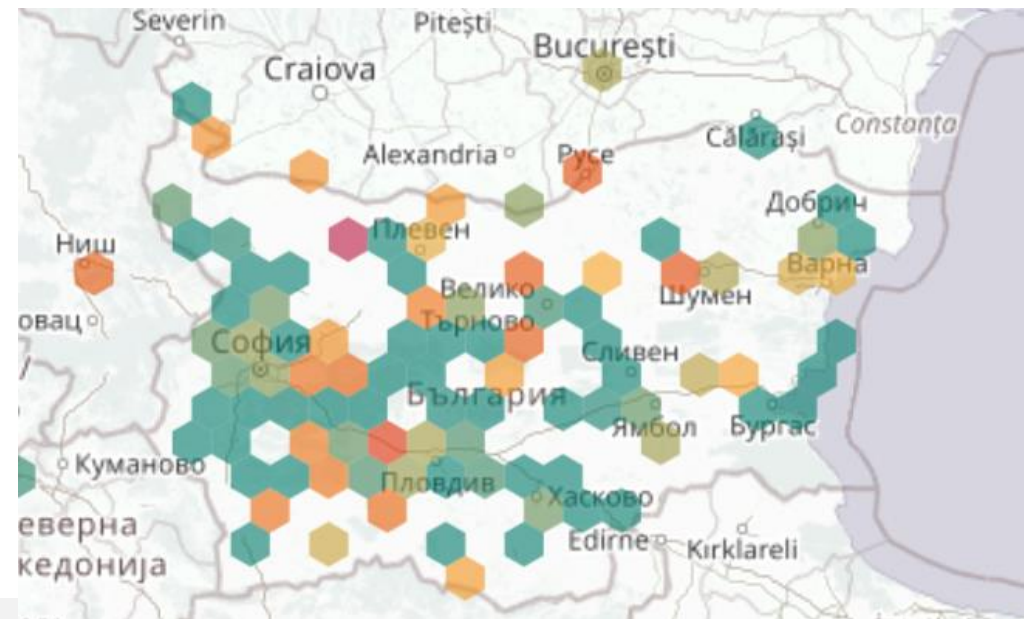
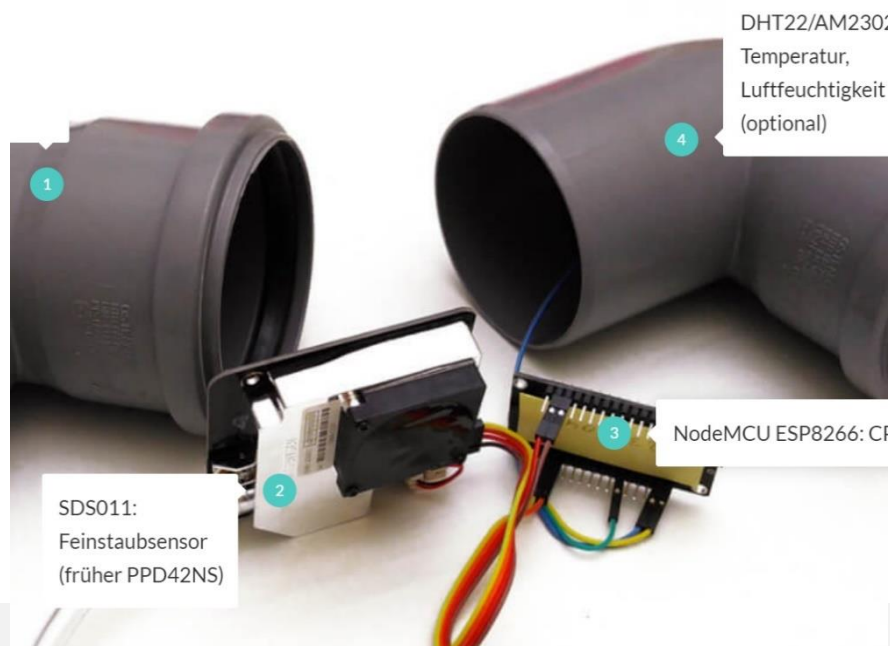
Der Mikrocontroller-VW Käfer: ESP 8266

- 32bit-Mikrocontroller der Fa. Espressif
- verschiedene Bauformen
- wir verwenden NodeMCU v3 Dev. Board
- Progr.sprachen:
 - Arduino / C++
 - Lua
 - Basic
- Features:
 - 80 MHz CPU
 - ~50 KB(!) freies RAM
 - 4 MB Flash für Code & FS



Bsp.: Feinstaub-Messung - luftdaten.info

- OK Lab Stuttgart: Open Data und Citizen Science fördern
- Daten sind Database Contents License (DbCL) v1.0
 - Über API frei abrufbar
- Weltweites Netzwerk mit Schwerpunkt (D)
- Gebaut mit ESP8266 Mikrocontroller



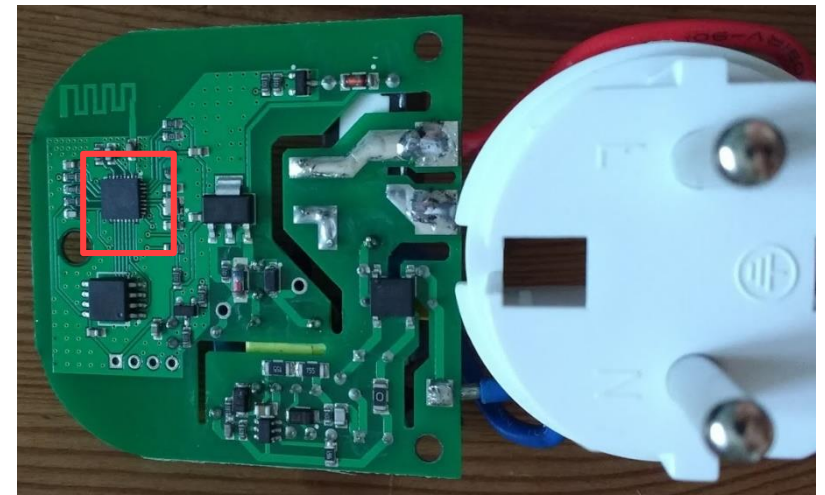
Bsp.: Stromverbrauch per IoT messen

- Beispiel: Stromverbrauch messen am SmartMeter

- Raspi mit Python-Skript
- Schreibt in Volkszähler (lokal)
- Und TimeSeries-DB (Cloud)

- Projekt:

- Strom einzelner Verbraucher messen
- WLAN-Steckdose mit ESP8266 umprogrammieren
- Open-Source: Tasmota
<https://github.com/arendst/Sonoff-Tasmota/>



- Arduino IDE installieren
- ggf. CH430 Treiber installieren
- Wer braucht ESP & Breadboard?
 - Sammelbestellung; ESP & Breadboard je ~5€
- Beispiel For Runaways: *Blink.ino*
- Nächste Woche: Temp. & Luftfeuchtigkeit messen
- Backend: IBM Cloud Produkte; Lite Plan

- Btw.: Essen?

Vielen Dank für Eure Aufmerksamkeit!

ESP8266 – first steps

- Arduino IDE herunterladen
 - <https://www.arduino.cc/en/Main/Software>
- Unter Datei/Voreinstellungen/Zusätzl. Boardverwalter im Feld URL eintragen:
 - http://arduino.esp8266.com/stable/package_esp8266com_index.json
- Werkzeuge/Board/Boardverwalter, nach ESP suchen, Installieren
- Werkzeug/Board: NodeMCU 1.0 (ESP-12E) auswählen
- Werkzeuge/PORT:COMx auswählen (ggf. im Geräte-Mngr nachsehen)
- Upload: 115200
- Ggf. CH340G Bridge Treiber installieren;
 - <http://www.arduined.eu/files/CH341SER.zip>

Quelle: <https://www.davidlackovic.de/beginners-guide-nodemcu-flashen/>


```
// simple program to make built-in LED blink

#define LED_BUILTIN 2

void setup() {
    pinMode(LED_BUILTIN, OUTPUT);    // Initialize the LED_BUILTIN
                                    // pin as an output
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(LED_BUILTIN, LOW);    // Turn the LED on
    delay(1000);                        // Wait for a second
    digitalWrite(LED_BUILTIN, HIGH);   // Turn the LED off
    delay(2000);                        // Wait for two seconds
}
```