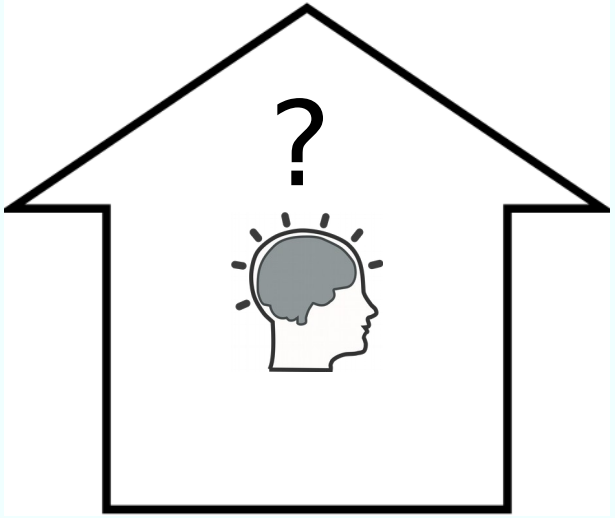
Cloudbusters

Wer kontrolliert dein Smart Home?



Prof. Dipl.-Ing. Klaus Knopper
<ki@knoppix.info>

Was ist „Smart“ im Home?

- ✔ Geräte mit WIFI-Steuerung und -Abfrage von Leistungsdaten, Statistiken, Sensoren: z.B. Steckdosen, Lampen, Kaffeemaschinen, Kühlschrank, Ofen, Waschmaschine
- ✔ Smart TV mit Streaming-Empfang und Apps
- ✔ Smartphones (heißen ja auch so…), Tablets
- ✔ Wifi-Kameras
- ✔ Türöffner ...

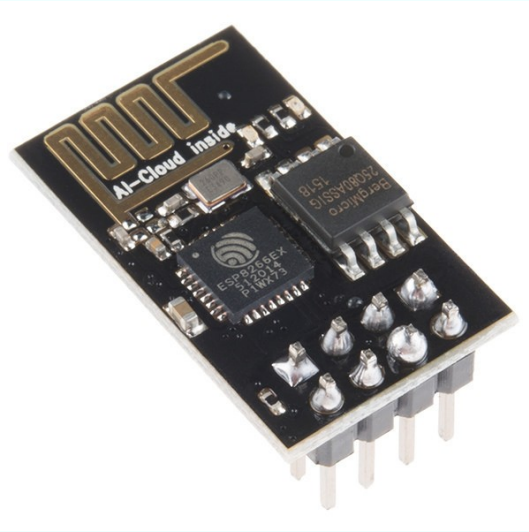
Was habe ich davon?

- ✔ Überblick über alle Funktionen in einer App
- ✔ Steuerungsmöglichkeit durch Sprache oder Gesten
- ✔ Vereinfachung / leichterer Zugriff auf Informationen
- ✔ Mit einem Signal mehrere Aktionen auslösen („Beleuchtung ein!“)
- ✔ Höherer (Standby-) Stromverbrauch

Wie funktioniert das? (1)

- ✔ Geräte sind i.d.R. per WLAN erreichbar, mit teilweise sehr kostengünstiger Hardware, die Schalter betätigen kann (GPIO, Relais) und ein integriertes WLAN-Modem besitzt.

- ✔ Beispiel: ESP 8266
  - 32-Bit-Prozessorkern, 80-160MHz
  - 64 kB RAM Befehlsspeicher
  - 96 kB RAM Datenspeicher
  - Firmware auf externem, seriellem Flash-Speicher
  - SPI-Schnittstelle
  - WLAN nach IEEE 802.11 b/g/n
  - Preis je nach Anbieter ab ca. 1,80 €



Wie funktioniert das? (2)

- ✔ Programmierung: z.B. mit Arduino-Entwicklungsumgebung, Software (monolithisches Programm, kein OS) wird per serieller Schnittstelle auf den externen Flash-Speicher transferiert,

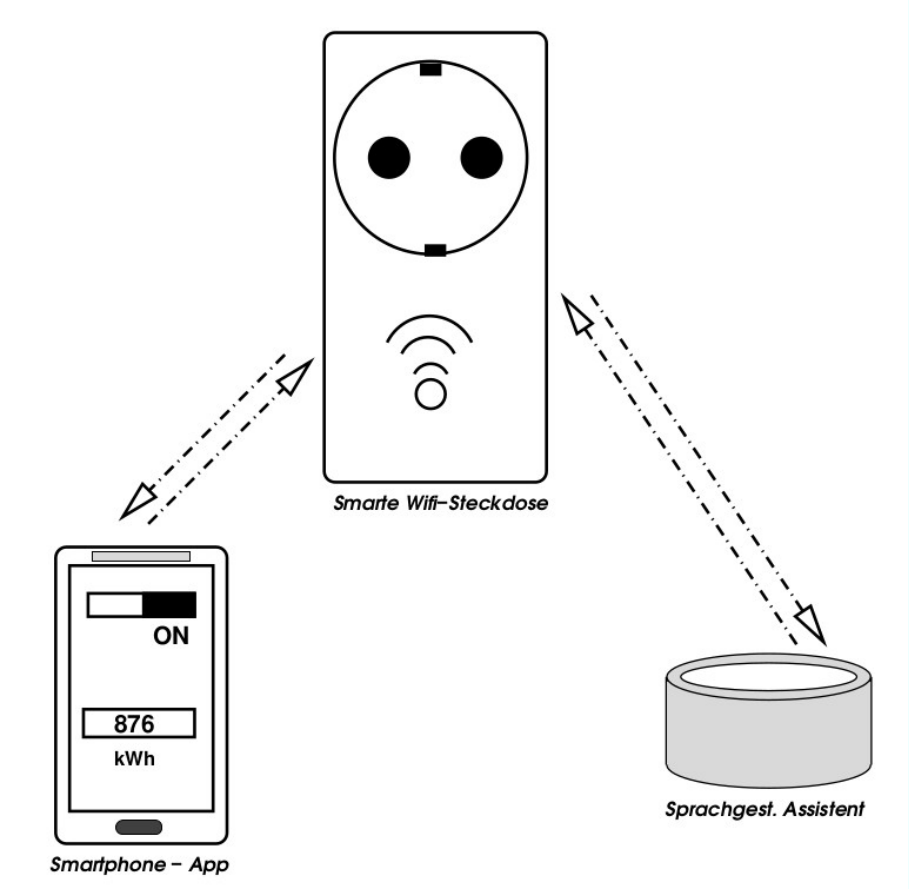
- ✔ Steuerung des Geräts über einfache Protokolle (MQTT oder herstellerspezifische Standards wie Wemo oder Hue Bridge), als Bibliothek für die Firmware verfügbar, auch komplett mit Open Source Software auf beiden Seiten (Microcontroller und Endgerät) möglich.

Beispiel: Wifi Steckdose aus dem Baumarkt

Was will uns der Hersteller mit diesem netten Bild auf der Verpackung sagen?

Bildquelle lt. Verpackung: „Made in PRC“, keine Angaben zum Hersteller, gekauft bei *BI Februar 2019I, Vertrieb durch div. Reseller in EU, CH, IT, PL, RU

User Experience (Die Wahrnehmung)



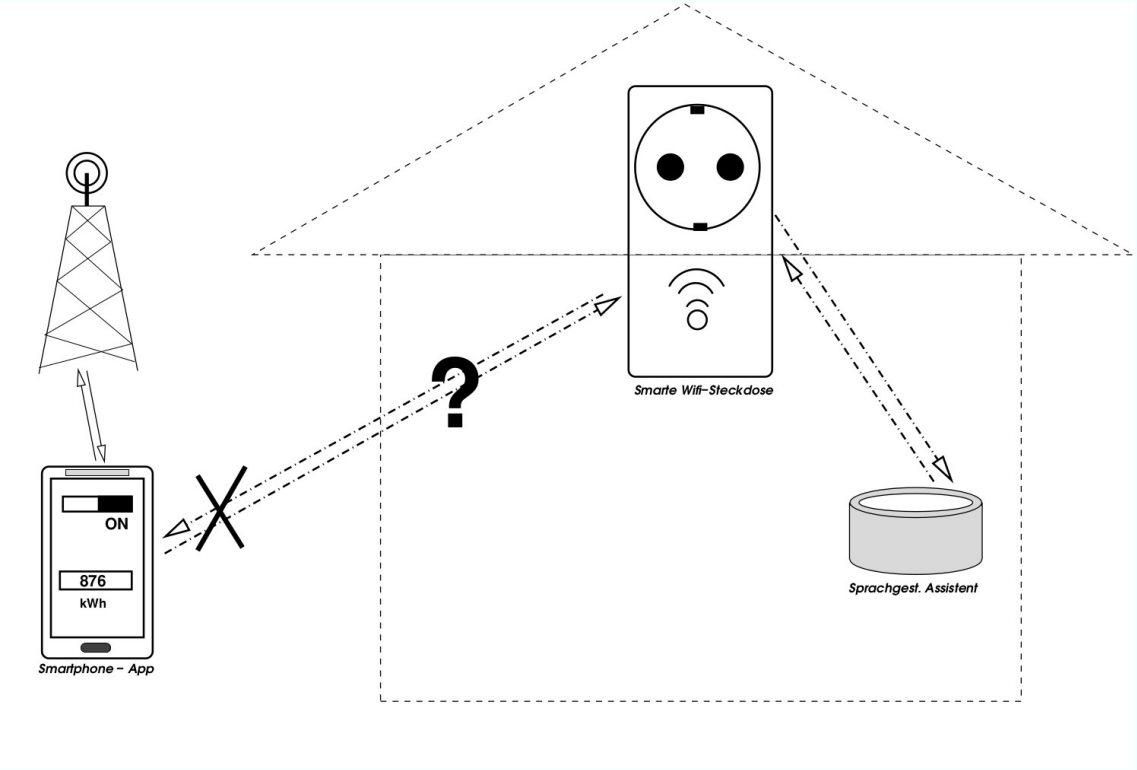
App-Einrichtung „Smarte Wifi Steckdose“ (1)

- ✔ Man soll ein Login und Passwort (beides frei wählbar) angeben - warum?
- ✔ Man soll Nutzungsbedingungen zustimmen.
- ✔ Die App will [unnötig] viele Rechte auf dem Smartphone freigegeben bekommen (Zugriff auf Daten/Bilder/Telefonnummern/…) - ist jedenfalls in allen bisher beobachteten Fällen so.

App-Einrichtung „Smarte Wifi Steckdose“ (2)

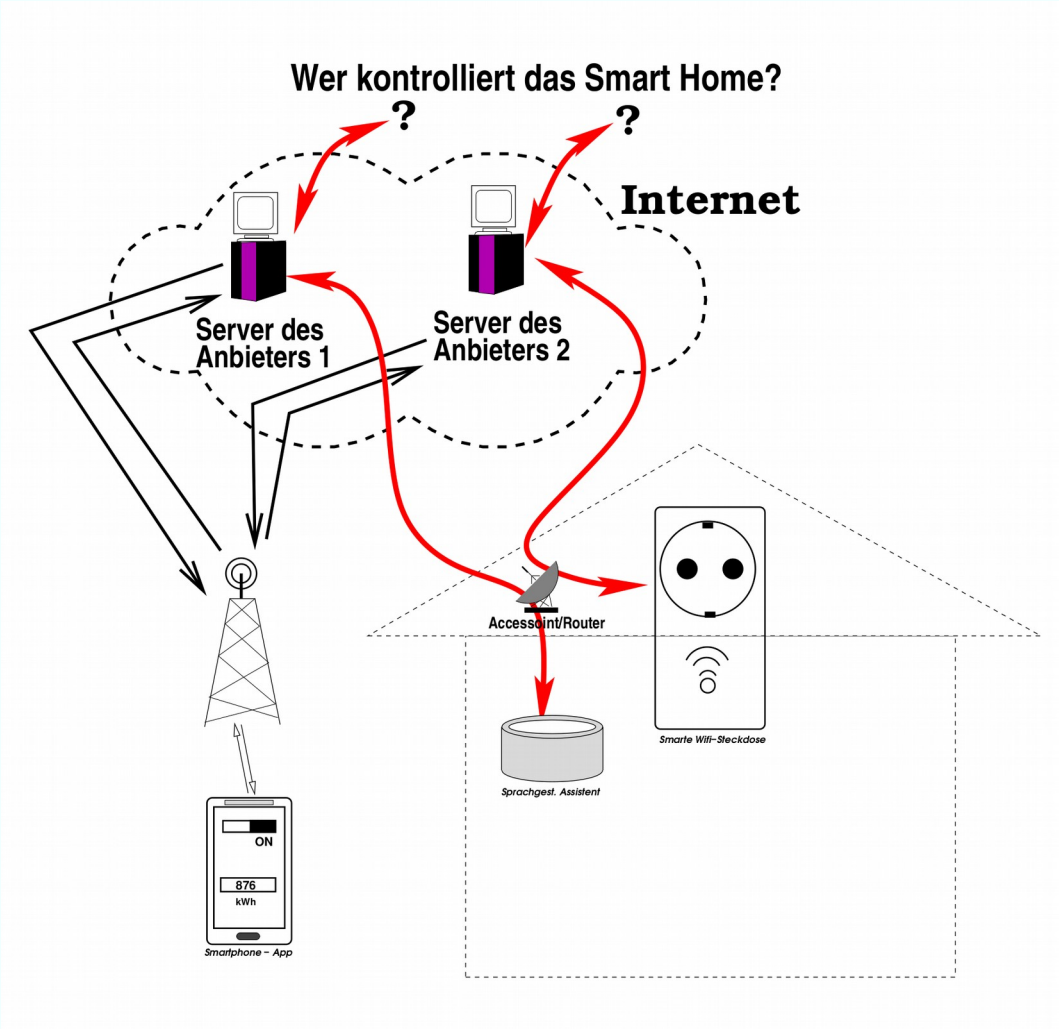
- ✔ App verbindet das Smartphone zunächst mit dem Wifi-AP der Steckdose, lässt dann Eingabe der eigenen WLAN-SSID und Passwort abfragen, anschließend verbindet sich die Steckdose mit dem hauseigenen WLAN und lässt sich per App ein- und ausschalten.
- ✔ Sprachassistent findet die Wifi-Steckdose auch und kann sie per Sprachkommando schalten.
- ✔ Sieht doch gut aus?
- ✔ Wir verlassen das Haus und fahren ein Dorf weiter…

Die Steckdose lässt sich immer noch per App schalten!



Kann praktisch sein, dass das geht, aber wie, wenn man sich doch außerhalb der WLAN-Reichweite befindet?

Die Wahrheit™



Auf dem Foto sah das anders aus...

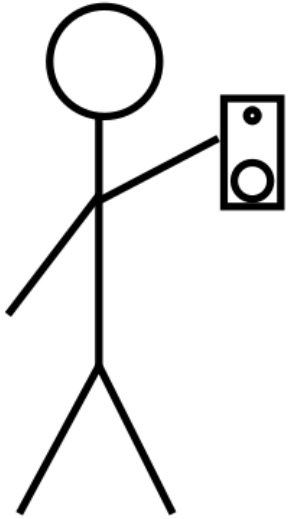
Erkenntnisse

- ✔ Die Steckdose wird von einem (dem Benutzer **unbekannten**) **Computer im Internet** aus gesteuert
- ✔ Die abgefragten Login-Daten sind offenbar die **Zugangsdaten zu diesem Server**, um zu verhindern, dass andere auch die Steckdose schalten können
- ✔ Ist der **Server nicht erreichbar**, kann es sein, dass die Steckdose **nicht funktioniert** (außer es gibt einen zweiten, lokalen Pfad → kommt noch)
- ✔ **Messung:** Die Steckdose bekommt, sobald sie über das heimische WLAN ins Internet gelangt, **ein Firmware-Update** mit dem Benutzer **nicht transparentem Inhalt**.

Das bedeutet…

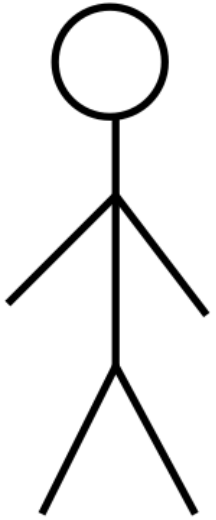
- ✔ Jeder, der Vollzugriff auf den **Server** hat, kann **ALLE Steckdosen ALLER** User **schalten**, oder
- ✔ eine **eigene Firmware** (inkl. **Schadsoftware**) installieren, oder
- ✔ die **WLAN-Passwörter ausspähen**, oder
- ✔ feststellen, wann Schaltvorgänge stattfinden (**wann ist der Benutzer zuhause?**), oder
- ✔ **Alle Steckdosen gleichzeitig ein- oder ausschalten**, oder „**blinken**“ lassen, bis die angeschlossenen **Geräte kaputt** sind, oder …
- ✔ […]

Ich möchte diese Wifi–Steckdose umtauschen, die wird von irgend einem Server aus dem internet ferngesteuert!



ich

Ist doch klar, noch nie vom "Internet of Things" gehört?



Fachverkäufer

Sollte man lieber auf Smart Home Funktionen verzichten?

- ✔ Falsche Frage. :-)

- ✔ Das **Problem** ist NICHT die gewünschte Funktionalität, sondern die **Implementierung** (die für den Hersteller sicher vorteilhaft ist, für den **Anwender aber hohe Risiken** birgt).

- ✔ Erster Lösungsansatz: „**Befreien**“ der Steckdose **aus der Cloud.**
  - ✔ Hierzu ist offensichtlich ein **Austausch der Firmware** notwendig
  - ✔ ...die es für die meisten Wifi-Schaltgeräte auf Basis ESP8266 schon als Open Source gibt: → **Sonoff-Tasmota** https://github.com/arendst/Sonoff-Tasmota

Ein ernstes Wort...

- ✔ Der hier gezeigte Ansatz erfordert fortgeschrittene Kenntnisse im Löten und Verständnis der Elektrotechnik-Grundlagen.
- ✔ Die Steckdose ist ZWINGEND vom Stromnetz zu trennen, so lange sie geöffnet ist bzw. daran gearbeitet wird.

Schritt 1: Öffnen des Gehäuses

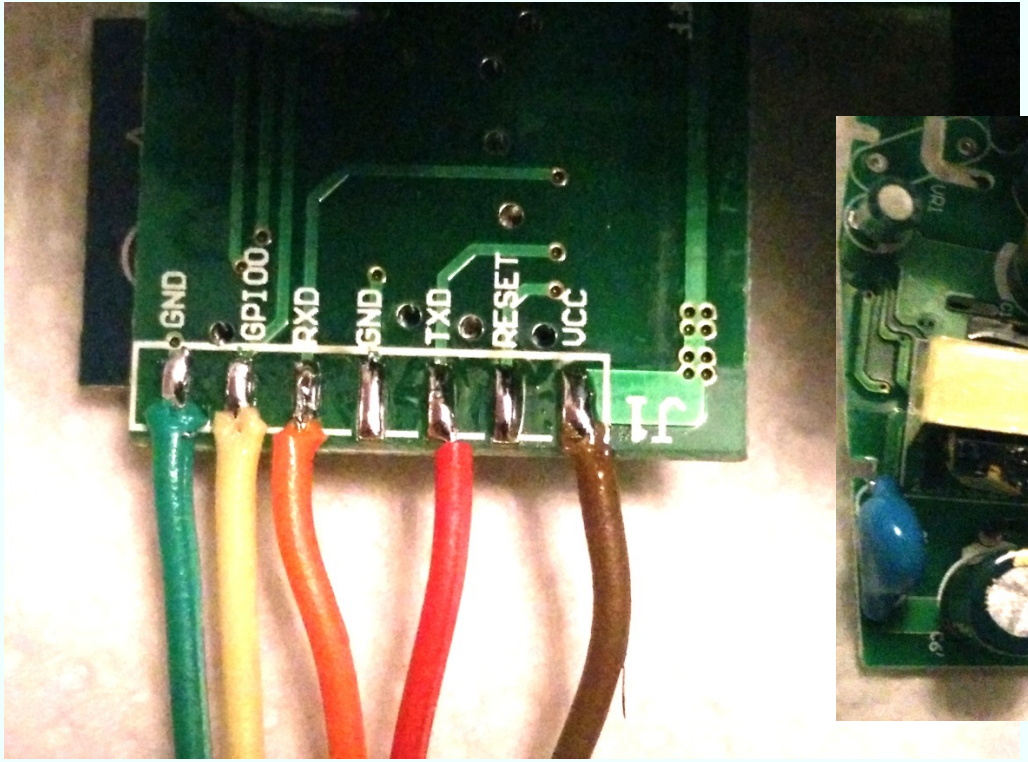
- ✔ … schwierig, da Spezialschrauben verbaut sind, um eben dies zu verhindern. Geht aber mit einem „normalen“ Schraubenzieher, wenn er genau die passende Größe (ca. 3,5mm breit) besitzt. Zum späteren wieder zuschrauben, Spax-Kreuzschlitz-Schrauben 2,5x12 verwenden.

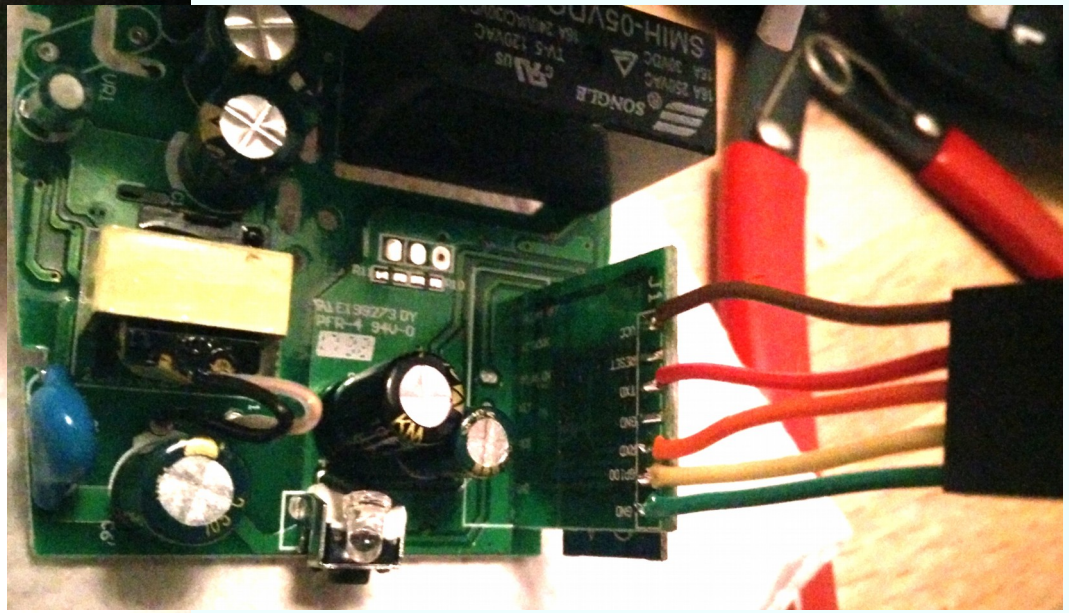
- ✔ Die Platine mit dem ESP8266-Board ist ebenfalls mit 1-3 kleinen (normalen) Kreuzschrauben befestigt.

Schritt 2: Anlöten von Kabeln für USB-serial TTL Adapter

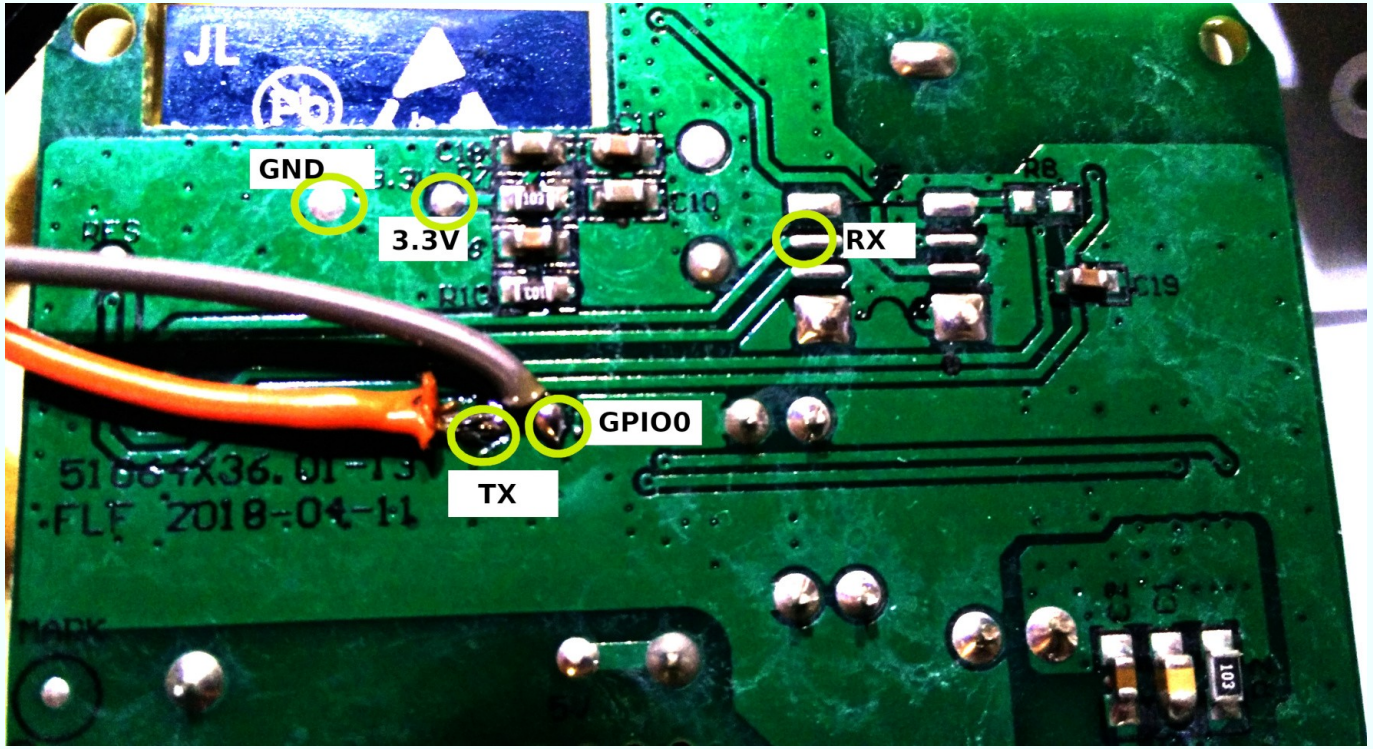
- ✔ 3,3V (!!!) Stromversorgung, NICHT 5V (entsprechenden TTL-Adapter mit 3,3V Ausgang besorgen),
- ✔ GND (Masse)
- ✔ RX (Receive) → wird mit TX auf dem Board verbunden
- ✔ TX (Transmit) → Wird mit RX auf dem Board verbunden
- ✔ GPIO 0 („Flash Modus“)

Bilder: *BI Socket Modell 1

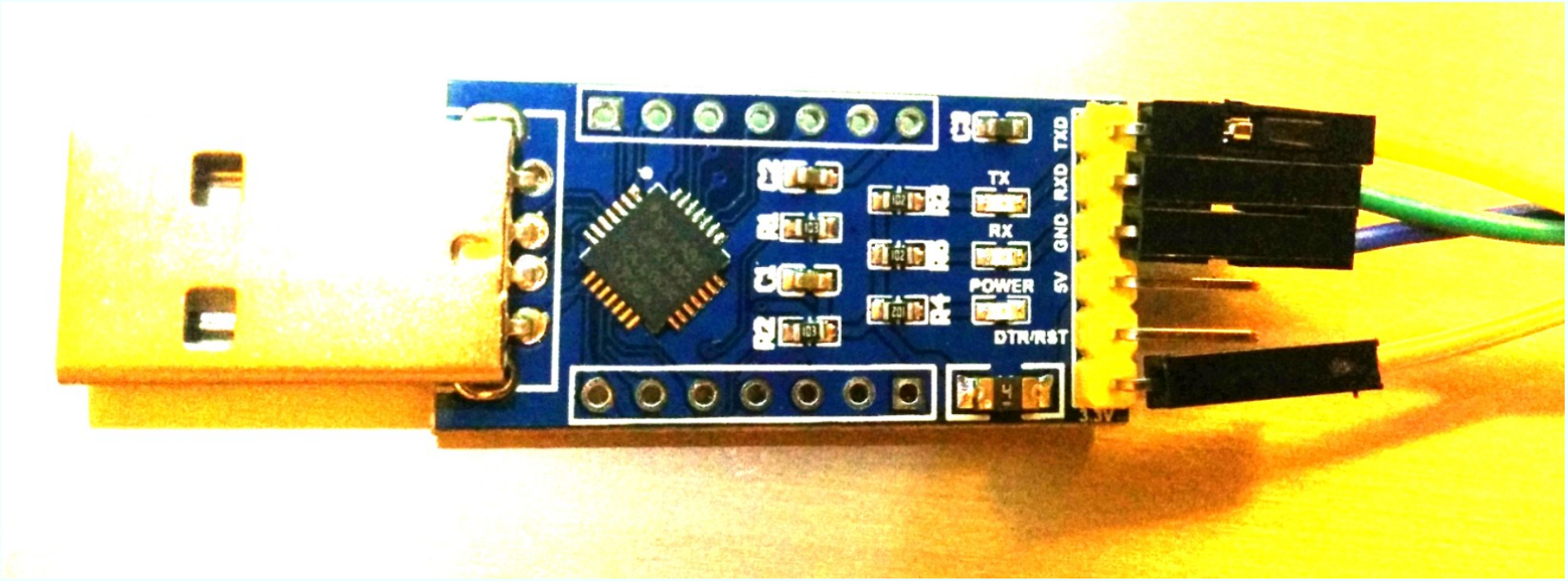




Bilder: *BI Socket Modell 2



TTL USB Stecker



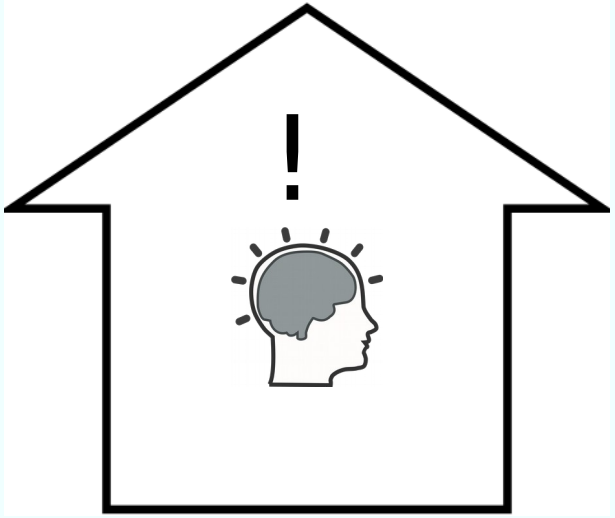
Flash! (Demo)

- ✔ ...mit Arduino-Umgebung (erlaubt auch das Recompilieren, oder
- ✔ mit Python-Skript → /dev/ttyUSB0
- ✔ Frisch geflasht geht das Gerät in den Accesspoint-Modus und lässt sich zunächst netzwerkmäßig konfigurieren, nach Neustart auch alle gewünschten Protokolle & Features inkl. Anbindung an Sprachassistenten (wenn man dies will)
- ✔ Firmware-Update mittels „intermediate minimal Firmware“, falls es einmal notwendig sein sollte

Sprachassistent ohne Cloud?

- ✔ Nicht so umfangreich (lokale Spracherkennung ohne Internet-Anbindung), aber möglich → Vielversprechende Projekte fürs nächste Semester, z.B. snips.ai
- ✔ Training mit einfachen Sprachbefehlen und festgelegter Syntax
- ✔ Keine externe Datenspeicherung
- ✔ Ggf. genügt ein Raspberry Pi 3B+, Anbindung aller Geräte per MQTT (mosquitto)

Feedback-Schleife



Prof. Dipl.-Ing. Klaus Knopper
<ki@knoppix.info>

https://knopper.net/tuebix/