



Hey Siri – Hallo Jasper

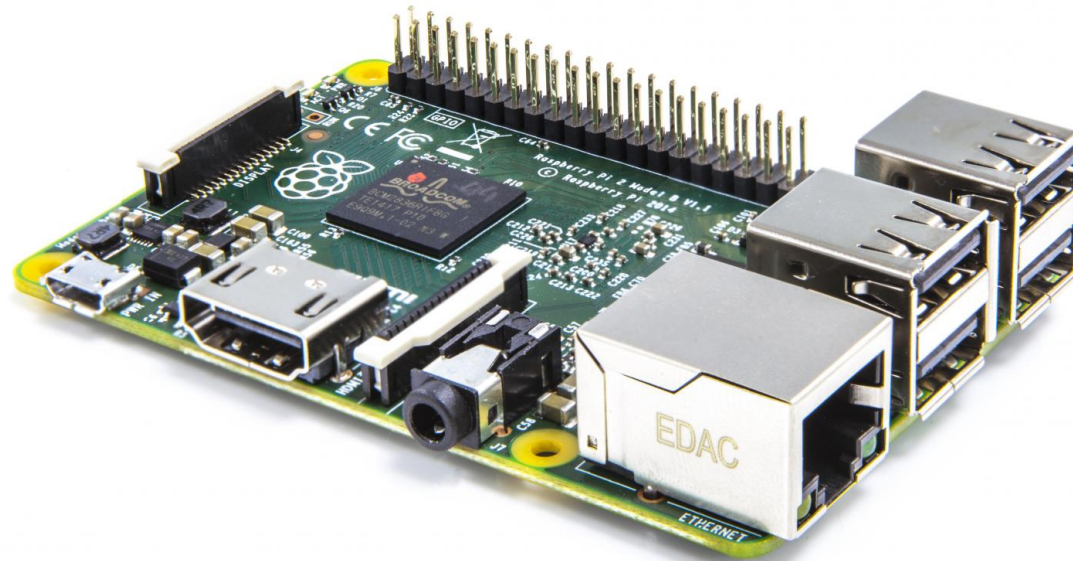
Eike Holtz – Infratix GmbH
Chemnitzer Linuxtage 2017

Jasper

- Open Source Plattform
- Spracherkennung
- Python
- IoT / Always On / Cloud based / Weitere Buzzwörter hier einfügen

Raspberry Pi

- Gibt es jemanden im Publikum, der ihn nicht kennt?



The easy part

- Installation von Jasper auf Raspberry Pi
- Webseite <http://jasperproject.github.io/>
 - Alle Anleitungen dort nicht benutzen!
- Webseite <https://groups.google.com/forum/#!forum/jasper-support-forum>
 - Mit deutlich aktuelleren Informationen

The easy part (II)

- In den Tiefen des Forums suchen
- Dort den Link finden:

https://www.amazon.com/clouddrive/share/XLewxYm20h6antbhXFDfaAP7aeZaCpD0K3VY0zExwR4?_encoding=UTF8&mgh=1&ref_=cd_ph_share_link_copy

- Das aktuelle Image herunterladen
- Mittels dd auf eine 4GB+ große microSD-Karte bannen

The easy part (III)

- Weiter in den Tiefen des Forums suchen
- Dort den Link finden:

<https://github.com/mattcurrycom/Documentation/blob/master/jasper/jasper-client/Jasper-Post-Image-Setup.md>

- Erstkonfiguration nach der dortigen Anleitung
 - Timezone, Ort, Name, Mailadresse, ...

The easy part (IV)

```
$ cat ~/.jasper/profile.yml
carrier: PHONE_CARRIER
first_name: Eike
gmail_address: MAIL_ADRESS
gmail_password: MAIL_PASSWORD
last_name: Holtz
location: 'Berlin'
phone_number: 'PHONE_NM'
prefers_email: true
timezone: Europe/Berlin
```

```
tts_engine: mary-tts
mary-tts:
  server: 'tts.mattcurry.com'
  port: '59125'
  language: 'en_US'
  voice: 'cmu-slt-hsmm'

stt_engine: witai
witai-stt:
  access_token: SERVER_TOKEN
```

The hard part

- Spracherkennung ist nicht einfach
- Diverse Engines in Jasper enthalten
- Lokal vs Internet

wit.ai

- Wit.ai ist die default Erkennungseengine in Jasper
- Internet Based
- Registrierung erforderlich
- Kann auch deutsch
- Ganze Sätze

wit.ai benutzen

- <https://wit.ai>
- Github oder Facebook Account notwendig
- Projekt erstellen
 - Server Access Token unter Settings abrufen
 - Unter Settings => Danger Zone => Make Private
- Später: Sprache auf Deutsch umstellen

pocketsphinx

- Lokale Lösung
- Erkennungsrate ist nicht so besonders
- Kann theoretisch auch Deutsch
 - Habe ich aber nicht zum laufen bekommen

pocketsphinx vorbereiten

```
# Vorbereitungen

sudo swapoff -a
sudo apt -y install pocketsphinx

# auf keinen Fall neuere Versionen von pocketsphinx via pip
# besorgen

wget https://storage.googleapis.com/google-code-archive-downloads/v2/code.google.com/phonetisaurus/is2013-conversion.tgz
tar xvzf is2013-conversion.tgz
cd is2013-conversion/phonetisaurus/src
make

# Kaffee trinken gehen

cd ../../
sudo cp bin/phonetisaurus-g2p /usr/local/bin/
```

pocketsphinx konfigurieren

```
# profile.yml

#stt_engine: witai
#witai-stt:
#  access_token: DIRZ2WI76VPCJV4HIVPESCJLZQMUTZYM

# Einrückung beachten!

stt_engine: sphinx
pocketsphinx:
  fst_model: '/home/pi/g014b2b/g014b2b.fst'
  hmm_dir: '/usr/share/pocketsphinx/model/hmm/en_US/hub4wsj_sc_8k'
```

Mikrofonsetup

- Da werden eigene Vorträge drüber gemacht
- Start mit 5€ USB-Mikro von ebay
- Weitere Tests mit Headset
- Aktuell im Test Zoom H1
 - Ergebnisse vielversprechend
 - Kostenpunkt ~100 €

Jasper spricht

- Jasper gibt Feedback per Audio
- High beep, low beep
- Texte (alle in englisch)
- Kleine Lautsprecher sinnvoll

mary-tts

- Default Text-To-Speech engine
- Qualitativ okay
- Vorkonfiguriert
- Deutsch möglich
- Internet Based
 - Server ist in der Hand vom Haupt-Maintainer von Jasper

espeak

- Lokale Lösung
- Klingt wie Techno aus den 90ern
- Kann nur englisch einigermaßen

espeak

```
# profile.yml

#tts_engine: mary-tts
#mary-tts:
#  server: 'tts.mattcurry.com'
#  port: '59125'
#  language: 'en-US'
#  voice: 'cmu-slt-hsmm'

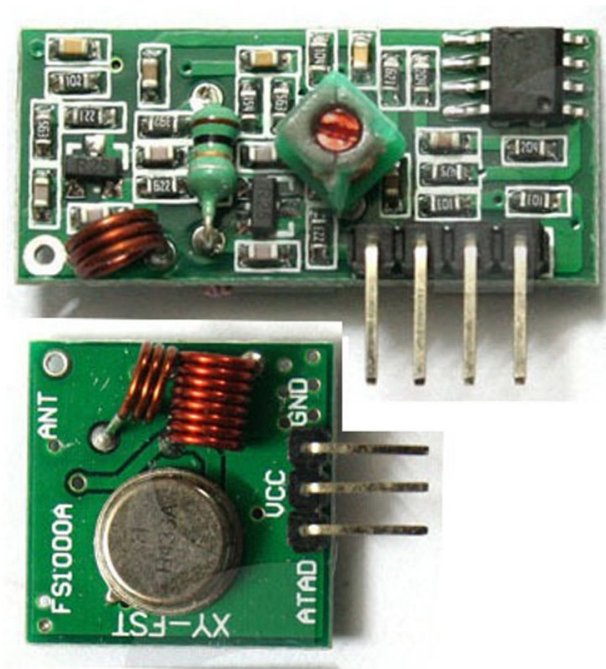
tts_engine: espeak
```

The really easy part

- Anbindung von 433Mhz Funksteckdosen
- Ich war total überrascht
- <https://www.einplatinencomputer.com/raspberry-pi-433-mhz-funksteckdose-schalten/>

The really easy part (II)

- 433 Mhz Sende/Empfangsmodul ~ 5€
- Funksteckdosenset



The really easy part (III)



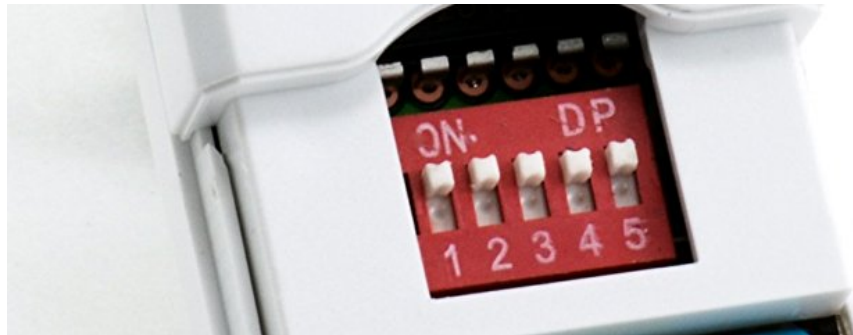
3v3 Power	1		2	5v Power
BCM 2 (SDA)	3		4	5v Power
BCM 3 (SCL)	5		6	Ground
BCM 4 (GPCLK0)	7		8	BCM 14 (TXD)
Ground	9		10	BCM 15 (RXD)
BCM 17	11		12	BCM 18 (PWM0)
BCM 27	13		14	Ground
BCM 22	15		16	BCM 23
3v3 Power	17		18	BCM 24
BCM 10 (MOSI)	19		20	Ground
BCM 9 (MISO)	21		22	BCM 25
BCM 11 (SCLK)	23		24	BCM 8 (CE0)
Ground	25		26	BCM 7 (CE1)

The really easy part (IV)

```
git clone git://git.drogon.net/wiringPi  
cd wiringPi  
sudo ./build
```

```
git clone git://github.com/xkonni/raspberry-remote.git  
cd raspberry-remote  
make send
```

```
# send <dipcode> <steckdosennr> <1/0>  
sudo ./send 11111 2 1
```



Jasper und Funkdosen koppeln

- Modul für Jasper schreiben
- Python
- <http://jasperproject.github.io/documentation/api/standard/>

client/modules/Light.py

```
import re
from subprocess import call
import pickle
import os.path

WORDS = ["SCHALTEN", "LICHT", "AN", "AUS"]

def isValid(text):
    return bool(re.search(r'\blicht\b', text, re.IGNORECASE))
```


client/modules/Light.py

```
def handle(text, mic, profile):
    fname="/dev/shm/light.pickle"
    if os.path.isfile(fname):
        lightstate=pickle.load( open( fname, "rb" ) )
    else:
        lightstate=0

    if lightstate == 0:
        lightstate=1
    else:
        lightstate=0

    pickle.dump( lightstate, open( fname, "wb" ) )
    call(["sudo", "/bin/send", "11111", "1", str(lightstate)])
    time.sleep(2)
    call(["sudo", "/bin/send", "11111", "1", str(lightstate)])
```

Jasper lokalisieren

```
# Namen in Karl ändern
sed -i "s/JASPER/KARL/g" client/local_mic.py
sed -i "s/JASPER/KARL/g" client/modules/MPDControl.py
sed -i "s/JASPER/KARL/g" jasper.py

# Begrüßung ändern
sed -i "s/How can I be of service/Stehe zu diensten/g" jasper.py
```

mary-tts auf deutsch

```
# profile.yml

tts_engine: mary-tts
mary-tts:
  server: 'tts.mattcurry.com'
  port: '59125'
  language: 'de'
  voice: 'dfki-stefan-hsmm'
```

Jasper interaktiv starten

```
~/jasper-client $ ./jasper.py
*****
*                JASPER - THE TALKING COMPUTER                *
* (c) 2015 Shubhro Saha, Charlie Marsh & Jan Holthuis        *
*****

No disturbance detected
INFO:client.stt:Transcribed: [u'HALLO KARL']
INFO:client.stt:Transcribed: [u'LICHT ANSCHALTEN']
using pin 0
sending systemCode[11111] unitCode[1] command[1]
INFO:client.stt:Transcribed: []
```

Fragen?

- <https://eholtz.de>
- clt2017@eholtz.de