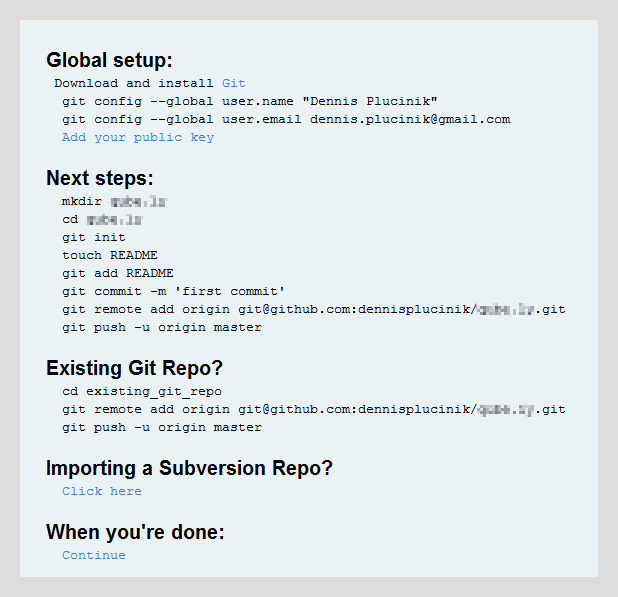
# 8.10.2014

Wallner.: Upload von dem Pflichtenheft auf Github, mittels des Git Bash Programms.

Danner, Heiss:Recherche zur Raspberryprogrammierung

Guide zum Upload von Dateien mittels Git Bash auf Github

http://www.srf.ch/wissen/digital/billig-pc-raspberry-pi-die-anleitung-fuer-einsteiger

[Set up your public key (http://help.github.com/win-set-up-git/#\_set\_up\_ssh\_keys)](http://help.github.com/win-set-up-git/#_set_up_ssh_keys)

You will not be able to connect without doing this first.

* set up some kind of config

git config --global user.name "Dennis Plucinik"

git config --global user.email dennis.plucinik@gmail.com

* Create a folder where you want your repository to live

For me (on Windows 7), GitBash opened up on c:\Users\Dennis so I had to navigate a bit to get to the right folder...

cd Dropbox/Clients/Current

Then we follow GitHub's lead

mkdir {projectName}

cd {projectName}

* Initialize repo

git init

If successful, you should see a comment "Initialized empty Git repository in c:/{wherever}/.git/

* Type in a bunch of other stuff

Apparently we are touching, adding, committing, remote adding, and pushing...

touch README

git add README

git commit -m 'first commit'

git remote add origin git@github.com:dennisplucinik/{projectName}.git

git push -u origin master

Hopefully I figure out what all this is for.

At this point I actually couldn't continue because the last command (git push -u ...) gave me an error about my account billing :\ I'll pick up on this when that gets cleared up later.

...the evening passes...

Ok, I checked this morning and my account is authenticated just fine. So after getting README to send up properly, open Explorer and copy your project files from wherever they were previously into the new folder containing your local .git repo. Now try this:

git add .

git commit -m 'adding entire project'

git push -u origin master

You should see your files going up along with a success message. Truthfully by this time I started monkeying around with setting up the EGit plugin in Eclipse. I had some issues with setting ssh paths correctly but I think I'll be writing another post to cover that soon.

UPDATE: [Configuring the EGit Plugin for Eclipse](http://www.dennisplucinik.com/blog/2011/03/25/configuring-the-egit-plugin-for-eclipse/).

Pinbelegung:

<http://raspberrypiguide.de/>

#make persistant

#Ls (zeigt die befehle die wir brauchen)

# 15.10.2014

Heiss, Wallner: Programmieranfänge mit raspberry (lampe über GPIO Ausgänge ansteuern)

Mit der Sprachsteuerungssoftware Jasper befasst.

Danner: Pflichtenheft Verbesserung und Bericht Erstellung

# 5.11.2014

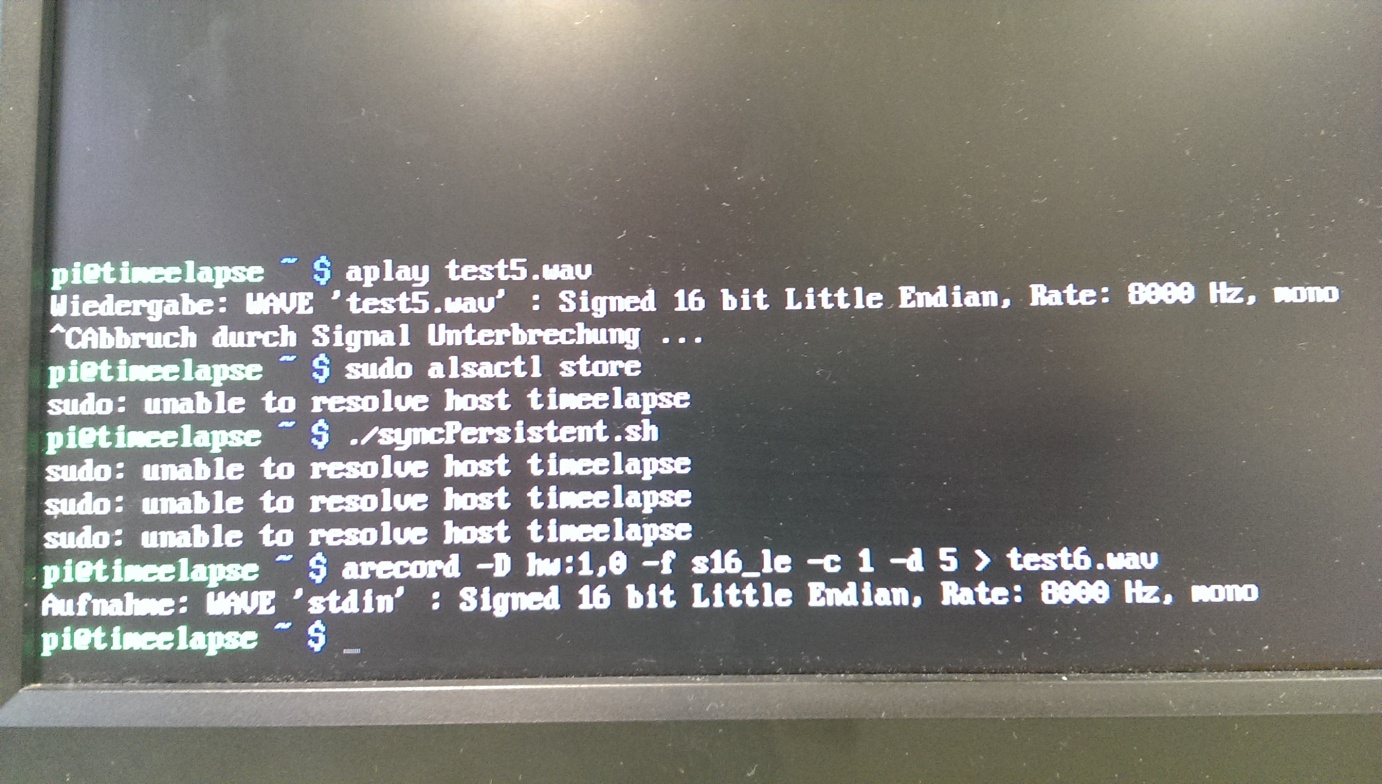
Anwesenheit: Heiss Tobias

Lukas Wallner

Fehlende: Danner Michael

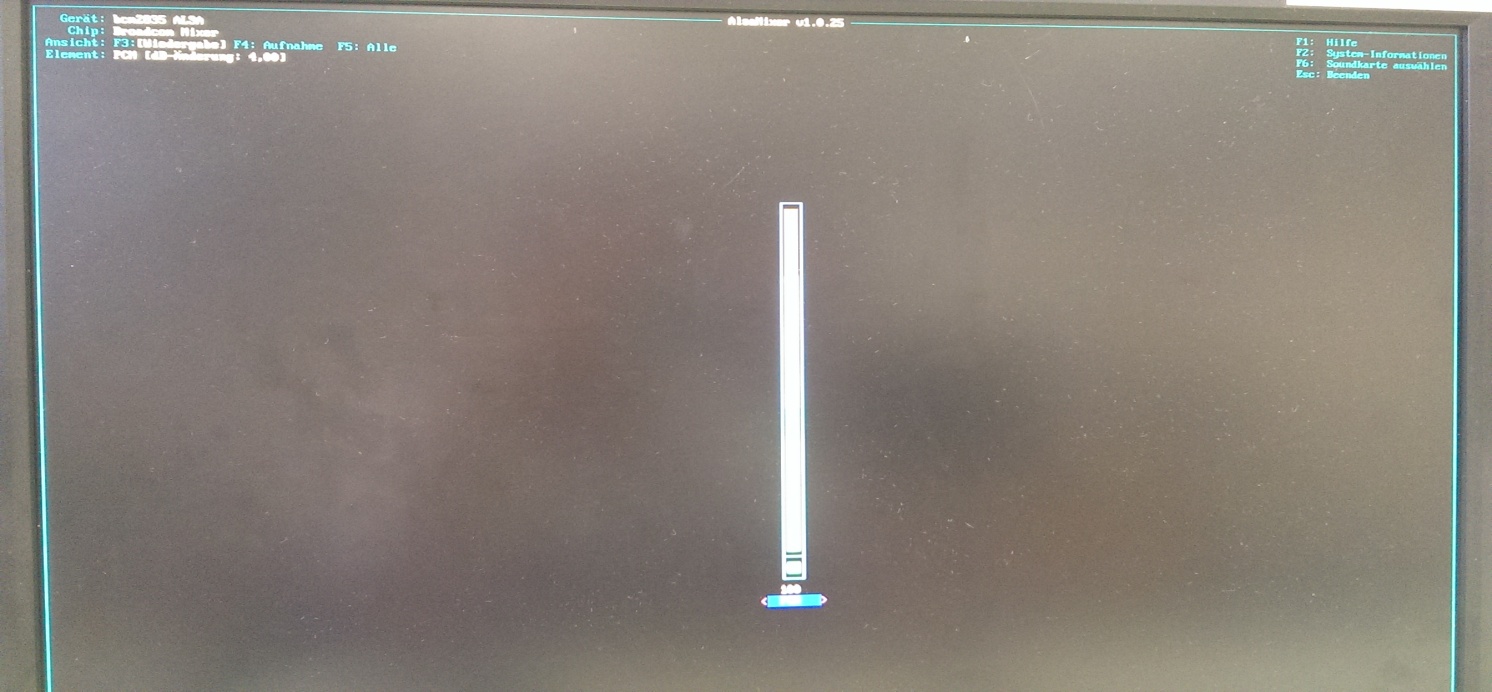
In der letzten Stunde wurde der Raspberry so konfiguriert, dass Audio-Aufnahmen über ein USB-Mikrofon gemacht werden können.

Wie im unteren Bild zu sehen ist, wurden einige Aufnahmen gemacht, da die Einstellungen (Lautstärke war zu niedrig, sodass man nichts Hören konnte) nicht gestimmt hatten.

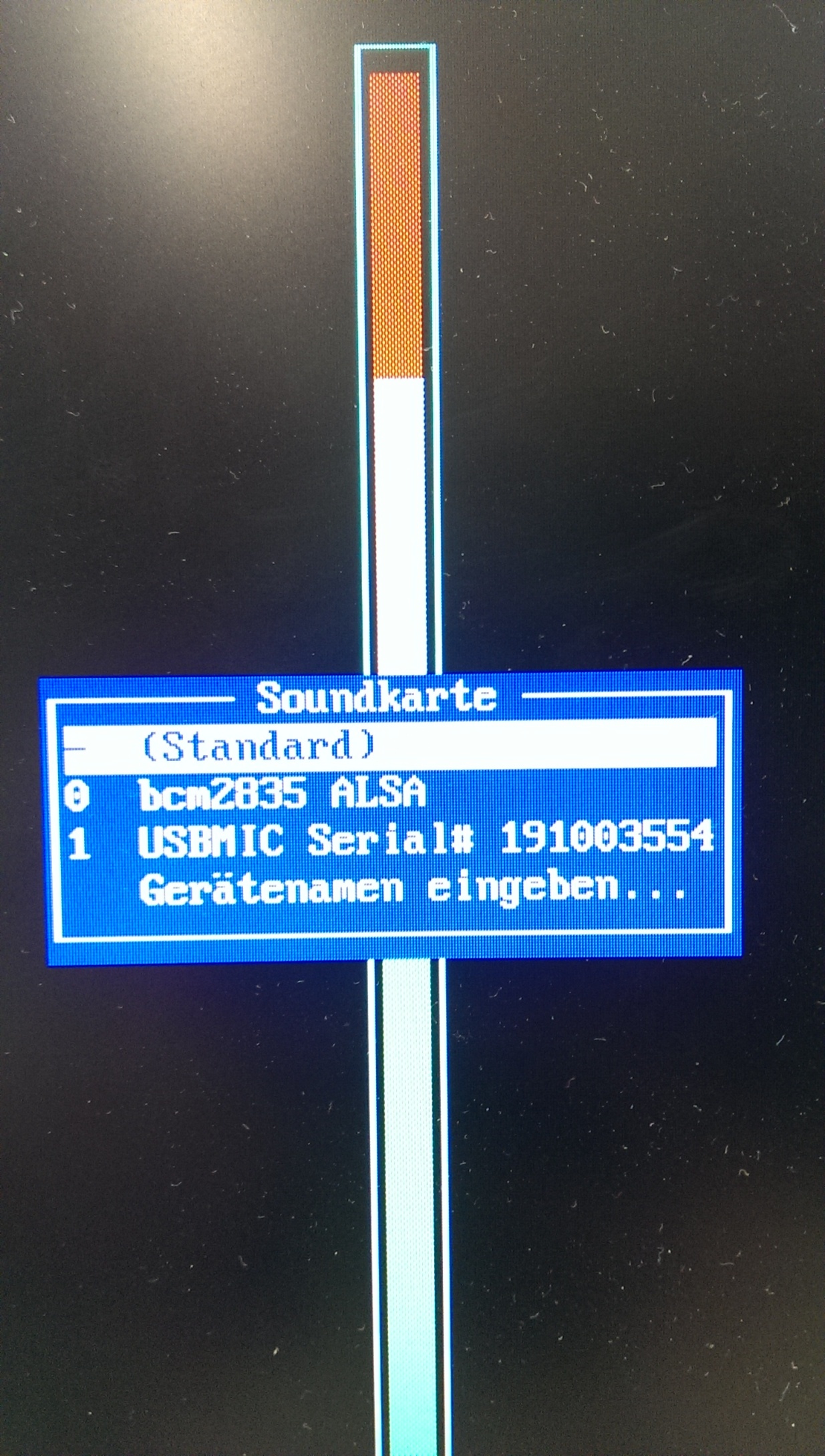


Anschließend wurden die DB Einstellungen im alsamixer angepasst.

Der alsamixer sieht so aus wie auf der unteren Abbildung zu sehen ist:



Dort wurden im Soundkarten Menü die Lautstärke der Kopfhöher und die Lautstärke des Mikrofons erhöht.



Als zweiten Teil der PPM-Einheit wurde das WLAN eingestellt, so dass über Hotspot auch vom Laptop auf den Raspberry zugegriffen werden kann.

Als letzten Teil der Übung wurde an der Jasper Installation weiter gearbeitet (Installieren von Pocketshpinx, …)

# Bericht vom 12.11.2014

Anwesenheit: Alle anwesend

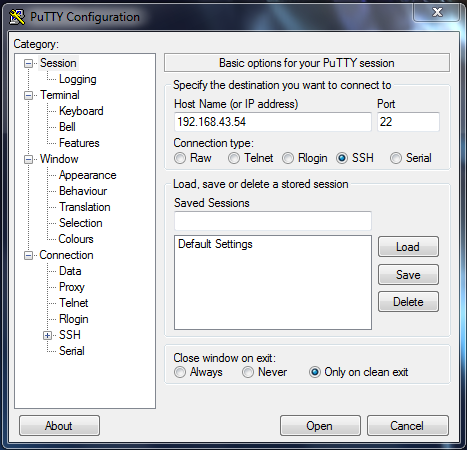
Arbeit:

Danner: Recherche über Python und .txt Dateien und SSH-Verbindung mit Mac-book

Wallner: Installation von M2M-Aligner,Phonetisaurus und Mitlm auf dem Raspberry

Heiss: Errichten der WLAN-Verbindung für SSH-Verbindungen und Raspberry Konfiguration

Über Putty:



# 

# 3.12.2014

Fehlende: Wallner

Heiss: Ausbessern der Fehler im Python Programm

Anschließen des Raspberrys an die Schaltung.

Steuern der Ausgänge mit Hilfe des Python Programms, und einer Textdatei

# 10.12.2014

Alle Anwesend

Heiss, Wallner:

Installation des voicecommand Programms

GPIO-Ausgänge mit Hilfe der Raspberry Befehle ansteuern: funktioniert nicht!!!

GIT-Downloads auf Raspberry möglich machen

# 17.12.2014

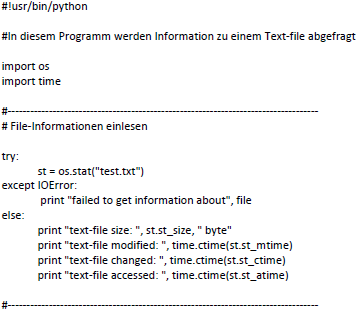
# Recherche über Informationen in einer Datei abfragen

Mit der Bibliothek os können alle wichtigen Informationen über ein File ganz einfach abgefragt werden.

<http://effbot.org/zone/python-fileinfo.htm>

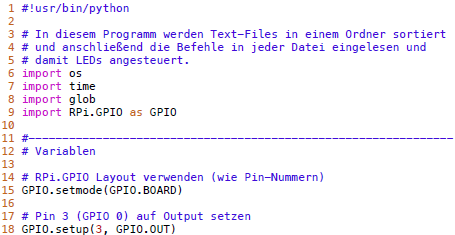
<http://stackoverflow.com/questions/168409/how-do-you-get-a-directory-listing-sorted-by-creation-date-in-python>

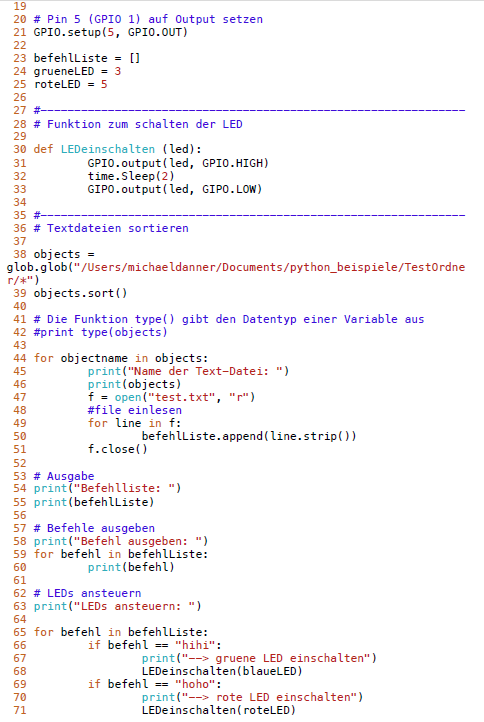
## Programm:



# 7.01.2015 + 14.01.2015

Fertigstellung des ersten gesamt Programms und Tests





## Kopie des Programms auf das Mac-Book

Michaels-MacBook-Air:Python\_Beispiele michaeldanner$ scp

pro\_LEDSteuerung02.py pi@192.168.1.227:

The authenticity of host '192.168.1.227 (192.168.1.227)' can't be

established.

RSA key fingerprint is

97:cd:86:c9:57:84:f9:be:8c:16:9a:6e:07:34:72:6f.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '192.168.1.227' (RSA) to the list of

known hosts.

pi@192.168.1.227's password:

pro\_LEDSteuerung02.py

100% 1661 1.6KB/s 00:00

Michaels-MacBook-Air:Python\_Beispiele michaeldanner$

# 21.01.2015

Abwesend: Danner Michael

## Fertigstellung der Sprachsteuerung

Programm so umgeschrieben, dass nach Aufnahme direkt an Google API geschickt wird, anschließend die Ergebnisse in eine txt-Datei geschrieben wird und danach am Monitor mit cat ausgegeben werden.

Dann wird die .flac-Datei gelöscht und die Ergebnisse als json in die txt-Datei gespeichert.

# 28.1.2015

## Links und Google Speech API Informationen

<http://blog.oscarliang.net/raspberry--‐pi--‐voice--‐recognition--‐works--‐like--‐siri/>

## <http://www.spech.de/blog/article/spracherkennung--‐mit--‐google--‐speech--‐api>

* API KEY: AIzaSyDlljDLpEnWm9NYZipr3fe14wcYHJnt8uk
* Nur 50 Anfragen pro Tag
* Link:

https://www.google.com/speech--‐api/v1/recognize

?output=json&client=luggi.lw@gmail.com&lang=en--‐us&key=

AIzaSyDlljDLpEnWm9NYZipr3fe14wcYHJnt8uk

# 04.02.2015

## Testen des wget-Befehls und Übung mit Json

Michaels-MacBook-Air:Audio michaeldanner$ wget -q -U "Mozilla/5.0" -

-post-file mynameislukastest.flac --header "Content-Type: audio/xflac;

rate=44100" -O - "https://www.google.com/speechapi/

v2/recognize?output=json&lang=enus&

key=AIzaSyB3CZWmazW6XL1hPCv9Y5RWLmeyhBj5aoY"

{"result":[]}

{"result":[{"alternative":[{"transcript":"hello my name is

Lucas","confidence":0.97492468},{"transcript":"hello my name is

Lukas"},{"transcript":"hello My name is Lucas"},{"transcript":"hello

my name is Nicholas"},{"transcript":"hello my name is

locust"}],"final":true}],"result\_index":0}

Michaels-MacBook-Air:Audio michaeldanner$

# 18.02.2015

## Aufbau der Speech API und Json Ergebnisse auswerten

Die Google Speech API liefert JSON-Ergebnisse und diese müssen in normale Text-Dateien umgeändert werden.

Ebenfalls wurde der Code der speech2text.sh geändert, sodass wir bestimmen können wie unsere .flac-Dateien aussehen sollen und müssen zusätzlich noch 5 Sekunden warten, bevor wir ins Mikrofon sprechen können.

# 25.02.2015

Fehlende: Wallner

Das Json-Programm wurde nochmals ausprogrammiert.

Funktionierte mit dem Übersetzen aus einem String ganz gut, allerdings muss das Hauptprogramm noch hinzugefügt werden und beide Programme gekoppelt werden.

# 04.03.2015

Koppelung des Json-Programms mit dem speech2text Programm.

In beiden Programmen debuggen.

Anschließen die fertigen Programme auf den Raspberry übertragen.

Testdurchläufe mit dem „fertigen Programm“. Leider kein Steckbrett vorhanden, weshalb die eigentliche Funktion des Programms nicht getestet werden kann.

# 18.03.2015

Es wurde das Python-Json Programm erneut umgeändert.

Die Schaltung wurde entsprechend an den Raspberry angehängt und dementsprechend getestet.

Am Ende der Einheiten ein Erfolg beim Wort green.