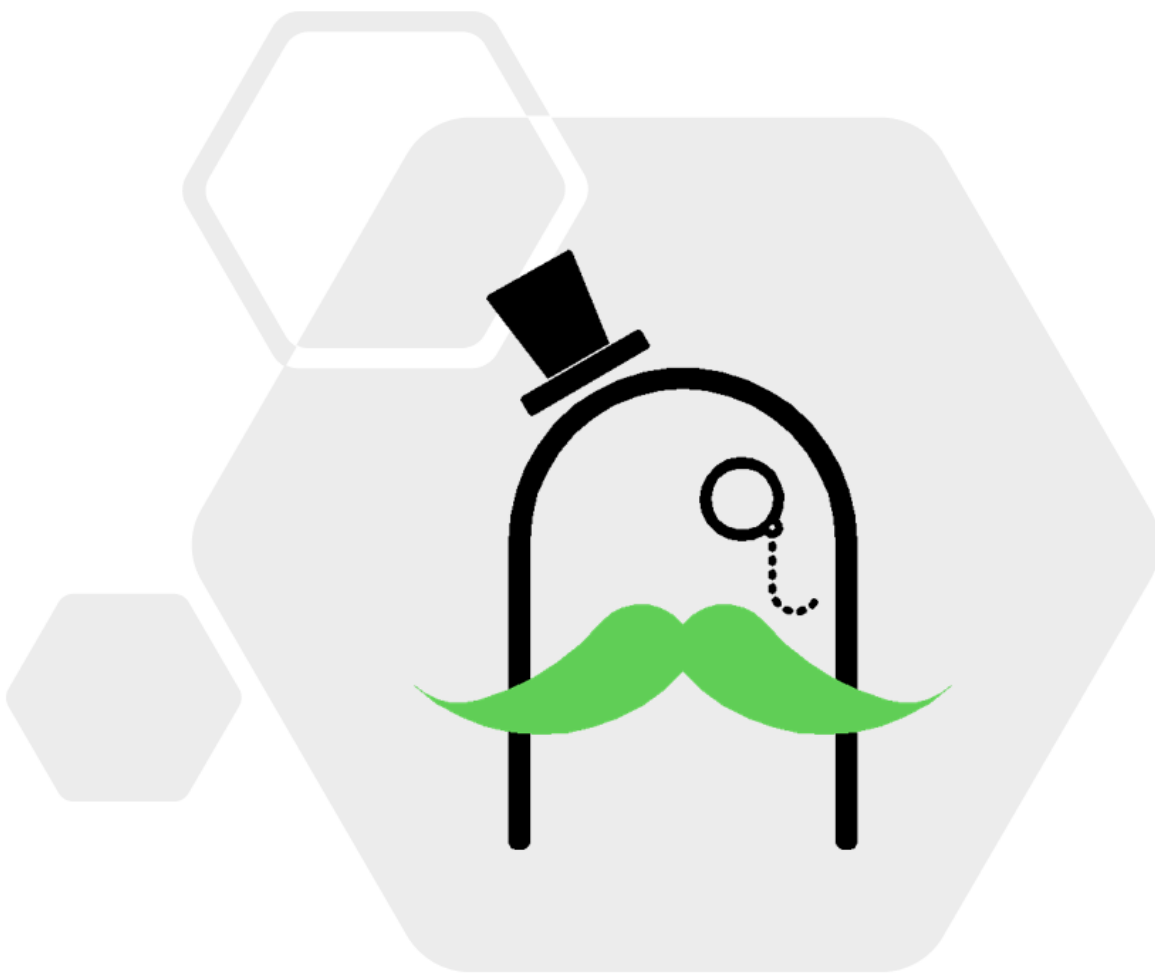


Dokumentation

Projekt Alfred - V1.0



Projektteam:

Stefan Dimnik

Saskia Willixhofer

Ladislav Zechmeister

17.09.2020



Inhalt

Hardware.....	4
Elektronik.....	4
3D-Druck.....	4
Software	5
Code.....	5
Alf_Main.py	5
Alf_GUI.py.....	5
Alf_Speech.py.....	5
Alf_ToolConnect.py	6
Alf_Verfolgung.py.....	6
Weitere Module	6
Sprachsteuerung/Spracheingabe	6
Emotionen	6
Probleme	6
Programm in mehreren Sessions ausführen	6
Raspberry einrichten	8
Raspbian installieren	8
Alfred von Github herunterladen.....	9
Benötigte Software.....	10
Serial Port ansteuern (Python Library)	10
Wiringpi – GPIO ansteuern (Python Library)	10
BME280 Temperatur Sensor (Python Library)	10
Jarvis (Sprachsteuerung)	10
Optionale Software	12
Matchbox-keyboard (Touchscreen/Onscreen Tastatur) - optional	12
Samba Server (Dateifreigabe) - optional.....	12
Software für Alfred Webinterface (in der aktuellen Version noch nicht verfügbar).....	13
Apache2 (Webserver).....	13
PHP	13
Raspberry Konfiguration	14
Bildschirm um 180° drehen:.....	14
Bootscreen ändern:.....	14
Default Audio Device ändern:	14
Init-Script erstellen	15
Desktop und Startmenü Verknüpfung	16



Probleme	18
Berechtigungen	18
Python Tkinter GUI über SSH starten	18



Hardware

Elektronik

Raspberry 3 B+

<https://www.rasppishop.de/Raspberry-Pi-3-Modell-B-Starterkit-Schwarz>

32GB SD

<https://www.rasppishop.de/Sandisk-32-GB-Noobs>

7 Zoll Raspberry Touch Screen

<https://www.rasppishop.de/Raspberry-Pi-7-Touchscreen-Display>

3D-Druck



Software

Für die Softwareentwicklung wurde ein github Repository angelegt um die Zusammenarbeit zu erleichtern und eine Versionierung zu ermöglichen.

Das Programm ist zurzeit noch nicht öffentlich.

<https://github.com/SDim44/Alfred>

Code

Alf_Main.py

Alf_Main.py wird in einer eigenen Session beim Programmstart ausgeführt.

Alfred verfügt über Modi, welche beliebig erweitert werden können.

Diese Modis werden über das Main Programm geschaltet und können mittels Touchscreen oder Sprachbefehl eingestellt werden.

Jeder Modi kann je nach Bedarf auf alle Funktionen zugreifen:

Modus 1: Objektverfolgung

=> Alfred verfolgt ein vorgegebenes Objekt das durch einen „Vision Sensor“ (Pixy2 Cam) erkannt wird. Wird ein Hindernis mittels Ultraschall Sensoren erkannt, weicht er mit einer Drehung aus. Sobald das Objekt nahe bzw. groß genug ist, bleibt er stehen.

Modul 2: Ausgangspunkt

=> das Modul 2 verhält sich ähnlich wie Modul 1, wobei es sich beim Ziel um ein anderes Objekt handelt.

Modul 3: In diesem Modus lässt sich das angeschlossene Tool manuell vom Steuern.

Alf_GUI.py

Alf_GUI.py wird in einer eigenen Session beim Programmstart ausgeführt.

Im Hauptfenster werden die Gesichtszüge von Alfred angezeigt die von der „Main“ gesteuert werden.

Mit einem Klick gelangt man in die Manuelle Steuerung von Alfred.

Neben Modi und Motoren wird hier eine Automatisch Tool Menü generiert, welches je nach angeschlossenen Toll aufgebaut wird.

Alf_Speech.py

Alf_Speech.py wird in einer eigenen Session beim Programmstart ausgeführt.

Dieses Programm überwacht die „speech.config“ Datei und spielt je nach Inhalt bestimmte Sound-Files ab.



Alf ToolConnect.py

Dieses Modul dient zur Erkennung und Ansteuerung des im Tool verbauten Arduinos.
Die Kommunikation erfolgt über eine Parameterübergabe über USB Mittels Software Serieller Schnittstelle.

Ablauf der Kommunikation:

1. Raspberry an Arduino -> 99 (Identifikation)
2. Arduino an Raspberry -> Name und Funktionen z.B. (1,Greifer,Zu,Auf)
3. Raspberry an Arduino -> Funktionen ausführen z.B. (0,1,20)
(Zu = 0, Auf = 1, Wie weit öffnen = 20)

Alf Verfolgung.py

Dieses Modul übernimmt das Auslesen der Pixy Cam und der Ultraschall Sensoren.
Je nach Aufruf aus der Main werden die Koordinaten des Objektes in Befehle an den Motor umgewandelt

Weitere Module

- Alf_LED.py -> LED Ein/Aus und Farbe
- Alf_Motor.py -> Motoren Ein/Aus und Geschwindigkeit
- Alf_Pixy2.py -> Pixy Cam auslesen
- Alf_Ultraschall.py -> Objekt Entfernung erreicht True/False
- Alf_Temperatur.py -> BME280 Sensor auslesen

Sprachsteuerung/Spracheingabe

Für die Spracheingabe wird die Software „Jarvis“ verwendet.

Emotionen

Probleme

Programm in mehreren Sessions ausführen

Da das Programm auf die Soundausgabe und bei der Erzeugung des GUI unterbrochen wird, mussten 3 Session gleichzeitig gestartet werden um alle Funktionen zu starten.



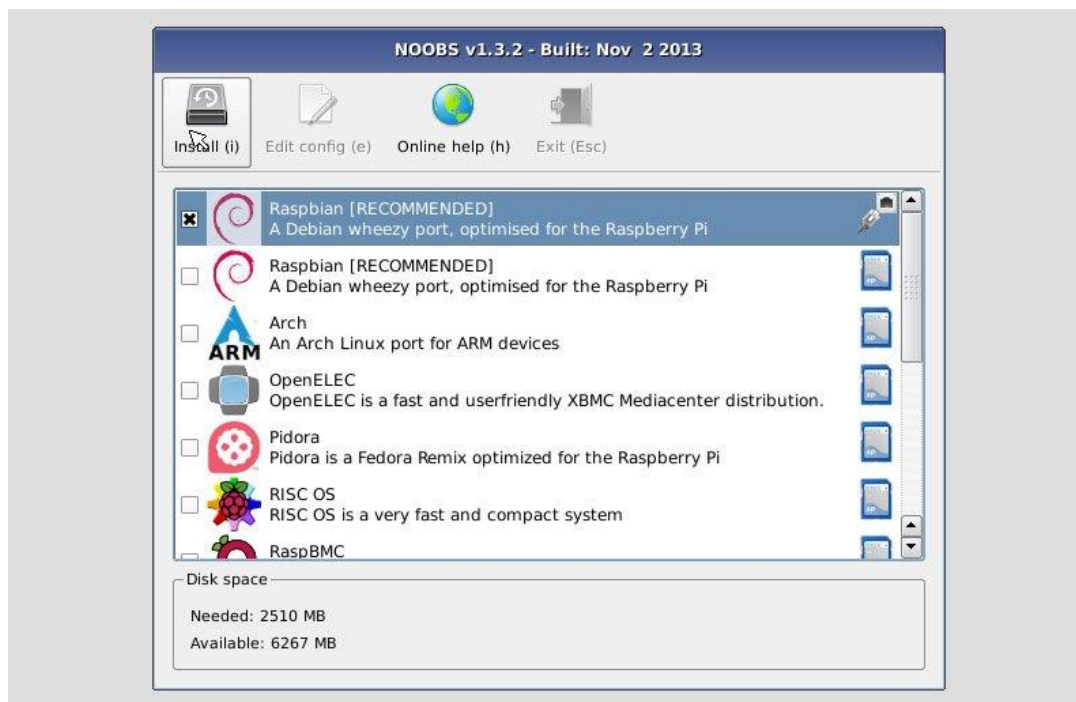
Alf_Main.py	->	Steuerung der Funktionen
Alf_GUI.py	->	Erzeugen der Grafischen Oberfläche
Alf_Speech.py	->	Ausgabe der Sounds



Raspberry einrichten

Raspbian installieren

1. Betriebssystem herunterladen:
Noobs herunterladen -> offizielle Webseite von Raspberry Pi.
<https://www.raspberrypi.org/downloads/noobs/>
Version: NOOBS LITE Network only. (Raspberry braucht bei der Installation eine Netzwerkverbindung)
2. SD-Karte des Raspberry Pi in FAT32 formatieren und die Noobs ZIP-Datei auf der SD-Karte entpackten.
3. SD-Karte in den Pi stecken und mit einem LAN-Kabel, Bildschirm (7 Zoll Raspberry Touch Screen), Maus, Tastatur und Stromstecker verbinden.
4. Bei der Betriebssystemauswahl Raspbian verwenden und den Installationsschritten folgen.



Hostname: Alfred

User: pi



Alfred von Github herunterladen

Die Daten werden zuerst von Github heruntergeladen.

Da ein Automatisches Setup noch nicht erstellt wurde, müssen danach noch einige Anpassungen auf dem Raspberry Pi gemacht werden.

Terminal auf dem Raspberry Pi öffnen:

```
cd /home/pi/  
git clone https://github.com/SDim44/Alfred.git
```

Alfred Updaten:

```
cd /home/pi/Alfred  
git pull
```

github Anmeldedaten nach der nächsten Eingabe speichern:

```
git config --global credential.helper store
```

Automatisches Setup durchführen

!!! Achtung !!!

Das Automatische Setup übernimmt alle Schritte unter den folgenden Punkten:

- Benötigte Software
- Raspberry Konfiguration

```
cd /home/pi/Alfred/setup  
sudo chmod +x setup.sh  
./setup.sh
```

(Während der Installation sind Benutzereingaben erforderlich.)



Benötigte Software

Serial Port ansteuern (Python Library)

```
sudo pip install pyserial
```

Wiringpi – GPIO ansteuern (Python Library)

(auf Raspbian vorinstalliert)

BME280 Temperatur Sensor (Python Library)

<https://canox.net/2017/09/raspberry-pi-der-bme280-sensor/>

```
sudo nano /etc/modules
```

Am Ende der Datei hinzufügen:

```
i2c-bcm2708
```

```
sudo nano /boot/config.txt
```

Am Ende der Datei hinzufügen:

```
dtoverlay=i2c1=on
```

```
dtoverlay=i2c_arm=on
```

Pixy Cam 2

https://docs.pixycam.com/wiki/doku.php?id=wiki:v2:building_the_libpixyusb_example_on_linux

libusb installieren:

```
sudo apt-get install git libusb-1.0-0-dev g++ build-essential
```

Pixy2 herunterladen:

```
cd /home/pi && git clone https://github.com/charmedlabs/pixy2
cd pixy2/scripts && ./build_libpixyusb2.sh
./build_get_blocks_cpp_demo.sh
```

Pixy2 testen:

```
cd ../build/get_blocks_cpp_demo/
sudo ./get_blocks_cpp_demo
```



Jarvis (Sprachsteuerung)

<https://github.com/alexylem/jarvis>

Jarvis herunterladen und starten:

```
cd /home/pi  
git clone https://github.com/alexylem/jarvis.git  
cd jarvis  
./jarvis.sh  
Jarvis
```

Anleitung zur Konfiguration unter:

<https://github.com/alexylem/jarvis>

Konfigurationshinweise:

Als Spracherkennung wird snowboy verwendet (offline-Erkennung)

Für alle Kommandos und das Schlüsselwort müssen vorab Aufnahmen in Jarvis erstellt werden.

Schlüsselwort: Alfred

Kommandos:

„Foge mir“ Mode 1

„Ladestation“ Mode 2

„stop“ Mode 3



Optionale Software

Matchbox-keyboard (Touchscreen/Onscreen Tastatur) - optional

```
sudo apt-get install matchbox-keyboard
```

Samba Server (Dateifreigabe) - optional

Der Samba Server wurde verwendet um mit dem Windows Explorer auf die Dateien am Raspberry zu zuzugreifen.

Installation:

```
sudo apt-get install samba samba-common smbclient
```

Default config sichern:

```
sudo mv /etc/samba/smb.conf /etc/samba/smb.conf_alt
```

Neue config-Datei erstellen und öffnen:

```
sudo nano /etc/samba/smb.conf
```

In Datei schreiben:

```
[global]
workgroup = WORKGROUP
security = user
encrypt passwords = yes
client min protocol = SMB2
client max protocol = SMB3

[root]
comment = root
path = /
read only = no
```

Samba Dienst neustarten:

```
sudo service smbd stop
sudo service smbd start
```

Samba Benutzer einrichten:

```
sudo smbpasswd -a pi
```

danach noch 2mal dein Passwort für den User „pi“ eingeben.

Jetzt kann der Pfad in der Ordner-Navigationsleiste am Windows Gerät eingegeben werden:

\\Alfred



Software für Alfred Webinterface
(in der aktuellen Version noch nicht verfügbar)

Apache2 (Webserver)

```
sudo apt-get install apache2
```

PHP

```
sudo apt install php
```

Website bereitstellen

```
cp /home/pi/Alfred/html /var/www/html
```

Jetzt kann über einen beliebigen Brower aus das Webinterface zugegriffen werden:

<http://alfred>



Raspberry Konfiguration

Bildschirm um 180° drehen:

Für viele Gehäuse des Raspberry Pi 7 Zoll Touchscreen ist es notwendig die anzeige zu drehen damit sie nicht auf dem Kopf steht.

Datei öffnen:

```
sudo nano /boot/config.txt
```

Am Ende des Dokuments eine Zeile einfügen:

```
lcd_rotate=2
```

Bootscreen ändern:

<https://scribes.net/customizing-boot-up-screen-on-raspberry-pi/>

Um das Bild, dass beim boot-vorgang angezeigt wird zu ändern, wird das Original Raspbian Bild mit dem Bild aus dem Alfred Ordner ersetzt.

```
sudo cp /home/pi/Alfred/DATA/design/splash.png  
/usr/share/plymouth/themes/pix/splash.png
```

Default Audio Device ändern:

Mit diesem Vorgang wird standardmäßig die externe Soundkarte verwendet.

Datei erstellen:

```
sudo nano /etc/asound.conf
```

In Datei schreiben:

```
pcm.!default  
{  
    type hw  
    card 1  
}  
  
ctl.!default  
{  
    type hw  
    card 1  
}
```



Init-Script erstellen

<http://raspberrypi.einsteiger/raspberry-pi-autostart-von-skripten-und-programmen-einrichten>

Vorteil eines Init-Scripts ist das ein Programm nicht nur beim booten des RasPi gestartet wird sondern beim runterfahren oder neustarten auch entsprechend behandelt wird.

Dazu eine Datei erstellt werden und der entsprechende Code in die Datei geschrieben werden.

```
sudo nano /etc/init.d/Alfred
```

In die Datei kopieren:

```
#!/bin/sh
### BEGIN INIT INFO
# Provides:          Alfred
# Required-Start:
# Required-Stop:
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Starts & Stops Alfred
### END INIT INFO
#Switch case fuer den ersten Parameter
case "$1" in
    start)
        #Aktion wenn start uebergeben wird
        echo "Starte Alfred"
        cd /home/pi/Alfred
        ./Alf_Start.sh

    stop)
        #Aktion wenn stop uebergeben wird
        echo "Stoppe Alfred"
        cd /home/pi/Alfred
        ./Alf_Stop.sh

    restart)
        #Aktion wenn restart uebergeben wird
        echo "Restarte Alfred"
        cd /home/pi/Alfred
        ./Alf_Stop.sh
        sleep 2
        ./Alf_Start.sh

    update)
        #Aktion wenn update uebergeben wird
        echo "Update Alfred"
        cd /home/pi/Alfred
        ./Alf_Update.sh

    *)
        #Standard Aktion wenn start|stop|restart nicht passen
        echo "(start|stop|restart)"
esac
exit 0
```

Jetzt kann Alfred mit folgendem Befehl gestartet, gestoppt und geupdatet werden.

```
Sudo /etc/init.d/Alfred start
Sudo /etc/init.d/Alfred stop
Sudo /etc/init.d/Alfred update
```



Desktop und Startmenü Verknüpfung

<https://www.raspberrypi.org/forums/viewtopic.php?t=9817>

Um alle Verknüpfungen zu erstellen, werden 3 Dateien angelegt und der entsprechenden Code geschrieben.

Alfred starten

Datei am Desktop erstellen:

```
sudo nano /home/pi/Desktop/Alfred_start.desktop
```

Ins Dokument schreiben:

```
[Desktop Entry]
Type=Application
Comment=Start Alfred
Name=Alfred
Exec=sudo /etc/init.d/Alfred start
Icon=/home/pi/Alfred/DATA/design/Alf_Logo_randlos_s_auf_t.png
Terminal=false
Categories=Utility
StartupNotify=true
```

Verknüpfung in das Startmenü kopieren:

```
sudo cp /home/pi/Desktop/Alfred_start.desktop /usr/share/raspi-ui-
overrides/applications/Alfred_start.desktop
```

Alfred beenden

Datei am Desktop erstellen:

```
sudo nano /home/pi/Desktop/Alfred_stop.desktop
```

Ins Dokument schreiben:

```
[Desktop Entry]
Type=Application
Comment=Stop Alfred
Name=Alfred
Exec=sudo /etc/init.d/Alfred stop
Icon=/home/pi/Alfred/DATA/design/stop.png
Terminal=false
Categories=Utility
StartupNotify=true
```

Verknüpfung in das Startmenü kopieren:

```
sudo cp /home/pi/Desktop/Alfred_stop.desktop /usr/share/raspi-ui-
overrides/applications/Alfred_stop.desktop
```




Alfred updaten

Datei am Desktop erstellen:

```
sudo nano /home/pi/Desktop/Alfred_update.desktop
```

Ins Dokument schreiben:

```
[Desktop Entry]
Type=Application
Comment=Update Alfred
Name=Alfred
Exec=sudo /etc/init.d/Alfred update
Icon=/home/pi/Alfred/DATA/design/update.png
Terminal=false
Categories=Utility
StartupNotify=true
```

Verknüpfung in das Startmenü kopieren:

```
sudo cp /home/pi/Desktop/Alfred_update.desktop /usr/share/raspi-ui-
overrides/applications/Alfred_update.desktop
```



Probleme

Berechtigungen

```
sudo chmod +x /home/pi/Alfred/Alf_Update.sh  
sudo chmod +x /home/pi/Alfred /Alf_Start.sh  
sudo chmod +x /home/pi/Alfred /Alf_Stop.sh  
sudo chmod +x /home/pi/Alfred /Alf_Main.py
```

Python Tkinter GUI über SSH starten

```
export DISPLAY=:0.0
```