Notes PREN

Pi-Features

- Time-To-Boot mit [Service] TimoutStartSec=5: ca. 20s
- Time-To-Reboot mit [Service] TimoutStartSec=5: ca. 23s

Wifi

`nano /etc/wpa_supplicant/wpa_supplicant.conf`

```
network={
        ssid=""
        psk=""
}
```

`wpa_cli -i wlan0 reconfigure`

Links

- http://www.wildow.com/blog/?p=1945
-

Python Threads

Theory

Da wir Tasks simultan auf der Raspberry Pi laufen lassen müssen, verwenden wir Threads, um dies sinnvoll umzusetzen.

Weiter unten ist ein konkretes Beispiel und als weitere Lektüre verwenden wir: O'reilly - Programmin Python - Mark Lutz

Source: Learn Raspberry Pi Programming With Python

```
# Define thread object

class myObject(threading.Thread):
    def __init__(self):
        #function used to initiate the class and thread
        threading.Thread.__init__(self) #necessary to start the thread
    def run(self):
        #function performed while thread is running
```

From the main portion of the program, we can start the thread by declaring a new `myObject` object (a new thread):

```
    newObject = myObject()
```

and then starting it with

```
    newObject.start()
```

The thread is now running with its own instance of the myObject, called newObject. Our thread (as shown in the final code at the end of the chapter) will be initiated with `threading.Thread.__init__(self)`. Once it has been started, it will continue to execute its function (in our case, collecting GPS data and taking pictures) until we quit the program.

Example

```
import os
from gps import *
from time import *
import time
import threading
import logging
from subprocess import call

#set up logfile
logging.basicConfig(filename='locations.log', level=logging.DEBUG, format='%(message

picnum = 0
gpsd = None

class GpsPoller(threading.Thread):
    def __init__(self): #initializes thread
        threading.Thread.__init__(self)
        global gpsd
        gpsd = gps(mode=WATCH_ENABLE)
        self.current_value = None
        self.running = True


    def run(self): #actions taken by thread
        global gpsd
        while gpsp.running:
            gpsd.next()

if __name__ == '__main__': #if in the main program section,
    gpsp = GpsPoller()      #start a thread and start logging
    try:                    #and taking pictures
        gpsp.start()
        while True: #log location from GPS
        logging.info    (str(gpsd.fix.longitude) + " " + str(gpsd.fix.latitude) + "
str(gpsd.fix.altitude))

        #save numbered image in correct directory
        call(["raspistill -o /home/pi/Documents/plane/image" + str(picnum) +
".jpg"], shell=True)
        picnum = picnum + 1 #increment picture number
        time.sleep(3)
    except (KeyboardInterrupt, SystemExit):
        gpsp.running = False
        gpsp.join()
```

Run Programs On Startup

Following things must be defined and done at startup of the Raspberry Pi:

Run scripts during boot:

`nano /etc/rc.local`

Automatic Login as Pi

Create a user `pi` and add any password:

`adduser pi`

Then change the settings to autologin on boot as `pi` in `raspi-config`:

`raspi-config`

Automatic Login as Root

Make sure in your `raspi-config`-settings that you have the following option enabled:

B1 Console: Text console, requiring user to login

Then edit the following file `/lib/systemd/system/getty@.service` as described below.

This line: `ExecStart=-/sbin/agetty --noclear %I $TERM` must be changed to:

`ExecStart=-/sbin/agetty --noclear -a root %I $TERM`

Setup a static IP

This enables us to connect to the Raspberry Pi via SSH just in case we need to connect to it this way.

```
ifconfig eth0 192.168.0.1 netmask 255.255.255.0
```

Miscellaneous Raspberry Pi Information

SSH Public Key

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCpMaWHXo+GGccUM8PiDJVWFdwPJGvkX/wxnMP7hwWAyq0b
```

Important Links

"A start job is running for LSB: raise network interfaces":

- https://ubuntuforums.org/showthread.php?t=2323253
- https://unix.stackexchange.com/questions/385281/why-im-giving-message-a-start-job-is-running-for-raise-network-interfaces-at
- https://www.raspberrypi.org/forums/viewtopic.php?t=135369
- Solution: https://unix.stackexchange.com/questions/186162/how-to-change-timeout-in-systemctl

Run bash script **after login**:

- https://stackoverflow.com/questions/39024657/run-bash-script-after-login