Anfangs des Semesters haben wir uns in der openHAB Dokumentation und in der Technik des Raspberry 2 eingelesen. Wir machten uns mit openHAB vertraut und installierten den openHAB Designer. Nachdem wir die Hardware erhalten haben, versuchten wir schrittweise die LED an den Raspberry anzubinden. Zuerst versuchten wir dies mit einer Anleitung aus dem Internet. 1Dies funktionierte nicht. Aber in weiteren Verlauf des Semesters hat es geklappt.

**Probleme**

Wir Studierenden kannten uns teilweise in der Gruppe nicht. Wir konnten die Fähigkeiten und Kenntnisse der Anderen nicht gut einschätzen. So war es schwierig Anfangs die Aufgaben gut zu verteilen.

Die Gruppenfindungsphase lief anfangs auch nicht optimal. Wir als Gruppe suchten anfangs des Semesters einen Raum, was sich als schwierig erwies. Im Raum E131 arbeitete schon eine andere Projektgruppe. Schlussendlich arbeiten im Raum E131 mehr als 20 Studierende, was zu viele waren.

Es gab auch Schwierigkeiten mit der Hardware. So war der Taste falsch verlötet und ein Kabel abgebrochen. Wir haben zuerst gedacht, der Taster reagiert falsch, weil der entsprechende Programmiercode nicht korrekt ist. So haben uns die Hardware Pannen viel Zeit gekostet.

Realisierung

Item in openHAB ist eine Konfigurationsdatei, wo alle einzelnen Teile definiert werden.

Die LEDs haben wir in der items Datei als switch definiert. Ein Switch in openHAB ist ein Schalter. Switch zeigt seinen Schalter Status an und kann Items an-, und ausschalten.   
Der Taster ist ein Contact Item. Dies ist ein Kontaktschalter.

Im Folgenden sind die Quellcode vom Taster und den beiden LEDs

Quellcode vom Taster

#Verwenden von GPIO

import RPi.GPIO as GPIO

import time

# Warnungen ausschalten

#GPIO.setwarnings(False)

# Pin Nummern verwenden

GPIO.setmode(GPIO.BCM)

# Pin 11 als Input

GPIO.setup(22, GPIO.IN)

# Pin 12 als Output

GPIO.setup(4, GPIO.OUT)

# Endlosschleife

while True:

# Solange Button nicht gedrueckt wird (False)

if not GPIO.input(22):

GPIO.output(4, False)

# Wenn der Button gedrückt wird

else:

GPIO.output(4, True)

# time.sleep(1)

GPIO.cleanup()

Quellcode von den LEDs

import RPi.GPIO as GPIO ## Import GPIO library

GPIO.setmode(GPIO.BCM) ## Use board pin numbering

GPIO.setup(4, GPIO.OUT) ## Setup GPIO Pin 7 to OUT

GPIO.output(4,False) ## Turn on GPIO pin 7