

SYT

# **Basement Shitting Prevention Team**

IoT in the Cloud

Rousavy, Pader-Barosch,  
Fentler, Deveci

14. Februar 2019

Bewertung:

Version:

1.0

Betreuer: Michael Borko

## Inhaltsverzeichnis

<b>1</b>	<b>Verwendete Technologien</b>	<b>3</b>
<b>2</b>	<b>Research</b>	<b>4</b>
2.1	Raspberry Setup . . . . .	4
2.2	Abtastrate der Informationen . . . . .	4
2.3	Aggregierung der Daten . . . . .	5
2.4	Schnittstellendefinition . . . . .	5
2.5	Energieversorgung . . . . .	5
2.6	Speicherverbrauch . . . . .	5
2.7	Verbindung . . . . .	5
<b>3</b>	<b>Implementierung</b>	<b>6</b>
3.1	Installation . . . . .	6
3.2	Schritte . . . . .	6
3.3	Bewegungssensor . . . . .	9

# 1 Verwendete Technologien

Es werden folgende Technologien verwendet:

- Raspberry Pi Zero W: Weil er klein und kostengünstig ist
- NodeJS: für den Webserver (Livestream), GPIO Zugriff (Bewegungssensor), Bilder, Mail

Verwendete NodeJS libraries:

- express: Für den Webserver/Livestream
- hls.js: Livestream in HLS format
- ip: IP-Adresse auslesen
- nodemailer: Mail mit IP/Bild senden
- onoff: Für GPIO pins
- pi-camera: Um auf die Kamera zuzugreifen

Um den Schaltplan zu verfassen wird die Desktop-Applikation **Fritzing** verwendet.

## 2 Research

### 2.1 Raspberry Setup

Die initiale Konfiguration des Raspberry Pi's erfolgt mittels der `raspi-info.md` Dokumentation.

### 2.2 Abtastrate der Informationen

Da eine fixe Abtastrate ineffizient wäre, wird ein GPIO callback verwendet um ein Bild zu senden.  
Der Bewegungssensor wird folgendermaßen initialisiert:

```
1  const pir = new Gpio(17, 'in', 'both')
```

Auflistung 1: GPIO Erstellung in JavaScript

Mittels dem erstellten Gpio Objekt kann ein Callback registriert werden:

```
2  pir.watch(function(err, value) {  
3      // Bewegungsmelder callback  
4      if (err) exit()  
5      console.log('Intruder detected')  
6      if (value == 1) {  
7          mail.sendMail()  
8      }  
9  })
```

Auflistung 2: GPIO Callback registrierung mittels watch in JavaScript

Zusätzlich wird ein Livestream über die Website erstellt, wobei keine fixe Abtastrate verwendet wird.

## 2.3 Aggregation der Daten

Bei jeder Rising Edge des Bewegungssensors wird eine Mail an den Besitzer gesendet. Der Code schaut folgendermaßen aus:

```
10  const pir = new Gpio(17, 'in', 'both')
11  pir.watch(function(err, value) {
12    if (err) exit()
13
14    console.log('Intruder detected')
15
16    if (value == 1) {
17      mail.sendMail()
18    }
19  })
```

Auflistung 3: Sender der Mail bei Erkennung einer Bewegung bzw. Rising Edge des Sensors

## 2.4 Schnittstellendefinition

Der Bewegungssensor wird über einen GPIO Pin angesprochen. Die Auswertung am Backend erfolgt über das onoff-Modul:

```
20  const Gpio = require('onoff').Gpio
```

Auflistung 4: Sender der Mail bei Erkennung einer Bewegung bzw. Rising Edge des Sensors

Der Livestream läuft als Webservice unter der IP (und Port 80) des Raspberry Pi. Realisiert wurde dies mittels dem Modul *Express*.

## 2.5 Energieversorgung

Der Raspberry Pi benötigt eine Stromversorgung über Micro-USB, welche entweder über eine normale Stromanbindung über eine Steckdose oder über ein Akku-pack erreicht wird.

## 2.6 Speicherverbrauch

Das **Basement Shitting Prevention** Programm benötigt 2.552 Bytes. Zusätzlich werden einige Packages benötigt, womit ein insgesamter Speicherverbrauch von idk Megabytes verwendet wird.

Der Raspberry Pi ist mit einer 32 GB SD-Karte ausgestattet, wobei neben Betriebssystem und Programm genügend Speicherplatz vorhanden bleibt.

## 2.7 Verbindung

## 3 Implementierung

### 3.1 Installation

Die folgenden Schritte wurden aus dem vorhandenen Dokument von Herr Professor Borko entnommen, und angepasst.

### 3.2 Schritte

1. Raspbian Image auf die SD-Karte flashen: Dazu wurde das neueste Raspbian-Stretch-Light Image von [hier](#) herunter geladen und mit der Software [Etcher](#) auf die SD Karte geflasht.
2. Image File mounten: Da man auf Windows nur den boot Ordner sieht, nicht aber den root Ordner, muss Linux verwendet werden. Dazu wurde ein VM mit Ubuntu verwendet. Um die SD Karte mit der VM zu verbinden braucht man einen USB zu SD Adapter. In der VM muss man noch diesen USB Port vom Host auf die VM ändern.

Anschließend muss man die 2 Folder (root und boot) auf zwei Ordner in der VM mounten, damit man diese nachher bearbeiten kann.

```
21         sudo mkdir /mnt/boot
22         sudo mkdir /mnt/root
23
24         sudo mount /dev/sdc1 /mnt/boot
25         sudo mount /dev/sdc2 /mnt/root
```

Auflistung 5: Mounting der SD Karte in Bash

Um herauszufinden auf welchem /dev/sdc . . . Verzeichnis der USB Adapter angeschlossen ist, führt man diesen Befehl aus:

```
26         sudo fdisk /dev/sdc
```

Auflistung 6: Mounting der SD Karte in Bash

3. Konfiguration im Boot-Folder In den Boot Ordner kommen jetzt folgende Dinge:
  - Ein leeres ssh File
  - Die `wpa_supplicant.conf`, in der die Netzwerkkonfiguration steht In dieses File kommt folgende Konfiguration, die uns den Pi mit dem TGM-Netzwerk verbindet. Als Passwort wurde ein hash verwendet.

```
27     network={
28         ssid="TGM1x"
29         scan_ssid=1
30         key_mgmt=WPA-EAP
31         identity="cbarosch"
32         password=hash:041096d6783b98b9fbc07c82d8f10b6a
33         eap=PEAP
34         phase2="auth=MSCHAPV2"
35     }
```

#### Auflistung 7: WPA\_supPLICANT datei

4. Konfiguration im Root-Folder Dort muss das File /etc/network/interfaces geändert werden. Dort sagen wir unter anderem, dass ein Python Script ausgeführt werden soll, dass uns die IP des PIs schickt sobald dieser hochgefahren ist.

```
36     source-directory /etc/network/interfaces.d
37
38     auto wlan0
39     allow-hotplug wlan0
40
41
42     iface wlan0 inet dhcp
43         wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
44         post-up /etc/network/if-up.d/sendIP.py
45
46     iface default inet dhcp
```

#### Auflistung 8: interface config datei

In dem Python File muss der Absender noch auf unseren Gmail Account geändert werden.

```
47     #!/usr/bin/python3
48     import smtplib
49     import datetime
50     import subprocess
51
52     passwd = "NoShittingAnymore"
53     user = "shittingpreventionagent@gmail.com"
54
55     sender = user
56     recipients = "chrispad2k@gmail.com cbarosch@student.tgm.ac.at
57     ↪ mfentler@student.tgm.ac.at mrousavy@student.tgm.ac.at
58     ↪ odeveci@student.tgm.ac.at".split()
59     subject = "RaspberryPi is now up ..."
60
61     now = datetime.datetime.now()
62
63     msg = ("From: %s\r\nTo: %s\r\nSubject: %s\r\n"
64           % (sender, ", ".join(recipients), subject))
65
66     msg = msg + "Testmail generated @" + str(now) + "\r\n\r\n"
67     msg = msg + subprocess.check_output('ip address', shell=True).decode()
68
69     def sendmail():
70         try:
71             server = smtplib.SMTP_SSL('smtp.gmail.com', 465)
72             server.set_debuglevel(0)
73             server.ehlo()
74             server.login(user, passwd)
75             server.sendmail(sender, recipients, msg)
76             server.quit()
77             print("Email sent successfully!")
78         except smtplib.SMTPAuthenticationError as e:
79             print("Username and Password not accepted.")
80             return False
81         except:
82             print("Unable to send the email!")
83             return False
84         return True
85
86     while sendmail() != True:
87         try:
88             print("Trying ... ")
89         except KeyboardInterrupt as e:
90             print("\nUser interrupt detected ...")
91             print("Email was not sent. Please check your connectivity!")
92         break
```

## Auflistung 9: Python script um eine email zu senden



Dieses Python File muss auch noch auf *executable* gesetzt werden. Dazu:

91

```
sudo chmod +x sendIP.py
```

Auflistung 10: Python datei als executable markieren

5. Unmounten Als letzter Schritt muss man die SD Karte noch unmounten. Dazu geht man in Linux in den File-Archiver und klickt auf den "Pfeil-Button"rechts neben dem Boot-Folder.

### 3.3 Bewegungssensor

Der Bewegungssensor wurde mit Jumper Kabeln mit den GPIO Ports des Raspberry Pis verbunden. Der dazugehörige Schaltplan:

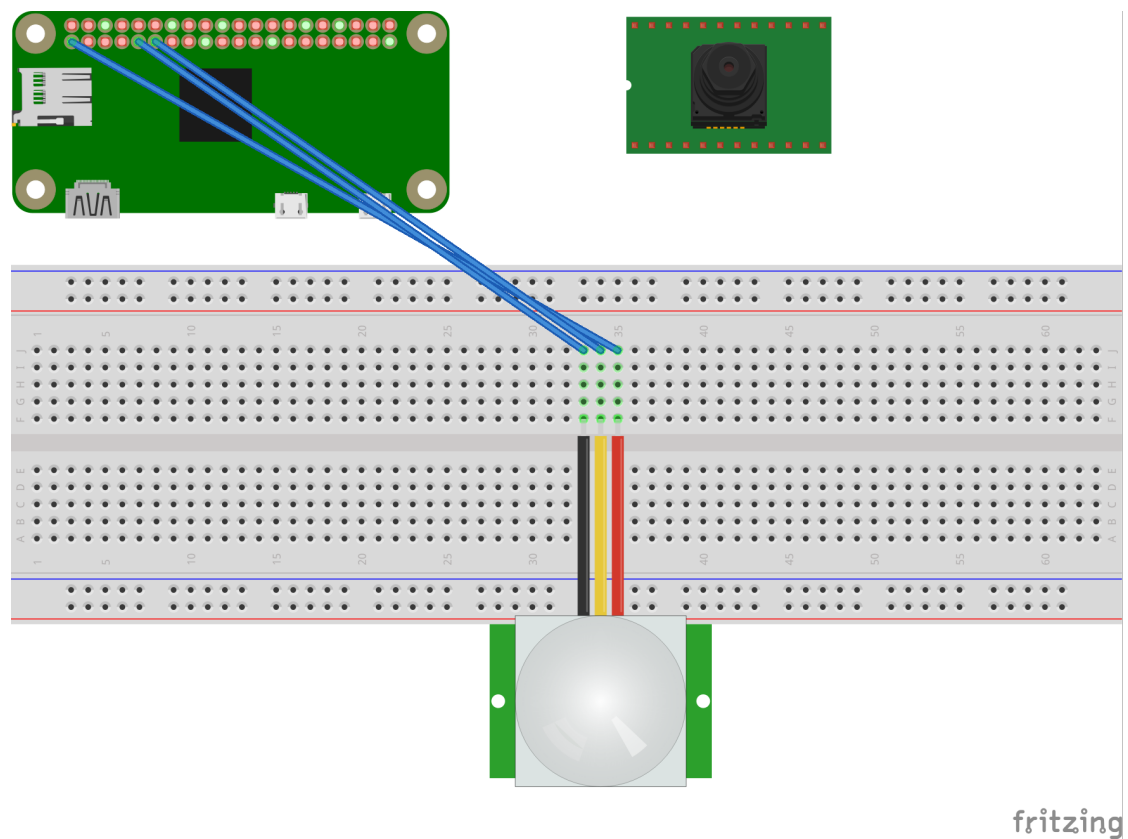


Abbildung 1: Der Schaltplan

## Abbildungsverzeichnis

1	Der Schaltplan . . . . .	9
---	--------------------------	---

## Auflistungsverzeichnis

1	GPIO Erstellung in JavaScript . . . . .	4
2	GPIO Callback registrierung mittels watch in JavaScript . . . . .	4
3	Sender der Mail bei Erkennung einer Bewegung bzw. Rising Edge des Sensors . . .	5
4	Sender der Mail bei Erkennung einer Bewegung bzw. Rising Edge des Sensors . . .	5
5	Mounting der SD Karte in Bash . . . . .	6
6	Mounting der SD Karte in Bash . . . . .	6
7	WPA_supplicant datei . . . . .	7
8	interface config datei . . . . .	7
9	Python script um eine email zu senden . . . . .	8
10	Python datei als executable markieren . . . . .	9