

PiTwitterCamera

Funktioner

Detta projekt är ett övervakningssystem som spelar in och tar bilder när rörelse upptäcks. Systemet rapporterar rörelser via Twitter “tweets”.

Krav

För att utföra detta projekt bör du ha dessa.

Hårdvara

- Raspberry Pi
- Raspberry Pi kamera modul
- PIR Rörelsesensor
- Taktill knapp
- Quick-connect sladdar

Mjukvara

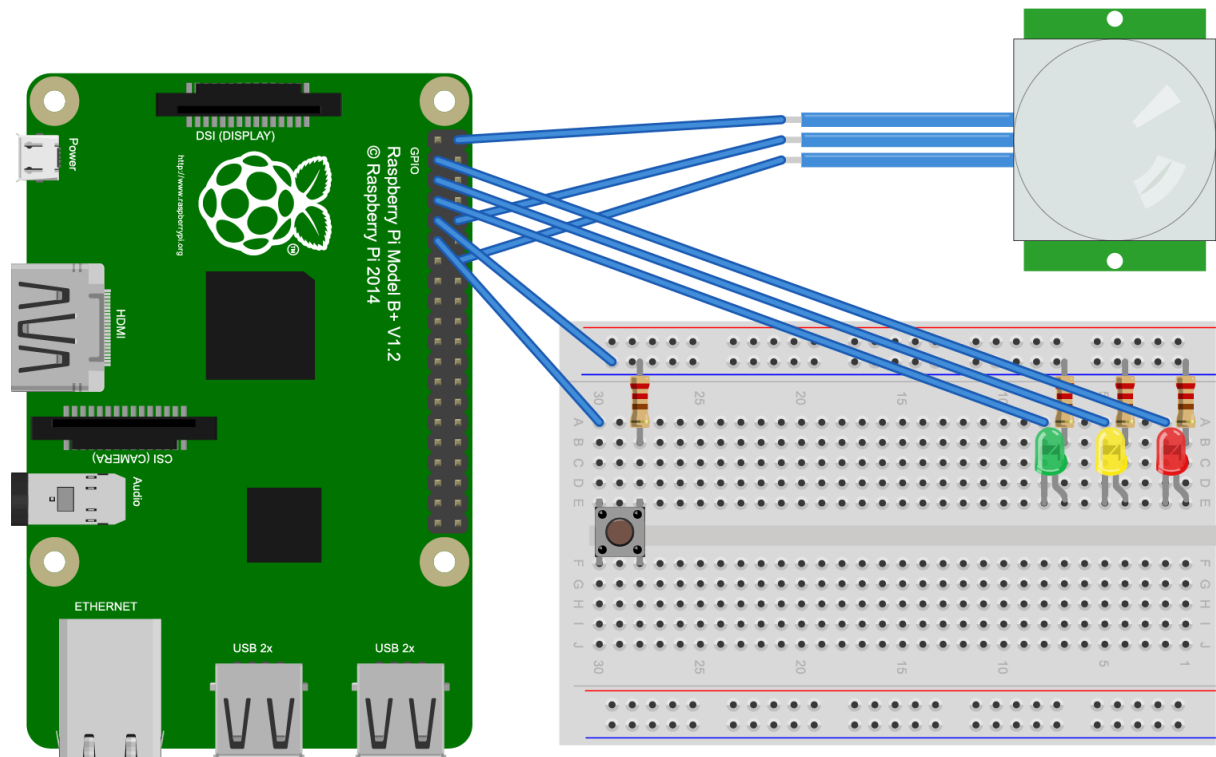
- Python 3
- pip

Python Moduler

- [gpiozero](#)
- [picamera](#)
- [datetime](#)
- [signal](#)
- [json](#)
- [tweepy](#)

Du bör även äga ett Twitter utvecklare konto.

Krets



PIN schema: https://gpiozero.readthedocs.io/en/stable/images/pin_layout.svg.

Kamera modulen bör även kopplas in.

Twitter Applikation

För att koden ska kunna autentisera applikationen genom API:n bör vi förse den med applikationens säkerhetsnycklar. Först bör en Twitter applikation skapas, detta här: <https://developer.twitter.com/en/apps/create>. Därefter finns säkerhets nycklarna under “Keys and Tokens” i applikationens inställningar.

En json fil bör skapas i samma katalog som själva Python koden för programmet (finns nedan). Denna json-fil bör följa formateringen som finns här:

https://github.com/KevinIssaDev/PiTweetCamera/blob/master/twitter_auth.json.

Kod

Källkoden finns här:

<https://github.com/KevinIssaDev/PiTweetCamera/blob/master/pitweetcamera.py>.

Importera moduler

Vi importerar de moduler vi behöver.

```
from gpiozero import Button, LED, TrafficLights, MotionSensor
from picamera import PiCamera
from datetime import datetime
from signal import pause
from json import load
import tweepy
```

Funktion: Vrint

Denna funktion används för att enklare kunna felsöka problem i koden.

```
def vrint(content):
    """ Verbose printing for debugging """
    if verbose:
        print(content)
```

Funktion: timestamp_format

Denna funktion skapar och returnerar en formaterad sträng av ett *datetime* objekt.

```
def timestamp_format(timestamp, filename=False, ext="jpg"):
    """ Formats timestamp to string """
    if filename:
        # 'day-month-year@hour;minute.jpg'
        return timestamp.strftime('%d-%m-%Y@%H;%M;%S.' + ext)
    else:
        # Day Month Date Hour:Minute:Second year
        return timestamp.strftime('%c')
```

Funktion: timestamp_format

Denna funktion läser in autentiserings datan från json filen och returnerar ett "dictionary"* med all data.

```
def load_auth_tokens(auth_file):
    """ Loads the authentication tokens from 'auth_file' """
    with open(auth_file, 'r') as f:
        auth_tokens = load(f)
    vrint("Authentication tokens loaded.")
    return auth_tokens
```

Funktion: tweet

Denna funktion skickar en "tweet" genom *tweepy* API:n med antingen bara text, eller med både media och text.

```
def tweet(content, media=False, media_path=None):
    """ Tweets the content with or without media """
    if media:
        api.update_with_media(media_path, content)
    else:
        api.update_status(content)
```

Funktion: on_motion

Denna funktion tar en bild som laddas upp till Twitter och börjar även spela in video.

```
def on_motion():
    """ Takes a photo then records until movement stops """
    lights.red.off()
    lights.amber.blink()
    vrint("Motion detected!")
    thumbnail_capture_time = datetime.now()
    thumbnail_filename = timestamp_format(thumbnail_capture_time,
    filename=True)
    camera.capture(thumbnail_filename)
    vrint("Thumbnail captured.")
    vrint("Recording...")
```

```
        camera.start_recording(timestamp_format(datetime.now()),
filename=True, ext='h264'))
        tweet("Motion detected at
{}!".format(timestamp_format(thumbnail_capture_time)), media=True,
media_path=thumbnail_filename)
        vrint("Posted to Twitter.")
```

Funktion: on_no_motion

Denna funktion avslutar pågående inspelning.

```
def on_no_motion():
    """ Stops the recording when there's no movement """
    lights.amber.off()
    lights.red.on()
    camera.stop_recording()
    vrint("Recording done.")
```

Funktion: on_button_press

Denna funktion tar en bild.

```
def on_button_press():
    """ Takes a photo when the button is pressed """
    lights.red.off()
    lights.green.on()
    vrint("Photo captured.")
    camera.capture(timestamp_format(datetime.now()),
filename=True))
    lights.green.off()
    lights.red.on()
```

När koden körs

När koden körs direkt (och inte importeras som en modul) vilket if `__name__ ==`

`"__main__":` ser till, skapas objekten för knappen, rörelsedetektorn, LED lamporna och kameramodulen. Alla lamporna släcks och koden körs i “verbose” läge. Autentiserings datan

läses in och ett *tweepy* API objekt skapas. Kretsens komponenter tilldelas funktioner som körs när “event” sker.

```
if __name__ == "__main__":
    button = Button(17)
    sensor = MotionSensor(15)
    lights = TrafficLights(2, 3, 4)
    camera = PiCamera()
    lights.red.off()
    lights.amber.off()
    lights.green.off()
    verbose = True
    auth_tokens = load_auth_tokens('twitter_auth.json')
    auth = tweepy.OAuthHandler(auth_tokens['consumer_key'],
auth_tokens['consumer_secret'])
    auth.set_access_token(auth_tokens['access_token'],
auth_tokens['access_token_secret'])
    api = tweepy.API(auth)
    lights.red.on()
    sensor.when_motion = on_motion
    sensor.when_no_motion = on_no_motion
    button.when_pressed = on_button_press
    pause()
```