PiTweetCamera

Functionality

This is a surveillance camera system scripted in Python. It detects and records movements. The recordings are saved locally, movements are reported to Twitter using *tweepy*.

Requirements

Hardware

- Raspberry Pi
- Raspberry Pi camera module
- PIR Motion Sensor
- Tactile button
- Quick-connect wires

Software

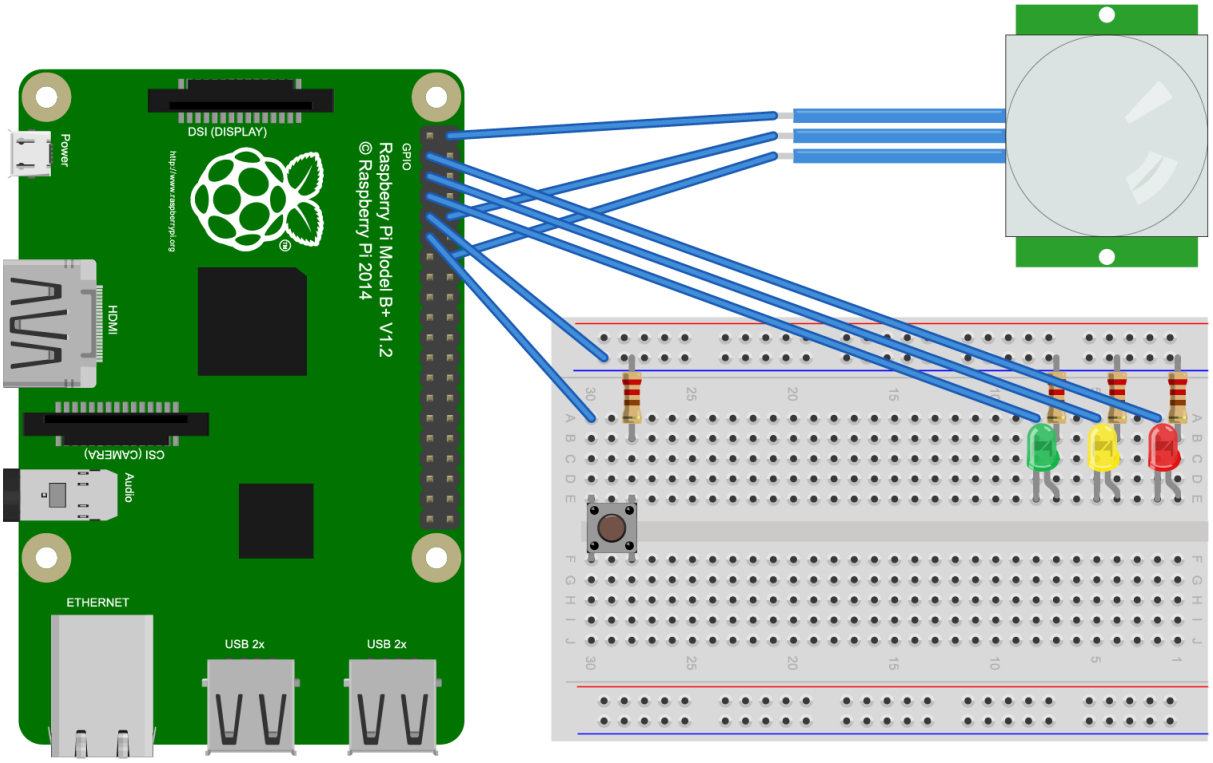
- Python 3
- pip

Python Modules

- gpiozero
- picamera
- datetime
- signal
- json
- tweepy

You also need to own a Twitter developer account.

Circuit



PIN layout: https://gpiozero.readthedocs.io/en/stable/_images/pin_layout.svg.

The camera module has to be connected as well.

Twitter Application

For the code to be able to authenticate the application through the API we need to provide it with the application's security tokens. First an application has to be created, which can be done here: https://developer.twitter.com/en/apps/create. The tokens can then be found under "Keys and Tokens" in the application's settings.

A JSON-file has to be created for the script to load the tokens from. The file should follow this format: https://github.com/KevinIssaDev/PiTweetCamera/blob/master/twitter_auth.json.

Code

Source code:

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

Import modules

The required modules are imported.

```
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
```

Function: Vrint

Function for debugging the code using verbose mode.

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

Function: timestamp_format

Returns formatted string from *datetime* object.

```
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')
```

Function: timestamp_format

Returns a dictionary containing the application's tokens read from the JSON-file.

```
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
```

Function: tweet

Posts a "Tweet" with or without media through the API using *tweepy*.

```
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)
```

Function: on_motion

Starts recording and "tweets" the timestamp along with a picture captured at the moment of movement detection.

```
def on_motion():
    """ Captures 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.")
```

Function: on_no_motion

Stops the recording.

```
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

Captures a photo.

```
def on_button_press():
    """ Captures 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()
```

When the code is run...

The instances of objects for the button, motion sensor, LEDs and the camera module are created. The *tweepy* object is created and the events of said components are assigned functions.

```
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()
```