**Intruder detector with Raspberry Pi and Pushbullet**

**Project Description:**

In this Project We are going to show you how to build a simple application that detects motion and sends notifications to your smart phone or Web browsers through Pushbullet using a Raspberry Pi with camera module. If someone moving in the camera then you will get notification to your mobile phone or web browsers. The notification comes with text, image, and video that you can view it right on the Mobile phone or Web Browsers.

**Software Required**:

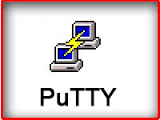
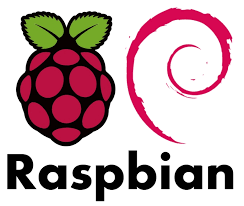
* Raspbian OS
* Putty
* Pushbullet

**Raspbian OS:**

This is the recommended os for raspberry pi. You can also installed other OS from third party. Raspbian OS is debian based OS. We can install it from noobs installer.

**Putty:**

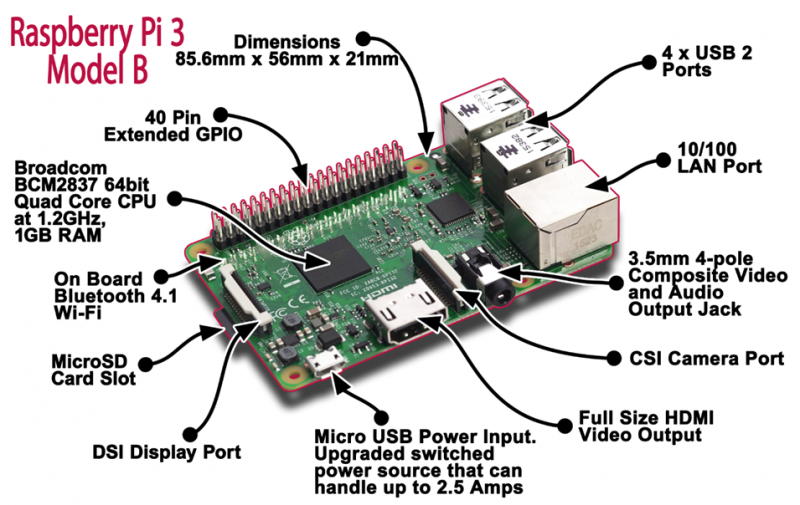
PuTTY is an SSH and telnet client, developed originally by Simon Tatham for the Windows platform. PuTTY is open source software that is available with source code and is developed and supported by a group of volunteers. Here we are using putty for accessing our raspberry pi remotely.



**Hardware Used:**

* Raspberry Pi
* Raspberry Pi Camera
* Power Supply

**Raspberry Pi 3 pin block diagram:**

* This is the latest version of raspberry pi. In this we have inbuilt Bluetooth and wi-fi, unlike previously we have to use Wi-Fi dongle in one of its usb port. There are total 40 pins in RPI3.  Of the 40 pins, 26 are GPIO pins and the others are power or ground pins (plus two ID EEPROM pins.)
* There are 4 USB Port and 1 Ethernet slot, one HDMI port, 1 audio output port and 1 micro usb port and also many other things you can see the diagram on right side. And also we have one micro sd card slot wherein we have to installed the recommended Operating system on micro sd card.
* There are two ways to interact with your raspberry pi. Either you can interact directly through HDMI port by connecting HDMI to VGA cable, and keyboard and mouse or else you can interact from any system through SSH. .

**Circuit Diagram:**

****

### You need to register in Pushbullet account for notification. Go to this link

### [https://www.pushbullet.com](https://www.pushbullet.com/)

### You can register either Google or Facebook.

### In my case, We are using Gmail Account. After successfully signup, log in and go to

### Settings –> Account to create an API access token.

### We need that key for Raspberry Pi to push notifications. Just note down that key in the code.

### Here are images for

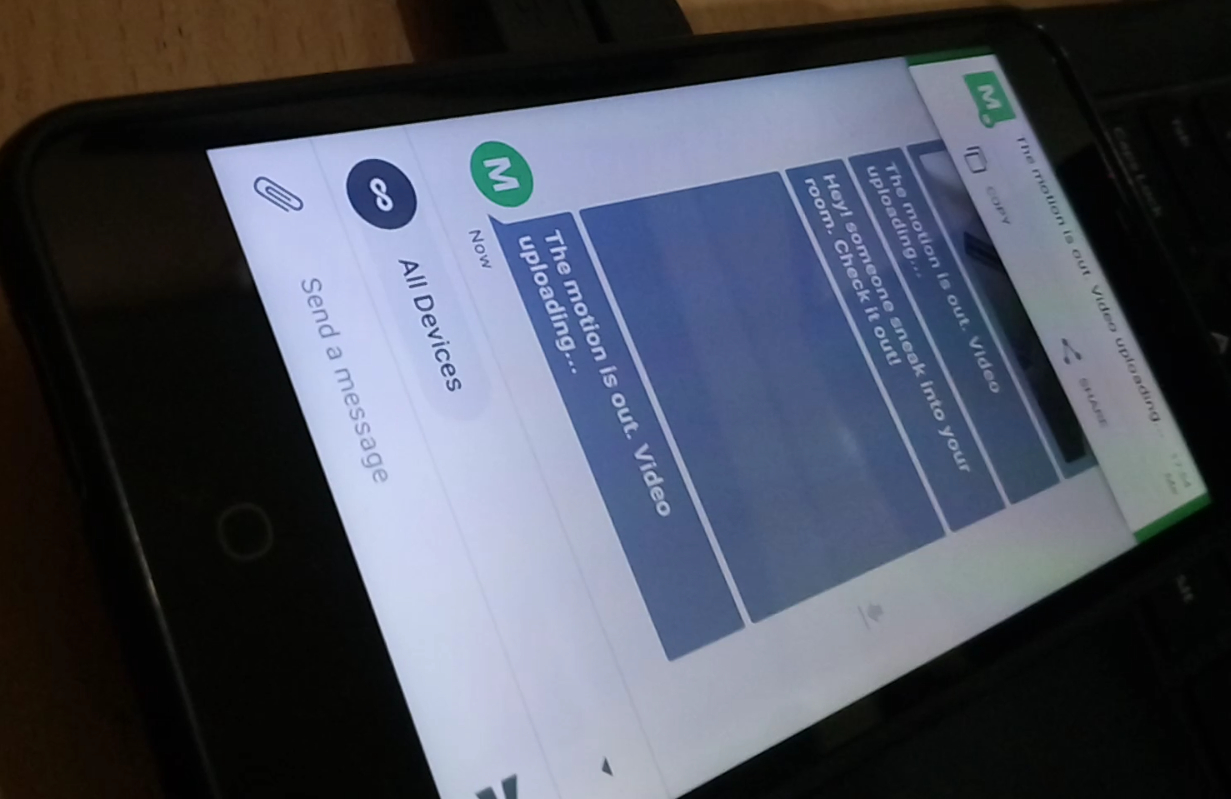
### Token Access Key:

### C:\Users\Deligence\Desktop\Motion Detection\motion3.png

### Pushbullet for Web Browser :

### C:\Users\Deligence\Desktop\Motion Detection\motion1.png

### Pushbullet for Mobile phone:



**Python Code:**

#!/usr/bin/python

import picamera

import picamera.array

import subprocess

import io

import os

import numpy as np

from pushbullet import Pushbullet

from time import sleep

from push import NotificationHandler

from threading import Thread

from Queue import Queue

import sched, time

import logging

#========= Customisable Parameters ======

#PUSHBULLET\_KEY='enter\_your\_pushbullet\_key\_here'

PUSHBULLET\_KEY = 'o.FjzdLl1ufpMuOuWD7WQmnOZVKrTiM1vP'

#========= Global variables ========

CAMERA\_OUT\_PATH = '/home/pi/Desktop/'

WORKING\_DIR="/home/pi/Desktop/PiCameraNotifier/"

LOG\_FILE\_PATH=WORKING\_DIR+'run.log'

VIDEO\_RECORDING\_PORT=1

MOTION\_ANALYSIS\_PORT=2

logging.basicConfig(format='%(asctime)s %(message)s', datefmt='%m/%d/%Y %I:%M:%S %p',filename=LOG\_FILE\_PATH,level=logging.INFO)

logging.info("=========== app launched ========")

camera = picamera.PiCamera()

camera.annotate\_background = True

stream = picamera.PiCameraCircularIO(camera, seconds=10, bitrate=1300000)

scheduler = sched.scheduler(time.time, time.sleep)

def didReceiveCommand(command):

global notificationHandler

if command == "@check":

logging.info("get system info")

process = subprocess.Popen([ WORKING\_DIR + 'systemInfo.sh'], stdout=subprocess.PIPE)

out, err = process.communicate()

pushData = {'type': 'TEXT\_MESSAGE', 'text': out}

notificationHandler.pushToMobile(pushData)

if command == "@snap" :

fileName=time.strftime("%Y%m%d\_%I:%M:%S%p") # '20170424\_12:53:15AM'

captureImage(fileName)

else:

logging.info("Command not supported: " + command)

logging.info("send notification to response")

logging.info("### Setup Notification Listener")

notificationHandler = NotificationHandler(PUSHBULLET\_KEY,didReceiveCommand)

class DetectMotion(picamera.array.PiMotionAnalysis):

def analyse(self,a):

a = np.sqrt(np.square(a['x'].astype(np.float)) + np.square(a['y'].astype(np.float))).clip(0, 255).astype(np.uint8)

if(a > 60).sum() > 10:

logging.info("motion just detected")

print("motion just detected")

didDetectMotion()

isRecordingMotion = False

def didDetectMotion():

global isRecordingMotion

if isRecordingMotion:

print("is Recording Motion ...")

else:

isRecordingMotion = True

print("start Recording Motion ...")

global notificationHandler

global camera

pushData = {'type': 'TEXT\_MESSAGE', 'text': 'Hey! someone sneak into your room. Check it out!'}

notificationHandler.pushToMobile(pushData)

fileName=time.strftime("%Y%m%d\_%I:%M:%S%p") # '20170606\_12:53:15PM'

logging.info("push image...")

captureImage(fileName)

camera.wait\_recording(7)

writeVideo(fileName)

isRecordingMotion = False

def captureImage(fileName):

global camera

global notificationHandler

camera.annotate\_text = fileName

filePath=CAMERA\_OUT\_PATH+fileName+'.jpg'

logging.info("capture still image to file: ", filePath)

camera.capture(filePath)

pushData = {'type': 'IMAGE\_MESSAGE', 'filePath': filePath, 'fileName': fileName+'.jpg'}

logging.info("push image data :", pushData)

notificationHandler.pushToMobile(pushData)

def writeVideo(fileName):

global stream

global notificationHandler

logging.info('Writing video with fileName: ', fileName)

filePath=CAMERA\_OUT\_PATH+fileName+'.h264'

with stream.lock:

stream.copy\_to(filePath)

# convert from h264 to mp4

outputFilePath=CAMERA\_OUT\_PATH+fileName+'.mp4'

logging.info("convert from h264 to mp4...")

subprocess.check\_call(["ffmpeg", '-framerate', '24', '-i', filePath, '-c', 'copy', outputFilePath])

pushData = {'type': 'VIDEO\_MESSAGE', 'filePath': outputFilePath, 'fileName': fileName+'.mp4'}

notificationHandler.pushToMobile(pushData)

def cameraInitialize():

logging.info("cameraInitialize: for (1) motion detection, and (2) circularIO recording")

global camera

# motion detection

camera.start\_recording(

'/dev/null',

splitter\_port=MOTION\_ANALYSIS\_PORT,

resize=(640,480),

format='h264',

motion\_output=DetectMotion(camera, size=(640,480))

)

# circularIO recording

global stream

camera.start\_recording(

stream,

format="h264",

resize=(640,480),

splitter\_port=VIDEO\_RECORDING\_PORT)

def main():

logging.info("Start main")

global notificationHandler

logging.info("### Initialize Camera")

cameraInitialize()

pushData = {'type': 'TEXT\_MESSAGE', 'text': 'PiCameraNotifier app starts !'}

notificationHandler.pushToMobile(pushData)

if \_\_name\_\_ == "\_\_main\_\_":

main()