Code :

# Python code:

import RPi.GPIO as GPIO

import time

import numpy as np

import cv2

from datetime import datetime

import os

import smtplib

from email.MIMEMultipart import MIMEMultipart

from email.MIMEBase import MIMEBase

from email.MIMEText import MIMEText

from email import Encoders

gmail\_user = "teamiot3@gmail.com" #Sender email address

gmail\_pwd = "team3@iot" #Sender email password

to = "dasrahulkumar89@gmail.com" #Receiver email address

subject = "Security Alert"

text = "Hello,Sorry to disturb you, here is some activity in your home. See the attached

picture."

sensor = 4

GPIO.setmode(GPIO.BCM)

GPIO.setup(sensor, GPIO.IN, GPIO.PUD\_DOWN)

previous\_state = False

current\_state = False

while True:

previous\_state = current\_state

current\_state = GPIO.input(sensor)

if current\_state != previous\_state

new\_state = "HIGH" if current\_state else "LOW"

print("GPIO pin %s is %s" % (sensor, new\_state))

if current\_state:

cap = cv2.VideoCapture(0)

ret, frame = cap.read()

cap = cv2.VideoCapture(0)

print "Saving Photo"

picname = datetime.now().strftime("%y-%m-%d-%H-%M")

picname = picname+'.jpg'

cv2.imwrite(picname, frame)

print "Sending email"

attach = picname

msg = MIMEMultipart()

msg['From'] = gmail\_user

msg['To'] = to

msg['Subject'] = subject

msg.attach(MIMEText(text))

part = MIMEBase('application', 'octet-stream'

part.set\_payload(open(attach, 'rb').read())

Encoders.encode\_base64(part)

part.add\_header('Content-Disposition', 'attachment; filename="%s"' % os.path.basename(attach))

msg.attach(part)

mailServer = smtplib.SMTP("smtp.gmail.com", 587)

mailServer.ehlo()

mailServer.starttls()

mailServer.ehlo()

mailServer.login(gmail\_user, gmail\_pwd)

mailServer.sendmail(gmail\_user, to, msg.as\_string())

mailServer.close()

print "Email Sent"

os.remove(picname)

# Java Code:

package pirrelay;

import javax.microedition.midlet.MIDlet;

import jdk.dio.DeviceManager;

import jdk.dio.gpio.GPIOPin;

public class Pirrelay extends MIDlet {

private static final int sensor=1;

private static final int relay=4;

private GPIOPin sensor1;

private GPIOPin relay1;

@Override

public void startApp() {

try{

sensor1=DeviceManager.open(sensor);

relay1=DeviceManager.open(buzzer);

while(true)

{

if(sensor1.getValue())

{

System.out.println("Some intruder came into your house.");

relay1.setValue(false);

Thread.sleep(1000);

while(sensor1.getValue()){ }

relay1.setValue(true);

} }

}

catch(Exception ex){System.out.println(ex.getMessage());

}

}

@Override

public void destroyApp(boolean unconditional){ } }

Pushover app code:

import time

import RPi.GPIO as GPIO

import httplib, urllib

# setup GPIO using Broadcom SOC channel numbering

GPIO.setmode(GPIO.BCM)

# define the GPIO port you will use for the motion detector

PIR\_SENSOR = 4

# number of seconds to delay between alarms

DELAY = 10

# set to pull-up (normally closed position for a PIR sensor dry contact)

GPIO.setup(PIR\_SENSOR, GPIO.IN, pull\_up\_down=GPIO.PUD\_UP)

# set the LED GPIO to output

GPIO.setup(LED, GPIO.OUT)

# Pushover API setup

PUSH\_TOKEN = "" # API Token/Key

PUSH\_USER = "" # Your User Key

PUSH\_MSG = "Motion detected!" # Push Message you want sent

# This function sends the push message using Pushover.

# Pass in the message that you want sent

def sendPush( msg ):

conn = httplib.HTTPSConnection("api.pushover.net:443")

conn.request("POST", "/1/messages.json",

urllib.urlencode({

"token": PUSH\_TOKEN,

"user": PUSH\_USER,

"message": msg,

}), { "Content-type": "application/x-www-form-urlencoded" })

conn.getresponse()

return

try:

# setup an indefinite loop that looks for the PIR sensor to be triggered

while True:

# motion is detected

GPIO.wait\_for\_edge(PIR\_SENSOR, GPIO.RISING)

# print and push message

print(PUSH\_MSG)

sendPush(PUSH\_MSG)

time.sleep(DELAY)

except KeyboardInterrupt:

# cleanup GPIOs on keyboard exit

GPIO.cleanup()

# cleanup GPIOs when program exits

GPIO.cleanup(