**60 HOURS CODE SPRINT**

**HOME SECURITY EMAIL ALERT SYSTEM**

****

**Date: 1 October 2016**

**Submitted to: Developed By:**

Prof. Sandeep Upadhayay Manasi Pandya

**HOME SECURITY EMAIL ALERT SYSTEM**

Description: This system takes picture of the intruder who enters your home and sends an email to you whenever you are not at your home.

**STEPS**

**Step-1: What you’ll need?**

Hardware Requirements:

* Raspberry pi
* Micro SD card
* PIR sensor
* Ethernet cable
* USB camera
* Power supply to power raspberry pi

Software Requirements:

* Python 2.7

**Step- 2: Setting up your Raspberry pi with Raspbian:**

First you need to install Raspbian on your micro SD card that you will be using in your Raspberry Pi.You can download the latest image of Raspbian from Raspberry Pi website at:

<http://www.raspberrypi.org/downloads/>

You will need to use an image writing tool to install the image on your SD card. You can find the steps to do that at:

[http://www.raspberrypi.org/documentation/installat...](http://www.raspberrypi.org/documentation/installation/installing-images/)

**Step- 3: Install OpenCV and required libraries:**

There are lots of methods to do this. But I would prefer using the simplest method that is available in OpenCV website at [http://docs.opencv.org/doc/tutorials/introduction/...](http://docs.opencv.org/doc/tutorials/introduction/linux_install/linux_install.html)

Open the terminal of your raspberry pi and execute the following commands...

Install compiler:

$ sudo apt-get install build-essential

Install required packages:

$ sudo apt-get install cmake git libgtk2.0-dev pkg-config libavcodec-dev libavformat-dev libswscale-dev

Install optional packages:

$ sudo apt-get install python-dev python-numpy libtbb2 libtbb-dev libjpeg-dev libpng-dev libtiff-dev libjasper-dev libdc1394-22-dev

Now download OpenCV to wherever you want to compile the source.

$ mkdir xxx

$ cd xxx

$ git clone <https://github.com/Itseez/opencv.git>

Create and build directory and onfigure OpenCV with cmake.

$ cd OpenCV-2.4

$ mkdir build

$ cd build

$ cmake -D CMAKE\_BUILD\_TYPE=RELEASE -D CMAKE\_INSTALL\_PREFIX=/usr/local -D WITH\_TBB=ON -D BUILD\_NEW\_PYTHON\_SUPPORT=ON -D WITH\_V4L=ON -D INSTALL\_C\_EXAMPLES=ON -D INSTALL\_PYTHON\_EXAMPLES=ON -D BUILD\_EXAMPLES=ON -D WITH\_QT=ON -D WITH\_OPENGL=ON ..

Now compile it

$ make

And finally install OpenCV

$ sudo make install

**Step- 4: Python code:**

(Logic)

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 = "xxxx@gmail.com" #Sender email address

gmail\_pwd = "xxxx" #Sender email password

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

subject = "Security Breach"

text = "There 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)

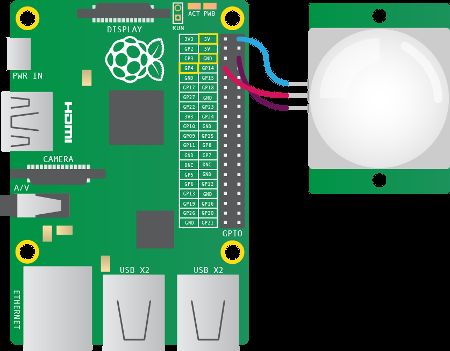
part.add\_header('Content-Disposition',

'attachment; filename="%s"' % os.path.basename(attach))

msg.attach(part)

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

**Step- 5: Make necessary connections and test the setup:**

[](http://www.instructables.com/file/FMK5OXAI82VRQDC/)

Now connect the USB camera and PIR sensor to the Raspberry Pi.

Using three female-to-female jumper cables, you'll need to connect each of the PIR sensor's connectors to the appropriate pins on the Raspberry Pi.

Connect the top one labelled VCC on the PIR sensor to the 5V pin on the Raspberry Pi, connect the middle one labelled OUT to GPIO pin 4, and connect the bottom one labelled GND to a ground pin also marked GND.

Now you run the python code with sudo to check the setup. Whenever there is a movement it detected by the PIR sensor, an image is captured and email is set to the email address you entered as the receivers email address.

**Step- 6: Output:**

Now, if you put the setup in front of your home door and run your python code and leave your home. If someone opens the door, you will get an email alert.