**Remote Monitoring and Manipulation of Appliances via E-Mail**

**Submitted by Ashwini Rajasekhar 15BEC0530**

**And Sushant Govindraj 15BEC0068**

**Abstract**

Need for remote automation is growing by the day, and in this project we present a way to control appliances, both digital and analog via e-mail, monitoring their statuses, making appropriate changes, and setting up a password lock in case of an intruder, and a camera to capture the same. This way there is all rounded protection and accessibility, which only requires an internet connection to survive.

**Components**

1. Raspberry Pi 3
2. PiCamera
3. LED’s
4. Breadboard
5. Resistors
6. Connector Cables

Since it is a basic setup we used a simple circuit, but in real time application it uses bulbs and motors.

**Proposed Methodology**

We have used libraries in Python and RPi and used two separate Gmail accounts, one as the Raspberry Pi end and one as the User End. To start off we have a password lock to safeguard the control from intruders. If the password entered is wrong the PiCamera captures the photo of an intruder, acting as CCTV camera, attaches the picture to the email body and sends it to the user. If the password entered is correct, the current status of the appliances is sent to the user, and they go back and forth with commands and statuses. The user end sends commands in a syntax we prescribe, and the Raspberry Pi end decodes this syntax to execute the required command. Then the commands are sequentially executed, new status is returned, and it goes on until user wants to terminate connection.

In an effort to make it user friendly, we made a new syntax for the RPi to interpret, so that any user not having knowledge of Python can use it as well.

1) To change appliance status of Appliance 1,2,3:

**A1-HIGH.  
A2-LOW.  
~**

The hyphens, dots, and tilde’s are used as indicators for the program. Hyphen, to separate appliance code and status, full stops to indicate end of line, and tilde to indicate end of program.

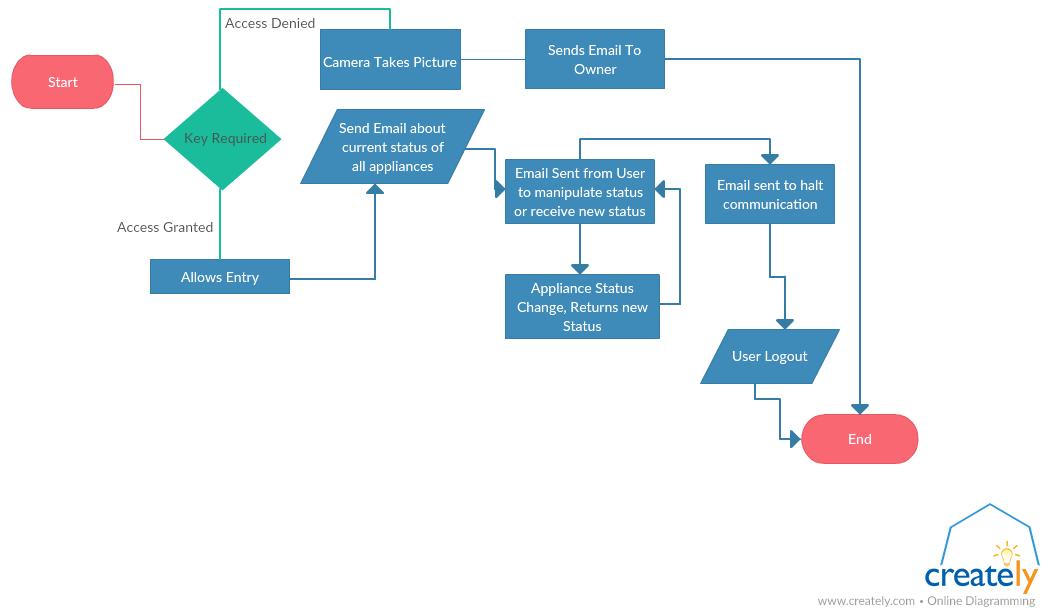
2) To ask for current status, send ‘**status.**’

3) To exit, send ‘**exit.**’

4) In the case of A4 Appliance controlled by PWM (fans, motors etc) we have 6 levels of speed (0-6) that alter the duty cycle of the appliance accordingly.

In such a way we can go back and forth, monitor and manipulate home appliances from remote areas.

**Flowchart**

****

**Code**

import imaplib

import RPi.GPIO as GPIO

import time

import smtplib

from email.mime.text import MIMEText

from email.mime.multipart import MIMEMultipart

import getpass

import feedparser

from email.mime.base import MIMEBase

from email import encoders

import picamera

pin0 = 11

pin1 = 12

pin2 = 13

pin3 = 15

GPIO.setmode(GPIO.BOARD)

GPIO.setwarnings(False) #To remove any warnings

GPIO.setup(pin1,GPIO.OUT)

GPIO.setup(pin2,GPIO.OUT)

GPIO.setup(pin0,GPIO.OUT)

GPIO.setup(pin3,GPIO.OUT)

p = GPIO.PWM(pin3,700)

p.start(0)

def takin():

a = [GPIO.input(pin0), GPIO.input(pin1), GPIO.input(pin2),GPIO.input(pin3)]

return a

def logout(): #For total logout from servers

sendser.quit()

server.close()

server.logout()

GPIO.cleanup()

return

def sending(mylist = [], \*a): #function to send email

text = "Current state: "

j=1

for i in mylist:

text = text + '\nAppliance ' + str((j)) + ': ' + str(i)

j=j+1

msg=MIMEText(text)

msg['Subject'] = "Testing"

msg['From'] = sendac

msg['To'] = myac

sendser.sendmail(sendac,myac,msg.as\_string())

print 'Status has been sent'

return

def sendingintrude():

sendac ="sendingac1@gmail.com" #Sending Account Information

pwdsend = "sendingaccount1"

myac= "receivingac1@gmail.com" #Receiving Account Information

sendser = smtplib.SMTP('smtp.gmail.com:587')

sendser.ehlo\_or\_helo\_if\_needed()

sendser.starttls()

sendser.ehlo\_or\_helo\_if\_needed()

sendser.login(sendac,pwdsend)

msg = MIMEMultipart()

msg['Subject'] = "Intruder Alert"

msg['From'] = sendac

msg['To'] = myac

body = 'Someboy trying to enter house!'

msg.attach(MIMEText(body,'plain'))

filename='xyz.jpg'

attachment =open(filename,'rb')

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

part.set\_payload((attachment).read())

encoders.encode\_base64(part)

part.add\_header('Content-Disposition',"attachment; filename= "+filename)

msg.attach(part)

text = msg.as\_string()

sendser.sendmail(sendac,myac,msg.as\_string())

return

def receiving(): #function to receive email

resp = feedparser.parse("https://" + sendac + ":" + pwdsend + "@mail.google.com/gmail/feed/atom")

unread = int(resp["feed"]["fullcount"]) #checking if any unread mail

if unread==0: #if no unread mail

print ('Mail Not Received')

else:

subj = resp['items'][0].title

if subj=='Testing': #checking if mail is one we need

stat, count = server.select('Inbox')

stat, data = server.fetch(count[0], '(UID BODY[TEXT])')

temp = data[0][1]

ogh = ''

new = ''

for i in range(76,len(temp)):

ogh = ogh + temp[i]

interpret(ogh)

state = takin() #new pin status

sending(state)

else:

print "Testing Mail not received."

return

def interpret(text):

d={"A1":"pin0","A2":"pin1","A3":"pin2","A4":"pin3"}

dd={"0":"0","1":"20","2":"40","3":"60","4":"80","5":"100"}

tosend=''

rep="GPIO.output("

nn=''

aa=0

for i in text:

nn=nn+i

if i=='~':

break

if i=='-':

nn=nn.replace('-','')

nn=nn.replace('\n','')

nn=nn.replace('\r','')

fin=d[nn]

if nn=="A4":

tosend=tosend+'p.ChangeDutyCycle('

aa=1

else:

tosend=tosend+rep+fin+",GPIO."

nn=''

if i=='.':

nn=nn.replace('.','')

nn=nn.replace('\n','')

nn=nn.replace('\r','')

print nn

if aa==1:

nn=dd[nn]

aa=0

if nn=='exit':

tosend='exit()'

elif nn=='state':

state=takin()

sending(state)

else:

tosend=tosend+nn+")"

print tosend

exec(tosend)

nn=''

tosend=''

return

pw = getpass.getpass()

if pw=='1235':

print ('You may proceed.')

#get current status and print

else:

print 'Incorrect. One more try.'

pw = getpass.getpass()

if pw=='1235':

print ('You may proceed.')

else:

print('Intruder alert!!/n Capturing photo!')

with picamera.PiCamera() as camera:

camera.resolution=(1280,720)

camera.capture("xyz.jpg")

sendingintrude()

exit()

sendac ="sendingac1@gmail.com" #Sending Account Information

pwdsend = "sendingaccount1"

myac= "receivingac1@gmail.com" #Receiving Account Information

sendser = smtplib.SMTP('smtp.gmail.com:587')

sendser.ehlo\_or\_helo\_if\_needed()

sendser.starttls()

sendser.ehlo\_or\_helo\_if\_needed()

sendser.login(sendac,pwdsend)

server=imaplib.IMAP4\_SSL("imap.gmail.com",993)

server.login(sendac,pwdsend)

#Sending Initial Status Email

state = takin()

sending(state)

time.sleep(3)

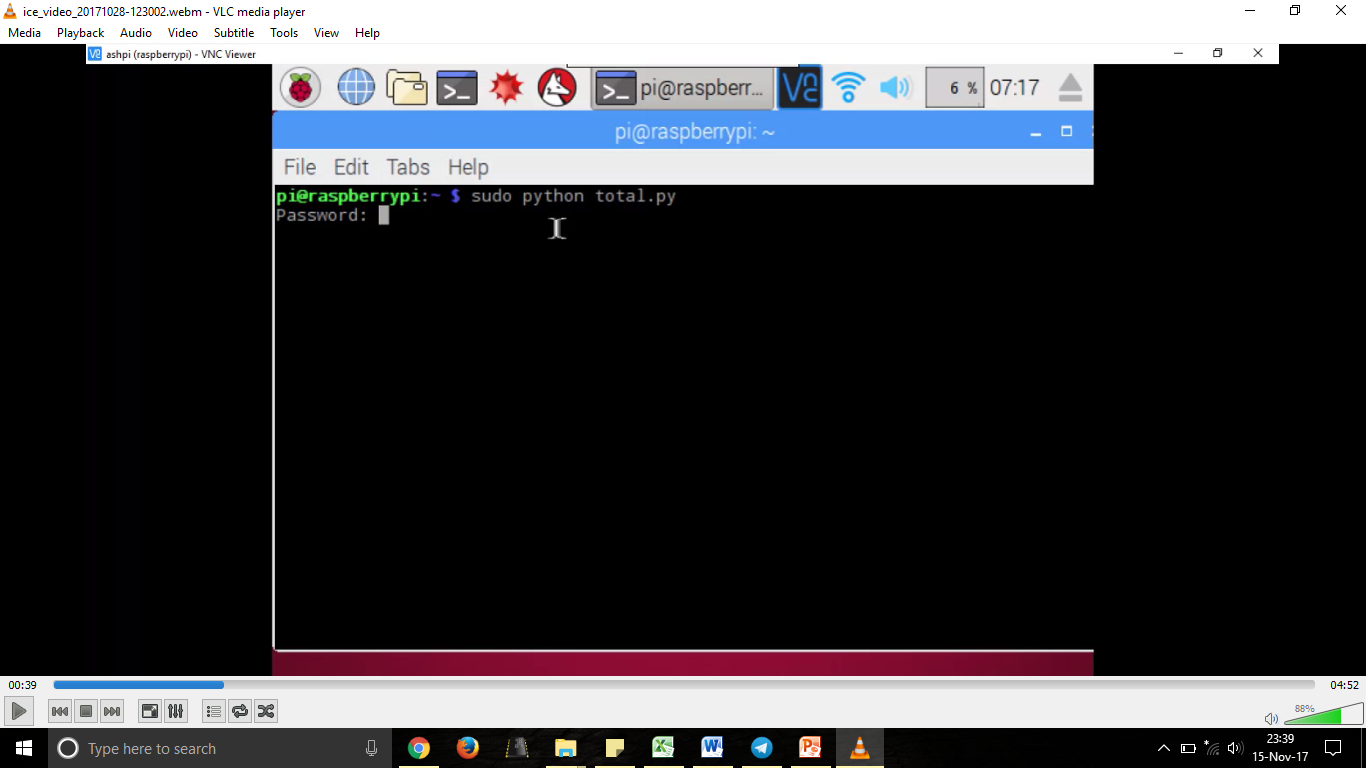
while(1):

receiving()

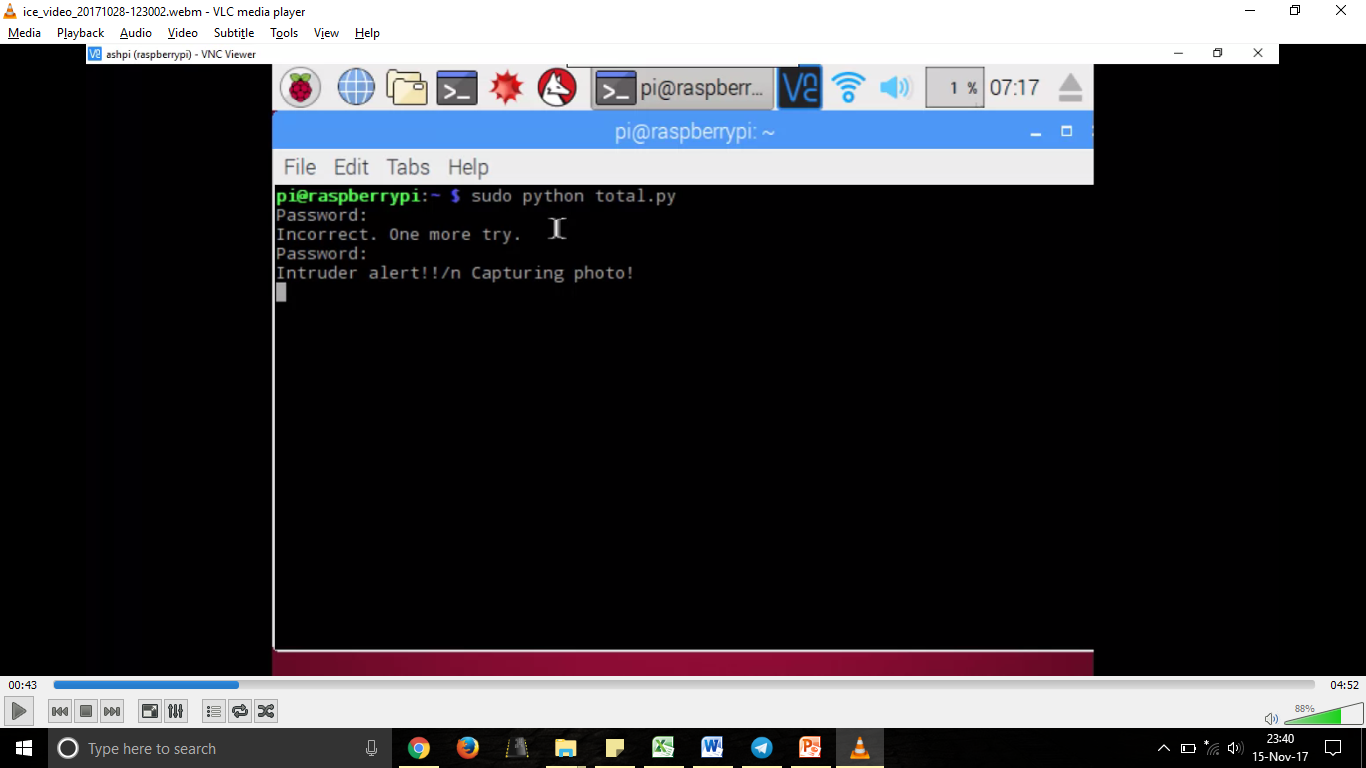
time.sleep(2)

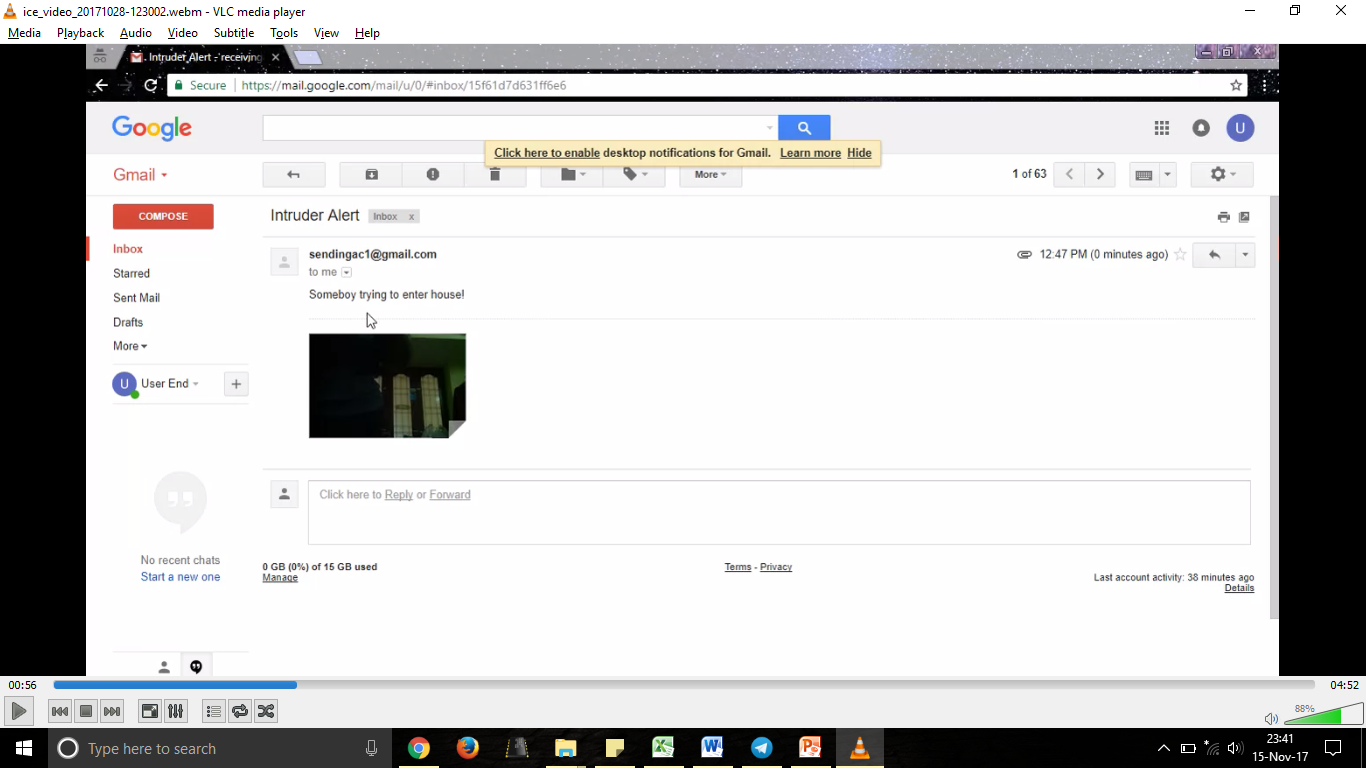
**Output**

1. **Asking for password:**



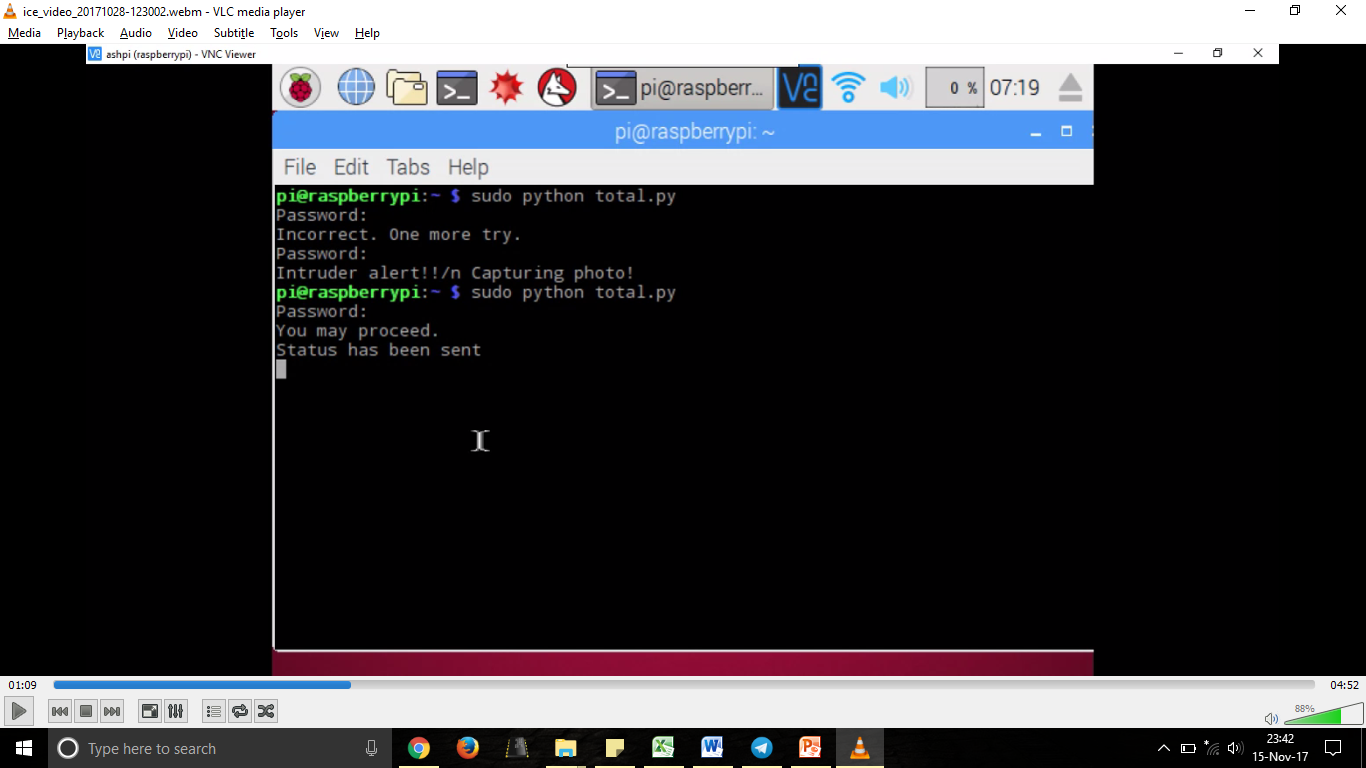
**Incorrect password, intruder alert:**



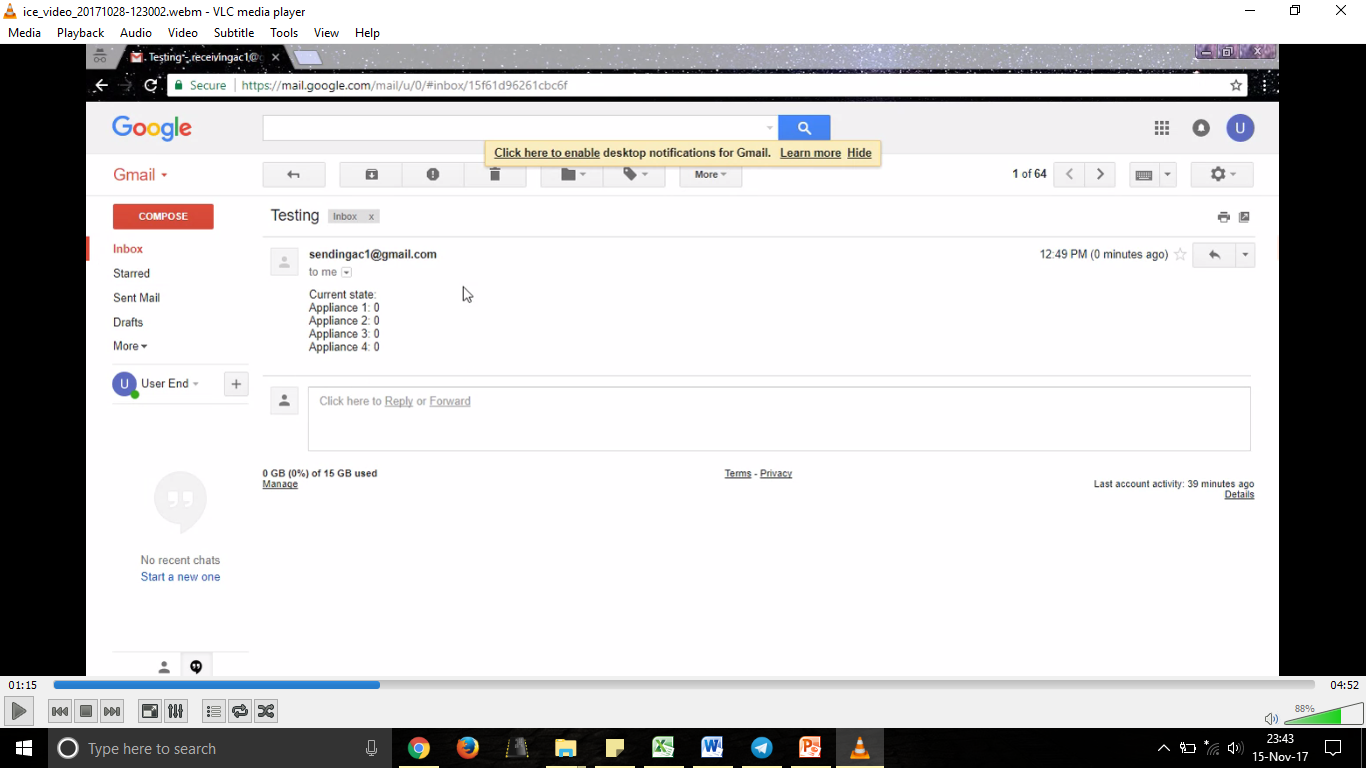


**Intruder picture sent.**

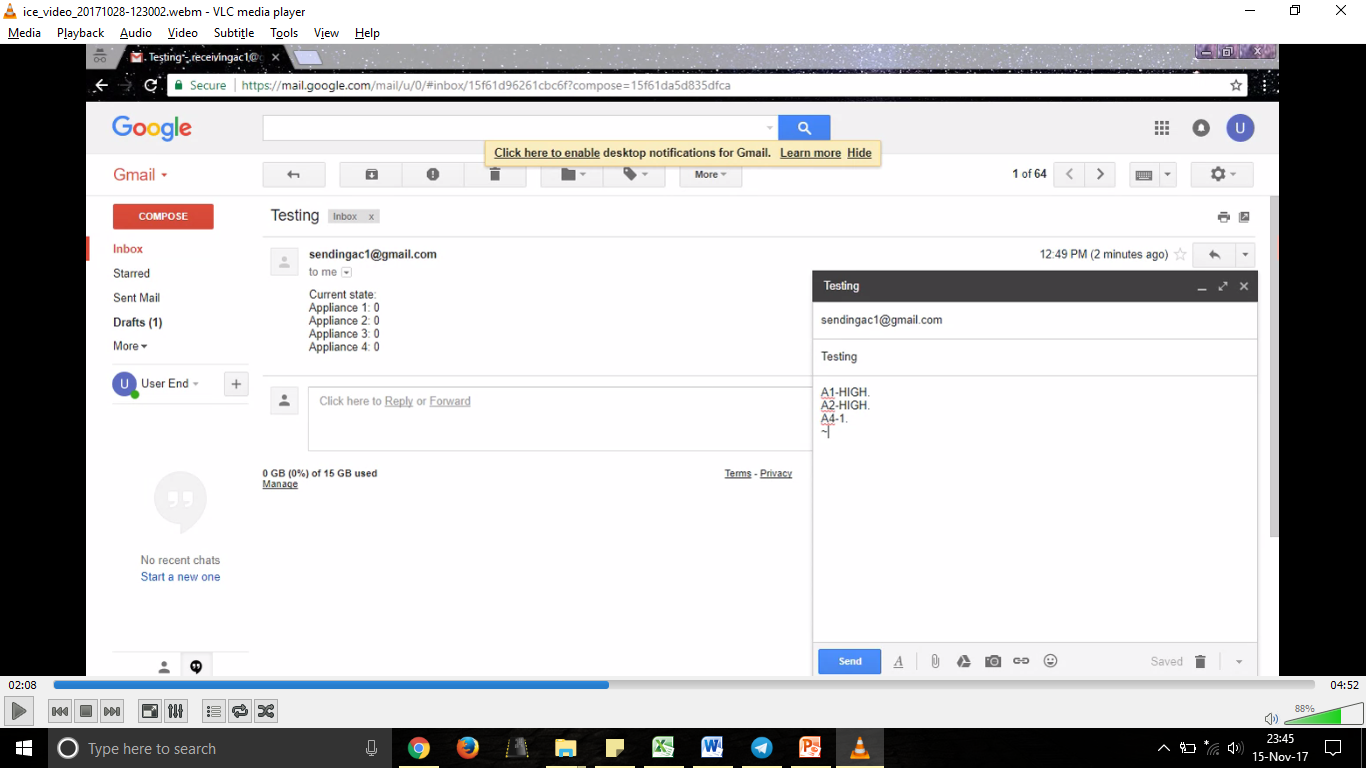
1. **Asking for password, giving the correct one:**



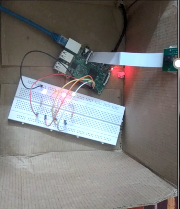
**Sending Current Status to the user email:**



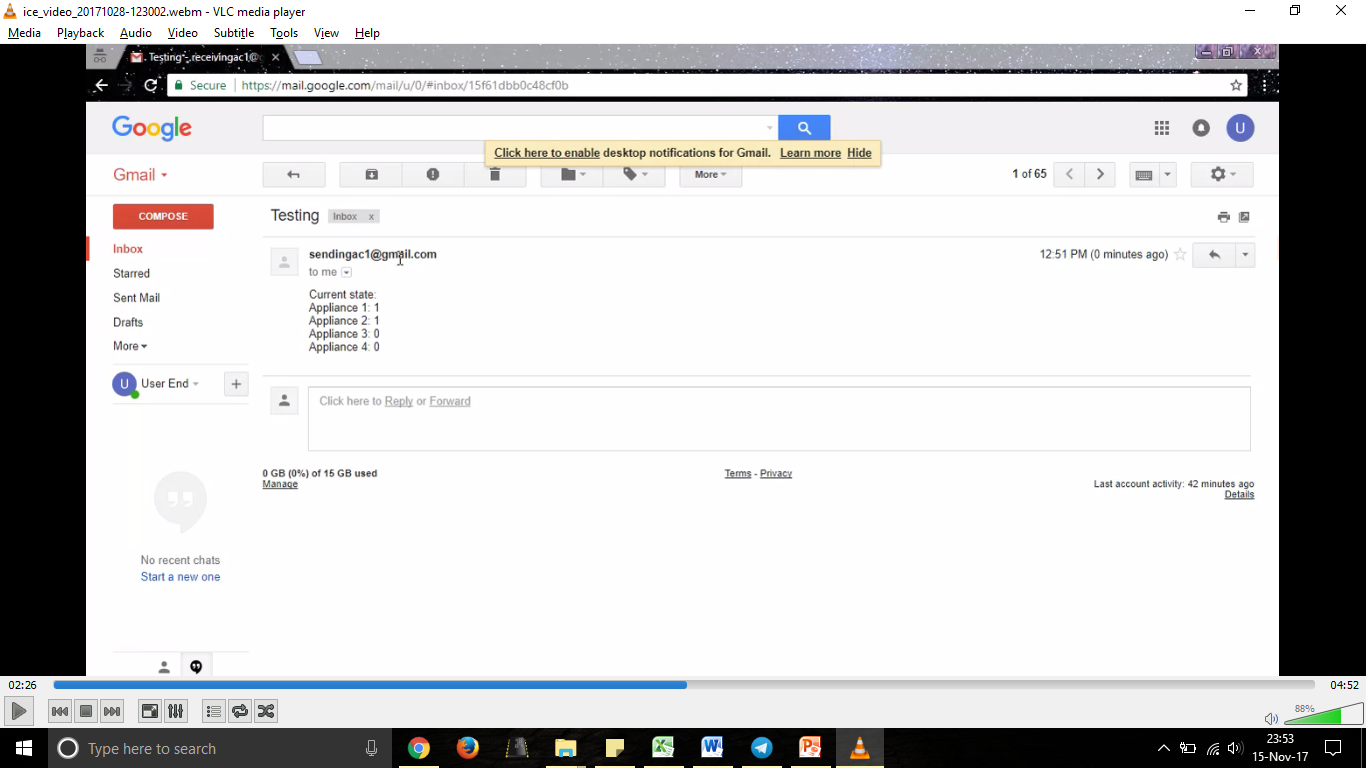
**Sending mail to change appliance status:**



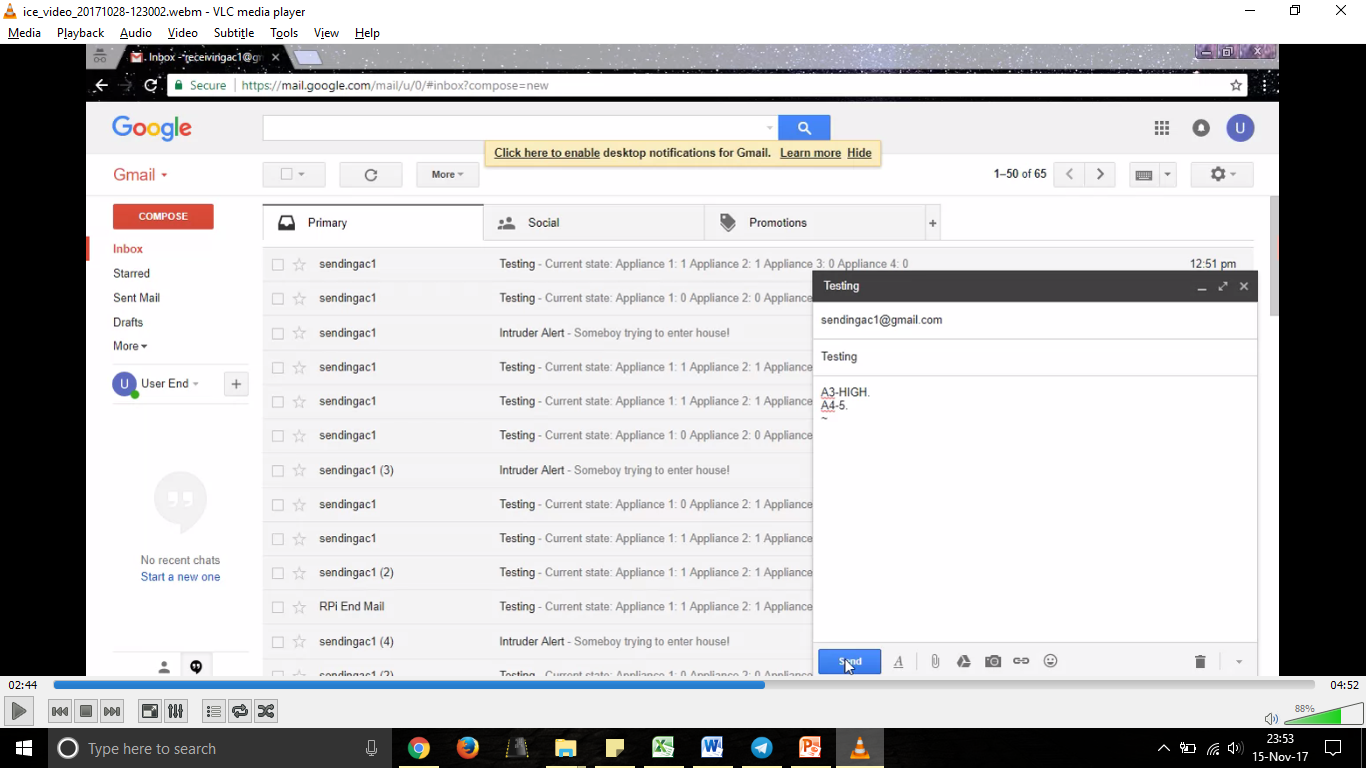
**Output LEDs:**



New status sent:



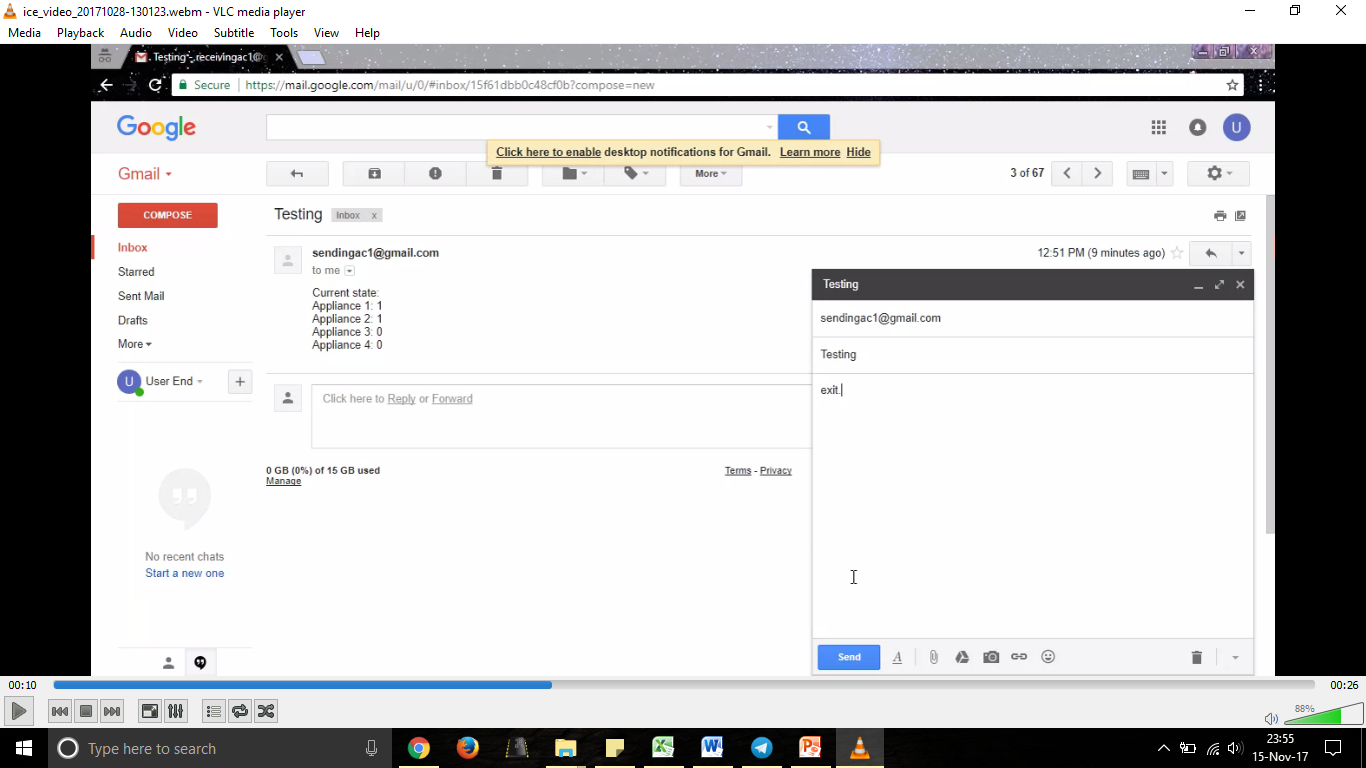
**Change of status again:**

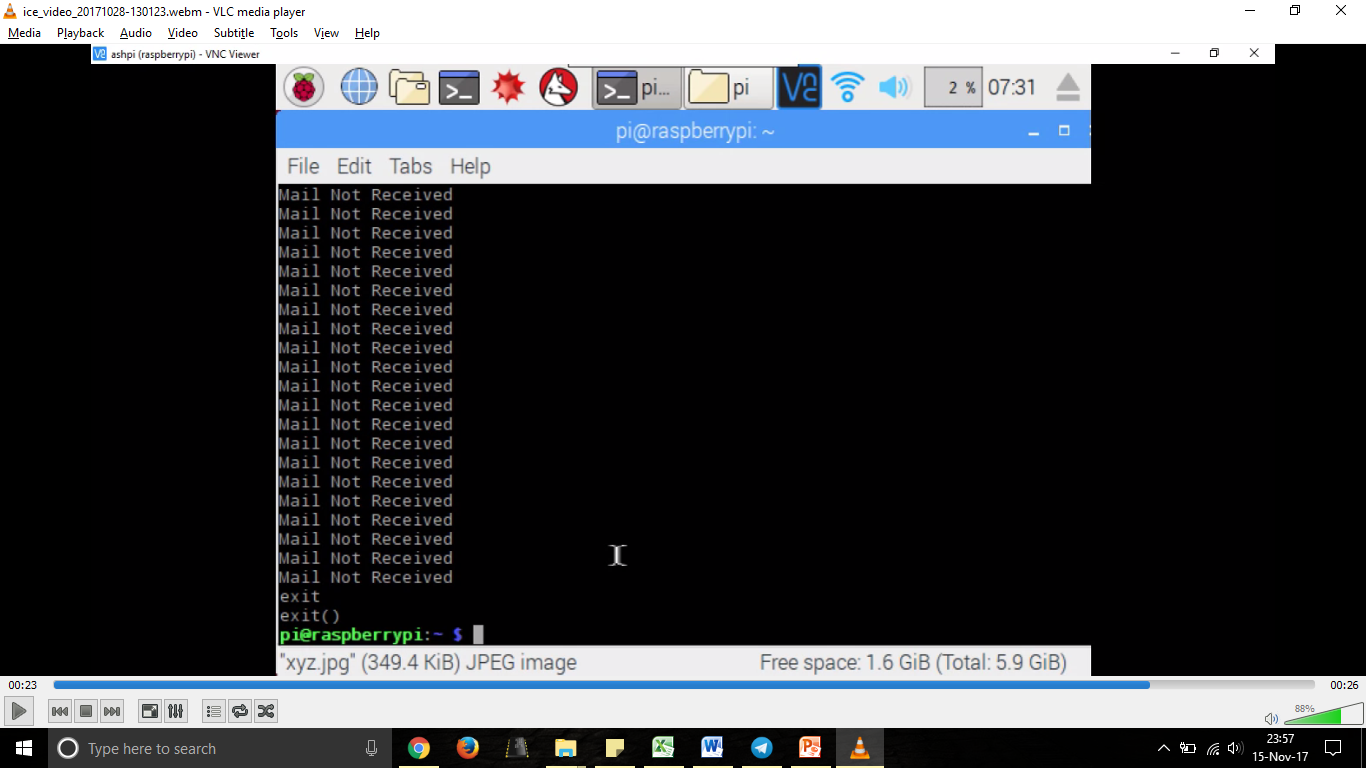


**Output LEDs:**



**To exit:**





\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_THANK YOU\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_