

Smart doorbell using Raspberry pi.

This article is about smart doorbell which is another useful application of IoT. IoT is basically an interconnection of things which communicate over the internet. IoT has been a point of focus of many companies these days and this DIY can help understanding the basic essence of IoT.

This smart doorbell captures an image whenever there is a visitor and sends the photograph as an attachment to the email id specified. You can see who has arrived on your phone without even opening the door. This superpower is by dint of IoT.

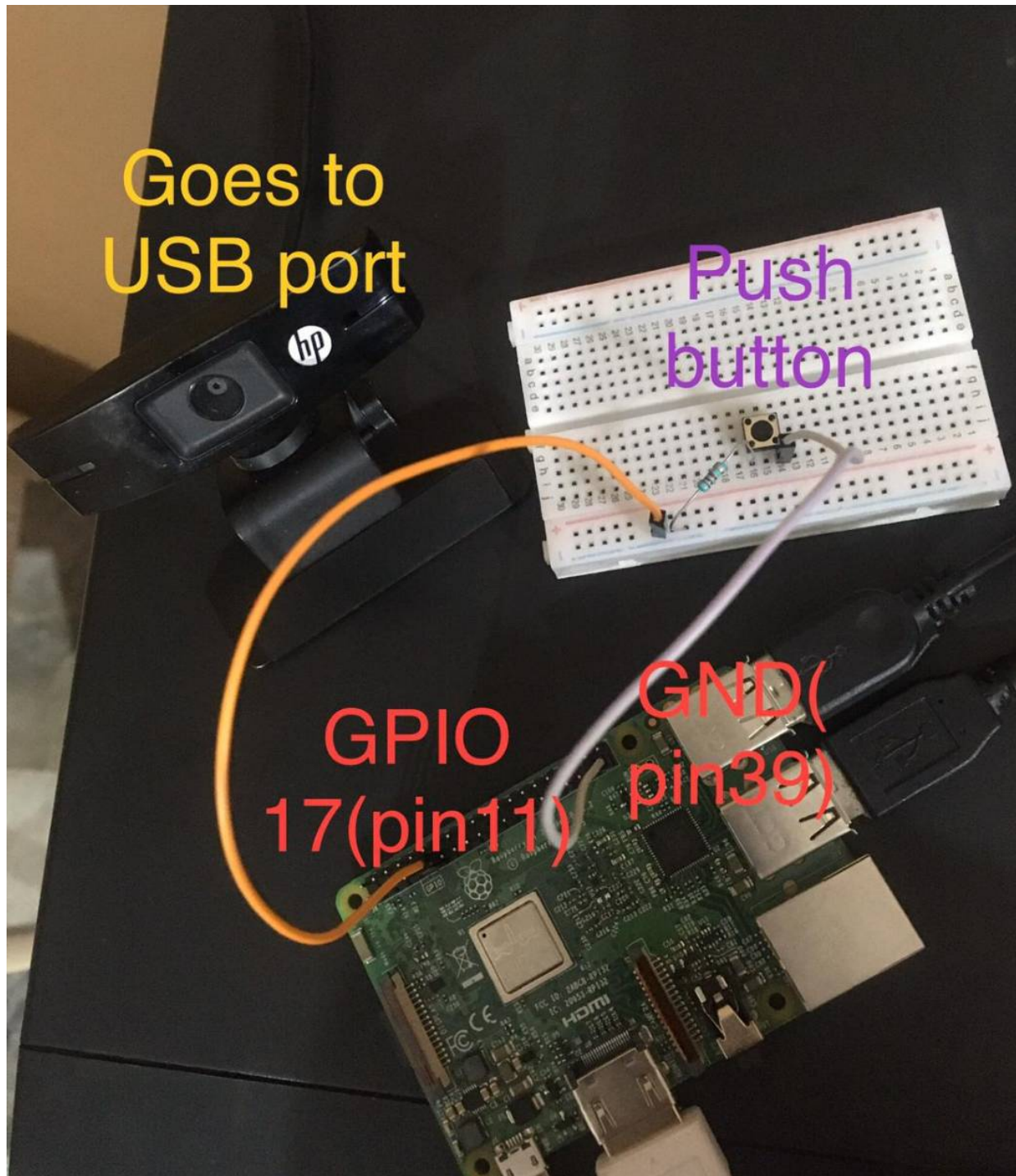
You can also use pi camera for this but I made use of a USB webcam.

THINGS REQUIRED:

1. Raspberry pi (I used model 3 B)with Raspbian OS installed.
2. Jumper wires
3. 220 ohm resistor.
4. USB camera (you can also use pi camera)
5. Power supply.

CIRCUIT CONNECTIONS:

Place a push button into the breadboard. It's one end goes to ground pin of Raspberry pi and other goes to GPIO 17(pin 11) via a resistor (220 ohm). Find the circuit connections below.



ESSENTIAL PREREQUISITE:

To get email alerts without any interruption, **turn on access for less secure apps**. This is an essential step as Google will block the emails it doesn't trust but you can stop this by going to Google your account and then to 'Security'-->'turn on access to less secure apps'. This step ensures you get the email.





Personalization

Security

People & sharing

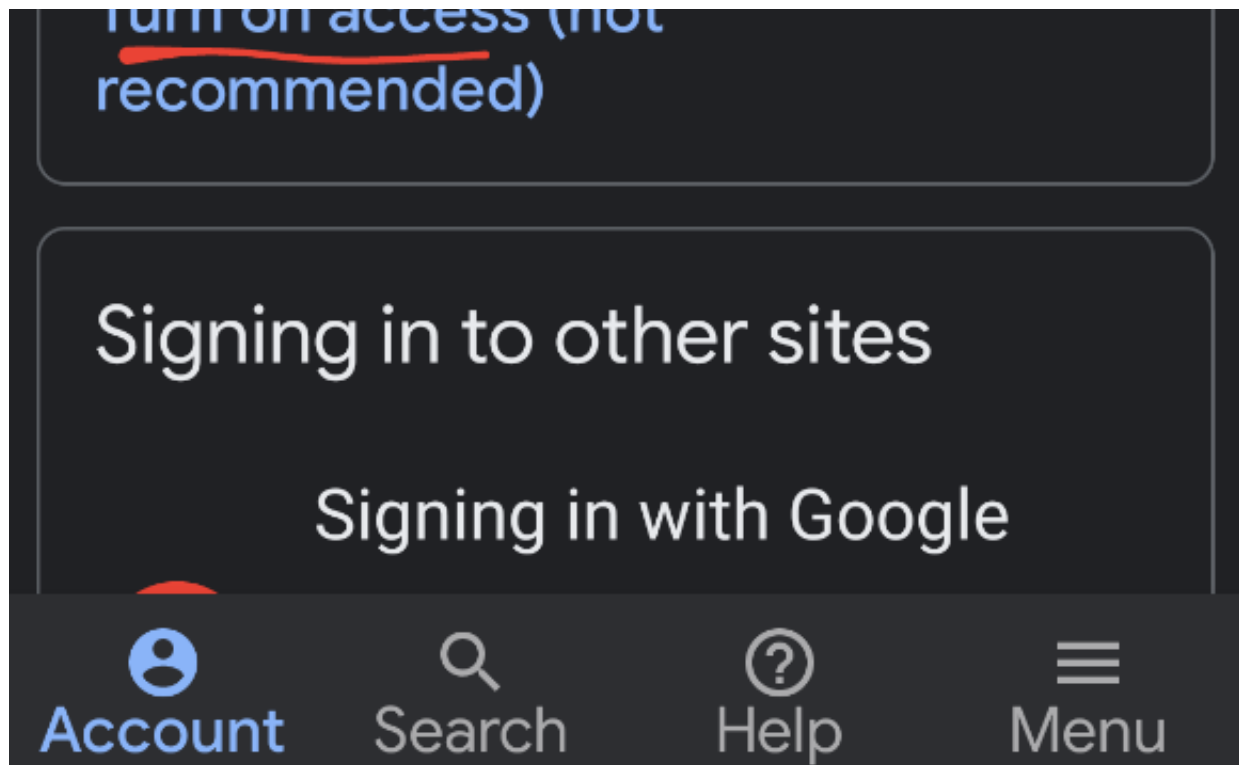
Less secure app access

To protect your account, apps and devices that use less secure sign-in technology are blocked. To keep your account secure, Google will automatically turn this setting OFF if it's not being used. [Learn more](#)



☐ Off

[Turn on access \(not recommended\)](#)



USB WEBCAM:

Raspberry pi installation is pretty simple. Use a memory card(at least 8 GB) to install NOOBS or Raspbian from the official site. Head on to the official site for more on installation of OS steps.

For using USB webcam you should install '**fswebcam**':

```
sudo apt-get install fswebcam
```

This is a basic command line utility that captures images with webcam. Verify installation using:

```
fswebcam filename.jpg
```

It will be saved in the **/home/pi** directory by default with the filename specified, as 'filename.jpg'

If you want to store the images in another directory then

```
mkdir /home/pi/output
```

The above command creates a new folder with name '**output**'.

To capture an image using a python script, one way is as below:

```
import os
import time
os.system("fswebcam -r 960x720 -d /dev/video0 /home/pi/output/file.jpg")
```

What we desire, is to capture an image when the push button is pressed. This can be achieved using :

```
#Example code that captures an image on being triggered by a button.
import os
import RPi.GPIO as GPIO
#adjust for where your button is connected
buttonPin = 17
GPIO.setmode(GPIO.BCM)
GPIO.setup(buttonPin, GPIO.IN)
#if the button is pressed capture a picture with specified resolution in the specified
if (GPIO.input(buttonPin)):
    os.system("fswebcam -r 960x720 -d /dev/video0 /home/pi/output/file.jpg")
```

TO SEND AN EMAIL:

SMTP(Simple Message Transfer Protocol) is used to send emails via command line or Python scripts. It is a collection of communication instructions that allows to send an electronic mail over the internet.

It can send a message to one or more recipients which can include text, voice or video.

1. First off, update and upgrade Raspbian OS using:

```
sudo apt-get update
sudo apt-get upgrade
```

2. Install SMTP library packages as below:

```
sudo apt-get install ssmtp
```

```
sudo apt-get install mailutils
```

CODE:

1. In the editor of your choice import the following libraries. I used Thony editor in Raspberry pi.

```
import RPi.GPIO as GPIO
import time
import os
import smtplib
```

2. Import the modules that are required to send email. **Multipurpose Internet Mail Extensions (MIME)** extends the email format for writing plain text, including attachment and subject and other application programs. Email messages with MIME formatting are transmitted with standard protocols like **SMTP**.

```
from email.mime.multipart import MIMEMultipart
from email.mime.base import MIMEBase
from email.mime.text import MIMEText
from email.utils import COMMASPACE, formatdate
from email import encoders
```

2. Store your username and password in two different variables.

```
USERNAME = 'type_your_email_id@gmail.com'
```

```
PASSWORD = 'type_your_password'
```

3. Set 'push button' to GPIO PIN17 of Raspberry pi so that it can act as a switch of doorbell.

```
buttonPin = 17
#using GPIO.BCM to use GPIO pins.
GPIO.setmode(GPIO.BCM)
GPIO.setup(buttonPin, GPIO.IN)
```

4. Whenever the button is pressed, the image is captured and the function to send the email is called.

```
while True:
    if (GPIO.input(buttonPin)):
        #captures image and saves at the specified path
        os.system("fswebcam -r 960x720 -d /dev/video0 /home/pi/output/file.jpg")
        sendMail( ["type_receiver's_email_id@gmail.com"],
                  "Doorbell notification",
                  "Someone is ringing the doorbell, picture attached",
                  ["/home/pi/output/file.jpg"] )
```

5. Let us define this function which sends the email. It accepts as arguments an email id, subject and body of the and the attachments and sends the email using SMTP.

```
def sendMail(to, subject, text, files):
    assert type(to)==list
    assert type(files)==list
    msg = MIMEMultipart()
    msg['From'] = USERNAME
    msg['To'] = COMMASPACE.join(to)
    msg['Date'] = formatdate(localtime=True)
    msg['Subject'] = subject

    ..
    ..

    server = smtplib.SMTP('smtp.gmail.com:587')
    server.ehlo_or_helo_if_needed()
    server.starttls()
    server.ehlo_or_helo_if_needed()
    server.login(USERNAME, PASSWORD)
    server.sendmail(USERNAME, to, msg.as_string())
    server.quit()
```

FINAL CODE:

```
import RPi.GPIO as GPIO
import time
import os

import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.base import MIMEBase
```

```

from email.mime.text import MIMEText
from email.utils import COMMASPACE, formatdate
from email import encoders

#adjust for where your button is connected
buttonPin = 17
GPIO.setmode(GPIO.BCM)
GPIO.setup(buttonPin, GPIO.IN)

USERNAME = 'type_your_email_id@gmail.com'
PASSWORD = 'type_your_password'

def sendMail(to, subject, text, files):
    assert type(to)==list
    assert type(files)==list

    msg = MIMEMultipart()
    msg['From'] = USERNAME
    msg['To'] = COMMASPACE.join(to)
    msg['Date'] = formatdate(localtime=True)
    msg['Subject'] = subject

    msg.attach( MIMEText(text) )

    for file in files:
        part = MIMEBase('application', "octet-stream")
        part.set_payload( open(file,"rb").read() )
        encoders.encode_base64(part)
        part.add_header('Content-Disposition', 'attachment; filename="%s"' % os.path.b
        msg.attach(part)

    server = smtplib.SMTP('smtp.gmail.com:587')
    server.ehlo_or_helo_if_needed()
    server.starttls()
    server.ehlo_or_helo_if_needed()
    server.login(USERNAME, PASSWORD)
    server.sendmail(USERNAME, to, msg.as_string())
    server.quit()

while True:
    if (GPIO.input(buttonPin)):
        #this is the script that will be called (as root)
        os.system("fswebcam -r 960x720 -d /dev/video0 /home/pi/output/file.jpg")
        sendMail( ["type_your_email_id@gmail.com"],
            "Doorbell notification",
            "Someone is ringing the doorbell, picture attached",
            ["/home/pi/output/file.jpg"] )

```

When you run this script you get an email within 7-8 seconds.

To avoid manually running the script, make it executable so that it runs automatically on every boot. Open **rc.local** file:

```
sudo nano /etc/rc.local
```

Then paste the path to your python script before **exit 0**.

Now reboot Raspberry Pi and you are done. Press the push button to receive the email at the comfort of your room.

