Capture Image & Record Videos With Raspberry Pi and Pi Camera

Installation Manual

In this tutorial, we are demonstrating Visitors Monitoring System with Image capture and recording functionality. This Monitoring system will digitize and automate the whole visitor entries, and there will be no need to maintain them manually.

We are interfacing Pi camera with Raspberry Pi to capture the image and record videos of every visitor which has entered through the Gate or door. This captured image is saved in the system with the Date and time of the entry.

This system is very useful in offices, factories where visitor entry record is maintained for visitors, employees. So this can be very useful for security purpose.

Hardware Requirements

- Raspberry Pi , SD Card
- Pi camera
- Buzzer
- 2 Push Buttons
- LED
- Bread Board
- Jumper Wires
- Power supply
- Ethernet Cable / Wi Fi
- Monitor , Keyboard, Mouse (**Optional** ,For without headless connection)

Software Requirements

1. Raspbian Stretch OS
2. Python Script

System Explanation

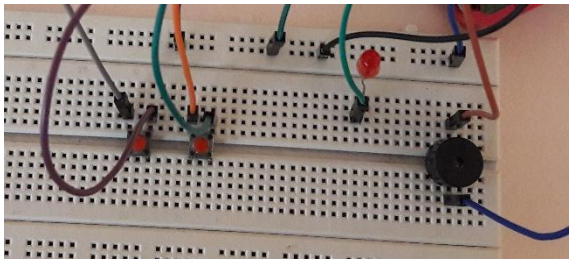
- **Pi camera** is used to capture the images and record the videos of visitors, when a **push button** is pressed or triggered.
- After **pushing the one button**, Raspberry Pi sends command to Pi Camera to **click the picture** and save it.
- Similarly, after **pushing another button**, Raspberry Pi sends command to Pi Camera to **record the video** and save it.
- The **buzzer** is used to generate sound when button pressed.
- **LED** is used for indicating that Raspberry Pi is ready to accept Push Button press, means when LED is ON, and system is ready for operation.

Circuit Explanation

❖ Connect Camera to Raspberry Pi

The Camera Module is a great accessory for the Raspberry Pi, allowing users to take still pictures and record video in full HD. Connect the **Camera Module** to the Raspberry Pi's camera port (Blue side of cable to face Ethernet connection).

❖ Connect LED,Push Button ,Buzzer with Raspberry Pi's GPIO Pins.



The buzzer is used to generate sound when button pressed.

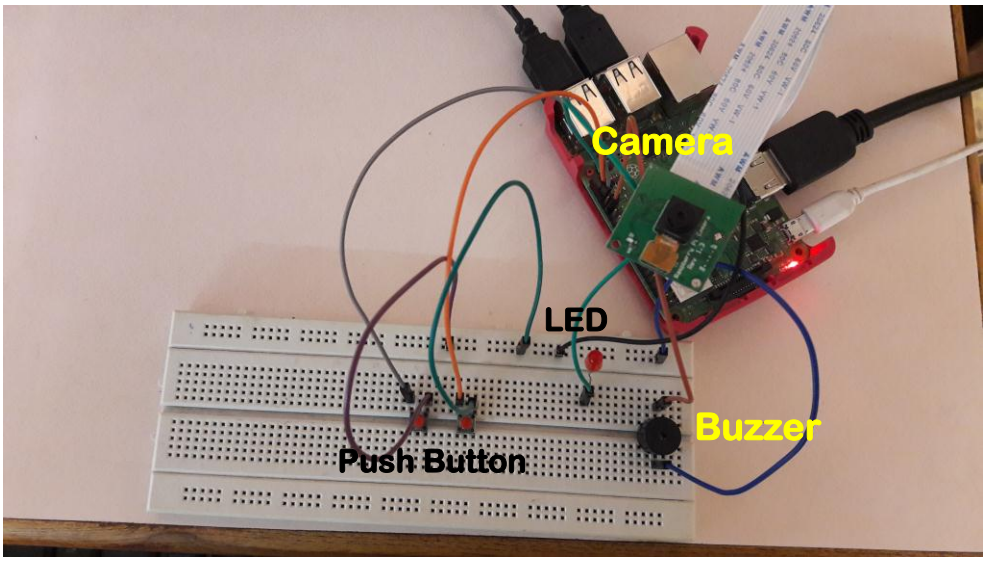
LED is used for indicating that Raspberry Pi is ready to accept **Push Button** press, means when LED is ON, and system is ready for operation.

I used following GPIO pins to connect above three components to Raspberry Pi.

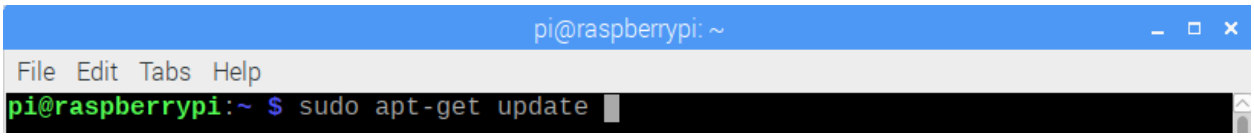
Component	Terminal	RPI Physical Pin	Raspberry Function
Buzzer	Positive Terminal	Pin 32	GPIO 12
Push Button1	Positive Terminal	Pin 36	GPIO 16
Push Button2	Positive Terminal	Pin 38	GPIO 20
LED	Positive Terminal	Pin 40	GPIO 21
all	Negative Terminal	Pin 20	GND

Note that the camera preview only works when a monitor is connected to the Pi, so remote access (such as SSH and VNC) will not allow you to see the camera preview.

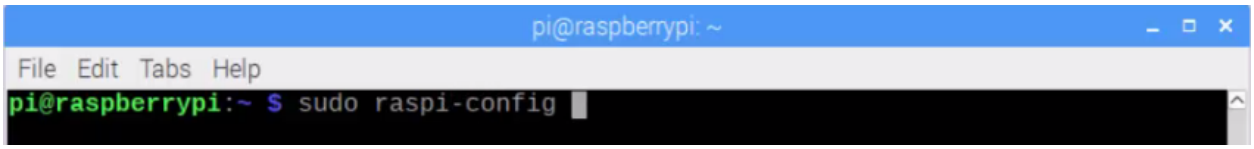
Complete Connection



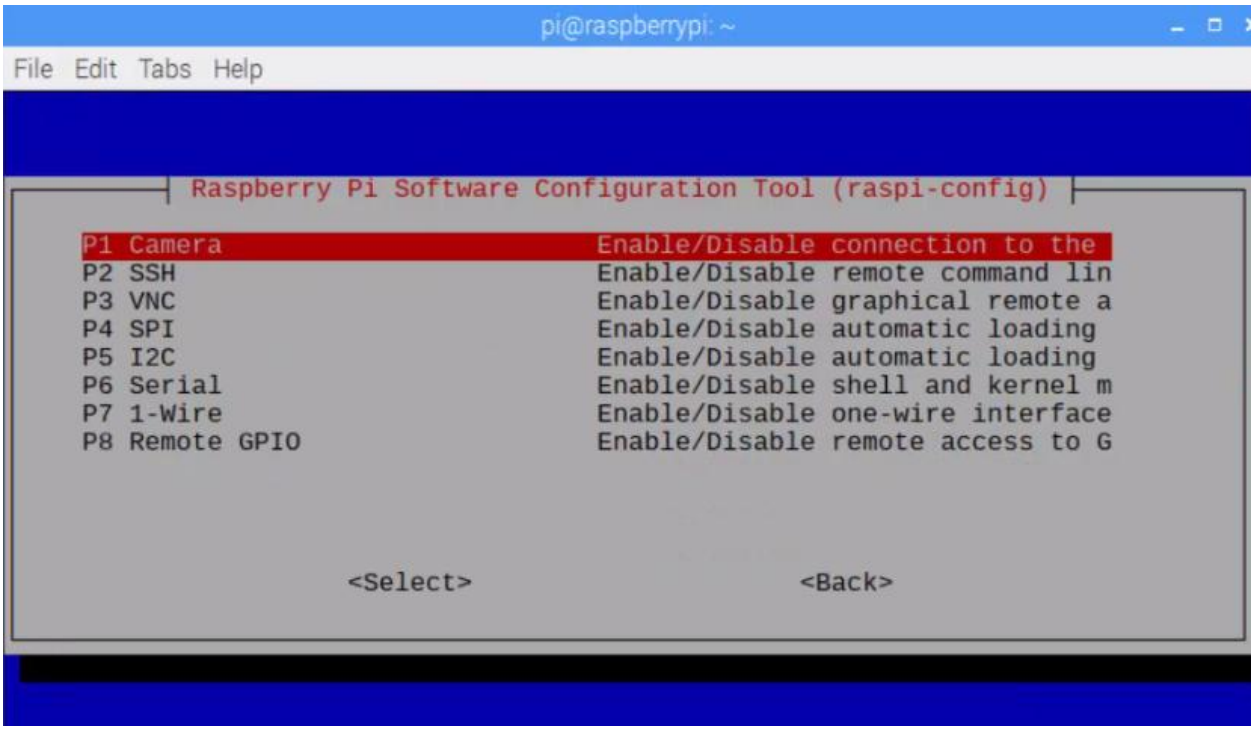
Step 1: Update Raspberry Pi



Step 3: Enable Raspberry Pi Camera by using Raspberry Pi Software Configuration Tool (raspi-config)



Select Interfacing Options → Camera → Enable



After this reboot the system, using **sudo reboot** command.

Step 4: After Reboot, Create a folder which is used to store pictures captured by Pi Camera. (E.g. /home/pi/Desktop/Visitors)

Step 5: Write a Python Script to capture images and save them in folder. (capture_record_picamera.py)

```
import RPi.GPIO as gpio
import picamera
import time

led=21              #pin no 40
buz=12              #pin no 32

butcapture=16       #pin no 36
butrecord=20        #pin no 38
```

```
HIGH=1
LOW=0

gpio.setwarnings(False)
gpio.setmode(gpio.BCM)
gpio.setup(led, gpio.OUT)
gpio.setup(buz, gpio.OUT)
gpio.setup(butcapture, gpio.IN, pull_up_down=gpio.PUD_UP)
gpio.setup(butrecord, gpio.IN, pull_up_down=gpio.PUD_UP)
gpio.output(led , 0)
gpio.output(buz , 0)
data=""

def capture_image():
    print('Please Wait...')
    data= time.strftime("%d_%b_%Y\%H:%M:%S")
    camera.start_preview()
    time.sleep(5)
    print(data)
    camera.capture('/home/pi/Desktop/Visitors/%s.jpg'%data)
    camera.stop_preview()
    print('Image Captured successfully')
    time.sleep(2)

def record_video():
    print('Please Wait...')
    data= time.strftime("%d_%b_%Y\%H:%M:%S")
    camera.start_preview()
    camera.start_recording('/home/pi/Desktop/Visitors/rec.h264')
    time.sleep(5)
    print(data)
    camera.stop_recording()
    camera.stop_preview()
    print('Video recorded successfully')
    time.sleep(2)

print('Welcome to my System')
time.sleep(2)

print('Visitor Monitor System using RPi')
time.sleep(3)

camera = picamera.PiCamera()
camera.rotation=180
camera.awb_mode= 'auto'
camera.brightness=55
```

```
print(" Please Press  Button")
time.sleep(2)
try:
   while 1:
         d= time.strftime("%d %b %Y")
         t= time.strftime("%H:%M:%S")
         print("Time: %s"%t)
         print("Date:%s"%d)
         gpio.output(led, 1)
         if gpio.input(butcapture)==False:
            gpio.output(buz, 1)
            gpio.output(led, 0)
            time.sleep(0.5)
            gpio.output(buz, 0)
            capture_image()

         if gpio.input(butrecord)==False:
            gpio.output(buz, 1)
            gpio.output(led, 0)
            time.sleep(0.5)
            gpio.output(buz, 0)
            record_video()

         time.sleep(0.5)

except KeyboardInterrupt:
   print("Done.....")
   time.sleep(3)
   gpio.output(led, 0)
finally:
   exit(0)
```

1. **Import** picamera ,gpio ,
2. **Initialize and setup** all GPIO(BCM Pattern)Pins used for led,buzzer,push button
3. Write **capture_image()** function to capture images and saved them in specified **folder with date & time.**
4. Write **record_video()** function to record videos and saved them in specified **folder with rec.h264 filename**
5. When you execute script, first it displays some **welcome messages on terminal**
6. After welcome messages, it will ask you to **Press Button,** and Displays **current Date & Time** on **terminal,**and **LED gets on** ,indicating Ready for process.
7. When you **press first push button**,
    a. first ,**buzzer generates sound**
    b. then **led gets off**,
    c. after half second ,
    d. **Stops buzzer sound**,
    e. And **call capture_image()** function which helps to capture image & saved in folder.

8. When you **press second push button**,
    a. first ,**buzzer generates sound**
    b. then **led gets off**,
    c. after half second ,
    d. **Stops buzzer sound**,
    e. And **call record_video()** function which helps to record the videos & saved in folder.

Step 6: Execute above script using python capture_record_picamera.py

Some useful methods

- **camera.capture()** is used to capture the still pictures. Provide Path(where to save) and image name as parameter.
- **start_recording()** is used to start the recording of the videos. Provide Path(where to save) and video name as parameter. Raspberry Pi' video supports .ht64 as extension.
- **stop_recording()** is used to stop the recording.
- To play the video, need to open a terminal window. Type the command **omxplayer videonme.h264** and Press Enter to play the video.
- It may actually play at a faster speed than what has been recorded, due to omxplayer's fast frame rate.

Effects

- The resolution of the capture is configurable. By default it's set to the resolution of your monitor, but the **maximum resolution is 2592 x 1944 for still photos** and **1920 x 1080 for video recording.** The **minimum resolution allowed is 64 x 64.** To set the resolution to max use following method. Note that you'll also need to set the frame rate to 15 to enable this maximum resolution.

```
camera.resolution = (2592, 1944)
camera.framerate = 15
camera.start_preview()
sleep(5)
camera.capture('/home/pi/Desktop/Visitors/max.jpg')
camera.stop_preview()
```

- To add text to your image with **annotate_text.**

```
camera.start_preview()
camera.annotate_text = "Hello world!"
sleep(5)
camera.capture('/home/pi/Desktop/Visitors/text.jpg')
camera.stop_preview()
```

- To set the **annotation text size** with the following code. Valid sizes are 6 to 160. The default is 32

```
camera.annotate_text_size = 50
```

- To alter the annotation colors. First of all, ensure that **Color** is imported by amending your **import** line at the top.

```
from picamera import PiCamera, Color
```

Then amend the rest of your code as follows:

```
camera.start_preview()
camera.annotate_background = Color('blue')
camera.annotate_foreground = Color('yellow')
camera.annotate_text = " Hello world "
sleep(5)
camera.stop_preview()
```

- To alter the **brightness** setting, which can be set from **0 to 100**. The default is 50. Try setting it to another value.

```
camera.start_preview()
camera.brightness = 70
sleep(5)
camera.capture('/home/pi/Desktop/Visitors/bright.jpg')
camera.stop_preview()
```

- To alter the **Contrast** setting, which can be set from **0 to 100**. The default is 50. Use similar to brightness

```
camera.start_preview()
camera.contrast = 70
sleep(5)
camera.capture('/home/pi/Desktop/Visitors/contrast.jpg')
camera.stop_preview()
```

- Use **camera.image_effect** to apply a particular image effect. The options are: none, negative, solarize, sketch, denoise, emboss, oilpaint, hatch, gpen, pastel, watercol or, film, blur, saturation, colorswap, washedout, posterise, colorpoint, colorbalance, cartoon, deinterlace1, and deinterlace2. The default is none. Pick one and try it out:

```
camera.start_preview()
camera.image_effect = 'colorswap'
sleep(5)
camera.capture('/home/pi/Desktop/Visitors/colorswap.jpg')
camera.stop_preview()
```

- Use **camera.awb_mode** to set the auto white balance to a preset mode to apply a particular effect. The options
areoff, auto, sunlight, cloudy, shade, tungsten, fluorescent, incandescent, flash, and horizon. The default is auto. Pick one and try it out:

```
camera.start_preview()
camera.awb_mode = 'sunlight'
sleep(5)
camera.capture('/home/pi/Desktop/Visitors/sunlight.jpg')
camera.stop_preview()
```

- You can use **camera.exposure_mode** to set the exposure to a preset mode to apply a particular effect. The options
  are: off, auto, night, nightpreview, backlight, spotlight, sports, snow, beach, verylong, fixedfps, anti shake, and fireworks. The default is auto. Pick one and try it out:

```
camera.start_preview()
camera.exposure_mode = 'beach'
sleep(5)
camera.capture('/home/pi/Desktop/Visitors/beach.jpg')
camera.stop_preview()
```

That's all !!!

Thank you….