**8. Write a program to capture image and video using pi camera and motion sensor when motion detected .**

**Introduction**

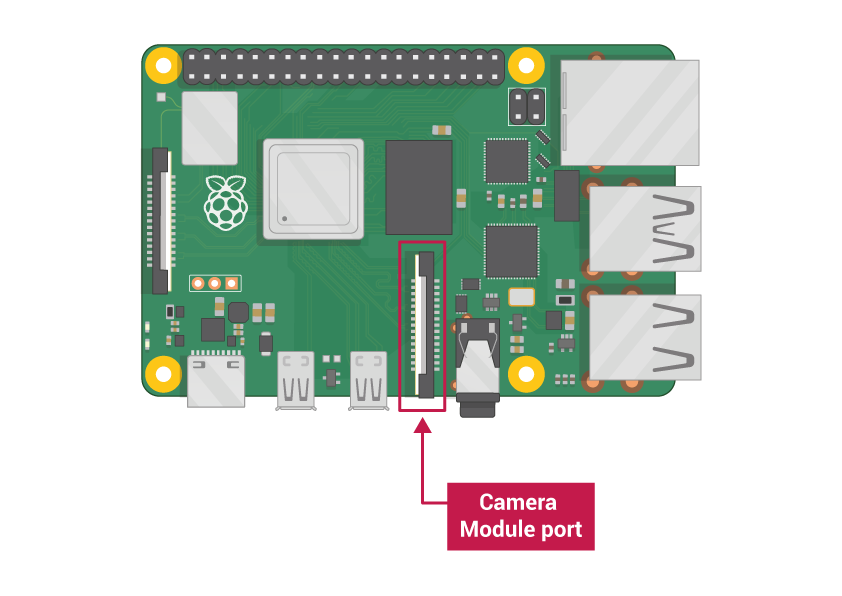
Connect the Raspberry Pi Camera Module to your Raspberry Pi and take pictures, record video, and apply image effects.



## What you will need

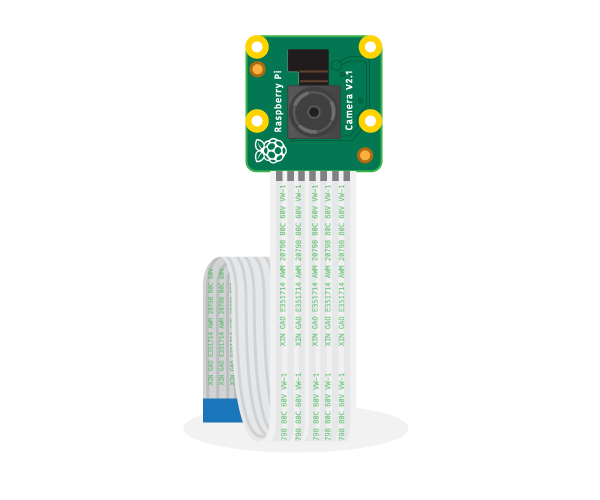
### Raspberry Pi computer with a Camera Module port

All current models of Raspberry Pi have a port for connecting the Camera Module.



**Note:** If you want to use a Raspberry Pi Zero, you need a Camera Module ribbon cable that fits the Raspberry Pi Zero’s smaller Camera Module port.

### Raspberry Pi Camera Module



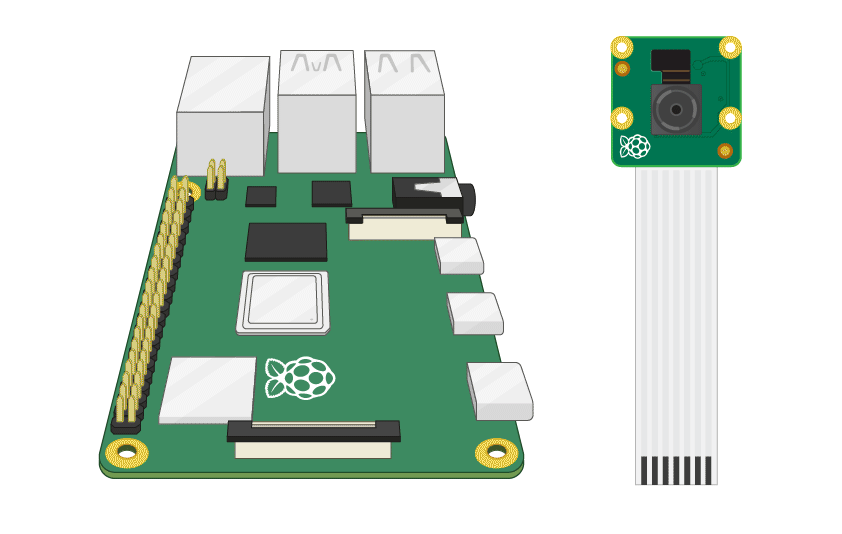
**There are two versions of the Camera Module:**

* [The standard version](https://www.raspberrypi.org/products/camera-module-v2/), which is designed to take pictures in normal light
* [The NoIR version](https://www.raspberrypi.org/products/pi-noir-camera-v2/), which doesn’t have an infrared filter, so you can use it together with an infrared light source to take pictures in the dark

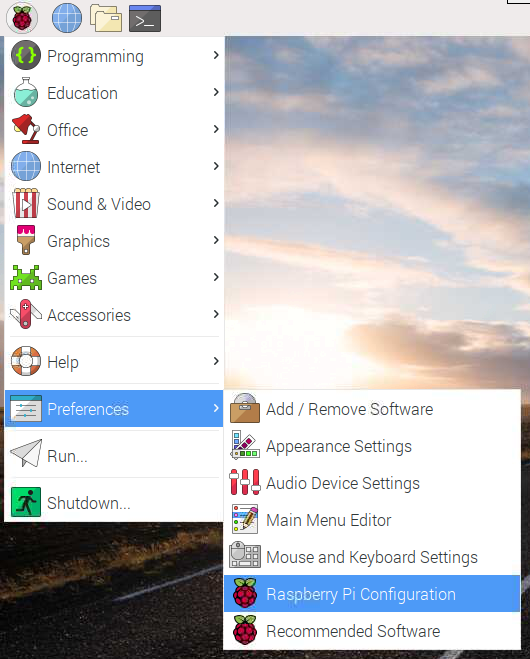
Connect the Camera Module

Ensure your Raspberry Pi is turned off.

1. Locate the Camera Module port
2. Gently pull up on the edges of the port’s plastic clip
3. Insert the Camera Module ribbon cable; make sure the cable is the right way round
4. Push the plastic clip back into place



* Start up your Raspberry Pi.
* Go to the main menu and open the Raspberry Pi Configuration tool.



* Select the Interfaces tab and ensure that the camera is enabled:

Camera enabled

* Reboot your Raspberry Pi.

How to control the Camera Module via the command line

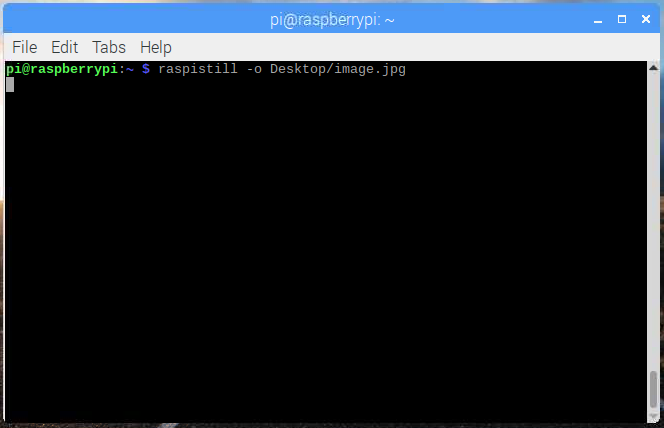
Now your Camera Module is connected and the software is enabled, try out the command line tools raspistill and raspivid.

* Open a terminal window by clicking the black monitor icon in the taskbar:

Open terminal

* Type in the following command to take a still picture and save it to the Desktop:

raspistill -o Desktop/image.jpg



* Press Enter to run the command.

When the command runs, you can see the camera preview open for five seconds before a still picture is taken.

* Look for the picture file icon on the Desktop, and double-click the file icon to open the picture.

Image on Desktop

By adding different options, you can set the size and look of the image the raspistill command takes.

* For example, add –h and –w to change the height and width of the image:

raspistill -o Desktop/image-small.jpg -w 640 -h 480

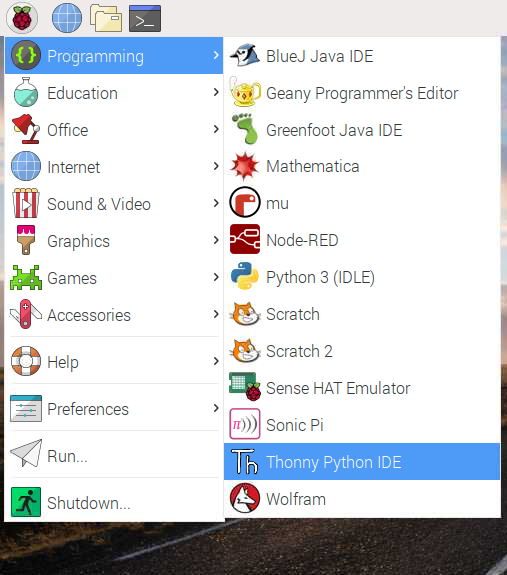
* Now record a video with the Camera Module by using the following raspivid command:

raspivid -o Desktop/video.h264

* In order to play the video file, double-click the video.h264 file icon on the Desktop to open it in VLC Media Player.

The Python picamera library allows you to control your Camera Module and create amazing projects.

* Open a Python 3 editor, such as **Thonny Python IDE**:



* Open a new file and save it as camera.py.

**Note:** it’s important that you **never save the file as** picamera.py.

* Enter the following code:

**from picamera import PiCamera**

**from time import sleep**

**camera = PiCamera()**

**camera.start\_preview()**

**sleep(5)**

**camera.stop\_preview()**

* Save and run your program. The camera preview should be shown for five seconds and then close again.
* f your preview is upside-down, you can rotate it by 180 degrees with the following code:

**camera = PiCamera()**

**camera.rotation = 180**

You can rotate the image by 90, 180, or 270 degrees. To reset the image, set rotation to 0 degrees.

It’s best to make the preview slightly see-through so you can see whether errors occur in your program while the preview is on.

* Make the camera preview see-through by setting an alpha level:

camera.start\_preview(alpha=200)

The alpha value can be any number between 0 and 255.

Take still pictures with Python code

Now use the Camera Module and Python to take some still pictures.

* Amend your code to add a camera.capture() line:

**camera.start\_preview()**

**sleep(5)**

**camera.capture('/home/pi/Desktop/image.jpg')**

**camera.stop\_preview()**

Note: it’s important to sleep for at least two seconds before capturing an image, because this gives the camera’s sensor time to sense the light levels.

* Run the code.

You should see the camera preview open for five seconds, and then a still picture should be captured. As the picture is being taken, you can see the preview briefly adjust to a different resolution.

Your new image should be saved to the Desktop.

* Now add a loop to take five pictures in a row:

**camera.start\_preview()**

**for i in range(5):**

**sleep(5)**

**camera.capture('/home/pi/Desktop/image%s.jpg' % i)**

**camera.stop\_preview()**

The variable i counts how many times the loop has run, from 0 to 4. Therefore, the images get saved as image0.jpg , image1.jpgmage1.jpg, and so on.

* Run the code again and hold the Camera Module in position.

The camera should take one picture every five seconds. Once the fifth picture is taken, the preview closes.

* Look at your Desktop to find the five new pictures.

## Recording video with Python code

Now record a video!

* Amend your code to remove capture() and instead add start recording() and stop recording()

Your code should look like this now:

**camera.start\_preview()**

**camera.start\_recording('/home/pi/Desktop/video.h264')**

**sleep(5)**

**camera.stop\_recording()**

**camera.stop\_preview()**

* Run the code.

Your Raspberry Pi should open a preview, record 5 seconds of video, and then close the preview.

## How to change the image settings and add image effects

The Python picamera software provides a number of effects and configurations to change how your images look.

**Note:** some settings only affect the preview and not the captured image, some affect only the captured image, and many others affect both.

### Set the image resolution

You can change the resolution of the image that the Camera Module takes.

By default, the image resolution is set to the resolution of your monitor. The maximum resolution is 2592×1944 for still photos, and 1920×1080 for video recording.

* Use the following code to set the resolution to maximum and take a picture.

**Note:** you 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/max.jpg')**

**camera.stop\_preview()**

The minimum resolution is 64×64.

* Try taking a picture with the minimum resolution.

### Add text to your image

You can add text to your image using the command annotate\_text.

* Run this code to try it:

**camera.start\_preview()**

**camera.annotate\_text = "Hello world!"**

**sleep(5)**

**camera.capture('/home/pi/Desktop/text.jpg')**

**camera.stop\_preview()**

### Change the look of the added text

* Set the text size with the following code:

camera.annotate\_text\_size = 50

You can set the text size to anything between 6 to 160. The default size is 32.

It’s also possible to change the text colour.

* First of all, add  color to your import line at the top of the program:

from picamera import PiCamera, Color

* Then below the import line, amend the rest of your code so it looks like this:

**camera.start\_preview()**

**camera.annotate\_background = Color('blue')**

**camera.annotate\_foreground = Color('yellow')**

**camera.annotate\_text = " Hello world "**

**sleep(5)**

**camera.stop\_preview()**

**Q1: code:for taking picture when motion detected.**

#import the necessary packages

from gpiozero import Button, MotionSensor

from picamera import PiCamera

from time import sleep

from signal import pause

#create objects that refer to a button,

#a motion sensor and the PiCamera

button = Button(2)

pir = MotionSensor(4)

camera = PiCamera()

#start the camera

camera.rotation = 180

camera.start\_preview()

#image image names

i = 0

#stop the camera when the pushbutton is pressed

def stop\_camera():

camera.stop\_preview()

#exit the program

exit()

#take photo when motion is detected

def take\_photo():

global i

i = i + 1

camera.capture('/home/pi/Desktop/image\_%s.jpg' % i)

print('A photo has been taken')

sleep(10)

#assign a function that runs when the button is pressed

button.when\_pressed = stop\_camera

#assign a function that runs when motion is detected

pir.when\_motion = take\_photo

pause()

***Q2.Code for recording video when motion detected.***

#import the necessary packages

from gpiozero import Button, MotionSensor

from picamera import PiCamera

from time import sleep

from signal import pause

#create objects that refer to a button,

#a motion sensor and the PiCamera

button = Button(2)

pir = MotionSensor(4)

camera = PiCamera()

#start the camera

camera.rotation = 180

camera.start\_preview()

#image image names

i = 0

#stop the camera when the pushbutton is pressed

def stop\_camera():

camera.stop\_preview()

#exit the program

exit()

def take\_video():

camera.resolution = (640, 360)

camera.framerate = 24

camera.start\_recording('one\_day.h264')

camera.wait\_recording(24 \* 60 \* 60)

camera.stop\_recording()

#assign a function that runs when the button is pressed

button.when\_pressed = stop\_camera

#assign a function that runs when motion is detected

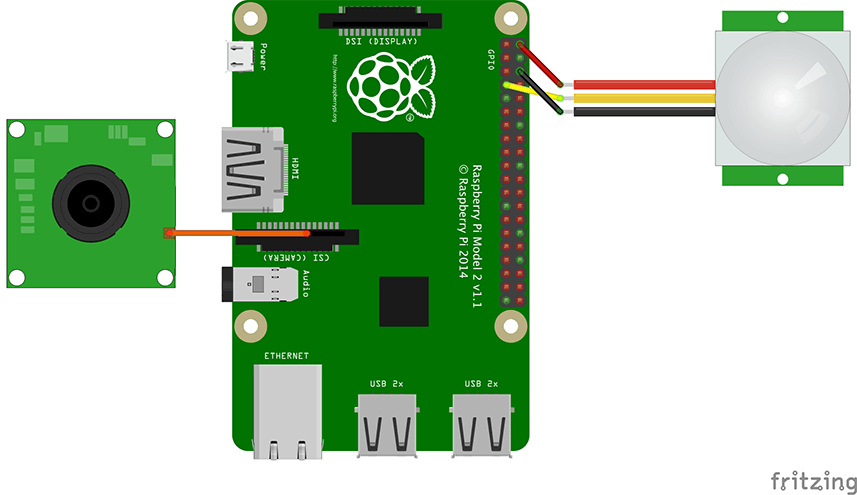
pir.when\_motion = take\_photo

button.when\_pressed = stop\_camera

#assign a function that runs when motion is detected

pir.when\_motion = take\_video()

pause()



import picamera # Importing the library for camera module

from time import sleep

camera = picamera.PiCamera() # Setting up the camera

camera.start\_preview()

sleep(5)

camera.capture('/home/pi/Desktop/picture/imag.jpg') # Capturing the image

camera.stop\_preview()

print('Done')

To record the video using the python code, type the below code in the python file. You will not see the preview window if you are controlling the raspberry pi through SSH connection.

import picamera # Importing the library for camera module

from time import sleep # Importing sleep from time library to add delay in program

camera = picamera.PiCamera() # Setting up the camera

camera.start\_preview() # You will see a preview window while recording

camera.start\_recording('/home/pi/Desktop/video.h264') # Video will be saved at desktop

sleep(5)

camera.stop\_recording()

camera.stop\_preview()