

Raspberry PI and Turtlebot Camera Guide

Raspberry PI PICAMERA - Command Line Control

To capture a picture:-

```
$ raspistill -o cam.jpg
```

<https://www.raspberrypi.org/documentation/usage/camera/raspicam/raspistill.md>

To capture a video:-

```
$ raspivid -o vid.h264
```

<https://www.raspberrypi.org/documentation/usage/camera/raspicam/raspivid.md>

To view pictures and videos on the Raspberry PI:-

Gpicview and vlc are both installed and available to use to view pictures and videos respectively

To export picture and videos:-

Scp and ftp over to your DiCE account and/or your home computer and view from there.

To stream a video from your Raspberry PI:-

```
$ raspivid -t 0 -l -o tcp://0.0.0.0:3333
```

Or

```
$ raspivid -o - -t 0 -n | cvlc -vvv stream:///dev/stdin --sout '#rtp{sdp=rtsp://:8554/}' :demux=h264
```

Note:- the second option requires vlc to be installed on your raspberry pi (the turtlebot Raspberry PI's do not have vlc installed as part of their original build)

To view your streamed video:-

VLC - media - open network stream

```
tcp/h264://192.168.0.77:3333
```

```
$ vlc tcp/h264://192.168.0.77:3333
```

Or

```
rtsp://192.168.0.77:8554/
```

Raspberry PI - PICAMERA - Display Camera

```
from picamera import PiCamera
```

```
from time import sleep
```

```
Camera = PiCamera()
```

```
camera.start_preview()
sleep(5)
camera.stop_preview()
```

Raspberry PI - PICAMERA - Capture Image

```
from picamera import PiCamera
from time import sleep
```

```
Camera = PiCamera()
```

```
camera.start_preview()
sleep(5)
camera.capture('/home/pi/Desktop/image.jpg')
camera.stop_preview()
```

Raspberry PI - PICAMERA - Record Video

```
from picamera import PiCamera
from time import sleep
```

```
Camera = PiCamera()
```

```
camera.start_preview()
camera.start_recording('/home/pi/Desktop/video.h264')
sleep(5)
camera.stop_recording()
camera.stop_preview()
```

To View the USB Webcam

```
$ ls /dev
```

On the standalone raspberry PI's the USB Webcam should appear as video1

On the turtlebots raspberry PI's the USB Webcam should appear as video0

```
$ vlc
```

```
Media - open_capture_device
```

Video device name = video0 or video1 (refer to note above)
Video standard = All

Raspberry PI USB Webcam - Capture Image

using pygame:

```
import pygame
import pygame.camera

pygame.camera.init()
pygame.camera.list_cameras() #Camera detected or not
cam = pygame.camera.Camera("/dev/video1", (640,480))
cam.start()
img = cam.get_image()
pygame.image.save(img, "filename.jpg")
```

Display Webcam Video

Note:- VNC needs to be enabled on Raspberry PI and code should be run from a VNC connection

With OpenCV

```
import cv2

cap = cv2.VideoCapture("/dev/video1")

# Check if the webcam is opened correctly
if not cap.isOpened():
    raise IOError("Cannot open webcam")

while True:
    ret, frame = cap.read()
    frame = cv2.resize(frame, None, fx=1, fy=1, interpolation=cv2.INTER_AREA)
    cv2.imshow('Input', frame)

    c = cv2.waitKey(1)
    if c == 27:
        break

cap.release()
cv2.destroyAllWindows()
```

With Pygame

```
import pygame
```

```

import pygame.camera
from pygame.locals import *

DEVICE = '/dev/video1'
SIZE = (640, 480)
FILENAME = 'capture.png'

def camstream():
    pygame.init()
    pygame.camera.init()
    display = pygame.display.set_mode(SIZE, 0)
    camera = pygame.camera.Camera(DEVICE, SIZE)
    camera.start()
    screen = pygame.surface.Surface(SIZE, 0, display)
    capture = True
    while capture:
        screen = camera.get_image(screen)
        display.blit(screen, (0,0))
        pygame.display.flip()
        for event in pygame.event.get():
            if event.type == QUIT:
                capture = False
            elif event.type == KEYDOWN and event.key == K_s:
                pygame.image.save(screen, FILENAME)
    camera.stop()
    pygame.quit()
    Return

if __name__ == '__main__':
    camstream()

```

Capture Webcam Stream

With OpenCV

```

import numpy as np
import cv2

cap = cv2.VideoCapture("/dev/video1")

# Define the codec and create VideoWriter object
fourcc = cv2.VideoWriter_fourcc(*'XVID')
out = cv2.VideoWriter('output.avi',fourcc, 20.0, (640,480))

while(cap.isOpened()):

```

```
ret, frame = cap.read()
if ret==True:
    frame = cv2.flip(frame,0)

    # write the flipped frame
    out.write(frame)

    cv2.imshow('frame',frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
    else:
        break

# Release everything if job is finished
cap.release()
out.release()
cv2.destroyAllWindows()
```