# Configurazioni testate

## Lettura singolo frame

#inizializzazione

camera = PiCamera()

rawCapture = PiRGBArray(camera)

#configurazione

camera.resolution = (webcamx, webcamy)

camera.framerate = 60

#lettura (alternativa 1)

rawCapture = PiRGBArray(camera)

camera.capture(rawCapture, format="bgr")

immagine = rawCapture.array

#lettura (alternativa 2)

camera.capture(rawCapture, format="bgr")

immagine = rawCapture.array

rawCapture.truncate(0)

### Telecamere aperte e chiuse ad ogni operazione

1. **Apro e chiudo ogni telecamera prima di configurare o prima di configurare e leggere l’immagine (close01.py)**

Esito: lettura corretta ma frame rate = 1 fps

apro CAM1

configuro CAM1

chiudo CAM1

apro CAM2

configuro CAM2

chiudo CAM2

apro CAM3

configuro CAM3

chiudo CAM3

loop:

apro CAM1

configuro CAM1

leggo CAM1

chiudo CAM1

apro CAM2

configuro CAM2

leggo CAM2

chiudo CAM2

apro CAM3

configuro CAM3

leggo CAM3

chiudo CAM3

### Telecamere lasciate aperte

1. **configuro le telecamere tenendole aperte, apro e chiudo ogni telecamera prima di configurarla e leggere l’immagine (open02.py)**

Esito:

* lettura corretta ma frame rate = 1 fps
* funziona sempre

apro CAM1

configuro CAM1

configuro CAM2

configuro CAM3

chiudo CAM3

loop:

apro CAM1

configuro CAM1

leggo CAM1

chiudo CAM1

apro CAM2

configuro CAM2

leggo CAM2

chiudo CAM2

apro CAM3

configuro CAM3

leggo CAM3

chiudo CAM3

1. **configuro le telecamere lasciandole aperte, apro e chiudo ogni tripletta di telecamere prima di configurare e leggere l’immagine (open03.py)**

Esito:

* lettura corretta ma frame rate = 1 fps
* funziona NON sempre, in caso di errore si ricomincia ad eseguire le procedure dalla 2

apro CAM1

configuro CAM1

configuro CAM2

configuro CAM3

chiudo CAM3

loop:

apro CAM1

configuro CAM1

leggo CAM1

configuro CAM2

leggo CAM2

configuro CAM3

leggo CAM3

chiudo CAM3

1. **Configuro le telecamere lasciandole aperte, configuro e leggo l’immagine (open04.py)**

Esito:

* lettura corretta ma frame rate = 1 fps
* funziona NON sempre, in caso di errore si ricomincia ad eseguire le procedure dalla 2

apro CAM1

configuro CAM1

configuro CAM2

configuro CAM3

loop:

configuro CAM1

leggo CAM1

configuro CAM2

leggo CAM2

configuro CAM3

leggo CAM3

1. **Configuro le telecamere lasciandole aperte, leggo l’immagine (open05.py)**

Esito:

* lettura corretta ma frame rate = 1 fps
* funziona NON sempre, in caso di errore si ricomincia ad eseguire le procedure dalla 2

apro CAM1

configuro CAM1

configuro CAM2

configuro CAM3

loop:

leggo CAM1

leggo CAM2

leggo CAM3

## Lettura flusso di immagini

1. **Configuro le telecamere lasciandole aperte, leggo flusso con capture\_continuous (open06.py)**

#inizializzazione

camera = PiCamera()

rawCapture = PiRGBArray(camera)

#configurazione

camera.resolution = (webcamx, webcamy)

camera.framerate = 60

#lettura flusso

for frame in camera.capture\_continuous(rawCapture, format="bgr", use\_video\_port=True):

immagine = frame.array

rawCapture.truncate(0)

Esito:

* Le immagini sono lette sempre da una sola telecamera, anche se cambio l’ordine di configurazione

1. **Configuro le telecamere lasciandole aperte, leggo flusso con capture\_continuous (open07.py)**

#inizializzazione

self.camera = PiCamera()

#configurazione

camera.resolution = (webcamx, webcamy)

camera.framerate = 60

rawCapture = PiRGBArray(camera)

stream = camera.capture\_continuous(rawCapture, format="bgr", use\_video\_port=True)

#lettura flusso

while True:

f = stream.next()

frame\_uncropped = f.array

rawCapture.truncate(0)

Esito:

* dopo due cicli di lettura (molto lenti) dalle tre telecamere, errore alla istruzione “f = stream.next()” - riga 299:

“*Incorrect buffer length for resolution %dx%d' % (width, height))*

*picamera.exc.PiCameraValueError: Incorrect buffer length for resolution 480x320*”

* non funziona