

Raspberry pi code:

```
import cv2

import numpy as np

import serial

import string

import RPi.GPIO as GPIO

import time

from skimage import measure

from matplotlib import pyplot as plt


GPIO.setmode(GPIO.BOARD)

GPIO.setup(32,GPIO.OUT)

GPIO.setup(36,GPIO.OUT)

GPIO.setup(38,GPIO.OUT)

GPIO.setup(40,GPIO.OUT)


port=serial.Serial(

    "/dev/ttyUSB0",

    baudrate=2400,

    parity=serial.PARITY_NONE,
```

```
stopbits=serial.STOPBITS_ONE,
```

```
bytesize=serial.EIGHTBITS,
```

```
writeTimeout = 0,
```

```
timeout = 10)
```

```
print(port.isOpen())
```

```
print("Port Opened...")
```

```
defdraw_hist(name, gray):
```

```
hist = cv2.calcHist([gray], [0], None, [256], [0,256])
```

```
globalc_val
```

```
c_val = np.mean([hist[np.nonzero(hist)]])
```

```
MAX = max(hist)
```

```
plot = np.zeros((256,512))
```

```
fori in range(255):
```

```
x1 = 4*i
```

```
x2 = 4*(i+1)
```

```
y1 = hist[i]*512/MAX
```

```
y2 = hist[i+1]*512/MAX
```

```
cv2.line(plot, (x1,y1), (x2,y2), 1, 3)
```

```
cv2.imshow(name + "-gray", gray)
```

```
cv2.imshow(name + "-hist", plot)
```

```
def main():
```

```
    cam = cv2.VideoCapture(0)
```

```
    while cv2.waitKey(10) == -1:
```

```
        ret, frame = cam.read()
```

```
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

```
            #draw_hist("cam", gray)
```

```
        draw_hist("cam", frame)
```

```
        print(c_val)
```

```
        edges = cv2.Canny(gray, 100, 200)
```

```
        cv2.imshow('Canny', edges)
```

```
        val_out = np.sum(edges)
```

```
            L = measure.label(edges)
```

```
        print "Number of components:", np.max(L)
```

```
            LL = np.max(L)
```

```
        if LL <= 10:
```

```
            print('Normal')
```

```
            port.write('[0]')
```

```
        elif LL > 10:
```

```
            print(val_out)
```

```
if ((val_out< 100000) & (c_val> 1800)):

    print ('Low')

    port.write('[1]')

    GPIO.output(36,True)

    time.sleep(1)

    GPIO.output(36,False)

        #time.sleep(1)

if ((val_out> 100000) & (val_out< 300000) & (c_val> 1800)): # ET

    print ('High')

    port.write('[2]')

    GPIO.output(38,True)

    time.sleep(1)

    GPIO.output(38,False)

        #time.sleep(1)

if ((val_out> 300000) & (c_val> 1800)):

    print ('Very High')

    port.write('[3]')

    GPIO.output(40,True)

    time.sleep(1)

    GPIO.output(40,False)

        #time.sleep(1)

cv2.waitKey(10)
```

```
if __name__=="__main__":
```

```
main()
```

PIC CODE FOR SPRAYING:

```
#include      <pic.h>
```

```
#include      <stdio.h>
```

```
#include      "delay.c"
```

```
unsigned char Rbuf[8];
```

```
shortintPtr=0,U1p=0;
```

```
shortintRrdy_Flag=0,Sync_Flag=0;
```

```
static void interrupt Int();
```

```
void main()
```

```
{
```

```
ADCON0 = 0x81;
```

```
ADCON1 = 0x80;
```

```
TRISA = 0x00;
```

```
TRISB = 0x00;
```

```
TRISE = 0x00;
```

```
TRISC = 0x80;
```

```
TRISD = 0x00;
```

```
PORTA = 0x00;
```

```
PORTB = 0x00;
```

```
PORTE = 0x00;
```

```
PORTC = 0x80;
```

```
PORTD = 0x00;
```

```
SPBRG = 103;
```

```
TXEN = 1;    // Enable transmit
```

```
BRGH = 1;    // ; Select high baud rate
```

```
SPEN = 1; // Enable Serial Port
```

```
CREN = 1; // Enable continuous reception
```

```
RCIF = 0; // Clear RCIF Interrupt Flag
```

```
RCIE = 1; // Set RCIE Interrupt Enable
```

```
SYNC = 0;
```

```
PEIE=1;
```

```
GIE=1;
```

```
for(;;)
```

```
{
```

```
    if(Rrdy_Flag == 1)
```

```
    {
```

```
        Rrdy_Flag = 0;
```

```
        if(Rbuf[0] == '1')
```

```
        {
```

```
            PORTC = 0x01;
```

```
DelayMs(1000);
```

```
DelayMs(1000);
```

```
DelayMs(1000);
```

```
PORTC = 0x00;
```

```
}
```

```
else if(Rbuf[0] == '2')
```

```
{
```

```
PORTC = 0x02;
```

```
DelayMs(1000);
```

```
DelayMs(1000);
```

```
DelayMs(1000);
```

```
DelayMs(1000);
```

```
DelayMs(1000);
```

```
DelayMs(1000);
```

```
PORTC = 0x00;
```

```
}
```

```
else if(Rbuf[0] == '3')
```

```
{
```

```
PORTC = 0x04;
```

```
DelayMs(1000);
```

```
DelayMs(1000);
```

```
DelayMs(1000);
```



```
    DelayMs(1000);
```

```
    DelayMs(1000);
```

```
    DelayMs(1000);
```

```
    DelayMs(1000);
```

```
    DelayMs(1000);
```

```
    DelayMs(1000);
```

```
    DelayMs(1000);
```

```
    DelayMs(1000);
```

```
    DelayMs(1000);
```

```
    PORTC = 0x00;
```

```
    }
```

```
    else if(Rbuf[0] == '0')
```

```
    {
```

```
        PORTC = 0x00;
```

```
    }
```

```
    }
```

```
    }
```

```
}
```

```
static void interrupt Int()
```

```
{
```

```
if(RCIF)

{

    RCIF = 0;

    if(Sync_Flag != 0)

    {

        if(RCREG == ']')

        {

            Rrdy_Flag = 1;

            Sync_Flag = 0;

        }

        else

        {

            if((RCREG >= 0x00) && (RCREG <= 0xFF))

                Rbuf[Ptr++] = RCREG;

        }

    }

    else

    {
```

```
if(RCREG == '[')
{
    Sync_Flag = 1;

    Ptr = 0;

}

}

}
```