PC-side color tracking

In this chapter, we will connect the Raspberry Pi to the PC, the Raspberry Pi as the TCP/IP socket server, the PC as the client, start the OpenCV control camera on the PC. And send data to the Raspberry Pi server by Socket.send() function.
Raspberry Pi will control servo by a simple count.

We need to input this command at the terminal：

ifconfig

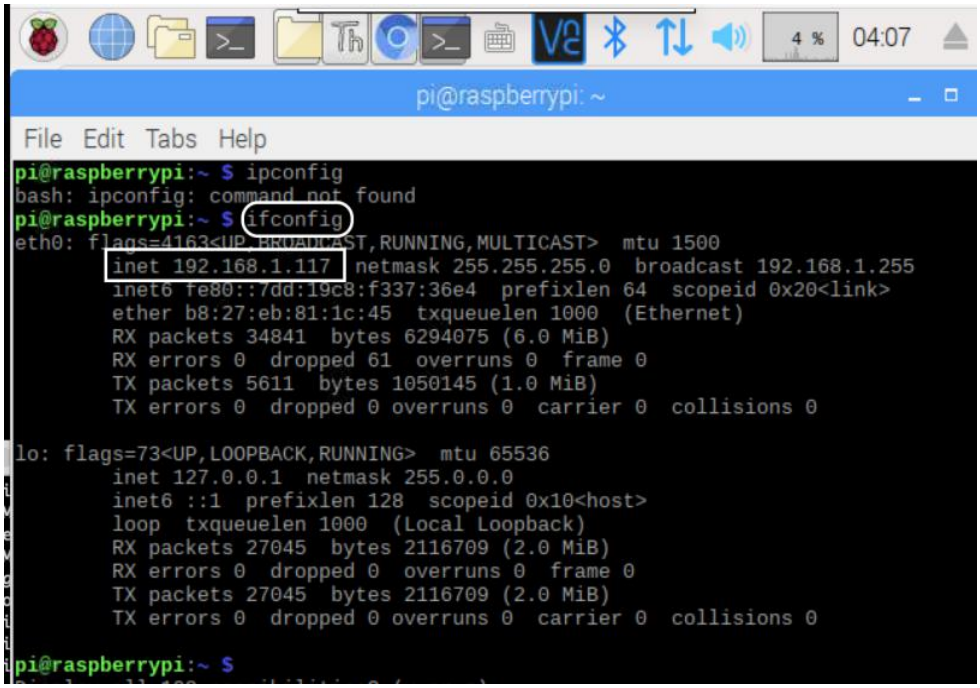

Figure 1-1

The Raspberry pi socket server source code of the program is located at:

/home/pi/Adafruit_PCA9685/socket_servo.py

The IP address in the program is the IP address of the host.

```
#!/usr/bin/env python2
# -*- coding: utf-8 -*-
"""
    Created on Tue Nov  6 01:18:45 2018
    * @par Copyright (C): 2010-2019, Shenzhen Yahboom Tech
    * @file        socket_servoMotor
    * @version      V1.0
    * @details
    * @par History

    @author: longfuSun
"""
from __future__ import division
import Adafruit_PCA9685
import time
import socket

address = ('192.168.1.66',7783)#Local IP address

#Complete the standard socket connection, binding, monitoring, to the raspberry
s = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
s.bind(address)
#Initialize servo
pwm = Adafruit_PCA9685.PCA9685()
pwm.set_pwm_freq(60)
pwm.set_pwm(1,0,400)
pwm.set_pwm(0,0,500)
time.sleep(1)
#Initialize PID
x=0
y=0
thisError_x=0
lastError_x=0
thisError_y=0
lastError_y=0
X_P=425
Y_P=425
flag=0
y=0

while 1:
```

```
    #Socket "receive" immediately after entering the steering gear
    data,addr=s.recvfrom(2048)
    if not data:
        break
    #print("got data from",addr)
    #Socket communication data needs to be decoded first
    x=data.decode()
    print(x)
    #The values sent are x and y, and in the actual "," is used as the flag.
    strX=str(x)
    arr=strX.split(',')
    #String type directly into int will report an error, so first convert to fl
    intX=int(float(arr[0]))
    intY=int(float(arr[1]))
    print('x:',intX,'y:',intY)

    ############
    thisError_x=intX-320
    thisError_y=intY-240
    pwm_x=thisError_x*6+1*(thisError_x-lastError_x)
    pwm_y=thisError_y*6+1*(thisError_y-lastError_y)

    lastError_x=thisError_x
    lastError_y=thisError_y
    XP=pwm_x/100
    YP=pwm_y/100
    X_P=X_P+int(XP)
    Y_P=Y_P+int(YP)
    if X_P>670:
        X_P=650
    if X_P<0:
        X_P=0
    if Y_P>670:
        Y_P=650
    if Y_P<50:
        Y_P=0

    ############
    print('***',X_P,Y_P)
    pwm.set_pwm(1,0,650-X_P)
    pwm.set_pwm(2,0,800-Y_P)


    time.sleep(0.02)
s.close()
```

The PC-side source code of the program is located at:

```
from __future__ import division
import cv2
import time
import numpy as np
import socket

addr = ('192.168.1.66',7783)#Target host IP
#readdr = ('192.168.1.110',7780)#Host IP
s = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)

cap = cv2.VideoCapture(0)
cap.set(3, 640)
cap.set(4, 480)
yellow_lower=np.array([9,135,231])
yellow_upper=np.array([31,255,255])

x=0;

while 1:
    ret,frame = cap.read()
    frame=cv2.GaussianBlur(frame,(5,5),0)
    hsv= cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)
    mask=cv2.inRange(hsv,yellow_lower,yellow_upper)
    mask=cv2.erode(mask,None,iterations=2)
    mask=cv2.dilate(mask,None,iterations=2)
    mask=cv2.GaussianBlur(mask,(3,3),0)
    res=cv2.bitwise_and(frame,frame,mask=mask)
    cnts=cv2.findContours(mask.copy(),cv2.RETR_EXTERNAL,
                          cv2.CHAIN_APPROX_SIMPLE)[-2]
    if len(cnts)>0:
        cnt=max(cnts,key=cv2.contourArea)
        (x,y),radius=cv2.minEnclosingCircle(cnt)
        cv2.circle(frame,(int(x),int(y)),int(radius),(255,0,255),2)
        #[Data, the (x, y) two values are combined and converted into a string type
        # the socket does not send a string string, but a binary code formed after utf-8 encoding.
        print('x',x);
        data =str(x)+','+str(y)
        s.sendto(data.encode("utf-8"),addr)
    cv2.imshow("capture", frame)
    if cv2.waitKey(1)==119:
        break
cap.release()
cv2.destroyAllWindows()
s.close()
```