Chapter19: QR code recognition

About QR code:

The content of this project is mainly for extracting and printing information of QR Code (Quick Response Code).

How to generate a QR code?
You can find ways to generate QR codes online.

Two open source software for QR code recognition, zbar and pyzbar.
In this chapter, we will provide two ways to identify the QR code:
1) pyzbar, which derived from zbar
2) Processing zbar.

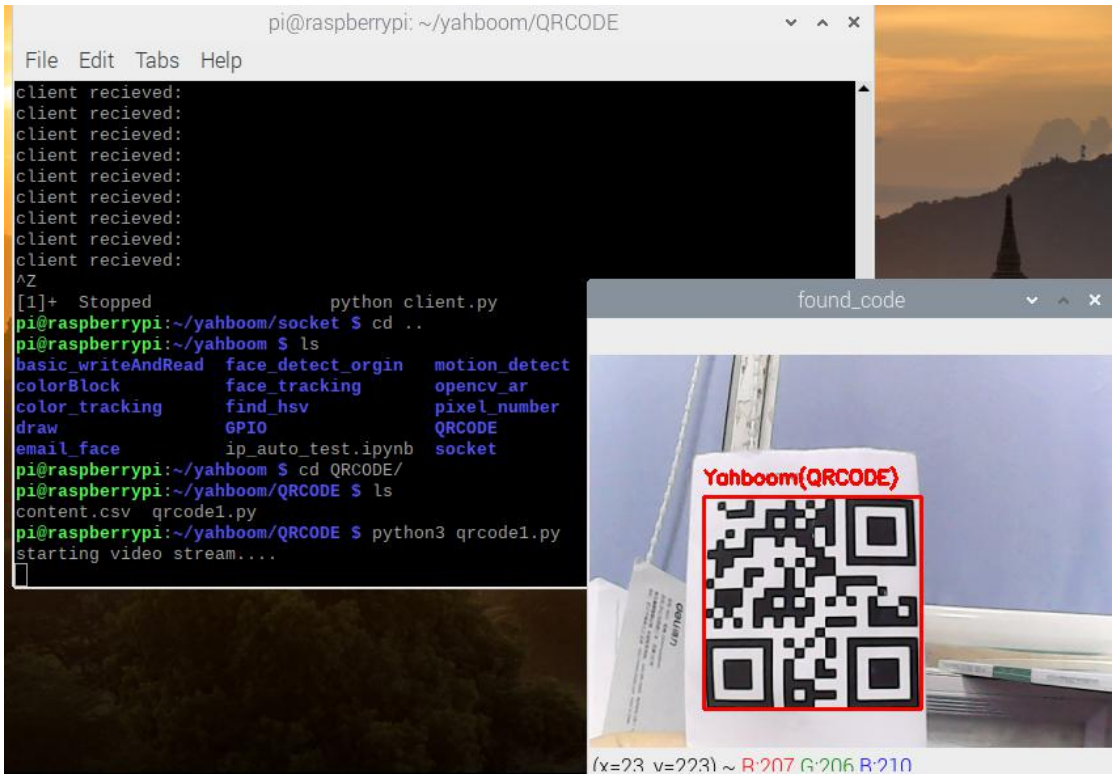
We need to input this command at the terminal:

```
sudo apt-get install lizbar0
sudo pip install pyzbar
```

After the download is complete, you can input import pyzbar in the spyder python to check if the installation is successful.
The source code of the program is located at:
/home/pi/yahboom/QRCODE/qrcode1.py
In this program, in addition to pyzbar, we will use imutils and argparse to simplify "camera use" and "parameter reference".
!Note: This program cannot be run using jupyter lab.
Please use the following command to run the program.

```
python3 qrcode1.py
```

operation result:



The code as shown in the figure below.

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
    * @par Copyright (C): 2010-2019, Shenzhen Yahboom Tech
    * @file        qr code
    * @version     V1.0
    * @details
    * @par History
    @author: longfuSun
"""

from imutils.video import VideoStream
from pyzbar import pyzbar
import argparse
import datetime
import imutils
import time
import cv2

ap=argparse.ArgumentParser()
#Provide a csv file, the QR code content can be displayed on the screen and asave it to a special file.
ap.add_argument("-o","--output",type=str,default="content.csv",
                help="path to output csv file containing barcode")
args=vars(ap.parse_args())

print('starting video stream....')
#Command of using web camera
vs=VideoStream(src=0).start()
#Command of using web Raspberry Pi's own camera
#vs=VideoStream(usePiCamera=True).start()
time.sleep(2.0)

#Write content to csv
csv=open(args["output"],"w")
found=set()
while True:
    frame=vs.read()
    frame=imutils.resize(frame,width=400)
    barcodes=pyzbar.decode(frame)
    for barcode in barcodes:
        (x,y,w,h)=barcode.rect

        cv2.rectangle(frame,(x,y),(x+w,y+h),(0,0,255),2)
        #Decode the contents of the QR code, enter the time, content and type
        barcodeData=barcode.data.decode("utf-8")
        barcodeType=barcode.type

        text="{}({})".format(barcodeData,barcodeType)
```

```python
        cv2.putText(frame,text,(x,y-10),cv2.FONT_HERSHEY_SIMPLEX,0.5,
                    (0,0,255),2)

        if barcodeData not in found:
            csv.write("{},{}\n".format(datetime.datetime.now(),barcodeData))
            csv.flush()
            found.add(barcodeData)
    cv2.imshow("found_code",frame)
    key=cv2.waitKey(1)&0xFF
    if key==ord("q"):
        break


csv.close()
cv2.destroyAllWindows()
vs.stop()
```

3)The combination of QR code and Raspberry Pi hardware

First, we need to prepare 5 QR codes, each corresponding to the encoding of a gpio command string.



1.red_yello_blue light up 2. right_left servo turn 0 3.right_left servo turn 180


4.turn down 5. turn up

No1 QR code : starts the RGB lights.
No2 QR code: control servo turn 0
No3 QR code: control servo turn 180
No4 QR code: control servo turn down
No5 QR code: control servo turn up

In addition, the control expansion board The buzzer on the alarm is given and the control is abandoned.
In addition, in order to ensure the safety of the hardware, we need to make certain restrictions on the rotation of the servo. When the servo is turned to the far right, the buzzer will alarm on the expansion board and the give up control.

When the camera detects the QR code, it immediately decodes the QR code and adjusts the corresponding execution function.
For example, the code for servo turn left is as follows:

```
def right():
    global servo_rightleft
    servo_rightleft-=50
    if servo_rightleft<=170 or servo_rightleft>=570: warning()
else :    pwm.set_pwm(0, 0, servo_rightleft)
```

We use two functions to control the rotation angle of the same servo, so use the global statement in the function.

The pulse value of each rotation is 50, about 20°, when the real-time pulse value is close to the critical value, call warning() function will start the buzzer.

There are two codes in this experiment. The first one is to decode and respond to the QR code. The second one will try to introduce Baidu speech. Each time, a QR code is recognized and an action is completed, it will broadcast action. The first source code of the program is located at:
/home/pi/Adafruit_Python_PCA9685/qrcode_motion.py

!Note: This program cannot be run using jupyter lab.
Please use the following command to run the program.
python3 qrcode_motion.py

```
pi@raspberrypi:~/Adafruit_Python_PCA9685 $ python3 qrcode_motion.py
qrcode_motion.py:26: RuntimeWarning: This channel is already in use, continuing
anyway.  Use GPIO.setwarnings(False) to disable warnings.
  GPIO.setup(10,GPIO.OUT)
qrcode_motion.py:27: RuntimeWarning: This channel is already in use, continuing
anyway.  Use GPIO.setwarnings(False) to disable warnings.
  GPIO.setup(9,GPIO.OUT)
qrcode_motion.py:28: RuntimeWarning: This channel is already in use, continuing
anyway.  Use GPIO.setwarnings(False) to disable warnings.
  GPIO.setup(11,GPIO.OUT)
starting video stream....
```

When the camera recognizes the above QR codes, we can see the corresponding experimental phenomenon.

Finally, we can press Ctrl+C to end the process and close the output.
The second source code of the program is located at:
/home/pi/Adafruit_Python_PCA9685/qrcode_speech.py

!Note: This program cannot be run using jupyter lab.
Please use the following command to run the program.
python3 qrcode_speech.py
The code as shown in the figure below.

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""

    * @par Copyright (C): 2010-2019, Shenzhen Yahboom Tech
    * @file       qrcode-speech
    * @version      V1.0
    * @details
    * @par History

    @author: longfuSun
"""
from __future__ import division
from imutils.video import VideoStream
from pyzbar import pyzbar
import argparse
import datetime
import imutils
import time
import cv2
import RPi.GPIO as GPIO
import Adafruit_PCA9685
from aip import AipSpeech
import pygame
import threading

#APP_ID='XXXX'
#API_KEY='XXX'
#SECRET_KEY='XXX'
APP_ID='20059657'
API_KEY='AOAFsdeeCwbQrEVDbSGjNjFE'
SECRET_KEY='8NzZdG1AZw8Q0G1mnqgAfh5RLbbTGzZv'
aipSpeech=AipSpeech(APP_ID,API_KEY,SECRET_KEY)

#Initialization gpio
GPIO.setmode(GPIO.BCM)
GPIO.setup(10,GPIO.OUT)
GPIO.setup(9,GPIO.OUT)
GPIO.setup(11,GPIO.OUT)
#Initialize the servo
servo_updown=500
servo_rightleft=390
pwm = Adafruit_PCA9685.PCA9685()
pwm.set_pwm_freq(60)
pwm.set_pwm(0, 0, servo_rightleft)
pwm.set_pwm(1, 0, servo_updown)
#When the steering gear exceeds the allowable angle, buzzer will alarm
def warning():
    GPIO.setup(16,GPIO.OUT)
```

```
    GPIO.output(16,True)
    time.sleep(1)
    GPIO.output(16,False)

def motion_speech(content):
    text=content
    result = aipSpeech.synthesis(text = text,
                            options={'spd':5,'vol':9,'per':0,})
    if not isinstance(result,dict):
        with open('audio.mp3','wb') as f:
            f.write(result)
    else:print(result)
    pygame.mixer.init()
    pygame.mixer.music.load('/home/pi/Adafruit_Python_PCA9685/audio.mp3')
    pygame.mixer.music.play()

def red_yellow_blue():
    content='红绿蓝小灯'
    motion_speech(content)
    for i in range(0,2):
        GPIO.output(9,True)
        time.sleep(0.1)
        GPIO.output(10,True)
        time.sleep(0.1)
        GPIO.output(11,True)
        time.sleep(0.1)
        GPIO.output(10,False)
        time.sleep(0.1)
        GPIO.output(9,False)
        time.sleep(0.1)
        GPIO.output(11,False)

def right():
    content='伺服电机右转'
    motion_speech(content)
    global servo_rightleft
    servo_rightleft-=50
    if servo_rightleft<=170 or servo_rightleft>=570:
        warning()
    else :
        pwm.set_pwm(1, 0, servo_rightleft)
def left():
    content='伺服电机左转'
    motion_speech(content)
    global servo_rightleft
    servo_rightleft+=50
    if servo_rightleft<=170 or servo_rightleft>=570:
        warning()
    else :
```

```
        pwm.set_pwm(1, 0, servo_rightleft)
def turn_down():
    content='伺服电机向下'
    motion_speech(content)
    global servo_updown
    servo_updown-=50
    if servo_updown<=170 or servo_updown>=570:
        warning()
    else:
        pwm.set_pwm(2,0,servo_updown)
def turn_up():
    content='伺服电机向上'
    motion_speech(content)
    global servo_updown
    servo_updown+=50
    if servo_updown<=170 or servo_updown>=570:
        warning()
    else:
        pwm.set_pwm(2,0,servo_updown)

ap=argparse.ArgumentParser()

ap.add_argument("-o","--output",type=str,default="content.csv",
                help="path to output csv file containing barcode")
args=vars(ap.parse_args())

print('starting video stream....')
#for web camera
vs=VideoStream(src=0).start()
#Raspberry Pi camera
#vs=VideoStream(usePiCamera=True).start()
time.sleep(2.0)

csv=open(args["output"],"w")
found=set()

lastData=''
sendDate=0
while True:
    frame=vs.read()
    frame=imutils.resize(frame,width=400)
    barcodes=pyzbar.decode(frame)
    for barcode in barcodes:
        (x,y,w,h)=barcode.rect

        cv2.rectangle(frame,(x,y),(x+w,y+h),(0,0,255),2)

        barcodeData=barcode.data.decode("utf-8")
        barcodeType=barcode.type
```

```
        text="{}({})".format(barcodeData,barcodeType)

        cv2.putText(frame,text,(x,y-10),cv2.FONT_HERSHEY_SIMPLEX,0.5,
                    (0,0,255),2)

        newData=barcodeData
        currentDate=time.time()

        if (currentDate-sendDate>4):
            print('1')
            if newData=='red_yello_blue light up':
                tid=threading.Thread(target=red_yellow_blue)
                tid.setDaemon(True)
                tid.start()
                sendDate=time.time()
            elif newData=='right_left servo turn 0':
                tid=threading.Thread(target=left)          #Less than 500
                tid.setDaemon(True)
                tid.start()
                sendDate=time.time()
            elif newData=='right_left servo turn 180':
                tid=threading.Thread(target=right)
                tid.setDaemon(True)
                tid.start()
                sendDate=time.time()
            elif newData=='turn down':
                tid=threading.Thread(target=turn_down)
                tid.setDaemon(True)
                tid.start()
                sendDate=time.time()
            elif newData=='turn up':
                tid=threading.Thread(target=turn_up)
                tid.setDaemon(True)
                tid.start()
                sendDate=time.time()
            else : print('incorrect data:',newData)
        else:
            continue

        if barcodeData not in found:
            csv.write("{},{}\n".format(datetime.datetime.now(),barcodeData))
            csv.flush()
            found.add(barcodeData)
```

```
    cv2.imshow("found_code",frame)
    key=cv2.waitKey(1)&0xFF
    if key==ord("q"):
        break
csv.close()
cv2.destroyAllWindows()
vs.stop()
```