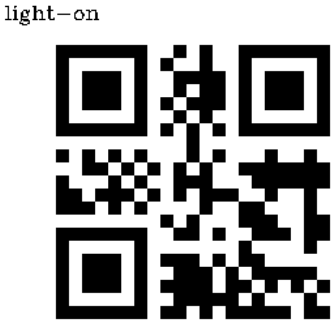
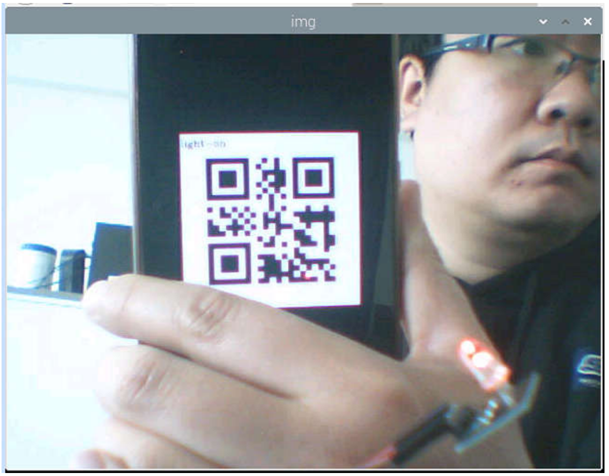
智能系统与控制



树莓派：
二维码的生成与扫描



于泓

鲁东大学

信息与电气工程学院

2022.4.27

实验内容

- 1 生二维码
- 2 利用二维码控制树莓派实现

(1) 开灯

(2) 关灯

(3) 开蜂鸣器

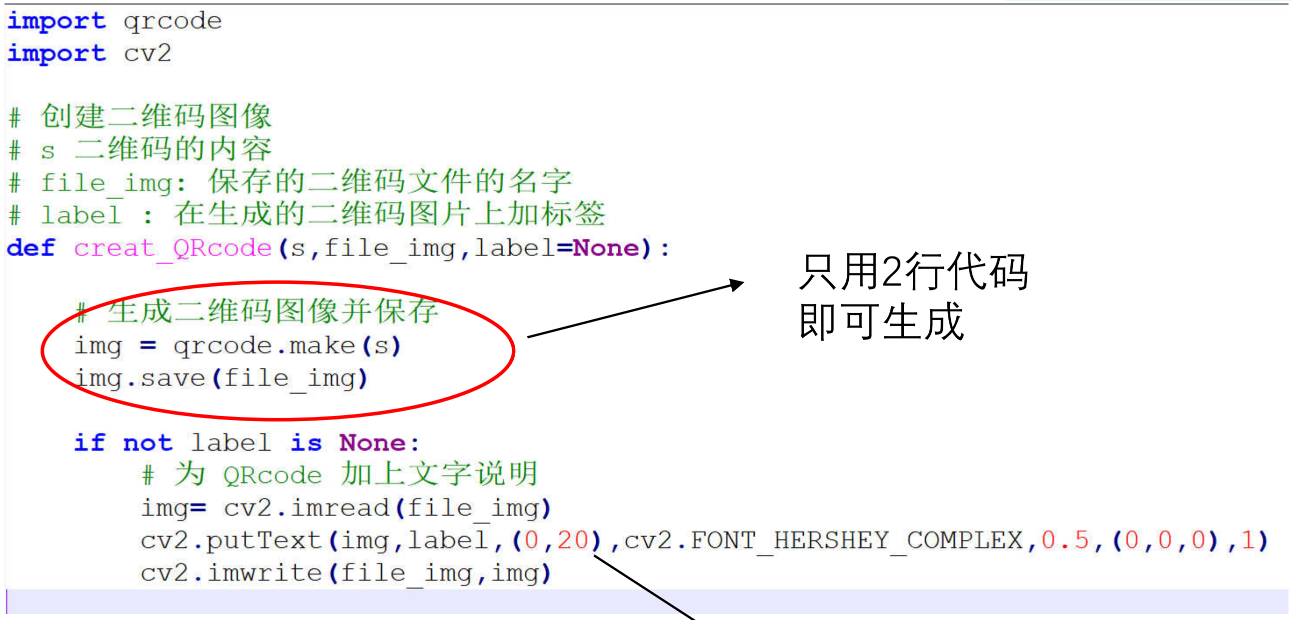
(4) 关蜂鸣器

(5) 测温度

1 二维码的生成

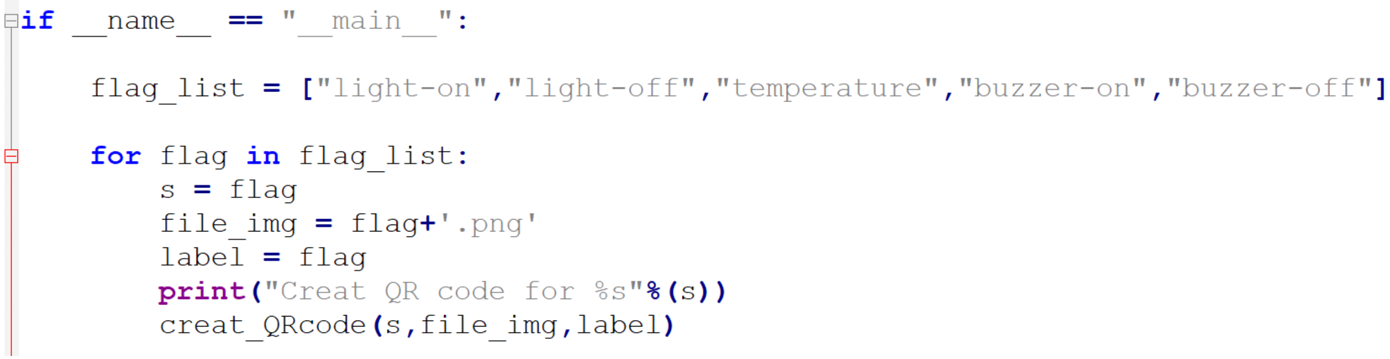
- 安装 qrcode

pip3 install qrcode



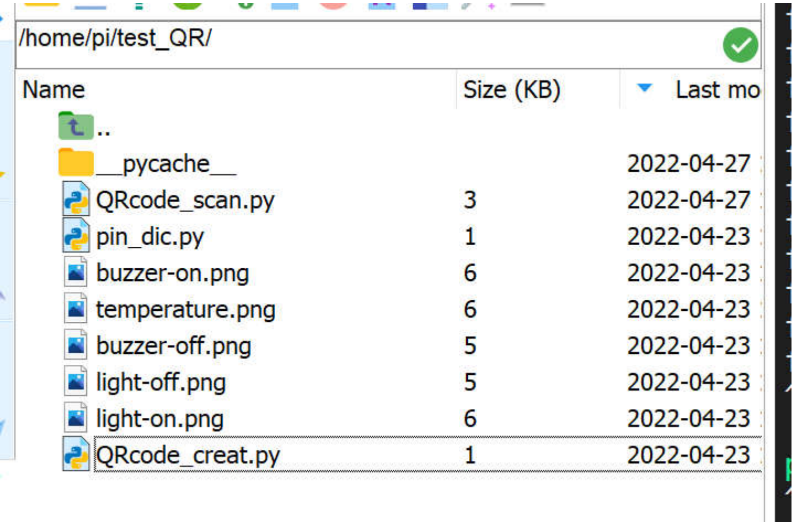
```
import qrcode
import cv2

# 创建二维码图像
# s 二维码的内容
# file_img: 保存的二维码文件的名字
# label : 在生成的二维码图片上加标签
def creat_QRcode(s,file_img,label=None):

    # 生成二维码图像并保存
    img = qrcode.make(s)
    img.save(file_img)

    if not label is None:
        # 为 QRcode 加上文字说明
        img= cv2.imread(file_img)
        cv2.putText(img,label,(0,20),cv2.FONT_HERSHEY_COMPLEX,0.5,(0,0,0),1)
        cv2.imwrite(file_img,img)
```

只用2行代码
即可生成



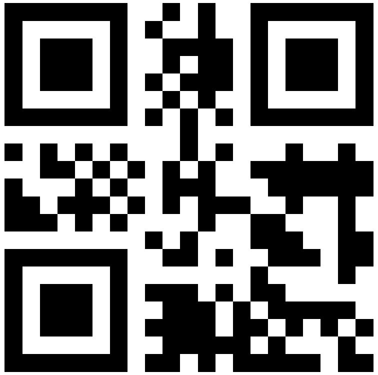
```
if __name__ == "__main__":

    flag_list = ["light-on","light-off","temperature","buzzer-on","buzzer-off"]

    for flag in flag_list:
        s = flag
        file_img = flag+'.png'
        label = flag
        print("Creat QR code for %s"%(s))
        creat_QRcode(s,file_img,label)
```

在生成的二维码
上加上标记

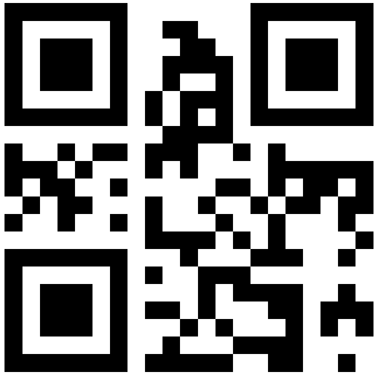


/home/pi/test_QR/

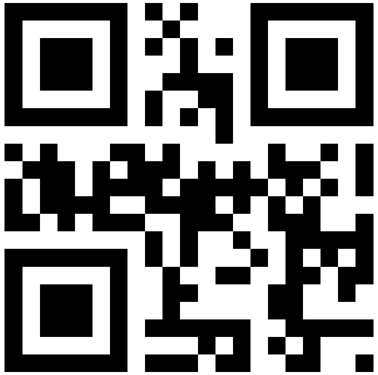
Name	Size (KB)	Last mo
..		
__pycache__		2022-04-27
QRcode_scan.py	3	2022-04-27
pin_dic.py	1	2022-04-23
buzzer-on.png	6	2022-04-23
temperature.png	6	2022-04-23
buzzer-off.png	5	2022-04-23
light-off.png	5	2022-04-23
light-on.png	6	2022-04-23
QRcode_creat.py	1	2022-04-23



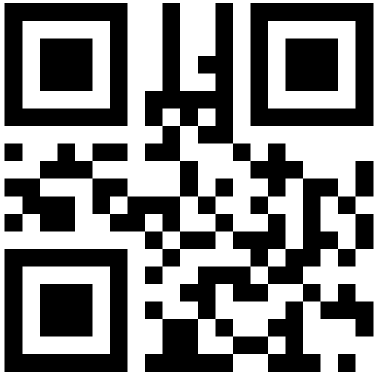
light-on



light-off



temperature

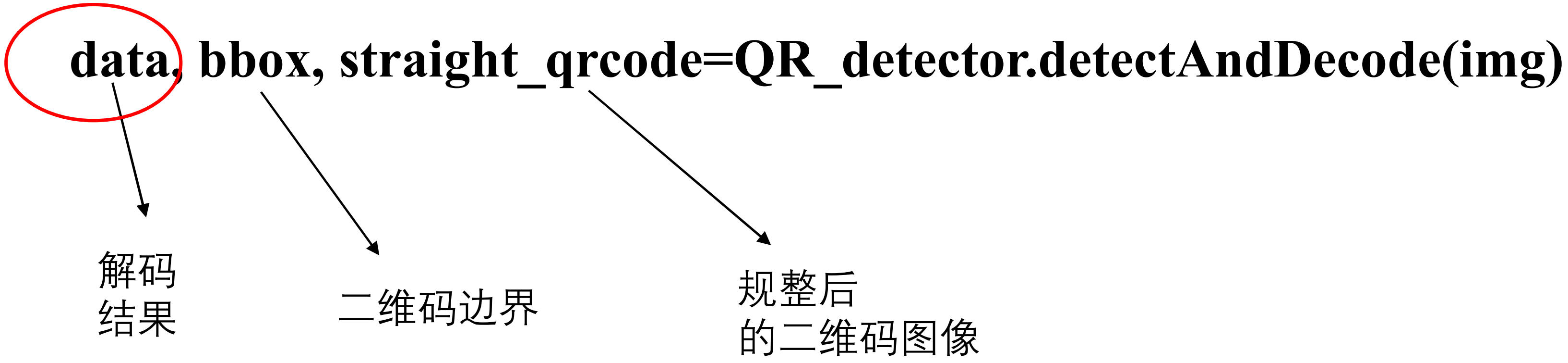


buzzer-on

buzzer-off

二维码扫描

```
QR_detector = cv2.QRCodeDetector()
```

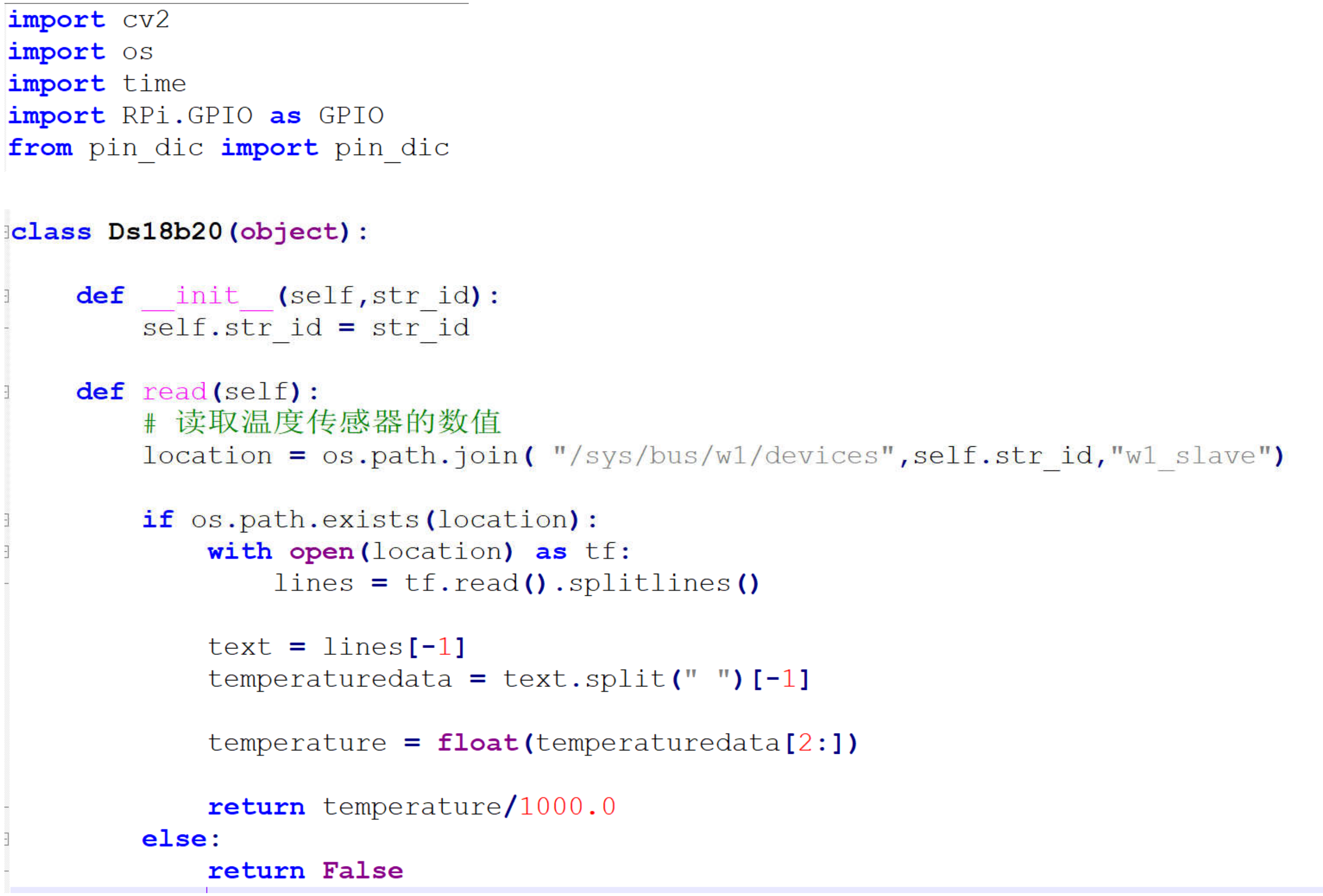


```
data, bbox, straight_qrcode=QR_detector.detectAndDecode(img)
```

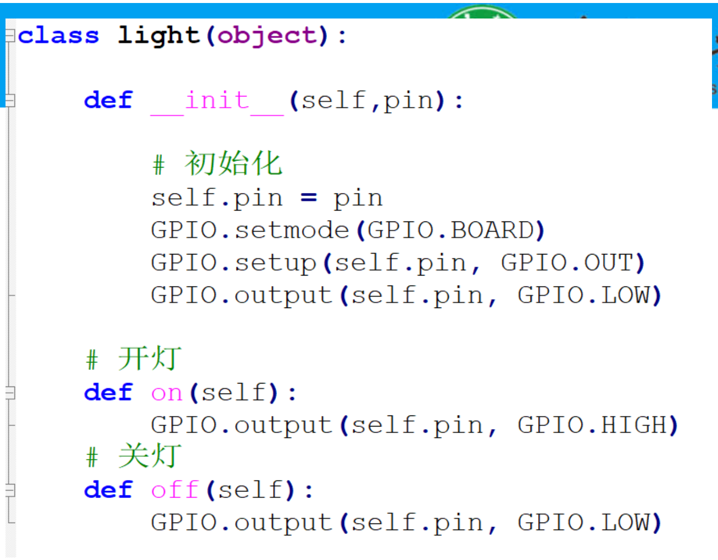
解码
结果

二维码边界

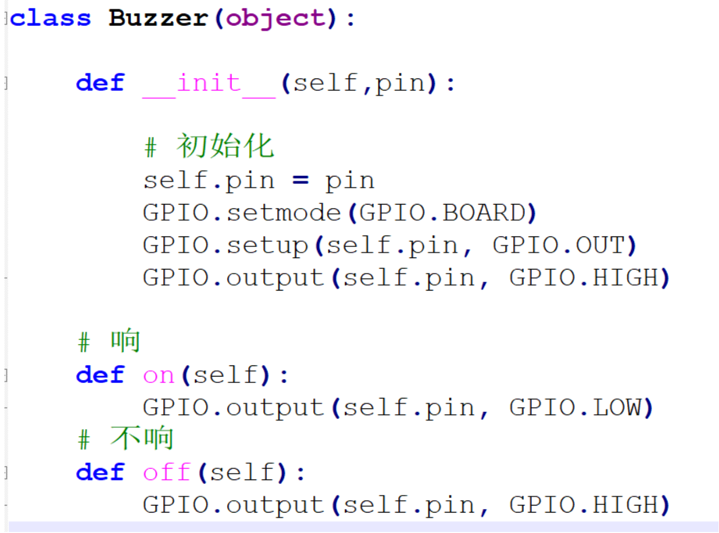
规整后
的二维码图像



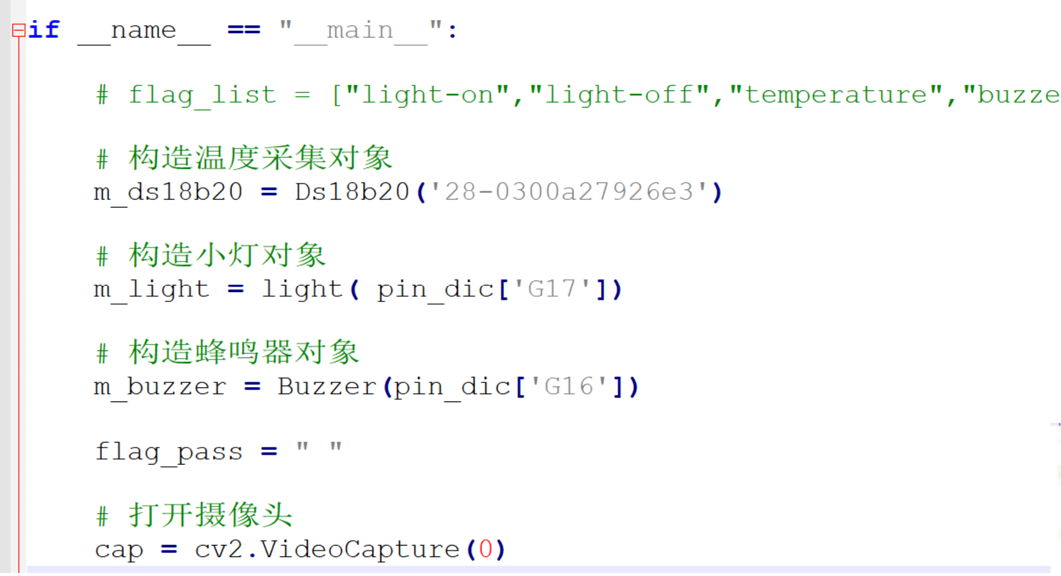
```
import cv2
import os
import time
import RPi.GPIO as GPIO
from pin_dic import pin_dic


class Ds18b20(object):

    def __init__(self,str_id):
        self.str_id = str_id

    def read(self):
        # 读取温度传感器的数值
        location = os.path.join( "/sys/bus/w1/devices",self.str_id,"w1_slave")

        if os.path.exists(location):
            with open(location) as tf:
                lines = tf.read().splitlines()

            text = lines[-1]
            temperaturedata = text.split(" ")[-1]

            temperature = float(temperaturedata[2:])

            return temperature/1000.0
        else:
            return False
```



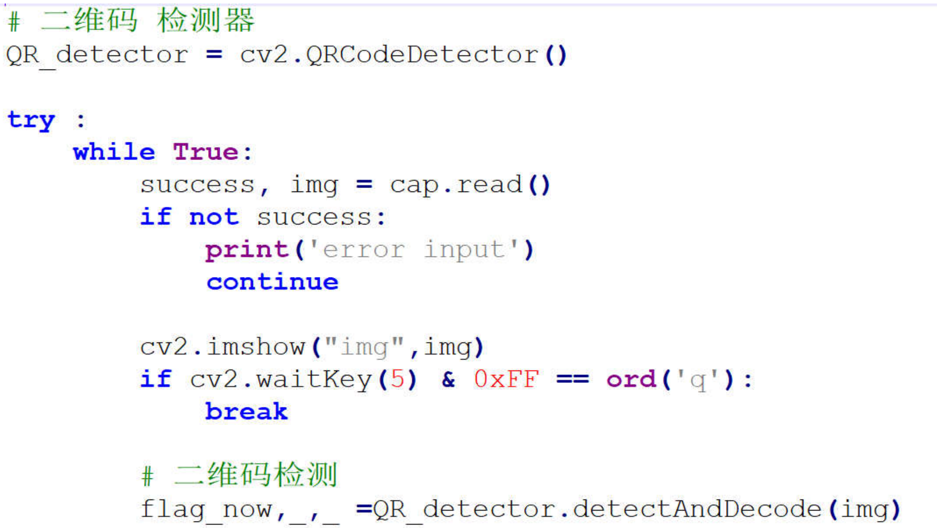
```
class light(object):

    def __init__(self,pin):

        # 初始化
        self.pin = pin
        GPIO.setmode(GPIO.BOARD)
        GPIO.setup(self.pin, GPIO.OUT)
        GPIO.output(self.pin, GPIO.LOW)

    # 开灯
    def on(self):
        GPIO.output(self.pin, GPIO.HIGH)
    # 关灯
    def off(self):
        GPIO.output(self.pin, GPIO.LOW)
```



```
class Buzzer(object):

    def __init__(self,pin):

        # 初始化
        self.pin = pin
        GPIO.setmode(GPIO.BOARD)
        GPIO.setup(self.pin, GPIO.OUT)
        GPIO.output(self.pin, GPIO.HIGH)

    # 响
    def on(self):
        GPIO.output(self.pin, GPIO.LOW)
    # 不响
    def off(self):
        GPIO.output(self.pin, GPIO.HIGH)
```



```
if __name__ == "__main__":

    # flag_list = ["light-on","light-off","temperature","buzzer-on","buzzer-off"]

    # 构造温度采集对象
    m_ds18b20 = Ds18b20('28-0300a27926e3')

    # 构造小灯对象
    m_light = light( pin_dic['G17'])

    # 构造蜂鸣器对象
    m_buzzer = Buzzer(pin_dic['G16'])

    flag_pass = " "

    # 打开摄像头
    cap = cv2.VideoCapture(0)
```



```
# 二维码 检测器
QR_detector = cv2.QRCodeDetector()

try :
    while True:
        success, img = cap.read()
        if not success:
            print('error input')
            continue

        cv2.imshow("img",img)
        if cv2.waitKey(5) & 0xFF == ord('q'):
            break

        # 二维码检测
        flag_now,_,_ =QR_detector.detectAndDecode(img)
        if flag_now:

            if flag_now == flag_pass:
                continue
            flag_pass = flag_now

            if flag_pass == "light-on":
                m_light.on()
                print("light on")

            elif flag_pass == "light-off":
                m_light.off()
                print("ligh off")

            elif flag_pass == "buzzer-on":
                m_buzzer.on()
                print("buzzer on")

            elif flag_pass == "buzzer-off":
                m_buzzer.off()
                print("buzzer off")

            elif flag_pass == "temperature":
                t = m_ds18b20.read()
                print(t)

except KeyboardInterrupt:
    print('\n Ctrl + C QUIT')

finally:

    GPIO.cleanup()

cap.release()
```

有检测结果

避免重复执行