

居家养老服务--基于树莓派STM32的人脸识别门禁

1903070311

许连鹏

2021 12 26

**成 绩 评 定 表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 学 院 | | 信息科学与工程 | 专 业 | 电子信息科学与技术 |
| 学生姓名 | | 许连鹏 | 班级学号 | 1903070311 |
| 实训题目 | | 居家养老服务--基于树莓派STM32的人脸识别门禁 | | |
| 评  语 | **【日常表现】**  1.学生学习态度： □认真; □比较认真; □一般; □不够认真  2.学生平时出勤： □全勤; □偶尔缺勤; □经常缺勤; □不来上课  **【题目属性】**  3.课题创意： □较好; □一般; □无  4.课题难度： □难; □有难度; □适中; □简单  5.技术先进性： □高; □较高; □一般；□不先进  6.课题工作量： □大; □较大; □一般；□小  **【完成情况】**  范增辉，刘诗扬  7.是否团队完成：□否; □是，队友：  8.如有团队分工是否明确：□是; □不够明确，有交集; □否;  9.对课题熟悉程度：□熟悉; □较熟悉; □部分熟悉; □不熟悉  10.独立完成程度： □大; □较大; □一般；□小  11.任务书完成度： □全部; □主要功能; □部分功能; □很少  12.作品运行效果：□非常好; □较好; □一般; □不好; □无作品;  **【报告情况】**  13.报告格式规范： □好; □较好; □一般; □不好; □有错误  14.报告内容情况： □好; □较好; □一般; □不好; □有错误  **【答辩情况】**  15.实训答辩情况：□好; □较好; □一般; □不好  16.实训整体感觉：□好; □较好 □一般 □勉强 □差  **其他说明：**  验收/评阅教师签字： | | | |
| 成绩 |  | | | |
| 日期 | 2021年 12月 27 日 | | | |

**实训任务书**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学 院 | 信息科学与工程 | | 专 业 | | 电子信息科学与技术 |
| 学生姓名 | 许连鹏 | | 班级学号 | | 1903070311 |
| 实训题目 | 居家养老服务--基于树莓派STM32的人脸识别门禁 | | | | |
| **实践教学要求与任务（如果是团队完成，需要写出具体分工）:**  基于树莓派和STM32最小系统，围绕居家养老服务主题，设计一个远程人脸识别门禁系统。设计有如下功能：   1. 实现人脸的检测和识别。 2. 实现手动输入密码的开门。 3. 实现树莓派端及门控端的手动关门功能。 4. 树莓派端有较为友好的UI界面。 5. 门控端有屏幕和蜂鸣器进行简单的人机交互。 6. 门的打开关闭由舵机模拟实现。 7. 开关门的信息可发送至其他平台进行记录。   **工作计划与进度安排:**  第3周: 实训动员，布置任务，选择题目， 查找资料  第4周: 确定题目，确定技术路线，开题  第5~14周: 详细设计与实现  第15周：调试、测试  第16周: 验收答辩 | | | | | |
| 指导教师：  2021 年9月 7 日 | | 专业负责人：  2021年 9月8 日 | | 学院教学副院长：  2021 年9月 10 日 | |

**摘 要**

本设计采用树莓派作为上位机，STM32单片机作为下位机，通过物联网协议MQTT进行互联通信，把人脸识别作为系统整体框架的主框架，设计完成了以树莓派为远端主控模块、外接摄像头设备，STM32为门控端、外接摄像头设备主控模块的远程人脸门禁功能。本设计以人脸识别为主，搭配门控端硬件交互，为远程人脸门禁的设计实现提供了一个切实可行的方案。本论文介绍了远程人脸门禁总体之设计，既囊括了硬件下位机系统设计，又囊括了上位机软件系统设计。通过上位机软件系统的UI界面选择人脸开门、手动关门和退出等功能模块。通过下位机硬件系统可实现手动输入密码开锁、手动关锁以及屏幕交互等功能。

关键词： 人脸识别；树莓派；STM32；MQTT

目 录

[1 引言 1](#_Toc90593604)

[2 需求设计 2](#_Toc90593605)

[3 系统（总体）设计 3](#_Toc90593606)

[4 详细设计 4](#_Toc90593607)

[4.1 硬件设计 4](#_Toc90593608)

[4.1.1树莓派主控模块 5](#_Toc90593609)

[4.1.2摄像头模块 6](#_Toc90593610)

[4.1.3 7寸显示器模块 7](#_Toc90593611)

[4.1.4 STM32F103C6T6最小系统 8](#_Toc90593612)

[4.1.5 ESP01-S模块 9](#_Toc90593613)

[4.1.6 有源蜂鸣器模块 10](#_Toc90593614)

[4.1.7 0.96寸OLED模块 10](#_Toc90593615)

[4.2 软件设计 12](#_Toc90593616)

[4.2.1 树莓派端软件设计 12](#_Toc90593617)

[4.2.2 门控端软件设计 13](#_Toc90593618)

[5 实验结果分析 19](#_Toc90593619)

[6 结论 22](#_Toc90593620)

[参考文献 23](#_Toc90593621)

1 引言

目前，我国已逐步进入老龄化社会。据2021年第7次全国人口普查数据，我国60岁以上的老人占总人口的18.7%，与2010年相比，60岁以上人口的比重上升6.79个百分点。我国人口老龄化发展速度，空巢老人也越来越多，老年人的人身安危亟待关注。随着人的年龄增长，身体机能也逐渐衰退，这点在老人身上尤为显著。记性不好、判断力下降是常见的症状。最早出现的是近记忆障碍，记不起新近发生的事性，经常会遗失东西，忘记东西置放的地方，忘记刚刚与别人谈话的内容;学习新事物的能力大大降低。到后期，远期记忆也会逐渐减退，即回忆不起从前的经历，思维和判断力障碍方面开始表现，不能掌握技术上或学识的要点，其后对原有认识也模糊不清。因此空巢老人每天离家回家这个环节就很容易出现问题。其一，记忆力的衰退容易使得老人忘记携带钥匙将自己锁在门外或是外出钥匙丢失无法进入家门。其二，老人思维和判断力迟缓，在开关家门的时候容易被不法分子盯上，导致财产或人身安全受损。如今社会公众安全意识的逐步增强，老人的安全防范已经得到人们的广泛关注。而门禁系统作为整个安全防范的一部分，已经成为安防领域关注的焦点之一。而且伴随计算机技术的快速发展，人脸识别技术得到大多数研发机构的青睐。 人脸是具有唯一性的生物特征，而人脸识别技术是通过采集人脸图像，提取图像中的人脸特征从而实现身份识别。由于操作简单、结果直观等特点，人脸识别技术广泛应用于信息安全等领域。

本设计并实现的远程人脸识别门禁系统主要针对家庭这种人数较少的应用场合。考虑到系统便携易用，功耗成本等因素，本文采用性能良好的嵌入式硬件平台进行系统的搭建。 本文硬件设备采用树莓派4B和STM32F103C6T6单片机，软件平台选择Linux操作系统， 通过对人脸图像的采集与处理，从而实现门禁系统的智能控制。

2 需求设计

在本设计中，需要两个客户端，为人脸识别端和门控端。用户可以进入到人脸识别端的UI界面，通过界面选择使用不同的功能，例如，人脸识别和关门等。人脸识别端发送控制指令给门控端，门控端收到指令后执行关门相关的机械控制。当人脸识别端网络较差无法正常运行人脸检测时，可以通过门控端密码锁进行传统方式的开门。

图示

描述已自动生成

图1 UML用例图

3 系统（总体）设计

本设计中，上位机采用树莓派，下位机采用STM32F103C6T6单片机，通过物联网协议MQTT进行通信，实现远程人脸门禁功能。MQTT服务器使用开源EMQ X 开源版，搭建在阿里云轻量应用服务器上，该MQTT服务器具有可视化管理的H5界面，方便易用。人脸识别功能由OpenCV及百度智能云人脸识别API实现。树莓派上运行带UI界面的人脸识别程序，UI界面使用框架PyQt5设计及实现。人脸识别程序开始运行后，由OpenCV对人脸进行检测及标识，通过百度智能云人脸识别API实现人脸的区分和识别。程序得到人脸信息后，将通过MQTT发送至下位机。STM32F103C6T6作为传统单片机并不支持网络连接，因此借助ESP01-S模块，对ESP01-S模块烧录MQTT及串口通信代码文件后，与STM32F103C6T6单片机串口相连接，便可实现传统单片机的网络连接，以及STM32F103C6T6下位机与树莓派上位机的互联通信。此外，门控端还设有4\*4矩阵键盘作为手动密码输入的工具，有源蜂鸣器的目的是提供一个按键及开关门的反馈提醒，还设有一个OLED屏幕进行简单的人机交互，门的打开关闭由舵机进行机械控制，舵机转动所需要的PWM信号由STM32F103C6T6单片机产生。

图示

描述已自动生成

图2 系统总体框图

4 详细设计

4.1 硬件设计

本设计中，下位机以STM32F103C6T6作为核心，辅以必要的电路，设计了一个简易的远程人脸门禁系统。下位机由5V直流电源供电，STM32F103C6T6单片机通过ESP01-S模块连接至阿里云上搭建的MQTT消息服务器。门控端同时设有4\*4矩阵键盘和蜂鸣器，可通过手动按键向STM32F103C6T6单片机输入密码信息，蜂鸣器可以对按键的操作进行响应和反馈。门控端连接一个OLED屏幕，在屏幕上显示网络连接状态，门的开关状态和密码输入状态。STM32F103C6T6单片机可以输出PWM信号对舵机进行控制。

图示, 示意图

描述已自动生成

图3 门控端总体原理图

本设计中，上位机以树莓派作为核心，树莓派为远端主控模块、外接摄像头设备，及7寸屏幕。

图示, 示意图

描述已自动生成

图4 树莓派原理图

4.1.1树莓派主控模块

树莓派开发者为以 Eben·Upton 为本项目领头人的“Raspberry Pi 基金会”开发制作而成。而后正式发售者为 Eben Epton。树莓派被称为世界上最小的台式电脑，又称卡片式电脑。其为一款为 ARM 之微型电脑主板，以 SD / MicroSD 卡为内硬盘，主板周有四方 USB、以太网口，可按照设计需要连接输入设备、输出设备以及网线，与此同时， 还带有可输出高清视频的 HDMI 接口，配有 PC 之基本功能。参数列表如表所示：

表格

中度可信度描述已自动生成

图5 参数列表图

树莓派 4B 的板载接口如图所示。

图示, 工程绘图, 示意图

描述已自动生成

图 6 板载接口图

图示, 示意图

描述已自动生成

图 7 主板尺寸图

4.1.2摄像头模块

Raspberry Pi NoIP Camera V2 红外高清摄像头，是树莓派专用的高清摄像头。支持1080P30、720P60、640\*480/90 摄像。因主板上小插槽接树莓使，而用专之 CSI 方。镜头无红外滤波器，可以红外线摄像和低光环境拍照。

Cable Extension Kit 树莓派摄像头线缆延长套件：把摄像头的 MIPI 信号转成 CML 信号后，把控制摄像头的 IIC 接口信号耦合到一对差分线进行传输，通过普通的 CAT5 非屏蔽线最长可传输 20 米，最低温度可到-20 摄氏度。无需额外驱动，把树莓派主板上的 CSI 接口通过 15 针的排线接到 RX 板上，通过 CAT5 网线连接发送模块，把摄像头通过 15 针的 FPC 排线接到发送板上，用树莓派标准的摄像头命令拍照、录制视频即可。参数列表如图所示：

表格

低可信度描述已自动生成

图 8 参数列表图

摄像头电路板与树莓派通过一条 15 芯的排线进行连接。具体连接操作如下：先将树莓派上连接座两端卡扣拉起，再把排线插入座中，并竖直，然后按下两端的卡扣。摄像头电路板连接同上，如图9所示。

电子邮件页面的手机截图

中度可信度描述已自动生成

图 9 摄像头实物图

4.1.3 7寸显示器模块

基本参数如图所示。

表格

描述已自动生成

图 10 参数列表图

HDMI 信号引脚如图 3.4 所示。

图示

描述已自动生成

图 11 信号引脚图

显示器模块如图12所示。

桌子上放着电脑

描述已自动生成

图 12 显示器模块图

4.1.4 STM32F103C6T6最小系统

STM32F103系列属于中低端的32位ARM微控制器，该系列芯片是意法半导体（ST）公司出品，其内核是Cortex-M3。该系列芯片按片内Flash的大小可分为三大类：小容量（16K和32K）、中容量（64K和128K）、大容量（256K、384K和512K）。芯片集成定时器Timer，CAN，ADC，SPI，I2C，USB，UART等多种外设功能。

硬件资源：  
1、STM32F103C6T6主芯片一片  
2、贴片8M晶振（通过芯片内部PLL最高达72M）ST官方标准参数  
3、3.3V稳压芯片，最大提供800mA电流  
4、一路miniUSB接口，可以给系统版供电，预留USB通讯功能  
5、复位按键  
6、标准SWD口一个，支持JLink，STLink，JLINK OB  
7、BOOT选择端口  
8、O扩展排针 20pin x 2  
9、电源指示灯1个  
10、功能指示灯一个，用于验证IO口基本功能  
11、预留串口接口，方便和5V开发板连接，用串口即可程序  
12、高性能爱普生32768Hz晶振，价格是直插晶振的10倍价格，易起振  
13、10K RAM，32K ROM ，TQFP48封装

图示, 示意图

描述已自动生成

图 13 STM32F103最小系统原理图

STM32F103最小系统实物图如下：

图片包含 游戏机, 电路, 测量

描述已自动生成

图 14 STM32F103最小系统实物图

4.1.5 ESP01-S模块

ESP8266 系列模组是深圳市安信可科技有限公司开发的一系列基于乐鑫ESP8266的低功耗UART-WiFi芯片模组，可以方便地进行二次开发，接入云端服务，实现手机3/4G全球随时随地的控制，加速产品原型设计。

模块核心处理器 ESP8266 在较小尺寸封装中集成了业界领先的 Tensilica L106 超低功耗 32 位微型 MCU，带有 16 位精简模式，主频支持 80 MHz 和 160 MHz，支持 RTOS，集成 Wi-Fi MAC/ BB/RF/PA/LNA，板载天线。支持标准的 IEEE802.11 b/g/n 协议，完整的 TCP/IP 协议栈。用户可以使用该模块为现有的设备添加联网功能，也可以构建独立的网络控制器。

ESP8266是高性能无线 SoC，以最低成本提供最大实用性，为 Wi-Fi 功能嵌入其他系统提供无限可能。

图示, 文本

描述已自动生成

图 15 ESP01-S原理图

ESP01-S模块实物图如下：

文本

中度可信度描述已自动生成

图 16 ESP01-S实物图

4.1.6 有源蜂鸣器模块

有源蜂鸣器是一种一体化结构的电子讯响器，采用直流电压供电，广泛应用于计算机、打印机、复印机、报警器、电子玩具、汽车电子设备、电话机、定时器等电子产品中作发声器件。蜂鸣器主要分为压电式蜂鸣器和电磁式蜂鸣器两种类型。蜂鸣器在电路中用字母“H”或“HA”（旧标准用“FM”、“LB”、“JD”等）表示。压电式蜂鸣器主要由多谐振荡器、压电蜂鸣片、阻抗匹配器及共鸣箱、外壳等组成。有的压电式蜂鸣器外壳上还装有发光二极管。多谐振荡器由晶体管或集成电路构成。当接通电源后（1.5~15V直流工作电压）,多谐振荡器起振,输出100~500Hz的音频信号，阻抗匹配器推动压电蜂鸣片发声。压电蜂鸣片由锆钛酸铅或铌镁酸铅压电陶瓷材料制成。在陶瓷片的两面镀上银电极，经极化和老化处理后，再与黄铜片或不锈钢片粘在一起。蜂鸣器模块实物图如下：

手机屏幕截图

低可信度描述已自动生成

图 17 蜂鸣器实物图

4.1.7 0.96寸OLED模块

图示

描述已自动生成

图18 0.96寸OLED屏幕原理图

OLED，即有机发光二极管（ Organic Light Emitting Diode）。 OLED 由于同时具备自发光，不需背光源、对比度高、厚度薄、视角广、反应速度快、可用于挠曲性面板、使用温度范围广、构造及制程较简单等优异之特性，被认为是下一代的平面显示器新兴应用技术，OLED 效果要来得好一些。以目前的技术，OLED 的尺寸还难以大型化，但是分辨率确可以做到很高。在此我们使用的是中景园电子的0.96 寸OLED 显示屏，该屏有以下特点：

1. 黄蓝屏上1/4 部分为黄光，下3/4 为蓝；而且是固定区域显示固定颜色，颜色和显示区域均不能修改。

2. 分辨率为128\*64。

3. IIC 接口方式（只需要 2 根线就可以控制 OLED）这种接口是通过屏上的BS0~BS2 来配置的。

4. 中景园电子的屏幕接口分别为四针的IIC 模块。模块都很方便使用。

4.2 软件设计

4.2.1 树莓派端软件设计

OpenCV人脸检测主要代码如下：

运行OpenCV人脸分类器，将检测到的人脸数据返回到元组中保存，根据元组中的数据在屏幕上用矩形框出人脸。

faces = self.face\_haar.detectMultiScale(self.frame, 1.2, 5)

for face\_x, face\_y, face\_w, face\_h in faces:

cv2.rectangle(self.frame, (face\_x, face\_y), (face\_x + face\_w, face\_y + face\_h), (0, 255, 0), 2)

百度人脸识别API相关代码如下：

将已经保存在本地的照片转换成base64字符串后上传给百度智能云，百度智能云将检测结果以Json形式返回给程序。程序对Json结果进行解析，得到相应的人脸信息。

class Baidu\_thread(QtCore.QThread):

faceSignal = QtCore.pyqtSignal(str)

def \_\_init\_\_(self, parent=None):

super(Baidu\_thread, self).\_\_init\_\_(parent)

def baidu\_timer\_deal(self):

f = open('faceimage.jpg', 'rb')

image = base64.b64encode(f.read())

result = client.search(str(image, 'utf-8'), IMAGE\_TYPE, GROUP)

if result['error\_msg'] == 'SUCCESS':

name = result['result']['user\_list'][0]['user\_id']

score = result['result']['user\_list'][0]['score']

if score > 80:

if name == 'gg':

print("欢迎%许连鹏")

self.faceSignal.emit('许连鹏')

else:

self.faceSignal.emit('Unknown')

print("对不起，我不认识你！")

# name = 'Unknown'

if result['error\_msg'] == 'pic not has face':

print("pic not has face")

pass

MQTT.py主要代码如下：

创建一个MQTT客户端，连接到IP为121.40.198.205、端口1883的MQTT消息服务器上。连接后调用相关函数进行保持MQTT心跳操作，防止被MQTT消息服务器踢下线。

from paho.mqtt import client as mqtt\_client

client = '' ''

def connect\_mqtt():

global client

client = mqtt\_client.Client("litbubo")

client.connect('121.40.198.205', 1883, 60)

client.loop\_forever()

def publish(msg):

result = client.publish("control", msg)

status = result[0]

if status == 0:

print(f"Send msg to topic")

else:

print(f"Failed to send message to topic")

4.2.2 门控端软件设计

PWM相关代码设计：

使用STM32的PWM功能，首先开启 TIM3时钟以及复用功能时钟，配置P A7为复用输出。其次初始化TIM3, 设置TIM3的ARR和PSC。然后设置TIM3\_CH2 的PWM模式，使能TIM3的CH2输出并使能TIM3。最后修改 TIM3\_CCR2来控制占空比。

void TIM3\_PWM\_Init(uint16\_t arr, uint16\_t psc)

{

GPIO\_InitTypeDef GPIO\_InitStruct;

TIM\_TimeBaseInitTypeDef TIM\_TimeBaseInitStruct;

TIM\_OCInitTypeDef TIM\_OCInitStruct;

RCC\_APB1PeriphClockCmd(RCC\_APB1Periph\_TIM3,ENABLE);

RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_GPIOA, ENABLE);

GPIO\_InitStruct.GPIO\_Pin = GPIO\_Pin\_7; //TIM\_CH2

GPIO\_InitStruct.GPIO\_Mode = GPIO\_Mode\_AF\_PP;

GPIO\_InitStruct.GPIO\_Speed = GPIO\_Speed\_50MHz;

GPIO\_Init(GPIOA, &GPIO\_InitStruct);

TIM\_TimeBaseInitStruct.TIM\_Period = arr;

TIM\_TimeBaseInitStruct.TIM\_Prescaler = psc;

TIM\_TimeBaseInitStruct.TIM\_ClockDivision = 0;

TIM\_TimeBaseInitStruct.TIM\_CounterMode = TIM\_CounterMode\_Up;

TIM\_TimeBaseInit(TIM3, &TIM\_TimeBaseInitStruct);

TIM\_OCInitStruct.TIM\_OCMode = TIM\_OCMode\_PWM2;

TIM\_OCInitStruct.TIM\_OutputState = TIM\_OutputState\_Enable;

TIM\_OCInitStruct.TIM\_OCPolarity = TIM\_OCPolarity\_High;

TIM\_OC2Init(TIM3, &TIM\_OCInitStruct);

TIM\_OC2PreloadConfig(TIM3, TIM\_OCPreload\_Enable);

TIM\_Cmd(TIM3, ENABLE);

}

串口相关代码设计：

使用STM32的串口功能，首先要串口时钟使能。然后进行串口参数初始化，设置波特率、字长、校验位以及停止位等信息。最后使能串口并进行相应中断的开启，在中断服务程序中发送或接收处理串口数据。

void UART2\_Init(uint32\_t bound)

{

NVIC\_InitTypeDef NVIC\_InitStruct;

USART\_InitTypeDef USART\_InitStruct;

UART2\_IO\_Init();

USART\_DeInit(USART2);

RCC\_APB1PeriphClockCmd(RCC\_APB1Periph\_USART2| RCC\_APB2Periph\_AFIO, ENABLE);

USART\_InitStruct.USART\_BaudRate = bound;

USART\_InitStruct.USART\_HardwareFlowControl = USART\_HardwareFlowControl\_None;

USART\_InitStruct.USART\_Mode = USART\_Mode\_Rx | USART\_Mode\_Tx;

USART\_InitStruct.USART\_Parity = USART\_Parity\_No;

USART\_InitStruct.USART\_StopBits = USART\_StopBits\_1;

USART\_InitStruct.USART\_WordLength = USART\_WordLength\_8b;

USART\_Init(USART2, &USART\_InitStruct);

NVIC\_InitStruct.NVIC\_IRQChannel = USART2\_IRQn;

NVIC\_InitStruct.NVIC\_IRQChannelCmd = ENABLE;

NVIC\_InitStruct.NVIC\_IRQChannelPreemptionPriority = 3;

NVIC\_InitStruct.NVIC\_IRQChannelSubPriority = 3;

NVIC\_Init(&NVIC\_InitStruct);

USART\_ITConfig(USART2,USART\_IT\_RXNE, ENABLE);

USART\_Cmd(USART2, ENABLE);

}

void USART2\_IRQHandler()

{

uint8\_t clear\_rev;

rev[rev\_pos++] = USART\_ReceiveData(USART2);

if (rev[0] != '#')

{

for (clear\_rev = 0; clear\_rev < 17; clear\_rev++)

rev[clear\_rev] = 0;

rev\_pos = 0;

}

if(rev\_pos == 16 && rev[15] == '\*')

{

rev[rev\_pos] = 0;

rev\_pos = 0;

if(strcmp( (char \*)rev, ack\_open) == 0)

{

USART\_SendData(USART2, '1');

TIM\_SetCompare2(TIM3, 190);

OLED\_Clear() ;

GPIO\_ResetBits(GPIOB,GPIO\_Pin\_0);

delay\_ms(100);

GPIO\_SetBits(GPIOB,GPIO\_Pin\_0);

}

if(strcmp( (char \*)rev, ack\_close) == 0)

{

USART\_SendData(USART2, '0');

TIM\_SetCompare2(TIM3, 180);

OLED\_Clear() ;

GPIO\_ResetBits(GPIOB,GPIO\_Pin\_0);

delay\_ms(100);

GPIO\_SetBits(GPIOB,GPIO\_Pin\_0);

}

if(strcmp( (char \*)rev, ack\_wifi) == 0)

for (clear\_rev = 0; clear\_rev < 17; clear\_rev++)

rev[clear\_rev] = 0;

}

else if(rev\_pos == 16 && rev[15] != '\*')

{

for (clear\_rev = 0; clear\_rev < 17; clear\_rev++)

rev[clear\_rev] = 0;

rev\_pos = 0;

}

}

main.c主要代码如下：

在主函数中对矩阵键盘不停扫秒，获取被按下按键的键码，根据键码执行不同的操作，例如，密码输入、密码确认和关门等操作。

while(1)

{

Key\_temp = Key\_Scan();

if(Key\_temp != KEY\_NULL)

{

switch (Key\_temp)

{

case 14:

for(clear\_passwd = 0; clear\_passwd < 6; clear\_passwd++)

passwd\_temp[clear\_passwd] = 0;

Key\_pos = 0;

break;

case 15:

for(clear\_passwd = 0; clear\_passwd < 6; clear\_passwd++)

if(passwd[clear\_passwd] != passwd\_temp[clear\_passwd])

lock\_flag = 0;

if(lock\_flag == 1)

TIM\_SetCompare2(TIM3, 190);

else

TIM\_SetCompare2(TIM3, 180);

lock\_flag = 1;

for(clear\_passwd = 0; clear\_passwd < 6; clear\_passwd++)

passwd\_temp[clear\_passwd] = 0;

break;

case 10:

TIM\_SetCompare2(TIM3, 190);

break;

case 11:

TIM\_SetCompare2(TIM3, 180);

break;

case 12:

OLED\_Display\_Off();

break;

case 13:

OLED\_Display\_On();

break;

default:

if (Key\_pos < 6)

{

passwd\_temp[Key\_pos] = Key\_temp;

OLED\_ShowChar(Key\_pos\*8, 6, '\*', 16);

Key\_pos ++;

}

break;

}

GPIO\_ResetBits(GPIOB,GPIO\_Pin\_0);

delay\_ms(100);

GPIO\_SetBits(GPIOB,GPIO\_Pin\_0);

}

}

5 实验结果分析

运行软件看到如下界面：时间标签正常走时，摄像头画面正常显示。

图形用户界面, 应用程序

描述已自动生成

图19 起始界面

点击人脸解锁按钮，程序开始检测人脸。当检测到人脸后，弹出提醒弹窗并停止检测人脸，同时发送开门命令给下位机。如图所示：

图形用户界面, 应用程序

描述已自动生成

图20 人脸识别界面

门禁收到指令，门已打开。

桌子上的手机

中度可信度描述已自动生成

图21 门禁已开门

点击关门按钮，程序发送关门指令。如图所示：

图形用户界面

描述已自动生成

图22 远程关门

桌子上有许多电子用品

描述已自动生成

图23 门禁已关门

门控端上电后会自动进行设备初始化，如下为WIFI已连接，初始化完成的效果图：

图形用户界面

中度可信度描述已自动生成

图24 门控端初始化

通过矩阵键盘手动输入6位密码后，大门解锁，门已打开：

桌子上的手机

中度可信度描述已自动生成

图25 密码解锁

6 结论

本设计实现了树莓派远端运行人脸识别，门控端接受指令远程开门的功能。做到了人脸识别与门禁完全分离开发。树莓派远端仅负责人脸检测及发送门禁控制指令，门控端仅负责机械控制等简单功能。两端通过MQTT进行远程通信，实现远程开门及对门禁状态的感知。本设计目前存在以下问题：树莓派远端对网络要求较高，当网络不通畅时，人脸扫描比较慢。门控端配网不灵活，未实现网页等灵活配网的手段。

关于本次设计，我花费了比较多的心思，既是对课程理论内容的一次复习和巩固，也让我丰富了更多与该专业相关的其他知识，比如软件应用等，在摸索中学习，在摸索中成长，在学习的过程中带着问题去学我发现效率很高，这是我做这次设计的又一收获。在真正设计之前我做了相当丰富的准备，首先巩固一下课程理论，再一遍熟悉课程知识的构架，然后结合加以理论分析、总结，有了一个清晰的思路和一个完整的软件流程图之后才着手设计。在设计程序时，不能妄想一次就将整个程序设计好，反复修改、不断改进是程序设计的必经之路；养成注释程序的好习惯是非常必要的，一个程序的完美与否不仅仅是实现功能，而应该让人一看就能明白你的思路，这样也能为资料的保存和交流提供了方便；我觉得在设计过程中遇到问题是很正常的，但我应该将每次遇到的问题记录下来，并分析清楚，以免下次再碰到同样的问题的设计又出错了。 除了对此次设计的准备工作之外，我还学到了很多平时难得的东西，首先是资料查找，在这次设计当中，难免遇到自己不了解的知识，通过这次设计，我体会到了懂得查阅文献的重要性及力量之强大，还有本次设计让我处理事情更加有条理，思路更加清晰明了，发现、提出、分析、解决问题和实践能力的提高都将受益于我在以后的学习、工作和生活中。 此次的设计，其实也是我所学知识的一次综合运用，让我深深的认识到了学习要有扎实的基础，应将所学的知识紧紧的结合在一起，综合运用，所谓设计，就是要求创新，只有将知识综合运用起来才能真正的设计好一个作品。

参考文献

[1] 孙志. 基于OpenCV的人脸识别算法实验平台研究与实现[D].吉林:吉林大学,2014.

[2] 类红乐,魏忠恒,彭延军.人脸识别机器人的设计与实现[J].山东:山东科技大学,2011.

[3] 舒连成.基于树莓派和云平台的智能家居控制装置设计[D].大连:大连理工大学,2020.

[4] 谭浩强.C程序设计[M]. 北京：清华大学出版社，1991.

[5] 张玉荣，王强强，吴琼，祝方清。基于Python-OpenCV图像处理技术研究[J]. 河南工业大学学报（自然科学版），：1-13.

[6] 董婉萍.计算机C++语言编辑程序技巧[J]. 电子技术与软件工程，2017.