**基于Blinker的智能宿舍——智能门锁和智能厕所**

产品文档

# 1 AS608系统功能与环境

## 1.1系统功能

ATK-AS608 指纹识别模块是 ALIENTEK 推出的一款高性能的光学指纹识别模块。芯片内置 DSP 运算单元，集成了指纹识别算法，能高效快速采集图像并识别指纹特征。模块配备了串口、USB 通讯接口，用户无需研究复杂的图像处理及指纹识别算法，只需通过简单的串口、USB 按照通讯协议便可控制模块。本模块可应用于各种考勤机、保险箱柜、指纹门禁系统、指纹锁等场合。

本小组通过对用户手册的阅读，了解该指纹模块基本控制原理并使用C++语言实现了该模块的关键功能（add，delete，search，list）。

## 1.2 硬件环境

CPU：ARM Cortex-A53 1.2GHz 四核；内存：1GB及以上；硬盘：16G及以上。

## 1.3 软件环境

Raspbian，Bash Shell

## 1.4 操作说明

**1.4.1 预处理**

使用树莓派的硬件进行串口通信，需要额外配置一下(关闭板载蓝牙功能等)：具体操作如下：

1.把serial打开，这个在当前最新的系统里面，能够直接在Raspberry Pi Configuration里面找到，在Interfaces里面把Serial Port改成Enable，重启即可

2.修改配置文件

sudo gedit /boot/cmdline.txt

删掉里面的 console=serial1,115200

1. 还需要把serial口子换一下，因为serial0和serial1不一样，我们需要的默认被蓝牙占用了

sudo gedit /boot/config.txt

在末尾加上一句话

dtoverlay=pi3-miniuart-bt

重启后在终端输入ls -l /dev进行查看：IMG_256

可知串口类型已修改成功。

**1.4.2 使用方法**

把本项目根目录下的as608.h和as608.c拷贝到你的程序目录下并包含头文件。还需要包含 <wiringPi.h> 和 <wiringSerial.h>。

1. 编译运行

cd example

make

./fp # 第一次使用，让程序初始化

1. 修改配置文件

方法一：编辑 ~/.fpconfig ：执行vim ~/.fpconfig

address=0xffffffff

password=none

baudrate=9600

detect\_pin=1

serial=/dev/ttyAMA0

方法二：使用命令

fp cfgaddr [address] ：修改address

fp cfgpwd [password] ：修改password

fp cfgserial [serialFile]：修改串口通信端口

fp cfgbaud [baudrate]：修改通信波特率

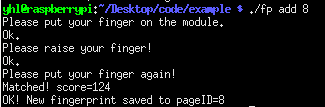
fp cfgpin [GPIO\_pin]：修改检测手指是否存在 对于的GPIO引脚

1. 示例

在进行完相关步骤后，即可通过使用命令行程序调用相应的函数来实现功能，示例如下：

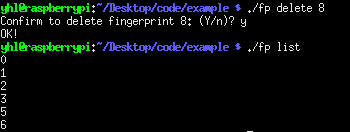
（1）添加指纹

以./fp add [fingerNum]格式调用，连续两次采集指纹信息，录入指纹。



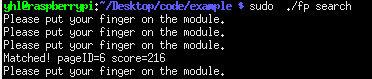
（2）删除指纹

以./fp delete [fingerNum]格式调用，删除标号为fingerNum的指纹。



（3）验证指纹

以./fp search格式调用，并一直进行循环。等待两秒，如果检测到指纹，与指纹库中的指纹进行比对，如果指纹特征符合度高于标准，则判断为指纹正确，亮蓝灯表示识别成功，并旋转舵机拉动门把手实现开门的操作。如果未检测到指纹，则继续进行循环。



# 2 Blinker物联网系系统功能与环境

## 2.1系统描述

Blinker（点灯）是一套专业且易用物联网解决方案，提供了服务器、应用、设备端sdk支持。具有IOS端、安卓端、电脑端。基于高性能异步框架开发的服务器端可以承载大量设备连接，让设备所有者方便的进行设备管理；简单便捷的应用配合多设备支持的sdk，可以让开发者在3分钟内实现设备的接入。

blinker支持多种主流通信方式，如：**蓝牙** 、 **WiFi** ，理论上只要是支持蓝牙或者WiFi的设备，都可以使用blinker连接。

使用WiFi接入，当设备和手机在同一个局域网中，为局域网通信，其余情况，使用MQTT远程通信。

## 2.2 硬件环境

树莓派：CPU：ARM Cortex-A53 1.2GHz 四核；内存：1GB及以上；硬盘：16G及以上。

Arduino：Arduino Mega 2560

## 2.3 软件环境

Raspbian，ArduinoIDE，Blinker配置环境

## 2.4 操作说明

首先Raspblian需要使用git下载blinker库，并配置环境，命令如下

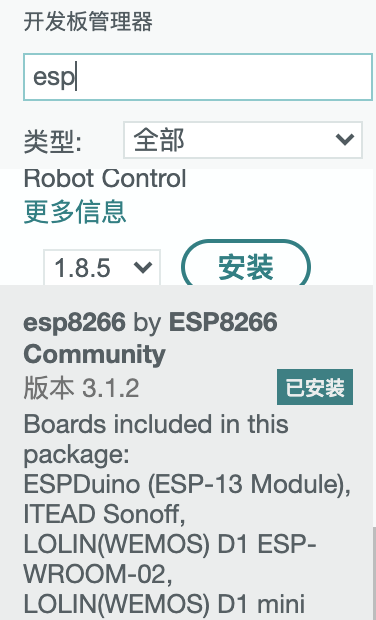
`git clone https://github.com/blinker-iot/blinker-py`

`cd blinker-py`

`sudo python3 setup.py install`

`sudo pip3 install -r requirements.txt`

Arduino需要在IDE中下载ESP8266WIFI模块库。



## 2.4.1添加设备

通过添加设备页面，您可以添加设备到您的账号。  
在**我的设备**页面点击右上角的“+”图标，即可进入**添加设备**页面，页面如下：  


图2.4.1-1 添加设备

会得到Cecret Key 设备识别码，用来唯一识别该物联网设备。

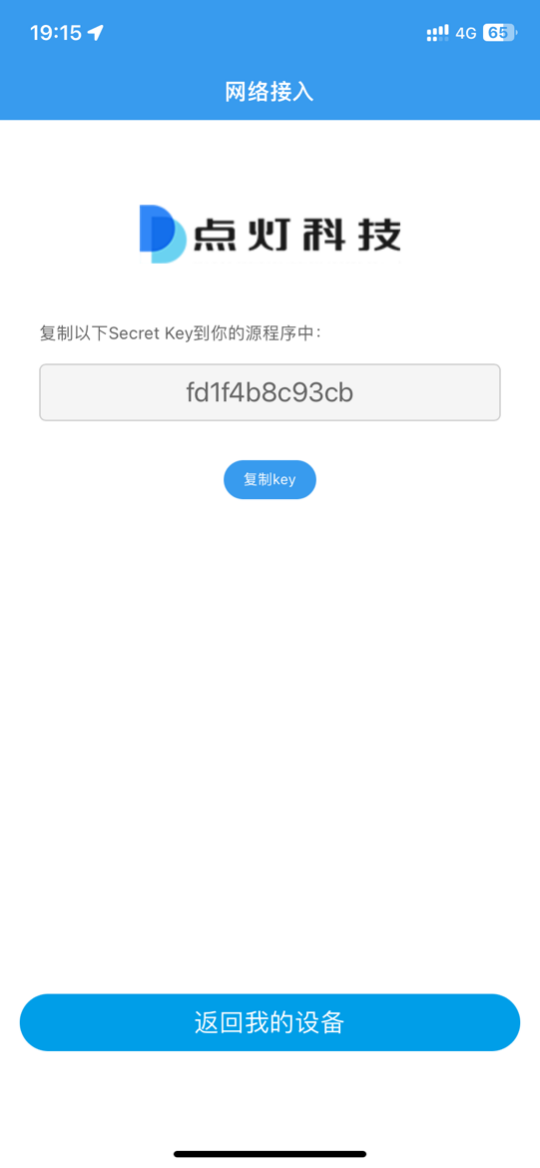


图2.4.1-2 设备识别key

## 2.4.2开启界面编辑

进入设备控制面板，点击页面右上方的编辑按钮，可以进入界面编辑状态。

#### 添加组件

在界面编辑状态下，点击下方组件图标，可以选择要添加的组件。

图2.4.2-1：编辑选项 图2.4.2-2 添加滑动组件

#### 改变组件位置

在界面编辑状态下，拖拽组件可以改变组件位置。

#### 编辑组件

点击组件，可以设置这个组件的显示风格，或改变其参数.

在编辑界面上方，是预览区域，可以看到组件设置改变后的外观,点击切换不同风格，可以改变组件的外观

中间为参数设置区域，用户可以修改其中内容，显示文本用于设置组件在app中显示的文字内容

左上不保存退出编辑模式，右上保存退出，页面底部为删除组件按钮。

**按键组件**、**开关组件**可绑定语音指令。

图2.4.2-3 按钮类型 图2.4.2-4编辑组件（按钮）

#### 保存界面

编辑好界面后，可以通过点击界面右上角的上传按钮，即可保存界面。

## 2.4.3软件运行状态

如图2.4.3-1和图2.4.3-2

1、当物联网设备连接上Blinker服务器时，软件界面正上方会显示在线。

2、Monitor监视器会显示设备状态与按钮点击情况。蓝色为手机操作反应，黑色为服务器返回的消息。

比如：get state表示手机向Blinker服务器发送请求。如果物联网设备连接了Blinker服务器，Blinker服务器会向手机发送Json格式的信息：state online。

当点金点我开关灯按钮时，monitor中会有 btn-abc “tap”的信息。btn-123是查看厕所状态，当它被点击时，服务器会返回Json格式的厕所状态。Tex-dui是文本信息的名称，分为tex和tex1：“厕所状态”和“有人”。



图2.4.3-1 图2.4.3-2

## 2.4.4设备共享

该功能允许blinker用户将自己的设备共享给其他用户使用。

blinker设备端SDK集成了用户鉴权功能，搭配blinker APP提供设备共享功能，可以实现多个用户对同一设备的控制。

**主用户 共享设备**

在 设备设置页>设备共享 或 用户中心>设备共享>对应设备 可进入 设备共享页

点击 +添加共享，输入要共享的用户的手机号（用户需已经注册blinker）

等待用户接受共享

**子用户 接受/拒绝 共享**

在用户中心>设备共享>接受 点击对应设备后的同意/拒绝

**主用户 取消共享**

在 设备设置页>设备共享 或 用户中心>设备共享>对应设备 可进入 设备共享页

点击对应用户的后的取消共享

**子用户 取消共享**

在用户中心>设备共享>接受 点击对应设备后的取消共享

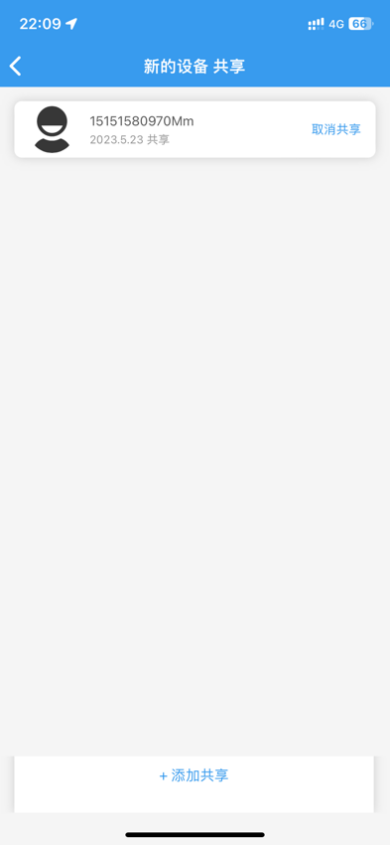


图2.4.4以上三张图从左到右展示了如何共享一个设备。