

第十四届“博创杯”全国大学生嵌入式设计大赛
作品设计报告

RFID 门禁

设
计
报
告

队伍编号： B-HB-20181210

参赛学校： 燕京理工学院

作 者： 崔文轩

指导教师： 王菊 张伟娟

组 别： 本科组

摘 要

整个系统的设计以 ArduinoMega2530 为核心,采用软硬件相结合的方式实现基于 RFID 技术的物联网化的门禁系统。本系统能实现如下功能:

- 1) App 显示设备的多种运行状态(设备 ID, 日期时间, 运行时间, 温湿度条, 附近是否有人, 锁的开关状态, 门的开关状态); App 控制锁的状态(开关, 死锁)、指示灯的开关、设备音量的大小。
- 2) 在 App 端设置为非死锁时, 允许用户刷卡锁控制锁的开关。
- 3) 使用管理卡添加或删除用户卡; 管理卡遗失可使用按键进入上帝模式删除原管理卡或所有用户; 使用按键修改设备 ID(字符串格式)。
- 4) 有人靠近时显示时间日期温度湿度等信息并记录日志和 sha1 校验值。
- 5) 硬件测试模式, 显示字符串转二维码, 显示内存卡里的图片, 等等。
- 6) 人性化的串口交互逻辑。

关键词: Arduino, Blynk, RFID, 物联网。

目 录

第一章 绪 论	4
1.1 研究背景及意义	4
1.2 本文研究的主要目标	4
第二章 系统方案	4
2.1 设计任务	4
2.2 具体模块	4
2.2.1 屏幕&tf 卡	4
2.2.1 读卡器	4
2.2.3 Arduino Mega 2530 r3	5
2.2.4 Arduino Nano	6
2.2.5 时钟模块	7
2.2.6 舵机门锁	7
2.2.7 三色灯	9
2.2.8 门磁模块	9
2.2.9 电容按键	9
2.2.10 喇叭	10
2.2.11 Wifi 模块	10
2.2.12 MP3 模块	11
2.2.13 人体检测模块	12
2.2.14 温湿度模块	12
第三章 系统流程图	12
第四章 硬件电路图	13
第五章 特色列举	15
致谢	16
参考文献	16
附录	17
1. 演示视频	17
2. 源代码	17
2.1 MegaCode_v0.1	17
2.2 NanoCode_v0.1	39

第一章 绪论

1.1. 研究背景及目的

当今科技飞速发展，给人们带来利益的同时，也带来了不法分子利用高科技进行盗窃、抢劫和犯罪等问题。怎样才能使人们的安全防范措施跟上科技的发展、有效的阻止这些犯罪行为呢？仅依靠普通的防盗门、门锁和监控是不够的。原始的安防措施就是把门锁上，由人巡逻保证安全，随着科技的进步，智能化的门禁系统已成为现代化管理的重要手段。

1.2. 研究的主要目标

针对国内目前的市场需求情况，通过 Arduino 实现了一套基于 RFID 的门禁系统。论文在探讨了基于射频识别技术的门禁系统的发展现状和技术基础上，设计了门禁系统的硬件设计和软件设计，给出了 Arduino 和 RFID 读卡器为核心的门禁系统设计方案，包括设计主程序流程图和模块子程序，并进行实际电路调试。

第二章 系统方案

2.1. 设计任务

使用树莓派搭建局域网内的流行物联网框架服务器 Blynk 和廉价开发版 Arduino 实现摘要部分所描述的功能功能。

2.2. 具体模块

2.2.1. 屏幕&tf 卡

规格参数

尺寸：2.0 寸 SPI 串行总线

分辨率：176*220

驱动 IC：ILI9225

更多

<http://pan.baidu.com/s/lgdwlZEj>

<https://item.taobao.com/item.htm?spm=alz09.2.0.0.33ba2e8dXYddtn&iid=520532136969&u=11gn5130d807>

2.2.2. 读卡器

MF522-AN 模块采用 Philips MFRC522 原装芯片设计读卡电路，使用方便，成

本低廉，适用于设备开发、读卡器开发等高级应用的用户、需要进行射频卡终端设计/生产的用户。本模块可直接装入各种读卡器模具。模块采用电压为 3.3V，通过 SPI 接口简单的几条线就可以直接与用户任何 CPU 主板相连接通信，可以保证模块稳定可靠的工作、读卡距离远；

规格参数

工作电流：13—26mA/直流 3.3V

空闲电流：10—13mA/直流 3.3V

休眠电流：<80uA

峰值电流：<30mA

工作频率：13.56MHz

支持的卡类型：mifare1 S50、mifare1 S70、mifare UltraLight、mifare Pro、mifare Desfire

产品物理特性：尺寸：40mm×60mm

环境工作温度：摄氏-20—80 度

环境储存温度：摄氏-40—85 度

环境相对湿度：相对湿度 5%—95%

模块接口 SPI 参数

数据传输速率：最大 10Mbit/s

更多

<http://box.cloud.taobao.com/s/cPKHyTldZE>

<https://item.taobao.com/item.htm?spm=alz09.2.0.0.33ba2e8dXYddtn&iid=39817054807&u=11gn51306a43>

2.2.3. Arduino Mega 2530 r3

Zduino MEGA 微控制器板，相较于其他型号的 Zduino 控制器，它提供了更多 I/O 口与更大运算储存空间。它包含了 54 个数字输入输出（其中 14 个可以作为 PWM 输出），16 个模拟输入，4 个串口（硬件串口），同时含有一个 USB 连接头，一个电源插孔，一个 ICSP 接口，一个复位按键等支持主控器运行的超小系统。用一根 USB 数据线连接到电脑，或者一个交流转直流的电源适配器，亦或是一个电池即可让它运行起来。

Zduino Mega2560 与之前的版本不同在于它未使用 USB 转串口驱动芯片 FTDI，而是用 ATmega16u2 作为 USB 转串口芯片（R2 版本则采用 ATmega8u2）。

R3 版本还有如下几点改动:

1、输出:在引脚 AREF 旁边添加了 SDA 和 SCL 接口,同时在 RESET 引脚旁边添加了 IOREF 引脚,允许扩展板采用下面主板提供的电压.使得扩展板不仅兼容主板同时还兼容用 5V 电压工作的 AVR,和用 3V3 操作的 DUE. IOREF 为空脚,预留给未来其他用途.

2、更强的复位电路

3、Atmega16u2 代替 8u2

规格参数

主控芯片: ATmega2560

工作电压: 5V

输入电压(推荐): 7-12V

输入电压(极限值): 6-20V

数字输入输出: 54(其中 15 个可被用作 PWM 输出)

模拟输出: 16

每个 IO 口的电流: 40mA

3.3V 引脚电流: 50mA

Flash: 256KB(其中 8KB 被用作 bootloader)

SRAM: 8KB

EEPROM: 4KB

时钟速率: 16Mhz

更多

<http://box.cloud.taobao.com/s/cPKHyT1dZE>

<https://item.taobao.com/item.htm?spm=alzo9.2.0.0.33ba2e8dXYddtn&id=19278476495&u=11gn5l30f1b7>

2.2.4. Arduino Nano

Arduino Nano 是一块基于开放原始代码的超小 Simple i/o 平台,与之前的 USB 版 Arduino duemilanove 相比较,Arduino Nano 在体积上占很大优势,Arduino 既可以用来开发出需要独立运行,并且具有互动效果的电子用品;也可以用来开发出与电脑相连接,同 Flash, Processing, Max/Msp, PD, VVVV 等软件一同合作完成的互动作品。

规格参数

数字接口：12 个数字输入/输出端口 D2—D13

模拟接口：8 个模拟输入端口 A0—A7

工作电压：5V 逻辑电平

输入电压（推荐）：7—12V

输入电压（限制）：6-20V

主控制器：采用 Atmel Atmega328P-AU 单片机

USB 接口：支持 USB 下载及供电

外形尺寸：45*18mm

产品重量：5g

支持 ISP 下载

更多

[http://www.alsrobot.cn/wiki/index.php/\(SKU:RB-01C012\)Arduino_Nano_mega328%E6%8E%A7%E5%88%B6%E5%99%A8](http://www.alsrobot.cn/wiki/index.php/(SKU:RB-01C012)Arduino_Nano_mega328%E6%8E%A7%E5%88%B6%E5%99%A8)
<https://item.taobao.com/item.htm?spm=alz09.2.0.0.33ba2e8dXYddtn&id=39389046541&u=11gn5130cb15>

2.2.5. 时钟模块

DS1307 I2C 实时时钟芯片 (RTC)

24C32 32K I2C EEPROM 存储器

采用 LIR2032 可充电锂电池，并带充电电路

解决 DS1307 带备用电池不能读写的问题。

充满电后，能提供 DS1307 计时 1 年。

设计小巧，27mm*28mm*8.4mm

引出 DS1307 的时钟引脚，为单片机提供时钟信号

更多

<http://pan.baidu.com/s/17vPDk>
<https://item.taobao.com/item.htm?spm=alz09.2.0.0.33ba2e8dXYddtn&id=39913853583&u=11gn513097e2>

2.2.6. 舵机门锁

产品名称：最新大扭力舵机 MG996R (MG995 升级产品)6v/11Kg

厂家编号：MG996R

产品净重：55g

产品尺寸：40.7*19.7*42.9mm

产品拉力：9.4kg/cm(4.8V)，11kg/cm(6V)

反应速度：0.17sec/60degree(4.8v) 0.14sec/60degree(6v)

工作电压：4.8-7.2V

工作温度：0℃-55℃

齿轮形式：金属齿轮

工作死区：5us（微秒）

适合机型：50 级-90 级甲醇固定翼机以及 26cc-50cc 汽油固定翼飞机等
更多

[https://item.taobao.com/item.htm?spm=alz09.2.0.0.33ba2e8dXYddtn&i
d=40434919944& u=11gn51303cc1](https://item.taobao.com/item.htm?spm=alz09.2.0.0.33ba2e8dXYddtn&i d=40434919944& u=11gn51303cc1)

2.2.7. 三色灯

全彩三色 LED 模块

更多

[https://item.taobao.com/item.htm?spm=alz09.2.0.0.33ba2e8dXYddtn&i
d=558036722139& u=11gn51300fa2](https://item.taobao.com/item.htm?spm=alz09.2.0.0.33ba2e8dXYddtn&i d=558036722139& u=11gn51300fa2)

2.2.8. 门磁模块

型号：MC-38 常常闭（合在一起导通的）

外观尺寸：27*14*10 mm

动作距离：18mm ± 6mm

寿命：100 万次

开关输出：常常闭（合在一起导通的）

明装式，适用于非铁质门或窗 表面安装

更多

[https://item.taobao.com/item.htm?spm=alz09.2.0.0.33ba2e8dXYddtn&i
d=566575087530& u=11gn51306855](https://item.taobao.com/item.htm?spm=alz09.2.0.0.33ba2e8dXYddtn&i d=566575087530& u=11gn51306855)

2.2.9. 电容按键

模块尺寸：15mmX11MM

供电：2.5v-5.5v

更多

<https://item.taobao.com/item.htm?spm=alz09.2.0.0.33ba2e8dXYddtn&iid=548917671034&u=11gn51306a72>

2.2.10. 喇叭

直径 4CM 8 欧 0.5W 0.5 瓦 小喇叭 扬声器

更多

<https://item.taobao.com/item.htm?spm=alz09.2.0.0.33ba2e8dXYddtn&iid=547359307904&u=11gn51306e72>

2.2.11. Wifi 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 的超低功耗的 UART-WiFi 模块的模组，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 WiFi 功能嵌入其他系统提供无限可能。

特点

802.11 b/g/n

内置 Tensilica L106 超低功耗 32 位微型 MCU，主频支持 80 MHz 和 160 MHz，支持 RTOS

内置 10 bit 高精度 ADC

内置 TCP/IP 协议栈

内置 TR 开关、balun、LNA、功率放大器和匹配网络

内置 PLL、稳压器和电源管理组件，802.11b 模式下+20 dBm 的输出功率

A-MPDU、A-MSDU 的聚合和 0.4 s 的保护间隔

WiFi @ 2.4 GHz，支持 WPA/WPA2 安全模式

支持 AT 远程升级及云端 OTA 升级

支持 STA/AP/STA+AP 工作模式

支持 Smart Config 功能（包括 Android 和 iOS 设备）

HSPI、UART、I2C、I2S、IR Remote Control、PWM、GPIO

深度睡眠保持电流为 10 uA，关断电流小于 5 uA

2 ms 之内唤醒、连接并传递数据包

待机状态消耗功率小于 1.0 mW (DTIM3)

工作温度范围：-20℃- 85℃

更多

<http://wiki.ai-thinker.com/esp8266>

<https://item.taobao.com/item.htm?spm=alz09.2.0.0.33ba2e8dXYddtn&id=538556979201&u=11gn5l30a55d>

<https://item.taobao.com/item.htm?spm=alz09.2.0.0.33ba2e8dXYddtn&id=570443243724&u=11gn5l30bd1f>

2.2.12. MP3 模块

JQ6500 是一个提供串口的 MP3 芯片，完美的集成了 MP3、WMV 的硬解码。同时软件支持 TF 卡驱动，支持电脑直接更新 spi flash 的内容，支持 FAT16、FAT32 文件系统。通过简单的串口指令即可完成播放指定的音乐，以及如何播放音乐等功能，无需繁琐的底层操作，使用方便，稳定可靠是此款产品的最大特点。另外该芯片也是深度定制的产品，专为固定语音播放领域开发的低成本解决方案。

- 1、支持采样率(KHz):8/11.025/12/16/22.05/24/32/44.1/48
- 2、24 位 DAC 输出，动态范围支持 90dB，信噪比支持 85dB
- 3、完全支持 FAT16、FAT32 文件系统，最大支持 32G 的 TF 卡，支持 32G 的 U 盘、64M 字节的 NORFLASH
- 4、多种控制模式，串口模式、AD 按键控制模式
- 5、广播语插播功能，可以暂停正在播放的背景音乐
- 6、音频数据按文件夹排序，最多支持 100 个文件夹，每隔文件夹可以分配 1000 首歌曲
- 7、30 级音量可调，10 级 EQ 可调
- 8、可以外挂 spi flash, 连接电脑可以显示 spi flash 的盘符进行更新内容；
- 9、可以通过单片机串口进行控制播放指定的音乐；
- 10、在按键模式下，可以进行播放模式选择：脉冲可重复、脉冲不可重复、

电平非保持可循环、电平保持可循环

更多

<https://item.taobao.com/item.htm?spm=alz09.2.0.0.33ba2e8dXYddtn&id=557193070770&u=11gn513090eb>

2.2.13. 人体检测模块

HC-SR505 小型人体感应模块是基于红外线技术的自动控制产品,灵敏度高,可靠性强,超小体积,超低电压工作模式。广泛应用于各类自动感应电器设备,尤其是干电池供电的自动控制产品。

产品特点:

全自动感应:人进入其感应范围则输出高电平,人离开感应范围则自动延时关闭高电平,输出低电平。

超小体积。

可重复触发方式:即感应输出高电平后,在延时时间段内,如果有人体在其感应范围活动,其输出将一直保持高电平,直到人离开后才延时将高电平变为低电平(感应模块检测到人体的每一次活动后会自动顺延一个延时时间段,并且以最后一次活动的时间为延时时间的起始点)。

工作电压范围宽:默认工作电压 DC4.5V-20V。

低功耗:静态电流<50 微安,特别适合干电池供电的自动控制产品。

输出高电平信号:可方便与各类电路实现对接。

更多

<http://pan.baidu.com/share/link?shareid=2730641214&uk=2302102993>
<https://item.taobao.com/item.htm?spm=alz09.2.0.0.33ba2e8dXYddtn&id=522740760591&u=11gn513062cc>

2.2.14. 温湿度模块

DHT22 单总线数字温湿度传感器 2302 模块电子积木

产品特点

尺寸: 28mm X 宽 12mm X 高 10mm

主要芯片: 奥松 DHT22 温湿度传感器

工作电压: 直流 3.3-5.5V

湿度测量范围: 0---100%RH

湿度测量精度：±2%RH

温度测量范围：-40---80℃

温度测量精度：±0.5℃

工作电压：DC5V 常用

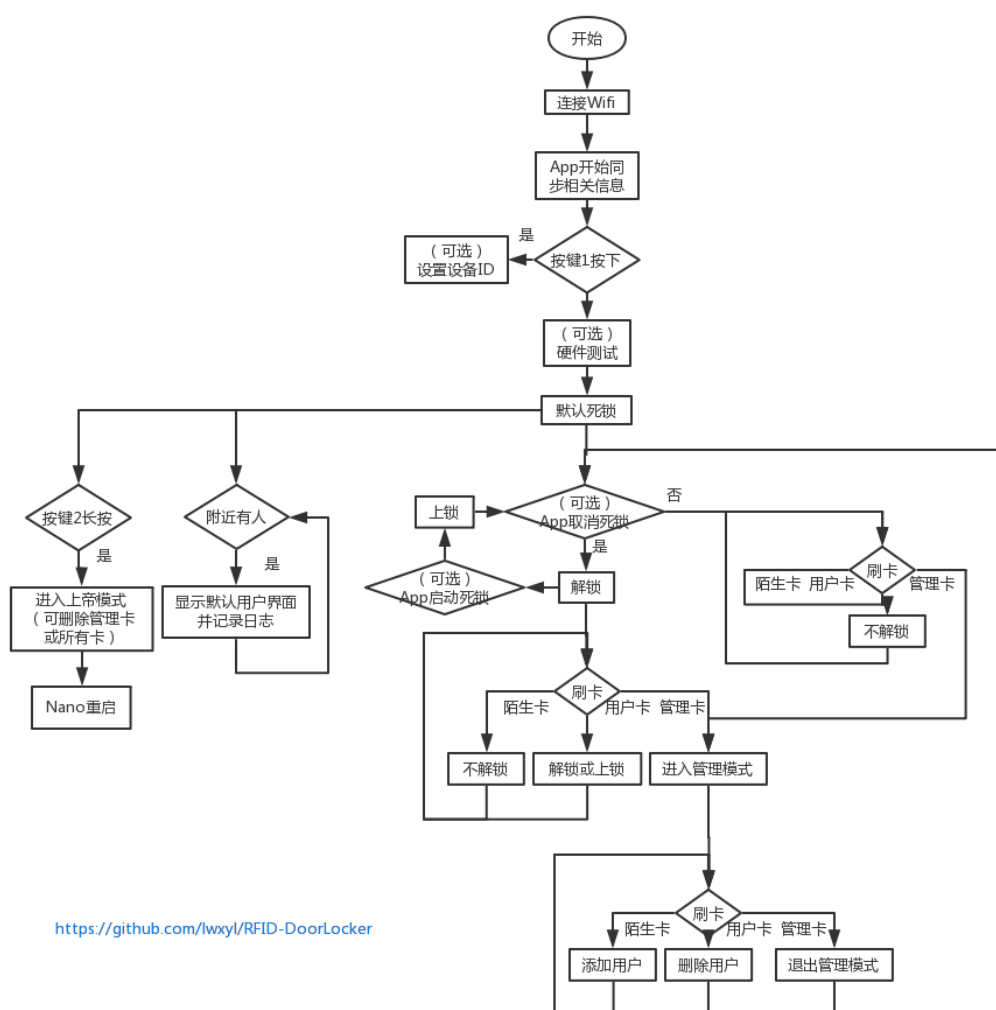
单总线数字信号输出，串口数据双向口；

更多

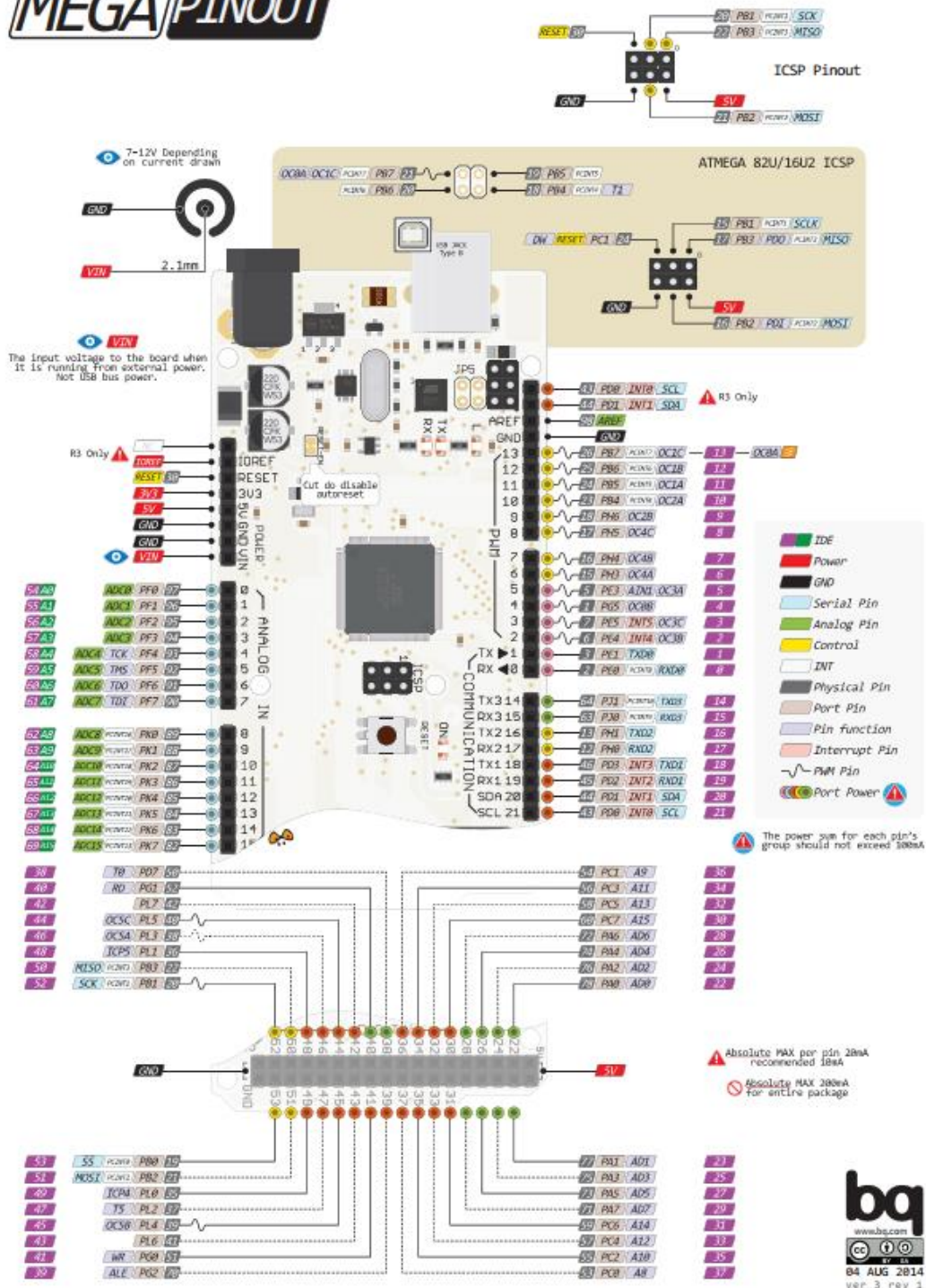
<http://pan.baidu.com/share/link?shareid=1820724183&uk=2302102993>

<https://item.taobao.com/item.htm?spm=alzo9.2.0.0.33ba2e8dXYddtn&id=40553755936&u=11gn5130b26b>

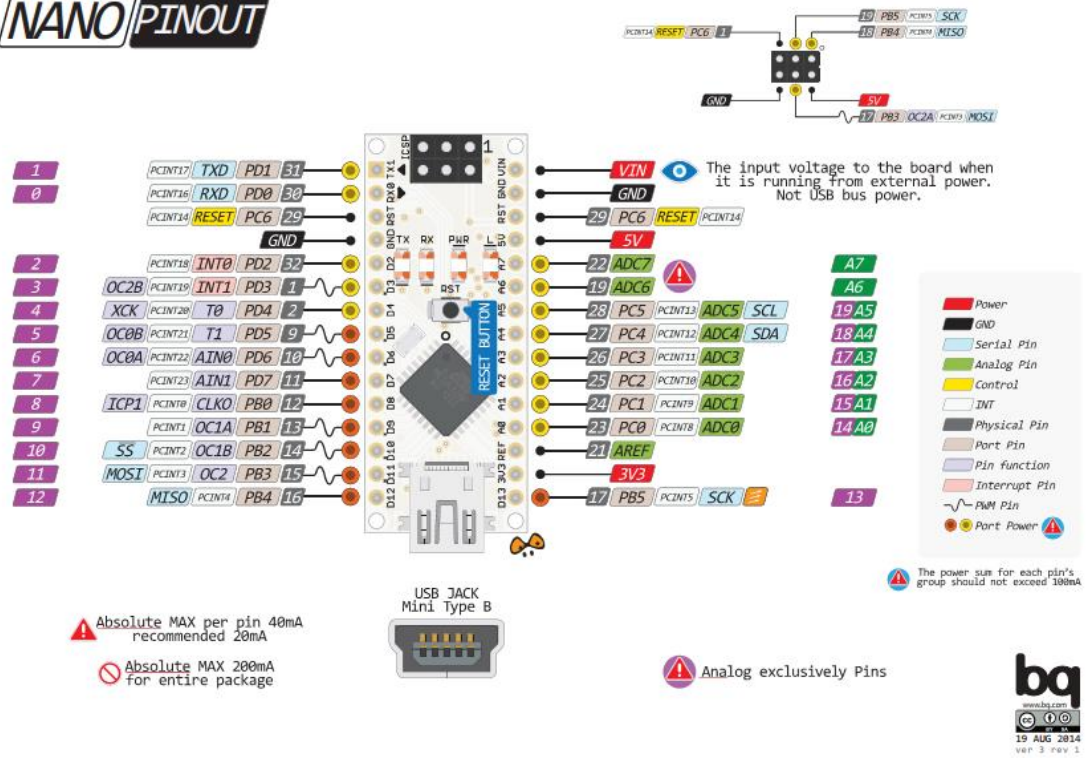
第三章 系统流程图



第四章 硬件电路图



NANO PINOUT



待完善

第五章 特色列举

1. 使用树莓派搭建了局域网内的流行物联网框架服务器 Blynk, 实现了 App 对门锁的监测和智能管理;
2. 使用廉价 Arduino 开发版实现了如日志加密存储, 绘制图像, 特殊字体显示, 字符串转二维码显示, 温湿度可视化显示等诸多功能;
3. 库文件全为 Github 进三年的新代码; 主代码简洁易懂, 技巧精妙, 且完全开源 (GNUv3.0), 作为教材也是极好的呢。

致谢

感谢博创比赛，没有这个比赛，我可能永远也不会把这个项目付诸实践。

感谢 PACKT 出版社和 Marco Schwartz, Tony Olsson 以及 Richard Grimmett, 感谢你们带我进入 Arduino 的世界。

感谢 Github 的众多开源作者和 csdn 社区的众多博主，让我有机会站在巨人的肩膀上。

感谢 Stackoverflow, arduino 论坛和 rasbian 论坛的各位回答者，在我遇到困局的时候，为我提供了很多思路。

感谢张秋菊老师，在我想要放弃的时候，给我鼓励。

感谢各位室友，在我通宵敲代码的时候，没有打我。

参考文献

- 【1.】 Marco Schwartz, Internet of Things with ESP8266, www.packpub.com , 2016, ISBN 978-1-78646-802-4
- 【2.】 Marco Schwartz, Arduino for Secret Agents, www.packpub.com , 2015, ISBN 978-1-78398-608-8
- 【3.】 Marco Schwartz, Arduino Home Automation Projects, www.packpub.com , 2014, ISBN 978-1-78398-606-4
- 【4.】 Tony Olsson , Arduino Wearable Projects , www.packpub.com , 2015 , ISBN 978-1-78528-330-7
- 【5.】 Richard Grimmett, Arduino Robotic Projects, www.packpub.com , 2014, ISBN 978-1-78398-982-9
- 【6.】 Robert C. Martin, 代码整洁之道, 人民邮电出版社 , 2016, ISBN 978-7-115-43415-9
- 【7.】 <https://examples.blynk.cc>
- 【8.】 <http://playground.arduino.cc/Code/SimpleTimer#Usage>
- 【9.】 <https://www.arduino.cc/reference/en/>
- 【10.】 <https://github.com/blynkkk/blynk-server>
- 【11.】 <https://github.com/blynkkk/blynk-library/releases/latest>
- 【12.】 <https://github.com/Makuna/Rtc/>
- 【13.】 <https://github.com/adafruit/DHT-sensor-library>
- 【14.】 <https://github.com/MickThomson21/UTFT-ILI9225>
- 【15.】 https://github.com/steelice/UTFT_DLB
- 【16.】 https://github.com/sleemanj/JQ6500_Serial
- 【17.】 https://github.com/lightcalamar/JQ6500_Serial_HardwareSerial
- 【18.】 <https://github.com/miguelbalboa/rfid>
- 【19.】 <https://github.com/filip505/EEPROM-String>
- 【20.】 <https://github.com/ricmoo/QRCode>
- 【21.】 <https://github.com/mr-glt/Arduino-SHA-1-Hash>
- 【22.】 <https://github.com/Nanotect/BitmapToArduinoArray>
- 【23.】 <https://blog.csdn.net/lishuhuakai/article/details/46048481>

附录

1. 演示视频

<https://www.bilibili.com/video/av25239370>

2. 源代码

<https://github.com/lwxyl/RFID-DoorLocker>



2.1. MegaCode_v0.1

```
1 /*****
2   RTC DS1307: https://github.com/Makuna/Rtc/
3   GND - GND
4   VCC - 5V
5   SDA - 20
6   SCL - 21
7   DS -
8
9   ESP-01S:
10  https://github.com/blynkkk/blynk-library/releases/latest
11  3V3 - 3V
12  RX1 - TX1
13  EN - 3V
14  TX1 - RX1
15  GND - GND
16
17  NANO:
18  D6 - RX2
19  D5 TX2
20  GND - GND
21
22  RGB LED:
23  R - 2
24  G - 3
25  B - 4
26  GND - GND
27
28  DHT22: https://github.com/adafruit/DHT-sensor-library
29  "+" - 5V
30  OUT - 5
31  "-" - GND
```

```

32 HCSR505:
33 "+" - 5V
34 OUT - 6
35 "-" - GND
36
37 SERVO:
38 GND - GND
39 5V - 5V
40 OUT - 7
41
42
43 TFT: https://github.com/MickThomson21/UTFT-ILI9225
44 5V - 5V
45 GND - GND
46 GND -
47 NC - VIN
48 NC -
49 LED - A0
50 SCL - A1
51 SDA - A2
52 RS - A3
53 RST - A4
54 CS - A5
55 SD_CS - 8
56 SD_MOSI - 51
57 SD_SCK - 52
58 SD_MISO - 50
59
60 TOUCH:
61 GND - GND
62 OUT - 9
63 VCC - 5V
64
65 Door Magnetic sensor:
66 GND - GND
67 OUT - 10
68
69 JQ6500: https://github.com/sleemanj/JQ6500\_Serial
70 RX+1k(L1) - 10
71 TX(L2) - 11
72 GND(L3) - GND
73 DC-5V(L4) - 5V
74 SPK+(L7) - SPK+
75 SPK-(L8) - SPK-

```

```

76
77   MFRC522: https://github.com/miguelbalboa/rfid
78   ->NANO
79
80   *****/
81
82   #define BLYNK_PRINT Serial
83   #define EspSerial Serial1
84   #define BLYNK_MAX_READBYTES 1024
85   #define BLYNK_EXPERIMENTAL
86   #define JQ6500Serial Serial2
87   #define NanoSerial Serial3
88
89   #define LED_RED 2
90   #define LED_GREEN 3
91   #define LED_BLUE 4
92   #define LED_RGB 0
93   #define FOREVER 0
94   #define LED_RGBLOOP_TYPE1 -1
95   #define LED_RGBLOOP_TYPE2 -2
96   #define LEDSTATE_CHANGEABLE HIGH
97   #define LEDSTATE_UNCHANGEABLE LOW
98   #define LED_ON HIGH
99   #define LED_OFF LOW
100  #define SHORTTIME 300
101  #define LONGTIME 1000
102
103  #define DHTPIN 5
104
105  #define HCSR505PIN 6
106  #define FINDSOMEBODY HIGH
107  #define NOBODY LOW
108
109  #define SERVOPIN 7
110  #define LOCKED HIGH
111  #define UNLOCKED LOW
112
113  #define SDPIN 8
114
115  #define TOUCHPIN 9
116  #define TOUCHED HIGH
117  #define UNTOUCHED LOW
118
119  #define DOORPIN 10

```

```

120 #define OPENED HIGH
121 #define CLOSED LOW
122
123
124
125 #include <ESP8266_Lib.h> //
https://github.com/blynkkk/blynk-library/releases/latest
126 #include <BlynkSimpleShieldEsp8266.h> //
https://github.com/blynkkk/blynk-library/releases/latest
127 #include <Wire.h> // must be included here so that Arduino library
    object file references work
128 #include <RtcDS1307.h> // https://github.com/Makuna/Rtc/
129 #include <DHT.h> //
https://github.com/adafruit/DHT-sensor-library
130 #include <Servo.h>
131 #include <UTFT.h>
    https://github.com/MickThomson21/UTFT-ILI9225
132 #include <UTFT_DLB.h> // https://github.com/steelice/UTFT\_DLB
133 #include <SPI.h>
134 #include <SD.h>
135 #include <qrcode.h> // https://github.com/ricmoo/QRCode
136 #include <Hash.h> // https://github.com/mr-glt/Arduino-SHA-1-Hash
137 #include <EEPROM.h>
138 #include <JQ6500_Serial_HardwareSerial.h>
    https://github.com/lightcalamar/JQ6500\_Serial\_HardwareSerial
139
140
141
142 char auth[] = "60c1636fb25b47d1900311a000503a01"; //blynk project
    auth tokens
143 char ssid[] = "USR-G800-118D";
144 char pass[] = "123456780";
145 ESP8266 wifi(&EspSerial);
146 BlynkTimer timer;
    http://playground.arduino.cc/Code/SimpleTimer#Usage
147 uint8_t FBtimer, CLtimer; //findBody, checkLocker
148
149 File Log;
150 RtcDS1307<TwoWire> Rtc(Wire);
151 DHT dht(DHTPIN, DHT22);
152 Servo myservo;
153 JQ6500_Serial_HardwareSerial mp3(JQ6500Serial);
154 uint8_t VOLUME = 30;
155 bool TEST = true;

```

```

156  bool LED_RED_STATE = LEDSTATE_CHANGEABLE, LED_GREEN_STATE =
    LEDSTATE_CHANGEABLE, LED_BLUE_STATE = LEDSTATE_CHANGEABLE;
157  bool LockStage = LOCKED, LockedByApp = true;
158  bool writedtlog = true, writetemlog = true, writehumlog = true;
159  UTFT_DLB myGLCD(QD220A, A2, A1, A5, A4, A3);
160  extern uint8_t BigFont[];
161  //extern uint8_t SmallFont[];
162  //extern uint8_t SevenSegNumFont[];
163  //extern uint8_t DejaVuSans18[];
164  extern uint8_t DejaVuSans24[];
165  extern uint8_t BeynoBlackPanther24[]; //logos
166  extern uint8_t BVS_13[]; //logs
167  extern uint8_t Farrington7BQiqi16[]; //idnums
168  extern uint8_t Lcd2u24[];
169
170
171
172  void setup() {
173      Serial.begin(115200);
174      Serial.println("==>setup-----");
175
176      EspSerial.begin(115200);
177      delay(10);
178      Blynk.begin(auth, wifi, ssid, pass, "192.168.1.5", 8080);
179
180      JQ6500Serial.begin(9600);
181      mp3.reset();
182      mp3.setVolume(VOLUME);
183
184      setDevicesId();
185
186      NanoSerial.begin(9600);
187
188      myGLCD.InitLCD();
189      myGLCD.clrScr();
190
191      setDataTime();
192
193      dht.begin();
194
195      pinMode(LED_RED, OUTPUT);
196      pinMode(LED_GREEN, OUTPUT);
197      pinMode(LED_BLUE, OUTPUT);
198

```

```

199     pinMode(SDPIN, OUTPUT);
200
201     pinMode(HCSR505PIN, INPUT);
202
203     pinMode(TOUCHPIN, INPUT);
204
205     myservo.attach(SERVOPIN);
206     pinMode(SERVOPIN, OUTPUT);
207
208     pinMode(DOORPIN, INPUT);
209
210     if (TEST) {
211         hardwareTest();
212         TEST = !TEST;
213     }
214
215     lockLocker();
216
217     FBtimer = timer.setInterval(1000, findBody);
218     CLtimer = timer.setInterval(1000, checkLocker);
219
220     Serial.println("==>setup-----Done!");
221 }
222
223
224 void loop() {
225     Blynk.run();
226     timer.run();
227 }
228
229
230 void hardwareTest() {
231     Serial.println("==>hardwareTest-----");
232     drawQrcode("http://yit.edu.cn");
233     delay(2000);
234     displayPictureInSdcard("yitlogo.raw");
235     delay(2000);
236     testRtc();
237     testDht();
238     //testRgbLed();
239     //testServo();
240     //testTouch();
241     //testmp3();
242     Serial.println("==>hardwareTest-----Done!");

```

```

243 }
244
245
246 void testRtc() {
247     Serial.println("==>testRtc-----");
248     setDataTime();
249     displayDataAndTime();
250     Serial.println("==>testRtc-----Done!");
251 }
252
253
254 void testDht() {
255     Serial.println("==>testDht-----");
256     displayTemperature();
257     displayHumidity();
258     Serial.println("==>testDht-----Done!");
259 }
260
261
262 void testRgbLed() {
263     Serial.println("==>testRgbLed-----");
264     ledBlink(LED_RED, 3, SHORTTIME);
265     ledBlink(LED_GREEN, 3, LONGTIME);
266     ledBlink(LED_BLUE, 3, SHORTTIME);
267     ledBlink(LED_RGBLOOP_TYPE1, 3, LONGTIME);
268     ledBlink(LED_RGBLOOP_TYPE2, 3, SHORTTIME);
269     ledSolid(LED_RED, 3000);
270     Serial.println("==>testRgbLed-----Done!");
271 }
272
273
274 void testServo() {
275     Serial.println("==>testServo-----");
276     lockLocker();
277     delay(2000);
278     unlockLocker();
279     delay(2000);
280     Serial.println("==>testServo-----Done!");
281 }
282
283
284 void testTouch() {
285     if (digitalRead(TOUCHPIN) == TOUCHED) {
286         Serial.println("TOUCHED");

```

```

287     mp3.playFileByIndexNumber(11);
288 }
289 else {
290     Serial.println("UNTOUCHED");
291 }
292 }
293
294
295 void testmp3() {
296     mp3.playFileByIndexNumber(1); //管理卡添加完成
297     delay(2000);
298     mp3.playFileByIndexNumber(2); //已进入管理模式
299     delay(2000);
300     mp3.playFileByIndexNumber(3); //已退出管理模式
301     delay(2000);
302     mp3.playFileByIndexNumber(4); //新卡添加完成
303     delay(2000);
304     mp3.playFileByIndexNumber(5); //删除完成
305     delay(2000);
306     mp3.playFileByIndexNumber(6); //屌丝请进
307     delay(2000);
308     mp3.playFileByIndexNumber(7); //管理员请进
309     delay(2000);
310     mp3.playFileByIndexNumber(8); //土豪请进
311     delay(2000);
312     mp3.playFileByIndexNumber(9); //余额不足
313     delay(2000);
314     mp3.playFileByIndexNumber(11); //dong ong ong ong ong ong
315     delay(2000);
316     mp3.setLoopMode(MP3_LOOP_ONE);
317     mp3.playFileByIndexNumber(10); //beep beep beep beep beep beep
318     delay(5000);
319     mp3.setLoopMode(MP3_LOOP_ONE_STOP);
320 }
321
322
323
324 #define MULTIPLIER 3
325 void drawQrcode(const char* url) {
326     Serial.println("==>drawQrcode-----");
327     QRCode qrcode;
328     uint8_t qrcodeData[qrcode_getBufferSize(9)];
329     qrcode_initText(&qrcode, qrcodeData, 9, 0, url);
330     myGLCD.InitLCD();

```



```

331 myGLCD.fillScr(255, 255, 255);
332 for (uint8_t y = 0; y < qrcode.size; y++) {
333     for (uint8_t x = 0; x < qrcode.size; x++) {
334         if (qrcode_getModule(&qrcode, x, y)) {
335             myGLCD.setColor(0, 0, 0);
336             myGLCD.fillRect(MULTIPLIER * x, MULTIPLIER * y, MULTIPLIER
* x + MULTIPLIER, MULTIPLIER * y + MULTIPLIER);
337         }
338     }
339 }
340 Serial.println("==>drawQrcode-----Done!");
341 }
342
343
344
345 BLYNK_READ(V0) { //transDevicesID
346     Blynk.virtualWrite(V0, eepromLoadString(0));
347 }
348 void setDevicesId() {
349     if (digitalRead(TOUCHPIN) == TOUCHED) {
350         Serial.println("Stop touching now then input DevicesID");
351         Serial.println("Or you will quit set DevicesID after 3
seconds");
352         if (touchedTime() > 3000) {
353             Serial.println("Quit set DevicesID");
354             return;
355         }
356         Serial.println("Please input a string, it will be set as
DevicesID:");
357         Serial.println("Or you can input 'Q' to quit set DevicesID");
358         while (1) {
359             if (Serial.available()) {
360                 String str = Serial.readString();
361                 if (str == "Q" || str == "q") {
362                     Serial.println("Quit set DevicesID");
363                     return;
364                 }
365                 char* devicesID = const_cast<char*>(str.c_str());
366                 Serial.print("Set DevicesID as: "); Serial.print(str);
Serial.println(" ?");
367                 Serial.println("Re input DevicesID again for commit");
368                 while (1) {
369                     if (Serial.available()) {
370                         String str2 = Serial.readString();

```

```

371         if (str == str2) {
372             eepromSaveString(devicesID, 0);
373             Serial.print("DevicesID had been seted as: ");
Serial.println(eepromLoadString(0));
374             Serial.println("==>setDevicesId-----Done!");
375             return;
376         }
377         else {
378             Serial.println("ERROR: Re INPUT DIFFERENT!!!");
379             Serial.println("Please input a string, it will be set
as DevicesID:");
380             break;
381         }
382     }
383 }
384 }
385 }
386 }
387 }
388 uint8_t touchedTime() {
389     uint8_t touchedtime = 0;
390     if (digitalRead(TOUCHPIN) == TOUCHED) {
391         Serial.println("Touched");
392         mp3.playFileByIndexNumber(11);
393         uint32_t starttime = millis();
394         while (digitalRead(TOUCHPIN) == TOUCHED) {
395             touchedtime = millis() - starttime;
396             if (touchedtime > 3000) {
397                 return 3001;
398             }
399         }
400     }
401     return touchedtime;
402 }
403 uint8_t count = 3; //https://github.com/filip505/EEPROM-String
404 void eepromSaveString(char *data, uint8_t id) {
405     uint8_t dataSize = strlen(data);
406     EEPROM.write(id, dataSize);
407     uint8_t start = 0;
408     for (uint8_t i = 0; i < id; i++)
409     {
410         start += EEPROM.read(i);
411     }
412     dataSize = dataSize + count + start;

```

```

413     for (uint8_t i = count + start; i < dataSize; i++)
414     {
415         EEPROM.write(i, data[i - count - start]);
416     }
417 }
418 String eepromLoadString(uint8_t id) {
419     uint8_t dataSize = EEPROM.read(id);
420     String rez;
421     uint8_t endNum = dataSize + count;
422     uint8_t start = 0;
423     for (uint8_t i = 0; i < id; i++)
424     {
425         start += EEPROM.read(i);
426     }
427     endNum += start;
428     for (uint8_t i = count + start; i < endNum; i++)
429     {
430         rez += char(EEPROM.read(i));
431     }
432     return rez;
433 }
434
435
436 BLYNK_READ(V1) { //transData e.g. 2018/06/22 W:5
437     RtcDateTime dt = Rtc.GetDateTime();
438     char datebufwithyear[20];
439     sprintf( datebufwithyear, "%04u/%02u/%02u", dt.Year(),
dt.Month(), dt.Day() );
440     uint8_t week = calculateWeek( dt.Year(), dt.Month(), dt.Day() );
441     String transdata = String(datebufwithyear) + " W:" +
String(week) ;
442     Blynk.virtualWrite(V1, transdata);
443 }
444 uint8_t calculateWeek(uint8_t y, uint8_t m, uint8_t d) {
445     if (m == 1) m = 13;
446     if (m == 2) m = 14;
447     return (d + 2 * m + 3 * (m + 1) / 5 + y + y / 4 - y / 100 + y /
400) % 7 + 1;
448 }
449
450
451 BLYNK_READ(V2) { //transTime e.g. 08:00
452     RtcDateTime dt = Rtc.GetDateTime();
453     char timebuf[20];

```

```

454     sprintf( timebuf, "%02u:%02u", dt.Hour(), dt.Minute() );
455     String transtime = String(timebuf);
456     Blynk.virtualWrite(V2, transtime);
457 }
458
459
460 BLYNK_READ(V3) { //transRunTime e.g. 123
461     Blynk.virtualWrite(V3, millis() / 1000);
462 }
463
464
465 BLYNK_READ(V4) { //transTemperature e.g. 26.5
466     Blynk.virtualWrite(V4, dht.readTemperature());
467 }
468
469
470 BLYNK_READ(V5) { //transHumidity e.g. 56,5
471     Blynk.virtualWrite(V5, dht.readHumidity());
472 }
473
474
475 WidgetLED bodyLED(V6); //transFindBody
476 void findBody() {
477     if (digitalRead(HCSR505PIN) == FINDSOMEBODY) {
478         //Serial.println();
479         //Serial.println("Find somebody");
480         //ledSolid(LED_BLUE, FOREVER);
481         mp3.playFileByIndexNumber(11);
482         bodyLED.on();
483         myGLCD.InitLCD();
484         displayDefaultInterface();
485     }
486     else {
487         //ledTurnOff(LED_BLUE);
488         bodyLED.off();
489     }
490 }
491
492
493 BLYNK_WRITE(V9) { //lockByApp
494     if (param.asInt()) {
495         lockLocker();
496         LockedByApp = true;
497     }

```

```

498     else {
499         unlockLocker();
500         LockedByApp = false;
501     }
502 }
503
504
505 WidgetLED lockLED(V7); //transLockStage
506 void changeLockStage() {
507     Serial.println("==>changeLockStage-----");
508     if (LockedByApp) {
509         Serial.println("locked by app");
510         Serial.println("==>changeLockStage-----Denied!");
511         ledBlink(LED_RED, 1, LONGTIME);
512     }
513     else {
514         Serial.println("lockstage changeable");
515         timer.disable(CLtimer); timer.disable(FBtimer);
516         if (LockStage == LOCKED) {
517             unlockLocker();
518         }
519         else {
520             lockLocker();
521         }
522         Serial.println("==>changeLockStage-----Done!");
523         timer.enable(CLtimer); timer.enable(FBtimer);
524     }
525 }
526 void lockLocker() {
527     myservo.attach(SERVOPIN);
528     myservo.write(100);
529     ledBlink(LED_RED, 1, LONGTIME);
530     timer.setTimeout(800L, []() {
531         myservo.detach();
532     });
533     LockStage = LOCKED;
534     lockLED.on();
535     Serial.println("Locked :)");
536 }
537 void unlockLocker() {
538     myservo.attach(SERVOPIN);
539     myservo.write(0);
540     ledBlink(LED_GREEN, 1, LONGTIME);
541     timer.setTimeout(800L, []() {

```

```

542     myservo.detach();
543   });
544   LockStage = UNLOCKED;
545   lockLED.off();
546   Serial.println("Unlocked :(");
547 }
548
549
550 WidgetLED doorLED(V8); //transDoorStage
551 bool checkDoor() {
552   bool doorstate = digitalRead(DOORPIN);
553   if (doorstate == OPENED) {
554     Serial.println();
555     Serial.println("Door opened");
556     doorLED.on();
557     mp3.playFileByIndexNumber(11);
558   }
559   else {
560     doorLED.off();
561   }
562   return doorstate;
563 }
564
565
566 BLYNK_WRITE(V10) { //changeVolumeByApp
567   VOLUME = param.asInt();
568   Serial.print("volume: ");
569   Serial.print(VOLUME);
570   mp3.setVolume(VOLUME);
571   mp3.playFileByIndexNumber(11);
572   Serial.println("==>setvolume-----Done!");
573 }
574
575
576 BLYNK_WRITE(V11) { //changeRedLedHintableByApp
577   if (param.asInt() == 0) {
578     ledTurnOff(LED_RED);
579     ledBlink(LED_RED, 2, SHORTTIME);
580     LED_RED_STATE = LEDSTATE_UNCHANGEABLE;
581   }
582   else {
583     LED_RED_STATE = LEDSTATE_CHANGEABLE;
584     ledBlink(LED_RED, 2, SHORTTIME);
585   }

```

```

586 }
587
588
589 BLYNK_WRITE(V12) { //changeGreenLedHintableByApp
590     if (param.asInt() == 0) {
591         ledTurnOff(LED_GREEN);
592         ledBlink(LED_GREEN, 2, SHORTTIME);
593         LED_GREEN_STATE = LEDSTATE_UNCHANGEABLE;
594     }
595     else {
596         LED_GREEN_STATE = LEDSTATE_CHANGEABLE;
597         ledBlink(LED_GREEN, 2, SHORTTIME);
598     }
599 }
600
601
602 BLYNK_WRITE(V13) { //changeBlueLedHintableByApp
603     if (param.asInt() == 0) {
604         ledTurnOff(LED_BLUE);
605         ledBlink(LED_BLUE, 2, SHORTTIME);
606         LED_BLUE_STATE = LEDSTATE_UNCHANGEABLE;
607     }
608     else {
609         LED_BLUE_STATE = LEDSTATE_CHANGEABLE;
610         ledBlink(LED_BLUE, 2, SHORTTIME);
611     }
612 }
613
614
615 void checkLocker() {
616     if (NanoSerial.available()) {
617         String command = NanoSerial.readString();
618         Serial.println(command);
619         if (command == "Welcome") {
620             changeLockStage();
621         }
622     }
623 }
624
625
626 void setDataTime() {
627     Rtc.Begin();
628     RtcDateTime dt = RtcDateTime(__DATE__, __TIME__);
629     if (!Rtc.IsDateTimeValid()) {

```

```

630     Serial.println("ERROR: RTC LOW BATTERY!!!");
631     myGLCD.setColor(255, 255, 255); myGLCD.setBackColor(0, 0, 0);
myGLCD.setFont(BVS_13);
632     myGLCD.print("ERROR: RTC LOW BATTERY!!!", 10, 10, 0);
633 }
634 if (!Rtc.GetIsRunning()) {
635     Rtc.SetIsRunning(true);
636 }
637 Rtc.SetDateTime(dt);
638 Rtc.SetSquareWavePin(DS1307SquareWaveOut_Low);
639 }
640
641
642 void displayPictureInSdcard(String PictureName) {
643     Serial.println("==>displayPictureInSdcard-----");
644     if (SD.begin(SDPIN)) {
645         File rawpicture = SD.open(PictureName, FILE_READ);
646         myGLCD.InitLCD();
647         uint8_t r, g, b;
648         for (uint8_t y = 0; y < 176; y++) {
649             for (uint8_t x = 0; x < 220; x++) {
650                 r = rawpicture.read();
651                 g = rawpicture.read();
652                 b = rawpicture.read();
653                 myGLCD.setColor(r, g, b);
654                 myGLCD.drawPixel(x, y);
655             }
656         }
657         rawpicture.close();
658     }
659
660     Serial.println("==>displayPictureInSdcard-----Done!");
661 }
662
663
664 #define BGSTR1_POSITION_X 53 //45
665 #define BGSTR1_POSITION_Y 65 //65
666 #define BGSTR2_POSITION_X 53 //53
667 #define BGSTR2_POSITION_Y 95 //95
668 void displayBackgroundText() {
669     String bgstr1 = "Y I T", bgstr2 = "BO CHUANG";
670     myGLCD.fillScr(20, 20, 20);
671     //myGLCD.setColor(50, 50, 50);
672     //myGLCD.fillRect(30, 30, 220 - 30, 176 - 30);

```



```

673     myGLCD.setColor(155, 0, 255); myGLCD.setBackColor(20, 20, 20);
myGLCD.setFont(BeynoBlackPanther24); //瓦坎达紫
674     //myGLCD.drawRect(30, 30, 220 - 30, 176 - 30);
675     myGLCD.print(bgstr1, BGSTR1_POSITION_X, BGSTR1_POSITION_Y, 0);
myGLCD.print(bgstr2, BGSTR2_POSITION_X, BGSTR2_POSITION_Y, 0);
676 }
677
678
679 void displayDefaultInterface() {
680     Serial.println("==>displayDefaultInterface-----");
681     timer.disable(CLtimer); timer.disable(FBtimer);
682     displayBackgroundText();
683     writedtlog = true; writetemlog = true; writehumlog = true;
684     //myGLCD.fillScr(0, 0, 100);
685     displayDataAndTime(); displayTemperature(); displayHumidity();
686     timer.enable(CLtimer);
687     timer.setTimeout(10000, []() {
688         myGLCD.clrScr();
689         timer.enable(FBtimer);
690         Serial.println("==>displayDefaultInterface-----Done!");
691     } );
692 }
693
694
695 #define DATA_POSITION_X 135
696 #define DATA_POSITION_Y 5
697 #define TIME_POSITION_X DATA_POSITION_X
698 #define TIME_POSITION_Y DATA_POSITION_Y+20
699 #define WEEK_POSITION_X DATA_POSITION_X+3*15
700 #define WEEK_POSITION_Y TIME_POSITION_Y+20
701 void displayDataAndTime() {
702     RtcDateTime dt2 = Rtc.GetDateTime();
703     if (!Rtc.IsDateTimeValid()) {
704         Serial.println("ERROR: RTC LOW BATTERY!!!");
705         myGLCD.setColor(255, 255, 255); myGLCD.setBackColor(0, 0, 0);
myGLCD.setFont(BVS_13);
706         myGLCD.print("ERROR: RTC LOW BATTERY!!!", 10, 10, 0);
707     }
708     char datebuf[20], datebufwithyear[20], timebuf[20],
timebufwithsecond[20];
709     sprintf( datebuf, "%02u/%02u", dt2.Month(), dt2.Day() );
710     sprintf( datebufwithyear, "%04u/%02u/%02u", dt2.Year(),
dt2.Month(), dt2.Day() );
711     sprintf( timebuf, "%02u:%02u", dt2.Hour(), dt2.Minute() );

```

```

712     sprintf( timebufwithsecond, "%02u:%02u:%02u", dt2.Hour(),
dt2.Minute(), dt2.Second() );
713     uint8_t week = calculateWeek( dt2.Year(), dt2.Month(),
dt2.Day() );
714     if (writedtlog == true) {
715         String dtlog = "DevicesID: " + eepromLoadString(0) + " Data:
" + String(datebufwithyear) + " Week: " + String(week) + " Time:
" + String(timebufwithsecond);
716         String sha1dtlog = "SHA1_dtlog: " + String(sha1(dtlog));
717         Serial.println(dtlog);
718         Serial.println(sha1dtlog);
719         Log = SD.open("Log.txt", FILE_WRITE);
720         if (Log) {
721             //Log.println(dtlog);
722             Log.println(sha1dtlog);
723             Log.close();
724         }
725         else
726         {
727             Serial.println("ERROR: OPEN LOG.TXT");
728         }
729         writedtlog = false;
730     }
731     myGLCD.setColor(255, 255, 255); myGLCD.setBackColor(0, 0, 0);
myGLCD.setFont(BigFont);
732     myGLCD.print(datebuf, DATA_POSITION_X, DATA_POSITION_Y, 0);
733     myGLCD.print(timebuf, TIME_POSITION_X, TIME_POSITION_Y, 0);
734     myGLCD.print(".", String(week), WEEK_POSITION_X,
WEEK_POSITION_Y);
735 }
736
737
738 #define TEMPERATURE_POSITION_X 5
739 #define TEMPERATURE_POSITION_Y DATA_POSITION_Y
740 void displayTemperature() {
741     float temperature = dht.readTemperature();
742     if ( isnan(temperature) ) {
743         Serial.println("ERROR: READ TEMPERATURE!!!");
744         myGLCD.setColor(255, 255, 255); myGLCD.setBackColor(0, 0, 0);
myGLCD.setFont(BVS_13);
745         myGLCD.print("ERROR: READ TEMPERATURE!!!", 10, 30, 0);
746     }
747     else {
748         if (writetemlog == true) {

```

```

749     String temlog = "Temperature: " + String(temperature) + " *C";
750     Serial.println(temlog);
751     Log = SD.open("Log.txt", FILE_WRITE);
752     if (Log) {
753         Log.println(temlog);
754         Log.close();
755     }
756     else
757     {
758         Serial.println("ERROR: OPEN LOG.TXT");
759     }
760     writetemlog = false;
761 }
762 uint8_t inttemperature = int(temperature);
763 graphicalTemperature(inttemperature);
764 myGLCD.setBackColor(0, 0, 0); myGLCD.setFont(BigFont);
765 myGLCD.print(String(inttemperature) + "*C",
TEMPERATURE_POSITION_X, TEMPERATURE_POSITION_Y);
766 }
767 }
768
769
770 void graphicalTemperature(uint8_t inttemperature) {
771     myGLCD.setColor(150, 150, 150); myGLCD.setBackColor(0, 0, 0);
772     myGLCD.fillCircle(12, 165, 5); myGLCD.fillRoundRect(10, 65, 15,
165);
773     if (inttemperature <= 10) {
774         myGLCD.setColor(30, 30, 220); //深蓝
775         Serial.println("WARNING: LOW TEMPERATURE !!!");
776     }
777     else if (inttemperature > 10 && inttemperature <= 25) {
778         myGLCD.setColor(0, 200, 255); //浅蓝
779     }
780     else if (inttemperature > 25 && inttemperature <= 32) {
781         myGLCD.setColor(255, 255, 0); //金黄
782     }
783     else if (inttemperature > 32 && inttemperature <= 40) {
784         myGLCD.setColor(255, 200, 0); //橙黄
785     }
786     else if (inttemperature > 40 && inttemperature <= 50) {
787         myGLCD.setColor(255, 0, 0); //红
788     }
789     else {
790         myGLCD.setColor(255, 0, 0); //红

```

```

791     Serial.println("WARNING: HIGH TEMPERATURE !!!");
792 }
793 myGLCD.fillCircle(12, 165, 4); myGLCD.fillRect(11, 165 -
inttemperature, 14, 165);
794 }
795
796
797 #define HUMIDITY_POSITION_X TEMPERATURE_POSITION_X
798 #define HUMIDITY_POSITION_Y TEMPERATURE_POSITION_Y+20
799 void displayHumidity() {
800     float humidity = dht.readHumidity();
801     if ( isnan(humidity) ) {
802         Serial.println("ERROR: READ HUMIDITY!!!");
803         myGLCD.setColor(255, 255, 255); myGLCD.setBackColor(0, 0, 0);
myGLCD.setFont(BVS_13);
804         myGLCD.print("ERROR: READ HUMIDITY!!!", 10, 30, 0);
805     }
806     else {
807         if (writehumlog == true) {
808             String humlog = "Humidity: " + String(humidity) + " %";
809             Serial.println(humlog);
810             Log = SD.open("Log.txt", FILE_WRITE);
811             if (Log) {
812                 Log.println(humlog);
813                 Log.close();
814             }
815             else
816             {
817                 Serial.println("ERROR: OPEN LOG.TXT");
818             }
819             writehumlog = false;
820         }
821         uint8_t inthumidity = int(humidity);
822         graphicalHumidity(inthumidity);
823         myGLCD.setBackColor(0, 0, 0); myGLCD.setFont(BigFont);
824         myGLCD.print(String(inthumidity) + "%", HUMIDITY_POSITION_X,
HUMIDITY_POSITION_Y);
825     }
826 }
827
828 void graphicalHumidity(uint8_t inthumidity) {
829     myGLCD.setColor(150, 150, 150); myGLCD.setBackColor(0, 0, 0);
830     myGLCD.fillCircle(27, 165, 5); myGLCD.fillRoundRect(25, 65, 30,
165);

```

```

831     if (inhumidity <= 45) {
832         myGLCD.setColor(0, 200, 255); //浅蓝
833     }
834     else if (inhumidity > 45 && inhumidity <= 75) {
835         myGLCD.setColor(255, 255, 0); //金黄
836     }
837     else {
838         myGLCD.setColor(255, 0, 0); //红
839     }
840     myGLCD.fillCircle(27, 165, 4); myGLCD.fillRect(26, 165 -
inhumidity, 29, 165);
841 }
842
843
844 void ledTurnOff(uint8_t LED_X) {
845     if (LED_X == LED_RGB) {
846         digitalWrite(LED_RED, LED_OFF);
847         digitalWrite(LED_GREEN, LED_OFF);
848         digitalWrite(LED_BLUE, LED_OFF);
849     }
850     digitalWrite(LED_X, LED_OFF);
851 }
852
853
854 bool checkLedState(uint8_t LED_X) {
855     if (LED_X == LED_RED) {
856         return LED_RED_STATE;
857     }
858     else if (LED_X == LED_GREEN) {
859         return LED_GREEN_STATE;
860     }
861     else if (LED_X == LED_BLUE) {
862         return LED_BLUE_STATE;
863     }
864     else {
865         return LEDSTATE_CHANGEABLE;
866     }
867 }
868
869
870 void ledSolid(uint8_t LED_X, uint8_t delaytime) {
871     if (checkLedState(LED_X) == LEDSTATE_UNCHANGEABLE) {
872         return;
873     }

```

```

874     ledTurnOff(LED_X);
875     digitalWrite(LED_X, LED_ON);
876     if (delaytime == FOREVER) {
877         return;
878     }
879     Blynk_Delay(delaytime);
880     digitalWrite(LED_X, LED_OFF);
881 }
882
883
884 void ledBlink(uint8_t LED_X, uint8_t times, uint8_t delaytime) {
885     if (checkLedState(LED_X) == LEDSTATE_UNCHANGEABLE) {
886         return;
887     }
888     ledTurnOff(LED_X);
889     if (LED_X == LED_RGBLOOP_TYPE1) {
890         for (uint8_t i = 0; i < (times); i++) {
891             digitalWrite(LED_RED, LED_ON);
892             Blynk_Delay(delaytime);
893             digitalWrite(LED_RED, LED_OFF);
894             digitalWrite(LED_GREEN, LED_ON);
895             Blynk_Delay(delaytime);
896             digitalWrite(LED_GREEN, LED_OFF);
897             digitalWrite(LED_BLUE, LED_ON);
898             Blynk_Delay(delaytime);
899             digitalWrite(LED_BLUE, LED_OFF);
900         }
901         return;
902     }
903     if (LED_X == LED_RGBLOOP_TYPE2) {
904         for (uint8_t i = 0; i < (times); i++) {
905             digitalWrite(LED_RED, LED_ON);
906             Blynk_Delay(delaytime);
907             digitalWrite(LED_RED, LED_OFF);
908             Blynk_Delay(delaytime);
909             digitalWrite(LED_GREEN, LED_ON);
910             Blynk_Delay(delaytime);
911             digitalWrite(LED_GREEN, LED_OFF);
912             Blynk_Delay(delaytime);
913             digitalWrite(LED_BLUE, LED_ON);
914             Blynk_Delay(delaytime);
915             digitalWrite(LED_BLUE, LED_OFF);
916             Blynk_Delay(delaytime);
917         }

```

```

918     return;
919 }
920 bool LED_STATE = LED_ON;
921 for (uint8_t i = 0; i < (times * 2); i++) {
922     digitalWrite(LED_X, LED_STATE);
923     LED_STATE = !LED_STATE;
924     Blynk_Delay(delaytime);
925 }
926 }
927
928
929 void Blynk_Delay(uint8_t delaytime) {
930     uint32_t starttime = millis();
931     while (millis() - starttime < delaytime) {
932         if (!TEST) {
933             Blynk.run();
934             timer.run();
935         }
936     }
937 }

```

2.2. NanoCode_v0.1

```

1  /*****
2  RGB LED:
3  R - D2
4  G - D3
5  B - D4
6  GND - GND
7
8  MEGA:
9  TX2 - D5
10 RX2 - D6
11 GND - GND
12
13 TOUCH:
14 VCC - 5V
15 OUT - D8
16 GND - GND
17
18
19 MFRC522: https://github.com/miguelbalboa/rfid
20 SDA - D10
21 SCK - D13
22 MOSI - D11

```

```

23     MISO - D12
24     IRQ -
25     GND - GND
26     RST - D9
27     3.3V - 3.3V
28
29     *****/
30
31
32
33     #define LED_RED 2
34     #define LED_GREEN 3
35     #define LED_BLUE 4
36     #define LED_RGB 0
37     #define LED_RGBLOOP_TYPE1 -1
38     #define LED_RGBLOOP_TYPE2 -2
39     #define LED_ON HIGH
40     #define LED_OFF LOW
41     #define SHORTTIME 300
42     #define LONGTIME 1000
43
44     #define SOFTRXPIN 5
45     #define SOFTTXPIN 6
46
47     #define TOUCHPIN 8
48
49     #define MFRC522_RST_PIN 9
50     #define MFRC522_SS_PIN 10
51
52
53
54     #include <SPI.h>
55     #include <MFRC522.h> // https://github.com/miguelbalboa/rfid
56     MFRC522 mfrc522(MFRC522_SS_PIN, MFRC522_RST_PIN);
57     #include <SoftwareSerial.h>
58     SoftwareSerial mySerial(SOFTRXPIN, SOFTTXPIN);
59     #include <EEPROM.h>
60
61
62
63     bool programMode = false;
64     uint8_t successRead; // Variable integer to keep if we have
        Successful Read from Reader
65     byte readCard[4]; // Stores scanned ID read from RFID Module

```



```

66 byte masterCard[4]; // Stores master card's ID read from EEPROM
67 byte storedCard[4]; // Stores an ID read from EEPROM
68
69
70
71 void setup() {
72     Serial.begin(115200);
73     Serial.println("==>setup-----");
74
75     SPI.begin();
76     mfrc522.PCD_Init();
77     ShowReaderDetails();
78
79     mySerial.begin(9600);
80
81     pinMode(LED_RED, OUTPUT);
82     pinMode(LED_GREEN, OUTPUT);
83     pinMode(LED_BLUE, OUTPUT);
84
85     pinMode(TOUCHPIN, INPUT);
86     touchToDoSomething();
87
88     if (EEPROM.read(1) != 233) {
89         Serial.println(F("No Admin"));
90         Serial.println(F("Scan a PICC to define as MasterCard"));
91         do {
92             successRead = getID(); // sets successRead to 1 when
we get read from reader otherwise 0
93             ledBlink(LED_BLUE, 2, SHORTTIME);
94         }
95         while (!successRead); // Program will not go
further while you not get a successful read
96         for ( uint8_t j = 0; j < 4; j++ ) { // Loop 4 times
97             EEPROM.write( 2 + j, readCard[j] ); // Write scanned PICC's
UID to EEPROM, start from address 3
98         }
99         EEPROM.write(1, 233); // Write to EEPROM we
defined Master Card.
100         Serial.println(F("MasterCard defined"));
101     }
102
103     Serial.println(F("-----"));
104     Serial.println(F("MasterCard's UID"));
105     for ( uint8_t i = 0; i < 4; i++ ) { // Read Master Card's

```

```

106     UID from EEPROM
107     masterCard[i] = EEPROM.read(2 + i);    // Write it to masterCard
108     Serial.print(masterCard[i], HEX);
109 }
110 Serial.println("==>setup-----Done!");
111 }
112
113
114 void loop () {
115
116     do {
117         successRead = getID(); // sets successRead to 1 when we get read
118         from reader otherwise 0
119         if (digitalRead(TOUCHPIN) == HIGH) { // Check if button is
120         pressed
121             touchToDoSomething();
122         }
123         if (programMode) {
124             ledTurnOn(LED_BLUE);
125         }
126         else {
127             ledTurnOff(LED_RED);
128         }
129     } while (!successRead); //the program will not go further while
130     you are not getting a successful read
131
132     if (programMode) {
133         if ( isMaster(readCard) ) { //When in program mode check First
134         If master card scanned again to exit program mode
135             ledTurnOff(LED_RGB);
136             Serial.println(F("MasterCard scanned"));
137             Serial.println(F("Exiting ProgramMode"));
138             Serial.println(F("-----"));
139             programMode = false;
140             return;
141         }
142         else {
143             if ( findID(readCard) ) { // If scanned card is known delete
144             it
145                 Serial.println(F("I know this PICC, removing..."));
146                 delateUser(readCard);
147                 hintScan();
148             }
149         }
150     }

```

```

144         else {                                     // If scanned card is not known add it
145             Serial.println(F("I do not know this PICC, adding..."));
146             addUser(readCard);
147             hintScan();
148         }
149     }
150 }
151 else {
152     if ( isMaster(readCard)) {    // If scanned card's ID matches
Master Card's ID - enter program mode
153         programMode = true;
154         Serial.println(F("Hello master - Entered ProgramMode"));
155         uint8_t count = EEPROM.read(0);    // Read the first Byte of
EEPROM that
156         Serial.print(F("I have "));    // stores the number of ID's
in EEPROM
157         Serial.print(count);
158         Serial.print(F(" record(s) on EEPROM"));
159         Serial.println("");
160         hintScan();
161         Serial.println(F("Scan MasterCard again to exit
ProgramMode"));
162         Serial.println(F("-----"));
163     }
164     else {
165         if ( findID(readCard) ) { // If not, see if the card is in the
EEPROM
166             Serial.println(F("Welcome, You shall pass"));
167             mySerial.print("Welcome");
168             ledBlink(LED_GREEN, 1, LONGTIME);
169         }
170         else {    // If not, show that the ID was not valid
171             Serial.println(F("Stranger, You shall not pass"));
172             mySerial.print("Stranger");
173             ledBlink(LED_RED, 1, LONGTIME);
174         }
175     }
176 }
177 }
178
179
180
181 void hintScan() {
182     Serial.println(F("-----"));

```

```

183     Serial.println(F("Scan a PICC to add or remove"));
184 }
185
186
187 void ledTurnOn(uint8_t LED_X) {
188     if (LED_X == LED_RGB) {
189         digitalWrite(LED_RED, LED_ON);
190         digitalWrite(LED_GREEN, LED_ON);
191         digitalWrite(LED_BLUE, LED_ON);
192     }
193     digitalWrite(LED_X, LED_ON);
194 }
195
196
197 void ledTurnOff(uint8_t LED_X) {
198     if (LED_X == LED_RGB) {
199         digitalWrite(LED_RED, LED_OFF);
200         digitalWrite(LED_GREEN, LED_OFF);
201         digitalWrite(LED_BLUE, LED_OFF);
202     }
203     digitalWrite(LED_X, LED_OFF);
204 }
205
206
207 void ledBlink(uint8_t LED_X, uint8_t times, uint8_t delaytime) {
208     ledTurnOff(LED_X);
209     if (LED_X == LED_RGBLOOP_TYPE1) {
210         for (uint8_t i = 0; i < (times); i++) {
211             digitalWrite(LED_RED, LED_ON);
212             delay(delaytime);
213             digitalWrite(LED_RED, LED_OFF);
214             digitalWrite(LED_GREEN, LED_ON);
215             delay(delaytime);
216             digitalWrite(LED_GREEN, LED_OFF);
217             digitalWrite(LED_BLUE, LED_ON);
218             delay(delaytime);
219             digitalWrite(LED_BLUE, LED_OFF);
220         }
221         return;
222     }
223     if (LED_X == LED_RGBLOOP_TYPE2) {
224         for (uint8_t i = 0; i < (times); i++) {
225             digitalWrite(LED_RED, LED_ON);
226             delay(delaytime);

```

```

227     digitalWrite(LED_RED, LED_OFF);
228     delay(delaytime);
229     digitalWrite(LED_GREEN, LED_ON);
230     delay(delaytime);
231     digitalWrite(LED_GREEN, LED_OFF);
232     delay(delaytime);
233     digitalWrite(LED_BLUE, LED_ON);
234     delay(delaytime);
235     digitalWrite(LED_BLUE, LED_OFF);
236     delay(delaytime);
237 }
238 return;
239 }
240 bool LED_STATE = LED_ON;
241 for (uint8_t i = 0; i < (times * 2); i++) {
242     digitalWrite(LED_X, LED_STATE);
243     LED_STATE = !LED_STATE;
244     delay(delaytime);
245 }
246 }
247
248
249 void ShowReaderDetails() {
250     // Get the MFRC522 software version
251     byte v = mfrc522.PCD_ReadRegister(mfrc522.VersionReg);
252     Serial.print(F("MFRC522 Software Version: 0x"));
253     Serial.print(v, HEX);
254     if (v == 0x91)
255         Serial.print(F(" = v1.0"));
256     else if (v == 0x92)
257         Serial.print(F(" = v2.0"));
258     else
259         Serial.print(F(" (unknown),probably a chinese clone?"));
260     Serial.println("");
261     // When 0x00 or 0xFF is returned, communication probably failed
262     if ((v == 0x00) || (v == 0xFF)) {
263         Serial.println(F("WARNING: Communication failure, is the
MFRC522 properly connected?"));
264         Serial.println(F("SYSTEM HALTED: Check connections."));
265         // Visualize system is halted
266         ledTurnOn(LED_RED);
267         while (true); // do not go further
268     }
269 }

```

```

270
271
272 void touchToDoSomething() {
273     ledTurnOn(LED_RED);
274     if (digitalRead(TOUCHPIN) == HIGH) {
275         bool buttonState = monitorTOUCHPIN(5000);
276         if (buttonState == true && digitalRead(TOUCHPIN) == HIGH) {
277             hintGodMode();
278             while (1) {
279                 if (Serial.available()) {
280                     String str = Serial.readString();
281                     if (str == "A" || str == "a") {
282                         Serial.println("Delate all users");
283                         wipeEeprom();
284                     }
285                     else if (str == "B" || str == "b") {
286                         Serial.println("Delate admin only");
287                         delateAdmin();
288                     }
289                     else {
290                         Serial.println("Do nothing, just quit");
291                         ledTurnOff(LED_RED);
292                         return;
293                     }
294                 }
295             }
296         }
297     }
298     else {
299         Serial.println("Not touched, do nothing, go on");
300         ledTurnOff(LED_RED);
301     }
302 }
303
304
305 void hintGodMode() {
306     ledTurnOn(LED_RED);
307     Serial.println("");
308     Serial.println("God mode");
309     Serial.println("Input 'A' to delate all users");
310     Serial.println("Input 'B' to delate admin only");
311     Serial.println("Input others to do nothing, just quit");
312 }
313

```

```

314
315 void wipeEeprom() {
316     ledTurnOn(LED_RED);
317     Serial.println(F("You have 3 seconds to cancel"));
318     Serial.println(F("This will delate all users and cannot be
undone"));
319     bool buttonState = monitorTOUCHPIN(3000);
320     if (buttonState == true && digitalRead(TOUCHPIN) == HIGH) {    //
If button still be pressed, wipe EEPROM
321         Serial.println(F("Starting wiping EEPROM"));
322         for (uint16_t x = 0; x < EEPROM.length(); x = x + 1) {    //Loop
end of EEPROM address
323             if (EEPROM.read(x) == 0) {                            //If EEPROM address 0
324                 // do nothing, already clear, go to the next address in order
to save time and reduce writes to EEPROM
325             }
326             else {
327                 EEPROM.write(x, 0);    // if not write 0 to clear, it takes
3.3mS
328             }
329         }
330         Serial.println(F("EEPROM successfully wiped"));
331         Serial.println(F("Please restart to add new MasterCard"));
332         while (1);
333     }
334     else {
335         Serial.println(F("Wiping cancelled")); // Show some feedback
that the wipe button did not pressed for 15 seconds
336         ledTurnOff(LED_RED);
337     }
338     hintGodMode();
339 }
340
341
342 void delateAdmin() {
343     ledTurnOn(LED_RED);
344     Serial.println(F("You have 3 seconds to cancel"));
345     Serial.println(F("This will delate admin and cannot be undone"));
346     bool buttonState = monitorTOUCHPIN(3000);
347     if (buttonState == true && digitalRead(TOUCHPIN) == HIGH) {    //
If button still be pressed, wipe EEPROM
348         EEPROM.write(1, 0);
349         Serial.println(F("MasterCard erased from device"));
350         Serial.println(F("Please restart to add new MasterCard"));

```

```

351     while (1);
352 }
353 else {
354     Serial.println(F("MasterCard erase cancelled")); // Show some
// feedback that the wipe button did not pressed for 15 seconds
355     ledTurnOff(LED_RED);
356 }
357 hintGodMode();
358 }
359
360
361 uint8_t getID() {
362     // Getting ready for Reading PICCs
363     if ( ! mfrc522.PICC_IsNewCardPresent()) { //If a new PICC placed
to RFID reader continue
364         return 0;
365     }
366     if ( ! mfrc522.PICC_ReadCardSerial()) { //Since a PICC placed
get Serial and continue
367         return 0;
368     }
369     // There are Mifare PICCs which have 4 byte or 7 byte UID care
if you use 7 byte PICC
370     // I think we should assume every PICC as they have 4 byte UID
371     // Until we support 7 byte PICCs
372     Serial.println(F("Scanned PICC's UID:"));
373     for ( uint8_t i = 0; i < 4; i++) { //
374         readCard[i] = mfrc522.uid.uidByte[i];
375         Serial.print(readCard[i], HEX);
376     }
377     Serial.println("");
378     mfrc522.PICC_HaltA(); // Stop reading
379     return 1;
380 }
381
382
383 void readID( uint8_t number ) {
384     uint8_t start = (number * 4) + 2; // Figure out starting position
385     for ( uint8_t i = 0; i < 4; i++ ) { // Loop 4 times to get the
4 Bytes
386         storedCard[i] = EEPROM.read(start + i); // Assign values read
from EEPROM to array
387     }
388 }

```



```

389
390
391 void addUser( byte a[] ) {
392     if ( !findID( a ) ) { // Before we write to the EEPROM, check
                             to see if we have seen this card before!
393         uint8_t num = EEPROM.read(0); // Get the number of used spaces,
                             position 0 stores the number of ID cards
394         uint8_t start = ( num * 4 ) + 6; // Figure out where the next
                             slot starts
395         num++; // Increment the counter by one
396         EEPROM.write( 0, num ); // Write the new count to the counter
397         for ( uint8_t j = 0; j < 4; j++ ) { // Loop 4 times
398             EEPROM.write( start + j, a[j] ); // Write the array values
                             to EEPROM in the right position
399         }
400         ledBlink(LED_GREEN, 3, SHORTTIME);
401         Serial.println(F("Succesfully added ID record to EEPROM"));
402     }
403     else {
404         ledBlink(LED_RED, 3, LONGTIME);
405         Serial.println(F("Failed! There is something wrong with ID or
                             bad EEPROM"));
406     }
407 }
408
409
410 void delateUser( byte a[] ) {
411     if ( !findID( a ) ) { // Before we delete from the EEPROM, check
                             to see if we have this card!
412         ledBlink(LED_RED, 3, LONGTIME); // If not
413         Serial.println(F("Failed! There is something wrong with ID or
                             bad EEPROM"));
414     }
415     else {
416         uint8_t num = EEPROM.read(0); // Get the number of used spaces,
                             position 0 stores the number of ID cards
417         uint8_t slot; // Figure out the slot number of the card
418         uint8_t start; // = ( num * 4 ) + 6; // Figure out where the
                             next slot starts
419         uint8_t looping; // The number of times the loop repeats
420         uint8_t j;
421         uint8_t count = EEPROM.read(0); // Read the first Byte of EEPROM
                             that stores number of cards
422         slot = findIDSLOT( a ); // Figure out the slot number of the

```

```

card to delete
423     start = (slot * 4) + 2;
424     looping = ((num - slot) * 4);
425     num--;      // Decrement the counter by one
426     EEPROM.write( 0, num ); // Write the new count to the counter
427     for ( j = 0; j < looping; j++ ) {      // Loop the card shift
times
428         EEPROM.write( start+j, EEPROM.read(start+4+j)); // Shift
the array values to 4 places earlier in the EEPROM
429     }
430     for ( uint8_t k = 0; k < 4; k++ ) {      // Shifting loop
431         EEPROM.write( start + j + k, 0);
432     }
433     ledBlink(LED_RED, 3, SHORTTIME);
434     Serial.println(F("Succesfully delated ID record from
EEPROM"));
435 }
436 }
437
438
439 bool checkTwo ( byte a[], byte b[] ) {
440     for ( uint8_t k = 0; k < 4; k++ ) { // Loop 4 times
441         if ( a[k] != b[k] ) { // IF a != b then false, because: one
fails, all fail
442             return false;
443         }
444     }
445     return true;
446 }
447
448
449 uint8_t findIDSLOT( byte find[] ) {
450     uint8_t count = EEPROM.read(0); // Read the first Byte of
EEPROM that
451     for ( uint8_t i = 1; i <= count; i++ ) { // Loop once for each
EEPROM entry
452         readID(i); // Read an ID from EEPROM, it is stored
in storedCard[4]
453         if ( checkTwo( find, storedCard ) ) { // Check to see if the
storedCard read from EEPROM
454             // is the same as the find[] ID card passed
455             return i; // The slot number of the card
456         }
457     }

```

```

458 }
459
460
461 bool findID( byte find[] ) {
462     uint8_t count = EEPROM.read(0);    // Read the first Byte of
EEPROM that
463     for ( uint8_t i = 1; i < count; i++ ) {    // Loop once for each
EEPROM entry
464         readID(i);        // Read an ID from EEPROM, it is stored in
storedCard[4]
465         if ( checkTwo( find, storedCard ) ) {    // Check to see if the
storedCard read from EEPROM
466             return true;
467         }
468         else {    // If not, return false
469         }
470     }
471     return false;
472 }
473
474
475 bool isMaster( byte test[] ) {
476     return checkTwo(test, masterCard);
477 }
478
479
480 bool monitorTOUCHPIN(uint32_t interval) {
481     uint32_t now = (uint32_t)millis();
482     while ((uint32_t)millis() - now < interval) {
483         // check on every half a second
484         if (((uint32_t)millis() % 500) == 0) {
485             if (digitalRead(TOUCHPIN) != HIGH)
486                 return false;
487         }
488     }
489     return true;
490 }

```

