

随着智能化的发展，智能门禁系统渐渐的出现同学们的事业中，尤其是可以支持多种解锁方式的，比如指纹、手机、门禁卡、按键密码、钥匙解锁功能等等，今天分享的这个毕设项目正是包含了这些。

智能门锁实现功能

- 可通过指纹模块增删更改家庭成员的指纹信息，增删更改是否成功的相关信息显示在OLED屏幕上
- 在指纹匹配过程中，如果采集的指纹与指纹模块库相匹配，OLED显示匹配成功，并转动步进电机一圈
- 可通过按键设定智能门锁密码，密码可设置为两个（密码六位），如果匹配两个中的一个成功，即可开锁，也可通过按键修改密码，所有的操作过程显示于OLED中
- 实现RFID与手机解锁（蓝牙解锁）

硬件准备

整个项目所需的硬件并不复杂，也非常容易配备：

- 1、步进电机(带驱动模块)，型号为：28BYJ4+ULN2003



- 2、自制4X4按键片



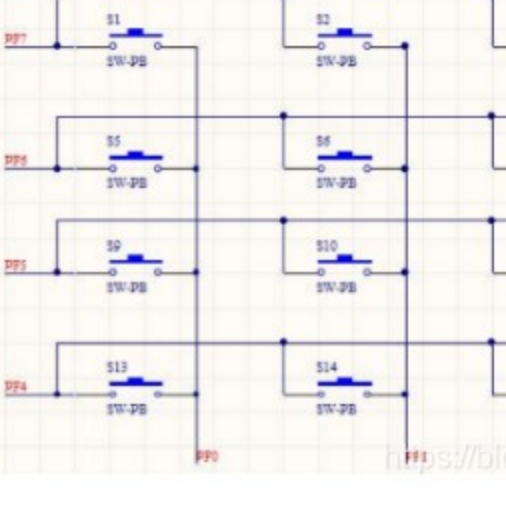
- 3、指纹模块AS608



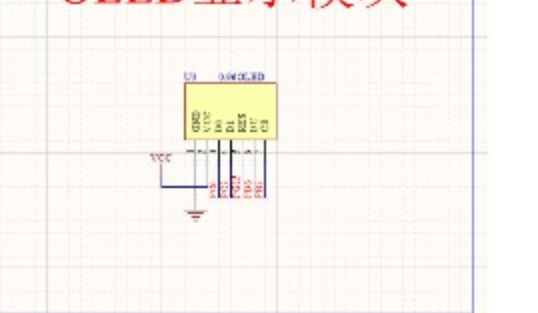
- 4、0.96寸OLED显示屏（IIC），型号：7针



- 5、蓝牙模块HC-05



- 6、主控STM32F407ZGT6最小系统板



- 7、RFID RC522 射频模块



再准备一些杜邦线就OK。

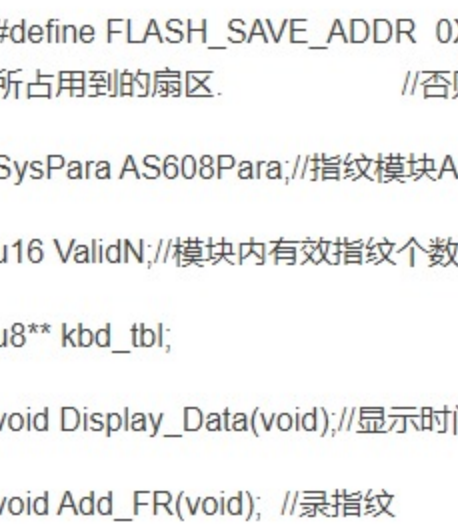
软、硬件设计

一）硬件设计

由于使用的是STM32最小核心板，所以可以自由添加外设。其中4x4按键分别使用了MCU的8个IO口：



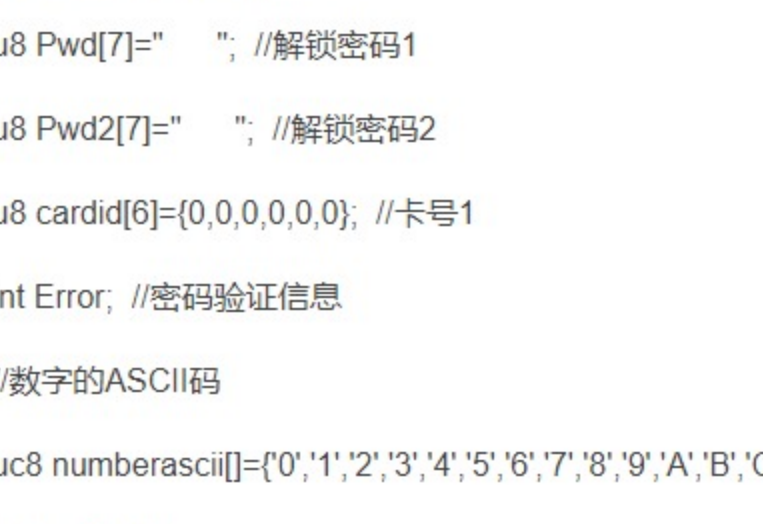
OLED显示模块及射频模块接口：



指纹和蓝牙模块接口为：



根据原理图连接好硬件电路：



二）程序设计

```
#define usart2_baud 57600//串口2波特率，根据指纹模块波特率更改
//要写入到STM32 FLASH的字符串数组
const u8 TEXT_Buffer[]={0x17,0x23,0x6f,0x60,0,0,0};
#define TEXT_LENGTH sizeof(TEXT_Buffer) //数组长度
#define SIZE TEXT_LENGTH/4+(TEXT_LENGTH%4)?1:0
#define FLASH_SAVE_ADDR 0X0802C124 //设置FLASH 保存地址(必须为偶数，且所在扇区 要大于本代码所占用到的扇区。 //否则，写操作的时候，可能会导致擦除整个扇区，从而引起部分程序丢失 引起死机
SysPara AS608Para;//指纹模块AS608参数
u16 ValidN;//模块内有效指纹个数
u8** kbd_tbl;
void Display_Data(void);//显示时间
void Add_FR(void); //录指纹
void Del_FR(void); //删除指纹
int press_FR(void);//刷指纹
void ShowErrorMessage(u8 ensure);//显示确认错误码信息
int password(void);//密码锁
void SetPassworld(void);//修改密码
void starting(void);//开机界面信息
u8 MFRC522_lock(void);//刷卡解锁
u8 Add_Rfid(void); //录入
void Set_Time(void);
void Massige(void);
u8 Pwd[7]=" ", //解锁密码1
u8 Pwd2[7]=" ", //解锁密码2
u8 cardid[6]={0,0,0,0,0,0}; //卡号1
int Error; //密码验证信息
//数字的ASCII码
u8 numberascii[]={"0","1","2","3","4","5","6","7","8","9","A","B","C","D","E","F"};
//显示缓冲区
u8 dispnumber5buf[6];
u8 dispnumber3buf[4];
u8 dispnumber2buf[3];
//MFRC522数据区
u8 mfrc522pidbuf[18];
u8 card_pydebuf[2];
u8 card_numberbuf[5];
u8 card_key0Abuf[6]={0xff,0xff,0xff,0xff,0xff,0xff};
u8 card_writebuf[16]={0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15};
u8 card_readbuf[18];
//SM05-S数据区
u8 sm05cmdbuf[15]={14,128,0,22,5,0,0,0,4,1,157,16,0,0,21};
//extern声明变量已在外部的C文件里定义，可以在主文件中使用
extern u8 sm05receivebuf[16]; //在中断C文件里定义
extern u8 sm05_OK; //在中断C文件里定义
u8 * week[7]={"Mon","Tue","Wed","Thu","Fri","Sat","Sun"};
u8 * setup[7]={"1、录入指纹","2、删除指纹","3、修改密码","4、修改时间","5、录入卡片","6、查看信息"};
int main(void)
{
    u16 set0;
    int key_num;
    int time1;
    int time2; //锁屏时间
    char arrow=0; //箭头位子
    NVIC_PriorityGroupConfig(NVIC_PriorityGroup_2);//设置系统中断优先级分组2
    My_RTC_Init(); //初始化RTC
    RTC_Set_WakeUp(RTC_WakeUpClock_CK_SPRE_16bits,0); //配置WAKE UP中断,1秒钟中断一次
    delay_init(168); //初始化延时函数
    uart_init(115200); //初始化串口1波特率为115200，用于支持USART
    uart3_init(9600);// 蓝牙初始化
    usart2_init(usart2_baud);//初始化串口2,用于与指纹模块通讯
    PS_StaGPIO_Init(); //初始化FR读状态引脚
    LED_Init(); //初始化LED
    // BEEP_Init(); //beep初始化
    Button4_4_Init();//矩阵按键初始化
    OLED_Init(); //oled初始化
    W25QXX_Init(); //初始化W25Q128
    Walkmotor_Init(); //步进电机初始化
    my_mem_init(SRAMIN); //初始化内部内存池
    my_mem_init(SRAMCCM); //初始化CCM内存池
    exfuncs_init(); //为fatfs相关变量申请内存
    f_mount(fs[0],"0",1); //挂载SD卡
    f_mount(fs[1],"1",1); //挂载FLASH.
    starting();//开机信息 logo
    STMFLASH_Read(0X08020004,(u32*)Pwd,2); //读取密码1
    STMFLASH_Read(0X08090004,(u32*)Pwd2,2); //读取密码2
    STMFLASH_Read(0X080F0004,(u32*)cardid,1); //读取卡号1
    上机调试
    把硬件连接好之后还需要做一些准备工作：
    1) 录入指纹：
    1) 程序下载之前需要给门禁一个解锁方式，门锁一共有4种解锁方式：指纹、蓝牙、卡片、密码。其中卡片需要录入过系统的才可以解锁，密码一开始保存在STM32内部Flash中，内部Flash做EEPROM需要先初始化，解锁后在设置中更改密码或者在代码中进行初始化密码，不然密码是乱码。
    2) 串口录入指纹：将串口转TTL CH340模块连接好指纹模块，用配套上位机录入自己的一个指纹。详细连接方法参考指纹模块资料
    3) 解锁后就可以进行设置密码和录入卡片等操作
    2) 移植汉字库：（如果没有SD卡接口可以自己汉字取模显示）
    1) 准备一个sd卡，一个读卡器，用来存储汉字库，显示到OLED
    2) 找到开发资料下的SYSTEM文件夹
    3) 把SYSTEM文件夹复制到SD卡根目录下（一定要是根目录下）
    4) 复制好之后把SD卡插到开发板
    5) 下载程序到开发板启动，如果可以显示汉字说已经移植成功。（注意：开始的化作尘&指纹锁界面是显示的图片，并不是汉字体）
    遇到的问题及解决办法
    问题一：如果进去只显示“化作尘指纹锁”然后黑屏，那么是字库的问题；
    问题二：如果不知道密码进不去主界面，那么打开代码，将密码打印出来，然后就可以用密码解锁了，一共两个密码，任意一个都可以解锁：图片因为没有考虑到字符不全为“0”-“9”的关系，密码打印出来不是数字，所以查看密码不成功。正确打开方式：
    步骤一：直接给密码赋值操作，然后编译下载，下载完成后输入密码“123456”是肯定可以进主界面的，然后点击修改密码，密码修改为你想要的密码“000111”，这个密码会保存到内部eeprom，密码初始化就完成了。
    步骤二：把刚刚写的代码注释，重新下载进去，代码就可以正常运行了，输入刚刚设置的密码“000111”就可以解锁。
    知识点扫描
    知识点一：汉字的显示
    （具体代码参考正点原子“汉字显示实验”）
    1) 汉字的显示原理：学习过屏幕驱动的都该知道，屏幕的显示无非就是在屏幕上打相应颜色的点，比如要显示一个字符‘A’那么我只需要知道‘A’相应的点阵数据就可以把它显示出来，汉字也一样。
    2) 汉字库：汉字库文件在正点原子的开发板目录->“SD卡根目录文件”中，可以看到正点原子官网论坛下载。
    汉字库有两类文件，.FON与.BIN文件，.FON代表汉字库点阵文件，.BIN表示的是编码表，与点阵一一对应，比如GBK编码的第一个字就对应着点阵里的第一个点阵数据，映射关系如下：
    UNIGBK->GBK12.FON
    UNIGBK->GBK24.FON
    UNIGBK->GBK32.FON
    有兴趣可以百度一下查看GBK编码表。
    知识点二：把汉字库里的点阵数据读取出来显示到MCU屏
    数据编码格式的问题：有关显示汉字就必然会涉及编码格式的问题，不同的编码格式对应的汉字编码是不同的，比如UTF-8编码的文件用GB2312编码打开就会乱码。
    同理，比如文档的格式是UTF-8编码，但是我的字库是GBK编码，读出来的数据直接显示到屏幕就必然会乱码。所以显示汉字如果乱码首先需要检查的就是编辑器的编码，保证编码一致。
    118 //0X08020004 密码1
    119 //0X08090004 密码2
    120 //0X080F0004 卡号1
    121 //0X080F0004 卡号1
    122 STMFLASH_Write(FLASH_SAVE_ADDR,(u32*)TEXT_Buffer,SIZE); //写入密码1
    123 STMFLASH_Read(0X08020004,(u32*)Pwd,2); //读取密码1
    124 STMFLASH_Read(0X08090004,(u32*)Pwd2,2); //读取密码2
    125 STMFLASH_Read(0X080F0004,(u32*)cardid,1); //读取卡号1
    126
    127 // Pwd[7] = "\0";
    128 // Pwd2[7] = "\0";
    129 // printf("Pwd=%s",Pwd);
    130 // printf("Pwd2=%s",Pwd2);
    131 // printf("cardid=%X,%X,%X,%X\n",cardid[0],cardid[1],cardid[2],cardid[3]);
    132 while(1)
    133 }
```

移植汉字库：（如果没有SD卡接口可以自己汉字取模显示）

- 1) 准备一个sd卡，一个读卡器，用来存储汉字库，显示到OLED
- 2) 找到开发资料下的SYSTEM文件夹



- 3) 把SYSTEM文件夹复制到SD卡根目录下（一定要是根目录下）



- 4) 复制好之后把SD卡插到开发板

- 5) 下载程序到开发板启动，如果可以显示汉字说已经移植成功。（注意：开始的化作尘&指纹锁界面是显示的图片，并不是汉字字体）



遇到的问题及解决办法

问题一：如果进去只显示“化作尘指纹锁”然后黑屏，那么是字库的问题；

问题二：如果不知道密码进不去主界面，那么打开代码，将密码打印出来，然后就可以用密码解锁了，一共两个密码，任意一个都可以解锁：图片因为没有考虑到字符不全为“0”-“9”的关系，密码打印出来不是数字，所以查看密码不成功。正确打开方式：

```
118 //0X08020004 密码1
119 //0X08090004 密码2
120 //0X080F0004 卡号1
121 //0X080F0004 卡号1
122 STMFLASH_Write(FLASH_SAVE_ADDR,(u32*)TEXT_Buffer,SIZE); //写入密码1
123 STMFLASH_Read(0X08020004,(u32*)Pwd,2); //读取密码1
124 STMFLASH_Read(0X08090004,(u32*)Pwd2,2); //读取密码2
125 STMFLASH_Read(0X080F0004,(u32*)cardid,1); //读取卡号1
126
127 // Pwd[7] = "\0";
128 // Pwd2[7] = "\0";
129 // printf("Pwd=%s",Pwd);
130 // printf("Pwd2=%s",Pwd2);
131 // printf("cardid=%X,%X,%X,%X\n",cardid[0],cardid[1],cardid[2],cardid[3]);
132 while(1)
133 }
```

步骤一：直接给密码赋值操作，然后编译下载，下载完成后输入密码“123456”是肯定可以进主界面的，然后点击修改密码，密码修改为你想要的密码“000111”，这个密码会保存到内部eeprom，密码初始化就完成了。

```
121 //0X080F0004 卡号1
122 // STMFLASH_Write(FLASH_SAVE_ADDR,(u32*)TEXT_Buffer,SIZE);
123 STMFLASH_Read(0X08020004,(u32*)Pwd,2); //读取密码1
124 STMFLASH_Read(0X08090004,(u32*)Pwd2,2); //读取密码2
125 STMFLASH_Read(0X080F0004,(u32*)cardid,1); //读取卡号1
126
127 // strcpy(Pwd, "123456");
128 // strcpy(Pwd2, "123456");
129 // printf("Pwd=%s",Pwd);
130 // printf("Pwd2=%s",Pwd2);
131 // printf("cardid=%X,%X,%X,%X\n",cardid[0],cardid[1],cardid[2],cardid[3]);
132 while(1)
133 }
```

步骤二：把刚刚写的代码注释，重新下载进去，代码就可以正常运行了，输入刚刚设置的密码“000111”就可以解锁。

```
122 // STMFLASH_Write(FLASH_SAVE_ADDR,(u32*)TEXT_Buffer,SIZE);
123 STMFLASH_Read(0X08020004,(u32*)Pwd,2); //读取密码1
124 STMFLASH_Read(0X08090004,(u32*)Pwd2,2); //读取密码2
125 STMFLASH_Read(0X080F0004,(u32*)cardid,1); //读取卡号1
126
127 // strcpy(Pwd, "123456");
128 // strcpy(Pwd2, "123456");
129 // printf("Pwd=%s",Pwd);
130 // printf("Pwd2=%s",Pwd2);
131 // printf("cardid=%X,%X,%X,%X\n",cardid[0],cardid[1],cardid[2],cardid[3]);
132 while(1)
133 }
```

知识点扫描

知识点一：汉字的显示

（具体代码参考正点原子“汉字显示实验”）

- 1) 汉字的显示原理：学习过屏幕驱动的都该知道，屏幕的显示无非就是在屏幕上打相应颜色的点，比如要显示一个字符‘A’那么我只需要知道‘A’相应的点阵数据就可以把它显示出来，汉字也一样。
- 2) 汉字库：汉字库文件在正点原子的开发板目录->“SD卡根目录文件”中，可以看到正点原子官网论坛下载。

汉字库有两类文件，.FON与.BIN文件，.FON代表汉字库点阵文件，.BIN表示的是编码表，与点阵一一对应，比如GBK编码的第一个字就对应着点阵里的第一个点阵数据，映射关系如下：

UNIGBK->GBK12.FON

UNIGBK->GBK24.FON

UNIGBK->GBK32.FON

有兴趣可以百度一下查看GBK编码表。

知识点二：把汉字库里的点阵数据读取出来显示到MCU屏

数据编码格式的问题：有关显示汉字就必然会涉及编码格式的问题，不同的编码格式对应的汉字编码是不同的，比如UTF-8编码的文件用GB2312编码打开就会乱码。

同理，比如文档的格式是UTF-8编码，但是我的字库是GBK编码，读出来的数据直接显示到屏幕就必然会乱码。所以显示汉字如果乱码首先需要检查的就是编辑器的编码，保证编码一致。

