



浙江工业大学

《单片机原理与实践》 实验报告书

综合设计实验

实验题目：基于 STM32F407 的智能手机系统

姓 名： 朱顺禄

学 号： 202105100415

指导老师： 欧县华

专业班级： 自动化 2104 班

实验地点： 信息楼 B408

实验日期： 2024 年 1 月 6 日

摘 要

本实验旨在通过 STM32F407 芯片实现一个功能强大的智能手机系统。系统涵盖了触摸屏、密码锁、相册、实时时钟、GPS 获取、拨号盘、温湿度家居控制、电机模拟、手电筒、音乐播放器等多项功能。通过硬件连接和软件设计的综合实践，展示 STM32F407 在嵌入式系统中的广泛应用。

目 录

一、 引言	3
二、 系统功能设计	3
2.1 整体系统功能流程介绍	3
2.2 各部分功能介绍	4
2.2.1 密码锁	4
2.2.2 主菜单交互以及返回功能	7
2.2.3 相册	7
2.2.4 实时时钟	8
2.2.5 GPS 定位	9
2.2.6 拨号盘	10
2.2.7 温湿度家居报警系统	11
2.2.8 MP3 音乐播放器	12
2.2.9 手电筒	13
2.2.10 电机模拟	14
三、 硬件介绍	15
3.1 STM32F407 实验板	15
3.2 各部分外设模块介绍	15
3.2.1 GPS+北斗模块	15
3.2.2 MP3 模块	16
3.2.3 数码管矩阵键盘模块	17
3.2.4 直流电机模块	18
四、 软件设计与程序实现	18
五、 总结	39

一、引言

智能手机的广泛应用催生了嵌入式系统技术的迅猛发展。STM32F407 芯片以其强大的性能和丰富的外设，成为嵌入式系统设计的理想选择。本实验通过设计一个多功能的智能手机系统，旨在全面展示 STM32F407 芯片的强大性能和多样化应用。

二、系统功能设计

2.1 整体系统功能流程介绍

本智能手机系统的功能流程为：开机需要输入正确的锁屏密码方能进入到后续的主菜单交互页面，成功进入后可进行交互功能，在主菜单界面有八个功能可供交互，用户可依据自己的需求或喜好进行任意功能交互，交互功能实现为触摸识别，即用户只需点击菜单对应功能即可，当用户选择某一功能后，系统将对应进入到对应功能的下一交互阶段，此时用户可根据需求进行操作，如若用户需要退出，仅需点击左上角返回按键即可退出当前功能，回到主菜单，此时用户可再次进入其他交互菜单内，余下操作同理。具体功能流程图如下图 2-1。

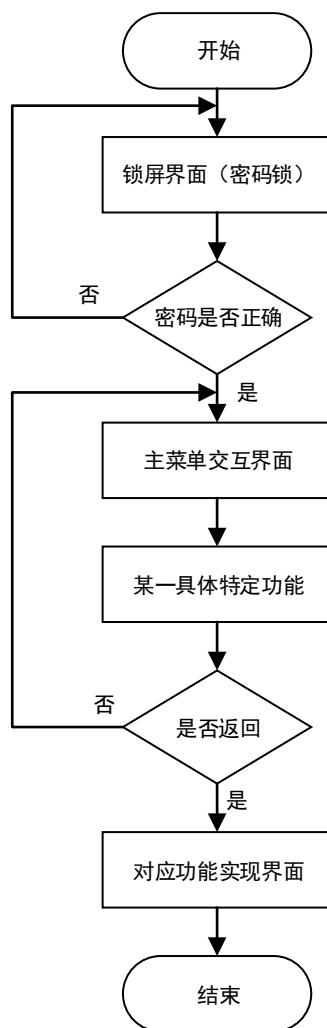


图 2-1 整体系统功能流程图

2.2 各部分功能介绍

本次实验设计功能包括，触摸屏密码锁和主菜单中包括的八个功能：相册、实时时钟、GPS 定位获取、拨号盘模拟、温湿家居报警系统、MP3 音乐功能、手电筒以及电机模拟功能，下面将依次介绍。

2.2.1 密码锁

密码锁功能主要由 UI 界面、密码输入以及密码判断三部分组成。下面将对这三个方面进行简要分析。

UI 界面主要模拟一般手机的数字密码锁功能，在主界面有 0-9 共 10 个数字对应 10 个数字密码单元，在数字输入按键上方有实时反应按键按下情况的四个输入小圆，在数字密码锁最上方有中文提示“请输入锁屏密码”样式提醒，具体 UI 界面如图 2-2 所示。

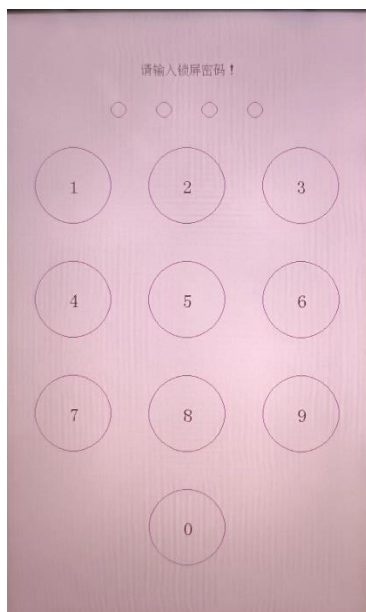


图 2-2 锁屏界面效果图

密码输入功能主要模拟用户输入密码的过程，本次密码输入采用触摸识别的方式，用户只需要点击 UI 界面对应的数字区域即可完成当前密码的输入，当用户点击对应数字后，在上方的输入小圆中会自动填满，表示当前输入完成，此次密码锁设置四位数字密码，用户在交互界面可以自行输入，输入密码后对应的界面变化如图 2-3。



图 2-3 输入密码效果图

密码判断部分主要是对用户输入的代码和预设密码进行判断，只有当密码输入正确才能进入后续的主菜单交互界面。本次功能设置了密码错误提示以及密码正确进入两个提示环节，并且设有蜂鸣器提示功能：当密码输入正确后，蜂鸣器短暂急速鸣声后停止，进入主菜单 UI 界面；当密码输入错误，蜂鸣器长鸣后返回锁屏界面，用户需重新输入。密码输入错误以及密码输入正确对应提示界面分别如下图 2-3a 和 2-3b。

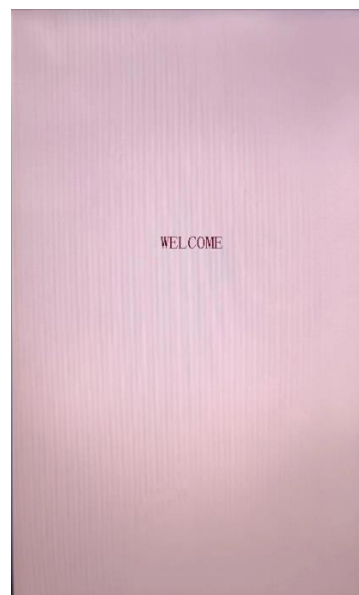


图 2-3a 密码错误效果图

图 2-3b 密码正确效果图

2.2.2 主菜单交互以及返回功能

当用户正确输入密码后，将进入主菜单交互界面，本系统的主菜单交互界面共设置 8 个可交互功能，分别为相册、实时时钟、GPS 定位获取、拨号盘模拟、温湿家居报警系统、MP3 音乐功能、手电筒以及电机模拟功能，在 UI 界面将各个功能图标按照尺寸以及颜色进行归一处理，以保持界面的美观性。主界面菜单图如下图 2-4 所示。

返回功能旨在实现流畅切换多功能，在每个功能对应界面左上角都有一个返回按键，当用户进入一个指定功能界面后需要退出当前界面而选择交互其他功能时，仅需点击返回按键，系统将自动退出到主菜单界面，此时用户可继续选择其他功能进行交互。



图 2-4 主菜单 UI 界面

2.2.3 相册

本系统的相册功能旨在模拟智能手机系统中的相册功能。相册功

能主要包括 3 个部分：相册 UI 界面、切换下一张图片以及切换上一张图片。

相册 UI 界面主要包括当前图片展示以及切换功能按钮：在屏幕中间有一张 320*240 大小的图片表示当前图片，在图片下方有 200*200 大小的切换按钮（共两个，分为上一张和下一张）。

切换功能主要通过实验板上的 S1 和 S2 按键实现，当用户按下 S1 键时，图片自动切换至上一张图片，当用户按下 S2 键时，图片自动切换至下一张图片。相册界面 UI 展示图如下图 2-5 所示。

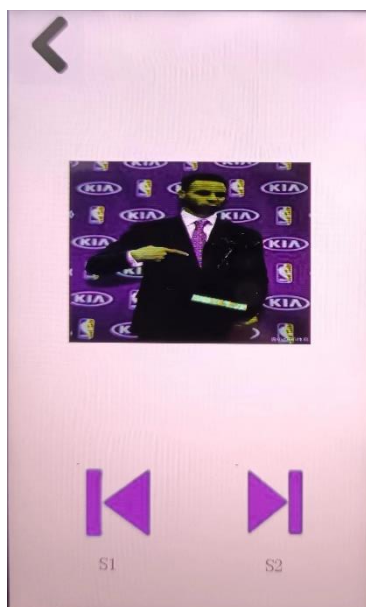


图 2-5 相册功能效果图

2.2.4 实时时钟

本系统的实时时钟功能旨在模拟智能手机中的实时时间功能。实时时钟功能包括时钟 UI 设计、屏幕数字时钟以及数码管分秒时钟三个表示部分。

时钟 UI 界面：在本次时钟设计中，在 LCD 屏幕中间有一个圆形钟

表样子的时钟，具有时分秒三根针随时间不断转动，在电子钟上有数字样式的实时时钟，用于判断当前时间的准确性。

数码管分秒显示功能，在实验板右侧有一个矩阵键盘，在上方的数码管将实时的时间分钟和秒钟显示出来，并不断随时间变化。实时时钟界面展示如图 2-6 所示。

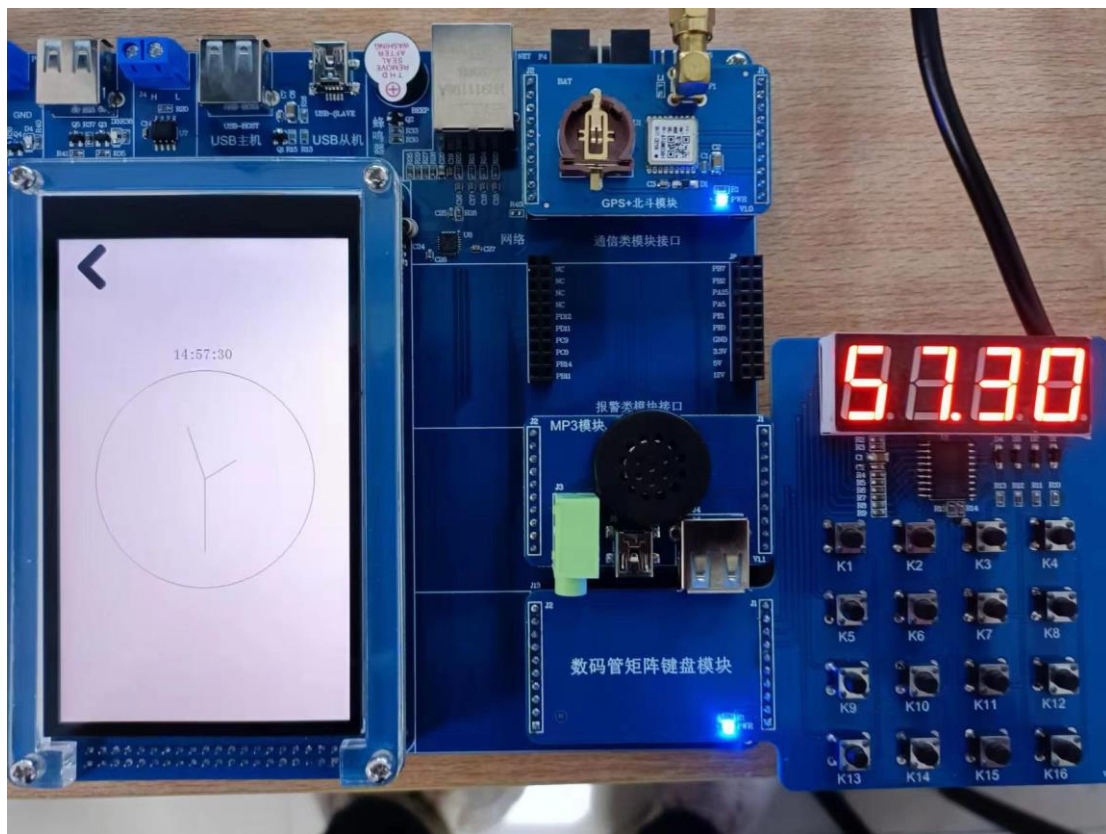


图 2-6 实时时钟效果图

2.2.5 GPS 定位

本系统的 GPS 定位功能旨在模拟智能手机中实时定位功能。GPS 定位功能主要实现对当前时间、所在地点的经纬度定位以及捕获的 GPS 和北斗卫星数量展示，具体 UI 界面如下图 2-7 所示。

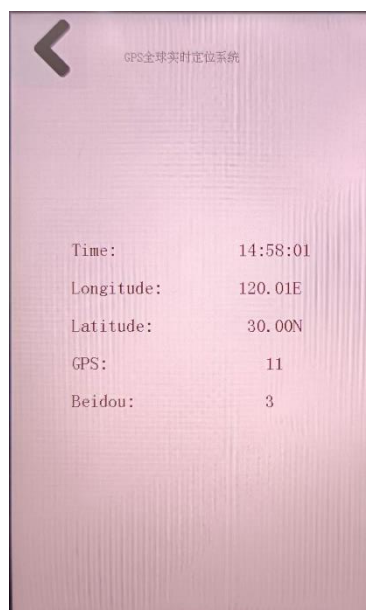


图 2-7 GPS 定位系统效果图

2.2.6 拨号盘

本系统的拨号盘功能旨在模拟智能手机中拨号功能。拨号盘功能主要由三部分实现：UI 界面、号码输入以及号码操作。

UI 界面：拨号盘 UI 界面主要仿照手机拨号盘功能，主要包括输入框、数字 0-9，删除以及清空四部分。

号码输入以及号码操作功能都是通过右边数码管矩阵键盘外设实现，用户只需在矩阵键盘上按下自己想输入的号码，在左侧屏幕输入框内会自动输出对应号码，其中需要说明的是 K10 按键对应数字 0，K11 按键对应删除当前某一号码，K12 按键对应将当前输入号码清空，用户可重新输入。拨号盘 UI 界面图如下图 2-8 所示。

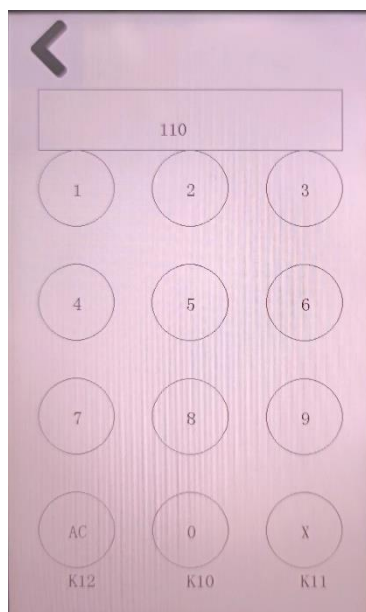


图 2-8 拨号盘功能 UI 图

2.2.7 温湿家居报警系统

本系统的温湿家居报警功能旨在模拟智能家居功能。温湿家居报警系统主要能实现对当前环境的温度湿度进行监测，同时当温度或湿度超出正常范围内，系统会实现报警，在 LCD 屏幕中体现为，对应区域变色以示区别，正常 UI 界面以及湿度报警界面分别如下图 2-9a 和 2-9b 所示。

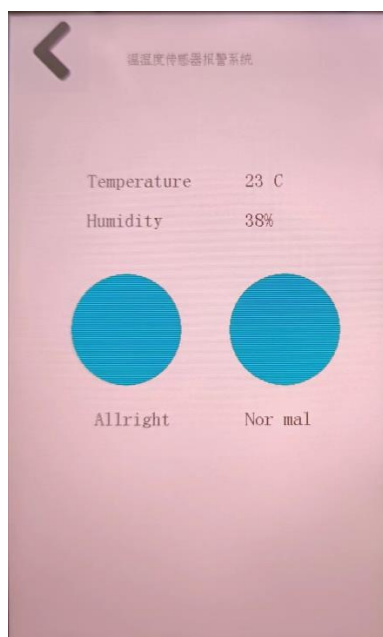


图 2-9a 系统正常效果图

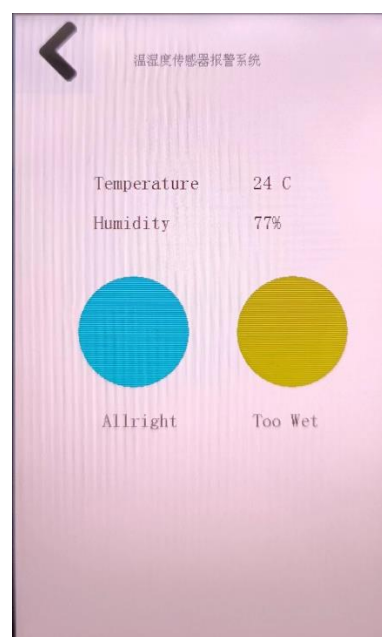


图 2-9b 湿度报警界面图

2.2.8 MP3 音乐播放器

本系统的 MP3 音乐播放器旨在模拟智能手机中的音乐功能。MP3 音乐播放器功能能够实现播放音乐、暂停播放、切换下一首、切换上一首、增加音量、减小音量、循环播放、循环暂停等多种功能，对应功能的实现在 LCD 屏幕中已经给出，如下图 2-10 所示。用户只需在右侧的矩阵键盘按下对应的操作按键便可使系统执行对应功能。同时在数码管中将实时显示当前曲目以及当前音量，能够人性化并且直观给用户提供操作。

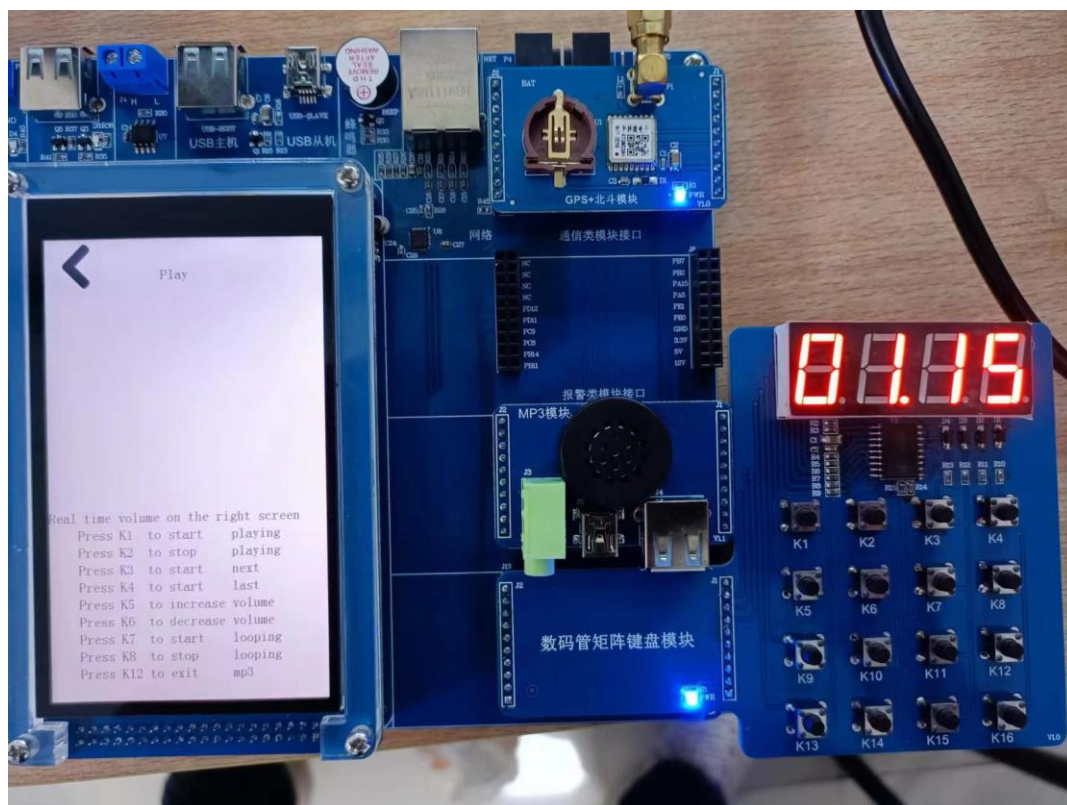


图 2-10 MP3 音乐播放器效果图

2.2.9 手电筒

本系统的手电筒功能旨在模拟智能手机中的手电筒功能。主要由 LED 灯以及 LCDUI 界面变化来实现，具体功能为：用户打开手电筒仅需按下按键 S1，实验板左侧的环状 LED 灯将以流水灯样式依次点亮，最后所有 LED 灯点亮实现手电筒打开，对应屏幕上有一个填充圆改变颜色代表手电筒打开状态；用户关闭手电筒功能只需按下按键 S2，手电筒关闭，所有 LED 灯熄灭，对应屏幕上填充圆改变颜色代表手电筒为关闭状态。手电筒打开以及关闭状态图分别如下图 2-11a 和 2-11b 所示。

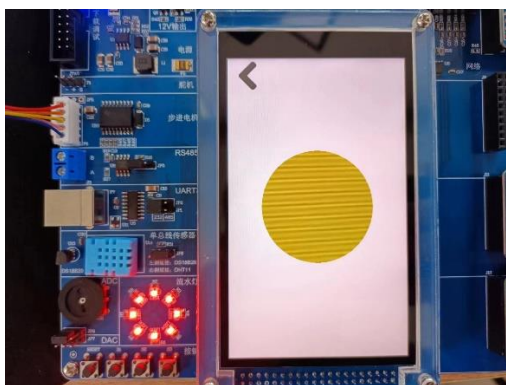


图 2-11a 手电筒打开效果图

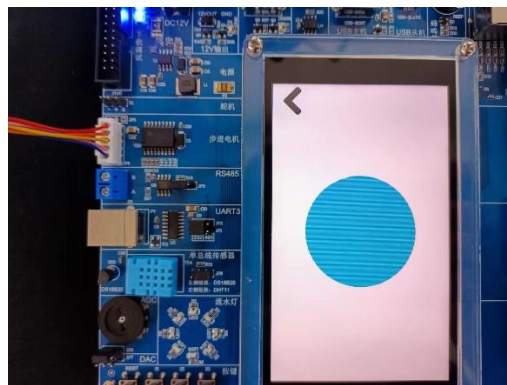


图 2-11b 手电筒关闭效果图

2.2.10 电机模拟

本系统的点电机模拟功能旨在模拟智能手机中的小游戏功能。用户可根据自己的需求来实现电机的转速调节，具体实现为：每次按下按键 S1 直流电机转速自动加 200，上限为 1000，按下按键 S2 直流电机转速自动减 200,下限为 0，按下按键 S3，直流电机转速归零。电机转速为 200 时状态图如下图 2-12 所示。

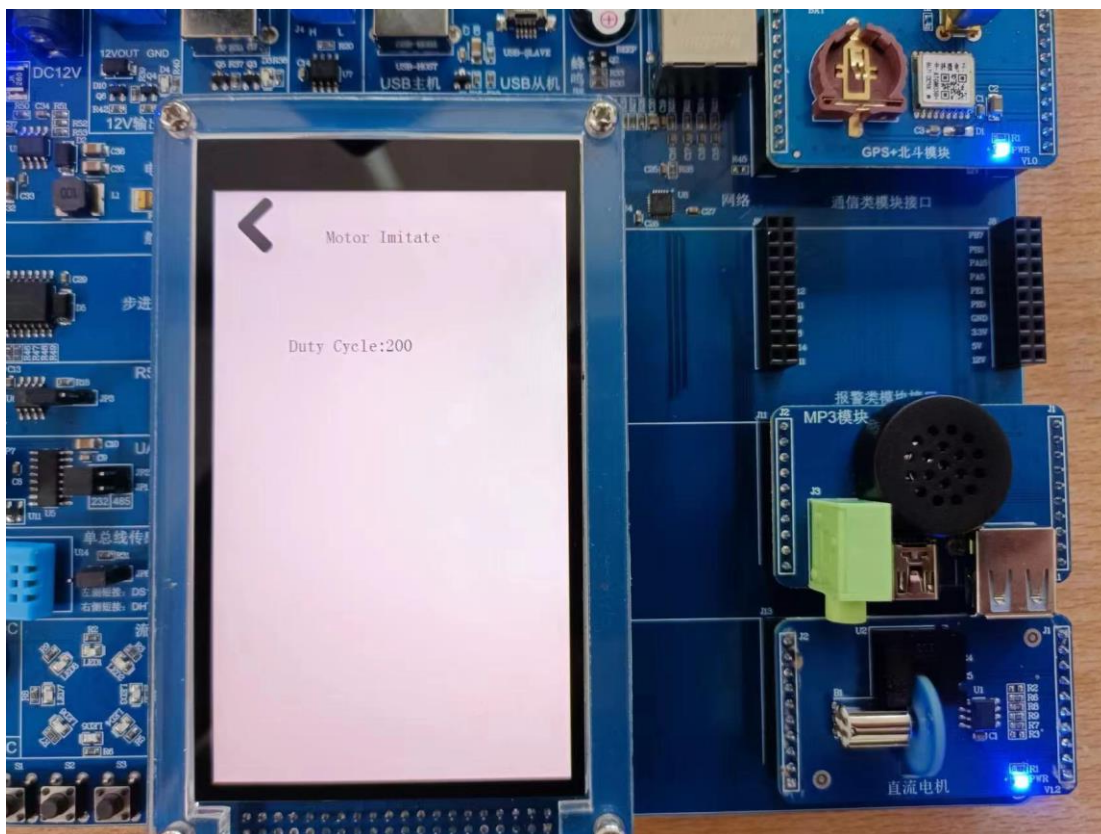


图 2-12 电机转速 200 效果图

三、硬件介绍

3.1 STM32F407 实验板

STM32F407 开发板基于 Cortex-M4 内核的 32 位 ARM 处理器，实现了多模块的应用实验。它是集学习、应用编程、开发研究于一体多功能创新平台。实验系统上的扩展模块接口能够拓展较为丰富的实验接口板，用户在了解扩展模块的接口定义后，更能研发出满足自身需求的实验接口板。系统资源分布如图 3-1 所示。

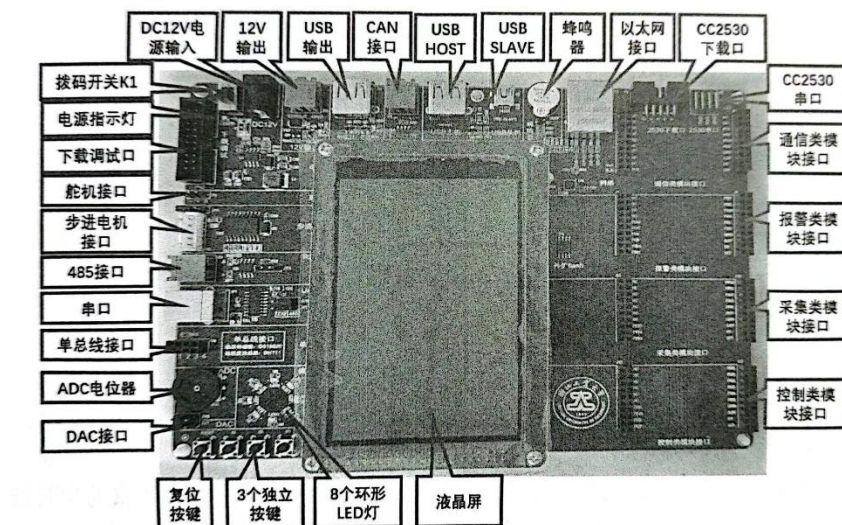


图 3-1 开发板资源图

3.2 各部分外设模块介绍

3.2.1 GPS+北斗模块

连接 GPS+北斗模块到 STM32F407 单片机，确保正确连接模块的电源、地线和通信线（串口等），使用 STM32 的 HAL 库，配置串口以

与 GPS+北斗模块进行通信。选择适当的波特率和串口引脚。GPS+北斗模块实物如下图 3-2 所示



图 3-2 GPS+北斗模块实物图

3.2.2 MP3 模块

将 MP3 模块连接到 STM32F407 单片机，确保正确连接模块的电源、地线以及任何必要的控制线。根据 MP3 模块的规格，选择合适的通信接口，使用 STM32 的 HAL 库，配置所选的通信接口。根据 MP3 模块的规格，设置正确的时钟频率、数据位宽等参数。根据 MP3 模块的指令集，发送命令以播放、暂停、停止等音频文件。通常，需要将音频文件存储在 SD 卡或其他存储介质上，并通过 STM32F407 发送指令来控制 MP3 模块。MP3 模块实物如下图 3-3 所示。



图 3-3 MP3 模块实物图

3.2.3 数码管矩阵键盘模块

将数码管矩阵键盘模块连接到 STM32F407 单片机, 确保正确连接模块的电源、地线以及数码管和矩阵键盘的引脚。根据数码管和矩阵键盘的规格, 选择合适的 GPIO 引脚用于连接。使用 STM32 的 HAL 库, 配置所选引脚的模式 (输入或输出)、上拉/下拉等参数。数码管矩阵键盘模块实物如下图 3-4 所示。



图 3-4 数码管矩阵键盘模块实物图

3.3.4 直流电机模块

将直流电机模块连接到 STM32F407 单片机, 确保正确连接模块的电源、地线以及电机的引脚。根据直流电机模块的规格, 选择合适的 GPIO 引脚用于连接。使用 STM32 的 HAL 库, 配置所选引脚为 PWM 输出模式。直流电机模块实物如下图 3-5 所示。

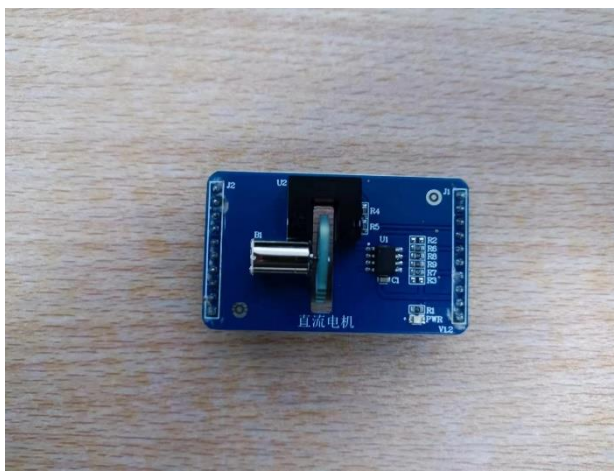


图 3-5 直流电机模块实物图

四、软件设计与程序实现

本系统主要涉及的知识点有: GPIO 相关配置 (LED、蜂鸣器)、定时器 PWM (直流电机)、GPIO 按键中断、定时器中断、串口通信 (MP3 模块、GPS+北斗、温湿度计 DHT11) 和 I2C (数码管显示、矩阵键盘)、LCD 显示、图片展示、中文字样显示、触摸屏交互。

核心代码如下:

1. 主函数

```
int main(void)
{
    int enter=0;
    float temp;
```

```

SysTick_Init();
LEDGpio_Init();
BEEPGpio_Init();
KEYGpio_Init();
LCD_Init();
Adc_Init();
UART3_Configuration();
UART5_Configuration(9600);
UART6_Configuration(9600);
setMp3Dev(FLASH0);      //设置 MP3 读取方式
delay_ms(200); DCMotorGpio_Init();
TIM2_Configuration();
EXTI_Config();
tp_dev.init();
POINT_COLOR = BLUE;
enter=In();    //进入锁屏界面
while(enter)
{
    menuset();    //主菜单界面
    apps();       //交互功能界面
}
}

```

2. 锁屏界面设计

```

int In()    //密码判断
{
    int k=1;
    while(k!=0)
    {
        int i=0,j=0;
        k=0;
        close();
        for(i=0;i<4;i++)
        {
            delay_ms(1000);
            ctp_test2(i);    //输入四位数字密码
        }
        delay_ms(200);
        for(j=0;j<4;j++)
        {
            if(password[j]!= Password[j])
            {
                k=1;
                break;
            }
        }
    }
}

```

```

        }
    }
    if(k==1)
    {
        BEEP2();
        LCD_Clear(WHITE);
        LCD_ShowString(0,300,450,24,24,"Incorrect password, please re-enter");
        delay_ms(1000);
        LCD_Clear(WHITE);
    }
    else
    {
        BEEP1();
        LCD_Clear(WHITE);
        LCD_ShowString(200,300,200,24,24,"WELCOME");
        delay_ms(1000);
        LCD_Clear(WHITE);
        return 1;
    }
}
}

```

3. 主菜单设计

```

void menuset()
{
    LCD_DrawPicture(60, 20, 210, 170, (u8*)gImage_photo);
    LCD_ShowString(110,180,200,24,24,"Photo");
    LCD_DrawPicture(250, 20, 400, 170, (u8*)gImage_timer);
    LCD_ShowString(310,180,200,24,24,"Time");
    LCD_DrawPicture(60, 210, 210, 360, (u8*)gImage_gps1);
    LCD_ShowString(120,370,200,24,24,"GPS");
    LCD_DrawPicture(250, 210, 400, 360, (u8*)gImage_phone);
    LCD_ShowString(310,370,200,24,24,"Phone");
    LCD_DrawPicture(60, 400, 210, 550, (u8*)gImage_a);
    LCD_ShowString(110,560,200,24,24,"TandW");
    LCD_DrawPicture(250, 400, 400, 550, (u8*)gImage_laba);
    LCD_ShowString(310,560,200,24,24,"MP3");
    LCD_DrawPicture(60, 590, 210, 740, (u8*)gImage_light);
    LCD_ShowString(110,750,200,24,24,"Light");
    LCD_DrawPicture(250, 590, 400, 740, (u8*)gImage_motor);
    LCD_ShowString(310,750,200,24,24,"Motor");
}

```

4. 应用交互

```

void apps()
{
    u8 t = 0;
    u8 i = 0;
    u16 lastpos[5][2];      //×ï°óÔ»´ÎμÄÊý¾Ý
    LED1_OFF;
    LED2_OFF;
    LED3_OFF;
    LED4_OFF;
    LED5_OFF;
    LED6_OFF;
    LED7_OFF;
    LED8_OFF;
    while(1)
    {
        tp_dev.scan(0);

        for(t = 0; t < 5; t++)
        {
            if((tp_dev.sta) & (1 << t))
            {
                if(tp_dev.x[t] < lcddev.width && tp_dev.y[t] < lcddev.height)
                {
                    if(lastpos[t][0] == 0xFFFF)
                    {
                        lastpos[t][0] = tp_dev.x[t];
                        lastpos[t][1] = tp_dev.y[t];
                    }
                }
            }
            if(exit_flag == 1)
            {
                //LCD_Clear(WHITE);

                menureset();

                photo_flag=0;
                light_flag=0;
                gps_flag=0;
                mp3_flag=0;
                temwet_flag = 0;
                motor_flag = 0;
                phone_flag = 0;
                time_flag=0;
            }
        }
        if(tp_dev.x[t]>60&&tp_dev.x[t]<210&&tp_dev.y[t]>20&&tp_dev.y[t]<170) //photo

```

```
        {  
            photo_flag=1;  
        }  
if(tp_dev.x[t]>60&&tp_dev.x[t]<210&&tp_dev.y[t]>590&&tp_dev.y[t]<740) //light  
    {  
        light_flag=1;  
    }  
if(tp_dev.x[t]>60&&tp_dev.x[t]<210&&tp_dev.y[t]>210&&tp_dev.y[t]<360) //gps  
    {  
        gps_flag=1;  
    }  
if(tp_dev.x[t]>250&&tp_dev.x[t]<400&&tp_dev.y[t]>400&&tp_dev.y[t]<550) //mp3  
    {  
        mp3_flag=1;  
    }  
if(tp_dev.x[t]>60&&tp_dev.x[t]<210&&tp_dev.y[t]>400&&tp_dev.y[t]<550) //temwet  
    {  
        temwet_flag=1;  
    }  
if(tp_dev.x[t]>250&&tp_dev.x[t]<400&&tp_dev.y[t]>590&&tp_dev.y[t]<740) //motor  
    {  
        motor_flag=1;  
    }  
if(tp_dev.x[t]>250&&tp_dev.x[t]<400&&tp_dev.y[t]>210&&tp_dev.y[t]<360) //phone  
    {  
        phone_flag=1;  
    }  
if(tp_dev.x[t]>250&&tp_dev.x[t]<400&&tp_dev.y[t]>21&&tp_dev.y[t]<170) //timer  
    time_flag=1;  
    if(photo_flag)  
        photo();  
    else if(light_flag)  
        Light();  
    else if(gps_flag)  
        Gps();  
    else if(mp3_flag)  
        music();  
    else if(temwet_flag)  
        tandw();  
    else if(motor_flag)  
        motor();  
    else if(phone_flag)  
        phone();  
    else if(time_flag)
```

```

                                timer();

                                delay_ms(5);
                                i++;
                                }
                                }

```

5. 各部分功能

```

void photo()
{
    u8 tempKEY=0;
    u8 number=1;
    LCD_Clear(WHITE);
    LCD_DrawPicture(0, 0, 100, 100, (u8*)gImage_back);
    LCD_DrawPicture(80, 200, 400, 440, (u8*)gImage_111);
    LCD_DrawPicture(90, 600, 190, 700, (u8*)gImage_shang);
    LCD_DrawPicture(300, 600, 400, 700, (u8*)gImage_xia);
    LCD_ShowString(120,720,450,24,24,"S1");
    LCD_ShowString(340,720,450,24,24,"S2");
    while(1)
    {
        tempKEY = KeyScan();
        if(tempKEY == 1)
        {
            number--;
            if(number<=0)
            number=1;
            switch(number)
            {
            case 1:
                LCD_DrawPicture(80, 200, 400, 440, (u8*)gImage_111);    break;
            case 2:
                LCD_DrawPicture(80, 200, 400, 440, (u8*)gImage_2015);  break;
            }
        }
        else if(tempKEY == 2)
        {
            number++;
            if(number>=2)
            number=2;
            switch(number)
            {
            case 1:
                LCD_DrawPicture(80, 200, 400, 440, (u8*)gImage_111);    break;
            case 2:

```

```

    LCD_DrawPicture(80, 200, 400, 440, (u8*)gImage_2015); break;
}
}
else if(tempKEY == 3)
{
    break;
}
}
while(1)
{
    Exit();
    if(exit_flag == 1)
        {
            photo_flag = 0;
            LCD_Clear(WHITE);
            break;
        }
}
}

void Gps()
{
    char dispbuf[100] = "";
    u8 key=0;
    LCD_Clear(WHITE);
    LCD_DrawPicture(0, 0, 100, 100, (u8*)gImage_back);
    GPS_Buff_Init();
    LCD_ShowString2(150, 50, "GPSÊ«ÇòÊµÊ±¶Í»ÎµÍ³", BLACK, WHITE);
    while (1)
    {
        if(RxDataOutPtr != RxDataInPtr)
        {
            if (memcmp(&RxDataOutPtr[2], "$GNGGA", 6) == 0)
            {
                GNGGA_info(&RxDataOutPtr[2]);
                sprintf(dispbuf, "%s", gps_time);
                LCD_ShowString(80,300,200, 24,24,"Time:");
                LCD_ShowString(300,300,200, 24,24,(u8*)dispbuf);
                sprintf(dispbuf, "%6.2f%c", longitude,longitude_dir);
                LCD_ShowString(80,350,200, 24,24,"Longitude:");
                LCD_ShowString(300,350,200, 24,24,(u8*)dispbuf);
                sprintf(dispbuf, "%6.2f%c", latitude,latitude_dir);
                LCD_ShowString(80,400,200, 24,24,"Latitude:");
            }
        }
    }
}

```



```

LCD_ShowString(300,400,200, 24,24,(u8*)dispbuf);
}
else if (memcmp(&RxDataOutPtr[2], "$GPGSV", 6) == 0)
{
    GSV_info(&RxDataOutPtr[2]);
    sprintf(dispbuf, "    %d\r\n ",gpgsv.numSv);
    LCD_ShowString(80,450,200, 24,24,"GPS:");
    LCD_ShowString(300,450,200, 24,24,(u8*)dispbuf);
}
else if (memcmp(&RxDataOutPtr[2], "$BDGSV", 6) == 0)
{
    GSV_info(&RxDataOutPtr[2]);
    sprintf(dispbuf, "    %d ",gpgsv.numSv);
    LCD_ShowString(80,500,200, 24,24,"Beidou:");
    LCD_ShowString(300,500,200, 24,24,(u8*)dispbuf);
}
RxDataOutPtr += RBUF_UNIT;
if(RxDataOutPtr == RxDataEndPtr)
RxDataOutPtr = RxDataBuf[0];
}
key = KeyScan();
if(key==3)
break;
    }
while(1)
{
    Exit();
    if(exit_flag == 1)
    {
        gps_flag = 0;
        LCD_Clear(WHITE);
        break;
    }
}
}

```

```

void music()
{
    u8 i=0;
    u8 flag=0;
    u8 voice=15;
    u8 number=1;
    u8 showH[2],showL[2];

```

```

float tempadc;
LCD_Clear(WHITE);
LCD_DrawPicture(0, 0, 100, 100, (u8*)gImage_back);
setMp3Vol(voice);
init_CH455();
CH455_Write( CH455_SYSON );
LCD_ShowString(50, 500, 400,24,24,"Press K1  to start  playing");
LCD_ShowString(50, 530, 400,24,24,"Press K2  to stop    playing");
LCD_ShowString(50, 560, 400,24,24,"Press K3  to start  next   ");
LCD_ShowString(50, 590, 400,24,24,"Press K4  to start  last   ");
LCD_ShowString(50, 620, 400,24,24,"Press K5  to increase volume ");
LCD_ShowString(50, 650, 400,24,24,"Press K6  to decrease volume ");
LCD_ShowString(50, 680, 400,24,24,"Press K7  to start  looping");
LCD_ShowString(50, 710, 400,24,24,"Press K8  to stop    looping");
LCD_ShowString(50, 740, 400,24,24,"Press K12 to exit  mp3    ");
LCD_ShowString(0, 470, 470,24,24,"Real time volume on the right screen");
showL[0] = BCD_decode_tab[voice%10];
showH[0] = BCD_decode_tab[voice/10];
showL[1] = BCD_decode_tab[0];
showH[1] = BCD_decode_tab[0];
CH455_Write( CH455_DIG3 | showL[0] );
CH455_Write( CH455_DIG2 | showH[0] );
CH455_Write( CH455_DIG1 | showL[1] | 0x80);
CH455_Write( CH455_DIG0 | showH[1] );
while(1)
{
#ifdef USE_CH455_KEY
    ch455_key = 0xff;
    while( ch455_key == 0xff);
    i = ch455_key & 0x3f;//0°%4üÖμ
    if( i<=7 )
    {
        i=i-3;
    }else if( 12<=i & i<=15 )
    {
        i=i-7;
    }else if( 20<=i & i<=23 )
    {
        i=i-11;
    }else
    {
        i=i-15;
    }
    if( i==1 )

```

```
{
LCD_ShowString(200, 50, 200,24,24,"Play ");
mp3Play();
if(flag=0)
{number++;
flag=1;}
}
else if(i==2)
{
mp3Stop();
LCD_ShowString(200, 50, 200,24,24,"Stop ");
}
else if(i==3)
{
mp3next();
number++;
if(number>=4)
number=1;
LCD_ShowString(200,150, 200,24,24,"Next ");
showH[1] = BCD_decode_tab[0];
showL[1] = BCD_decode_tab[number];
}
else if(i==4)
{
mp3last();
LCD_ShowString(200, 150, 200,24,24,"Last ");
number--;
if(number<=0)
number=3;
showH[1] = BCD_decode_tab[0];
showL[1] = BCD_decode_tab[number];
}
else if(i==5)
{
mp3voiceadd();
voice++;
if(voice>30)
voice=30;
LCD_ShowString(200, 250, 200,24,24,"Loud ");
showH[0] = BCD_decode_tab[voice/10];
showL[0] = BCD_decode_tab[voice%10];
}
else if(i==6)
{
```

```

mp3voicedet();
voice--;
if(voice<0)
    voice=0;
    LCD_ShowString(200, 250, 200,24,24,"Small");
showH[0] = BCD_decode_tab[voice/10];
    showL[0] = BCD_decode_tab[voice%10];
    }
else if(i==7)
    {
mp3loopplay();
    LCD_ShowString(200,350, 200,24,24,"LoopStart");
    }
else if(i==8)
    {
mp3loopstop();
    LCD_ShowString(200,350, 200,24,24,"LoopEnd  ");
    }

else if(i==16)
{
    LCD_ShowString(200, 450, 200,24,24,"Exit ");
break;
    }
CH455_Write( CH455_DIG3 | showL[0] );
    CH455_Write( CH455_DIG2 | showH[0] );
    CH455_Write( CH455_DIG1 | showL[1] |0x80 );
    CH455_Write( CH455_DIG0 | showH[1] );
while(1)
{
    i = CH455_Read( );
    if( i & 0x40 )
    {
        CH455_Write( CH455_DIG3 );
        CH455_Write( CH455_DIG2 );
        delay_ms(50);
        CH455_Write( CH455_DIG3 | showL[0] );
        CH455_Write( CH455_DIG2 | showH[0] );
        delay_ms(50);
    }
    else
    {
        break;
    }
}

```

```

    }
#endif

}

while(1)
{
    Exit();
    if(exit_flag == 1)
    {
        phone_flag = 0;
        LCD_Clear(WHITE);
        break;
    }
}
}

void tandw()
{
    unsigned char temperature;
    unsigned char humidity;
    u8 key;
    char dispbuf[100] = "";
    LCD_Clear(WHITE);
    LCD_DrawPicture(0, 0, 100, 100, (u8*)gImage_back);
    LCD_ShowString2(150, 50, "ÎÂÊ'«,ÐÆ÷±³¼¬µ³", BLACK, WHITE);
    while(1)
    {
        DTH11_ReadData(&temperature, &humidity);
        sprintf(dispbuf, "%d C", temperature);
        LCD_ShowString(100,200,200, 24,24,"Temperature");
        LCD_ShowString(300,200,200, 24,24,(u8*)dispbuf);
        sprintf(dispbuf, "%d% %", humidity);
        LCD_ShowString(100,250,200, 24,24,"Humidity");
        LCD_ShowString(300,250,200, 24,24,(u8*)dispbuf);
        if(temperature>=35)
        {
            gui_fill_circle(150,400,70,RED);
            LCD_ShowString(110, 500, 200,24,24,"Too High");
            BEEP2();
            delay_ms(1000);
        }
        if(temperature<=10)
        {

```

```
gui_fill_circle(150,400,70,BLUE);
LCD_ShowString(110, 500, 200,24,24,"Too Cold");
BEEP2();
delay_ms(1000);
}
else
{
gui_fill_circle(150,400,70,GREEN);
LCD_ShowString(110, 500, 200,24,24,"Allright");

}
if(humidity>=70)
{
gui_fill_circle(350,400,70,RED);
LCD_ShowString(300, 500, 200,24,24,"Too Wet");
BEEP2();
delay_ms(1000);
}
if(humidity<=10)
{
gui_fill_circle(350,400,70,BLUE);
LCD_ShowString(300, 500, 200,24,24,"Too Dry");
BEEP2();
delay_ms(1000);
}
else
{
gui_fill_circle(350,400,70,GREEN);
LCD_ShowString(300, 500, 200,24,24,"Nor mal");
delay_ms(1000);
}
key = KeyScan();
if(key==3)
break;
}
while(1)
{
Exit();
if(exit_flag == 1)
{
temwet_flag = 0;
LCD_Clear(WHITE);
// BEEP_ON;
break;
```

```
    }
}
}
void motor()
{
    u8 t=0;
    unsigned int data = 0;
    char dispbuf[100] = "";
    LCD_Clear(WHITE);

    LCD_DrawPicture(0, 0, 100, 100, (u8*)gImage_back);
    LCD_ShowString(150, 50, 200,24,24,"Motor Imitate");
    LCD_ShowString(100, 200,200,24,24, "Duty Cycle:0");
    while(1)
    {
        t = KeyScan();
        if (t == 1)
        {
            data += 200;
            if(data > 1000)
            {
                data = 1000;
            }

            TIM_SetCompare1(TIM10, data);
            sprintf(dispbuf, "Duty Cycle:%-5d", data);
            LCD_ShowString(100, 200,200,24,24, (u8*)dispbuf);
        }
        else if(t == 2)
        {
            if(data!=0)
            {
                data -= 200;
                if(data < 100)
                {
                    data = 0;
                }
            }

            TIM_SetCompare1(TIM10, data);
            sprintf(dispbuf, "Duty Cycle:%-5d", data);
            LCD_ShowString(100, 200,200,24,24, (u8*)dispbuf);
        }
        else if(t == 3)
        {

```

```

TIM_SetCompare1(TIM10, 0);
LCD_ShowString(100, 200, 200, 24, 24, "Duty Cycle:0");
break;
}
}
while(1)
{
Exit();
if(exit_flag == 1)
{
motor_flag = 0;
LCD_Clear(WHITE);
break;
}
}
}

```

```

void phone()
{
u8 i;
u8 key;
u8 showH[2], showL[2];
u8 number[20] = {0};
u8 n=0; u8 j=0; u8 k=0;
char dispbuf[100] = "";
u32 data=0;
u32 datatem=0;
closeforphone();
init_CH455();
CH455_Write( CH455_SYSON );
showL[0] = BCD_decode_tab[0];
showH[0] = BCD_decode_tab[0];
showL[1] = BCD_decode_tab[0];
showH[1] = BCD_decode_tab[0];
CH455_Write( CH455_DIG3 | showL[0] );
CH455_Write( CH455_DIG2 | showH[0] );
CH455_Write( CH455_DIG1 | showL[1] );
CH455_Write( CH455_DIG0 | showH[1] );
while(1)
{
#ifdef USE_CH455_KEY
ch455_key = 0xff;
while( ch455_key == 0xff);
i = ch455_key & 0x3f; // 0~14

```



```
    if( i<=7 )
    {
        i=i-3;
    }else if( 12<=i & i<=15 )
    {
        i=i-7;
    }else if( 20<=i & i<=23 )
    {
        i=i-11;
    }else
    {
        i=i-15;
    }
    if( i<10 )
    {
        data=data*10;
    }
    datatem=i;
    data=data+datatem;
    sprintf(disbuf, "%d", data);
    LCD_ShowString(200,140,200, 24,24,(u8*)disbuf);
    //LCD_ShowString2(200, 150, (u8*)disbuf, BLACK, WHITE);
    showH[0] = BCD_decode_tab[0];
    showL[0] = BCD_decode_tab[i];
    BEEP3();
}
else if(i==10)
{
    data=data*10;
    sprintf(disbuf, "%d", data);
    LCD_ShowString(200, 140,200, 24,24,(u8*)disbuf);
    showH[0] = BCD_decode_tab[i/10];
    showL[0] = BCD_decode_tab[i-10];
    BEEP3();
}
else if(i==11)//x
{
    closeforphone();
    data=data/10;
    sprintf(disbuf, "%d", data);
    LCD_ShowString(200, 140,200, 24,24,(u8*)disbuf);
    showH[0] = BCD_decode_tab[i/10];
    showL[0] = BCD_decode_tab[i-10];
    BEEP3();
}
```

```
n++;  
    }  
else if(i==12)    //ac  
{  
data=0;  
closeforphone();  
    showH[0] = BCD_decode_tab[i/10];  
    showL[0] = BCD_decode_tab[i-10];  
    BEEP3();  
  
    }  
else if(i==16)    //1636  
{  
  
    break;  
    }  
    CH455_Write( CH455_DIG3 | showL[0] );  
    CH455_Write( CH455_DIG2 | showH[0] );  
    CH455_Write( CH455_DIG1 | showL[1] );  
    CH455_Write( CH455_DIG0 | showH[1] );  
  
while(1)  
{  
    i = CH455_Read( );  
    if( i & 0x40 )  
    {  
        CH455_Write( CH455_DIG3 );  
        CH455_Write( CH455_DIG2 );  
        delay_ms(50);  
        CH455_Write( CH455_DIG3 | showL[0] );  
        CH455_Write( CH455_DIG2 | showH[0] );  
        delay_ms(50);  
    }  
    els  
    {  
        break;  
    }  
}  
#endif  
  
}  
while(1)  
{  
Exit();
```

```

if(exit_flag == 1)
{
    phone_flag = 0;
    LCD_Clear(WHITE);
    break;
}
}

void timer()
{
    char dispbuf[100] = "";
    u8 key=0;
    u8 rs=120;
    u8 rm=90;
    u8 rh=60;
    u8 showH[2],showL[2];
    u8  m0;u8  m1;u8  s0;u8 s1;u8 s;u8 m;u8  h0;u8h1;u8h;
    float angle = 6.0;
    float radian = angle * 3.1415926 / 180.0;
    timescreen();
    init_CH455();
    CH455_Write( CH455_SYSON );
    showL[0] = BCD_decode_tab[0];
    showH[0] = BCD_decode_tab[0];
    showL[1] = BCD_decode_tab[0];
    showH[1] = BCD_decode_tab[0];
    CH455_Write( CH455_DIG3 | showL[0] );
    CH455_Write( CH455_DIG2 | showH[0] );
    CH455_Write( CH455_DIG1 | showL[1] );
    CH455_Write( CH455_DIG0 | showH[1] );
    GPS_Buff_Init();
    LCD_Draw_Circle(240,400,180);
    while (1)
    {
        if(RxDataOutPtr != RxDataInPtr)
        {
            if (memcmp(&RxDataOutPtr[2], "$GNGGA", 6) == 0)
            {
                GNGGA_info(&RxDataOutPtr[2]);
                sprintf(dispbuf, "%s", gps_time);
                LCD_ShowString(190,180,200, 24,24,(u8*)dispbuf);
                h0=dispbuf[0]-'0';
            }
        }
    }
}

```

```

h1=dispbuff[1]-'0';
m0=dispbuff[3]-'0';
m1=dispbuff[4]-'0';
s0=dispbuff[6]-'0';
s1=dispbuff[7]-'0';
s=s0*10+s1;
m=m0*10+m1;
h=h0*10+h1;
showL[0] = BCD_decode_tab[s1];
    showH[0] = BCD_decode_tab[s0];
    showL[1] = BCD_decode_tab[m1];
    showH[1] = BCD_decode_tab[m0];
    CH455_Write( CH455_DIG3 | showL[0] );
    CH455_Write( CH455_DIG2 | showH[0] );
    CH455_Write( CH455_DIG1 | showL[1]|0x80 );
    CH455_Write( CH455_DIG0 | showH[1] );
gui_fill_circle(240,400,150,WHITE);

LCD_DrawLine(240,400,240+rs*sin(angle *s* 3.1415926 / 180.0),400-rs*cos(angle *s* 3.1415926 / 180.0));
LCD_DrawLine(240,400,240+rm*sin(angle *m* 3.1415926 / 180.0),400-rm*cos(angle *m* 3.1415926 / 180.0));
LCD_DrawLine(240,400,240+rh*sin(30.0 *h* 3.1415926 / 180.0),400-rh*cos(30.0 *h* 3.1415926 / 180.0));
}
RxDataOutPtr += RBUFF_UNIT;
if(RxDataOutPtr == RxDataEndPtr)
RxDataOutPtr = RxDataBuf[0];
}
key = KeyScan();
if(key==3)
{
showL[0] = BCD_decode_tab[0];
    showH[0] = BCD_decode_tab[0];
    showL[1] = BCD_decode_tab[0];
showH[1] = BCD_decode_tab[0];
    CH455_Write( CH455_DIG3 | showL[0] );
    CH455_Write( CH455_DIG2 | showH[0] );
    CH455_Write( CH455_DIG1 | showL[1] );
    CH455_Write( CH455_DIG0 | showH[1] );
break;
}
}
while(1)
{
Exit();
if(exit_flag == 1)

```

```
    {  
        time_flag = 0;  
        LCD_Clear(WHITE);  
        break;  
    }  
}  
  
}  
  
void Light()  
{  
    u8 tempLIGHT=0;  
    LCD_Clear(WHITE);  
    LCD_DrawPicture(0, 0, 100, 100, (u8*)gImage_back);  
    gui_fill_circle(240,400,150,BLACK);  
    while(1)  
    {  
  
        tempLIGHT = KeyScan();  
        if(tempLIGHT == 1)  
        {  
            gui_fill_circle(240,400,150,RED);  
            LED1_ON;  
            delay_ms(500);  
            LED1_OFF;  
            LED2_ON;  
            delay_ms(500);  
            LED2_OFF;  
            LED3_ON;  
            delay_ms(500);  
            LED3_OFF;  
            LED4_ON;  
            delay_ms(500);  
            LED4_OFF;  
            LED5_ON;  
            delay_ms(500);  
            LED5_OFF;  
            LED6_ON;  
            delay_ms(500);  
            LED6_OFF;  
            LED7_ON;  
            delay_ms(500);  
            LED7_OFF;  
            LED8_ON;
```

```
delay_ms(500);
LED8_OFF;

    LED1_ON;
    LED2_ON;
    LED3_ON;
    LED4_ON;
    LED5_ON;
    LED6_ON;
    LED7_ON;
    LED8_ON;
}
else if(tempLIGHT == 2)
{
gui_fill_circle(240,400,150,GREEN);
LED1_OFF;
    LED2_OFF;
    LED3_OFF;
    LED4_OFF;
    LED5_OFF;
    LED6_OFF;
    LED7_OFF;
    LED8_OFF;
}
else if(tempLIGHT == 3)
{
    break;
}
}
while(1)
{
Exit();
    if(exit_flag == 1)
        {
light_flag = 0;
    LCD_Clear(WHITE);
        break;
        }
}
}
```

五、总结

本系统将本学期所学过的单片机知识与单片机实验相结合，通过 STM32F407 芯片实现一个功能强大的智能手机系统。系统涵盖了触摸屏、密码锁、相册、实时时钟、GPS 获取、拨号盘、温湿度家居控制、电机模拟、手电筒、音乐播放器等多项功能。通过硬件连接和软件设计的综合实践，展示 STM32F407 在嵌入式系统中的广泛应用。通过本实验，学生更加加强对单片机知识的理解，同时对单片机的使用也更为熟练。

最后，感谢学校提供的实验开发板设备，以及课程授课老师欧县华老师的悉心指点！