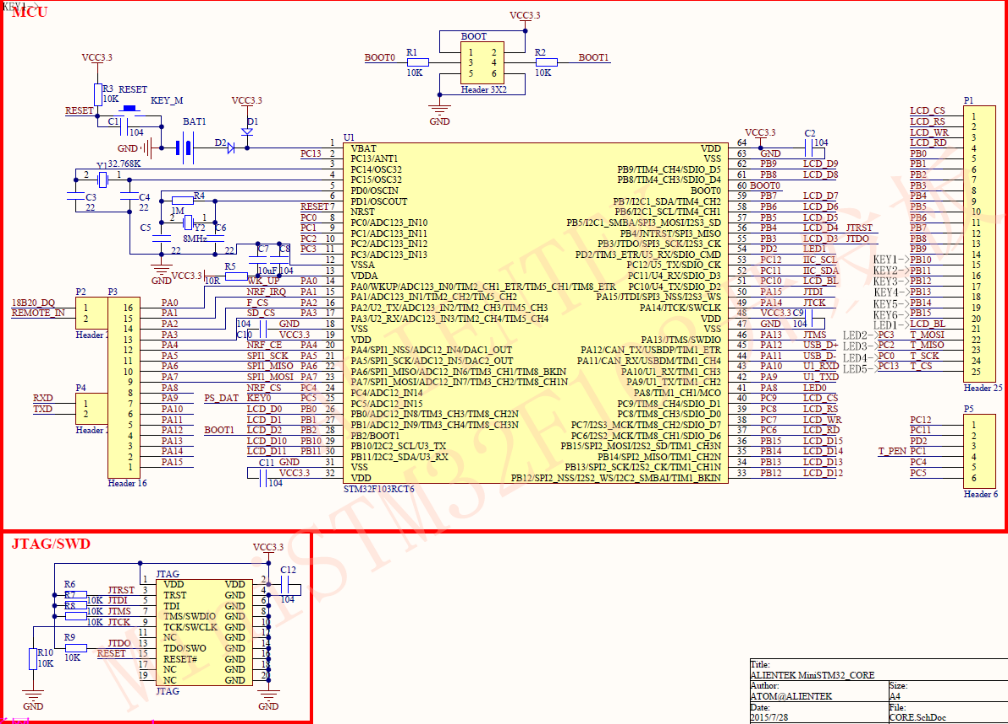
**评估板硬件和软件部分详细介绍**

**1、硬件原理图：详见“硬件原理图”文件夹**

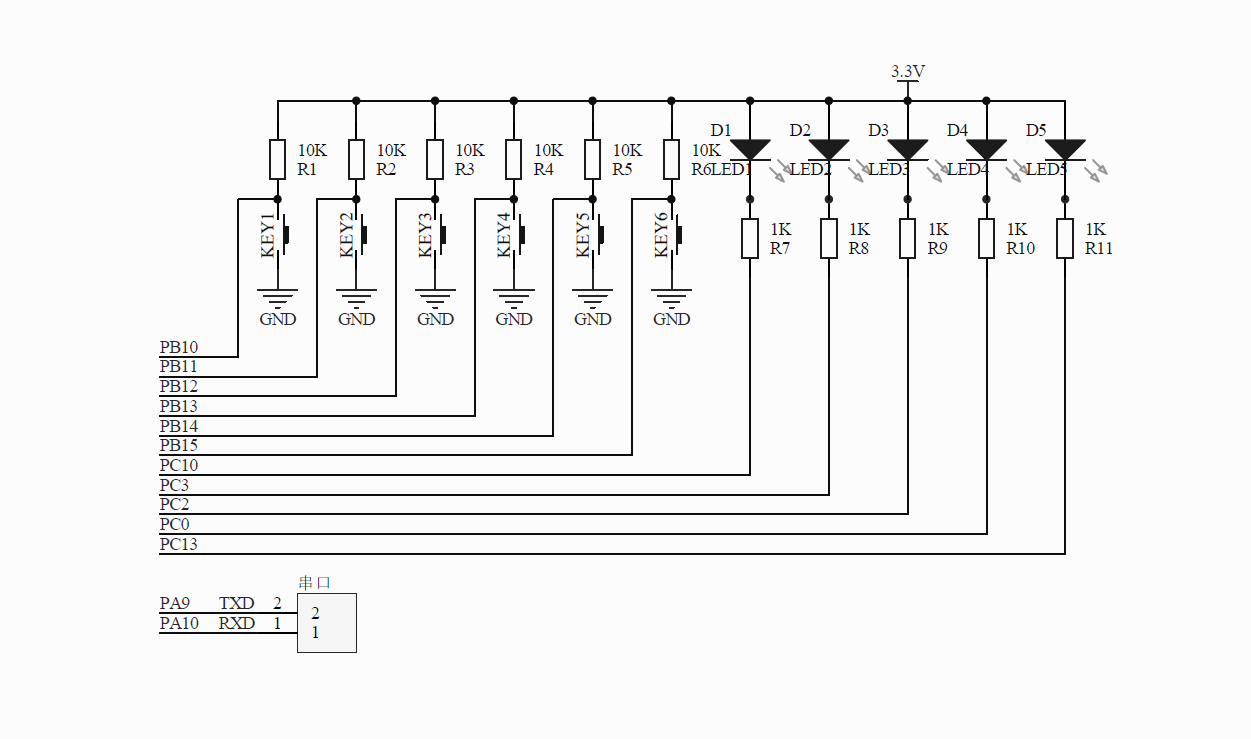
（1）硬件接口定义：

|  |  |  |
| --- | --- | --- |
| 功能 | 对应STM32F103的IO |  |
| KEY1（屏幕亮度开关） | PB10 |  |
| KEY2（温度+） | PB11 |  |
| KEY3（温度-） | PB12 |  |
| KEY4（播放/停止） | PB13 |  |
| KEY5（音量+） | PB14 |  |
| KEY6（音量-） | PB15 |  |
| LED1 | PC10 |  |
| LED2 | PC3 |  |
| LED3 | PC2 |  |
| LED4 | PC0 |  |
| LED5 | PC13 |  |
| TXD2(触摸屏端子显示) | PA10 |  |
| RXD2(触摸屏端子显示) | PA9 |  |

（2）硬件原理图：



*图1：STM32F103原理图*



*图2：LED及KEY原理图*

**2、软件程序**

界面工程文件在“DGUS\_II工程”文件夹中。STM32程序在“MCU\_STM32F103RCT6程序”文件夹中。

（1）界面工程

LED1图标：0X1000

LED1按键：0X1200

LED2图标：0X1001

LED2按键：0X1201

LED3图标：0X1002

LED3按键：0X1202

LED4图标：0X1003

LED4按键：0X1203

LED5图标：0X1004

LED5按键：0X1204

音量显示框地址：0X1210

温度显示框地址：0X1110

（2）MCU程序

1）.main():硬件初始化，触摸屏初始化，按键扫描，接收触摸屏数据处理

int main(void)

{

uint8\_t i;

HAL\_Init();

SystemClock\_Config();

MX\_GPIO\_Init();

MX\_USART1\_UART\_Init();

while (1)

{

HAL\_UART\_Receive\_IT(&huart1,&usart1\_data,1);

if(UP\_POWER\_FLAG == 0)//上电初始化触摸屏

{

UP\_POWER\_FLAG = 1;

HMI\_SEND\_BUF[0] = TEMP;

PLC\_Data\_Wr2HMI(0X1110,0,1);

delay\_ms(100);

HMI\_SEND\_BUF[0] = VOLUME;

PLC\_Data\_Wr2HMI(0X1210,0,1);//ÒôÁ¿ÏÔÊ¾

delay\_ms(100);

WRITE\_HMI(0X00A1,VOLUME,0x00,0,0,2);//Ä¬ÈÏÒôÁ¿50

delay\_ms(100);

for(i=0;i<5;i++)

{

HMI\_SEND\_BUF[i] = 0;

}

for(i=0;i<20;i++)

{

usart1\_rxBuf[i] = 0x00;

}

HMI\_HANDLEFLAG = 0;

}

if(PLAY ==1)//监测是否播放完音乐

{

READ\_HMI(0x00a1,1);

delay\_ms(100);

if(HMI\_HANDLEFLAG)

{

if(usart1\_rxBuf[8] == 0)

{

PLAY = 0;

for(i=0;i<50;i++)

{

usart1\_rxBuf[i] = 0x00;

}

usart1\_counter = 0;

}

}

}

SCAN\_KEY(0);//按键扫描

if(HMI\_HANDLEFLAG)//接收触摸屏数据处理

{

HMI\_HANDLEFLAG = 0;

CLOSE\_OPEN\_IO(0x01);

CLOSE\_OPEN\_IO(0x02);

CLOSE\_OPEN\_IO(0x03);

CLOSE\_OPEN\_IO(0x04);

CLOSE\_OPEN\_IO(0x05);

for(i=0;i<50;i++)

{

usart1\_rxBuf[i] = 0x00;

}

usart1\_counter = 0;

}

}

2）. PLC\_Data\_Wr2HMI ():数据写入触摸屏变量寄存器

void PLC\_Data\_Wr2HMI(uint16\_t HMI\_start\_addr,uint16\_t PLC\_return\_data\_start\_addr,uint8\_t len)

{

uint8\_t i;

usart1\_txBuf[0]=0x5a;

usart1\_txBuf[1]=0xa5;

usart1\_txBuf[2]=len\*2+3;

usart1\_txBuf[3]=0x82;

usart1\_txBuf[4]=((uint8\_t)(HMI\_start\_addr>>8))&0XFF;

usart1\_txBuf[5]=(uint8\_t)(HMI\_start\_addr&0XFF);

for(i=0;i<len;i++)

{

usart1\_txBuf[i\*2+6]=((HMI\_SEND\_BUF[i+PLC\_return\_data\_start\_addr] >>8)&0xff);

usart1\_txBuf[i\*2+7]=(HMI\_SEND\_BUF[i+PLC\_return\_data\_start\_addr] &0xff);

}

HAL\_UART\_Transmit(&huart1, usart1\_txBuf, len\*2+6, 10);//串口发送

}

3）. WRITE\_HMI ():数据写入触摸屏寄存器

void WRITE\_HMI(uint16\_t addr,uint8\_t data1,uint8\_t data2,uint8\_t data3,uint8\_t data4,uint8\_t len)

{

usart1\_txBuf[0]=0x5a;

usart1\_txBuf[1]=0xa5;

usart1\_txBuf[2]=len\*2+3;

usart1\_txBuf[3]=0x82;

usart1\_txBuf[4]=(uint8\_t)((addr>>8)&0xff);

usart1\_txBuf[5]=(uint8\_t)(addr&0xff);

usart1\_txBuf[6]=data1;

usart1\_txBuf[7]=data2;

usart1\_txBuf[8]=data3;

usart1\_txBuf[9]=data4;

HAL\_UART\_Transmit(&huart1, usart1\_txBuf, len\*2+6 , 20);

}

4）. READ\_HMI ():读触摸屏寄存器数据

void READ\_HMI(uint16\_t addr,uint8\_t num)

{

usart1\_txBuf[0]=0x5a;

usart1\_txBuf[1]=0xa5;

usart1\_txBuf[2]=0x04;

usart1\_txBuf[3]=0x83;

usart1\_txBuf[4]=(uint8\_t)((addr>>8)&0xff);

usart1\_txBuf[5]=(uint8\_t)(addr&0xff);

usart1\_txBuf[6]=num;

HAL\_UART\_Transmit(&huart1, usart1\_txBuf, 7 , 20);

}

5). CLOSE\_OPEN\_IO ():开关LED

void CLOSE\_OPEN\_IO(uint8\_t hmi\_key\_value)

{

uint8\_t i;

if(usart1\_rxBuf[7] == 0x00&&usart1\_rxBuf[8] == hmi\_key\_value)//LED1

{

if(LED\_FLAG[hmi\_key\_value-1] == 0)

{

if(hmi\_key\_value == 0x01)

{

HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_10, GPIO\_PIN\_RESET);//开LED1

}

if(hmi\_key\_value == 0x02)

{

HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_3, GPIO\_PIN\_RESET); //开LED2

}

if(hmi\_key\_value == 0x03)

{

HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_2, GPIO\_PIN\_RESET); //开LED3

}

if(hmi\_key\_value == 0x04)

{

HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_0, GPIO\_PIN\_RESET); //开LED4

}

if(hmi\_key\_value == 0x05)

{

HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_13, GPIO\_PIN\_RESET); //开LED5

}

LED\_FLAG[hmi\_key\_value-1] = 1;

}

else if(LED\_FLAG[hmi\_key\_value-1] == 1)

{

if(hmi\_key\_value == 0x01)

{

HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_10, GPIO\_PIN\_SET);//关LED1

}

if(hmi\_key\_value == 0x02)

{

HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_3, GPIO\_PIN\_SET); //关LED2

}

if(hmi\_key\_value == 0x03)

{

HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_2, GPIO\_PIN\_SET); //关LED3

}

if(hmi\_key\_value == 0x04)

{

HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_0, GPIO\_PIN\_SET); //关LED4

}

if(hmi\_key\_value == 0x05)

{

HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_13, GPIO\_PIN\_SET); //关LED5

}

LED\_FLAG[hmi\_key\_value-1] = 0;

}

for(i=0;i<9;i++)

{

usart1\_rxBuf[i] = 0xff;

}

for(i=0;i<5;i++)

{

HMI\_SEND\_BUF[i] = LED\_FLAG[i];

}

PLC\_Data\_Wr2HMI(0X1000,0,5);//发送LED状态到触摸屏，改变LED显示图标；

delay\_ms(50);

}

}

6). CLOSE\_OPEN\_IO ():开关LED

void SCAN\_KEY(uint8\_t io\_key\_value)

{

uint8\_t i;

if(HAL\_GPIO\_ReadPin(GPIOB,GPIO\_PIN\_10)==GPIO\_PIN\_RESET)//KEY1按下

{

delay\_ms(150);//去抖

if(HAL\_GPIO\_ReadPin(GPIOB,GPIO\_PIN\_10)==GPIO\_PIN\_RESET)

{

if(HMI\_FLAG[0] == 0)

{

WRITE\_HMI(0X82,0x00,0x00,0,0,1);//熄灭屏幕

HMI\_FLAG[0] = 1;

}else if(HMI\_FLAG[0] == 1)

{

WRITE\_HMI(0X82,0X64,0x00,0,0,1);//点亮屏幕

HMI\_FLAG[0] = 0;

}

}

for(i=0;i<50;i++)

{

usart1\_rxBuf[i] = 0x00;

}

usart1\_counter = 0;

}

if(HAL\_GPIO\_ReadPin(GPIOB,GPIO\_PIN\_11)==GPIO\_PIN\_RESET)//KEY2按下

{

delay\_ms(150);//去抖

if(HAL\_GPIO\_ReadPin(GPIOB,GPIO\_PIN\_11)==GPIO\_PIN\_RESET)

{

TEMP += 1;

HMI\_SEND\_BUF[0] = TEMP;

PLC\_Data\_Wr2HMI(0X1110,0,1);//温度+

}

for(i=0;i<50;i++)

{

usart1\_rxBuf[i] = 0x00;

}

usart1\_counter = 0;

}

if(HAL\_GPIO\_ReadPin(GPIOB,GPIO\_PIN\_12)==GPIO\_PIN\_RESET)//KEY3按下

{

delay\_ms(150);//去抖

if(HAL\_GPIO\_ReadPin(GPIOB,GPIO\_PIN\_12)==GPIO\_PIN\_RESET)

{

TEMP -= 1;

HMI\_SEND\_BUF[0] = TEMP;

PLC\_Data\_Wr2HMI(0X1110,0,1);//温度-

}

for(i=0;i<50;i++)

{

usart1\_rxBuf[i] = 0x00;

}

usart1\_counter = 0;

}

if(HAL\_GPIO\_ReadPin(GPIOB,GPIO\_PIN\_13)==GPIO\_PIN\_RESET)//KEY4按下

{

delay\_ms(150);//去抖

if(HAL\_GPIO\_ReadPin(GPIOB,GPIO\_PIN\_13)==GPIO\_PIN\_RESET)

{

if(PLAY == 0)

{

WRITE\_HMI(0X00A0,0X00,0x0E,0,0,1);

PLAY =1;//播放

}else if(PLAY == 1)

{

WRITE\_HMI(0X00A0,0X00,0x00,0,0,1);

PLAY =0;//停止

}

}

for(i=0;i<50;i++)

{

usart1\_rxBuf[i] = 0x00;

}

usart1\_counter = 0;

}

if(HAL\_GPIO\_ReadPin(GPIOB,GPIO\_PIN\_14)==GPIO\_PIN\_RESET)//KEY5按

{

delay\_ms(150);//去抖

if(HAL\_GPIO\_ReadPin(GPIOB,GPIO\_PIN\_14)==GPIO\_PIN\_RESET)

{

if(VOLUME<250)

{

VOLUME += 10;

WRITE\_HMI(0X00A1,VOLUME,0x00,0,0,2);//音量+

delay\_ms(50);

HMI\_SEND\_BUF[0] = VOLUME;

PLC\_Data\_Wr2HMI(0X1210,0,1);//写当前音量到触摸屏

delay\_ms(50);

}

}

for(i=0;i<50;i++)

{

usart1\_rxBuf[i] = 0x00;

}

usart1\_counter = 0;

}

if(HAL\_GPIO\_ReadPin(GPIOB,GPIO\_PIN\_15)==GPIO\_PIN\_RESET)//KEY6按下

{

delay\_ms(150);//去抖

if(HAL\_GPIO\_ReadPin(GPIOB,GPIO\_PIN\_15)==GPIO\_PIN\_RESET)

{

if(VOLUME >= 10)

{

VOLUME -= 10;

WRITE\_HMI(0X00A1,VOLUME,0x00,0,0,2);//音量-

delay\_ms(50);

HMI\_SEND\_BUF[0] = VOLUME;

PLC\_Data\_Wr2HMI(0X1210,0,1);//写当前音量到触摸屏

delay\_ms(50);

}

}

for(i=0;i<50;i++)

{

usart1\_rxBuf[i] = 0x00;

}

usart1\_counter = 0;

}

}

7). delay\_ms ():延时函数，不精确慎用

void delay\_ms(uint16\_t nms)

{

uint16\_t i,j;

for(i=0;i<nms;i++)

{

for(j=0;j<10000;j++)

{

;

}

}

}

8). USART1\_IRQHandler ():数据接收中断

void USART1\_IRQHandler(void)

{

HAL\_UART\_IRQHandler(&huart1);

if(usart1\_rxBuf[0] == 0x5a&&usart1\_rxBuf[1] == 0xa5&&usart1\_rxBuf[2])

{

if(usart1\_counter >= 3&&usart1\_counter<(usart1\_rxBuf[2]+3))

{

usart1\_rxBuf[usart1\_counter] = usart1\_data;

usart1\_counter++;

}

if(usart1\_counter >= 3&&usart1\_counter>=(usart1\_rxBuf[2]+3))

{

HMI\_HANDLEFLAG = 1;//接收完成标志

}

}

if(usart1\_counter < 3)

{

usart1\_rxBuf[usart1\_counter] = usart1\_data;

usart1\_counter++;

}

if(HMI\_HANDLEFLAG)

{

usart1\_counter = 0;

}

HAL\_UART\_Receive\_IT(&huart1,&usart1\_data,1);//开接收中断

}