

---

主函数程序部分

```
#include "stm32f10x.h"

#include "sys.h"

#include "delay.h"

#include "syn6288.h"//语音模块头文件

#include "led.h"

#include "oled.h"//OLED 显示屏模块头文件

#include "key.h"//按键头文件

u8 i=0;

int main()

{

    /*****各个模块初始化函数*****/

    delay_init();          //延时函数初始化

    syn6288_Init();        //syn6288 语音模块初始化

    NVIC_Configuration();  //设置 NVIC 中断分组 2:2 位抢占优先级，2 位响应优先级

    LED_Init();            //LED 端口初始化

    OLED_Init();           //初始化 OLED

    LED_Init();//LED 初始化

    KEY_Init();//按键初始化

    //////////////////////////////////////

    /*****OLED 初始化界面*****/

    OLED_Clear();//oled 清屏语句

    OLED_ShowChinese(30,3,0);//语

    OLED_ShowChinese(48,3,1);//音

    OLED_ShowChinese(66,3,2);//播

    OLED_ShowChinese(84,3,3);//报

    delay_ms(8000);

    delay_ms(8000);

    OLED_Clear();

    OLED_ShowChinese(0,0,4);

    OLED_ShowChinese(18,0,5);//语文
```

---

```
OLED_ShowChinese(90,0,6);

    OLED_ShowChinese(108,0,7);//数学

    OLED_ShowChinese(0,5,8);

    OLED_ShowChinese(18,5,9);//英语

OLED_ShowChinese(90,5,10);

    OLED_ShowChinese(108,5,11);//政治

    delay_ms(8000);

delay_ms(8000);
```

```
while(1)
{
    /*****按键检测*****/

    if(!S1)//如果按下第一个按键

    {

        delay_ms(10);//延时消抖

        if(!S1)

            {

                while(!S1);//等待按键释放

                syn6288_SpeakStr("语文",1);//播报语文

            }

    }

    if(!S2)//如果按下第二个按键

    {

        delay_ms(10);//延时消抖

        if(!S2)    //如果按下第二个按键

            {

                while(!S2);//等待按键释放

                syn6288_SpeakStr("数学",1);//播报数学

            }

    }

    if(!S3)//如果按下第三个按键
```

---

```
{  
  
    delay_ms(10); //延时消抖  
  
    if(!S3)      //如果按下第三个按键  
  
    {  
  
        while(!S3); //等待按键释放  
  
        syn6288_SpeakStr("英语",1); //播报英语  
  
    }  
}  
  
if(!S4)  
{  
  
    delay_ms(10);  
  
    if(!S4)  
  
    {  
  
        while(!S4);  
  
        syn6288_SpeakStr("政治",1);  
  
    }  
}  
  
}  
}
```

OLED 驱动程序:

```
#include "oled.h"  
  
#include "stdlib.h"  
  
#include "oledfont.h"  
  
#include "delay.h"  
  
#if OLED_MODE==1  
  
void OLED_WR_Byte(u8 dat,u8 cmd)  
{  
  
    DATAOUT(dat);  
  
    if(cmd)  
  
        OLED_DC_Set();  
}
```

---

```
else

    OLED_DC_Clr();

    OLED_CS_Clr();

    OLED_WR_Clr();

    OLED_WR_Set();

    OLED_CS_Set();

    OLED_DC_Set();

}

#else

void OLED_WR_Byte(u8 dat,u8 cmd)

{

    u8 i;

    if(cmd)

        OLED_DC_Set();

    else

        OLED_DC_Clr();

    OLED_CS_Clr();

    for(i=0;i<8;i++)

    {

        OLED_SCLK_Clr();

        if(dat&0x80)

            OLED_SDIN_Set();

        else

            OLED_SDIN_Clr();

        OLED_SCLK_Set();

        dat<<=1;

    }

    OLED_CS_Set();

    OLED_DC_Set();

}

#endif

void OLED_ShowCHinese(u8 x,u8 y,u8 no)

{

    u8 t,adder=0;
```

---

```
OLED_Set_Pos(x,y);
for(t=0;t<16;t++)
{
    OLED_WR_Byte(Hzk[2*no][t],OLED_DATA);
    adder+=1;
}
OLED_Set_Pos(x,y+1);
for(t=0;t<16;t++)
{
    OLED_WR_Byte(Hzk[2*no+1][t],OLED_DATA);
    adder+=1;
}
}

void OLED_Init(void)
{
    GPIO_InitTypeDef  GPIO_InitStructure;

    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB|RCC_APB2Periph_GPIOD|RCC_APB2Periph_GPIOG,
ENABLE);  //使能 PC,D,G 端口时钟

    GPIO_InitStructure.GPIO_Pin=
GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_6|GPIO_Pin_7|GPIO_Pin_3|GPIO_Pin_8;  //PD3,PD6 推挽输出

    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;      //推挽输出

    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;//速度 50MHz

    GPIO_Init(GPIOD, &GPIO_InitStructure);  //初始化 GPIOD3,6

    GPIO_SetBits(GPIOD,GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_6|GPIO_Pin_7|GPIO_Pin_3|GPIO_Pin_8);

    #if OLED_MODE==1

    GPIO_InitStructure.GPIO_Pin =0xFF; //PC0~7 OUT 推挽输出

    GPIO_Init(GPIOC, &GPIO_InitStructure);

    GPIO_SetBits(GPIOC,0xFF); //PC0~7 输出高

    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_13|GPIO_Pin_14|GPIO_Pin_15;

    GPIO_Init(GPIOG, &GPIO_InitStructure);

    GPIO_SetBits(GPIOG,GPIO_Pin_13|GPIO_Pin_14|GPIO_Pin_15);
```

---

```
OLED_RST_Set();

delay_ms(100);

OLED_RST_Clr();

delay_ms(100);

OLED_RST_Set();


OLED_WR_Byte(0xAE,OLED_CMD);//--turn off oled panel

OLED_WR_Byte(0x00,OLED_CMD);//---set low column address

OLED_WR_Byte(0x10,OLED_CMD);//---set high column address

OLED_WR_Byte(0x40,OLED_CMD);//--set start line address   Set Mapping RAM Display Start Line
(0x00~0x3F)

OLED_WR_Byte(0x81,OLED_CMD);//--set contrast control register

OLED_WR_Byte(0xCF,OLED_CMD); // Set SEG Output Current Brightness

OLED_WR_Byte(0xA1,OLED_CMD);//--Set SEG/Column Mapping      0xa0 左右反置 0xa1 正常

OLED_WR_Byte(0xC8,OLED_CMD);//Set COM/Row Scan Direction     0xc0 上下反置 0xc8 正常

OLED_WR_Byte(0xA6,OLED_CMD);//--set normal display

OLED_WR_Byte(0xA8,OLED_CMD);//--set multiplex ratio(1 to 64)

OLED_WR_Byte(0x3f,OLED_CMD);//--1/64 duty

OLED_WR_Byte(0xD3,OLED_CMD);//--set display offset   Shift Mapping RAM Counter (0x00~0x3F)

OLED_WR_Byte(0x00,OLED_CMD);//--not offset

OLED_WR_Byte(0xd5,OLED_CMD);//--set display clock divide ratio/oscillator frequency

OLED_WR_Byte(0x80,OLED_CMD);//--set divide ratio, Set Clock as 100 Frames/Sec

OLED_WR_Byte(0xD9,OLED_CMD);//--set pre-charge period

OLED_WR_Byte(0xF1,OLED_CMD);//Set Pre-Charge as 15 Clocks & Discharge as 1 Clock

OLED_WR_Byte(0xDA,OLED_CMD);//--set com pins hardware configuration

OLED_WR_Byte(0x12,OLED_CMD);

OLED_WR_Byte(0xDB,OLED_CMD);//--set vcomh

OLED_WR_Byte(0x40,OLED_CMD);//Set VCOM Deselect Level

OLED_WR_Byte(0x20,OLED_CMD);//-Set Page Addressing Mode (0x00/0x01/0x02)

OLED_WR_Byte(0x02,OLED_CMD);//

OLED_WR_Byte(0x8D,OLED_CMD);//--set Charge Pump enable/disable

OLED_WR_Byte(0x14,OLED_CMD);//--set(0x10) disable

OLED_WR_Byte(0xA4,OLED_CMD);// Disable Entire Display On (0xa4/0xa5)

OLED_WR_Byte(0xA6,OLED_CMD);// Disable Inverse Display On (0xa6/a7)
```

---

```
OLED_WR_Byte(0xAF,OLED_CMD);/--turn on oled panel
```

```
OLED_WR_Byte(0xAF,OLED_CMD); /*display ON*/
```

```
OLED_Clear();
```

```
OLED_Set_Pos(0,0);
```

```
}
```

SYN6288 语音播报模块驱动程序

```
void USART1_Init(void)
```

```
{
```

```
GPIO_InitTypeDef GPIO_InitStructure;
```

```
USART_InitTypeDef USART_InitStructure;    RCC_APB2PeriphClockCmd(RCC_APB2Periph_USART1 |  
RCC_APB2Periph_GPIOA, ENABLE);
```

```
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_9;
```

```
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
```

```
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
```

```
GPIO_Init(GPIOA, &GPIO_InitStructure);
```

```
/* Configure USART1 Rx (PA.10) as input floating */
```

```
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_10;
```

```
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
```

```
GPIO_Init(GPIOA, &GPIO_InitStructure);
```

```
USART_InitStructure.USART_BaudRate = 9600;
```

```
USART_InitStructure.USART_WordLength = USART_WordLength_8b;
```

```
USART_InitStructure.USART_StopBits = USART_StopBits_1;
```

```
USART_InitStructure.USART_Parity    = USART_Parity_No ;
```

```
USART_InitStructure.USART_HardwareFlowControl = USART_HardwareFlowControl_None;
```

```
USART_InitStructure.USART_Mode = USART_Mode_Rx | USART_Mode_Tx;
```

```
USART_Init(USART1, &USART_InitStructure);
```

```
    USART_ITConfig(USART1,USART_IT_RXNE,ENABLE); //使能中断
```

```
NVIC_InitStructure.NVIC_IRQChannel = USART1_IRQn;
```

```
NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 2;
```

```
NVIC_InitStructure.NVIC_IRQChannelSubPriority = 1;
```

```
NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
```

```
NVIC_Init(&NVIC_InitStructure);
```

---

```
    USART_Cmd(USART1, ENABLE);
}

void syn6288_SpeakStr(char *text,char MusicID)    //MusicID 为需要播放的背景音乐变化，0 不播放
{
    char FrameHead[5];

    int  cnt=0;

    char length=0;

    char check=0;

    length=strlen(text);

    FrameHead[0]=0xfd;
    FrameHead[1]=0x00;
    FrameHead[2]=length+3;
    FrameHead[3]=01;
    FrameHead[4]=00+(MusicID<<3); //GB2312

    for(cnt=0;cnt<5;cnt++)
    {
        check=check^(FrameHead[cnt]);

        syn6288_SendByte(FrameHead[cnt]);
    }

    for(cnt=0;cnt<length;cnt++)
    {
        check=check^(*text);

        syn6288_SendByte(*text);

        text++;
    }

    syn6288_SendByte(check);

    syn_state=1;

    while(syn_state);

    syn_Delaynms(8);
```



---

```
}
```

```
void USART1_IRQHandler()
```

```
{
```

```
if(USART_GetITStatus(USART1,USART_IT_RXNE)!= RESET) //确保是否产生了中断
```

```
{
```

```
    u2reccdat=USART_ReceiveData(USART1);
```

```
    if(u2reccdat==SYN6288_ACK_IDLE)
```

```
    {
```

```
        LED0=1;
```

```
        syn_state=0;//idle
```

```
    }
```

```
}
```

```
}
```