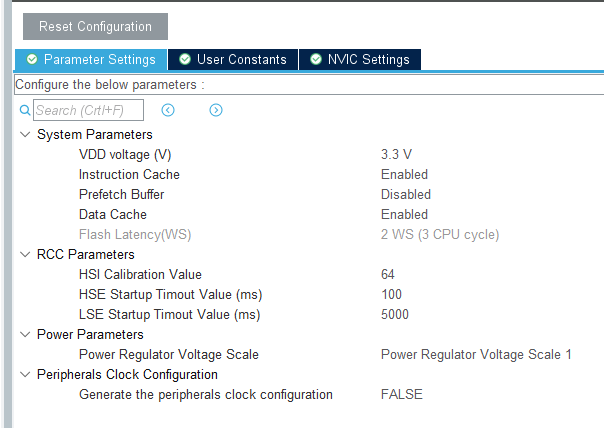
# RTC与IIC的电子钟设计

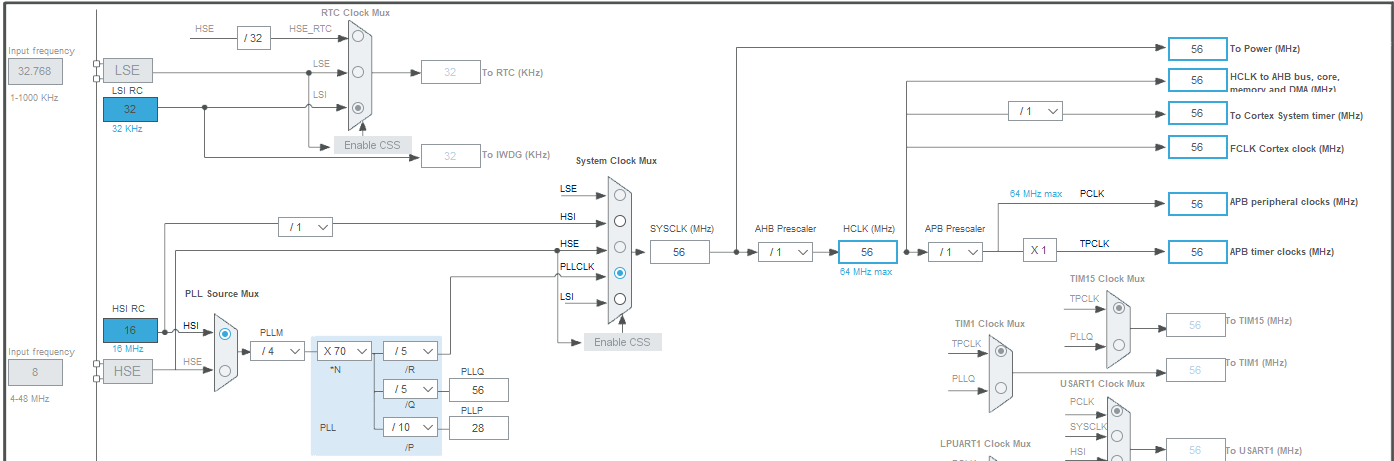
## 创建工程

将PA5引脚设置为GPIO\_OUTPUT模式用以作为提示灯。

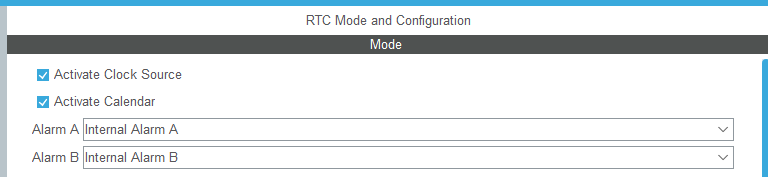
将RCC如下设置更改为“FALSE”



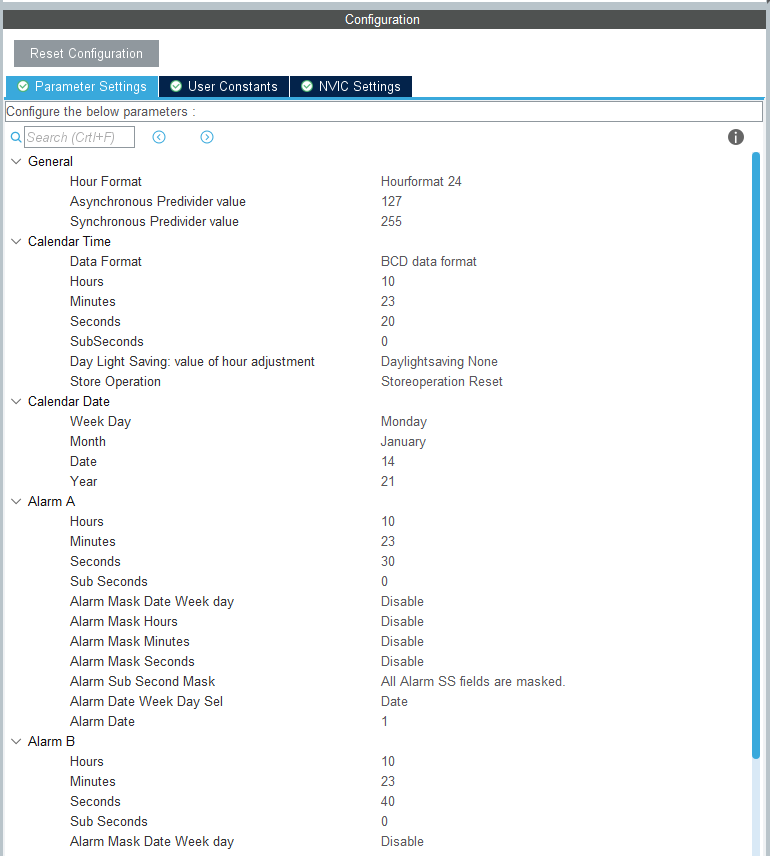
按照下图所示配置系统时钟

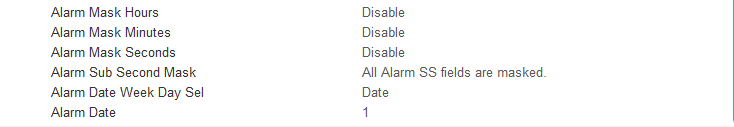


RTC的日历、时钟、闹钟A、闹钟B开启

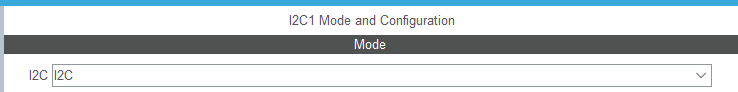


RTC其他参数配置如下图所示：

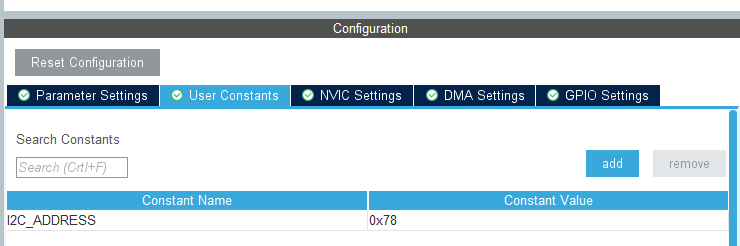




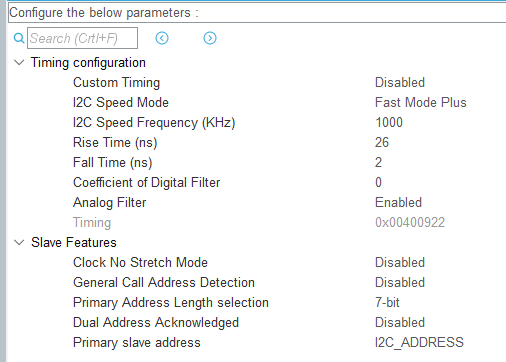
I2C1配置：其模式选择为IIC模式



添加从设备（OLED显示屏地址标号）



其他IIC设置如下：



添加管脚映射：将IIC\_SDA映射在PB8引脚，将IIC\_SCL映射在PB6引脚。

创建工程

## stm32g0xx\_hal\_msp.c文件修改

修改如下函数：

void HAL\_RTC\_MspInit(RTC\_HandleTypeDef\* hrtc)

{

if(hrtc->Instance==RTC)

{

RCC\_OscInitTypeDef RCC\_OscInitStruct = {0};

RCC\_PeriphCLKInitTypeDef PeriphClkInitStruct = {0};

\_\_HAL\_RCC\_PWR\_CLK\_ENABLE();

HAL\_PWR\_EnableBkUpAccess();

HAL\_RCCEx\_GetPeriphCLKConfig(&PeriphClkInitStruct);

if (PeriphClkInitStruct.RTCClockSelection == RtcClockSource)

{ }

else

{

PeriphClkInitStruct.PeriphClockSelection = RCC\_PERIPHCLK\_RTC;

if (PeriphClkInitStruct.RTCClockSelection != RCC\_RTCCLKSOURCE\_NONE)

{

PeriphClkInitStruct.RTCClockSelection = RCC\_RTCCLKSOURCE\_NONE;

if (HAL\_RCCEx\_PeriphCLKConfig(&PeriphClkInitStruct) != HAL\_OK)

Error\_Handler();

}

RCC\_OscInitStruct.OscillatorType = RCC\_OSCILLATORTYPE\_LSI | RCC\_OSCILLATORTYPE\_LSE;

RCC\_OscInitStruct.PLL.PLLState = RCC\_PLL\_NONE;

RCC\_OscInitStruct.LSIState = RCC\_LSI\_ON;

RCC\_OscInitStruct.LSEState = RCC\_LSE\_OFF;

if (HAL\_RCC\_OscConfig(&RCC\_OscInitStruct) != HAL\_OK)

Error\_Handler();

PeriphClkInitStruct.RTCClockSelection = RtcClockSource;

if (HAL\_RCCEx\_PeriphCLKConfig(&PeriphClkInitStruct) != HAL\_OK)

Error\_Handler();

}

\_\_HAL\_RCC\_RTC\_ENABLE();

\_\_HAL\_RCC\_RTCAPB\_CLK\_ENABLE();

HAL\_NVIC\_SetPriority(RTC\_TAMP\_IRQn, 0, 0);

HAL\_NVIC\_EnableIRQ(RTC\_TAMP\_IRQn);

}

}

void HAL\_RTC\_MspDeInit(RTC\_HandleTypeDef\* hrtc)

{

if(hrtc->Instance==RTC)

{

/\* Peripheral clock disable \*/

\_\_HAL\_RCC\_RTC\_DISABLE();

\_\_HAL\_RCC\_RTCAPB\_CLK\_DISABLE();

/\* RTC interrupt DeInit \*/

HAL\_NVIC\_DisableIRQ(RTC\_TAMP\_IRQn);

}

}

## main.c文件中编写代码

### 变量定义与函数声明

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*RTC相关变量

#define Init\_Time\_Year 0x21

#define Init\_Time\_Month 0x06

#define Init\_Time\_Date 0x014

#define Init\_Time\_Hours 0x08

#define Init\_Time\_Minutes 0x20

#define Init\_Time\_Seconds 0x00

#define Init\_Time\_SubSeconds 0x00

#define Alarm\_A\_Hours 0x08

#define Alarm\_A\_Minutes 0x20

#define Alarm\_A\_Seconds 0x10

#define Alarm\_A\_SubSeconds 0x00

#define Alarm\_B\_Hours 0x08

#define Alarm\_B\_Minutes 0x20

#define Alarm\_B\_Seconds 0x20

#define Alarm\_B\_SubSeconds 0x00

#include <stdio.h>

uint8\_t Flag\_Alarm = 0; //闹钟标志置位，判定是否处于闹钟模式

uint8\_t Flag\_Second\_Old = 0; //降低OLED刷新速率--------------------避免卡BUG

uint8\_t aShowTime[8] = "hh:ms:ss";

uint8\_t aShowDate[10] = "dd-mm-yyyy";

static void RTC\_CalendarShow(uint8\_t \*showtime, uint8\_t \*showdate);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*IIC的数据缓冲区与基层函数

#define TXBUFFERSIZE 2

unsigned char aTxBuffer\_Command[TXBUFFERSIZE] = {0x00,0x00};

unsigned char aTxBuffer\_Data[TXBUFFERSIZE] = {0x40,0x00};

void Write\_IIC\_Command(unsigned char IIC\_Command);

void Write\_IIC\_Data(unsigned char IIC\_Data);

void Initial\_M096128x64\_ssd1306(void);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*IIC的图片显示

const unsigned char biaoqinbao[][128] =

{

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X80,0X80,0XC0,0XC0,0X60,0X60,0X30,0X30,0X30,

0X30,0X30,0X30,0X30,0X30,0X30,0X30,0X30,0XB0,0X30,0X20,0X60,0X60,0XC0,0XC0,0X80,

0X80,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X80,0XE0,0X78,0X1C,0X0E,0X07,0X03,0X01,0X00,0X80,0X03,0X03,0X01,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X80,0X80,0X01,0X03,0X03,0X00,0X00,0X00,0X00,0X00,0X01,

0X03,0X07,0X0E,0X1C,0X70,0XE0,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0XF0,0XF8,

0X0E,0X07,0X01,0X08,0X1C,0X1C,0X1C,0X1C,0X1C,0X0F,0X07,0X33,0X30,0X60,0X60,0X60,

0X60,0X60,0X60,0X30,0X33,0X03,0X0F,0X0D,0X0C,0X0C,0X0C,0X0C,0X0C,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X87,0XFF,0X30,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X07,0X1F,

0X38,0X70,0X60,0XC0,0XC0,0X80,0X80,0X80,0X80,0X80,0X20,0X60,0X40,0XC0,0XC0,0X80,

0XC0,0XC0,0XC0,0X60,0X30,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X80,0X80,0X80,

0XC0,0X60,0X70,0X38,0X1C,0X0F,0X03,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X80,0XC0,0XE0,0X70,0X1C,0X8F,0XC3,0X61,0X31,0X19,0X08,0X00,0X80,0XC0,0X61,0X39,

0X1D,0X01,0X81,0XE0,0X70,0X38,0X00,0X00,0XF0,0XFE,0X0F,0X03,0X03,0X01,0X01,0X01,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0XE0,0XE0,0XF0,0XF8,0XFE,0XE6,

0XC3,0XC7,0XC6,0X83,0X83,0XF9,0XFC,0X86,0X86,0X82,0X82,0X07,0X0D,0X0C,0X9C,0X9E,

0X9F,0X9B,0X99,0XB0,0XF0,0XF0,0XF8,0XFF,0XF7,0XF0,0XE0,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X01,0X01,0X03,0X03,0X03,

0X07,0X07,0X07,0X07,0X07,0X07,0X07,0X07,0X07,0X07,0X07,0X07,0X07,0X07,0X07,0X07,

0X07,0X07,0X07,0X07,0X03,0X03,0X03,0X03,0X01,0X01,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

};

unsigned char show[][128]=

{

{0x00,0x00,0x06,0x0A,0xFE,0x0A,0xC6,0x00,0xE0,0x00,0xF0,0x00,0xF8,0x00,0x00,0x00,

0x00,0x00,0x00,0xFE,0x7D,0xBB,0xC7,0xEF,0xEF,0xEF,0xEF,0xEF,0xEF,0xEF,0xC7,0xBB,

0x7D,0xFE,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x08,0x0C,0xFE,0xFE,0x0C,0x08,0x20,0x60,0xFE,0xFE,0x60,0x20,0x00,0x00,0x00,0x78,

0x48,0xFE,0x82,0xBA,0xBA,0x82,0xBA,0xBA,0x82,0xBA,0xBA,0x82,0xBA,0xBA,0x82,0xFE},

{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,

0x01,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},

{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0xFE,0xFF,0x03,0x03,0x03,0x03,0x03,0x03,0x03,0x03,0x03,0xFF,0xFF,0x00,0x00,0xFE,

0xFF,0x03,0x03,0x03,0x03,0x03,0x03,0x03,0x03,0x03,0xFF,0xFE,0x00,0x00,0x00,0x00,

0xC0,0xC0,0xC0,0x00,0x00,0x00,0x00,0xFE,0xFF,0x03,0x03,0x03,0x03,0x03,0x03,0x03,

0x03,0x03,0xFF,0xFE,0x00,0x00,0xFE,0xFF,0x03,0x03,0x03,0x03,0x03,0x03,0x03,0x03,

0x03,0xFF,0xFE,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},

{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0xFF,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,0xFF,0x00,0x00,0xFF,

0xFF,0x0C,0x0C,0x0C,0x0C,0x0C,0x0C,0x0C,0x0C,0x0C,0xFF,0xFF,0x00,0x00,0x00,0x00,

0xE1,0xE1,0xE1,0x00,0x00,0x00,0x00,0xFF,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0xFF,0xFF,0x00,0x00,0xFF,0xFF,0x0C,0x0C,0x0C,0x0C,0x0C,0x0C,0x0C,0x0C,

0x0C,0xFF,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},

{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x0F,0x1F,0x18,0x18,0x18,0x18,0x18,0x18,0x18,0x18,0x18,0x1F,0x0F,0x00,0x00,0x0F,

0x1F,0x18,0x18,0x18,0x18,0x18,0x18,0x18,0x18,0x18,0x1F,0x0F,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0x1F,0x18,0x18,0x18,0x18,0x18,0x18,0x18,

0x18,0x18,0x1F,0x0F,0x00,0x00,0x0F,0x1F,0x18,0x18,0x18,0x18,0x18,0x18,0x18,0x18,

0x18,0x1F,0x0F,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},

{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xE2,0x92,0x8A,0x86,0x00,0x00,0x7C,0x82,0x82,

0x82,0x7C,0x00,0xFE,0x00,0x82,0x92,0xAA,0xC6,0x00,0x00,0xC0,0xC0,0x00,0x7C,0x82,

0x82,0x82,0x7C,0x00,0x00,0x02,0x02,0x02,0xFE,0x00,0x00,0xC0,0xC0,0x00,0x7C,0x82,

0x82,0x82,0x7C,0x00,0x00,0xFE,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},

{0x00,0x00,0x00,0x24,0xA4,0x2E,0x24,0xE4,0x24,0x2E,0xA4,0x24,0x00,0x00,0x00,0xF8,

0x4A,0x4C,0x48,0xF8,0x48,0x4C,0x4A,0xF8,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xC0,0x20,

0x10,0x10,0x10,0x10,0x20,0xC0,0x00,0x00,0xC0,0x20,0x10,0x10,0x10,0x10,0x20,0xC0},

{0x00,0x00,0x00,0x12,0x0A,0x07,0x02,0x7F,0x02,0x07,0x0A,0x02,0x00,0x00,0x00,0x0B,

0x0A,0x0A,0x0A,0x7F,0x0A,0x0A,0x0A,0x0B,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1F,0x20,

0x40,0x40,0x40,0x50,0x20,0x5F,0x80,0x00,0x1F,0x20,0x40,0x40,0x40,0x50,0x20,0x5F}

};

void fill\_picture(unsigned char fill\_Data);

void Picture(int i);

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*IIC的字符显示

const unsigned char L8H16[][8]=

{

{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},

{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},//space 0

{0x00,0x00,0x00,0xF8,0x00,0x00,0x00,0x00},

{0x00,0x00,0x00,0x33,0x30,0x00,0x00,0x00},//! 1

{0x00,0x10,0x0C,0x06,0x10,0x0C,0x06,0x00},

{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},//" 2

{0x40,0xC0,0x78,0x40,0xC0,0x78,0x40,0x00},

{0x04,0x3F,0x04,0x04,0x3F,0x04,0x04,0x00},//# 3

{0x00,0x70,0x88,0xFC,0x08,0x30,0x00,0x00},

{0x00,0x18,0x20,0xFF,0x21,0x1E,0x00,0x00},//$ 4

{0xF0,0x08,0xF0,0x00,0xE0,0x18,0x00,0x00},

{0x00,0x21,0x1C,0x03,0x1E,0x21,0x1E,0x00},//% 5

{0x00,0xF0,0x08,0x88,0x70,0x00,0x00,0x00},

{0x1E,0x21,0x23,0x24,0x19,0x27,0x21,0x10},//& 6

{0x10,0x16,0x0E,0x00,0x00,0x00,0x00,0x00},

{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},//' 7

{0x00,0x00,0x00,0xE0,0x18,0x04,0x02,0x00},

{0x00,0x00,0x00,0x07,0x18,0x20,0x40,0x00},//( 8

{0x00,0x02,0x04,0x18,0xE0,0x00,0x00,0x00},

{0x00,0x40,0x20,0x18,0x07,0x00,0x00,0x00},//) 9

{0x40,0x40,0x80,0xF0,0x80,0x40,0x40,0x00},

{0x02,0x02,0x01,0x0F,0x01,0x02,0x02,0x00},//\* 10

{0x00,0x00,0x00,0xF0,0x00,0x00,0x00,0x00},

{0x01,0x01,0x01,0x1F,0x01,0x01,0x01,0x00},//+ 11

{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},

{0x80,0xB0,0x70,0x00,0x00,0x00,0x00,0x00},//, 12

{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},

{0x00,0x01,0x01,0x01,0x01,0x01,0x01,0x01},//- 13

{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},

{0x00,0x30,0x30,0x00,0x00,0x00,0x00,0x00},//. 14

{0x00,0x00,0x00,0x00,0x80,0x60,0x18,0x04},

{0x00,0x60,0x18,0x06,0x01,0x00,0x00,0x00},/// 15

{0x00,0xE0,0x10,0x08,0x08,0x10,0xE0,0x00},

{0x00,0x0F,0x10,0x20,0x20,0x10,0x0F,0x00},//0 16

{0x00,0x10,0x10,0xF8,0x00,0x00,0x00,0x00},

{0x00,0x20,0x20,0x3F,0x20,0x20,0x00,0x00},//1 17

{0x00,0x70,0x08,0x08,0x08,0x88,0x70,0x00},

{0x00,0x30,0x28,0x24,0x22,0x21,0x30,0x00},//2 18

{0x00,0x30,0x08,0x88,0x88,0x48,0x30,0x00},

{0x00,0x18,0x20,0x20,0x20,0x11,0x0E,0x00},//3 19

{0x00,0x00,0xC0,0x20,0x10,0xF8,0x00,0x00},

{0x00,0x07,0x04,0x24,0x24,0x3F,0x24,0x00},//4 20

{0x00,0xF8,0x08,0x88,0x88,0x08,0x08,0x00},

{0x00,0x19,0x21,0x20,0x20,0x11,0x0E,0x00},//5 21

{0x00,0xE0,0x10,0x88,0x88,0x18,0x00,0x00},

{0x00,0x0F,0x11,0x20,0x20,0x11,0x0E,0x00},//6 22

{0x00,0x38,0x08,0x08,0xC8,0x38,0x08,0x00},

{0x00,0x00,0x00,0x3F,0x00,0x00,0x00,0x00},//7 23

{0x00,0x70,0x88,0x08,0x08,0x88,0x70,0x00},

{0x00,0x1C,0x22,0x21,0x21,0x22,0x1C,0x00},//8 24

{0x00,0xE0,0x10,0x08,0x08,0x10,0xE0,0x00},

{0x00,0x00,0x31,0x22,0x22,0x11,0x0F,0x00},//9 25

{0x00,0x00,0x00,0xC0,0xC0,0x00,0x00,0x00},

{0x00,0x00,0x00,0x30,0x30,0x00,0x00,0x00},//: 26

{0x00,0x00,0x00,0x80,0x00,0x00,0x00,0x00},

{0x00,0x00,0x80,0x60,0x00,0x00,0x00,0x00},//; 27

{0x00,0x00,0x80,0x40,0x20,0x10,0x08,0x00},

{0x00,0x01,0x02,0x04,0x08,0x10,0x20,0x00},//< 28

{0x40,0x40,0x40,0x40,0x40,0x40,0x40,0x00},

{0x04,0x04,0x04,0x04,0x04,0x04,0x04,0x00},//= 29

{0x00,0x08,0x10,0x20,0x40,0x80,0x00,0x00},

{0x00,0x20,0x10,0x08,0x04,0x02,0x01,0x00},//> 30

{0x00,0x70,0x48,0x08,0x08,0x08,0xF0,0x00},

{0x00,0x00,0x00,0x30,0x36,0x01,0x00,0x00},//? 31

{0xC0,0x30,0xC8,0x28,0xE8,0x10,0xE0,0x00},

{0x07,0x18,0x27,0x24,0x23,0x14,0x0B,0x00},//@ 32

{0x00,0x00,0xC0,0x38,0xE0,0x00,0x00,0x00},

{0x20,0x3C,0x23,0x02,0x02,0x27,0x38,0x20},//A 33

{0x08,0xF8,0x88,0x88,0x88,0x70,0x00,0x00},

{0x20,0x3F,0x20,0x20,0x20,0x11,0x0E,0x00},//B 34

{0xC0,0x30,0x08,0x08,0x08,0x08,0x38,0x00},

{0x07,0x18,0x20,0x20,0x20,0x10,0x08,0x00},//C 35

{0x08,0xF8,0x08,0x08,0x08,0x10,0xE0,0x00},

{0x20,0x3F,0x20,0x20,0x20,0x10,0x0F,0x00},//D 36

{0x08,0xF8,0x88,0x88,0xE8,0x08,0x10,0x00},

{0x20,0x3F,0x20,0x20,0x23,0x20,0x18,0x00},//E 37

{0x08,0xF8,0x88,0x88,0xE8,0x08,0x10,0x00},

{0x20,0x3F,0x20,0x00,0x03,0x00,0x00,0x00},//F 38

{0xC0,0x30,0x08,0x08,0x08,0x38,0x00,0x00},

{0x07,0x18,0x20,0x20,0x22,0x1E,0x02,0x00},//G 39

{0x08,0xF8,0x08,0x00,0x00,0x08,0xF8,0x08},

{0x20,0x3F,0x21,0x01,0x01,0x21,0x3F,0x20},//H 40

{0x00,0x08,0x08,0xF8,0x08,0x08,0x00,0x00},

{0x00,0x20,0x20,0x3F,0x20,0x20,0x00,0x00},//I 41

{0x00,0x00,0x08,0x08,0xF8,0x08,0x08,0x00},

{0xC0,0x80,0x80,0x80,0x7F,0x00,0x00,0x00},//J 42

{0x08,0xF8,0x88,0xC0,0x28,0x18,0x08,0x00},

{0x20,0x3F,0x20,0x01,0x26,0x38,0x20,0x00},//K 43

{0x08,0xF8,0x08,0x00,0x00,0x00,0x00,0x00},

{0x20,0x3F,0x20,0x20,0x20,0x20,0x30,0x00},//L 44

{0x08,0xF8,0xF8,0x00,0xF8,0xF8,0x08,0x00},

{0x20,0x3F,0x00,0x3F,0x00,0x3F,0x20,0x00},//M 45

{0x08,0xF8,0x30,0xC0,0x00,0x08,0xF8,0x08},

{0x20,0x3F,0x20,0x00,0x07,0x18,0x3F,0x00},//N 46

{0xE0,0x10,0x08,0x08,0x08,0x10,0xE0,0x00},

{0x0F,0x10,0x20,0x20,0x20,0x10,0x0F,0x00},//O 47

{0x08,0xF8,0x08,0x08,0x08,0x08,0xF0,0x00},

{0x20,0x3F,0x21,0x01,0x01,0x01,0x00,0x00},//P 48

{0xE0,0x10,0x08,0x08,0x08,0x10,0xE0,0x00},

{0x0F,0x18,0x24,0x24,0x38,0x50,0x4F,0x00},//Q 49

{0x08,0xF8,0x88,0x88,0x88,0x88,0x70,0x00},

{0x20,0x3F,0x20,0x00,0x03,0x0C,0x30,0x20},//R 50

{0x00,0x70,0x88,0x08,0x08,0x08,0x38,0x00},

{0x00,0x38,0x20,0x21,0x21,0x22,0x1C,0x00},//S 51

{0x18,0x08,0x08,0xF8,0x08,0x08,0x18,0x00},

{0x00,0x00,0x20,0x3F,0x20,0x00,0x00,0x00},//T 52

{0x08,0xF8,0x08,0x00,0x00,0x08,0xF8,0x08},

{0x00,0x1F,0x20,0x20,0x20,0x20,0x1F,0x00},//U 53

{0x08,0x78,0x88,0x00,0x00,0xC8,0x38,0x08},

{0x00,0x00,0x07,0x38,0x0E,0x01,0x00,0x00},//V 54

{0xF8,0x08,0x00,0xF8,0x00,0x08,0xF8,0x00},

{0x03,0x3C,0x07,0x00,0x07,0x3C,0x03,0x00},//W 55

{0x08,0x18,0x68,0x80,0x80,0x68,0x18,0x08},

{0x20,0x30,0x2C,0x03,0x03,0x2C,0x30,0x20},//X 56

{0x08,0x38,0xC8,0x00,0xC8,0x38,0x08,0x00},

{0x00,0x00,0x20,0x3F,0x20,0x00,0x00,0x00},//Y 57

{0x10,0x08,0x08,0x08,0xC8,0x38,0x08,0x00},

{0x20,0x38,0x26,0x21,0x20,0x20,0x18,0x00},//Z 58

{0x00,0x00,0x00,0xFE,0x02,0x02,0x02,0x00},

{0x00,0x00,0x00,0x7F,0x40,0x40,0x40,0x00},//[ 59

{0x00,0x0C,0x30,0xC0,0x00,0x00,0x00,0x00},

{0x00,0x00,0x00,0x01,0x06,0x38,0xC0,0x00},//\ 60

{0x00,0x02,0x02,0x02,0xFE,0x00,0x00,0x00},

{0x00,0x40,0x40,0x40,0x7F,0x00,0x00,0x00},//] 61

{0x00,0x00,0x04,0x02,0x02,0x02,0x04,0x00},

{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},//^ 62

{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},

{0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80},//\_ 63

{0x00,0x02,0x02,0x04,0x00,0x00,0x00,0x00},

{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},//` 64

{0x00,0x00,0x80,0x80,0x80,0x80,0x00,0x00},

{0x00,0x19,0x24,0x22,0x22,0x22,0x3F,0x20},//a 65

{0x08,0xF8,0x00,0x80,0x80,0x00,0x00,0x00},

{0x00,0x3F,0x11,0x20,0x20,0x11,0x0E,0x00},//b 66

{0x00,0x00,0x00,0x80,0x80,0x80,0x00,0x00},

{0x00,0x0E,0x11,0x20,0x20,0x20,0x11,0x00},//c 67

{0x00,0x00,0x00,0x80,0x80,0x88,0xF8,0x00},

{0x00,0x0E,0x11,0x20,0x20,0x10,0x3F,0x20},//d 68

{0x00,0x00,0x80,0x80,0x80,0x80,0x00,0x00},

{0x00,0x1F,0x22,0x22,0x22,0x22,0x13,0x00},//e 69

{0x00,0x80,0x80,0xF0,0x88,0x88,0x88,0x18},

{0x00,0x20,0x20,0x3F,0x20,0x20,0x00,0x00},//f 70

{0x00,0x00,0x80,0x80,0x80,0x80,0x80,0x00},

{0x00,0x6B,0x94,0x94,0x94,0x93,0x60,0x00},//g 71

{0x08,0xF8,0x00,0x80,0x80,0x80,0x00,0x00},

{0x20,0x3F,0x21,0x00,0x00,0x20,0x3F,0x20},//h 72

{0x00,0x80,0x98,0x98,0x00,0x00,0x00,0x00},

{0x00,0x20,0x20,0x3F,0x20,0x20,0x00,0x00},//i 73

{0x00,0x00,0x00,0x80,0x98,0x98,0x00,0x00},

{0x00,0xC0,0x80,0x80,0x80,0x7F,0x00,0x00},//j 74

{0x08,0xF8,0x00,0x00,0x80,0x80,0x80,0x00},

{0x20,0x3F,0x24,0x02,0x2D,0x30,0x20,0x00},//k 75

{0x00,0x08,0x08,0xF8,0x00,0x00,0x00,0x00},

{0x00,0x20,0x20,0x3F,0x20,0x20,0x00,0x00},//l 76

{0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x00},

{0x20,0x3F,0x20,0x00,0x3F,0x20,0x00,0x3F},//m 77

{0x80,0x80,0x00,0x80,0x80,0x80,0x00,0x00},

{0x20,0x3F,0x21,0x00,0x00,0x20,0x3F,0x20},//n 78

{0x00,0x00,0x80,0x80,0x80,0x80,0x00,0x00},

{0x00,0x1F,0x20,0x20,0x20,0x20,0x1F,0x00},//o 79

{0x80,0x80,0x00,0x80,0x80,0x00,0x00,0x00},

{0x80,0xFF,0xA1,0x20,0x20,0x11,0x0E,0x00},//p 80

{0x00,0x00,0x00,0x80,0x80,0x80,0x80,0x00},

{0x00,0x0E,0x11,0x20,0x20,0xA0,0xFF,0x80},//q 81

{0x80,0x80,0x80,0x00,0x80,0x80,0x80,0x00},

{0x20,0x20,0x3F,0x21,0x20,0x00,0x01,0x00},//r 82

{0x00,0x00,0x80,0x80,0x80,0x80,0x80,0x00},

{0x00,0x33,0x24,0x24,0x24,0x24,0x19,0x00},//s 83

{0x00,0x80,0x80,0xE0,0x80,0x80,0x00,0x00},

{0x00,0x00,0x00,0x1F,0x20,0x20,0x00,0x00},//t 84

{0x80,0x80,0x00,0x00,0x00,0x80,0x80,0x00},

{0x00,0x1F,0x20,0x20,0x20,0x10,0x3F,0x20},//u 85

{0x80,0x80,0x80,0x00,0x00,0x80,0x80,0x80},

{0x00,0x01,0x0E,0x30,0x08,0x06,0x01,0x00},//v 86

{0x80,0x80,0x00,0x80,0x00,0x80,0x80,0x80},

{0x0F,0x30,0x0C,0x03,0x0C,0x30,0x0F,0x00},//w 87

{0x00,0x80,0x80,0x00,0x80,0x80,0x80,0x00},

{0x00,0x20,0x31,0x2E,0x0E,0x31,0x20,0x00},//x 88

{0x80,0x80,0x80,0x00,0x00,0x80,0x80,0x80},

{0x80,0x81,0x8E,0x70,0x18,0x06,0x01,0x00},//y 89

{0x00,0x80,0x80,0x80,0x80,0x80,0x80,0x00},

{0x00,0x21,0x30,0x2C,0x22,0x21,0x30,0x00},//z 90

{0x00,0x00,0x00,0x00,0x80,0x7C,0x02,0x02},

{0x00,0x00,0x00,0x00,0x00,0x3F,0x40,0x40},//{ 91

{0x00,0x00,0x00,0x00,0xFF,0x00,0x00,0x00},

{0x00,0x00,0x00,0x00,0xFF,0x00,0x00,0x00},//| 92

{0x00,0x02,0x02,0x7C,0x80,0x00,0x00,0x00},

{0x00,0x40,0x40,0x3F,0x00,0x00,0x00,0x00},//} 93

{0x00,0x06,0x01,0x01,0x02,0x02,0x04,0x04},

{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},//~ 94

};

void OLED\_ShowChar(unsigned char x,unsigned char y,unsigned char chr);

void OLED\_ShowString(unsigned char x,unsigned char y, unsigned char \*p);

void OLED\_ShowString\_Short(unsigned char x,unsigned char y, unsigned char \*p,unsigned char l);

### 主函数编写

int main(void)

{

HAL\_Init();

SystemClock\_Config();

MX\_GPIO\_Init();

MX\_I2C1\_Init();

MX\_RTC\_Init();

Initial\_M096128x64\_ssd1306();

HAL\_Delay(5);

Picture(1);//显示一张图片--壁纸

OLED\_ShowString(22,2," "); //清空原图片中间部分（显示日期与时间）

OLED\_ShowString(22,4," ");

RTC\_CalendarShow(aShowTime, aShowDate); //获取日期

OLED\_ShowString(22,2,aShowDate); //字符串显示日期

while (1)

{

RTC\_CalendarShow(aShowTime, aShowDate);

if((Flag\_Alarm==0)&&(Flag\_Second\_Old!=aShowTime[7])) //非闹钟模式显示时钟

{

OLED\_ShowString\_Short(28,4,aShowTime,8); //限定长度8位

Flag\_Second\_Old = aShowTime[7];

}

}

}

### RTC初始化函数修改（方便自定义起始时间）

static void MX\_RTC\_Init(void)

{

RTC\_TimeTypeDef sTime = {0};

RTC\_DateTypeDef sDate = {0};

RTC\_AlarmTypeDef sAlarm = {0};

/\*\* Initialize RTC Only \*/

hrtc.Instance = RTC;

hrtc.Init.HourFormat = RTC\_HOURFORMAT\_24;

hrtc.Init.AsynchPrediv = 127;

hrtc.Init.SynchPrediv = 255;

hrtc.Init.OutPut = RTC\_OUTPUT\_DISABLE;

hrtc.Init.OutPutRemap = RTC\_OUTPUT\_REMAP\_NONE;

hrtc.Init.OutPutPolarity = RTC\_OUTPUT\_POLARITY\_HIGH;

hrtc.Init.OutPutType = RTC\_OUTPUT\_TYPE\_OPENDRAIN;

hrtc.Init.OutPutPullUp = RTC\_OUTPUT\_PULLUP\_NONE;

if (HAL\_RTC\_Init(&hrtc) != HAL\_OK)

{

Error\_Handler();

}

/\*\* Initialize RTC and set the Time and Date \*/

sTime.Hours = Init\_Time\_Hours;

sTime.Minutes = Init\_Time\_Minutes;

sTime.Seconds = Init\_Time\_Seconds;

sTime.SubSeconds = Init\_Time\_SubSeconds;

sTime.DayLightSaving = RTC\_DAYLIGHTSAVING\_NONE;

sTime.StoreOperation = RTC\_STOREOPERATION\_RESET;

if (HAL\_RTC\_SetTime(&hrtc, &sTime, RTC\_FORMAT\_BCD) != HAL\_OK)

{

Error\_Handler();

}

sDate.WeekDay = RTC\_WEEKDAY\_MONDAY;

sDate.Month = Init\_Time\_Month; //RTC\_MONTH\_JUNE;

sDate.Date = Init\_Time\_Date;

sDate.Year = Init\_Time\_Year;

if (HAL\_RTC\_SetDate(&hrtc, &sDate, RTC\_FORMAT\_BCD) != HAL\_OK)

{

Error\_Handler();

}

/\*\* Enable the Alarm A \*/

sAlarm.AlarmTime.Hours = Alarm\_A\_Hours;

sAlarm.AlarmTime.Minutes = Alarm\_A\_Minutes;

sAlarm.AlarmTime.Seconds = Alarm\_A\_Seconds;

sAlarm.AlarmTime.SubSeconds = Alarm\_A\_SubSeconds;

sAlarm.AlarmTime.DayLightSaving = RTC\_DAYLIGHTSAVING\_NONE;

sAlarm.AlarmTime.StoreOperation = RTC\_STOREOPERATION\_RESET;

sAlarm.AlarmMask = RTC\_ALARMMASK\_NONE;

sAlarm.AlarmSubSecondMask = RTC\_ALARMSUBSECONDMASK\_ALL;

sAlarm.AlarmDateWeekDaySel = RTC\_ALARMDATEWEEKDAYSEL\_WEEKDAY;

sAlarm.AlarmDateWeekDay = RTC\_WEEKDAY\_MONDAY;

sAlarm.Alarm = RTC\_ALARM\_A;

if (HAL\_RTC\_SetAlarm\_IT(&hrtc, &sAlarm, RTC\_FORMAT\_BCD) != HAL\_OK)

{

Error\_Handler();

}

/\*\* Enable the Alarm B \*/

sAlarm.AlarmTime.Hours = Alarm\_B\_Hours;

sAlarm.AlarmTime.Minutes = Alarm\_B\_Minutes;

sAlarm.AlarmTime.Seconds = Alarm\_B\_Seconds;

sAlarm.AlarmTime.SubSeconds = Alarm\_B\_SubSeconds;

sAlarm.Alarm = RTC\_ALARM\_B;

if (HAL\_RTC\_SetAlarm\_IT(&hrtc, &sAlarm, RTC\_FORMAT\_BCD) != HAL\_OK)

{

Error\_Handler();

}

}

### 功能函数添加

/\* USER CODE BEGIN 4 \*/

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*RTC

void HAL\_RTC\_AlarmAEventCallback(RTC\_HandleTypeDef \*hrtc)

{

HAL\_GPIO\_WritePin(GPIOA,GPIO\_PIN\_5,1); /\* 点亮LED灯 \*/

Picture(2);//显示二张图片--起床表情包

Flag\_Alarm=1; //闹钟模式标志位启用

}

void HAL\_RTCEx\_AlarmBEventCallback(RTC\_HandleTypeDef \*hrtc)

{

HAL\_GPIO\_WritePin(GPIOA,GPIO\_PIN\_5,0); /\* 关闭LED灯 \*/

Picture(1);//显示一张图片--壁纸

OLED\_ShowString(22,2," "); //清空原图片中间部分（显示日期与时间）

OLED\_ShowString(22,4," ");

OLED\_ShowString(22,2,aShowDate); //字符串显示日期

Flag\_Alarm=0; //闹钟模式标志位关闭

}

static void RTC\_CalendarShow(uint8\_t \*showtime, uint8\_t \*showdate)

{

RTC\_DateTypeDef sdatestructureget;

RTC\_TimeTypeDef stimestructureget;

/\* Get the RTC current Time \*/

HAL\_RTC\_GetTime(&hrtc, &stimestructureget, RTC\_FORMAT\_BIN);

/\* Get the RTC current Date \*/

HAL\_RTC\_GetDate(&hrtc, &sdatestructureget, RTC\_FORMAT\_BIN);

/\* Display time Format : hh:mm:ss \*/

sprintf((char \*)showtime, "%2d:%2d:%2d", stimestructureget.Hours, stimestructureget.Minutes, stimestructureget.Seconds);

/\* Display date Format : mm-dd-yy \*/

sprintf((char \*)showdate, "%2d-%2d-%2d", sdatestructureget.Month, sdatestructureget.Date, 2000 + sdatestructureget.Year);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*IIC基层函数

void Write\_IIC\_Command(unsigned char IIC\_Command)

{

aTxBuffer\_Command[1]=IIC\_Command;

HAL\_I2C\_Master\_Transmit(&hi2c1, (uint16\_t)I2C\_ADDRESS, (uint8\_t \*)aTxBuffer\_Command, TXBUFFERSIZE, 10000);

}

void Write\_IIC\_Data(unsigned char IIC\_Data)

{

aTxBuffer\_Data[1]=IIC\_Data;

HAL\_I2C\_Master\_Transmit(&hi2c1, (uint16\_t)I2C\_ADDRESS, (uint8\_t \*)aTxBuffer\_Data, TXBUFFERSIZE, 10000);

}

void Initial\_M096128x64\_ssd1306()

{

Write\_IIC\_Command(0xAE); //display off

Write\_IIC\_Command(0x20); //Set Memory Addressing Mode

Write\_IIC\_Command(0x10); //00,Horizontal Addressing Mode;01,Vertical Addressing Mode;10,Page Addressing Mode (RESET);11,Invalid

Write\_IIC\_Command(0xb0); //Set Page Start Address for Page Addressing Mode,0-7

Write\_IIC\_Command(0xc8); //Set COM Output Scan Direction

Write\_IIC\_Command(0x00);//---set low column address

Write\_IIC\_Command(0x10);//---set high column address

Write\_IIC\_Command(0x40);//--set start line address

Write\_IIC\_Command(0x81);//--set contrast control register

Write\_IIC\_Command(0xdf);

Write\_IIC\_Command(0xa1);//--set segment re-map 0 to 127

Write\_IIC\_Command(0xa6);//--set normal display

Write\_IIC\_Command(0xa8);//--set multiplex ratio(1 to 64)

Write\_IIC\_Command(0x3F);//

Write\_IIC\_Command(0xa4);//0xa4,Output follows RAM content;0xa5,Output ignores RAM content

Write\_IIC\_Command(0xd3);//-set display offset

Write\_IIC\_Command(0x00);//-not offset

Write\_IIC\_Command(0xd5);//--set display clock divide ratio/oscillator frequency

Write\_IIC\_Command(0xf0);//--set divide ratio

Write\_IIC\_Command(0xd9);//--set pre-charge period

Write\_IIC\_Command(0x22); //

Write\_IIC\_Command(0xda);//--set com pins hardware configuration

Write\_IIC\_Command(0x12);

Write\_IIC\_Command(0xdb);//--set vcomh

Write\_IIC\_Command(0x20);//0x20,0.77xVcc

Write\_IIC\_Command(0x8d);//--set DC-DC enable

Write\_IIC\_Command(0x14);//

Write\_IIC\_Command(0xaf);//--turn on oled panel

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*IIC的图片显示

void Picture(int i)

{

for(unsigned char y=0;y<8;y++)

{

Write\_IIC\_Command(0xb0+y);

Write\_IIC\_Command(0x0);

Write\_IIC\_Command(0x10);

for(unsigned char x=0;x<128;x++)

{

if(i==1)

Write\_IIC\_Data(show[y][x]);

else if(i==2)

Write\_IIC\_Data(biaoqinbao[y][x]);

else ;

}

}

}

void fill\_picture(unsigned char fill\_Data)

{

for(unsigned char m=0;m<8;m++)

{

Write\_IIC\_Command(0xb0+m); //page0-page1

Write\_IIC\_Command(0x00); //low column start address

Write\_IIC\_Command(0x10); //high column start address

for(unsigned char n=0;n<128;n++)

{

Write\_IIC\_Data(fill\_Data);

}

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*IIC的字符显示

void OLED\_ShowChar(unsigned char x,unsigned char y,unsigned char chr)

{

unsigned char nomber=0;

nomber=chr-' ';//得到偏移后的值即ASC码偏移量 设置空格为0号字符

if(x>127) //如果超出这一行自动跳转到下一行（+2）

{

x=0;

y=y+2;

}

Write\_IIC\_Command(0xb0+y);

Write\_IIC\_Command(0x00+x%16); //低四位横坐标

Write\_IIC\_Command(0x10+x/16);//高四位横坐标

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

Write\_IIC\_Data(L8H16[nomber\*2][i]);

Write\_IIC\_Command(0xb0+y+1);

Write\_IIC\_Command(0x00+x%16);

Write\_IIC\_Command(0x10+x/16);

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

Write\_IIC\_Data(L8H16[nomber\*2+1][i]);

}

void OLED\_ShowString(unsigned char x,unsigned char y,unsigned char \*chr)

{

unsigned char i=0;

while (chr[i]!='\0') //不是字符串的结束则一直循环

{

OLED\_ShowChar(x,y,chr[i]); //在x，y处显示字符

x+=8; //x=x+8 列地址加8准备显示下一字符

if(x>120) //位置不够显示当前字符，去下一行显示

{

x=0;

y+=2;

}

i++; //扫描下一字符

}

}

void OLED\_ShowString\_Short(unsigned char x,unsigned char y, unsigned char \*chr,unsigned char l)

{

unsigned char i=0;

while (chr[i]!='\0'&&i<l) //不是字符串并且小于长度

{

OLED\_ShowChar(x,y,chr[i]); //在x，y处显示字符

x+=8; //x=x+8 列地址加8准备显示下一字符

if(x>120) //位置不够显示当前字符，去下一行显示

{

x=0;

y+=2;

}

i++; //扫描下一字符

}

}

/\* USER CODE END 4 \*/