

YS-IR01F 红外学习模块使用说明



目录

- 一、程序移植步骤
- 二、电路连接方法
- 三、调试说明
- 四、其他

一、程序移植步骤

在配套的资料中找到“IR01F 红外底层驱动”文件夹，本程序是 STM32 系统移植编译过来的。只需要将其中的 2 个程序 C 文件添加到工程中，接着我们根据以下步骤进行相关修改。

1、SDA/SCL/BUSY 三个 I2C 通讯端口的定义

根据自己单片机系统的情况进行 IO 口定义，并根据程序文件“IR01F_IC_Driver.c”中的提示进行修改，具体修改程序从以下图片位置开始：

```
/*-----以下包含的为用户进行修改的参数-----  
-----根据每个函数的要求写入对应的配置-----  
----- begin -----  
/*-----*/  
  
#include "stm32f10x.h"  
/*-----编写好延时函数后，这里宏定义自定义修改-----*/  
#define DELAY_TIME 60 //50-100uS  
#define DELAY_ST 20 //20mS  
  
/*-----IO口的定义-----*/  
#define SDA_PIN GPIO_Pin_0  
#define SCL_PIN GPIO_Pin_1  
#define BUSY_PIN GPIO_Pin_2  
GPIO_InitTypeDef GPIO_InitStructure;  
  
/*-----  
* 名 称: Set_SDA_IO(void)  
* 功 能: 定义SDA端口  
* 入口参数: 无  
* 出口参数: 无  
-----*/
```

到以下图片位置结束：

```
/*
 * 名称: RemoteDelayms(UINT16 data)
 * 功能: 毫秒延时函数
 * 入口参数: data=1; 表示1毫秒
 * 出口参数: 无
 * 说明: 根据不同的单片机系统进行调整
 */
void RemoteDelayms(UINT16 data)
{
    uint16_t i=0;
    while(data-->0)
    {
        i=12000;
        while(i-->0);
    }
}

/*
 * 以上包含的为用户进行修改的参数
 * 根据每个函数的要求写入对应的配置
 */
end
```

整个程序的修改，除了以上规定部分外，其他程序都不允许改变。

2、修改延时函数

延时函数的修改主要根据对应的单片机系统的工作时序情况计算出延时时间，分别需要的延时为 微秒和毫秒级函数，具体的修改位置如下：（对应函数为：void DelayTime(UINT16 lTime); void RemoteDelayms(UINT16 data); ）

```

*****/
void DelayTime( UINT16 lTime )
{
    uint16_t i=0;
    while(lTime-->0)
    {
        i=10;
        while(i-->0);
    }
}

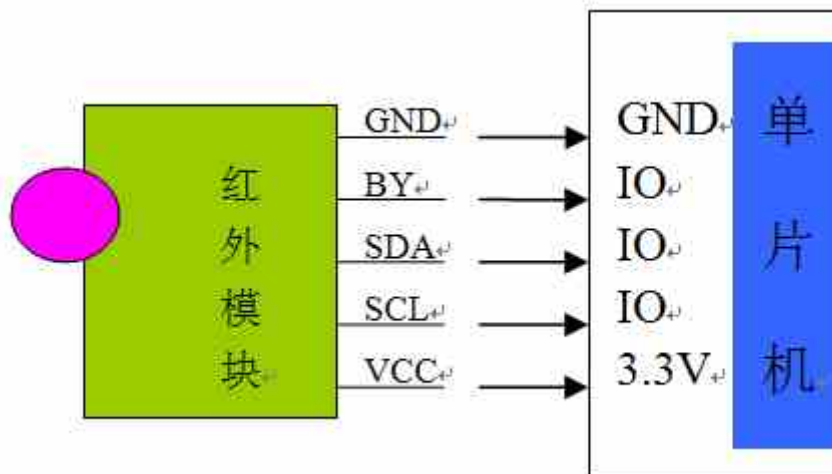
*****/
* 名    称: RemoteDelaysms(UINT16 data)
* 功    能: 毫秒延时函数
* 入口参数: data=1; 表示1毫秒
* 出口参数: 无
* 说    明: 根据不同的单片机系统进行调整
*****/
void RemoteDelaysms (UINT16 data)
{
    uint16_t i=0;
    while(data-->0)

```

3、 编译程序、调试

根据以上 2 个步骤移植好程序后，可以进行程序的编译，如发现编译不通过，请查明原因根据情况在不改变原来意思的情况下进行修改程序，本程序是 C 语音编写的，在不同的单片机系统上，表达方式存在不一致，故需要保证原意的情况下进行逐步修改，注意单片机内存（RAM）必须要 1K 以上才能够满足本程序的编译空间。

二、电路连接方法



红外模块与单片机的连接示意图

三、调试说明

根据以上步骤做好前期工作后，下面可以对程序进行实际应用，我们下面说明本模块的“学习红外”方法。

1、 红外学习过程

1) 在主函数中，定义数据和调用初始化函数：

```
unsigned char data_out[112]; // 红外数据缓存
```

```
IR01F_Init_Cfg();
```

2) 在 `while (1) { };` 中通过触发的方式调用学习函数：`IR_Learn_Start();`（如按键触发后调用）

在调用了学习函数后，将被学习的遥控器对准红外模块的接收头（10cm 距离以内）。

3) 等待红外模块的 BY（BUSY）端口变高电平。这是说明学习成功，那么我们调用 `Get_Learn_Data(data_out);`（可以采用串口输出的方式查看学习到的数

据)

2、 发射过程

我们学习到的数据后可以通过 flash 存储起来，然后调用发送函数：

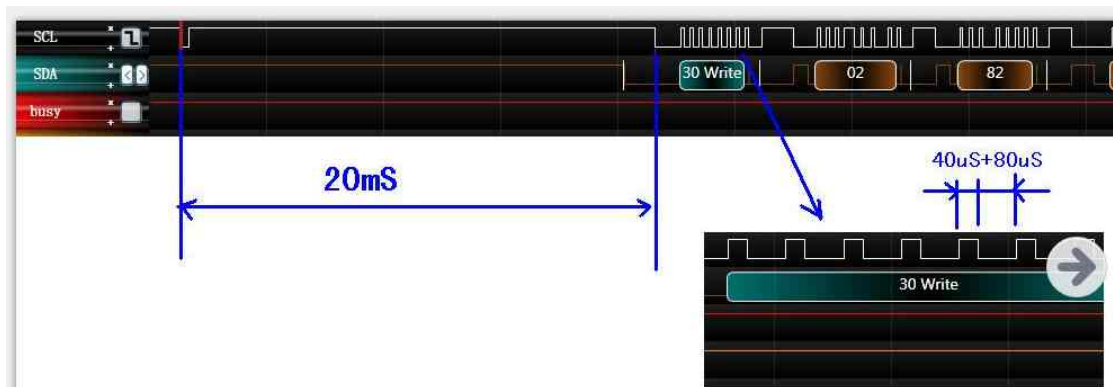
Send_IR_Data (char* data, UINT8 count); 将学习到的数据通过发送头发送出去。其中参数 *data 为红外数据如本例子中的 data_out; count 为数组的长度为 112。

3、 完整的一个收发过程参考程序：

```
unsigned char data_out[112]; //定义红外数据缓存
remote_poweron_init(); //调用初始化函数
while(1){
    if(GPIO_ReadInputDataBit(GPIOB,GPIO_Pin_14)==0){
        ID_Delay_ms(20);
        while(GPIO_ReadInputDataBit(GPIOB,GPIO_Pin_14)!=0);
        MODE_Flag++;
        if(MODE_Flag==2) MODE_Flag=0;
        switch(MODE_Flag){
            case 0:
                break;
            case 1:Learn_start(); //按键触发调用
                    Learn_Flag=1;
                break;}
        }
        if(GetBUSYStatus()==1 && Learn_Flag==1){ //BUSY 高电平检测
            Learn_Flag=0;
            MODE_Flag=0;
            Learn_data_flag=1;
            learn_data_in_out(data_out); //读取学习到的红外数据
            USART_OUT(USART1,data_out,112);
        }
        if(Learn_data_flag==1)
            {writeI2C(data_out, 112);} //发送学习到的红外数据
    }
}
```

四、其他

1、 IR01F 红外学习芯片工作时序



2、 程序移植测试不通过办法

本程序的移植和模块的测试成功，一般具备单片机基础的只需要半天至一天时间可以弄明白使用，如出现比较多的错误，是因系统表达方式不兼容或内存不足的原因，请细心检查修改正确，直至编译通过。同时需要严格 注意 三个引脚的定义和函数设置，程序的延时时间务必保证正确，供电为 3.3V。只需要保证这 3 点模块完全可以根据要求进行工作，如出现学习不成功，请反复检查保证正确，同时调整相关的时序宏定义！

```
#define DELAY_TIME 60 //50-100uS  
#define DELAY_ST 20 //20ms
```