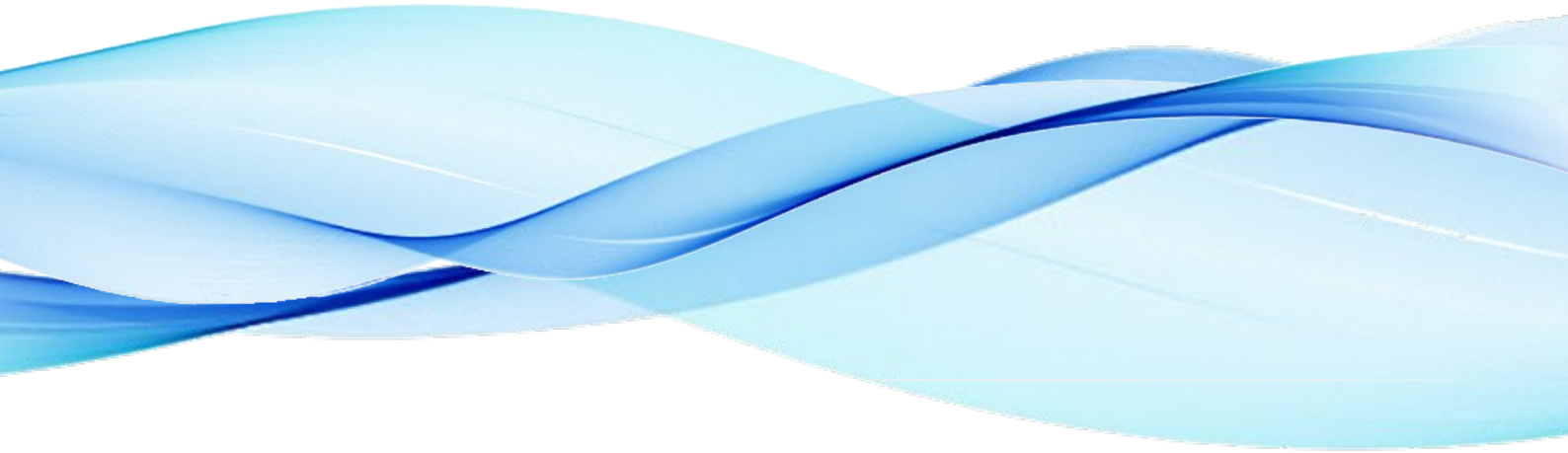
Ashining
泽耀科技

Si24R1_Stm8L101

Demo 程序说明



官网：www.ashining.com
邮箱：support@ashining.com
地址：四川省·成都市·高新西区百草路898号
智能信息产业园2层、5层

Si24R1_Stm8L101 的 demo 程序说明

本 demo 程序是基于 Stm8L101 单片机和 SI24R1 开发设计的。本程序包含了主函数文件 main.c，SPI 文件 drv_spi.c，串口文件 drv_uart.c,指示灯相关函数的文件 drv_led.c,按键相关函数文件 drv_button.c，延时函数文件 drv_delay.c 以及 Si24R1 的驱动文件 drv_RF24L01.c 等。

本程序实现的功能是使用 Si24R1 进行透明传输的功能，但只支持单独的接收或单独的发送。若想实现收发一体的功能需要用户自行修改程序。发送功能中分为了 2 种模式，固定发送模式和自由发送模式，由按键控制。自由发送模式是发送串口收到的数据。

1. 切换发送功能或接收功能的进行编译

打开工程文件后，在 main.c 文件中有__RF24L01_TX_TEST__的一个宏定义，若该参数未被定义的话发送功能则未被编译，若就当前状态进行编译下载，则该模块有了接收功能。要想编译下载发送功能的程序，需要点开 main.h 文件，将#define __RF24L01_TX_TEST__释放出来即可。如图：

打开 main.h 文件

```
#ifndef __MAIN_H__
#define __MAIN_H__

#include "drv_button.h"
#include "drv_delay.h"
#include "drv_led.h"
#include "drv_spi.h"
#include "drv_uart.h"
#include "drv_RF24L01.h"

//#define    __RF24L01_TX_TEST__                    //**@@ 如果测试
//#define    __USE_SOFT_SPI_INTERFACE__                 //**@@ 如果

/** 发送模式定义 */
enum
{
  TX_MODE_1 = 0,     //鏊戟€侗a寮?锛屽嚒鏅閻佸浐瀹氬彂閫佹ā寮?
  TX_MODE_2       //鏊戟€侗a寮?锛屽嚒鏅閻佹槸鏊 f 帴鏀跺埌鍚庣?
};

#endif
```

led.h | drv_RF24L01.c | main.h | main.c | drv_RF24L01.h

释放掉圈出部分

```
#ifndef __MAIN_H__
#define __MAIN_H__


#include "drv_button.h"
#include "drv_delay.h"
#include "drv_led.h"
#include "drv_spi.h"
#include "drv_uart.h"
#include "drv_RF24L01.h"


#define    __RF24L01_TX_TEST__              //**@@ 如果测试发? 凸δ兼
//#define    __USE_SOFT_SPI_INTERFACE__          //**@@ 如果? 褂萌?


/** 发送模式定义 */
enum
{
  TX_MODE_1 = 0,    //鎏戟e俺a寮?锛岮骖闇佸佸沪瀹氯殑瀛樻珋? 覆
  TX_MODE_2     //鎏戟e俺a寮?锛岮骖闇佷覆鎏 f 幔鏀跺埌鐨勬? 版嵫
};


#endif
```

drv_led.h | drv_RF24L01.c | main.h * | main.c | drv_RF24L01.h

现在的主程序，发送功能可编译，接受功能部分不可编译

2. 更改串口波特率

本程序默认的串口波特率是 9600，我们可以通过更改 drv_uart_init()中的参数更改串口波特率。如图：

```
void main( void )
{
  uint8_t i = 0;

  //串口初始化
  drv_uart_init( 9600 );

  //LED初始化
  drv_led_init( );

  //SPI初始化
  drv_spi_init( );

  //RF24L01初始化
  NRF24L01_Gpio_Init( );
  NRF24L01_check( );
  RF24L01_Init( );

  led_red_off( );
  led_green_off( );
  for( i = 0; i < 6; i++ )
  {
    led_red_flashing( );
    led_green_flashing( );
    drv_delay_ms( 500 );
  }

#ifdef  __RF24L01_TX_TEST__
  //=============================================================================
  //*****************************************************************************
  //************************************ 发送 ***********************************
  //*****************************************************************************
  //=============================================================================

  //按键初始化
  drv_button_init( );
```

drv_led.h | drv_RF24L01.c | main.h * | main.c | drv_RF24L01.h

本程序默认波特率为 9600，用户可根据需要更改为对应参数即可。

3. 更改 Si24R1 的通信地址

若需更改 Si24R1 的通信地址，我们需要打开 drv_RF24L01.h 文件，更改 INIT_ADDR 的宏定义参数。值得注意的是发送模块和接收模块的通信地址要一致才能通信。

```
13  **********************************************************************
14  */
15
16
17  #ifndef __DRV_RF24L01_H__
18  #define __DRV_RF24L01_H__
19
20
21  #include "drv_spi.h"
22
23
24  /** 配置和选项定义 */
25  #define DYNAMIC_PACKET      1       //1:动态数据包, 0:固定
26  #define FIXED_PACKET_LEN    32      //包长度
27  #define REPEAT_CNT          15      //重复次数
28  #define INIT_ADDR           0x34,0x43,0x10,0x10,0x01
29
30  /** RF24L01硬件IO定义 */
31  #define RF24L01_CE_GPIO_PORT      GPIOB
```

4. 更改 Si24R1 的通信配置

若需更改 Si24R1 的通信配置则需要在 drv_rf24l1.c 文件中的 RF24L01_Init()函数修改对应参数。其中通信配置中最重要的是信道，空速，发射功率等。需注意的是发送方与接收方的信道，空速，发射功率都需一致。

a. 若需要更改信道则更改 RF_CH 寄存器的参数。

```
*/
void RF24L01_Init( void )
{
    uint8_t addr[5] = {INIT_ADDR};

    RF24L01_SET_CE_HIGH( ); //拉高CE
    NRF24L01_Clear_IRQ_Flag( IRQ_ALL );
#if DYNAMIC_PACKET == 1

    NRF24L01_Write_Reg( DYNPD, ( 1 << 0 ) );  //使能通道1动? 频  享ざ?
    NRF24L01_Write_Reg( FEATRUE, 0x07 );     //设置特征寄存器，使能动态负载长度，使能ACK负载，使能
    NRF24L01_Read_Reg( DYNPD );          //读取使能动态负载长度
    NRF24L01_Read_Reg( FEATRUE );        //读取特征寄存器

#elif DYNAMIC_PACKET == 0

    L01_WriteSingleReg( L01REG_RX_PW_P0, FIXED_PACKET_LEN );  //固定数据长度

#endif  //DYNAMIC_PACKET

    NRF24L01_Write_Reg( CONFIG, /*( 1<<MASK_RX_DR ) |*/   //接收中断
                                ( 1 << EN_CRC ) |      //使能CRC 1个字节
                                ( 1 << PWR_UP ) );     //开启设备
    NRF24L01_Write_Reg( EN_AA, ( 1 << ENAA_P0 ) );       //通道0自动应答
    NRF24L01_Write_Reg( EN_RXADDR, ( 1 << ERX_P0 ) );    //通道0接收
    NRF24L01_Write_Reg( SETUP_AW, AW_5BYTES );           //地址宽度 5个字节
    NRF24L01_Write_Reg( SETUP_RETR, ARD_4000US |
                        ( REPEAT_CNT & 0x0F ) );          //重复等待时间 250us
    NRF24L01_Write_Reg( RF_CH, 60 );                    //初始化通道
    NRF24L01_Write_Reg( RF_SETUP, 0x26 );

    NRF24L01_Set_TxAddr( &addr[0], 5 );                  //设置TX地址
    NRF24L01_Set_RxAddr( 0, &addr[0], 5 );               //设置RX地址
}
```

寄存器参数设置：(2.4GHz---2.525GHz)

05	RF_CH				RF Channel
	Reserved	7	0	R/W	Only '0' allowed
	RF_CH	6:0	0000010	R/W	Sets the frequency channel nRF24L01+ operates on

b. 若需更改空速和发射功率则更改 RF_SETUP 寄存器的参数。

```
622  void RF24L01_Init( void )
623  {
624      uint8_t addr[5] = {INIT_ADDR};
625
626      RF24L01_SET_CE_HIGH( ); //拉高CE
627      NRF24L01_Clear_IRQ_Flag( IRQ_ALL );
628  #if DYNAMIC_PACKET == 1
629
630      NRF24L01_Write_Reg( DYNPD, ( 1 << 0 ) );  //使能通道1动? 颊  苹ざ?
631      NRF24L01_Write_Reg( FEATRUE, 0x07 );    //设置特征寄存器，使能动态负载长度，使
632      NRF24L01_Read_Reg( DYNPD );          //读取使能动态负载长度
633      NRF24L01_Read_Reg( FEATRUE );        //读取特征寄存器
634
635  #elif DYNAMIC_PACKET == 0
636
637      L01_WriteSingleReg( L01REG_RX_PW_P0, FIXED_PACKET_LEN );  //固定数据长度
638
639  #endif  //DYNAMIC_PACKET
640
641      NRF24L01_Write_Reg( CONFIG, /*( 1<<MASK_RX_DR ) |*/   //接收中断
642                                  ( 1 << EN_CRC ) |      //使能CRC 1个字节
643                                  ( 1 << PWR_UP ) );     //开启设备
644      NRF24L01_Write_Reg( EN_AA, ( 1 << ENAA_P0 ) );      //通道0自动应答
645      NRF24L01_Write_Reg( EN_RXADDR, ( 1 << ERX_P0 ) );   //通道0接收
646      NRF24L01_Write_Reg( SETUP_AW, AW_5BYTES );          //地址宽度 5个字节
647      NRF24L01_Write_Reg( SETUP_RETR, ARD_4000US |
648                                  ( REPEAT_CNT & 0x0F ) );      //重复等待时间 250us
649      NRF24L01_Write_Reg( RF_CH, 60 );                  //初始化通道
650      NRF24L01_Write_Reg( RF_SETUP, 0x26 );
651
652      NRF24L01_Set_TxAddr( &addr[0], 5 );                    //设置TX地址
653      NRF24L01_Set_RxAddr( 0, &addr[0], 5 );                 //设置RX地址
654  }
655
```

drv_led.h | drv_RF24L01.c | main.h * | main.c | drv_RF24L01.h

寄存器参数设置：

06	RF_SETUP				RF Setup Register
	CONT_WAVE	7	0	R/W	Enables continuous carrier transmit when high.
	Reserved	6	0	R/W	Only '0' allowed
	RF_DR_LOW	5	0	R/W	Set RF Data Rate to 250kbps. See RF_DR_HIGH for encoding.
	PLL_LOCK	4	0	R/W	Force PLL lock signal. Only used in test
	RF_DR_HIGH	3	1	R/W	Select between the high speed data rates. This bit is don't care if RF_DR_LOW is set. Encoding: [RF_DR_LOW, RF_DR_HIGH]: '00' – 1Mbps '01' – 2Mbps '10' – 250kbps '11' – Reserved
	RF_PWR	2:1	11	R/W	Set RF output power in TX mode '00' – -18dBm '01' – -12dBm '10' – -6dBm '11' – 0dBm
	Obsolete	0			Don't care

例如参数为 0x26:空速为 250kbps，发射功率为 0dBm。