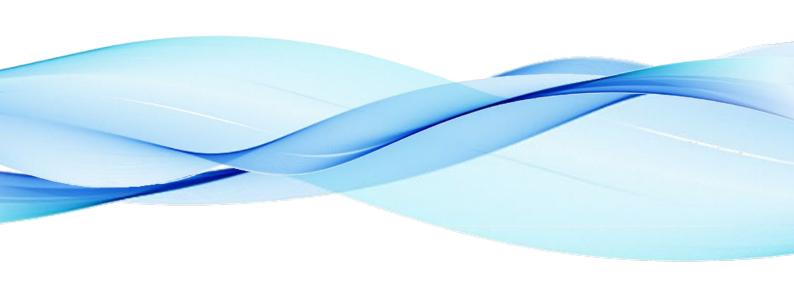


Si24R1_Stm32F103

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_Stm32F103 的 demo 程序说明

本 demo 程序是基于 Stm32F103 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,会看到发送功能部分是灰色的,即不可编译状态。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:

```
drv_led.c main.c drv_RF24L01.c main.h* drv_spi.c drv_uart.c drv_button.c
        for( i = 0; i < 6; i++)
 63 🖹
 64
           led_red_flashing();
 65
           led_green_flashing();
           drv_delay_500Ms(1); //检测到芯片 闪烁3个周期
 66
 67
 68
 69
 70 = #ifdef
           __RF24L01_TX_TEST_
 71
      73
 75
 76
 77
        //按键初始化
 78
        drv_button_init();
 79
        RF24L01_Set_Mode(MODE_TX); //发送模式
 80
 81
        while(1)
 82 =
           //模式切换
 83
           if( BUTOTN PRESS DOWN == drv button check())
 84
 85
                                        //模式会在 TX MODE 1(0),TX MODE 2(
 86
               g_TxMode = 1 - g_TxMode;
 87
               //状态显示清零
               led_green_off();
               led_red_off( );
 91
```

打开 main.h 文件

```
drv_led.c main.c drv_RF24L01.c main.h drv_spi.c drv_uart.c drv_button.c
                                                                                 ₹ ×
        * @author 泽耀科技 ASHINING
        * @version V3.0
        * @date 2016-10-08
* @brief 主配置H文件
    5
                   主配置H文件
    6
         8
        * @attention
   9
        * 官网 : http://www.ashining.com
   10
        * 淘宝 : https://shop105912646.taobao.com
* 阿里巴巴: https://cdzeyao.1688.com
   11
   12
   13
   14
   15
   16
      17
  18 = #ifndef MAIN_H
19 #define MAIN_H
  20
       #include "drv_RF24L01.h"
  21
      #include "drv_uart.h"
#include "drv_button.h"
#include "drv_delay.h"
   22
   23
   24
       #include "drv_led.h"
  25
  26
   27
                                                           //**@@ 如果测试发送功能则系
//**@@ 如果使用软件SPI则需
      28
   29
   30
```

释放掉圈出部分

```
drv_led.c main.c drv_RF24L01.c main.h* drv_spi.c drv_uart.c drv_button.c
   1 🗐 / * *
   2
   3
        * @author 泽耀科技 ASHINING
        * @version V3.0
   4
5
        * @date 2016-10-08
* @brief 主配置H文件
   6
        ******************
   8
        * @attention
   9
        * 官网 : http://www.ashining.com
* 淘宝 : https://shop105912646.taobao.com
* 阿里巴巴: https://cdzeyao.1688.com
  10
  11
  13
        */
  14
  15
  16
  18 = #ifndef MAIN_H
19 #define MAIN_H
  20
  21
      #include "drv_RF24L01.h"
      #include "drv_uart.h"
#include "drv_button.h"
  22
  23
      #include "drv_delay.h"
#include "drv_led.h"
  24
  25
  26
  27
      //**@@ 如果测试发送功能则是
  28
                                                               //**@@ 如果使用软件SPI则需
  29
  30
```

现在的主程序, 发送功能可编译, 接受功能部分不可编译

demo 程序说明文档 成都泽耀科技有限公司

样品网址: https://zeyaotech.taobao.com

更多产品请登录: www.ashining.com

```
68
69
70 #ifdef __RF24L01_TX_TEST_
71
  72
  73
  //***********************
74
75
76
     //按键初始化
77
78
     drv_button_init();
79
80
     RF24L01 Set Mode ( MODE TX );
                            //发送模式
81
     while (1)
82 🖹
        //模式切换
83
84
        if( BUTOTN_PRESS_DOWN == drv_button_check())
85 ់
                                //模式会在 TX MODE 1(0),TX MODE
          g TxMode = 1 - g TxMode;
86
87
          //状态显示清零
88
89
          led green off();
90
          led red off();
91
92
          if( TX_MODE_1 == g_TxMode )
93 📥
94
             for( i = 0; i < 6; i++ )
95 🗀
                                //固定发送模式, 红灯闪烁3次
                led_red_flashing( );
96
97
                drv delay 500Ms(1);
```

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:

```
drv_led.c main.c drv_RF24L01.c main.h* drv_spi.c drv_uart.c drv_button.c
  26
  27
  28 -/**
        * @brief :主函数
  29
        * @param :无
  30
        * @note :无
  31
        * @retval:无
  32
  33
  34
     int main( void )
  35 ⊟ {
  36
          uint8_t i = 0;
  37
  38
          //串口初始化
  39
          drv_uart_init( 9600 );
  40
          //延时初始化
  41
  42
          drv_delay_init();
  43
          . .___>++4/\ //.
```

函数中的参数 9600 代表的是波特率为 9600, 用户可根据自己需要进行更改。

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。

```
drv_RF24L01.h
                      drv_RF24L01.c
                                   main.h*
                                             drv_spi.c drv_uart.c
       *****************
       * @attention
  8
  9
  10
                :
                    http://www.ashining.com
       * 淘宝
  11
                    https://shop105912646.taobao.com
        * 阿里巴巴: https://cdzeyao.1688.com
  12
  13
 14
       */
 15
 16
 17 ☐ #ifndef __DRV_RF24L01_H_
18 #define __DRV_RF24L01_H_
  19
 20
     #include "drv spi.h"
 21
 22
 23
     /** 配置和选项定义 */
 24
     #define DYNAMIC PACKET
                                       //1:动态数据包, 0:固定
  25
                                       //包长度
  26
     #define FIXED_PACKET_LEN
                               32
  27
     #define REPEAT CNT
                                15
                                       <u>//</u>重复次数
 28 #define INIT ADDR
                               0x34,0x43,0x10,0x10,0x01
 29
  30
  31 /** RF24L01硬件接口定义 */
```

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。

更多产品请登录: www.ashining.com

a. 若需要更改信道则更改 RF_CH 寄存器的参数。

```
drv_RF24L01.h main.c main.c main.c main.c main.h main.h main.h main.h drv_spi.c drv_uart.c drv_button.c
          RF24L01_SET_CE_LOW(); //使能设备
RF24L01_SET_CS_HIGH(); //取消SPI
 630
                                     //取消SPI片选
 631
 633 }
 634 🗐 /**
 635
        * @brief:RF24L01模块初始化
       * @param :无
 637
       * @note :无
        * @retval:无
 638
 639 */
 640 void RF24L01_Init( void )
 641 □ {
 642
          uint8_t addr[5] = {INIT_ADDR};
 643
 644
          RF24L01_SET_CE_HIGH();
 645
          NRF24L01_Clear_IRQ_Flag( IRQ_ALL );
 646 #if DYNAMIC PACKET == 1
 647
          NRF24L01_Write_Reg( DYNPD, ( 1 << 0 ) );
NRF24L01_Write_Reg( FEATRUE, 0x07 );
                                                     //使能通道1动态数据长度
 648
                                                     //设置特征寄存器, 使能动态负载长
 649
          NRF24L01_Read_Reg( DYNPD );
NRF24L01_Read_Reg( FEATRUE );
                                                     //读取使能动态负载长度
 650
                                                     //读取特征寄存器
 651
 652
 653
      #elif DYNAMIC_PACKET == 0
 654
 655
          L01_WriteSingleReg(L01REG_RX_PW_P0, FIXED_PACKET_LEN); //固定数据长度
 656
 657
      #endif //DYNAMIC PACKET
 658
 659 🖨
          NRF24L01 Write Reg( CONFIG, /*( 1<<MASK RX DR ) |*/
                                                                //接收中断
                                           ( 1 << EN CRC ) |
                                                                //使能CRC 1个字节
 660
                                            ( 1 << PWR_UP ) );
                                                                //开启设备
 661
                                                                //通道0自动应答
          NRF24L01_Write_Reg(EN_AA, (1 << ENAA_P0 ));
 662
                                                                //通道0接收
          NRF24L01_Write_Reg( EN_RXADDR, ( 1 << ERX_P0 ) );
 663
                                                                //地址宽度 5个字节
 664
          NRF24L01_Write_Reg( SETUP_AW, AW_5BYTES );
 665 🖨
          NRF24L01_Write_Reg( SETUP_RETR, ARD_4000US |
 666
                              ( REPEAT_CNT & 0x0F ) );
                                                                //重复等待时间 250us
      NRF24L01_Write_Reg( RF_CH, 60 );
                                                                //初始化通道
 667
          NRF24LU1_Write_Reg( RF_SETUP, 0x26 );
 668
                                                                //设置通信速度为1M
 669
                                                                 //设置TX地址
 670
          NRF24L01_Set_TxAddr( &addr[0], 5 );
                                                                 //设置RX地址
          NRF24L01_Set_RxAddr( 0, &addr[0], 5 );
 671
 672
      }
 673
```

寄存器参数设置: (2.4GHz---2.525GHz)

- 1				I .		
	05	RF_CH				RF Channel
		Reserved	7	0	R/W	Only '0' allowed
1		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。

```
drv_RF24L01.h main.c drv_RF24L01.c main.h* drv_spi.c drv_uart.c drv_button.c ▼
                                     //使能设备
          RF24L01_SET_CE_LOW();
 630
                                     //取消SPI片选
 631
          RF24L01_SET_CS_HIGH();
 632
     }
 633
 634 🗏 /**
       * @brief :RF24L01模块初始化
 635
       * @param :无
 636
       * @note :无
 637
       * @retval:无
 638
 639
 640 void RF24L01_Init( void )
 641 □ {
 642
         uint8 t addr[5] = {INIT ADDR};
 643
          RF24L01 SET CE HIGH();
 644
 645 NRF24L01_Clear_IRQ_Flag( IRQ_ALL );
646 = #if DYNAMIC_PACKET == 1
 647
          NRF24L01_Write_Reg( DYNPD, ( 1 << 0 ) ); //使能通道1动态数据长度
 648
         NRF24L01 Write Reg( FEATRUE, 0x07 );
NRF24L01 Read Reg( DYNPD );
                                                   //设置特征寄存器, 使能动态负载长
 649
                                                   //读取使能动态负载长度
 650
 651
          NRF24L01 Read Reg ( FEATRUE );
                                                    //读取特征寄存器
 652
 653
      #elif DYNAMIC PACKET == 0
 654
 655
          LO1_WriteSingleReg(LO1REG_RX_PW_PO, FIXED_PACKET_LEN); //固定数据长度
 656
 657
      #endif //DYNAMIC_PACKET
 658
                                                               //接收中断
 659
          NRF24L01_Write_Reg( CONFIG, /*( 1<<MASK_RX_DR ) |*/
 660
                                          ( 1 << EN_CRC ) |
                                                               //使能CRC 1个字节
                                                               //开启设备
 661
                                           ( 1 << PWR_UP ) );
                                                               //通道0自动应答
 662
          NRF24L01_Write_Reg( EN_AA, ( 1 << ENAA_P0 ) );
                                                               //涌消∩接收
 663
          NRF24L01_Write_Reg( EN_RXADDR, ( 1 << ERX_P0 ) );
                                                               //地址宽度 5个字节
 664
          NRF24L01_Write_Reg( SETUP_AW, AW_5BYTES );
 665 🖨
          NRF24L01_Write_Reg( SETUP_RETR, ARD_4000US |
                                                               //重复等待时间 250us
 666
                             ( REPEAT_CNT & 0x0F ) );
         NRF24L01 Write Reg( RF CH, 60 );
NRF24L01 Write Reg( RF SETUP, 0x26 );
                                                               //初始化通道
 667
                                                               //设置通信速度为1M
 668
 669
                                                                //设置TX地址
 670
          NRF24L01 Set TxAddr( &addr[0], 5 );
                                                                //设置RX地址
          NRF24L01_Set_RxAddr( 0, &addr[0], 5 );
 671
 672
      1
 673
```

demo 程序说明文档

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
		0:4	14	D 0.04	
	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。