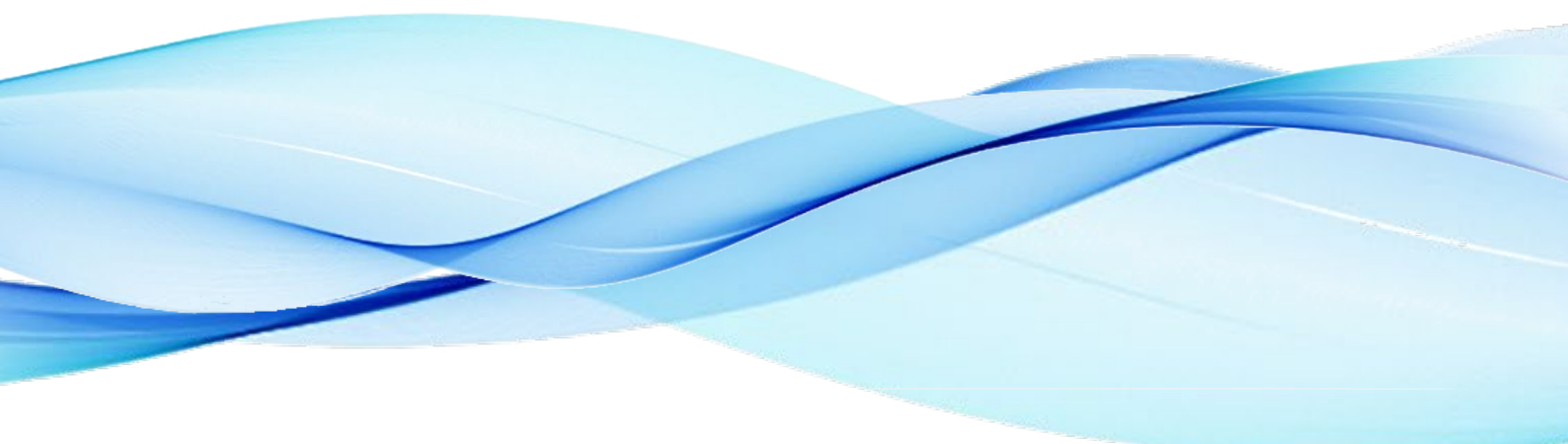




nRF24L01_Stm32F103

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

nRF24L01_Stm32F103 的 demo 程序说明

本 demo 程序是基于 Stm32F103 单片机和 nRF24L01 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 nRF24L01 的驱动文件 drv_RF24L01.c。

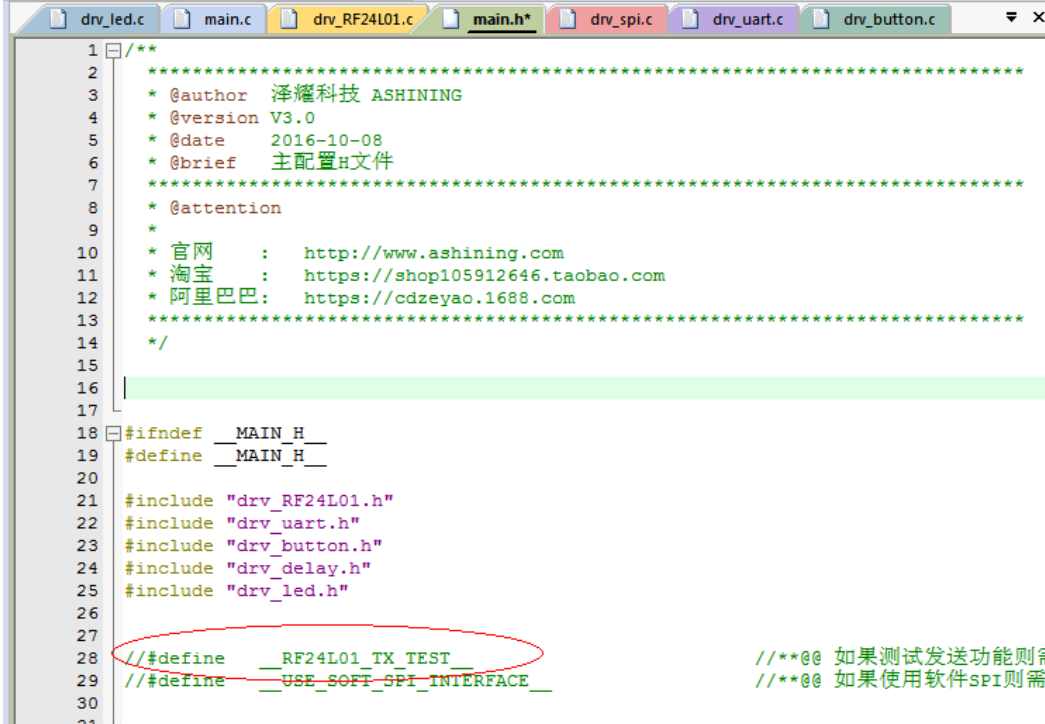
本程序实现的功能是使用 nRF24L01 进行透明传输的功能, 但只支持单独的接收或单独的发送。若想实现收发一体的功能需要用户自行修改程序。发送功能中分为了 2 种模式, 固定发送模式和自由发送模式, 由按键控制。自由发送模式是发送串口收到的数据。

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:

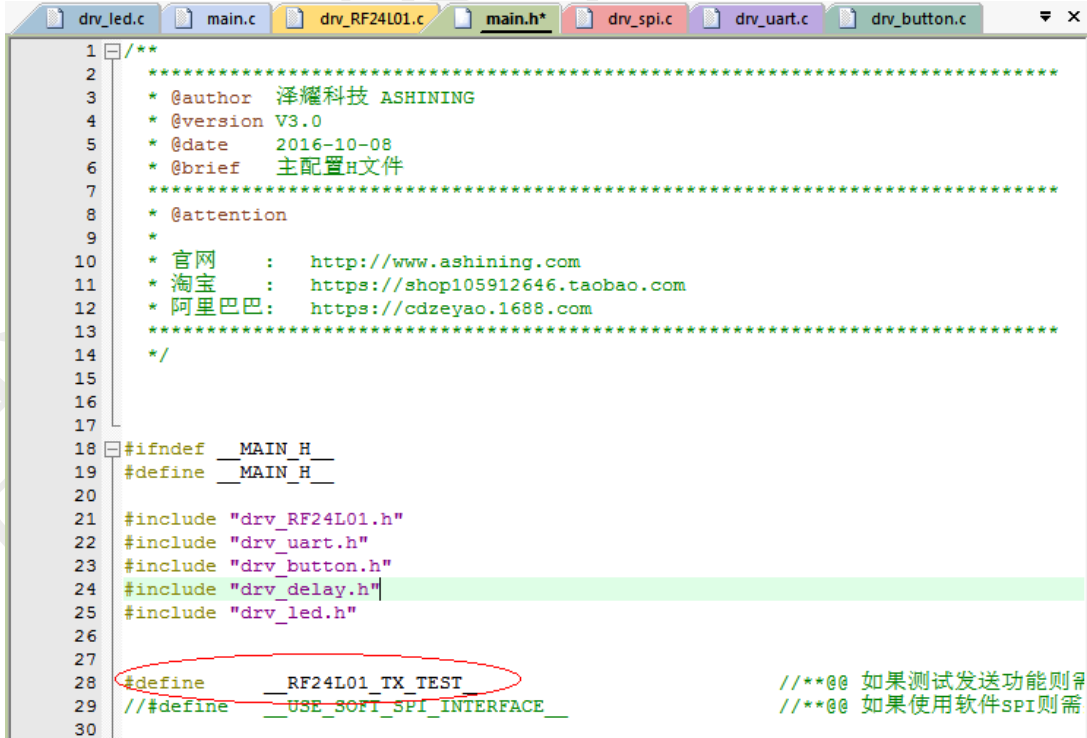


打开 main.h 文件



```
1 /**
2  * @author 泽耀科技 ASHINING
3  * @version V3.0
4  * @date 2016-10-08
5  * @brief 主配置H文件
6  *
7  * @attention
8  *
9  * 官网 : http://www.ashining.com
10 * 淘宝 : https://shop105912646.taobao.com
11 * 阿里巴巴: https://cdzeyao.1688.com
12 *
13 */
14
15
16
17
18 #ifndef MAIN_H
19 #define MAIN_H
20
21 #include "drv_RF24L01.h"
22 #include "drv_uart.h"
23 #include "drv_button.h"
24 #include "drv_delay.h"
25 #include "drv_led.h"
26
27
28 // #define RF24L01_TX_TEST // ** 如果测试发送功能则需要
29 // #define USE_SOFT_SPI_INTERFACE // ** 如果使用软件SPI则需要
30
31
```

释放掉圈出部分



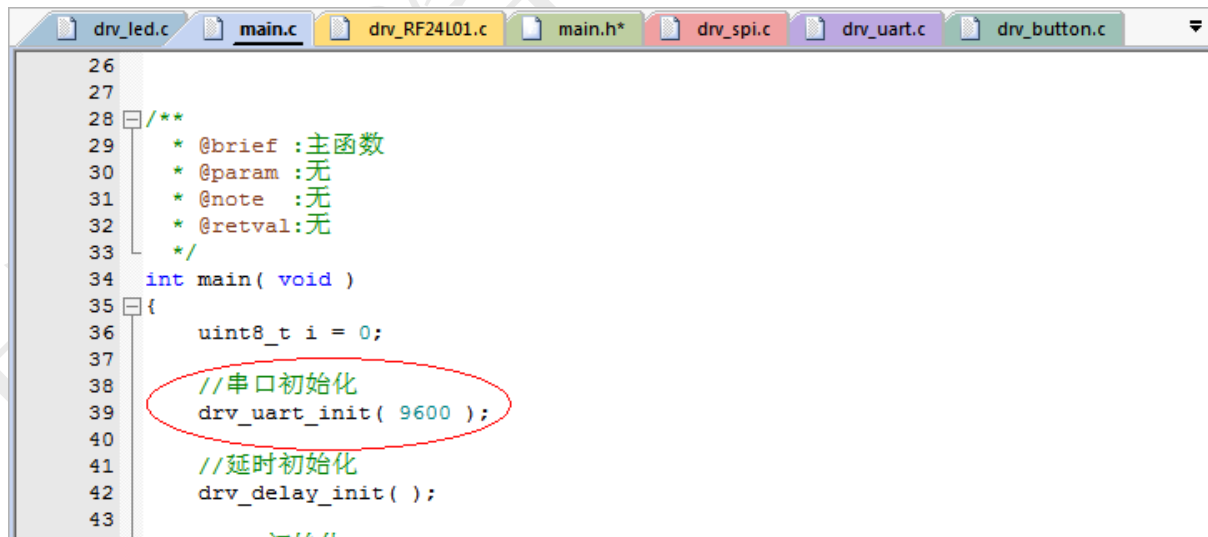
```
1 /**
2  * @author 泽耀科技 ASHINING
3  * @version V3.0
4  * @date 2016-10-08
5  * @brief 主配置H文件
6  *
7  * @attention
8  *
9  * 官网 : http://www.ashining.com
10 * 淘宝 : https://shop105912646.taobao.com
11 * 阿里巴巴: https://cdzeyao.1688.com
12 *
13 */
14
15
16
17
18 #ifndef MAIN_H
19 #define MAIN_H
20
21 #include "drv_RF24L01.h"
22 #include "drv_uart.h"
23 #include "drv_button.h"
24 #include "drv_delay.h"
25 #include "drv_led.h"
26
27
28 // #define RF24L01_TX_TEST // ** 如果测试发送功能则需要
29 // #define USE_SOFT_SPI_INTERFACE // ** 如果使用软件SPI则需要
30
```

现在的主程序，发送功能可编译，接受功能部分不可编译

```
68
69
70 #ifdef __RF24L01_TX_TEST__
71 //=====
72 //***** 发送 *****
73 //***** 发送 *****
74 //=====
75
76
77 //按键初始化
78 drv_button_init( );
79
80 RF24L01_Set_Mode( MODE_TX ); //发送模式
81 while( 1 )
82 {
83     //模式切换
84     if( BUTOTN_PRESS_DOWN == drv_button_check( ) )
85     {
86         g_TxMode = 1 - g_TxMode; //模式会在 TX_MODE_1( 0 ),TX_MODE_
87
88         //状态显示清零
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             {
96                 led_red_flashing( ); //固定发送模式，红灯闪烁3次
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：



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 nRF24L01 的通信地址

若需更改 nRF24L01 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。



```
7  *****
8  * @attention
9  *
10 * 官网      :   http://www.ashining.com
11 * 淘宝      :   https://shop105912646.taobao.com
12 * 阿里巴巴:   https://cdzeyao.1688.com
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
31 /** RF24L01硬件接口定义 */
```

4. 更改 nRF24L01 的通信配置

若需更改 nRF24L01 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。

```
drv_RF24L01.h  main.c  drv_RF24L01.c  main.h*  drv_spi.c  drv_uart.c  drv_button.c
630 RF24L01_SET_CE_LOW( ); //使能设备
631 RF24L01_SET_CS_HIGH( ); //取消SPI片选
632 }
633
634 /**
635  * @brief :RF24L01模块初始化
636  * @param :无
637  * @note :无
638  * @retval:无
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
648     NRF24L01_Write_Reg( DYNPD, ( 1 << 0 ) ); //使能通道1动态数据长度
649     NRF24L01_Write_Reg( FEATRUE, 0x07 ); //设置特征寄存器, 使能动态负载长
650     NRF24L01_Read_Reg( DYNPD ); //读取使能动态负载长度
651     NRF24L01_Read_Reg( FEATRUE ); //读取特征寄存器
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
660                        ( 1 << EN_CRC ) | //使能CRC 1个字节
661                        ( 1 << PWR_UP ) ); //开启设备
662     NRF24L01_Write_Reg( EN_AA, ( 1 << ENAA_P0 ) ); //通道0自动应答
663     NRF24L01_Write_Reg( EN_RXADDR, ( 1 << ERX_P0 ) ); //通道0接收
664     NRF24L01_Write_Reg( SETUP_AW, AW_5BYTES ); //地址宽度 5个字节
665     NRF24L01_Write_Reg( SETUP_RETR, ARD_4000US |
666                        ( REPEAT_CNT & 0x0F ) ); //重复等待时间 250us
667     NRF24L01_Write_Reg( RF_CH, 60 ); //初始化通道
668     NRF24L01_Write_Reg( RF_SETUP, 0x26 ); //设置通信速度为1M
669
670     NRF24L01_Set_TxAddr( &addr[0], 5 ); //设置Tx地址
671     NRF24L01_Set_RxAddr( 0, &addr[0], 5 ); //设置Rx地址
672 }
673
```

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。

```
drv_RF24L01.h  main.c  drv_RF24L01.c  main.h*  drv_spi.c  drv_uart.c  drv_button.c
630     RF24L01_SET_CE_LOW( );           //使能设备
631     RF24L01_SET_CS_HIGH( );          //取消SPI片选
632 }
633
634 /**
635  * @brief :RF24L01模块初始化
636  * @param :无
637  * @note :无
638  * @retval:无
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
648     NRF24L01_Write_Reg( DYNPD, ( 1 << 0 ) ); //使能通道1动态数据长度
649     NRF24L01_Write_Reg( FEATRUE, 0x07 );      //设置特征寄存器, 使能动态负载长
650     NRF24L01_Read_Reg( DYNPD );              //读取使能动态负载长度
651     NRF24L01_Read_Reg( FEATRUE );            //读取特征寄存器
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
660                        ( 1 << EN_CRC ) | //使能CRC 1个字节
661                        ( 1 << PWR_UP ) ); //开启设备
662     NRF24L01_Write_Reg( EN_AA, ( 1 << ENAA_P0 ) ); //通道0自动应答
663     NRF24L01_Write_Reg( EN_RXADDR, ( 1 << ERX_P0 ) ); //通道0接收
664     NRF24L01_Write_Reg( SETUP_AW, AW_5BYTES ); //地址宽度 5个字节
665     NRF24L01_Write_Reg( SETUP_RETR, ARD_4000US |
666                        ( REPEAT_CNT & 0x0F ) ); //重复等待时间 250us
667     NRF24L01_Write_Reg( RF_CH, 60 ); //初始化通道
668     NRF24L01_Write_Reg( RF_SETUP, 0x26 ); //设置通信速度为1M
669
670     NRF24L01_Set_TxAddr( &addr[0], 5 ); //设置TX地址
671     NRF24L01_Set_RxAddr( 0, &addr[0], 5 ); //设置RX地址
672 }
673
```


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。