

SX127x Demo 说明

1. Demo 使用说明

硬件：STM32F103ZE_{xx}、RFM95/RFM96/RFM97/RFM98.

平台：keil5.

文件说明：

模拟 SPI：STM32F103ZE_sx127x_模拟 SPI

硬件 SPI：STM32F103ZE_sx127x_硬件 SPI

RFM98：137~525MHZ

RFM97：SF 6~9

2. 驱动文件组成

sx1276Regs-Fsk.h:

- (1) 主要定义 sx127x, Fsk 的寄存器。

sx1276Regs-LoRa.h:

- (1) 主要定义 sx127x, LoRa 的寄存器。

board-config.h:

- (1) 主要定义 GPIO 的相关宏定义。

sx1276-board.h:

- (1) 定义 GPIO 的初始化函数。
- (2) 定义相关控制引脚的函数。

sx1276-board.c:

- (1) 实现 GPIO 的初始化函数。
- (2) 实现相关控制引脚的函数。

SX1276.h:

- (1) 主要定义 sx127x 的 API 接口函数。

SX1276.c:

- (1) 实现定义 sx127x 的 API 接口函数。

radio.h:

- (1) 定义 sx127x 的应用接口函数。
- (2) 定义 sx127x 的中断事件回调函数。

app.c

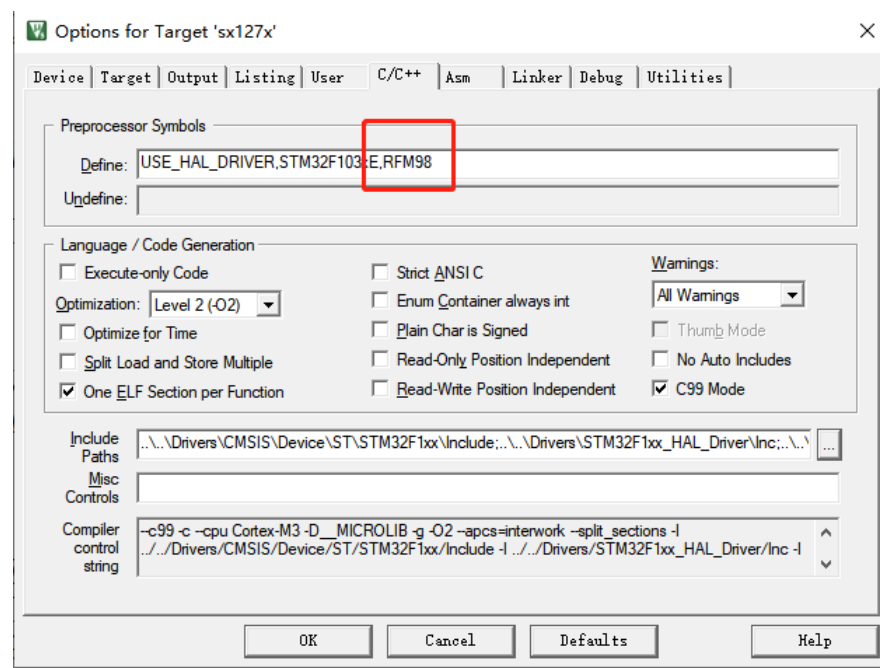
- (1) 实现 sx127x 的 ping pong 应用。

3. 使用说明

(1) 以下宏定义可对 Frequency、Power、Bandwidth、Spread factor、Coding Rate 等 LoRa 参数进行修改。

```
#define RF_FREQUENCY 434000000 // Hz
#define TX_OUTPUT_POWER 14 // dBm
#define LORA_BANDWIDTH 0 // [0: 125 kHz,
// 1: 250 kHz,
// 2: 500 kHz,
// 3: Reserved]
#define LORA_SPREADING_FACTOR 9 // [SF7..SF12]
#define LORA_CODINGRATE 1 // [1: 4/5,
// 2: 4/6,
// 3: 4/7,
// 4: 4/8]
#define LORA_PREAMBLE_LENGTH 16 // Same for Tx and Rx
#define LORA_SYMBOL_TIMEOUT 0 // Symbols
#define LORA_FIX_LENGTH_PAYLOAD_ON false
#define LORA_IQ_INVERSION_ON false
```

(2) 宏定义模块名称。



(3) 烧录运行之后实验现象通过串口打印信息

两块模块通讯成功现象：

```
[2023-08-08 16:46:18.942]# RECV ASCII>  
RX:PONG  
  
[2023-08-08 16:46:19.447]# RECV ASCII>  
SEND:PING  
  
[2023-08-08 16:46:20.089]# RECV ASCII>  
RX:PONG  
  
[2023-08-08 16:46:20.591]# RECV ASCII>  
SEND:PING
```

两块模块通讯失败现象：

```
[2023-08-08 17:12:44.425]# RECV ASCII>  
RX_TIMEOUT  
  
[2023-08-08 17:12:44.987]# RECV ASCII>  
SEND:PING  
  
[2023-08-08 17:12:46.898]# RECV ASCII>  
RX_TIMEOUT  
  
[2023-08-08 17:12:47.461]# RECV ASCII>  
SEND:PING
```

4. 移植说明

(1) radio.h、SX1276.h、SX1276.c 可直接移植不用做修改。

(2) 主要修改 board-config.h 和 sx1276-board.c 文件。

修改对 GPIO 引脚和输入、输出寄存器的宏定义。对于 Dio2、Dio3、Dio4 有需要的话自行配置。

```

#define SX1276_Port1      GPIOF

#define SX1276DIO0_Pin    GPIO_PIN_0
#define SX1276DIO1_Pin    GPIO_PIN_1
#define SX1276POR_Pin     GPIO_PIN_2
#define SX1276Nss_Pin     GPIO_PIN_3
#define SX1276Switch_Pin  GPIO_PIN_7//暂时没用到

#define SX1276_Port2      GPIOA
#define SX1276SCK_Pin     GPIO_PIN_5
#define SX1276MISO_Pin    GPIO_PIN_6
#define SX1276MOSI_Pin    GPIO_PIN_7

//#define SX1276DIO2_Pin   GPIO_PIN_3
//#define SX1276DIO3_Pin   GPIO_PIN_4
#define SX1276TXCO_Pin    GPIO_PIN_5//暂时没用到

#define Dio0_INT_IRQn      EXTI0_IRQn
#define Dio0_INT_IRQHandler EXTI0_IRQHandler

#define Dio1_INT_IRQn      EXTI1_IRQn
#define Dio1_INT_IRQHandler EXTI1_IRQHandler

#define SetSX1276SCK()     (SX1276_Port2->BSRR = SX1276SCK_Pin)
#define ClrSX1276SCK()     (SX1276_Port2->BRR = SX1276SCK_Pin)
#define SetSX1276MOSI()    (SX1276_Port2->BSRR = SX1276MOSI_Pin)
#define ClrSX1276MOSI()    (SX1276_Port2->BRR = SX1276MOSI_Pin)

#define SetSX1276POR()     (SX1276_Port1->BSRR = SX1276POR_Pin)
#define ClrSX1276POR()     (SX1276_Port1->BRR = SX1276POR_Pin)
#define SetSX1276Nss()     (SX1276_Port1->BSRR = SX1276Nss_Pin)
#define ClrSX1276Nss()     (SX1276_Port1->BRR = SX1276Nss_Pin)
#define SetSX1276TXCO()    (SX1276_Port1->BSRR = SX1276TXCO_Pin)//暂时没用到
#define ClrSX1276TXCO()    (SX1276_Port1->BRR = SX1276TXCO_Pin) //暂时没用到

//引脚判断
#define SX1276MISO()       (SX1276_Port2->IDR & SX1276MISO_Pin)
#define SX1276DIO1()       (SX1276_Port1->IDR & SX1276DIO1_Pin)

```

修改对 GPIO 初始化的操作。注意对 Dio0、Dio1 口做上升沿中断处理，射频状态机主要由检测 Dio0、Dio1 的中断，在中断回调函数中进行。

```

void SX1276IoInit( void )
{
    GPIO_InitTypeDef gpio_init_struct;

    SX1276_GPIO_CLK_ENABLE();

    gpio_init_struct.Pin = SX1276Nss_Pin;           /* Nss引脚 */
    gpio_init_struct.Mode = GPIO_MODE_OUTPUT_PP;    /* 推挽输出 */
    gpio_init_struct.Speed = GPIO_SPEED_FREQ_HIGH; /* 高速 */
    HAL_GPIO_Init(SX1276_Port1, &gpio_init_struct);

    gpio_init_struct.Pin = SX1276MOSI_Pin;          /* MOSI引脚 */
    HAL_GPIO_Init(SX1276_Port2, &gpio_init_struct);

    gpio_init_struct.Pin = SX1276SCK_Pin;           /* SCK引脚 */
    HAL_GPIO_Init(SX1276_Port2, &gpio_init_struct);

    gpio_init_struct.Pin = SX1276MISO_Pin;          /* MISO引脚 */
    gpio_init_struct.Mode = GPIO_MODE_INPUT;        /* 输入 */
    gpio_init_struct.Pull = GPIO_NOPULL;
    HAL_GPIO_Init(SX1276_Port2, &gpio_init_struct);

    gpio_init_struct.Pin = SX1276DIO0_Pin;          /* DIO0引脚 */
    gpio_init_struct.Mode = GPIO_MODE_IT_RISING;    /* 输入 */
    gpio_init_struct.Pull = GPIO_NOPULL;
    HAL_GPIO_Init(SX1276_Port1, &gpio_init_struct);
    HAL_NVIC_SetPriority(Dio0_INT_IRQn, 0, 2);
    HAL_NVIC_EnableIRQ(Dio0_INT_IRQn);              /* 抢占0, 子优先级2 */
                                                    /* 使能中断线1 */

    gpio_init_struct.Pin = SX1276DIO1_Pin;          /* DIO1引脚 */
    HAL_GPIO_Init(SX1276_Port1, &gpio_init_struct);
    HAL_NVIC_SetPriority(Dio1_INT_IRQn, 0, 2);
    HAL_NVIC_EnableIRQ(Dio1_INT_IRQn);              /* 抢占0, 子优先级2 */
                                                    /* 使能中断线1 */

    SetSX1276Nss();
    ClrSX1276SCK();
}

```

修改对收发器复位的操作。

```
void SX1276Reset( void )
{
    GPIO_InitTypeDef gpio_init_struct;

    // Enables the TCXO if available on the board design
    // SX1276SetBoardTcxo( true );

    SX1276IoInit();

    // Set RESET pin to 0
    gpio_init_struct.Pin = SX1276POR_Pin;           /* Reset引脚 */
    gpio_init_struct.Mode = GPIO_MODE_OUTPUT_PP;     /* 推挽输出 */
    gpio_init_struct.Speed = GPIO_SPEED_FREQ_HIGH;  /* 高速 */
    HAL_GPIO_Init(SX1276_Port1, &gpio_init_struct);
    ClrSX1276POR();

    // Wait 1 ms
    delay_ms( 1 );

    SetSX1276POR();
    // Configure RESET as input
    gpio_init_struct.Pin = SX1276POR_Pin;           /* reset引脚 */
    gpio_init_struct.Mode = GPIO_MODE_INPUT;        /* 输入 */
    gpio_init_struct.Pull = GPIO_PULLUP;
    HAL_GPIO_Init(SX1276_Port1, &gpio_init_struct);

    // Wait 6 ms
    delay_ms( 6 );
}
```

Dio 中断函数的处理

```
static void DioIrqHanlderProcess( uint8_t index )
{
    switch(index)
    {
        case 0:
            SX1276nDio0Irq();
            break;
        case 1:
            SX1276nDio1Irq();
            break;
        case 2:
            SX1276nDio2Irq();
            break;
        case 3:
            SX1276nDio3Irq();
            break;
        default:
            break;
    }
}

void Dio0_INT_IRQHandler(void)
{
    __HAL_GPIO_EXTI_CLEAR_IT(SX1276DIO0_Pin);
    DioIrqHanlderProcess( 0 );
}

void Dio1_INT_IRQHandler( void )
{
    __HAL_GPIO_EXTI_CLEAR_IT(SX1276DIO1_Pin);
    DioIrqHanlderProcess( 1 );
}
```