

ZM470SX-DEMO

470 模块

UM01010101

V1.00

Date: 2015/9/25

产品用户手册

类别	内容
关键词	无线通信；470 模块；驱动例程；LORA 调制
摘 要	



修订历史

版本	日期	原因
V1.00	2015/9/25	创建文档

目 录

1. 产品简介.....	1
2. 功能测试.....	2
2.1 安装驱动.....	2
2.2 驱动例程简介.....	2
2.3 通信.....	2
2.3.1 按键发送数据.....	2
2.3.2 串口收发数据.....	2
2.4 按键调节发射功率.....	3
3. ZM470SX 模块硬件指南	5
3.1 硬件电路.....	5
3.2 主要寄存器描述.....	5
3.2.1 中断状态.....	5
3.2.2 中断使能控制寄存器.....	6
3.3 ISP 下载程序.....	6
3.4 调试接口.....	7
4. ZM470SX 驱动简介	8
4.1.1 射频参数配置.....	8
4.1.1 初始化模块.....	8
4.1.2 发送数据.....	8
4.1.3 主函数流程.....	9
5. 解决故障.....	12
6. 免责声明.....	13

1. 产品简介

ZM470SX-M 是广州致远电子针对 470MHz 频段远距离通信需求而开发的超值无线模块。ZM470SX-DEMO 是为帮助用户快速上手 ZM470SX-M 模块而开发的评估套件，评估板实物如图 1.1 所示。**ZM470SX-DEMO 这个型号包含了两个板子。**

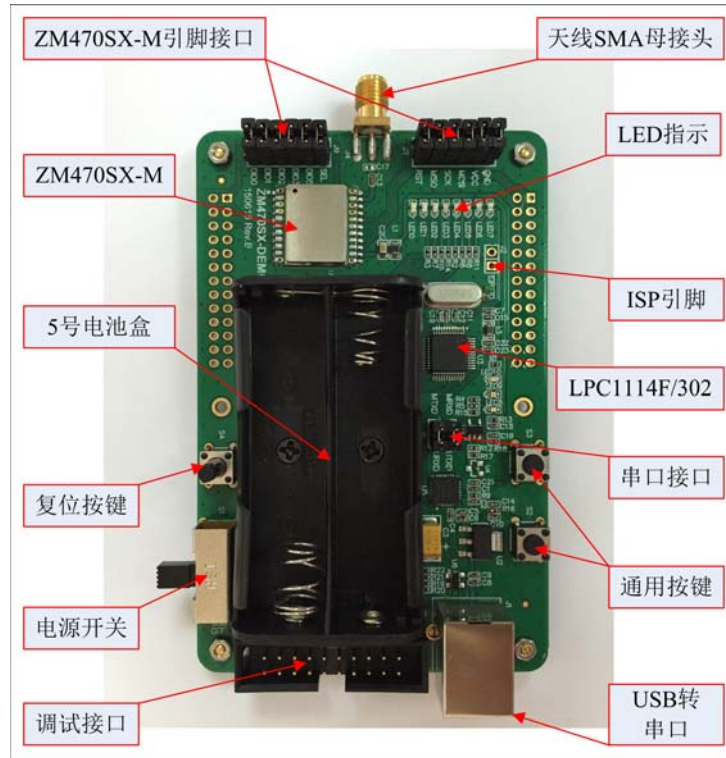


图 1.1 评估板实物图

评估板的正面安装有电池盒，评估板套件不带电池，需要客户自备 2 节 5 号电池，当然也可以用 USB 供电。评估板部件相关描述见表 1.1。

表 1.1 评估板部件描述

部件	描述
电源开关	选择电源供电方式，USB 或电池
串口(USB 形式)	USB 转串口，用 USB 电缆直接连到电脑，电脑需要安装相应的驱动， 可用来下载程序和数据收发
按键	通用按键，可作为外中断输入
LED 指示灯	指示灯
调试接口	可用 J-Link 进行调试
天线	天线需要用频点在 470MHz，阻抗 50Ω 的
ZM470SX-M	470 MHz 无线收发模块
ISP 选择	先短接 J2 和（串口接口）J6，然后复位 MCU 进入 ISP 状态
模块引脚接口	需要用短路器短接，否则模块引脚与 MCU 引脚是断开的
电池盒	电池输出经电源开关（S1）连接到 VCC 网络

2. 功能测试

2.1 安装驱动

使用前请在产品光盘中的 CP210x_VCP_Windows.zip 找到 CP210X 驱动程序，安装完毕后再进行后面的操作。如果需要使用最新的驱动程序可到 Silicon Labs 官方网站下载，下载连接：<http://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx>。

2.2 驱动例程简介

产品出厂时，我们下载了附件中的 keil-zm470SX V1.00.zi 程序到评估板上。如需要下载其它的程序，用户可通过串口自行将程序下载到板子上，具体操作步骤在下文中的 ISP 下载程序小节进行讲解。出厂程序具备了本章后面介绍的两个功能：通信和功率调整。

2.3 通信

产品出厂时，我们已将两个板子的程序设置为可以相互通信：一方面可以通过按键发送固定的数据，另一方面可以将串口接收到的数据发送出去。程序上默认采用 LoRa 调制模式，空中传输速率约 2kbps，并且没有数据重发和缓冲溢出处理等措施。

2.3.1 按键发送数据

按下 S2 发送一包数据，发送数据期间 LED7 亮起，发送结束 LED7 熄灭。如果另一个评估板接收到另一个评估板发送的数据，LED0~LED6 指示灯会有（计数加 1 形式的）闪动，同时会把接收到的数据从串口发送出去。

当一个人拿两个评估板测试距离时，可以把其中一个评估板上 J6 的 MTXD 短接到 MRXD，放在离地面 1 米高以上的位置，并且尽量使天线垂直向上；手持另一块评估板移动，手持那个评估板的 J6 需要正常短接如图 1.1 所示。手持的评估板按下 S2 会发一包数据出去，如果固定的那个评估板能正确收到数据，那么它会把收到的数据回传；若手持的评估板也能正确收到数据，则 LED0~LED6 指示灯会有（计数加 1 形式的）闪动，并把数据从串口发送出去，串口发送数据期间板上的 LED8 会闪动。

2.3.2 串口收发数据

使用配送的 USB 通讯电缆分别连接电脑和 2 块评估板，把 J6、J7 和 J9 上的排针用短路器短接（默认情况下，出厂时已短接好），分别打开两个串口调试软件（如丁丁串口调试助手等），对应评估板 1 和评估板 2。选择对应的串口号，设置波特率为 115200，数据位为 8，停止位为 1，校验位为无，流控制为无。在发送窗口填入数据（不能超过 60 个字节，否则后面的数据会丢失），然后点击发送，板上的 LED9 会闪烁，在另一个已打开的串口软件中将收到刚发送的数据。如图 2.1 和图 2.2 所示。



图 2.1 串口 1



图 2.2 串口 2

2.4 按键调节发射功率

按下 S3 可循环调节发射功率，每按一下会增加 1dB，如果超出最大值则回到最小值，调整范围：5dBm~20dBm。按下 S3 后 LED0~LED4 指示灯指示当前的发射功率，对应的功率关系如表 2.1 所示；1 代表 LED 亮，0 代表灭。



表 2.1 LED 与发射功率对应关系

LED4~LED0 状态		10100	10011	10010	01000	00111	00110	00101
发射 功率	单位 dBm	20	19	18	8	7	6	5
	单位 mW	100	80	64	6.4	5.0	4.0	3.2

3. ZM470SX 模块硬件指南

这部分内容针对 ZM470SX 模块的电路设计，SX1278 的几个重要的寄存器，评估板的程序下载，模块的布局注意事项和模块的部分驱动程序进行介绍。

3.1 硬件电路

ZM470SX 模块必须配合 MCU 才能使用，模块通过表 3.1 所示引脚与 MCU 相连。ZM470SX 模块与 MCU 之间通过 SPI 总线进行通信；ZM470SX 模块通过状态寄存器和 DIOx 引脚将状态反馈给 MCU。

表 3.1 ZM470SX 模块引脚功能

引脚	描述
GND	模块地引脚。
VCC	模块电源引脚。供电不可超过 3.6V，否则烧坏模块，可通取下 VCC 标注的跳帽后串电流表上去测试电流。
MOSI	SPI 总线从机输入引脚。当 MCU 向模块写数据时，数据从此引脚写入。
SCK	SPI 总线时钟引脚，常态为高电平。
MISO	SPI 总线从机输出引脚。当 MCU 读取模块数据时，模块数据由此引脚输出。
RST	复位引脚，低电平有效
SEL	SPI 片选引脚，低电平有效。
DIO5	中断输出引脚，可编程决定
DIO3	中断输出引脚，可编程决定
DIO2	中断输出引脚，可编程决定
DIO1	中断输出引脚，可编程决定
DIO0	可编程决定，常用作数据发送完成和接收到数据这两个中断输出

原理图设计可参考《ZM470SX-DEMO sch.pdf》，模块的 SPI 引脚尽量加上上拉电阻，评估板的设计案例中使用了 MCU 的片内上拉功能而没有外加上拉电阻。为提高模块的接收灵敏度，尽量使用线性稳压器供电而不用开关电源；布局设计中，尽量使模块远离噪声源（如晶体和时钟信号线等），天线安置在较开阔的地方并远离大电容等金属体。另外，射频产品很忌讳系统中的某个方波信号频率的整数倍恰好是模块的载波频率，例如，模块载波设 470MHz，而模块附近有个 47 MHz 峰峰值为 3.3V 方波信号，那么这个信号的其中一个谐波恰好是 470MHz，因此这个信号会降低模块的接收灵敏度从而导致通信距离下降，当然可把载波频率设为 471 MHz 解决这种问题。

3.2 主要寄存器描述

附件中的《SX1278_ag.pdf》末尾部分有针对 SX1278 寄存器有专门的描述章节，故在此仅对几个常用的寄存器进行简要描述。用户在使用我们的驱动例程过程中也没有必要详细了解 SX1278 寄存器，了解例程中的相关函数后即用我们模块实现数据收发。

3.2.1 中断状态

地址：0x12

LoRa 模式中断状态寄存器功能描述如表 3.2 所示。当某位对应的事件发生后，此位置 1，若此时中断使能控制寄存器（地址：0x11）对应的位事先写 0，则相应 DIO 引脚输出高电

平；此后，向相应的位写 1 可清除中断，同时相应的 DIOx 变为低电平。

表 3.2 中断状态寄存器

Add	名称	功能描述
Bit7	RxTimeout	接收超时中断，若产生中断此位置 1
Bit6	RxDone	包接收完成中断，当接收到一个有效数据包时，此位置 1
Bit5	PayloadCrcError	接收数据包 CRC 校验出错时，此位置 1
Bit4	ValidHeader	接收到有效包头时，此位置 1
Bit3	TxDone	FIFO 发送完成一包数据时，此位置 1
Bit2	CadDone	通道检测完成后，此位置 1
Bit1	FhssChangeChannel	通道切换完成后，此位置 1
Bit0	CadDetected	在 Cad 操作时检测到 LoRa 信号后，此位置 1

3.2.2 中断使能寄存器

地址：0x11

这个寄存用来使能的失能中断，它的每一位与中断状态寄存器（0x12）一一对应。当其中的某些位为 0 时，表示这些位对应的中断开启；反之，则表示关闭。例如，需要打开接收完成和 CRC 错误这两个中断时，应向该寄存器里写入 0x9f。如果需要当收到有效数据包后 DIO0 产生高电平，那么还需要将 DIO0 引脚功能设置为 RxDone 选项，具体操作详见致远电子提供的驱动例程。当中断产生后，可通过读取中断状态寄存器来确认到底是哪些中断产生，而向中断状态寄存器写某些位写 1 时可以清除相应位中断的标志，并使 DIOx 返回到低电平。

3.3 ISP 下载程序

焊接在板上的 MCU 是 NXP 的 LPC1114F/302，开发过程中生成的 hex 文件可用 Flash Magic 软件烧写到 MCU 的 Flash 中。Flash Magic 按图 3.1 所示配置后，把板上的 J2 和 J6 分别短接，用配送的 USB 通讯电缆连接 PC 和评估板的 J1，然后给评估板上电（S1 打到 USB 档位），最后点击 Flash Magic 的 Start 下载程序。下载完成后，在 J2 没有短接情况下重新上电，MCU 运行刚下载的程序；若 J2 短接上电，MCU 进入 ISP 模式。

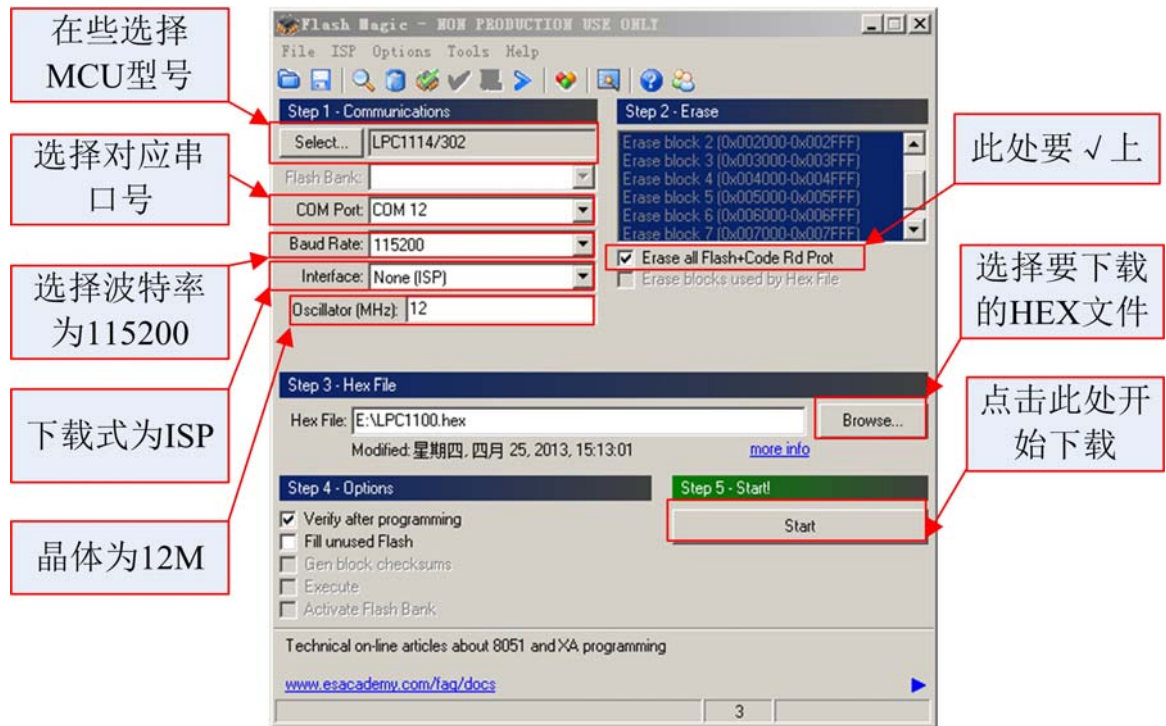


图 3.1 Flash Magic 配置

3.4 调试接口

评估板上的 J8 是 SW 调试接口，可以 J-Link 进行烧写程序及仿真。调试时，接 SWDIO、SWCKL、VT 和 GND 即可。J8 的引脚及名称如表 3.3 所示，其中第 1 脚的焊盘是方形的，第 9 和第 11 脚位于牛角座的缺口处。

表 3.3 调试引脚接口

引脚	2	4	6	8	10	12	14	16	18	20
名称	NC		GND	GND	GND	GND	GND	GND	GND	GND
引脚	1	3	5	7	9	11	13	15	17	19
名称	VT	NC	NC	SWDIO	SWCKL	NC	NC	RST	NC	NC

4. ZM470SX 驱动简介

在我们提供的 ZM470SX 开发资料压缩包中包含了两个驱动例程：“sx1278 官方驱动-SX12xxDrivers-2.0.0.zip”和“致远驱动例程-keil-zm470SX V1.00.zip”，但只对后面这个驱动程序做技术支持。我们会根据用户在实际应用中反应的问题作相应修改后并更新到致远电子官网上，用户可从我们的网站上下载得到最新驱动程序。ZM470SX 支持传统的调制方式（OOK 和 FSK）和 LoRa 调制方式，由于这个模块的优势在独有的 LoRa 调制方式，所以我们提供的驱动例程也仅适用于 LoRa 调制方式。

4.1.1 射频参数配置

ZM470SX 模块可以从程序上设定与射频相关的参数：载波频率、发射功率、带宽（BW）、扩频因数（SF）和编码率（CR）等。在 LoRa 调制模式下并没有明确的通信速率的概念，但有等效通信速率，最高可以 37.5kbps，欲了解等效通信速率的介绍可参考《SX1278_ag.pdf》第 30 页前后的内容。通信速率与带宽、扩频因数和编码率相关，带宽越大速率越高，扩频因数越大速率越低，编码率越大速率越低。

程序清单 4.1 射频参数初始化值

```
tLoRaSettings LoRaSettings =
{
    470000000,      /* 载波频率*/
    20,             /* 发射功率*/
    9,              /* 带宽 [0: 7.8kHz, 1: 10.4 kHz, 2: 15.6 kHz, 3: 20.8 kHz, 4: 31.2 kHz,
                    5: 41.6 kHz, 6: 62.5 kHz, 7: 125 kHz, 8: 250 kHz, 9: 500 kHz, other: Reserved] */
    10,             /* 扩频因数[6: 64, 7: 128, 8: 256, 9: 512, 10: 1024, 11: 2048, 12: 4096 chips] */
    2,              /* ErrorCoding [1: 4/5, 2: 4/6, 3: 4/7, 4: 4/8]
    true,           /* CRC 校验[0: OFF, 1: ON] */
    false,          /* ImplicitHeaderOn [0: OFF, 1: ON] /* 包头模式*/
    0,              /* RxSingleOn [0: Continuous, 1 Single] /* 固定为设置*/
    0,              /* FreqHopOn [0: OFF, 1: ON] /* 固定为设置*/
    4,              /* HopPeriod Hops every frequency hopping period symbols
    100,            /* TxPacketTimeout
    100,            /* RxPacketTimeout
    128,            /* PayloadLength (used for implicit header mode)
};
```

4.1.1 初始化模块

在初始化模块前，先调用 `Radio = RadioDriverInit()` 这个函数来，从而使与操作模块相关的函数被函数结构体指针 `Radio` 所向。在后面的操作中都通过操作这个函数指针来实现与操作原函数相同的功能。

初始模块函数：`Radio->Init()`。执行这个函数后，将对 SX1278 的寄存器进行配置，即根据结构体 `LoRaSettings` 的参数要求来配置无线模块的寄存器。

4.1.2 发送数据

当需要发送数据时调用 `uint8_t LoRaTxData(uint8_t *pbuf, uint8_t size)` 这个函数，函数

的第一个入参是被发送数据的指针，第二个入参数是数据长。这个函数里有一发送等待，直到数据发送完成才跳出该函数，这一点可能会导致程序的实时性被破坏，用户可根据实现需要来修改这个发送等待部分的程序。函数流程大体是这样执行的，首先把射频开关切换到天线与发送电路连通状态，然后填充数据到发送缓冲区，接着使模块进入发送状态，最后等待发送完成并跳出函数。

4.1.3 接收函数

ZM470SX-M 模块只有在接收状态下才有可以接到数据。因此需要接收发送发出来的数据时，必先调 `Radio->RFRxStateEnter()` 这个函数。程序执行完该函数后，一方面后把射频开关切换到天线与接收电路连通状态；另一方面使模块进入接状态；与此同时还打开了接收到有效数据中断：收到数据包后 DIO0 引脚由低电平跳变为高电平。

若模块开启了接收中，那么在模块接到数据后 DIO0 引脚会跳变为高电平，此时单片机可以通过 `Radio->RFRxDataRead(&ucRxbuf[0], &j)` 这个函数读取模块接收到的数据。函数的第一个入参数是存放数据的指针，第二个入参是需要读取数据长度的指针。函数内部会将接收中断清除，该函数执行结束后，DIO0 引脚变低电平状态。

4.1.4 休眠状态

`Radio->RFOpModeSet(RFLR_OPMODE_SLEEP)` 当从休眠状切换到其它新状态时，直接调用所需要状态的函数即可。比如从休眠切换到接收状态，在休眠状态时直接调用 `Radio->RFRxStateEnter()` 即可进入接收状态。

4.1.5 主函数流程

出厂程序的主函数如下面的程序清单所示，板子上电后首先初始化 MCU 和串口，然后初始化 ZM470SX 模块并使它进入接收状态，接着点亮 LED，最后进入循环体。在循环里有接收到有效数据包，发送数据包，调整发送功率的处理。在进入接收状态的函数里面打开了接收到有效数据包中断，当接收到有效数据包时，DIO0 引脚拉高，第一 if 语句为真，在 if 语句里面把 ZM470SX 模块接收缓冲区里的数据读出来并发送出去。第二个 if 里面处理数据发送部分，当按键（S2）中断产生或串口接收到数据都会使这个 if 语句成立，该 if 语句里面把缓冲区的数据通过 ZM470SX 模块发送出去。再往后的一个 if 是休眠处理，如果成立那么，模块进入休眠状态。再接着是调整功率处理，当按下 S3 键后，if 条件为真，if 语句内部进行功率调整。最后是延时处理。

程序清单 4.2 主函数流程

```
int main(void)
{
//    double crssi,prssi;
    uint8_t   PwrFlag       = UPwr;
    uint8_t   j              = 0;
    uint16_t   i              = 0;
    uint16_t   temp          = 0x5f;
    uint8_t   tempbuf[PACK_LENGTH] = {0};

    SystemInit();                                /* 系统初始化；晶体.0592MHz    */
    LPC_SYSCON->SYSAHBCLKCTRL |= (1 << 6);
    LED_INIT();                                  /* 初始化 LED                */
    uartInit();                                  /* 初始化 UART                */
    KEY3_INT();                                  /* 初始按钮 S3 中断;对应功率 */
}
```



```

KEY2_INT();                                /* 初始按钮 S2 中断 */
Radio = RadioDriverInit();                 /* 注册相关的函数 */
Radio->Init();                             /* 初始无线模块 */
Radio->RFRxStateEnter();                   /* 进入接收状态 */
LED_Show(0x18);
while (1) {
    if (1 == Radio->RFDio0State()) {        /* 如果接收到数据 DIO0 引脚引高 */
        Radio->RFRxDataRead( &ucRxbuf[0], &j ); /* 把 RF 芯片 FIFO 中的数据读出来 */
        Radio->RFRxStateEnter();           /* 进入接收状态 */
        if(j > 0) {
            for (temp = 0; temp < j; temp++) {
                uartSendByte(ucRxbuf[temp]); /* 把数据从串发送出去 */
            }
            led++;                          /* 收到数据计数加 */
            led = led % 128;
            LED_Show(led);
        }
    }
    /* 发送 */
    if(uiByetNum >= PACK_LENGTH || uiMs == 1) { /* 若缓冲区满或者接收超时 */
        DisableIRQ();                       /* 失能中断,防止 SPI 时序混乱 */
        NVIC_DisableIRQ(UART_IRQn);        /* 失能 UART 中断 */

        for (i = 0; i < uiByetNum; i++) {
            tempbuf[i] = ucTxbuf[i];        /* 把串口缓冲区的数据读出来 */
        }
        uiByetNum = 0;                     /* 缓冲区数据标志清零 */
        NVIC_EnableIRQ(UART_IRQn);         /* 使能 UART 中断 */
        LED_Show(0x80);
        Radio->RFTxData(&tempbuf[0], i);    /* 发送数据出去 */
        Radio->RFRxStateEnter();           /* 进入接收状态 */
        EnableIRQ();                       /* 恢复按键中断 */
        LED_Show(led);
        uiMs = 0;
        // ucsleep = 1;
    }
    /* 休眠 */
    if (ucsleep == 1) {
        Radio->RFOpModeSet(RFLR_OPMODE_SLEEP); /* 进入休眠,电流.2uA */
        ucsleep = 0;
    }
    if(PwrFlag != UPwr) {
        PwrFlag = UPwr;
        LED_Show(UPwr);
    }
}

```



```
        Radio->RFTxPower(UPwr);                /* 调整发射功率          */
    }
    if (uiByetNum > 0) {
        dellayxms(1);                            /* 误差很小的 ms 延时          */
        if(uiMs > 0) {
            uiMs--;                               /* 等待串口接收延时减          */
        }
    }
}
}
```

5. 解决故障

1. 通过 ZM470SX-M 的电流可以判断其工作状态，处在准备状态时电流在 2mA，处在接收状态时电流在 14mA 左右。
2. 设置频率通道时，最好将步进设置大于 2MHz，因为步进太小的话相互间的干扰比较大，并且还可能接收到邻道的信息。
3. 当读取到 CRC 校验错误后，最好复位接收 FIFO 然后再进入接收状态。
4. 为加大链路预算（间接等价于通信距离）可通过降低数据传输速度来实现，具体操作是增大 SF 或者减小 BW。

6. 免责声明

应用信息

本应用信息适用于 ZM470SX-DEMO 的操作使用。客户在开发产品前必须根据其产品特性给与修改并验证。

修改文档的权利

本手册所陈述的产品文本及相关软件版权均属广州致远电子有限公司所有，其产权受国家法律绝对保护，未经本公司授权，其它公司、单位、代理商及个人不得非法使用和拷贝，否则将受到国家法律的严厉制裁。广州致远电子有限公司保留在任何时候修订本用户手册且不需通知的权利。

您若需要我公司产品及相关信息，请及时与我们联系，我们将热情接待。

销售与服务网络

广州致远电子股份有限公司

地址：广州市天河区车陂路黄洲工业区 7 栋 2 楼

邮编：510660

传真：(020)38601859

网址：www.zlg.cn



全国销售与服务电话：400-888-4005

全国服务电话：400-888-4005

销售与服务网络：

广州总公司

广州市天河区车陂路黄洲工业区 7 栋 2 楼

电话：(020)28267985 22644261

上海分公司—上海

上海市北京东路 668 号科技京城东楼 12E 室

电话：(021) 53865521 53083451

北京分公司

北京市海淀区知春路 108 号豪景大厦 A 座 19 层

电话：(010)62536178 62635573

上海分公司—南京

南京市珠江路 280 号珠江大厦 1501 室

电话：(025) 68123923 68123920

深圳分公司

深圳市福田区深南中路 2072 号电子大厦 12 楼

电话：(0755)83640169 83783155

上海分公司—杭州

杭州市天目山路 217 号江南电子大厦 502 室

电话：(0571)89719491 89719493

武汉分公司

武汉市洪山区广埠屯珞瑜路 158 号 12128 室(华中电脑数码市场)

电话：(027) 87168497 87168397

重庆分公司

重庆市九龙坡区石桥铺科园一路二号大西洋国际大厦（赛格电子市场）2705 室

电话：(023)68796438 68797619

成都分公司

成都市一环路南二段 1 号数码科技大厦 403 室

电话：(028) 85439836 85432683

西安办事处

西安市长安北路 54 号太平洋大厦 1201 室

电话：(029)87881295 87881296

请您用以上方式联系我们，我们会为您安排样机现场演示，感谢您对我公司产品的关注！