

# ZM433SX-L-DEMO

## 433 模块

UM01010101

V1.00

Date: 2016/6/6

产品用户手册

类别	内容
关键词	无线通信；433 模块；驱动例程；
摘要	

## 修订历史

版本	日期	原因
V1.00	2016/6/6	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. ZM433SX 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
3.3 ISP 下载程序.....	6
3.4 调试接口.....	7
4. ZM433SX-L 驱动简介.....	8
4.1.1 模块寄存器配置.....	8
4.1.1 初始化模块.....	9
4.1.2 发送数据.....	9
4.1.3 接收函数.....	10
4.1.4 主函数流程.....	10
5. 解决故障.....	13
6. 免责声明.....	14

## 1. 产品简介

ZM433SX-L 是广州致远电子针对 433MHz 频段远距离通信需求而开发的低功耗无线模块。ZM433SX-L-DEMO 是为帮助用户快速上手 ZM433SX-L 模块而开发的评估套件，评估板实物如图 1.1 所示。ZM433SX-L-DEMO 这个型号包含了两个板子，一个发送，一个接收。

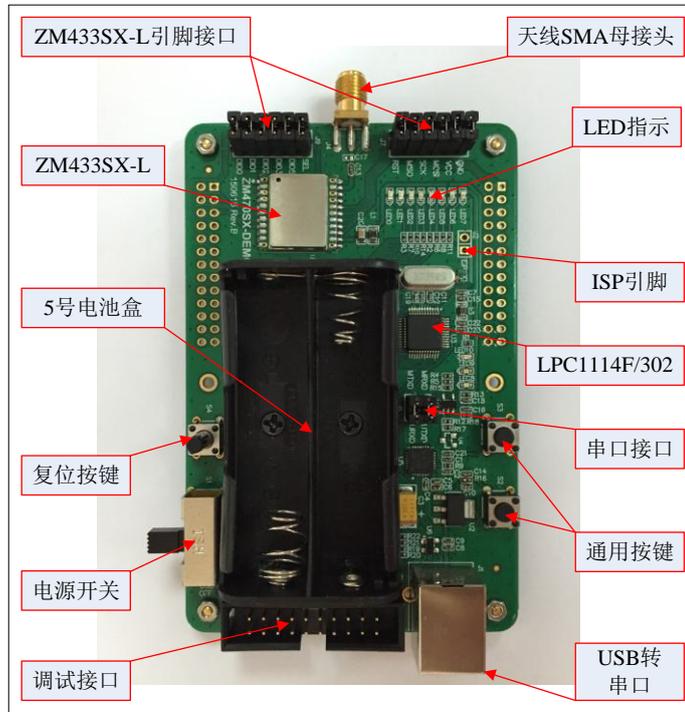


图 1.1 评估板实物图

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， 可用于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 433MHz，阻抗 50Ω 的
ZM433SX-L	433 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-ZM433SX-L V1.00 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**错误! 找不到引用源**。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 FSK 调制模式, 空中传输速率约 25kbps, 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。

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。

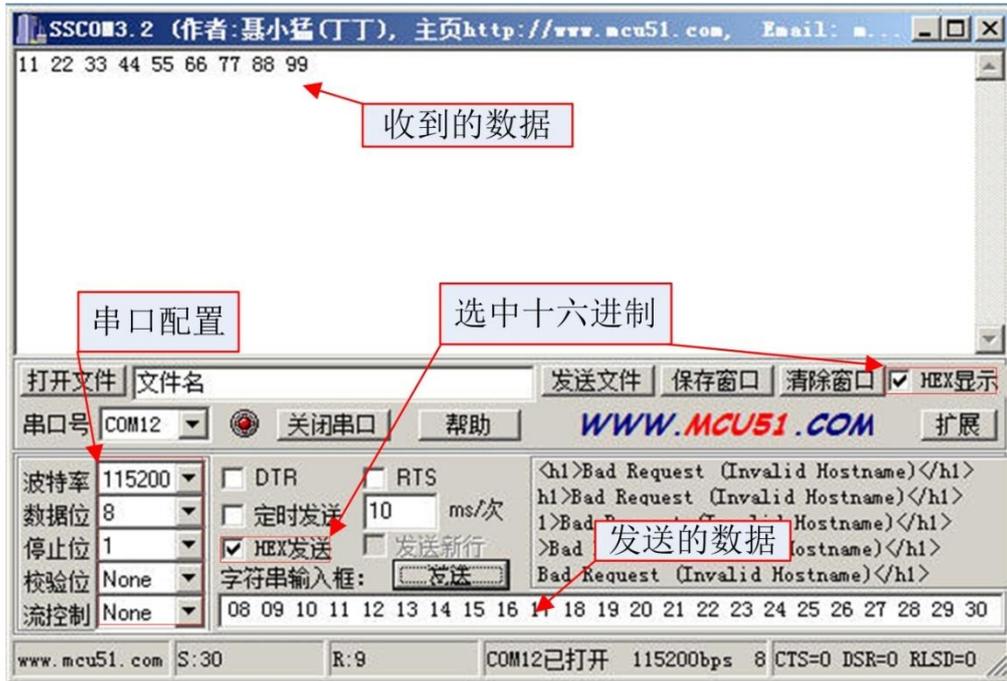


图 2.1 串口 1



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，每按一下会减少 3dB，如果超出最大值则回到最小值，调整范围：12dBm~8dBm。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

LED2~LED0 状态		000	001	010	011	100	101	110	111
发射 功率	单位 dBm	12	9	6	3	0	-3	-6	-9
	单位 mW	15.8	7.94	3.98	1.99	1	0.50	0.25	0.125

### 3. ZM433SX-L 模块硬件指南

#### 3.1 硬件电路

ZM433SX-L 模块必须配合 MCU 才能使用，模块通过表 3.1 所示的短路排针 J7、J9 与 MCU 相连。用户可以把排针的短路器取下，用杜邦线连接模块引脚和自己的核心板来调试 ZM433SX-L 模块。

表 3.1 ZM433SX-L-DEMO 排针功能

排针	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。
CSEL	配置模块寄存器时，SPI 片选引脚，低电平有效。
DSEL	读写 FIFO 数据时，SPI 片选引脚，低电平有效。
PLL	锁相环锁定输出，正常上电工作后，输出高电平。
DAT	数据输入和输出(包模式该引脚无效)。
CLK	时钟输出，模块正常上电，会输出 427.4Khz 方波。
IRQ1	中断输出，功能由编程决定。
IRQ0	中断输出，功能由编程决定。

原理图设计可参考《ZM433SX-L-DEMO sch.pdf》，模块的 SPI 引脚尽量加上拉电阻，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 433MHz，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 433MHz，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《SX1212.pdf》有针对 SX1212 寄存器的描述章节，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 SX1212 寄存器，了解例程中的相关函数后即可用我们模块实现数据收发。

##### 3.2.1 中断状态

地址：13 (0x0d)

Packet 模式(例程使用的模式是包模式)中断状态寄存器功能描述如表 3.2 所示。当模块进入 RX 模式后，且 Rx\_stby\_irq\_1=00，那么模块收到数据，完成 CRC 校验后会在 IRQ\_0 引脚输出一个高电平。当模块进入 TX 模式后，且 Tx\_irq\_1=1，那么模块发送完数据后，会

在 IRQ\_1 引脚输出一个高电平。所以要通过 IRQ 来确定模块的工作状态，首先要确定模块是处于 Rx、Tx、Standby、Sleep 哪种工作模式

表 3.2 中断状态寄存器

Add	名称	数值	RW	功能描述
Bit7	Rx_stby_irq_0	00	r/w	数据包准备好(默认)
		01		写数据
10		FIFO 缓冲区无数据(待机模式也有效)		
11		同步或地址匹配(如果开启了地址过滤功能)		
Bit6	Rx_stby_irq_1	00	r/w	CRC 校验成功(默认)
Bit5		01		FIFO 缓冲区满载(待机模式也有效)
		Bit4		10
11				FIFO 缓冲区载荷达到阈值(待机模式也有效)
Bit3	Tx_start_irq_0	0	r/w	当 FIFO 载荷超过阈值才开始发送数据包, IRQ_0 映射到 FIFO 阈值
		1		FIFO 缓冲区有数据即开始发送数据, IRQ_0 映射到 FIFOempty(默认)
Bit2	Tx_irq_1	0	r/w	FIFO 缓冲区满载
		1		发送完成(默认)
Bit1	Fifofull	0	r	FIFO 缓冲区满载时置 1
		1		
Bit0	/Fifoempty	0	r	FIFO 缓冲区空载时置 0
		1		

### 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。

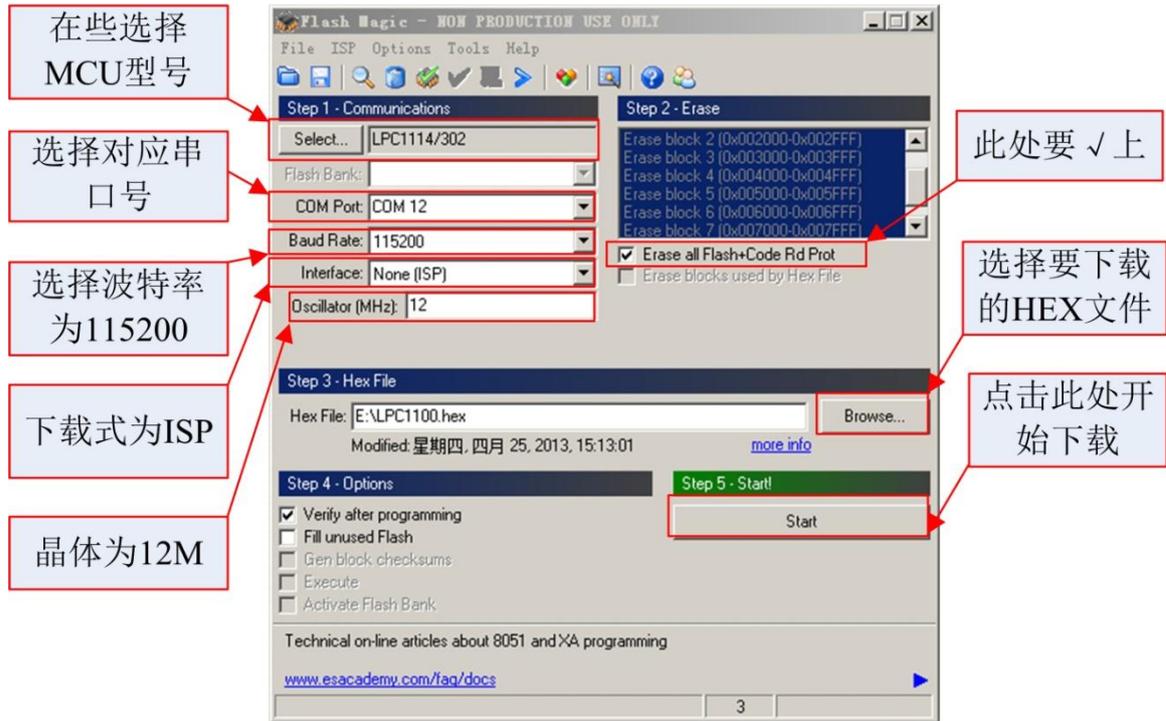


图 3.1 Flash Magic 配置

### 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. ZM433SX-L 驱动简介

ZM433SX-L 的驱动主要包括 SPI 时序控制、寄存器初始化、中断输出判断。

### 4.1.1 模块寄存器配置

ZM433SX-L 模块有 32 个配置寄存器。具体每个寄存器的功能，可以参考 SX1212.pdf 的 58 页。

程序清单 4.1 模块的配置寄存器

```
_U16 RegistersCfg[] = { // SX1212 configuration registers values
    //0 :运行状态, 及频率范围
    DEF_MCPARAM1 | RF_MC1_STANDBY | RF_MC1_BAND_430470 | RF_MC1_SUBBAND_FIRST,
    //1 :包模式, 调制模式, IF 增益
    DEF_MCPARAM2 | RF_MC2_DATA_MODE_PACKET | RF_MC2_MODULATION_FSK |
RF_MC2_OOK_THRESH_TYPE_PEAK | RF_MC2_GAIN_IF_00,
    //2 :频偏
    DEF_FDEV | RF_FDEV_100,
    //3 :空中传输速率
    DEF_BITRATE_MSB | RF_BITRATE_MSB_25000,
    //4 :空中传输速率
    DEF_BITRATE_LSB | RF_BITRATE_LSB_25000,
    //5 :
    DEF_MCPARAM6 | RF_MC6_PARAMP_11 | RF_MC6_LOW_POWER_RX_OFF |
RF_MC6_VCO_TRIM_11 | RF_MC6_RPS_SELECT_1,
    //6 :PLL 分频寄存器 1 433Mhz
    DEF_R1 | 0x8f,
    //7 :PLL 分频寄存器 1 433Mhz
    DEF_P1 | 0x38,
    //8 :PLL 分频寄存器 1 433Mhz
    DEF_S1 | 0x37,
    //9 :PLL 分频寄存器 2
    DEF_R2 | RF_R2_VALUE,
    //10:PLL 分频寄存器 2
    DEF_P2 | RF_P2_VALUE,
    //11:PLL 分频寄存器 2
    DEF_S2 | RF_S2_VALUE,
    //12:FIFO 深度, FIFO 门
    DEF_FIFOCONFIG | RF_FIFO_SIZE_64 | RF_FIFO_THRESH_VALUE,
    //13:IRQ0 和 IRQ1 中断源
    DEF_IRQPARAM1 | RF_IRQ0_RX_STDBY_PAYLOADREADY | RF_IRQ1_RX_STDBY_CRCOK |
RF_IRQ0_TX_FIFOEMPTY_START_FIFONOTEMPTY | RF_IRQ1_TX_TXDONE,
    //14:FIFI Tx_Done RSSI_irq PLL_lock
    DEF_IRQPARAM2 | RF_IRQ2_PLL_LOCK_PIN_ON,
    //15:信号强度门槛
```

```

DEF_RSSIIRQTHRESH | RF_RSSIIRQTHRESH_VALUE,
//16:滤波器 BW
DEF_RXPARAM1 | RF_RX1_PASSIVEFILT_378 | RF_RX1_FC_FOPLUS100,
//17:中心频率滤波器
DEF_RXPARAM2 | RF_RX2_FO_100,
//18:同步码相关的配置
DEF_RXPARAM3 | RF_RX3_POLYPFILT_OFF | RF_RX3_SYNC_SIZE_32 | RF_RX3_SYNC_TOL_0,
//19:OOK 解调相关
DEF_OOKFLOORTHRESH | RF_OOKFLOORTHRESH_VALUE,
//20:RSSI 信号强度，只读寄存器
DEF_RSSIVALUE,
//21:OOK 解调相关
DEF_RXPARAM6 | RF_RX6_OOK_THRESH_DECSTEP_000 |
RF_RX6_OOK_THRESH_DECPERIOD_000 | RF_RX6_OOK_THRESH_AVERAGING_00,
//22:同步码 1
DEF_SYNCBYTE1 | 0x69, // 1st byte of Sync word,
//23:同步码 2
DEF_SYNCBYTE2 | 0x81, // 2nd byte of Sync word,
//24:同步码 3
DEF_SYNCBYTE3 | 0x7E, // 3rd byte of Sync word,
//25:同步码 4
DEF_SYNCBYTE4 | 0x96, // 4th byte of Sync word,
//26:发射滤波带宽，功率
DEF_TXPARAM | RF_TX_FC_200 | RF_TX_POWER_MAX | RF_TX_ZERO_IF_OFF,
//27:CLK 输出
DEF_OSCPARAM | RF_OSC_CLKOUT_ON | RF_OSC_CLKOUT_427,
//28:包长度，与包格式相关
DEF_PKTPARAM1 | RF_PKT1_MANCHESTER_OFF | 64,
//29:地址过滤
DEF_NODEADRS | RF_NODEADRS_VALUE,
//30:包格式，引导码长度，数据美化，CRC，地址过滤禁使能
DEF_PKTPARAM3 | RF_PKT3_FORMAT_VARIABLE | RF_PKT3_PREAMBLE_SIZE_32 |
RF_PKT3_WHITENING_OFF | RF_PKT3_CRC_ON | RF_PKT3_ADRSFILT_00,
//31:在 standby 模式时，选择 FIFO 可读还是可写
DEF_PKTPARAM4 | RF_PKT4_AUTOCLEAR_ON | RF_PKT4_FIFO_STANDBY_WRITE
};

```

#### 4.1.1 初始化模块

初始化 SX1212 芯片，调用 InitRFChip()函数，该函数主要是把上述的 32 个寄存器的值写入到芯片，但是第 20 个寄存器 RSSI，是只读的，所以中间会跳过这个寄存器，总共写入 31 个寄存器值。

#### 4.1.2 发送数据

当需要发送数据时调用 SendRfFrame(\_U8 \*buffer, \_U8 size, \_U8 \*pReturnCode)函数，函

数的第一个参数是发送缓冲区的指针，第二个参数是数据长度，第三个参数反馈函数的处理结果，模块成功发送数据，该参数返回“OK”，否则返回“ERROR”。这个函数里有发送等待，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，首先配置 FIFO 为可写模式，然后把 size 写进 FIFO 的第一位，可以让接收端知道这包数据有多少位，接着往后继续写入要发送的数据，配置寄存器使模块进入发射状态并等待模块发送完成输出 IRO\_1 的中断信号，最后把模块设置为 Standby 模式。

#### 4.1.3 接收函数

当需要发送数据时调用 ReceiveRfFrame (\_U8 \*buffer, \_U8 \*size, \_U8 \*pReturnCode)函数，函数的第一个入参是接收缓冲区的指针，第二个参数反馈接收到的字节数，第三个参数反馈函数的处理结果，模块成功发送数据，该参数返回“OK”，否则返回“ERROR”。函数流程大体是这样执行的，首先设置模块为 Standby 模式并配置 FIFO 为可读状态，然后从 FIFO 读出第一个 byte，根据这个 byte 的大小，再读取随后的数据并把这些数据保存在 buffer 指向的缓冲区。

#### 4.1.4 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 ZM433SX-L 模块并使它进入接收状态，接着点亮 LED，最后进入循环体。循环里包含三个部分，分别是接收数据包、发送数据包、调整发送功率。串口没有数据时，循环体一直在检测 if (IRQ1\_READ())。当接收到数据时，模块 IRQ\_1 输出高电平，循环进入 ReceiveRfFrame()函数，将数据从 FIFO 中读取出来并发送到串口。串口接收到 PC 传来的数据时，循环体不会立即进入发射函数，而是等待串口数据超过(PACK\_LENGTH-4)=60 或者等待串口停止发送数据。当这两个条件中的一个满足，则进入 SendRfFrame()函数，发送一帧数据，发送完成后切换回接收模式。

程序清单 4.2 主函数流程

```
int main(void)
{
    uint8_t    ReturnFlag          = 0;
    uint8_t    j                   = 0;
    uint16_t   i                   = 0;
    uint16_t   temp                = 0x5f;
    uint8_t    tempbuf[PACK_LENGTH] = {0};
    /* 系统初始化; 晶体 11.0592MHz */
    SystemInit();
    LPC_SYSCON->SYSAHBCLKCTRL |= (1 << 6);
    /* 初始化 LED */
    LED_INIT();
    /* 初始化 UART */
    uartInit();
    /* 初始按钮 S3 中断;对应功率 */
    KEY3_INT();
    /* 初始按钮 S2 中断 */
}
```

```
KEY2_INT());
/* 初始 SX1212 */
InitRFChip();
/* 设置载波频率 */
Freq_Set(433000);
SetRFMode(RF_RECEIVER);
LED_Show(0x18);
while (1) {
    /* 接收 */
    /* 如果接收到数据 DIO0 引脚引高 */
    if (IRQ1_READ()) {
        ReceiveRfFrame(&ucRxbuf[0], &j, &ReturnFlag);
        SetRFMode(RF_RECEIVER);
        if(j > 0) {
            /* 把数据从串口发送出去 */
            for (temp = 0; temp < j; temp++) {
                uartSendByte(ucRxbuf[temp]);
            }
            /* 收到数据计数加 1 */
            led++;
            led = led % 128;
            LED_Show(led);
        }
    }
    /* 发送 */
    /* 若缓冲区满或者接收超时 */
    if(uiByteNum >=(PACK_LENGTH-4) || uiMs == 1) {
        /* 失能中断,防止 SPI 时序混乱 */
        DisableIRQ();
        /* 失能 UART 中断 */
        NVIC_DisableIRQ(UART_IRQn);
        for (i = 0; i < uiByteNum ; i++) {
            /* 把串口缓冲区的数据读出来 */
            tempbuf[i] = ucTxbuf[i];
        }
        /* 缓冲区数据标志清零 */
        uiByteNum = 0;
        /* 使能 UART 中断 */
        NVIC_EnableIRQ(UART_IRQn);
        LED_Show(0x80);
        /* 发送一帧数据 */
        SendRfFrame(&tempbuf[0], i, &ReturnFlag);
        SetRFMode(RF_RECEIVER);
        /* 恢复按键中断 */
    }
}
```

```
EnableIRQ();
LED_Show(led);
uiMs = 0;
//ucsleep = 1;
}
/* 休眠 */
if (ucsleep == 1) {
    SetRFMode(RF_SLEEP);
    ucsleep = 0;
}
if (uiByteNum > 0) {
    delayxms(1);
    if(uiMs > 0) {
        /* 进入串口中断，uiMs=3，延时 3ms 还没有收到串口数据则把缓存发送出去 */
        uiMs--;
    }
}
}
```

## 5. 解决故障

1. 通过 ZM433SX-L 的电流可以判断其工作状态，处在准备状态时电流在 1.3mA，处在接收状态时电流在 3mA 左右。

## 6. 免责声明

### *应用信息*

本应用信息适用于 ZM433SX-L-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股份有限公司所有，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。广州致远电子股份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。