

# ZM433S-M-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.....	3
2.1 安装驱动.....	3
2.2 驱动例程简介.....	3
2.3 通信.....	3
2.3.1 按键发送数据.....	3
2.3.2 串口收发数据.....	3
2.4 按键调节发射功率.....	4
3. ZM433SX 模块硬件指南 .....	5
3.1 硬件电路.....	5
3.2 寄存器说明.....	5
3.3 ISP 下载程序.....	5
3.4 调试接口.....	6
4. ZM433S-M 驱动简介 .....	7
4.1 模块寄存器配置.....	7
4.2 发送数据.....	8
4.3 接收函数.....	8
4.4 速率函数.....	8
4.5 功率函数.....	8
4.6 掉电函数.....	8
4.7 主函数流程.....	8
5. 解决故障.....	12
6. 免责声明.....	13

## 1. 产品简介

ZM433S-M 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。ZM433S-M-DEMO 是为帮助用户快速上手 ZM433S-M 模块而开发的评估套件，评估板也适用于 ZM470S-M 模块，但是要修改主控 IC 的频率函数。评估板实物如图 1.1 所示。

**ZM433S-M-DEMO** 这个型号包含了两个板子，一个发送，一个接受。

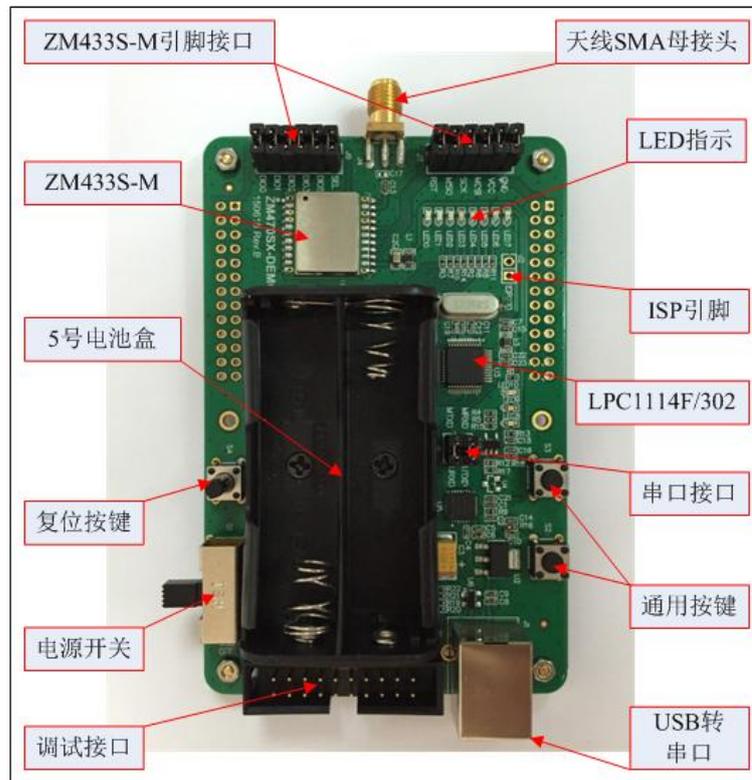


图 1.1 评估板实物图

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， 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 433MHz，阻抗 50Ω 的
ZM433S-M	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-ZM433S-M 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! 未找到引用源。**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。

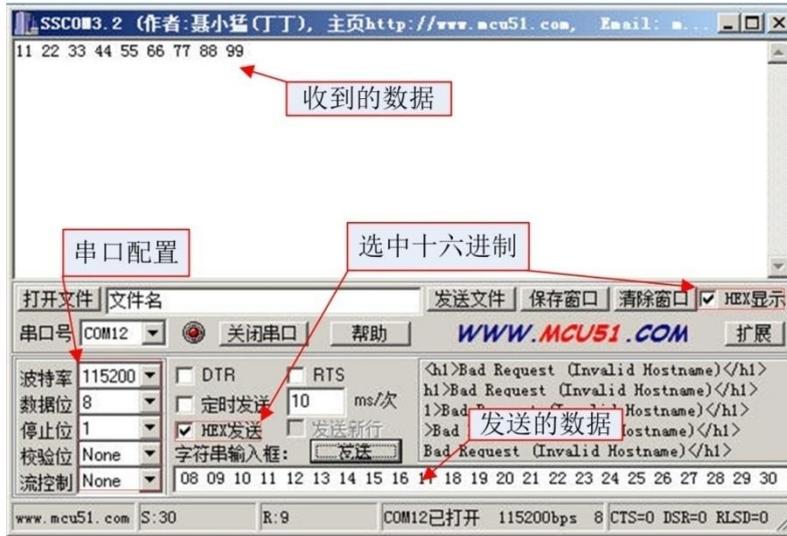


图 2.1 串口 1



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 5dB，如果超出最大值则回到最小值，调整范围：20dBm~-18dBm。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	单位 dBm	20	15	10	5	0	-5	-10	-18
功率	单位 mW	100	31	10	3.1	1	0.31	0.1	0.015

### 3. ZM433SX 模块硬件指南

#### 3.1 硬件电路

ZM433S-M 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 ZM433S-M 模块。

表 3.1 ZM433S-M-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。
SDN	芯片内部电源使能引脚，低电平使能（芯片供电）；如果为高电平，模块电源掉电，功耗达到最小。
SEL	SPI 片选引脚，低电平有效。
IO3	可编程决定。
IO2	可编程决定。
IO1	可编程决定。
IO0	可编程决定。
IRQ	中断输出，功能由编程决定。

原理图设计可参考《ZM433S-M-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 寄存器说明

SI4438 有上百个寄存器，因此不建议用户自己修改寄存器，比较可行的方法是在软件上设置相关参数后生成配置头文件来静态修改寄存器值。

#### 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。

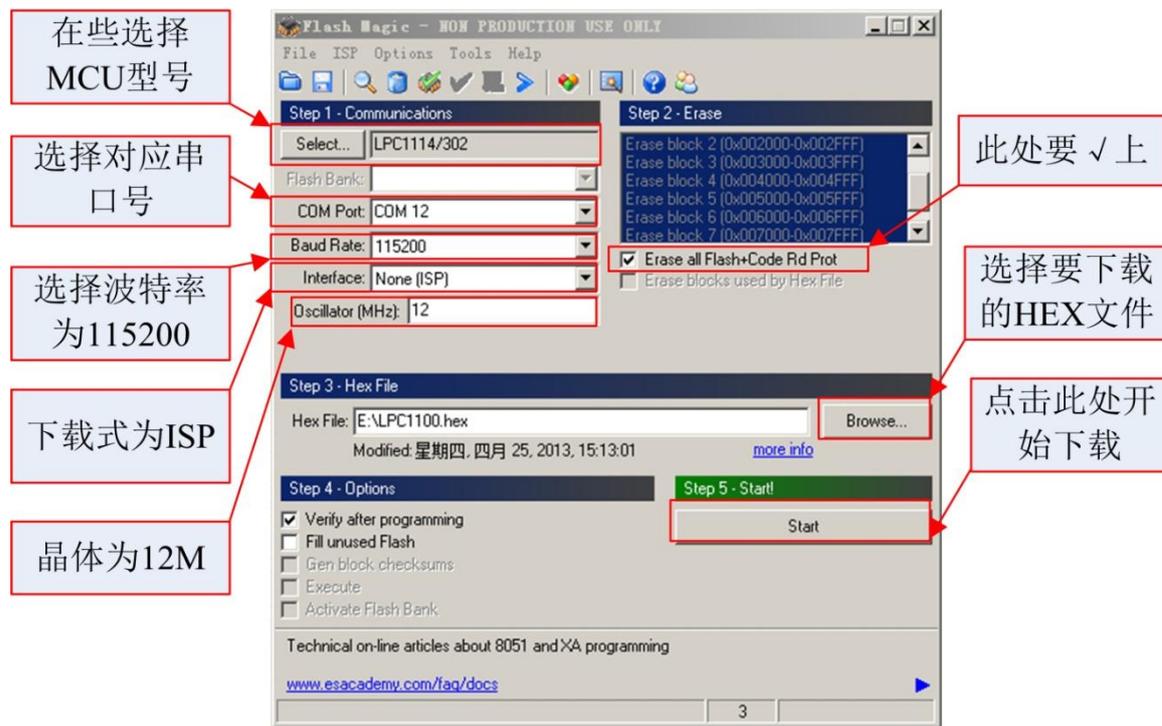


图 3.1 Flash Magic 配置

### 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 3.2 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 3.2 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. ZM433S-M 驱动简介

ZM433S-M 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 模块寄存器配置

SI4438 上百个配置寄存器。用户自己配置具体的寄存器不太现实，一般都是通过软件选择参数然后生成配置代码的头文件。主要的参数有中心频率、空中速率、发射功率。

程序清单 4.1 初始化 SI4438 的配置指令

```
#define RADIO_CONFIGURATION_DATA_ARRAY { \  
    0x07, RF_POWER_UP, \  
    0x08, RF_GPIO_PIN_CFG, \  
    0x06, RF_GLOBAL_XO_TUNE_2, \  
    0x05, RF_GLOBAL_CONFIG_1, \  
    0x06, RF_INT_CTL_ENABLE_2, \  
    0x08, RF_FRR_CTL_A_MODE_4, \  
    0x0D, RF_PREAMBLE_TX_LENGTH_9, \  
    0x0A, RF_SYNC_CONFIG_6, \  
    0x0B, RF_PKT_CRC_CONFIG_7, \  
    0x10, RF_PKT_LEN_12, \  
    0x10, RF_PKT_FIELD_2_CRC_CONFIG_12, \  
    0x10, RF_PKT_FIELD_5_CRC_CONFIG_12, \  
    0x0D, RF_PKT_RX_FIELD_3_CRC_CONFIG_9, \  
    0x10, RF_MODEM_MOD_TYPE_12, \  
    0x05, RF_MODEM_FREQ_DEV_0_1, \  
    0x10, RF_MODEM_TX_RAMP_DELAY_12, \  
    0x10, RF_MODEM_BCR_NCO_OFFSET_2_12, \  
    0x07, RF_MODEM_AFC_LIMITER_1_3, \  
    0x05, RF_MODEM_AGC_CONTROL_1, \  
    0x07, RF_MODEM_AGC_WINDOW_SIZE_3, \  
    0x08, RF_MODEM_OOK_PDTC_4, \  
    0x0C, RF_MODEM_RAW_CONTROL_8, \  
    0x07, RF_MODEM_RSSI_CONTROL_3, \  
    0x06, RF_MODEM_RAW_SEARCH2_2, \  
    0x06, RF_MODEM_SPIKE_DET_2, \  
    0x05, RF_MODEM_RSSI_MUTE_1, \  
    0x09, RF_MODEM_DSA_CTRL1_5, \  
    0x10, RF_MODEM_CHFLT_RX1_CHFLT_COE13_7_0_12, \  
    0x10, RF_MODEM_CHFLT_RX1_CHFLT_COE1_7_0_12, \  
    0x10, RF_MODEM_CHFLT_RX2_CHFLT_COE7_7_0_12, \  
    0x08, RF_PA_MODE_4, \  
    0x0B, RF_SYNTH_PFDPCP_CPFF_7, \  
    0x10, RF_MATCH_VALUE_1_12, \  
    0x0C, RF_FREQ_CONTROL_INTE_8, \  
}
```

```
    0x00 \  
}
```

初始化 SI4438 芯片，调用 `vRadio_Init()` 函数，函数会把以上所有配置属性写入到芯片中。

## 4.2 发送数据

当需要发送数据时调用 `vRadio_StartTx_Variable_Packet(U8 channel, U8 *pioRadioPacket, U8 length)` 函数，`channel` 是通道号（决定通信频率，中心频率 =  $420 + \text{channel} * 0.5$  MHz），对于 433MHz，`channel=26`。`pioRadioPacket` 是指向数据缓冲区的指针。`length` 是数据包的总长度，如果需要发送 25 个字节的数据包，`length = 26`，`pioRadioPacket` 的长度也是 26，其中第一个字节的数值是 25，用于指示后面有一个长度为 25 字节的数据包。

## 4.3 接收函数

`bRadio_Check_Tx_RX()` 是一个检测函数。函数首先判断 `IRQ` 引脚的电平，若是高电平则返回 0，若是低电平则发送 `SPI` 指令获取中断寄存器状态值，然后判断中断源是命令错误、发送完成、接收完成、`CRC` 错误中的哪一种。若检测到接收完成，则再次发送 `SPI` 指令，读取 `FIFO` 的数据，把数据保存在 `customRadioPacket` 这个全局变量中，也把 `RSSI` 值保存在 `rsssi` 中。

## 4.4 速率函数

`vRadio_Set_Rate(U8 rate)` 是一个设置空中速率的函数，提供了 8 种不同的速率。`rate` 的取值从 0-7，对应的速率，0→600、1→1200、2→2400、3→4800、4→9600、5→40k、6→100k、7→200k。两个模块要能正常通信，空中速率必须要一致。`vRadio_Init()` 初始化函数默认的空中速率是 9600。

## 4.5 功率函数

`U8 vRadio_Set_PA_Level(U8 level)` 是功率设置函数，提供了 8 种不同的输出功率设置。`Level` 的取值范围 0-7。对应的输出功率，0→20dBm、1→15dBm、2→10dBm、3→5dBm、4→0 dBm、5→-5 dBm、6→-10 dBm、7→-18 dBm。

## 4.6 掉电函数

`vRadio_ShutDown()` 函数控制 `SDN` 引脚输出高电平，使模块进入掉电模式。要使模块恢复正常工作，要调用 `vRadio_Init()` 函数。

## 4.7 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 `ZM433S-M` 模块并使它进入接收状态，接着点亮 `LED`，最后进入主循环体。主循环里包含两个部分，分别是接收数据包、发送数据包。串口没有数据时，循环体一直在检测

bRadio\_Check\_Tx\_RX() 的返回值。当接收到数据时，模块 IRQ 输出低电平，bRadio\_Check\_Tx\_RX()函数通过 SPI 获取模块的中断状态，判断的状态有 4 个，分别是命令错误、发送完成、接收完成、CRC 错误。不同的状态会返回不同的数值指示不同的中断源。如果中断源是接收完成，函数还会把数据从 FIFO 中读取到 customRadioPacket 这个全局变量中。

检测完 bRadio\_Check\_Tx\_RX()函数后，如果串口缓冲区有数据，则进入发送部分，把串口数据转移至发送缓冲区 customRadioPacket[1]地址之后，并且把缓冲区的第一个字节 customRadioPacket[0]设置为发送的数据包的长度值(uiByteNum)，以指示数据包的长度。最后调用 vRadio\_StartTx\_Variable\_Packet()函数，发送数据。

程序清单 4.6 主函数流程

```
int main(void)
{
    //中心频率 = 420 + channel * 0.5 (MHz)
    uint8_t j = 0,channel=26;
    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();
    MCU_PinInit();
    vRadio_Init();
    vRadio_StartRX(channel,0u);
    LED_Show(0x18);
    while (1) {
        /* 接收 */
        switch (bRadio_Check_Tx_RX())
        {
            /* 发送成功 */
            case SI446X_CMD_GET_INT_STATUS_REP_PH_PEND_PACKET_SENT_PEND_BIT:
                /* 从发射模式转换到接收模式需要重新配置 RF_PKT_LEN_12 的寄存器 */
                vRadio_Tx_TO_Rx();
                /* 进入接收状态 */
                vRadio_StartRX(channel, 0u);
                break;
            /* 接收到数据 */

```

```
case SI446X_CMD_GET_INT_STATUS_REP_PH_PEND_PACKET_RX_PEND_BIT:
    /* 每次接收到一个包， rssi 会刷新 */
    //rssi=rssi;
    /* 进入接收状态 */
    vRadio_StartRX(channel, 0);
    if(ucPackLgth > 0) {
        for (temp = 0; temp < ucPackLgth; temp++) {
            /* 把数据从串发送出去 */
            uartSendByte(customRadioPacket[temp]);
        }
        /* 收到数据计数加 1 */
        led++;
        led = led % 128;
        LED_Show(led);
    }
    break;
default:
    break;
}
/* 发送 */
/* 若缓冲区满或者接收超时 */
if(uiByteNum >= PACK_LENGTH || 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 */
        customRadioPacket[i+1] = ucTxbuf[i];
    }
    /* 包的长度放在第一位，总发送的包的长度=uiByteNum+1 */
    customRadioPacket[0]=uiByteNum;
    /* 缓冲区数据标志清零 */
    uiByteNum = 0;
    /* 使能 UART 中断 */
    NVIC_EnableIRQ(UART_IRQn);
    /* LED 指示发送数据中 */
    LED_Show(0x80);
    /* 发送数据， customRadioPacket[0]是数据包长度， +1 则是整个发送的包的长度 */
    vRadio_StartTx_Variable_Packet(channel, &customRadioPacket[0],customRadioPacket[0]+1);
    /* 恢复按键中断 */
    EnableIRQ();
    /* LED 指示数据发送完成 */
    LED_Show(led);
}
```

```
        uiMs = 0;
    }
    if (uiByteNum > 0) {
        /* 误差很小的 1ms 延时          */
        delayxms(1);
        if(uiMs > 0) {
            /* 每次进入串口中断，uiMs 都会刷新为 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 ZM433S-M 的电流可以判断其工作状态,处在接收状态时电流在 14mA 左右,处于发射状态时电流在 75mA 作用。

2. 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,具体操作是减小空中速率。

## 6. 免责声明

### *应用信息*

本应用信息适用于 ZM433S-M-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。