

利用利尔达 eZ430-RF5137 实现 TI 无线实验

利尔达的 eZ430-RF5137-433 是 TI 公司的 EM430F5137RF900 的山寨版。或许为了“避嫌”，在外围电路的设置上，与 TI 的 EM430F5137RF900 稍有不同。其区别参见下表：

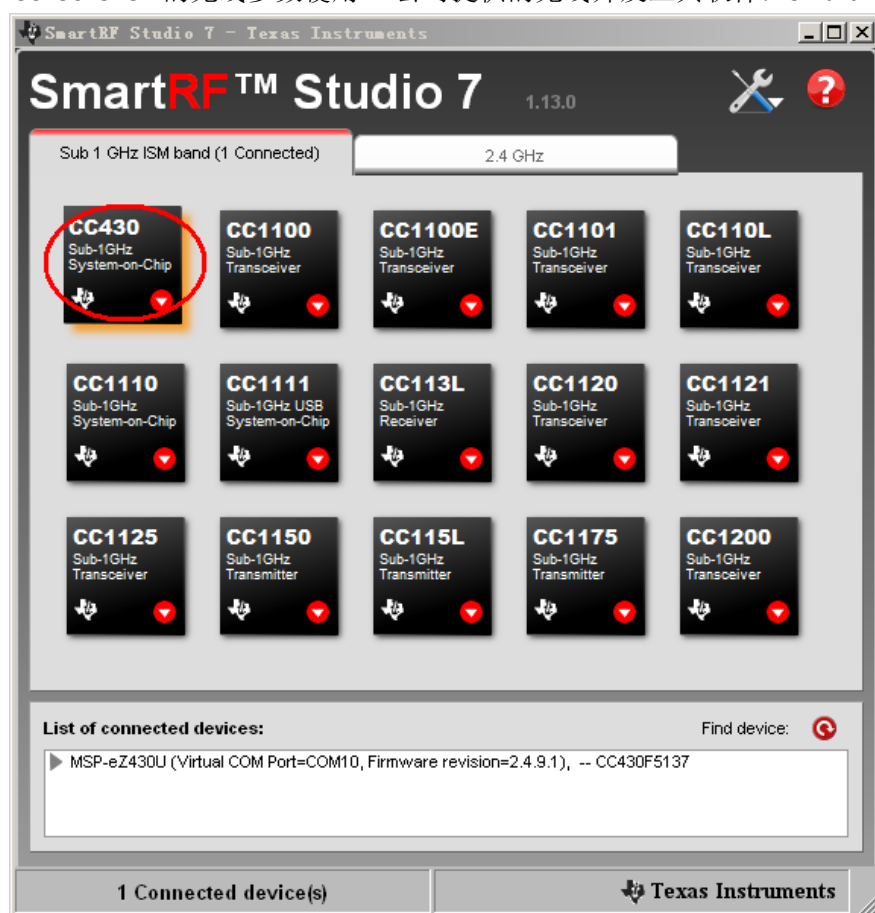
功能	利尔达 eZ430	TI EM430
按钮	P J.0	P 1.1
红色 LED	P 2.6	P 3.6
蓝色 LED	P 2.7	P 2.7

使用 eZ430-RF5137-433 稍作调整，就可以完成 EM430F5137RF900 所有的例程。

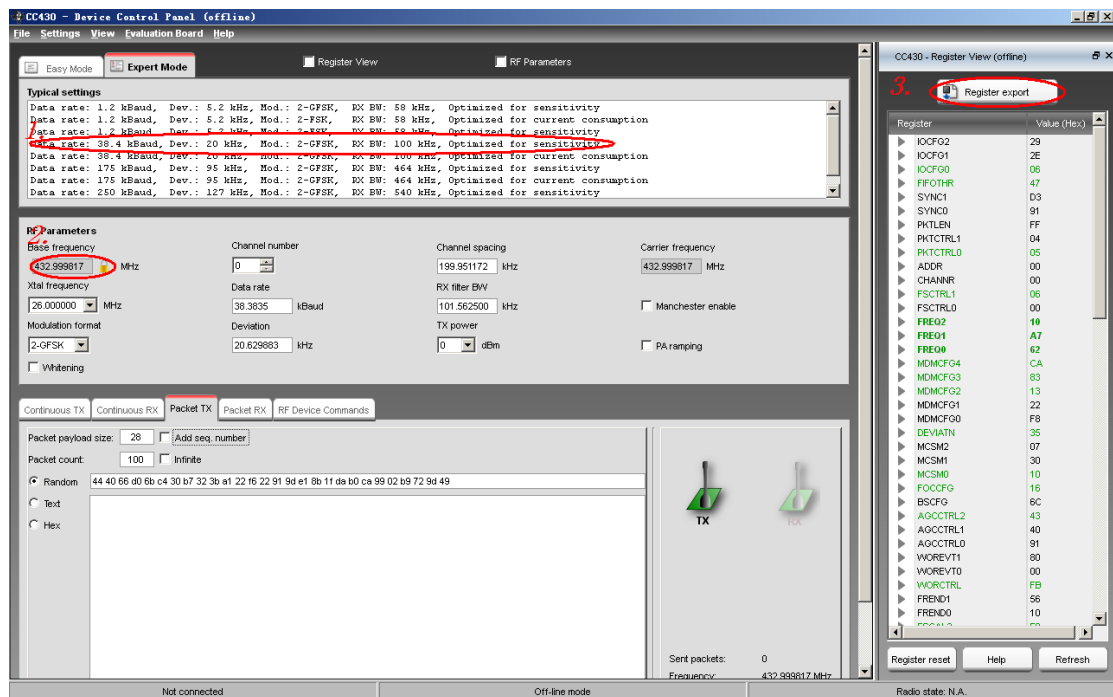
建立正确的无线参数

eZ430-RF5137-433 与 EM430F5137RF900 最大的区别在于无线频率的差别，故此，需要重新建立 CC430F5137 所需要的无线参数。

建立 CC430F5137 的无线参数使用 TI 公司提供的无线开发工具软件：SmartRF Studio 7。



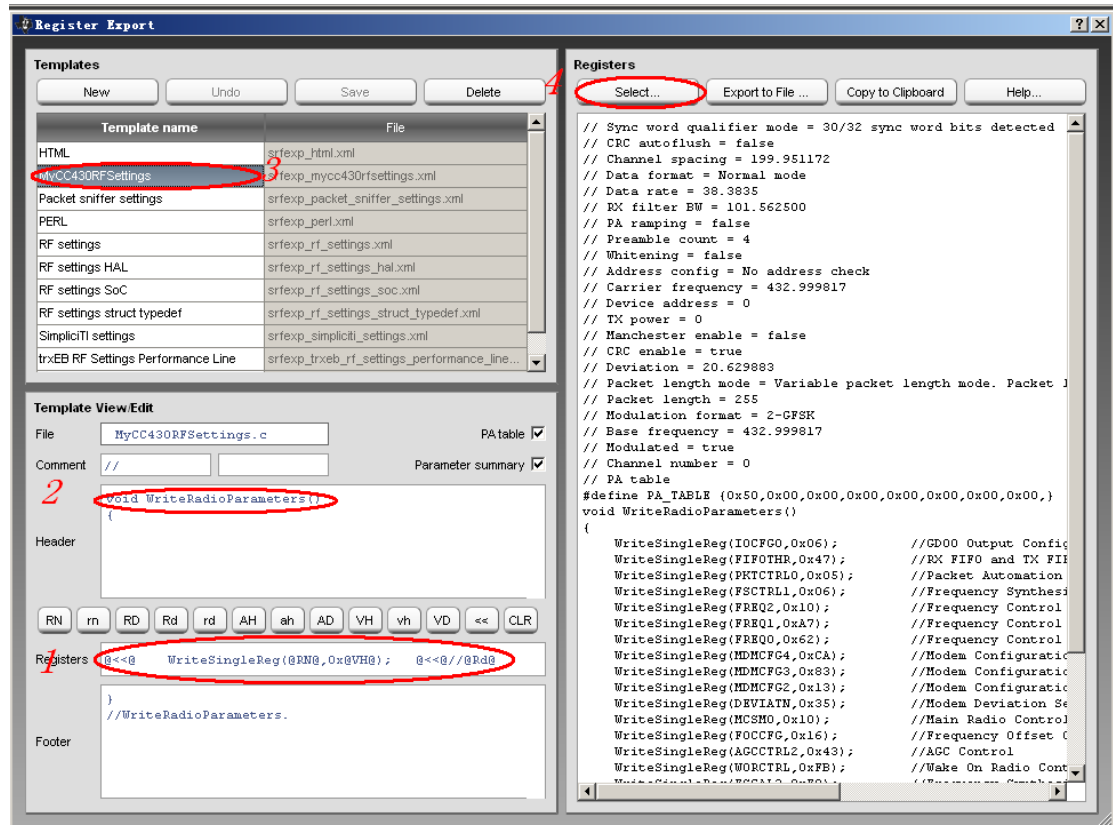
在启动 SmartRF Studio 7 后，选择 CC430→Open Device in Off-line Mode，也可以在连接 eZ430-RF5137-433 时，选择 Show Connected Device，进入参数设置页。



设置无线参数的时候，可以简单地选择 SmartRF Studio 7 提供的优化后的参数。

- 1、在系统提供的优化参数中，选择 38.4K Bps 的优化参数；
- 2、将通讯频率调整为 433MHz；
- 3、选择导出寄存器参数。

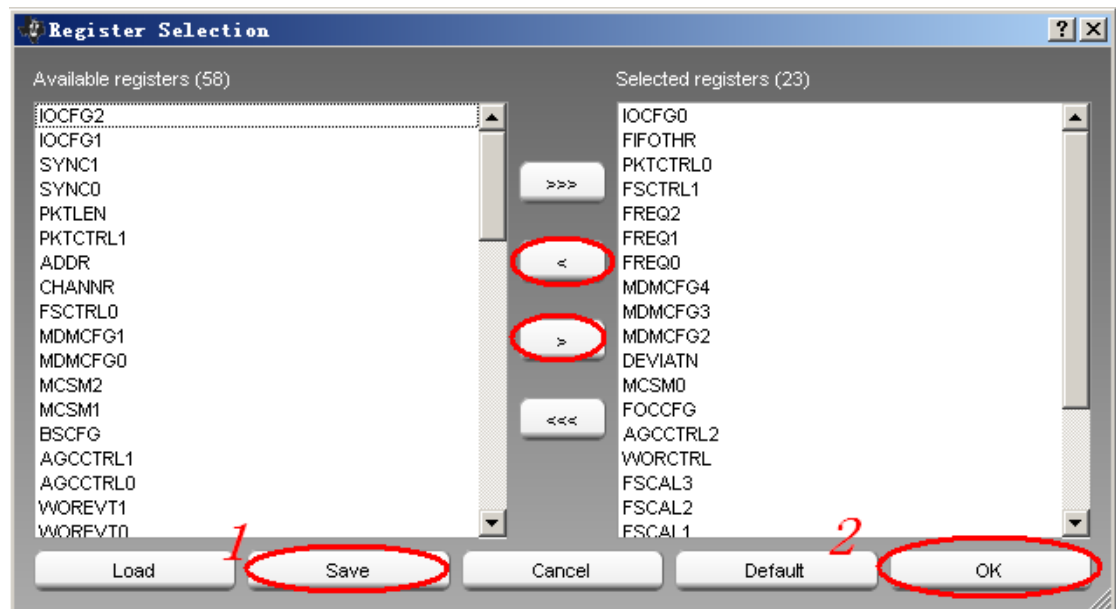
在导出参数的操作中，需要建立一套 CC430 常用的参数的模板，从而方便参数导出。

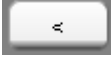
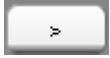


- 1、将无线寄存器设置直接使用 RF1A 中自带的函数 WriteSingleReg () 的方式导出；

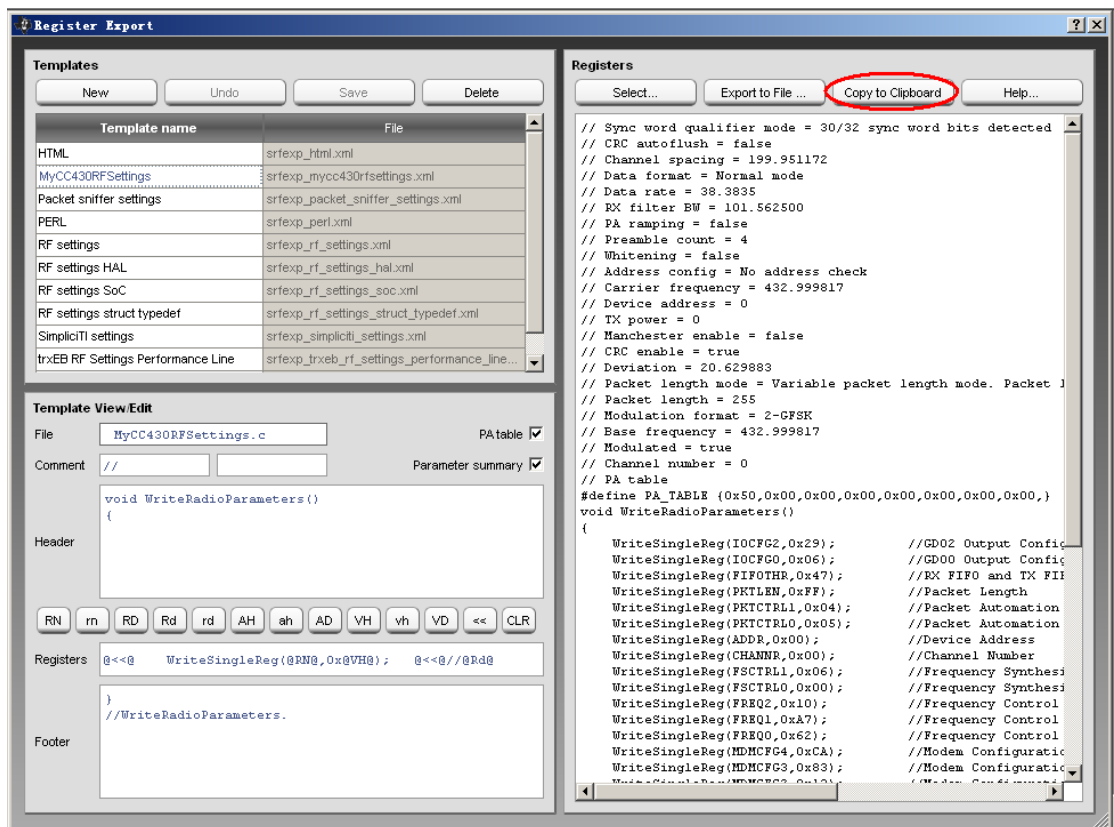
- 2、将函数名设置为 WriteRadioParameters ();
- 3、保存该设置为自己特定的模板;
- 4、选择需要导出的寄存器。

接下来，在寄存器选择列表中，按照例程提供 RF_Settings 的数据结构中所采用的寄存器，来选择自己模板中所需要的寄存器。



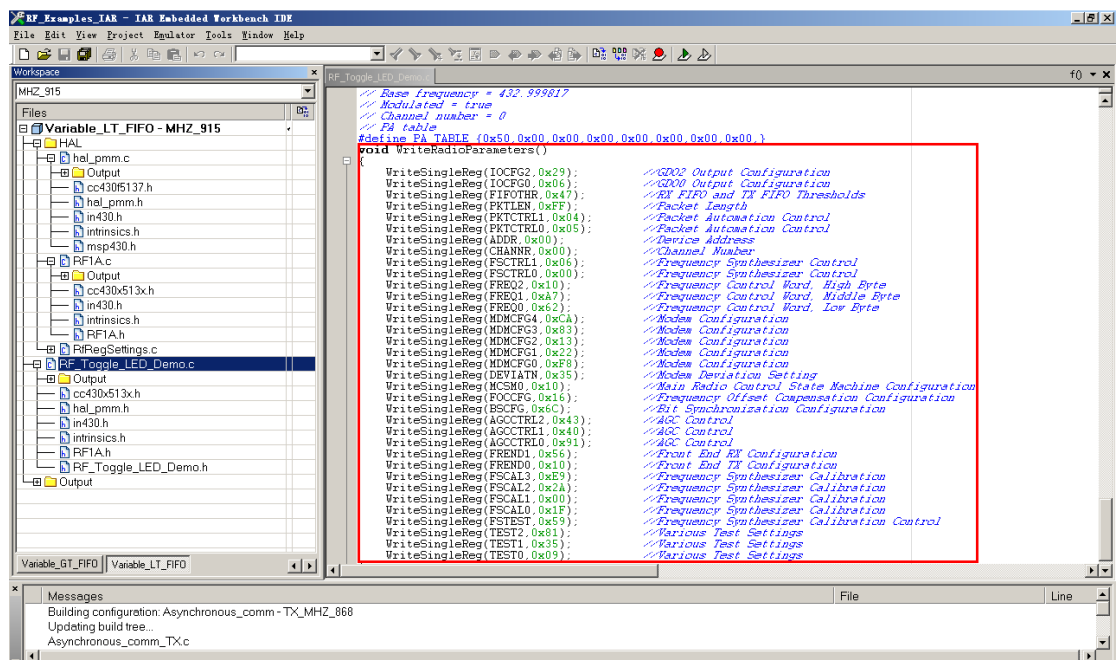
- 1、按照 RF_Settings 数据结构中所需要的寄存器，使用  或者  建立所需要的寄存器清单（注意：这里介绍的方法可以不按照例程中的寄存器次序）。在检查所需要的寄存器都被选择之后，选择 Save 保存；
- 2、点击 OK 退出。
注意，按照这样的方式建立了寄存器列表之后，在接下来所需要的寄存器清单已经保存在系统中，下次选择寄存器的时候，只需要选择 Load，保存了的寄存器清单则被调出。

接下来，可以将 SmartRF Studio 7 生成的无线参数设置函数，Copy 到无线例程中。



调整例程

点击 Copy to Clipboard，进入 IAR，打开无线例程的 WorkSpace，选择 Variable_LT_FIFO，打开 RF_Toggle_LED_Demo.c，将 SmartRF Studio 7 生成的函数 Copy 进文件中。



调整无线初始化函数，注释原来的无线参数函数调用，调用新建立的无线参数设置

函数:

```
void InitRadio(void)
{
    // Set the High-Power Mode Request Enable bit so LPN3 can be entered
    // with active radio enabled
    PMMCTL0_H = 0xA5;
    PMMCTL0_L |= PMMHPMRE_L;
    PMMCTL0_H = 0x00;

    //WriteRfSettings(&rfSettings);

    WriteRadioParameters();

    WriteSinglePATable(PATABLE_VAL);
}
```

在 main () 之前声明新的无线参数:

```
extern RF_SETTINGS rfSettings;
void WriteRadioParameters();
unsigned char packetReceived;
unsigned char packetTransmit;

unsigned char RxBuffer[64];
unsigned char RxBufferLength = 0;
const unsigned char TxBuffer[6] = {PACKET_LEN, 0x27, 'L', 'S', 'D', '.'};
unsigned char buttonPressed = 0;
unsigned int i = 0;
```

接下来, 调整 LED 以及 Button 初始化函数:

```
void InitButtonLeds(void)
{
    // Initialize Port J
    PJOUT = 0x01;
    PJDIR = 0xFF;

    // Set up the button as interruptible
    BUTTON_DIR&=~(1<<BUTTON_BIT); // 按键设置为输入
    BUTTON_REN|=BIT0; // 上拉

    // Set up LEDs
    LED_OUT&=~((1<<LED_RBIT)|(1<<LED_GBIT)); // LED端口输出0
    LED_DIR|=(1<<LED_RBIT)|(1<<LED_GBIT); // LED端口方向设置为输出
}
```

编译、下载、运行, 一切都顺利完成。

其他例程

TI 所提供的其他无线例程, 使用同样的方式建立参数、修改例程, 即可以实现。需要注意的是: 在异步、同步连续数据传输的例程中, 会使用的 P 2.7 和 P 2.6, 而在利尔达 eZ430-RF5137-433 中, 这两个 GPIO 口用于了两个 LED, 特别是 P 2.6 会映射成为无线模块的 GDO0。不过没有太大的影响, 如果需要从 P 2.6 做数据输入, 只需要适当地提高输入的电流, 毕竟还需要驱动一个 LED, 不过也不错, 正好可以做信号输入指示☺。