Application Notes
ZHCA743 – Dec. 2017

CC2640R2F 如何和 Nordic52832 的 2.4G 私有模式互连互通

Daniel Fang — *East CHINA EP FAE*
Yan Zhang — *East CHINA EP FAE*

ABSTRACT

CC2640R2F 是 TI 公司最新出的超低功耗 BLE SOC，支持 BLE 和 2.4G 私有协议。Nordic52832 除了 BLE 以外，同样支持 2.4G 私有通信协议。BLE 因为是蓝牙协议栈的标准规范，互相通信相对简单，但 2.4G 私有通信协议因为各个厂家的各自私有配置不同而造成很难互连互通。在实际产品中，往往面临需要各厂家的私有协议互连互通，或者兼容已有产品。本文分析了 Nordic 52832 和 CC2640R2F 的私有协议格式，并通过修改参数来实现 CC2640R2F 和 Nordic 52832 之间的互连互通，最后实际测试收发通信，CRC 校验和组包测试。通过 CC2640R2F 的灵活性，运用合理的配置方式，满足多种通信方式的互连互通。

Contents

1 Nordic52832 私有协议数据包格式 3
2 CC2640R2F 私有协议数据包格式 3
3 难点分析 4
4 CC2640R2F 和 Nordic52832 私有协议互连互通 4
4.1 基本射频参数配置 4
4.2 数据包格式对应 5
4.3 CRC 校验配置 6
5 关键代码修改实现 6
5.1 RF 的配置 6
5.2 CRC 校验的配置 8
5.3 TX 的 packet 配置 9
5.4 TX 的实施 10
5.5 RX 的 packet 解析配置 11
5.6 RX 的实施 13
6 测试结果 14
7 参考文献 15

Figures

图 1. Nordic 52832 私有协议数据包格式 3

图 2. CC2640R2F 私有协议标准数据包格式 3
图 3. Nordic52832 射频参数配置 5
图 4. 数据包格式对应 5
图 5. CMD_RADIO_SETUP 参数配置 7
图 6. CMD_FS 参数配置 8
图 7. CC2640R2F Advanced 数据包格式 9
图 8. CMD_PROP_TX_ADV 参数配置 10
图 9. CMD_PROP_RX_ADV 参数配置 13
图 10. CC2640R2F 上电初始化界面 14
图 11. CC2640R2F 发送界面 15
图 12. CC2640R2F 和 Nordic52832 收发界面 15

1 Nordic52832 私有协议数据包格式

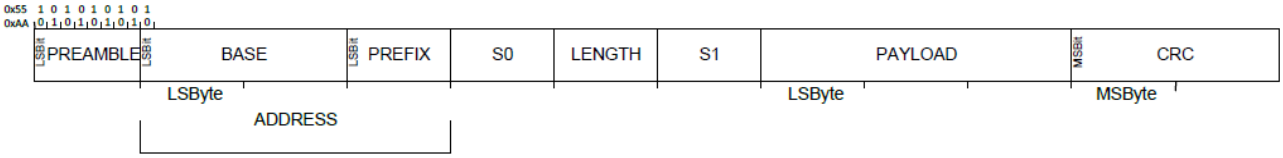

图 1 Nordic 52832 私有协议数据包格式

图 1 为 Nordic52832 私有协议的数据格式，分别由 Preamble、Base、Prefix、S0、Length、S1、Payload 和 CRC 组成。

Preamble 为前导码，除了在 2 Mbit/s 的 BLE 模式下为 2 个字节，其他所有模式下都是 1 个字节。根据 Address 的第一位来判断。如果为 0 ，Preamble 为 0xAA，否则为 0x55。

Address 地址由 Base 和 Prefix 两部分组成，可通过 PCNF1 寄存器 BALEN 设置长度。

S0，LENGTH 和 S1 可以分别通过 PCNF0 寄存器的 S0LEN ，LFLEN 和 S1LEN 来设置长度。

Payload 为实际用户发送数据。S0、LENGTH、S1 和 PAYLOAD 的总长度不能超过 258 字节。

CRC 由芯片自动计算出校验值，从 Address 开始直到 Payload 结束。

2 CC2640R2F 私有协议数据包格式

1 bit to 32 bytes	8 to 32 bits	0 or 1 byte	0 or 1 byte	0 to 255 bytes	0 or 16 bits (0 to 32 bits)
Preamble	Sync word	Length field	Address	Payload	CRC

图 2 CC2640R2F 私有协议标准数据包格式

CC2640R2F 私有协议有两种。图 2 为标准数据包格式，可满足绝大部分数据包格式要求，另外一种为更灵活的 Advanced 数据包格式。

CC2640R2F 的私有模式标准数据包格式由 Preamble、Sync Word、Length field、Address、Payload 和 CRC 组成。Length field、Address、CRC 为可选，可根据实际应用分别省略。

Preamble 由 0 和 1 间隔组成，1-30 字节，长度可修改，根据第一位为 0 还是 1，可设其为 0x55 或 0xAA。

Sync word 同步字，收发两端保持相同，否则会收不到，建议在调通收发的例程基础上修改为自定义参数，避免都用默认例程的参数而发生碰撞。通常设为 1-4 字节，长度越长越能避免和同概率包的冲突，但是处于发送接收的时间也相应越长，功耗也相应增大。

Length field 为可选部分，1 字节。私有协议长度设置有定长和不定长两种模式，定长模式收发两端用设定好的固定长度通信，不定长模式通过 Length field 来告知接收端此包接下来的 Payload 数据包长度，帮助接收端正确解析数据包。

Address 为可选部分，1 字节，可用于做地址过滤。

Payload: 这里是用户实际想要发送或接收的数据，0-255 字节。

CRC 为可选，2 或 4 字节，发送时从 Sync word 到 Payload 由芯片 CRC 校验自动生成，加在数据包最后发送。接收时对整包数据进行校验，如果有一 bit 变化，都会报 CRC 校验出错，整包数据无效。

3 难点分析

用不同芯片调私有协议互连互通，底层物理层和协议层必须要一样，否则如果有一个地方不同就可能造成无法互相通信，其中最重要的参数包括：基本射频配置参数、数据包格式、包括定长或不定长、白化、CRC 校验、LSB 还是 MSB 等。

这里用 SimpleLink CC2640R2F -v1.40.00.45 版本 SDK 下面的 rfPacketErrorRate 例程来调试和 Nordic52832 私有协议 esb 例程通信，主要面临下面三个难点：

1. 两家数据包格式不一致，CC2640R2F 以字节为单位，以一个字节跟一个字节方式，而 Nordic 是字节和比特位混杂方式，每个数据所代表的含义在两个数据包的解析中完全不同。
2. Nordic 使用 1Mbps 和 2Mbps 两种速率，默认为 1Mbps, 而 CC2640R2F 目前 SDK *v1.40.00.45* 下 rfPacketErrorRate 例程支持不了 1Mbps。 用 SmartRF Studio7 v2.6.1 目前还不能直接生成该射频配置文件。
3. Nordic52832 和 CC2640R2F 的 CRC 校验方式，两者实际空中数据 CRC 值相同。

4 CC2640R2F 和 Nordic52832 私有协议互连互通

4.1 基本射频参数配置

基本射频参数包含下面几个关键参数: 通讯频点(frequency)、调制方式(modulation)、频偏(deviation)、通讯符号速率(symbol rate)、接收机滤波器带宽(rx filter bandwidth)。

这些基本射频参数必须要保证收发两端设置一致才可以正常通信。Nordic52832 基本射频参数配置如下：

射频参数	配置
Frequency(MHz)	24xx
Modulation	GFSK
Deviation(KHz)	250
Symbol rate(Kbps)	1000

图 3 Nordic52832 射频参数配置

在 CC2640R2F 端，我们修改射频配置参数以满足 Nordic 默认设置。

4.2 数据包格式对应

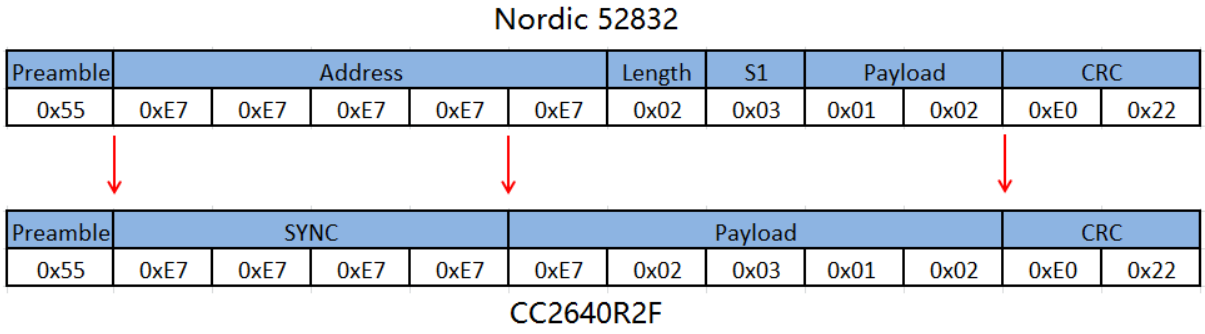

图 4 数据包格式对应

我们以实际要发送 2 个字节的 0x01、0x02 数据为例，把 Nordic52832 的数据包格式用 CC2640R2F 的标准私有协议数据包格式来一一对应。

把 Nordic 数据包内容全部改为字节为单位或者省略，比如在这里 Nordic S1 默认是 3 比特，为了和 CC2640R2F 字节相对应，我们把其长度改为 1 个字节。

CC2640R2F 的同步字 SYNC 对应 4 个字节的 0xE7，这里用的默认数值，但在实际应用中建议修改为自定义数值，避免冲突，同时也降低错包率。

Payload 部分以实际要发送的 0x01、0x02 为例，在 CC2640F2F 实际要发送的 0x01、0x02 数据之前人为添加固定的 0xE7、0x02、0x03 作前缀来和 Nordic 数据包对齐。CC2640R2F 在发送的时候添加固定数据来做前缀，接收的时对该部分丢弃来实现格式对齐。

把 Nordic 的字节和比特相混合的数据方式改为全部以字节为单位，以及通过应用层灵活解析来实现两种不同数据包格式的一一对应。

4.3 CRC 校验配置

Nordic52832 设 CRC 为 2 个字节, NRF_ESB_CRC_16BIT，初始值 CRCINIT = 0xFFFFUL，多项式为 CRCPOLY = 0x11021UL，CRC poly: x^16+x^12^x^5+1。读 RXCRC 寄存器可以知道上一包收到的数据包 CRC 值，不管是否通过 CRC 校验。Nordic 的私有协议空中报文比较特殊，因为默认并不是按字节对齐，例如发送的数据为：0xE7、0xE7、0xE7、0xE7、0xE7、0x02、0x03、0x01、0x02, 其中地址域(5Bytes)：0xE7、0xE7、0xE7、0xE7、0xE7, 长度域(6bits)：0b000010, S1 域(3bits)：0b011，Payload(2Bytes)：0x01、0x02。

S1 是作为 3bits 还是一个字节来计算，CRC 校验值不同，导致实际空中发送的数据包格式也完全不同。因此，我们把长度域和 S1 都改为一个字节，通过整字节的方式来计算 CRC，计算出 CRC 值为 0xE0、0x22，此时实际空中发送的数据为 0x55、0xE7、0xE7、0xE7、0xE7、0xE7、0x02、0x03、0x01、0x02、0xE0、0x22，才能和 CC2640R2F 实际空中数据相对应。

5 关键代码修改实现

5.1 RF 的配置

由于采用的是 1Mbps 速率的模式，而 BLE 的正常模式的速率正好是 1Mbps，所以这里用 CC2640R2F 的 BLE 模式为基础，进行一些私有模式的改造，配置成特殊的 2.4G 私有模式。

在 smartrf_settings_ble.c 中：

```
// TI-RTOS RF Mode Object
RF_Mode RF_modeBle =
{
    .rfMode = RF_MODE_PROPRIETARY_2_4,
    .cpePatchFxn = NULL,
    .mcePatchFxn = NULL,
    .rfePatchFxn = NULL,
};
```

.rfMode，配置成 2.4G 私有模式，而非 BLE 模式：RF_MODE_PROPRIETARY_2_4。

```
// CMD_RADIO_SETUP
rfc_CMD_RADIO_SETUP_t RF_ble_cmdRadioSetup =
{
    .commandNo = CMD_RADIO_SETUP,
    .status = 0x0000,
    .pNextOp = 0, // INSERT APPLICABLE POINTER: (uint8_t*)&xxx
    .startTime = 0x00000000,
    .startTrigger.triggerType = 0x0,
    .startTrigger.bEnaCmd = 0x0,
    .startTrigger.triggerNo = 0x0,
    .startTrigger.pastTrig = 0x0,
    .condition.rule = 0x1,
    .condition.nSkip = 0x0,
    .mode = 0x00,
    .__dummy0 = 0x00,
    .config.frontEndMode = 0x0,
    .config.biasMode = 0x0,
    .config.analogCfgMode = 0x0,
```

```
    .config.bNoFsPowerUp = 0x0,
    .txPower = 0x3161, //0x3161 0 dB
    .pRegOverride = pBleOverrides,
};
```

关键参数：.mode = 0x00，（.mode 前面的参数是 TI 特有 RF command chain 配置，在此可以忽略）表示选择的是 BLE 模式的射频。对应的参数对照表可以在 TRM 中找到：

Table 23-15. CMD_RADIO_SETUP Command Format

Byte Index	Field	Bit Index	Bit Field name	Type	Description
14	mode			W	This is the main mode to use. 0x00: Bluetooth low energy 0x01: IEEE 802.15.4 0x02: 2-Mbps GFSK 0x05: 5-Mbps coded 8-FSK 0xFF: Keep existing mode; update overrides only.
15	loDivider			W	CC13x0: LO divider setting to use. Supported values: 0 (equivalent to 2), 2, 5, 6, 10, 12, 15, and 30. Value of 0 or 2 only supported for CC1350. CC26x0: Reserved
16–17	config	0–2	frontEndMode		0x00: Differential mode 0x01: Single-ended mode RFP 0x02: Single-ended mode RFN 0x05: Single-ended mode RFP with external front-end control on RF pins (RFN and RXTX) 0x06: Single-ended mode RFN with external front-end control on RF pins (RFP and RXTX) Others: Reserved
		3	biasMode	W	0: Internal bias 1: External bias
		4-9	analogCfgMode	W	0x00: Write analog configuration. Required first time after boot and when changing frequency band or front-end configuration 0x2D: Keep analog configuration. May be used after standby or when changing mode with the same frequency band and front-end configuration. Others: Reserved
		10	bNoFsPowerup	W	0: Power up frequency synthesizer. 1: Do not power up frequency synthesizer.
		11–15			Reserved
18–19	txPower			W	Output power setting; use value from SmartRF Studio. For more details, see Section 23.3.4.16.
20–23	pRegOverride			W	Pointer to a list of hardware and configuration registers to override. If NULL, no override is used.

图 5 CMD_RADIO_SETUP 参数配置

频率合成器的配置，这里主要配置发射&接收的频率/频段：

```
// CMD_FS
rfc_CMD_FS_t RF_ble_cmdFs =
{
    .commandNo = CMD_FS,
    .status = 0x0000,
    .pNextOp = 0, // INSERT APPLICABLE POINTER: (uint8_t*)&xxx
    .startTime = 0x00000000,
    .startTrigger.triggerType = 0x0,
    .startTrigger.bEnaCmd = 0x0,
    .startTrigger.triggerNo = 0x0,
    .startTrigger.pastTrig = 0x0,
    .condition.rule = 0x1,
    .condition.nSkip = 0x0,
    .frequency = 0x0978,
    .fractFreq = 0x0000,
    .synthConf.bTxMode = 0x0,
    .synthConf.refFreq = 0x0,
```

```
    .__dummy0 = 0x00,
    .__dummy1 = 0x00,
    .__dummy2 = 0x00,
    .__dummy3 = 0x0000,
};
```

主要参数就是.frequency 和.fractFreq。该命令的对应参数对照表可以在 TRM 中找到：

Table 23-24. CMD_FS Command Format

Byte Index	Field Name	Bit Index	Bit Field Name	Type	Description
14–15	frequency			W	The frequency in MHz to which the synthesizer should be tuned
16–17	fractFreq			W	Fractional part of the frequency to which the synthesizer should be tuned
18	synthConf	0	bTxMode	W	0: Start synthesizer in RX mode. 1: Start synthesizer in TX mode.
		1–6	refFreq	W	CC13x0: 0: Use default reference frequency. Others: Use reference frequency 24 MHz/refFreq. CC26x0: Reserved
		7	Reserved	W	Reserved
19–23			Reserved	W	Reserved

图 6 CMD_FS 参数配置

其中，实际使用的频率的换算方法为 (frequency + fractFreq / 65536) MHz，即（0x0978 + 0x0000 / 65536）= 0x0978MHz，换算成十进制就是 2424MHz。

5.2 CRC 校验的配置

CRC 校验的配置在 pBleOverrides 结构里面实现。

在 smartrf_settings_ble.c 中：

```
// Overrides for CMD_RADIO_SETUP
static uint32_t pBleOverrides[] =
{
 HW_REG_OVERRIDE(0x6084, 0x05F8), // RFC_RFE:SPARE0. Select R1-style gain table
 (uint32_t)0x04280243,  //10 us added to the RF SYNTH calibration
 (uint32_t)0x00FF8A43,  // set advLenMask to 0xFF to avoid ROM patch
#ifdef CACHE_AS_RAM
 (uint32_t)0x00018063,
#endif //CACHE_AS_RAM
// 0x10040103, // Set MSB first, no whitener, 16-bit CRC
 0x10000103, // Set LSB first, no whitener, 16-bit CRC
 0xC0040051, // Set CRC initialization
 0xFFFF0000, // CRC initialization value
 0x40012005, // Set CRC polynomial
 0x10210000, // CRC polynomial
 (uint32_t)0xFFFFFFFF,
};
```

可以看到，CRC 初始值 0xFFFFUL：

```
0xC0040051, // Set CRC initialization
0xFFFF0000, // CRC initialization value
```

多项式为 CRCPOLY = 0x11021UL，CRC poly: x^16+x^12^x^5+1：

```
0x40012005, // Set CRC polynomial
0x10210000, // CRC polynomial
```

5.3 TX 的 packet 配置

Packet 配置主要包括 preamble，syncword，CRC 的一些使用方式、包长、包的内容等，包的内容可以由后续代码运行中动态改变。

由于 CC2640R2F 的 BLE 模式下的 Preamble 和 Syncword 没法满足我们的需要，且无法更改，所以我们不用 BLE 模式下的 CMD_BLE_ADV_NC（这里 ADV=Advertising，NC=None Connectable），转而用私有模式命令 CMD_PROP_TX_ADV（这里 ADV=Advanced）来完成发送 packet 的配置。

在 smartrf_settings_ble.c 中：

```
// CMD_BLE_ADV_NC
rfc_CMD_PROP_TX_ADV_t RF_ble_cmdBleAdvNc  =
{
     .commandNo = 0x3803,
     .status = 0x0000,
     .pNextOp = 0,
     .startTime = 0x00000000,
     .startTrigger.triggerType = 0x0,
     .startTrigger.bEnaCmd = 0x0,
     .startTrigger.triggerNo = 0x0,
     .startTrigger.pastTrig = 0x0,
     .condition.rule = 0x1,
     .condition.nSkip = 0x0,
     .pktConf.bFsOff = 0x0,
     .pktConf.bUseCrc = 0x1,
     .pktConf.bCrcIncSw = 1,
     .pktConf.bCrcIncHdr = 1,
     .numHdrBits = 0,
     .pktLen = PAYLOAD_LENGTH, // SET APPLICATION PAYLOAD LENGTH
     .syncWord = 0xE7E7E7E7,
     .preTrigger.triggerType = TRIG_NOW,
     .pPkt = 0,
};
```

主要的参数是.pktConf.bUseCrc，.pktConf.bCrcIncSw，.pktLen，.syncWord，.pPkt 等等。其中，.pktLen 暂且设置为 5，.syncWord 则根据要求，设置成 0xE7E7E7E7，.pPkt 暂且设置成 0，具体内容在代码运行时修改。

CC2640R2F 的 Advanced 数据包格式如下：

Figure 23-10. Advanced Packet Format

1 bit to 32 bytes or repetition	8 to 32 bits	0 to 32 bits	0 to 8 bytes	Arbitrary	0 or 16 bits (0 to 32 bits)
Preamble	Sync word	Header	Address	Payload	CRC

图 7 CC2640R2F Advanced 数据包格式

.numHdrBits 配置成 0，所以我们 TX 的包格式里面不存在 header。Payload 部分经过运行时修改后，会最终得到的 TX 包会有如下格式：

Preamble	SYNC				Payload					CRC	
0x55	0xE7	0xE7	0xE7	0xE7	0xE7	0x02	0x03	0x01	packet	0xE0	0x22

该命令的对应参数对照表可以在 TRM 中找到：

Table 23-135. CMD_PROP_TX_ADV Command Structure

Byte Index	Field Name	Bits	Bit Field name	Type	Description
14	pktConf	0	bFsOff	W	0: Keep frequency synthesizer on after command. 1: Turn frequency synthesizer off after command.
		1–2			Reserved
		3	bUseCrc	W	0: Do not append CRC. 1: Append CRC.
		4	bCrcIncSw	W	0: Do not include sync word in CRC calculation. 1: Include sync word in CRC calculation.
		5	bCrcIncHdr	W	0: Do not include header in CRC calculation. 1: Include header in CRC calculation.
		6–7			Reserved
15	numHdrBits			W	Number of bits in header (0 to 32)
16–17	pktLen			W	Packet length. 0: Unlimited
18	startConf	0	bExtTxTrig	W	0: Start packet on a fixed time from the command start trigger. 1: Start packet on an external trigger (Contact TI to enable this feature).
		1–2	inputMode	W	Input mode if external trigger is used for TX start. 00: Rising edge 01: Falling edge 10: Both edges 11: Reserved
		3–7	source	W	RAT input event number used for capture if external trigger is used for TX start.
19	preTrigger			W	Trigger for transition from preamble to sync word. If this is set to “now," one preamble as configured in the setup is sent. Otherwise, the preamble is repeated until this trigger is observed.
20–23	preTime			W	Time parameter for preTrigger
24–27	syncWord			W	Sync word to transmit
28–31	pPkt			W	Pointer to packet, or TX queue for unlimited length

图 8 CMD_PROP_TX_ADV 参数配置

5.4 TX 的实施

具体的 TX 的实施是在工程的 tx.c 文件里，都是跟前面几章的配置相关。这里介绍一下几个关键步骤。

先根据设置打开射频功能：

```
RF_open(&rfObject, RF_pModeBle, (RF_RadioSetup*)RF_ble_pCmdRadioSetup, &rfParams);
```

接着手动选择频率，自然也能用前面设置的固定频率，完全取决于下面代码中的 RF_ble_pCmdFs 参数：

```
RF_runCmd(rfHandle, (RF_Op*)RF_ble_pCmdFs, RF_PriorityNormal, NULL, 0);
```

最后就是发送了，用 BLE 广播的模式把配置好的数据发送出去：

```
RF_runCmd(rfHandle, (RF_Op*)RF_ble_pCmdBleAdvNc, RF_PriorityNormal, NULL, 0);
```

5.5 RX 的 packet 解析配置

RX 的配置也用私有模式的 CMD_PROP_RX_ADV。

在 smartrf_settings_ble.c 中：

```
uint8_t AddrArray[1] = {0xE7};

rfc_CMD_PROP_RX_ADV_t RF_ble_cmdBleGenericRx =
{
 .commandNo = 0x3804,
 .status = 0x0000,
 .pNextOp = 0,
 .startTime = 0x00000000,
 .startTrigger.triggerType = TRIG_NOW,
 .startTrigger.bEnaCmd = 0x0,
 .startTrigger.triggerNo = 0x0,
 .startTrigger.pastTrig = 0x0,
 .condition.rule = 0x1,
 .condition.nSkip = 0x0,
 .pktConf.bFsOff = 0x0,
 .pktConf.bRepeatOk = 0x0,
 .pktConf.bRepeatNok = 0x0,
 .pktConf.bUseCrc = 0x1,
 .pktConf.bCrcIncSw = 0x1,
 .pktConf.bCrcIncHdr = 0x1,
 .pktConf.endType = 0x0,
 .pktConf.filterOp = 0x0,
 .rxConf.bAutoFlushIgnored = 0x0,
 .rxConf.bAutoFlushCrcErr = 0x0,
 .rxConf.bIncludeHdr = 0x1,
 .rxConf.bIncludeCrc = 0x1,
 .rxConf.bAppendRssi = 0x0,
 .rxConf.bAppendTimestamp = 0x0,
 .rxConf.bAppendStatus = 0x0,
 .syncWord0 = 0xE7E7E7E7,
 .syncWord1 = 0,
 .maxPktLen = 64,
 .hdrConf.numHdrBits = 24,
 .hdrConf.lenPos = 8,
 .hdrConf.numLenBits = 8,
 .addrConf.addrType = 1,
 .addrConf.addrSize = 8,
 .addrConf.addrPos = 0, //For LBSfirst: addrConf.addrPos = 0 and For MBSfirst: addrConf.addrPos = 16
 .addrConf.numAddr = 1, // 1 unless several values for Prefix are in use in the protocol
 .lenOffset = 0,
 .endTrigger.triggerType = TRIG_NEVER,
 .endTrigger.bEnaCmd = 0x0,
 .endTrigger.triggerNo = 0x0,
 .endTrigger.pastTrig = 0x0,
 .endTime = 0x00000000,
 .pAddr =  AddrArray,            // uint8_t AddrArray[1] = {0xE7};
 .pQueue = 0,
 .pOutput = 0,
};
```

其中，CRC 参数的配置和 TX 包的结构一样。

特别要注意的是，RX 的包的解析方式，这里用的方法，把包的 header 用上了。具体关键配置如下：

.hdrConf.numHdrBits = 24　　：表示 Header 有 24 bit，3 个字节。

.hdrConf.lenPos = 8　　：表示 Header 的 LSB 数过来第 8bit 开始表示 payload 长度，这里是第 2 个字节。

.hdrConf.numLenBits = 8 ： 表示 Header 里面有 8bit，就是 1 个字节表示 payload 长度。

.addrConf.addrType = 1 ： 表示 Address 包含在 Header 里。

.addrConf.addrSize = 8 ： 表示 Address 只有 8bit，1 个字节。

.addrConf.numAddr = 1 ： 表示只带有一个 Address。

.pAddr = AddrArray ： 表示地址的常量，这里暂且固定为 0xE7。

最终，得到如下包的格式，Header 的第 3 个字节，是为了符合 Nordic 的格式而留：

Preamble	SYNC				Header			Payload		CRC	
					Address	Payload Length					
0x55	0xE7	0xE7	0xE7	0xE7	0xE7	0x02	0x03	0x01	packet	0xE0	0x22

其实这只是某一种 RX 的包的配置方法，根据 CC2640R2F 的 CMD_PROP_RX_ADV 命令的配置方法，结合应用层的改动，有很多种其他方法，比如说干脆像 TX 一样，不用 Header，直接用 Payload，相应固定某些字节就可以。

该命令的对应参数对照表可以在 TRM 中找到：

Table 23-137. CMD_PROP_RX_ADV and CMD_PROP_RX_ADV_SNIFF Command Structure

Byte Index	Field Name	Bits	Bit Field Name	Type	Description
14	pktConf	0	bFsOff	W	0: Keep frequency synthesizer on after command. 1: Turn frequency synthesizer off after command.
		1	bRepeatOk	W	0: End operation after receiving a packet correctly. 1: Go back to sync search after receiving a packet correctly.
		2	bRepeatNok	W	0: End operation after receiving a packet with CRC error. 1: Go back to sync search after receiving a packet with CRC error.
		3	bUseCrc	W	0: Do not check CRC. 1: Check CRC.
		4	bCrcIncSw	W	0: Do not include sync word in CRC calculation. 1: Include sync word in CRC calculation.
		5	bCrcIncHdr	W	0: Do not include header in CRC calculation. 1: Include header in CRC calculation.
		6	endType	W	0: Packet is received to the end if end trigger occurs after sync is obtained. 1: Packet reception is stopped if end trigger occurs.
		7	filterOp	W	0: Stop receiver and restart sync search on address mismatch. 1: Receive packet and mark it as ignored on address mismatch.
15	rxConf			W	RX configuration, see Table 23-143 for details.
16–19	syncWord0			W	Sync word to listen for
20–23	syncWord1			W	Alternative sync word if nonzero
24–25	maxPktLen			W	Maximum length of received packets: 0: Unlimited or unknown length
26–27	hdrConf	0–5	numHdrBits	W	Number of bits in header (0–32)
		6–10	lenPos	W	Position of length field in header (0–31)
		11–15	numLenBits	W	Number of bits in length field (0–16)
28–29	addrConf	0	addrType	W	0: Address after header 1: Address in header
		1–5	addrSize	W	If addrType = 0: Address size in bytes. If addrType = 1: Address size in bits.
		6–10	addrPos	W	If addrType = 1: Bit position of address in header. If addrType = 0: Nonzero to extend address with sync word identifier.
		11–15	numAddr	W	Number of addresses in address list
30	lenOffset			W	Signed value to add to length field
31	endTrigger			W	Trigger classifier for ending the operation
32–35	endTime			W	Time to end the operation
36–39	pAddr			W	Pointer to address list
40–43	pQueue			W	Pointer to receive queue
44–47	pOutput			W	Pointer to output structure
48–59	CMD_PROP_RX_ADV_SNIFF only: carrier sense options as given in Table 23-142 (CC13x0 only)				

图 9 CMD_PROP_RX_ADV 参数配置

5.6 RX 的实施

RX 的实施和 TX 类似，有一个 rx.c 的文件。

先根据设置打开射频功能：

```
RF_open(&rfObject, RF_pModeBle, (RF_RadioSetup*)RF_ble_pCmdRadioSetup, &rfParams);
```

接着手动选择频率，自然也能用前面设置的固定频率，完全取决于下面代码中的 RF_ble_pCmdFs 参数：

```
RF_runCmd(rfHandle, (RF_Op*)RF_ble_pCmdFs, RF_PriorityNormal, NULL, 0);
```

最后就开始打开接收窗口开始接收：

```
RF_postCmd(rfHandle, (RF_Op*)RF_ble_pCmdBleGenericRx, RF_PriorityNormal, &rx_callback, RF_EventRxEntryDone);
```

注意 RX 的实施中， RF_EventRxEntryDone 事件可以用于在 rx_callback()函数中对接收到的数据进行相应的处理。

6 测试结果

开发板：LAUNCHXL-CC2640R2

开发环境：CCS-v 7.3.0.00019

软件例程：基于 SDK -v1.40.00.45 下的 rfPacketErrorRate 例程对代码进行修改。
C:\ti\simplelink_cc2640r2_sdk_1_40_00_45\examples\rtos\CC2640R2_LAUNCHXL\drivers\rfPacketErrorRate

CC2640R2F 作为发送端，Nordic52832 作为接收端。发送 100 个数据包，发送两个字节实际数据，第一个字节固定为 0x01, 第二个字节为 packet counter, 用于计算包的个数，十六进制，用于测试 100 个包能正确收到的个数。

数据包格式：

Preamble	SYNC				Payload					CRC	
0x55	0xE7	0xE7	0xE7	0xE7	0xE7	0x02	0x03	0x01	packet	0xE0	0x22

射频参数：
频点： 2424MHz，调制方式：GFSK，Symbol rate: 1Mbps，发送功率 0dBm。
CC2640R2F 上电接 PC 端串口调试工具，115200 波特率，上电初始化界面如下：

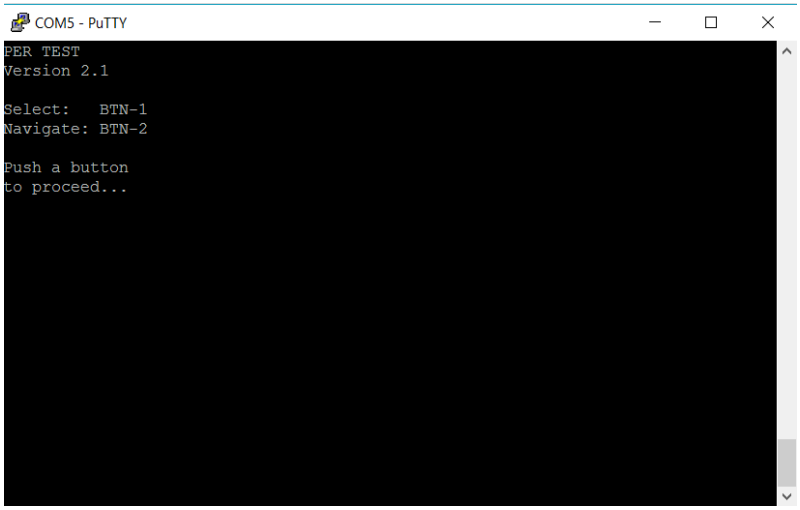

图 10 CC2640R2F 上电初始化界面

通过左右按键可在菜单中进行切换选择，选择进入 BLE mode, 频点 2424MHz，100 个包，Tx 模式。

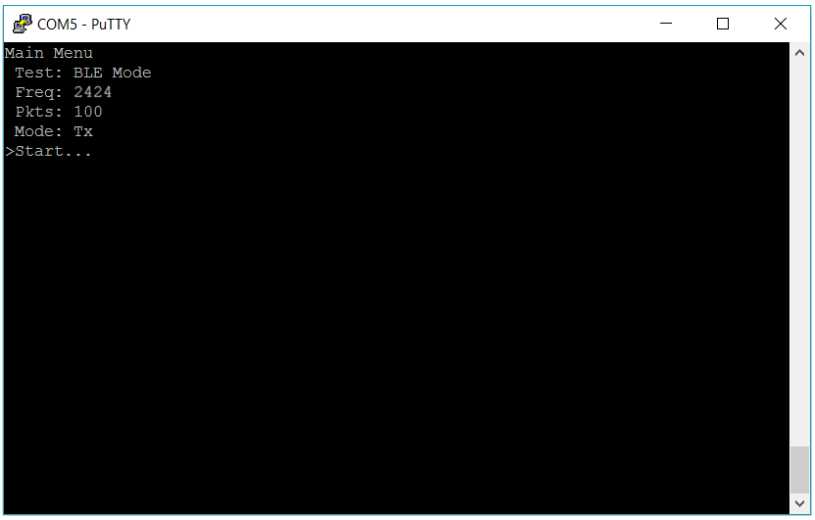

图 11 CC2640R2F 发送界面

最后点击 Start，开始发送，在 Nordic 接收端，收到正确数据包并通过校验后，串口打印 Payload 值，否则不打印。如下图，左侧为 Nordic 收到的数据，右侧发 CC2640R2F 发送状态，可以看到右侧 100 个包(0x64)全部发送完成，左侧 Nordic52832 100 个包全部正确收到。

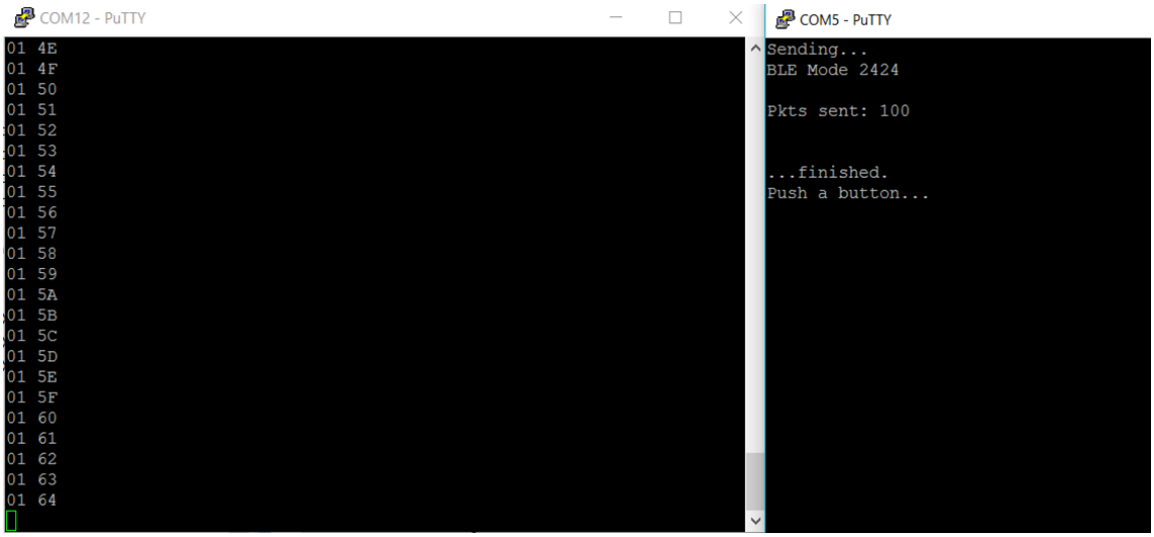

图 12 CC2640R2F 和 Nordic52832 收发界面

7 参考文献

1. *CC13x0, CC26x0 SimpleLink™ Wireless MCU Technical Reference Manual (SWCU117G)*
2. *nRF52832 Product Specification v1.3*
3. *SimpleLink CC2640R2 SDK v1.40.00.45, rfPacketErrorRate Project*
4. *Nordic 52832 SDK v14.1.0, esb_ptx, esb_prx Project*

有关 TI 设计信息和资源的重要通知

德州仪器 (TI) 公司提供的技术、应用或其他设计建议、服务或信息，包括但不限于与评估模块有关的参考设计和材料（总称“TI 资源”），旨在帮助设计人员开发整合了 TI 产品的 应用； 如果您（个人，或如果是代表贵公司，则为贵公司）以任何方式下载、访问或使用了任何特定的 TI 资源，即表示贵方同意仅为该等目标，按照本通知的条款进行使用。

TI 所提供的 TI 资源，并未扩大或以其他方式修改 TI 对 TI 产品的公开适用的质保及质保免责声明；也未导致 TI 承担任何额外的义务或责任。TI 有权对其 TI 资源进行纠正、增强、改进和其他修改。

您理解并同意，在设计应用时应自行实施独立的分析、评价和 判断， 且应全权负责并确保 应用的安全性， 以及您的 应用 （包括应用中使用的所有 TI 产品）） 应符合所有适用的法律法规及其他相关要求。你就您的 应用声明，您具备制订和实施下列保障措施所需的一切必要专业知识，能够 (1) 预见故障的危险后果，(2) 监视故障及其后果，以及 (3) 降低可能导致危险的故障几率并采取适当措施。您同意，在使用或分发包含 TI 产品的任何 应用前， 您将彻底测试该等 应用 和该等应用所用 TI 产品的 功能。除特定 TI 资源的公开文档中明确列出的测试外，TI 未进行任何其他测试。

您只有在为开发包含该等 TI 资源所列 TI 产品的 应用时， 才被授权使用、复制和修改任何相关单项 TI 资源。但并未依据禁止反言原则或其他法理授予您任何TI知识产权的任何其他明示或默示的许可，也未授予您 TI 或第三方的任何技术或知识产权的许可，该等产权包括但不限于任何专利权、版权、屏蔽作品权或与使用TI产品或服务的任何整合、机器制作、流程相关的其他知识产权。涉及或参考了第三方产品或服务的信息不构成使用此类产品或服务的许可或与其相关的保证或认可。使用 TI 资源可能需要您向第三方获得对该等第三方专利或其他知识产权的许可。

TI 资源系“按原样”提供。TI 兹免除对 TI 资源及其使用作出所有其他明确或默认的保证或陈述，包括但不限于对准确性或完整性、产权保证、无屡发故障保证，以及适销性、适合特定用途和不侵犯任何第三方知识产权的任何默认保证。

TI 不负责任何申索，包括但不限于因组合产品所致或与之有关的申索，也不为您辩护或赔偿，即使该等产品组合已列于 TI 资源或其他地方。对因 TI 资源或其使用引起或与之有关的任何实际的、直接的、特殊的、附带的、间接的、惩罚性的、偶发的、从属或惩戒性损害赔偿，不管 TI 是否获悉可能会产生上述损害赔偿，TI 概不负责。

您同意向 TI 及其代表全额赔偿因您不遵守本通知条款和条件而引起的任何损害、费用、损失和/或责任。

本通知适用于 TI 资源。另有其他条款适用于某些类型的材料、TI 产品和服务的使用和采购。这些条款包括但不限于适用于 TI 的半导体产品 (http://www.ti.com/sc/docs/stdterms.htm)、评估模块和样品 (http://www.ti.com/sc/docs/sampterms.htm) 的标准条款。

邮寄地址：上海市浦东新区世纪大道 1568 号中建大厦 32 楼，邮政编码：200122
Copyright © 2017 德州仪器半导体技术（上海）有限公司