



Panchip Microelectronics Co., Ltd.

XN297L 应用 FAQ 文档

当前版本: 1.4

发布日期: 2022.01

上海磐启微电子有限公司

地址: 上海张江高科技园区盛夏路 666 号 D 栋 3 楼

联系电话: 021-50802371

网址: <http://www.panchip.com>

文档说明

由于版本升级或存在其他原因，本文档内容会不定期进行更新。除非另有约定，本文档内容仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

商标

磐启是磐启微电子有限公司的商标。本文档中提及的其他名称是其各自所有者的商标/注册商标。

免责声明

本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，磐启微电子有限公司对本文档内容不做任何明示或暗示的声明或保证。

修订历史

版本	修订时间	描述
V1.0	2019.11	初始版本创建
V1.1	2020.03	修改 3.11 节内容描述
V1.2	2021.05	更新 RF 参数
V1.3	2021.11	更新公司地址和水印
V1.4	2022.01	根据最新 SDK 更新一些内容描述

目录

1 资料介绍.....	1
1.1 芯片的官方资料从哪里获取?	1
2 芯片特性.....	2
2.1 芯片可以工作在哪些频点?	2
2.2 芯片可以设置多大的功率?	2
2.3 XN297L 距离能达到多远?	2
2.4 芯片的工作电压为多少?	2
2.5 芯片最低休眠功耗为多大?	2
2.6 使用软件命令控制 CE 信号时, 硬件 CE 口是否可以悬空?	3
2.7 普通模式和增强模式有什么区别?	3
2.8 不同封装的芯片有什么区别吗?	3
2.9 XN297L 的通信接口是什么?	4
2.10 xn297l 是否能过认证?	4
2.11 过认证需要注意什么?	4
3 软件相关.....	5
3.1 芯片如何跳频?	5
3.2 芯片如何对码?	5
3.3 芯片 RSSI 如何获取?	5
3.4 芯片上电初始化为接收的时候为什么要加 10ms 延时?	5
3.5 芯片发射数据时 CE 应该一直保持为高吗?	6
3.6 芯片发射数据的时候为什么要加延时?	7
3.7 普通模式下, 芯片由接收状态转为发射时, 为什么对端无法收到数据?	8
3.8 芯片如何进入载波模式?	8
3.9 芯片的休眠和唤醒如何操作?	9
3.10 为什么我清除了 status 寄存器, IRQ 电平却还是保持为低?	9
3.11 增强模式下发射端只能收到 ack, 但是收不到 ack payload?	10
4 硬件相关.....	12
4.1 xn297l 硬件设计有哪些注意事项?	12

4.2 选择什么样的天线比较合适?	
4.3 天线电路如何匹配?	12
4.4 晶振和匹配电容该如何选择?	12
5 异常分析.....	13
5.1 发射接收无法正常通信如何定位?	13
5.2 发射接收距离近如何定位?	14
6 联系方式.....	15

Confidential

1 资料介绍

1.1 芯片的官方资料从哪里获取？

芯片官方资料均可从 BBS 上获取，地址为：

<http://bbs.panchip.com/forum.php?mod=viewthread&tid=7508#lastpost>

2 芯片特性

2.1 芯片可以工作在哪些频点？

ISM 频段 2400-2483 均可以使用，1M 带宽，总计 84 个频点。但是在晶振 16MHz 的整数倍(如 2416、2432MHz)等的频道及相邻正负 1MHz 的频道的接收灵敏度退化 2dB，发射信号调制精度（EVM）退化 10%，所以这些频点不建议使用。

例如 2416 是 16 的整数倍，那么 2415，2416，2417 连续三个频点均不建议使用。

另外，如果需要过 FCC、CE 等认证的话，考虑到边带指标，建议使用频点范围 2420-2465M。

2.2 芯片可以设置多大的功率？

芯片最低功率可以设置为-23dbm，最高可以设置 11dbm。中间有多个功率档位可供选择，具体请参考软件调试文档或者直接使用参考代码中给出的参考档位。通常建议使用 10dbm 以下。

2.3 XN297L 距离能达到多远？

距离跟板子的 layout，发射端使用的功率和通信的速率都有比较大的关系。1M 速率下，设置 9dbm 功率，空旷距离可达 80 米以上。

2.4 芯片的工作电压为多少？

2.3-3.3V，同样芯片 IO 口承受电压也不能超过 3.3V。

2.5 芯片最低休眠功耗为多大？

典型值 2uA。

2.6 使用软件命令控制 CE 信号时，硬件 CE 口是否可以悬空？

可以，但是注意当使用软件命令方式控制 CE 时，需要使能内部 CE 引脚弱下拉电阻，防止休眠时漏电现象发生。具体操作参考《XN297L 软件设计和调试参考.pdf》。

2.7 普通模式和增强模式有什么区别？

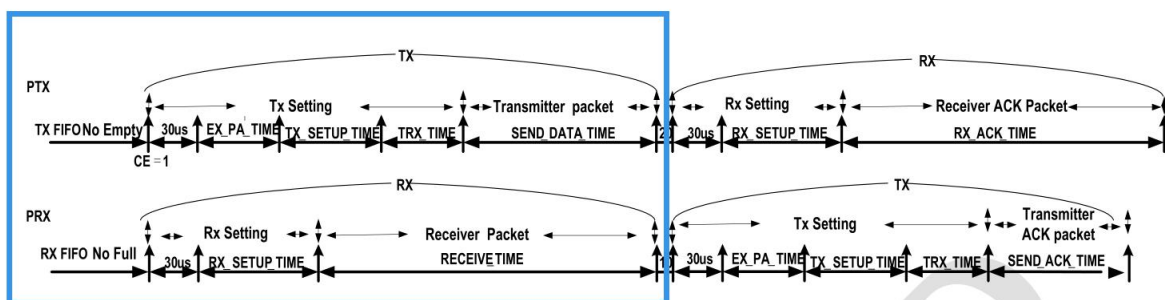


图 5-2 增强模式的时序图

普通模式通常我们称为单向模式，即发射端只发射数据，接收端只接收数据。时序图相当于只有上图中蓝色框部分。

增强模式通常我们称为双向模式，发射端发完数据后会自动切换到接收状态。同样，接收端收到数据后会自动切换到发射状态。如上图所示。

注意：因为增强模式下芯片会自动切换收发状态，如果想要达到预期的发射接收效果，就必须遵循上图中的时序。具体的计算方式请参考《XN297L 产品说明书_v4.8-ZH.pdf》第 5.6 节说明以及 XN297L_Time_set_V1d1.xlsx 文件。

2.8 不同封装的芯片有什么区别吗？

目前常见的封装为 SOP8 和 QFN20(SOP16)，在使用上有如下区别：

(1) SPI 读写驱动不同

QFN20 为标准的四线 SPI(CS,SCK,MOSI,MISO),MOSI 与 MISO 分为对应输入与输出数据。

SOP8 为三线 SPI(CS,SCK,DAT),读写数据时，DAT 首先配置为输出，写完需要读的地址后切换为输入。详细请参考《XN297L SOP8 使用手册.pdf》第 4 节。

(2) CE 使用上的区别

QFN20 封装的芯片有 CE 脚。在软件上支持通过硬件 IO 口或者软件命令控制 CE 的高低状态。SOP8 没有 CE 脚，只能使用软件命令去控制 CE。

软件命令控制示例:

```
#define CE_HIGH RF_WriteReg(CE_FSPI_ON, 0)
#define CE_LOW RF_WriteReg(CE_FSPI_OFF, 0)
```

硬件 IO 控制示例:

```
#define CE_HIGH GPIOB->ODR |= GPIO_Pin_1
#define CE_LOW GPIOB->ODR &= (~GPIO_Pin_1)
```

(3) IRQ 使用

QFN20 封装有 IRQ 脚, 在设置了 CONFIG 寄存器 bit4-bit6 为 0 后, 可以通过 IRQ 判断 RF 的状态(发射数据成功/接收数据成功/发送失败), 或者软件查询 STATUS 寄存器。

SOP8 没有 IRQ 脚, 只能通过查询 STATUS 寄存器来判断 RF 的状态。

具体代码实现请参考三线/四线 SPI 的 demo 程序。

2.9 XN297L 的通信接口是什么?

xn297l 通过 SPI 接口与 MCU 通信。

休眠模式和待机模式-I, SPI 最高速率为 1Mbps, 其它状态 SPI 最高速率为 4Mbps。推荐使用软件模拟 SPI, 速率控制在 1Mbps 以下。

2.10 xn297l 是否能过认证?

xn297l 可以过 FCC、CE、KC、日本等多国认证, 已经有很多客户通过了认证机构的检测, 拿到相应的报告。

2.11 过认证需要注意什么?

接收过认证时, VCO 配置最小, RF_CAL 的配置(06, 37, 5D)不能用于发射模式。

详细说明请参考安规调试文档。

3 软件相关

3.1 芯片如何跳频？

跳频原理：通过更改 RF_CH 寄存器的值即可切换频点，实际的频点为 $2400 + \text{RF_CH}$ ，切换之后新频点会有一段稳定时间，大约为 400us。

跳频机制：客户根据实际方案定制，可以参考 XN297L_SDK 中 hopping 例程。

3.2 芯片如何对码？

对码实现的方向：发射和接收双方同时将通信地址或者频点改为一个特定的值来区分没配对的设备。具体根据实际方案定制，可以参考 XN297L_SDK 中 pair 例程。

3.3 芯片 RSSI 如何获取？

xn297l 的 rssi 值不是很精确，不适合用来判断距离。读取方法：将寄存器 RSSI_EN 和 RSSI_SEL 设置为高，DATAOUT_SEL 设置为低，就可以从寄存器 DATAOUT 读出 RSSI 的值。DATAOUT 低四位表示接收数据的信号强度，DATAOUT 高四位表示接收信号前干扰信号的强度。要求接收数据的信号强度需要在收到数据包后的 100ms 内读出。可以参考 XN297L_SDK 中 rssi 例程。

3.4 芯片上电初始化为接收的时候为什么要加 10ms 延时？

在提供的参考 demo 中，一般设置成接收模式的代码如下：

```
void RF_RxMode(void)
{
    RF_WriteReg(W_REGISTER + CONFIG, 0X8F);
    delay_ms(10);
    CE_HIGH;
    delay_ms(10);
}
```

芯片在上电的过程中，晶振会有一段稳定时间，大约在 7ms。在软件配置 PWR_UP (CONFIG 寄存器 BIT1) 为 1 后，至少需要延时 7ms 以上确保稳定。

3.5 芯片发射数据时 CE 应该一直保持为高吗？

分为两种情况，250K 速率和 2M/1M 速率有所区别。

(1) 250K 速率

250K 速率下 CE 不能频繁操作，通常 CE 在初始化为发射模式的时候设置为高，在之后发数据的过程中一直保持为高(2M/1M 速率也可以选择这种操作方式)。

```

/*****
// @brief RF设置为TX mode
// @param None
// @retval None
*****/
void RF_TxMode(void)
{
    RF_WriteReg(W_REGISTER + CONFIG, 0X8E);
    delay_ms(10);
    CE_HIGH;
    delay_ms(10);
}

/*****
// @brief RF发射数据
// @param ucPayload: 发射数据的起始地址
//          length: 发射数据长度
// @retval None
*****/
void RF_TxData( uint8_t *ucPayload, uint8_t length)
{
    RF_WriteBuf(W_TX_PAYLOAD, ucPayload, length);
    delay_ms(2);
    while(!(RF_GetStatus() & TX_DS_FLAG))
    {
        delay_ms(1);
    }
    RF_WriteReg(FLUSH_TX, 0); //清除RF的TX FIFO
    RF_ClearStatus();
}

```

(2) 2M/1M 速率

2M/1M 速率下，通常 CE 在初始化为发射模式的时候设置为低，需要发数据的时候，把 CE 置高，发送完成后把 CE 置低。

```

/*****
// @brief RF设置为TX mode
// @param None
// @retval None
*****/
void RF_TxMode(void)
{
    RF_WriteReg(W_REGISTER + CONFIG, 0X8E);
    CE_LOW;
}

```

```

/*****
// @brief RF发射数据
// @param ucPayload: 发射数据的起始地址
//          length: 发射数据长度
// @retval None
*****/
void RF_TxData( uint8_t *ucPayload,  uint8_t length)
{
    CE_HIGH;
    delay_100us(1);
    RF_WriteBuf(W_TX_PAYLOAD, ucPayload, length);
    delay_ms(1);
    while(!(RF_GetStatus() & TX_DS_FLAG))
    {
        delay_ms(1);
    }
    RF_WriteReg(FLUSH_TX, 0); //清除RF的TX FIFO
    RF_ClearStatus();
    CE_LOW;
}

```

后者由于在没发数据的时候 CE 置低，会使 RF 状态进入 STB3 状态（不置低会进入 STB2 状态），会更节省功耗一些。

3.6 芯片发射数据的时候为什么要加延时？

如 3.5 中发送流程所示，在将需要发送的数据写入 TX_FIFO 后，需要加入一定的延时时间，如下图所示：

```

RF_WriteBuf(W_TX_PAYLOAD, ucPayload, length);
delay_ms(1);
while(!(RF_GetStatus() & TX_DS_FLAG))
{
    delay_ms(1);
}

```

XN297L 在发射数据的过程中，不能频繁操作 SPI，否则会影响到 RF 的正常发射。一般程序中使用延时来规避其它意外的干扰操作。这个时间跟 BB_CAL 的值，通讯的速率，数据长度有关，可以根据《XN297L 产品说明书_v4.8-ZH.pdf》5.6 节中图 5-2 计算出发射的时间(30+EX_PA_TIME + TX_SETUP_TIME + TRX_TIME+SEND_DATA_TIME)。上面的操作中，1ms 后查询是否发送成功，如果没成功延时 1ms 再次查询，可避免持续查询 SPI。如果用户确认 1ms 能发送完成，后面查询的操作可以省略，延时后直接清 FIFO 和 status 即可。

3.7 普通模式下，芯片由接收状态转为发射时，为什么对端无法收到数据？

xn297l 接收转发射，第一包发送的时间会延时，延时的时间 =RX_SETUP_TIME+RX_ACK_TIME+ARD。发完之后需要加 3ms 以上的延时。

接收转发射后，在 CE_HIGH 的情况下，如果不往 TX FIFO 写数据，TX 也会发射出去，发送数据是随机的，而且此时的 TX FIFO 无法清除，所以在接收转发送后，需要在 10us 内往 TX FIFO 内写 payload。

接收/发射转换的具体流程请参考《XN297L 软件设计和调试参考.pdf》3.6 节。

3.8 芯片如何进入载波模式？

单载波是未经调制过的波形，一般用于测量频偏。

代码示例：

```

/*****
// @brief RF设置为载波模式
// @param ucChannel_Set: 载波的频点
// @retval None
*****/
void RF_Carrier( uint8_t ucChannel_Set)
{
    uint8_t BB_cal_data[] = {0x12,0xED,0x67,0x9C,0x46};
    uint8_t RF_cal_data[] = {0xF6,0x3F,0x5D};
    uint8_t RF_cal2_data[] = {0x45,0x21,0xEF,0x2C,0x5A,0x42};
    uint8_t Dem_cal_data[] = {0xE1};
    uint8_t Dem_cal2_data[] = {0x0B,0xDF,0x02};

    RF_WriteReg(RST_FSPI, 0x5A); //Software Reset
    RF_WriteReg(RST_FSPI, 0xA5);

    uint8_t feature = 0x20; //使用软件CE

    RF_WriteReg(W_REGISTER + FEATURE, feature);

    CE_LOW;
    delay_100us(5); //delay 500us
    RF_WriteReg(W_REGISTER + CONFIG, 0x8E); //tx mode
    RF_WriteReg(W_REGISTER + RF_CH, ucChannel_Set); //单载波频点
    RF_WriteReg(W_REGISTER + RF_SETUP, RF_POWER); //9dbm
    RF_WriteBuf(W_REGISTER + BB_CAL, BB_cal_data, sizeof(BB_cal_data));
    RF_WriteBuf(W_REGISTER + RF_CAL2, RF_cal2_data, sizeof(RF_cal2_data));
    RF_WriteBuf(W_REGISTER + DEM_CAL, Dem_cal_data, sizeof(Dem_cal_data));
    RF_WriteBuf(W_REGISTER + RF_CAL, RF_cal_data, sizeof(RF_cal_data));
    RF_WriteBuf(W_REGISTER + DEM_CAL2, Dem_cal2_data, sizeof(Dem_cal2_data));
    delay_100us(5);
}

```

代码示例中①、②的顺序不能颠倒，①中 feature 的取值根据客户程序中使用的是软件控制 CE(0x20)还是硬件控制 CE(0x00)而定。

3.9 芯片的休眠和唤醒如何操作？

芯片的休眠操作和唤醒操作无特定的指令，是需要客户设置寄存器的值来控制。请参考《XN297L 产品说明书_v4.8-ZH.pdf》图 4-1(状态图)和表 4-1。

表 4-1 XN297L 操作模式配置

模式		PWR_DN	STB1	STB3	STB2	RX	TX
控制位	PWR_UP	0	1	1	1	1	1
	EN_PM	0	0	1	1	1	1
	CE	0	0	0	1	1	1
	PRIM_RX	X	X	X	0	1	0
	FIFO state	X	X	X	TX FIFO 空	X	数据在 TX FIFOs 中

例如要进入休眠模式(即 PWR_DN 模式),将表格左侧的几个 PWR_UP,EN_PM,CE 置为低就可以了。反之唤醒就根据要进入的模式设置左侧的寄存器值和控制信号即可。

休眠示例程序：

```
void RF_SleepMode(void)
```

```
{
```

```
    RF_WriteReg(W_REGISTER + CONFIG, 0X0C);
```

```
    CE_LOW;
```

```
}
```

注意事项：

XN297L 的 MISO 输出为高阻，当 MCU 与 MISO 相连的 PIN 脚输入设置为浮空，由于 RF 的输出是高阻，会导致 MCU 的 PIN 脚的 pmos 和 nmos 同时导通，产生漏电，当输入模式设置为下拉和上拉时问题解决。

3.10 为什么我清除了 status 寄存器，IRQ 电平却还是保持为低？

先清除 FIFO，再清除 status 寄存器。

3.11 增强模式下发射端收只能收到 ack，但是收不到 ack payload?

增强模式下，接收端收到数据后，芯片会自动切换到发射状态。具体时序图如下：

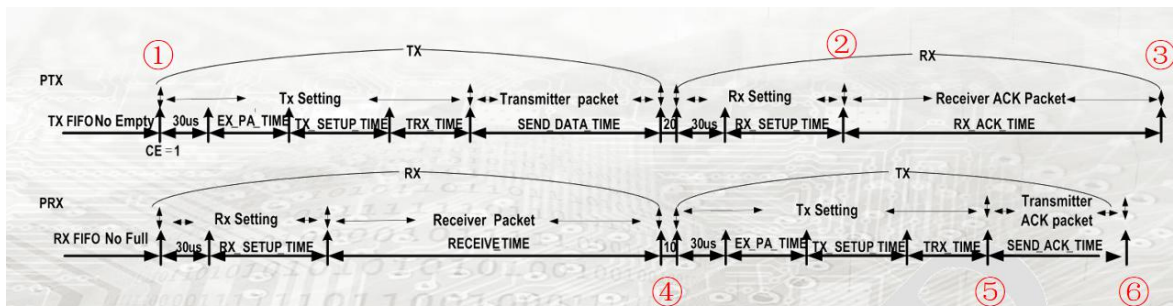


图 5-2 增强模式的时序图

重要的几个时间点：

- | | |
|------------------|------------------------------|
| ①：发射端(PTX)开始发射数据 | ④：接收端(PRX)开始切换到发射状态（注意和①的区别） |
| ②：发射端(PTX)接收窗口开始 | ⑤：接收端(PRX)发射窗口开始 |
| ③：发射端(PTX)接收窗口结束 | ⑥：接收端(PRX)发射窗口结束 |

如要确保双向通信成功，需要满足以下三个条件：

- (1) PTX（或 PRX）发射的锁相环稳定+功放使能+锁相环开环的三段时间之和，大于 PRX（或 PTX）接收的锁相环稳定时间 20us 以上，这样可以保证 PTX（或 PRX）发射数据的时间段落在 PRX（或 PTX）接收数据的时间段内，即： $EX_PA_TIME + TX_SETUP_TIME + TRX_TIME > RX_SETUP_TIME + 20us$;
- (2) PRX 发送 ACK 的锁相环稳定+功放使能+锁相环开环+发送 ACK 的四段时间之和，小于 PTX 接收的锁相环稳定+等待 ACK 的两端时间之和 80us 以上，保证 PRX 回复 ACK 的时间端落在 PTX 等待 ACK 的时间段内，即： $EX_PA_TIME + TX_SETUP_TIME + TRX_TIME + SEND_ACK_TIME < RX_SETUP_TIME + RX_ACK_TIME - 80us$ 。
- (3) 对于 SPI 速率不高的单片机(一般指 500k 以下)，需要注意如下：

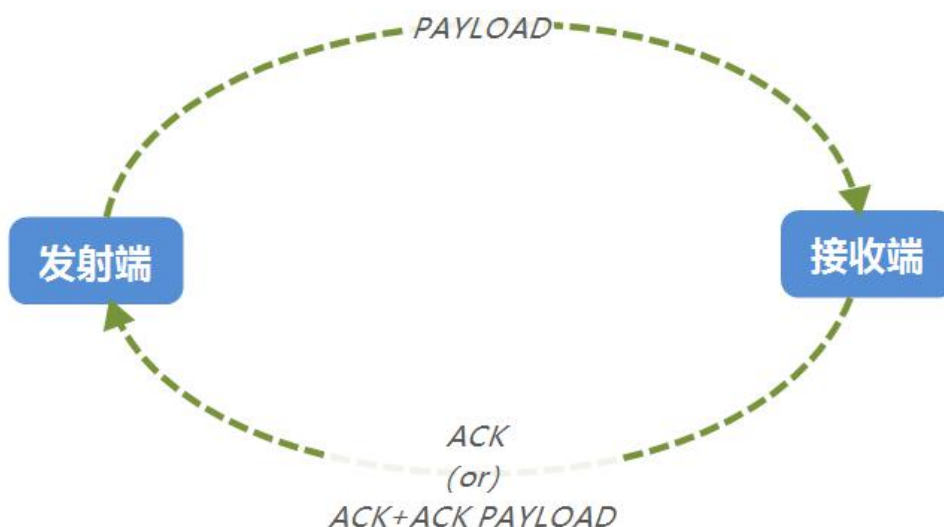
接收端收到数据以后需要在⑤(发射窗口)前把需要发送的数据写入 TX_FIFO,否则只会发射 ack 包(组帧格式 III)。

值得注意的是，芯片在④处就会自动切换为发射状态，到⑤处就会开始发射数据。那么留给用户写入发射数据的时间为 $(10+30+EX_PA_TIME + TX_SETUP_TIME + TRX_TIME)$ 。以 BB_CAL 设置 0x12,0xED,0x67,0x9C,0x46, 1M 速率，数据长度 8bytes 为例，我们可以算出 ④-⑤之间的时间为 647.5us。意味着我们必须在芯片接收到数据后(④处)，要在 647.5us 内把数据写入 TX_FIFO 中。但遗憾的是，我们在应用过程中，往往会出现以下情况：

- (I) 第一种情况是接收端没有及时地监测到芯片已经收到了数据，比如程序中设定几个 ms 才去查询一次芯片的接收情况，那么芯片的发射窗口早就结束了，所以发射端

只能收到芯片自动回复的 ack 包，而不是 ack+payload 包。

(II) 另外一种情况是即使用户很及时地监测到芯片收到了数据，假设是从④处就开始准备把数据写入 TX_FIFO,但是由于 SPI 速率比较慢，假设 SPI 将 ack payload 写入 FIFO 中需要约 647us 以上。这个时间意味着当还没将发射数据写入完成就已经开始发射了。而且在实际应用中，用户在收到数据后往往是先将收取到的数据先读取下来，经过逻辑判断后再将需要回复的数据写入 TX_FIFO,那么这个时间就需要更长了，导致最终发出的也是 ack 包，而不是 ack+payload 包。



4 硬件相关

4.1 xn297l 硬件设计有哪些注意事项？

请参考《XN297L 硬件设计和调试参考.pdf》。

4.2 选择什么样的天线比较合适？

根据具体应用选择，常见的几种天线比较如下：

种类	特点
板载 PCB 天线	成本低，但是性能有限
SMT 陶瓷天线	占用面积少，集成度高，容易更换
棒状天线	性能好，方便更换

4.3 天线电路如何匹配？

如果需要过认证，请参考《XN297L 安规设计和调试参考.pdf》。

4.4 晶振和匹配电容该如何选择？

请参考《XN297L 外接晶振匹配电容应用指南.pdf》。

5 异常分析

5.1 发射接收无法正常通信如何定位？

- (1) 有条件的话，先定位是发射有问题还是接收有问题

如果手头有能确认正常工作的发射/接收的情况下，用来作为参照板，定位是发射有问题还是接收有问题。

- (2) 测量硬件是否有异常

使用万用表或者示波器，测试芯片 XC1, XC2, ANT 脚。在设置为接收模式/发射模式 (CE 保持为高) 时, XC1, XC2 电压为 0.6-1V, 如果使用示波器则能看到 16MHz 的正弦波。ANT 脚电压为 1.8V。

如果 XC1, XC2 电压不正常，说明晶振起振不正常，请做以下测试去定位分析：

硬件方面：

- ① 是否有虚焊/短路。
- ② XC2 脚是否有串 510 欧姆电阻，如果没有，务必加上。
- ③ 晶振与芯片之间是否走线太长。
- ④ 晶振和外围电容是否匹配，参考本文档 4.4 节。
- ⑤ 如果使用的是 3225 等四脚的晶振封装，请确认管脚对应是否正确。
- ⑥ 测量 XC1, XC2 二极管特性和对地阻值是否正常，二极管特性测量方法：用万用表二极管档测量各个管脚对地的二极管特性，地接正极，管脚接负极，正常的值应该在 0.3~0.8。

软件方面：

- ① 验证 SPI 读写是否正常。

对具备 R/W 特性的寄存器，先写入一个特定的值，然后再读出来，如果写入和读出一致，说明 SPI 读写正常。如果不正常，请检查 SPI 是否符合标准。SPI 速率请参考本文档 2.9 节。

- ② 检查软件初始化流程是否正确，具体参考给出的软件调试文档或者软件 demo。
- ③ 芯片在上电的过程中，晶振会有一段稳定时间，大约在 7ms，检查下软件上是否有留够足够长的时间，参考本文档 3.4 节。

如果 XC1, XC2 脚正常，ANT 脚无电压，可能的原因有：

- ① CE 未置高。注意 CE 分为软件控制和硬件 IO 控制，取决于 FEATURE 寄存器的 BIT5 设置。Bit5=0；CE 由 CE 引脚控制，Bit5=1；CE 由命令方式控制。在操作 CE 之前，请先确认 FEATURE 寄存器 BIT5 有被正确设置，否则对 CE 的操作无效。
- ② QFN20 封装的芯片有两个 VDD，请确认电路上都有接入电源。
- (3) 如果如上硬件测试未见明显异常，可从以下软件方面去定位：
 - ① 检查发射/接收软件的参数是否按照 PANCHIP 给出的参考文档配置。
 - ② 如果有逻辑分析仪的话，最好使用逻辑分析仪抓取 SPI 的数据验证写入的寄存器数据是否与软件的逻辑一致。
- (4) 使用频谱仪查看发射接收状态

如果客户公司有频谱仪，在检查硬件和软件均没发现明显异常的情况下，请使用频谱仪查看以下状态是否正常：

 - ① 发射端：是否能在频谱仪上查看到发射频谱信号。
 - ② 接收端：是否能在频谱仪上查看到本振信号，本振频点按照如下公式计算：
$$LO = (CH+2) * 8/7, CH \text{ 为软件设置的频点。}$$
 - ③ 设置成载波模式查看发射端和接收端之间是否存在较大的频偏。

5.2 发射接收距离近如何定位？

- (1) 使用频谱仪测试发射功率是否正常。
- (2) 使用频谱仪测试频偏是否正常。
- (3) 检查软件在发射数据的时候是否有操作 SPI，参考本文档 3.6 节说明。
- (4) 检查是否是程序中跳频、双向模式等流程中一些因素不合理导致的丢包，推荐先采用 1M 速率，普通模式先测试距离。
- (5) 排除是否因为测试的场景下干扰太大导致丢包严重。可以去外面空旷场景下测试或者在软件上使用 2.5G 的频点去测试，避开 2.4G 的蓝牙，WIFI 等信号干扰。(合法的频点为 2400-2483，但是测试时可以使用 2.5G 频点，比如将 RF_CH 寄存器设置为 110，那么通信频点为 2510。)

6 联系方式

上海磐启微电子有限公司

电话：021-50802371

传真：021-50802372

地址：上海张江高科技园区盛夏路 666 号 D 栋 3 楼

苏州磐启微电子有限公司

电话：0512-68136052

传真：0512-68136051

地址：苏州工业园区崇文路 199 号富华科技大厦 3-A

上海磐启微电子有限公司深圳分公司

电话：0755-26403799

传真：0755-26403799

地址：深圳市南山区高新南一道 9 号中科大厦 3A08