



Panchip Microelectronics Co., Ltd.

PAN3029/3060 系列提前中断应用参考文档

当前版本: 1.1

发布日期: 2024.04

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 302 室

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	2023.06	初始版本
V1.1	2024.04	修改函数接口和芯片描述

目录

1 功能介绍	1
2 软件设计参考	2
2.1 软件设计流程	2
2.2 软件设计验证	2
2.2.1 验证步骤	2
2.2.2 SDK 示例	3
2.2.3 验证结果	5
2.3 逻辑分析仪抓取	6
2.3.1 验证步骤	6
2.3.2 验证结果	6
2.4 注意事项	6
2.5 临界情况测试	6

1 功能介绍

PAN3029/3060 系列芯片提供提前中断功能。提前中断功能是在芯片读取一帧数据的过程中，查看已经解出来的数据，判断是不是自己想要的，再决定继续读取还是放弃这帧数据。

流程图如下所示：

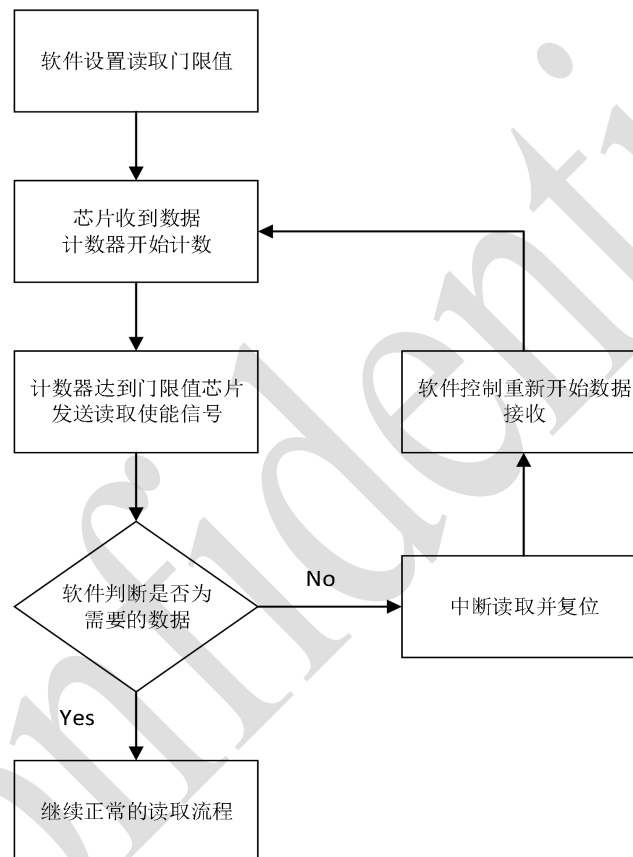


图 1-1 提前中断流程图

2 软件设计参考

2.1 软件设计流程

1、芯片初始化；

2、配置射频参数；

3、将芯片配置为提前中断模式，通过寄存器设置提前中断起始字节和包头长度（包头长度也称为计数器门限值）。起始字节是指提前中断从第几个字节开始检查，包头长度是指提前中断检查多少个字节的数据（包头长度只支持 8 字节或 16 字节，分别使用 PLHD_LEN8/PLHD_LEN16 表示）；

4、芯片进入接收模式；

5、芯片接收数据，内部计数器开始计数，收到一个字节加 1，直至计数器达到包头长度之后，芯片会产生提前中断信号，供软件读取；

6、软件判断是否为自己想要的的数据，如果是，则继续往下读取，如果不是，则停止读取。

2.2 软件设计验证

2.2.1 验证步骤

1、发送模组周期性发送 100 字节数据包，前 30 字节数据为

```
uint8_t tx_test_buf[100] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
```

2、接收模组配置提前中断模式，并设置从第 5 字节开始，共检查 16 字节数据；

```
rf_set_plhd_rx_on(5, PLHD_LEN8);
```

3、当产生提前中断信号时，将提前中断得到的数据打印出来；

4、继续往下接收，将本帧数据全部打印出来；

5、通过串口助手查看打印结果。

2.2.2 SDK 示例

Main.c 参考代码：

```
ret = rf_init();
if(ret != OK)
{
    printf("  RF Init Fail");
    while(1);
}
rf_set_default_para();
rf_set_plhd_rx_on(5,PLHD_LEN16);
rf_enter_continuous_rx();
while (1)
{
    rf_irq_process();
    if(rf_get_rcv_flag() == RADIO_FLAG_PLHDRXDONE)
    {
        rf_set_rcv_flag(RADIO_FLAG_IDLE);
        printf("###Plhd Rev :##\r\n");
        for(i = 0; i < RxDoneParams.PlhdSize; i++)
        {
            printf("0x%02x ", RxDoneParams.PlhdPayload[i]);
        }
        printf("\r\n");
    }

    if(rf_get_rcv_flag() == RADIO_FLAG_RXDONE)
```

```
{  
    rf_set_recv_flag(RADIO_FLAG_IDLE);  
    printf("Rx : SNR: %f ,RSSI: %f\r\n", RxDoneParams.Snr, RxDoneParams.Rssi);  
    for(i = 0; i < RxDoneParams.Size; i++)  
    {  
        printf("0x%02x ", RxDoneParams.Payload[i]);  
    }  
    printf("\r\n");  
    cnt ++;  
    printf("###Rx cnt %d##\r\n", cnt);  
}  
if((rf_get_recv_flag() == RADIO_FLAG_RXTIMEOUT) || (rf_get_recv_flag() ==  
RADIO_FLAG_RXERR))  
{  
    rf_set_recv_flag(RADIO_FLAG_IDLE);  
    printf("Rxerr\r\n");  
}  
}
```

Radio.c 参考代码:

```
void rf_irq_process(void)函数中  
if(irq & REG_IRQ_RX_PLHD_DONE)  
{  
    RxDoneParams.PlhdSize = rf_get_plhd_len();  
    RxDoneParams.PlhdSize = rf_plhd_receive(RxDoneParams.PlhdPayload,  
RxDoneParams.PlhdSize);  
    irq &= ~REG_IRQ_RX_PLHD_DONE;  
    rf_clr_irq(REG_IRQ_RX_PLHD_DONE);  
    rf_set_recv_flag(RADIO_FLAG_PLHDRXDONE);  
}
```

示例代码配置了提前中断模式，并设置从第 5 字节开始，共检查 16 字节数据。接收到提前中断信号后，主函数中选择将提前中断接收到的内容打印出来，并继续接收；随后模组会再产生一个接收中断信号，主函数中再将完整接收到的数据内容打印出来。

如果需要提前中止接收，只需要在接收到提前中断信号后执行 `rf_set_refresh()`；即可。即：

```
if(rf_get_recv_flag() == RADIO_FLAG_PLHDRXDONE)
{
    rf_set_recv_flag(RADIO_FLAG_IDLE);
    printf("###Plhd Rev :##\r\n");
    for(i = 0; i < RxDoneParams.PlhdSize; i++)
    {
        printf("0x%02x ", RxDoneParams.PlhdPayload[i]);
    }
    printf("\r\n");
    rf_set_refresh();//stop it
}
```

2.2.3 验证结果

串口助手显示结果为：

```
###Plhd Rev :##
0x05 0x06 0x07 0x08 0x09 0x0a 0x0b 0x0c 0x0d 0x0e 0x0f 0x10 0x11 0x12 0x13 0x00
Rx : SNR: 14.084321 ,RSSI: -60.000000
0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0a 0x0b 0x0c 0x0d 0x0e 0x0f 0x10
0x11 0x12 0x13 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09
###Rx cnt 1##
###Plhd Rev :##
0x05 0x06 0x07 0x08 0x09 0x0a 0x0b 0x0c 0x0d 0x0e 0x0f 0x10 0x11 0x12 0x13 0x00
Rx : SNR: 16.130527 ,RSSI: -60.000000
0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0a 0x0b 0x0c 0x0d 0x0e 0x0f 0x10
0x11 0x12 0x13 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09
###Rx cnt 2##
```

图 2-1 串口助手显示结果

根据结果显示，接收模组发生了提前中断，获取到了指定的数据，并继续执行接收，接收到完整数据包。

2.3 逻辑分析仪抓取

2.3.1 验证步骤

- 1、发送模组周期性发送数据包；
- 2、接收模组分别使用提前中断接收模式和正常接收模式进行接收，两种模式交替进行接收；
- 3、使用逻辑分析仪 Channel8 抓取发送端开始和结束信号，Channel11 抓取接收端信号。

2.3.2 验证结果

抓取结果如下图所示：

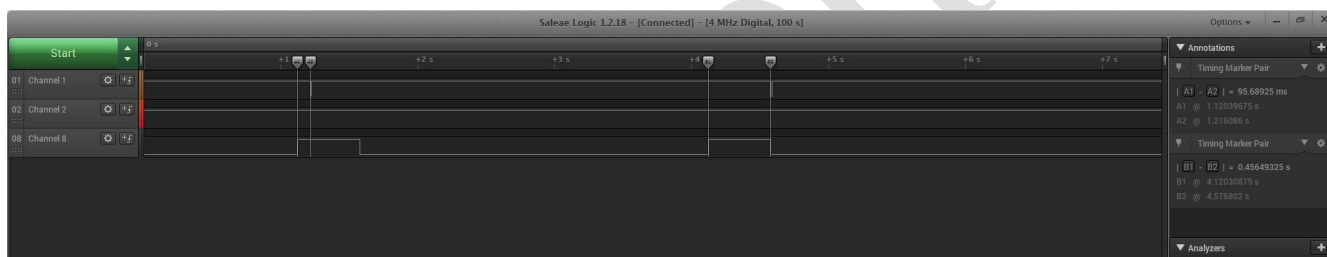


图 2-2 抓取结果

由结果可以看出，提前中断接收模式在 95ms 时就产生了提前中断，供用户判断。正常接收模式需要在 456ms 才产生完整的接收中断。

2.4 注意事项

提前中断功能只支持读取两种数据长度，即 8 字节/16 字节，分别使用 PLHD_LEN8/PLHD_LEN16 表示。不可以使用自定义参数。

提前中断功能获取数据时，使用 `rf_plhd_receive()` 接口函数，与普通数据包 `rf_receive()` 接口函数不同，其内部 FIFO 地址不同。

2.5 临界情况测试

对于临界情况，参考如下测试结果（SF7，BW500K）：

发送端配置	接收端配置	接收结果
-------	-------	------

发送端发包长度	提前中断起始字节	提前中断包头长度	提前中断结果	接收中断结果
100 字节	分别设置 80/81/82	PLHD_LEN16	均有	均有
100 字节	分别设置 83/84	PLHD_LEN16	均有且数据内容不可靠	均有
100 字节	分别设置 85~100	PLHD_LEN16	均无	均有

可以看到，提前中断功能的设计初衷是在数据内容的包头位置，做提前检测，如果包头数据内容不符合需求，就可以提前退出接收状态，以达到节省接收时间的目的。

在临界值附近，提前中断功能的软件应用价值不大，临界值附近不推荐使用此功能。