



武汉芯源半导体有限公司

WUHAN XINYUAN SEMICONDUCTOR CO., LTD

# 提前中断应用手册

应用笔记

版本号：Rev 1.0



# 目录

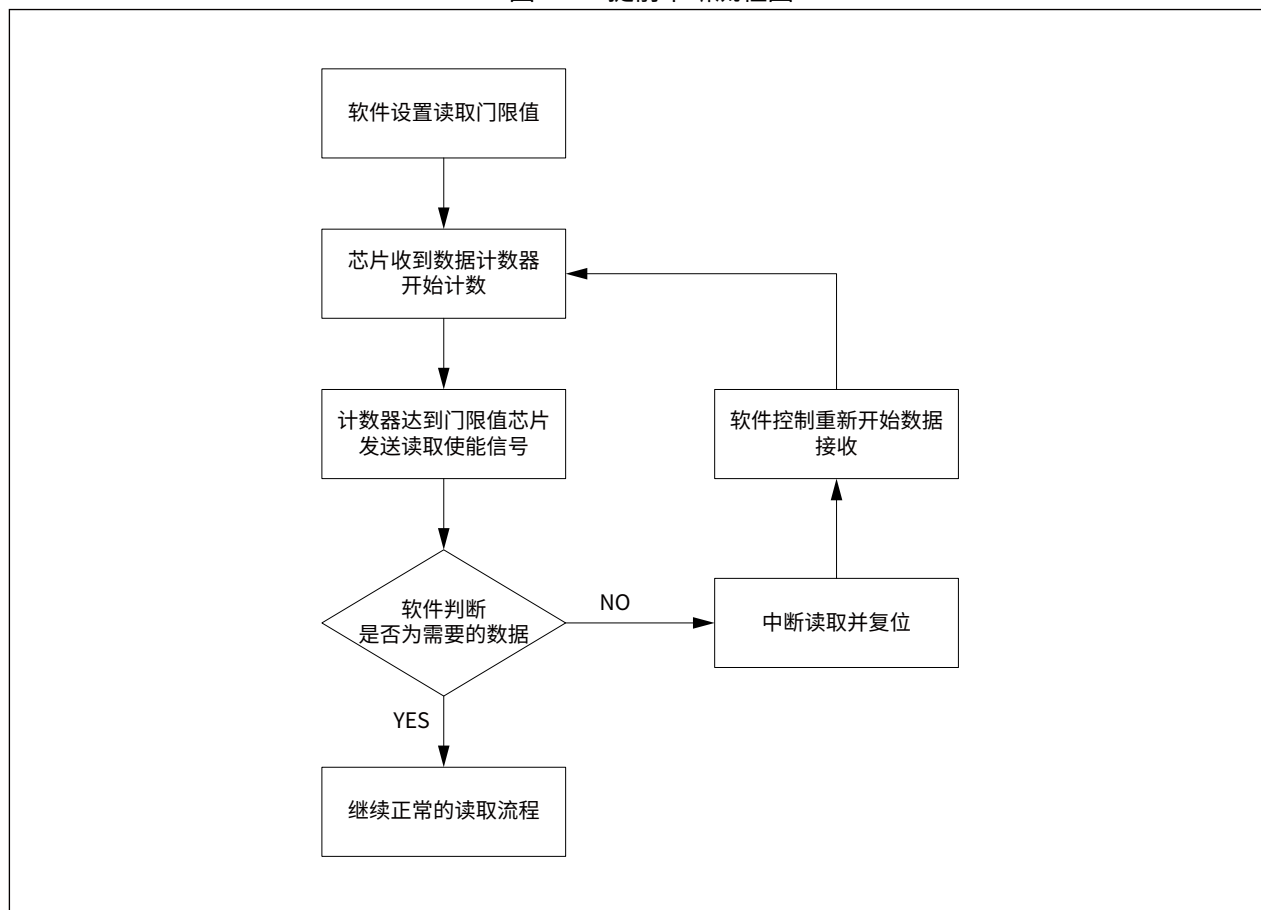
1	功能介绍 .....	3
2	软件设计参考 .....	4
2.1	软件设计流程.....	4
2.2	软件设计验证.....	4
2.2.1	验证步骤.....	4
2.2.2	SDK 示例 .....	5
2.2.3	验证结果.....	7
2.3	逻辑分析仪抓取 .....	8
2.3.1	验证步骤.....	8
2.3.2	验证结果.....	8
2.4	注意事项 .....	8
3	版本信息 .....	9

## 1 功能介绍

CW32W031 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. 将芯片配置为提前中断模式；
3. 通过寄存器设置包头长度，或者称为计数器门限值。提前中断功能需要设置确定从第几个字节开始检查，以及检查多少个字节的数据（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_buf[256]={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_LEN16);
```

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)
{
    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_rcv_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_rcv_flag() == RADIO_FLAG_RXTIMEOUT) || (rf_get_rcv_flag() == RADIO_FLAG_RXERR))
    {
        rf_set_rcv_flag(RADIO_FLAG_IDLE);
        printf("Rxerr\r\n");
    }
}
```

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“PAN3028\_rst();”，即：

```
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");
    PAN3028_rst();
}
```

### 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: 11.464460 ,RSSI: -85.000000  
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: 8.774908 ,RSSI: -82.000000  
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##
```

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

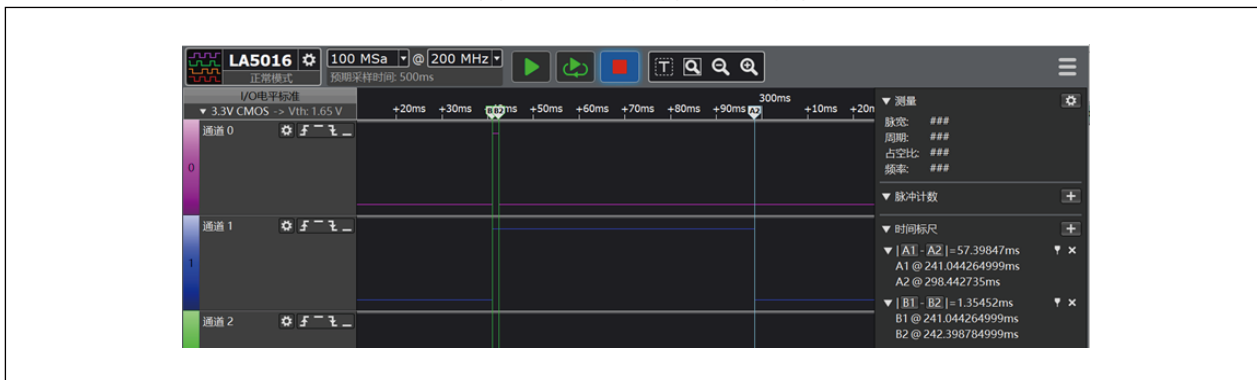
### 2.3.1 验证步骤

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
3. 使用逻辑分析仪 Channel1 抓取提前中断信号，Channel1 抓取正常接收端信号。

### 2.3.2 验证结果

抓取结果如下图所示：

图 2-1 逻辑分析仪抓取结果



由结果可以看出，提前中断接收模式在 1.35ms 时就产生了提前中断，供用户判断。正常接收模式需要在 57.39ms 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 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 PAN3028\_plhd\_receive() 接口函数，与普通数据包 PAN3028\_recv\_packet() 接口函数不同，其内部 FIFO 地址不同。



### 3 版本信息

表 3-1 文档修订信息

日期	版本	变更信息
2023-03-09	Rev 1.0	初始发布