



武汉芯源半导体有限公司

WUHAN XINYUAN SEMICONDUCTOR CO., LTD

# CAD 应用参考手册

应用笔记

版本号：Rev 1.1



# 目录

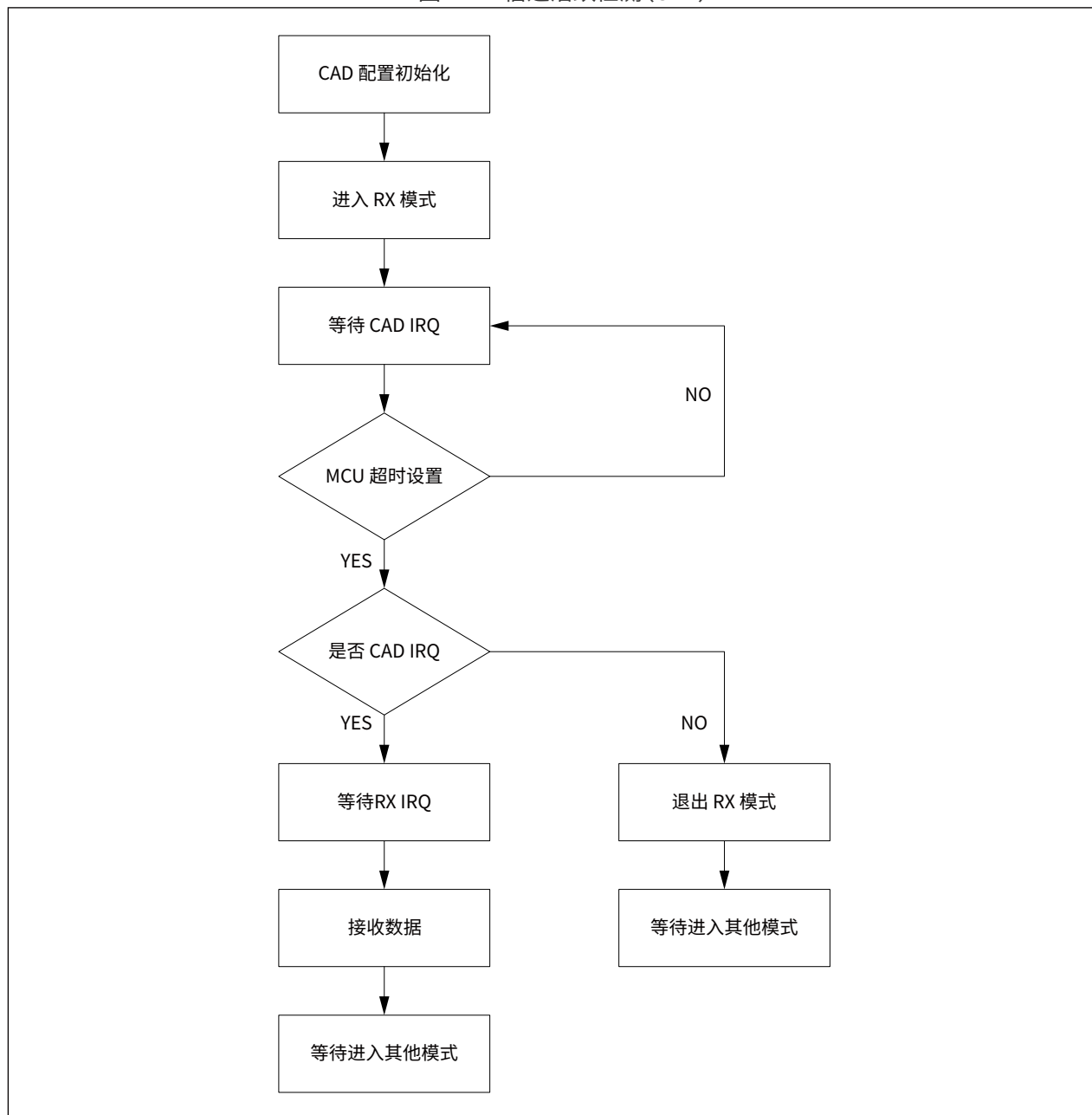
1	功能介绍 .....	3
2	软件设计参考 .....	4
2.1	软件设计流程.....	4
2.2	软件设计验证.....	4
2.2.1	验证步骤.....	4
2.2.2	SDK 示例.....	4
2.2.3	验证结果.....	5
3	注意事项 .....	6
3.1	关于 CAD 影响芯片的接收灵敏度 .....	6
3.2	关于 SDK 及演示系统板.....	6
3.3	关于 CAD 使用方法.....	7
3.3.1	对 preamble 检查方式.....	7
3.3.2	对 payload 检查方式.....	8
3.3.3	软件应用参考.....	9
3.3.4	两个 CAD 典型应用的软件设计参考 .....	11
4	版本信息 .....	16

## 1 功能介绍

CW32W031 的射频部分支持 CAD 中断。从 DeepSleep 进入 STB3，开启 CAD 功能并进入 RX 模式后，CW32W031 会检测信道中是否有 ChirpIoT™ 信号，如果存在将 CAD-IRQ 置高，MCU 内核可以通过一定的时间来检测 CAD-IRQ 信号是否拉高来判断信道中是否存在 ChirpIoT™ 信号。

用户可通过 GPIO11 端口检测 CAD-IRQ 信号，信号检测流程如下：

图 1-1 信道活跃检测 (CAD)



注：

CW32W031 的 RF 部分有多种中断源，MCU 超时设置是在等待中断的产生，然后判断中断是否为 CAD 中断，从而执行不同的命令。

## 2 软件设计参考

### 2.1 软件设计流程

1. 芯片初始化;
2. 配置 CAD 初始化;
3. 芯片进入接收模式;
4. 观察 CAD-IRQ 信号。

### 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数据包;
2. 接收模组配置为接收模式;
3. 使用逻辑分析仪抓取接收端 CAD-IRQ 信号。

#### 2.2.2 SDK 示例

参考代码

```
ret = rf_init();

if(ret != OK)
{
    dis_err(" RF Init Fail");
    while(1);
}
rf_set_default_para();
rf_set_cad(CAD_DETECT_THRESHOLD_10);
rf_enter_continuous_rx();
while (1)// 等待逻辑分析仪检测 CAD-IRQ 信号
{
    rf_irq_process();
}
```

示例代码配置了 CAD 初始化，配置 GPIO11 作为 CAD 检测 IO 口，随后进入接收模式。

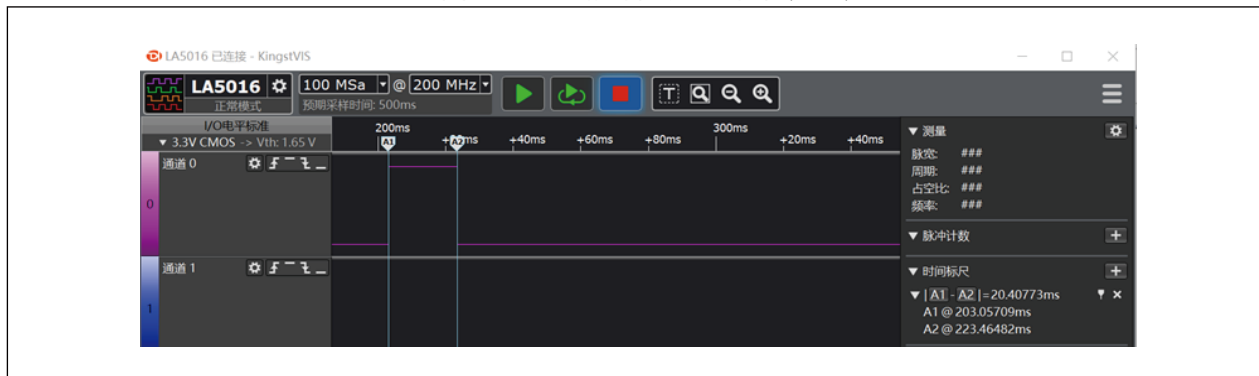
发送模组周期性发送数据包（数据包 preamble+payload 的持续时间约 20.5ms），用逻辑分析仪抓取接收模组 GPIO11 波形，观察检测结果。



### 2.2.3 验证结果

逻辑分析仪抓取结果如下图所示：

图 2-1 逻辑分析仪抓取结果 (CAD)



根据结果显示，当发送模组发送数据包时，接收模组发生了 CAD-IRQ，CAD 检测引脚 GPIO11 被拉高约 20.5ms，维持一个完整 ChirpIoT™数据包的时间长度。

### 3 注意事项

#### 3.1 关于 CAD 影响芯片的接收灵敏度

CAD 功能初始化时，修改了芯片的接收阈值，设置不同的接收阈值，会影响芯片的接收灵敏度，并可能存在 CAD 误触发的情况。

```
uint32_t PAN3028_cad_en(uint8_t threshold)
{
    PAN3028_set_gpio_output(MODULE_GPIO_CAD_IRQ);
    if(PAN3028_write_spec_page_reg(PAGE1_SEL,0x0f,threshold)!=OK)
    {
        return FAIL;
    }
    return OK;
}
```

接收阈值的设置，需修改 PAN3028\_cad\_en() 函数中的寄存器配置，默认值为 0x10，修改接收阈值对接收灵敏度及误触发概率的影响如下（实验数据在屏蔽放环境下测试）：

接收阈值	误触发概率	接收灵敏度
0x0a	极高（每分钟数次）	不影响
0x10	较高（数分钟一次）	下降 1dB
0x15	较低（数小时一次）	下降 3dB
0x20	极低（24 小时 0 次）	下降 5dB

用户在使用 CAD 功能时，需根据应用场景选择修改 PAN3028\_cad\_en() 函数中的寄存器值（PAGE1\_SEL,0x0f, 默认值为 0x10），在使用完 CAD 功能后，建议调用 rf\_set\_cad\_off() 函数，rf\_set\_cad\_off() 函数可以关闭 CAD 功能并将接收阈值恢复。

#### 3.2 关于 SDK 及演示系统板

SDK 中提供了 CAD 功能所需的函数接口，CAD-IRQ 被触发时，检测引脚 GPIO11 会被拉高。演示系统板将 GPIO11 连接到了 PB07, 具体信息可前往官网查看 CW32W031 的开发板原理图。



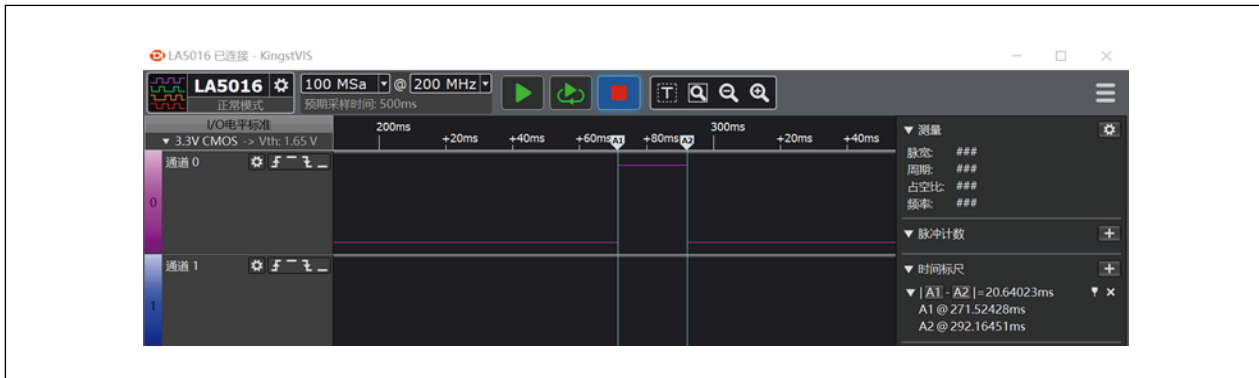
### 3.3 关于 CAD 使用方法

CW32W031 的射频部分可以对 preamble 和 payload 进行 CAD 检测。

#### 3.3.1 对 preamble 检查方式

当完整的 preamble+payload 信号到来时，用户可以在接收端通过 GPIO 口读取到 CAD-IRQ 信号，CAD 检测引脚 GPIO11 被拉高，拉高时间为 preamble+payload 的持续时间。此时，接收端可以产生正确的 rxdone 结果。

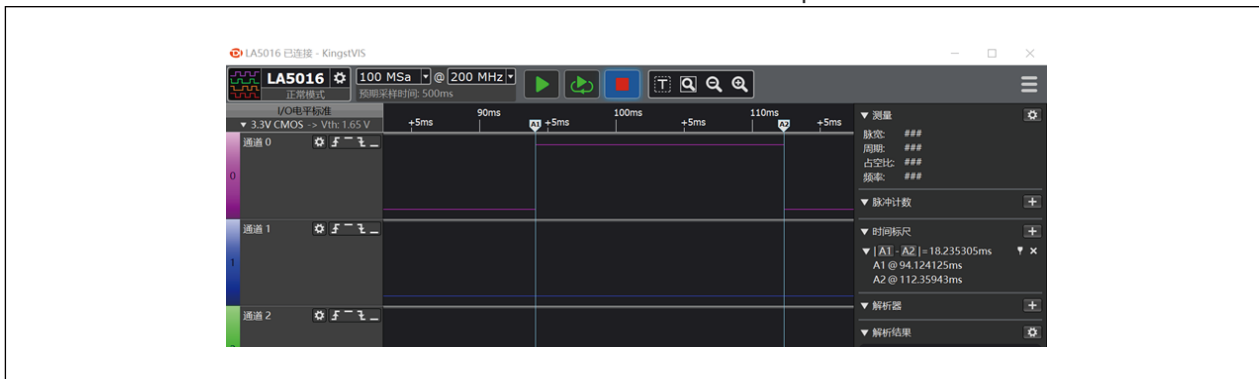
图 3-1 逻辑分析仪抓取结果 (完整 CAD)



当不完整的 preamble+payload 信号到来时（发射端先进行数据发送，随后接收端在 preamble 时间段内打开了 CAD 检测），此时，用户可以在接收端通过 GPIO 口读取到 CAD-IRQ 信号，CAD 检测引脚 GPIO11 的变化有两种情况：

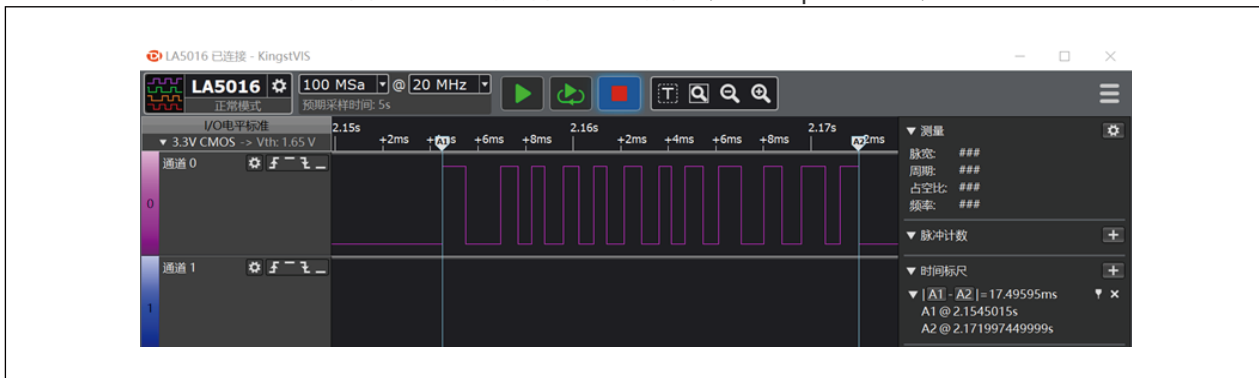
1. preamble（部分）内含有较完整信息，GPIO11 会被拉高 preamble（部分）+payload 的持续时间。此时，接收端可以产生正确的 rxdone 结果。

图 3-2 逻辑分析仪抓取结果 (较完整 preamble)



2. preamble（部分）内未含有完整信息，GPIO11 会呈现不规则高 - 低 - 高 - 低变换的现象。此时，接收端不会产生正确的接收结果。

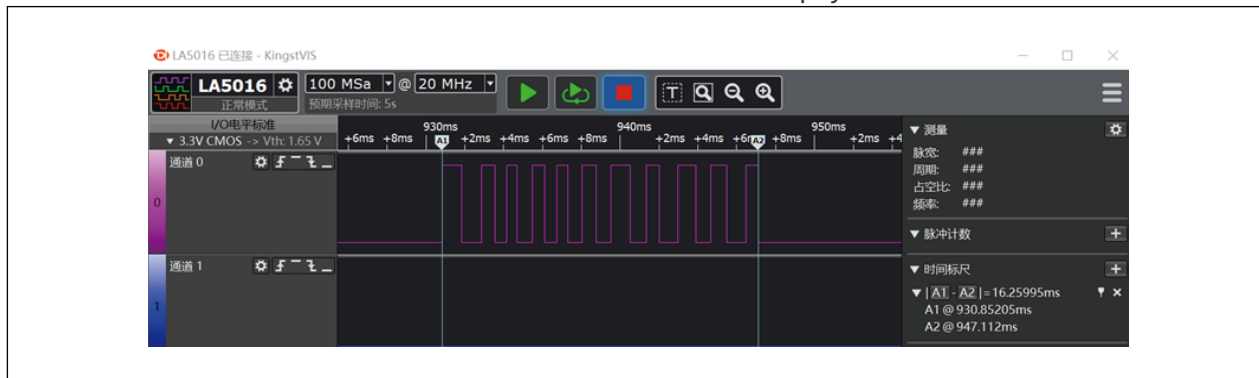
图 3-3 逻辑分析仪抓取结果 (不完整 preamble)



### 3.3.2 对 payload 检查方式

当只有 payload 信号到来时（发射端先进行数据发射，随后接收端在 payload 时间段内打开了 CAD 检测），此时，由于 RX-CAD 检测不到 preamble，CAD-IRQ 会呈现不规则高 - 低 - 高 - 低变换的现象。此时，接收端不会产生正确的接收结果。

图 3-4 逻辑分析仪抓取结果（只含 payload）





### 3.3.3 软件应用参考

当设置的 CAD 误触发概率较少，或存在少量误触发率但不影响软件应用时，可以通过检测 CAD-IRQ，GPIO11 上升沿来判断 CAD 触发，可以认为只要收到上升沿就是收到了 payload 信号（建议使用这种方法的接收阈值设为 0x15~0x20）。

当设置的 CAD 误触发概率较高时，需要结合 CAD 检测时间用软件方法来判断 payload 信号，CAD 检测的时间以单个 chirp 持续时间作为单位计算。单个 chirp 持续时间为  $2^{SF}/BW$ （SF 为扩频因子，BW 为带宽，BW 单位为 Hz，时间单位为秒）。同时，软件设计需要参考 [3.3.1 对 preamble 检查方式](#)和 [3.3.2 对 payload 检查方式](#)，进行灵活调整。

在打开 CAD 接收后，如果当前空中存在信号，那么芯片需要至少 2 个 chirp+360 $\mu$ s 持续时间的检测，GPIO11 才能首次对外输出高电平。一个典型的应用方法为：

1. 计算  $one\_chirp\_time = 2^{SF}/BW$ ;
2. 配置需要检测的信道，`rf_set_cad()`，`enter_rx`;
3. 检测 `check_cad_inactive()`，判断是否检测到 CAD 信号（下面的例程检测了三次，适用于误触发较多的情况）。

```
uint32_t check_cad_inactive(void)
{
    delay10us(one_chirp_time*2/10);
    delay10us(36);
    if(GPIO_ReadPin(CW_GPIOB, GPIO_PIN_7) != 1)
    {
        return LEVEL_INACTIVE;// 没有 cad
    }
    delay10us(one_chirp_time/10);
    if(GPIO_ReadPin(CW_GPIOB, GPIO_PIN_7) != 1)
    {
        return LEVEL_INACTIVE;
    }
    delay10us(one_chirp_time/10);
    if(GPIO_ReadPin(CW_GPIOB, GPIO_PIN_7) != 1)
    {
        return LEVEL_INACTIVE;
    }
    return LEVEL_ACTIVE;// 有 cad
}
```

如果误触发较少，或存在少量误触发但不影响软件应用时，可以将检测次数减少为一次（例程如下），或者通过 GPIO11 上升沿作为判断依据。



```
uint32_t check_cad_inactive(void)
{
    delay10us(one_chirp_time*2/10);
    delay10us(36);
    If(GPIO_ReadPin(CW_GPIOB, GPIO_PIN_7) != 1)
    {
        return LEVEL_INACTIVE;// 没有 cad
    }
    return LEVEL_ACTIVE;// 有 cad
}
```

### 3.3.4 两个 CAD 典型应用的软件设计参考

以下函数为计算 CW32W031 不同 BW、SF 配置下 chirp 的时长，为后面两个例程都需用到的函数：

```

/*return us*/
uint32_t get_chirp_time(uint32_t bw,uint32_t sf)
{
    switch(bw)
    {
        case 6:
            bw = 62500;
            break;
        case 7:
            bw = 125000;
            break;
        case 8:
            bw = 250000;
            break;
        case 9:
            bw = 500000;
            break;
        default:
            return FAIL;
    }
    return (1000000/bw)*(1<<sf);
}

```

1. CAD 功能被用于发送前的信号检测，以保证当前信道空闲，随后进行数据发射，避免无线信号碰撞干扰，提高通信成功率。

CAD IO 中断处理函数，将 CAD 事件置位为 ACTIVE.

```

void Cad_Irq_Callback(void)
{
    cad_tx_timeout_flag = MAC_EVT_TX_CAD_ACTIVE;
}

```

CAD 定时器超时回调函数，将 CAD 事件置位为 TIMEOUT。

```

void cad_tx_timeout_cb(void)
{
    stimer_stop(&stimer_txcad_event);
    cad_tx_timeout_flag |= MAC_EVT_TX_CAD_TIMEOUT;
}

```



CAD 检测函数，配置进入 CAD 接收状态，将 CAD 事件置位为 NONE, 并打开定时器。

```
uint32_t check_cad_tx_inactive(void)
{
    uint32_t bw,sf;
    rf_get_para(RF_PARA_TYPE_BW, &bw);
    rf_get_para(RF_PARA_TYPE_SF, &sf);
    uint32_t one_chirp_time = get_chirp_time(bw,sf);
    if(rf_set_cad(CAD_DETECT_THRESHOLD_10) != OK)
    {
        return FAIL;
    }
    cad_tx_timeout_flag = MAC_EVT_TX_CAD_NONE;
    if(rf_enter_continuous_rx() != OK)
    {
        return FAIL;
    }
    SET_TIMER_MS(one_chirp_time*3/1000 + 1);
    // 对于毫秒延时函数，需避免出现 0ms，可以在整除的基础上增加 1ms
    return OK;
}
```

等待并检测 CAD 事件即可。如果为 TIMEOUT 则可以立即发射，如果为 ACTIVE 则需要退避并等待一段时间再发射，再次发射之前需要再按上述步骤检测一次信道状态。

2. CAD 功能被用于接收前的信道检测，用来检查当前信道是否存在有用信号，随后决定是继续接收，或是关闭接收进入待机或休眠状态，以降低功耗。

```
uint32_t check_cad_rx_inactive(void)
{
    uint8_t i = 0;
    rf_delay_us(one_chirp_time);
    // 首次检测需要额外延时 360us，这是 RF 启动 RX 所需要的耗时
    rf_delay_us(360);
    for (i = 0; i < 3; i++)
    {
        // 首次检测延时 one_chirp_time*2，之后循环都延时 one_chirp_time
        rf_delay_us(one_chirp_time);
        if(CHECK_CAD() != 1)
        {
            // 可以选择关闭 RX，以降低功耗。这里是测量（程序切换耗时）的起点
            if(rf_set_mode(PAN3028_MODE_STB3) != OK)
            {
                return FAIL;
            }
        }
    }
}
```

```

        return LEVEL_INACTIVE;
    }
}
return LEVEL_ACTIVE;
}

```

根据返回值判断 CAD 检测结果，如果为 LEVEL\_ACTIVE 则继续接收，等待接收结果；如果为 LEVEL\_INACTIVE，可以立即关闭 RX，以降低功耗；如果为 FAIL，代表 RF 芯片异常。

// 主函数可以使用如下逻辑调用 check\_cad\_rx\_inactive

```

while (1)
{
    if(check_cad_rx_inactive(one_chirp_time) == LEVEL_ACTIVE)
    {
        // 接收处理
    }
    else
    {
        rf_set_mode(PAN3028_MODE_STB3);
        rf_set_para(RF_PARA_TYPE_FREQ, freq);
        rf_enter_continuous_rx();// 这里是测量（程序切换耗时）的终点
    }
}

```

关于 check\_cad\_rx\_inactive 和 check\_cad\_tx\_inactive 的软件应用示例可以参考固件库跳频例程中的 TX DEMO 和 RX DEMO。

如果需要多频点切换接收时（参考 RX DEMO 双频点切换接收），需要根据不同的代码逻辑实测一下 RX 程序切换耗时，然后根据这个耗时计算 TX 发射端应当配置的前导码个数。

具体方法如下：

例如对于双频点切换的场景，根据上文程序中（程序切换耗时）的起点和终点测得程序切换耗时为 T1，根据 SF 和 BW 计算得 one\_chirp\_time，RF 启动 RX 时有一个固定的启动耗时约 360us。双频点场景极限情况下所需前导码示例如下：

RX (双频点扫描)	FREQ1 (未检测到，至多 3 个 chirp 时间)	切换 (程序执行时 间，实测)	FREQ2	切换	FREQ1	FREQ1 接收	
TX(FREQ1)	额外前导码					前导码 8	数据

项目	耗时 (us)	备注
FREQ1	$\text{one\_chirp\_time} * 3 + 360$	考虑误触发、干扰等情况，极限时间需要 3 个 one_chirp_time
程序切换	T1	实测
FREQ2	$\text{one\_chirp\_time} * 3 + 360$	考虑误触发、干扰等情况，极限时间需要 3 个 one_chirp_time
TX 前导码	$(\text{one\_chirp\_time} * 3 + 360) * 3 + T1 * 2$	实际配置前导码个数： $((\text{one\_chirp\_time} * 3 + 360) * 3 + T1 * 2) / \text{one\_chirp\_time} + 8$

TX DEMO 中，TX 端根据计算结果  $((\text{one\_chirp\_time} * 3 + 360) * 3 + T1 * 2) / \text{one\_chirp\_time} + 8$

配置前导码个数。对于 SF7 等高速率场景，由于单个 chirp 时间短，使用理论值的前导码长度发射容易出现收包率不足 100% 的情况，可以酌情增加一些前导码长度，以获得更稳定的收包效果。

对于 SF11 等低速率场景，由于总发包时间较长，长包易受干扰，收包效果会不理想，实际使用时建议控制包长。详情可参考下表测试结果：

SF	BW	RF 检测 延时 (μs)	程序切换 延时 (μs)	chirp 时间 (μs)	理论总 时间 (μs)	理论前 导码个 数	测试结果： 收包数 / 发包数 = 比率 %	备注
							接收阈值 0x10	
7	62500	360	2600	2048	24712	21	屏蔽房 400/500=99.2%	前导码加 到 25 后 500/500
7	125000	360	2600	1024	15496	24	屏蔽房 500/500	
7	250000	360	2600	512	10888	30	屏蔽房 1783/1791=99.5%	前导码加 到 35 后 1000/1000
7	500000	360	2600	256	8584	42	屏蔽房 9000/9000	
8	62500	360	2600	4096	43144	19	屏蔽房 500/500	
8	125000	360	2600	2048	24712	21	500/500	
8	250000	360	2600	1024	15496	24	500/500	
8	500000	360	2600	512	10888	30	500/500	
9	62500	360	2600	8192	80008	18	屏蔽房 500/500	
9	125000	360	2600	4096	43144	19	500/500	
9	250000	360	2600	2048	24712	21	500/500	
9	500000	360	2600	1024	15496	24	500/500	
10	62500	360	2600	16384	153736	18	屏蔽房 200/200	
10	125000	360	2600	8192	80008	18	屏蔽房 200/200	

SF	BW	RF 检测 延时 (μs)	程序切换 延时 (μs)	chirp 时间 (μs)	理论总 时间 (μs)	理论前 导码个 数	测试结果: 收包数 / 发包数 = 比率 %	备注
							接收阈值 0x10	
10	250000	360	2600	4096	43144	19	屏蔽房 500/500	
10	500000	360	2600	2048	24712	21	屏蔽房 500/500	
11	62500	360	2600	32768	301192	18	屏蔽房 96/100	4 个 rxerr, 修改最大包 长为 64 后 100/100
11	125000	360	2600	16384	153736	18	屏蔽房 200/200	
11	250000	360	2600	8192	80008	18	屏蔽房 500/500	
11	500000	360	2600	4096	43144	19	屏蔽房 500/500	

在 RX DEMO 中，检测到 CAD 信号后，进入等待接收结果状态。这里在等待接收结果期间，建议多读取两次 CAD 信号，确保更高的 CAD 准确性。同是，建议设置一个 RX 端最大接收超时时间，超时时间根据 TX 端最大发包时间确定，可以选择 RX DEMO 中的计时方法，也可以选用 `rf_enter_single_timeout_rx(timeout)` 来接收。

## 4 版本信息

表 4-1 文档修订信息

日期	版本	变更信息
2023-03-09	Rev 1.0	初始发布
2024-01-09	Rev 1.1	增加 3.3.4 两个 CAD 典型应用的软件设计参考

