



Panchip Microelectronics Co., Ltd.

PAN3029/3060 系列 CAD 应用参考文档

当前版本: 1.3

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	2023.08	修改 CAD 引脚
V1.2	2024.03	修改 CAD 函数接口形参
V1.3	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	2
2.2.3 验证结果	3
3 注意事项	3
3.1 关于 CAD 影响芯片的接收灵敏度	3
3.2 关于 SDK 及演示系统板	4
3.3 关于 CAD 使用方法	4
3.3.1 对 preamble 检查方式	4
3.3.2 软件应用参考	5

1 功能介绍

PAN3029/3060 系列芯片支持 CAD-IRQ 中断，开启 CAD 功能并进入 Rx 模式后，芯片会检测信道中是否存在 ChirpIOT™信号，如果存在则将 CAD-IRQ 置高，外部 MCU 可以通过在一定时间内检测 CAD-IRQ 信号是否拉高来判断信道中是否存在 ChirpIOT™信号。

用户可以通过 GPIO 口读取 CAD-IRQ 信号，信道活跃检测流程如图 1-1 所示：

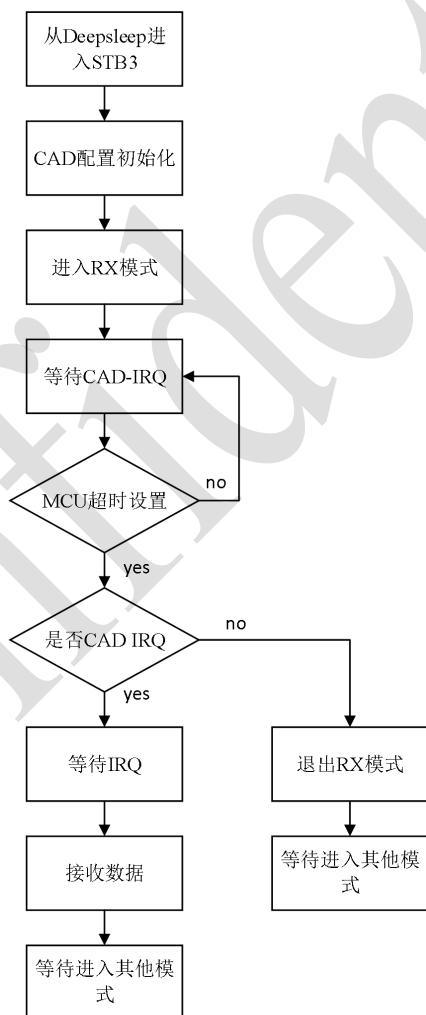


图 1-1 信道活跃检测（CAD）

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、使用示波器抓取接收端 CAD-IRQ 信号。

2.2.2 SDK 示例

参考代码：

```
ret = rf_init();
if(ret != OK)
{
    printf("  RF Init Fail");
    while(1);
}
rf_set_default_para();
rf_set_cad(CAD_DETECT_THRESHOLD_10, CAD_DETECT_NUMBER_3); // 使能
CAD 模式，并设置 Chirp-IoT 信号阈值及 preamble 有效信号个数
rf_enter_continuos_rx();
```

```
while (1); //等待示波器检测 CAD-IRQ 信号
```

示例代码配置了 CAD 初始化，配置 GPIO11 作为 CAD 检测 IO，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 42ms），用逻辑分析仪抓取接收模组 GPIO11 波形，观察逻辑分析仪结果。

2.2.3 验证结果

示波器抓取结果如图 2-1 所示：

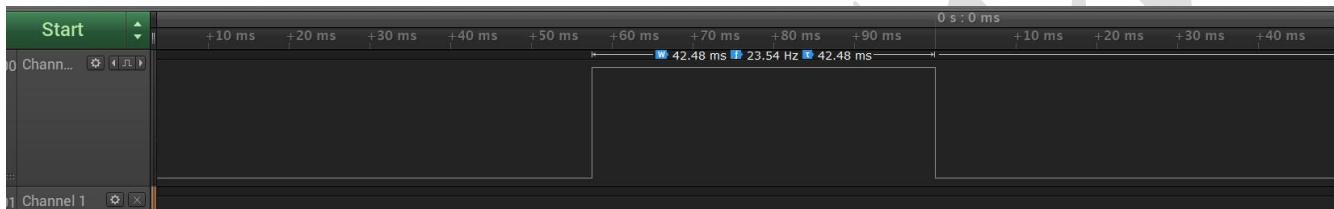


图 2-1 示波器抓取结果 (CAD)

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 42ms，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，修改了芯片的接收信号检测阈值和 CAD 有效信号个数阈值，设置不同的接收阈值和有效信号个数，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 rf_cad_on(uint8_t threshold, uint8_t chirps)
{
    rf_set_gpio_output(11);
```

```
    RF_ASSERT(rf_reset_spec_page_reg_bits(PAGE0_SEL, 0x5E, BIT6));  
    RF_ASSERT(rf_write_spec_page_reg_bits(PAGE1_SEL, 0x25, chirps, BIT0|BIT1));  
    RF_ASSERT(rf_write_spec_page_reg(PAGE1_SEL, 0x0f, threshold));  
    return OK;  
}
```

接收阈值的设置，需要修改 `rf_cad_on(uint8_t threshold, uint8_t chirps)` 函数的传参，修改接收阈值对接收灵敏度及误触发概率的影响如下（实验数据在办公室日常应用场景测试）：

表 3-1 接收阈值设置

接收阈值	有效信号个数	误触发概率	接收灵敏度
0xa	0x03	较低(数小时一次)	不影响
0x10	0x03	极低(24 小时 0 次)	恶化 1dBm

用户在使用 CAD 功能时，需要根据应用场景选择修改 `rf_cad_on(uint8_t threshold, uint8_t chirps)` 函数中的传参，在使用完 CAD 功能后，建议调用 `rf_cad_off()` 函数，`rf_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 功能所需的接口函数，CAD-IRQ 被触发时，检测引脚 GPIO11 会被拉高。GPIO11 接入演示系统板 MCU 的 PA09。

```
#define GPIO_PIN_CAD Pin09  
#define GPIO_PORT_CAD PortA  
#define CHECK_CAD() PORT_GetBit(GPIO_PORT_CAD, GPIO_PIN_CAD);
```

3.3 关于 CAD 使用方法

PAN3029/3060 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。

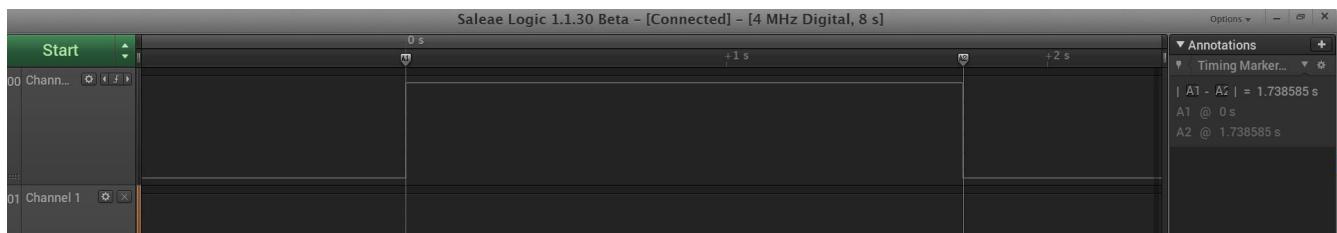


图 3-1 逻辑抓取结果（完整 preamble）

当不完整的 preamble+payload 信号到来时（发射端先进行数据发射，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、如果 preamble 比较完整，内部含 preamble 数量大于配置的 chirp_num 时，GPIO11 会被拉高 preamble+payload 的持续时间（preamble 不完整，因此 GPIO11 整体拉高时间短于图 3-1 的情况）。此时，接收端可以产生正确的 rxdone 结果。

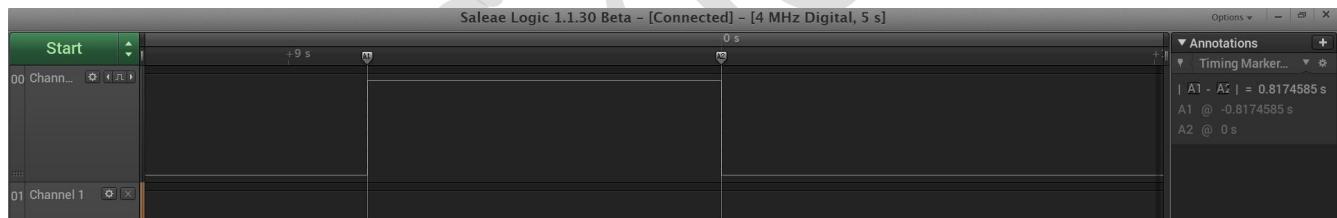


图 3-2 逻辑抓取结果（较完整 preamble）

2、如果 preamble 不完整，内部含 preamble 数量小于配置的 chirp_num 时，GPIO11 会一直处于低电平。此时，接收端不会产生正确的接收结果。

3、只有 payload 信号时，GPIO11 会一直处于低电平。此时，接收端不会产生正确的接收结果。

3.3.2 软件应用参考

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 $2^{\text{SF}}/\text{BW}$ (SF

为扩频因子，BW 为带宽，BW 单位为 Hz，时间单位为秒）。

下面给出两个 CAD 典型应用的软件设计参考：

```
uint32_t rf_get_chirp_time(uint8_t bw, uint8_t sf)

{
    const uint32_t bw_table[4] = {62500, 125000, 250000, 500000};

    if(bw < BW_62_5K || bw > BW_500K)
    {
        return 0;
    }

    return (1000000 / bw_table[bw - BW_62_5K]) * (1 << sf);
}
```

1、CAD 功能被用于发射前的信道检测，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)

{
    rf_set_cad_detect_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);
    rf_set_cad_detect_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 rf_cad_detect_start(void)
{
    uint32_t bw, sf;
    uint32_t one_chirp_time;
    bw = rf_get_bw();
    sf = rf_get_sf();
    one_chirp_time = rf_get_chirp_time(bw, sf); // us
    RF_ASSERT(rf_cad_on(CAD_DETECT_THRESHOLD_10,
CAD_DETECT_NUMBER_3));
    cad_tx_detect_flag = MAC_EVT_TX_CAD_NONE;
    RF_ASSERT(rf_enter_continuos_rx())
    SET_TIMER_MS(one_chirp_time * 7 / 1000 + 1);
    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，继续接收，或是关闭接收，进入待机或休眠状态，以降低功耗。

```
bool check_cad_rx_inactive(uint32_t one_chirp_time)
{
    rf_delay_us(one_chirp_time * 7);
    rf_delay_us(360); // state machine start up time after enter rx state

    if (CHECK_CAD() != 1)
    {
        rf_set_mode(RF_MODE_STB3);
        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。

```
//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)  
        {//检测到信号，等待接收完成  
    }else{  
        //未检测到信号，不需要继续接收，可进行休眠处理  
    }  
}
```

关于 check_cad_rx_inactive 和 rf_cad_detect_start 的软件应用示例可以参考 CAD 例程。