



**武汉芯源半导体有限公司**  
WUHAN XINYUAN SEMICONDUCTOR CO., LTD

# CW32W031 CAD Application Reference

Application note

Rev 1.0



# Contents

|       |  |    |
|-------|--|----|
| 1     | Introduction.....  | 3  |
| 2     | Software design reference .....                                  | 4  |
| 2.1   | Software design process.....                                     | 4  |
| 2.2   | Software design verification .....                               | 4  |
| 2.2.1 | Verification step .....  | 4  |
| 2.2.2 | SDK Sample .....   | 4  |
| 2.2.3 | Validation results .....   | 5  |
| 3     | Notice .....   | 6  |
| 3.1   | Effect of CAD detection thresholds on reception sensitivity..... | 6  |
| 3.2   | About the SDK and Demo Board .....                               | 6  |
| 3.3   | About CAD Usage .....  | 7  |
| 3.3.1 | Preamble detection method.....                                   | 7  |
| 3.3.2 | Payload detection method.....                                    | 8  |
| 3.3.3 | Software application reference .....                             | 9  |
| 4     | Revision history .....   | 11 |

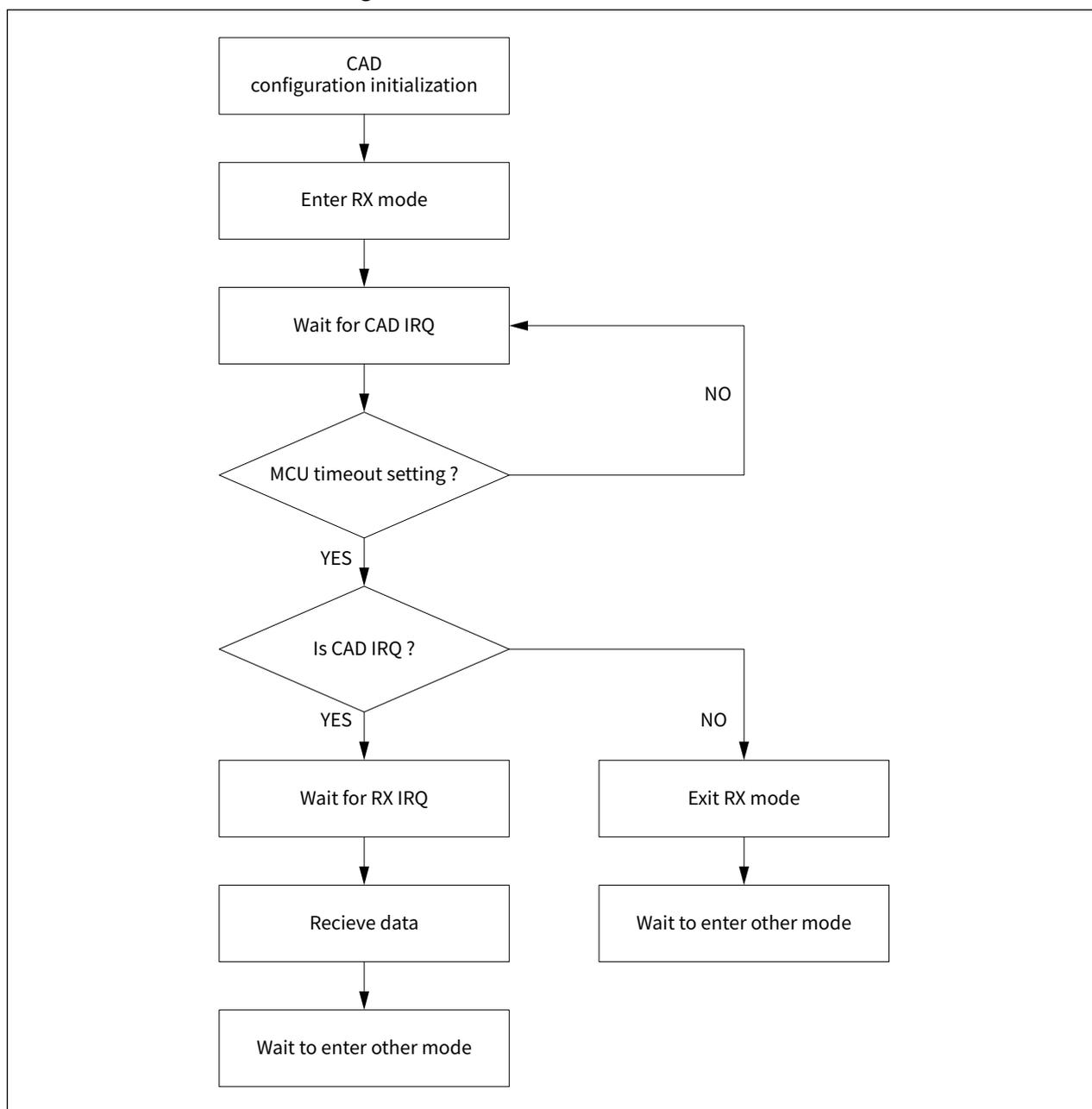


# 1 Introduction

The RF section of the CW32W031 supports CAD interrupt. After entering STB3 from DeepSleep, enabling the CAD function and entering RX mode, the CW32W031 will detect if there is a ChirpIOT™ signal in the channel, and if so, set the CAD-IRQ high, the MCU kernel can detect if the CAD-IRQ signal is pulled high for a certain period of time to determine if there is a ChirpIOT™ signal in the channel.

The user can detect the CAD-IRQ signal via the GPIO11 port, the signal detection process is as follows:

Figure 1-1 Channel Active Detection (CAD)



**Caution:**

*The RF section of the CW32W031 has multiple interrupt sources, the MCU timeout is set to wait for an interrupt to be generated and then determine if the interrupt is a CAD interrupt, thus executing different commands.*

## 2 Software design reference

### 2.1 Software design process

1. Chip initialisation;
2. CAD configuration initialisation;
3. The chip enters RX mode;
4. Observe the CAD-IRQ signal.

### 2.2 Software design verification

#### 2.2.1 Verification step

1. The sending module periodically sends data packets;
2. The receiving module is configured in a RX mode;
3. Use the logic analyzer to capture the CAD-IRQ signal on the receive side.

#### 2.2.2 SDK Sample

Reference Code:

```
ret = rf_init();

if(ret != OK)
{
    dis_err(" RF Init Fail");
    while(1);
}
rf_set_default_para();
rf_set_cad();
rf_enter_continuous_rx();
while (1)//Wait for the logic analyzer to detect the CAD-IRQ signal
{
    rf_irq_process();
}
```

The sample code configures the CAD initialization, configures GPIO11 as the CAD detection IO port, and then enters the receive mode.

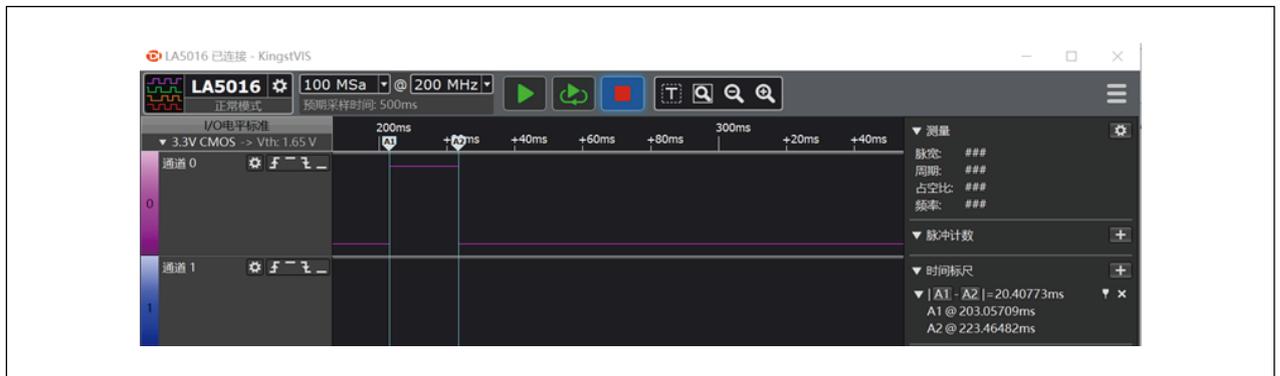
The sending module sends data packets periodically (the duration of the preamble+payload of the data packet is about 20.5ms). Use the logic analyzer to capture the GPIO11 waveform of the receiving module and observe the detection results.



### 2.2.3 Validation results

The capture result of logic analyzer is shown in the following figure:

Figure 2-1 Logic analyzer capture result



According to the results, when the sending module sends a data packet, the receiving module generates a CAD-IRQ, and the CAD detection pin GPIO11 is pulled high for about 20.5ms, maintaining the time length of a complete ChirpIOT™ data packet.

## 3 Notice

### 3.1 Effect of CAD detection thresholds on reception sensitivity

When the CAD function is initialised, the chip's CAD signal detection threshold is modified. Setting different detection thresholds will affect the chip's reception sensitivity and may result in CAD false triggering.

```
uint32_t PAN3028_cad_en(void)
{
    PAN3028_set_gpio_output(MODULE_GPIO_CAD_IRQ);
    If(PAN3028_write_spec_page_reg(PAGE1_SEL,0x0f,0x10)!=OK)
    {
        return FAIL;
    }
    return OK;
}
```

To set the detection threshold, modify the register configuration in the PAN3028\_cad\_en() function, the default value is 0x10. The effect of different detection thresholds on the reception sensitivity and false trigger probability is as follows (experimental data tested in a shielded discharge environment).

| Detection threshold | False trigger probability            | Reception sensitivity |
|---------------------|--------------------------------------|-----------------------|
| 0x0A                | Very high (several times per minute) | No effect             |
| 0x10                | Higher (once every few minutes)      | Decrease by 1dB       |
| 0x15                | Lower (once every few hours)         | Decrease by 3dB       |
| 0x20                | Very low (0 times in 24 hours)       | Decrease by 5dB       |

When using the CAD function, the user needs to set the register value(PAGE1\_SEL,0x0f, default value is 0x10) in the PAN3028\_cad\_en() function according to the application scenario. After using the CAD function, it is recommended to call the rf\_set\_cad\_off() function to turn off the CAD function and restore the detection threshold.

### 3.2 About the SDK and Demo Board

The SDK provides the interface to the functions required for the CAD function. When the CAD-IRQ is triggered, the detection pin GPIO11 is pulled high. The demo board connects GPIO11 to the PB07, for more information go to the CW32W031 development board schematic.



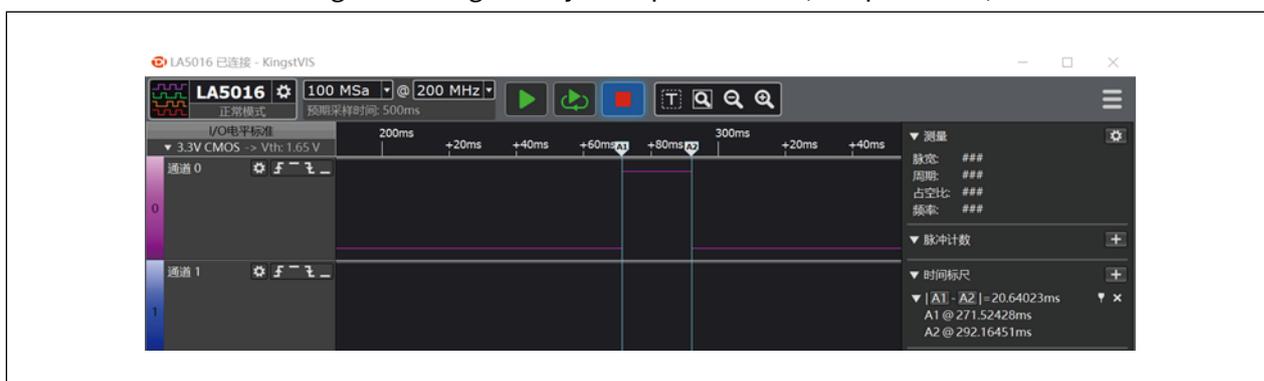
### 3.3 About CAD Usage

The RF section of the CW32W031 provides CAD detection for both preamble and payload.

#### 3.3.1 Preamble detection method

When the full preamble+payload signal arrives, the user can read the CAD-IRQ signal on the receiver side via the GPIO port and the CAD detection pin GPIO11 is pulled high for the duration of the preamble+payload. At this point, the correct rxdone result is generated at the receiver.

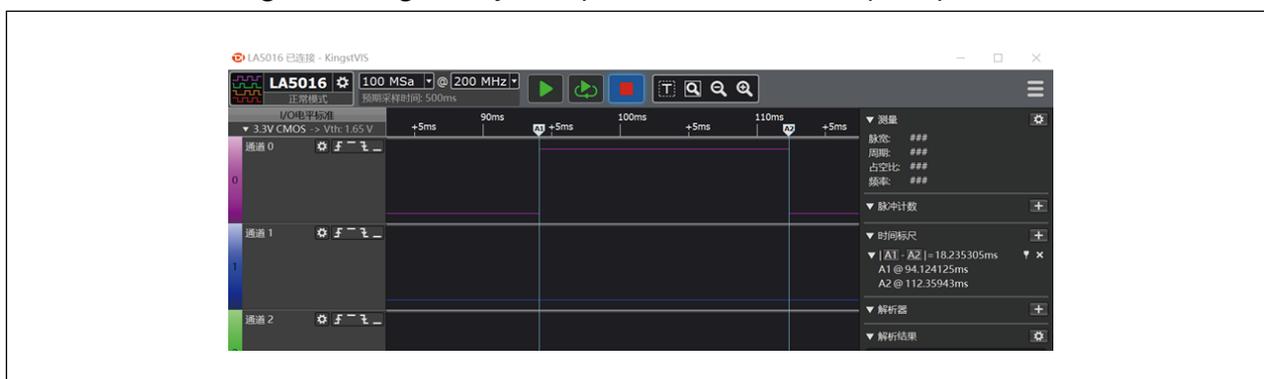
Figure 3-1 Logic analyser capture result (complete CAD)



When an incomplete preamble+payload signal arrives (data transmission at the transmitter first, followed by CAD detection turned on at the receiver during the preamble time period), the user can then read the CAD-IRQ signal at the receiver via the GPIO port, and the CAD detection pin GPIO11 changes in two ways:

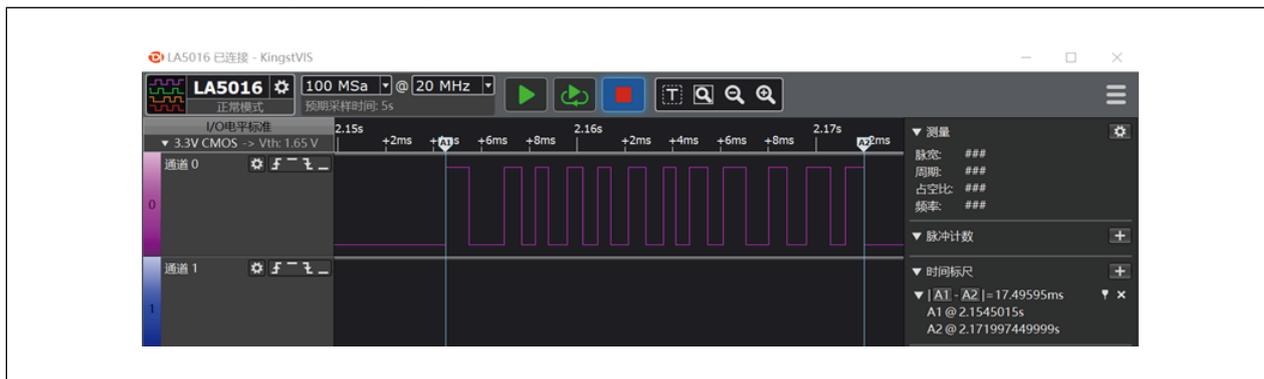
1. The preamble (part) contains more complete information and GPIO11 will be pulled up for the duration of preamble (part) + payload. At this point the receiver can produce the correct rx done result.

Figure 3-2 Logic analyser capture result (more complete preamble)



2. The preamble does not contain the full information and GPIO11 will show an irregular high-low-high-low transition. In this case, the receiver will not produce the correct reception result.

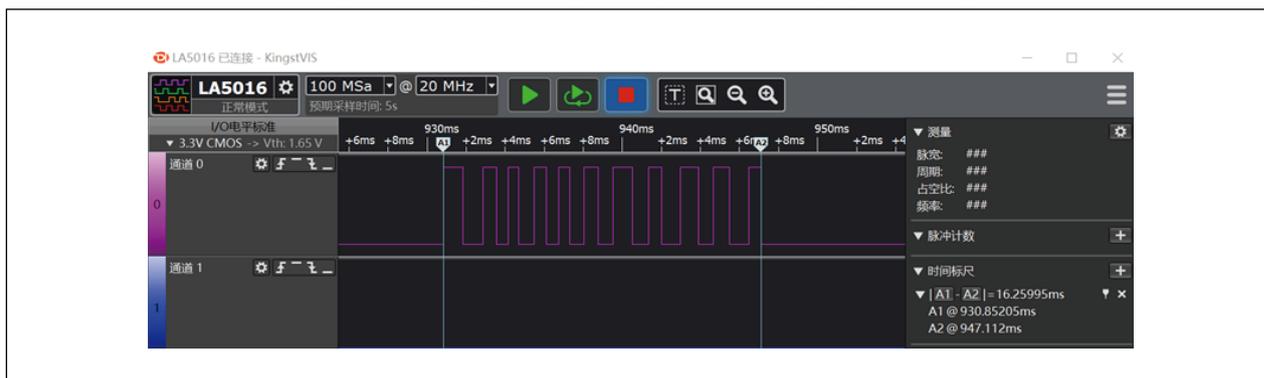
Figure 3-3 Logic analyser capture result (incomplete preamble)



### 3.3.2 Payload detection method

When only the payload signal arrives (the transmitter transmits the data first and then the receiver turns on CAD detection during the payload time period), the CAD-IRQ shows an irregular high-low-high-low transition because the RX-CAD does not detect the preamble. At this point, the receiver does not produce a correct reception result.

Figure 3-4 Logic analyser capture result (payload only)



### 3.3.3 Software application reference

When the set CAD false trigger probability is small, or there is a small false trigger rate but it does not affect the software application, the CAD trigger can be determined by detecting the CAD-IRQ, the rising edge of GPIO11. It can be assumed that the payload signal is received whenever a rising edge is received (it is recommended that the detection threshold for this method is set to 0x15~0x20).

When a high probability of false CAD triggering is set, a software method is required to determine the payload signal in conjunction with the CAD detection time, which is calculated in units of single chirp duration. The duration of a single chirp is  $2^{SF}/BW$  (SF is the spreading factor, BW is the bandwidth, BW in Hz and time in seconds). For use, it is advisable to set the detection time window to 3 chirp durations. The user can detect the CAD signal at each of the 3 chirp positions and if the CAD signal is detected at the same time (GPIO11 is high), it is assumed that a signal is present. At the same time, the software design needs to be adapted flexibly, taking into account [3.3.1 Preamble detection method](#) and [3.3.2 Payload detection method](#).

After turning on CAD reception, if there is a signal in the air, the chip needs at least 2 chirp duration detection before GPIO11 can output high level for the first time. A typical application method is:

1. Calculate  $one\_chirp\_time = 2^{SF}/BW$ ;
2. Configure the channels to be detected, `rf_set_cad()`, `enter_rx`;
3. Detects `check_cad_inactive()` to determine if a CAD signal is detected (the following routine detects it three times, for cases where there are many false triggers).

```
uint32_t check_cad_inactive(void)
{
    delay10us(one_chirp_time*2/10);
    if(GPIO_ReadPin(CW_GPIOB, GPIO_PIN_7) != 1)
    {
        return LEVEL_INACTIVE;//No CAD signal
    }
    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;//With CAD signal
}
```

If there are few false triggers, or if there are a few false triggers but they do not affect the software application, the number of detections can be reduced to one (example procedure below) or the rising edge of GPIO11 can be used as a basis for judgement.



```
uint32_t check_cad_inactive(void)
{
    delay10us(one_chirp_time*2/10);
    If(GPIO_ReadPin(CW_GPIOB, GPIO_PIN_7) != 1)
    {
        return LEVEL_INACTIVE;//No CAD signal
    }
    return LEVEL_ACTIVE;//With CAD signal
}
```



## 4 Revision history

Table 4-1 Document revision history

| Date         | Revision | Changes          |
|--------------|----------|------------------|
| May 18, 2023 | Rev 1.0  | Initial release. |
|              |          |                  |
|              |          |                  |
|              |          |                  |
|              |          |                  |
|              |          |                  |
|              |          |                  |
|              |          |                  |
|              |          |                  |

