

实例开发指南

一、概述

本文开发实例是针对初学者能快速使用 CMT2300A 实现无线收、发功能。

开发硬件平台：RF-EB 主板，STM32F103 芯片

开发软件：Keil uVision V5

二、RFPDK 软件配置

首先通过 RFPDK 软件生成配置参数，参考例程配置如下(单收、单发配置)：

The screenshot displays the RFPDK software configuration interface, which is divided into three main sections: RF Settings, Baseband 2 Settings, and Baseband 1 Settings. Each section contains various configuration parameters for the CMT2300A chip.

RF Settings:

- Frequency (126.334-1020): 433.920 MHz
- Xtal Frequency: 26.0000 MHz
- (De)Modulation: GFSK
- AGC: On
- Data Rate (0.5-300): 9.6 kbps
- Deviation (2-200): 20.0 kHz
- Tx Xtal Tol. (0-50): +/- 20 ppm
- Rx Xtal Tol. (0-50): +/- 20 ppm
- TRx Matching Network Type: 20 dBm
- Tx Power: +20 dBm
- Gaussian BT: 0.5
- PA Ramp: On

Baseband 2 Settings:

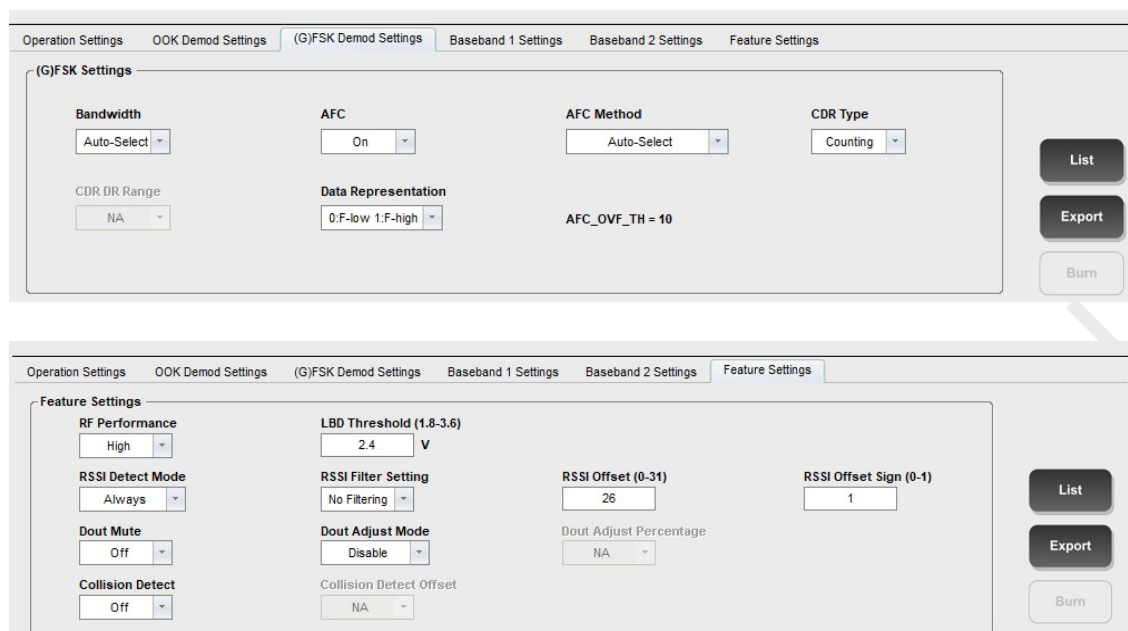
- Packet Type: Fixed Length
- Node-Length Position: NA
- Payload Bit Order: Start from msb
- Preamble Rx Size (0-31): 2
- Preamble Tx Size (0-65535): 8
- Hex: ☒ Dec: ☐
- Preamble Value (0-0xFF): AA
- Packet Structure: Preamble (0xAA), Sync (0x9375), Node ID (None), Data (32-byte), CRC (None)

Baseband 1 Settings:

- Data Mode: Packet
- Whitening: Disable
- Manchester: Disable
- Tx Prefix Type: 0
- Whiten Type: NA
- Manchester Type: NA
- Tx Packet Number (1-256): 1 packet(s)
- Whiten Seed Type: NA
- FEC: Disable
- Tx Packet Gap (1-256): 32 symbol(s)
- Whiten Seed (0-511): NA
- FEC Type: NA

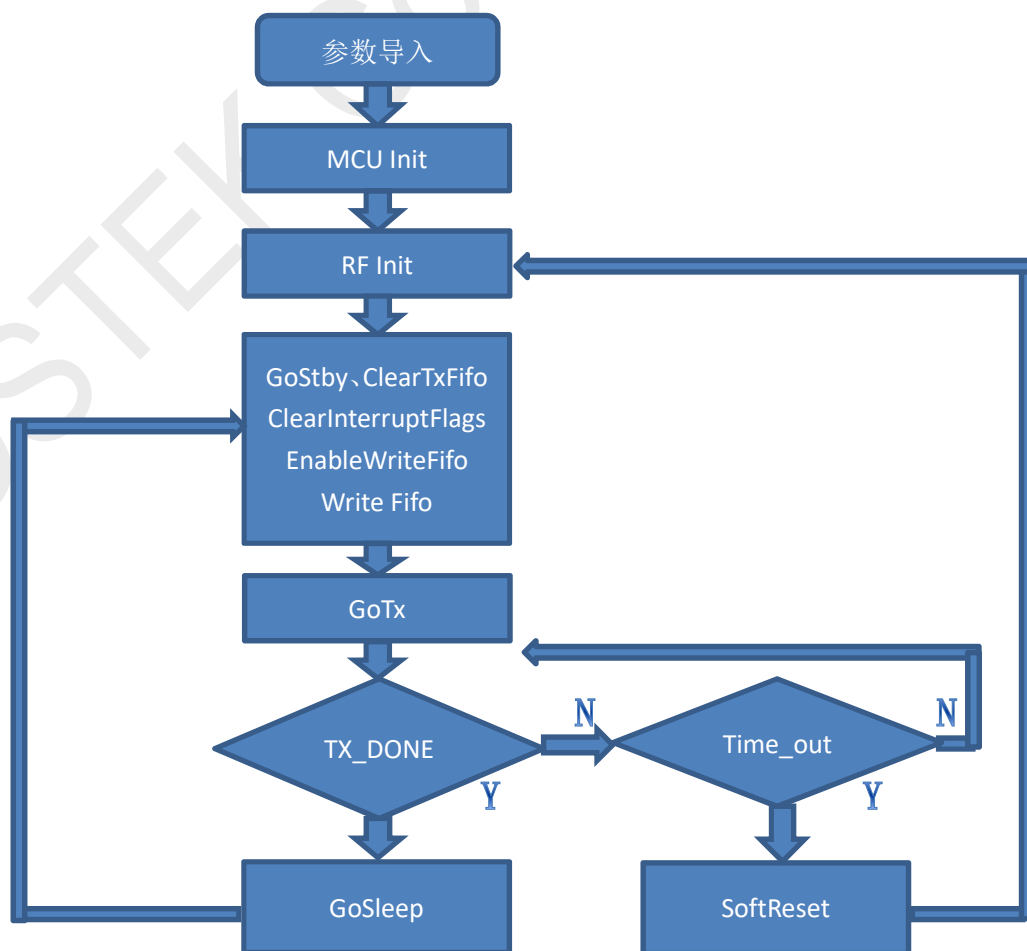
Operation Settings:

- Rx Duty-Cycle: Off
- Rx Timer: Off
- Tx Exit State: STBY
- SLP Mode: Disable
- Tx Duty-Cycle: Off
- Rx Time T1 (0.04-2683043): NA ms
- LFOSC Calibration: On
- RSSI Valid Source: PJD
- Sleep Timer: Off
- Rx Time T2 (0.04-2683043): NA ms
- XTAL Stable Time: 155 us
- PJD Window: 8 Jumps
- Sleep Time (0.03125-41922560): NA ms
- Rx Exit State: STBY
- RSSI Compare TH (-127-0): NA dBm



三、单发实例

1、流程图



2、实例

```

/*****
; Frequency          = 433.920 MHz
; Demodulation       = GFSK
; Data Rate          = 9.6 kbps
; Deviation          = 20.0 kHz
; Tx Power           = +20 dBm
; Payload Length     = 32
*****/
#define RF_PACKET_SIZE 32          /* Define the payload size here */
static u8 g_rxBuffer[RF_PACKET_SIZE]; /* RF Rx buffer */
static u8 g_txBuffer[RF_PACKET_SIZE]; /* RF Tx buffer */

void Mcu_Init(void)
{
    /* system init */
    SystemInit();
    GPIO_Config();
    NVIC_Config();
    SystemTimerDelay_Config();
    Timer5_Config();
    buzzer_init();
}

u8 Radio_Send_FixedLen(const u8 pBuf[], u8 len)
{
    u8 wrLen = 0;
    u32 delay;
    CMT2300A_GoStby();
    CMT2300A_ClearInterruptFlags();
    CMT2300A_ClearTxFifo();
    CMT2300A_EnableWriteFifo();
    CMT2300A_WriteFifo(pBuf, len); // 写 TX_FIFO
    CMT2300A_GoTx(); // 启动发送
    delay = 1000;
    while(1)
    {
        if(CMT2300A_ReadGpio1()) // TX_DONE
        {
            CMT2300A_ClearInterruptFlags();
            CMT2300A_GoSleep();
            buzzer_on(); //发送完成蜂鸣器响一次
            system_delay_ms(100);
            buzzer_off();
            return 1; //
        }
        if(delay==0) ///超时溢出，防止芯片死机，客户可根据实际开发情况调整超时时间
        {
            CMT2300A_SoftReset(); //复位芯片
            RF_Init();
            return 0; // 发送超时
        }
        system_delay_100us(2);
        delay--;
    }
}

int main(void) //单发例程
{
    int i;
    for(i=0; i<RF_PACKET_SIZE; i++)
        g_txBuffer[i] = 1+i;

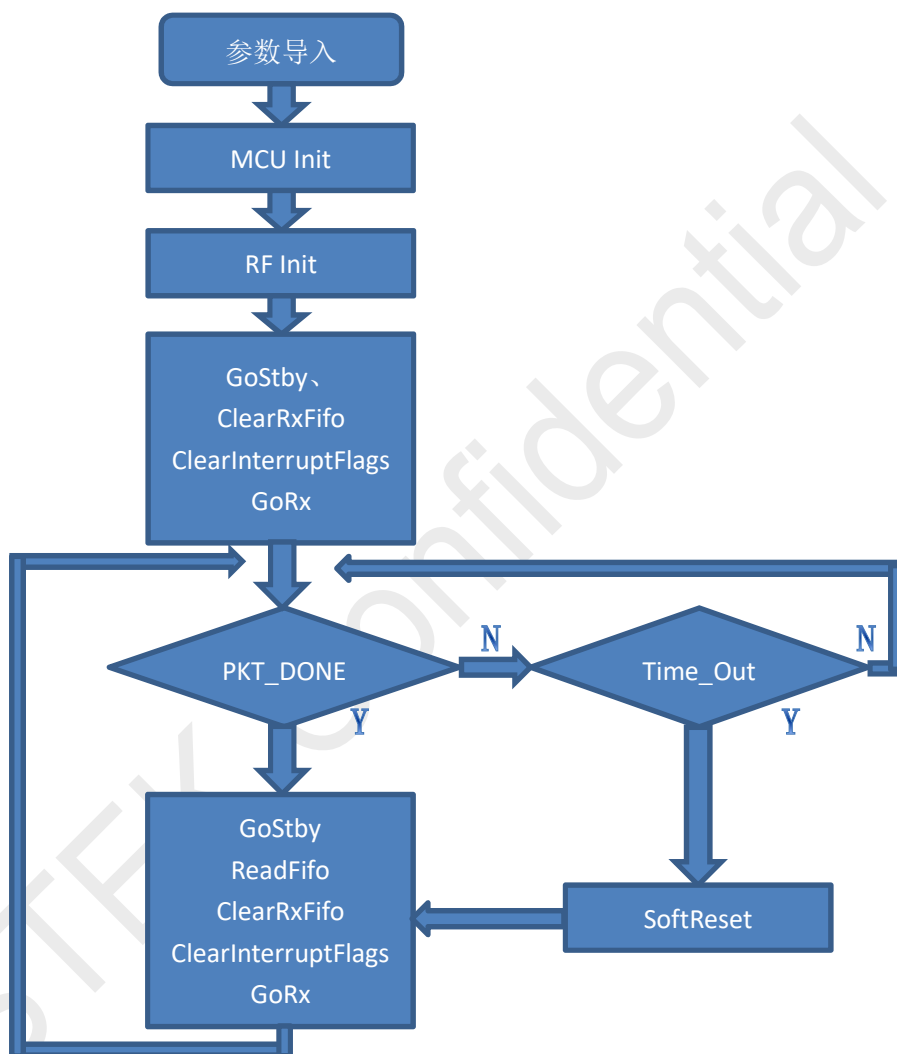
    Mcu_Init();
    RF_Init();

    while(1) //连续发送
    {
        Radio_Send_FixedLen(g_txBuffer, RF_PACKET_SIZE);
        system_delay_ms(3000); //间隔时间
    }
}

```

四、单收实例

1、流程图



2、实例

```

/*****
; Frequency                = 433.920 MHz
; Demodulation              = GFSK
; Data Rate                 = 9.6 kbps
; Deviation                 = 20.0 kHz
; Tx Power                  = +20 dBm
; Payload Length            = 32
*****/
#define RF_PACKET_SIZE 32 /* Define the payload size here */
static u8 g_rxBuffer[RF_PACKET_SIZE]; /* RF Rx buffer */
static u8 g_txBuffer[RF_PACKET_SIZE]; /* RF Tx buffer */

void Mcu_Init(void)
{
    /* system init */
    SystemInit();
    GPIO_Config();
    NVIC_Config();
    SystemTimerDelay_Config();
    Timer5_Config();
}

u8 Radio_Recv_FixedLen(u8 pBuf[], u8 len)
{
    if(CMT2300A_ReadGpio2()) // PKT_OK
    {
        CMT2300A_GoStby();
        CMT2300A_ReadFifo(pBuf, len);
        CMT2300A_ClearRxFifo();
        CMT2300A_ClearInterruptFlags();
        CMT2300A_GoRx();
        return 1;
    }

    return 0;
}

int main(void) // 单收例程
{
    u8 i=0;
    u32 Time_out=0xFFFFFFFF;
    Mcu_Init();
    RF_Init();

    CMT2300A_GoStby();
    /* Must clear FIFO after enable SPI to read or write the FIFO */
    CMT2300A_EnableReadFifo();
    CMT2300A_ClearInterruptFlags();
    CMT2300A_ClearRxFifo();
    CMT2300A_GoRx();
    while(1)
    {
        if(Radio_Recv_FixedLen(g_rxBuffer, RF_PACKET_SIZE))
        {
            for(i=0; i<RF_PACKET_SIZE; i++) // Clear Buff
                g_rxBuffer[i]=0;
        }
        Time_out--;
        if(Time_out==0) // 超时溢出，防止芯片死机，客户可根据实际开发情况调整超时时间
        {
            CMT2300A_SoftReset(); // 复位芯片
            RF_Init();
            CMT2300A_GoStby();
            /* Must clear FIFO after enable SPI to read or write the FIFO */
            CMT2300A_EnableReadFifo();
            CMT2300A_ClearInterruptFlags();
            CMT2300A_ClearRxFifo();
            CMT2300A_GoRx();
        }
        system_delay_10us(2);
    }
}

```

五、低功耗接收实例

关于 CMT2300A 低功耗使用详细说明请参考《AN146-CMT2300A 低功耗模式使用指南_CN_V0.9》文档，本文只对其中 Mode12 低功耗接收模式写 Demo Code。

编号	RX 的延长方式	RX 的延长条件
1	T1 内一旦满足检测条件，就离开 T1，将控制权交给 MCU	RSSI_VLD 有效
2		PREAM_OK 有效
3		RSSI_VLD 与 PREAM_OK 同时有效
4	T1 内只要检测到 RSSI 有效，就退出 T1 并一直处于 RX，直到 RSSI 不满足就退出 RX	RSSI_VLD 有效
5	T1 内一旦满足检测条件，就切换到 T2，T2 计时结束后就退出 RX	RSSI_VLD 有效
6		PREAM_OK 有效
7		RSSI_VLD 与 PREAM_OK 同时有效
8		PREAM_OK 或 SYNC_OK 任意一个有效
9		PREAM_OK 或 NODE_OK 任意一个有效
10		PREAM_OK 或 SYNC_OK 或 NODE_OK 任意一个有效
11	T1 内一旦满足检测条件，就切换到 T2，T2	RSSI_VLD 有效
12	内一旦检测到 SYNC 就退出 T2 并将控制权	PREAM_OK 有效
13	交给 MCU，否则 T2 计时结束后就退出 RX	RSSI_VLD 与 PREAM_OK 同时有效

假设发送端 RFPDK 设置如下：

为了保证接收稳定，发射端连发2包，包与包间隔100ms，Preamble 长度为360byte，数据区长度为20byte。（用户可以根据需求，为了接收稳定性，增加发包个数）

Chip Parameters

RF Settings

Frequency (126.334-1020) 433.920 MHz Xtal Frequency 26.0000 MHz (De)Modulation GFSK AGC On

Data Rate (0.5-300) 9.6 kbps Deviation (2-200) 20.0 kHz Tx Xtal Tol. (0-50) +/- 20 ppm Rx Xtal Tol. (0-50) +/- 20 ppm

TRx Matching Network Type 20 dBm Tx Power +20 dBm Gaussian BT 0.5 PA Ramp On

Operation Settings OOK Demod Settings (G)FSK Demod Settings Baseband 1 Settings Baseband 2 Settings Feature Settings

Baseband 2 Settings

Packet Type Fixed Length

Node-Length Position NA Length (1-2048) 20 byte(s)

Payload Bit Order Start from msb

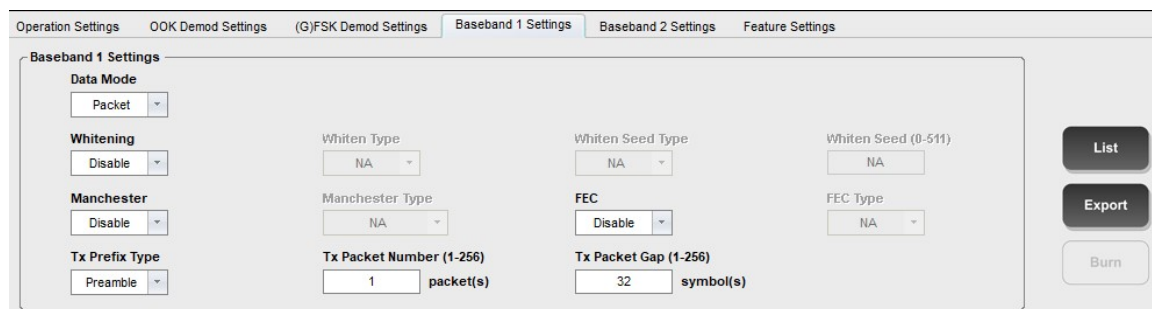
0xAA 0x9375 None 20-byte None

Preamble Sync Node ID Data CRC

Preamble Rx Size (0-31) 2 Preamble Tx Size (0-65535) 360 Hex Preamble Value (0-0xFF) AA

Payload Bit Order Start from msb Preamble Unit 8-bit

List Export Burn



Baseband 1 Settings

Data Mode: Packet

Whitening: Disable

Manchester: Disable

Tx Prefix Type: Preamble

Whiten Type: NA

Manchester Type: NA

Tx Packet Number (1-256): 1 packet(s)

Whiten Seed Type: NA

FEC: Disable

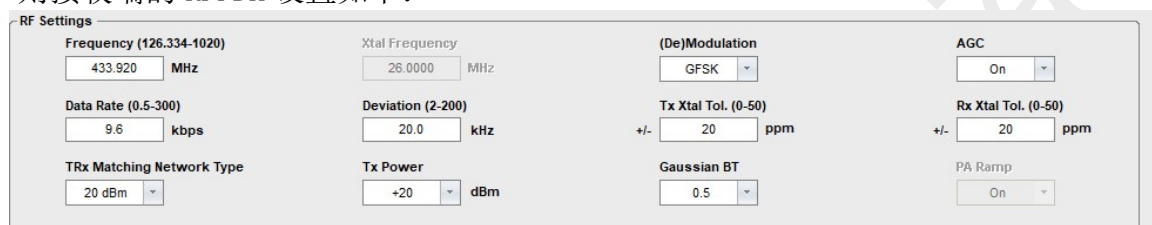
Tx Packet Gap (1-256): 32 symbol(s)

Whiten Seed (0-511): NA

FEC Type: NA

Buttons: List, Export, Burn

则接收端的 RFPDK 设置如下:



RF Settings

Frequency (126.334-1020): 433.920 MHz

Xtal Frequency: 26.0000 MHz

(De)Modulation: GFSK

AGC: On

Data Rate (0.5-300): 9.6 kbps

Deviation (2-200): 20.0 kHz

Tx Xtal Tol. (0-50): +/- 20 ppm

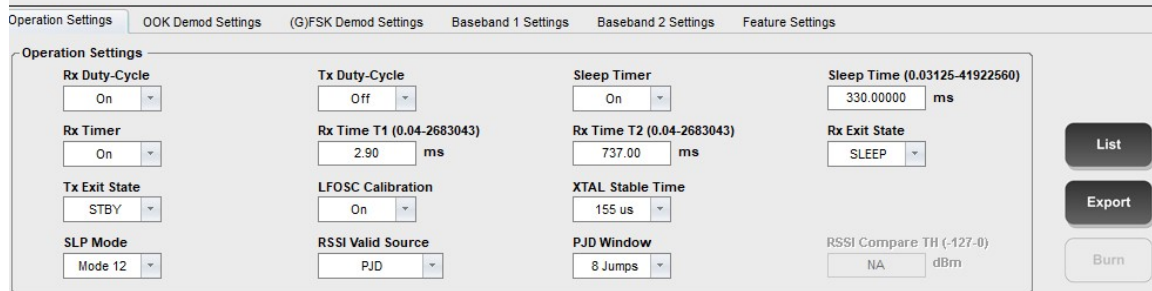
Rx Xtal Tol. (0-50): +/- 20 ppm

TRx Matching Network Type: 20 dBm

Tx Power: +20 dBm

Gaussian BT: 0.5

PA Ramp: On



Operation Settings

Rx Duty-Cycle: On

Tx Duty-Cycle: Off

Sleep Timer: On

Sleep Time (0.03125-41922560): 330.00000 ms

Rx Timer: On

Rx Time T1 (0.04-2683043): 2.90 ms

Rx Time T2 (0.04-2683043): 737.00 ms

Rx Exit State: SLEEP

Tx Exit State: STBY

LFOSC Calibration: On

XTAL Stable Time: 15S us

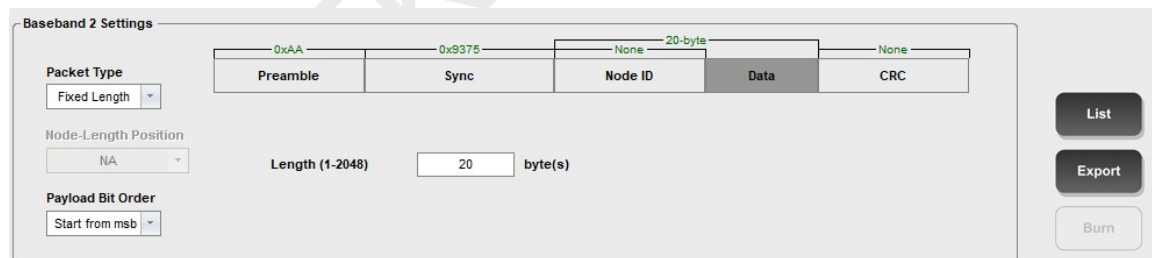
SLP Mode: Mode 12

RSSI Valid Source: PJD

PJD Window: 8 Jumps

RSSI Compare TH (-127-0): NA dBm

Buttons: List, Export, Burn



Baseband 2 Settings

Packet Type: Fixed Length

Node-Length Position: NA

Payload Bit Order: Start from msb

Length (1-2048): 20 byte(s)

Diagram: 0xAA Preamble, 0x9375 Sync, None Node ID, 20-byte Data, None CRC

Buttons: List, Export, Burn

T1 设置方法如下:

预留 8 个 symbol 给接收机做 AFC, 将 PJD 的跳变数设为 8, 再多预留 2 个 symbol, 在接收 preamble 时, 跳变数才等同于 symbol 数, 如下图 14, 采用 Mode12, T1 的时间一共为 18 个 symbol, GoSleep 到进入 GoRX 工作要 1ms 左右, 通信速率为 9.6Kbps, $T1 = (1/9600) * 18 + 1 = 2.87\text{ms}$, 取 2.9ms

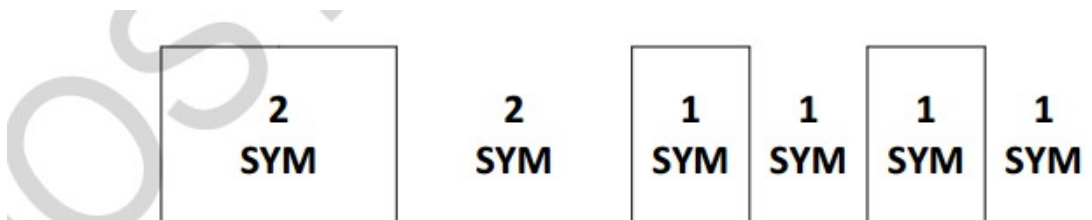


图 14.接收信号跳变图

T2 设置方法如下：

当模式设置成 11-13 的时候，RX T1 内一旦满足检测条件，就切换到 RX T2，RX T2 内一旦检测到 SYNC_OK，RX T2 就停止计时，芯片停留在 RX，为了确保数据接收稳定性，T2 设置为两包数据总时间（包括两包数据之间的间歇时间），一包数据时间为 $(1/9600) * 8 * 382 = 318.3\text{ms}$ ；

$T2 = 318.3 * 2 + 100 = 736.6\text{ms}$ ，T2 取 737ms；

Sleep Time 设置方法如下：

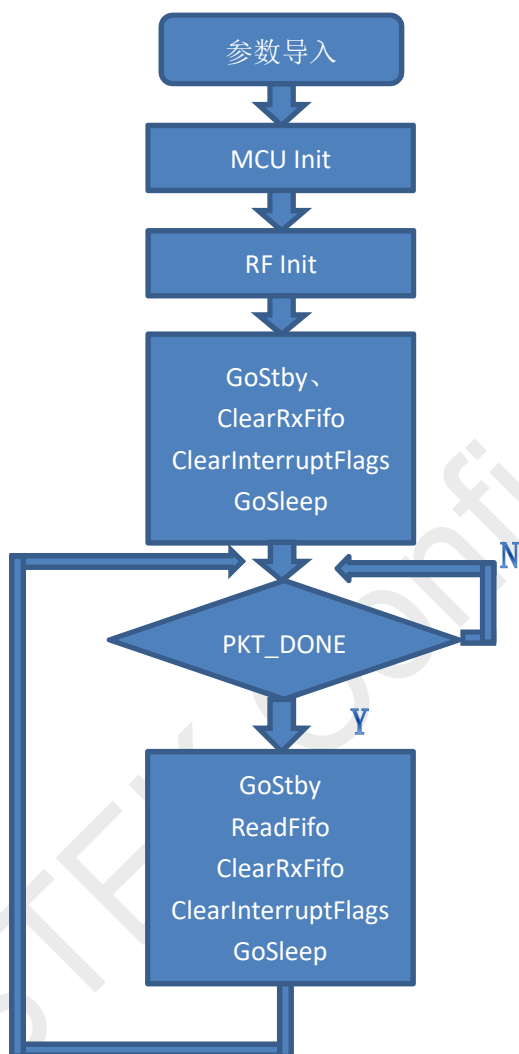
Sleep Time 为 Preamble 时间，即 $(1/9600) * 8 * 360 = 330\text{ms}$

接收机平均电流计算：

$$I = (T1 / T_{\text{sleep}}) * 8.5\text{mA} = (2.9 / 330) * 8.5 = 74.7\mu\text{A}$$

从上述电流计算公式可知，当增加发射机 Preamble 个数，接收机的电流会更低；
当将 PJD 的跳变个数减少，则 T1 时间减少，接收机的电流也会更低。

2、流程图



3、 实例

```

/*****
; Frequency          = 433.920 MHz
; Demodulation       = GFSK
; Data Rate          = 9.6 kbps
; Deviation          = 20.0 kHz
; Tx Power           = +20 dBm
; Payload Length     = 20
*****/
#define RF_PACKET_SIZE 20 /* Define the payload size here */
static u8 g_rxBuffer[RF_PACKET_SIZE]; /* RF Rx buffer */
static u8 g_txBuffer[RF_PACKET_SIZE]; /* RF Tx buffer */

void Mcu_Init(void)
{
    /* system init */
    SystemInit();
    GPIO_Config();
    NVIC_Config();
    SystemTimerDelay_Config();
    Timer5_Config();
    buzzer_init();
}

u8 Radio_Recv_FixedLen(u8 pBuf[],u8 len)
{
    if(CMT2300A_ReadGpio2()) // PKT_OK
    {
        CMT2300A_GoStby();
        CMT2300A_ReadFifo(pBuf,len);
        CMT2300A_ClearRxFifo();
        CMT2300A_ClearInterruptFlags();
        CMT2300A_GoSleep ();
        return 1;
    }

    return 0;
}

int main(void) //单收例程
{
    u8 i=0;
    Mcu_Init();
    RF_Init();

    CMT2300A_GoStby();

    /* Must clear FIFO after enable SPI to read or write the FIFO */
    CMT2300A_EnableReadFifo();
    CMT2300A_ClearInterruptFlags();
    CMT2300A_ClearRxFifo();
    CMT2300A_GoSleep ();

    while(1)
    {
        if(Radio_Recv_FixedLen(g_rxBuffer,RF_PACKET_SIZE))
        {
            for(i=0;i<RF_PACKET_SIZE;i++) //Clear Buff
                g_rxBuffer[i]=0;
            system_delay_10us(2);
        }
    }
}

```

文档变更记录

表 1.文档变更记录表

版本号	章节	变更描述	日期
0.1	所有	初始发布版本	2018-11-12

Copyright. CMOSTEK Microelectronics Co., Ltd. All rights are reserved.

The information furnished by CMOSTEK is believed to be accurate and reliable. However, no responsibility is assumed for inaccuracies and specifications within this document are subject to change without notice. The material contained herein is the exclusive property of CMOSTEK and shall not be distributed, reproduced, or disclosed in whole or in part without prior written permission of CMOSTEK. CMOSTEK products are not authorized for use as critical components in life support devices or systems without express written approval of CMOSTEK. The CMOSTEK logo is a registered trademark of CMOSTEK Microelectronics Co., Ltd. All other names are the property of their respective owners.