



Panchip Microelectronics Co., Ltd.

PAN1026 SDK 用户指南

当前版本: 1.1

发布日期: 2022.03

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 3 楼

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	2021.03	初始版本创建
V1.1	2022.03	模板更新

目 录

1 概述.....	1
1.1 开发平台.....	1
1.2 SDK 结构.....	1
1.3 功能介绍.....	2
2 SDK 工作流程介绍.....	3
2.1 TX/RX 工作时序介绍.....	3
2.2 默认参数更改.....	4
3 移植.....	5
3.1 驱动接口实现.....	5
3.1.1 接口函数封装.....	5
3.1.2 SPI 的 CS 操作注意事项.....	6
3.1.3 三线 SPI 和 I2C 注意事项.....	7
3.2 延时接口实现.....	9
3.3 收发包实现.....	9
4 不同模式收发流程介绍.....	10
4.1 普通模式及 BLE 模式发送.....	10
4.1.1 BLE 组包.....	11
4.2 增强型模式发送.....	11
4.2.1 ACK Payload 功能.....	11
4.3 普通模式与 BLE 模式接收.....	12
4.4 增强型模式接收.....	13
4.4.1 ACK Payload 功能.....	13
4.5 环境温度变化较大的情况.....	14

1 概述

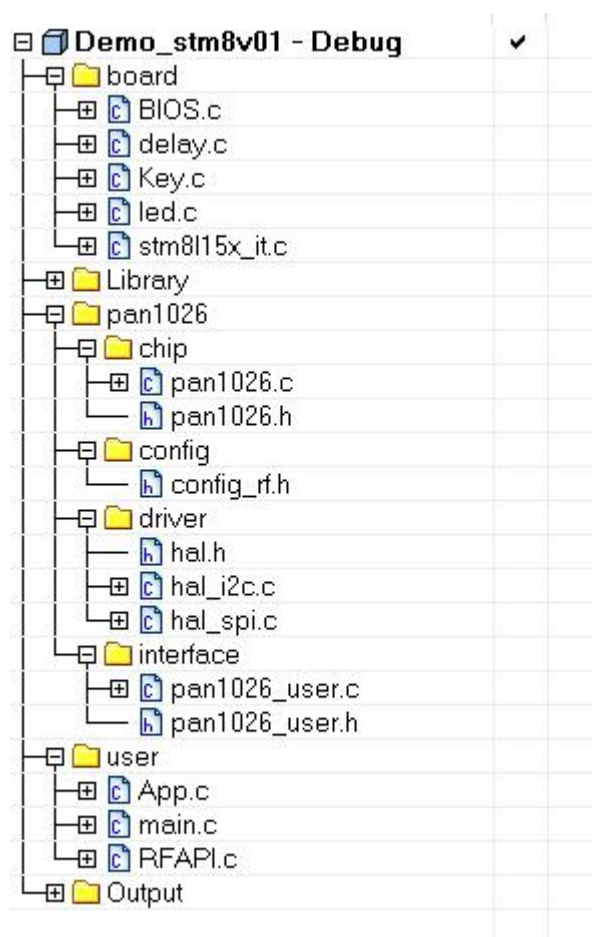
1.1 开发平台

PAN1026 评估板是 PAN1026 SDK 的配套硬件，板载 STM8L151G6 为主控 MCU。

PAN1026 SDK 使用 IAR 7.3 开发。

1.2 SDK 结构

PAN1026 SDK 结构如下图所示：



BOARD:

测试板硬件相关驱动(KEY, LED, USART, TIMER, RF)。

Library:

ST 官方库。

PAN1026:

CHIP: PAN1026 底层驱动。

DRIVE: PAN1026 接口驱动(包含 I2C, 3 线 SPI, 4 线 SPI)。

INTERFACE: PAN1026 应用接口(对底层做进一步封装), 供应用层使用。

CONFIG: PAN1026 常用参数配置。

USER:

main.c: 主函数入口。

App.c: 用户事件处理(按键响应, LED 显示)。

RFAPI.c: PAN1026 收发包处理。

1.3 功能介绍

PAN1026 有三种兼容模式。

兼容模式	说明
XN297L 模式	兼容 XN297L 的普通模式、增强型模式、增强型带 Payload 功能。 PAN1026 可与 XN297L 通信。
BLE 模式	兼容 BLE 4.0 广播包、数据包, 但无蓝牙协议栈。 PAN1026 可与手机 App 或其他蓝牙设备进行广播蓝牙通信。
NORDIC 模式	兼容 NORDIC NRF24L01 普通模式、增强型模式。 PAN1026 可与 NRF24L01 通信。

每种兼容模式, PAN1026 相互间都可通信。

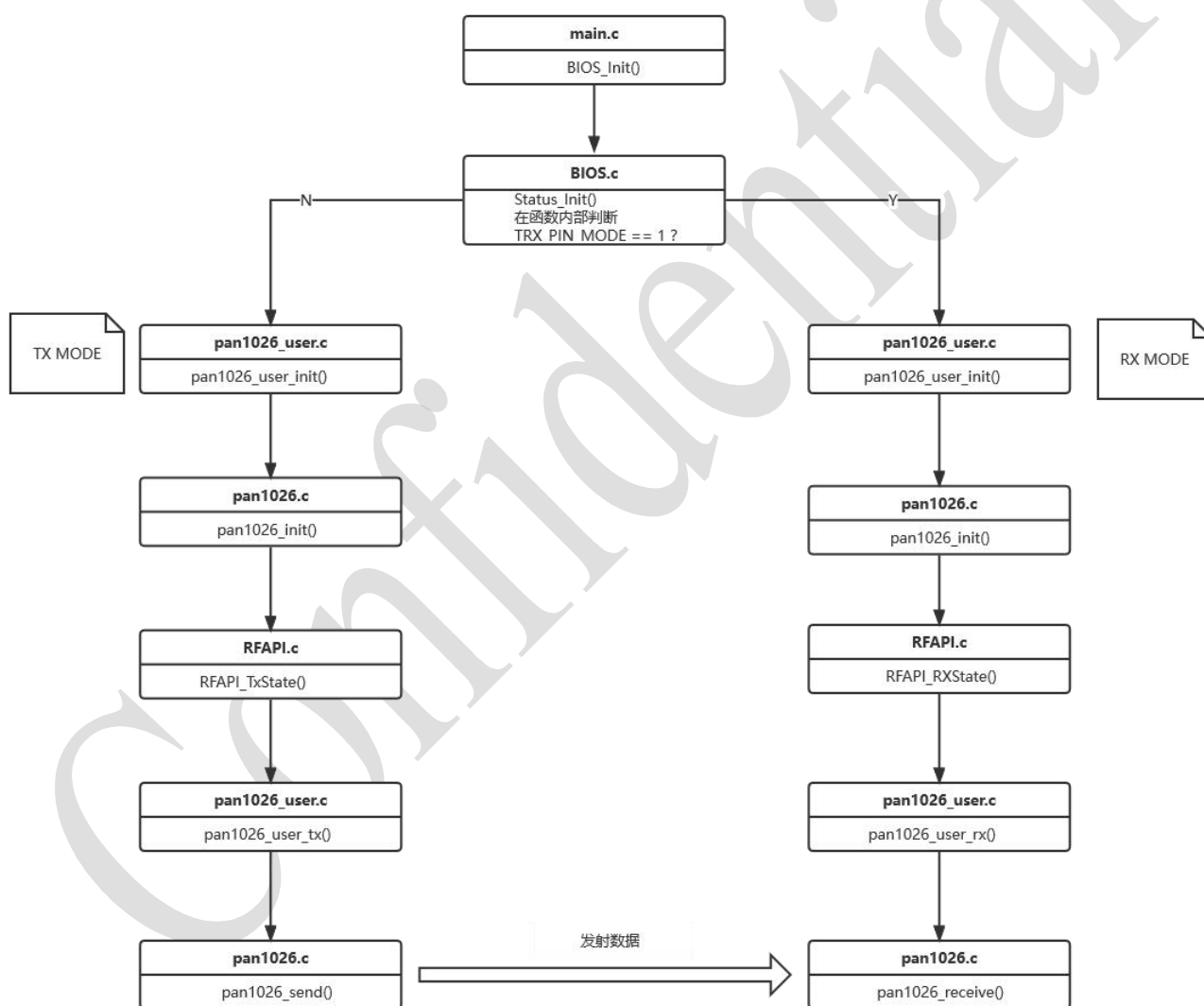
普通模式指发送端硬件只负责发, 接收端硬件只负责收, 收发转换完全依赖软件控制。

增强型模式指发送端硬件发完当前数据包后自动转为接收, 等待接收端的确认应答 (ACK), 接收端硬件收到数据包后自动转为发送, 发出 ACK; 增强型带 Payload 功能是在增强型模式的基础上, 接收端发 ACK 的同时, 附带了用户自定义的数据, 且数据长度可配。

2 SDK工作流程介绍

2.1 TX/RX 工作时序介绍

在应用中，一端初始化为发射，一端初始化为接收。简单的框架流程如下图所示：



- (1) TX/RX 模式默认在“config_rf.h”中设置，但在 DEMO 例程中最后会在 Status_Init()中通过 TRX_PIN_MODE(PA4)的状态更改 TX/RX 模式，用户移植时根据默认参数设置即可。
- (2) 初始化完成后在 RFAPL.c 中根据 TX/RX 状态处理不同的任务。收发包调用的路径：应用层 (RFAPL.c) -> 接口层(pan1026_user.c)->底层驱动(pan1026.c)。

2.2 默认参数更改

在 `pan1026_init(pan1026_com_interface* interface)` 中，根据默认的参数对 RF 进行初始化。

默认的参数在 `config_rf.h` 中修改，如下所示：

```

/***** PAN1026 参数设置 *****/
#define CONFIG_WORK_MODE          PAN1026_BURST_MODE          /* 工作模式为 XN297L 普通型 */
#define CONFIG_DRIVE_MODE        PAN1026_SPI_NO_MOSI_MODE     /* 驱动接口为 3 线 SPI */
#define CONFIG_IRQ                PAN1026_IRQ_DISABLE        /* 不使用 IRQ 引脚 */

#define CONFIG_TRX_MODE           PAN1026_TX_MODE             /* 设置 RF 为 TX MODE */
#define CONFIG_PAYLOAD_WIDTH      8                           /* 设置数据长度为 8bytes */
#define CONFIG_CHANNEL            78                          /* 设置频点为 2478 Mhz */
#define CONFIG_ADDR_LEN          5                           /* 设置地址长度为 5 字节 */
#define CONFIG_POWER              PAN1026_POWER_P_8           /* 设置功率为 8dbm */
#define CONFIG_RATE              PAN1026_RATE_1M              /* 设置速率为 1Mbps */

#define CONFIG_SCR                PAN1026_SCR_ENABLE          /* 打开扰码 */
#define CONFIG_CRC                PAN1026_CRC16               /* 打开 CRC，设置类型为 CRC16 */
#define CONFIG_RX_PIPE            (ERX_P0)                    /* 默认打开接收通道 0 */

#define CONFIG_ADDR_BYTE0         0x23
#define CONFIG_ADDR_BYTE1         0x71
#define CONFIG_ADDR_BYTE2         0x55
#define CONFIG_ADDR_BYTE3         0x0f
#define CONFIG_ADDR_BYTE4         0x71

```

3 移植

移植前将 PAN1026 文件夹添加到当前工程中，该文件夹下有四个子文件夹，CONFIG，DRIVER，INTERFACE 分别为配置文件，驱动文件，接口文件，这三个文件夹不是非必须的，可根据实际情况选择。CHIP 为 PAN1026 底层驱动，必须保留。

3.1 驱动接口实现

3.1.1 接口函数封装

PAN1026 支持标准四线 SPI、三线 SPI、I2C，其中四线 SPI 仅 QFN20 封装支持。

实现当前 MCU 的 SPI 或 I2C 读写功能后，按 pan1026.h 中要求的原型封装接口，可参考 hal_spi.c 或 hal_i2c.c。

```
typedef struct
{
    /**
     * @brief 向指定地址写入指定长的数据
     * @param addr 指定的地址
     * @param buf 待写入的数据
     * @param len 待写入数据的长度，字节为单位
     * @retval None
     */
    void (*rf_write_buf)(uint8_t addr, uint8_t *buf, uint8_t len);

    /**
     * @brief 从指定地址读取指定长的数据
     * @param addr 指定的地址
     * @param buf 待读取数据的存储空间
     * @param len 待读取数据的长度，字节为单位
     * @retval None
     */
    void (*rf_read_buf)(uint8_t addr, uint8_t *buf, uint8_t len);

}pan1026_com_interface;
```


实现好任意一种方式的接口后，通过 `pan1026_init(pan1026_com_interface* interface)` 函数传入到底层进行寄存器初始化工作。

```
/* 设置 PAN1026 通信接口，并初始化 */
#if(CONFIG_DRIVE_MODE == PAN1026_I2C_MODE )
    soft_i2c_for_rf_init();
    interface.rf_write_buf = soft_i2c_write_buf;
    interface.rf_read_buf = soft_i2c_read_buf;
    pan1026_init(&interface);
#elif(CONFIG_DRIVE_MODE == PAN1026_SPI_MODE)
    spi_for_rf_init();
    interface.rf_write_buf = spi_for_rf_write_buf;
    interface.rf_read_buf = spi_for_rf_read_buf;
    pan1026_init(&interface);
#else
    soft_spi_for_rf_init();
    interface.rf_write_buf = soft_spi_for_rf_write_buf;
    interface.rf_read_buf = soft_spi_for_rf_read_buf;
    pan1026_init(&interface);
#endif
```

3.1.2 SPI 的 CS 操作注意事项

写或读 PAN1026，分为写地址、写或读数据两部分，CS 要在这两部分整个过程中保持使能，即输出低电平，如下示例代码。MCU 硬件 SPI 一般都支持硬件自动控制 CS，可能无法满足要求，推荐软件独立控制 CS。

```
/**
 * @brief 硬件 SPI 向 PAN1026 指定地址发送指定长度的字节数据
 * @param addr 地址
 * @param buf 要发送的数据
 * @param len 数据长度, 字节为单位
 * @retval None
 */
void spi_for_rf_write_buf(uint8_t addr, uint8_t* buf, int len)
{
    GPIOB->ODR &= (uint8_t)(~GPIO_Pin_4);    /* CS Enable */

    /* 写地址 */
    while (SPI_GetFlagStatus(SPI1, SPI_FLAG_TXE) == RESET)
        ;
    SPI_SendData(SPI1, addr);
    while (SPI_GetFlagStatus(SPI1, SPI_FLAG_RXNE) == RESET)
        ;
    SPI_ReceiveData(SPI1);

    /* 写数据 */
    while (len--)
    {
        while (SPI_GetFlagStatus(SPI1, SPI_FLAG_TXE) == RESET)
            ;
        SPI_SendData(SPI1, *buf++);
        while (SPI_GetFlagStatus(SPI1, SPI_FLAG_RXNE) == RESET)
            ;
        SPI_ReceiveData(SPI1);
    }

    GPIOB->ODR |= GPIO_Pin_4;    /* CS Disable */
}
```

SPI 速率很慢时, CS 拉低后先短暂延时, 再读写, 读写完毕后再次短暂延时, 然后 CS 拉高。延时时间根据实际情况实测, 不加延时可能读写寄存器异常。

3.1.3 三线 SPI 和 I2C 注意事项

PAN1026 无 IRQ 引脚的封装, Rx 状态时 MCU 需不断读状态寄存器来查询是否产生接收中断。

I2C 和三线 SPI 的数据线是双向的, 即输出又输入, 若 MCU 和 PAN1026 的数据线同时是输出状态, 数据线容易出现较大电流, 影响射频接收数据。

使用 I2C，PAN1026 I2C 的 IO 会使能上拉电阻，MCU I2C 的 IO 配置成开漏输出，可避免异常。

使用三线 SPI，PAN1026 SPI 的 IO 不会使能上下拉。只有向 PAN1026 读数据时才会出现数据线输出输入状态的切换，读数据时写地址的最后一个时钟拉低后，PAN1026 立即将数据线切换成输出状态，MCU 要抢在之前完成输入状态切换才能避免异常；PAN1026 捕捉到 CS 拉高后才会将数据线切换成输入，所以读完数据后 MCU 不要立即将数据线恢复成输出，可以在下次写数据时设成输出；如下，hal_spi.c 中三线 SPI 写一个字节函数的实现。

```
/**
 * @brief IO 模拟三线 SPI，写一字节数据
 * @param dat 数据
 * @retval None
 */
static uint8_t soft_spi_for_rf_write_byte(uint8_t dat)
{
    uint8_t cnt;

    /* 每次写时配成输出，读函数中不要配置输入输出方向 */
    SPI_SET_MOSI_OUT();

    for (cnt = 0; cnt < 8; cnt++)
    {
        SPI_SCK_LOW();
        SOFT_SPI_DELAY();

        /* 发送数据 */
        if (dat & 0x80)
            SPI_MOSI_HIGH();
        else
            SPI_MOSI_LOW();

        dat <<= 1;
        SPI_SCK_HIGH();
        SOFT_SPI_DELAY();
    }

    /* 重要：最后一个时钟拉低前，就将数据线切换成输入 */
    SPI_SET_MOSI_IN();
    SPI_SCK_LOW();
    SOFT_SPI_DELAY();

    return 0;
}
```

3.2 延时接口实现

PAN1026 初始化阶段和发射数据时需要用到延时，SDK 中实现了一个 ms 级和 us 级的延时函数，用户根据实际 MCU 实现。

```
/**
 * @brief delay for some time in ms unit
 * @param n_ms is how many ms of time to delay
 * @retval None
 */
void delay_ms(unsigned int n_ms);

/**
 * @brief delay for some time in 10us unit
 * @param n_10us is how many 10us of time to delay
 * @retval None
 */
void delay_10us(unsigned int n_10us);
```

3.3 收发包实现

完成驱动接口后，使用 `pan1026_init(pan1026_com_interface *interface)` 完成初始化。完成后可调用 `pan1026_user` 中提供的接口进行数据的收发工作。

```
/**
 * @brief 发送指定长数据
 * @param buf 待发送数据
 * @param len 待发送数据长度
 * @retval 返回 PAN1026_TX_DONE 表示发送成功
 */
uint8_t pan1026_user_tx(uint8_t *buf, uint8_t len);

/**
 * @brief 接收
 * @param buf 接收数据缓存
 * @param max_size 接收数据最大长度，大于这个长度的数据包会被丢弃
 * @retval 接收到的数据长度，0 表示未收到数据
 */
uint8_t pan1026_user_rx(uint8_t *buf, uint8_t max_size);
```

4 不同模式收发流程介绍

PAN1026 SDK 中，将本章描述的典型收发流程作了封装，可直接使用或按需修改。

4.1 普通模式及 BLE 模式发送

步骤序号	步骤内容
1	设置工作模式（根据实际情况三选一）： <code>pan1026_set_mode(PAN1026_BURST_MODE);</code> /* XN297L 普通模式 */ <code>pan1026_set_mode(PAN1026_NORDIC_BURST_MODE);</code> /* NORDIC 普通模式 */ <code>pan1026_set_mode(PAN1026_BLE_MODE);</code> /* BLE 模式 */
2	设置 Tx 地址以及地址宽度（BLE 模式无需设置）： <code>pan1026_set_tx_addr(addr, addr_len);</code> <code>pan1026_set_addr_width(addr_len);</code>
3	设置频率（BLE 模式需区分广播频率和数据频率）： <code>pan1026_set_channel(78);</code>
4	进 Tx 模式： <code>pan1026_tx_mode();</code>
5	写 Tx FIFO： <code>pan1026_send(tx_buf, len);</code>
6	等待中断： 有 IRQ 引脚的封装，判断 IRQ 为低电平表示中断到来，然后调用 <code>pan1026_get_status()</code> 获取中断状态； 无 IRQ 引脚的封装，写完 Tx FIFO 后，需要先延时再调用 <code>pan1026_get_status()</code> 查询中断状态，延时时间要大于硬件发送数据包的时间，否则会影响发射性能； 判断中断： 判断 <code>pan1026_get_status()</code> 返回值是否为 PAN1026_TX_DONE，是则发送成功 否则发送失败
7	清 Tx FIFO，清中断： <code>pan1026_flush_tx_fifo();</code> <code>pan1026_clear_status();</code>
8	重复第 5-7 步，继续发送数据

上表为普通模式发送的典型流程，设置 Tx 地址和频率可根据需要随时调用，但注意不要在写 Tx FIFO 后、中断到来前调用。

4.1.1 BLE 组包

发送 BLE 广播包，需按 BLE 4.0 规定的格式组包。如下，PDU 就是要写入 PAN1026 Tx FIFO 的数据，其余部分硬件生成。

LSB		MSB	
Preamble (1 octet)	Access Address (4 octets)	PDU (2 to 39 octets)	CRC (3 octets)

PDU 格式要求如下，Header 先写入 Tx FIFO 且低字节在前。

LSB	MSB
Header (16 bits)	Payload (as per the Length field in the Header)

4.2 增强型模式发送

步骤序号	步骤内容
1	设置工作模式（根据实际情况两选一）： <code>pan1026_set_mode(PAN1026_ENHANCE_MODE);</code> /* XN297L 增强型 */ <code>pan1026_set_mode(PAN1026_NORDIC_ENHANCE_MODE);</code> /* NORDIC 增强型 */
2	设置 Tx 与 Rx 地址以及地址宽度 <code>pan1026_set_tx_addr(addr, addr_len);</code> <code>pan1026_set_rx_addr(addr, addr_len);</code> <code>pan1026_set_addr_width(addr_len);</code>
3-8	同普通模式 3-8 步骤

增强型模式，PAN1026_TX_DONE 的含义是数据发送成功且收到接收端的 ACK，即数据确认送达接收端，这与普通模式不同，普通模式下 PAN1026_TX_DONE 的含义仅仅表示数据已经发出，并不保证送达接收端。

NORDIC 增强型模式发射状态，发射第一包时硬件 VCO 校正消耗部分时间导致接收窗口延后，无法收到接收端的 ACK，软件将第一包发射两次即可。一般而言，软件应用都会有重试机制，该问题对应用影响不大。

4.2.1 ACK Payload 功能

在收发数据前调用 `pan1026_set_noack_ackpayload(1)` 启用 ACK Payload 功能。ACK Payload 功能目前仅支持 XN297L 增强型模式。

发送端发出数据后等待到的中断状态若是 PAN1026_TRX_DONE，表示收到 ACK 且收到 ACK Payload；若是 PAN1026_TX_DONE 表示仅收到 ACK，并未收到 ACK Payload；其他状态则发送失败。

发送端收到 ACK Payload 后，要调用 pan1026_read_ackpayload 读出 ACK Payload 具体数据，然后继续发送新的数据或进行其他操作。

可接收的 ACK Payload 长短与相关时间配置有关。

4.3 普通模式与 BLE 模式接收

步骤序号	步骤内容
1	设置工作模式（根据实际情况三选一）： pan1026_set_mode(PAN1026_BURST_MODE); /* XN297L 普通模式 */ pan1026_set_mode(PAN1026_NORDIC_BURST_MODE); /* NORDIC 普通模式 */ pan1026_set_mode(PAN1026_BLE_MODE); /* BLE 模式 */
2	设置 Rx 地址以及地址宽度（BLE 模式无需设置）： pan1026_set_rx_addr(addr, addr_len); pan1026_set_addr_width(addr_len);
3	设置频率(BLE 模式需区分广播频率和数据频率)： pan1026_set_channel(78);
4	设置收包长度（BLE 模式无需设置）： pan1026_set_rx_payload_len(8); /* 接收长 8 字节的数据包 */
5	进 Rx 模式： pan1026_rx_mode();
6	等待中断： 有 IRQ 引脚的封装，判断 IRQ 为低电平表示中断到来，然后调用 pan1026_get_status()获取中断状态； 无 IRQ 引脚的封装，要不断调用 pan1026_get_status()查询中断状态； 判断中断： 判断 pan1026_get_status()返回值是否为 PAN1026_RX_DONE，是则接收到数据；
7	调用 pan1026_receive 读出数据内容
8	请 Rx FIFO，清中断： pan1026_flush_rx_fifo(); pan1026_clear_status();
9	重复第 6-8 步，继续接收数据

4.4 增强型模式接收

步骤序号	步骤内容
1	设置工作模式（根据实际情况二选一）： <code>pan1026_set_mode(PAN1026_ENHANCE_MODE);</code> <i>/* XN297L 增强型 */</i> <code>pan1026_set_mode(PAN1026_NORDIC_ENHANCE_MODE);</code> <i>/* NORDIC 增强型 */</i>
2	设置 Tx 与 Rx 地址以及地址宽度 <code>pan1026_set_tx_addr(addr, addr_len);</code> <code>pan1026_set_rx_addr(addr, addr_len);</code> <code>pan1026_set_addr_width(addr_len);</code>
3	设置频率： <code>pan1026_set_channel(78);</code>
4	设置收包长度： <code>pan1026_set_rx_payload_len(8);</code> <i>/* 接收长 8 字节的数据包 */</i>
5	进 Rx 模式： <code>pan1026_rx_mode();</code>
6	等待中断： 有 IRQ 引脚的封装，判断 IRQ 为低电平表示中断产生； 延时一段时间； 然后调用 <code>pan1026_get_status()</code> 获取中断状态； 判断中断： 判断 <code>pan1026_get_status()</code> 返回值是否为 <code>PAN1026_RX_DONE</code> ，是则接收到数据
7	调用 <code>pan1026_receive</code> 读出数据内容
8	请 Rx FIFO，清中断： <code>pan1026_flush_rx_fifo();</code> <code>pan1026_clear_status();</code>
9	重复第 6-8 步，继续接收数据

注意步骤 6 中通过 IRQ 引脚得知中断产生后需要延时，因为增强型模式接收端收到数据后硬件自动发送 ACK，发送期间如果读写 PAN1026 会影响射频性能。SDK 默认配置，收到数据后 196us 才开始发送 ACK，若步骤 6-8 能在 196us 内完成，不加延时也可以；若来不及完成，发送 ACK 消耗的 90us 一并计入，延时 300us 以上。

无 IRQ 的封装，无法较好的规避读写对射频的影响。

4.4.1 ACK Payload 功能

在收发数据前调用 `pan1026_set_noack_ackpayload(1)` 启用 ACK Payload 功能。ACK

Payload 功能目前仅支持 XN297L 增强型模式。

接收端收到数据产生中断后，自动发送 ACK 前，软件必须在该时间内调用 `pan1026_write_ackpayload` 写入 ACK Payload 并执行完毕。该时间段是可配的。

4.5 环境温度变化较大的情况

若环境温度不稳定，会变化 20 度以上，需定期任意切换一下频率让 PAN1026 重新校准相关参数。