

## **PAN1070 RADIO B250K TX 使用说明**

PAN-CLT-VER-A0, Rev 1.1

PANCHIP

PanchipMicroelectronics

[www.panchip.com](http://www.panchip.com)

## 修订历史

版本	修订日期	描述
V1.0	2021-02-19	初始版本创建

## 目录

第 1 章 测试目的 .....	4
第 2 章 测试内容 .....	5
2.1 测试内容 .....	5
2.2 环境配置 .....	5
第 3 章 测试说明 .....	6
3.1 环境说明 .....	6
3.2 测试结果 .....	6
3.3 开发说明 .....	7
3.3.1 2.4G 发射初始化 .....	7
3.3.2 2.4G 初始化配置说明 .....	8
3.3.3 2.4G 中断处理 .....	13

## 第1章 测试目的

1. 测试 PAN1070 2.4G 250K 速率 TX 功能。

PANCHIP

## 第2章 测试内容

### 2.1 测试内容

1. 此项目演示了 2.4G 发射端功能：每隔 333ms 发送一次 2.4G 数据包，长度 5 个字节。
2. 接收端（参考 [PAN1070 RADIO B250K RX 使用说明.docx](#)）接收发送端的 2.4G 信号，并将接收到的数据通过串口打印出来。

### 2.2 环境配置

#### a) 环境要求

- board: pan1070\_evb
  - uart: 显示串口输出 log
  - PC 串口工具: Panchip Serial Assistant V0.0.006.exe
- 需要搭配一个运行“PRF\_B250K\_RX”的板子一起使用。

#### b) 编译和烧录

项目位置：“03\_MCU\mcu\_samples\PRF\_B250K\_TX”。

用 keil 编译程序，用 j-link 烧录编译后的 hex 文件到 pan1070\_evb 板子中。

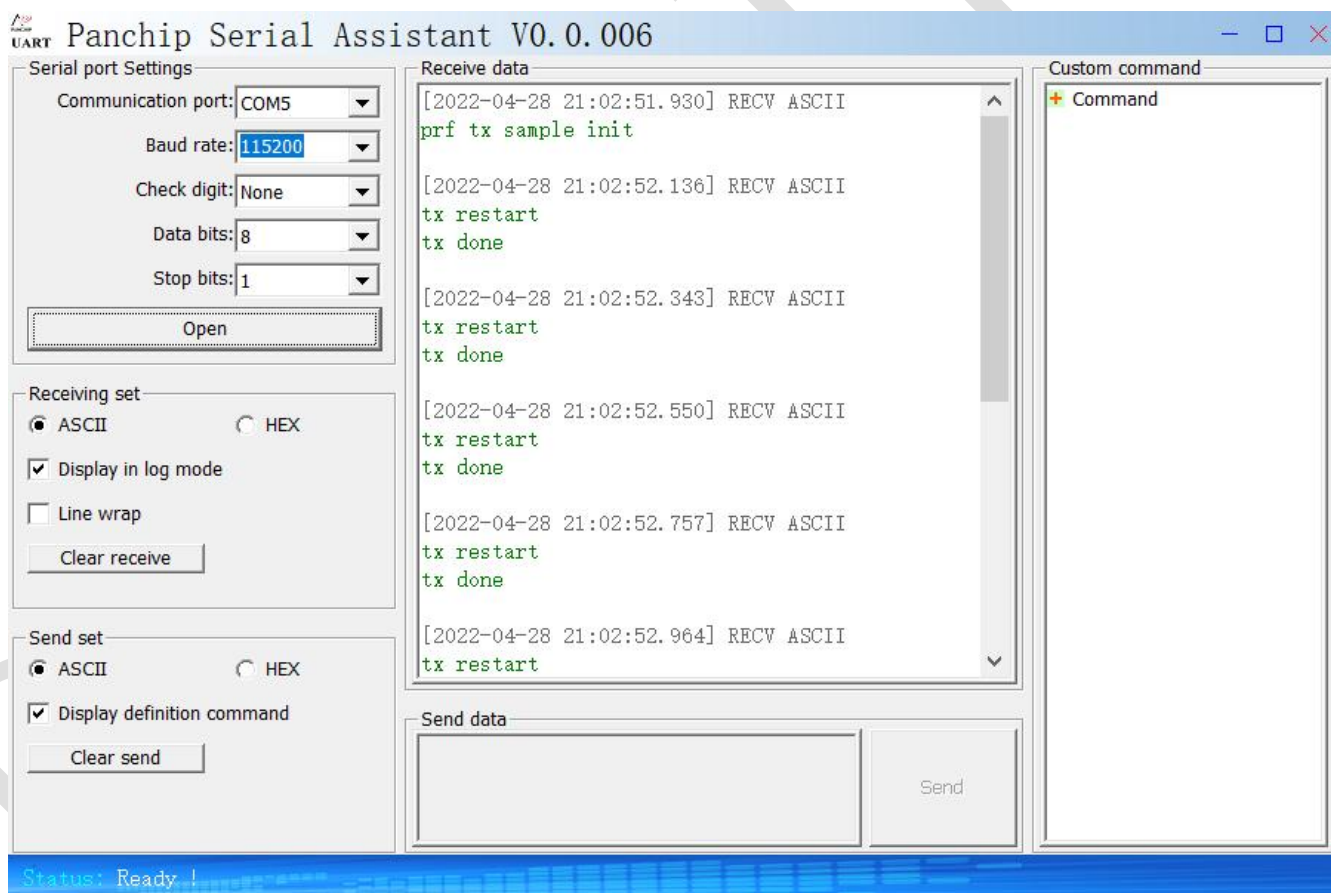
## 第3章 测试说明

### 3.1 环境说明

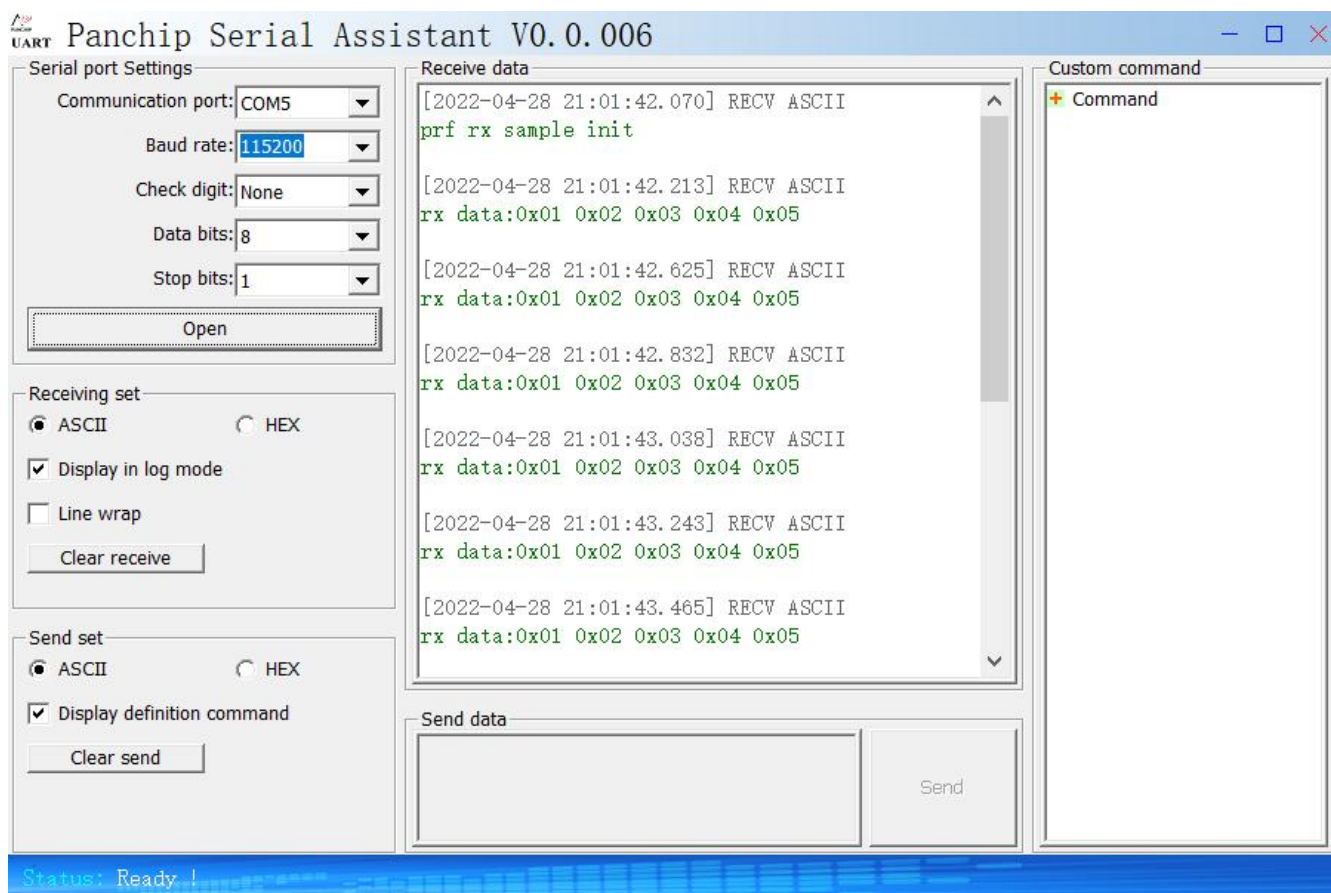
- a) 将接收端串口和发射端串口分别接到 PC 的 USB 端口上。
- b) 配置接收端和发送端（参考 [PAN1070\\_RADIO\\_B250K\\_RX 使用说明.docx](#)）。
- c) 观察 PC 串口工具的输出结果。

### 3.2 测试结果

- 1. 发射端输出结果：



## 2. 接收端输出结果:



## 3.3 开发说明

发射端启动了一个定时器,每隔 333ms 发送一包 2.4G 数据。发送成功串口输出 log“tx done”。

### 3.3.1 2.4G 发射初始化

```
1. pan_prf_config_t tx_config = {
2.     .work_mode           = PRF_MODE_NORMAL,
3.     .chip_mode          = PRF_CHIP_MODE_SEL_NORDIC,
4.     .trx_mode           = PRF_TX_MODE,
5.     .phy                = PRF_PHY_250K,
6.     .crc                = PRF_CRC_SEL_CRC16,
7.     .src                = PRF_SRC_SEL_NOSRC,
8.     .dev                = PRF_DEV_BLE,
9.     .rx_timeout         = 50000,        //us
10.    .rf_channel          = 2410,
11.    .tx_no_ack           = DISABLE,
12.    .trf_type            = PRF_TRF_BOE,
13.    .rx_length           = 0,
```

```

14.     .sync_length      = 4,
15.     .crc_include_sync  = DISABLE,
16.     .src_include_sync  = DISABLE,
17.     .tx_trans_time     = 140,          //us
18.     .rx_trans_time     = 150,          //us
19.     .sync              = { 0x91, 0xd3, 0x91, 0xd3 },
20.     .pipe              = PRF_PIPE0,
21. };
22.
23. panchip_prf_payload_t tx_payload = {
24.     .data_length       = 5,
25.     .data              = { 0x01, 0x02, 0x03, 0x04, 0x05 },
26. };
27.

```

### 3.3.2 2.4G 初始化配置说明

Type	name	Description
prf_mode_t	work_mode	工作模式配置，包括普通型和增强型
prf_chip_mode_sel_t	chip_mode	xn297 通信协议和 nordic 通信协议配置
prf_trx_mode_t	trx_mode	收发模式配置
prf_phy_t	phy	通信速率配置，可配置为 1M 和 2M
prf_crc_sel_t	crc	数据包 CRC 配置，可配置为 crc 16bit, crc 8bit, crc 24bit , no crc
prf_scramble_sel_t	src	数据包扰码的配置，可配置为使用扰码和不使用扰码
uint16_t	rx_timeout	接收超时时间配置，最大 50000us
uint16_t	rf_channel	2.4g 频点配置，任意频点可设 (2402Mhz~2480Mhz)
uint8_t	tx_no_ack	配置增强型模式下 tx 是否需要 ack
prf_trf_t	trf_type	nordic 的长包模式配置，最大 payload 的长



Type	name	Description
		度为 255
uint8_t	rx_length	rx 接收数据包长度配置, 增强型模式下可不配置
uint8_t	sync_length	接入地址长度配置, 可配置为 3、4、5 字节
uint8_t	sync[5]	接入地址的内容(xn297 模式下可白化地址, 防止出现长 0 和长 1 的地址)
prf_dev_sel_t	dev	设置 deviation, 可以选择 BLE 模式 (1M 250k; 2M 500k), NRF 模式 (1M 160K; 2M 320)
int8_t	tx_power	设置发射功率, 范围(-45dbm~7dbm)
uint8_t	pid_manual_flag	pid 手动配置的标志, 使能后可以自定义 pid
uint8_t	crc_include_sync	crc 计算包含地址
uint8_t	src_include_sync	白化包含地址
uint16_t	tx_trans_time	发送传输时间设置
uint16_t	rx_trans_time	接收传输时间设置
prf_pipe_t	pipe	管道配置, 可配置为 0~7

**prf\_mode\_t:**

Type	Value	Description
PRF_MODE_NORMAL	0	普通型
PRF_MODE_ENHANCE	1	增强型
PRF_MODE_NORMAL_M1	2	普通型 M1 模式

**prf\_chip\_mode\_sel\_t:**

Type	Value	Description
PRF_CHIP_MODE_SEL_BLE	1	蓝牙模式
PRF_CHIP_MODE_SEL_XN297	2	XN297 模式
PRF_CHIP_MODE_SEL_NORDIC	3	NORCDIC 模式

**prf\_trx\_mode\_t:**

Type	Value	Description
PRF_TX_MODE	0	2.4G 发射
PRF_RX_MODE	1	2.4G 接收

**prf\_phy\_t:**

Type	Value	Description
PRF_PHY_1M	1	1M 通信速率
PRF_PHY_2M	2	2M 通信速率
PRF_PHY_CODED_S8	3	S8 模式
PRF_PHY_CODED	4	S2 模式
PRF_PHY_250K	5	250K 模式

**prf\_crc\_sel\_t:**

Type	Value	Description
PRF_CRC_SEL_NOCRC	0	no crc
PRF_CRC_SEL_CRC8	1	crc 8bit
PRF_CRC_SEL_CRC16	2	crc 16bit
PRF_CRC_SEL_CRC24	3	crc 24bit

**prf\_scramble\_sel\_t:**

Type	Value	Description
PRF_SRC_SEL_NOSRC	0	不使能扰码
PRF_SRC_SEL_EN	1	使能扰码

**prf\_dev\_sel\_t:**

Type	Value	Description
PRF_DEV_NRF	1	NRF 模式 deviation 配置, 1M 170k, 2M 340K
PRF_DEV_BLE	2	NRF 模式 deviation 配置, 1M 250k, 2M 500K

**prf\_addr\_length\_sel\_t:**

Type	Value	Description
PRF_ADDR_LENGTH_SEL_3	3	3 BYTE 地址长度
PRF_ADDR_LENGTH_SEL_4	4	4 BYTE 地址长度
PRF_ADDR_LENGTH_SEL_5	5	5 BYTE 地址长度

**prf\_pipe\_t:**

Type	Value	Description
PRF_PIPE0	1<<0	管道 0
PRF_PIPE1	1<<1	管道 1
PRF_PIPE2	1<<2	管道 2
PRF_PIPE3	1<<3	管道 3
PRF_PIPE4	1<<4	管道 4
PRF_PIPE5	1<<5	管道 5
PRF_PIPE6	1<<6	管道 6
PRF_PIPE7	1<<7	管道 7

prf\_trf\_t:

Type	Value	Description
PRF_TRF_NORMAL	0	普通模式传输
PRF_TRF_NRF52	1	NRF 模式传输
PRF_TRF_BOE	2	250k 模式传输

### 3.3.3 2.4G 中断处理

#### 1. Tx 普通型触发的中断

Tx 完成后会触发 tx done 中断，其他中断不会执行，tx 中断执行完后 tx 流程就结束了。  
Tx 中断接口如下：

```
1. void event_tx_fun(void)
2. {
3.     printf("tx done\n");
4. }
```

#### 2. Tx 增强型触发的中断

Tx 完成后会触发 tx done 中断，tx 中断执行完后会进入 rx 模式，rx 有以下中断接口：

##### 1) RX 收到数据

```
2) void event_rx_fun(void)
3) {
4)     panchip_prf_payload_t rx_payload;
5)
6)     rx_payload.data_length = panchip_prf_data_rec(&rx_payload);
7)     printf("rx data:");
8)     data_printk(rx_payload.data, rx_payload.data_length);
9) }
```

将收到的 ack 数据打印出来。

##### 2) RX 接收超时

```
3) void event_rx_timeout_fun(void)
4) {
5)     printf("rx timeout\n");
6) }
```

超时时间内未收到数据则会触发该中断，超时时间就是初始化设置的时间。

### 3) RX 接收 CRC 错误

```
4) void event_crc_err_fun(void)
5) {
6)     printf("rx data crc err\n");
7) }
```

接收到数据但是 CRC 校验错误则会触发该中断，该中断也可将收到的 ack 数据打印出来。CRC 未使能该中断不会触发。

### 4) RX 接收 PID 错误

```
5) void event_pid_err_fun(void)
6) {
7)
8) }
```

当接收重发包的 pid 异常时会触发该中断，该中断可将收到的 ack 数据打印出来。