

PAN1070 PRF TRX 开发指南

PAN-CLT-VER-A0, Rev 1.1

PANCHIP

PanchipMicroelectronics

www.panchip.com

修订历史

版本	修订日期	描述
V0.1	2023-11-13	初始版本创建
V1.0	2024-05-30	新增 tx_noack 和地址注意事项
V1.1	2024-09-28	新增多 pipe 相关接口

目录

第 1 章 例程概述	4
1.1 例程功能	4
1.2 环境配置	4
第 2 章 测试说明	5
2.1 环境说明	5
2.2 测试结果	5
第 3 章 开发说明	7
3.1 2.4G 初始化配置说明	7
3.2 2.4G API 介绍	10
3.3 2.4G 中断介绍	16
第 4 章 Sample 运行流程	17
4.1 PRF_RX_SAMPLE 例程运行流程	17
4.2 PRF_TX_SAMPLE 例程运行流程	18
第 5 章 2.4G 帧结构介绍	20

第1章 例程概述

1.1 例程功能

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。

1.2 环境配置

a) 环境要求

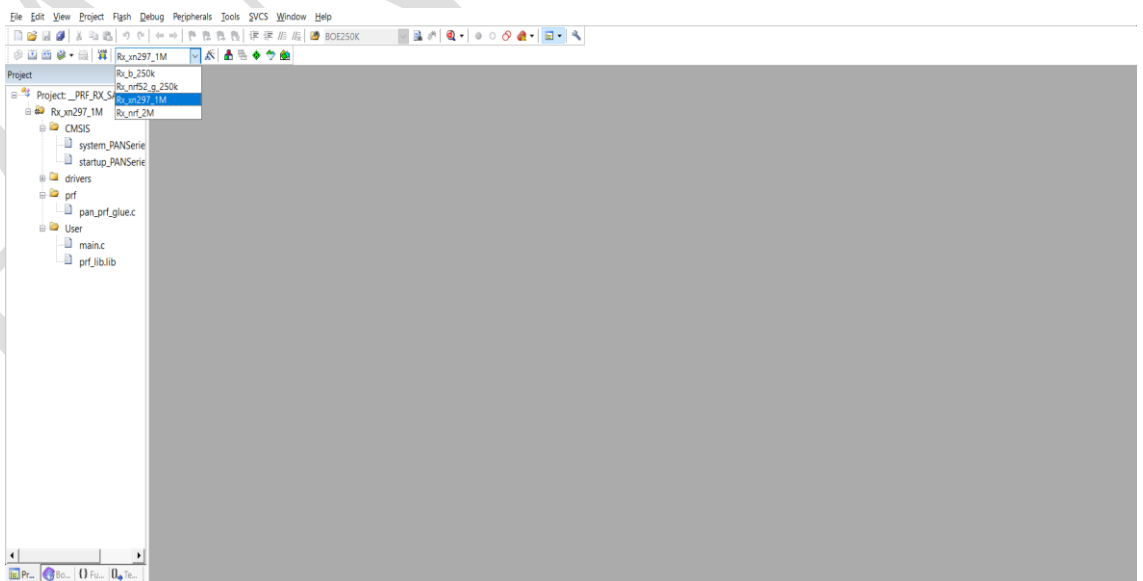
- board: pan1070_evb
 - uart: 显示串口输出 log，uart 端口 P16 (UART_RX)、P17 (UART_TX)
 - PC 串口工具：Panchip Serial Assistant V0.0.010.exe
- “PRF_TX_SAMPLE”的板子需要搭配和“PRF_RX_SAMPLE”的板子一起使用。

b) 编译和烧录

1. 项目位置：

- TX 端：“03_MCU\mcu_samples\PRF_TX_SAMPLE”
- RX 端：“03_MCU\mcu_samples\PRF_RX_SAMPLE”

2. TX 端和 RX 端选择 project target，如下图所示：



工程有四种 project target，分别对应四种不同的配置。

- B_250k, 250k 私有调制方式。

- NRF52 250k, 可以和 nrf52832 250k 通信, GFSK 调制方式
- XN297 1M, 可以和 xn297 1M phy 通信
- NRF 2M, 可以和 nrf24101 2M phy 通信

不同的配置也是可以修改的, 分别在不同 target 的 “sample_config” 文件中, 用户可以根据自己的需求修改相应的配置。

3. 选择好后编译程序, 用 j-link 烧录编译后的 hex 文件到 pan1070_evb 板子中。

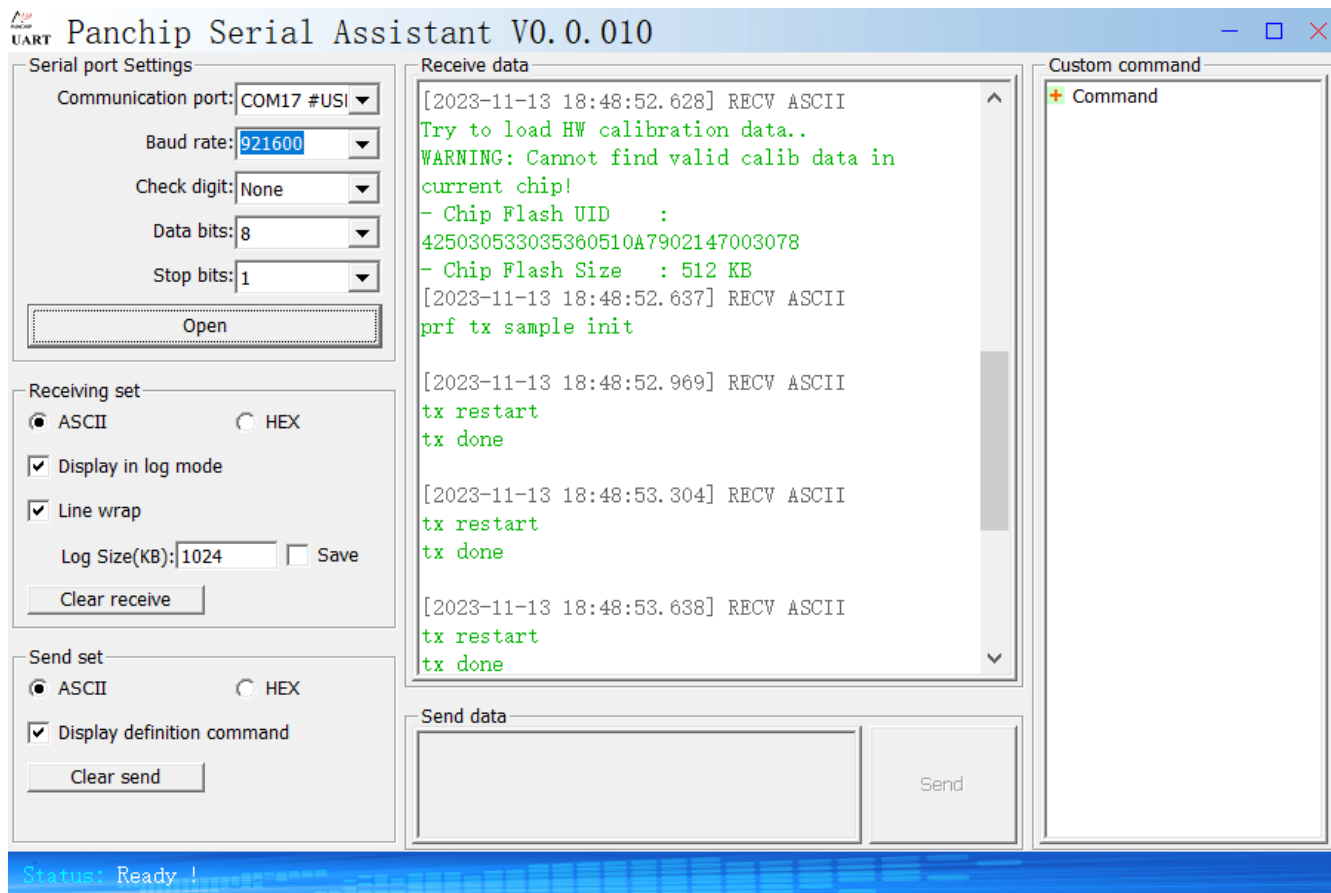
第2章 测试说明

2.1 环境说明

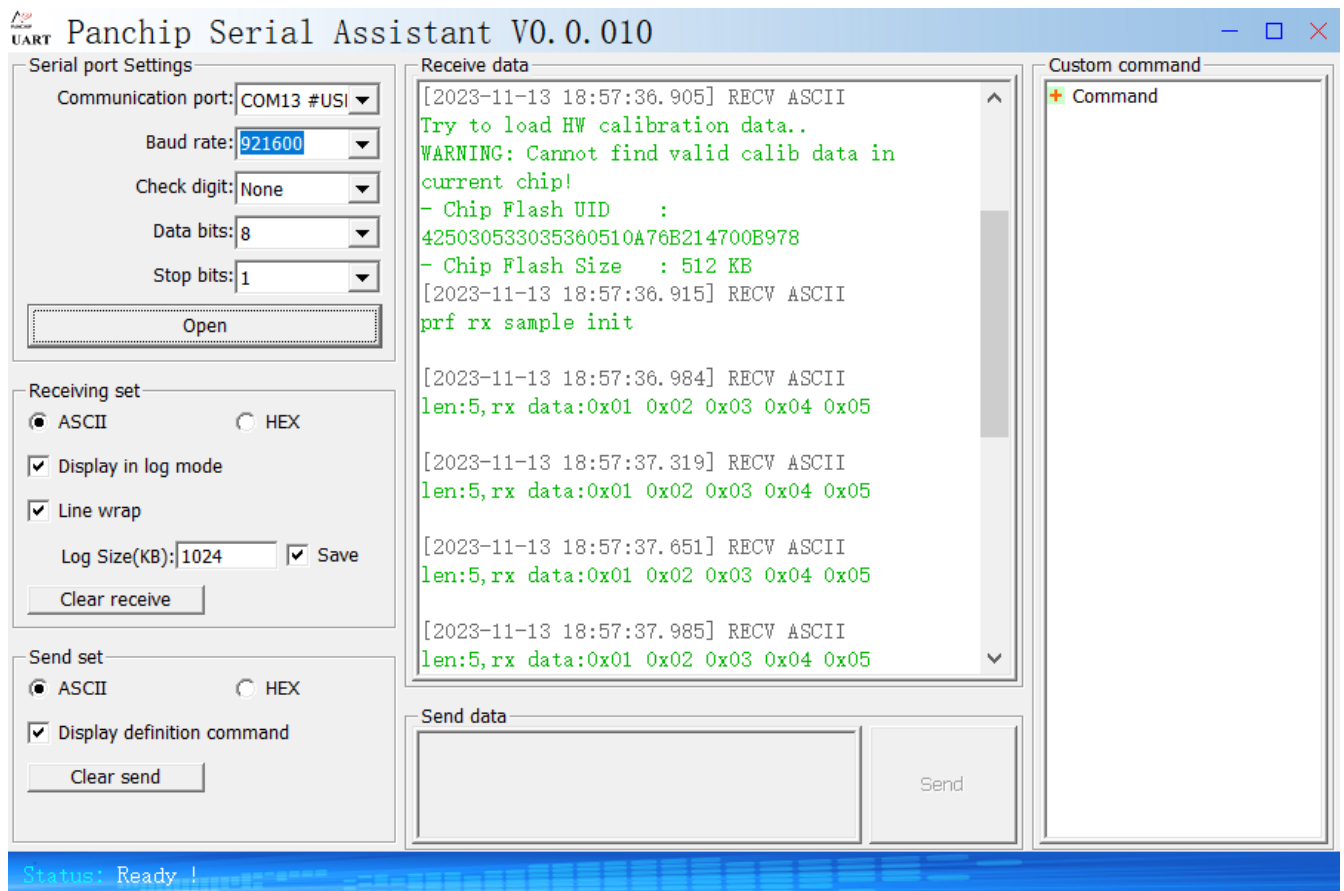
- a) 将接收端串口和发射端串口分别接到 PC 的 USB 端口上。
- b) 配置接收端和发送端。
- c) 观察 PC 串口工具的输出结果。

2.2 测试结果

1. 发射端输出结果:



2. 接收端输出结果:



第3章 开发说明

3.1 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、增强型和自定义模式
prf_chip_mode_sel_t	chip_mode	xn297 通信协议和 nrf 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、2M、CODED_S8、CODED_S2、250K
prf_crc_sel_t	crc	数据包 CRC 配置，可配置为 crc 24bit, crc 16bit, crc 8bit, 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
prf_mode_conf_sel_t	mode_conf	2.4g 不同模式射频参数配置
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，配置范围 2402-2480Mhz 和带外
uint8_t	tx_no_ack	配置增强型模式下 tx 是否需要 ack
prf_trf_t	trf_type	特殊通信模式设置
uint8_t	rx_length	rx 接收数据包长度配置

uint8_t	sync_length	接入地址长度配置，可配置为 2、3、4、5 字节
uint8_t	sync[5]	接入地址的内容
uint8_t	crc_include_sync	crc 是否包含接入地址（XN297 和 NRF24L01 必须使能）
uint8_t	src_include_sync	扰码是否包含接入地址
prf_pipe_t	pipe	设置不同地址通信通道（0~7）
int8_t	tx_power	设置发射功率（-40dbm~9dbm）
uint8_t	pid_manual_flag	是否手动设置 pid

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	自定义帧格式（payload 最大 255byte）

prf_chip_mode_sel_t:

Type	Value	Description
PRF_CHIP_MODE_SEL_XN297	2	XN297 模式
PRF_CHIP_MODE_SEL_NRF	3	NRF 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 编码 PHY
PRF_PHY_CODED	4	S2 编码 PHY
PRF_PHY_250K	5	250k 通信速率

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_mode_conf_sel_t:

Type	Value	Description
PRF_BLE_CONF	1	BLE 射频参数配置, deviation 250k(1M)
PRF_NRF_CONF	2	NRF 射频参数配置, deviation 170k(1M)
PRF_G_250k_CONF	3	GFSK 调制 250K 通信速率射频参数配置
PRF_B_250K_CONF	4	私有调制 250K 通信速率射频参数配置 (B_250K)

prf_addr_length_sel_t:

Type	Value	Description
PRF_ADDR_LENGTH_SEL_2	2	2 BYTE 地址长度
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 长包模式 (payload 最大 255byte) 传输
PRF_TRF_B_250K	2	B_250K 通信速率模式传输

prf_endian_t:

Type	Value	Description
PRF_BIG_ENDIAN	0	大端模式传输
PRF_LITTLE_ENDIAN	1	小端模式传输

3.2 2.4G API 介绍

3.2.1 panchip_prf_init

```
void panchip_prf_init(pan_prf_config_t *p_config);
```

初始化配置的结构体“pan_prf_config_t”

3.2.2 panchip_prf_trx_start

```
void panchip_prf_trx_start(void);
```

调用此接口，普通型模式 **RX** 状态下开始 **RX**，在 **TX** 状态下开始 **TX**，增强型模式下，进行收发转换。

3.2.3 panchip_prf_set_data

```
void panchip_prf_set_data(panchip_prf_payload_t *p_payload);
```

TX 模式下设置 **payload** 的内容和长度。

3.2.4 panchip_prf_set_ack_data

```
void panchip_prf_set_ack_data(panchip_prf_payload_t *p_payload);
```

RX 模式下设置 **ack payload** 的内容和长度，**ack** 空包需要将 **data length** 设为 0。

3.2.5 panchip_prf_data_rec

```
uint8_t panchip_prf_data_rec(panchip_prf_payload_t *p_payload);
```

RX 模式下获取接收的数据和长度，返回值是数据长度。必须在 **TRX 中断中读取数据**，在其他地方读数据会异常。

3.2.6 panchip_switch_prf

```
void panchip_switch_prf(pan_prf_config_t *p_config);
```

切换通信协议的帧结构，**xn297** 模式和 **NRF** 模式。

3.2.7 panchip_prf_set_chn

```
void panchip_prf_set_chn(pan_prf_config_t *p_config);
```

设置通信的频点，带内 **2400~2480** 任意频点可设，也可设置带外频点。必须在 **TX** 和 **RX** 退出后使用。

3.2.8 panchip_prf_set_tx_pwr

```
void panchip_prf_set_tx_pwr(int8_t tx_pwr);
```

设置发射的功率，功率范围 **-40dbm~9dbm**。

3.2.9 panchip_prf_mode_conf_set

```
void panchip_prf_mode_conf_set(prf_mode_conf_sel_t conf);
```

设置 2.4g 不同模式射频参数配置，可以选择 **BLE** 模式、**NRF** 模式、**GFSK 250K**、私有调制 **250K**。

3.2.10 panchip_prf_set_phy

```
void panchip_prf_set_phy(pan_prf_config_t *p_config);
```

设置通信速率，可以选择 **1M**、**2M**、**PRF_PHY_CODED_S8**、**PRF_PHY_CODED**、**PRF_PHY_250K**。

3.2.11 panchip_prf_set_trx_mode

```
void panchip_prf_set_trx_mode(prf_trx_mode_t trx_mode);
```

设置接收模式和发射模式，在 **prf** 是 **idle** 状态下调用。

3.2.12 panchip_prf_set_work_mode

```
void panchip_prf_set_work_mode(prf_mode_t work_mode);
```

设置普通型工作模式和增强型工作模式，在 **prf** 是 **idle** 状态下调用。

3.2.13 panchip_prf_set_addr

```
void panchip_prf_set_addr(uint8_t *addr, uint8_t len, prf_pipe_t pipe, uint8_t trx_addr);
```

设置通信地址内容、长度、地址通道、地址类型。

地址类型分为以下三种：

PRI_RF_MODE_SEL_TX	0	设置 TX 地址
PRI_RF_MODE_SEL_RX	1	设置 RX 地址
PRI_RF_MODE_SEL_TRX	2	设置 TRX 地址，TX 和 RX 地址相同

TX 地址只有一个 **PIPE**，**RX** 地址有 **8** 个 **PIPE**。

NOTE: 地址的最后一个字节不能出现连续的 **6** 个 **bit** 为 **0** 或者为 **1**。

3.2.14 panchip_prf_set_tx_noack

```
void panchip_prf_set_tx_noack(bool flag);
```

设置 TX 增强型模式下是否需要接收 ACK，设置成 true 后，tx 后不会进接收。

NOTE: 该字段和 NRF52 系列的 noack 字段是相反的，PAN1070 的 true 对应 NRF52 的 false，PAN1070 的 false 对应 NRF52 的 true。由于 PAN1070 该字段是硬件操作的，适配需要修改 NRF52 的软件。

PAN1070 和 NRF2401、XN297 通信无此问题。

3.2.15 panchip_prf_rx_timeout

```
void panchip_prf_rx_timeout(uint16_t time);
```

设置接收的超时时间，单位 us，最大 50000us。

3.2.16 panchip_prf_reset

```
void panchip_prf_reset(void);
```

复位 PRF 状态机，使其处于空闲模式。

3.2.17 panchip_prf_rx_stop

```
void panchip_prf_rx_stop(void);
```

退出接收模式，RX 退出后会产生 timeout 中断。

3.2.18 panchip_prf_pid_cfg

```
void panchip_prf_pid_cfg(uint8_t pid);
```

增强型模式手动设置 TX PID，PID 设置的范围是 0~3。

3.2.19 panchip_white_init_value

```
void panchip_white_init_value(uint8_t value);
```

白化使能后，白化初始值设置，默认是 **0x7f**。

3.2.20 panchip_prf_carrier_start

```
void panchip_prf_carrier_start(uint16_t tx_channel);
```

发射单载波，参数设置任意频点。必须在 **TX** 和 **RX** 退出后使用。

3.2.21 panchip_prf_carrier_stop

```
void panchip_prf_carrier_stop(void);
```

退出发射单载波模式。

3.2.22 panchip_prf_read_rssi

```
int16_t panchip_prf_read_rssi(void);
```

收到数据后，在接收中断中获取数据强度的 **RSSI** 值。这个 **rssi** 会缓存，收到新的数据后值会覆盖。

3.2.23 panchip_prf_enable_carrier_rssi

```
void panchip_prf_enable_carrier_rssi(void);
```

使能获取载波强度的 **RSSI** 模式，调用后不能接收数据。

3.2.24 panchip_prf_read_carrier_rssi

```
int16_t panchip_prf_read_carrier_rssi(void);
```

获取载波强度的 **RSSI** 值，这个 **rssi** 值是实时的，不会缓存。

3.2.25 panchip_prf_disable_carrier_rssi

```
void panchip_prf_disable_carrier_rssi(void);
```

载波强度 **RSSI** 获取后，关闭 **rssi** 功能。关闭后可以正常接收数据。

3.2.26 panchip_prf_rx_length_irq_cfg

```
void panchip_prf_rx_length_irq_cfg(uint8_t value);
```

配置是否使能 **rx length err** 中断，当 **rx** 收到数据硬件解析出来的 **length** 值大于配置的 **rx length**（结构体中“**rx_length**”字段），**rx length err irq** 会产生，普通型和增强型接收都有效。在使用前需要先注册中断函数。

3.2.27 panchip_prf_get_pipe

```
uint8_t panchip_prf_get_pipe(void);
```

收到数据获取地址通道号，范围 0~7。

3.2.28 panchip_prf_set_crc

```
void panchip_prf_set_crc(pcrc_sel_t crc, uint8_t crc_include_sync);
```

设置 **crc** 的类型和 **crc** 的范围是否包含地址。

3.2.29 panchip_prf_set_whitening

```
void panchip_prf_set_whitening(prf_scramble_sel_t src, uint8_t src_include_sync);
```

设置白化类型和白化范围是否包含地址。

3.2.30 panchip_prf_set_endian

```
void panchip_prf_set_endian(prf_endian_t endian);
```

设置 **payload** 数据传输的大小端。

3.3 MULTI PIPE 使用介绍

多 **pipe** 的使用主要在于接收端，发送端同时只有一个 **pipe**，接收端同时有 **8** 个地址。

涉及到两个接口，**panchip_prf_set_addr** 和 **panchip_prf_get_pipe**。

panchip_prf_set_addr 为每个 **pipe** 设置不同的地址，接收端收到数据后可以使用 **panchip_prf_get_pipe** 获取当前收到数据对应的 **pipe** 地址。

详细使用情况请参考例程“**PRF_RX_MULTI_PIPE**”和“**PRF_TX_MULTI_PIPE**”。RX 端将收到的数据打印出来，同时打印当前的 **pipe** 通道号，TX 端每 **1s** 切换不同 **pipe** 通道，同时每个 **pipe** 通道配置相应的地址。注意：接收端在增强型模式下收到数据后需要配置 TX 地址，地址为当前 **pipe** 通道的地址。

3.4 2.4G 中断介绍

中断介绍如如下代码所示：

```
1. void event_tx_fun(void)
2. {
3.     printk("tx done\n");
4. } //增强型和普通型模式 TX 结束后会产生 TX 中断，TX 退出，增强型 TX 完了之后会马上进入 RX。
5.
6. void event_rx_fun(void)
7. {
8.     panchip_prf_payload_t rx_payload;
9.
10.    rx_payload.data_length = panchip_prf_data_rec(&rx_payload);
11.    printk("rx data:");
12.    data_printk(rx_payload.data, rx_payload.data_length);
13. } //增强型和普通型模式 RX 收到数据后会进入 RX 中断,RX 退出，增强型 RX 完了之后会马上进入 TX。
14.
15. void event_rx_timeout_fun(void)
16. {
17.     printk("rx timeout\n");
18. } //接收超时后会进入 timeout 中断,RX 退出
19.
20. void event_crc_err_fun(void)
```



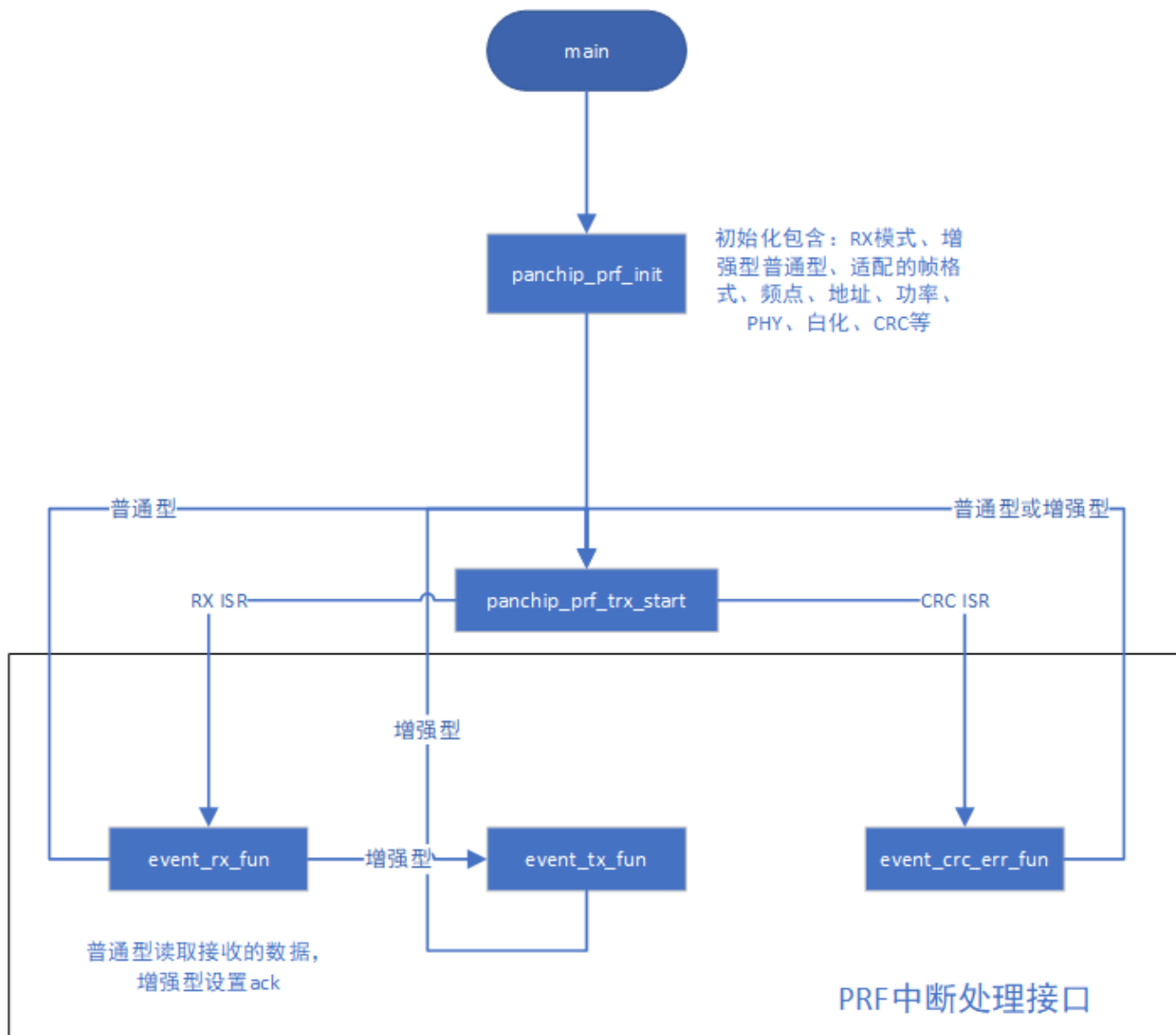
```
21. {
22.     printk("crc err\n");
23. } //接收数据错误会进入 crc 中断，RX 退出，不会进入 TX
24.
25. /*普通型 TX，TX 结束退出后产生 TX 中断，进入下一次 TX 后须调用"panchip_prf_trx_start"。*/
26.
27. /*普通型 RX，收到数据后会进入 RX 中断，接收超时会进入 timeout 中断，收到错误数据会进入 crc 中断，进入下一次 RX 后须调用"panchip_prf_trx_start"。*/
28.
29. /*增强型 TX，TX 结束后硬件会自动进入 RX，RX 收到数据后会进入 RX 中断，接收超时会进入 timeout 中断，收到错误数据会进入 crc 中断，进入下一次 TX 后须调用"panchip_prf_trx_start"。*/
30.
31. /*增强型 RX，RX 收到数据后会硬件会自动进入 TX，TX 结束退出后产生 TX 中断，接收超时会进入 timeout 中断，收到错误数据会进入 crc 中断，进入下一次 RX 后须调用"panchip_prf_trx_start"。*/
```

Note: 在高性能的 2.4g 应用中尽量不要使用打印，会影响射频收发性能。增强型 RX 中断中不能有延时或者打印或者执行时间很长的接口，不然会影响 TX 中断从而导致时序错误。

第4章 Sample 运行流程

4.1 PRF_RX_SAMPLE 例程运行流程

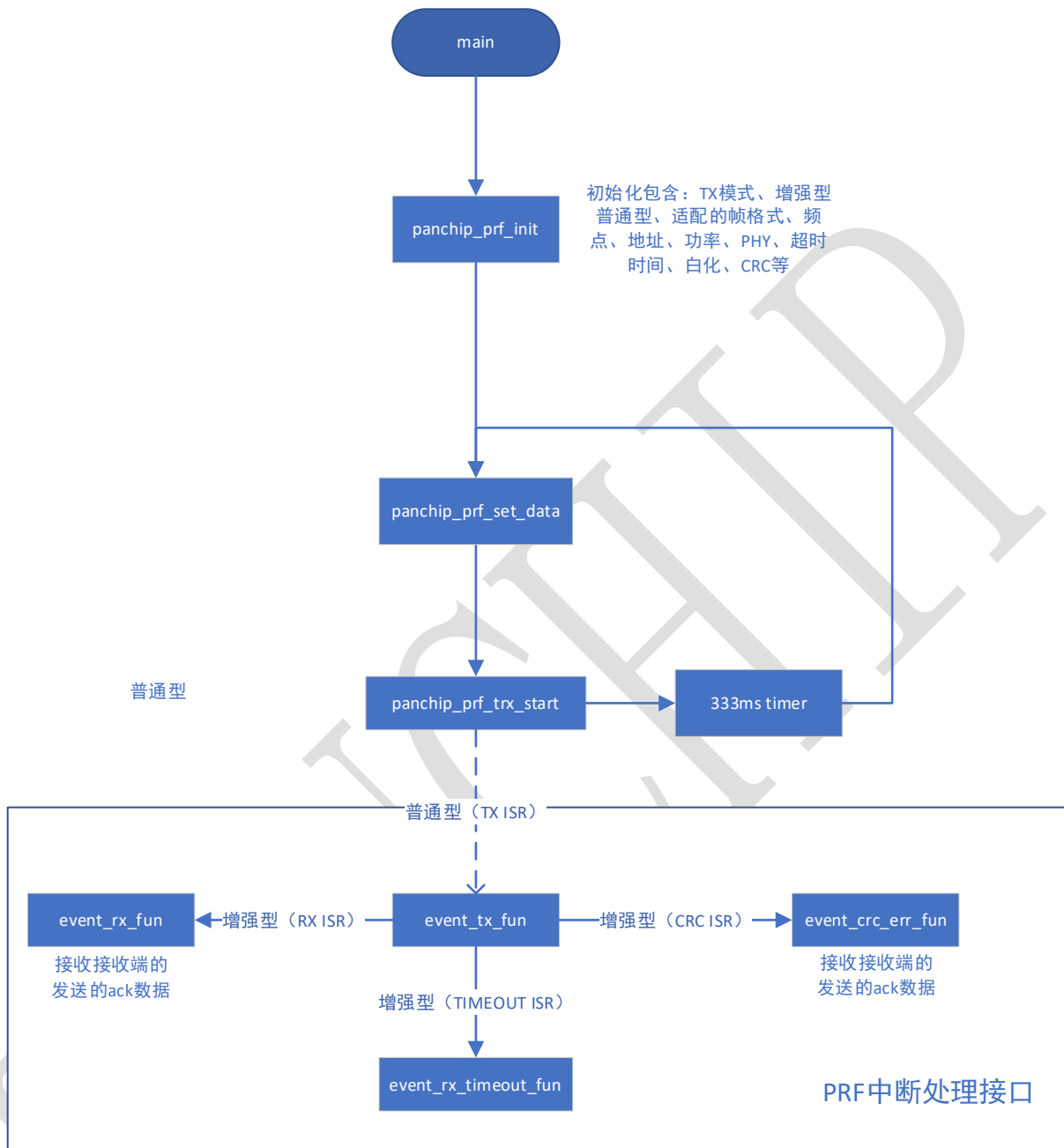
如下图所示：



RX 例程流程图

4.2 PRF_TX_SAMPLE 例程运行流程

如下图所示:



TX 例程流程图

第5章 2.4G帧结构介绍

兼容三种帧结构：**xn297**、**nrf24l01**、**nrf52**

5.1.1 xn297 兼容帧结构

普通型：

3 byte	3~5 byte	1~64 byte	0/1/2/3 byte
preamble(0x710f55)	addr	payload	crc(包含 addr 和 payload)

空中 **bit** 序：大端模式

增强型：在地址和 **payload** 之间插入 **signal(10bit)**数据。

3 byte	3~5 byte	10bit	1~64 byte	0/1/2/3 byte
preamble(0x710f55)	addr	signal	payload	crc(包含 addr 和 payload)

signal 结构：

7bit	2bit	1bit
数据长度标识(动态 payload)	PID 标识 (判断是否是重发包)	NO_ACK 标识 (tx 完是否需要 ack)

数据长度 **7bit** 最大包长发送可支持 **127byte**，接收最大包长可支持 **128byte**。

白化和 **crc** 的作用域包括 **addr+payload+signal**。

5.1.2 nrf2401 兼容帧结构

普通型：

空中 **bit** 序：大端模式

1 byte	3~5 byte	1~32byte	0/1/2/3 byte
preamble(0x55 或者 0xaa)	addr	payload	crc(包含 addr 和 payload)

preamble: 地址第一个 **bit** 为 **0**，取 **0xaa**；地址第一个 **bit** 为 **1**，取 **0x55**。

增强型：在地址和 **payload** 之间插入 **signal(9bit)**数据。

1 byte	3~5 byte	9bit	1~32byte	0/1/2/3 byte
preamble(0x55 或者 0xaa)	addr	signal	payload	crc(包含 addr 和 payload)

signal 结构:

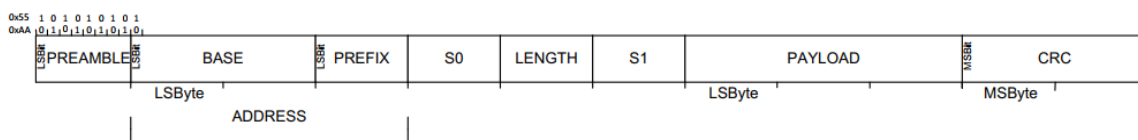
6bit	2bit	1bit
数据长度标识(动态 payload)	PID 标识 (判断是否是重发包)	NO_ACK 标识 (tx 完是否需要 ack)

数据长度 **6bit** 最大包长发送可支持 **63byte**，接收最大包长可支持 **64byte**。

nrf2401 无白化，**crc** 的作用域包括 **addr+payload+signal**。

5.1.3 nrf52 兼容帧结构

nrf52 帧结构如下图所示:



nrf52 帧结构

当 **s0=0**，**length=0**，**s1=0** 时，和 **nrf2401** 普通型帧结构一样。

当 **s0=0**，**length=6**，**s1=3** 时，和 **nrf2401** 增强型帧结构一样。**payload** 长度最大 **32byte**。

当 **s0=0**，**length=8**，**s1=3** 时，长包模式。**payload** 长度最大 **255byte**。

空中 **bit** 序：大小端模式可配

crc 作用域: **addr(可配)+s0+length+s1+payload**

长包模式下, **PAN1070 signal** 结构如下:

8bit	2bit	1bit
数据长度标识(动态 payload)	PID 标识 (判断是否是重发包)	NO_ACK 标识 (tx 完是否需要 ack)

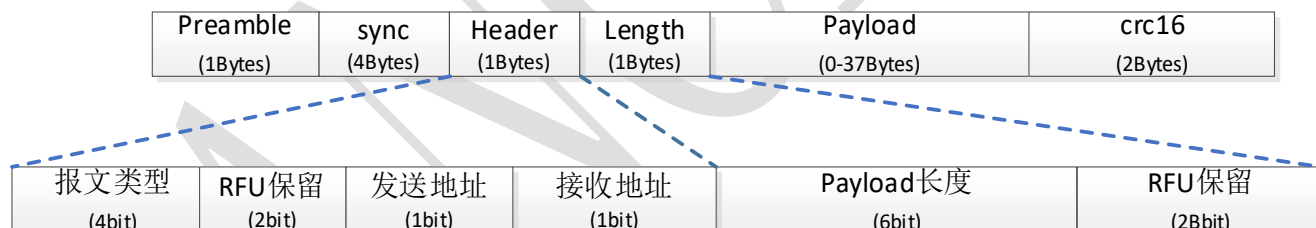
5.1.4 自定义帧结构

chip_mode 字段为“PRF_CHIP_MODE_SEL_NRF”模式时, 帧结构如下:

Preamble (1Bytes)	Addr (3-5Bytes)	Header1 (0-1Bytes)	Header0 (0-1Bytes)	Length (0-1Bytes)	Payload (0-255Bytes)	Crc24/crc16/crc8 (3/2/1Bytes)
----------------------	--------------------	-----------------------	-----------------------	----------------------	-------------------------	----------------------------------

header0、**header1**、**length** 为可选字段, 可以适配 **BLE** 和 **NRF52** 不同模式的帧结构,

例如 **BLE** 广播帧结构:



蓝牙广播中的报头和长度对应 **PRF_MODE_NORMAL_M1** 模式中的 **header0** 和 **length**,

header1 没有。

5.1.5 B_250k 帧结构

Preamble (2-24Bytes)	sync (2/4Bytes)	Length (0-1Bytes)	Payload (0-255Bytes)	crc16 (2Bytes)
-------------------------	--------------------	----------------------	-------------------------	-------------------

使用 **nrf** 普通型帧结构进行兼容。 **B_250K** 中 **payload** 长度为 8bit, 而且最大 **payload**(包含 **addr** 在内)长度为 255.另外, **length** 字段固定为 1 个字节。

Preamble 长度有: 2/3/4/6/8/12/16/24byte, 帧速率为 250k。