



# PAN312x 开发套件使用手册

发布 0.2.0



磐启微电子 PAN312x 项目组  
2025 年 08 月 06 日

# Table of contents

|        |                              |     |
|--------|------------------------------|-----|
| 1      | 快速入门                         | 1   |
| 1.1    | SDK 快速入门                     | 1   |
| 1.1.1  | 1 概述                         | 1   |
| 1.1.2  | 2 PAN312x EVB 硬件资源介绍         | 1   |
| 1.1.3  | 3 PAN312x SDK 开发环境确认         | 1   |
| 2      | 硬件资料                         | 5   |
| 2.1    | PAN312x Evaluation Board 介绍  | 5   |
| 2.1.1  | 1 概述                         | 5   |
| 2.1.2  | 2 使用说明                       | 5   |
| 2.2    | PAN312x 硬件参考设计               | 11  |
| 2.2.1  | 1 概述                         | 11  |
| 2.2.2  | 2 典型应用原理图                    | 11  |
| 2.2.3  | 3 PCB 设计建议                   | 15  |
| 2.2.4  | 4 RF 基本性能                    | 18  |
| 3      | 演示例程                         | 21  |
| 3.1    | 基础例程                         | 21  |
| 3.1.1  | SDK_CTK_PacketRx             | 21  |
| 3.1.2  | SDK_CTK_PacketTx             | 30  |
| 3.1.3  | SDK_FixedLongPacketRx        | 37  |
| 3.1.4  | SDK_FixedLongPacketTx        | 42  |
| 3.1.5  | SDK_FixedPacketRx            | 45  |
| 3.1.6  | SDK_FixedPacketTx            | 50  |
| 3.1.7  | SDK_GpioOutPutOclk           | 55  |
| 3.1.8  | SDK_SemiAutomaticDutyCycleRx | 56  |
| 3.1.9  | SDK_SemiAutomaticDutyCycleTx | 60  |
| 3.1.10 | SDK_TestDemo                 | 63  |
| 3.1.11 | SDK_VariableLongPacketRx     | 67  |
| 3.1.12 | SDK_VariableLongPacketTx     | 72  |
| 3.1.13 | SDK_VariablePacketRx         | 74  |
| 3.1.14 | SDK_VariablePacketTx         | 81  |
| 4      | 开发指南                         | 87  |
| 5      | 量产测试                         | 89  |
| 6      | 开发工具                         | 91  |
| 6.1    | PANCHIP-RF-CTK               | 91  |
| 6.1.1  | 基本 UI 介绍                     | 91  |
| 6.1.2  | 新建工程                         | 93  |
| 6.1.3  | 芯片配置解析                       | 93  |
| 7      | 其他文档                         | 101 |
| 8      | 应用文档                         | 103 |
| 8.1    | PAN312x 寄存器说明                | 103 |

|       |                                   |     |
|-------|-----------------------------------|-----|
| 8.1.1 | 1 COMMAND 寄存器描述 . . . . .         | 103 |
| 8.1.2 | 2 特殊寄存器描述 . . . . .               | 105 |
| 8.1.3 | 3 PROPERTY GROUP1 寄存器描述 . . . . . | 106 |
| 8.1.4 | 4 PROPERTY GROUP2 寄存器描述 . . . . . | 106 |
| 8.1.5 | 5 PROPERTY GROUP3 寄存器描述 . . . . . | 106 |
| 8.1.6 | 6 PROPERTY GROUP4 寄存器描述 . . . . . | 107 |
| 8.1.7 | 7 PROPERTY GROUP5 寄存器描述 . . . . . | 108 |
| 9     | <b>更新日志</b> . . . . .             | 111 |
| 9.1   | PAN312x DK v0.2.0 . . . . .       | 111 |
| 9.1.1 | 1. SDK . . . . .                  | 111 |
| 9.1.2 | 3. DOC . . . . .                  | 111 |
| 9.1.3 | 4. TOOLS . . . . .                | 112 |
| 9.1.4 | 5. 已知问题 . . . . .                 | 112 |
| 9.2   | PAN312x DK v0.1.0 . . . . .       | 112 |
| 9.2.1 | 1. SDK . . . . .                  | 112 |
| 9.2.2 | 3. DOC . . . . .                  | 113 |
| 9.2.3 | 4. TOOLS . . . . .                | 113 |
| 9.2.4 | 5. 已知问题 . . . . .                 | 113 |
| 9.3   | PAN312x DK v0.0.0 . . . . .       | 113 |
| 9.3.1 | 1. SDK . . . . .                  | 113 |
| 9.3.2 | 2. HDK . . . . .                  | 114 |
| 9.3.3 | 3. DOC . . . . .                  | 114 |
| 9.3.4 | 4. TOOLS . . . . .                | 115 |
| 9.3.5 | 5. 已知问题 . . . . .                 | 115 |

## Chapter 1

# 快速入门

## 1.1 SDK 快速入门

### 1.1.1 1 概述

本文是 PAN312x Radio 开发的快速入门指引，旨在帮助使用者快速入门 PAN312x SDK 的开发，搭建软硬件环境，并编译、运行、调试一个例程。

### 1.1.2 2 PAN312x EVB 硬件资源介绍

在开始使用 PAN312x SDK 之前，我们建议您先阅读[PAN312x Evaluation\\_Board 硬件资源介绍](#)文档，熟悉 Evaluation\_Board 开发板的基本使用方法。

### 1.1.3 3 PAN312x SDK 开发环境确认

#### 3.1 PC 环境检查

目前 PAN312x 提供的编译工具链只支持 Windows 7 及以上版本的 64 位操作系统，请确保您的开发环境满足此要求。

您可以按照如下操作快速确认您的 PC 是否满足要求：

- 1、按快捷键 Win + R，在弹出的运行对话框中输入 dxdiag 并回车：
- 2、稍等片刻，在弹出的 DirectX 诊断工具对话框中，可以看到当前 PC 的系统信息：

#### 3.2 获取并配置 SDK 及编译工具链环境

- 1、请确认您已经获取到如下 1 个必要的压缩包：
  - pan312x-dk-v<x>.<y>.<z>.zip: PAN312x Development Kit 开发套件
- 2、请确保 PC 上已经正常解压缩了 PAN312x Development Kit 开发包：

#### 3.5 快速编译烧录运行一个简单的例程

下面我们以 SDK\_FixedPacketTx 例程为例，演示如何编译烧录运行一个 demo 工程

- 3.5.1. 硬件环境准备 1、将 PAN312x EVB 板和 PAN312x 模组通过 spi 接口相连接；
- 2、将 PAN312x EVB 板通过 usb 接口连接到 pc；



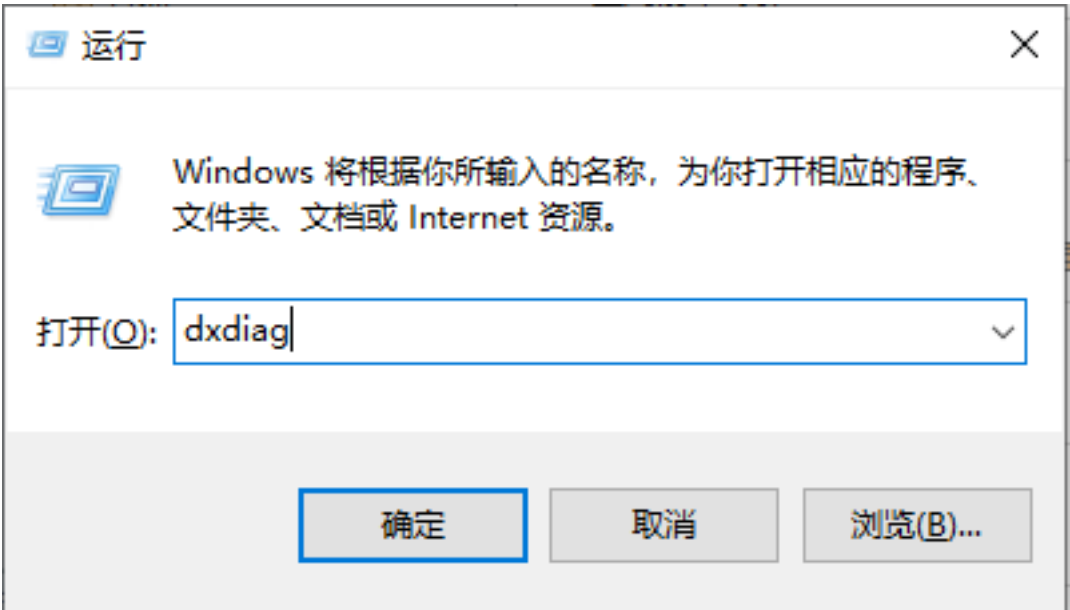


图 1: 运行 dxdiag

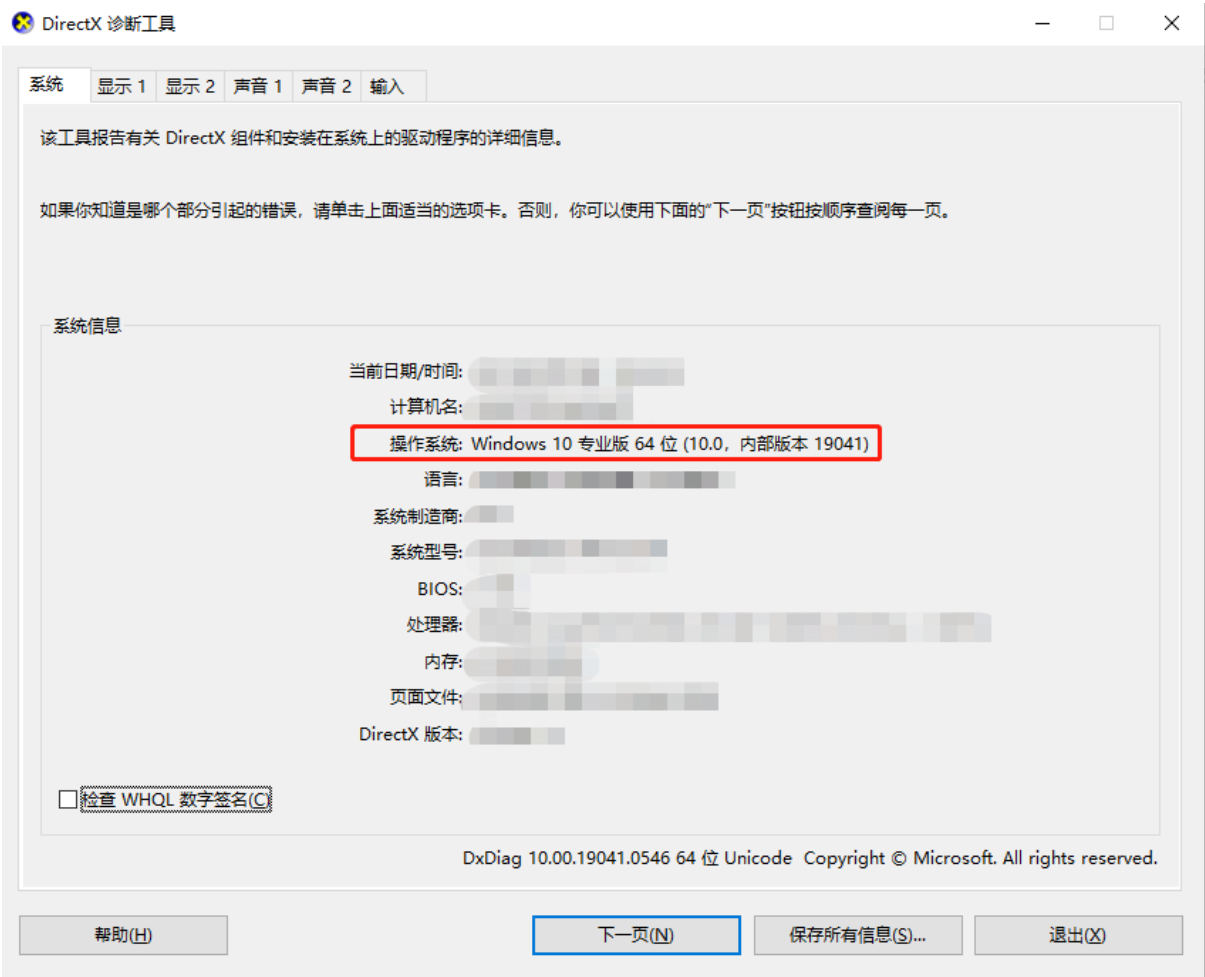


图 2: DirectX 诊断工具对话框

| 名称           | 修改日期            | 类型            | 大小   |
|--------------|-----------------|---------------|------|
| 01_SDK       | 2023/3/6 16:48  | 文件夹           |      |
| 02_HDK       | 2023/2/23 16:46 | 文件夹           |      |
| 03_DOC       | 2023/3/6 16:48  | 文件夹           |      |
| 04_TOOLS     | 2023/3/6 16:48  | 文件夹           |      |
| CHANGELOG.md | 2023/2/23 17:15 | Markdown File | 1 KB |
| README.md    | 2022/7/22 14:18 | Markdown File | 1 KB |

图 3: PAN312x DK 开发包主目录结构

3.5.2 编译 1、进入目录 01\_SDK\PAN312x\_DK\_Develop\Project\MDK-ARM(AC5)，双击 project.uvprojx，如下图所示：



图 4: 打开 PAN312x 开发工程

2、打开 project 工程，选择 SDK\_FixedPacketTx 例程，如下图所示

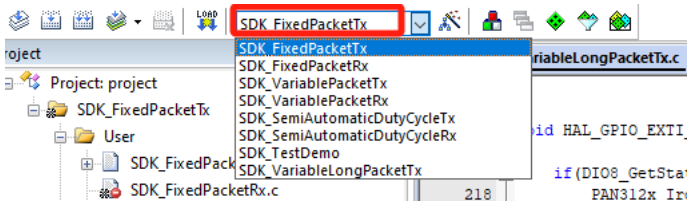


图 5: 打开 hello\_world 工程

(2)、选择 build 按钮，编译整个工程，选择 download 按钮，下载程序，如下图所示：

3.5.4 运行 程序运行后，串口会输出如下信息：

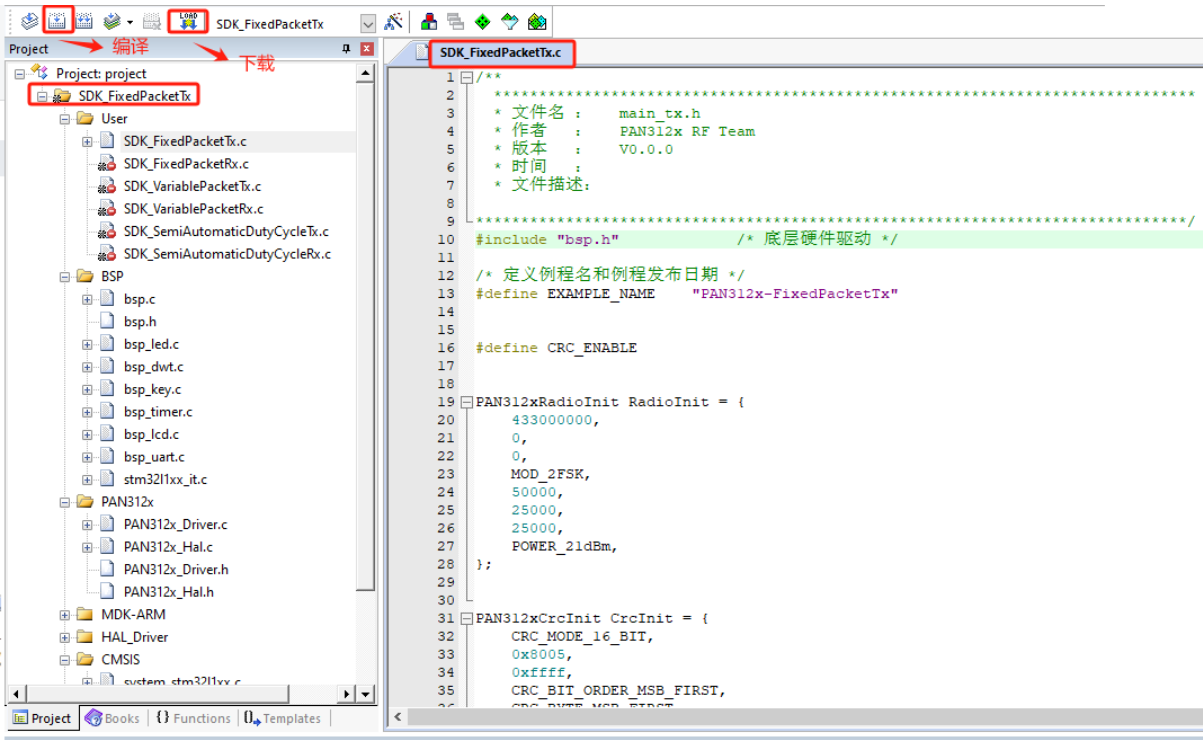


图 6: 构建 SDK\_FixedPacketTx 例程

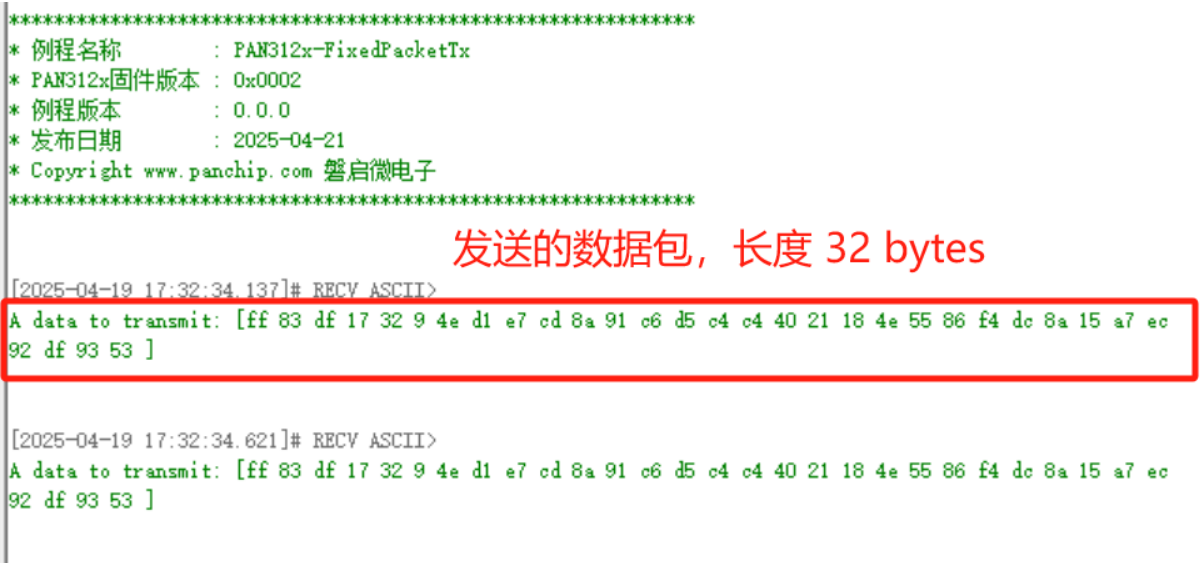


图 7: 串口输出信息

## Chapter 2

# 硬件资料

## 2.1 PAN312x Evaluation Board 介绍

### 2.1.1 1 概述

#### 2.1 Evaluation Board 介绍

本文档是 Evaluation Board 开发板的介绍，包括相关板级硬件模块、各模块在 Evaluation Board 板的位置以及对应电路原理图，旨在帮助开发者快速了解 Evaluation Board 开发板。

Evaluation Board 开发板需配合待测模块使用，Evaluation Board 开发板对外接口提供了 SPI (CSN、SCK、MISO、MOSI)、UART (TX、RX) 和 5 个 GPIO。

Evaluation Board 开发板包含了很多人机交互模块，包括：电源管理系统、USB\_Micro 转串口模块、无源蜂鸣器、独立按键、状态指示 LED、液晶显示屏等。

#### 2.1 Evaluation Board 外观

### 2.1.2 2 使用说明

#### 2.1 供电及功耗测试

Evaluation Board 开发板支持 3 节干电池和 Micro-USB 两种供电方式，当两种电源都提供时选择 USB 供电。Evaluation Board 开发板的电源通过自锁开关 K1 控制上下电。MCU 和待测模块的功耗可以通过 J3 和 J5 测试，如果不需要测试功耗可以使用跳线帽将其短接。

### 2.2 电源电压检测

Evaluation Board 开发板提供了电源电压检测接口，当使用干电池电压低时可以及时提示更换电池。使用的是 PA0 采集电源电压的值。

### 2.3 通信

J1 为 Evaluation Board 开发板对外通信的 Micro-USB 口，MCU 引脚 PA9/TX、PA10/RX 通过 USB 转串口芯片 CHN340N 与电脑通信。

### 2.4 无源蜂鸣器

Evaluation Board 开发板提供无源蜂鸣器外设，可以通过 MCU 引脚 PA1 输出 PWM 信号进行控制。

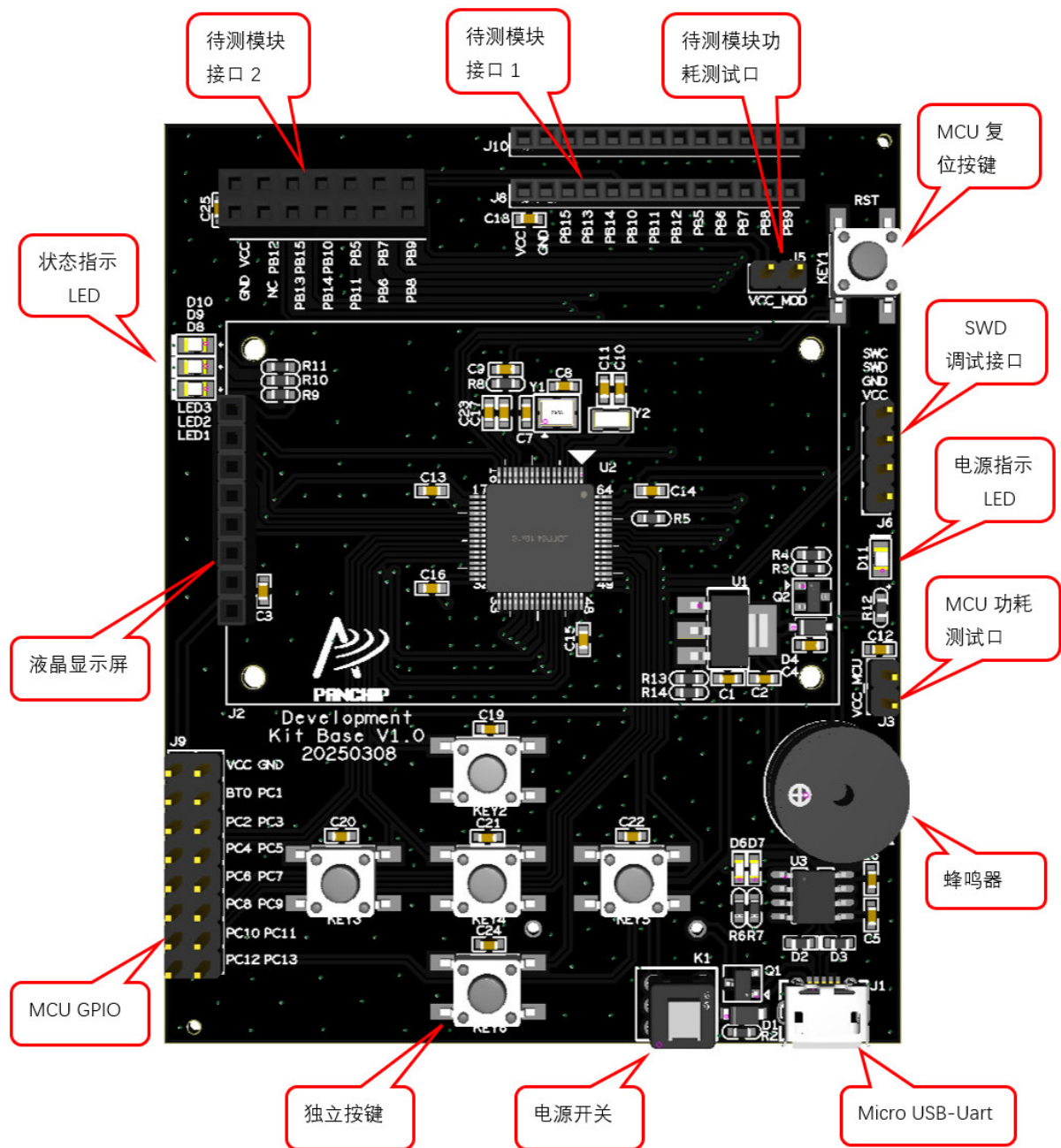


图 1: PAN312x Evaluation Board 外观

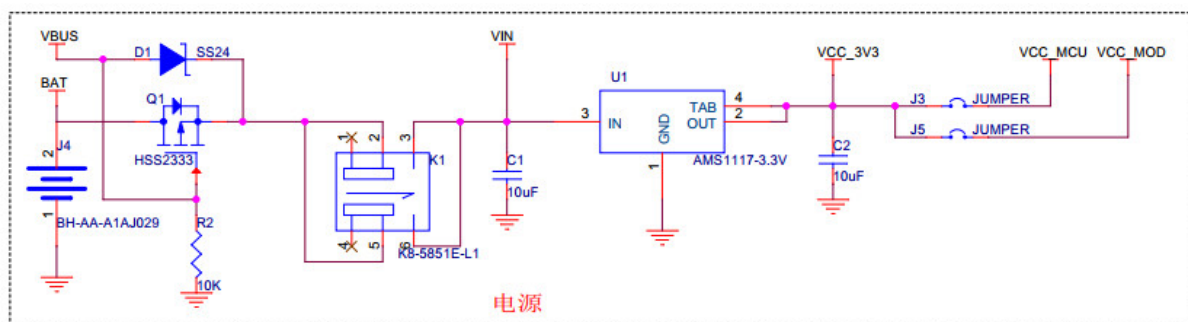


图 2: PAN312x Evaluation Board 供电方式



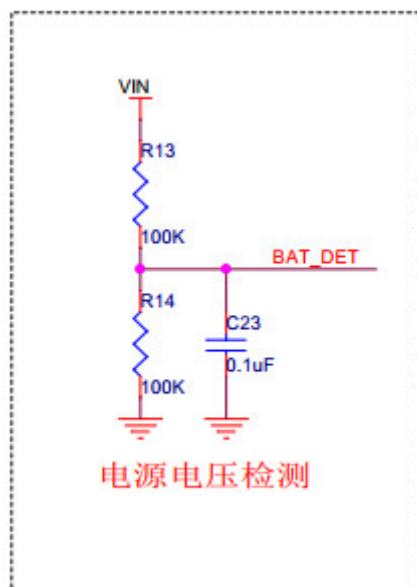


图 3: PAN312x Evaluation Board 电源电压检测接口

## 2.5 LED 指示灯

Evaluation Board 开发板提供三组 LED 指示灯，一组时对外串口通信指示，一组电源指示，还有一组用户自定义指示灯。用户自定义指示灯通过引脚 PA11、PA12、PA8 控制，分别对应 LED1、LED2、LED3，输出低电平点亮 LED。

## 2.6 液晶显示屏

Evaluation Board 开发板配有液晶显示屏来提供人机交互界面，控制引脚对应情况如下：PA2-LCD\_BL、PA3-LCD\_RST、PA4-LCD\_CS、PA5-LCD\_SCK、PA6-LCD\_DC、PA7-LCD\_MOSI；

## 2.7 独立按键

Evaluation Board 开发板提供 5 个按键配合液晶显示屏进行菜单选择，控制引脚对应情况如下：PB0-KEY\_LEFT、PB1-KEY\_OK、PB2-KEY\_UP、PB3-KEY\_DOWN、PB4-KEY\_RIGHT。

## 2.8 SPI 数字通信接口

Evaluation Board 开发板和待测模块的连接有两种接口，一种是 2.54mm\_2X7 排母，另一种是 2mm\_1X13 排母。对外接口提供了一组 SPI (CSN、SCK、MISO、MOSI)、一路 UART (TX、RX) 和 5 个 GPIO。分别是 PB12-CSN、PB13-SCK、PB14-MISO、PB15-MOSI、PB10/TX、PB11/RX、PB5、PB6、PB7、PB8、PB9。

## 2.9 预留 GPIO

Evaluation Board 开发板提供了 16 个预留的 GPIO 可以扩展功能和 Debug。分别是 PC1、PC2、PC3、PC4、PC5、PC6、PC7、PC8、PC9、PC10、PC11、PC12、PC13。

## 2.9 SWD 下载接口

Evaluation Board 开发板提供了两种下载方式，一种是通过 SWD 下载，一种是串口下载。默认使用 J-Link 通过 SWD 下载。如果需要使用串口下载，需要将 J9 的 BOOT0 通过跳线帽和 VCC\_3V3 短接，

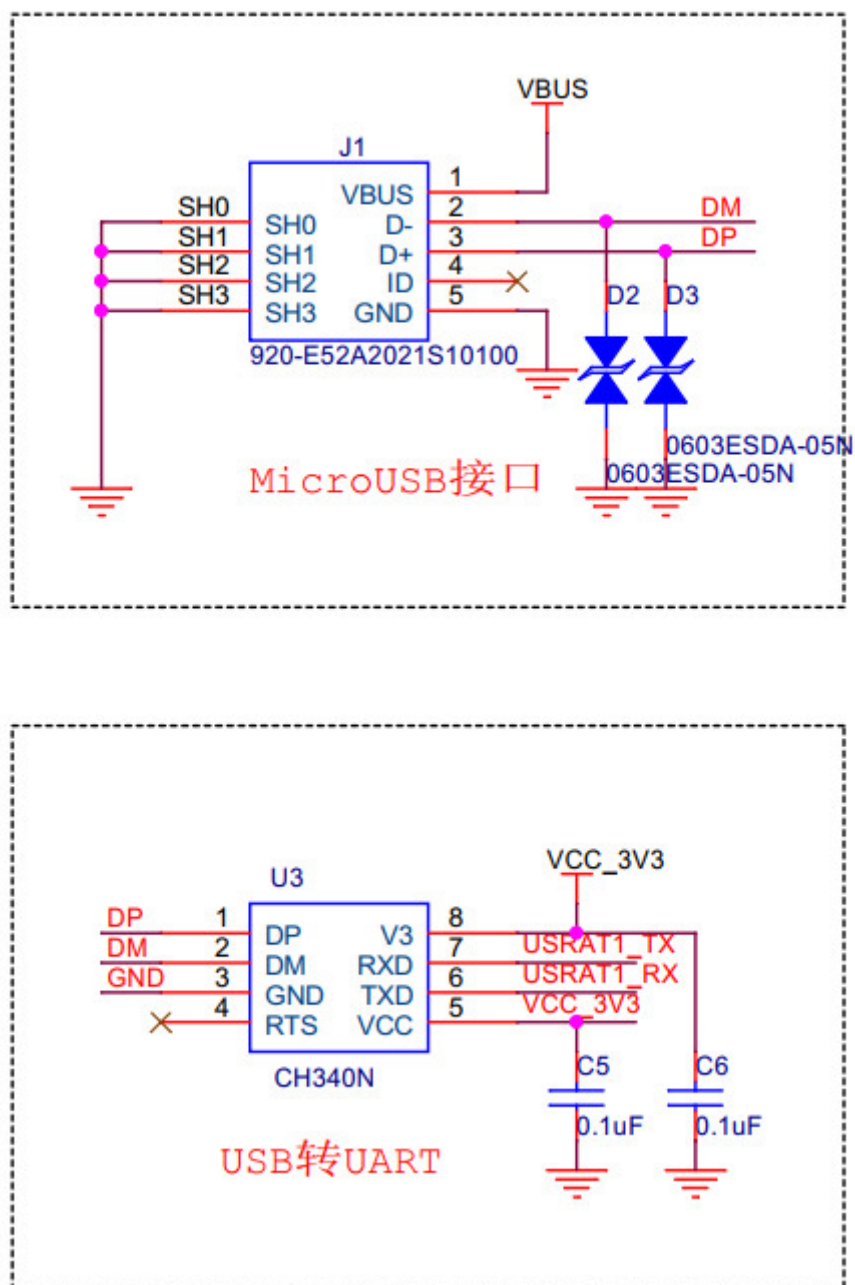


图 4: PAN312x Evaluation Board 通信接口

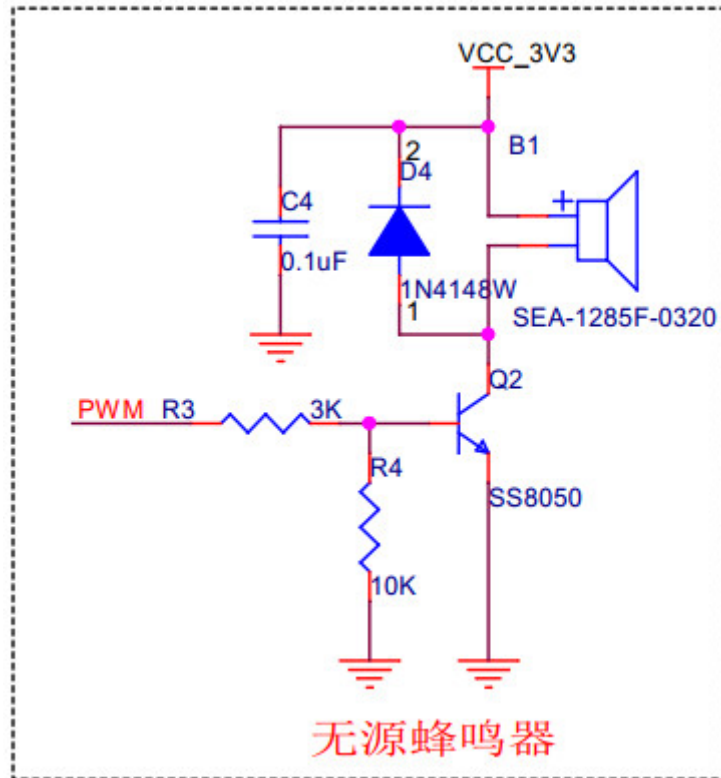


图 5: PAN312x Evaluation Board 无源蜂鸣器

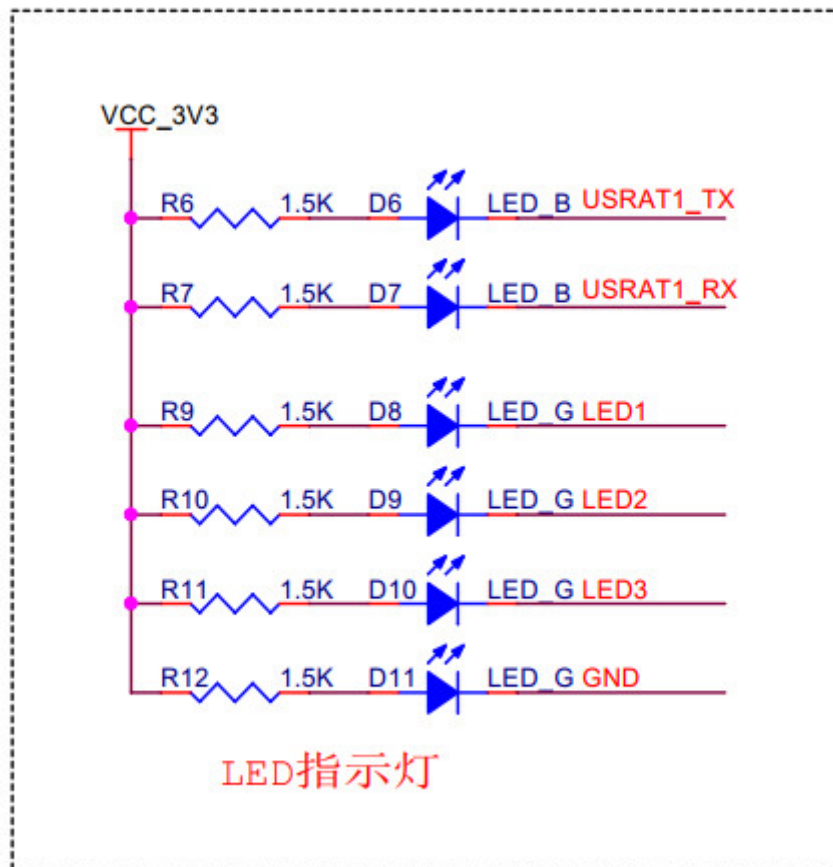


图 6: PAN312x Evaluation Board led 指示灯

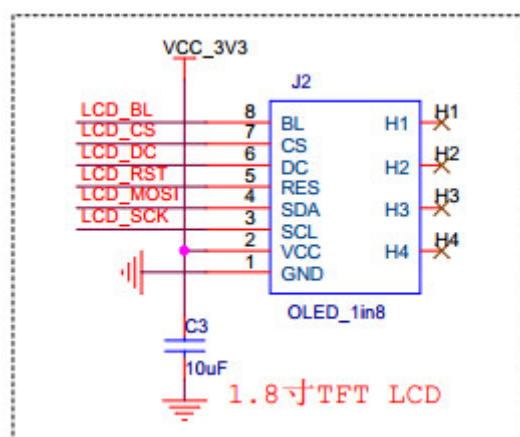


图 7: PAN312x Evaluation Board 液晶屏显示

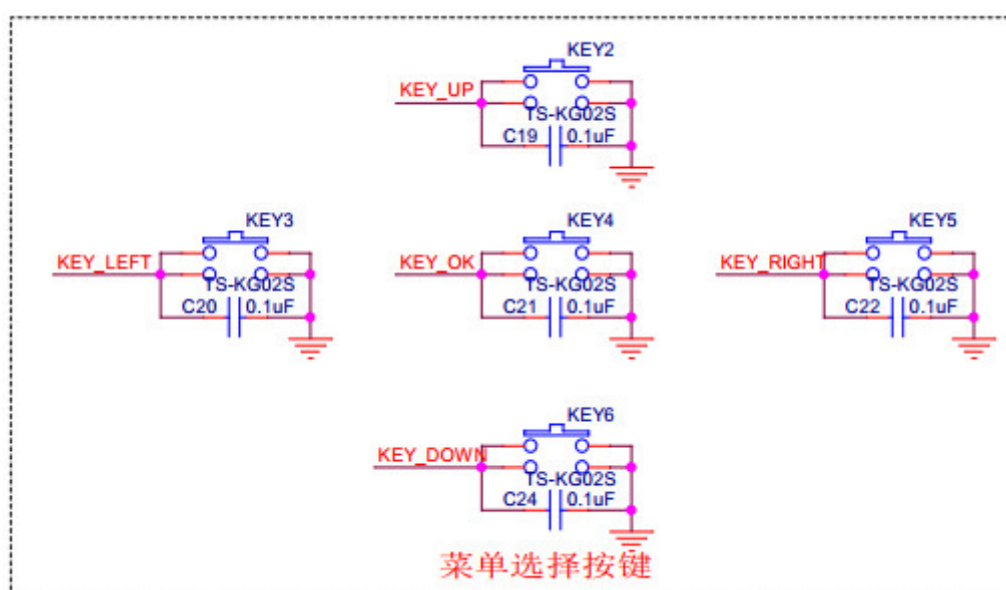


图 8: PAN312x Evaluation Board 按键

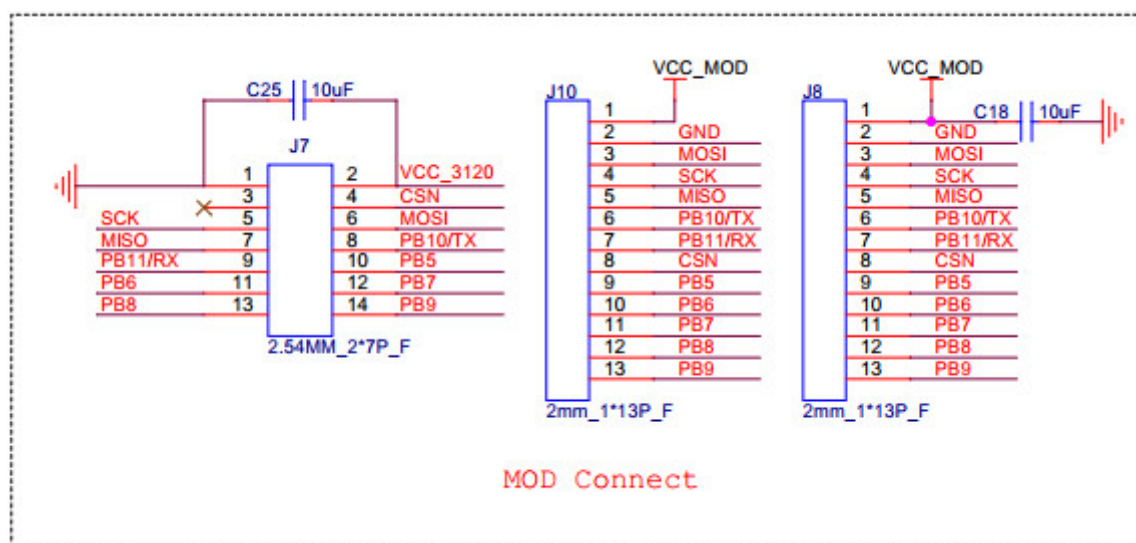


图 9: PAN312x Evaluation Board SPI 数字通信接口

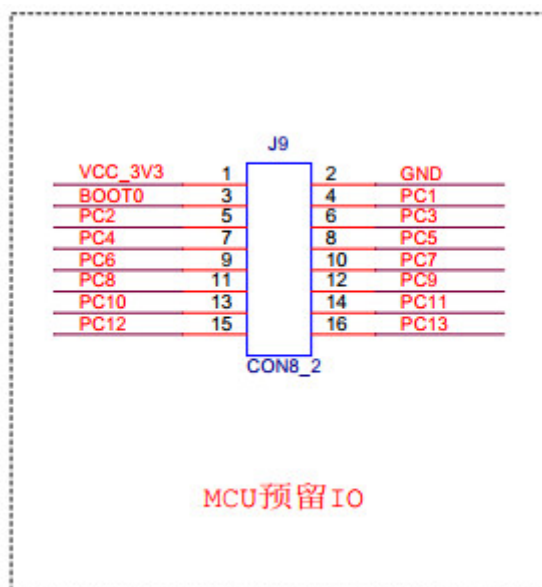


图 10: PAN312x Evaluation Board 预留 GPIO

按住 KEY2 上电进入串口下载模式

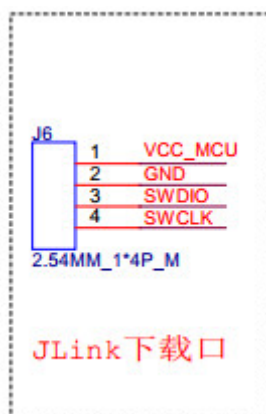


图 11: PAN312x Evaluation Board SWD 下载接口

## 2.2 PAN312x 硬件参考设计

### 2.2.1 1 概述

本文档主要介绍 PAN312x 芯片方案的硬件原理图设计、PCB 设计建议、静电防护、RF 性能等，以及提供一些外围电路的硬件设计方法。

### 2.2.2 2 典型应用原理图

#### 2.1 直连 (Direct Tie) 原理图

如下图，电路系统由 DC-DC 电路、晶振电路、天线匹配网络和一些电容组成。GPIO3/NRST、CSN、SCK、MISO、MOSI 信号需要与控制芯片相连。

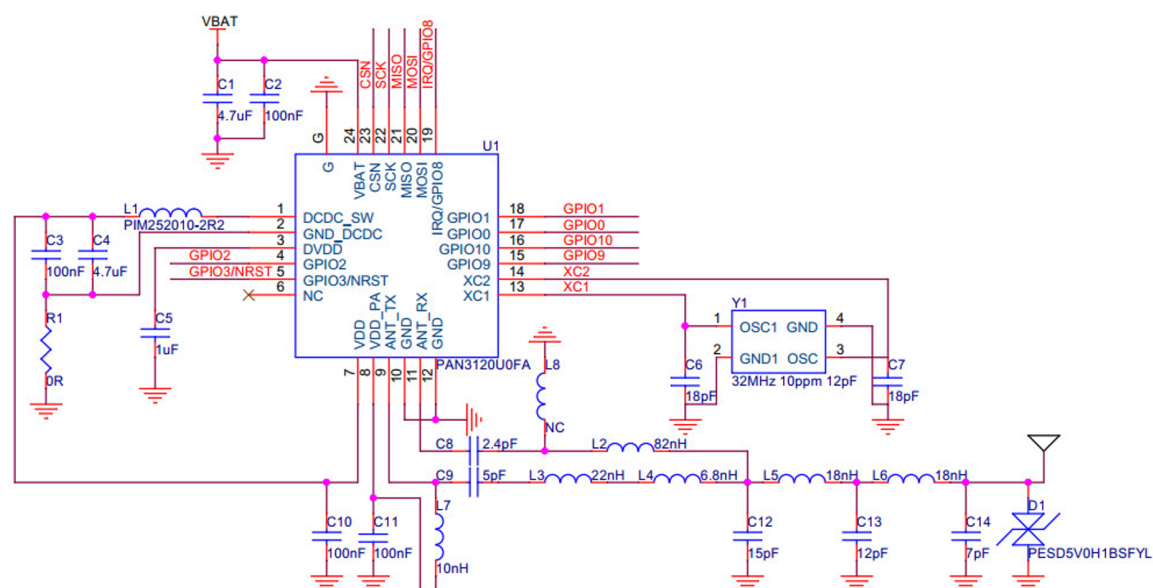


图 12: 图 2-1. 20dBm 直连 (Direct Tie) DCDC 模式典型应用原理图

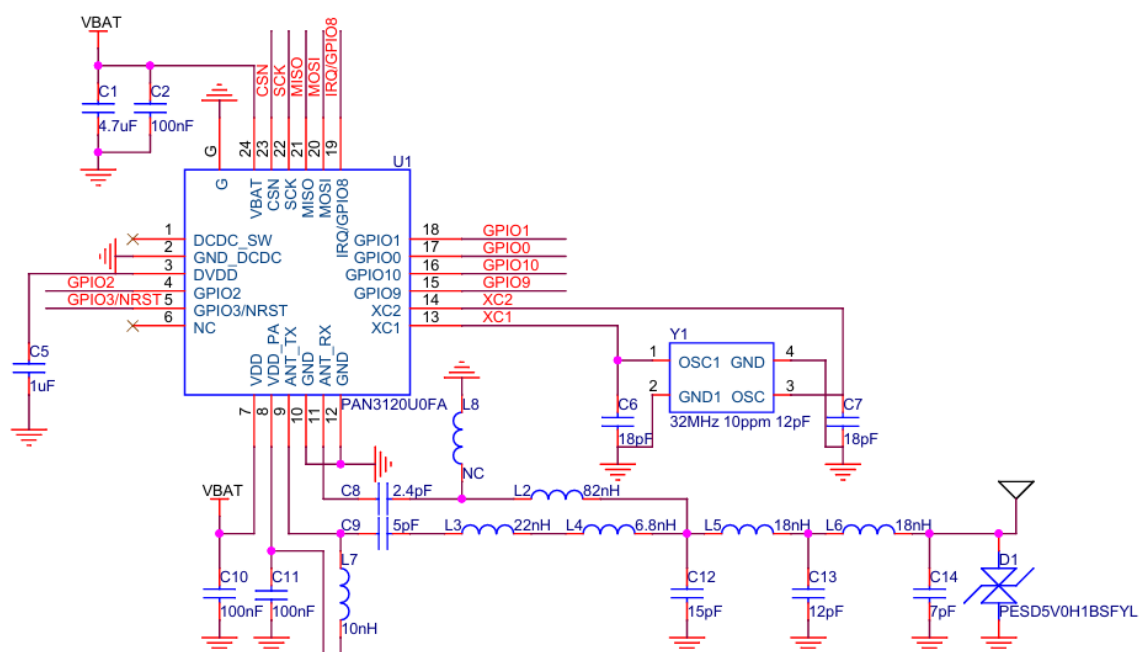


图 13: 图 2-1. 20dBm 直连 (Direct Tie) LDO 模式典型应用原理图



## 2.2 典型应用物料清单

射频匹配网络推荐优先使用 0603 封装电感, 描述中的型号描述为电感型号, 电容不做型号推荐。

| 标号  | 描述                                   | 元件值                |                    |                    |                    |                    |
|-----|--------------------------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
|     |                                      | 320 MHz<br>+20 dBm | 433 MHz<br>+20 dBm | 490 MHz<br>+20 dBm | 868 MHz<br>+20 dBm | 915 MHz<br>+20 dBm |
| C1  | ±5%, 0402 NP0, 50 V                  | 4.7uF              |                    |                    |                    |                    |
| C2  | ±5%, 0402 NP0, 50 V                  | 100nF              |                    |                    |                    |                    |
| C3  | ±5%, 0402 NP0, 50 V                  | 100nF              |                    |                    |                    |                    |
| C4  | ±5%, 0402 NP0, 50 V                  | 4.7uF              |                    |                    |                    |                    |
| C5  | ±5%, 0402 NP0, 50 V                  | 1uF                |                    |                    |                    |                    |
| C6  | ±5%, 0402 NP0, 50 V                  | 18pF               |                    |                    |                    |                    |
| C7  | ±5%, 0402 NP0, 50 V                  | 18pF               |                    |                    |                    |                    |
| C8  | ±5%, 0402 NP0, 50 V                  | 0Ω                 | 3pF                | 2.4pF              | 0Ω                 | 0Ω                 |
| C9  | ±5%, 0402 NP0, 50 V                  | 5pF                | 4.3pF              | 3.6pF              | 2.7pF              | 3pF                |
| C10 | ±5%, 0402 NP0, 50 V                  | 100nF              |                    |                    |                    |                    |
| C11 | ±5%, 0402 NP0, 50 V                  | 100nF              |                    |                    |                    |                    |
| C12 | ±5%, 0402 NP0, 50 V                  | 12pF               | 15pF               | 15pF               | 6pF                | 5pF                |
| C13 | ±5%, 0402 NP0, 50 V                  | 12pF               | 12pF               | 12pF               | 5pF                | 5pF                |
| C14 | ±5%, 0402 NP0, 50 V                  | 10pF               | 6.2pF              | 5pF                | 4.3pF              | 4.3pF              |
| L1  | ±5%, 2520 功率电感<br>PIM252010-2R2MTS00 | 2.2uH              |                    |                    |                    |                    |
| L2  | ±5%, 0402 SDCL1005C系列                | 5.6pF              | 82nH               | 68nH               | 1.8pF              | 2pF                |
| L3  | ±5%, 0402 SDCL1005C系列                | 47nH               | 27nH               | 22nH               | 10nH               | 7.5nH              |
| L4  | ±5%, 0402 SDCL1005C系列                | 12nH               | 6.8nH              | 7.5nH              | 3.9nH              | 3nH                |
| L5  | ±5%, 0402 SDCL1005C系列                | 39nH               | 18nH               | 15nH               | 9.1nH              | 9.1nH              |
| L6  | ±5%, 0402 SDCL1005C系列                | 39nH               | 22nH               | 18nH               | 9.1nH              | 10nH               |
| L7  | ±5%, 0402 SDCL1005C系列                | 22nH               | 10nH               | 10nH               | 10nH               | 10nH               |
| L8  | ±5%, 0402 SDCL1005C系列                | 3.9nH              | NC                 | NC                 | 9.1nH              | 8.2nH              |
| R1  | ±5%, 0402电阻                          | 0Ω                 |                    |                    |                    |                    |
| Y1  | ±10ppm, SMD_3225                     | 32MHz              |                    |                    |                    |                    |
| D1  | DSN0603-2, ESD保护元件<br>PESD5V0H1BSFYL |                    |                    |                    |                    |                    |
| U1  | PAN3120, 超低功耗Sub-1GHz射<br>频收发器       |                    |                    |                    |                    |                    |

图 14: 图 2-2. 20dBm 直连 (Direct Tie) 0402 封装典型应用物料清单

## 2.3 电源

- VBAT 为芯片电源脚, 要求供电能力不小于 200mA, 供电范围为 1.8V~3.8V。
- DVDD、VDD、VDD\_PA、VBAT 等电源相关引脚需要至少预留 1 个电容, 预留一大一小 2 个电容更佳。
- DVDD 电容 1uF, VDD、VBAT 引脚电容不小于 4.7 F, 应尽可能靠近芯片引脚摆放, 电容焊盘和芯片焊盘之间最大距离不超过 5mm。

\* 注: DC-DC ON 模式下工作电压范围 2V~3.8V

### 2.3.1 DC-DC

- DC-DC 芯片外围电路
1. 芯片外围电路组成为: L1、C3、C4。

| 标号  | 描述                                   | 元件值                |                    |                    |                    |                    |
|-----|--------------------------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
|     |                                      | 320 MHz<br>+20 dBm | 433 MHz<br>+20 dBm | 490 MHz<br>+20 dBm | 868 MHz<br>+20 dBm | 915 MHz<br>+20 dBm |
| C1  | ±5%, 0402 NP0, 50 V                  | 4.7uF              |                    |                    |                    |                    |
| C2  | ±5%, 0402 NP0, 50 V                  | 100nF              |                    |                    |                    |                    |
| C3  | ±5%, 0402 NP0, 50 V                  | 100nF              |                    |                    |                    |                    |
| C4  | ±5%, 0402 NP0, 50 V                  | 4.7uF              |                    |                    |                    |                    |
| C5  | ±5%, 0402 NP0, 50 V                  | 1uF                |                    |                    |                    |                    |
| C6  | ±5%, 0402 NP0, 50 V                  | 18pF               |                    |                    |                    |                    |
| C7  | ±5%, 0402 NP0, 50 V                  | 18pF               |                    |                    |                    |                    |
| C8  | ±5%, 0402 NP0, 50 V                  | -                  | 2.4pF              | 6pF                | 0Ω                 | 0Ω                 |
| C9  | ±5%, 0402 NP0, 50 V                  | -                  | 5pF                | 3.3pF              | 2.4pF              | 2.4pF              |
| C10 | ±5%, 0402 NP0, 50 V                  | 100nF              |                    |                    |                    |                    |
| C11 | ±5%, 0402 NP0, 50 V                  | 100nF              |                    |                    |                    |                    |
| C12 | ±5%, 0402 NP0, 50 V                  | -                  | 15pF               | 10pF               | 6pF                | 5pF                |
| C13 | ±5%, 0402 NP0, 50 V                  | -                  | 10pF               | 10pF               | 5pF                | 5pF                |
| C14 | ±5%, 0402 NP0, 50 V                  | -                  | 10pF               | 9pF                | 4.3pF              | 4.3pF              |
| L1  | ±5%, 2520 功率电感<br>PIM252010-2R2MTS00 | 2.2uH              |                    |                    |                    |                    |
| L2  | ±5%, 0603 SDCL1608C系列                | -                  | 82nH               | 47nH               | 1.8pF              | 1.8pF              |
| L3  | ±5%, 0603 SDCL1608C系列                | -                  | 22nH               | 22nH               | 10nH               | 8.2nH              |
| L4  | ±5%, 0603 SDCL1608C系列                | -                  | 6.8nH              | 6.8nH              | 4.7nH              | 5.6nH              |
| L5  | ±5%, 0603 SDCL1608C系列                | -                  | 18nH               | 15nH               | 8.2nH              | 6.8nH              |
| L6  | ±5%, 0603 SDCL1608C系列                | -                  | 18nH               | 18nH               | 6.8nH              | 6.8nH              |
| L7  | ±5%, 0603 SDCL1608C系列                | -                  | 10nH               | 10nH               | 10nH               | 10nH               |
| L8  | ±5%, 0603 SDCL1608C系列                | -                  | NC                 | NC                 | 10nH               | 8.2nH              |
| R1  | ±5%, 0402电阻                          | 0Ω                 |                    |                    |                    |                    |
| Y1  | ±10ppm, SMD 3225                     | 32MHz              |                    |                    |                    |                    |
| D1  | DSN0603-2, ESD保护元件<br>PESD5V0H1BSFYL |                    |                    |                    |                    |                    |
| U1  | PAN3120, 超低功耗Sub-1GHz射<br>频收发器       |                    |                    |                    |                    |                    |

图 15: 图 2-3. 20dBm 直连 (Direct Tie) 0603 封装典型应用物料清单

| Name   | Type  | Voltage(V)   | Note                          |
|--------|-------|--------------|-------------------------------|
| DVDD   | Power | 1.2(typ)     | 数字电源LDO输出                     |
| VDD    | Power | 1.5(typ)~3.8 | 模拟电源, DCDC模式连接VFB, LDO模式连接总电源 |
| VDD_PA | Power | 0.8~2.9      | PA电源LDO输出                     |
| VBAT   | Power | 1.8~3.8      | 模拟电源, 连接总电源                   |

图 16: 电源引脚描述表

2. L1 推荐型号: PIM252010-2R2MTS00, 参数详见章节 2.2-典型应用物料清单。选择功率电感, 2.2 H, **额定电流不小于 800mA**, DCR **不超过 80mΩ**, 未满足要求在 DC-DC 模式可能会造成 RF 功能异常。

3. DCR 过大会影响 BUCK 效率, 能量会转化成热量损耗掉, DC-DC 输出的驱动电流是有限的, 效率越低, 能够供给到芯片的有效能量就越少。

- DC-DC 的两种工作模式:

1. 开启 DC-DC 模式可以降低系统功耗。

2. 开启 LDO (Bypass) 模式后芯片内部将 VBAT 连接到 DCDC\_SW, 这时 DCDC\_SW 处的 2.2uH 电感作用为一段导体, 可以用 0Ω 电阻替换。

3. DC-DC、LDO 两种模式不能同时开启。

4. 在不考虑功耗的前提下, 可将 VDD 直接连接到 VBAT, 此时应将电源模式设置为 LDO 模式。

- DC-DC 相关引脚说明:

1. VBAT 为 DC-DC 的供电引脚。

2. DCDC\_SW 为 DC-DC 的功率开关 (P-MOS) 漏极输出引脚, 功率电感应靠近该引脚放置。

3. VDD 为 DC-DC 的反馈引脚, 电容应靠近该引脚放置。

4. GND\_DCDC 为 DC-DC 电源的 GND 引脚。

\* 注: DC/DC 转换器仅支持对低功率 PA 电路提供电源, 用户选择用高功率 PA 时, 需要用外部电源供电。

**2.3.2 DVDD** DVDD 需要**放置 1uF 电容**。电容最大不应超过 1uF, 否则会影响芯片正常启动, 电容应靠近该引脚放置。

## 2.4 晶振

### 2.4.1 32M 晶振

- 32M 晶振电路由 Y1, C6, C7 构成, 为芯片提供稳定的外部 32MHz 时钟。推荐型号为: X322532MOB4SI 32MHz 12pF  $\pm 10\text{ppm}$ , ESR 小于 50Ω, 晶振的频率温度 (-20~70°C) 特征  $\pm 10\text{ppm}$ , C6, C7 = 18pF, 此搭配的射频频率较为准确, 为  $\pm 10\text{KHz}$  左右。

- 32MHz 晶振如果用有源晶振, GPIO9 和 GPIO10 可以给有源晶振供电。XC2 为芯片的时钟输入引脚。

## 2.5 静电防护

**2.5.1 天线端静电防护** 无论是板载天线还是其他天线, 本质上都是一段长导体, 必然有概率吸引到静电电荷, 为预防静电打坏芯片 RF, \*\* 首先常用板载天线一定要有馈地, 其次天线端建议预留静电防护元件, 必须使用低容值和低钳位电压 (Clamping Voltage) 不高于 5V 的 TVS 元器件, 尽可能不影响 RF 阻抗。

## 2.2.3 3 PCB 设计建议

### 3.1 板材选择和特殊说明

- 关于 PCB 设计, 建议使用四层板, 四层板安规特性更好, 如需考虑成本等因素, 使用两层板布局, 需要将中间的 EPAD 通过顶层 GND 脚接到外面的大地上, 同时中间的 EPAD 接地过孔尽可能多, 比如 16 个。

- 芯片下面的 PCB 板的 Bottom 层不要走线, 特别是靠近射频匹配电路的部分, 完整的地平面能保证射频性能。

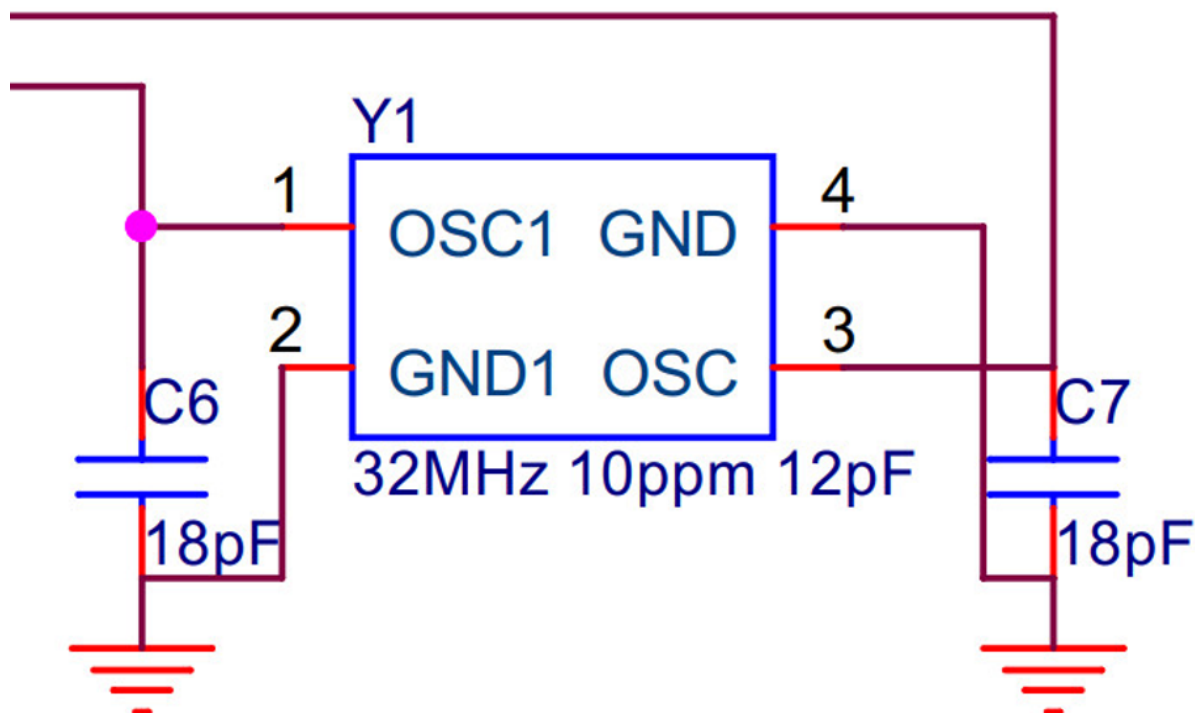


图 17: 32MHz 晶振外围电路示意图

\* 线宽推荐如下:

| 板材属性          | 参数      |
|---------------|---------|
| PCB 板材        | FR4     |
| PCB 板厚        | 1.2mm   |
| 50 欧姆 RF 线宽   | 13.7mil |
| 接地铺铜与 RF 走线间距 | 15mil   |

### 3.2 DC/DC layout 说明

- GND\_DCDC 需要通过 0Ω 电阻与 PCB 地单点连接，不要与 PCB 地直接相接。
- DCDC\_SW 管脚与电感距离尽量短且走线尽量粗，提升 DCDC 的效率。
- DC/DC 电感底部需要净空，电容靠近电感放置。

### 3.3 电源和地 layout 说明

- 电源线宽度要求能达到 0.5mm 以上，承受 200 毫安的瞬态电流。在靠近芯片电源引脚放置去耦电容，其中小容值电容摆放在更靠近芯片引脚的位置，以便较好地滤除高频噪声。
- 电源线和地线采用放射状的连接方式，单点接电源/地并且单独走线，RF 芯片的电源/地线走线与其它芯片或器件分开来，从总参考电源/地线单独引线，防止受到干扰。如果是从 LDO 或者 DCDC 等器件引出电源线，也需要单独引线并且做好滤波措施。芯片底部的 GND 引脚需要和电路板顶层的 GND 平面直接相连。
- 覆地的地线也建议与噪声较少的地线或者总参考地线连接，不与强信号或者强干扰器件地线电源线相连，可以有效地减少整个印制板的工作噪声。

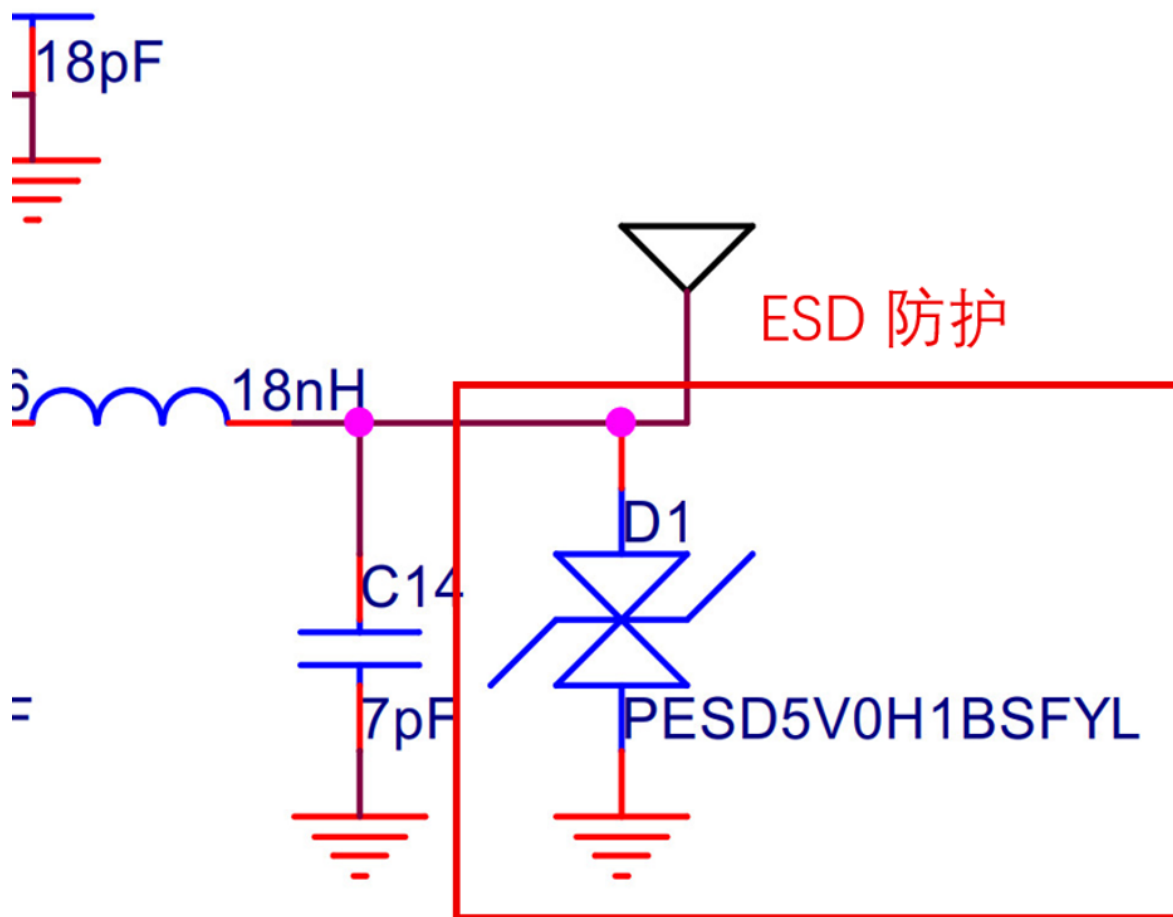


图 18: ESD 防护电路示意图

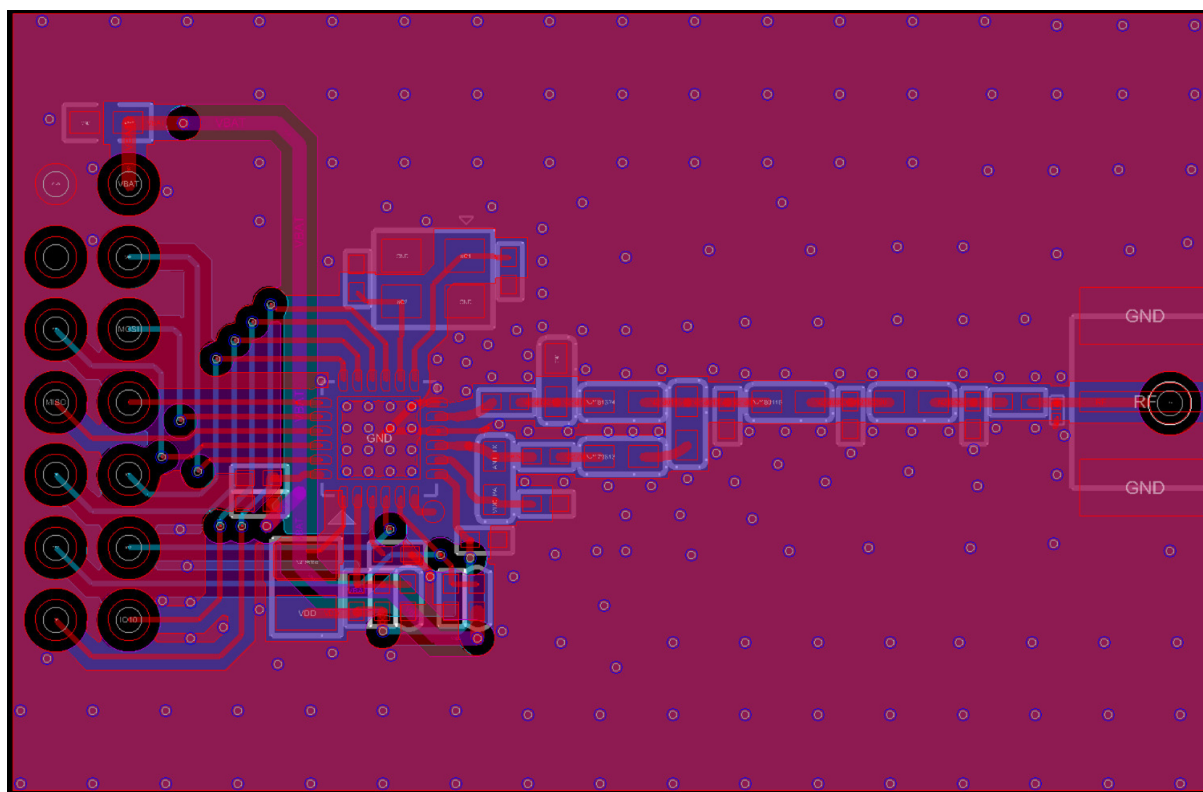


图 19: PCB layout 布局

### 3.4 晶振 layout 说明

- 在面积允许的程度下, 晶体跟芯片之间保持一定的距离, 做好隔热处理。
- 直插的晶振的焊盘需要保证外径与内径差值有 0.2mm 以上。
- 为防止晶振信号干扰到射频信号, 印制板上在晶振焊盘和走线的两边需要做覆地处理。
- 为避免晶振受到天线的发射功率干扰, 印制板上的天线部分与晶振焊盘走线部分之间要用 0.5mm 以上地线作为间隔带, 同时晶振的外壳需要离天线 3mm 以上。

### 3.5 控制线 layout 说明

- 控制类的 SPI 线、IRQ 线需要减少走线干扰, 布线时走线较短并且走线两边有完整的覆地。

### 3.6 射频走线 layout 说明

- 射频匹配链路按照 50Ω 阻抗走线 (与周边铺地间距 0.3mm, 背面完整的参考地), 可以参考 TOP 和 BOTTOM 层的 GND 平面, RF 走线尽可能短, RF 线与焊盘宽度一致。
- RF 线有完整的参考地, 从 IC 端出来就进行包地处理, 两边打 GND 过孔, 底层地平面尽量宽, 可以使得发射能量较多地从天线端出去。
- 禁止射频线打过孔换层。
- 天线旁边的 GND 可以预留露铜, 方便焊接调试天线。
- 芯片底部多打过孔, QFN 封装则打在 E-PAD 上。
- 晶振应远离天线, TOP 层挖空, 周围包地, 降低对电源和 RF 的干扰。
- 天线辐射区域不要摆放金属器件, 净空区挖空处理。

## 2.2.4 4 RF 基本性能

### 4.1 顺络 0402 叠层电感模组性能

#### 4.1.1 320M 频段

| 编号               | Fre-<br>quency(MHz) | DataRate(kbps) | Power(dBm) | EVM(%) | Car-<br>rOfst(KHz) | Devia-<br>tion(KHz) |
|------------------|---------------------|----------------|------------|--------|--------------------|---------------------|
| PAN3120_EMD_315M | 320                 | 50             | 18.3       | 5%     | -4.3               | 24.1                |

| 编号               | Frequency(MHz) | IF(kbps) | DataRate(kbps) | Sen_LDO(dBm) |
|------------------|----------------|----------|----------------|--------------|
| PAN3120_EMD_315M | 320            | 500      | 50             | -108         |

#### 4.1.2 433M 频段

| 编号               | Fre-<br>quency(MHz) | DataRate(kbps) | Power(dBm) | EVM(%) | Car-<br>rOfst(KHz) | Devia-<br>tion(KHz) |
|------------------|---------------------|----------------|------------|--------|--------------------|---------------------|
| PAN3120_EMD_433M | 433                 | 50             | 20         | 3.8%   | -3.3               | 24.2                |

| 编号               | Fre-<br>quency(MHz) | IF(kbps) | DataRate(kbps) | Sen_LDO(dBm) | Sen_DCDC(dBm) |
|------------------|---------------------|----------|----------------|--------------|---------------|
| PAN3120_EMD_433M | 433                 | 500      | 50             | -108         | -107          |



## 4.1.3 490M 频段

| 编号               | Fre-<br>quency(MHz) | DataRate(kbps) | Power(dBm) | EVM(%) | Car-<br>rOfst(KHz) | Devia-<br>tion(KHz) |
|------------------|---------------------|----------------|------------|--------|--------------------|---------------------|
| PAN3120_EMD_490M | 490                 | 50             | 20         | 2.5%   | -5.1               | 24.4                |

| 编号               | Fre-<br>quency(MHz) | IF(kbps) | DataRate(kbps) | Sen_LDO(dBm) | Sen_DCDC(dBm) |
|------------------|---------------------|----------|----------------|--------------|---------------|
| PAN3120_EMD_490M | 490                 | 500      | 50             | -108         | -107          |

## 4.1.4 868M 频段

| 编号               | Fre-<br>quency(MHz) | DataRate(kbps) | Power(dBm) | EVM(%) | Car-<br>rOfst(KHz) | Devia-<br>tion(KHz) |
|------------------|---------------------|----------------|------------|--------|--------------------|---------------------|
| PAN3120_EMD_868M | 865                 | 50             | 20         | 3.1%   | -7                 | 24.3                |

| 编号               | Fre-<br>quency(MHz) | IF(kbps) | DataRate(kbps) | Sen_LDO(dBm) | Sen_DCDC(dBm) |
|------------------|---------------------|----------|----------------|--------------|---------------|
| PAN3120_EMD_868M | 865                 | 500      | 50             | -108         | -108          |

## 4.1.5 915M 频段

| 编号               | Fre-<br>quency(MHz) | DataRate(kbps) | Power(dBm) | EVM(%) | Car-<br>rOfst(KHz) | Devia-<br>tion(KHz) |
|------------------|---------------------|----------------|------------|--------|--------------------|---------------------|
| PAN3120_EMD_915M | 915                 | 50             | 20         | 2.7%   | -6.2               | 24.2                |

| 编号               | Fre-<br>quency(MHz) | IF(kbps) | DataRate(kbps) | Sen_LDO(dBm) | Sen_DCDC(dBm) |
|------------------|---------------------|----------|----------------|--------------|---------------|
| PAN3120_EMD_915M | 915                 | 500      | 50             | -108         | -108          |

## 4.2 网络 0603 叠层电感模组性能

## 4.1.1 433M 频段

| 编号               | Fre-<br>quency(MHz) | DataRate(kbps) | Power(dBm) | EVM(%) | Car-<br>rOfst(KHz) | Devia-<br>tion(KHz) |
|------------------|---------------------|----------------|------------|--------|--------------------|---------------------|
| PAN3120_EMD_433M | 433                 | 50             | 20         | 5.1%   | -0.3               | 25                  |

| 编号               | Fre-<br>quency(MHz) | IF(kbps) | DataRate(kbps) | Sen_LDO(dBm) | Sen_DCDC(dBm) |
|------------------|---------------------|----------|----------------|--------------|---------------|
| PAN3120_EMD_433M | 433                 | 500      | 50             | -108         | -107          |

## 4.2.2 490M 频段

| 编号               | Fre-<br>quency(MHz) | DataRate(kbps) | Power(dBm) | EVM(%) | Car-<br>rOfst(KHz) | Devia-<br>tion(KHz) |
|------------------|---------------------|----------------|------------|--------|--------------------|---------------------|
| PAN3120_EMD_490M | 490                 | 50             | 20         | 4.3%   | -0.1               | 24.6                |

| 编号               | Fre-<br>quency(MHz) | IF(kbps) | DataRate(kbps) | Sen_LDO(dBm) | Sen_DCDC(dBm) |
|------------------|---------------------|----------|----------------|--------------|---------------|
| PAN3120_EMD_490M | 490                 | 500      | 50             | -108         | -107          |

## 4.2.3 868M 频段

| 编号               | Fre-<br>quency(MHz) | DataRate(kbps) | Power(dBm) | EVM(%) | Car-<br>rOfst(KHz) | Devia-<br>tion(KHz) |
|------------------|---------------------|----------------|------------|--------|--------------------|---------------------|
| PAN3120_EMD_868M | 865                 | 50             | 20         | 3.1%   | 0.9                | 24.2                |

| 编号               | Fre-<br>quency(MHz) | IF(kbps) | DataRate(kbps) | Sen_LDO(dBm) | Sen_DCDC(dBm) |
|------------------|---------------------|----------|----------------|--------------|---------------|
| PAN3120_EMD_868M | 865                 | 500      | 50             | -108         | -108          |

## 4.2.4 915M 频段

| 编号               | Fre-<br>quency(MHz) | DataRate(kbps) | Power(dBm) | EVM(%) | Car-<br>rOfst(KHz) | Devia-<br>tion(KHz) |
|------------------|---------------------|----------------|------------|--------|--------------------|---------------------|
| PAN3120_EMD_915M | 915                 | 50             | 20         | 3.1%   | 0.5                | 24                  |

| 编号               | Fre-<br>quency(MHz) | IF(kbps) | DataRate(kbps) | Sen_LDO(dBm) | Sen_DCDC(dBm) |
|------------------|---------------------|----------|----------------|--------------|---------------|
| PAN3120_EMD_915M | 915                 | 500      | 50             | -108         | -108          |

## Chapter 3

# 演示例程

### 3.1 基础例程

#### 3.1.1 SDK\_CTK\_PacketRx

##### 1 功能概述

本代码示例主要演示如何通过 PANCHIP\_RF\_CTK 生成 rf 配置，并实现数据接收的功能。

1、PANCHIP\_RF\_CTK：一款可用于配置 PAN3120 的 rf 参数工具；

##### 2 环境要求

- Board: PAN312x\_EVB
- PAN312x 模组
- USB 转串口模块
- PC 端串口数据显示上位机
- PANCHIP\_RF\_CTK

##### 3 PANCHIP\_RF\_CTK 参数配置

1、打开配置工程，如下图所示：

打开\01\_SDK\PAN312x\_DK\_Develop\User\PAN312x\_Example\SDK\_CTK\_Packet 目录下 PAN3120\_433.ctkp 工程，如下图所示

2、选择生成的配置的输出模式、输出目录，输出名称等信息，如下图所示：

3、射频参数设置，包括频点、速率、调制方式、Deviation、发射功率等参数，如下图所示：

4、中断使能配置，客户可根据自己应用需求开启，如下图所示：

5、操作设置

6、数据包设置之前导码字段，包括前导码模式，前导码长度，如果前导模式选择非标准的话，还需要配置前导码内容，是否使能前导码 manchester 编码，具体如下图所示：

7、数据包设置之同步字字段，包括同步字大小，同步字内容，是否使能同步字 manchester 编码，具体如下图所示：

8、数据包设置之 NodeId 字段，如果帧结构中，NodeId 字段的长度不为 0，则需要配置 NodeId 位置，NodeId 内容，是否使能数据过滤功能，具体配置如下图所示：

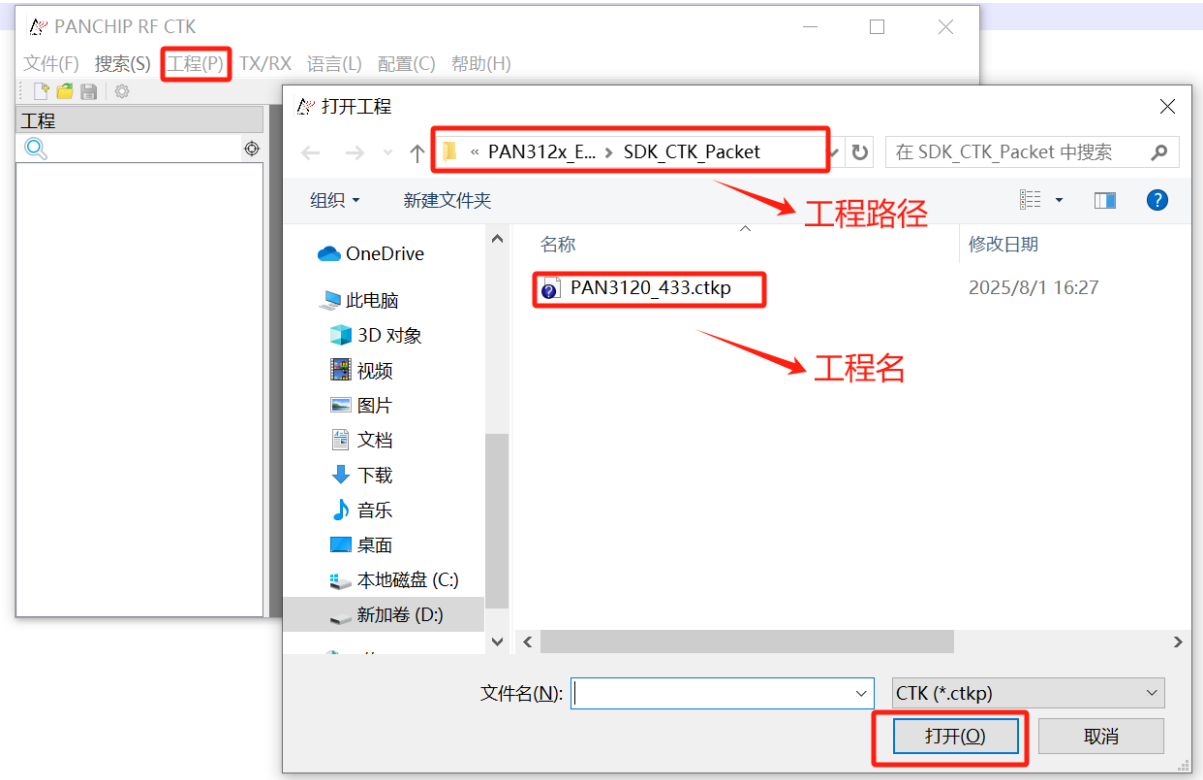


图 1: 打开 PAN3120\_433.ctpk 工程



图 2: 配置输出信息

|                                           |                        |                                                      |
|-------------------------------------------|------------------------|------------------------------------------------------|
| PAN3120_433 x                             |                        |                                                      |
| 芯片信息 工程选项 射频设置 中断使能设置 操作设置 Direct设置 数据包设置 |                        |                                                      |
| + 频段 (MHz)                                | 420 <= frequency < 440 | Frequency = Base + Step * Channel, Frequency必须在频带范围内 |
| + 基频 (MHz)                                | 433.000000             | Frequency = Base + Step * Channel, Frequency必须在频带范围内 |
| + 频率步进 (MHz)                              | 0.000000               | Frequency = Base + Step * Channel, Frequency必须在频带范围内 |
| + 信道                                      | 0                      |                                                      |
| 数据速率配置模式                                  | Symbol Rate B          |                                                      |
| 数据速率 (kbps)                               | 50                     |                                                      |
| 调制方式                                      | 2FSK                   |                                                      |
| 发送频偏 (Hz)                                 | 25000                  |                                                      |
| 接收频偏 (Hz)                                 | 25000                  |                                                      |
| 供电模式                                      | LDO                    | 将会影响功率配置                                             |
| 电感封装                                      | SMD0603                | 将会影响功率配置                                             |
| 发送功率 (dBm)                                | 20                     |                                                      |

图 3: 射频参数设置

|                                           |                                     |         |
|-------------------------------------------|-------------------------------------|---------|
| PAN3120_433 x                             |                                     |         |
| 芯片信息 工程选项 射频设置 中断使能设置 操作设置 Direct设置 数据包设置 |                                     |         |
| 中断使能-发送完成                                 | <input checked="" type="checkbox"/> | 支持自动清除  |
| 中断使能-接收流程完成                               | <input checked="" type="checkbox"/> | 不支持自动清除 |
| 中断使能-数据接收完成                               | <input type="checkbox"/>            | 支持自动清除  |
| 中断使能-CRC错误                                | <input checked="" type="checkbox"/> | 不支持自动清除 |
| 中断使能-接收超时                                 | <input type="checkbox"/>            | 支持自动清除  |
| 中断使能-Node ID错误                            | <input type="checkbox"/>            | 支持自动清除  |
| 中断使能-Lengh接收完成                            | <input type="checkbox"/>            | 支持自动清除  |
| 中断使能-Node ID接收完成                          | <input type="checkbox"/>            | 支持自动清除  |
| 中断使能-包头CRC完成                              | <input type="checkbox"/>            | 支持自动清除  |
| 中断使能-唤醒                                   | <input type="checkbox"/>            | 支持自动清除  |
| 中断使能-应答超时                                 | <input type="checkbox"/>            | 支持自动清除  |
| 中断使能-应答错误                                 | <input type="checkbox"/>            | 支持自动清除  |
| 中断使能-同步字超时                                | <input type="checkbox"/>            | 支持自动清除  |
| 中断使能-发送FIFO阈值                             | <input type="checkbox"/>            | 不支持自动清除 |
| 中断使能-接收FIFO阈值                             | <input type="checkbox"/>            | 不支持自动清除 |
| 中断使能-RSSI有效                               | <input type="checkbox"/>            | 支持自动清除  |
| 中断使能-前导码完成                                | <input type="checkbox"/>            | 支持自动清除  |
| 中断使能-同步字完成                                | <input type="checkbox"/>            | 支持自动清除  |
| 中断使能-前导码超时                                | <input type="checkbox"/>            | 支持自动清除  |
| 中断使能-RSSI超时                               | <input type="checkbox"/>            | 支持自动清除  |

图 4: 中断使能设置



图 5: 操作设置

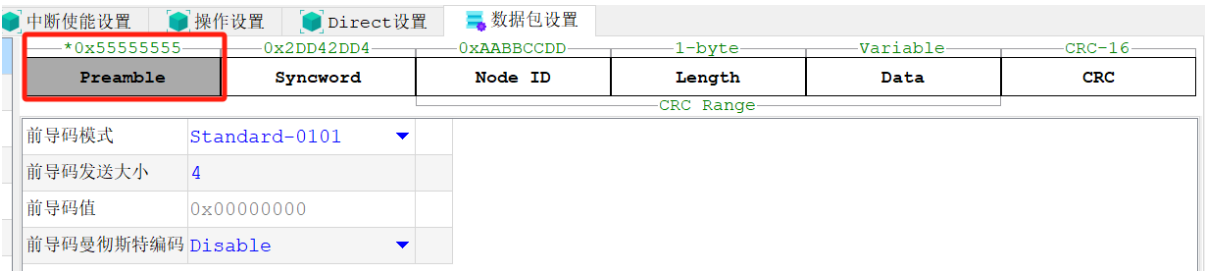


图 6: 数据包设置之前导码字段

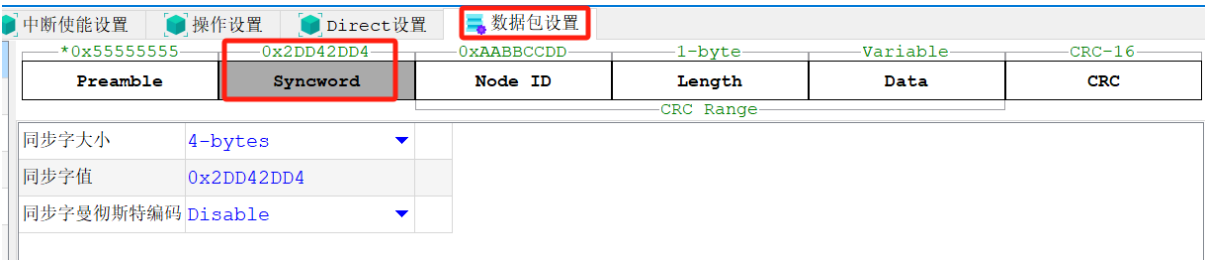


图 7: 数据包设置之同步字字段



| *0x55555555 | 0x2DD42DD4                          | 0xAABBCCDD               | 1-byte | Variable | CRC-16 |
|-------------|-------------------------------------|--------------------------|--------|----------|--------|
| Preamble    | Syncword                            | Node ID                  | Length | Data     | CRC    |
| CRC Range   |                                     |                          |        |          |        |
| Node ID位置   | Before Length                       |                          |        |          |        |
| Node ID发送值  | 0xAABBCCDD                          |                          |        |          |        |
| 数据包过滤模式     | Reset                               |                          |        |          |        |
| 数据包过滤中断     | <input type="checkbox"/>            |                          |        |          |        |
| 数据包过滤掩码     | 0x00000000                          | 对于每个比特, 1表示禁用对应bit的过滤功能。 |        |          |        |
| 启用数据包过滤匹配1  | <input checked="" type="checkbox"/> |                          |        |          |        |
| 数据包过滤匹配1的值  | 0xA0B0C0D0                          |                          |        |          |        |
| 启用数据包过滤匹配2  | <input checked="" type="checkbox"/> |                          |        |          |        |
| 数据包过滤匹配2的值  | 0xA1B1C1D1                          |                          |        |          |        |
| 启用数据包过滤匹配3  | <input checked="" type="checkbox"/> |                          |        |          |        |
| 数据包过滤匹配3的值  | 0xAABBCCDD                          |                          |        |          |        |

图 8: 数据包设置之 NodeId 字段

9、数据包设置之 Length 字段, 包括 Length 的长度, Length 字节序, 具体如下图所示:

| *0x55555555 | 0x2DD42DD4 | 0xAABBCCDD | 1-byte | Variable | CRC-16 |
|-------------|------------|------------|--------|----------|--------|
| Preamble    | Syncword   | Node ID    | Length | Data     | CRC    |
| CRC Range   |            |            |        |          |        |
| Length大小    | 1-byte     |            |        |          |        |
| Length字节序   | Big Endian |            |        |          |        |

图 9: 数据包设置之 Length 字段

10、数据包设置之 Data 字段, 包括数据长度的设置, 具体如下图所示:

| *0x55555555 | 0x2DD42DD4 | 0xAABBCCDD         | 1-byte | Variable | CRC-16 |
|-------------|------------|--------------------|--------|----------|--------|
| Preamble    | Syncword   | Node ID            | Length | Data     | CRC    |
| CRC Range   |            |                    |        |          |        |
| 数据长度        | 20         | for CTK TX/RX only |        |          |        |

图 10: 数据包设置之 Data 字段

11、数据包设置之 CRC, 包括 CRC 初始值, CRC 多项式, CRC 作用范围, CRC 输出是否取反, CRC 字节序设置, CRC 位序设置等, 具体如下图所示:

12、数据包之其他一些参数配置, 主要包括数据类型, NodeId 大小, CRC 模式, Payload 位序, 是否 Payload 的 Manchester 编码, 是否使能白化, 编码方式的选择等, 具体如下图所示:

#### 4 PANCHIP\_RF\_CTK 参数输出

待完成上述第 3 章描述的参数配置后, 然后通过工具 PANCHIP\_RF\_CTK 输出一个.c 和.h 文件, 输出文件路径为上述设置的输出目, 具体如下图所示:

#### 5 编译和烧录

例程位置: 01\_SDK\PAN312x\_DK\_Develop\Projectc\MDK-ARM(AC5)

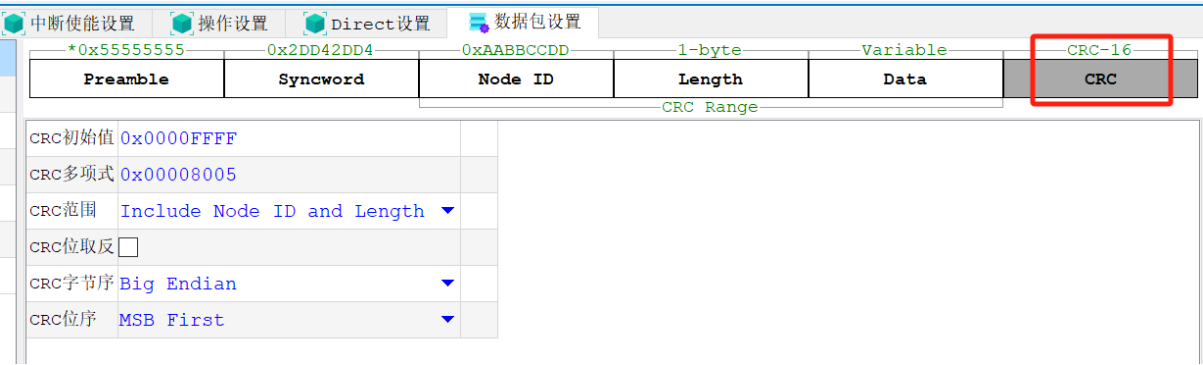


图 11: 数据包设置之 CRC 字段



图 12: 数据包设置之其他一些参数配置



图 13: 数据包设置之其他一些参数配置

1、打开\Projectc\MDK-ARM(AC5) 目录下 project.uvprojx 工程，选择 SDK\_CTK\_PacketTx，如下图所示

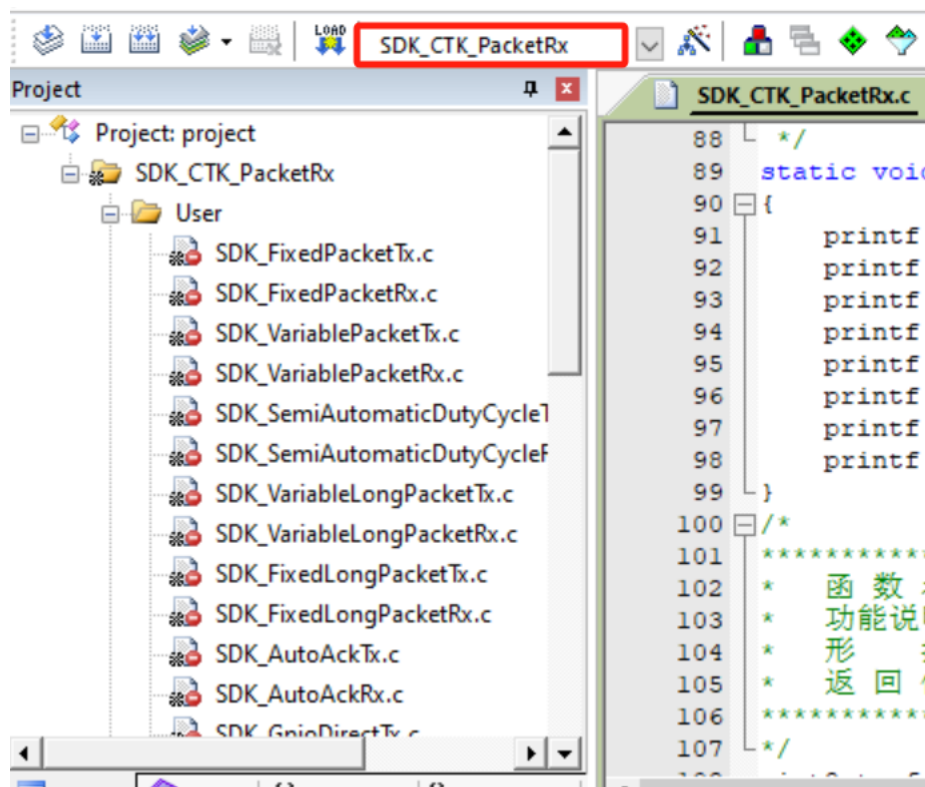


图 14: 工程选择

然后编译并下载该程序。

2、将 PANCHIP\_RF\_CTK 输出的文件添加到工程目录中，如下图所示：

## 6 测试方法

- 1、复位 PAN312x、等待 PAN312x Ready；
- 2、初始化一些寄存器；
- 3、调用宏定义 **\*\*\_\_PAN312x\_LOAD\_CTKCONFIG\*\*** 加载 rf 配置文件到芯片；
- 4、在 while 循环中，等待 Rx 接收完成标志，待等到 Rx 接收完成后，然后读取 fifo，并清除 Rx\_Done 状态；如下图所示：

### 备注：

1、关于 int PAN312x\_Enter\_Rx(uint16\_t channel, uint8\_t condition, uint16\_t rx\_len)，需要注意，当设置 PAN312x 的帧结构为固定包长时，第三个参数 rx\_len 为实际发送的数据长度，当设置 PAN312x 的帧结构为可变包长时，第三个参数 rx\_len 为 0；

## 7 测试现象

- 1、测试现象，串口会打印收取到的数据，如下图所示：
- 2、其他模组作为 Tx，发送的数据，如下图所示：
- 3、通过对比 Tx 和 Rx 数据，两边是一致的，证明 PAN312x 可变包长帧结构的接收功能正确；

**备注：**1、测试过程中，可以用另外一块 PAN312x 模组或其他模组作为 Tx，配置相同的 rf 参数，就可以收取到 PAN312x 或其他模组发送过来的数据。

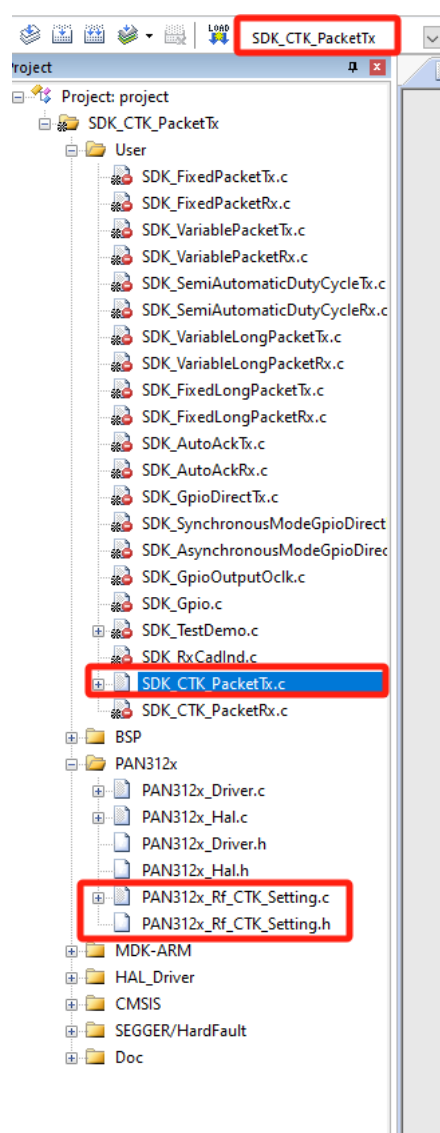


图 15: RF 配置文件添加

```

while (1)
{
    /* 判断定时器超时时间 */
    if(bsp_CheckTimer(0)){
        /* 每隔1000ms进来一次 表示mcu运行正常*/
        bsp_LedToggle(1);
    }

    if(xRxDoneFlag){
        xRxDoneFlag = S_RESET;
        rx_cnt++;

        rssi_value = PAN312x_GetRSSI();

        nPayloadLength = PAN312x_Get_Rx_Length();

        PAN312x_Read_Fifo(vectRxBuffer, nPayloadLength);
        PAN312x_Irq_Clear_RxFifo_Status();
        PAN312x_Irq_Clear_RxDone_Status();
        printf("Rx Cnt = %d\r\n", rx_cnt);
        printf("Rssi Value = %d\r\n", rssi_value);
        printf("Payload length = %d\r\n", nPayloadLength);
        /* print the received data */
        printf("B data received: [");
        for(uint8 t i = 0; i < nPayloadLength; i++)
            printf("%02x ", vectRxBuffer[i]);
        printf("]\r\n");

        if(memcmp(vectRxBuffer, ExpectedData, nPayloadLength) == 0){
            //rx_cnt ++ ;
        }else{
        }
    }
}

```

图 16: 可变包长帧结构接收

```

*****
* 例程名称       : PAN312x-VariableFifoRxB
* PAN312x固件版本 : 0x0002
* 例程版本       : 0.0.0
* 发布日期       : 2025-04-21
* Copyright ver.panchip.com 星启微电子
*****

[2025-04-19 19:44:04.207]# RECV ASCII>

*****
* 例程名称       : PAN312x-VariableFifoRxB
* PAN312x固件版本 : 0x0002
* 例程版本       : 0.0.0
* 发布日期       : 2025-04-21
* Copyright ver.panchip.com 星启微电子
*****

[2025-04-19 19:44:15.684]# RECV ASCII>
Rx Cnt = 1
Rssi Value = 38
Payload length = 38
B data received: [04 11 57 60 a0 d7 23 a0 ea fb ]

[2025-04-19 19:44:27.589]# RECV ASCII>
Rx Cnt = 2
Rssi Value = 39
Payload length = 40
B data received: [42 fe 10 fa 05 fb 86 94 8a a1 15 9a 01 20 af 9d 64 e0 23 6d b1 ee d9 29 29 e0 c2 4a 6e 4a 8f 12 e7 5f 21 23 44 9d 41 8b ]

[2025-04-19 19:44:36.823]# RECV ASCII>
Rx Cnt = 3
Rssi Value = 44
Payload length = 64
B data received: [00 dd 1a f4 a5 07 f2 aa 0a 7c e2 2b 43 85 a0 28 36 8b 7f 77 04 2c b6 e4 8c 8f a3 83 b4 f4 0f 28 8a 8f 74 9b 93 90 3c a0 01 99 6c 31 b1 12 e1 f1 9b 3d fa 5c 5b 29 d6 20 41 0f 7f 54 5a e4 e7 af ]

[2025-04-19 19:44:45.045]# RECV ASCII>
Rx Cnt = 4
Rssi Value = 49
Payload length = 128
B data received: [7a 20 5b af f9 bd 35 30 07 5c 43 8b aa 90 61 f9 a0 55 b5 9d a0 f4 a4 25 a7 60 04 fe d7 aa e2 a0 85 38 41 20 55 ff 16 ba 6b 29 94 91 4a b1 a0 8d ac 5d 53 e0 d4 89 97 3d aa a1 23 f9 d6 ed 32 5a 4b a0 b0 ab 3a ba 9a 88 52 a0 b9 1c e2 10 56 24 44 f9 91 d4 5a 89 a6 7a d7 29 9f 1d a2 e0 30 a1 b7 aa d8 8c a5 67 ab 00 51 5f 62 47 7a 17 c3 e5 59 a3 47 a0 2b 4b 92 8c aa e1 6f e2 03 ]

```

PAN312x可变包长帧结构接收数据

图 17: 可变包长帧结构测试现象

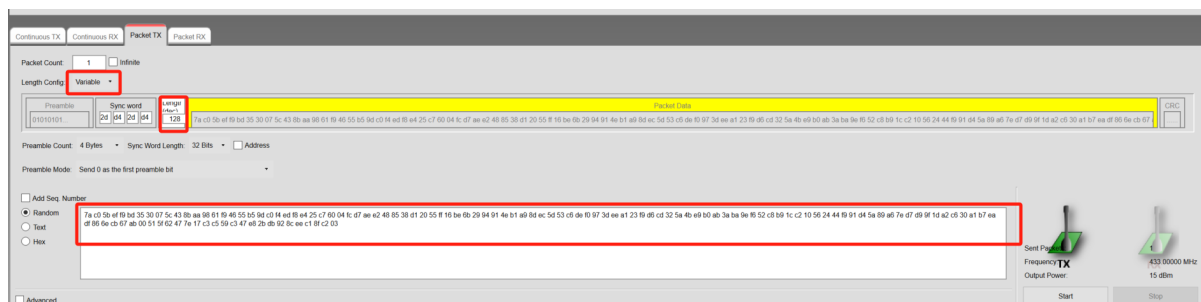


图 18: 其他模组收取到的数据

### 3.1.2 SDK\_CTK\_PacketTx

#### 1 功能概述

本代码示例主要演示如何通过 PANCHIP\_RF\_CTK 生成 rf 配置，并实现数据发送的功能。

- 1、PANCHIP\_RF\_CTK：一款可用于配置 PAN3120 的 rf 参数工具；

#### 2 环境要求

- Board: PAN312x\_EVB
- PAN312x 模组
- USB 转串口模块
- PC 端串口数据显示上位机
- PANCHIP\_RF\_CTK

### 3 PANCHIP\_RF\_CTK 参数配置

1、打开配置工程，如下图所示：

打 开\01\_SDK\PAN312x\_DK\_Develop\User\PAN312x\_Example\SDK\_CTK\_Packet 目 录 下 PAN3120\_433.ctkp 工程，如下图所示

2、选择生成的配置的输出模式、输出目录，输出名称等信息，如下图所示：

3、射频参数设置，包括频点、速率、调制方式、Deviation、发射功率等参数，如下图所示：

4、中断使能配置，客户可根据自己应用需求开启，如下图所示：

5、操作设置

6、数据包设置之前导码字段，包括前导码模式，前导码长度，如果前导模式选择非标准的话，还需要配置前导码内容，是否使能前导码 manchester 编码，具体如下图所示：

7、数据包设置之同步字字段，包括同步字大小，同步字内容，是否使能同步字 manchester 编码，具体如下图所示：

8、数据包设置之 NodeId 字段，如果帧结构中，NodeId 字段的长度不为 0，则需要配置 NodeId 位置，NodeId 内容，是否使能数据过滤功能，具体配置如下图所示：

9、数据包设置之 Length 字段，包括 Length 的长度，Length 字节序，具体如下图所示：

10、数据包设置之 Data 字段，包括数据长度的设置，具体如下图所示：

11、数据包设置之 CRC，包括 CRC 初始值，CRC 多项式，CRC 作用范围，CRC 输出是否取反，CRC 字节序设置，CRC 位序设置等，具体如下图所示：

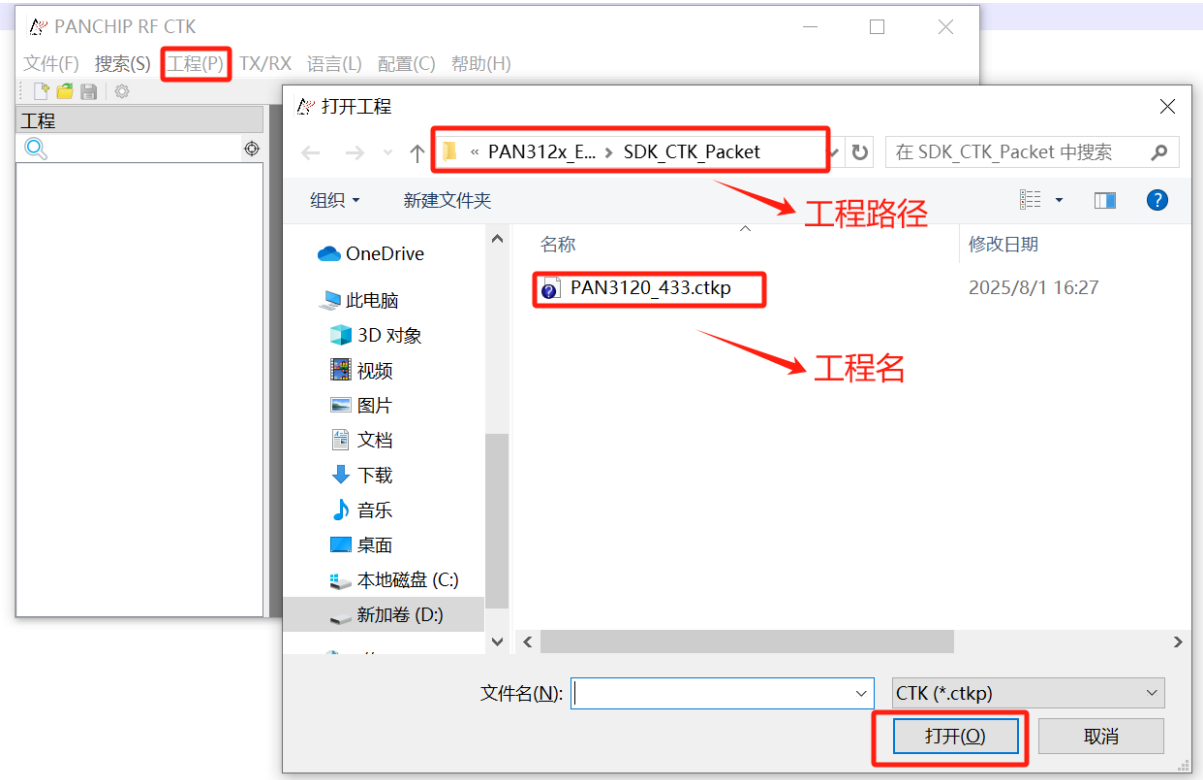


图 19: 打开 PAN3120\_433.ctpk 工程



图 20: 配置输出信息



|               |                        |                                                      |
|---------------|------------------------|------------------------------------------------------|
| PAN3120_433 x |                        |                                                      |
| 芯片信息          | 工程选项                   | 射频设置                                                 |
| 中断使能设置        | 操作设置                   | Direct设置                                             |
| 数据包设置         |                        |                                                      |
| + 频段 (MHz)    | 420 <= frequency < 440 | Frequency = Base + Step * Channel, Frequency必须在频带范围内 |
| + 基频 (MHz)    | 433.000000             | Frequency = Base + Step * Channel, Frequency必须在频带范围内 |
| + 频率步进 (MHz)  | 0.000000               | Frequency = Base + Step * Channel, Frequency必须在频带范围内 |
| + 信道          | 0                      |                                                      |
| 数据速率配置模式      | Symbol Rate B          |                                                      |
| 数据速率 (kbps)   | 50                     |                                                      |
| 调制方式          | 2FSK                   |                                                      |
| 发送频偏 (Hz)     | 25000                  |                                                      |
| 接收频偏 (Hz)     | 25000                  |                                                      |
| 供电模式          | LDO                    | 将会影响功率配置                                             |
| 电感封装          | SMD0603                | 将会影响功率配置                                             |
| 发送功率 (dBm)    | 20                     |                                                      |

图 21: 射频参数设置

|                  |                                     |          |
|------------------|-------------------------------------|----------|
| PAN3120_433 x    |                                     |          |
| 芯片信息             | 工程选项                                | 射频设置     |
| 中断使能设置           | 操作设置                                | Direct设置 |
| 数据包设置            |                                     |          |
| 中断使能-发送完成        | <input checked="" type="checkbox"/> | 支持自动清除   |
| 中断使能-接收流程完成      | <input checked="" type="checkbox"/> | 不支持自动清除  |
| 中断使能-数据接收完成      | <input type="checkbox"/>            | 支持自动清除   |
| 中断使能-CRC错误       | <input checked="" type="checkbox"/> | 不支持自动清除  |
| 中断使能-接收超时        | <input type="checkbox"/>            | 支持自动清除   |
| 中断使能-Node ID错误   | <input type="checkbox"/>            | 支持自动清除   |
| 中断使能-Lengh接收完成   | <input type="checkbox"/>            | 支持自动清除   |
| 中断使能-Node ID接收完成 | <input type="checkbox"/>            | 支持自动清除   |
| 中断使能-包头CRC完成     | <input type="checkbox"/>            | 支持自动清除   |
| 中断使能-唤醒          | <input type="checkbox"/>            | 支持自动清除   |
| 中断使能-应答超时        | <input type="checkbox"/>            | 支持自动清除   |
| 中断使能-应答错误        | <input type="checkbox"/>            | 支持自动清除   |
| 中断使能-同步字超时       | <input type="checkbox"/>            | 支持自动清除   |
| 中断使能-发送FIFO阈值    | <input type="checkbox"/>            | 不支持自动清除  |
| 中断使能-接收FIFO阈值    | <input type="checkbox"/>            | 不支持自动清除  |
| 中断使能-RSSI有效      | <input type="checkbox"/>            | 支持自动清除   |
| 中断使能-前导码完成       | <input type="checkbox"/>            | 支持自动清除   |
| 中断使能-同步字完成       | <input type="checkbox"/>            | 支持自动清除   |
| 中断使能-前导码超时       | <input type="checkbox"/>            | 支持自动清除   |
| 中断使能-RSSI超时      | <input type="checkbox"/>            | 支持自动清除   |

图 22: 中断使能设置



图 23: 操作设置



图 24: 数据包设置之前导码字段

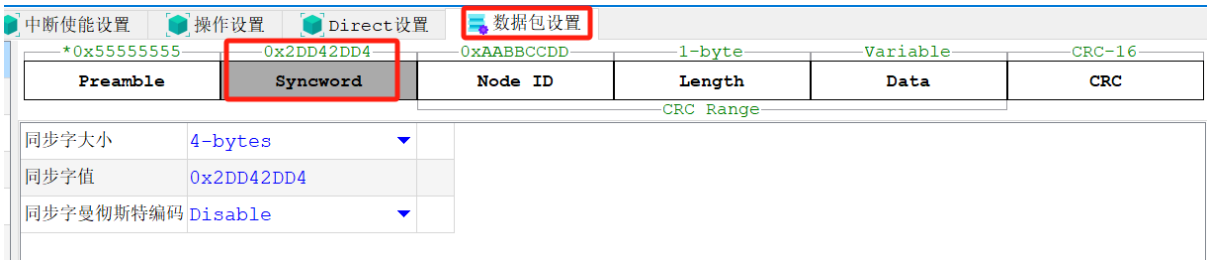


图 25: 数据包设置之同步字字段

中断使能设置 操作设置 Direct设置 数据包设置

|             |            |            |        |          |        |
|-------------|------------|------------|--------|----------|--------|
| *0x55555555 | 0x2DD42DD4 | 0xAABBCCDD | 1-byte | Variable | CRC-16 |
| Preamble    | Syncword   | Node ID    | Length | Data     | CRC    |

CRC Range

|            |                                     |                          |
|------------|-------------------------------------|--------------------------|
| Node ID位置  | Before Length                       |                          |
| Node ID发送值 | 0xAABBCCDD                          |                          |
| 数据包过滤模式    | Reset                               |                          |
| 数据包过滤中断    | <input type="checkbox"/>            |                          |
| 数据包过滤掩码    | 0x00000000                          | 对于每个比特, 1表示禁用对应bit的过滤功能。 |
| 启用数据包过滤匹配1 | <input checked="" type="checkbox"/> |                          |
| 数据包过滤匹配1的值 | 0xA0B0C0D0                          |                          |
| 启用数据包过滤匹配2 | <input checked="" type="checkbox"/> |                          |
| 数据包过滤匹配2的值 | 0xA1B1C1D1                          |                          |
| 启用数据包过滤匹配3 | <input checked="" type="checkbox"/> |                          |
| 数据包过滤匹配3的值 | 0xAABBCCDD                          |                          |

图 26: 数据包设置之 NodeId 字段

中断使能设置 操作设置 Direct设置 数据包设置

|             |            |            |        |          |        |
|-------------|------------|------------|--------|----------|--------|
| *0x55555555 | 0x2DD42DD4 | 0xAABBCCDD | 1-byte | Variable | CRC-16 |
| Preamble    | Syncword   | Node ID    | Length | Data     | CRC    |

CRC Range

|           |            |
|-----------|------------|
| Length大小  | 1-byte     |
| Length字节序 | Big Endian |

图 27: 数据包设置之 Length 字段

中断使能设置 操作设置 Direct设置 数据包设置

|             |            |            |        |          |        |
|-------------|------------|------------|--------|----------|--------|
| *0x55555555 | 0x2DD42DD4 | 0xAABBCCDD | 1-byte | Variable | CRC-16 |
| Preamble    | Syncword   | Node ID    | Length | Data     | CRC    |

CRC Range

|      |    |                    |
|------|----|--------------------|
| 数据长度 | 20 | for CTK TX/RX only |
|------|----|--------------------|

图 28: 数据包设置之 Data 字段

中断使能设置 操作设置 Direct设置 数据包设置

|             |            |            |        |          |        |
|-------------|------------|------------|--------|----------|--------|
| *0x55555555 | 0x2DD42DD4 | 0xAABBCCDD | 1-byte | Variable | CRC-16 |
| Preamble    | Syncword   | Node ID    | Length | Data     | CRC    |

CRC Range

|        |                            |
|--------|----------------------------|
| CRC初始值 | 0x0000FFFF                 |
| CRC多项式 | 0x00008005                 |
| CRC范围  | Include Node ID and Length |
| CRC位取反 | <input type="checkbox"/>   |
| CRC字节序 | Big Endian                 |
| CRC位序  | MSB First                  |

图 29: 数据包设置之 CRC 字段

12、数据包之其他一些参数配置，主要包括数据类型，NodeId 大小，CRC 模式，Payload 位序，是否 Payload 的 Manchester 编码，是否使能白化，编码方式的选择等，具体如下图所示：



图 30: 数据包设置之其他一些参数配置

4 PANCHIP\_RF\_CTK 参数输出

待完成上述第 3 章描述的参数配置后，然后可以通过工具 PANCHIP\_RF\_CTK 输出一个.c 和.h 文件，输出文件路径为上述设置的输出目，具体如下图所示：



图 31: 数据包设置之其他一些参数配置

5 编译和烧录

例程位置：01\_SDK\PAN312x\_DK\_Develop\Projectc\MDK-ARM(AC5)

1、打开\Projectc\MDK-ARM(AC5) 目录下 project.uvprojx 工程，选择 SDK\_CTK\_PacketTx，如下图所示

然后编译并下载该程序。

2、将 PANCHIP\_RF\_CTK 输出的文件添加到工程目录中，如下图所示：

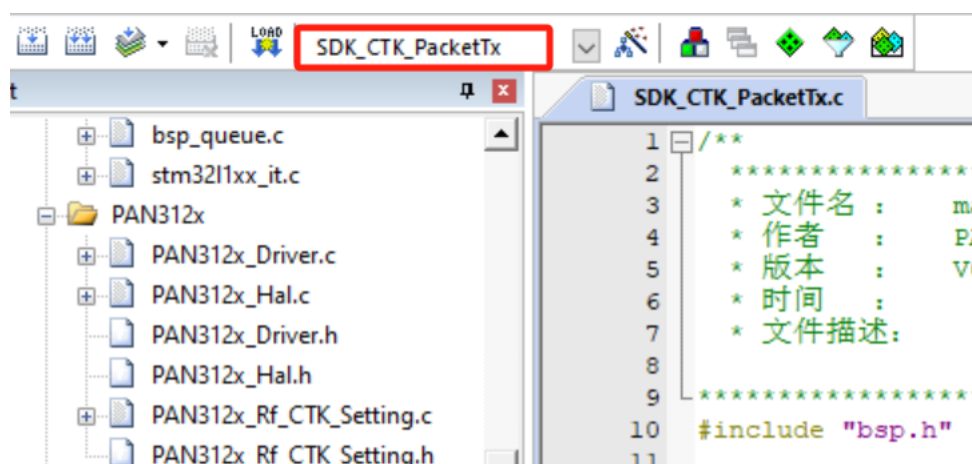


图 32: 工程选择

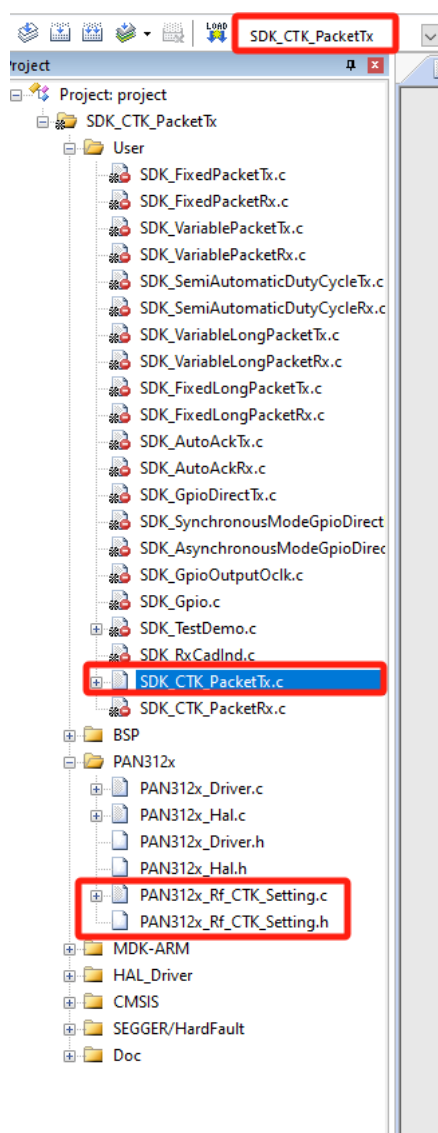


图 33: RF 配置文件添加

## 6 测试方法

- 1、复位 PAN312x、等待 PAN312x Ready;
- 2、初始化一些寄存器;
- 3、调用宏定义 `**__PAN312x_LOAD_CTKCONFIG**` 加载 rf 配置文件到芯片;
- 4、在 while 循环中, 隔一段时间往 fifo 中写入数据, 并执行发送命令, 同时更改数据长度; 如下图所示:

```
while (1)
{
    /* 判断定时器超时时间 */
    if(bsp_CheckTimer(0)){
        /* 每隔1000ms进来一次 表示mcu运行正常*/
        bsp_LedToggle(1);
    }

    if(bsp_CheckTimer(1)){
        printf("\n\rPayload length=%d\n\r", nPayloadLength);
        printf("A data to transmit: [");
        for(uint8_t i = 0; i < nPayloadLength; i++){
            printf("%x ", vectTxBuffer[i]);
        }
        printf("]\r\n");
        PAN312x_Write_Fifo((uint8_t *)vectTxBuffer, nPayloadLength);
        PAN312x_Enter_Tx(0, 0, nPayloadLength);
        while(!xTxDoneFlag);
        xTxDoneFlag = S_RESET;

        nPayloadLength = (nPayloadLength+1)%PAYLOAD_LENGTH_FIX;
    }
}
```

图 34: 可变包长帧结构发送

**备注:**

1、若使能设置系统自动清除部分中断状态, 除了 `IRQ_RX_DONE`、`IRQ_RX_CRC_ERROR`、`IRQ_TX_FIFO_TH`、`IRQ_RX_FIFO_TH` 这些中断状态, 需要额外发送命令来清除, 其它中断状态都可以由 PAN312x 自动清除掉;

## 7 测试现象

- 1、测试现象, 串口会打印将要发送出去的数据, 如下图所示:
- 2、其他模组作为 Rx, 接收 PAN312x 发送的数据, 如下图所示:
- 3、通过对比 Tx 和 Rx 数据, 两边是一致的, 证明 PAN312x 可变包长帧结构的发送功能正确;

**备注:** 1、测试过程中, 可以用另外一块 PAN312x 模组或其他模组作为 Rx, 配置相同的 rf 参数, 就可以收取到 PAN312x 发送过来的数据。

### 3.1.3 SDK\_FixedLongPacketRx

#### 1 功能概述

本代码示例主要演示 PAN312x 固定包长帧结构怎么实现长包接收。

**备注:** 1、当接收的字节数长度大于 fifo 长度 (128bytes) 时, 可采用 rx fifo 阈值的方式来实现长包接收; 2、当前 rx fifo 阈值固定为 64 bytes;

#### 2 环境要求

- Board: PAN312x\_EVB
- PAN312x 模组

```
*****
* 例程名称       : PAN312x-VariablePacketTx
* PAN312x固件版本 : 0x0002
* 例程版本       : 0.0.0
* 发布日期       : 2025-04-21
* Copyright www.panchip.com 磐启微电子
*****

[2025-04-19 19:31:32.395]# RECV ASCII>

Payload length=0

A data to transmit: []

[2025-04-19 19:31:32.901]# RECV ASCII>

Payload length=1

A data to transmit: [ff ]

[2025-04-19 19:31:33.391]# RECV ASCII>
```

可变包长帧结构发送,  
长度(bytes): 0~128

图 35: 可变包长帧结构测试现象



图 36: 其他模组收取到的数据



- USB 转串口模块
- PC 端串口数据显示上位机

### 3 编译和烧录

例程位置：01\_SDK\PAN312x\_DK\_Develop\Projectc\MDK-ARM(AC5)

打开\Projectc\MDK-ARM(AC5) 目录下 project.uvprojx 工程，选择 SDK\_FixedLongPacketRx，如下图所示

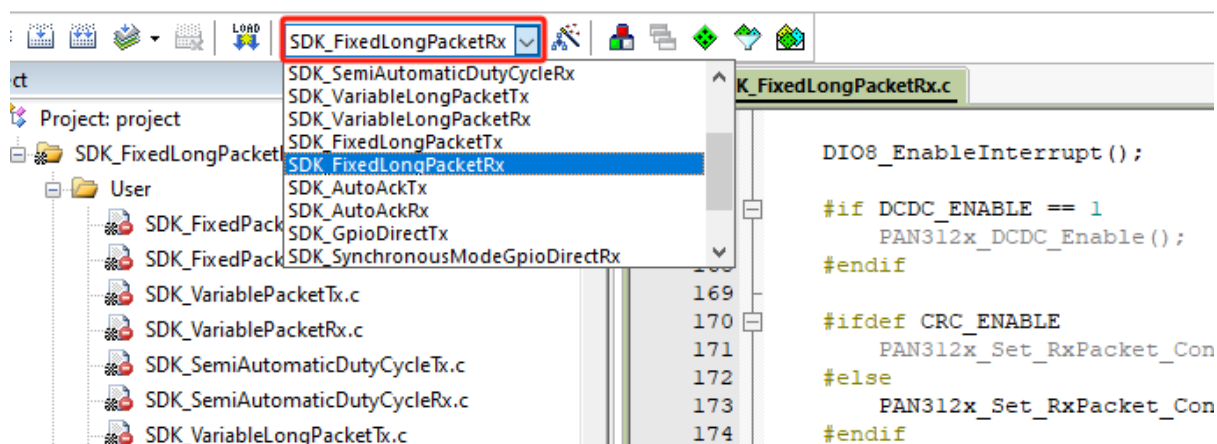


图 37: 工程选择

然后编译并下载该程序。

### 4 RF 参数参数设置

```
typedef struct{
    uint32_t FrequencyBase;
    uint32_t FrequencyStep;
    uint16_t FrequencyChannelNumber;
    PAN312xModulationSelect ModulationSelect;
    uint32_t DataRate;
    uint32_t TxDeviation;
    uint32_t RxDeviation;
    PAN312xPowerdBm Power;
}PAN312xRadioInit;
```

图 38: RF 参数设置

如上图所示：RF 参数目前主要包括频点、调制方式、速率、Tx Deviation、Rx Deviation、发射功率等，其中频点由基础频点，频点步进，通道 3 部分组成；

举例配置频点为 493000000Hz，调制方式为 2FSK，速率为 50000bps，Tx Deviation 为 25000Hz，Rx Deviation 为 25000Hz，发射功率为 21dBm，如下所示：

```
PAN312xRadioInit RadioInit = {
    493000000,
    0,
    0,
    MOD_2FSK,
    50000,
    25000,
    25000,
    POWER_21dBm,
};
```

图 39: RF 参数设置举例

## 5 帧结构参数设置

```
typedef struct{
    uint8_t PreambleLength;
    PAN312xPreambleSelect PreambleSeclect;
    uint32_t NonStandardPreamblePattern;
    uint8_t SyncwordLength;
    uint32_t SyncwordPattern;
    PAN312xManchesterMode PreambleManchesterMode;
    SFunctionalState PreambleManchester;
    PAN312xManchesterMode SyncwordManchesterMode;
    SFunctionalState SyncwordManchester;
    PAN312xManchesterMode PayloadManchesterMode;
    SFunctionalState PayloadManchester;
    PAN312xFecEncoding FecEncoding;
    PAN312xWhiteningPattern WhiteningPattern;
    PAN312xDataMode DataMode;
}PAN321xPacketInit;
```

图 40: 帧结构参数设置

如上图所示：帧结构参数目前主要包前导码长度，前导码内容，同步字长度，同步字内容，前导码、同步字和数据区域的 manchester 编码是否模式，以及 manchester 编码是否使能，Fec 编码，白化选择等参数；

举例配置帧结构，如下图示：

```
PAN321xPacketInit PacketInit = {
    4,
    PREAMBLE_0101,
    0,
    4,
    0x2dd42dd4,
    MANCHESTER_ZeroToTwo,
    S_DISABLE,
    MANCHESTER_ZeroToTwo,
    S_DISABLE,
    MANCHESTER_ZeroToTwo,
    S_DISABLE,
    FEC_HAMING_DISABLE,
    WHITENING_DISABLE,
    DATA_MODE_PACKET,
};
```

图 41: 帧结构参数设置举例

## 6 CRC 参数设置

```
typedef struct{
    PAN312xCrcMode CrcMode;
    uint32_t CrcPolynomial;
    uint32_t CrcSeed;
    PAN312xCrcBitOrder CrcBitOrder;
    PAN312xCrcByteSwap CrcByteSwap;
    PAN312xCrcRange CrcRange;
    SFunctionalState CrcBitInv;
}PAN312xCrcInit;
```

图 42: CRC 参数设置

如上图所示：CRC 参数目前主要 CrcMode、Crc 多项式、Crc 种子、CrcBitOrder、CrcByteSwap、Crc 作用范围，Crc 输出是否取反等。

举例配置 CRC 为 CRC\_16\_IBM，如下图所示：

```

PAN312xCrcInit CrcInit = {
    CRC_MODE_16_BIT,
    0x8005,
    0xffff,
    CRC_BIT_ORDER_MSB_FIRST,
    CRC_BYTE_MSB_FIRST,
    CRC_RANGE_WHOLE_PAYLOAD,
    S_DISABLE,
};

```

图 43: CRC 参数设置举例

## 7 测试方法

- 1、复位 PAN312x、等待 PAN312x Ready;
- 2、初始化一些寄存器;
- 3、设置 RF 参数、帧结构参数、CRC 参数;
- 4、设置 RxPpacket 为**固定包长**, 根据需要确认是否需要使能 CRC;
- 5、将 IRQ\_MASK\_RX\_FIFO\_TH 中断映射到 IRQ(GPIO8) 引脚上;
- 6、设置系统自动清除部分中断状态;
- 7、设置 RxTimeout 时间;
- 8、设置 RxTimeout、RxInvalid、RxValid 后, RF 的状态为 Rx;
- 9、设置 RF 的进入 Rx;
- 9、长包发送处理逻辑, 如下图所示::

```

while (1)
{
    nRxIndex = 0;
    nResidualPcktLength = 0;

    nPayloadLength = 0;

    while(!xRxDoneFlag);
    xRxDoneFlag = S_RESET;
    if(memcmp(&vectRxBuffer[2], vectExpectBuffer, nPayloadLength) == 0){
        printf("received data ok\r\n");
    }else{
        printf("received data error\r\n");
    }

    printf("B data received: [");
    for(uint16_t i = 0; i < (nPayloadLength + 2); i++){
        printf("%02x ", vectRxBuffer[i]);
    }
    printf("]\r\n");

    PAN312x_Enter_Rx(0, 0, 0xffff);
}

```

图 44: 固定包长帧结构接收主函数处理逻辑

备注:

```

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    if(DIO8_GetState() == GPIO_PIN_SET){

        PAN312x_Irq_Get_Status(&xIrqStatus);

        if(xIrqStatus.IRQ_STATUS2_F.IRQ_RX_FIFO_TH){
            PAN312x_Read_Fifo(&vectRxBuffer[nRxIndex], THRESHOLD_RX_FIFO);
            nRxIndex += THRESHOLD_RX_FIFO;
            if(nPayloadLength == 0){
                nPayloadLength = ((vectRxBuffer[1] << 8) | (vectRxBuffer[0]));
                nResidualPcktLength = nPayloadLength;
                printf("nPayloadLength: %d\r\n", nPayloadLength);
            }

            if(nResidualPcktLength <= THRESHOLD_RX_FIFO){
                xRxDoneFlag = S SET;
                PAN312x_Enter_Ready();
                PAN312x_Flush_RxFifo();
                PAN312x_Reset_Modem();
                return;
            }
            nResidualPcktLength = nPayloadLength - nRxIndex;
            printf("nResidualPcktLength: %d\r\n", nResidualPcktLength);

            PAN312x_Irq_Clear_RxFifo_Status();
        }
    }
}

```

读取数据长度

强制PAN312x停止接收

图 45: 固定包长帧结构接收中断处理逻辑

- 1、若使能设置系统自动清除部分中断状态，除了 IRQ\_RX\_DONE、IRQ\_RX\_CRC\_ERROR、IRQ\_TX\_FIFO\_TH、IRQ\_RX\_FIFO\_TH 这些中断状态，需要额外发送命令来清除，其它中断状态都可以由 PAN312x 自动清除掉；
- 2、关于 int PAN312x\_Enter\_Rx(uint16\_t channel, uint8\_t condition, uint16\_t rx\_len)，需要注意，当设置 PAN312x 的帧结构为固定包长时，第三个参数 rx\_len 为实际接收的数据长度，当设置 PAN312x 的帧结构为可变包长时，第三个参数 rx\_len 为 0；
- 3、因为固定包长帧结构，无法知道发送端发送的数据长度，所以在进入接收时，需要将接收的数据长度设置为最大 0x1fff

### 3.1.4 SDK\_FixedLongPacketTx

#### 1 功能概述

本代码示例主要演示 PAN312x 固定包长帧结构怎么实现长包发送。

备注：1、当发送的字节数长度大于 fifo 长度 (128bytes) 时，可采用 tx fifo 阈值的方式来实现长包发送；2、当前 tx fifo 阈值固定为 64 bytes；

#### 2 环境要求

- Board: PAN312x\_EVB
- PAN312x 模组
- USB 转串口模块
- PC 端串口数据显示上位机

### 3 编译和烧录

例程位置：01\_SDK\PAN312x\_DK\_Develop\Projectc\MDK-ARM(AC5)

打开\Projectc\MDK-ARM(AC5) 目录下 project.uvprojx 工程，选择 SDK\_FixedLongPacketTx，如下图所示

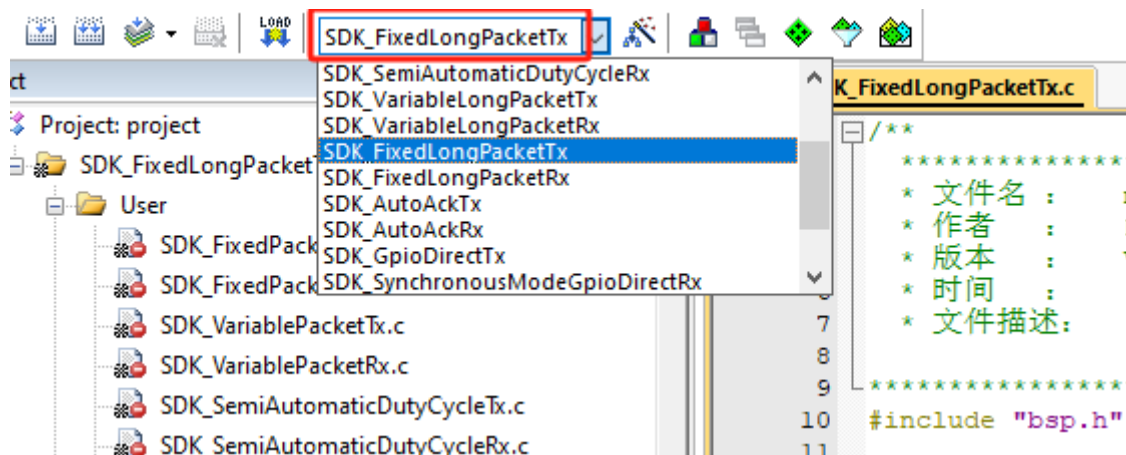


图 46: 工程选择

然后编译并下载该程序。

### 4 RF 参数参数设置

```
typedef struct{
    uint32_t FrequencyBase;
    uint32_t FrequencyStep;
    uint16_t FrequencyChannelNumber;
    PAN312xModulationSelect ModulationSelect;
    uint32_t DataRate;
    uint32_t TxDeviation;
    uint32_t RxDeviation;
    PAN312xPowerdBm Power;
}PAN312xRadioInit;
```

图 47: RF 参数设置

如上图所示：RF 参数目前主要包括频点、调制方式、速率、Tx Deviation、Rx Deviation、发射功率等，其中频点由基础频点，频点步进，通道 3 部分组成；

举例配置频点为 493000000Hz，调制方式为 2FSK，速率为 50000bps，Tx Deviation 为 25000Hz，Rx Deviation 为 25000Hz，发射功率为 21dBm，如下所示：

```
PAN312xRadioInit RadioInit = {
    433000000,
    0,
    0,
    MOD_2FSK,
    50000,
    25000,
    25000,
    POWER_21dBm,
};
```

图 48: RF 参数设置举例

```
typedef struct{
    uint8_t PreambleLength;
    PAN312xPreambleSelect PreambleSeclect;
    uint32_t NonStandardPreamblePattern;
    uint8_t SyncwordLength;
    uint32_t SyncwordPattern;
    PAN312xManchesterMode PreambleManchesterMode;
    SFunctionalState PreambleManchester;
    PAN312xManchesterMode SyncwordManchesterMode;
    SFunctionalState SyncwordManchester;
    PAN312xManchesterMode PayloadManchesterMode;
    SFunctionalState PayloadManchester;
    PAN312xFecEncoding FecEncoding;
    PAN312xWhiteningPattern WhiteningPattern;
    PAN312xDataMode DataMode;
}PAN321xPacketInit;
```

图 49: 帧结构参数设置

## 5 帧结构参数设置

如上图所示：帧结构参数目前主要包前导码长度，前导码内容，同步字长度，同步字内容，前导码、同步字和数据区域的 manchester 编码是否模式，以及 manchester 编码是否使能，Fec 编码，白化选择等参数；

举例配置帧结构，如下图示：

```
PAN321xPacketInit PacketInit = {
    4,
    PREAMBLE_0101,
    0,
    4,
    0x2dd42dd4,
    MANCHESTER_ZeroToTwo,
    S_DISABLE,
    MANCHESTER_ZeroToTwo,
    S_DISABLE,
    MANCHESTER_ZeroToTwo,
    S_DISABLE,
    FEC_HAMING_DISABLE,
    WHITENING_DISABLE,
    DATA_MODE_PACKET,
};
```

图 50: 帧结构参数设置举例

## 6 CRC 参数设置

```
typedef struct{
    PAN312xCrcMode CrcMode;
    uint32_t CrcPolynomial;
    uint32_t CrcSeed;
    PAN312xCrcBitOrder CrcBitOrder;
    PAN312xCrcByteSwap CrcByteSwap;
    PAN312xCrcRange CrcRange;
    SFunctionalState CrcBitInv;
}PAN312xCrcInit;
```

图 51: CRC 参数设置

如上图所示：CRC 参数目前主要 CrcMode、Crc 多项式、Crc 种子、CrcBitOrder、CrcByteSwap、Crc 作用范围，Crc 输出是否取反等。

举例配置 CRC 为 CRC\_16\_IBM，如下图所示：

```

|PAN312xCrcInit CrcInit = {
|    CRC_MODE_16_BIT,
|    0x8005,
|    0xffff,
|    CRC_BIT_ORDER_MSB_FIRST,
|    CRC_BYTE_MSB_FIRST,
|    CRC_RANGE_WHOLE_PAYLOAD,
|    S_DISABLE,
|};

```

图 52: CRC 参数设置举例

## 7 测试方法

- 1、复位 PAN312x、等待 PAN312x Ready;
- 2、初始化一些寄存器;
- 3、设置 RF 参数、帧结构参数、CRC 参数;
- 4、设置 TxPpacket 为**固定包长**, 根据需要确认是否需要使能 CRC;
- 5、将 IRQ\_MASK\_TX\_DONE 和 IRQ\_MASK\_TX\_FIFO\_TH 中断映射到 IRQ(GPIO8) 引脚上;
- 6、设置系统自动清除部分中断状态;
- 7、清除所有中断状态;
- 8、长包发送处理逻辑, 如下图所示:

### 备注:

- 1、若使能设置系统自动清除部分中断状态, 除了 IRQ\_RX\_DONE、IRQ\_RX\_CRC\_ERROR、IRQ\_TX\_FIFO\_TH、IRQ\_RX\_FIFO\_TH 这些中断状态, 需要额外发送命令来清除, 其它中断状态都可以由 PAN312x 自动清除掉;
- 2、往 fifo 中填充数据时, 将第一个和第二个 bytes 作为数据长度;

### 3.1.5 SDK\_FixedPacketRx

#### 1 功能概述

本代码示例主要演示 PAN312x 固定包长帧结构的接收功能。

#### 2 环境要求

- Board: PAN312x\_EVB
- PAN312x 模组
- USB 转串口模块
- PC 端串口数据显示上位机

#### 3 编译和烧录

例程位置: 01\_SDK\PAN312x\_DK\_Develop\Projectc\MDK-ARM(AC5)

打开\Projectc\MDK-ARM(AC5) 目录下 project.uvprojx 工程, 选择 SDK\_FixedPacketRx, 如下图所示

然后编译并下载该程序。



```

while(1){
    nResidualPcktLength = nPayloadLength;
    nTxIndex = 0;
    fitTxBuffer();
    printf("\n\rPayload length = %d\n\n\r", nPayloadLength);

    printf("A data to transmit: [");
    for(uint16_t i = 0; i < nPayloadLength; i++){
        printf("%x ", vectTxBuffer[i]);
    }
    printf("]\r\n");

    if(nPayloadLength > DEV_FIFO_SIZE){
        if((nPayloadLength - DEV_FIFO_SIZE)%256 == 0){
            TxFifoSize = DEV_FIFO_SIZE - 2;
        }else{
            TxFifoSize = DEV_FIFO_SIZE - 1;
        }
        PAN312x_Write_Fifo((uint8_t*)vectTxBuffer, TxFifoSize);

        /* store the number of transmitted bytes */
        nTxIndex = TxFifoSize; 当发送数据个数大于fifo长度时

        /* update the residual number of bytes to be transmitted */
        nResidualPcktLength -= TxFifoSize;
    }else{
        PAN312x_Write_Fifo((uint8_t*)vectTxBuffer, nPayloadLength);
        nResidualPcktLength -= nPayloadLength;
    } 当发送数据个数小于fifo长度时

    PAN312x_Enter_Tx(0, 0, nPayloadLength);
    while(!xTxDoneFlag);
    xTxDoneFlag = S_RESET;

    nPayloadLength = (nPayloadLength+1)%PAYLOAD_LENGTH_FIX;

    tx_cnt++;
    printf("\n\r tx_cnt = %d\n\n\r", tx_cnt);
    bsp_DelayMS(500);
}

```

图 53: 固定包长帧结构长包发送主循环处理逻辑

```

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    if(DIO8_GetState() == GPIO_PIN_SET){
        PAN312x_Irq_Get_Status(&xIrqStatus);
        if(xIrqStatus.IRQ_STATUS0_F.IRQ_TX_DONE){
            xTxDoneFlag = S_SET;
            printf("irq tx done\r\n");
        }
        if(xIrqStatus.IRQ_STATUS2_F.IRQ_TX_FIFO_TH){
            printf("irq tx fifo th\r\n");
            if(nResidualPcktLength == 0){
                printf("0 nResidualPcktLength: %d\r\n", nResidualPcktLength);
                xIrqStatus.IRQ_STATUS2 |= 0x02;
            }else if((nResidualPcktLength + THRESHOLD_TX_FIFO) >= DEV_FIFO_SIZE){
                printf("1 nResidualPcktLength: %d\r\n", nResidualPcktLength);
                PAN312x_Write_Fifo(&vectTxBuffer[nTxIndex], DEV_FIFO_SIZE - THRESHOLD_TX_FIFO);
                nResidualPcktLength -= (DEV_FIFO_SIZE - THRESHOLD_TX_FIFO);
                nTxIndex += (DEV_FIFO_SIZE - THRESHOLD_TX_FIFO);
            }else{
                printf("2 nResidualPcktLength: %d\r\n", nResidualPcktLength);

                PAN312x_Write_Fifo(&vectTxBuffer[nTxIndex], nResidualPcktLength);
                nTxIndex += nResidualPcktLength;
                nResidualPcktLength = 0;
                xIrqStatus.IRQ_STATUS2 |= 0x02;
            }
        }
        PAN312x_Irq_Clear_TxFifo_Status(xIrqStatus);
    }
}
}

```

Tx 完成状态判断

Tx 阈值状态判断

图 54: 固定包长帧结构长包发送中断处理逻辑

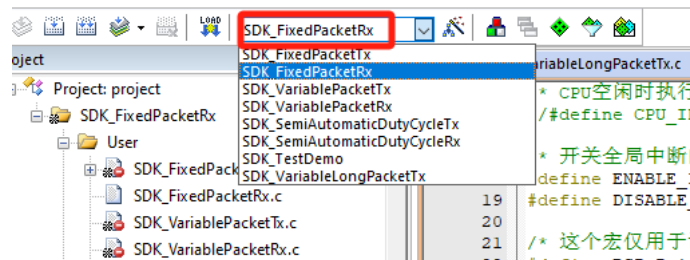


图 55: 工程选择

```

typedef struct{
    uint32_t FrequencyBase;
    uint32_t FrequencyStep;
    uint16_t FrequencyChannelNumber;
    PAN312xModulationSelect ModulationSelect;
    uint32_t DataRate;
    uint32_t TxDeviation;
    uint32_t RxDeviation;
    PAN312xPowerdBm Power;
}PAN312xRadioInit;

```

图 56: RF 参数设置

## 4 RF 参数参数设置

如上图所示：RF 参数目前主要包括频点、调制方式、速率、Tx Deviation、Rx Deviation、发射功率等，其中频点由基础频点，频点步进，通道 3 部分组成；

举例配置频点为 493000000Hz，调制方式为 2FSK，速率为 50000bps，Tx Deviation 为 25000Hz，Rx Deviation 为 25000Hz，发射功率为 21dBm，如下所示：

```
PAN312xRadioInit RadioInit = {
    433000000,
    0,
    0,
    MOD_2FSK,
    50000,
    25000,
    25000,
    POWER_21dBm,
};
```

图 57: RF 参数设置举例

## 5 帧结构参数设置

```
typedef struct{
    uint8_t PreambleLength;
    PAN312xPreambleSelect PreambleSeclect;
    uint32_t NonStandardPreamblePattern;
    uint8_t SyncwordLength;
    uint32_t SyncwordPattern;
    PAN312xManchesterMode PreambleManchesterMode;
    SFunctionalState PreambleManchester;
    PAN312xManchesterMode SyncwordManchesterMode;
    SFunctionalState SyncwordManchester;
    PAN312xManchesterMode PayloadManchesterMode;
    SFunctionalState PayloadManchester;
    PAN312xFecEncoding FecEncoding;
    PAN312xWhiteningPattern WhiteningPattern;
    PAN312xDataMode DataMode;
}PAN321xPacketInit;
```

图 58: 帧结构参数设置

如上图所示：帧结构参数目前主要包前导码长度，前导码内容，同步字长度，同步字内容，前导码、同步字和数据区域的 manchester 编码是否模式，以及 manchester 编码是否使能，Fec 编码，白化选择等参数；

举例配置帧结构，如下图示：

```
PAN321xPacketInit PacketInit = {
    4,
    PREAMBLE_0101,
    0,
    4,
    0x2dd42dd4,
    MANCHESTER_ZeroToTwo,
    S_DISABLE,
    MANCHESTER_ZeroToTwo,
    S_DISABLE,
    MANCHESTER_ZeroToTwo,
    S_DISABLE,
    FEC_HAMING_DISABLE,
    WHITENING_DISABLE,
    DATA_MODE_PACKET,
};
```

图 59: 帧结构参数设置举例

```

typedef struct{
    PAN312xCrcMode CrcMode;
    uint32_t CrcPolynomial;
    uint32_t CrcSeed;
    PAN312xCrcBitOrder CrcBitOrder;
    PAN312xCrcByteSwap CrcByteSwap;
    PAN312xCrcRange CrcRange;
    SFunctionalState CrcBitInv;
}PAN312xCrcInit;

```

图 60: CRC 参数设置

## 6 CRC 参数设置

如上图所示: CRC 参数目前主要 CrcMode、Crc 多项式、Crc 种子、CrcBitOrder、CrcByteSwap、Crc 作用范围, Crc 输出是否取反等。

举例配置 CRC 为 CRC\_16\_IBM, 如下图所示:

```

PAN312xCrcInit CrcInit = {
    CRC_MODE_16_BIT,
    0x8005,
    0xffff,
    CRC_BIT_ORDER_MSB_FIRST,
    CRC_BYTE_MSB_FIRST,
    CRC_RANGE_WHOLE_PAYLOAD,
    S_DISABLE,
};

```

图 61: CRC 参数设置举例

## 7 测试方法

- 1、复位 PAN312x、等待 PAN312x Ready;
- 2、初始化一些寄存器;
- 3、设置 RF 参数、帧结构参数、CRC 参数;
- 4、设置 RxPpacket 为**固定包长**, 根据需要确认是否需要使能 CRC;
- 5、将 IRQ\_MASK\_RX\_DONE 和 IRQ\_MASK\_RX\_CRC\_ERROR 中断映射到 IRQ(GPIO8) 引脚上;
- 6、设置系统自动清除部分中断状态;
- 7、设置 RxTimeout 时间;
- 8、设置 RxTimeout、RxInvalid、RxValid 后, RF 的状态为 Rx;
- 9、设置 RF 的进入 Rx;
- 9、在 while 循环中, 等待 Rx 接收完成标志, 待等到 Rx 接收完成后, 然后读取 fifo, 并清除 Rx\_Done 状态; 如下图所示:

### 备注:

- 1、若使能设置系统自动清除部分中断状态, 除了 IRQ\_RX\_DONE、IRQ\_RX\_CRC\_ERROR、IRQ\_TX\_FIFO\_TH、IRQ\_RX\_FIFO\_TH 这些中断状态, 需要额外发送命令来清除, 其它中断状态都可以由 PAN312x 自动清除掉;
- 2、关于 int PAN312x\_Enter\_Rx(uint16\_t channel, uint8\_t condition, uint16\_t rx\_len), 需要注意, 当设置 PAN312x 的帧结构为固定包长时, 第三个参数 rx\_len 为实际发送的数据长度, 当设置 PAN312x 的帧结构为可变包长时, 第三个参数 rx\_len 为 0;

```

while (1)
{
    /* 判断定时器超时时间 */
    if(bsp_CheckTimer(0)){
        /* 每隔1000ms进来一次 表示mcu运行正常*/
        bsp_LedToggle(1);
    }

    if(xRxDoneFlag){
        xRxDoneFlag = S_RESET;

        rssi_value = PAN312x_GetRSSI();

        nPayloadLength = PAN312x_Get_Rx_Length();

        PAN312x_Read_Fifo(vectRxBuffer, nPayloadLength);
        PAN312x_Irq_Clear_RxDone_Status();

        printf("Rssi Value = %d\r\n", rssi_value);
        printf("Payload length = %d\r\n", nPayloadLength);
        /* print the received data */
        printf("B data received: [");
        for(uint8_t i = 0; i < nPayloadLength; i++)
            printf("%02x ", vectRxBuffer[i]);
        printf("]\r\n");

        if(memcmp(vectRxBuffer, ExpectedData, nPayloadLength) == 0){
            //rx_cnt ++ ;
        }else{
        }
    }
}

```

图 62: 固定包长帧结构接收

## 8 测试现象

1、测试现象，串口会打印收取到的数据，如下图所示：

```

*****
* 例程名称       : PAN312x-FixedPacketRx
* PAN312x固件版本 : 0x0002
* 例程版本       : 0.0.0
* 发布日期       : 2025-04-21
* Copyright www.panchip.com 磐启微电子
*****
PAN312x收取到的数据，长度：32bytes

[2025-04-19 18:59:30.357]# RECV ASCII>
Rssi Value = 41
Payload length = 32
B data received: [9e 36 61 78 1c 68 09 0a 07 06 57 e4 6e 2d 09 9c 23 bd 8d b5 28 00 16 68 64 8c ef a3 9a
6a 10 53 ]

```

图 63: 固定包长帧结构测试现象

2、其他模组作为 Tx，发送的数据，如下图所示：

3、通过对比 Tx 和 Rx 数据，两边是一致的，证明 PAN312x 固件包长帧结构的接收功能正确；

**备注：**1、测试过程中，可以用另外一块 PAN312x 模组或其他模组作为 Tx，配置相同的 rf 参数，就可以收取到 PAN312x 或其他模组发送过来的数据。

### 3.1.6 SDK\_FixedPacketTx

#### 1 功能概述

本代码示例主要演示 PAN312x 固定包长帧结构的发送功能。

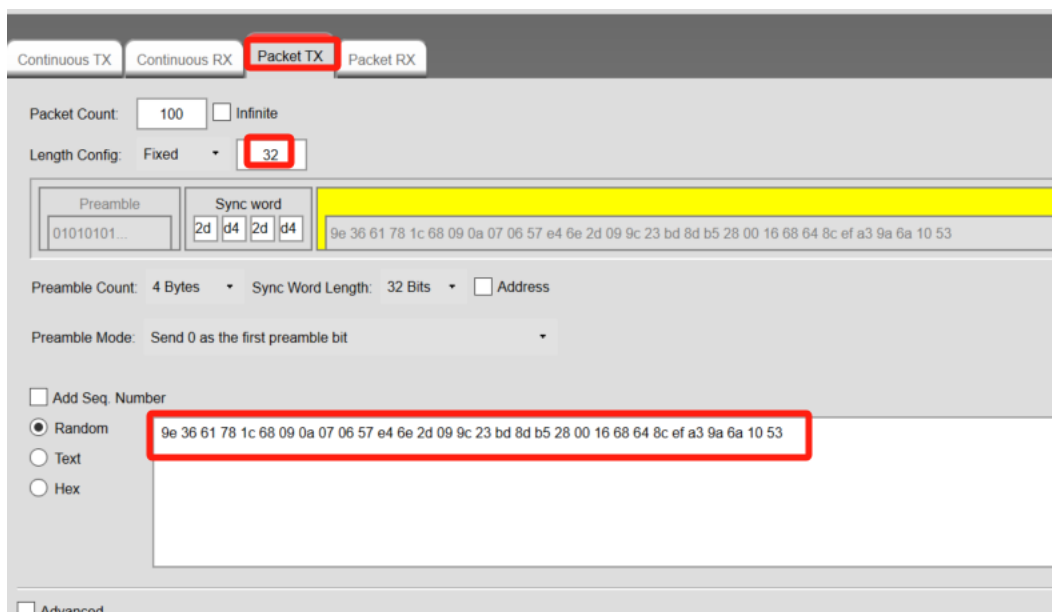


图 64: 其他模组收取到的数据

## 2 环境要求

- Board: PAN312x\_EVB
- PAN312x 模组
- USB 转串口模块
- PC 端串口数据显示上位机

## 3 编译和烧录

例程位置: 01\_SDK\PAN312x\_DK\_Develop\Projectc\MDK-ARM(AC5)

打开\Projectc\MDK-ARM(AC5) 目录下 project.uvprojx 工程, 选择 SDK\_FixedPacketTx, 如下图所示

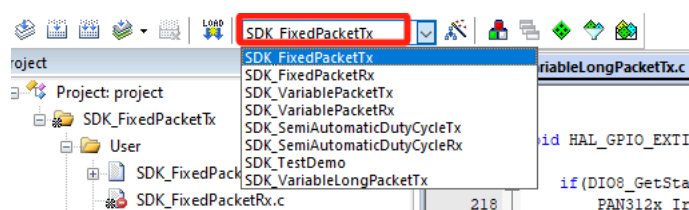


图 65: 工程选择

然后编译并下载该程序。

## 4 RF 参数参数设置

如上图所示: RF 参数目前主要包括频点、调制方式、速率、Tx Deviation、Rx Deviation、发射功率等, 其中频点由基础频点, 频点步进, 通道 3 部分组成;

举例配置频点为 493000000Hz, 调制方式为 2FSK, 速率为 50000bps, Tx Deviation 为 25000Hz, Rx Deviation 为 25000Hz, 发射功率为 21dBm, 如下所示:

```
typedef struct{
    uint32_t FrequencyBase;
    uint32_t FrequencyStep;
    uint16_t FrequencyChannelNumber;
    PAN312xModulationSelect ModulationSelect;
    uint32_t DataRate;
    uint32_t TxDeviation;
    uint32_t RxDeviation;
    PAN312xPowerdBm Power;
}PAN312xRadioInit;
```

图 66: RF 参数设置

```
PAN312xRadioInit RadioInit = {
    433000000,
    0,
    0,
    MOD_2FSK,
    50000,
    25000,
    25000,
    POWER_21dBm,
};
```

图 67: RF 参数设置举例

## 5 帧结构参数设置

```
typedef struct{
    uint8_t PreambleLength;
    PAN312xPreambleSelect PreambleSelect;
    uint32_t NonStandardPreamblePattern;
    uint8_t SyncwordLength;
    uint32_t SyncwordPattern;
    PAN312xManchesterMode PreambleManchesterMode;
    SFunctionalState PreambleManchester;
    PAN312xManchesterMode SyncwordManchesterMode;
    SFunctionalState SyncwordManchester;
    PAN312xManchesterMode PayloadManchesterMode;
    SFunctionalState PayloadManchester;
    PAN312xFecEncoding FecEncoding;
    PAN312xWhiteningPattern WhiteningPattern;
    PAN312xDataMode DataMode;
}PAN312xPacketInit;
```

图 68: 帧结构参数设置

如上图所示：帧结构参数目前主要包前导码长度，前导码内容，同步字长度，同步字内容，前导码、同步字和数据区域的 manchester 编码是否模式，以及 manchester 编码是否使能，Fec 编码，白化选择等参数；

举例配置帧结构，如下图示：

## 6 CRC 参数设置

如上图所示：CRC 参数目前主要 CrcMode、Crc 多项式、Crc 种子、CrcBitOrder、CrcByteSwap、Crc 作用范围，Crc 输出是否取反等。

举例配置 CRC 为 CRC\_16\_IBM，如下图所示：

## 7 测试方法

- 1、复位 PAN312x、等待 PAN312x Ready；
- 2、初始化一些寄存器；



```

PAN321xPacketInit PacketInit = {
    4,
    PREAMBLE_0101,
    0,
    4,
    0x2dd42dd4,
    MANCHESTER_ZeroToTwo,
    S_DISABLE,
    MANCHESTER_ZeroToTwo,
    S_DISABLE,
    MANCHESTER_ZeroToTwo,
    S_DISABLE,
    FEC_HAMING_DISABLE,
    WHITENING_DISABLE,
    DATA_MODE_PACKET,
};

```

图 69: 帧结构参数设置举例

```

typedef struct{
    PAN312xCrcMode CrcMode;
    uint32_t CrcPolynomial;
    uint32_t CrcSeed;
    PAN312xCrcBitOrder CrcBitOrder;
    PAN312xCrcByteSwap CrcByteSwap;
    PAN312xCrcRange CrcRange;
    SFunctionalState CrcBitInv;
}PAN312xCrcInit;

```

图 70: CRC 参数设置

```

PAN312xCrcInit CrcInit = {
    CRC_MODE_16_BIT,
    0x8005,
    0xffff,
    CRC_BIT_ORDER_MSB_FIRST,
    CRC_BYTE_MSB_FIRST,
    CRC_RANGE_WHOLE_PAYLOAD,
    S_DISABLE,
};

```

图 71: CRC 参数设置举例

- 3、设置 RF 参数、帧结构参数、CRC 参数；
- 4、设置 TxPpacket 为**固定包长**，根据需要确认是否需要使能 CRC；
- 5、将 IRQ\_MASK\_TX\_DONE 中断映射到 IRQ(GPIO8) 引脚上；
- 6、设置系统自动清除部分中断状态；
- 7、清除所有中断状态；
- 8、在 while 循环中，隔一段时间往 fifo 中写入数据，并执行发送命令；如下图所示：

```

while (1)
{
    /* 判断定时器超时时间 */
    if(bsp_CheckTimer(0)){
        /* 每隔1000ms进来一次 表示mcu运行正常*/
        bsp_LedToggle(1);
    }

    if(bsp_CheckTimer(1)){
        printf("A data to transmit: [");
        for(uint8_t i = 0; i < nPayloadLength; i++){
            printf("%x ", vectTxBuffer[i]);
        }
        printf("]\n");
        PAN312x_Write_Fifo((uint8_t *)vectTxBuffer, nPayloadLength);
        PAN312x_Enter_Tx(0, 0, nPayloadLength);
        while(!xTxDoneFlag);
        xTxDoneFlag = S_RESET;
    }
}

```

图 72: 固定包长帧结构发送

#### 备注：

- 1、若使能设置系统自动清除部分中断状态，除了 IRQ\_RX\_DONE、IRQ\_RX\_CRC\_ERROR、IRQ\_TX\_FIFO\_TH、IRQ\_RX\_FIFO\_TH 这些中断状态，需要额外发送命令来清除，其它中断状态都可以由 PAN312x 自动清除掉；

## 8 测试现象

- 1、测试现象，串口会打印将要发送出去的数据，如下图所示：

```

*****
* 例程名称       : PAN312x-FixedPacketTx
* PAN312x固件版本 : 0x0002
* 例程版本       : 0.0.0
* 发布日期       : 2025-04-21
* Copyright www.panchip.com 磐启微电子
*****

发送的数据包，长度 32 bytes

[2025-04-19 17:32:34.137]# RECV ASCII>
A data to transmit: [ff 83 df 17 32 9 4e d1 e7 cd 8a 91 c6 d5 c4 c4 40 21 18 4e 55 86 f4 dc 8a 15 a7 ec
92 df 93 53 ]

[2025-04-19 17:32:34.621]# RECV ASCII>
A data to transmit: [ff 83 df 17 32 9 4e d1 e7 cd 8a 91 c6 d5 c4 c4 40 21 18 4e 55 86 f4 dc 8a 15 a7 ec
92 df 93 53 ]

```

图 73: 固定包长帧结构测试现象

- 2、其他模组作为 Rx，接收 PAN312x 发送的数据，如下图所示：

```

17.40.36.334 | ff 83 df 17 32 09 4e d1 e7 cd 8a 91 c6 d5 c4 c4 40 21 18 4e 55 86 f4 dc 8a 15 a7 ec 92 df 93 53 | -46
17.40.36.832 | ff 83 df 17 32 09 4e d1 e7 cd 8a 91 c6 d5 c4 c4 40 21 18 4e 55 86 f4 dc 8a 15 a7 ec 92 df 93 53 | -48
17.40.37.327 | ff 83 df 17 32 09 4e d1 e7 cd 8a 91 c6 d5 c4 c4 40 21 18 4e 55 86 f4 dc 8a 15 a7 ec 92 df 93 53 | -47
17.40.37.821 | ff 83 df 17 32 09 4e d1 e7 cd 8a 91 c6 d5 c4 c4 40 21 18 4e 55 86 f4 dc 8a 15 a7 ec 92 df 93 53 | -49
17.40.38.318 | ff 83 df 17 32 09 4e d1 e7 cd 8a 91 c6 d5 c4 c4 40 21 18 4e 55 86 f4 dc 8a 15 a7 ec 92 df 93 53 | -47
17.40.38.815 | ff 83 df 17 32 09 4e d1 e7 cd 8a 91 c6 d5 c4 c4 40 21 18 4e 55 86 f4 dc 8a 15 a7 ec 92 df 93 53 | -46
17.40.39.309 | ff 83 df 17 32 09 4e d1 e7 cd 8a 91 c6 d5 c4 c4 40 21 18 4e 55 86 f4 dc 8a 15 a7 ec 92 df 93 53 | -47
17.40.39.806 | ff 83 df 17 32 09 4e d1 e7 cd 8a 91 c6 d5 c4 c4 40 21 18 4e 55 86 f4 dc 8a 15 a7 ec 92 df 93 53 | -47
17.40.40.302 | ff 83 df 17 32 09 4e d1 e7 cd 8a 91 c6 d5 c4 c4 40 21 18 4e 55 86 f4 dc 8a 15 a7 ec 92 df 93 53 | -47
17.40.40.798 | ff 83 df 17 32 09 4e d1 e7 cd 8a 91 c6 d5 c4 c4 40 21 18 4e 55 86 f4 dc 8a 15 a7 ec 92 df 93 53 | -47
17.40.41.293 | ff 83 df 17 32 09 4e d1 e7 cd 8a 91 c6 d5 c4 c4 40 21 18 4e 55 86 f4 dc 8a 15 a7 ec 92 df 93 53 | -46

```

图 74: 其他模组收取到的数据

3、通过对比 Tx 和 Rx 数据，两边是一致的，证明 PAN312x 固件包长帧结构的发送功能正确；

**备注：**1、测试过程中，可以用另外一块 PAN312x 模组或其他模组作为 Rx，配置相同的 rf 参数，就可以收取到 PAN312x 发送过来的数据。

### 3.1.7 SDK\_GpioOutputOclk

#### 1 功能概述

本代码示例主要演示 PAN312x 的 GPIO 如何输出时钟信号。

**备注：**1、通过配置 GPIO0、GPIO1、GPIO2、GPIO9、GPIO10 输出时钟信号；2、输出时钟的频率可以为 500Khz、2Mhz、4Mhz、8Mhz、16Mhz；

#### 2 环境要求

- Board: PAN312x\_EVB
- PAN312x 模组
- 逻辑分析仪

#### 3 编译和烧录

例程位置：01\_SDK\PAN312x\_DK\_Develop\Projectc\MDK-ARM(AC5)

打开\Projectc\MDK-ARM(AC5) 目录下 project.uvprojx 工程，选择 SDK\_GpioOutputOclk，如下图所示

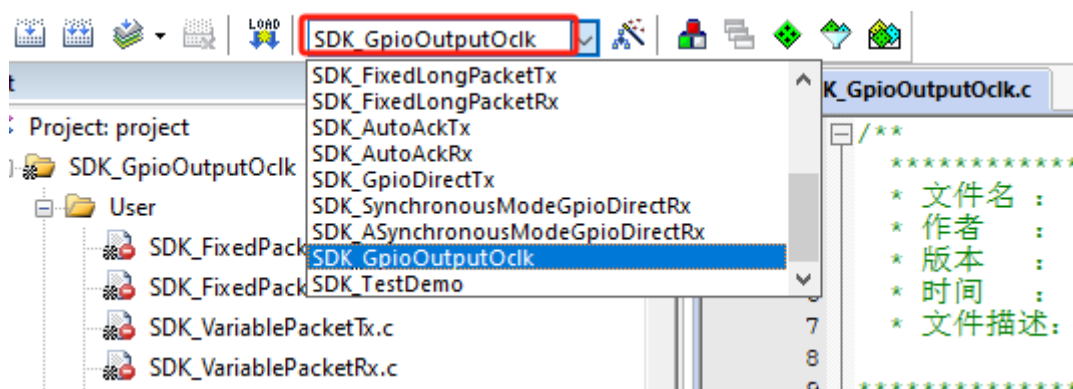


图 75: 工程选择

然后编译并下载该程序。

## 4 测试方法

- 1、复位 PAN312x、等待 PAN312x Ready;
- 2、初始化一些寄存器;
- 3、配置 GPIO0、GPIO1、GPIO2、GPIO9、GPIO10 输出时钟信号;

## 5 测试现象

- 1、500Khz 时钟输出，如下图所示:

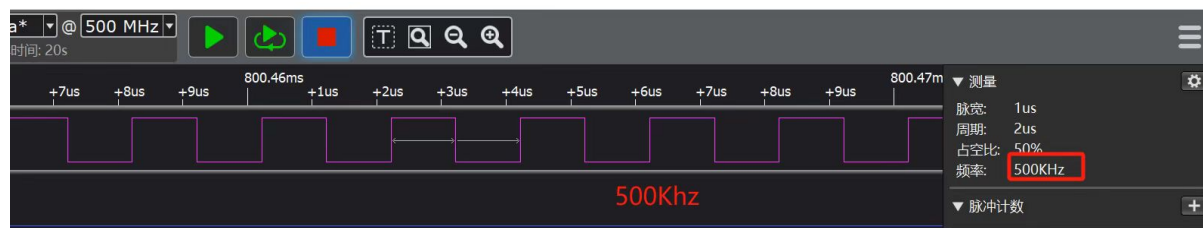


图 76: 500Khz 时钟输出

- 2、2Mhz 时钟输出，如下图所示:

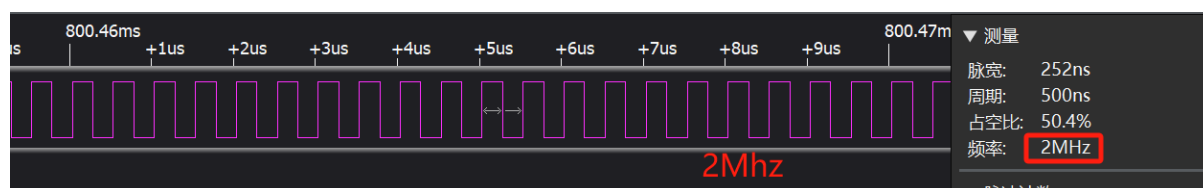


图 77: 2Mhz 时钟输出

- 3、4Mhz 时钟输出，如下图所示:

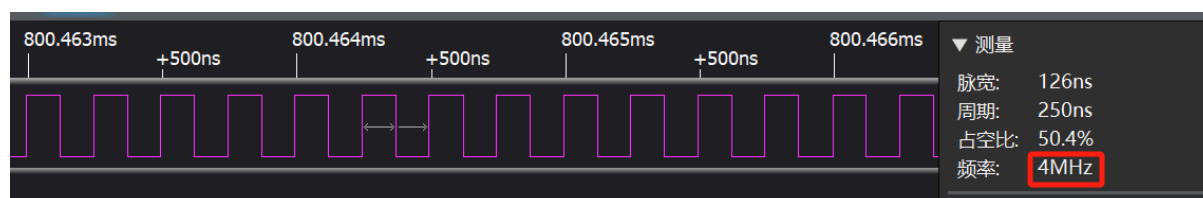


图 78: 4Mhz 时钟输出

- 4、8Mhz 时钟输出，如下图所示:

- 5、16Mhz 时钟输出，如下图所示:

### 3.1.8 SDK\_SemiAutomaticDutyCycleRx

#### 1 功能概述

本代码示例主要演示通过外部 MCU 结合 PAN312x 的低功耗特性，如何实现 DutyCycleRx 功能。

备注: 1、通过主控 MCU 来定时，控制 PAN312x 进入 Rx 的时间，该时间可理解为 PAN312x 的开窗时间，如果在开窗时间内，有收取到有效的 preamble 或 syncword，则延长 Rx 接收时间，直至收取到整个数据包；如果在开窗时间内，没有收取到有效的 preamble 或 syncword，则设置芯片进入休眠，同时计时休眠时间，待到达休眠时间，则将 PAN312x 唤醒，重新进入 Rx 开窗时间。



图 79: 8Mhz 时钟输出

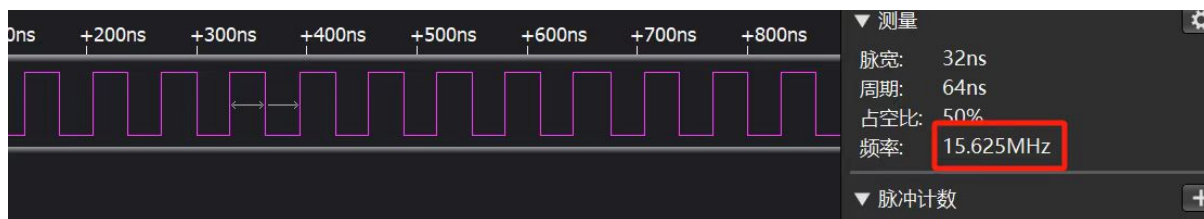


图 80: 16Mhz 时钟输出

## 2 环境要求

- Board: PAN312x\_EVB
- PAN312x 模组
- USB 转串口模块
- PC 端串口数据显示上位机

## 3 编译和烧录

例程位置: 01\_SDK\PAN312x\_DK\_Develop\Projectc\MDK-ARM(AC5)

打开\Projectc\MDK-ARM(AC5) 目录下 project.uvprojx 工程, 选择 SDK\_SemiAutomaticDutyCycleRx, 如下图所示

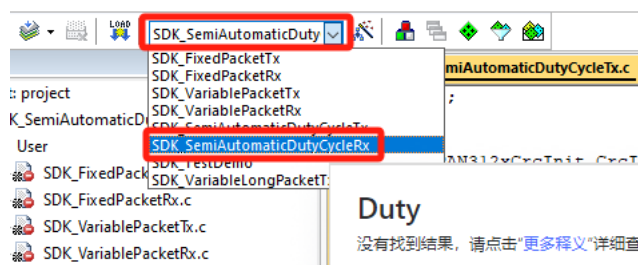


图 81: 工程选择

然后编译并下载该程序。

## 4 RF 参数参数设置

如上图所示: RF 参数目前主要包括频点、调制方式、速率、Tx Deviation、Rx Deviation、发射功率等, 其中频点由基础频点, 频点步进, 通道 3 部分组成;

举例配置频点为 493000000Hz, 调制方式为 2FSK, 速率为 50000bps, Tx Deviation 为 25000Hz, Rx Deviation 为 25000Hz, 发射功率为 21dBm, 如下所示:

```
typedef struct{
    uint32_t FrequencyBase;
    uint32_t FrequencyStep;
    uint16_t FrequencyChannelNumber;
    PAN312xModulationSelect ModulationSelect;
    uint32_t DataRate;
    uint32_t TxDeviation;
    uint32_t RxDeviation;
    PAN312xPowerdBm Power;
}PAN312xRadioInit;
```

图 82: RF 参数设置

```
PAN312xRadioInit RadioInit = {
    433000000,
    0,
    0,
    MOD_2FSK,
    50000,
    25000,
    25000,
    POWER_21dBm,
};
```

图 83: RF 参数设置举例

```
typedef struct{
    uint8_t PreambleLength;
    PAN312xPreambleSelect PreambleSelect;
    uint32_t NonStandardPreamblePattern;
    uint8_t SyncwordLength;
    uint32_t SyncwordPattern;
    PAN312xManchesterMode PreambleManchesterMode;
    SFunctionalState PreambleManchester;
    PAN312xManchesterMode SyncwordManchesterMode;
    SFunctionalState SyncwordManchester;
    PAN312xManchesterMode PayloadManchesterMode;
    SFunctionalState PayloadManchester;
    PAN312xFecEncoding FecEncoding;
    PAN312xWhiteningPattern WhiteningPattern;
    PAN312xDataMode DataMode;
}PAN312xPacketInit;
```

图 84: 帧结构参数设置

## 5 帧结构参数设置

如上图所示：帧结构参数目前主要包前导码长度，前导码内容，同步字长度，同步字内容，前导码、同步字和数据区域的 manchester 编码是否模式，以及 manchester 编码是否使能，Fec 编码，白化选择等参数；

举例配置帧结构，如下图示：

```
PAN321xPacketInit PacketInit = {
    4,
    PREAMBLE_0101,
    0,
    4,
    0x2dd42dd4,
    MANCHESTER_ZeroToTwo,
    S_DISABLE,
    MANCHESTER_ZeroToTwo,
    S_DISABLE,
    MANCHESTER_ZeroToTwo,
    S_DISABLE,
    FEC_HAMING_DISABLE,
    WHITENING_DISABLE,
    DATA_MODE_PACKET,
};
```

图 85: 帧结构参数设置举例

## 6 CRC 参数设置

```
typedef struct{
    PAN312xCrcMode CrcMode;
    uint32_t CrcPolynomial;
    uint32_t CrcSeed;
    PAN312xCrcBitOrder CrcBitOrder;
    PAN312xCrcByteSwap CrcByteSwap;
    PAN312xCrcRange CrcRange;
    SFunctionalState CrcBitInv;
}PAN312xCrcInit;
```

图 86: CRC 参数设置

如上图所示：CRC 参数目前主要 CrcMode、Crc 多项式、Crc 种子、CrcBitOrder、CrcByteSwap、Crc 作用范围，Crc 输出是否取反等。

举例配置 CRC 为 CRC\_16\_IBM，如下图所示：

```
PAN312xCrcInit CrcInit = {
    CRC_MODE_16_BIT,
    0x8005,
    0xffff,
    CRC_BIT_ORDER_MSB_FIRST,
    CRC_BYTE_MSB_FIRST,
    CRC_RANGE_WHOLE_PAYLOAD,
    S_DISABLE,
};
```

图 87: CRC 参数设置举例

## 7 测试方法

- 1、复位 PAN312x、等待 PAN312x Ready；
- 2、初始化一些寄存器；
- 3、设置 RF 参数、帧结构参数、CRC 参数；
- 4、设置 RxPpacket 为**可变包长**，根据需要确认是否需要使能 CRC；

- 5、将 RX\_DONE、RX\_PREAMBLE\_DONE、RX\_CRC\_ERROR、WAKEUP 中断映射到 IRQ(GPIO8) 引脚上；
- 6、设置系统自动清除部分中断状态；
- 7、设置 RxTimeout 时间；
- 8、设置 RxTimeout、RxInvalid、RxValid 后，RF 的状态为 Rx；
- 9、设置 RF 的进入 Rx；
- 10、通过主控 MCU 来定时，控制 PAN312x 进入 Rx 的时间，该时间可理解为 PAN312x 的开窗时间，如果在开窗时间内，有收取到有效的 preamble 或 syncword，则延长 Rx 接收时间，直至收取到整个数据包；如果在开窗时间内，没有收取到有效的 preamble 或 syncword，则设置芯片进入休眠，同时计时休眠时间，待到达休眠时间，则将 PAN312x 唤醒，重新进入 Rx 开窗时间。

#### 备注：

- 1、若使能设置系统自动清除部分中断状态，除了 IRQ\_RX\_DONE、IRQ\_RX\_CRC\_ERROR、IRQ\_TX\_FIFO\_TH、IRQ\_RX\_FIFO\_TH 这些中断状态，需要额外发送命令来清除，其它中断状态都可以由 PAN312x 自动清除掉；
- 2、关于 int PAN312x\_Enter\_Rx(uint16\_t channel, uint8\_t condition, uint16\_t rx\_len)，需要注意，当设置 PAN312x 的帧结构为固定包长时，第三个参数 rx\_len 为实际发送的数据长度，当设置 PAN312x 的帧结构为可变包长时，第三个参数 rx\_len 为 0；

## 8 测试现象

- 1、测试功耗如下图所示：

| SemiAutomaticDutyCycleRx |              |              |     |          |          |              |
|--------------------------|--------------|--------------|-----|----------|----------|--------------|
| Rx Window<br>(ms)        | 休眠时间<br>(ms) | 工作周期<br>(ms) | 模式  | 峰值电流(mA) | 休眠电流(uA) | 平均电流<br>(mA) |
| 200                      | 1000         | 1200         | LDO | 14.19    | 0.92     | 1.48         |

图 88: SemiAutomaticDutyCycleRx 功耗

### 3.1.9 SDK\_SemiAutomaticDutyCycleTx

#### 1 功能概述

本代码示例主要演示通过外部 MCU 结合 PAN312x 的低功耗特性，如何实现 DutyCycleTx 功能。

#### 备注：

- 1、通过主控 MCU 来定时，控制 PAN312x 进入休眠的时间，待达到休眠的时间后，通过 spi 将 PAN312x 唤醒，待唤醒 PAN312x 后，然后在发送数据，发送数据的这段时间依赖于发送数据的长度，速率等参数；

#### 2 环境要求

- Board: PAN312x\_EVB
- PAN312x 模组
- USB 转串口模块
- PC 端串口数据显示上位机



### 3 编译和烧录

例程位置：01\_SDK\PAN312x\_DK\_Develop\Projectc\MDK-ARM(AC5)

打开\Projectc\MDK-ARM(AC5) 目录下 project.uvprojx 工程，选择 SDK\_SemiAutomaticDutyCycleTx，如下图所示

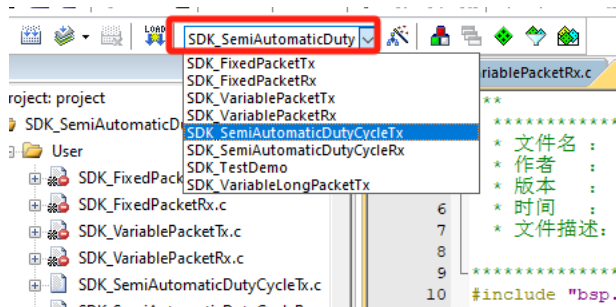


图 89: 工程选择

然后编译并下载该程序。

### 4 RF 参数参数设置

```
typedef struct{
    uint32_t FrequencyBase;
    uint32_t FrequencyStep;
    uint16_t FrequencyChannelNumber;
    PAN312xModulationSelect ModulationSelect;
    uint32_t DataRate;
    uint32_t TxDeviation;
    uint32_t RxDeviation;
    PAN312xPowerdBm Power;
}PAN312xRadioInit;
```

图 90: RF 参数设置

如上图所示：RF 参数目前主要包括频点、调制方式、速率、Tx Deviation、Rx Deviation、发射功率等，其中频点由基础频点，频点步进，通道 3 部分组成；

举例配置频点为 493000000Hz，调制方式为 2FSK，速率为 50000bps，Tx Deviation 为 25000Hz，Rx Deviation 为 25000Hz，发射功率为 21dBm，如下所示：

```
PAN312xRadioInit RadioInit = {
    493000000,
    0,
    0,
    MOD_2FSK,
    50000,
    25000,
    25000,
    POWER_21dBm,
};
```

图 91: RF 参数设置举例

### 5 帧结构参数设置

如上图所示：帧结构参数目前主要包前导码长度，前导码内容，同步字长度，同步字内容，前导码、同步字和数据区域的 manchester 编码是否模式，以及 manchester 编码是否使能，Fec 编码，白化选择等参数；

举例配置帧结构，如下图示：

```
typedef struct{
    uint8_t PreambleLength;
    PAN312xPreambleSelect PreambleSeclect;
    uint32_t NonStandardPreamblePattern;
    uint8_t SyncwordLength;
    uint32_t SyncwordPattern;
    PAN312xManchesterMode PreambleManchesterMode;
    SFunctionalState PreambleManchester;
    PAN312xManchesterMode SyncwordManchesterMode;
    SFunctionalState SyncwordManchester;
    PAN312xManchesterMode PayloadManchesterMode;
    SFunctionalState PayloadManchester;
    PAN312xFecEncoding FecEncoding;
    PAN312xWhiteningPattern WhiteningPattern;
    PAN312xDataMode DataMode;
}PAN321xPacketInit;
```

图 92: 帧结构参数设置

```
PAN321xPacketInit PacketInit = {
    4,
    PREAMBLE_0101,
    0,
    4,
    0x2dd42dd4,
    MANCHESTER_ZeroToTwo,
    S_DISABLE,
    MANCHESTER_ZeroToTwo,
    S_DISABLE,
    MANCHESTER_ZeroToTwo,
    S_DISABLE,
    FEC_HAMING_DISABLE,
    WHITENING_DISABLE,
    DATA_MODE_PACKET,
};
```

图 93: 帧结构参数设置举例

## 6 CRC 参数设置

```
typedef struct{
    PAN312xCrcMode CrcMode;
    uint32_t CrcPolynomial;
    uint32_t CrcSeed;
    PAN312xCrcBitOrder CrcBitOrder;
    PAN312xCrcByteSwap CrcByteSwap;
    PAN312xCrcRange CrcRange;
    SFunctionalState CrcBitInv;
}PAN312xCrcInit;
```

图 94: CRC 参数设置

如上图所示: CRC 参数目前主要 CrcMode、Crc 多项式、Crc 种子、CrcBitOrder、CrcByteSwap、Crc 作用范围, Crc 输出是否取反等。

举例配置 CRC 为 CRC\_16\_IBM, 如下图所示:

## 7 测试方法

- 1、复位 PAN312x、等待 PAN312x Ready;
- 2、初始化一些寄存器;
- 3、设置 RF 参数、帧结构参数、CRC 参数;
- 4、设置 TxPpacket 为**可变包长**, 根据需要确认是否需要使能 CRC;
- 5、将 TX\_DONE、WAKEUP 中断映射到 IRQ(GPIO8) 引脚上;
- 6、设置系统自动清除部分中断状态;

```

PAN312xCrcInit CrcInit = {
    CRC_MODE_16_BIT,
    0x8005,
    0xffff,
    CRC_BIT_ORDER_MSB_FIRST,
    CRC_BYTE_MSB_FIRST,
    CRC_RANGE_WHOLE_PAYLOAD,
    S_DISABLE,
};

```

图 95: CRC 参数设置举例

7、清除所有中断状态；

8、通过主控 MCU 来定时，控制 PAN312x 进入休眠的时间，待达到休眠的时间后，通过 spi 将 PAN312x 唤醒，待唤醒 PAN312x 后，然后在发送数据，发送数据的这段时间依赖于发送数据的长度，速率等参数：

**备注：**

1、若使能设置系统自动清除部分中断状态，除了 IRQ\_RX\_DONE、IRQ\_RX\_CRC\_ERROR、IRQ\_TX\_FIFO\_TH、IRQ\_RX\_FIFO\_TH 这些中断状态，需要额外发送命令来清除，其它中断状态都可以由 PAN312x 自动清除掉；

## 8 测试现象

1、测试功耗如下图所示：

| SemiAutomaticDutyCycleTx |                      |              |              |              |                   |     |          |              |              |
|--------------------------|----------------------|--------------|--------------|--------------|-------------------|-----|----------|--------------|--------------|
| DataRate<br>(kbps)       | DataLength<br>(byte) | 发送时间<br>(ms) | 休眠时间<br>(ms) | 工作周期<br>(ms) | tx power<br>(dbm) | 模式  | 峰值电流(mA) | 休眠电流<br>(uA) | 平均电流<br>(uA) |
| 50                       | 20                   | 3.2          | 1000         | 1003.2       | 21                | LDO | 163.88   | 0.89         | 648.79       |
| 50                       | 20                   | 3.2          | 1000         | 1003.2       | 15                | LDO | 72.79    | 0.88         | 291.23       |

图 96: SemiAutomaticDutyCycleTx 功耗

### 3.1.10 SDK\_TestDemo

#### 1 功能概述

本代码示例主要通过评估板 (PAN312x\_EVB) 来评估 PAN312x 芯片的主要性能，如通信测试，单载波测试，距离测试，丢包率测试，功耗测试等。

#### 2 评估板 (PAN312x\_EVB)

评估板主要部件如下图所示：

#### 3 无线模块 (PAN312x\_Module)

无线模块主要部件如下图所示：

#### 4 编译和烧录

例程位置：01\_SDK\PAN312x\_DK\_Develop\Projectc\MDK-ARM(AC5)

打开\Projectc\MDK-ARM(AC5) 目录下 project.uvprojx 工程，选择 SDK\_TestDemo，如下图所示然后编译并下载该程序。

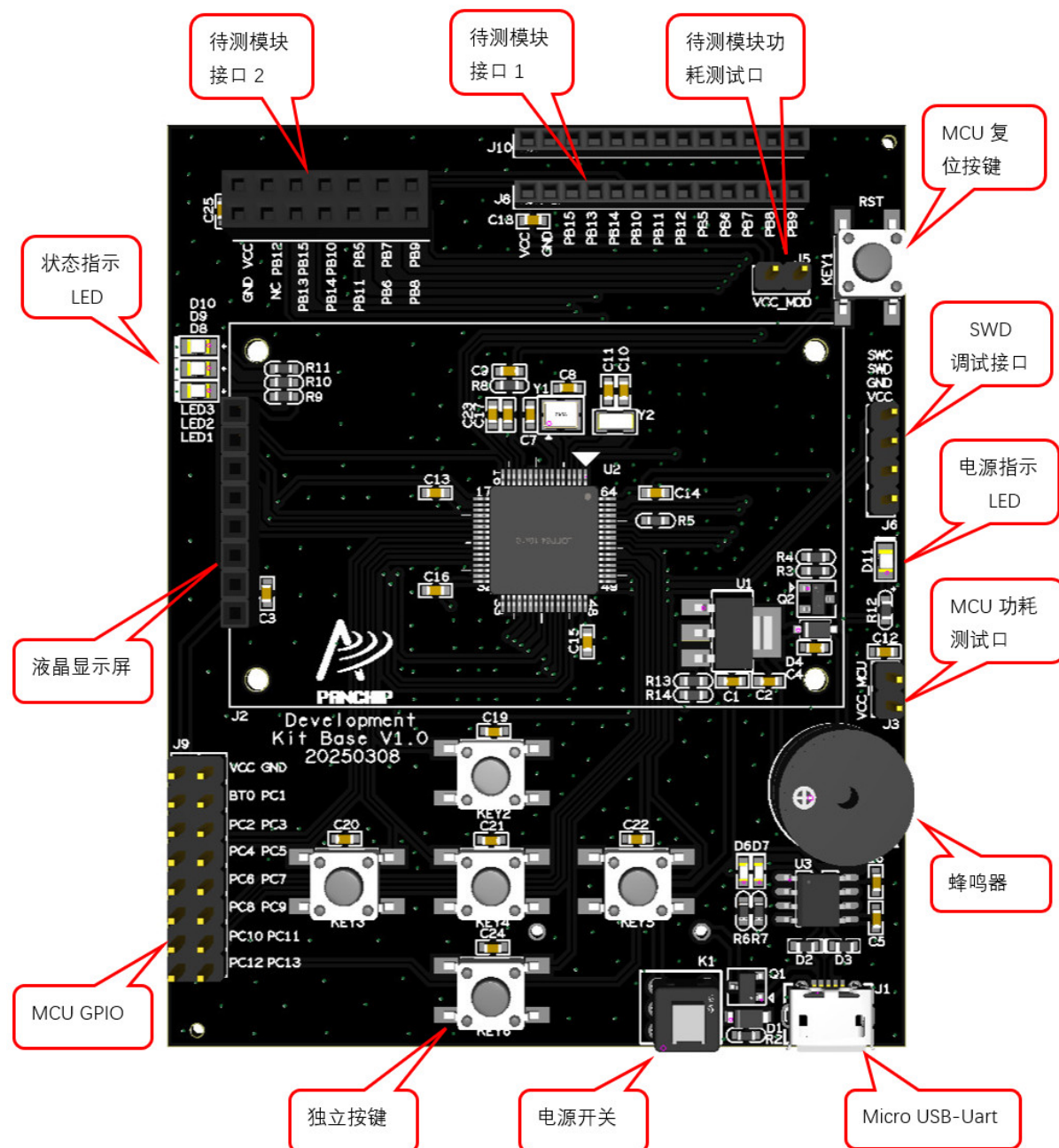


图 97: PAN312x\_EVB

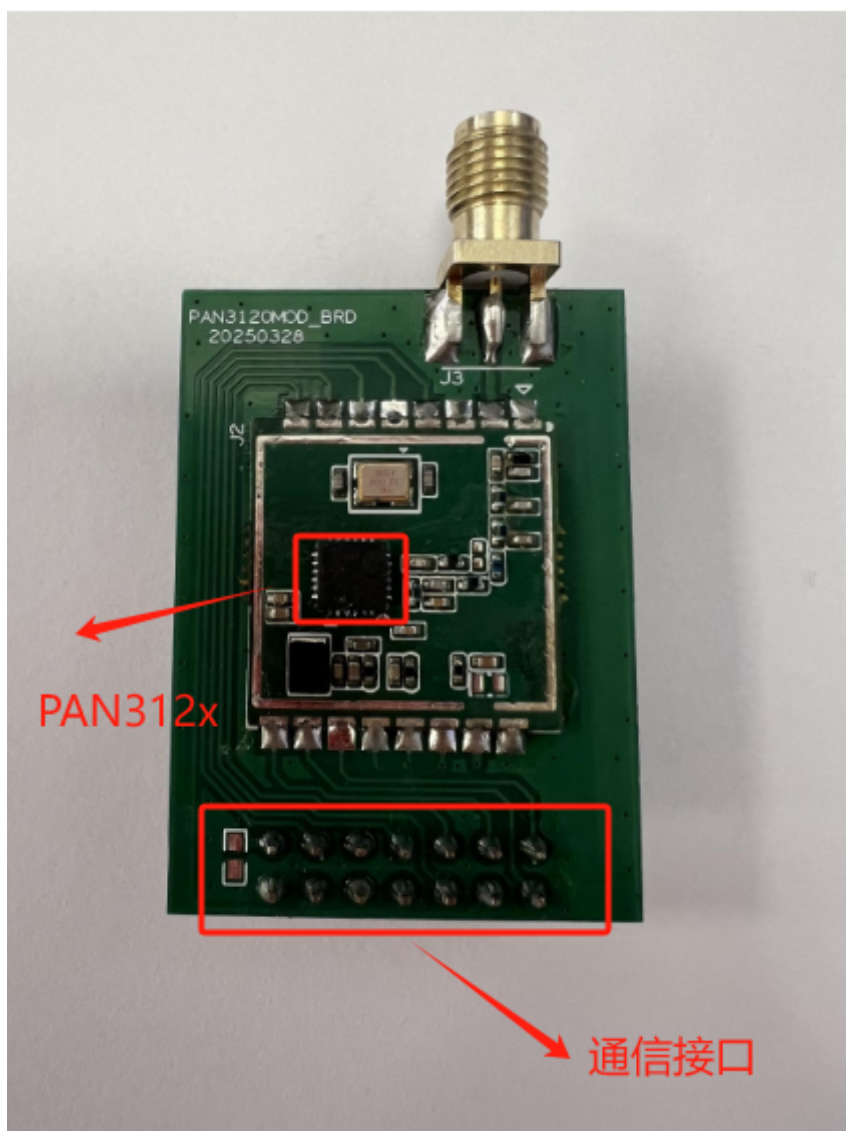


图 98: PAN312x\_Module

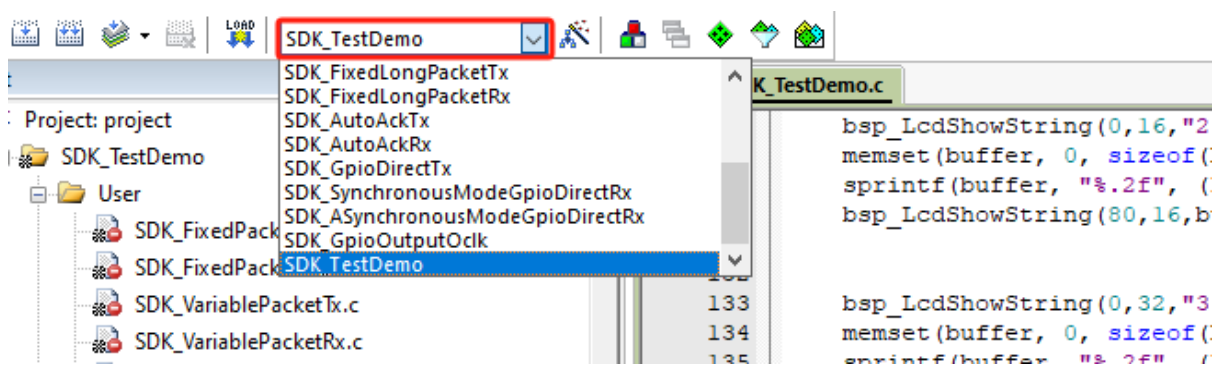


图 99: 工程选择

## 5 评估板操作指南

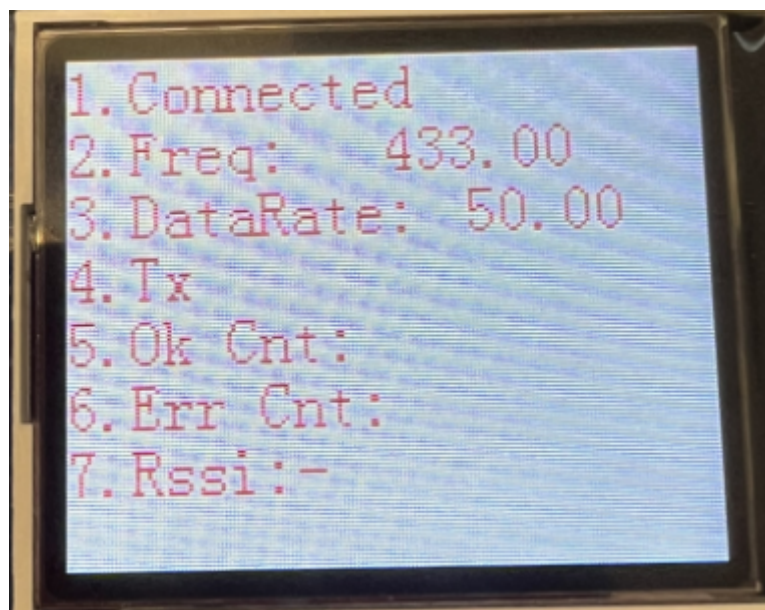


图 100: LCD 菜单显示

### 5.1 LCD 菜单显示:

5.2 **按键功能:** 1、短按 key2 按键, 单次发送数据包;

2、长按 key2 按键, 一直发送数据包;

3、短按 key3 按键, 发送数据包 500 次;

4、短按 key4 按键, 切换模式, 包括 Rx、Tx、TRxMaster、TRxSlave

TRxMaster: 设备 Master 发送完数据后, 进入接收模式, 并等待 Slave 设备回应的 ack 包

TRxSlave: 设备 Slave 接收到数据后, 进入发送模式, 并发送 ack 包给 Master 设备

5、长按 key4 按键, 清除计数值;

6、短按 key5 按键, 切换速率, 目前可配置的速率如下:



| DataRate(Kbps) | TRxDeviation(KHz) |
|----------------|-------------------|
| 2.4            | 10                |
| 2.5            | 10                |
| 4.8            | 10                |
| 5              | 10                |
| 9.6            | 10                |
| 10             | 10                |
| 20             | 10                |
| 32.8           | 16.4              |
| 40             | 20                |
| 50             | 25                |
| 62.5           | 31.25             |
| 80             | 40                |
| 100            | 50                |
| 125            | 62.5              |
| 200            | 100               |
| 250            | 125               |
| 400            | 200               |
| 500            | 250               |

7、短按 key6 按键，切换频点，目前可配置频点为：315Mhz、317Mhz、433.92Mhz、436Mhz、470Mhz、840Mhz、868Mhz、915Mhz、920Mhz、

备注：

1、每次切换完速率和频点后，需要通过短按 key4 按键，切换芯片的模式；

### 3.1.11 SDK\_VariableLongPacketRx

#### 1 功能概述

本代码示例主要演示 PAN312x 可变包长帧结构怎么实现长包接收。

备注：1、当接收的字节数长度大于 fifo 长度 (128bytes) 时，可采用 rx fifo 阈值的方式来实现长包接收；2、当前 rx fifo 阈值固定为 64 bytes；

#### 2 环境要求

- Board: PAN312x\_EVB
- PAN312x 模组
- USB 转串口模块
- PC 端串口数据显示上位机

#### 3 编译和烧录

例程位置：01\_SDK\PAN312x\_DK\_Develop\Projectc\MDK-ARM(AC5)

打开\Projectc\MDK-ARM(AC5) 目录下 project.uvprojx 工程，选择 SDK\_VariableLongPacketRx，如下图所示

然后编译并下载该程序。

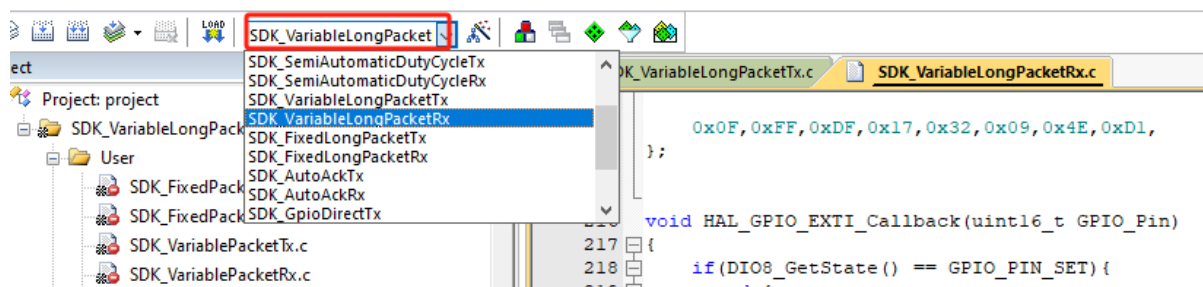


图 101: 工程选择

```
typedef struct{
    uint32_t FrequencyBase;
    uint32_t FrequencyStep;
    uint16_t FrequencyChannelNumber;
    PAN312xModulationSelect ModulationSelect;
    uint32_t DataRate;
    uint32_t TxDeviation;
    uint32_t RxDeviation;
    PAN312xPowerdBm Power;
}PAN312xRadioInit;
```

图 102: RF 参数设置

#### 4 RF 参数设置

如上图所示：RF 参数目前主要包括频点、调制方式、速率、Tx Deviation、Rx Deviation、发射功率等，其中频点由基础频点，频点步进，通道 3 部分组成；

举例配置频点为 493000000Hz，调制方式为 2FSK，速率为 50000bps，Tx Deviation 为 25000Hz，Rx Deviation 为 25000Hz，发射功率为 21dBm，如下所示：

```
PAN312xRadioInit RadioInit = {
    493000000,
    0,
    0,
    MOD_2FSK,
    50000,
    25000,
    25000,
    POWER_21dBm,
};
```

图 103: RF 参数设置举例

#### 5 帧结构参数设置

如上图所示：帧结构参数目前主要包前导码长度，前导码内容，同步字长度，同步字内容，前导码、同步字和数据区域的 manchester 编码是否模式，以及 manchester 编码是否使能，Fec 编码，白化选择等参数；

举例配置帧结构，如下图示：

#### 6 CRC 参数设置

如上图所示：CRC 参数目前主要 CrcMode、Crc 多项式、Crc 种子、CrcBitOrder、CrcByteSwap、Crc 作用范围，Crc 输出是否取反等。

举例配置 CRC 为 CRC\_16\_IBM，如下图所示：



```
typedef struct{
    uint8_t PreambleLength;
    PAN312xPreambleSelect PreambleSeclect;
    uint32_t NonStandardPreamblePattern;
    uint8_t SyncwordLength;
    uint32_t SyncwordPattern;
    PAN312xManchesterMode PreambleManchesterMode;
    SFunctionalState PreambleManchester;
    PAN312xManchesterMode SyncwordManchesterMode;
    SFunctionalState SyncwordManchester;
    PAN312xManchesterMode PayloadManchesterMode;
    SFunctionalState PayloadManchester;
    PAN312xFecEncoding FecEncoding;
    PAN312xWhiteningPattern WhiteningPattern;
    PAN312xDataMode DataMode;
}PAN321xPacketInit;
```

图 104: 帧结构参数设置

```
PAN321xPacketInit PacketInit = {
    4,
    PREAMBLE_0101,
    0,
    4,
    0x2dd42dd4,
    MANCHESTER_ZeroToTwo,
    S_DISABLE,
    MANCHESTER_ZeroToTwo,
    S_DISABLE,
    MANCHESTER_ZeroToTwo,
    S_DISABLE,
    FEC_HAMING_DISABLE,
    WHITENING_DISABLE,
    DATA_MODE_PACKET,
};
```

图 105: 帧结构参数设置举例

```
typedef struct{
    PAN312xCrcMode CrcMode;
    uint32_t CrcPolynomial;
    uint32_t CrcSeed;
    PAN312xCrcBitOrder CrcBitOrder;
    PAN312xCrcByteSwap CrcByteSwap;
    PAN312xCrcRange CrcRange;
    SFunctionalState CrcBitInv;
}PAN312xCrcInit;
```

图 106: CRC 参数设置

```
PAN312xCrcInit CrcInit = {
    CRC_MODE_16_BIT,
    0x8005,
    0xffff,
    CRC_BIT_ORDER_MSB_FIRST,
    CRC_BYTE_MSB_FIRST,
    CRC_RANGE_WHOLE_PAYLOAD,
    S_DISABLE,
};
```

图 107: CRC 参数设置举例

## 7 测试方法

- 1、复位 PAN312x、等待 PAN312x Ready;
- 2、初始化一些寄存器;
- 3、设置 RF 参数、帧结构参数、CRC 参数;
- 4、设置 RxPpacket 为**可变包长**，根据需要确认是否需要使能 CRC;
- 5、将 IRQ\_MASK\_RX\_DONE、IRQ\_MASK\_RX\_LENGTH\_DONE、IRQ\_MASK\_RX\_COMPLETE、IRQ\_MASK\_RX\_FIFO\_TH 中断映射到 IRQ(GPIO8) 引脚上;
- 6、设置系统自动清除部分中断状态;
- 7、设置 RxTimeout 时间;
- 8、设置 RxTimeout、RxInvalid、RxValid 后，RF 的状态为 Rx;
- 9、设置 RF 的进入 Rx;
- 9、长包发送处理逻辑，如下图所示::

```
while (1)
{
    nRxIndex = 0;
    nResidualPcktLength = 0;

    nPayloadLength = 0;

    while(!xRxDoneFlag);
    xRxDoneFlag = S_RESET;

    rssi_value = PAN312x_GetRSSI();

    printf("Rssi Value = %d\r\n", rssi_value);
    printf("Payload length = %d\r\n", nPayloadLength);
    /* print the received data */
    printf("B data received: [");
    for(uint8_t i = 0; i < nPayloadLength ; i++)
        printf("%02x ", vectRxBuffer[i]);
    printf("]\r\n");

    if(memcmp(vectRxBuffer, ExpectedData, nPayloadLength) == 0){
        printf("data correct\r\n");
    }else{
        printf("data error\r\n");
    }
}
```

图 108: 可变包长帧结构接收主函数处理逻辑

### 备注:

- 1、若使能设置系统自动清除部分中断状态，除了 IRQ\_RX\_DONE、IRQ\_RX\_CRC\_ERROR、IRQ\_TX\_FIFO\_TH、IRQ\_RX\_FIFO\_TH 这些中断状态，需要额外发送命令来清除，其它中断状态都可以由 PAN312x 自动清除掉;
- 2、关于 int PAN312x\_Enter\_Rx(uint16\_t channel, uint8\_t condition, uint16\_t rx\_len)，需要注意，当设置 PAN312x 的帧结构为固定包长时，第三个参数 rx\_len 为实际发送的数据长度，当设置 PAN312x 的帧结构为可变包长时，第三个参数 rx\_len 为 0;

```

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    if(DIOS_GetState() == GPIO_PIN_SET){
        do{
            PAN312x_Irq_Get_Status(&xIrqStatus);
            if(xIrqStatus.IRQ_STATUS0_F.IRQ_RX_LENGTH_DONE){
                nPayloadLength = PAN312x_Get_Rx_Length();
                nResidualPcktLength = nPayloadLength;
                printf("nPayloadLength: %d\r\n", nPayloadLength);
            }

            if(xIrqStatus.IRQ_STATUS2_F.IRQ_RX_FIFO_TH){
                //printf("nResidualPcktLength:%d\r\n", nResidualPcktLength);
                if(nResidualPcktLength >= cThresholdRxFifoAF){
                    nResidualPcktLength -= cThresholdRxFifoAF;
                    PAN312x_Read_Fifo(&vectRxBuffer[nRxIndex], 64);
                    nRxIndex += cThresholdRxFifoAF;
                }
                PAN312x_Irq_Clear_RxFifo_Status();
            }

            if(xIrqStatus.IRQ_STATUS0_F.IRQ_RX_COMPLETE){
                printf("rx complete\r\n");
                if(nResidualPcktLength <= cThresholdRxFifoAF){
                    PAN312x_Read_Fifo(&vectRxBuffer[nRxIndex], nResidualPcktLength);
                    nResidualPcktLength = 0;
                }
            }

            if(xIrqStatus.IRQ_STATUS0_F.IRQ_RX_DONE){
                //rx done
                printf("rx done\r\n");
                xRxDoneFlag = S_SET;
                PAN312x_Irq_Clear_RxDone_Status();
                break;
            }

            if(xIrqStatus.IRQ_STATUS0_F.IRQ_RX_CRC_ERROR){
                PAN312x_Irq_Clear_RxCrcError_Status();
                xRxDoneFlag = S_SET;
                break;
            }
        }while(DIOS_GetState() == GPIO_PIN_SET);
    }
}

```

图 109: 可变包长帧结构接收中断处理逻辑

### 3.1.12 SDK\_VariableLongPacketTx

#### 1 功能概述

本代码示例主要演示 PAN312x 可变包长帧结构怎么实现长包发送。

备注：1、当发送的字节数长度大于 fifo 长度 (128bytes) 时，可采用 tx fifo 阈值的方式来实  
现长包发送；2、当前 tx fifo 阈值固定为 64 bytes；

#### 2 环境要求

- Board: PAN312x\_EVB
- PAN312x 模组
- USB 转串口模块
- PC 端串口数据显示上位机

#### 3 编译和烧录

例程位置：01\_SDK\PAN312x\_DK\_Develop\Projectc\MDK-ARM(AC5)

打开\Projectc\MDK-ARM(AC5) 目录下 project.uvprojx 工程，选择 SDK\_VariableLongPacketTx，如  
下图所示

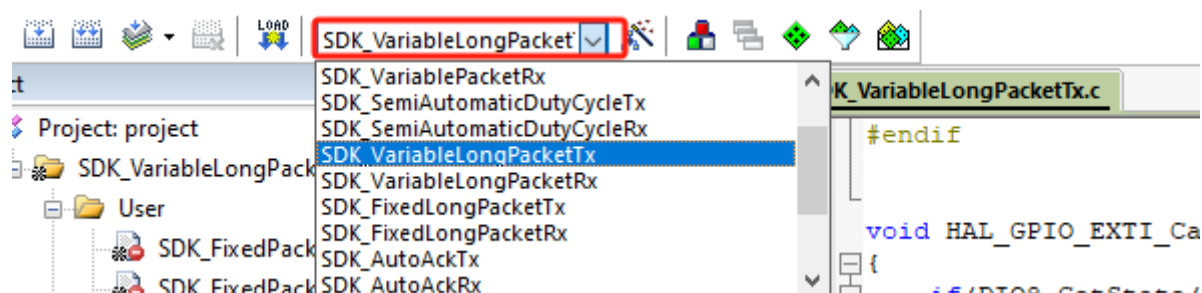


图 110: 工程选择

然后编译并下载该程序。

#### 4 RF 参数参数设置

```
typedef struct{
    uint32_t FrequencyBase;
    uint32_t FrequencyStep;
    uint16_t FrequencyChannelNumber;
    PAN312xModulationSelect ModulationSelect;
    uint32_t DataRate;
    uint32_t TxDeviation;
    uint32_t RxDeviation;
    PAN312xPowerdBm Power;
} PAN312xRadioInit;
```

图 111: RF 参数设置

如上图所示：RF 参数目前主要包括频点、调制方式、速率、Tx Deviation、Rx Deviation、发射功率等，其中频点由基础频点，频点步进，通道 3 部分组成；

举例配置频点为 493000000Hz，调制方式为 2FSK，速率为 50000bps，Tx Deviation 为 25000Hz，Rx Deviation 为 25000Hz，发射功率为 21dBm，如下所示：

```
PAN312xRadioInit RadioInit = {
    433000000,
    0,
    0,
    MOD_2FSK,
    50000,
    25000,
    25000,
    POWER_21dBm,
};
```

图 112: RF 参数设置举例

## 5 帧结构参数设置

```
typedef struct{
    uint8_t PreambleLength;
    PAN312xPreambleSelect PreambleSeclect;
    uint32_t NonStandardPreamblePattern;
    uint8_t SyncwordLength;
    uint32_t SyncwordPattern;
    PAN312xManchesterMode PreambleManchesterMode;
    SFunctionalState PreambleManchester;
    PAN312xManchesterMode SyncwordManchesterMode;
    SFunctionalState SyncwordManchester;
    PAN312xManchesterMode PayloadManchesterMode;
    SFunctionalState PayloadManchester;
    PAN312xFecEncoding FecEncoding;
    PAN312xWhiteningPattern WhiteningPattern;
    PAN312xDataMode DataMode;
}PAN321xPacketInit;
```

图 113: 帧结构参数设置

如上图所示：帧结构参数目前主要包前导码长度，前导码内容，同步字长度，同步字内容，前导码、同步字和数据区域的 manchester 编码是否模式，以及 manchester 编码是否使能，Fec 编码，白化选择等参数；

举例配置帧结构，如下图示：

```
PAN321xPacketInit PacketInit = {
    4,
    PREAMBLE_0101,
    0,
    4,
    0x2dd42dd4,
    MANCHESTER_ZeroToTwo,
    S_DISABLE,
    MANCHESTER_ZeroToTwo,
    S_DISABLE,
    MANCHESTER_ZeroToTwo,
    S_DISABLE,
    FEC_HAMING_DISABLE,
    WHITENING_DISABLE,
    DATA_MODE_PACKET,
};
```

图 114: 帧结构参数设置举例

## 6 CRC 参数设置

如上图所示：CRC 参数目前主要 CrcMode、Crc 多项式、Crc 种子、CrcBitOrder、CrcByteSwap、Crc 作用范围，Crc 输出是否取反等。

举例配置 CRC 为 CRC\_16\_IBM，如下图所示：

```

typedef struct{
    PAN312xCrcMode CrcMode;
    uint32_t CrcPolynomial;
    uint32_t CrcSeed;
    PAN312xCrcBitOrder CrcBitOrder;
    PAN312xCrcByteSwap CrcByteSwap;
    PAN312xCrcRange CrcRange;
    SFunctionalState CrcBitInv;
}PAN312xCrcInit;

```

图 115: CRC 参数设置

```

PAN312xCrcInit CrcInit = {
    CRC_MODE_16_BIT,
    0x8005,
    0xffff,
    CRC_BIT_ORDER_MSB_FIRST,
    CRC_BYTE_MSB_FIRST,
    CRC_RANGE_WHOLE_PAYLOAD,
    S_DISABLE,
};

```

图 116: CRC 参数设置举例

## 7 测试方法

- 1、复位 PAN312x、等待 PAN312x Ready;
- 2、初始化一些寄存器;
- 3、设置 RF 参数、帧结构参数、CRC 参数;
- 4、设置 TxPpacket 为**可变包长**, 根据需要确认是否需要使能 CRC;
- 5、将 IRQ\_MASK\_TX\_DONE 和 IRQ\_MASK\_TX\_FIFO\_TH 中断映射到 IRQ(GPIO8) 引脚上;
- 6、设置系统自动清除部分中断状态;
- 7、清除所有中断状态;
- 8、长包发送处理逻辑, 如下图所示:

### 备注:

- 1、若使能设置系统自动清除部分中断状态, 除了 IRQ\_RX\_DONE、IRQ\_RX\_CRC\_ERROR、IRQ\_TX\_FIFO\_TH、IRQ\_RX\_FIFO\_TH 这些中断状态, 需要额外发送命令来清除, 其它中断状态都可以由 PAN312x 自动清除掉;

### 3.1.13 SDK\_VariablePacketRx

#### 1 功能概述

本代码示例主要演示 PAN312x 可变包长帧结构的接收功能。

#### 2 环境要求

- Board: PAN312x\_EVB
- PAN312x 模组
- USB 转串口模块
- PC 端串口数据显示上位机

```

while(1){
    nResidualPcktLength = nPayloadLength;
    nTxIndex = 0;
    fitTxBuffer();
    printf("\n\rPayload length = %d\n\n\r", nPayloadLength);

    printf("A data to transmit: [");
    for(uint16_t i = 0; i < nPayloadLength; i++){
        printf("%x ", vectTxBuffer[i]);
    }
    printf("]\r\n");

    if(nPayloadLength > DEV_FIFO_SIZE){
        if((nPayloadLength - DEV_FIFO_SIZE)%256 == 0){
            TxFifoSize = DEV_FIFO_SIZE - 2;
        }else{
            TxFifoSize = DEV_FIFO_SIZE - 1;
        }
        PAN312x_Write_Fifo((uint8_t*)vectTxBuffer, TxFifoSize);

        /* store the number of transmitted bytes */
        nTxIndex = TxFifoSize; 当发送数据个数大于fifo长度时

        /* update the residual number of bytes to be transmitted */
        nResidualPcktLength -= TxFifoSize;
    }else{
        PAN312x_Write_Fifo((uint8_t*)vectTxBuffer, nPayloadLength);
        nResidualPcktLength -= nPayloadLength;
    } 当发送数据个数小于fifo长度时

    PAN312x_Enter_Tx(0, 0, nPayloadLength);
    while(!xTxDoneFlag);
    xTxDoneFlag = S_RESET;

    nPayloadLength = (nPayloadLength+1)%PAYLOAD_LENGTH_FIX;

    tx_cnt++;
    printf("\n\r tx_cnt = %d\n\n\r", tx_cnt);
    bsp_DelayMS(500);
}

```

图 117: 可变包长帧结构长包发送主循环处理逻辑



```

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    if(DIO8_GetState() == GPIO_PIN_SET){
        PAN312x_Irq_Get_Status(&xIrqStatus);
        if(xIrqStatus.IRQ_STATUS0_F.IRQ_TX_DONE){ // Tx 完成状态判断
            xTxDoneFlag = S_SET;
            printf("irq tx done\r\n");
        }
        if(xIrqStatus.IRQ_STATUS2_F.IRQ_TX_FIFO_TH){ // Tx 阈值状态判断
            printf("irq tx fifo th\r\n");
            if(nResidualPcktLength == 0){
                printf("0 nResidualPcktLength: %d\r\n", nResidualPcktLength);
                xIrqStatus.IRQ_STATUS2 |= 0x02;
            }else if((nResidualPcktLength + THRESHOLD_TX_FIFO) >= DEV_FIFO_SIZE){
                printf("1 nResidualPcktLength: %d\r\n", nResidualPcktLength);
                PAN312x_Write_Fifo(&vectTxBuffer[nTxIndex], DEV_FIFO_SIZE - THRESHOLD_TX_FIFO);
                nResidualPcktLength -= (DEV_FIFO_SIZE - THRESHOLD_TX_FIFO);
                nTxIndex += (DEV_FIFO_SIZE - THRESHOLD_TX_FIFO);
            }else{
                printf("2 nResidualPcktLength: %d\r\n", nResidualPcktLength);

                PAN312x_Write_Fifo(&vectTxBuffer[nTxIndex], nResidualPcktLength);
                nTxIndex += nResidualPcktLength;
                nResidualPcktLength = 0;
                xIrqStatus.IRQ_STATUS2 |= 0x02;
            }
        }
        PAN312x_Irq_Clear_TxFifo_Status(xIrqStatus);
    }
}
}

```

图 118: 可变包长帧结构长包发送中断处理逻辑

### 3 编译和烧录

例程位置: 01\_SDK\PAN312x\_DK\_Develop\Projectc\MDK-ARM(AC5)

打开\Projectc\MDK-ARM(AC5) 目录下 project.uvprojx 工程, 选择 SDK\_VariablePacketRx, 如下图所示

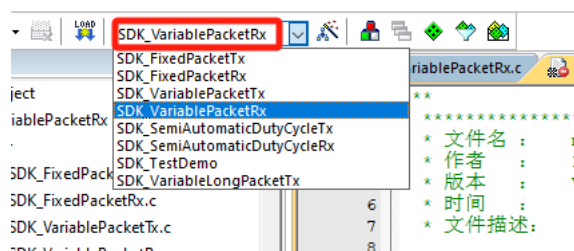


图 119: 工程选择

然后编译并下载该程序。

### 4 RF 参数参数设置

如上图所示: RF 参数目前主要包括频点、调制方式、速率、Tx Deviation、Rx Deviation、发射功率等, 其中频点由基础频点, 频点步进, 通道 3 部分组成;

举例配置频点为 493000000Hz, 调制方式为 2FSK, 速率为 50000bps, Tx Deviation 为 25000Hz, Rx Deviation 为 25000Hz, 发射功率为 21dBm, 如下所示:

```
typedef struct{
    uint32_t FrequencyBase;
    uint32_t FrequencyStep;
    uint16_t FrequencyChannelNumber;
    PAN312xModulationSelect ModulationSelect;
    uint32_t DataRate;
    uint32_t TxDeviation;
    uint32_t RxDeviation;
    PAN312xPowerdBm Power;
}PAN312xRadioInit;
```

图 120: RF 参数设置

```
PAN312xRadioInit RadioInit = {
    433000000,
    0,
    0,
    MOD_2FSK,
    50000,
    25000,
    25000,
    POWER_21dBm,
};
```

图 121: RF 参数设置举例

```
typedef struct{
    uint8_t PreambleLength;
    PAN312xPreambleSelect PreambleSelect;
    uint32_t NonStandardPreamblePattern;
    uint8_t SyncwordLength;
    uint32_t SyncwordPattern;
    PAN312xManchesterMode PreambleManchesterMode;
    SFunctionalState PreambleManchester;
    PAN312xManchesterMode SyncwordManchesterMode;
    SFunctionalState SyncwordManchester;
    PAN312xManchesterMode PayloadManchesterMode;
    SFunctionalState PayloadManchester;
    PAN312xFecEncoding FecEncoding;
    PAN312xWhiteningPattern WhiteningPattern;
    PAN312xDataMode DataMode;
}PAN312xPacketInit;
```

图 122: 帧结构参数设置

## 5 帧结构参数设置

如上图所示：帧结构参数目前主要包前导码长度，前导码内容，同步字长度，同步字内容，前导码、同步字和数据区域的 manchester 编码是否模式，以及 manchester 编码是否使能，Fec 编码，白化选择等参数；

举例配置帧结构，如下图示：

```
PAN321xPacketInit PacketInit = {
    4,
    PREAMBLE_0101,
    0,
    4,
    0x2dd42dd4,
    MANCHESTER_ZeroToTwo,
    S_DISABLE,
    MANCHESTER_ZeroToTwo,
    S_DISABLE,
    MANCHESTER_ZeroToTwo,
    S_DISABLE,
    FEC_HAMING_DISABLE,
    WHITENING_DISABLE,
    DATA_MODE_PACKET,
};
```

图 123: 帧结构参数设置举例

## 6 CRC 参数设置

```
typedef struct{
    PAN312xCrcMode CrcMode;
    uint32_t CrcPolynomial;
    uint32_t CrcSeed;
    PAN312xCrcBitOrder CrcBitOrder;
    PAN312xCrcByteSwap CrcByteSwap;
    PAN312xCrcRange CrcRange;
    SFunctionalState CrcBitInv;
}PAN312xCrcInit;
```

图 124: CRC 参数设置

如上图所示：CRC 参数目前主要 CrcMode、Crc 多项式、Crc 种子、CrcBitOrder、CrcByteSwap、Crc 作用范围，Crc 输出是否取反等。

举例配置 CRC 为 CRC\_16\_IBM，如下图所示：

```
PAN312xCrcInit CrcInit = {
    CRC_MODE_16_BIT,
    0x8005,
    0xffff,
    CRC_BIT_ORDER_MSB_FIRST,
    CRC_BYTE_MSB_FIRST,
    CRC_RANGE_WHOLE_PAYLOAD,
    S_DISABLE,
};
```

图 125: CRC 参数设置举例

## 7 测试方法

- 1、复位 PAN312x、等待 PAN312x Ready；
- 2、初始化一些寄存器；
- 3、设置 RF 参数、帧结构参数、CRC 参数；
- 4、设置 RxPpacket 为**可变包长**，根据需要确认是否需要使能 CRC；

- 5、将 IRQ\_MASK\_RX\_DONE 和 IRQ\_MSK\_RX\_CRC\_ERROR 中断映射到 IRQ(GPIO8) 引脚上;
- 6、设置系统自动清除部分中断状态;
- 7、设置 RxTimeout 时间;
- 8、设置 RxTimeout、RxInvalid、RxValid 后, RF 的状态为 Rx;
- 9、设置 RF 的进入 Rx;
- 10、在 while 循环中, 等待 Rx 接收完成标志, 待等到 Rx 接收完成后, 然后读取 fifo, 并清除 Rx\_Done 状态; 如下图所示:

```

while (1)
{
    /* 判断定时器超时时间 */
    if(bsp_CheckTimer(0)){
        /* 每隔1000ms进来一次 表示mcu运行正常*/
        bsp_LedToggle(1);
    }

    if(xRxDoneFlag){
        xRxDoneFlag = S_RESET;
        rx_cnt++;

        rssi_value = PAN312x_GetRSSI();

        nPayloadLength = PAN312x_Get_Rx_Length();

        PAN312x_Read_Fifo(vectRxBuffer, nPayloadLength);
        PAN312x_Irq_Clear_RxFifo_Status();
        PAN312x_Irq_Clear_RxDone_Status();
        printf("Rx Cnt = %d\r\n", rx_cnt);
        printf("Rssi Value = %d\r\n", rssi_value);
        printf("Payload length = %d\r\n", nPayloadLength);
        /* print the received data */
        printf("B data received: [");
        for(uint8_t i = 0; i < nPayloadLength; i++)
            printf("%02x ", vectRxBuffer[i]);
        printf("]\r\n");

        if(memcmp(vectRxBuffer, ExpectedData, nPayloadLength) == 0){
            //rx_cnt ++ ;
        }else{
        }
    }
}

```

图 126: 可变包长帧结构接收

**备注:**

- 1、若使能设置系统自动清除部分中断状态, 除了 IRQ\_RX\_DONE、IRQ\_RX\_CRC\_ERROR、IRQ\_TX\_FIFO\_TH、IRQ\_RX\_FIFO\_TH 这些中断状态, 需要额外发送命令来清除, 其它中断状态都可以由 PAN312x 自动清除掉;
- 2、关于 int PAN312x\_Enter\_Rx(uint16\_t channel, uint8\_t condition, uint16\_t rx\_len), 需要注意, 当设置 PAN312x 的帧结构为固定包长时, 第三个参数 rx\_len 为实际发送的数据长度, 当设置 PAN312x 的帧结构为可变包长时, 第三个参数 rx\_len 为 0;

**8 测试现象**

- 1、测试现象, 串口会打印收取到的数据, 如下图所示:
- 2、其他模组作为 Tx, 发送的数据, 如下图所示:
- 3、通过对比 Tx 和 Rx 数据, 两边是一致的, 证明 PAN312x 可变包长帧结构的接收功能正确;

**备注:** 1、测试过程中, 可以用另外一块 PAN312x 模组或其他模组作为 Tx, 配置相同的 rf 参数, 就可以收取到 PAN312x 或其他模组发送过来的数据。

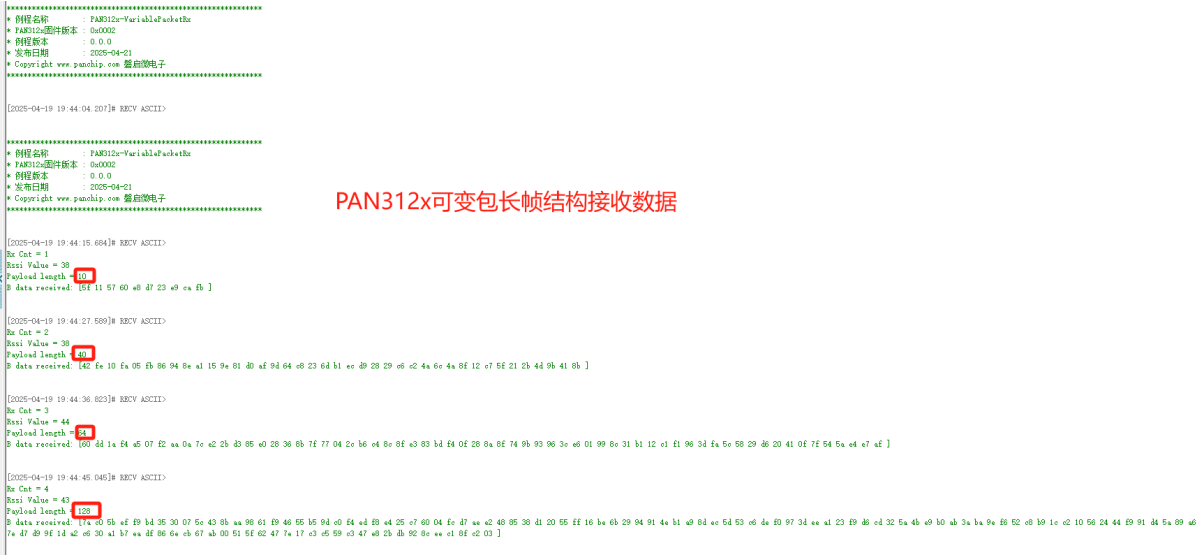


图 127: 可变包长帧结构测试现象

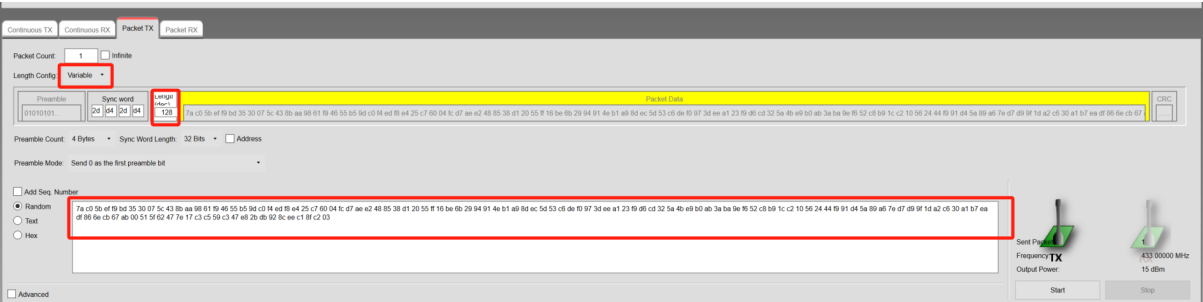


图 128: 其他模组收取到的数据

### 3.1.14 SDK\_VariablePacketTx

#### 1 功能概述

本代码示例主要演示 PAN312x 可变包长帧结构的发送功能。

#### 2 环境要求

- Board: PAN312x\_EVB
- PAN312x 模组
- USB 转串口模块
- PC 端串口数据显示上位机

#### 3 编译和烧录

例程位置：01\_SDK\PAN312x\_DK\_Develop\Projectc\MDK-ARM(AC5)

打开\Projectc\MDK-ARM(AC5) 目录下 project.uvprojx 工程，选择 SDK\_VariablePacketTx，如下图所示

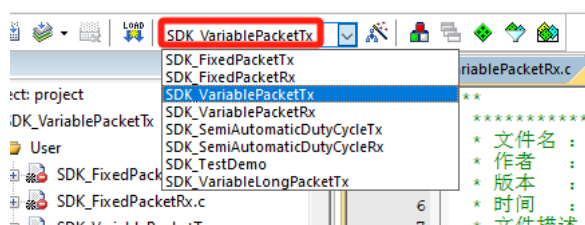


图 129: 工程选择

然后编译并下载该程序。

#### 4 RF 参数参数设置

```
typedef struct{
    uint32_t FrequencyBase;
    uint32_t FrequencyStep;
    uint16_t FrequencyChannelNumber;
    PAN312xModulationSelect ModulationSelect;
    uint32_t DataRate;
    uint32_t TxDeviation;
    uint32_t RxDeviation;
    PAN312xPowerdBm Power;
}PAN312xRadioInit;
```

图 130: RF 参数设置

如上图所示：RF 参数目前主要包括频点、调制方式、速率、Tx Deviation、Rx Deviation、发射功率等，其中频点由基础频点，频点步进，通道 3 部分组成；

举例配置频点为 493000000Hz，调制方式为 2FSK，速率为 50000bps，Tx Deviation 为 25000Hz，Rx Deviation 为 25000Hz，发射功率为 21dBm，如下所示：

```
PAN312xRadioInit RadioInit = {
    433000000,
    0,
    0,
    MOD_2FSK,
    50000,
    25000,
    25000,
    POWER_21dBm,
};
```

图 131: RF 参数设置举例

```
typedef struct{
    uint8_t PreambleLength;
    PAN312xPreambleSelect PreambleSelect;
    uint32_t NonStandardPreamblePattern;
    uint8_t SyncwordLength;
    uint32_t SyncwordPattern;
    PAN312xManchesterMode PreambleManchesterMode;
    SFunctionalState PreambleManchester;
    PAN312xManchesterMode SyncwordManchesterMode;
    SFunctionalState SyncwordManchester;
    PAN312xManchesterMode PayloadManchesterMode;
    SFunctionalState PayloadManchester;
    PAN312xFecEncoding FecEncoding;
    PAN312xWhiteningPattern WhiteningPattern;
    PAN312xDataMode DataMode;
}PAN321xPacketInit;
```

图 132: 帧结构参数设置

## 5 帧结构参数设置

如上图所示：帧结构参数目前主要包前导码长度，前导码内容，同步字长度，同步字内容，前导码、同步字和数据区域的 manchester 编码是否模式，以及 manchester 编码是否使能，Fec 编码，白化选择等参数；

举例配置帧结构，如下图示：

```
PAN321xPacketInit PacketInit = {
    4,
    PREAMBLE_0101,
    0,
    4,
    0x2dd42dd4,
    MANCHESTER_ZeroToTwo,
    S_DISABLE,
    MANCHESTER_ZeroToTwo,
    S_DISABLE,
    MANCHESTER_ZeroToTwo,
    S_DISABLE,
    FEC_HAMING_DISABLE,
    WHITENING_DISABLE,
    DATA_MODE_PACKET,
};
```

图 133: 帧结构参数设置举例

## 6 CRC 参数设置

如上图所示：CRC 参数目前主要 CrcMode、Crc 多项式、Crc 种子、CrcBitOrder、CrcByteSwap、Crc 作用范围，Crc 输出是否取反等。

举例配置 CRC 为 CRC\_16\_IBM，如下图所示：

```
typedef struct{
    PAN312xCrcMode CrcMode;
    uint32_t CrcPolynomial;
    uint32_t CrcSeed;
    PAN312xCrcBitOrder CrcBitOrder;
    PAN312xCrcByteSwap CrcByteSwap;
    PAN312xCrcRange CrcRange;
    SFunctionalState CrcBitInv;
}PAN312xCrcInit;
```

图 134: CRC 参数设置

```
PAN312xCrcInit CrcInit = {
    CRC_MODE_16_BIT,
    0x8005,
    0xffff,
    CRC_BIT_ORDER_MSB_FIRST,
    CRC_BYTE_MSB_FIRST,
    CRC_RANGE_WHOLE_PAYLOAD,
    S_DISABLE,
};
```

图 135: CRC 参数设置举例

## 7 测试方法

- 1、复位 PAN312x、等待 PAN312x Ready;
- 2、初始化一些寄存器;
- 3、设置 RF 参数、帧结构参数、CRC 参数;
- 4、设置 TxPpacket 为**可变包长**, 根据需要确认是否需要使能 CRC;
- 5、将 IRQ\_MAKS\_TX\_DONE 中断映射到 IRQ(GPIO8) 引脚上;
- 6、设置系统自动清除部分中断状态;
- 7、清除所有中断状态;
- 8、在 while 循环中, 隔一段时间往 fifo 中写入数据, 并执行发送命令, 同时更改数据长度; 如下图所示:

```
while (1)
{
    /* 判断定时器超时时间 */
    if(bsp_CheckTimer(0)){
        /* 每隔1000ms进来一次 表示mcu运行正常*/
        bsp_LedToggle(1);
    }

    if(bsp_CheckTimer(1)){
        printf("\n\rPayload length=%d\n\r", nPayloadLength);
        printf("A data to transmit: [");
        for(uint8_t i = 0; i < nPayloadLength; i++){
            printf("%x ", vectTxBuffer[i]);
        }
        printf("]\r\n");
        PAN312x_Write_Fifo((uint8_t *)vectTxBuffer, nPayloadLength);
        PAN312x_Enter_Tx(0, 0, nPayloadLength);
        while(!xTxDoneFlag);
        xTxDoneFlag = S_RESET;

        nPayloadLength = (nPayloadLength+1)%PAYLOAD_LENGTH_FIX;
    }
}
```

图 136: 可变包长帧结构发送

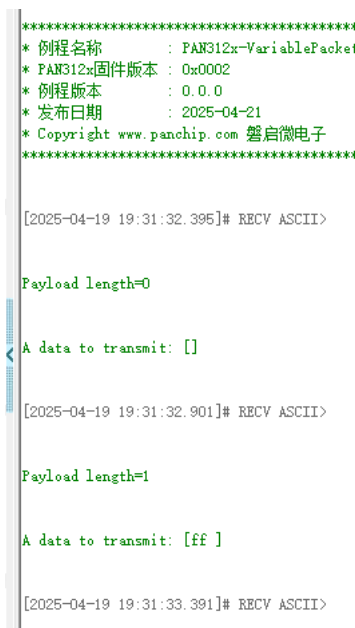
### 备注:

- 1、若使能设置系统自动清除部分中断状态, 除了 IRQ\_RX\_DONE、IRQ\_RX\_CRC\_ERROR、IRQ\_TX\_FIFO\_TH、IRQ\_RX\_FIFO\_TH 这些中断状态, 需要额外发送命令来清除, 其它中断状态都可以由 PAN312x 自动清除掉;



## 8 测试现象

1、测试现象，串口会打印将要发送出去的数据，如下图所示：



```
*****
* 例程名称       : PAN312x-VariablePacketTx
* PAN312x固件版本 : 0x0002
* 例程版本       : 0.0.0
* 发布日期       : 2025-04-21
* Copyright www.panchip.com 磐启微电子
*****

[2025-04-19 19:31:32.395]# RECV ASCII>

Payload length=0

A data to transmit: []

[2025-04-19 19:31:32.901]# RECV ASCII>

Payload length=1

A data to transmit: [ff]

[2025-04-19 19:31:33.391]# RECV ASCII>
```

可变包长帧结构发送，  
长度(bytes): 0~128

图 137: 可变包长帧结构测试现象

2、其他模组作为 Rx，接收 PAN312x 发送的数据，如下图所示：



图 138: 其他模组收取到的数据

3、通过对比 Tx 和 Rx 数据，两边是一致的，证明 PAN312x 可变包长帧结构的发送功能正确；

**备注：**1、测试过程中，可以用另外一块 PAN312x 模组或其他模组作为 Rx，配置相同的 rf 参数，就可以收取到 PAN312x 发送过来的数据。

源码路径: PAN312x\_DK/01\_SDK/PAN312x\_DK\_Develop

- [Basic: SDK\\_CTK\\_PacketTx](#): 演示如何通过 PANCHIP\_RF\_CTK 生成 rf 配置，并实现数据发送的功能。
- [Basic: SDK\\_CTK\\_PacketRx](#): 演示如何通过 PANCHIP\_RF\_CTK 生成 rf 配置，并实现数据接收的功能。
- [Basic: SDK\\_FixedLongPacketTx](#): 演示固定包长帧结构长包发送功能。
- [Basic: SDK\\_FixedLongPacketRx](#): 演示固定包长帧结构长包接收功能。
- [Basic: SDK\\_FixedPacketTx](#): 演示固定包长帧结构发送功能。
- [Basic: SDK\\_FixedPacketRx](#): 演示固定包长帧结构接收功能。

- Basic: [SDK\\_GpioOutPutOclk](#): 演示 PAN312x 的 GPIO 如何输出时钟信号。
- Basic: [SDK\\_SemiAutomaticDutyCycleTx](#): 演示通过外部 MCU 结合 PAN312x 的低功耗特性, 如何实现 DutyCycleTx 功能。
- Basic: [SDK\\_SemiAutomaticDutyCycleRx](#): 演示通过外部 MCU 结合 PAN312x 的低功耗特性, 如何实现 DutyCycleRx 功能。
- Basic: [SDK\\_TestDemo](#): 演示主要通过评估板 (PAN312x\_EVB) 来评估 PAN312x 芯片的主要性能, 如通信测试, 单载波测试, 距离测试, 丢包率测试, 功耗测试等。
- Basic: [SDK\\_VariableLongPacketTx](#): 演示可变包长帧结构长包发送功能。
- Basic: [SDK\\_VariableLongPacketRx](#): 演示可变包长帧结构长包接收功能。
- Basic: [SDK\\_VariablePacketTx](#): 演示可变包长帧结构发送功能。
- Basic: [SDK\\_VariablePacketRx](#): 演示可变包长帧结构接收功能。



## Chapter 4

# 开发指南



## Chapter 5

# 量产测试



# Chapter 6

## 开发工具

### 6.1 PANCHIP-RF-CTK

PANCHIP-RF-CTK 是一个为部分 RF 芯片提供图形界面配置的工具。

#### 6.1.1 基本 UI 介绍



图 1: UI 展示

菜单栏





图 2: 文件

文件

| 菜单     | 说明             |
|--------|----------------|
| 保存     | 保存当前编辑器中的配置工程  |
| 另存为... | 另存当前编辑器对应的配置工程 |
| 全部保存   | 保存编辑器中打开的所有工程  |
| 退出     | 退出软件           |



图 3: 工程

工程

| 菜单     | 说明               |
|--------|------------------|
| 新建工程   | 新建一个配置工程         |
| 打开工程   | 打开配置工程 (加载到工程列表) |
| 关闭工程   | 关闭当前工程           |
| 关闭所有工程 | 关闭工程列表中的所有工程     |
| 构建     | 构建当前工程，生成配置文件    |

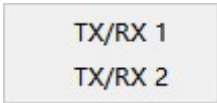


图 4: TX/RX

TX/RX

| 菜单      | 说明         |
|---------|------------|
| TX/RX 1 | TX/RX 窗口 1 |
| TX/RX 2 | TX/RX 窗口 2 |

打开关闭 TX/RX 窗口。

语言 切换 UI 语言。

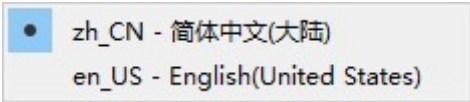


图 5: 语言



图 6: 配置

配置

| 菜单           | 说明                                 |
|--------------|------------------------------------|
| 提醒 CTK 更新    | 启用后，软件启动时会自动检查 CTK 更新，如果有更新则会提示    |
| 提醒 Device 更新 | 启用后，软件启动时会自动检查 Device 更新，如果有更新则会提示 |



图 7: 帮助

帮助

| 菜单           | 说明                     |
|--------------|------------------------|
| 检查 CTK 更新    | 检查 CTK 更新，如果有更新则会提示    |
| 检查 Device 更新 | 检查 Device 更新，如果有更新则会提示 |
| 关于           | 打开关于窗口                 |

工具栏

工程列表

配置编辑器

6.1.2 新建工程

- 菜单-> 新建工程
- 选择芯片
- 保存工程
- 打开编辑器

6.1.3 芯片配置解析

PAN3120

配置分为多个组：



图 8: 工具栏

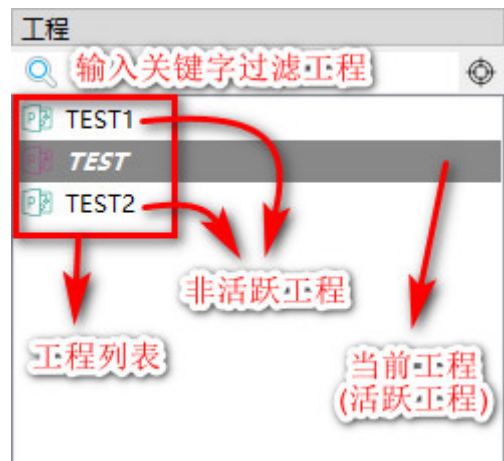


图 9: 工程列表

Device Information    Device Information 并非配置组，仅展示一些芯片信息。

| Title                 | Value       | 说明     |
|-----------------------|-------------|--------|
| Device                | PAN3120     | 名称     |
| Function              | Tx/Rx       | 功能     |
| Band                  | 130~1110MHz | 带宽     |
| Package               |             | 封装     |
| Modem                 |             | 调制     |
| Data Rate             | 2~500kbps   | 速率     |
| Firmware ID           | 0x0003      | 固件 ID  |
| Version(for CTK only) | 0x0000      | 配置模板版本 |

Project Options    Project Options 为工程本身的一些配置。

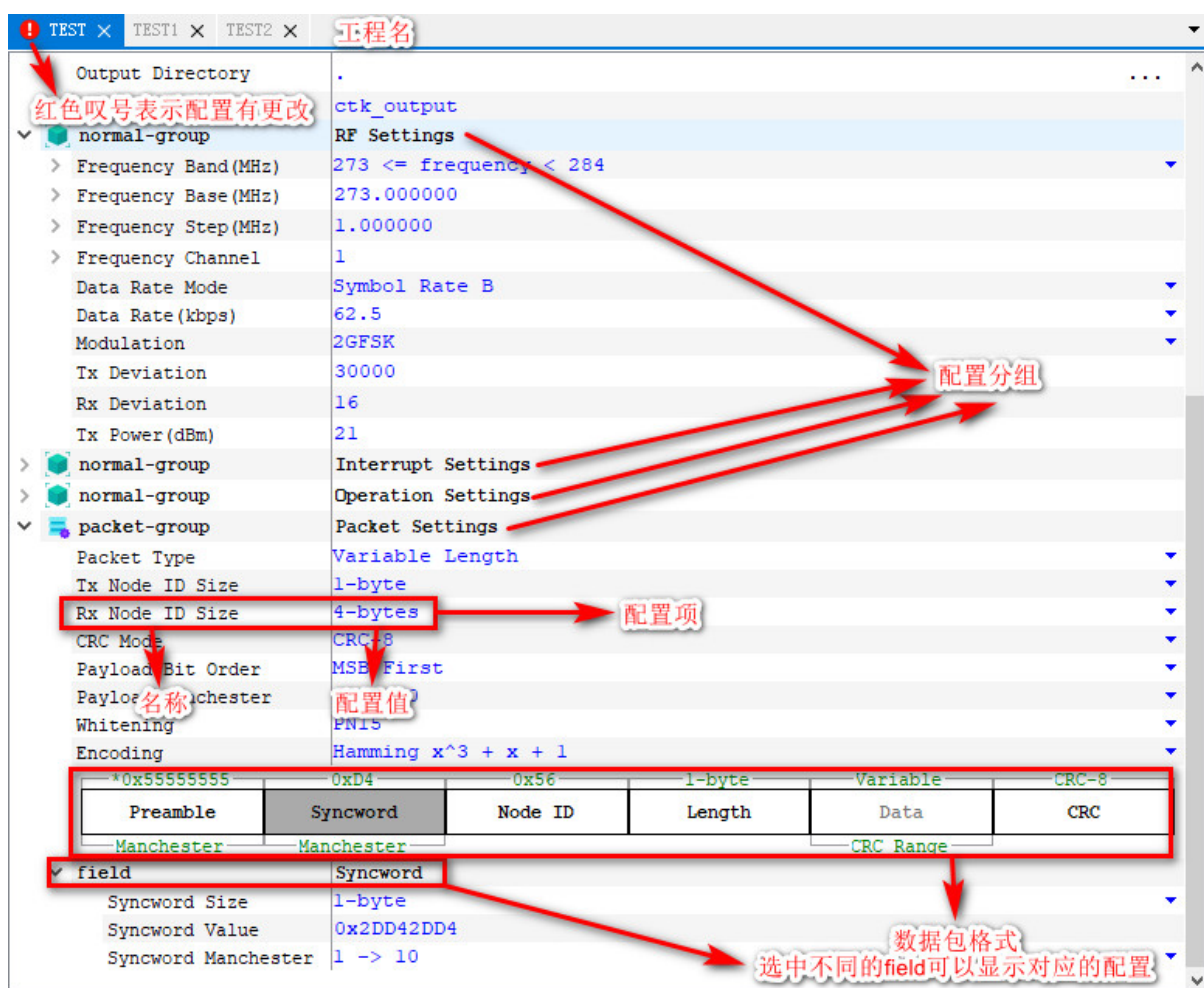


图 10: 配置编辑器

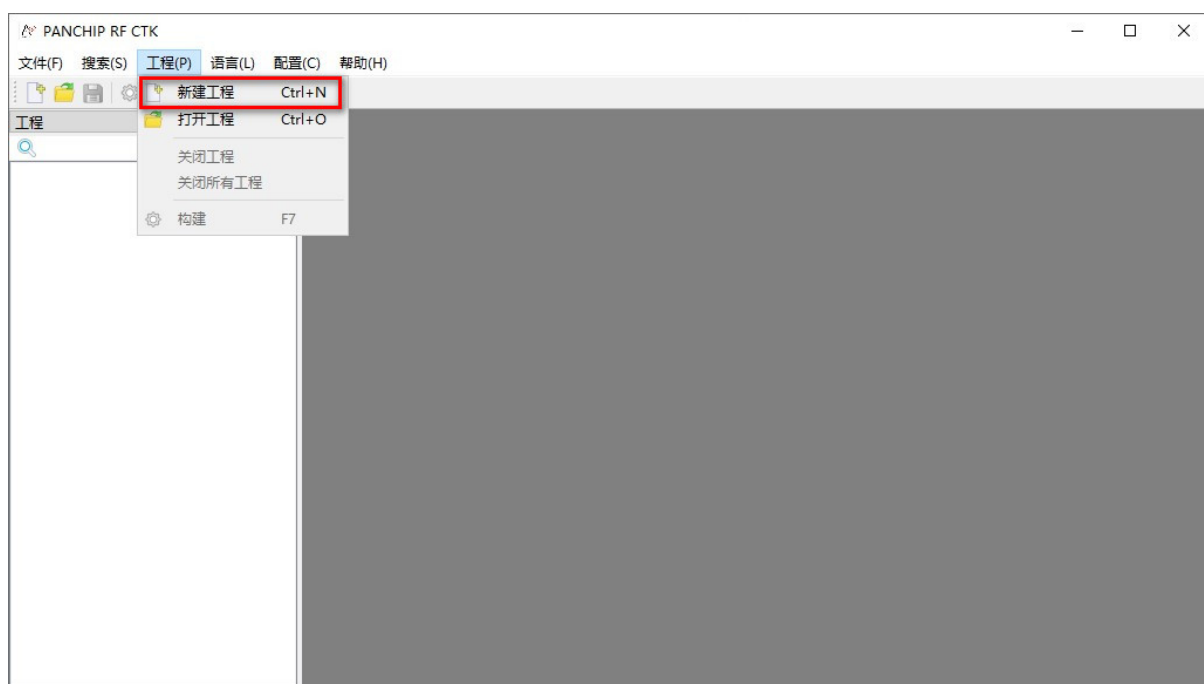


图 11: 新建工程

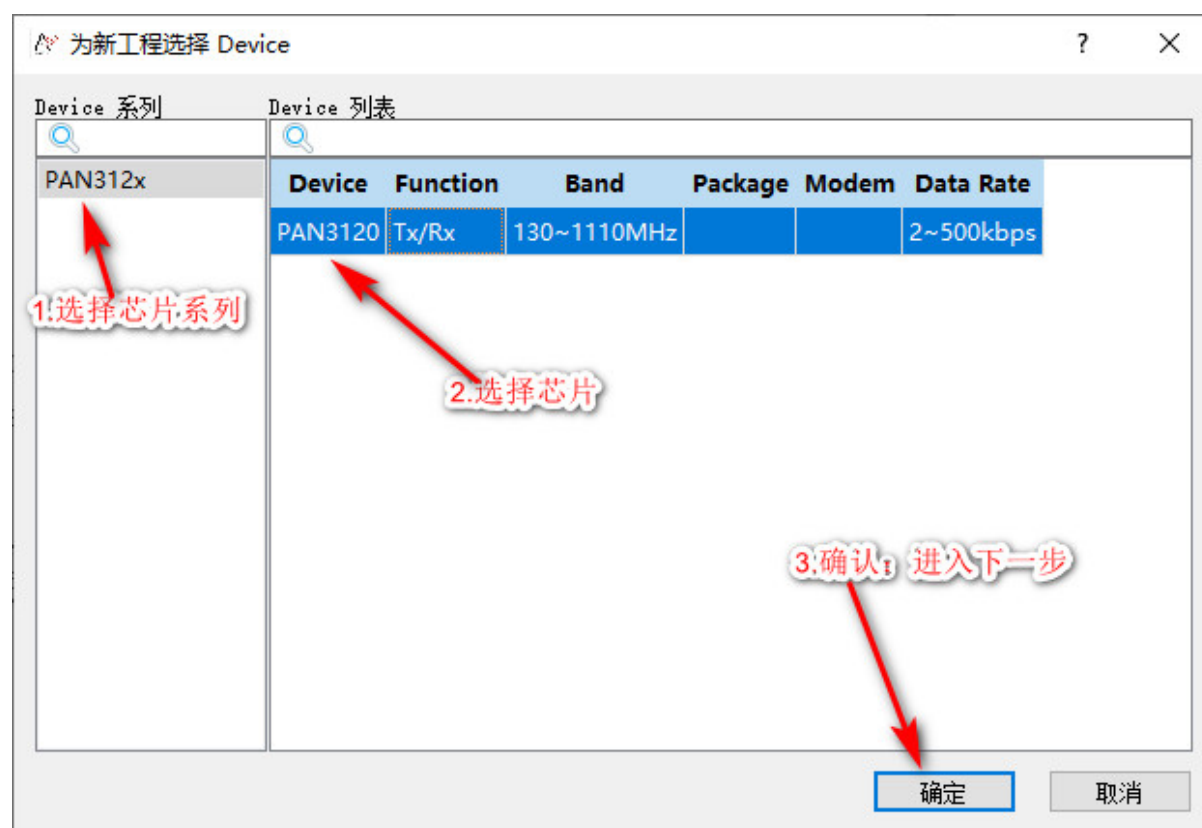


图 12: 选择芯片

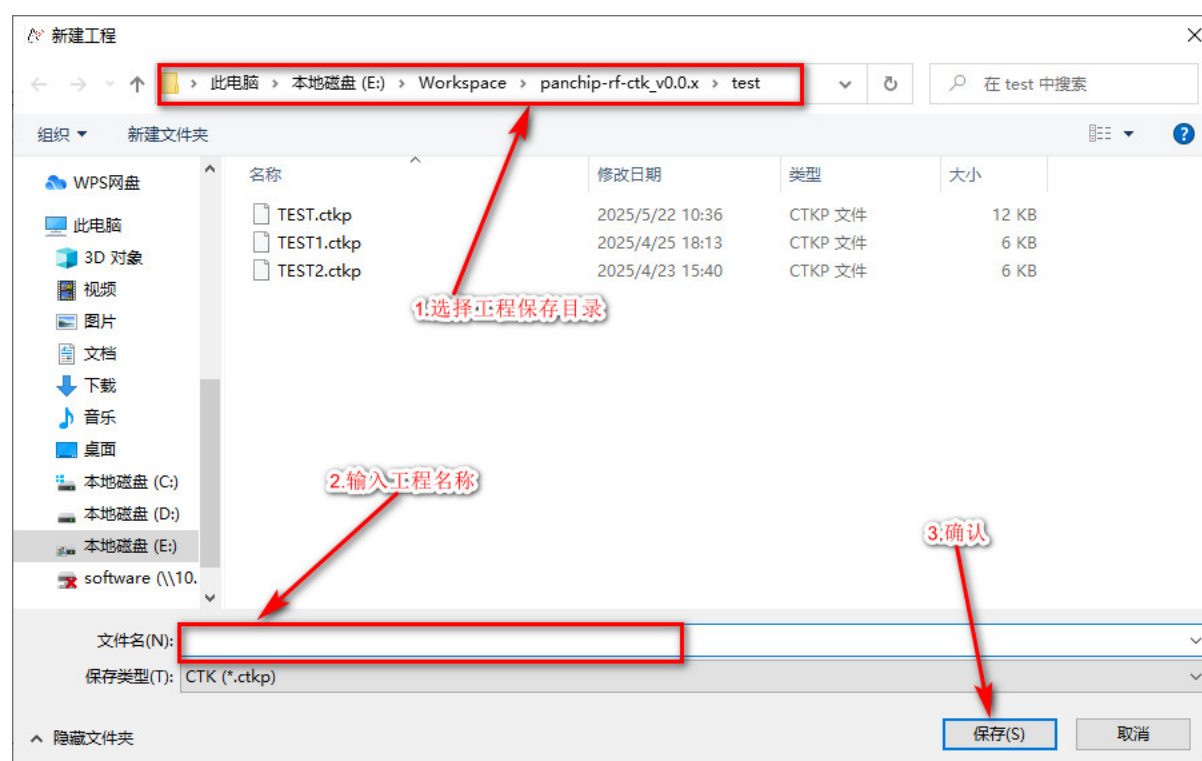


图 13: 保存工程

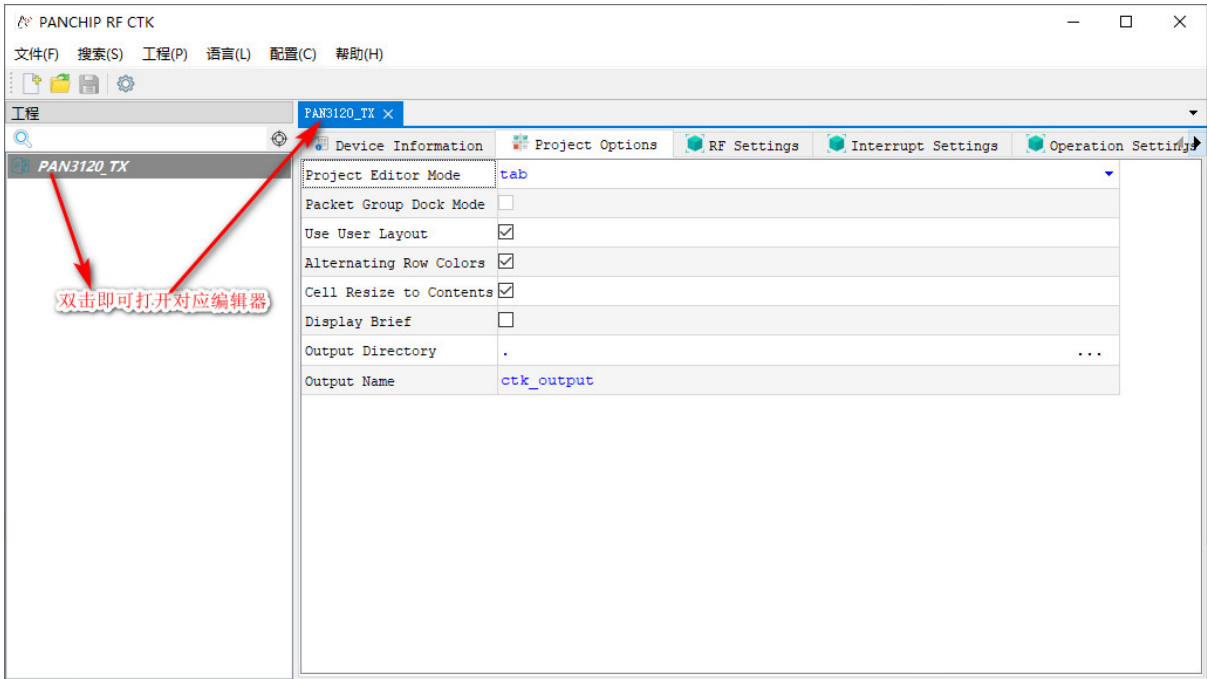


图 14: 打开编辑器

| Title                              | 说明                           |
|------------------------------------|------------------------------|
| Project Editor Mode                | 配置工程编辑器的模式，需重新打开编辑器生效        |
| Packet Group Dock Mode             | 包配置组是否启用 Dock 模式，建议关闭        |
| Use User Layout                    | 是否保存用户布局                     |
| Alternating Row Colors             | 配置项是否启用行颜色交错，建议启用            |
| Cell Resize to Contents            | 配置单元格大小是否按内容调整               |
| Display Brief                      | 是否启用描述列，需重新打开编辑器生效           |
| Page Language Follow CTK Settings  | 配置页面语言是否跟随 CTK 设置，需重新打开编辑器生效 |
| Output Mode                        | 输出模式，选择 C-standard 即可        |
| Output Directory                   | 构建输出目录                       |
| Output Name                        | 构建输出文件名                      |
| Generate Configuration as Comments | 构建输出时将配置信息以注释的形式输出           |
| Generate Configuration as Defines  | 构建输出时将配置信息以宏的形式输出            |

Project Editor Mode

| mode | 说明                    |
|------|-----------------------|
| tab  | 选项卡模式，页面分为多个选项卡       |
| tree | 树形模式                  |
| dock | 类似 tab 模式，但选项卡可随意拖动停靠 |

## RF Settings

| Title               | 说明              |
|---------------------|-----------------|
| Frequency Band(MHz) | 选择频带            |
| Frequency Base(MHz) | 设置基频, 必须在带宽之内   |
| Frequency Step(MHz) | 设置频率步进          |
| Frequency Channel   | 设置频率 channel    |
| Data Rate Mode      | 选择数据速率模式        |
| Data Rate(kbps)     | 设置或选择数据速率       |
| Modulation          | 选择调制方式          |
| Tx Deviation(Hz)    | 发送频偏            |
| Rx Deviation(Hz)    | 接收频偏            |
| Power Supply Mode   | 供电模式, 会影响发送功率配置 |
| Inductor Package    | 电感封装, 会影响发送功率配置 |
| Tx Power(dBm)       | 设置发送功率          |

**频率** 最终频率:  $\text{Frequency} = \text{Base} + \text{Step} * \text{Channel}$

如果用户想要跳频, 需要注意任何 **Channel** 带入上面公式得到的频率都必须在选择的带宽范围内。

**速率** 数据速率模式有两种:

- Symbol Rate A
- Symbol Rate B

不同的模式对于数据速率的配置有影响。

**Interrupt Settings** 中断发生后, 将会被记录到 STATUS 寄存器中, 可以通过对应的命令进行读取; 如果以下配置开启, 则对应的中断发生时反应到中断 IO 口通知 host。

| Title                   | 说明                                              |
|-------------------------|-------------------------------------------------|
| IE of Tx Done           | 发送结束中断 (所有发送流程结束), <b>支持自动清除</b>                |
| IE of Rx Done           | 接收结束中断 (所有接收流程结束)                               |
| IE of Rx Complete       | 接收完成中断 (数据已经全部接收完成, 但整个接收流程并未结束), <b>支持自动清除</b> |
| IE of CRC Error         | CRC 错误中断 (所有接收流程结束)                             |
| IE of Rx Timeout        | 接收超时中断, <b>支持自动清除</b>                           |
| IE of Node ID Error     | Node ID 错误中断, <b>支持自动清除</b>                     |
| IE of Length Done       | Length 接收完成中断, <b>支持自动清除</b>                    |
| IE of Node ID Done      | Node ID 接收完成中断, <b>支持自动清除</b>                   |
| IE of Header CRC Done   | 数据头 CRC 计算完成中断 (Node ID, Length), <b>支持自动清除</b> |
| IE of Wakeup            | 休眠唤醒中断, <b>支持自动清除</b>                           |
| IE of Ack Timeout       | ACK 接收超时中断, <b>支持自动清除</b>                       |
| IE of Ack Error         | ACK 错误中断, <b>支持自动清除</b>                         |
| IE of Syncword Timeout  | 同步字接收超时中断, <b>支持自动清除</b>                        |
| IE of Tx FIFO Threshold | 数据发送阈值中断                                        |
| IE of Rx FIFO Threshold | 数据接收阈值中断                                        |
| IE of RSSI Valid        | RSSI 有效中断, <b>支持自动清除</b>                        |
| IE of Preamble Done     | 前导码接收完成中断, <b>支持自动清除</b>                        |
| IE of Syncword Done     | 同步字接收完成中断, <b>支持自动清除</b>                        |
| IE of Preamble Timeout  | 前导码接收超时中断, <b>支持自动清除</b>                        |
| IE of RSSI Timeout      | RSSI 超时中断, <b>支持自动清除</b>                        |



## Operation Settings

| Title                     | 说明                           |
|---------------------------|------------------------------|
| WUT Clock Divider         | WUT 时钟分频值选择                  |
| RCL Trim Timer            | 是否启用 RCL 校准定时器               |
| RCL Trim Period           | RCL 校准定时器周期                  |
| Auto Clear Status         | 是否自动清除状态 (host 读时, 非所有状态都支持) |
| Idle If Possible          | 是否在可能时进入 IDLE 状态             |
| Duty Cycle Period         | Duty Cycle 周期                |
| Sleep Timer               | 是否启用休眠定时器                    |
| Sleep Time                | 休眠时间                         |
| Tx Extra Times            | 包发送额外次数                      |
| Tx Packet Gap(us)         | 当包发送额外次数不为 0 时的发包间隔          |
| Rx Timeout(us)            | 接收超时时间                       |
| Ack Delay Time(us)        | ACK 发送延迟时间                   |
| Ack Timeout(us)           | ACK 接收超时时间                   |
| Ack Check                 | 是否启用 ACK 校验                  |
| Ack Check Byte            | ACK 校验字节                     |
| Ack Tx Byte               | ACK 发送字节                     |
| Exit State of Tx Ok       | 发送完成退出状态                     |
| Exit State of Ack Timeout | ACK 接收超时退出状态                 |
| Exit State of Ack Error   | ACK 错误退出状态                   |
| Exit State of Rx Valid    | RX 完成且数据正确退出状态               |
| Exit State of Rx Invalid  | RX 完成但数据错误退出状态               |
| Exit State of Rx Timeout  | RX 接收超时退出状态                  |

## Exit State

| 状态    | 说明                  |
|-------|---------------------|
| Ready | 就绪状态                |
| Tx    | 发送状态                |
| Rx    | 接收状态                |
| Sleep | 休眠状态 (Normal Sleep) |

Direct Settings 当 **Direct** 使能时 **Packet** 的所有配置将被忽略。

| Title           | 说明             |
|-----------------|----------------|
| Direct Enable   | 是否使能 Direct    |
| Direct Bypass   | 是否启用 Bypass    |
| Direct Mode     | 模式选择: 异步/同步    |
| Direct CLK Gpio | 时钟引脚选择, 同步模式有效 |
| Direct DAT Gpio | 数据引脚选择         |

Packet Settings 当 **Direct** 关闭时 **Packet** 模式才有效。

## Common Settings

| Title              | 说明                  |
|--------------------|---------------------|
| Packet Type        | 数据包类型               |
| Node ID Size       | Node ID 字节数         |
| CRC Mode           | CRC 模式              |
| Payload Bit Order  | Payload 数据位序        |
| Payload Manchester | 是否启用 Payload 曼彻斯特编码 |
| Whitening          | 是否启用白化              |
| Encoding           | 编码选择                |



## Field: Preamble

| Title               | 说明            |
|---------------------|---------------|
| Preamble Mode       | 前导码模式         |
| Preamble Tx Size    | 前导码字节数        |
| Preamble Value      | 用户前导码值        |
| Preamble Manchester | 是否启用前导码曼彻斯特编码 |

## Field: Syncword

| Title               | 说明            |
|---------------------|---------------|
| Syncword Size       | 同步字字节数        |
| Syncword Value      | 同步字值          |
| Syncword Manchester | 是否启用同步字曼彻斯特编码 |

## Field: Node ID

| Title                     | 说明                    |
|---------------------------|-----------------------|
| Node ID Position          | Node ID 位置            |
| Tx Node ID Value          | Node ID 发送值           |
| Packet Filter Mode        | 包过滤模式                 |
| Packet Filter Interrupt   | 是否使能中断 (使能后才会改变状态寄存器) |
| Packet Filter Mask        | 包过滤掩码                 |
| Packet Filter Pat1 Enable | 包过滤是否启用 PAT1          |
| Packet Filter Pat1 Value  | 包过滤 PAT1 值            |
| Packet Filter Pat2 Enable | 包过滤是否启用 PAT2          |
| Packet Filter Pat2 Value  | 包过滤 PAT2 值            |
| Packet Filter Pat3 Enable | 包过滤是否启用 PAT3          |
| Packet Filter Pat3 Value  | 包过滤 PAT3 值            |

## Field: Length

| Title             | 说明      |
|-------------------|---------|
| Length Size       | 长度字段字节数 |
| Length Byte Order | 长度字节序   |

## Field: Data

| Title       | 说明   |
|-------------|------|
| Data Length | 数据长度 |

## Field: CRC

| Title          | 说明             |
|----------------|----------------|
| CRC Seed       |                |
| CRC Poly       |                |
| CRC Range      | CRC 计算范围       |
| CRC Bit Invert | CRC 计算结果是否按位反转 |
| CRC Byte Order | CRC 字节序        |
| CRC Bit Order  | CRC 比特序        |

## Chapter 7

# 其他文档

PAN312x SoC 相关的其他文档请参考：

- [PAN312x 产品说明书](#)
- [PAN312x 寄存器说明](#)



## Chapter 8

# 应用文档

### 8.1 PAN312x 寄存器说明

#### 8.1.1 1 COMMAND 寄存器描述

##### COMMAND\_OP\_REGISTER

备注：

- 1、INCA: 1: 地址自增, 0: 地址保持
- 2、CFLG: 1: 读取 info 区域数据, 0: 读取其它区域数据
- 3、WFLG: 1: 写数据, 0: 读数据
- 4、XFLG: 1: xdata, 0: data
- 5、XADH: xdata[10:8] 地址
- 6、ADDR/XADL: data 地址或者 xdata[7:0] 地址

##### COMMAND\_START\_TX

备注：

- 1、CHANNEL: 信道
- 2、RETX: 重传, ACK = 1 时有效
- 3、ECHN: 是否使用 CHANNEL
- 4、EACK: 是否使能 ACK
- 5、START: 启动条件, 0-立即启动, 1-延迟启动 (休眠唤醒后进入)
- 6、TXLEN: 发送数据长度

注意：

COMMAND\_START\_TX 命令可以只发 CMD, 此时表示使用之前设置的命令参数直接启动 TX

##### COMMAND\_START\_RX

备注：

- 1、CHANNEL: 信道

- 2、ERTO: 是否启用 RX 超时
- 3、ECHN: 是否使用 CHANNEL
- 4、EACK: 是否使能 ACK
- 5、START: 启动条件, 0-立即启动, 1-延迟启动 (休眠唤醒后进入)
- 6、RXLEN: 接收数据长度

注意:

COMMAND\_START\_RX 命令可以只发 CMD, 此时表示使用之前设置的命令参数直接启动 RX

#### COMMAND\_TX\_ACK

#### COMMAND\_GO\_SLEEP

备注:

- 1、SLEEP\_MODE: 0: SLEEP, 1: DEEP\_SLEEP;
- 2、WAKEUP\_STATE: 0: Ready, 1: TX, 2: RX ; (对 DEEP\_SLEEP 无效)

#### COMMAND\_GO\_READY

备注:

- 1、在 Tx、Rx 状态下使用, 立即生效;

#### COMMAND\_SET\_PROPERTY

注意:

**参数值:** 该命令最多可以写入 30 个 bytes 的参数值;

#### COMMAND\_GET\_PROPERTY

注意:

**参数值:** 该命令最多可以读取 30 个 bytes 的参数值;

#### COMMAND\_GET\_STATUS

备注:

- 1、TX\_DONE: 发送完成中断;
- 2、RX\_DONE: 接收完成中断 (不支持自动清除);
- 3、RX\_COMPLETE: 数据接收完成中断, 此时可读取 FIFO 中的所有数据, 但接收仍未完成;
- 4、RX\_CRC\_ERROR: 接收数据 CRC 错误中断 (不支持自动清除);
- 5、RX\_TIMEOUT: 接收数据超时中断;
- 6、RX\_NODEID\_ERROR: 接收数据 NODEID 匹配错误中断;
- 7、RX\_LENGTH\_DONE: 接收数据 LENGHT 字段完成中断;
- 8、RX\_NODEID\_DONE: 接收数据 NODEID 字段完成中断;
- 9、WAKE\_UP: 芯片被唤醒中断;

- 10、ACK\_TIMEOUT: 接收 ACK 超时中断;
- 11、ACK\_ERROR: 接收 ACK 错误中断;
- 12、RX\_RSSI\_VALID: 接收 RSSI 有效中断;
- 13、TX\_FIFO\_TH: 发送 FIFO 数据个数低于阈值中断 (不支持自动清除);
- 15、RX\_FIFO\_TH: 接收 FIFO 数据个数高于阈值中断 (不支持自动清除);
- 16、RX\_RSSI\_TIMEOUT: 接收 RSSI 超时中断;
- 17、RX\_PRE\_DONE: 接收前导码完成中断;
- 18、RX\_SYNC\_DONE: 接收同步字完成中断;
- 19、RX\_PRE\_TIMEOUT: 接收前导码同步超时中断;
- 20、RX\_SYNC\_TIMEOUT: 接收同步字超时中断;

## COMMAND\_CLEAR\_STATUS

备注:

- 1、TX\_DONE: 发送完成中断;
- 2、RX\_DONE: 接收完成中断 (不支持自动清除);
- 3、RX\_COMPLETE: 数据接收完成中断, 此时可读取 FIFO 中的所有数据, 但接收仍未完成;
- 4、RX\_CRC\_ERROR: 接收数据 CRC 错误中断 (不支持自动清除);
- 5、RX\_TIMEOUT: 接收数据超时中断;
- 6、RX\_NODEID\_ERROR: 接收数据 NODEID 匹配错误中断;
- 7、RX\_LENGTH\_DONE: 接收数据 LENGHT 字段完成中断;
- 8、RX\_NODEID\_DONE: 接收数据 NODEID 字段完成中断;
- 9、WAKE\_UP: 芯片被唤醒中断;
- 10、ACK\_TIMEOUT: 接收 ACK 超时中断;
- 11、ACK\_ERROR: 接收 ACK 错误中断;
- 12、RX\_RSSI\_VALID: 接收 RSSI 有效中断;
- 13、TX\_FIFO\_TH: 发送 FIFO 数据个数低于阈值中断 (不支持自动清除);
- 15、RX\_FIFO\_TH: 接收 FIFO 数据个数高于阈值中断 (不支持自动清除);
- 16、RX\_RSSI\_TIMEOUT: 接收 RSSI 超时中断;
- 17、RX\_PRE\_DONE: 接收前导码完成中断;
- 18、RX\_SYNC\_DONE: 接收同步字完成中断;
- 19、RX\_PRE\_TIMEOUT: 接收前导码同步超时中断;
- 20、RX\_SYNC\_TIMEOUT: 接收同步字超时中断;

## COMMAND\_GET\_FWID

### 8.1.2 2 特殊寄存器描述

R006

R007

R00E

R015

R016

R017

R018

R019

R01A

R01B

### 8.1.3 3 PROPERTY GROUP1 寄存器描述

PROPERTY\_ADDR\_IRQ\_ENABLE

### 8.1.4 4 PROPERTY GROUP2 寄存器描述

PROPERTY\_ADDR\_SYS\_CTRL0

PROPERTY\_ADDR\_PACKET\_FILTER\_CTRL

PROPERTY\_ADDR\_NODE\_ID\_CONF

PROPERTY\_ADDR\_TXPKT\_CONF

PROPERTY\_ADDR\_RXPKT\_CONF

### 8.1.5 5 PROPERTY GROUP3 寄存器描述

PROPERTY\_ADDR\_ACK\_TIMEOUT

PROPERTY\_ADDR\_RX\_TIMEOUT

PROPERTY\_ADDR\_PACKET\_FILTER\_MASK

PROPERTY\_ADDR\_PACKET\_FILTER\_PAT1

PROPERTY\_ADDR\_PACKET\_FILTER\_PAT2

PROPERTY\_ADDR\_PACKET\_FILTER\_PAT3

PROPERTY\_ADDR\_RX\_NODE\_ID

PROPERTY\_ADDR\_RX\_TIMEOUT\_EXIT\_STATE

PROPERTY\_ADDR\_RX\_INVALID\_EXIT\_STATE

PROPERTY\_ADDR\_RX\_VALID\_EXIT\_STATE

PROPERTY\_ADDR\_ACK\_TIMEOUT\_EXIT\_STATE

PROPERTY\_ADDR\_TX\_OK\_EXIT\_STATE

#### 8.1.6 6 PROPERTY GROUP4 寄存器描述

PROPERTY\_ADDR\_R106

PROPERTY\_ADDR\_R107

PROPERTY\_ADDR\_R108

PROPERTY\_ADDR\_R109

PROPERTY\_ADDR\_R10A

PROPERTY\_ADDR\_R10B

PROPERTY\_ADDR\_R10C

PROPERTY\_ADDR\_R10D

PROPERTY\_ADDR\_R10E

PROPERTY\_ADDR\_R10F

PROPERTY\_ADDR\_R110

PROPERTY\_ADDR\_R111

PROPERTY\_ADDR\_R112

PROPERTY\_ADDR\_R113

PROPERTY\_ADDR\_R114

PROPERTY\_ADDR\_R115

PROPERTY\_ADDR\_R116

PROPERTY\_ADDR\_R117

PROPERTY\_ADDR\_R118

PROPERTY\_ADDR\_R119



PROPERTY\_ADDR\_R11A

PROPERTY\_ADDR\_R11B

PROPERTY\_ADDR\_R11C

PROPERTY\_ADDR\_R11D

PROPERTY\_ADDR\_R11E

PROPERTY\_ADDR\_R11F

PROPERTY\_ADDR\_R120

PROPERTY\_ADDR\_R121

PROPERTY\_ADDR\_R122

PROPERTY\_ADDR\_R123

PROPERTY\_ADDR\_R124

PROPERTY\_ADDR\_R125

PROPERTY\_ADDR\_R12B

PROPERTY\_ADDR\_R12C

PROPERTY\_ADDR\_R12E

### 8.1.7 7 PROPERTY GROUP5 寄存器描述

PROPERTY\_ADDR\_R186

PROPERTY\_ADDR\_R187

PROPERTY\_ADDR\_R188

PROPERTY\_ADDR\_R189

PROPERTY\_ADDR\_R18A

PROPERTY\_ADDR\_R18B

PROPERTY\_ADDR\_R18C

PROPERTY\_ADDR\_R191

PROPERTY\_ADDR\_R192

PROPERTY\_ADDR\_R193

PROPERTY\_ADDR\_R194

PROPERTY\_ADDR\_R195

PROPERTY\_ADDR\_R196

PROPERTY\_ADDR\_R197

PROPERTY\_ADDR\_R198

PROPERTY\_ADDR\_R199

PROPERTY\_ADDR\_R19A

PROPERTY\_ADDR\_R19B

PROPERTY\_ADDR\_R1A4

PROPERTY\_ADDR\_R1AF

PROPERTY\_ADDR\_R1B0

PROPERTY\_ADDR\_R1B1

PROPERTY\_ADDR\_R1B2

PROPERTY\_ADDR\_R1B3

PROPERTY\_ADDR\_R1C0

PROPERTY\_ADDR\_R1C1



## Chapter 9

# 更新日志

### 9.1 PAN312x DK v0.2.0

PAN312x Development Kit v0.2.0 (2025-8-7) 已发布:

#### 9.1.1 1. SDK

- 优化两点式校准流程
- 两点式校准流程中, 添加  $h = 0.5$  的配置 ( $h = 2 * \text{Deviation} / \text{DataRate}$ )
- 添加功率配置参数, 包括 LDO 模式 (电感 0603 和 0402), DCDC 模式 (电感 0603 和 0402)
- 修复 RF 进入单载波模式的异常的问题
- 修复配置 RxDeviation 时,  $h > 1$  的寄存器配置 (可提升接收灵敏度)
- 修复 SDK\_TestDemo 中, TRxMarster 中, 发送计数错误的问题

#### 基础例程

- 新增如下例程:
- SDK\_CTK\_PacketTx: 演示如何通过 PANCHIP\_RF\_CTK 生成 RF 配置, 并实现数据发送的功能
- SDK\_CTK\_PacketRx: 演示如何通过 PANCHIP\_RF\_CTK 生成 RF 配置, 并实现数据接收的功能

#### 9.1.2 3. DOC

- 修改 硬件资料文档
  - 02\_hardware/hardware\_reference\_design: pan312x\_hardware\_reference\_design:
    - \* 添加 20dBm 直连 (Direct Tie)DCDC 模式典型应用原理图
    - \* 添加 20dBm 直连 (Direct Tie)LDO 模式典型应用原理图
    - \* 添加 0603 封装典型应用物料清单
    - \* 添加 DCDC 部分电路的描述
- 新增 演示例程文档:
  - 03\_samples/basic\_demo: SDK\_CTK\_PacketTx 文档
  - 03\_samples/basic\_demo: SDK\_CTK\_PacketRx 文档
- 新增 工具类文档:

- 06\_dev\_tools/panchip\_rf\_ctk: PANCHIP RF CTK 使用介绍文档
- 修改 其它类文档
  - 07\_others: PAN312x 寄存器说明 V1.1.pdf: 在 V1.0 的基础上, 增加其它寄存器的描述
  - 07\_others: PAN312x 系列产品说明书 V1.4.pdf: 具体可参考文档中的修订历史
- 新增 应用类文档
  - 07\_application/pan312x\_register\_introduction: PAN312x 寄存器使用说明

### 9.1.3 4. TOOLS

- 新增工具 PANCHIP-RF-CTK :
  - 可以配置 PAN3120 的 RF 参数, 包括频点, 速率、TRxDeviation、发射功率等参数;
  - 可以使能 PAN3120 的中断;
  - 可以配置一些操作设置, 比如接收数据成功后, RF 的退出撞他设置;
  - 可以配置数据包的帧格式, 包括前导码、同步字、Length、Data、CRC 等字段;

### 9.1.4 5. 已知问题

- 暂无

## 9.2 PAN312x DK v0.1.0

PAN312x Development Kit v0.1.0 (2025-6-3) 已发布:

### 9.2.1 1. SDK

- 修改部分 Regmap 寄存器默认值
- 修改获取 rssi 值接口函数, 根据不同速率, 对 rssi 值, 添加相应的补偿值
- 修复频率设置函数中的影响 868Mhz 频段灵敏度的问题;
- 修复两点式校准在 868Mhz 频段 evm 异常的问题;
- 修改设置 TxDeviation 接口函数, 支持小于 2.4Khz 的 TxDeviation 配置;

### 基础例程

- 新增如下例程:
  - SDK\_FixedLongPacketTx: 演示固定包长帧结构怎么实现长包发送的功能
  - SDK\_FixedLongPacketRx: 演示固定包长帧结构怎么实现长包接收的功能
  - SDK\_GpioOutput0clk: 演示 PAN312x 的 GPIO 如何输出时钟信号
  - SDK\_VariableLongPacketTx: 演示可变包长帧结构怎么实现长包发送的功能
  - SDK\_VariableLongPacketRx: 演示可变包长帧结构怎么实现长包接收的功能
  - SDK\_TestDemo: 演示通过评估板 (PAN312x\_EVB) 来评估 PAN312x 芯片的主要性能, 如通信测试, 单载波测试, 距离测试, 丢包率测试, 功耗测试等

### 9.2.2 3. DOC

- 修改 快速入门文档
- 01\_quick\_start: quick\_start\_pan312x\_sdk 文档中的错误描述
- 修改 硬件资料文档
  - 02\_hardware/hardware\_reference\_design: pan312x\_hardware\_reference\_design 文档中 VDD 和 VBAT 最高工作电压由 3.6V 改为 3.8V
- 新增 演示例程文档:
  - 03\_samples/basic\_demo: SDK\_FixedLongPacketTx 文档
  - 03\_samples/basic\_demo: SDK\_FixedLongPacketRx 文档
  - 03\_samples/basic\_demo: SDK\_GpioOutPutOclk 文档
  - 03\_samples/basic\_demo: SDK\_VariableLongPacketTx 文档
  - 03\_samples/basic\_demo: SDK\_VariableLongPacketRx 文档
  - 03\_samples/basic\_demo: SDK\_TestDemo 文档

### 9.2.3 4. TOOLS

- 暂无

### 9.2.4 5. 已知问题

- 暂无

## 9.3 PAN312x DK v0.0.0

PAN312x Development Kit v0.0.0 (2025-4-22) 已发布:

### 9.3.1 1. SDK

- 提供丰富的基础代码例程 & 演示文档
- 支持 PAN312x Evaluation Board V1.0 开发板
- 支持 PAN312x RF Transceiver Module 频段: 915Mhz、868Mhz、490Mhz、433Mhz
- 支持**固定包长帧结构、可变包长帧结构**
- 在可支持范围内, Frequency 任意可配
- 支持**速率**: 2kbps~40kbps, 步进 0.1kbps
- 支持**速率**: 50kbps、62.5kbps、80kbps、100kbps、125kbps、200kbps、250kbps、400kbps、500kbps
- 支持配置 TxDeviation 和 RxDeviation
- 发射功率可配置范围为 -20dBm ~ 21dBm, 步进 1dBm
- 支持 CRC 参数配置, 包括 CrcMode、CrcPolynomial、CrcSeed、CrcBitOrder、CrcByteSwap、CrcRange、CrcBitInv
- 支持 Preamble、Syncword、Payload&Crc 字段 Manchester 模式配置和使能
- 支持 Fec\_Haming 编码, 包括 Fec\_Haming\_X3\_X\_1、Fec\_Haming\_X3\_X2\_1

- 支持 Whitening 编码选择, 包括 Whitening\_PN6、Whitening\_PN7、Whitening\_PN9、Whitening\_PN9\_IBM、Whitening\_IEEE\_802154g、Whitening\_PN9\_CCITT、Whitening\_PN11、Whitening\_PN13、Whitening\_PN15
- SemiAutomaticDutyCycleTx 功耗:

| DataRate(kbps) | Packet Length(byte) | 发送时间(ms) | 休眠时间(ms) | 工作周期(ms) | Tx-Power(dBm) | 模式  | 峰值电流(mA) | 休眠电流(uA) | 平均电流(uA) |
|----------------|---------------------|----------|----------|----------|---------------|-----|----------|----------|----------|
| 50             | 20                  | 3.2      | 1000     | 1003.2   | 21            | LDO | 163.88   | 0.89     | 648.79   |
| 50             | 20                  | 3.2      | 1000     | 1003.2   | 15            | LDO | 72.79    | 0.88     | 291.23   |

- SemiAutomaticDutyCycleRx 功耗:

| Rx Window(ms) | 休眠时间(ms) | 工作周期(ms) | 模式  | 峰值电流(mA) | 休眠电流(uA) | 平均电流(mA) |
|---------------|----------|----------|-----|----------|----------|----------|
| 200           | 1000     | 1200     | LDO | 14.19    | 0.92     | 1.48     |

## 基础例程

- SDK\_FixedPacketTx: 演示固定包长帧结构发送功能
- SDK\_FixedPacketRx: 演示固定包长帧结构接收功能
- SDK\_VariablePacketTx: 演示可变包长帧结构发送功能
- SDK\_VariablePacketRx: 演示可变包长帧结构接收功能
- SDK\_SemiAutomaticDutyCycleTx: 演示通过外部 MCU 结合 PAN312x 的低功耗特性, 如何实现 DutyCycleTx 功能
- SDK\_SemiAutomaticDutyCycleRx: 演示通过外部 MCU 结合 PAN312x 的低功耗特性, 如何实现 DutyCycleRx 功能

## 9.3.2 2. HDK

目前版本提供了如下硬件相关资料:

- PAN312x Evaluation Board V1.0.pdf.pdf
- PAN312x Hardware Reference Design V1.0.pdf

## 9.3.3 3. DOC

目前版本提供了如下开发文档:

- 新增 快速入门文档
  - 01\_quick\_start: quick\_start\_pan312x\_sdk 文档
- 新增 硬件资料文档
  - 02\_hardware/hardware\_reference\_design: pan312x\_hardware\_reference\_design 文档
  - 02\_hardware/evaluation\_board\_introduction: pan312x\_evaluation\_board\_introduction 文档
- 新增 演示例程文档:
  - 03\_samples/basic\_demo: SDK\_FixedPacketTx 文档
  - 03\_samples/basic\_demo: SDK\_FixedPacketRx 文档

- 03\_samples/basic\_demo: SDK\_VariablePacketTx 文档
- 03\_samples/basic\_demo: SDK\_VariablePacketRx 文档
- 03\_samples/basic\_demo: SDK\_SemiAutomaticDutyCycleTx 文档
- 03\_samples/basic\_demo: SDK\_SemiAutomaticDutyCycleRx 文档
- 新增 其它文档:
  - 07\_other: PAN312x 系列产品说明书 V1.2.pdf 文档
  - 07\_other: PAN312x 寄存器说明 V1.0.pdf 文档

#### 9.3.4 4. TOOLS

- 暂无

#### 9.3.5 5. 已知问题

- 暂无