

TinySDR 安装指南和学习流程

V1.3.2

泰瑞合劲（天津）科技有限公司

2026-09-03

注意事项

本资料是当您购买了 TinySDR 之后需要阅读的第一本资料，因为在本资料中我将向您介绍如何 Setup TinySDR 和如何一步一步的向下学习 PlutoSDR，这也是我本人的学习心得，属于一叶障目，难免会有偏颇，还请各位读者自行斟酌。。

购买 TinySDR 后首先要明确的一点，就是我到底是只要简单的使用 TinySDR 还是我要通过学习 TinySDR，成为一个深入射频专业底层的工程师。

如果只是简单的使用，那其实非常简单，只需要掌握信号源发生器软件，频谱仪软件和 ADI IIO Oscilloscope 就足够了。

但是如果你想进行嵌入式的底层开发，或者开发自己的协议，又或者想破解别人的编码，制造自己未来特有的通讯设备，你就必须一步一步的深入学习整个 TinySDR 的软件和硬件架构，直到最终以 TinySDR 为基本蓝本，作出适合自己的软件和硬件。其中的学习之路非常漫长而且艰难，但是相信我，只要你一步一个脚印，上来就给自己制定一个目标的话，你即使最后做不出来，也会对通信射频有一个完全不同的理解，因为确实太难了。

这个手册主要是配合 TinyRFLab 基于 Pluto SDR 原型出品的 TinySDR 这款产品的相关应用知识，TinySDR 不含有电池，所以可以适合于所有运输方式。

因为本产品被少数不法商家仿制，这些不法商家炒作信誉同时根本不进行射频性能优化，所以非本公司通过淘宝或者公对公销售的产品，本公司一律不负责维护和培训服务。

法律内容：

TinySDR 是本公司拥有的注册商标。未经书面许可，不允许其他人使用 TinySDR 这个名字或用于任何 SDR 产品。

版本信息：

版本信息	撰写人	更新内容	更新日期
V1.0	Terry Kong,	初始版本	2026-07-26
V1.1	Terry Kong, Pio Wonsler	目录优化，更新 Q&A，更新 SDR++文档	2026-08-25
V1.2	Terry Kong, Pio Wonsler	删除多余的 SDR++部分，该内容详见《SDR++ 用户指南》	2026-08-26
V1.2.1	Terry Kong, Pio Wonsler	初步增加 TinySDR 的软件层配置内容及例程 1-3	2026-08-28
V1.2.2	Terry Kong, Pio Wonsler	初步增加例程 4-5，修复例程 3；初步完成 TinySDR 特性与功能	2026-08-31
V1.3	Terry Kong, Pio Wonsler	重构文档，完善例程 1-5，完善 Q&A，增加 Windows 的初始设置，增加硬件模块图和实拍照	2026-09-01

V1.3.1	Terry Kong, Pio Wonsler	完善例程 5 截图；完善 Q&A	2026-09-02
V1.3.2	Terry Kong	修改了格式和内容	2026-09-03

手册编号：TR-260726

手册出品方：

泰瑞合劲（天津）科技有限公司，

营业地址：天津市南开区科研东路西侧天津科技广场 6-2-1108（天开园）

电话号码：13920616671

Email：14025259@qq.com

公司主页：<https://www.tinyrfequip.com/>

公司主营：

提供定制化的微型，高效的射频微波测试设备，射频模块以及测试用 **Firmware** 和 **Software** 软件。这些射频产品主要用来满足工厂、科研单位、无线电爱好者基本的测试需求，以及学生学习的需求。本公司秉持自由开放的合作原则，互利互惠，互相学习，互相进步，希望和客户端进行更专业，更快捷地交流，解决射频领域的实际问题，本公司力争为客户获得最大的经济投入产出比。

在线培训咨询，更多信息请直接联系淘宝或者 email。

目前硬件产品主要有：

TinySA, TinySA Ultra, LiteVNA, LibreVNA, NanoVNA-F V2, NanoVNA-F V3, RFDetectorV2, TinyRFSG, TinyTracker, NearFieldRFProbe, TinySDR

目前的软件产品主要有：

TinySA and Ultra 上位机 App 定制化开发。TinyRFSG 硬件和上位机定制开发, RF Detector。TinySDR 编程。

视频教学：

可以在 B 站也就是哔哩哔哩网站搜索 TinyRFEquip 账号找到介绍 TinyRFEquip 仪器的主要功能介绍视频。例如：

TinySA Ultra 对于无线话筒的频率特性检定测量_bilibili

https://www.bilibili.com/video/BV1yt421u7Y2/?vd_source=8d60a323bdde2d574cb0a85c07cb3212

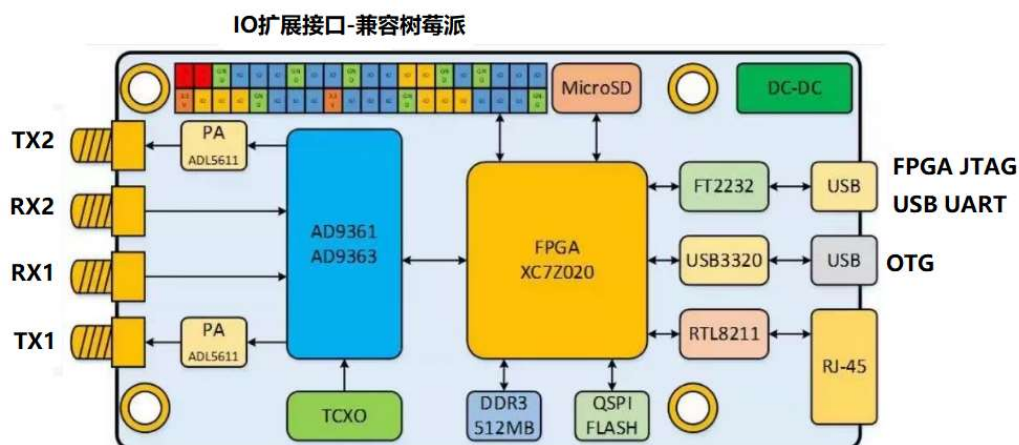
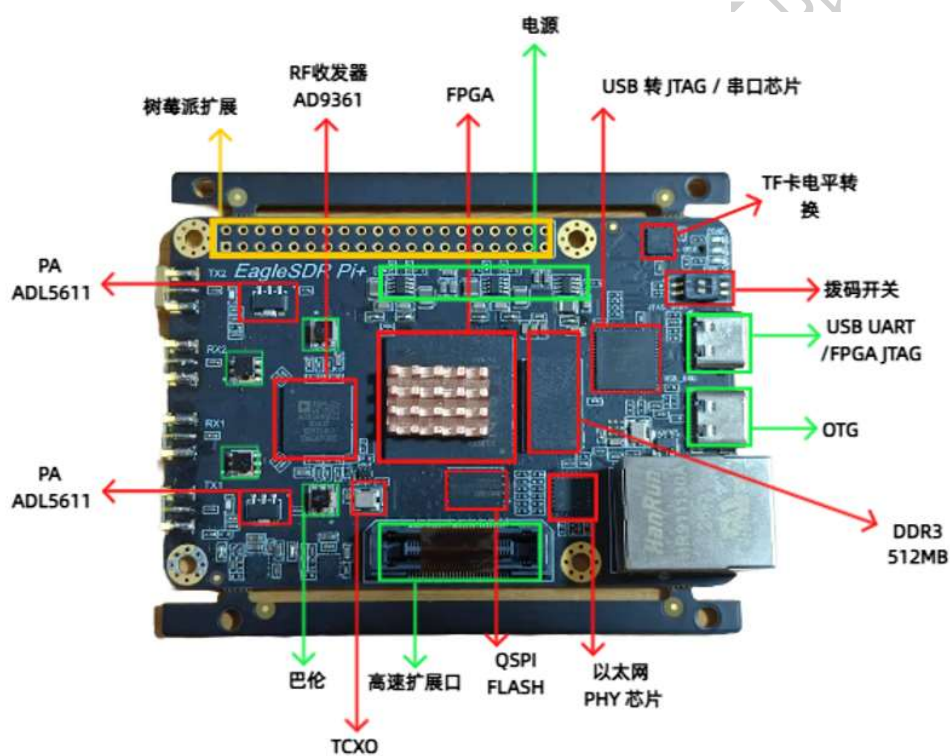
2

中文手册：

网盘里面的 TinySDR 文件夹有相关的 app 和我写的中文用户手册，可以获得机器最大的性能链接：<https://pan.baidu.com/s/1tPY1LlOzZ3m6y7eBUuKlbQ?pwd=478q>

提取码:478q

1. TinySDR 软硬件概述



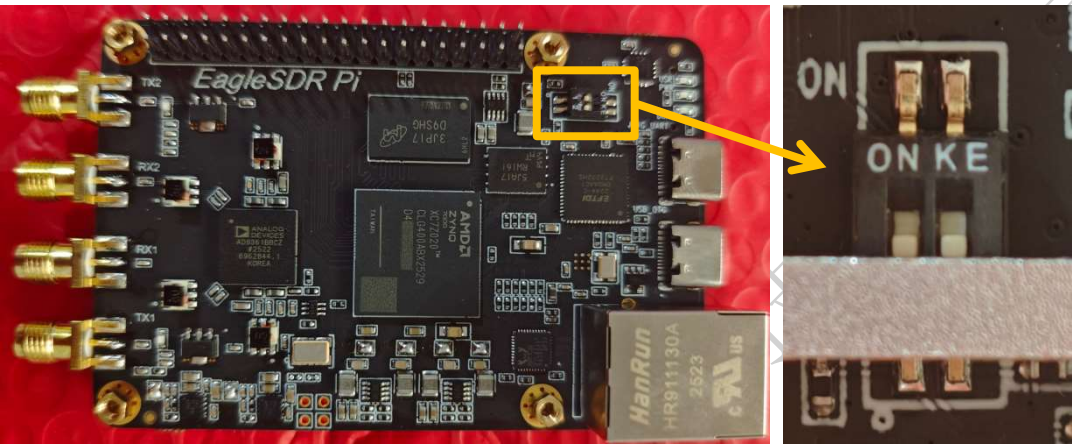
注：TinySDR 仍在不断优化硬件设计，但整体变化不大，上图为加载风扇后的外形图

- Plutosdr-fw 版本: 0.34
- 内核: Linux 5.4.0
- 架构: armv7l (armhf)
- glibc: 2.28 (Linux 的 C 库版本)
- 预期开机时间: 30s-3min

2. TinySDR 系统的 Setup

本 TinySDR 基于 PlutoSDR 开发，与 PlutoSDR 兼容。

开发板发货前已经烧写好了 PlutoSDR 镜像，初始拨码开关设置为 00（上为导通/ON/1），为 QSPI 启动模式，如图。（拨码开关位置均设置在指示灯附近）



拨码开关与启动模式的对应关系如表：

	拨码开关位 1	拨码开关位 2
QSPI（初始）	0	0
JTAG	0	1
SD	1	0
NAND	1	1

下面我们分为 Windows 和 Linux 两个系统的 Setup 分别进行介绍：

A. Windows 系统的 Setup:

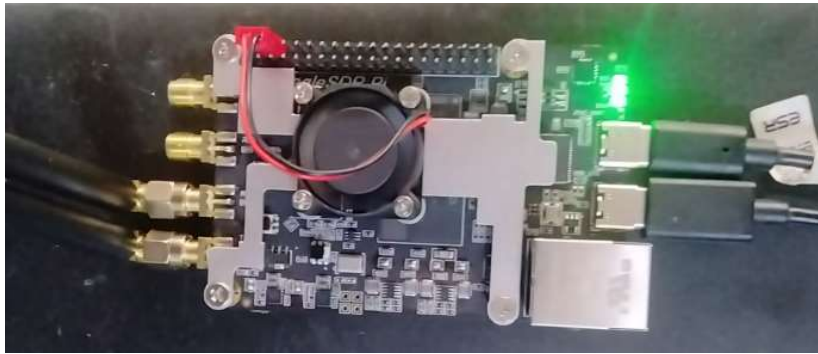
从网盘中下载 PlutoSDR 也就是 TinySDR 必备的软件驱动：

- LibIIO-0.26（旧的版本可以到 0.21）
- PlutoSDR-M2k-USB-Drivers.exe
- adi-osc-setup

运行并安装驱动程序，正确安装后，在控制面板的程序界面我们会看到如下列表：

Windows Driver Package - Analog Devices, Inc. Net (04/04/2022 6.0.6000.16384)	Analog Devices, Inc.	2026/4/30	04/04/2022 6.0.6000....
Windows Driver Package - Analog Devices, Inc. (WinUSB) Universal Serial Bus de...	Analog Devices, Inc.	2026/4/30	04/04/2022 6.1.7600....
Windows Driver Package - Analog Devices, Inc. (usbser) Ports (04/04/2022 5.1.2...	Analog Devices, Inc.	2026/4/30	04/04/2022 5.1.2600.0
Windows Driver Package - Analog Devices, Inc. (WinUSB) Universal Serial Bus de...	Analog Devices, Inc.	2026/4/30	04/04/2022 6.1.7600....
IIO Oscilloscope version v0.18-main	Analog Devices, Inc.	2026/7/20	231 MB v0.18-main
Libiio version 0.26	Analog Devices, Inc.	2026/7/26	6.09 MB 0.26
PlutoSDR-M2k-USB-Win-Drivers version 0.9	Analog Devices, Inc.	2026/4/30	8.95 MB 0.9

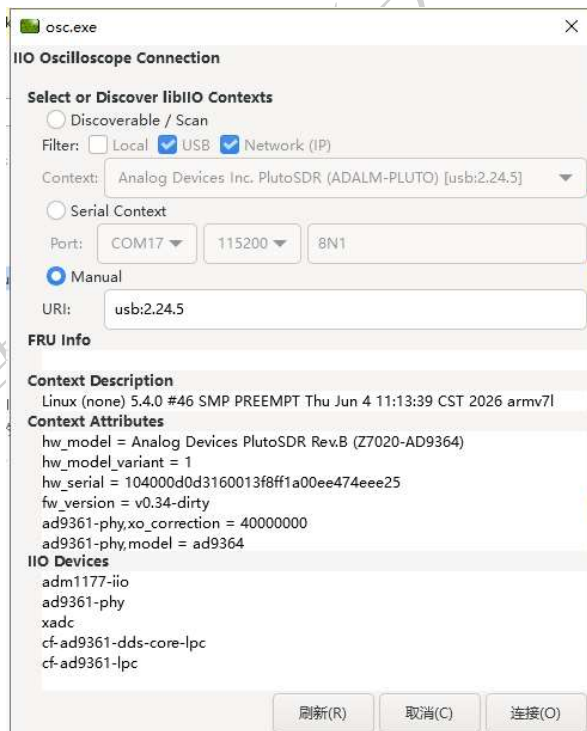
如下图设置连接好开发板 USB 线和电脑：



正常接上 TinySDR 后，我们会在仪器列表中看到如下信息，此时 TinySDR 已经被正常识别出串口了，会同时多出来两个虚拟串口，如下图所示出现了 Com6 和 Com17（每台电脑主机不同）：



理论上如果只连接 OTG 口，也就是更加靠近网口的 TypeC 接口，我们会只有 Com6，此时也能正常工作。我们也能在系统中看到一个硬盘。在 ADI OSC 中可以刷新后看到 TinySDR 的产品信息。如果连接上 USB UART 会出现 Com17。当电脑 USB 口供电不足的时候，必须同时连接两个 USB 提高供电能力。



PlutoSDR 已经准备就绪，可以开始测试了。这时候我们就可以从网盘下载下列软件并使用了：

IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!

Someone could be eavesdropping on you right now (man-in-the-middle attack)!

It is also possible that a host key has just been changed.

The fingerprint for the ECDSA key sent by the remote host is

SHA256:7ak0IjinIJXb6G1LCj8lp6LW/EKFGQjt9Hm4BmfA6Pc.

Please contact your system administrator.

Add correct host key in /home/user/.ssh/known_hosts to get rid of this message.

Offending ECDSA key in /home/user/.ssh/known_hosts:3

Host key for pluto.local has changed and you have requested strict checking.

Host key verification failed.

请在 ~/.ssh/known_hosts 文件中删除包含 pluto@local 字段的那一行。

当然还有一个“一劳永逸”的办法，在 /etc/ssh/sshd_config 文件中设置 StrictModes 为 No。但是这样会降低安全性，如果您的电脑同时担任生产环境运维等需要用到 ssh 的重要场合，请不要更改此参数。

a. 让电脑认识 TinySDR

在 Linux 环境下，请：

- systemctl enable avahi-daemon --now（用于让 SoapySDRUtil 找到 TinySDR，仅此而已，并非必要）
- 通过系统的包管理器（如 apt、dnf/yum 和 pacman）安装 soapyplutosdr 或 soapy-plutosdr 或者其它包含 plutosdr 这个字段的包（安装这个包会同时安装所有与 TinySDR 相关的所有软件依赖）

关于 Docker：我们不建议在这种环境使用 Docker。Docker 的优势是环境隔离，但这种场景中 Docker 带来的优势带来的收益不大。

如果您非要使用 Docker，如下命令可配置开发环境（请务必修改路径）：

```
docker run -it --rm \
    --net=host \
    -e DISPLAY=$DISPLAY \
    -v /tmp/.X11-unix:/tmp/.X11-unix \
    -v /dev/bus/usb/:/dev/bus/usb/ \
    -v /whatever/Desktop/PLUTOSDR:/home/gnuradio/PLUTOSDR \
    --privileged \
    --group-add=audio \
    ubuntu:gnuradio-plutosdr bash
sudo chmod +666 PLUTOSDR
zsh
sudo rm -rf /run/dbus
sudo rm -rf /run/avahi-daemon/pid
sudo mkdir -p /run/dbus # 确保 dbus 运行所需的目录存在
sudo dbus-daemon --system --fork # 以后台模式启动 dbus-daemon
sudo avahi-daemon -D # 后台模式运行
```

3. 交叉编译工具链的配置

A. 关于最新最热发行版和 TinySDR 的奇怪兼容问题

如果你在使用最新最热的 Linux 发行版（如 Arch Linux 等滚动发行 Linux 和 Debian Bookworm 等较新版本的点发行 Linux），请不要使用发行版自带的工具链。TinySDR 自带的软件环境，Glibc 库的版本为 2.28，这个版本是一个很古老的版本，用更新的 C 库会导致一些奇奇怪怪的问题，最终导致你甚至连 Hello World 都跑不起来。

当然这是有一个变通方案的，就是**静态编译**。但由于 TinySDR 存储空间并不大，我们不推荐使用这种方案。

B. 配置交叉编译工具链

找到老版本系列：

<https://developer.arm.com/Downloads/-/Legacy%20Linaro%20GNU%20Toolchains>

并找到相对较新的 7.5.0，基于 x86-64 机器为 ARMhf 构建的版本：

https://developer.arm.com/-/cdn-downloads/permalink/legacy-linaro-gnu-toolchains/7.5-2019.12/gcc-linaro-7.5.0-2019.12-x86_64_arm-linux-gnueabi.tar.xz

并解压。

但请注意，这个版本理论上仍然达不到我们推荐，但是可以使用。我们推荐的 GCC 版本为 8.2.0，但 ARM 官网和 Linaro 官网现不提供，目前我们自己也找不到。所以如果客户自身能力比较强，可以自行下载。

例程 1：Tiny Come!

本例程放在专门的文件夹里面。用于检验交叉编译环境、TinySDR 软件环境和 PC 与 TinySDR 的沟通途径的可用性。

你只需要关注 Makefile，并注意一下各参数——是的，之后的例程都会用到这些。

如果你已经配置好了环境，那接下来，修改其中的参数后，你只需要敲下 make，并敲下 make run，一个 Hello pluto sdr 就在你的机器上跑起来了。

如果你恰巧有一些怪癖，可以修改显示的内容为 Ciallo~(∠ · ω<)∩★，虽然这无关紧要就是。

```
Terminal 终端 - ~/p/g/P/0/01-hello_plutosdr
文件(F) 编辑(E) 视图(V) 终端(T) 标签(A) 帮助(H)
~/p/g/P/0/01-hello_plutosdr
user@archlinux-tinyrfequip ~/p/g/P/0/01-hello_plutosdr (main) [2]> make run
scp -O hello_plutosdr root@pluto.local:~/
The authenticity of host 'pluto.local (192.168.2.1)' can't be established.
ECDSA key fingerprint is: SHA256:yPx1yOELRwbD/lWGzIhqaeZ7cBd+H4+A0uUKBpB9ZEs
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'pluto.local' (ECDSA) to the list of known hosts.
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
root@pluto.local's password:
hello_plutosdr
24 333.2KB/s 00:00
ssh root@pluto.local "chmod +x ~/hello_plutosdr && ~/hello_plutosdr"
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
root@pluto.local's password:
hello pluto sdr
user@archlinux-tinyrfequip ~/p/g/P/0/01-hello_plutosdr (main)>
```

例程 2: Hello AD9361 IIOStream

本例程用于检验 TinySDR 软硬件联动的可用性。

这个例程中，我们将不止参考 GCC 等的事情，还要考虑 libiio 的感受。

LibIIO 是由亚德诺（也就是 Analog 公司）开发的一个用于与 Linux 的 IIO 设备交互的跨平台库。它是我们和 TinySDR、PlutoSDR 等设备的“外交官”。

Makefile 也因此多加了几个参数，需要链接 libiio 库。当然如果你选择静态编译，需同时准备已经静态编译好的 libiio 库。

冷提醒，若需停止该程序，你只需要按下 Ctrl-C。

```
Terminal 终端 - ~/p/g/P/0/02-ad9361-iiostream
文件(F) 编辑(E) 视图(V) 终端(T) 标签(A) 帮助(H)
~/p/g/P/0/02-ad9361-iiostream
user@archlinux-tinyrfequip ~/p/g/P/0/02-ad9361-iiostream (main)> make run
scp -O ad9361-iiostream root@pluto.local:~/
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
root@pluto.local's password:
ad9361-iiostream 100% 9820 2.6MB/s 00:00
#ssh -t root@pluto.local "chmod +x ~/ad9361-iiostream; /bin/sh"
ssh -t root@pluto.local "chmod +x ~/ad9361-iiostream; ./ad9361-iiostream"
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
root@pluto.local's password:
* Acquiring IIO context
* Acquiring AD9361 streaming devices
* Configuring AD9361 for streaming
* Acquiring AD9361 phy channel 0
* Acquiring AD9361 RX lo channel
* Acquiring AD9361 phy channel 0
* Acquiring AD9361 TX lo channel
* Initializing AD9361 IIO streaming channels
* Enabling IIO streaming channels
* Creating non-cyclic IIO buffers with 1 MiS
* Starting IO streaming (press CTRL+C to cancel)
RX 1. 1.05 MSmp, TX 1. 0.05 MSmp
RX 2. 2.10 MSmp, TX 2. 0.10 MSmp
RX 3. 3.15 MSmp, TX 3. 0.15 MSmp
RX 4. 4.19 MSmp, TX 4. 0.19 MSmp
RX 5. 5.24 MSmp, TX 5. 0.24 MSmp
RX 6. 6.29 MSmp, TX 6. 0.29 MSmp
RX 7. 7.34 MSmp, TX 7. 0.34 MSmp
RX 8. 8.39 MSmp, TX 8. 0.39 MSmp
RX 9. 9.44 MSmp, TX 9. 0.44 MSmp
RX 10. 10.49 MSmp, TX 10. 0.49 MSmp
Waiting for process to finish... Got signal 2
RX 11. 11.53 MSmp, TX 11. 0.53 MSmp
* Destroying buffers
* Disabling streaming channels
* Destroying context
Connection to pluto.local closed.
user@archlinux-tinyrfequip ~/p/g/P/0/02-ad9361-iiostream (main)>
```

例程 3：在 TinySDR 上直接进行 FM 解调，并回传语音信息到电脑主机

在该例程中，我们将让 PlutoSDR 完成 FM 调制解调的各步骤，并把音频流传回电脑播放。你需要修改源码中的以下部分：

pluto_fm_radio.c 202-207 行：

```
/* ---- 网络配置 ---- */
/* 注意: 192.168.2.1 是 Pluto 自身 USB 网卡的地址!
* USB 连接时, Pluto 会通过内置 DHCP 把 PC 分配为 192.168.2.10,
* 音频必须发往 PC (192.168.2.10), 而不是 Pluto 自己 (192.168.2.1) */
#define DEST_IP "192.168.2.10" // 接收端 PC 的 IP
#define DEST_PORT 1234 // 接收端 PC 的端口
```

pluto_fm_radio.c 182-185 行：

```
////////////////////////////////////
// FM 相关的逻辑
////////////////////////////////////
#define FM_LO_FREQ 99000000ULL // 目标 FM 频率 99.0 MHz
```

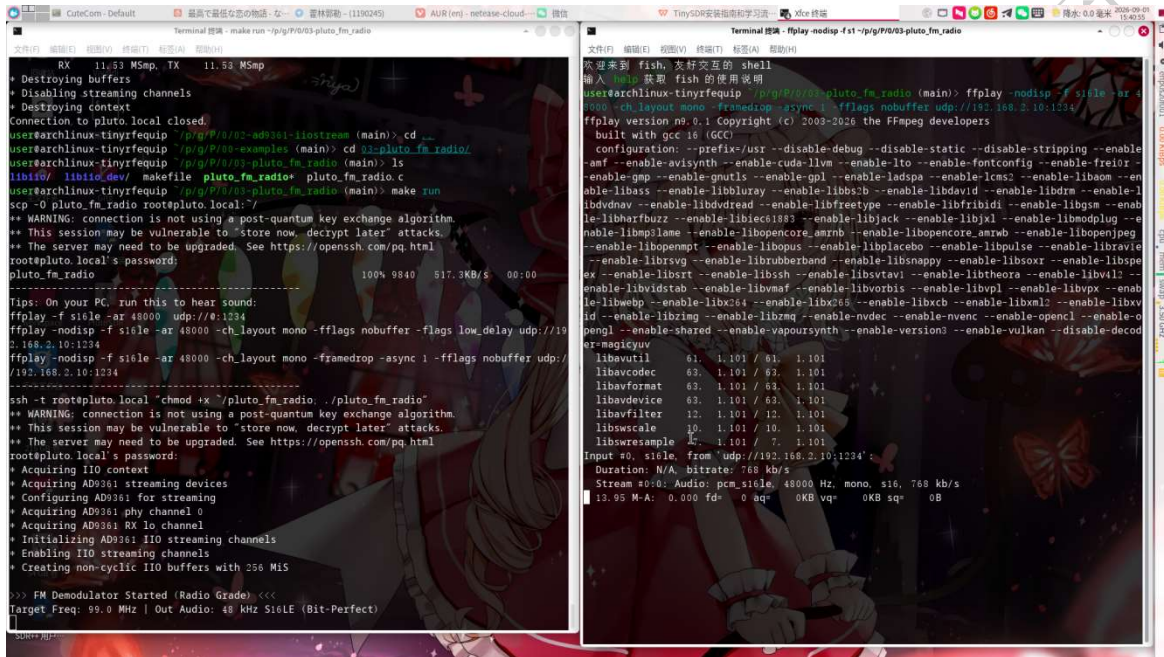
Makefile：如例程 1。

此时 TinySDR 会将音频流通过 UDP 发送给上位机，你需要让 ffmpeg 接受这个音频流，把它发送至 DSP，并由喇叭或耳机发出声音。

如下是具体的将音频流播放出来的命令：

- `ffplay -f s16le -ar 48000 udp://@:1234`
- `ffplay -nodisp -f s16le -ar 48000 -ch_layout mono -fflags nobuffer -flags low_delay udp://192.168.2.10:1234`
- `ffplay -nodisp -f s16le -ar 48000 -ch_layout mono -framedrop -async 1 -fflags nobuffer udp://192.168.2.10:1234`

三条命令选一条喜欢的执行即可。



Q & A

为什么我听不到声音？

请首先检查拦截包的部分组件，例如：

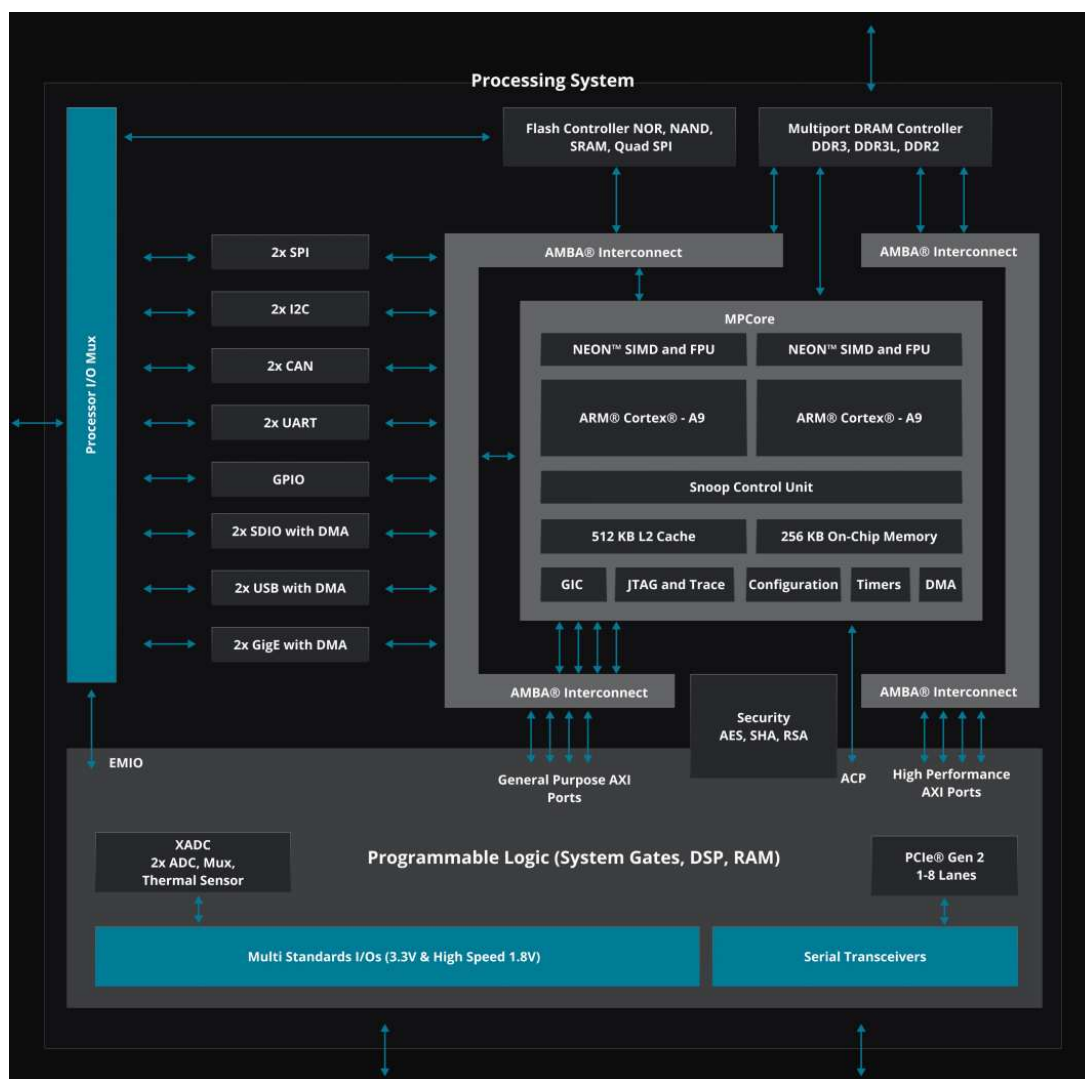
- 代理、网络分程序：mihomo/Clash Meta 等（一般启用它们最多只会造成卡顿）
- 防火墙：ufw、firewalld 等

无论如何，关闭它们，或放行 **UDP 流** 即可。请不要在公网环境下随意关闭防火墙。

例程 4：FFT 频谱仪

这个例程中，我们将令 PlutoSDR 实现吞吐 10MHz 采样率、动态自适应硬件带宽、汉宁窗（Hamming）平滑时域信号、片上 1024 点高级 kissFFT 变换、功率谱密度（dB）换算与 FFT-Shift 变频、多帧片上平滑过滤，输出 1024 个浮点数送到上位机生成频谱。

Makefile 中，我们可以注意到其启用了一些针对硬件特性的优化选项。如 NEON 指令集等，如果你翻 Cortex A9 的文档，你会发现是可选的，但 TinySDR 所使用的 Zynq 7020：



巧了，它真的有！

这个例程包含四个你需要注意的文件：

- Makefile: 编译脚本
- pluto_spectrum_analyzer.c: 下位机频谱分析源码
- pluto_spec_viewer.py: 上位机源码（GTK）
- pluto_spec_viewer_terminal.py: 上位机源码（TUI/终端）（不推荐使用）

你需要开两个终端页面，一个挂在前台在下位机跑频谱分析（执行 `make` 和 `make run`），一个打开上位机（`python pluto_spec_viewer.py`）。

makefile 有几个参数：

使用 `?` 允许命令行传参覆盖。若命令行未指定，则使用默认值

`make run` `FREQ=100.0` `SAMPLE=4`

`IP` `?` `= 192.168.2.10` # 你的主机 IP

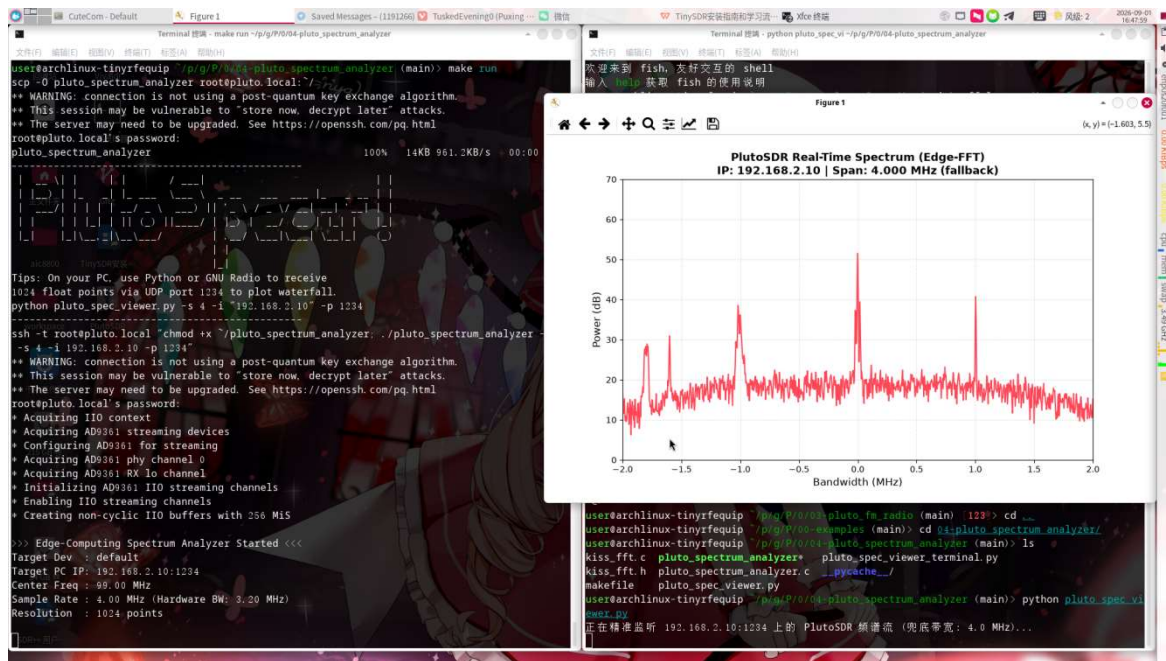
`PORT` `?` `= 1234` # 主从机端口号

`FREQ` `?` `= 93` # 中心频点

`SAMPLE` `?` `= 10` # 采样频率

这些参数主要影响软件运行（`make run`），不影响编译。

上位机运行指令为 `python pluto_spec_viewer.py`。



例程 5：实现飞机的 ADS-B 信号的解码，并发送至电脑显示

该例程中，我们将使用 TinySDR 实现 ADS-B 信号的解码，并发送至电脑的上位机对信息进行显示。

ADS-B (Automatic Dependent Surveillance – Broadcast) 是民航飞机定期广播的航班信息报文，工作在 1090 MHz 频率。每一架飞机大约每秒发送 6~10 条 112 位的短报文，包含了 ICAO 地址、航班号、高度、航向速度等关键信息。

这个例程包含四个文件：

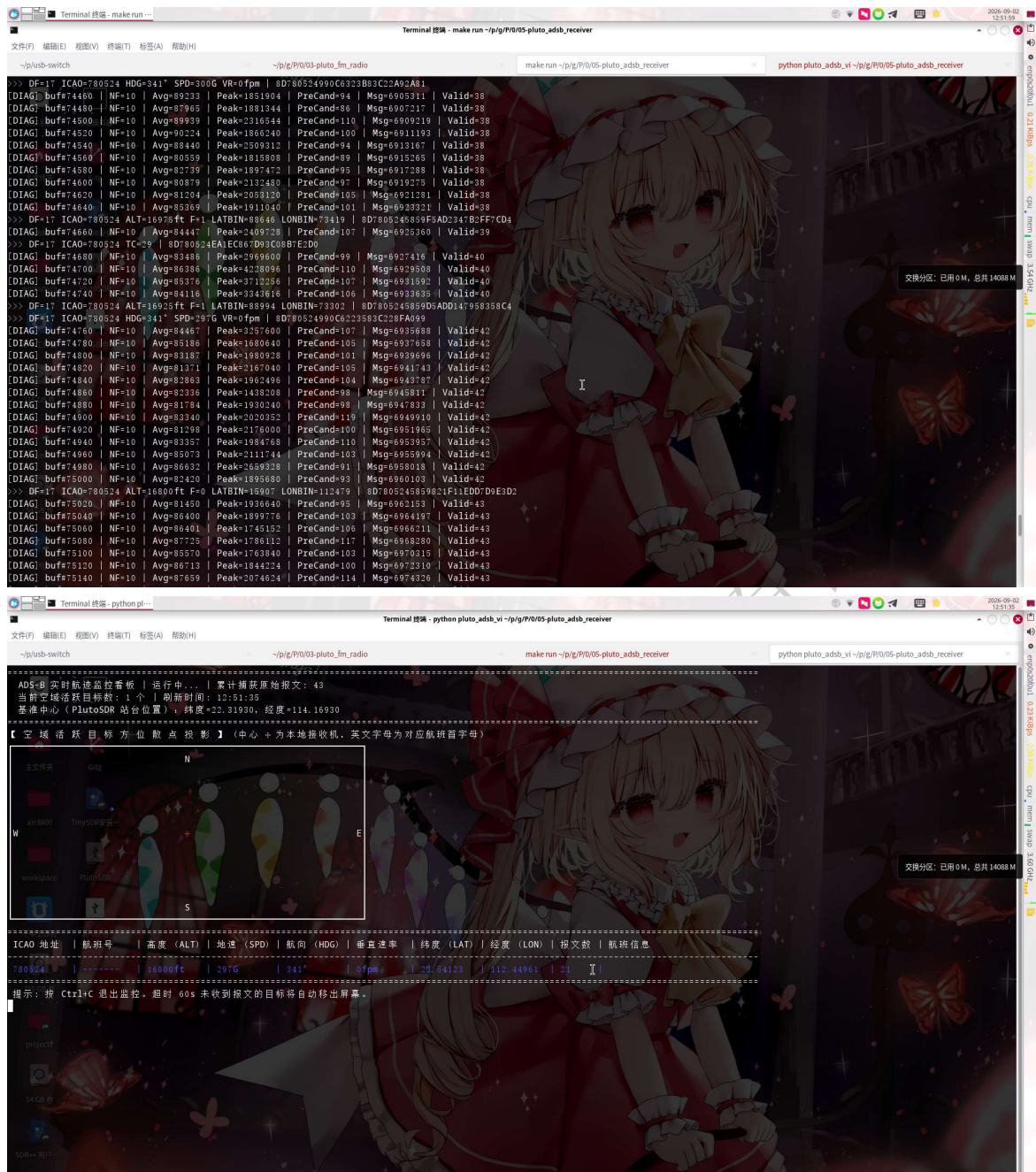
- Makefile：编译脚本
- pluto_adsb_receiver.c：下位机源码（解析 ADS-B 并发送至上位机）
- pluto_adsb_receiver.py：上位机源码（解析下位机发送的信息）
- flight_info.py：航班信息库

makefile 等如例程 4。

因为 TinySDR 执行了从解码到解析的全流程，到达上位机的只有一个包含诸如以下内容的字符串：

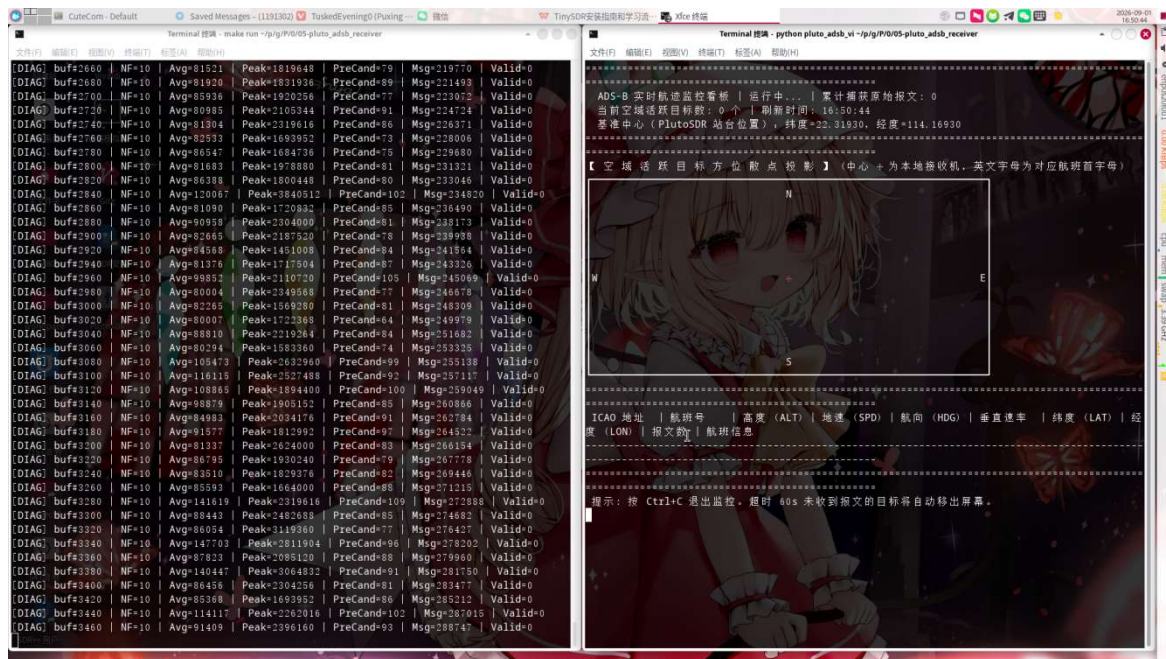
DF=17 ICAO=7808B7 ALT=29325ft F=1 LATBIN=82686 LONBIN=80566 |
8D7808B7589755A17F4EADAAA2C1

故上位机不需要管那些弯弯绕，它只需要知道该从哪拿到消息（下位机 IP 与端口，通过 UDP 通信），并展示出来即可。



Q & A

为什么下位机输出了一堆日志，但上位机画不出任何东西？
是不是像这样？



原因非常简单且生草，附近没有飞机。文档中展示的截图我们是等到中午才截到的。

C. Q & A

设备/电脑上有音频，但在应该有信号的地方却没有信号？

并非所有电台都持续广播——请调谐到已知的持续广播信号，例如本地广播电台或 24 小时 VOLMET 电台。如果仍然没有信号，请检查以下各项：

- 天线是否连接？
- AGC 或增益是否设置得当？
- 如果您使用 RTL-SDR 棒，直接采样（direct sampling）是否设置为 Q 支路（用于 28 MHz 以下电台）或禁用（用于 VHF/UHF 传输）？
- 是否选择了正确的调制方式，例如 AM、FM、下边带/上边带等？
- 音频输出（sink）是否设置为扬声器输出？

FFT 或瀑布图中显示了信号，但点击/轻触调谐时，VFO 没有移动到信号峰值？

将捕捉间隔（snap interval）调小，以便使用更小的调谐步进。

我已经等了 10 秒了，为什么还没开机？

请再等等，三分钟后若仍未开机，请重新刷写系统或进行售后。

我应该如何提供日志？

将随附的 A to C 线接入开发板的 JTAG Type-C 口，并打开串口工具（如 MobaXterm、minicom）。预期会看到一长串如下格式的信息：

```
CuteCom - Default  わんだーらんど・でさくら...  霍林郭勒 - (1190066)  AUR (en) - netaease-cloud-...  微信
Terminal 终端
文件(F)  编辑(E)  视图(V)  终端(T)  标签(A)  帮助(H)
sudo minicom -D /dev ~
sudo minicom

Copying Linux from QSPI flash to RAM...
SF: Detected N25Q256 with page size 256 Bytes, erase size 4 KiB, total 32 MiB
device 0 offset 0x500000, size 0x400000
SF: 4194304 bytes @ 0x500000 Read: OK
device 0 offset 0x930000, size 0x8a0000
SF: 9043968 bytes @ 0x930000 Read: OK
device 0 offset 0x900000, size 0x5000
SF: 20480 bytes @ 0x900000 Read: OK
## Booting kernel from Legacy Image at 02080000 ...
  Image Name:      uImage
  Image Type:      ARM Linux Kernel Image (uncompressed)
  Data Size:       4137808 Bytes = 3.9 MiB
  Load Address:   02080000
  Entry Point:     02080000
  Verifying Checksum ... OK
## Loading init Ramdisk from Legacy Image at 04000000 ...
  Image Name:
  Image Type:      ARM Linux RAMDisk Image (gzip compressed)
  Data Size:       8970118 Bytes = 8.6 MiB
  Load Address:   00000000
  Entry Point:     00000000
  Verifying Checksum ... OK
## Flattened Device Tree blob at 02000000
  Booting using the fdt blob at 0x2000000
  Loading Kernel Image ... OK
  Loading Ramdisk to 1e286000, end 1eb13f86 ... OK
  Loading Device Tree to 1e27e000, end 1e285a5d ... OK

Starting kernel ...

Booting Linux on physical CPU 0x0
Linux version 5.4.0 (zkf@ubuntu) (gcc version 8.2.0 (GCC)) #46 SMP PREEMPT Thu Jun 4 11:13:39 CST 2026
CPU: ARMv7 Processor [413fc090] revision 0 (ARMv7), cr=18c5387d
CPU: PIPT / VIPT nonaliasing data cache, VIPT aliasing instruction cache
OF: fdt: Machine model: Analog Devices PlutoSDR Rev.B (Z7020/AD9364)
printk: bootconsole [earlycon0] enabled
Memory policy: Data cache writealloc
cma: Reserved 256 MiB at 0x0e000000
On node 0 totalpages: 131072
  Normal zone: 1024 pages used for memmap
  Normal zone: 0 pages reserved
```

(Linux 中, 通常为/dev/ttyUSB0 或/dev/ttyUSB1, 建议通过 minicom 提前打开串口并为设备硬重启/下电后等三秒再上电)

这些是 U-Boot 的启动日志和 Linux 内核的启动日志。请把这些完整复制下来, 发送给售后, 或自行通过各手段诊断。

若未看到任何日志, 请检查拨码开关, 并考虑重刷固件。

为什么我使用旧版本的交叉编译工具链编译我的程序, 仍然可以在 TinySDR 上跑?

Glibc (GNU/Linux 使用的 C 语言库) 向下兼容。

为什么我使用新版本的交叉编译工具链编译我的程序, 却没办法在 TinySDR 上跑?

Glibc (GNU/Linux 使用的 C 语言库) 虽然向下兼容, 但不是这样向下。

使用新版本的 glibc 编译出来的程序会使用一些新的接口, TinySDR 上使用的较旧的可能

没有，因此会遇到一些奇奇怪怪的错误。

beautifulzzz/btfzzz 的例程和我们提供的例程有什么区别？

你可能发现我们保留了源码中的版权内容。

我们修改了一些例程，试图去除或修正了一部分错误的部分。我们保证提供的例程可用、正确且专业，避免出现 AI slop 或开源灵车大冲击的情形。我们同样保证，TR-260726 由我们负责地审核，且会明确标注使用 AI 的情况。

这些错误我们将回馈给上游。

泰瑞合创(天津)科技有限公司