



Adam Hua

Central FAE

摘要

本文提供了一份完整的卫星雷达系统指南，详细介绍如何将 AWR2188 76-81 GHz 毫米波射频前端 (MMIC，millimeter wave integrated circuit) 与 NVIDIA Jetson AGX Orin 平台连接，用于卫星雷达应用。该系统利用 TI 的 FPD-Link IV 加解串器 (SerDes) 技术将高速雷达数据远距离传输至 Orin 的 CSI-2 接口。本文详细阐述了硬件架构、雷达配置流程、SerDes 配置以及 Orin 平台配置，并提供了验证方法以确保系统正常运行。

内容

|                                        |    |
|----------------------------------------|----|
| 1 简介.....                              | 2  |
| 2 AWR2188+FPDLINK+AGX ORIN 系统详细描述..... | 3  |
| 2.1 系统架构.....                          | 3  |
| 2.2 AWR2188 雷达配置.....                  | 4  |
| 2.3 FPD-Link SerDes 配置.....            | 4  |
| 2.4 NVIDIA Orin 平台配置示例.....            | 5  |
| 2.5 完整系统验证实例.....                      | 7  |
| 2.6 系统运行过程中故障处理建议.....                 | 9  |
| 3 总结.....                              | 10 |
| 4 参考文献.....                            | 10 |

插图清单

|                                        |   |
|----------------------------------------|---|
| 图 2-1. Orin-FPD-Link-AWR2188 硬件连接..... | 3 |
| 图 2-2. AWR2188 数据流管线.....              | 4 |
| 图 2-3. lsmod   grep ov5693 示例.....     | 9 |
| 图 2-4. dmesg   grep -i ov5693 示例.....  | 9 |

商标

所有商标均为其各自所有者的财产。

## 1 简介

现代卫星雷达系统即需要优秀的射频前端芯片，也需要高性能数据处理平台。TI 的 AWR2188 雷达收发器与 NVIDIA 的 Jetson AGX Orin 的组合提供了一个优秀解决方案，具有以下优势：

- 高分辨率雷达感知 - AWR2188 的 76-81 GHz 调频连续波 ( Frequency Modulated Continuous Wave , FMCW ) 能力
- 稳健的远距离数据传输 - 使用 FPD-Link IV SerDes 技术

本文重点关注系统集成方面：雷达配置、SerDes 设置以及 Orin 平台配置，确保能够在 Orin 上通过 V4L2(Video4Linux2)接口成功捕获原始雷达数据。点云生成的信号处理算法不在本文范围内。

## 2 AWR2188+FPDLINK+AGX ORIN 系统详细描述

### 2.1 系统架构

#### 2.1.1 硬件组件

完整系统由四个主要硬件组件组成:

1. AWR2188 卫星板 - 包含 TI 的 76-81 GHz 雷达 MMIC (集成了发射和接收链路) 和 UB971 加串器。请访问 TI 网站以获取该[参考设计](#)。
2. UB9702 解串器评估板 - TI FPD-Link IV SerDes, 用于高速 CSI-2 数据传输。请访问 TI 网站以获取该[参考设计](#)。
3. NVIDIA Jetson AGX Orin 64G 开发套件 - 运行 Linux for Tegra(L4T)R36.4.4 的处理平台。
4. UB9702 至 Orin 适配板 - 将解串器输出连接至 Orin CSI 端口的定制接口板。由于 Orin 版本的不同, 接口定义可能有差异, 用户可以自行定义和制作转接板。

硬件参考连接如下如:

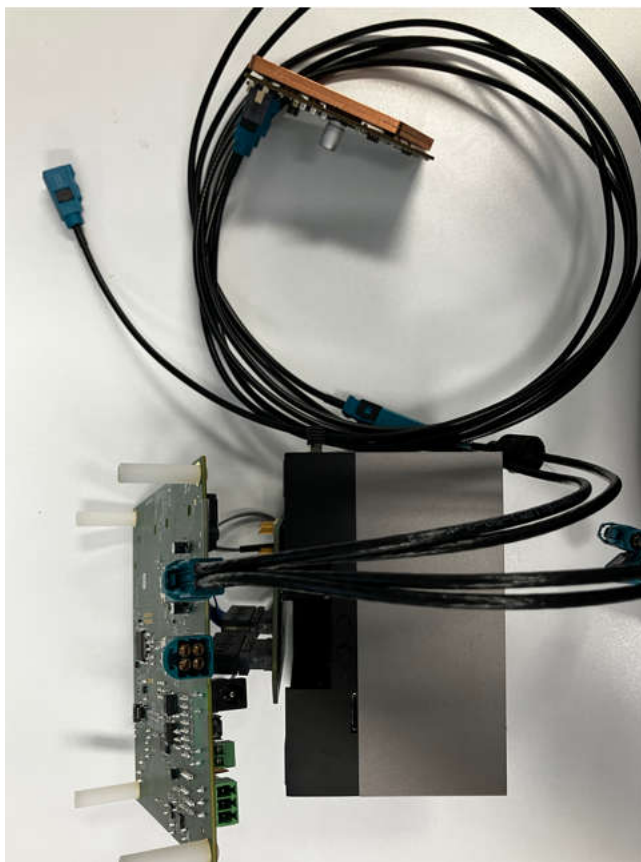


图 2-1. Orin-FPD-Link-AWR2188 硬件连接

#### 2.1.2 数据流管线

AWR2188 生成的雷达数据使用 FPD-Link IV 加串器, 将 AWR2188 输出的 CSI2 数据转换为串行数据, 通过 FAKRA 连接线传输到 UB9702 解串器。解串器将串行数据转换为 CSI-2 格式并传输至 Orin 的 CSI-2 摄像头接口, 在 Orin 的 Linux 系统中显示为 V4L2 视频设备。

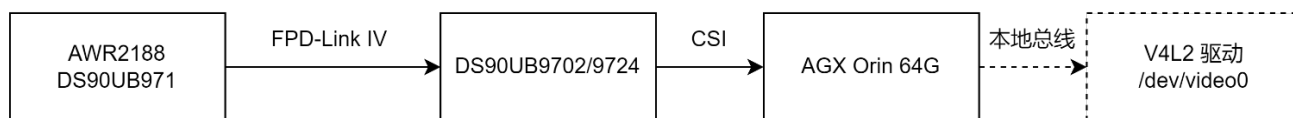


图 2-2. AWR2188 数据流管线

### 2.1.3 Orin 平台软件栈

表 2-1. 实例使用软件版本

| 组件         | 版本/详情                  |
|------------|------------------------|
| NVIDIA L4T | R36.4.4 (Jetson Linux) |
| Linux 内核   | 5.15.148-tegra (RT 内核) |
| 视频接口       | V4L2 (Video4Linux2)    |

## 2.2 AWR2188 雷达配置

AWR2188 的 CSI 配置模式支持 RAW8，RAW12 和 RAW14。对于单独一帧，所有的数据必须是相同的配置。在发送一个 chirp 的数据时，能够灵活配置行头行尾的位置，通常会在一个 chirp 的头上配置一个行开始 ( Horizontal Start, HS )，在 chirp 结尾配置一个行尾 ( Horizontal End, HE )。在采用 16 位采样率进行采样时，ADC 数据应以 RAW8 格式从 CSI-2 发出，在采用 12 位采样时，数据应以 RAW12 发出。在 AWR2188 中可以配置 CSI 测试序列生成器以快速测试 CSI 报文内容。

### 2.3 FPD-Link SerDes 配置

UB9702 解串器提供了远程 AWR2188 雷达与 Orin 平台之间的关键链路。该配置可通过两种方式完成：①将解串器板的 I<sup>2</sup>C 信号直接接至 Orin 套件的 I<sup>2</sup>C 接口，利用 Orin 的 I<sup>2</sup>C 读写操作进行配置；②使用 USB2ANY 工具直接对解串器进行操作。在配置解串器时，一般需设置端口的使能、波特率以及接收端口与发送端口的映射关系等参数。TI 在 Orin 平台上支持完整的加串器 UB971 和解串器 UB9702 的寄存器配置脚本。

在配置完成后，需要监控解串器的部分寄存器来验证链路是否完好。

表 2-2. 解串器关键状态寄存器

| 寄存器       | 名称       | 良好链路值                     |
|-----------|----------|---------------------------|
| 0x4D      | 端口状态 1   | 0x03 (端口 0) 或 0x43 (端口 1) |
| 0x4E      | 端口状态 2   | 0x04                      |
| 0x55/0x56 | 奇偶校验错误   | 0x00                      |
| 0x7A      | CSI 状态   | 0x00                      |
| 0x78      | CSI 错误计数 | 0x00                      |

## 2.4 NVIDIA Orin 平台配置示例

### 2.4.1 L4T 版本与配置

本示例使用 [NVIDIA Jetson Linux\(L4T\)R36.4.4](#)。版本选择和相关 SDK 使用请参考 NVIDIA 提供的[官方教程](#)。

初始设置如下：

```
# 解压 L4T 包
tar xjf Jetson_Linux_r36.4.4_aarch64.tbz2

# 解压示例根文件系统
cd Linux_for_Tegra/rootfs/
sudo tar xjpf ../../Jetson_Linux_R36.4.4_aarch64.tbz2

# 应用二进制文件
cd ..
sudo ./apply_binaries.sh

# 烧录设备
sudo ./flash.sh jetson-agx-orin-devkit mmcblk0p1
```

### 2.4.2 设备树配置

定制设备树覆盖(.dtbo 文件)对于定义 CSI 接口参数至关重要。

所需的设备树修改：

1. **CSI 端口映射**：定义 CSI 通道 ↔ V4L2 视频设备的映射关系
2. **帧格式支持**：指定支持的分辨率和像素格式
3. **传感器模式表**：定义每个支持模式的时序参数

设备树片段代码示例：

```
/ {
    camera_sensor: camera@10 {
        compatible = "nvidia,nv_ov5693";
        reg = <0x10>;

        mode0 { // 4096x32 RAW8
            active_w = "4096";
            active_h = "32";
            pixel_format = "bayer_rggb8";
            csi_pixel_bit_depth = "8";
            line_length = "4096";
            inherent_gain = "1";
            mclk_multiplier = "2.0";
            // 其他时序参数...
        };

        mode1 { // 4096x1536 RAW12
            active_w = "4096";
            active_h = "1536";
            pixel_format = "bayer_rggb12";
            csi_pixel_bit_depth = "12";
            line_length = "6144";
            inherent_gain = "1";
            mclk_multiplier = "2.0";
            // 其他时序参数...
        };
        // 其他模式...
    };
};
```

**备注**

在 L4T R36.4.4 中，设备树二进制文件(.dtb)和覆盖文件(.dtbo)存储在 UEFI 分区中。更新需要烧录引导加载程序分区：

```
# 构建设备树后
cd Linux_for_Tegra/sources
# 在此处构建...
cp -r kernel-devicetree/generic-dts/dtbs/* ../kernel/dtb/

# 仅烧录引导加载程序分区(包含 UEFI + DTBO)
cd ..
sudo ./flash.sh -k A_cpu-bootloader jetson-agx-orin-devkit mmcblk0p1
```

**2.4.3 内核驱动修改**

本文以 nv\_ov5693.ko 摄像头驱动作为基础制作 AWR2188 的驱动。

树外(OOT)模块的构建过程：

```
# 导航到内核源码
cd Linux_for_Tegra/sources

# 使用 source_sync.sh 获取内核源码
./source_sync.sh -k jetson_36.4.4

# 导航到驱动源码
cd kernel/nvidia/drivers/media/i2c

# 修改 nv_ov5693.c 以添加自定义模式表
# 关键修改：
# 1. 添加雷达帧大小的模式定义
# 2. 更新支持的格式列表

# 构建 OOT 模块
cd ../../../../..
export CROSS_COMPILE=/usr/bin/aarch64-linux-gnu-
export KERNEL_HEADERS=<内核头文件路径>
make -C kernel/nvidia modules

# 查找编译的模块
find . -name "nv_ov5693.ko"
```

部署修改后的驱动：

为了快速迭代而无需完整系统重新烧录，具体流程见下：

```
# 在构建机器上
cp Linux_for_Tegra/rootfs/lib/modules/5.15.148-tegra/updates/drivers/media/i2c/nv_ov5693.ko /
media/usb/

# 将 USB 驱动器插入 Orin 并复制
sudo cp /media/usb/nv_ov5693.ko /lib/modules/5.15.148-tegra/updates/drivers/media/i2c/

# 重新加载模块
sudo rmmod nv_ov5693
sudo modprobe nv_ov5693
```

**备注**

在设备树中必须有计划使用的分辨率和格式。

**2.4.4 配置验证**

配置后，验证系统设置：

```
# 检查可用的视频设备
ls /dev/video*

# 列出 video0 支持的格式
v4l2-ctl --device /dev/video0 --list-formats-ext
```

```
# 预期输出应显示:
# [0]: 'RGGB' (8-bit Bayer RGRG/GBGB)
#       Size: Discrete 4096x32
#       Interval: Discrete 0.033s (30.000 fps)
#       Size: Discrete 4096x1536
#       Interval: Discrete 0.033s (30.000 fps)
#       ...

# 检查媒体控制器拓扑
media-ctl -p

# 验证 CSI 接口
dmesg | grep -i csi

# 测试数据捕获
v4l2-ctl -d /dev/video0 --stream-mmap
# 当雷达传输时应无错误地流式传输
```

## 2.5 完整系统验证实例

### 2.5.1 软硬件启动顺序

1. 启动 Orin
2. 启动 AWR2188 并等待初始化
3. 配置 9702/9724
4. 验证 SerDes 链路
5. 验证 V4L2 设备
6. 通过 V4L2 进行数据捕获

需要注意的是，步骤 2 可以发生在步骤 1 之前。

### 2.5.2 SerDes 链路验证

运行 SerDes 配置脚本：

```
python3 python/B8_2188.py
```

预期输出：

配置端口 0...

配置端口 1...

DES CSI\_Port 0 时钟通道 = CONT

DES CSI\_Port 0 通道数 = 4

DES CSI\_Port 1 时钟通道 = CONT

DES CSI\_Port 1 通道数 = 4

DES CSI 端口 0 设置为轮询转发 Rx 端口 [0]

DES CSI 端口 1 设置为轮询转发 Rx 端口 [1]

##### 链路测试 #####

#端口 0 链路良好

端口 0 状态寄存器值：

端口 0 状态 1 (0x4D) = 0x3

端口 0 状态 2 (0x4E) = 0x4

端口 0 奇偶校验错误 (0x55, 0x56) = 0

端口 0 CSI 状态 (0x7A) = 0x0

端口 0 CSI 错误计数 (0x78) = 0

#端口 1 链路良好

端口 1 状态寄存器值:

端口 1 状态 1 (0x4D) = 0x43

端口 1 状态 2 (0x4E) = 0x4

端口 1 奇偶校验错误 (0x55, 0x56) = 0

端口 1 CSI 状态 (0x7A) = 0x0

端口 1 CSI 错误计数 (0x78) = 0

### 2.5.3 V4L2 捕获测试

[illegible]



## 2.6 系统运行过程中故障处理建议

### 2.6.1 Orin 启动后没有视频设备

- 验证内核驱动已加载: `lsmod | grep ov5693`
- 检查内核日志: `dmesg | grep -i ov5693`

通常该问题出现在驱动没有加载和设备树没有覆盖。图 2 和图 3 分别给出验证驱动示例。

```
nv_ov5693          16384  0
tegra_camera      249856  4 nvhost_isp5,nvhost_nvcsi_t194,nv_ov5693,nvhost_vi5
```

图 2-3. `lsmod | grep ov5693` 示例

```
[ 10.331858] ov5693 2-0038: probing v4l2 dummy sensor.
[ 10.332018] ov5693 2-0038: tegracam sensor driver:ov5693_v2.0.6
[ 10.332094] tegra-camrtc-capture-vi tegra-capture-vi: subdev ov5693 2-0038 bound
[ 10.346332] ov5693 2-0038: Detected OV5693 dummy sensor
[ 10.346899] ov5693 2-0036: probing v4l2 dummy sensor.
[ 10.347293] ov5693 2-0036: tegracam sensor driver:ov5693_v2.0.6
[ 10.347542] tegra-camrtc-capture-vi tegra-capture-vi: subdev ov5693 2-0036 bound
[ 10.355041] ov5693 2-0036: Detected OV5693 dummy sensor
```

图 2-4. `dmesg | grep -i ov5693` 示例

### 2.6.2 SerDes 链路检查报错

- 验证线缆连接。
- 检查 FPD-Link 端口寄存器值 (如表 2 所示)。初次检查出错时可以重复读取, 大部分错误寄存器是读清的。首次读取的错误可以忽略。
- 确保所有组件的电源供应正常。

### 2.6.3 V4L2 捕获未显示数据

- 验证 SerDes 链路良好(按照表 2 检查状态寄存器)。
- 确认 AWR2188 正在传输(如有, 检查雷达状态 LED)。
- 检查解串器的输出计数寄存器如 0x90, 0x91 (帧数计数), 0x95, 0x96 (行计数)。
- 在 Orin 上运行 `dmesg`, 检查 CSI 驱动 log。在长时间没有收到任何数据时, 会有超时错误, 在收到的数据格式与预期不一致时, 会有丢帧错误。
- 检查 CSI 时钟是否存在且频率正确。

### 3 总结

本文提供了一份通过 FPD-Link SerDes 将 TI 的 AWR2188 雷达 MMIC 连接到 NVIDIA 的 Jetson AGX Orin 平台的完整系统集成指南。该系统能够以高帧率和高分辨率实时捕获雷达数据，为高级雷达信号处理和 AI 应用提供了坚实的基础。

### 4 参考文献

1. <https://www.ti.com/product/AWR2188>
2. <https://www.ti.com/interface/fpdlink-serdes/overview.html>
3. <https://www.ti.com/product/DS90UB971-Q1>
4. <https://www.ti.com/product/DS90UB9702-Q1>
5. <https://docs.nvidia.com/jetson/archives/r36.4.4/DeveloperGuide/>
6. <https://www.kernel.org/doc/html/latest/userspace-api/media/v4l/v4l2.html>
7. TIDA-020093 reference design | TI.com

## 重要通知和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、与某特定用途的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他安全、安保法规或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的相关应用。严禁以其他方式对这些资源进行复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。对于因您对这些资源的使用而对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，您将全额赔偿，TI 对此概不负责。

TI 提供的产品受 [TI 销售条款](#)、[TI 通用质量指南](#) 或 [ti.com](#) 上其他适用条款或 TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。除非德州仪器 (TI) 明确将某产品指定为定制产品或客户特定产品，否则其产品均为按确定价格收入目录的标准通用器件。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

版权所有 © 2026，德州仪器 (TI) 公司

最后更新日期：2025 年 10 月