Application Note

Integrating AWR2188 Satellite Radar with AGX ORIN via FPD-Link

Adam Hua *Central FAE*

ABSTRACT

This document was translated from a simplified Chinese source. (ZHCAG78)

This document provides a comprehensive guide for satellite radar systems, detailing the methodology for interfacing the AWR2188 76 - 81GHz millimeter-wave RF front-end (MMIC) with the NVIDIA Jetson AGX Orin platform for satellite radar applications. The system leverages TI's FPD-Link IV SerDes technology to transmit high-speed radar data over extended distances to the Orin's CSI-2 interface. This document elaborates on the hardware architecture, radar configuration workflow, SerDes setup, and Orin platform configuration, along with validation methodologies to ensure proper system operation.

Table of Contents

List of Figures

Trademarks

1 Introduction

Modern satellite radar systems demand both exceptional RF front-end chips and high-capability data processing platforms. The combination of TI's AWR2188 radar transceiver and NVIDIA's Jetson AGX Orin offers an optimal solution, providing:

- High-resolution radar sensing – leveraging the AWR2188's 76–81GHz frequency modulated continuous wave (FMCW) capabilities
- Robust long-range data transmission – enabled by FPD-Link IV SerDes technology

This document focuses on system integration aspects: radar configuration, SerDes setup, and Orin platform configuration, ensuring successful raw radar data capture via the V4L2 (Video4Linux2) interface on the Orin. Signal processing algorithms for point-cloud generation are outside the scope of this document.

2 AWR2188+FPDLINK+AGX ORIN System Detailed Description

2.1 System Architecture

2.1.1 Hardware Components

The complete system consists of four primary hardware components:

1. AWR2188 Satellite Board – Contains the TI 76–81GHz radar MMIC (integrating transmit and receive chains) and the UB971 serializer. Please refer to the TI website for this reference design.
2. UB9702 Deserializer Evaluation Board – TI FPD-Link IV SerDes for high-speed CSI-2 data transmission. Please refer to the TI website for this reference design.
3. NVIDIA Jetson AGX Orin 64G Developer Kit – Processing platform running Linux for Tegra (L4T) R36.4.4.
4. UB9702-to-Orin Adapter Board – Custom interface board connecting the deserializer output to the Orin CSI ports. As interface definitions may vary across Orin revisions, users may define and fabricate their own adapter boards.

The hardware reference connection is shown below:

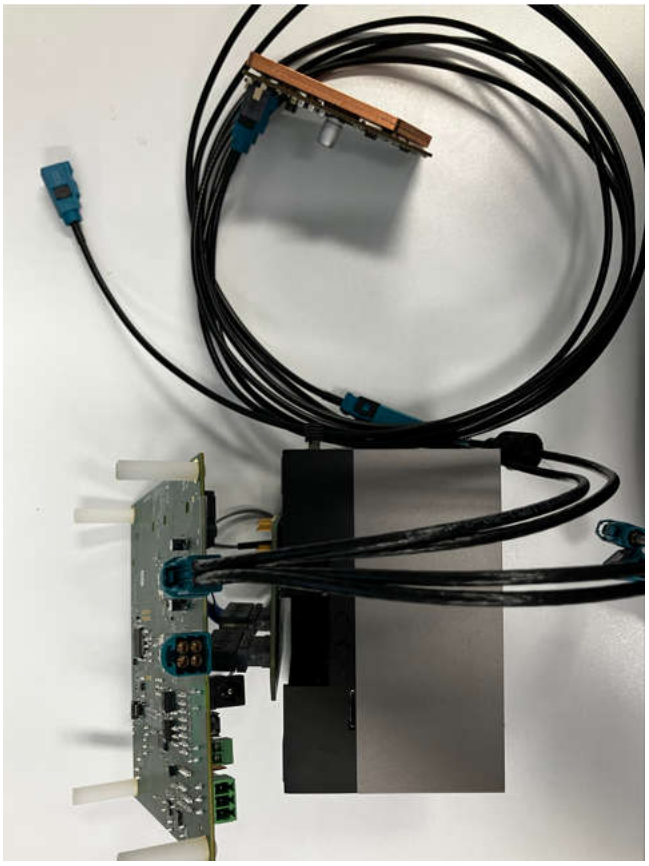

Figure 2-1. Orin-FPD-Link-AWR2188 Hardware Connection

2.1.2 Data Flow Pipeline

The radar data generated by the AWR2188 is transmitted via the FPD-Link IV serializer, which converts the CSI2 data output from the AWR2188 into serialized data. This data is transmitted over FAKRA cabling to the UB9702 deserializer. The deserializer converts the serial data back to CSI-2 format and forwards it to the Orin's CSI-2 camera interface, where it appears as a V4L2 video device within the Orin's Linux system.

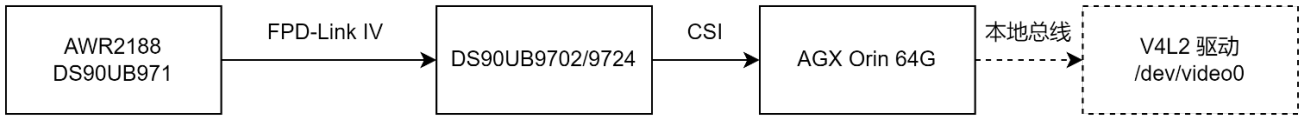

Figure 2-2. AWR2188 Data Flow Pipeline

2.1.3 Orin Platform Software Stack

Table 2-1. Software Versions Used in This Example

Component	Version/Details
NVIDIA L4T	R36.4.4 (Jetson Linux)
Linux Kernel	5.15.148-tegra (RT kernel)
Video interface	V4L2 (Video4Linux2)

2.2 AWR2188 Radar Configuration

The AWR2188 supports CSI configuration modes for RAW8, RAW12, and RAW14. For any given frame, all data must use the same configuration. When transmitting data for a single chirp, the placement of line start and line end markers can be flexibly configured. Typically, a Horizontal Start (HS) marker is placed at the beginning of a chirp, and a Horizontal End (HE) marker at the end of a chirp. When sampling at 16-bit resolution, ADC data shall be transmitted from CSI-2 in RAW8 format; when sampling at 12-bit resolution, data shall be transmitted in RAW12 format. The AWR2188 supports configuration of a CSI test pattern generator for rapid validation of CSI packet content.

2.3 FPD-Link SerDes Configuration

The UB9702 deserializer provides the critical link between the remote AWR2188 radar and the Orin platform. Configuration can be accomplished through two methods: ① connecting the deserializer board's I²C signals directly to the Orin kit's I²C interface, utilizing Orin I²C read/write operations for configuration; ② using the USB2ANY tool to directly manipulate the deserializer. During deserializer configuration, parameters such as port enable, baud rate, and receive-to-transmit port mapping must typically be set. TI provides complete register configuration scripts for both the UB971 serializer and UB9702 deserializer on the Orin platform.

After configuration, certain deserializer registers should be monitored to verify link integrity.

Table 2-2. Deserializer Key Status Registers

Register	Name	Good Link Value
0x4D	Port Status 1	0x03 (Port 0) or 0x43 (Port 1)
0x4E	Port Status 2	0x04
0x55/0x56	Parity Error	0x00
0x7A	CSI Status	0x00
0x78	CSI Error Count	0x00

2.4 NVIDIA Orin Platform Configuration Example

2.4.1 L4T Version and Setup

This example uses NVIDIA Jetson Linux (L4T) R36.4.4. For version selection and associated SDK usage, please refer to the official NVIDIA documentation.

Initial setup:

```
# 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 Device Tree Configuration

Custom device tree overlays (.dtbo files) are essential for defining CSI interface parameters.

Required device tree modifications:

1. **CSI port mapping:** Defines the mapping between CSI channels and V4L2 video devices
2. **Frame format support:** Specify supported resolutions and pixel formats
3. **Sensor mode table:** Define timing parameters for each supported mode

Device tree fragment example:

```
/ {
    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...
    };
};
```

Note

In L4T R36.4.4, device tree binaries (.dtb) and overlay files (.dtbo) are stored in the UEFI partition. Updates require flashing the bootloader partition:

```
# 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 Kernel Driver Modifications

This document uses the nv_ov5693.ko camera driver as a base for developing the AWR2188 driver.

Out-of-Tree (OOT) Module Build Process:

```
# 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"
```

Deploying the Modified Driver:

To enable rapid iteration without full system re-flashing, follow the procedure below:

```
# 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
```

Note

The resolutions and formats to be used must be defined in the device tree.

2.4.4 Configuration Verification

After configuration, verify the system setup:

```
# 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 Complete System Validation Example

2.5.1 Software and Hardware Boot Sequence

1. Power on Orin
2. Power on AWR2188 and wait for initialization
3. Configure 9702/9724
4. Verify SerDes link
5. Verify V4L2 device
6. Capture data via V4L2

Note: Step 2 may occur before Step 1.

2.5.2 SerDes Link Verification

Run the SerDes configuration script:

python3 python/B8_2188.py

Expected output:

Configuring Port 0...

Configuring Port 1...

DES CSI_Port 0 clock channel = CONT

DES CSI_Port 0 number of lanes = 4

DES CSI_Port 1 clock channel = CONT

DES CSI_Port 1 number of lanes = 4

DES CSI Port 0 set to polling forwarding Rx Port [0]

DES CSI Port 1 set to polling forwarding Rx Port [1]

########## Link Test ####################################

Port 0 link good

Port 0 Status Register Values:

Port 0 Status 1 (0x4D) = 0x3

Port 0 Status 2 (0x4E) = 0x4

Port 0 Parity Error (0x55, 0x56) = 0

Port 0 CSI Status (0x7A) = 0x0

Port 0 CSI Error Count (0x78) = 0

Port 1 link good

Port 1 Status Register Values:

Port 1 Status 1 (0x4D) = 0x43

Port 1 Status 2 (0x4E) = 0x4

Port 1 Parity Error (0x55, 0x56) = 0

Port 1 CSI Status (0x7A) = 0x0

Port 1 CSI Error Count (0x78) = 0

2.5.3 V4L2 Capture Test

```
# 终端 1: 测试 video0
v4l2-ctl --device /dev/video0 --set-fmt-video=width=4096,height=1536,pixelformat=RG12 --verbose

# 应成功设置格式为 4096x1536 RAW12

v4l2-ctl -d /dev/video0 --stream-mmap

# 预期输出(当雷达传输时):
# <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
# (重复模式表示正在捕获帧)
# 使用 Ctrl+C 停止
```

2.6 Troubleshooting Recommendations During System Operation

2.6.1 No Video Device After Orin Boot

- Verify that the kernel driver is loaded: lsmod | grep ov5693
- Check kernel logs: dmesg | grep -i ov5693

This issue typically arises when the driver fails to load or the device tree overlay is not applied. Figure 2 and Figure 3 provide examples of driver verification.

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

Figure 2-3. lsmod | grep ov5693 Example

```
[   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
```

Figure 2-4. dmesg | grep -i ov5693 Example

2.6.2 SerDes Link Check Reports Errors

- Verify cable connections.
- Check FPD-Link port register values (as shown in Table 2). When errors are detected during initial inspection, re-read the registers, as most error registers are clear-on-read. Errors observed on the first read can be ignored.
- Ensure all components are receiving proper power supply.

2.6.3 V4L2 Capture Shows No Data

- Verify that the SerDes link is healthy (check status registers per Table 2).
- Confirm that the AWR2188 is transmitting (check radar status LED, if available).
- Check deserializer output count registers, e.g., 0x90, 0x91 (frame count), 0x95, 0x96 (line count).
- Run dmesg on the Orin and check CSI driver logs. Timeout errors may appear when no data is received for an extended period; frame drop errors may occur when the received data format does not match the expected configuration.
- Verify that the CSI clock is present and at the correct frequency.

3 Summary

This document provides a comprehensive system integration guide for connecting TI's AWR2188 radar MMIC to the NVIDIA Jetson AGX Orin platform via FPD-Link SerDes. The system enables real-time radar data capture at high frame rates and high resolution, establishing a robust foundation for advanced radar signal processing and AI-based applications.

4 References

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

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you fully indemnify TI and its representatives against any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale, TI’s General Quality Guidelines, or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

TI objects to and rejects any additional or different terms you may propose.

Last updated 10/2025