D3 Engineering

Define | Design | Deploy

DesignCore®
RVP-TDA4Vx Development Kit
Quick Start Guide

Document No. 00C-255
Version 2.0
Release Date: 2/18/21

Table of Contents

Table of Figures

1.0 Introduction

The DesignCore™ RVP-TDA4Vx Development Kit from D3 Engineering provides qualified developers with a fully functioning evaluation platform for on-vehicle testing and development of applications, for use primarily in the automotive, transportation, robotic, and material-handling industries.

Built on the **DesignCore® Rugged Vision Platform (RVP) Reference Design with Texas Instruments TDA4VM Automotive Vision Processor**, the RVP-TDA4Vx Development Kit comes ready to use out of the box and includes processor module, cables, software, and tools.

The RVP-TDA4Vx Development Kit's base support package also provides optimized and verified hardware and software to use as a known-good launch point for design and development.

In addition, this Development Kit includes a single-use sublicense for all Texas Instruments and D3 Engineering firmware and application code.

2.0 Getting to Know Your RVP-TDA4Vx Development Kit

The RVP-TDA4Vx Development Kit comes in a rugged aluminum enclosure, ideal for in-vehicle testing and evaluation of ADAS applications. Video inputs VIN1 through VIN8 use the FPD-Link III SerDes system to communicate with external FPD-Link III enabled cameras. These ports use FAKRA connectors; to purchase cables, contact D3 Engineering.

External power and additional I/O for the system are available through the Aux I/O and Main Connectors.

VOUT provides an FPD-Link III high-speed data video output; for more information about the VOUT port, contact D3.

For detailed connector pinouts and additional hardware information, see **D3 Engineering Document No. 00C-254 DesignCore® RVP-TDA4Vx Reference Design Technical Reference Manual,** provided with shipment. For updated version, contact D3 Engineering.

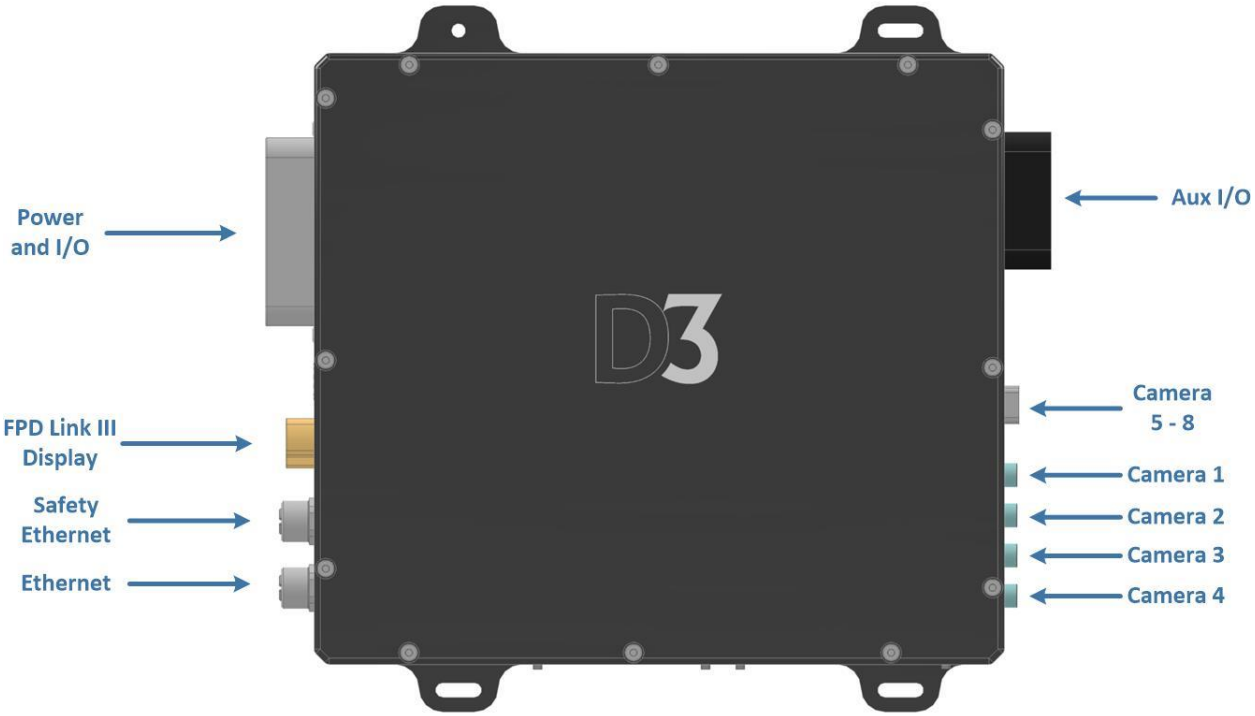


Figure 1. RVP-TDA4Vx, Top View

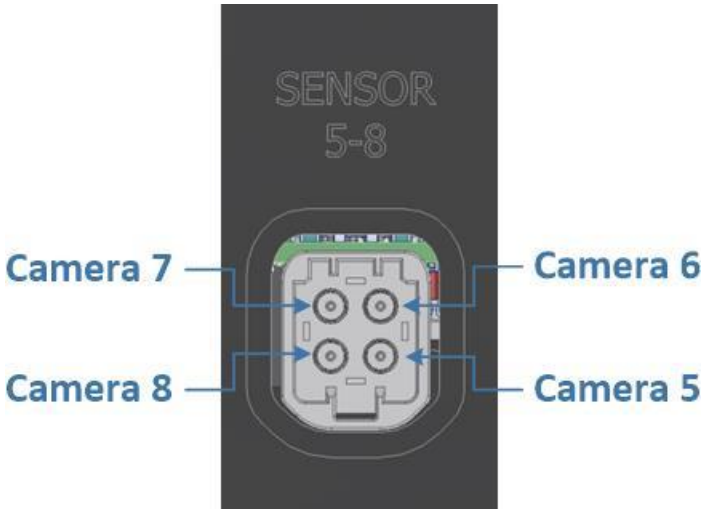


Figure 2. Sensors for Cameras 5-8

Removing the labeled access panel on the side of the device reveals the RVP-TDA4Vx's remaining ports. To run your first demo, you will need the micro-USB serial port, mini DisplayPort port, and micro-SD card slot. **See Figure 3**, below.

To debug the TDA4Vx processor, you can use the built-in XDS110 USB port (USB Console).

Figure 3. Access panel with mini DisplayPort, USB C, USB Console, PCIe, SD Card, and JTAG connectors visible

3.0 Configuring Your PC

Windows

D3 Engineering recommends using a Linux PC to develop with this platform; however, you can interact with the RVP-TDA4Vx via a console on a Windows PC.

You will need to install drivers and set up a serial console on your PC to interface with the RVP-TDA4Vx Development Kit. The RVP-TDA4Vx uses the FT232 USB to Serial bridge from FTDI, which should

automatically install upon plugging in the USB port to your computer. If installation does not occur, download the drivers from the FTDI Website.

Use any serial interface program to communicate with the RVP-TDA4Vx. Configure your serial terminal program as shown in **Figure 4** (below). Verify a baud rate of 115200, 8 data bits, 1 stop bit, no parity bit, and no flow control.

The COM port will vary by machine. The RVP-TDA4Vx will enumerate two separate COM ports; use the one with the lower number to connect (as shown in **Figure 4**, below, this is COM6).

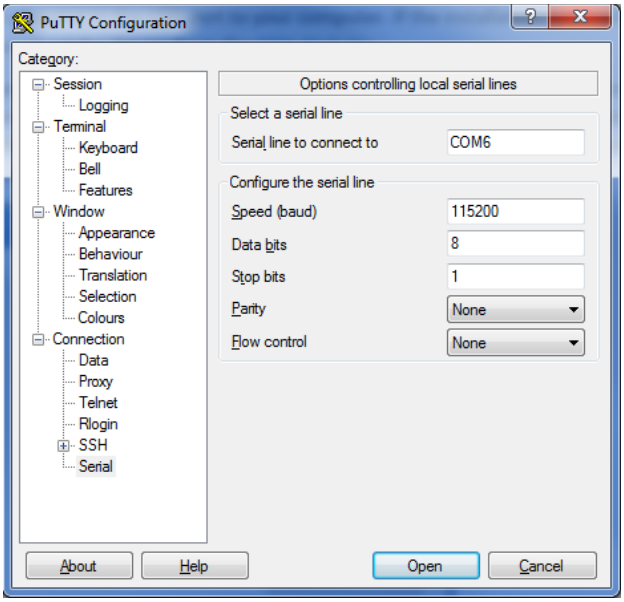


Figure 4. Windows PuTTY serial settings for communicating with the RVP-TDA4Vx

Linux

If you're running Linux on the PC you're using to communicate with the RVP-TDA4Vx Development Kit, make sure that the appropriate serial-over-USB driver is installed (this comes pre-installed on most Ubuntu variants), as well as a suitable serial terminal program such as Minicom or GtkTerm.

The RVP-TDA4Vx will show four USB serial devices as **/dev/ttyUSBx**. Choose the lower number to connect. A sample configuration for Minicom appears in **Figure 5** (below); note that this sample uses **/dev/ttyUSB0** .

Figure 5. Minicom serial settings for communicating with the RVP-TDA4Vx

4.0 Initial Setup

1. The RVP-TDA4Vx ships with a cover over the access panel housing the USB Serial, mini Display Port, PCIe Expansion, SD card and JTAG connectors. Use a small Phillips screwdriver to remove the cover.
2. Using FPD-Link III cables, connect up to eight cameras to the VIN ports **starting with VIN1**. You will need a Quad breakout cable – D3 PN: 1001302– to access the upper four camera ports; to order one, visit D3engineering.com or contact your sales representative.
3. Use the provided mini DisplayPort cable to connect the RVP-TDA4Vx to your external monitor. If the monitor only has HDMI input, an active mini display port to HDMI adapter can be used. Make sure that it is an active converter. D3 has tested successfully the 'Cable Matters Active DisplayPort to HDMI Adapter Supporting Eyefinity Technology 4K Resolution'.
4. Use the provided USB cable to connect your computer to the TDA4Vx Console 2 port. Check that the device enumerates on your computer as a COM port. If multiple new COM ports appear, use the one with the lower number for the next steps.
5. Insert the provided micro-SD card, pre-programmed with application software, in the access panel slot. If you need to create your own, see **Section 6.0 Creating an SD Card**.
6. Connect the power and I/O cable to the RVP-TDA4Vx port Main Conn.

5.0 Running Your First Demo

The RVP-TDA4Vx comes with the necessary software to get up and running quickly. After completing the steps in Section 4.0 Initial Setup:

1. Apply power to the system and verify the following:
   - Boot text appears in the serial console window.
   - A three-second countdown to system boot appears, after which the device continues on to boot.
   - When successfully booted, the screen will look like **Figure 6** (below). You are now ready to use the demo software.
   -

Figure 6. Login screen and Vision SDK demo startup

2. Log into the system by typing **root** and pressing **Enter .**
3. Navigate to the vision_apps directory and start the multi-camera demo:

```
# cd /opt/vision_apps
# ./vision_apps_init.sh
# ./run_app_multi_cam.sh
```

4. Startup text for the multi-camera capture application will appear, followed in a few seconds by the camera selection menu; see **Figure 7** (below).

```
root@j7-evm:/opt/vision_apps# ./run_app_multi_cam.sh
APP: Init ... !!!
MEM: Init ... !!!
MEM: Initialized DMA HEAP (fd=4) !!!
MEM: Init ... Done !!!
IPC: Init ... !!!
IPC: Init ... Done !!!
REMOTE_SERVICE: Init ... !!!
REMOTE_SERVICE: Init ... Done !!!
APP: Init ... Done !!!
   607.913336 s:  VX_ZONE_INIT:Enabled
   607.913352 s:  VX_ZONE_ERROR:Enabled
   607.913365 s:  VX_ZONE_WARNING:Enabled
   607.913945 s:  VX_ZONE_INIT:[tivxInit:71] Initialization Done !!!
   607.914118 s:  VX_ZONE_INIT:[tivxHostInit:48] Initialization Done for HOST
 !!!
   607.917247 s: ISS: Enumerating sensors ... !!!
   607.917655 s: ISS: Enumerating sensors ... found 0 : IMX390-UB953_D3
   607.917661 s: ISS: Enumerating sensors ... found 1 : AR0233-UB953_MARS
   607.917666 s: ISS: Enumerating sensors ... found 2 : AR0820-MAX9295
   607.917671 s: ISS: Enumerating sensors ... found 3 : UB9xxx_RAW12_TESTPATT
ERN
   607.917675 s: ISS: Enumerating sensors ... found 4 : UB96x_UYVY_TESTPATTER
N
   607.917679 s: ISS: Enumerating sensors ... found 5 : GW_AR0233_UYVY
6 sensor(s) found
Supported sensor list:
a : IMX390-UB953_D3
b : AR0233-UB953_MARS
c : AR0820-MAX9295
d : UB9xxx_RAW12_TESTPATTERN
e : UB96x_UYVY_TESTPATTERN
f : GW_AR0233_UYVY
Select a sensor
[MCU2_0]    607.917353 s: ImageSensor_RemoteServiceHandler: IM_SENSOR_CMD_CRE
ATE
[MCU2_0]    607.917423 s: Sensor at index 0 = IMX390-UB953_D3
[MCU2_0]    607.917459 s: Sensor at index 1 = AR0233-UB953_MARS
[MCU2_0]    607.917490 s: Sensor at index 2 = AR0820-MAX9295
[MCU2_0]    607.917516 s: Sensor at index 3 = UB9xxx_RAW12_TESTPATTERN
[MCU2_0]    607.917546 s: Sensor at index 4 = UB96x_UYVY_TESTPATTERN
[MCU2_0]    607.917573 s: Sensor at index 5 = GW_AR0233_UYVY
```

Figure 7. Vision SDK menu and demo options

5. Select the appropriate camera from the menu; for the IMX390 D3 Rugged Camera Module, select option **a** and press **Enter .**

8

6. Next select Whether Lens Distortion Correction (LDC) is enabled by selection 1 (Enable) or 0 (Disable) as shown in **Figure 9** (below).

```
Querying IMX390-UB953_D3
   884.570249 s: ISS: Querying sensor [IMX390-UB953_D3] ... !!!
   884.570458 s: ISS: Querying sensor [IMX390-UB953_D3] ... Done !!!
Enable LDC? Yes:1 No: 0
Invalid selection
. Try again
Enable LDC? Yes:1 No: 0
[MCU2_1]    942.003555 s: ImageSensor_RemoteServiceHandler: IM_SENSOR_CMD_QUERY
[MCU2_1]    942.003604 s: Received Query for IMX390-UB953_D3
```

Figure 9. LDC enable/disable menu

7. Select the number of cameras you are using.

   Note: *Cameras be plugged into the lowest-number ports. If you have two cameras, for example, use VIN1 and VIN2.*

```
Max number of cameras supported by sensor IMX390-UB953_D3 = 8
Please enter number of cameras to be enabled
Invalid selection
. Try again
Max number of cameras supported by sensor IMX390-UB953_D3 = 8
Please enter number of cameras to be enabled
```

Figure 10. Number of cameras

8. The application will now start. After a few seconds, video will appear on the connected DisplayPort Monitor, followed by the application menu.

```
==========================
Demo : Camera Demo
==========================

s: Save CSIx, VISS and LDC outputs

p: Print performance statistics

x: Exit

Enter Choice:
```

Figure 11. Multi Camera Demo menu

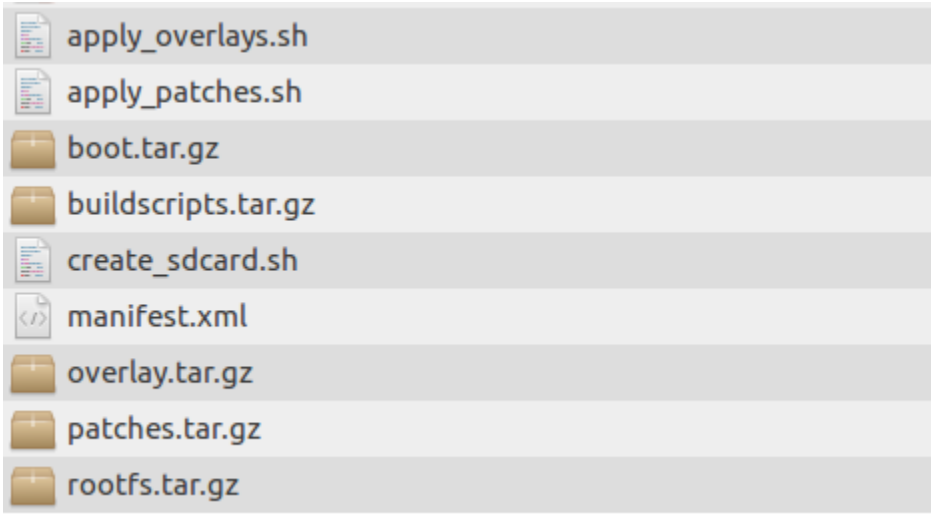
9. Selecting option **p** will print to screen various performance and framerate statistics of the system; selecting **s** will save one raw image from each channel to the SD card.

6.0 Creating an SD Card from Prebuilt Binaries

To create a new SD card with prebuilt binaries, follow the steps below:

Note: *These steps will work ONLY on a Linux PC.*

1. Unzip the release to find the following files:



2. Plug the SD card into the Linux computer and identify the device name — typically **/dev/sdX**
3. ***Important note:*** *This process* ***REMOVES DATA*** *from the device, so* ***ENSURE YOU HAVE THE CORRECT NAME! WRONG NAME = UNINTENDED DATA LOSS!*** From the directory containing the files listed in step 1, use the following commands:

```
# chmod u+x create_sdcard.sh

# sudo ./create_sdcard.sh --device /dev/sdb --name $(whoami) --path ./
```

4. Eject the SD card. It is now ready for booting by the TDA4Vx RVP.

7.0 Building the SDK from Source

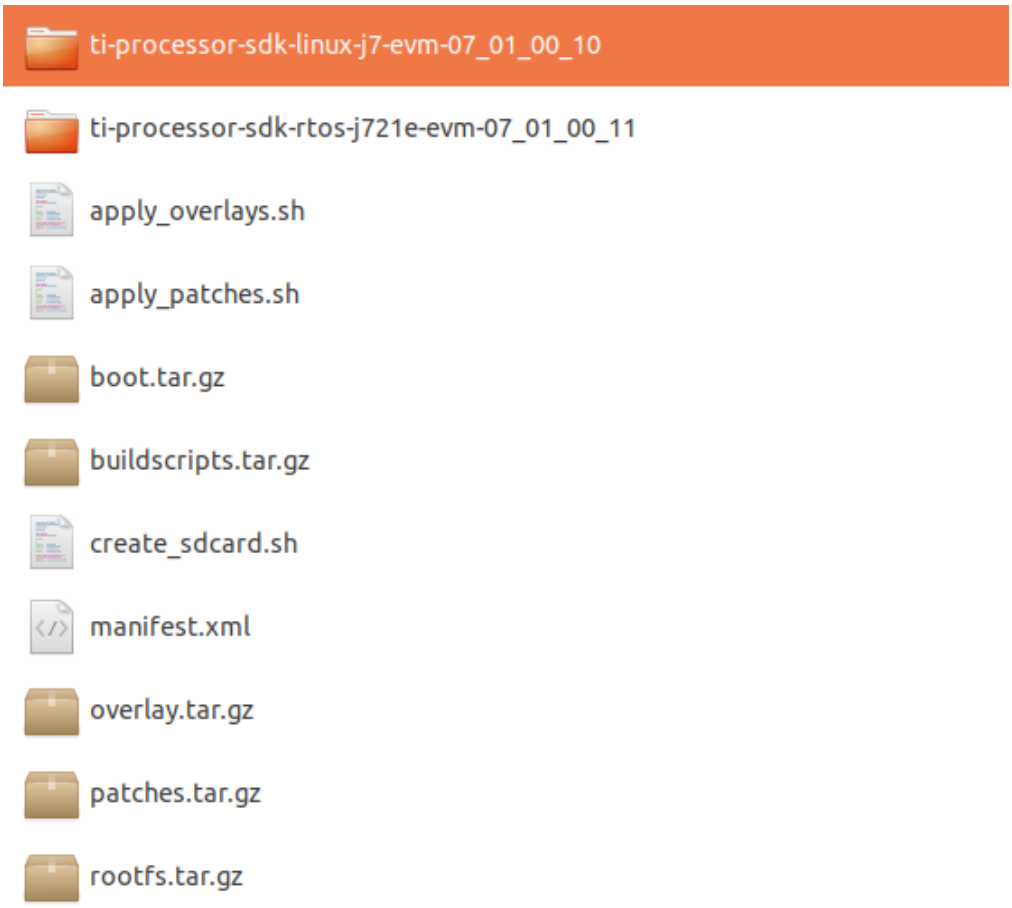
To begin development using the RVP-TDA4Vx, you will need to rebuild the SDK using the following steps:

1. Unpack the zip file from D3 Engineering.

2. Download and install the following two packages into the same folder as the D3 Engineering Release Package:

   - PSDK RTOS: https://software-dl.ti.com/jacinto7/esd/processor-sdk-rtos-jacinto7/07_02_00_06/exports/ti-processor-sdk-rtos-j721e-evm-07_02_00_06.tar.gz
   - PSDK Linux: https://software-dl.ti.com/jacinto7/esd/processor-sdk-linux-jacinto7/07_02_00_07/exports/ti-processor-sdk-linux-j7-evm-07_02_00_07-Linux-x86-Install.bin

Note: *Be sure to KEEP THE DEFAULT FOLDER NAMES for the TI components.*



3. Unpack buildscripts.tar.gz

   ```
   # tar xf buildscripts.tar.gz
   ```

4.
5. Enter the following commands to apply the provided overlays, which add D3 Engineering support to the SDK:

   # chmod u+x apply_overlays.sh

   # ./apply_overlays.sh --sdk ./

6. Run these steps to set up the Linux devkit, download compilers, and install other dependencies:

   # cd build

   # ./setup.sh -d

7. Build the SDK:

```
# ./build.sh
```

Note: *If you have not already initialized your SD card, complete that task by following the steps in* ***Section 6.0 Creating an SD card from Prebuilt Binaries.***

8. Install your newly built binaries:

```
# cd ti-processor-sdk-rtos-j721e-evm-07_02_00_06/vision_apps
# make linux_fs_install_sd
```

If you get access errors during this procedure, change the owner and group ids of the sd card content by using the command chown as follow:

```
sudo chown -R <user_id>:<group_id> /media/<user_id>/rootfs
```

Where <user_id> and <group_id> are your login id and group id on the Linux desktop.

Note*: For future incremental builds, D3 recommends building from the folder:* ***ti-processor-sdk-rtos-j721e-evm-07_02_00_06/vision_apps****.*

8.0 Additional Information

Additional software information from TI can be found inside the software release at **ti-processor-sdk-rtos-j721e-evm-07_02_00_06/vision_apps /docs/user_guide** .

For additional camera and cables, contact D3Engineering.

9.0 Reference Documentation

Document	Description	Location
DesignCore® RVP-TDA4Vx Reference Design Technical Reference Manual	Complete Technical Reference Manual for RVP-TDA4Vx Development Kit	Documents/
Vision Apps User Guide	Texas Instruments Vision Apps Documentation	ti-processor-sdk-rtos-j721e-evm-07_02_00_06/vision_apps/docs/user_guide/000__main__page_8md.html