

---

## TCP3D Driver

---

# Release Notes

Applies to Product Release: 02.01.00.02  
Publication Date: September 12, 2014



---

Texas Instruments, Incorporated  
20450 Century Boulevard  
Germantown, MD 20874 USA

---

VP00102-Form-1  
Revision D

# Contents

---

- Overview.....1
- LLD Dependencies .....1
- New/Updated Features and Quality .....1
- Resolved Incident Reports (IR) .....3
- Known Issues/Limitations.....3
- Licensing .....3
- Delivery Package .....3
- Installation Instructions.....3
- Test and Example .....4
  - TCP3D Example .....5
  - TCP3D Test .....5
- Customer Documentation List.....6

# TCP3D Driver version 02.01.00.02

## Overview

This document provides the release information for the latest TCP3D driver which should be used by drivers and application that interface with TCP3D IP.

TCP3D Driver module includes:

- Compiled library (Big and Little) Endian of TCP3D Driver.
- Source code.
- API reference guide
- Design Documentation
- Release Notes

## LLD Dependencies

LLD is dependent on following external components delivered in PDK package:

- CSL
- EDMA3 LLD

## New/Updated Features and Quality

This is an **engineering release**, tested by the development team. New and updated features are in reference to version TCP3D Driver 01.00.00.

### Release 2.01.00.02:

- Addressing IRs SDOCM00112706, SDOCM00112751, and SDOCM00105727 that are outlined in Table 4.

### Release 2.01.00.01:

- Addressing IR in Table 1 below.

**Table 1 Resolved IRs for Release 2.01.00.01**

IR Parent/ Child Number	Severity Level	IR Description
SDOCM00107359	S2 – Major	[Wireless LLD] Need to build and test the wireless LLD to enable the call stack trace

**Release 2.01.00.00:**

- Addition of K2L directories.
- Bug fixes as shown in Table 2.

**Table 2 Resolved IRs for Release 2.01.00.00**

IR Parent/ Child Number	Severity Level	IR Description
SDOCM00106473 (Parent SDOCM00106472)	S2 - Major	Critical Bug in TCP3d driver. Race condition found for incorrect programming of LINK and CHAIN fields for a previous code block, when polling is used
SDOCM00108178 (Parent SDOCM00108177)	S1-Minor	TCP3D: fix coverity issue for logical dead code (choosing between ping/pong buffer)

**Release 2.00.00.03:**

- Bug fixes as shown in Table 3.

**Table 3 Resolved IRs for Release 2.00.00.03**

IR Parent/ Child Number	Severity Level	IR Description
SDOCM00098324	S2 - Major	TCP3D driver compilation issues for non-KeyStone2 platforms on latest code base
SDOCM00093432	S3 - Minor	(Child) TCP3D LLD: osal defines memcpy and so on by itself
SDOCM00099259	S2 - Major	sync TCP3D keystone-II updates between Clearcase and GIT

**Release 2.00.00.02:**

- Renamed the device specific folders tci6634 to k2k as per new naming conventions.
- Support for TCI6636K2H device (k2h).

**Release 2.00.00.01:**

- Updates to work with auto generated CSL device file.

**Release 2.00.00.00:**

- One library to support multiple devices and updated RTSC scripts accordingly.
- Tested with new EDMA, BIOS & XDC tools which has support for C6634 device and updated CPINTC.
- Test and Example projects are updated and tested on core0. Currently test code supports only first 4 cores.

### **Release 2.00.00.1000:**

- Driver directory structure modified to support Keystone-II platforms to support multiple devices. This release supports C6634 device only.
- Tested with new EDMA, BIOS & XDC tools which has support for C6634 device.
- Limited testing is done for the projects in this release.
- Release notes document is added to the release.

### **Resolved Incident Reports (IR)**

Table 4 provides information on IR resolutions incorporated into this release.

**Table 4 Resolved IRs for this Release**

<b>IR Parent/ Child Number</b>	<b>Severity Level</b>	<b>IR Description</b>
SDOCM00112706	S1-Critical	TCP3D LLD not built for K2K, K2H platforms
SDOCM00112751	S3-Minor	TCP3D LLD: Check for number of instances is not correct
SDOCM00112818	S3-Minor	TCP3D LLD: Incorrect chain to next code block notification link for PONG side exists
SDOCM00105727	S2 - Major	Missing the extern "C" construct in tcp3d_drv.h

### **Known Issues/Limitations**

**Table 5 Known Issue IRs for this Release**

<b>IR Parent/ Child Number</b>	<b>Severity Level</b>	<b>IR Description</b>

### **Licensing**

Please refer to the software Manifest document for the details.

### **Delivery Package**

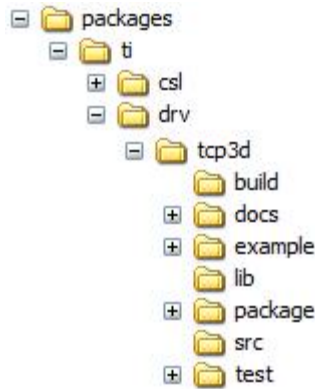
There is no separate delivery package. The TCP3D Driver is being delivered as part of PDK.

### **Installation Instructions**

The LLD is currently bundled as part of Platform Development Kit (PDK). Refer installation instruction to the release notes provided for PDK.

### **Directory structure**

The following is the directory structure after the TCP3D driver package has been installed:



The following table explains the contents of the TCP3D package:-

Directory Name	Description
ti/drv/tcp3d	The top level directory contains the following:- <ol style="list-style-type: none"> <li><u>XDC Build and Package files</u> These files (<code>config.bld</code>, <code>package.xdc</code> etc) are the XDC build files which are used to create the TCP3D package.</li> <li><u>Exported Driver header file</u> Header files which are provided by the TCP3D driver and should be used by the application developers for driver customization and usage.</li> <li><u>Makefile file</u> The file “<code>makefile</code>” is used to rebuild the libraries using standard make procedures.</li> </ol>
ti/drv/tcp3d/build	The directory contains internal XDC build related files which are used to create the TCP3D Driver package.
ti/drv/tcp3d/docs	The directory contains the TCP3D driver documentation.
ti/drv/tcp3d/example	The “ <code>example</code> ” directory in the TCP3D driver has a usage example which explains how the TCP3D driver API’s are used to send code blocks for decoding and receive decoded bits using EDMA.
ti/drv/tcp3d/lib	The “ <code>lib</code> ” folder has pre-built Big and Little Endian libraries for the TCP3D driver along with their <u>code/data size information</u> .
ti/drv/tcp3d/package	Internal TCP3D driver package files.
ti/drv/tcp3d/src	Source code for the TCP3D Driver. It may contain private header files which should not be used by application developers.
ti/drv/tcp3d/test	The “ <code>test</code> ” directory in the TCP3D driver may contain test code used for development and doing exhaustive unit testing.

## Test and Example

The section documents information about the test and example code located in the TCP3D driver.

## TCP3D Example

Example project is provided to test the pre-built libraries which are provided by TCP3D driver and to ensure the libraries are validated for all three modes supported by driver.

- WiMax dual MAP
- LTE dual MAP
- HSUPA+ split MAP

Please refer the PDK documentation for project creation procedure.

The example project depends on the pre-generated test vectors packaged with the TCP3D release.

### Running the example application:

1. In this method, ensure that the entire “example\testvectors” folder is placed two levels higher than the unit example project .out file location. For example, if the unit example project out file is located under “C:\MyPDKWorkspace\tcp3dExampleProject\Debug” folder, then please copy the folder “example\testvectors” from the PDK package to “C:\MyPDKWorkspace”
2. Ensure that the compile flag USE\_PDK\_INSTALL\_BASE is **undefined** in the project “tcp3dExampleProject” before building.
3. Build the project.
4. Launch the debug session to load and run the out file.

## TCP3D Test

Test project uses the TCP3D driver source files and is intended to test all the modes supported by driver for various combinations:

- 1) Mix of input parameters like code block sizes and optional outputs (status registers and soft decisions).
- 2) Randomly generated input configuration parameters using Matlab script focused on stress testing the driver APIs.
- 3) It also validates the notification mechanism and the TCP3D\_DRV\_INPUT\_LIST\_FULL case.
- 4) Attempts to test most of the driver run-time API with many possible commands and also the utility APIs.
- 5) Test the supported modes
  - a. WiMax dual MAP
  - b. LTE dual MAP
  - c. HSUPA+ split MAP (3GPP)

Please refer the PDK documentation for project creation procedure.

Running the TCP3D test project requires the input test vectors to be generated. Multiple unit test configurations for testing different modes are provided in the sub-folders in the

“test\gen\_test\_vectors” folder in the PDK package. This folder also has pre-built executable for test vector generation using these configurations.

**Running the test application:**

1. In this method, ensure that the entire “test\gen\_test\_vectors” folder is placed two levels higher than the unit test project .out file location. For example, if the unit test project out file is located under “C:\MyPDKWorkspace\tcp3dTestProject\Debug” folder, then please copy the folder “test\gen\_test\_vectors” from the PDK package to “C:\MyPDKWorkspace”
2. Next, execute the batch script “gen\_test\_vectors\genTestVect.bat” from the copied location to generate all the necessary test vector files.
3. Ensure that the compile flag USE\_PDK\_INSTALL\_BASE is **undefined** in the project “tcp3dTestProject” before building.
4. Build the project.
5. Launch the debug session to load and run the out file.

**Customer Documentation List**

Table 6 lists the documents that are accessible through the /docs folder on the product installation CD or in the delivery package.

**Table 6 Product Documentation included with this Release**

Document #	Document Title	File Name
1	API documentation (generated by Doxygen)	docs/ TCP3D_DRV_APIIF.chm
2	Design Document	docs/ TCP3D_DriverSDS.pdf
3	Software Manifest document	docs/ TCP3D_LLD_SoftwareManifest.pdf