Application Note

Jacinto7 HS Device Customer Return Process

Biao Li, Linjun Meng, and Yong Zhang

ABSTRACT

At present, hundreds of millions of cars equipped with TI ADAS processors are driving on the road, and with the mass production of TI's latest generation of Jacinto 7 automotive processors on the market, there are competitive device in every market segment, such as DRA8x/TDA4x series processors. While customer satisfaction is very important to Texas Instruments (TI), customer returns are handled with care and urgency. To ensure that customer concerns are addressed in a timely manner, TI has established a customer returns process for customers wishing to return devices.

Compared with the return of other devices, Jacinto7 series chips in the Customer Return Process (CRP), due to the unlocking of the High Security (HS) device, etc. Increase the complexity of the CRP process and prolong the time of the entire process. This application note introduces in detail the additional unlocking operations necessary for the Jacinto7 series HS device return process, which simplifies the CRP to the greatest extent, speeds up the CRP cycle, and optimizes customer experience.

Table of Contents

List of Figures

List of Tables

Trademarks

1 Introduction

In the stage of customer R&D and mass production, it is inevitable to encounter many hardware or software related problems, when the customer suspects that a problem is related to the internal module of the chip, customer need contact with your support window first, because TI have high confidence to our device quality, most issue high possibility related customer's hardware or software design. In most cases, TI can help you solve the problem, so there is no need to return your device.

In some special corner cases that customer really need to return customer device, it is necessary to judge whether your chip meets our return analysis standard according to TI's general CRP standard. The above is the standard process for returning TI problematic devices that should be read first before you submit your request to TI. You need to fill in and submit the corresponding CRP request form in the system. At this time, the system will automatically assign the corresponding TI FQE colleagues to follow up.

In addition to the standard return process, customer is required to assist TI in some experiments or further analysis in customer system level to locate the specific module failure, including but not limited to ABA experiments, X-ray solder joint photos, TI default software testing, signal waveform measurement, and so forth. These experiments need to be implemented case by case. Processor is so complex and difficult to analysis, so need strong cooperation from customers. This helps accelerate the process of this device analysis.

Over all, Jacinto7 devices need to follow Figure 1-1 to move on the process. TI support window will help if you have any question.

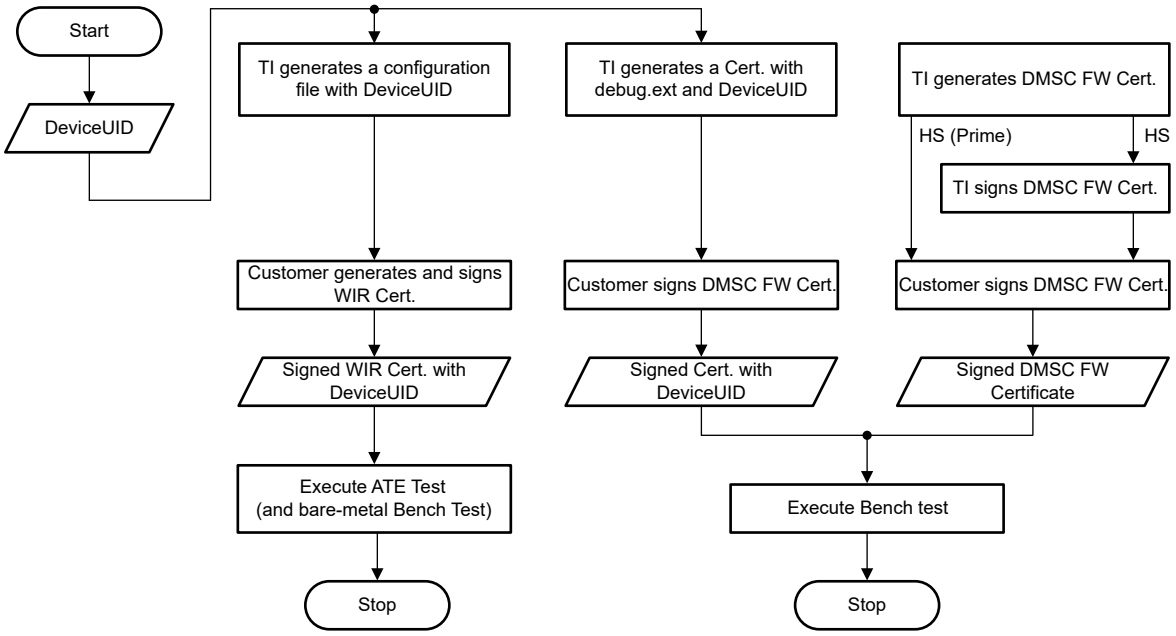

Figure 1-1. Jacinto7 Device Return Diagram

2 Device Type and Key Information Provided

You can get the part number from your device surface or search it from your order placed to TI. And then you can download the data sheet from ti.com. Take TDA4VM as an example. For more information, see the *Device and Documentation Support* section in *TDA4VM Jacinto™ Processors for ADAS and Autonomous Vehicles Silicon Revisions 1.0 and 1.1*. You can identify what type of device you have. There is lot of detail for device types. You only need to care about the three type described in Table 2-1.

Table 2-1. The JTAG State in GP/HS Silicon

Device Type	Variant	DMSC/SMS JTAG States	R5F JTAG States
General Purpose (GP)	NA	Open	Open
High Security	Security Enforced (SE)	Closed (TI)	Closed (Customer)
High Security-Prime	Security Enforced (SE)	Closed (Customer)	Closed (Customer)

For example: TDA4VM88T**G**BALFRQ1 --- General Purpose can return to TI directly.

TDA4VM88T**5**BALFRQ1 --- High Security need unlock Preparation, don't return directly.

The return process for the three device types is different. In particular, the difference between general-purpose chips and high-security chips is even greater, the return for HS device will be more complex.

2.1 Get Other Key Information via MCU UART

It has to be done to read key information such as the UID (Unique ID), DIE ID in advance before removing it from the customer board and sending it back for testing. This article recommends using the UART BOOT mode to analyze the characters printed by the MCU UART. The specific operation steps are as follows.

1. Get UID.
   a. Configure boot mode of the board to UART boot and connect second MCU UART serial port of the board to the host PC, see the EVM Setup for J721E, and power on the EVM.
   b. The terminal prints some log as shown below. You need to remove the extra CCC at the end and save as a log file. Default HS Board log as below.
   c.
```
mi02000000011a00006a3765730000000000000000485353450200010002000100002a600000100010033c74f0c863
1aa67a56d53b06f250d75cb2a9cf7a52d6eb5e21b5e824250d7e09c22d997f09dc9389ecaa3f7d2b64d3a76d6163a
a09e928ea050e1da95507e661f6002b07cd9b0b7c47d9ca8d1aae57b8e8784a12f636b2b760d7d98a18f189760dfd
0f23e2b0cb10ec7edc7c6edac3d9bdfefe0eddc3fff7fe9ad875195527df02f2a23c0ed9d5fcf6dfb3a097ee4207c
b1e2a5956e07ba144b73fe71143982CCCCCCCCCCC
```
   d. Download the python script to parse the log in step 2.
   e. Use the following command to parse the log after getting the above two files. The parsed information is as shown:

```
@Ubuntu18:~/Documents/summary/parse_uart_log$ python 7080.uart_boot_socid.py
default_uart_hs.log
----------------------
SoC ID Header Info:
----------------------
NumBlocks            : [2]
----------------------
SoC ID Public ROM Info:
----------------------
SubBlockId           :
SubBlockSize         :
DeviceName           : j7es
DeviceType           : HSSE
DMSC ROM Version     : [0, 1, 0, 2]
R5 ROM Version       : [0, 1, 0, 2]
----------------------
SoC ID Secure ROM Info:
----------------------
Sec SubBlockId       : 2
Sec SubBlockSize     : 166
Sec Prime            : 0
Sec Key Revision     : 1
```

```
Sec Key Count        : 1
Sec TI MPK Hash      :
33c74f0c8631aa67a56d53b06f250d75cb2a9cf7a52d6eb5e21b5e824250d7e09c22d997f09dc9389ecaa3f7d2b64
d3a76d6163aa09e928ea050e1da95507e66
Sec Cust MPK Hash    :
1f6002b07cd9b0b7c47d9ca8d1aae57b8e8784a12f636b2b760d7d98a18f189760dfd0f23e2b0cb10ec7edc7c6eda
c3d9bdfefe0eddc3fff7fe9ad875195527d
Sec Unique ID        : f02f2a23c0ed9d5fcf6dfb3a097ee4207cb1e2a5956e07ba144b73fe71143982
```

2. Get DIE ID.

   This document recommends that after entering the Linux of the customer board, type in the command line and use the following command line to read the DIE ID.

```
echo `devmem2 0x43000020 w | tail -n1`
echo `devmem2 0x43000024 w | tail -n1`
echo `devmem2 0x43000028 w | tail -n1`
echo `devmem2 0x4300002c w | tail -n1`
```

   After you get all the key information list above, you need share it to your support window. This will help the subsequent process for HS device return. If your device is a General Purpose (GP) type, provide this key information to your TI support window, then you can return this device directly without the files below for HS device return. Otherwise, you need to follow the steps in Section 3 to generate more binaries for TI can further test your device.

3 HS Device Return

HS device returns are more complicated as the JTAG interface is closed and TI cannot have access to and do more testing for this. TI needs customer help to generate wait in reset (WIR) Cert, SBL/SPL Cert, and DMSC FW. Those binaries will help TI unlock the JTAG and log in to the HLOS continue to follow-up process. The return test process of Jacinto7 series device is shown in Figure 3-1. HS credentials handshake is mandatory to enable HS device test.

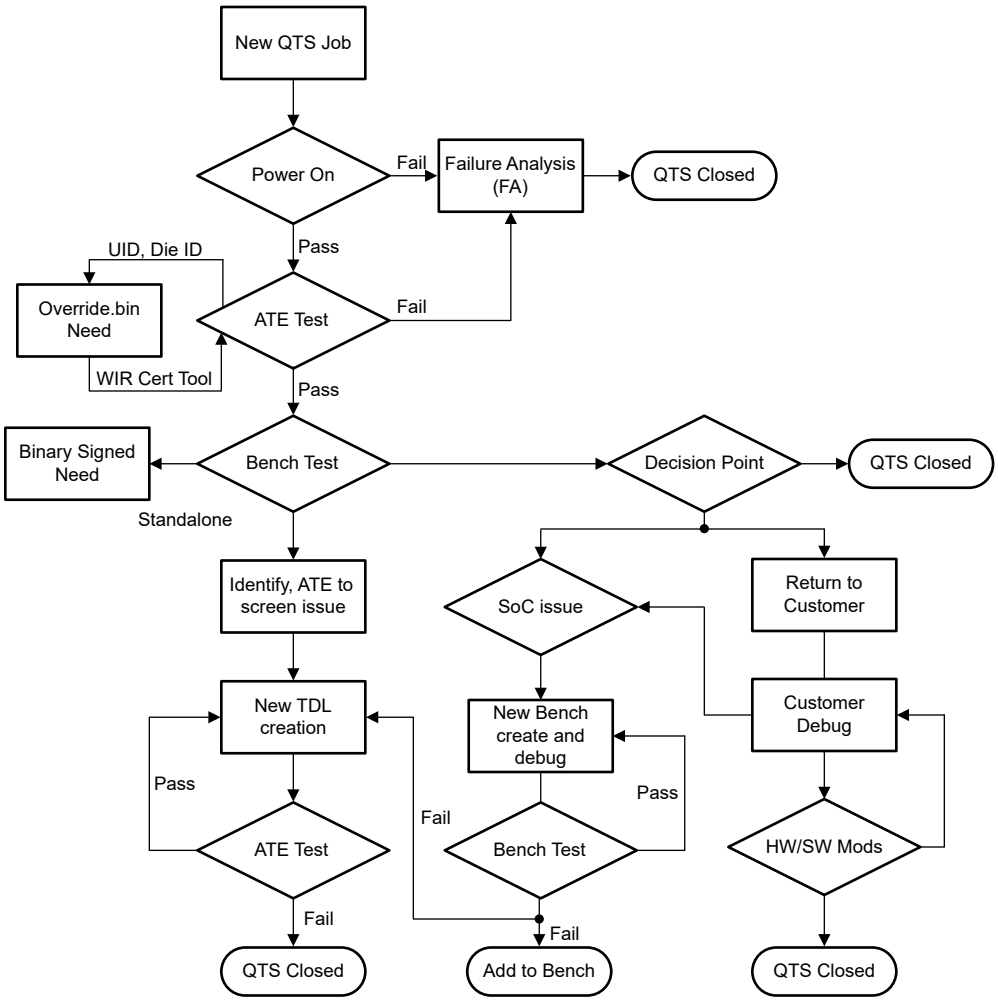

Figure 3-1. Customer Return Unit Test Flow in TI

Customer signed WIR certificate required to enable both: ATE Testing; Bench Testing. Customers can use allow_wildcard_unlock = 0x5A in the Boardcfg file in the development stage to skip the UID verification, but in the mass production stage, this parameter must be set to 0 to enable UID verification. CRP is mainly aimed at mass production projects. The corresponding firmware provided by TI (allow_wildcard_unlock = 0x0) is to enable UID verification. After the customer signs the firmware, the UID needs to be verified before the firmware enters the system. Only after passing the verification can it succeed Start the system. Therefore, this firmware can only be used to unlock the device that you have provided the UID to further protect the customer's information security. Device UID is unique per device. Certificates enables debug on only one device. This involves exchange of information between Customer and TI for each returned device. Secondary App debug requires additional signed images. Customer must sign TI bootloader and firmware images with device root key (Private Key).

After the New QTS Job be created, the CRP cannot start until TI:

- Receives device (customer removes from board and ships)
- Re-balls device

- Extracts Device UID and creates certificate inputs and sends to customer contact
- Receives customer signed certificates

3.1 CRP Script Tool

In order to standardize and simplify the process of customers providing binary, TI can provide the CRP Script Tools. Currently this tool reference tar only support **TDA4VM**, more device need chose your SDK path to generate the binary signed. The running logic of this CRP script is shown in Figure 3-2.

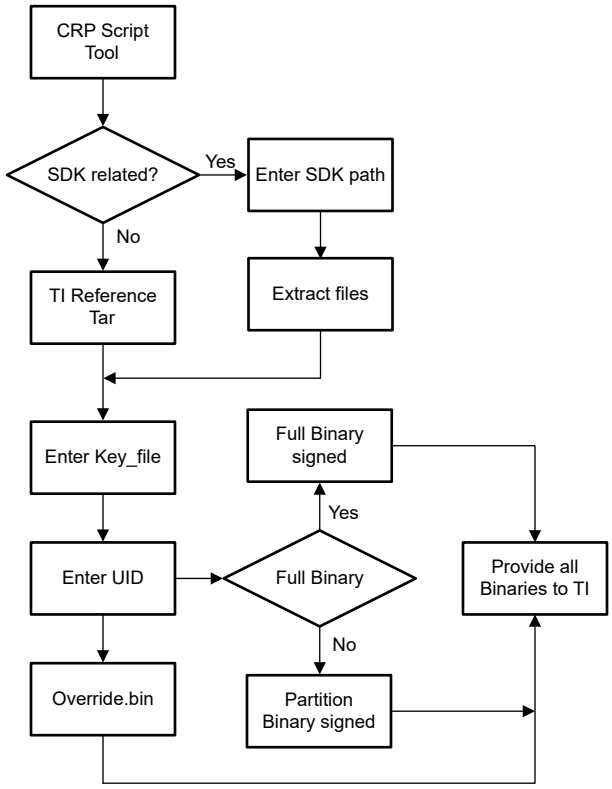

Figure 3-2. TI CRP Script Tool Sign and Generate Flow

After set up the environment for generate the binary, TI will provide a reference environment tar (base on TDA4VM SDK8.6), customer only need enter the KEY_FILE path (this means the customer needs to be able to access the private key), Device UID, and the output path, the tool will help sign and generate all the binaries for TI CRP need. Include the override.bin, SBL/SPL Certificate binary, and the DMSC Certificate binary. The tool can choose that other binaries except tiboot3.bin and tifs.bin or sysfw.itb need to be signed or not. And for the difference sign process of the HS device and HS-Prime device, tool can also handle this.

Use this tool to generate all the files that need to be signed in the CRP process at one time, and modify the parameters to configure the number of files generated by the tool, and the tool is still being improved. The specific steps to use the tool are as follows:

1. Download and extract the j7_crp_tool.zip tar to your ubuntu PC.
2. Go to the path of this tool installed, and execute the script tool use the command: **./ j7_crp_spl_tool.sh**.
3. Enter your private key path: KEY_FILE path, the tool provide the TI dummy key in this tool for example as below: {cwd}/j7_crp_spl_too_for_reference/core-secdev-k3/keys/custMpk.pem.
4. You need to enter the device type (hs/hsp). The tool will check to see whether the key file exists and exit immediately if it does not exists.
5. You need enter the Device UID and the output path for the signed binaries.
6. You can find the all binaries generated by this tool in the output path, and you need to pack them into one file (tar) and send to TI contact window.

3.2 Generate and Signs the WIR Certificate Binary (override.bin) Standalone

This is mandatory for all Jacinto7 HS device return ATE test. WIR certificate is used to unlock the JTAG debug port in the ATE test stage. After the JTAG is unlocked, ATE tests can be move on, and some basic bare mental test programs can also be run. But this test can only run single-module test on a specific failure module. If the above tests are all passed, it is necessary to enter the high-level operating system (HLOS) to perform bench system-level tests, and conduct targeted system-level tests according to customer failure scenarios. To generate the WIR certificate, you need to replace the UID in bold with the following template code and save it as x509_sec_override.txt.

In the following code template, certType = INTEGER:2147483649 (0x80000001 in decimal), representing Override Cert mode. debugType = INTEGER:4 stands for DEBUG_FULL, which is used to unlock the JTAG function and enable full debugging function.

```
[ req ]
distinguished_name = req_distinguished_name
x509_extensions = v3_ca
prompt = no dirstring_type = nobmp
[ req_distinguished_name ]
C = gc
ST = cW
L = y6qqF9wh61
O = vGtcXq5gItAeCDXDyVCtdVayXh
OU = tcDeqFyxG4r
CN = rgH4qfPTF
emailAddress = lQeqF8F1HQuc2@lrIP7hPUyQ03x.com
[ v3_ca ]
basicConstraints = CA:true
1.3.6.1.4.1.294.1.1=ASN1:SEQUENCE:boot_seq
1.3.6.1.4.1.294.1.8=ASN1:SEQUENCE:debug
[ boot_seq ]
certType = INTEGER:2147483649
bootCore = INTEGER:0
bootCoreOpts = INTEGER:0
destAddr = FORMAT:HEX,OCT:00000000
imageSize = INTEGER:0
[ debug ]
debugUID = FORMAT:HEX,OCT:486227340651ed7670e840191e064dbb8d0ad5164737980ed860ebd81672b8cc
debugType = INTEGER:4
coreDbgEn = INTEGER:0
coreDbgSecEn = INTEGER:0
```

Use the following command to generate a WIR Cert file named override.bin. The custkey.pem in the following command is the root private key of the customer, and the command needs to be run in the same path.

```
$ openssl req -new -x509 -key custkey.pem -nodes -outform DER –out override.bin -config
x509_sec_override.txt -sha512
```

This has been integrated into the CPR Script tool.

3.3 Generate the Binaries for Bench Test Standalone

This is mandatory for all Jacinto7 HS device return Bench tests. TI may need to log into the HLOS system to run more tests for further analysis. More signed binary are needed to unlock the device. The reason is that for Bench testing, TI needs to enter the HLOS (like Linux) to obtain more log information, so more signed binaries is needed. The following introduction is mainly for SPL boot mode.

1. Generate signed cfg files.

   TI will prepare board-cfg.bin; sec-cfg.bin; rm-cfg.bin; pm-cfg.bin (location in /ti-processor-sdk-linux-j7-evm-xx_xx_xx_xx/board-support/k3-image-gen-2021.01a/out/soc/j721e/evm) to you, and request that you sign these files by secure-binary-image.sh. Then, return the signed image. Use the following command to generate the binary signed:

   ```
   /ti-processor-sdk-linux-j7-evm-07_03_00_05/board-support/core-secdev-k3/scripts/secure-binary-
   image.sh
   out/soc/j721e/evm/board-cfg.bin out/soc/j721e/evm/board-cfg.bin-signed
   ```

2. Generate the signed sysfw.bin-hs.

Sign the sysfw inner certificate with custMpk.pem by ./gen_x509_cert.sh, This is different between the HS and HS-Prime device. You only need to chose one way to generate.

a. For HS device:

TI will prepare the ti-fs-firmware-j721e_sr1_1-hs-enc.bin and ti-fs-firmware-j721e_sr1_1-hs-cert.bin for customer sign.

```
./gen_x509_cert.sh -d -c m3 -b /home/chris/J7/J721e/86/hs/board-support/prebuilt-images/ti-
fs-firmware-
 j721e_sr1_1-hs-cert.bin -o ti-fs-firmware-j721e_sr1_1-hs-certs.bin -l 0x40000 -k /home/
chris/J7/J721e/86/hs/board-
 support/core-secdev-k3/keys/custMpk.pem -r 1
```

You need to generate sysfw.bin-hs by cat command.

```
cat ti-fs-firmware-j721e_sr1_1-hs-certs.bin /home/chris/J7/J721e/86/hs/board-support/
prebuilt-images/ti-fs-
firmware-j721e_sr1_1-hs-enc.bin > out/soc/j721e/evm/sysfw.bin-hs
```

b. For HS-Prime device:

TI will prepare only ti-fs-firmware-j721e-hs.bin for customer sign.

```
./gen_x509_cert.sh -d -c m3 -b /home/chris/J7/J721e/86/hs/board-support/prebuilt-images/
ti-fs-firmware-j721e-hs.bin -o out/soc/j721e/evm/sysfw.bin-hs -l 0x40000 -k /home/chris/J7/
J721e/86/hs/board-support/core-secdev-k3/keys/custMpk.pem -r 1
```

Customer will only need sign this binary only. No more cat command need be executed.

3. Generate its file by script gen_its.sh and finally return the sysfw.itb.

This is mandatory for all type HS device. Use the command below to generate the its file first.

```
./gen_its.sh j721e_sr1_1 hs evm out/soc/j721e/evm/sysfw.bin-hs out/soc/j721e/evm/board-cfg.bin-
signed
out/soc/j721e/evm/pm-cfg.bin-signed out/soc/j721e/evm/rm-cfg.bin-signed out/soc/j721e/evm/sec-
cfg.bin-signed >
out/soc/j721e/evm/sysfw-j721e_sr1_1-evm.its
```

Use this command mkimage to generate sysfw-j721e_sr1_1-evm.itb and rename to sysfw.itb.

```
mkimage -f out/soc/j721e/evm/sysfw-j721e_sr1_1-evm.its -r sysfw-j721e_sr1_1-evm.itb
move out/soc/j721e/evm/sysfw-j721e_sr1_1-evm.itb out/soc/j721e/evm/sysfw.itb
```

4. Generate tiboot3.bin for SPL boot.

You need the patch below to U-boot first, and regenerate the u-boot-spl.bin. This patch skips the need to sign subsequent kernel or app files.

```
diff --git a/arch/arm/mach-k3/security.c b/arch/arm/mach-k3/security.c
index 092588f4b5..c55d1da689 100644
--- a/arch/arm/mach-k3/security.c
+++ b/arch/arm/mach-k3/security.c
@@ -53,6 +53,14 @@ void ti_secure_image_post_process(void **p_image, size_t *p_size)
     if (!image_size)
         return;
 +    if (get_device_type() == K3_DEVICE_TYPE_HS_SE &&
+        !ti_secure_cert_detected(*p_image)) {
+        printf("Warning: Did not detect image signing certificate. "
+               "Skipping authentication to prevent boot failure for CRP. "
+               "This will fail on Security Enforcing(HS-SE) devices\n");
+        return;
+    }
+
     if (get_device_type() == K3_DEVICE_TYPE_GP) {
         if (ti_secure_cert_detected(*p_image)) {
             printf("Warning: Detected image signing certificate on GP device. "
```

Use k3_gen_x509_cert.sh to generate tiboot3.bin.

```
u-boot-2021.01+gitAUTOINC+62a9e51344-g62a9e51344/tools/k3_gen_x509_cert.sh -c 16 -b s -o
tiboot3.bin -l
0x41c00000 -r 1 -k /home/chris/J7/J721e/86/l/board-support/core-secdev-k3/keys/custMpk.pem
```

You need to provide the tiboot3.bin and sysfw.bin to TI only.

4 Summary

This application note summarizes the J7 HS device customer return process, and provide CPR Script Tool for customer to standardize and simplify the process of customer signature, which maximizes the convenience of customers Exercising the Right to Return for Testing. When returning HS devices, due to some issue such as device unlocking, the cycle of the customer return process is greatly prolonged and the customer experience is affected. Currently, the CRP Script Tool is developed for TDA4VM that the most common HS version of the Jacinto7 series. And the tool for other types of device in the Jacinto7 series will be updated in the future.

5 References

1. TDA4VM Product Page
2. Texas Instruments: *DRA829/TDA4VM Technical Reference Manual*
3. TISCI User Guide
4. Texas Instruments: *Jacinto7 HS Device Development*
5. Texas Instruments: *K3 Security Hardware Architecture User Guide* (SPRUIM0C)
6. How to Check if Device Type is HS-SE, HS-FS or GP

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), 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 will 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 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.

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

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265