



Test Plan Execution Report

Test Project: VISIONSDK

Test Plan: PSDKR_Test_Plan_3_3_Functional_All_Platform

Printed by TestLink on 06/04/2018

2017 (c) Testlink Community

Table Of Contents

1.1.Mono_Cam

1.1.1.MISC

VISIONSDK-325: VSDK_restructuring_directory_structure

1.2.Radar

VISIONSDK-150: Radar_AR12_Capture_Null

VISIONSDK-152: Radar_AR12_Capture_Radar_FrameCopy_DSP1_Null

VISIONSDK-154: NullSrc_Capture_Radar_FFT_EVE1_Null_Read_Frames_SDcard

VISIONSDK-155: NullSrc_Capture_Radar_FFT_EVE1_Null_Write_Frames_SDcard

VISIONSDK-156: NullSrc_Capture_Radar_FFT_EVE1_Null_Read_Frames_NW

VISIONSDK-157: NullSrc_Capture_Radar_FFT_EVE1_Null_Write_Frames_NW

VISIONSDK-232: Radar_AR12_Capture_Radar_Object_Detect_EVE1_Null

VISIONSDK-233: Radar_AR12_Capture_Radar_Object_Detect_EVE1_Display

VISIONSDK-243: Radar_Flash_AR12_Firmware

VISIONSDK-313: Radar_AR12_Multi_Capture_Radar_FFT_EVE1_Display

VISIONSDK-314: Radar_Test_Source_Object_Detection

VISIONSDK-331: Radar_AR12_Capture_Null_Multiple_Times

1.3.Build

1.3.1.Radar_Builds

VISIONSDK-242: Radar_default_build

VISIONSDK-280: Radar_different_builds

1.4.Release_Process

VISIONSDK-245: VSDK_Radar_release_check_list

VISIONSDK-247: Radar_package_creation_and_installation

1.5.Boot_Modes

1.5.1.SD_Boot

VISIONSDK-273: Load_BIOS_Binaries_using_SD_Card

1.5.2.QSPI_Boot

VISIONSDK-274: Load_Binaries_using_QSPI

1.5.3.QSPI_SD_Boot

VISIONSDK-275: Load_Binaries_using_QSPI_SD

1.5.4.CCS_Boot

VISIONSDK-332: Load_Binaries_using_CCS

Test Project: VISIONSDK

Project: VISIONSDK Location: TII Owner: Sivasankaran, Shiju

Test Plan: PSDKR_Test_Plan_3_3_Functional_All_Platform

Radar Functional Test Plan

Will cover all functional test

1.1.Test Suite : Mono_Cam

1.1.1.Test Suite : MISC

Test Case VISIONSDK-325: VSDK_restructuring_directory_structure			
<u>Summary:</u> restructuring directory structure for VSDk 3.0 release			
<u>#:</u>	<u>Step actions:</u>	<u>Expected Results:</u>	<u>Execution Status:</u>
1	Restructure directory structure for VSDK into separate Folder as below link_fw Make System (Common for FW & all Apps modules) sample_app apps algorithms docs testsuite	Directory structure should be as stated	
<u>Execution type:</u>	Manual		
<u>Estimated exec. duration (sec):</u>			
<u>Priority:</u>	Medium		
<u>Keywords:</u>	None		
Execution Details			
Build	REL_3_3		
Tester	x0246581		
<u>Execution Result:</u>	Passed		
<u>Execution Mode:</u>	Manual		
<u>Execution duration (sec):</u>			

1.2.Test Suite : Radar

Test Case VISIONSDK-150: Radar_AR12_Capture_Null			
<u>Summary:</u>			
Radar Capture Null UC			
Input : AR12			
Output : Null			
<u>Preconditions:</u>			
Ensure AR12 sensor Radar HW is connected to TDA3x EVM			
Debug prints will be in UART2			
<u>#:</u>	<u>Step actions:</u>	<u>Expected Results:</u>	<u>Execution Status:</u>
1	Boot TDA3x with Radar setup/TDA3xx ALPS Board	Shoul display Main Menu	
2	Run "Radar (Single AR1243) Capture + Null (TDA3xx Only) usecase" UC	No Display	
3	Press "P"	Check performance stats	
<u>Execution type:</u>	Manual		
<u>Estimated exec. duration (sec):</u>			
<u>Priority:</u>	Medium		
<u>Keywords:</u>	c_regression c_qualification tda3xx-alps tda3xx-AR12-Booster		
Execution Details			
Build	REL_3_3		
Tester	x0246581		
<u>Execution Result:</u>	Passed		
<u>Execution Mode:</u>	Manual		
<u>Execution duration (sec):</u>			

Test Case VISIONSDK-152: Radar_AR12_Capture_Radar_FrameCopy_DSP1_Null			
<u>Summary:</u>			
Radar Capture Radar Frame copy on DSP1 Null UC			
Input : AR12			
Output : Null			
<u>Preconditions:</u>			
Ensure AR12 sensor Radar HW is connected to TDA3x EVM			
Debug prints will be in UART2			
<u>#:</u>	<u>Step actions:</u>	<u>Expected Results:</u>	<u>Execution Status:</u>
1	Boot TDA3x with Radar setup/TDA3xx ALPS Board	Should display Main Menu	
2	Run "Radar (Single AR1243) Capture + Radar Frame Copy (DSP1) + Null (TDA3xx Only) usecase" UC	No Display	
3	Press "P"	Check performance stats	

<u>Execution type:</u>	Manual
<u>Estimated exec. duration (sec):</u>	
<u>Priority:</u>	Medium
<u>Keywords:</u>	c_stress c_qualification c_stability tda3xx-alps tda3xx-AR12-Booster
Execution Details	
Build	REL_3_3
Tester	x0246581
<u>Execution Result:</u>	Passed
<u>Execution Mode:</u>	Manual
<u>Execution duration (sec):</u>	

Test Case VISIONSDK-154: NullSrc_Capture_Radar_FFT_EVE1_Null_Read_Frames_SDCard

Summary:

Null Source Capture(SD card) Radar FFT on EVE1 Null UC

Input : AR12

Output : Null

Preconditions:

Input files present in SD card

Debug prints will be in

UART1 for TDA2x & UART2 for TDA3x

<u>#:</u>	<u>Step actions:</u>	<u>Expected Results:</u>	<u>Execution Status:</u>
1	Boot TDA2x/TDA3x	Should display Main Menu	
2	Run "Null Source (SD/Network) Input + Radar FFT (EVE1) + Null (SD/Network)" UC Select Data Read/Write Mode as SD card	No display	
3	Press "P"	Check performance stats	

<u>Execution type:</u>	Manual
<u>Estimated exec. duration (sec):</u>	
<u>Priority:</u>	Medium
<u>Keywords:</u>	tda2xx-evm tda3xx-evm
Execution Details	
Build	REL_3_3
Tester	x0246581
<u>Execution Result:</u>	Passed
<u>Execution Mode:</u>	Manual
<u>Execution duration (sec):</u>	

Test Case VISIONSDK-155: NullSrc_Capture_Radar_FFT_EVE1_Null_Write_Frames_SDCard

Summary:

Null Source Capture(SD card) Radar FFT on EVE1 Null UC

Input : AR12

Output : Null

Preconditions:

Input files present in SD card

Debug prints will be in

UART1 for TDA2x & UART2 for TDA3x

#:	Step actions:	Expected Results:	Execution Status:
1	Boot TDA2x/TDA3x	Should display Main Menu	
2	Run "Null Source (SD/Network) Input + Radar FFT (EVE1) + Null (SD/Network)" UC Select Data Read/Write Mode as SD card	No display	
3	Select File IO menu Write single frame to SD card	Writing single frame to SD card should be successful	
<u>Execution type:</u>	Manual		
<u>Estimated exec. duration (sec):</u>			
<u>Priority:</u>	Medium		
<u>Keywords:</u>	tda2xx-evm tda3xx-evm		
Execution Details			
Build	REL_3_3		
Tester	x0246581		
<u>Execution Result:</u>	Passed		
<u>Execution Mode:</u>	Manual		
<u>Execution duration (sec):</u>			

Test Case VISIONSDK-156: NullSrc_Capture_Radar_FFT_EVE1_Null_Read_Frames_NW

Summary:

Null Source Capture(Network) Radar FFT on EVE1 Null UC

Input : AR12

Output : Null

Preconditions:

Ensure NDK is enabled in build

Input files sent through network using network_tx

Debug prints will be in

UART1 for TDA2x & UART2 for TDA3x

#:	Step actions:	Expected Results:	Execution Status:
1	Boot TDA2x/TDA3x	Should display Main Menu	
2	Run "Null Source (SD/Network) Input + Radar FFT (EVE1) + Null (SD/Network)" UC Select Data Read/Write Mode as Network	No display	
3	Press "P"	Check performance stats	
<u>Execution type:</u>	Manual		
<u>Estimated exec. duration (sec):</u>			
<u>Priority:</u>	Medium		
<u>Keywords:</u>	tda2xx-evm tda3xx-evm		
Execution Details			
Build	REL_3_3		
Tester	x0246581		
<u>Execution Result:</u>	Passed		
<u>Execution Mode:</u>	Manual		

Execution duration (sec):

Test Case VISIONSDK-157: NullSrc_Capture_Radar_FFT_EVE1_Null_Write_Frames_NWSummary:

Null Source Capture(Network) Radar FFT on EVE1 Null UC

Input : AR12

Output : Null

Preconditions:

Ensure NDK is enabled in build

Input files sent through network using network_tx

Debug prints will be in

UART1 for TDA2x & UART2 for TDA3x

#:	Step actions:	Expected Results:	Execution Status:
1	Boot TDA2x/TDA3x	Should display Main Menu	
2	Run "Null Source (SD/Network) Input + Radar FFT (EVE1) + Null (SD/Network)" UC Select Data Read/Write Mode as Network	No display	
3	Run network_rx to dump files	Should be able to dump frmaes	
<u>Execution type:</u>	Manual		
<u>Estimated exec. duration (sec):</u>			
<u>Priority:</u>	Medium		
<u>Keywords:</u>	tda2xx-evm tda3xx-evm		
Execution Details			
Build	REL_3_3		
Tester	x0246581		
<u>Execution Result:</u>	Passed		
<u>Execution Mode:</u>	Manual		
<u>Execution duration (sec):</u>			

Test Case VISIONSDK-232: Radar_AR12_Capture_Radar_Object_Detect_EVE1_NullSummary:

Radar Capture Radar Object Detect on EVE1 Null UC

Input : AR12

Output : Null

Preconditions:

Ensure AR12 sensor Radar HW is connected to TDA3x EVM

Debug prints will be in UART2

#:	Step actions:	Expected Results:	Execution Status:
1	Boot TDA3x with Radar setup/TDA3xx ALPS Board	Should display Main Menu	
2	Run "Radar (Single AR1243) Capture + Radar Object Detect (EVE1) + Null (TDA3xx Only) usecase" UC	No Display	
3	Select Normal Frame/Advanced Frame.	Depending upon selection Normal Frame/Advanced Frame should be selected	
4	Press "P"	Check performance stats	
5	Press 'c' to read back and verify parameters.	Should be able to read and verify parameters	

6	Press 'd' to dynamically change the slope.	Slope should be changed dynamically	
<u>Execution type:</u>	Manual		
<u>Estimated exec. duration (sec):</u>			
<u>Priority:</u>	Medium		
<u>Keywords:</u>	tda3xx-alps tda3xx-AR12-Booster		
Execution Details			
Build	REL_3_3		
Tester	x0246581		
<u>Execution Result:</u>	Passed		
<u>Execution Mode:</u>	Manual		
<u>Execution duration (sec):</u>			

Test Case VISIONSDK-233: Radar_AR12_Capture_Radar_Object_Detect_EVE1_DisplaySummary:

Radar Capture Radar Object Detect on EVE1 Display UC

Input : AR12

Output : HDMI

Preconditions:

Ensure AR12 sensor Radar HW is connected to TDA3x EVM

Debug prints will be in UART2

<u>#:</u>	<u>Step actions:</u>	<u>Expected Results:</u>	<u>Execution Status:</u>
1	Boot TDA3x with Radar setup	Should display Main Menu	
2	Run "Radar (Single AR1243) Capture + Radar Object Detect (EVE1) + Display (TDA3xx Only) usecase" UC	Display should come up & no buffer drops should observed	
3	Select Normal Frame/Advanced Frame.	Depending upon selection Normal Frame/Advanced Frame should be selected	
4	Press "P"	Check performance stats	
5	Press 'c' to read back and verify parameters.	Should be able to read and verify parameters	
6	Press 'd' to dynamically change the slope.	Slope should be changed dynamically	
<u>Execution type:</u>	Manual		
<u>Estimated exec. duration (sec):</u>			
<u>Priority:</u>	Medium		
<u>Keywords:</u>	c_regression c_stress c_stability tda3xx-AR12-Booster		
Execution Details			
Build	REL_3_3		
Tester	x0246581		
<u>Execution Result:</u>	Passed		
<u>Execution Mode:</u>	Manual		
<u>Execution duration (sec):</u>			

Test Case VISIONSDK-243: Radar_Flash_AR12_FirmwareSummary:

Radar AR12 Firmware Flash UC

supported on TDA3x ALPS board

Input : AR12 Firmware

Preconditions:

AR12 firmware is part of binaries

Debug prints will be in UART2

#:	Step actions:	Expected Results:	Execution Status:
1	Boot TDA3xx ALPS Board	Should display Main Menu	
2	Run "AR12 Firmware Flash (ALPS board Only)" UC	No Display	
3	Erase AR12xx Flash	Should erase previous firmware from flash	
4	Flash AR12xx Firmware	New firmware should be flashed	
<u>Execution type:</u>	Manual		
<u>Estimated exec. duration (sec):</u>			
<u>Priority:</u>	Medium		
<u>Keywords:</u>	tda3xx-alps		
Execution Details			
Build	REL_3_3		
Tester	x0246581		
<u>Execution Result:</u>	Passed		
<u>Execution Mode:</u>	Manual		
<u>Execution duration (sec):</u>			

Test Case VISIONSDK-313: Radar_AR12_Multi_Capture_Radar_FFT_EVE1_DisplaySummary:

Radar Capture Radar FFT on EVE1 Display UC

Input : AR12

Output : HDMI

Preconditions:

Ensure AR12 sensor Radar HW is connected to TDA3x EVM

Debug prints will be in UART2

#:	Step actions:	Expected Results:	Execution Status:
1	Boot TDA3x/RVP with Radar setup	Should display Main Menu	
2	Run "Radar (Single AR1243) Capture + Radar FFT (EVE1) + Display (TDA3xx Only) usecase" UC	Display should come up & no buffer drops should observed	
3	Select Normal Frame/Advanced Frame.	Depending upon selection Normal Frame/Advanced Frame should be selected	
4	Press "P"	Check performance stats	
5	Press 'c' to read back and verify parameters.	Should be able to read and verify parameters	
6	Press 'd' to dynamically change the slope.	Slope should be changed dynamically	
<u>Execution type:</u>	Manual		
<u>Estimated exec. duration (sec):</u>			
<u>Priority:</u>	Medium		
<u>Keywords:</u>	c_regression c_stress c_stability tda3xx-AR12-Booster		
Execution Details			
Build	REL_3_3		
Tester	x0246581		
<u>Execution Result:</u>	Passed		
<u>Execution Mode:</u>	Manual		
<u>Execution duration (sec):</u>			

Test Case VISIONSDK-314: Radar_Test_Source_Object_DetectionSummary:

Radar Test Source Object Detection

Input : testdata

Output : HDMI

Preconditions:

Ensure AR12 sensor Radar HW is connected to TDA3x EVM

Debug prints will be in UART2

#:	Step actions:	Expected Results:	Execution Status:
1	Enable Macro ENABLE_TEST_SOURCE in chains_common_ar12xx.c & configure test source in ChainsCommon_ar12xxEnableTestSource	Should be able to configure test source	
2	Build the code by running below command make -s -j depend; make -s -j	should be able to build	
3	Run "Radar (Single AR1243) Capture + Radar Object Detect (EVE1) + Display (TDA3xx Only) usecase" UC Select Normal Frame/Advanced Frame.	Depending upon selection Normal Frame/Advanced Frame should be selected	
4	Press "P"	Check performance stats	
5	Press 'c' to read back and verify parameters.	Should be able to read and verify parameters	
6	Press 'd' to dynamically change the slope.	Slope should be changed dynamically	
Execution type:	Manual		
Estimated exec. duration (sec):			
Priority:	Medium		
Keywords:	c_regression c_stress c_stability tda3xx-AR12-Booster		
Execution Details			
Build	REL_3_3		
Tester	x0246581		
Execution Result:	Passed		
Execution Mode:	Manual		
Execution duration (sec):			

Test Case VISIONSDK-331: Radar_AR12_Capture_Null_Multiple_TimesSummary:

Radar Capture Null UC

Input : AR12

Output : Null

Preconditions:

Ensure AR12 sensor Radar HW is connected to TDA3x EVM

Debug prints will be in UART2

#:	Step actions:	Expected Results:	Execution Status:
1	Boot TDA3x with Radar setup/TDA3xx ALPS Board	Shoul display Main Menu	
2	Run "Radar (Single AR1243) Capture + Null (TDA3xx Only) usecase" UC	No Display	
3	Press "P"	Check performance stats	
<u>Execution type:</u>	Manual		

<u>Estimated exec. duration (sec):</u>	
<u>Priority:</u>	Medium
<u>Keywords:</u>	c_regression c_qualification tda3xx-alps tda3xx-AR12-Booster
Execution Details	
Build	REL_3_3
Tester	x0246581
<u>Execution Result:</u>	Passed
<u>Execution Mode:</u>	Manual
<u>Execution duration (sec):</u>	

1.3.Test Suite : Build

1.3.1.Test Suite : Radar_Builds

Test Case VISIONSDK-242: Radar_default_build			
<u>Summary:</u>			
Radar Default Build			
<u>Preconditions:</u>			
Follow UG to Install release package			
Copy all necessary components (gcc tool)			
<u>#:</u>	<u>Step actions:</u>	<u>Expected Results:</u>	<u>Execution Status:</u>
1	Navigate to (radar_install_path)/vision_sdk/build & run make -s showconfig	Should display config for tda3xx_evm_bios_radar	
2	Check default config	By default all IPU1_0, IPU1_1, DSP1, EVE1 are enabled Memory should be 128MB NDK should be disabled & A15_TARGET_OS=Bios	
3	run make -s -j depend & then make -s -j	Should build binaries without any error	
4	run make -s appimage	should create Appimage	
5	run make -s sbl	Should create SBL	
<u>Execution type:</u>	Manual		
<u>Estimated exec. duration (sec):</u>			
<u>Priority:</u>	Medium		
<u>Keywords:</u>	tda3xx-evm c_qualification		
Execution Details			
Build	REL_3_3		
Tester	x0246581		
<u>Execution Result:</u>	Passed		
<u>Execution Mode:</u>	Manual		
<u>Execution duration (sec):</u>			

Test Case VISIONSDK-280: Radar_different_builds			
<u>Summary:</u>			
Radar different configurations Build			
<u>Preconditions:</u>			
Follow UG to Install release package			
All ti_components (including PDK) should be part of release package			
Copy all necessary components (gcc tool,linaro tool chain)			
<u>#:</u>	<u>Step actions:</u>	<u>Expected Results:</u>	<u>Execution Status:</u>
1	Navigate to (v sdk_install_path)/vision_sdk/build & run make -s showconfig	Should display config for tda3xx_evm_bios_radar	

2	Modify Rules.mk file to other available MAKECONFIG & run make -s showconfig	Should display config for MAKECONFIG selected	
3	run make -s -j depend & then make -s -j	Should build binaries without any error	
4	run make -s appimage	should create Appimage	
5	run make -s sbi	Should create SBL	
<u>Execution type:</u>	Manual		
<u>Estimated exec. duration (sec):</u>			
<u>Priority:</u>	Medium		
<u>Keywords:</u>	tda2xx-evm tda3xx-evm tda3xx_rvp tda3xx-alps tda3xx-AR12-Booster c_integration		
<u>Attached files</u>	<ul style="list-style-type: none"> • Radar Different Build Config : build_radar.sh • build_radar.sh 		
Execution Details			
Build	REL_3_3		
Tester	x0246581		
<u>Execution Result:</u>	Passed		
<u>Execution Mode:</u>	Manual		
<u>Execution duration (sec):</u>			

1.4.Test Suite : Release_Process

Test Case VISIONSDK-245: VSDK_Radar_release_check_list			
<u>Summary:</u>			
VSDK & Radar release check list			
<u>Preconditions:</u>			
VSDK & Radar RC package already installed & tested			
Verify that release goes through the standard release process			
<u>#:</u>	<u>Step actions:</u>	<u>Expected Results:</u>	<u>Execution Status:</u>
1	Check for licenses, manifest, release notes, test reports, datasheets	Release shall comply for the basic release process such as export license, OSRB approval etc.	
2	Check there are test cases for all product requirements (planned in release) & executed in testing phase	Traceability report (Req -> Test) should have all req mapped to tc Test result matrix should have nothing in "Not Run" state	
3	Check updated project plan, test plan, test strategy docs for release are all available in clearcase	All updated version of docs should be available in clearcase	
4	Check for all docs available in vision_sdk/docs folder	All updated docs for current release should be available	
5	Check for all docs available in vision_sdk/docs folder	All updated docs for current release should be available	
<u>Execution type:</u>	Manual		
<u>Estimated exec. duration (sec):</u>			
<u>Priority:</u>	Medium		
<u>Keywords:</u>	None		
Execution Details			
Build	REL_3_3		
Tester	x0246581		
<u>Execution Result:</u>	Passed		
<u>Execution Mode:</u>	Manual		
<u>Execution duration (sec):</u>			

Test Case VISIONSDK-247: Radar_package_creation_and_installation			
<u>Summary:</u>			
Radar package creation & installation on windows & linux machine			
<u>Preconditions:</u>			
Radar RC package installed & tested			
#:	Step actions:	Expected Results:	Execution Status:
1	Modify MPI files to pick correct ti_components Modify InstallJammer Environment script Trigger Jenking project for packaging	Windows & Linux installer should be created	

2	Install on windows machine Check for all customer collaterals & Build with default config	Installation should be success Release package should include all customer collaterals such as user guide, data sheet, Release notes, Test reports, Developer guide etc Build should be success	
3	Install on Linux machine Check for all customer collaterals & Build with default config	Installation should be success Release package should include all customer collaterals such as user guide, data sheet, Release notes, Test reports, Developer guide etc Build should be success	
<u>Execution type:</u>		Manual	
<u>Estimated exec. duration (sec):</u>			
<u>Priority:</u>		Medium	
<u>Keywords:</u>		c_qualification	
Execution Details			
Build	REL_3_3		
Tester	x0246581		
<u>Execution Result:</u>		Passed	
<u>Execution Mode:</u>		Manual	
<u>Execution duration (sec):</u>			

1.5.Test Suite : Boot_Modes

1.5.1.Test Suite : SD_Boot

Test Case VISIONSDK-273: Load_BIOS_Binaries_using_SD_Card			
<u>Summary:</u>			
Load Binaries using SD Card			
supported on TDA2x/TDA2Ex/TDA2Ex Entry			
<u>Preconditions:</u>			
Build & Copy Appimage & MLO to SD card			
<u>#:</u>	<u>Step actions:</u>	<u>Expected Results:</u>	<u>Execution Status:</u>
1	Insert SD card into card slot & Follow UG to set SYSBOOT PIN for SD boot	SYSBOOT PINs should be for SD boot	
2	Boot EVM	EVM should boot with binaries & Display Main Menu	
<u>Execution type:</u>	Manual		
<u>Estimated exec. duration (sec):</u>			
<u>Priority:</u>	Medium		
<u>Keywords:</u>	tda2xx-evm tda2ex-evm tda2ex-entry tda2px-evm c_qualification		
Execution Details			
Build	REL_3_3		
Tester	x0246581		
<u>Execution Result:</u>	Passed		
<u>Execution Mode:</u>	Manual		
<u>Execution duration (sec):</u>			

1.5.2.Test Suite : QSPI_Boot

Test Case VISIONSDK-274: Load_Binaries_using_QSPI			
<u>Summary:</u>			
Load Binaries using QSPI			
<u>Preconditions:</u>			
Build Appimage & SBL for QSPI			
<u>#:</u>	<u>Step actions:</u>	<u>Expected Results:</u>	<u>Execution Status:</u>
1	Connect EVM through CCS debug & Follow UG to set SYSBOOT PIN for CCS debug	SYSBOOT PINs should be for debug	
2	Follow UG to Flash SBL & ApplImage to QSPI	SBL & ApplImage should be flashed to QSPI	
3	Discoonnect CCS & Follow UG to set SYSBOOT PIN for QSPI Boot	SYSBOOT PIN should be for QSPI Boot	
4	Boot EVM	EVM should boot with binaries & Display Main Menu	
<u>Execution type:</u>	Manual		
<u>Estimated exec. duration (sec):</u>			
<u>Priority:</u>	Medium		
<u>Keywords:</u>	tda2xx-evm tda2ex-evm tda3xx-evm tda2ex-entry tda2px-evm tda3xx_rvp		
Execution Details			
Build	REL_3_3		
Tester	x0246581		
<u>Execution Result:</u>	Passed		
<u>Execution Mode:</u>	Manual		
<u>Execution duration (sec):</u>			

1.5.3.Test Suite : QSPI_SD_Boot

Test Case VISIONSDK-275: Load_Binaries_using_QSPI_SD

Summary:

Load Binaries using QSPI SD
supported only on TDA3x/RVP

Preconditions:

Build Appimage & SBL for QSPI SD Boot
Copy AppImage to SD card

<u>#:</u>	<u>Step actions:</u>	<u>Expected Results:</u>	<u>Execution Status:</u>
1	Connect EVM through CCS debug & Follow UG to set SYSBOOT PIN for CCS debug	SYSBOOT PINs should be for debug	
2	Follow UG to Flash SBL	SBL should be flashed to QSPI	
3	Disconnect CCS Insert SD card to SD card slot Follow UG to set SYSBOOT PIN for QSPI SD Boot	SYSBOOT PIN should be for QSPI SD Boot	
4	Boot EVM	EVM should boot with binaries & Display Main Menu	
<u>Execution type:</u>	Manual		
<u>Estimated exec. duration (sec):</u>			
<u>Priority:</u>	Medium		
<u>Keywords:</u>	tda3xx-evm tda3xx_rvp		
Execution Details			
Build	REL_3_3		
Tester	x0246581		
<u>Execution Result:</u>	Passed		
<u>Execution Mode:</u>	Manual		
<u>Execution duration (sec):</u>			

1.5.4.Test Suite : CCS_Boot

Test Case VISIONSDK-332: Load_Binaries_using_CCS			
<u>Summary:</u>			
Load Binaries using CCS			
<u>Preconditions:</u>			
Build binaries			
<u>#:</u>	<u>Step actions:</u>	<u>Expected Results:</u>	<u>Execution Status:</u>
1	Connect EVM through CCS debug & Follow UG to set SYSBOOT PIN for CCS debug	SYSBOOT PINs should be for debug	
2	Load binaries on each core separately or use the ".js" script available under vision_sdk/build/rtos/scripts to load on all cores at once	Binaries should be load on each core successfully & Display main menu on uart console	
3	From Main Menu run any UC	UC should run successfully	
4	Check for few register address whether displaying proper data or not	Data should be proper	
<u>Execution type:</u>	Manual		
<u>Estimated exec. duration (sec):</u>			
<u>Priority:</u>	Medium		
<u>Keywords:</u>	tda2xx-evm tda2ex-evm tda3xx-evm tda2ex-entry tda2px-evm tda3xx_rvp tda3xx-alps tda3xx-AR12-Booster		
Execution Details			
Build	REL_3_3		
Tester	x0246581		
<u>Execution Result:</u>	Passed		
<u>Execution Mode:</u>	Manual		
<u>Execution duration (sec):</u>			