Vision SDK TDA2xx

(v02.09.00)

User Guide

TABLE OF CONTENTS

1 Introduction

Vision Software Development Kit (Vision SDK) is a multi-processor software development package for TI's family of ADAS SoCs. The software framework allows users to create different ADAS application data flows involving video capture, video pre-processing, video analytics algorithms, and video display. The framework has sample ADAS data flows which exercises different CPUs and HW accelerators in the ADAS SoC and demonstrates how to effectively use different sub-systems within the SoC. Frame work is generic enough to plug in application specific algorithms in the system.

Vision SDK is currently targeted for the TDA2xx family of SoCs

1.1 References

Refer the below additional documents for more information about Vision SDK

Document	**Description**
VisionSDK_ReleaseNotes.pdf	Release specific information
VisionSDK_UserGuide.pdf	This document. Contains install, build, execution information
VisionSDK_DataSheet.pdf	Summary of features supported, not supported in a release. Performance and benchmark information.
VisionSDK_ApiGuide.CHM	User API interface details
VisionSDK_SW_Architecture.pdf	Overview of software architecture
VisionSDK_DevelopmentGuide.pdf	Details how to create data flow (s) & add new functionality
VisionSDK_SurroundView_DemoSetUpGuide.pdf	Document contains the steps for hardware setup for calibrated surround view demo

2 System Requirements

This chapter provides a brief description on the system requirements (hardware and software) and instructions for installing Vision SDK.

2.1 PC Requirements

Installation of this release needs a windows machine with about 6GB of free disk space. Building of the SDK is supported on windows environment.

2.2 Software Requirements

All software packages required to build and run the Vision SDK are included as part of the SDK release package except for the ones mentioned below

2.2.1 A15 Compiler, Linker

The windows installer for the linaro tools should be downloaded from below link

https://launchpad.net/gcc-arm-embedded/+milestone/4.7-2013-q3-update

The tools need to be installed in "<install dir>/ti_components/cg_tools/windows/gcc-arm-none-eabi-4_7-2013q3" location.

IMPORTANT NOTE: A15 Compiler and linker MUST be installed before proceeding else compile will fail. Also make sure the compiler is installed at the exact path mentioned above

2.2.2 Code Composer Studio

CCS is needed to load, run and debug the software. CCS can be downloaded from the below link. CCS version 6.0.1.00040 should be installed.

SDK also works with 5.4.0.00091 and 5.5.0.00077.

http://processors.wiki.ti.com/index.php/Download_CCS

2.3 Hardware Requirements

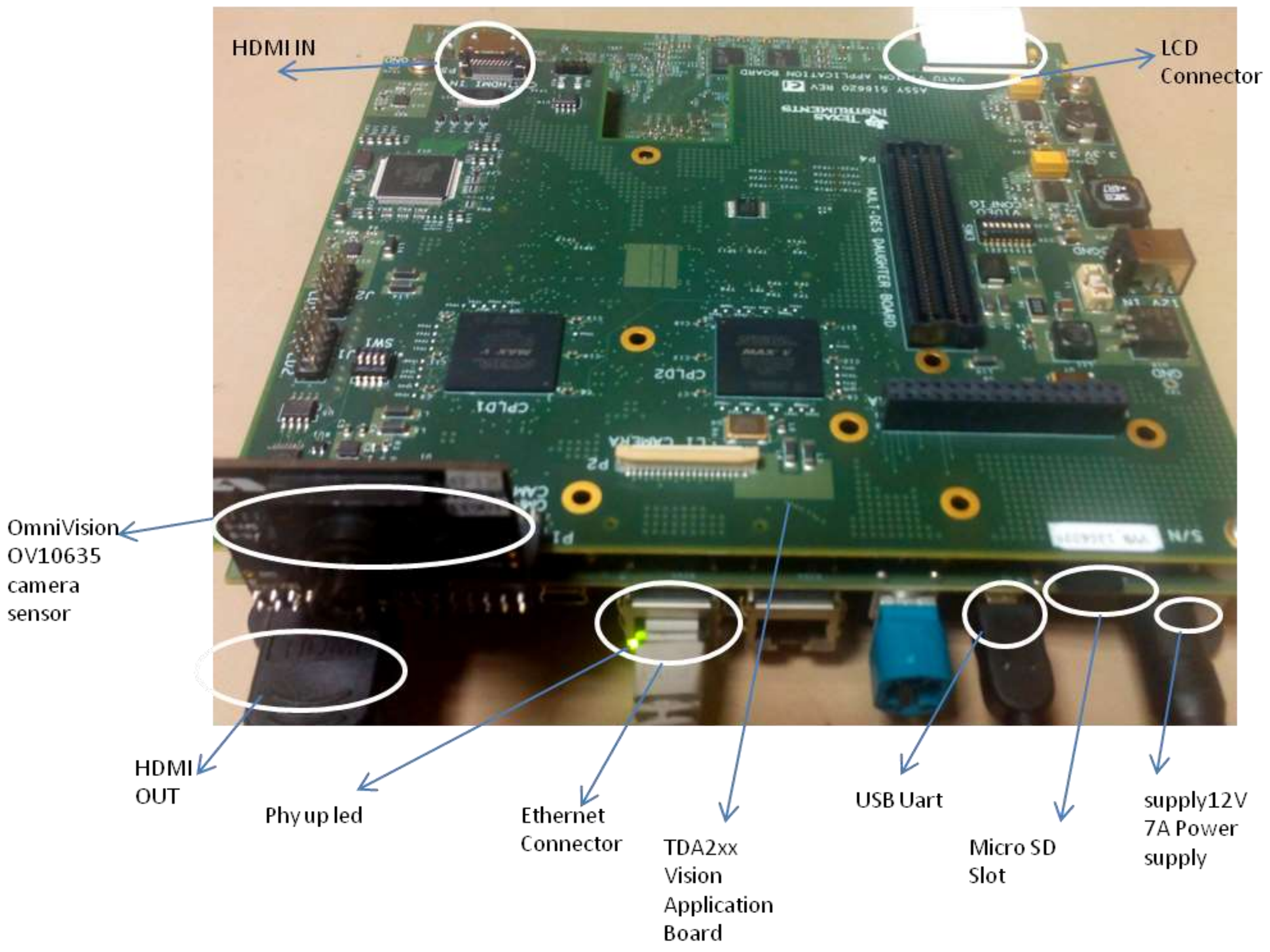
Hardware setup for Single Camera View (SCV), Multichannel AVB Multi-Channel View usecase and LVDS Multi Camera View (LVDS MCV) use-case is described in this section

2.3.1 SCV/AVB Use-case Hardware Setup

SCV/AVB use-case needs the below hardware

1. TDA2xx EVM (Rev D)
2. TDA2xx Vision Application Board (Rev C)
3. OV10635 Sensor (for SCV only)
4. 1Gbps Ethernet Cable (for AVB only)
5. WVGA LCD DC from Spectrum Digital (part #703663) OR
6. HDMI 1080p60 capable Display Monitor

Setup is shown below


HDMI IN
LCD Connector
OmniVision OV10635 camera sensor
HDMI OUT
Phy up led
Ethernet Connector
TDA2xx Vision Application Board
USB Uart
Micro SD Slot
supply12V 7A Power supply

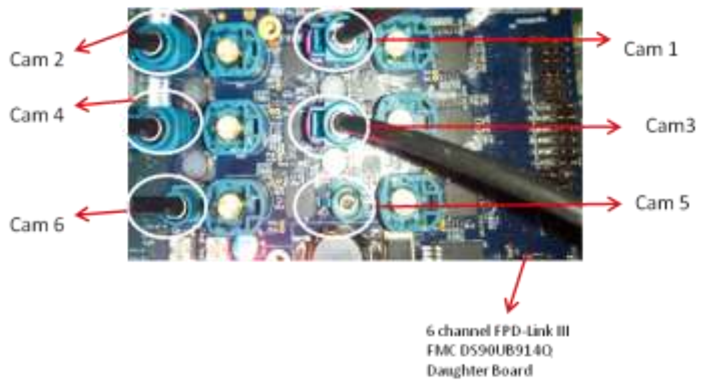
LVDS MCV Use-case Hardware Setup

LVDS MCV use-case needs the below hardware

1. TDA2xx EVM (Rev D)
2. TDA2xx Vision Application Board (Rev C)
3. 6 channel FPD-Link III FMC SV600964 Daughter Board (Rev E1)
4. 4 x DS90UB913A EVMs (Rev A)
5. 4 x OV10635 Sensor (additional 5th DS90UB913A EVM, OV10635 sensor and cable would be required in order to run the Surround view demo).
6. 4 x Rosenberger HSD connectors and cables
7. WVGA LCD DC from Spectrum Digital (part #703663) OR
8. HDMI 1080p60 capable Display Monitor

The LVDS MCV use case setup is shown in the snapshots below:

2.3.1.1 DeSerializer board



IMPORTANT NOTE: *Camera 1, Camera 2, Camera 3 and Camera 4 are used for 4 channel LVDS use-case and MUST be connected as shown in above figure.*
5th Camera is used for Edge detection currently connected for Cam 6 to enable both AVB and LVDS usecase

IMPORTANT NOTE: *To enable 6th Camera or 6CH LVDS capture, networking MUST be disabled, since there is pinmux conflict between VIP port used for 6th camera capture and Ethernet port. To disable Ethernet and enable 6CH capture do below change,*

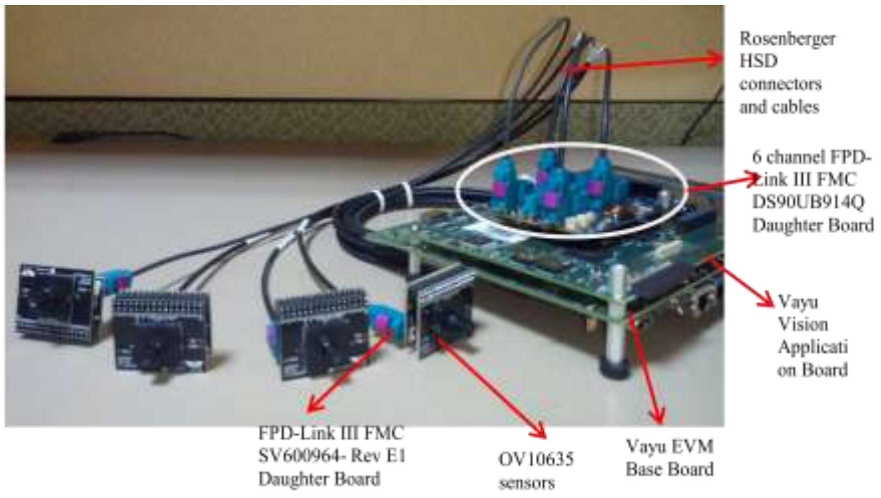
File: \vision_sdk\Rules.make
ifeq ($(NDK_PROC_TO_USE),)
*NDK_PROC_TO_USE=**none***
endif

File: \vision_sdk\examples\tda2xx\include\video_sensor.h
*#define VIDEO_SENSOR_NUM_LVDS_CAMERAS (**6**)*

*Do, "gmake –s config" to and check value of **NDK_PROC_TO_USE** to confirm this setting will get applied.*
Do "gmake –s depend_ndk_fatfs;gmake -s" to do an incremental build with modified settings

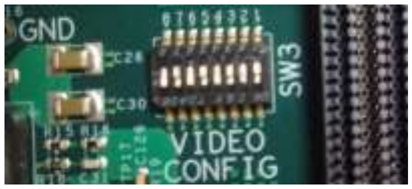
For SRV Demo setup Ref: VisionSDK_SurroundView_DemoSetUpGuide.pdf

2.3.1.2 *Complete LVDS Setup*


Rosenberger HSD connectors and cables
6 channel FPD-Link III FMC DS90UB914Q Daughter Board
Vayu Vision Application Board
FPD-Link III FMC SV600964- Rev E1 Daughter Board
OV10635 sensors
Vayu EVM Base Board

2.3.2 Capture Pin Settings

Video Config pins needs to set for different capture inputs



VIDEO CONFIG switch settings (SW3 on TDA2xx Vision Application Board (set for Ov10635 in Original version of CPLD))

Capture Type	**Hardware controlled pin settings Vision Application Board (Rev C CPLD) (default cpld image)**								**Software controlled pin settings New Version Of CPLD flashed (cpld_1_cam3_shift.pof)**							
	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
OV10635	**OFF**	**ON**	**OFF**	**ON**	**OFF**	**ON**	**OFF**	**ON**	OFF	OFF	OFF	OFF	OFF	OFF	OFF	**OFF**
LVDS	**OFF**	**OFF**	**ON**	**OFF**	**OFF**	**ON**	**OFF**	**ON**	OFF	OFF	OFF	OFF	OFF	OFF	OFF	**OFF**
HDMI	**OFF**	**OFF**	**ON**	**ON**	**OFF**	**ON**	**OFF**	**ON**	OFF	OFF	OFF	OFF	OFF	OFF	OFF	**OFF**

Cpld image is required for VIP input Muxing,

With the cpld_1_cam3_shift.pof and later the software control will work,

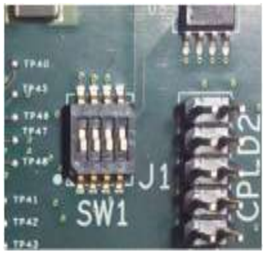
On default Rev C board the captured image won't be proper. Either program the Cpld with new image or use hardware controlled pin settings.

2.3.3 EDID Programming for HDMI Capture.

EDID information needs to be programmed on the EEPROM present on Vision Application board. This is required for the HDMI source to recognize the format and resolution supported by the receiver (TDA2xx SoC). If this step is not done or if this step fails, then TDA2xx SoC will not be able to receive data via HDMI.

It's recommended to program the HDMI receivers EDID. The default EDID is programmed to receive 1080P60 video streams only. If stream of different resolution is required (or EDID is corrupted), the EDID would require an update. The following steps details the procedure to re-program the EDID.

1. Change pins 1 and 2 of SW1 (on vision application board, near CPLD2 connector) to ON.



2. Connect usb cable from board to PC and setup UART for logs (Ref Uart settings)
3. Connect to a15 core and load 'edid_programming_1080p_60.xa15fg' binary using CCS (Refer Load using CCS till step 8).

   Binary placed under vision_sdk\docs\edid\edid_programming_1080p_60.xa15fg
4. Run the core and wait till "EDID programming success full" message comes on UART.
5. Terminate the JTAG session and restart the board.
6. Before Running Vision Sdk binaries, change pins 1 and 2 of SW1 to OFF (towards numbers 1 and 2).

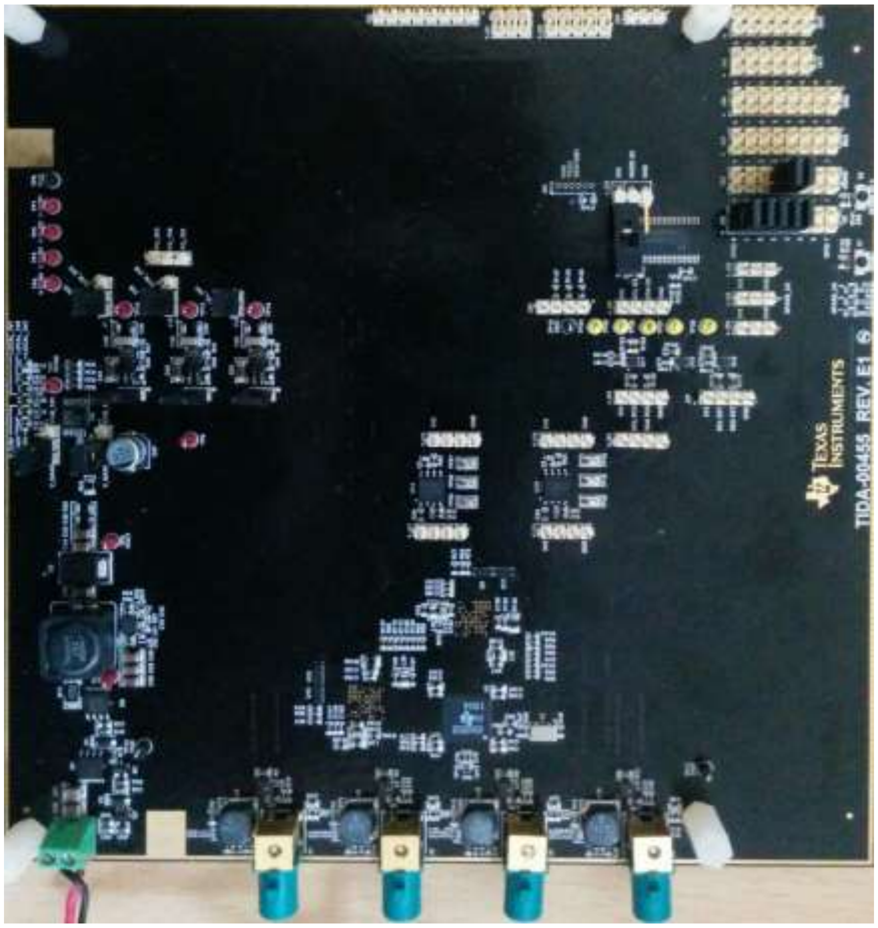
NOTE: Refer Load and Run using CCS for details of running the binaries

IMPORTANT NOTE: *If LVDS Setup is connected, EDID programming may fail. Disconnect the LVDS Daughter board and then program EDID.*
If EDID is not programmed correctly detect video will fail when HDMI capture is done.

2.3.4 Surround view use-case using TIDA00455/OV490

This use-case need following hardware

1. TDA2XX base EVM
2. TIDA00455 daughter card



Modifications needed:

a. SPI flashes on-board TIDA00455 must be configured with correct firmware

b. MSP430 on-board TIDA00455 must be configured with correct firmware

c. Modify R30 on TIDA00455 to ensure correct output on Power-on-coax network as required by camera modules\

3. 4 cameras modules sending serialized video data in RAW(Bayer) format connected to TIDA00455 using FPD-Link-III coax-cables.

   DS90UB913EVM/SAT0074 with an appropriate camera module and adapter can be used. For testing, DS90UB813EVM with OV10640 camera's was used.

Modifications needed:

a. If using DS90UB913EVM for camera modules, modify R56 to appropriate value to ensure correct I/O voltage to camera

b. If using DS90UB913EVM for camera modules, modify MSP430 firmware such that GPIO1 is not set to "1" to limit current usage by GPIO1 LED.

Contact your local FAE to get access to appropriate firmware for MSP430 and OV490.

2.3.5 INA226 Self power measurement setup

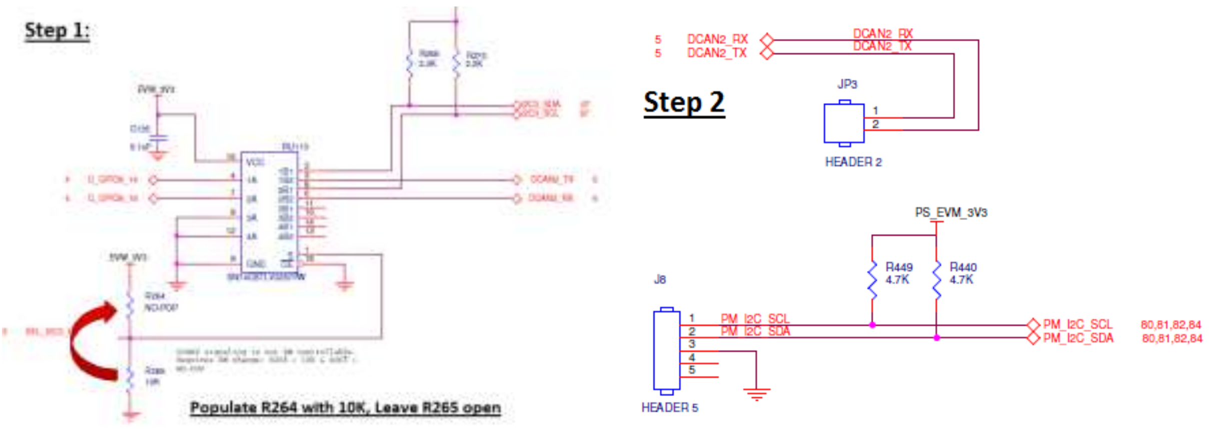
In order to measure power of the TDA2xx voltage rails from the TDA2xx device itself without the need for an extra daughter card the TDA2xx EVM should be modified as shown below. Without this board modification the power measurement software which runs on the device will fail.

STEP 1: Change the select for RU113 multiplexer on the board by making R264 = 10k & R265 = NO-POP. This allows DCAN2 Signaling.

STEP 2: Then perform a blue wiring for the following connections:

- DCAN2_TX (JP3 pin 1) to PM_I2C_SDA (J8 pin 2)
- DCAN2_RX (JP3 pin 2) to PM_I2C_SCL (J8 pin 1)

From Software Configure the PAD configuration registers such that the gpio6_14 and gpio6_15 pads operate as I2C3_SDA and I2C3_SCL respectively. Note that this is taken care from software. Additionally ensure SEL_I2C3_CAN2 is high.



2.4 Software Installation

vision_sdk_02_xx_xx_xx_setupwin32.exe is the SDK package installer.

Copy the installer to the path of your choice.

Double click the installer to begin the installation.

Follow the self-guided installer for installation.

IMPORTANT NOTE: On some computers running as administrator is needed. Right click on the installer and select option of "Run as administrator". If this is not done then you may see a message like "This program might not have installed correctly"

On completion of installation a folder by name VISION_SDK_02_xx_xx_xx would have got created in the installation path.

2.4.1 Uninstall Procedure

To uninstall, double click on uninstall.exe created during installation in the folder VISION_SDK_02_xx_xx_xx.

At the end of uninstall, VISION_SDK_02_xx_xx_xx folder still remains. It is just an empty folder. It can be deleted manually.

3 Build and Run

This chapter provides a brief overview of the sample application or use case present in the SDK and procedure to build and run it.

3.1 Overview of application in release

The Vision SDK supports the following use-cases as examples

- Single channel capture use-cases
    - Single channel capture, display use case
    - Single channel capture, Frame copy algorithm on DSP1, display use case
    - Single channel capture, Frame copy algorithm on EVE1, display use case
    - Single channel capture, Frame copy algorithm on A15, display use case
    - Single channel capture, Edge Detect algorithm on EVE1, display use case
    - Single channel capture, Dense Optical Flow algorithm, display use case
    - Single channel capture, Pedestrian and Traffic Sign algorithm, display use case
    - Single channel capture, Sparse Optical Flow algorithm, display use case
    - Single channel capture, Lane Detect, display use case
    - Single channel capture, Subframe Copy, display use case
    - Single channel capture, FrontCam Analytics PD+TSR+LD, display use case
    - Single channel capture, Freeze Frame detect and display usecase
    - Single channel capture, Encode, Decode, resize and display usecase
- Multi-channel LVDS capture use-cases
    - 4Ch LVDS capture, VPE scale, Timestamp Sync, 2x2 DMA SW Mosaic on IPU1-0, display use case
    - 5CH LVDS capture, Surround View Stitching Algorithm on DSP, Edge Detect Algorithm on (EVE1), display use case
    - 5CH LVDS capture, Surround View Stitching Algorithm on DSP, Dense Optical Flow Algorithm on (EVE1), display use case
    - 5CH LVDS capture, Surround View Stitching Algorithm on DSP, PD (DSP/EVE) and ED on (EVE), display use case
- AVB RX Use-cases,
    - 4CH AVB capture on (IPU1_1), Decode, VPE scale, 2x2 DMA SW Mosaic on IPU1-0 display use case
    - 4CH AVB Capture + Surround View (DSPx) + Display (HDMI)

Refer to VisionSDK_DataSheet.pdf for detailed description of each use-case.

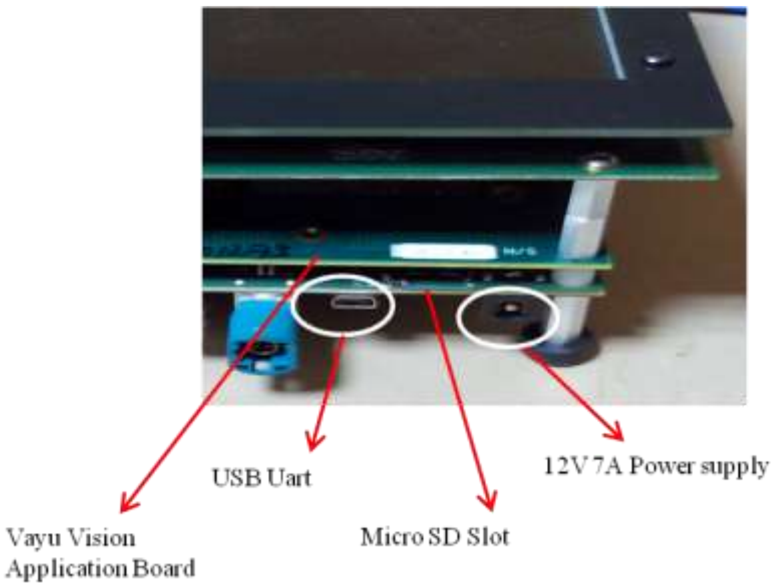
The demos support following devices as capture source
- OV10635 sensor (default)
- HDMI source

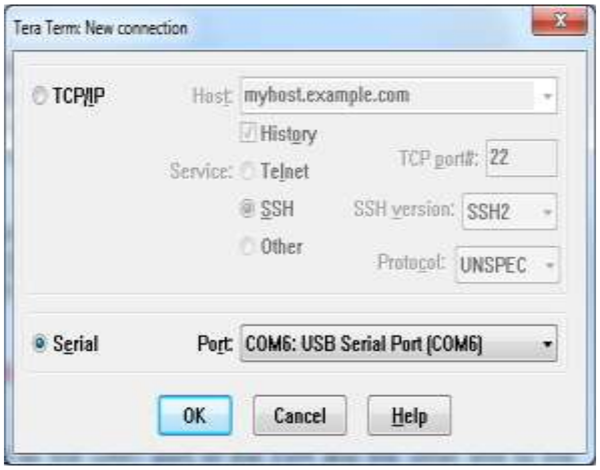
The demos support following devices as display devices
- LCD 800x480
- HDMI 1080p60 (default)

Use option "s" on the main menu in UART to select different capture and display devices.

3.2 Uart settings

Connect a serial cable to the UART port of the EVM and the other end to the serial port of the PC (*configure the HyperTerminal at 115200 baud rate*) to obtain logs and select demo.




Tera Term: New connection

TCP/IP — Host: myhost.example.com

[✓] History

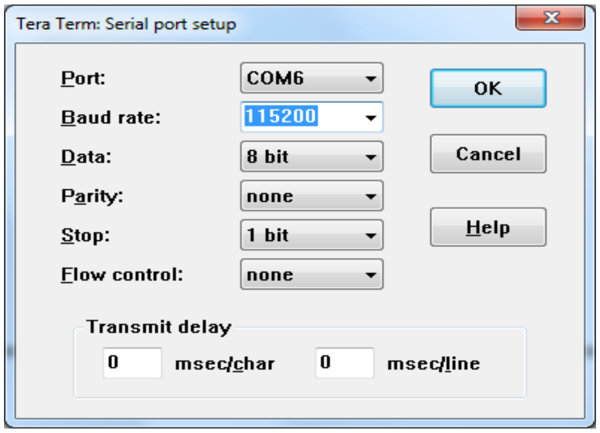
Service: Telnet / [✓ SSH] / Other

TCP port#: 22

SSH version: SSH2

Protocol: UNSPEC

[✓ Serial] — Port: COM6: USB Serial Port (COM6)


Tera Term: Serial port setup

Port: COM6
Baud rate: 115200
Data: 8 bit
Parity: none
Stop: 1 bit
Flow control: none

Transmit delay
0 msec/char 0 msec/line

3.3 Load using SD card

NOTE: The application can be run using SD card and SD card boot or using CCS. This section shows how to run using SD card boot.

Application image is run on the SoC via Secondary Boot Loader (SBL) present in SD card.

3.3.1 Option 1: Steps to prepare a bootable SD card

- Ensure Empty SD card (at least 256MB, preferably 4GB SDHC) is available.
- Ensure SD memory card reader is available.
- Create a primary FAT partition on MMC/SD card (FAT32 format with sector size 512) and mark it as Active. A partition manager utility has to be used for the same.
- Format SD card from DOS command line as below.
  "format <drive> /A:512 /FS:FAT32"
  Make SD card partition as active using below tool
  http://www.pcdisk.com/download.html
  IMPORTANT NOTE: Create a primary FAT partition on MMC/SD card (FAT32 format with sector size 512 bytes mark the partition as active.

3.3.2 Option 2: Steps to prepare a bootable SD card using DISKPART

- Open windows 7 Command prompt and Run as Administrator mode
- Enter command "diskpart.exe"
  C:\Windows\system32>diskpart.exe will take you DISKPART prompt
  Warning: Enter below command carefully w.r.t your computer/laptop SD card disk number. Choosing wrong disk number may delete data present in other drive

  To list all disk drive present on computer
  DISKPART> list disk

  Select the SD card disk number, in my case it is disk 1
  DISKPART> select disk 1
  Now all next command applicable only to disk 1(SD card)

Delete entire partition

DISKPART> clean

To create Primary partition

DISKPART> create partition primary

To list partition

DISKPART> list partition

Select partition

DISKPART> select partition 1

To list volume

DISKPART> list volume

Select volume associated with SD card, In my case its 3

DISKPART> select volume 3

Format SD card, please wait this may take few seconds

DISKPART>format quick fs=fat32 unit=512 label=SD_BOOT

Make disk active

DISKPART> active

To exit utility

DISKPART> exit

3.3.3 Steps to generate MLO

NOTE: SBL MLO image is built from starterware package.
To build MLO Run the command **gmake –s sbl_sd** from vision_sdk root dir
And run the **sbl_mlo_create.bat** placed at vision_sdk\build\scripts
This generates an MLO under vision_sdk\build\scripts\mlo

To build the mlo for different memory map, select required configuration in Makefile under vision_sdk (follow comments from Makefile under SBL build Targets).

3.3.4 Steps to generate appImage

Following steps need to be followed to generate the application image

1. Make sure the executables are built as shown in Building the application
2. To generate the application image run the batch file shown below
   `vision_sdk\MulticoreImageGen.bat`

IMPORTANT NOTE: If some cores are disabled from build, comment them from MulticoreImageGen.bat and generate the AppImage.

REM is the comment used to comment out in .bat file

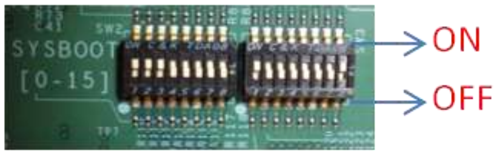
REM set App_IPU1_CPU1 is sufficient

3.3.5 SD Card setup

Once the AppImage and MLO are generated , Copy the MLO and AppImage at root folder of formatted SD Card

3.3.6 Hardware Pin settings for SD Boot

Make sure the Boot Mode Select Switch is set for the SD boot mode **on TDA2xx Base EVM**. This is done by setting the pins SYSBOOT (SW2+SW3)



- Please refer **Boot Modes of SBL** section In **SBL_UserGuide.pdf**

(VISION_SDK_xx_xx_xx_xx\ti_components\drivers\starterware_xx_xx_xx_xx\bootloader)

Pre-build MLO is located at C:\VISION_SDK_XX_XX_XX_XX\vision_sdk\prebuilt\tda2xx-evm\sbl_boot

3.4 Load using QSPI

3.4.1 Steps to generate qspi writer tools

NOTE: SBL qspi image is built from starterware package.
To build qspi Run the command **gmake –s sbl_qspi** from vision_sdk root dir
And run the **sbl_qspi_create.bat** placed at vision_sdk\build\scripts
This generates all required tools under vision_sdk\build\scripts\qspi

1. sbl_a15host_release.xa15fg
2. qspiFlashWriter_m4_release.xem4
3. sbl_qspi

To build the qspi for different memory map, select required configuration in Makefile under vision_sdk (follow comments from Makefile under SBL build Targets).

IMPORTANT NOTE: There is a board modification required for qspi boot for boards prior to rev C. Contact local TI support for information.

3.4.2 Steps to generate appImage

Following steps need to be followed to generate the application image

1. Make sure the executables are built as shown in Building the application
2. To generate the application image run the batch file shown below
   `vision_sdk\MulticoreImageGen.bat`

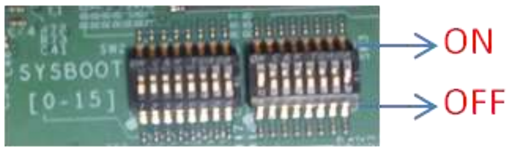
IMPORTANT NOTE: If some cores are disabled from build, comment them from MulticoreImageGen.bat and generate the AppImage.

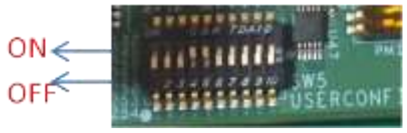
REM is the comment used to comment in .bat file

REM set App_IPU1_CPU1 is sufficient

3.4.3 Flashing steps

Flashing pin settings:





SW5 [1:10] =0001100000

- Please refer **Boot Modes of SBL** section In **SBL_UserGuide.pdf**

(VISION_SDK_xx_xx_xx_xx\ti_components\drivers\starterware_xx_xx_xx_xx\bootloader)

For loading binaries using CCS refer Load using CCS till step 8.

1. Connect A15.

Select CortexA15_0, navigate to Scripts->DRA7xx MULTICORE Initialization DRA7xx_MULTICORE_EnableALLCores

2. Connect M4 (IPU)

Halt A15 core, and Load image on M4

C:\VISION_SDK_XX_XX_XX_XX\vision_sdk\build\scripts\qspi\ qspiFlashWriter_m4_release.xem4

Run the core.

Console outputs

[Cortex_M4_IPU1_C0]

QSPI Flash writer application

MID - 1

DID - 18

Enter the File Name **C:\ VISION_SDK_XX_XX_XX_XX\vision_sdk\build\scripts\qspi\sbl_qspi**

Enter the Offset in bytes (HEX) **0x00**

Erase Options:

----

0 -> Erase Only Required Region

1 -> Erase Whole Flash

2 -> Skip Erase

```
Enter Erase Option:   1

Load Options:
-------------
    0 -> fread using code (RTS Library)
    1 -> load raw using CCS (Scripting console)
Enter Load Option:  0

Read xxxxxx bytes from [100%] file...Done.
 QSPI whole chip erase in progress
 QSPI file write started
 ************QSPI flash completed sucessfully**************
```

3. Reset the board and Repeat step 1 and 2.

   Console outputs

```
[Cortex_M4_IPU1_C0]
QSPI Flash writer application
MID - 1
DID - 18
Enter the File Name
C:\VISION_SDK_XX_XX_XX_XX\vision_sdk\build\scripts\qspi\sbl_qspi
Enter the Offset in bytes (HEX)  0x00
Erase Options:
---------------
          0 -> Erase Only Required Region
          1 -> Erase Whole Flash
          2 -> Skip Erase
Enter Erase Option:   2

Load Options:
-------------
    0 -> fread using code (RTS Library)
    1 -> load raw using CCS (Scripting console)
Enter Load Option: 0
      Read xxxxxx bytes from [100%] file...Done.
       QSPI file write started
       ************QSPI flash completed sucessfully**************
```

4. Reset the board and Repeat step 1 and 2.

```
[Cortex_M4_IPU1_C0]
QSPI Flash writer application
MID - 1
DID - 18
```

Enter the File Name
C:\VISION_SDK_XX_XX_XX_XX\vision_sdk\binaries\vision_sdk\bin\tda2xx-evm\sbl_boot\AppImage_BE

Enter the Offset in bytes (HEX): ***0x80000***

Erase Options:

----

0 -> Erase Only Required Region

1 -> Erase Whole Flash

2 -> Skip Erase

Enter Erase Option: **2**

Load Options:

----

0 -> fread using code (RTS Library)

1 -> load raw using CCS (Scripting console)

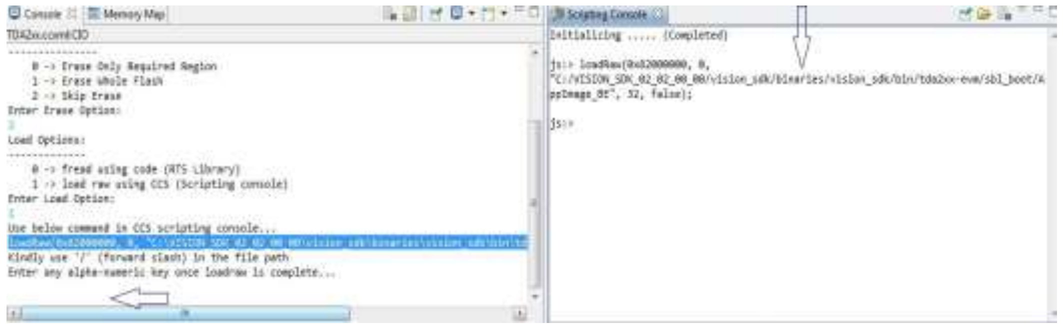
Enter Load Option: **1**

Open Scripting console window by clicking “Menu -> View -> Scripting console” and enter below command on scripting console as shown 3.5.3.1

loadRaw(0x82000000, 0, "C:/VISION_SDK_XX_XX_XX_XX/vision_sdk/binaries/vision_sdk/bin/tda2xx-evm/sbl_boot/AppImage_BE", 32, false);

In CCS console Enter any alpha-numeric key once loadraw is complete... as shown in 3.5.3.1

3.5.3.1 CCS console and scripting console



```
---------------
    0 -> Erase Only Required Region
    1 -> Erase Whole Flash
    2 -> Skip Erase
Enter Erase Option:
2
Load Options:
-------------
    0 -> fread using code (RTS Library)
    1 -> load raw using CCS (Scripting console)
Enter Load Option:
1
Use below command in CCS scripting console...
loadRaw(0x82000000, 0, "C:\VISION_SDK_02_02_00_00\vision_sdk\binaries\vision_sdk\bin\t
Kindly use '/' (forward slash) in the file path
Enter any alpha-numeric key once loadraw is complete...
```

```
Initializing ..... (Completed)

js:> loadRaw(0x82000000, 0,
"C:/VISION_SDK_02_02_00_00/vision_sdk/binaries/vision_sdk/bin/tda2xx-evm/sbl_boot/A
ppImage_BE", 32, false);

js:>
```

QSPI file write started

**************QSPI flash completed successfully***************

5. On completion change the pin setting

   Boot mode:

   - Please refer **Boot Modes of SBL** section In **SBL_UserGuide.pdf**

     (VISION_SDK_xx_xx_xx_xx\ti_components\drivers\starterware_xx_xx_xx_xx\bootloader)

3.5 Load using NOR

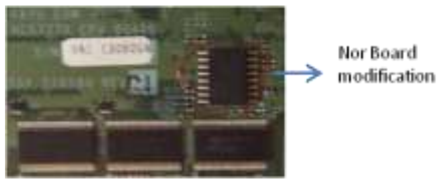
3.5.1 Steps to generate qspi writer tools

NOTE: SBL nor image is built from starterware package.
To build nor Run the command **gmake –s sbl_nor** from vision_sdk root dir
And run the **sbl_nor_create.bat** placed at vision_sdk\build\scripts
This generates all required tools under vision_sdk\build\scripts\nor

1. nor_flash_writer_m4_release.xem4
2. sbl_nor

To build the nor for different memory map, select required configuration in Makefile under vision_sdk (follow comments from Makefile under SBL build Targets).

IMPORTANT NOTE: There is a board modification required for nor boot. Contact local TI support for information.



3.5.2 Steps to generate appImage

Following steps need to be followed to generate the application image

1. Make sure the executables are built as shown in Building the application
2. To generate the application image run the batch file shown below
   `vision_sdk\MulticoreImageGen.bat`

IMPORTANT NOTE: If some cores are disabled from build, comment them from MulticoreImageGen.bat and generate the AppImage.

REM is the comment used to comment in .bat file

REM set App_IPU1_CPU1 is sufficient

3.5.3 Flashing steps

Flashing pin settings: Please refer **Boot Modes of SBL** section In **SBL_UserGuide.pdf**

(VISION_SDK_xx_xx_xx_xx\ti_components\drivers\starterware_xx_xx_xx_xx\bootloader)

For loading binaries using CCS refer Load using CCS till step 8.

1. Connect A15.

Load image

C:\VISION_SDK_XX_XX_XX_XX\vision_sdk\build\scripts\nor\nor_flash_writer_m4_release.xem4

Run the core.

Console outputs

```
[CortexA15_0] Starting NOR Flash Writer.
CFI Query...passed.
NOR Initialization:
        Command Set: Spansion
        Manufacturer: SPANSION
        Size: 0x40 MB

Enter the File Name
C:\VISION_SDK_XX_XX_XX_XX\vision_sdk\build\scripts\nor\sbl_nor
Enter the Offset in bytes (HEX) 0x00
Erase Options:
---------------
      0 -> Erase Only Required Region
      1 -> Erase Whole Flash
      2 -> Skip Erase
Enter Erase Option:   1
Erasing the NOR Flash
Erased through 0x8020000
:
:
Erased through 0x9000000
```

2. Reset the board and Repeat step 1.
   Console outputs

```
[CortexA15_0] Starting NOR Flash Writer.
CFI Query...passed.
NOR Initialization:
        Command Set: Spansion
        Manufacturer: SPANSION
        Size: 0x40 MB

Enter the File Name
C:\VISION_SDK_XX_XX_XX_XX\vision_sdk\build\scripts\nor\sbl_nor

Enter the Offset in bytes (HEX) 0x00
Erase Options:
---------------
      0 -> Erase Only Required Region
      1 -> Erase Whole Flash
```

```
          2 -> Skip Erase
Enter Erase Option:   2
Load Options:
-------------
   0 -> fread using code (RTS Library)
   1 -> load raw using CCS (Scripting console)
Enter Load Option: 0

    Reading xxxx bytes from file...
    Read XXXX bytes [100%] from file. Done!!
    Writing 69156 bytes to NOR...
    NOR Write OK through 0x8008000.
    NOR Write OK through 0x8010000.
    NOR Write OK through 0x8010E24.
    Done.
         !!! Successfully Flashed !!!
         NOR boot preparation was successful!
```

3. Reset the board and Repeat step 1.

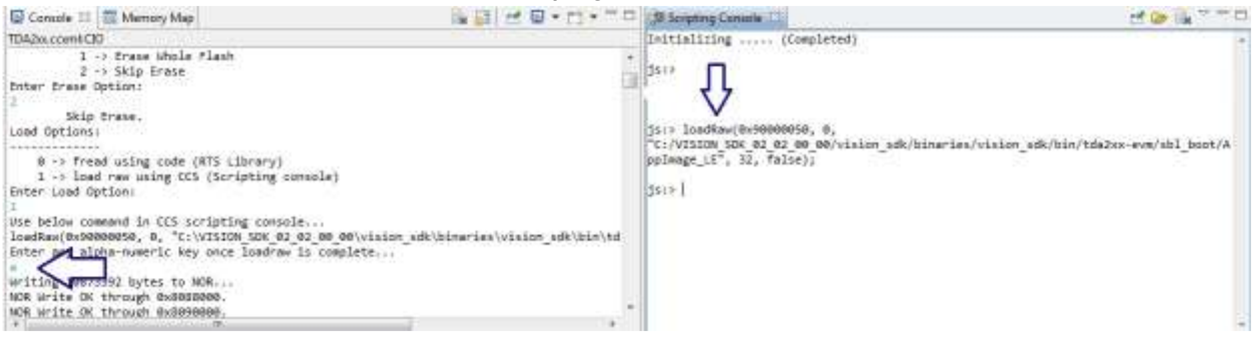
```
[CortexA15_0] Starting NOR Flash Writer.
CFI Query...passed.
NOR Initialization:
     Command Set: Spansion
     Manufacturer: SPANSION
     Size: 0x40 MB

Enter the File Name
C:\VISION_SDK_XX_XX_XX_XX\vision_sdk\binaries\vision_sdk\bin\tda2xx-evm\sbl_boot\AppImage_LE
Enter the Offset in bytes (HEX) 0x80000
Erase Options:
---------------
        0 -> Erase Only Required Region
        1 -> Erase Whole Flash
        2 -> Skip Erase
Enter Erase Option:  2

Load Options:
-------------
   0 -> fread using code (RTS Library)
   1 -> load raw using CCS (Scripting console)
Enter Load Option: 1
```

Open Scripting console window by clicking "Menu -> View -> Scripting console" and enter below command on scripting console as shown 3.6.3.1

loadRaw(0x90000050, 0, "C:/VISION_SDK_XX_XX_XX_XX/vision_sdk/binaries/vision_sdk/bin/tda2xx-evm/sbl_boot/AppImage_LE", 32, false);

In CCS console Enter any alpha-numeric key once loadraw is complete... as shown in 3.6.3.1

3.6.3.1 CCS console and scripting console



```
TDA2xx.ccent CIO
          1 -> Erase Whole Flash
          2 -> Skip Erase
Enter Erase Option:
2
        Skip Erase.
Load Options:
-------------
    0 -> fread using code (RTS Library)
    1 -> load raw using CCS (Scripting console)
Enter Load Option:
1
Use below command in CCS scripting console...
loadRaw(0x90000050, 0, "C:\VISION_SDK_02_02_00_00\vision_sdk\binaries\vision_sdk\bin\td
Enter any alpha-numeric key once loadraw is complete...
a
Writing 10873392 bytes to NOR...
NOR Write OK through 0x8088000.
NOR Write OK through 0x8090000.
```

```
Initializing ..... (Completed)

js:>

js:> loadRaw(0x90000050, 0,
"C:/VISION_SDK_02_02_00_00/vision_sdk/binaries/vision_sdk/bin/tda2xx-evm/sbl_boot/A
ppImage_LE", 32, false);

js:> |
```

Writing 10873392 bytes to NOR...

NOR Write OK through 0x8088000.

NOR Write OK through 0x8090000.

.

.

NOR Write OK through 0x8AD8000.

NOR Write OK through 0x8ADEA30.

Done.

!!! Successfully Flashed !!!

NOR boot preparation was successful!

4. On completion change the pin setting

   Boot mode:

   Flashing pin settings: Please refer **Boot Modes of SBL** section In **SBL_UserGuide.pdf**

   (VISION_SDK_xx_xx_xx_xx\ti_components\drivers\starterware_xx_xx_xx_xx\bootloader)

3.6 Load using CCS

After installing CCS, follow below steps to complete the platform setup,

1. GELs are available in
   <Install_dir>\ti_components\ccs_csp \auto_device_support_1.0.0.zip
   NOTE:
   - GELs are also be available at
     http://processors.wiki.ti.com/index.php/Device_support_files
     Under Automotive pick

Automotive vX.X.X

- To install the new GEL versions, you need to extract the zip to <CCS_INSTALL_DIR>/ccsv6/ccs_base

Change the following GEL files for vision SD as below,

- TDA2xx_ddr_config.gel
  - Set MEMMAP_512MB_SINGLE_EMIFX1 to 1
    - For older versions of VisionSDK (2.08 and older) MEMMAP_1GB_SINGLE_EMIFX1 should be set to 1 and MEMMAP_512MB_SINGLE_EMIFX1 should be set to 0
  - Set IS_EMIF2_AVAILABLE to 1
  - Set all other MEMMAP_**** macros to 0
  - For MONSTER CAM, set MONSTER_CAM to 1
- TDA2xx_multicore_reset.gel
  - Set VISION_SDK_CONFIG to 1
  - Set VISION_SDK_CONFIG_OLD to 0
    - For older versions of VisionSDK (2.08 and older) this should be set to 1
  - Set EVE_SW_CONFIG to 0
  - Add the following two lines

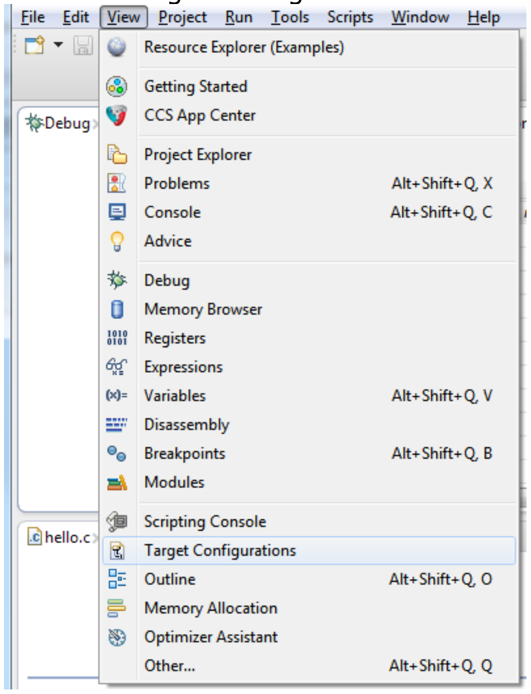
```
SetupEveMmuEntry(cpu_num, mmu_num,  3, 0x40000000, 0x40000000,
EVE_MMU_PAGESIZE_16M);
SetupEveMmuEntry(cpu_num, mmu_num,  4, 0x48000000, 0x48000000,
EVE_MMU_PAGESIZE_16M);
```

    after the line (line number 736)

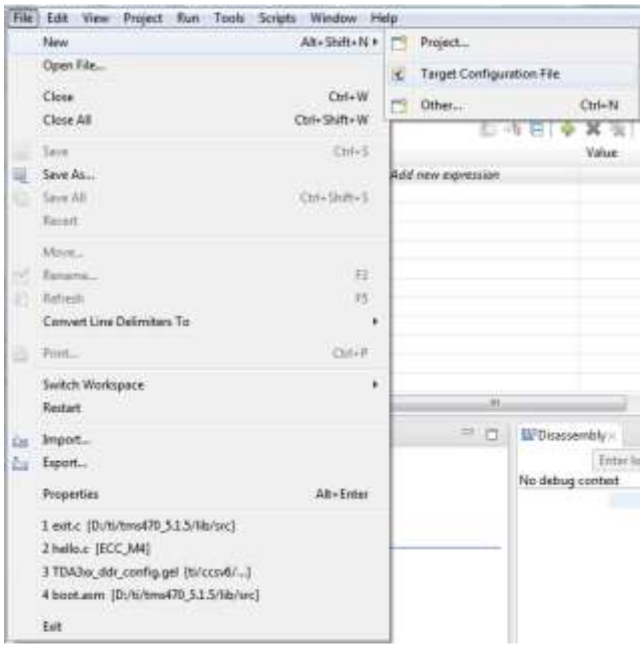
```
SetupEveMmuEntry(cpu_num, mmu_num,  2,   vec_base,   vec_base,
EVE_MMU_PAGESIZE_16M);
```

    This step will be absorbed in future GEL release packages.

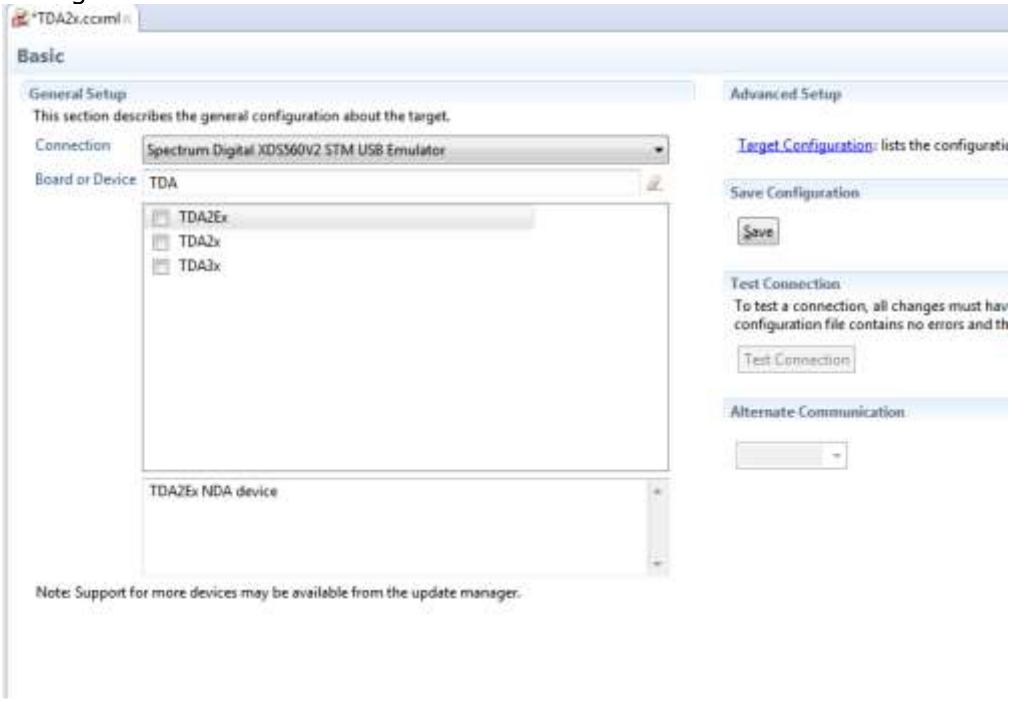
2. CCS Target Configuration creation:
   a. Open "Target Configurations" tab, by navigating through the menu "View ->Target Configurations".



   b. Create a new Target Configuration (TDA2xx Target Configuration) by navigating through the menu "File->New->Target Configuration File".



c. Specify Connections as "Spectrum Digital XDS560V2 STM USB Emulator". Specify Board or Device as "**TDA2x**". Then click on "Target Configuration link"


*TDA2x.ccxml

Basic

General Setup

This section describes the general configuration about the target.

Connection: Spectrum Digital XDS560V2 STM USB Emulator

Board or Device: TDA

- TDA2Ex
- TDA2x
- TDA3x

TDA2Ex NDA device

Note: Support for more devices may be available from the update manager.

Advanced Setup

Target Configuration: lists the configurati

Save Configuration

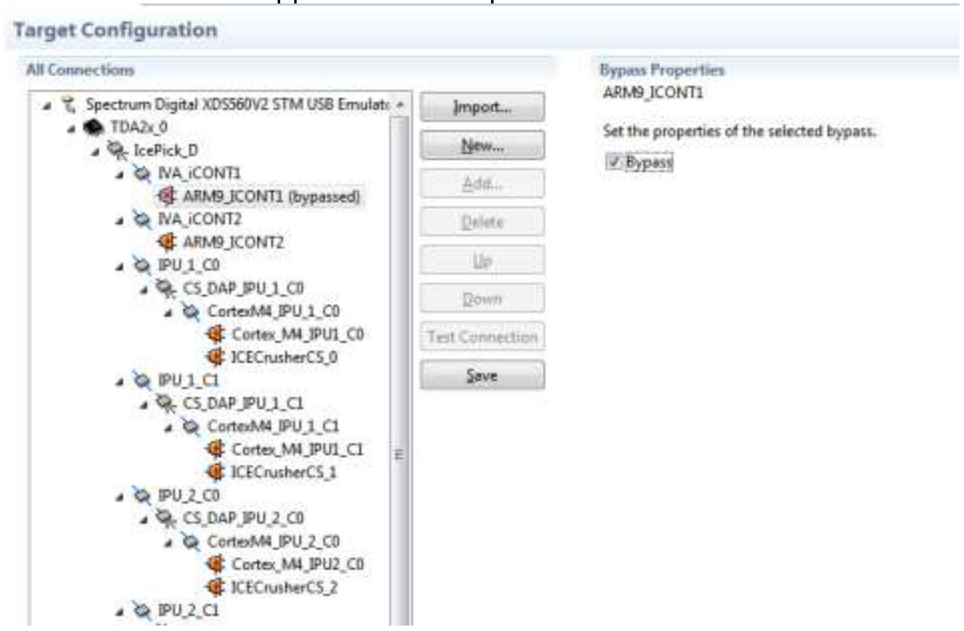
Test Connection

To test a connection, all changes must hav configuration file contains no errors and th

Alternate Communication

d. ByPass unused cores. Click on the core which needs to be bypassed and check ByPass under Bypass Properties.

The settings is under advance setup tab. Following image is example for TDA2x. Similar applies for other platforms.



Target Configuration

All Connections

- Spectrum Digital XDS560V2 STM USB Emulat
  - TDA2x_0
    - IcePick_D
      - IVA_iCONT1
        - ARM9_ICONT1 (bypassed)
      - IVA_iCONT2
        - ARM9_ICONT2
      - IPU_1_C0
        - CS_DAP_IPU_1_C0
          - CortexM4_IPU_1_C0
            - Cortex_M4_IPU1_C0
            - ICECrusherCS_0
      - IPU_1_C1
        - CS_DAP_IPU_1_C1
          - CortexM4_IPU_1_C1
            - Cortex_M4_IPU1_C1
            - ICECrusherCS_1
      - IPU_2_C0
        - CS_DAP_IPU_2_C0
          - CortexM4_IPU_2_C0
            - Cortex_M4_IPU2_C0
            - ICECrusherCS_2
      - IPU_2_C1

Bypass Properties

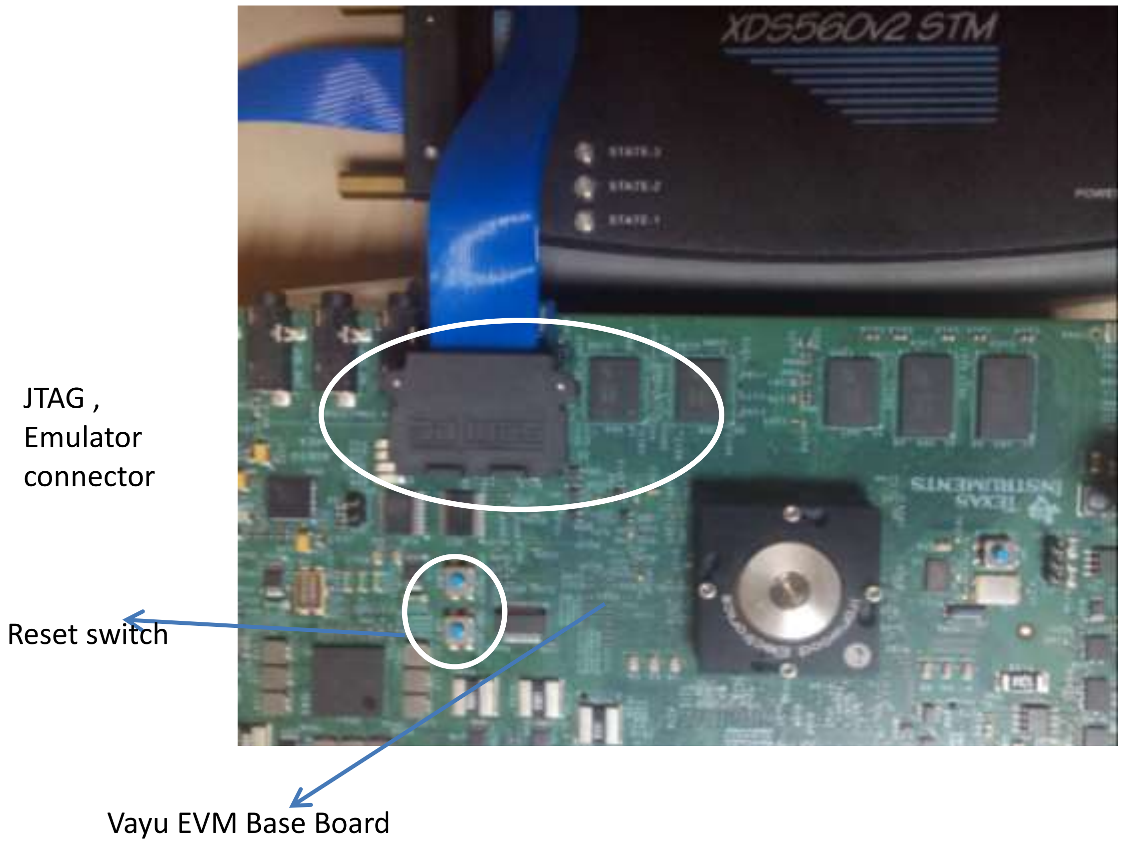
ARM9_ICONT1

Set the properties of the selected bypass.

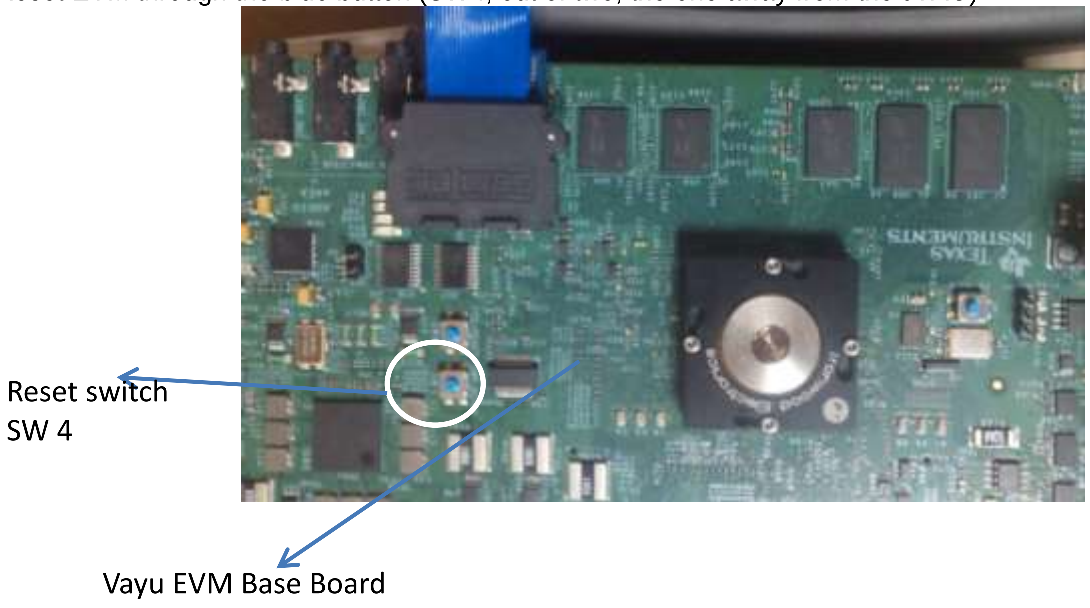
[✓] Bypass

3. Connect JTAG to the board.

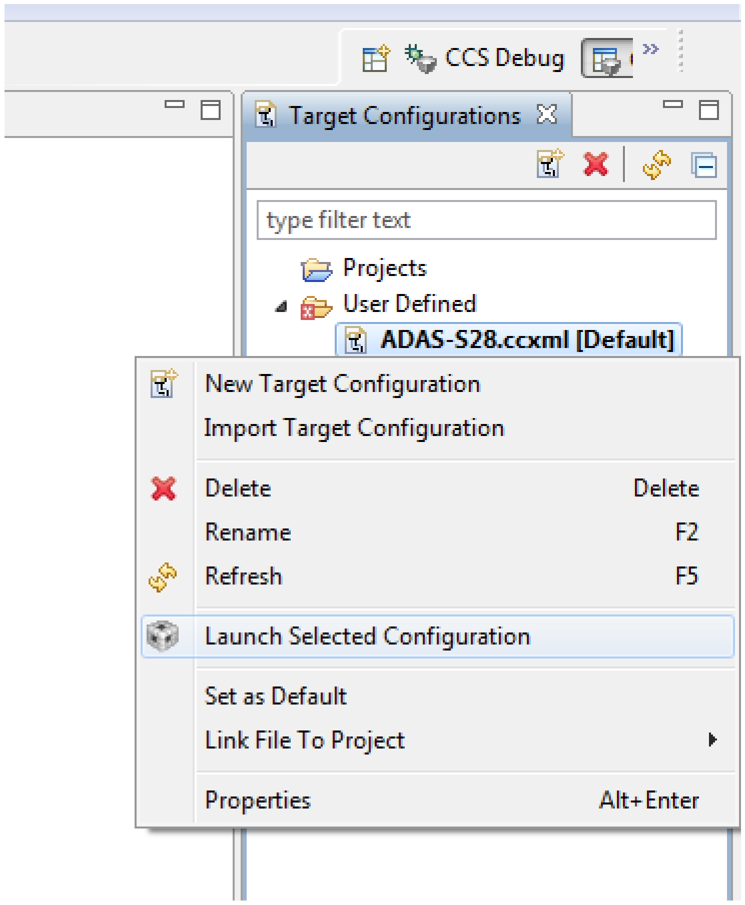
IMPORTANT NOTE: There are two JTAG connectors on the board. The one shown below MUST be used for CCS debug.

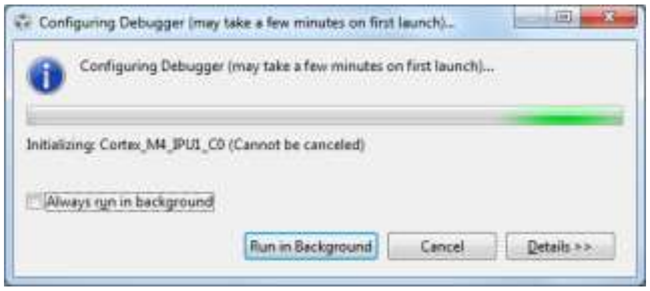


4. Reset EVM through the blue button (SW4, out of two, the one away from the JTAG).



5. Now launch the previously created TDA2xx Target Configuration.




Configuring Debugger (may take a few minutes on first launch)...

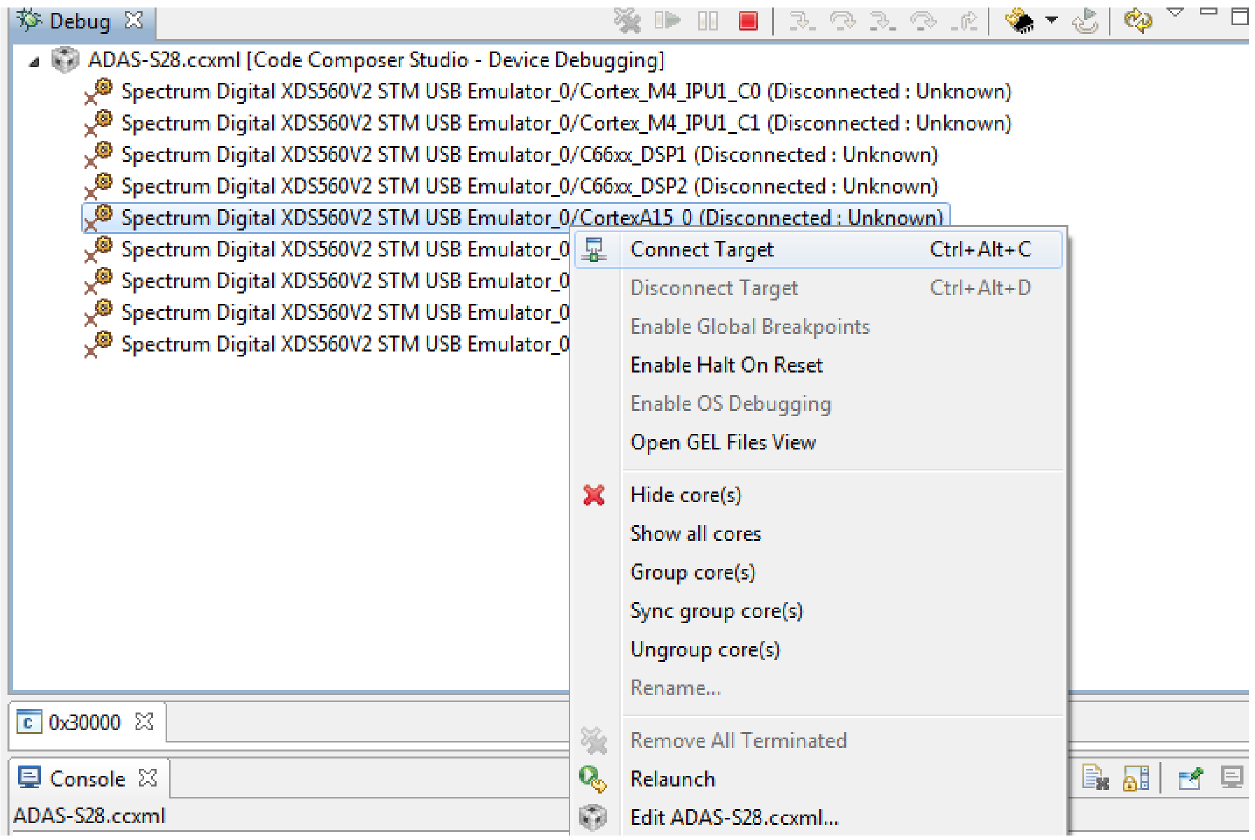
Configuring Debugger (may take a few minutes on first launch)...

Initializing: Cortex_M4_IPU1_C0 (Cannot be canceled)

6. Once the target configuration is launched successfully, the following log should be observed on the CCS console:

```
CortexA15_1: GEL Output: --->>> DRA7xx Cortex A15 Startup Sequence DONE! <<<---
```

7. Connect to core CortexA15_0.

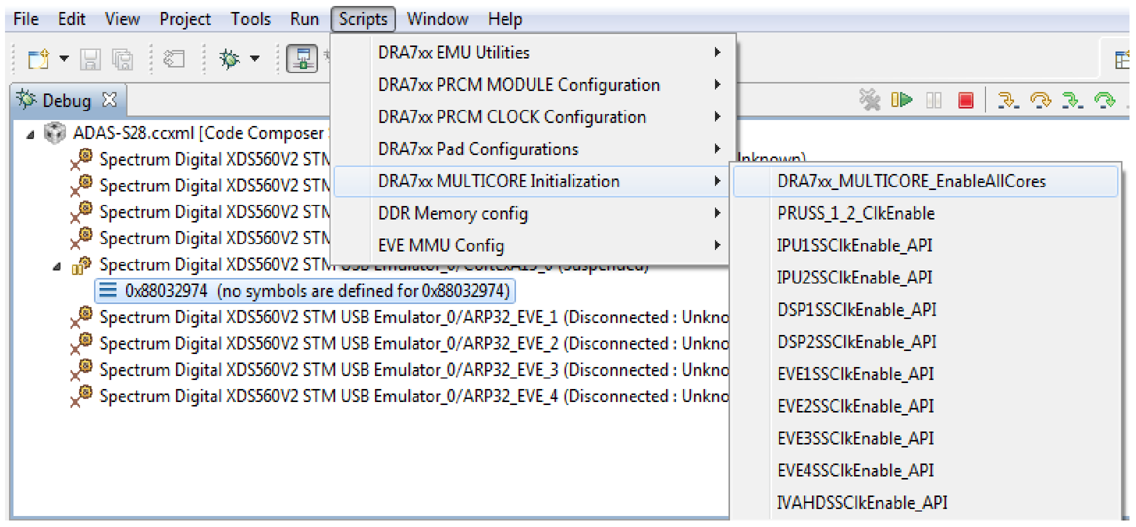


```
ADAS-S28.ccxml [Code Composer Studio - Device Debugging]
  Spectrum Digital XDS560V2 STM USB Emulator_0/Cortex_M4_IPU1_C0 (Disconnected : Unknown)
  Spectrum Digital XDS560V2 STM USB Emulator_0/Cortex_M4_IPU1_C1 (Disconnected : Unknown)
  Spectrum Digital XDS560V2 STM USB Emulator_0/C66xx_DSP1 (Disconnected : Unknown)
  Spectrum Digital XDS560V2 STM USB Emulator_0/C66xx_DSP2 (Disconnected : Unknown)
  Spectrum Digital XDS560V2 STM USB Emulator_0/CortexA15_0 (Disconnected : Unknown)
  Spectrum Digital XDS560V2 STM USB Emulator_0
  Spectrum Digital XDS560V2 STM USB Emulator_0
  Spectrum Digital XDS560V2 STM USB Emulator_0
  Spectrum Digital XDS560V2 STM USB Emulator_0
ADAS-S28.ccxml
```

8. On successful connect, the following log appears on CCS console:

```
CortexA15_0: GEL Output: --->>> DRA7xx Target Connect Sequence DONE !!!!! <<<---
```

9. Select CortexA15_0, navigate to Scripts->DRA7xx MULTICORE Initialization DRA7xx_MULTICORE_EnableALLCores



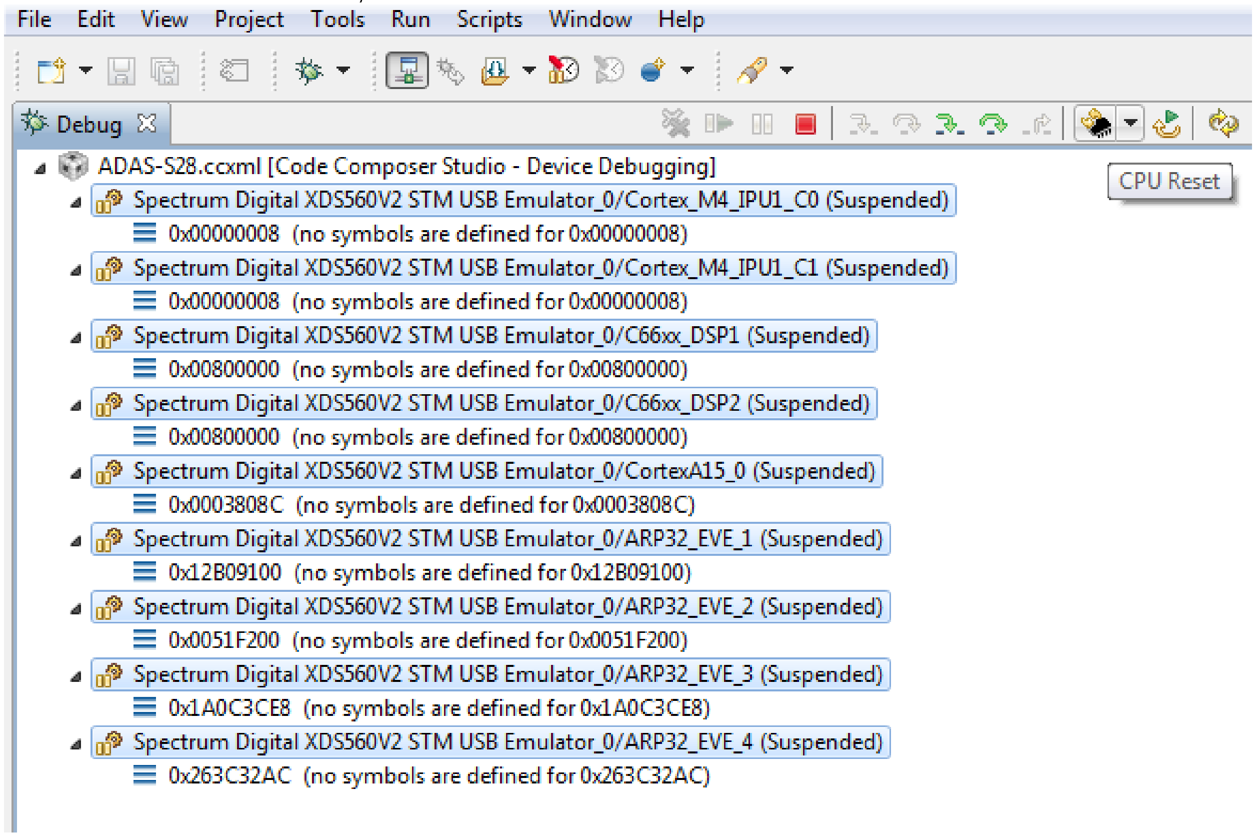
```
ADAS-S28.ccxml [Code Composer
  Spectrum Digital XDS560V2 STM
  Spectrum Digital XDS560V2 STM
  Spectrum Digital XDS560V2 STM
  Spectrum Digital XDS560V2 STM
  Spectrum Digital XDS560V2 STM
    0x88032974 (no symbols are defined for 0x88032974)
  Spectrum Digital XDS560V2 STM USB Emulator_0/ARP32_EVE_1 (Disconnected : Unkno
  Spectrum Digital XDS560V2 STM USB Emulator_0/ARP32_EVE_2 (Disconnected : Unkno
  Spectrum Digital XDS560V2 STM USB Emulator_0/ARP32_EVE_3 (Disconnected : Unkno
  Spectrum Digital XDS560V2 STM USB Emulator_0/ARP32_EVE_4 (Disconnected : Unkno
```

10. On successful script execution, the following log appears on CCS console:

```
CortexA15_0: GEL Output: --->>> PRUSS 1 and 2 Initialization is in complete ... <<<---
```

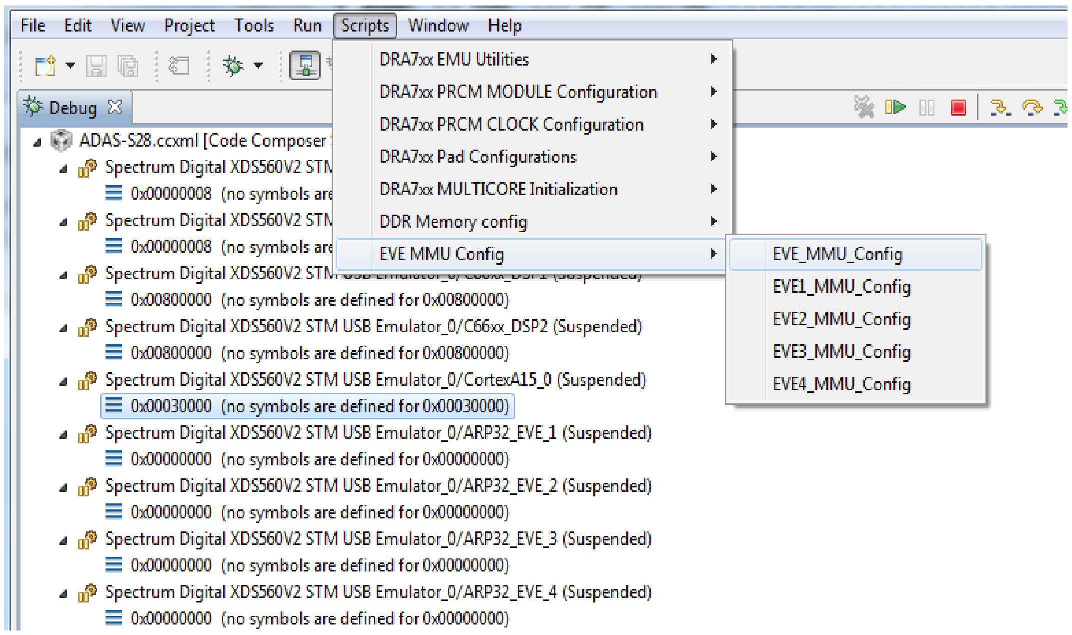
11. Now connect the core shown below,
    ARP32_EVE_1, ARP32_EVE_2, ARP32_EVE_3, ARP32_EVE_4
    C66xx_DSP1, C66xx_DSP2, Cortex_M4_IPU1_C0, Cortex_M4_IPU1_C1.

12. Once the cores are connected, do CPU Reset for all the cores.



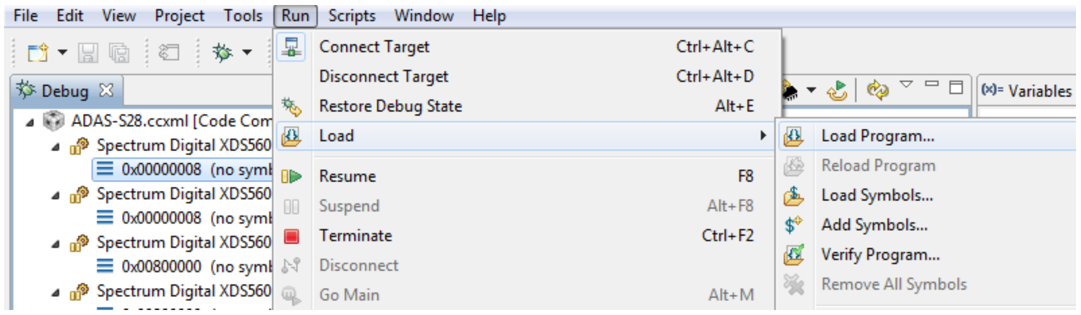
```
ADAS-S28.ccxml [Code Composer Studio - Device Debugging]
  Spectrum Digital XDS560V2 STM USB Emulator_0/Cortex_M4_IPU1_C0 (Suspended)
    0x00000008  (no symbols are defined for 0x00000008)
  Spectrum Digital XDS560V2 STM USB Emulator_0/Cortex_M4_IPU1_C1 (Suspended)
    0x00000008  (no symbols are defined for 0x00000008)
  Spectrum Digital XDS560V2 STM USB Emulator_0/C66xx_DSP1 (Suspended)
    0x00800000  (no symbols are defined for 0x00800000)
  Spectrum Digital XDS560V2 STM USB Emulator_0/C66xx_DSP2 (Suspended)
    0x00800000  (no symbols are defined for 0x00800000)
  Spectrum Digital XDS560V2 STM USB Emulator_0/CortexA15_0 (Suspended)
    0x0003808C  (no symbols are defined for 0x0003808C)
  Spectrum Digital XDS560V2 STM USB Emulator_0/ARP32_EVE_1 (Suspended)
    0x12B09100  (no symbols are defined for 0x12B09100)
  Spectrum Digital XDS560V2 STM USB Emulator_0/ARP32_EVE_2 (Suspended)
    0x0051F200  (no symbols are defined for 0x0051F200)
  Spectrum Digital XDS560V2 STM USB Emulator_0/ARP32_EVE_3 (Suspended)
    0x1A0C3CE8  (no symbols are defined for 0x1A0C3CE8)
  Spectrum Digital XDS560V2 STM USB Emulator_0/ARP32_EVE_4 (Suspended)
    0x263C32AC  (no symbols are defined for 0x263C32AC)
```

13. For VisionSDK 2.6 and older, on CortexA15_0, run the GEL “Scripts -> EVE MMU Config -> EVE_MMU_Config”.
    IMPORTANT NOTE: If this step is not done you will not be able to load executables on the EVE cores



```
ADAS-S28.ccxml [Code Composer
  Spectrum Digital XDS560V2 STM
    0x00000008  (no symbols are
  Spectrum Digital XDS560V2 STM
    0x00000008  (no symbols are
  Spectrum Digital XDS560V2 STM
    0x00800000  (no symbols are defined for 0x00800000)
  Spectrum Digital XDS560V2 STM USB Emulator_0/C66xx_DSP2 (Suspended)
    0x00800000  (no symbols are defined for 0x00800000)
  Spectrum Digital XDS560V2 STM USB Emulator_0/CortexA15_0 (Suspended)
    0x00030000  (no symbols are defined for 0x00030000)
  Spectrum Digital XDS560V2 STM USB Emulator_0/ARP32_EVE_1 (Suspended)
    0x00000000  (no symbols are defined for 0x00000000)
  Spectrum Digital XDS560V2 STM USB Emulator_0/ARP32_EVE_2 (Suspended)
    0x00000000  (no symbols are defined for 0x00000000)
  Spectrum Digital XDS560V2 STM USB Emulator_0/ARP32_EVE_3 (Suspended)
    0x00000000  (no symbols are defined for 0x00000000)
  Spectrum Digital XDS560V2 STM USB Emulator_0/ARP32_EVE_4 (Suspended)
    0x00000000  (no symbols are defined for 0x00000000)
```

14. On the cores load the binaries as mentioned below



```
ADAS-S28.ccxml [Code Com
  Spectrum Digital XDS560
    0x00000008  (no symb
  Spectrum Digital XDS560
    0x00000008  (no symb
  Spectrum Digital XDS560
    0x00800000  (no symb
  Spectrum Digital XDS560
```

On ARP32_EVE_4, load the binary, “vision_sdk_arp32_4_release.xearp32F”.
On ARP32_EVE_3, load the binary, “vision_sdk_arp32_3_release.xearp32F”.
On ARP32_EVE_2, load the binary, “vision_sdk_arp32_2_release.xearp32F”.
On ARP32_EVE_1, load the binary, “vision_sdk_arp32_1_release.xearp32F”.
On C66xx_DSP2, load the binary, “vision_sdk_c66xdsp_2_release.xe66”.
On C66xx_DSP1, load the binary, “vision_sdk_c66xdsp_1_release.xe66”.
On Cortex_M4_IPU1_C0, load the binary, “vision_sdk_ipu1_0_release.xem4”.
On Cortex_M4_IPU1_C1, load the binary, “vision_sdk_ipu1_1_release.xem4”.
On CortexA15_0, load the binary, "vision_sdk_a15_0_debug.xa15fg”

IMPORTANT NOTE: Binary for Cortex_M4_IPU1_C0 MUST be loaded before Cortex_M4_IPU1_C1 since IPU1-0 does MMU config for the complete IPU1 system. Other binaries can be loaded in any order.

3.7 Run the demo

3.7.1 Single channel demos with HDMI input

IMPORTANT NOTE: To demonstrate better output all single channel usecases that require HDMI input should use video clips mentioned in the table below. These clips are part of VISION_SDK_02.XX.XX.XX_INPUTS.tar.gz

Usecase No. "Runtime Menu"	**Usecase**	**Input clip to be played by HDMI player**
7	1CH VIP capture + Sparse Optical Flow (EVE1) + Display	Clip2
b	b: 1CH VIP capture (HDMI) + Lane Detect (DSP1) + Display	Clip1
c	c: 1CH VIP capture (HDMI) + SOF (EVE1) + SFM (DSP1) + Display	Clip2
d	d: 1CH VIP capture (HDMI) + Traffic Light Recognition (TLR) (DSP1) + Display	Clip2
e	e: 1CH VIP capture (HDMI) + Pedestrian, Traffic Sign, Vehicle Detect 2 (EVE1 + DSP1) + Display	Clip2
f	f: 1CH VIP capture (HDMI) + FrontCam Analytics 2 (PD+TSR+VD+LD+TLR+SFM) (DSPx, EVEx) + Display (HDMI)	Clip3

SFM_POSE.bin – SFM (Usecase 'c') and EUNCAP demo (Usecase 'f') needs SFM_POSE.bin on the **SD card**. It is part of VISION_SDK_02.XX.XX.XX_INPUTS.tar.gz

3.7.2 Steps to run

1. Power-on the Board after loading binaries by (SD, QSPI, NOR or CCS) and follow Uart settings to setup the console for logs and selecting demo.

2. For HDMI as input select capture source as HDMI "s: System Settings"-> "Capture Settings" -> "2: HDMI Capture 1080P60"

3. Select demo required from the menu by keying in corresponding option from uart menu.

IMPORTANT NOTE: Make sure you select SCV (1Ch VIP capture) use-case or LVDS MCV (4CH LVDS VIP capture) use-case depending on the hardware you run the application.

IMPORTANT NOTE: For AVB MCV Demo Ethernet port must be connected as shown in SCV/AVB Use-case Hardware Setup

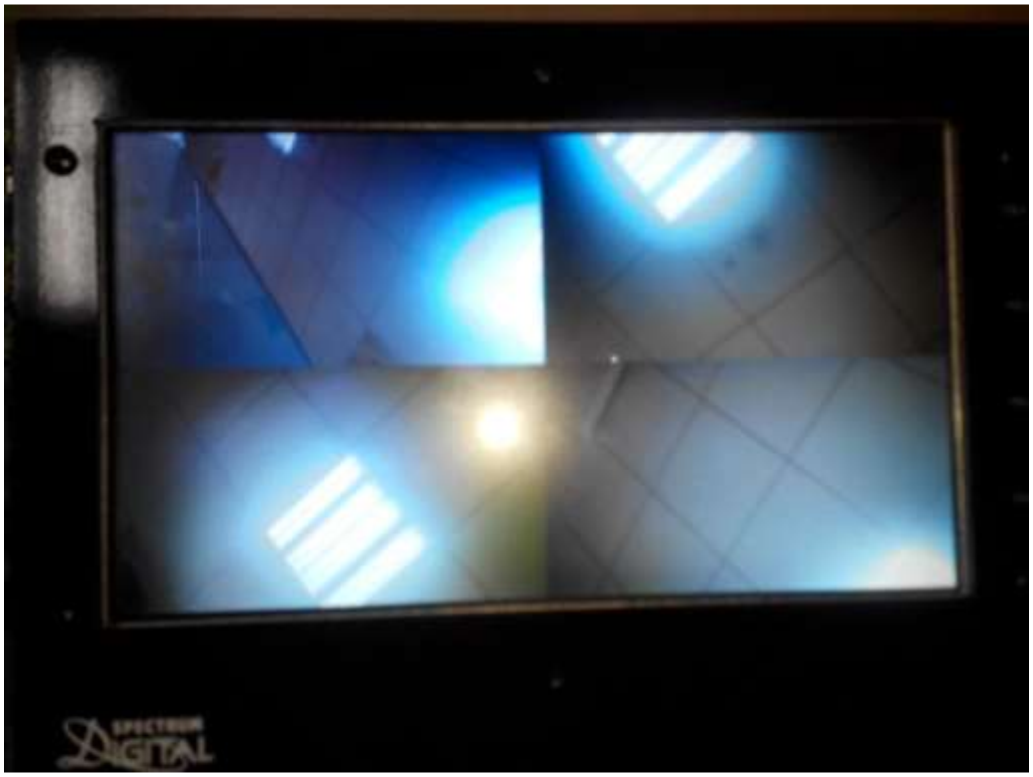
After successful initialization of the use-case, you will see video been display on the LCD as shown below,

a. SCV use-cases:

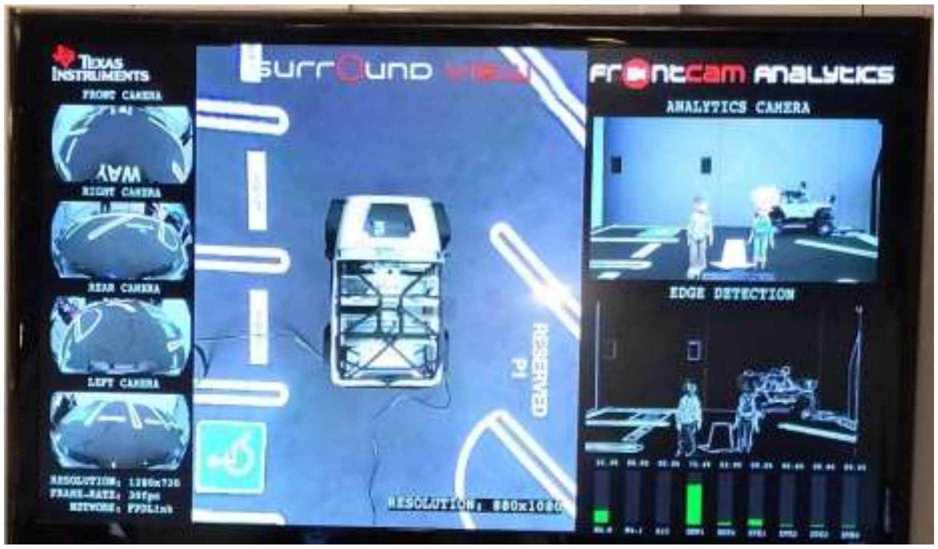
b. EDGE Detect use-case:

c. Dense Optical Flow Usecase

d. LVDS Usecase



e. Surround View + Edge Usecase



f. AVB Usecase

Data is streamed from Linux talker (Ref: AVB Used guide for building talker binaries)

VMware can also be used but the throughput of talker is not as desired and depends on PC configurations.

NOTE: AVB Talker code present at **ti_components\networking\avbtp_0_09_00_01\utils\avbtp_talker is replaced with avbtp_00_06_02_05\utils\avbtp_talker.**

4 Frequently Asked Questions

4.1 Hardware Board Related FAQs

Q. I selected a use-case and it hangs during initialization
Make sure you are running the SCV use-case on SCV hardware setup and LVDS use-case on LVDS hardware setup. For LVDS hardware use 12V and 7A supply only.
Q. Even after following all the steps I see a incorrect / distorted image on the LCD ?
The issue could be with the CPLD programming on the Vision Application Board. Contact TI local support for reprogramming the CPLD image. The CPLD controls the how the sensor data lines are routed to the VIP port of the SoC. By default CPLD is programmed for the SCV use-case.
Q. Sometimes I see a message "LCD not connected" on the UART console after running the use-case but I see normal display on the LCD
Ignore this message, the software / hardware is falsely reporting LCD is not connected.
Q. LVDS use-case init hangs on first try after power-cycle
Make sure all Sensors are connected as mentioned in earlier section. For LVDS hardware use 12V and 7A supply only.
Q. Sometimes LVDS setup hangs during use-case initialization second time after power-cycle
Suspecting this to be a board connectivity issue. Tighten the application / deserializer boards, Power cycle and retry. For LVDS hardware use 12V and 7A supply only.
Q. After CCS reload without power-cycle LVDS setup hangs during use-case initialization
Suspecting this to be a board connectivity issue. Power cycle, reset the board and retry.
Q. Sometimes when connecting to A15 via CCS we get a message "Router sub-path could not be accessed"
Power cycle, reset the board and JTAG emulator and retry.
Q. I select the LCD demo but I see only colors/color bar no display , logs show proper numbers
Please verify the LCD selected, is it 7" or 10". If its wrongly selected it LCD runs the default test.

5 Revision History

Version	Date	Revision History
1.0	31st March 2016	Updated for release 2.9

««« § »»»