IPNC Reference Design Kit Installation Guide

User's Guide

Literature Number: SPRUI83
September 2016

Contents

1 System Installation 5
- 1.1 Installing the Release Package 6
- 1.2 Hardware Installation 7
- 1.3 Software Installation 10
- 1.4 Flashing NAND Flash 11

2 Building IPNC Application 17
- 2.1 IPNC Application Build Procedure 17
- 2.2 Building U-boot and Kernel Image 21

3 Running IPNC Application 21
- 3.1 Running IPNC Application 21
- 3.2 Troubleshooting 24
- 3.3 Network Connections Setting 24
- 3.4 Streaming Demo on Web Browser 27
- 3.5 RTSP Streaming Demo on VLC 32
- 3.6 End-to-End Low Latency Measurement 36
- 3.7 ARM® CPU Loading Measurement 39
- 3.8 MJPEG Rate Control API Details 40

Copyright © 2016, Texas Instruments Incorporated

List of Figures

1 IPNC_RDK Directory Tree 7
2 Full Setup Required for Testing With Component Video Input 7
3 Analog Component Cable Connected to EVM 8
4 Serial Cable Connected to EVM 9
5 XDS510 USB JTAG 9
6 JTAG Board Connector 10
7 Flashing Using CCS 11
8 IPNC RDK Directory Structure 17
9 Boot Mode Order 24
10 PC Connected to DM38x EVM 25
11 PC Connected to IPNC Through a Router 25
12 Remote Monitor With Web Browser 26
13 Network Data Display 26
14 Security Warning 27
15 Downloaded ActiveX Controls (32 bit) 28
16 GoDB Tech Private Limited 28
17 Manage Add-Ons 29
18 Restart Browser 29
19 Install the Software 30
20 Main Login Screen 31
21 Live Video 32
22 Open Network Stream 33
23 Configure Network Protocol 34
24 Preferences 35
25 Overlay Video Output 35
26 Input/Codecs 36
27 End-to-End Latency 36
28 Tools → Preferences 37
29 Reset Preferences 37
30 Overlay Video Output 38
31 Input/Codecs 38
32 Latency Measurement 39

Copyright © 2016, Texas Instruments Incorporated

List of Tables

1 Acronyms and Description 5
2 IPNC_RDK Release Directories 6
3 Source Directory 6
4 Setting the Correct Environment 18
5 MAKE Command Options 20

Copyright © 2016, Texas Instruments Incorporated

User's Guide
SPRUI83–September 2016

IPNC Reference Design Kit Installation Guide

This document provides a brief description of the purpose and construction of the Internet Protocol (IP) Network Camera (using component video input as source), along with hardware and software environment requirements in the context of IP Network Camera deployment.

1 System Installation

Table 1. Acronyms and Description

Acronym/Abbreviation	Description
I2C	Inter-Integrated Circuit
DHCP	Dynamic Host Configuration Protocol
FTP	File Transfer Protocol
GPIO	General Purpose I/O
HTTP	Hypertext Transfer Protocol
MMC	Multimedia Card
NTP	Network Time Protocol
OSS	Open Source Software
SPI	Serial Peripheral Interface
PoE	Power over Ethernet
RTP	Real-time Transport Protocol
RTSP	Real Time Streaming Protocol
SSL	Secure Socket Layer
SMTP	Simple Mail Transfer Protocol
UBL	User Boot Loader
UDP	User Datagram Protocol
UART	Universal asynchronous receiver/transmitter
UPnP	Universal Plug and Play
VPFE	Video Processing Front End
VPBE	Video Processing Back End
VGA	640 x 480 resolution
WDT	Watch Dog Timer

Code composer studio is a trademark of Texas Instruments.
ARM, Cortex are registered trademarks of ARM Limited.
Linux is a registered trademark of Linux Torvalds in the U.S. and other countries.

Copyright © 2016, Texas Instruments Incorporated

1.1 Installing the Release Package

The release contains the following components:

- IPNC_RDK_V3.8.1.tar.gz:
  This main file for the IPNC reference design kit (RDK) installation contains the following:
  - Pre-built binary files for the DM38x platform
  - Set of documents
  - Source code required to build IPNC RDK
  - Hardware Package with build of material (BoM), schematics and gerber file
  - Utility files like GEL, Nand-programmer, and so forth
- *IPNC Reference Design Kit Installation Guide* (SPRUI83)
  Guide to enable installation and evaluation of IPNC features
- IPNC_RDK_Release_Notes.pdf. The release notes will be available online for each software release.
  Details on features added, known issues, version details, bug fixes, and so forth

1.1.1 Installation Procedure

1. Extract/untar the IPNC_RDK_V3.8.1.tar.gz file at the install directory: tar –zxvf IPNC_RDK_V3.8.1.tar.gz.
2. Table 2 shows a list of release directories that should be available after the above installation procedure.

Table 2. IPNC_RDK Release Directories

Files/Directories	Description
Collaterals	Various application reports, user's guides, integration guide, data sheets and training on multi-channel framework (McFW)
Binaries	Pre-built binaries for executing the IPNC RDK application on DM38x IPNC - includes uImage, uBoot and file system for two different configurations: low_power and full_feature.
Hardware	Contains gerber, schematics and hardware documents
Utils	Contains NAND flash utilities, such as NAND flash writer and Gel files
Source	IPNC RDK source code resides here

3. The source directory is a base work directory containing source code for developing the IPNC RDK. The description of the folders under source directory is shown in Table 3.

Table 3. Source Directory

Folders	Description
ipnc_rdk/target	Target File System. This can be exported from the NFS server
ipnc_rdk/tftp	Stores uImage, uBoot and filesys that can be exported from the TFTP server
ti_tools	All components and build tools needed for software development of IPNC RDK
ipnc_rdk/ipnc_app	Core IPNC application code
ipnc_rdk/ipnc_mcfw	Source code for multi-channel framework

Copyright © 2016, Texas Instruments Incorporated

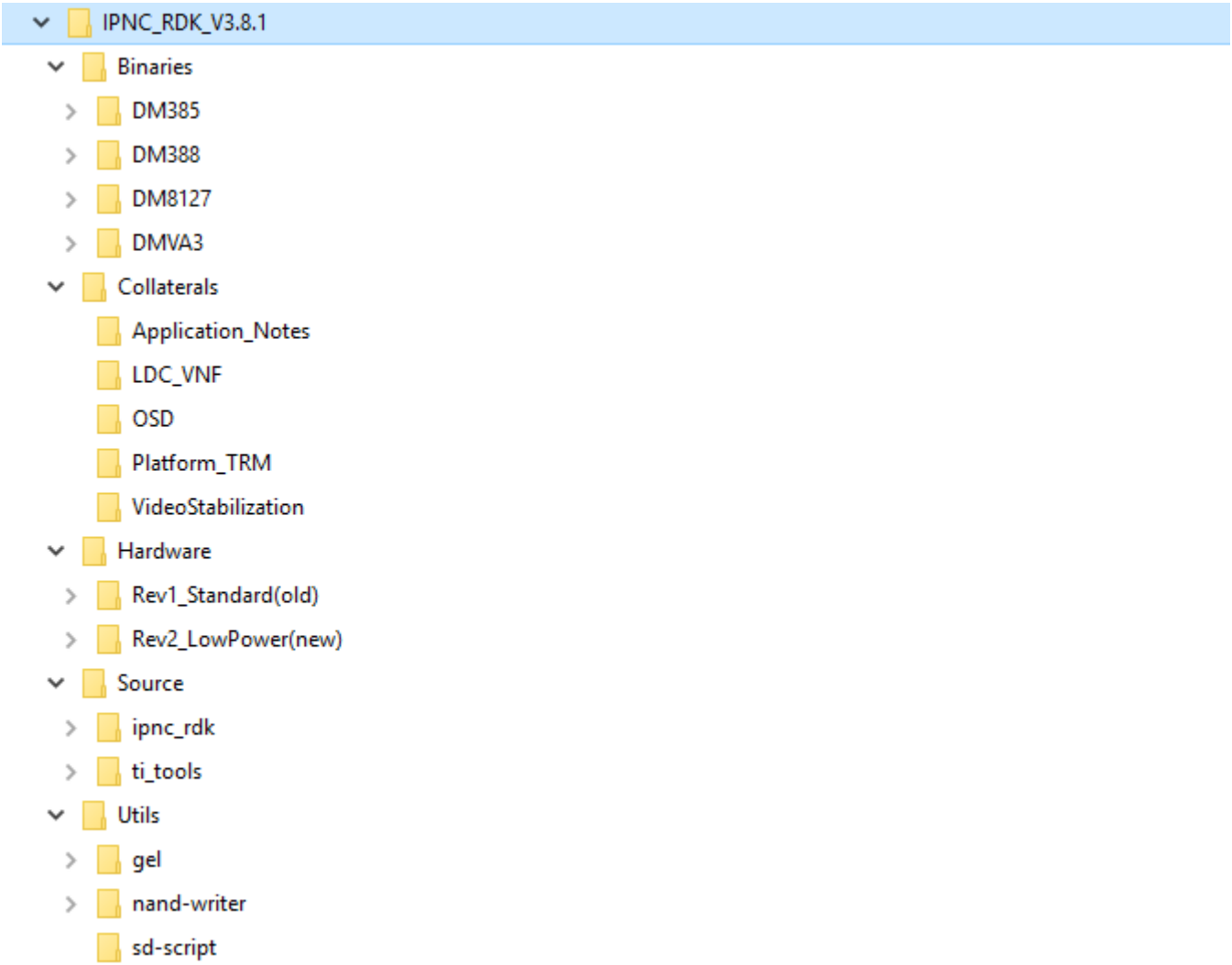

Figure 1. IPNC_RDK Directory Tree

1.2 Hardware Installation

1.2.1 Full System Setup Overview

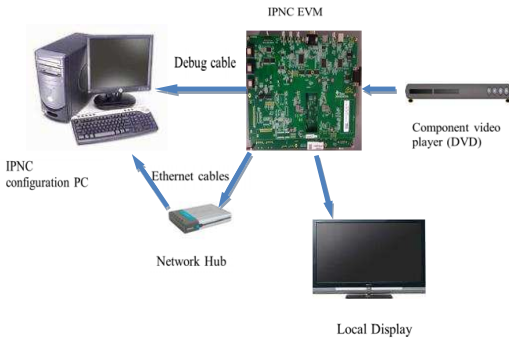

Figure 2. Full Setup Required for Testing With Component Video Input

Copyright © 2016, Texas Instruments Incorporated

Steps to setup testing environment with component video input source:

1. Connect analog component cable from component video player (DVD Player) to DM38x EVM as shown in Figure 2 and Figure 3.
2. Connect debug (serial) cable to the configuration PC as shown in Figure 2 and Figure 4.
3. Connect HDMI cable from base-board (on DM38x EVM) to 'Local Display' as shown in Figure 2.
4. Connect DM38x EVM and 'IPNC configuration PC' through Ethernet cable directly or using network hub as shown in Figure 2.
5. Make sure that 'IPNC Configuration PC' and EVM are connected in same network to ensure proper network connectivity between the two.

1.2.2 Connecting Analog Component Video Cable and Serial Cable

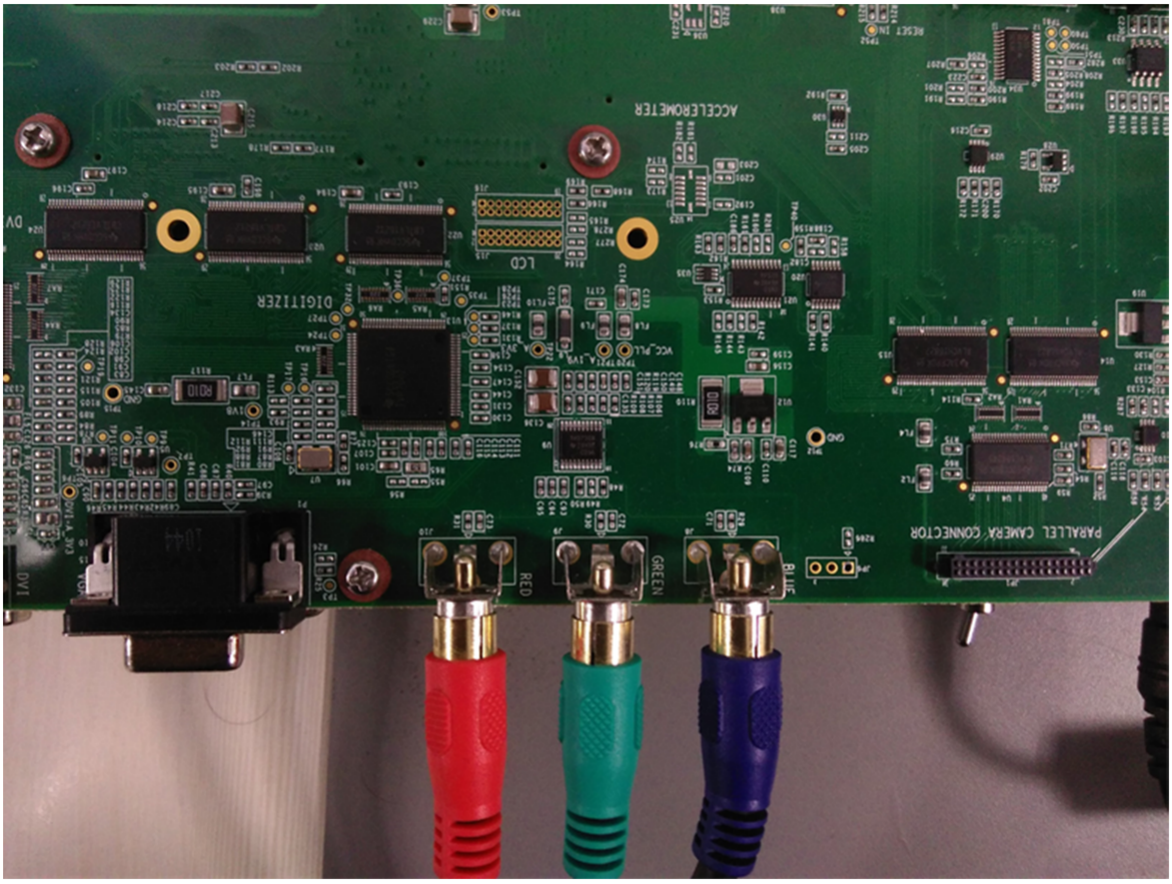

Figure 3. Analog Component Cable Connected to EVM

Copyright © 2016, Texas Instruments Incorporated

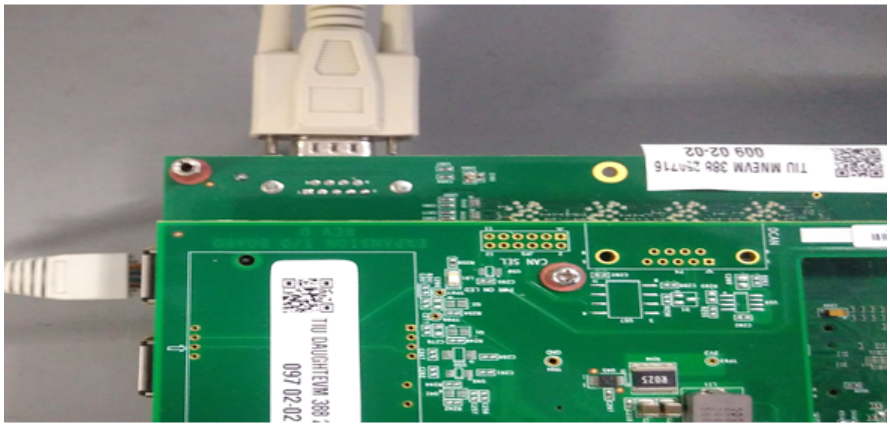

Figure 4. Serial Cable Connected to EVM

1.2.3 Connecting JTAG to Hardware

This section provides information about the two types of JTAG, their uses with the debug board, and their hardware jumper setting.

Figure 5. XDS510 USB JTAG

Copyright © 2016, Texas Instruments Incorporated

Figure 6. JTAG Board Connector

NOTE: Figure 5 depicts XDS510USB JTAG. However, it is recommended to use the XDS560V2 STM JTAG emulator for better reliability and higher speeds.

1.3 Software Installation

1.3.1 Installation of the Basic Tools and Host Software

Install the following items prior to installing the IPNC specific software:

- Code composer studio™ (CCS) 4.2.x/5.x or above installed on IPNC Configuration PC (running Windows XP service pack 3 or above) – This is necessary only for CCS-based NAND programming or for debugging the application on the Cortex®-M3 or DSP core.
- NFS server that can export IPNC file system. Use the following steps to install it (if not already installed) in the UBUNTU 10.04 Linux® host machine.

```
#sudo apt-get install nfs-kernel-server
```

- TFTP server (for example, http://tftpd32.jounin.net/). Use `#sudo apt-get install tftpd-hpa` to install this in the UBUNTU 10.04 Linux host machine.
  - The files that need to be downloaded on the DM38x EVM via tftp first need to be copied or moved to <TFTP DIRECTORY>. The default location of <TFTP DIRECTORY> can be found in the /etc/default/tftpd-hpa location. You can either copy the binaries to this default <TFTP DIRECTORY> or change the path of the <TFTP_DIRECTORY> to point to the binaries.
- To build the code, install FLEX and BISON on the Linux host machine.
  - sudo apt-get install bison
  - sudo apt-get install flex
- The following package needs to be installed for a successful build:
  - Install the acl library (needed to build file system)
    - sudo apt-get install libacl1-dev
  - Install the zlib library (needed to build file system)
    - sudo apt-get install zlib1g-dev
  - Install the lzo library (needed to build file system)
    - sudo apt-get install liblzo2-dev
  - Install the uuid library (needed to build file system)
    - sudo apt-get install uuid-dev

Copyright © 2016, Texas Instruments Incorporated

- For creating the UBIFS file system: check that mtd-utils is installed.
  - See MTD_Utilities
  - You can install the mtd-utils on the UBUNTU 10.04 linux host machine as:
    - #sudo apt-get install mtd-utils
  - mtd-utils are also provided with the file system along with the release source package
    - This is located at the Source/ipnc_rdk/target folder. Untar the file under root privileges (filesys_ipnc_rdk.tar.gz).

These steps could vary on different variants of Linux distributions.

1.3.2 Installation of the File System

1. Install the decompressed file system located at *<installDir>/Source/ipnc_rdk/target* on your Linux desktop.
   - `#sudo tar –zxvf filesys_ipnc_dm81xx.tar.gz`
   - `#chown –R <useracct>:<useracct> ./*`

   NOTE: Root privileges are necessary to untar the file system or an error could occur and untar could be incomplete, which makes the build non-functional.
2. For NFS usage, set the export path in the Linux host machine where the file system is mounted.
3. Modify /etc/exports file as mentioned with root privileges.
   - /home/user/workdir/filesys *(rw,no_root_squash,no_all_squash,sync)
4. Run this command to export the file system:
   - /usr/sbin/exportfs –av
5. Start NFS service using this command:
   - /etc/init.d/nfs-kernel-server restart

1.4 Flashing NAND Flash

In DM38x IPNC, the ROM code serves as the first stage bootloader. The second and the third stage bootloaders are based on U-Boot. When referring to the binaries, the binary for the second stage is referred to as uboot.min and the binary for the third stage is referred to as simply uboot.

Flashing uboot on NAND for the first time can be done using CCS, SD boot or universal asynchronous receiver/transmitter (UART) boot. Once uboot is flashed to NAND, further updates to the uboot or flashing of the kernel binary and file system can be done via U-Boot itself. This section describes the methods for transferring images to NAND using CCS, Uboot and SD boot.

1.4.1 Flashing Using CCS

Use the following steps to burn U-Boot for DM38x IPNC:

Figure 7. Flashing Using CCS

Copyright © 2016, Texas Instruments Incorporated

1. Connect JTAG to the IPNC EVM board.
2. Start Code Composer Studio v5.0.1.
   (a) Click on 'Start' → 'All Programs' → 'Texas Instruments' → 'Code Composer Studio v5.0.1' → 'Code Composer Studio v5'.
3. At the 'Select a Workspace' prompt, pick a convenient workspace directory to store the code composer project files and context. Click 'OK'.
4. Creating New target configuration:
   (a) Click on 'View' → 'Target Configurations'. On the Target Configurations (Tabbed Window), right click to get a menu.
   (b) Select 'New Target Configuration'. Enter the Target Configuration name (for example, TI814xPF.ccxml). In the New Target Configuration window, under the Connection tab choose ‘Spectrum Digital XDS560V2 STM USB Emulator’. Under 'Device', choose ‘TI814x’. Save the Configuration. TI814xPF.ccxml appears in the Target Configurations (Tabbed Window). For "Spectrum Digital XDS510USB Emulator" users, select TI814x (no STM, ETB only) under 'Devices'.
   (c) On the Target Configurations (Tabbed Window), right click on the TI814xPF.ccxml and select ‘Launch Selected Configuration’. Wait for Launch operation to complete (bottom right corner of the CCS App Window shows the status) and the 'Debug' window appears.
   (d) In the 'Debug' window, you should be able to see ‘TI814xPF.ccxm [Code Composer Studio – Device Debugging]’. On expanding, all cores open up. Make sure that Cortx-A8, CortexM3_ISS_0 (M3_DSS), and CortexM3_RTOS_0 (M3_Video) are among the list and that all of the cores show ‘Disconnected: Unknown’ as the status.

NOTE: Assuming XDS560 is connected to the EVM, make sure that the XDS560 USB/Ethernet driver is installed.

If 'Launch selected configuration' fails, disconnect and re-connect JTAG to PC and try again.

5. Loading and executing GEL File :
   (a) After status of A8 is shown as suspended, right click on the core and click on ‘Open GEL'.
   (b) A GEL files tab should open in CCS. Right click on the GEL files window and click on ‘Load GEL…’. The release package contains the gel file IPNC_A8_DDR3.gel at the location *<installDir>/Utils/gel*. Load this file on A8.
6. Connecting to Host: Connect to the Cortex A8 processor, which is the main host on TI812x/TI814x. Right click on the Cortex-A8 (Host) core and select ‘Connect Target’.
7. Loading and running A8 binary: Do a reset of the A8 core by clicking on the CPU reset icon in CCS (alternately Ctrl + Shift + R) with A8 selected. In CCS, (after highlighting the A8 processor in the debug window), click on the following menu item 'Run' → 'Load' → 'Load program' and select the nandflash-writer.out program that is included with this package at the *<installDir>/Utils/bin* location.
8. Start the application from 'Run' → 'Resume' or select 'F8'.

NOTE: If this fails, pause the program and reload the 'nandflash-writer.out'. The flash writer will be loaded successfully and run the program.

9. After success of this step 8, you should be able to see the following path on the CCS console.

```
Choose your operation
Enter 1 ---> To Flash an Image
Enter 2 ---> To ERASE the whole NAND
Enter 3 ---> To EXIT
```

   Select '1' to flash the binary to NAND.

```
Enter image file path
```


Copyright © 2016, Texas Instruments Incorporated

1.4.1.1 Flashing First Stage of U-boot on NAND

The first stage u-boot image 'u-boot.min.nand' needs to be flashed. This image should be located at: *<installDir>/Binaries/DMxxx/nand*. Copy this to a convenient location. After seeing the message on the console, enter the full path of the location where 'u-boot.min.nand' is now located.

1. When prompted, enter the offset as shown below. This offset is the start location from where the image should be flashed. Enter 0x0 for flashing the u-boot.min.nand image.

   (a) Enter offset (in hex):

2. Select ECC for flashing. Always select BCH8 for U-Boot as the ROM code uses the BCH8 ECC scheme. Enter 1 for u-boot.min.nand.

```
Choose the ECC scheme from given options
Enter 1 ---> BCH 8 bit
Enter 2 ---> HAM
Enter 3 ---> T0 EXIT
Please enter ECC scheme type :
```

3. Ensure that the flash info displayed by the tool matches the NAND flash in the EVM. After this, the tool should erase the required region in flash and then start flashing the new image.
4. Finally, you should be able to see the following message. Then, power restart the IPNC hardware.

```
Application is successfully flashed
NAND boot preparation was successful!
```

1.4.1.2 Flashing Second Stage of U-boot on NAND:

1. Assuming that you are still connected to A8, reload in CCS 'nandflash-writer.out' as mentioned in step in Section 1.4.1.
2. Start the application from 'Run' → 'Resume' or select 'F8'.
3. When prompted, select option 1. Select option 2 only if you want to erase the NAND.

```
Choose your operation

Enter 1 ---> To Flash an Image
Enter 2 ---> To ERASE the whole NAND
Enter 3 ---> To EXIT
```

4. When prompted, enter the image path to uboot.bin to flash as shown below. Provide the complete path of uboot.bin. For example, *<installDir>/Binaries/DMxxx/nand/uboot.bin*.

```
Enter image file path
```

5. When prompted, enter the offset as 0x20000 as shown below. This offset is the start location from where the uboot.bin should be flashed.

```
Enter offset (in hex):
```

6. Select ECC for flashing. Always select BCH8 for U-Boot as the ROM code uses the BCH8 ECC scheme. Enter 1 for u-boot.bin.

```
Choose the ECC scheme from given options
Enter 1 ---> BCH 8 bit
Enter 2 ---> HAM
Enter 3 ---> T0 EXIT
Please enter ECC scheme type :
```

7. Finally, you should see the following message, then power restart the IPNC hardware.

```
Application is successfully flashed
NAND boot preparation was successful!
```


Copyright © 2016, Texas Instruments Incorporated

1.4.2 Flashing U-Boot to IPNC

In this mode, the assumption is that the older version of uBoot already exists on the NAND.

1. Setup the TFTP server on the PC (assume the IP address of PC is 192.168.1.15).
2. Connect the network cable to the ethernet port of IPNC. Make sure that the PC hosting TFTP and the IPNC are in same network domain.
3. Let the first stage Uboot finish and halt Uboot at the second stage uboot.
4. You should get the prompt TI8148_IPNC.
5. Use the following commands on the uBoot second stage shell prompt to flash the first stage.

   For setting up the TFTP, the following ENV is needed:

   ```
   DM388_IPNC# setenv serverip <TFTP SERVER IP ADDRESS>
   DM388_IPNC# setenv ethaddr <DEVICE MAC ADDRESS>
   DM388_IPNC# setenv gatewayip <NETWORK GATEWAY ADDRESS>
   DM388_IPNC# setenv netmask <NETWORK NETMASK ADDRESS>
   DM388_IPNC# setenv dnsip <NETWORK DNS IP ADDRESS>
   DM388_IPNC# setenv dnsip2 <ALTERNATE DNS IP ADDRESS>
   ```

   Choose any of the following steps for setting up the IP address of IPNC during uBoot operations:

   Using static IP address

   ```
   DM388_IPNC# setenv ipaddr ‘192.168.1.1’
   ```

   Using DHCP

   ```
   DM388_IPNC# dhcp
   ```

   For flashing U-Boot-MIN to NAND

   ```
   DM388_IPNC# mw.b 0x81000000 0xFF 0x20000
   DM388_IPNC# tftp 0x81000000 u-boot.min.nand
   DM388_IPNC# nand erase 0x0 0x20000
   DM388_IPNC# nand write.i 0x81000000 0x0 0x20000
   ```

6. Use the following commands on the uBoot shell prompt to flash the second stage.

   For flashing U-Boot to NAND

   ```
   DM388_IPNC# mw.b 0x81000000 0xFF 0x40000
   DM388_IPNC# tftp 0x81000000 u-boot.bin
   DM388_IPNC# nand erase 0x20000 0x60000
   DM388_IPNC# nand write.i 0x81000000 0x20000 0x60000
   ```

1.4.3 Flashing Linux Kernel From U-Boot

1. Setup the TFTP server on the PC (assume the IP address of PC is 192.168.1.15).
2. Use the following commands on the uBoot second stage shell prompt to flash the first stage.

   For setting up the TFTP, the following ENV is needed:

   ```
   DM388_IPNC# setenv serverip <TFTP SERVER IP ADDRESS>
   DM388_IPNC# setenv ethaddr <DEVICE MAC ADDRESS>
   DM388_IPNC# setenv gatewayip <NETWORK GATEWAY ADDRESS>
   DM388_IPNC# setenv netmask <NETWORK NETMASK ADDRESS>
   DM388_IPNC# setenv dnsip <NETWORK DNS IP ADDRESS>
   DM388_IPNC# setenv dnsip2 <ALTERNATE DNS IP ADDRESS>
   ```

3. Choose any of the following steps for setting up for setting up the IP address of IPNC during uBoot operations:

   Using static IP address

   ```
   DM388_IPNC# setenv ipaddr ‘192.168.1.1’
   ```

   Using DHCP

   ```
   DM388_IPNC# dhcp
   ```

   For flashing uImage to NAND

   ```
   DM388_IPNC# mw.b 0x81000000 0xFF 0x20000
   DM388_IPNC# tftp 0x81000000 uImage
   DM388_IPNC# nand erase 0x0 0x20000 0x00300000
   DM388_IPNC# nand write.i 0x81000000 0x00280000 0x300000
   ```


Copyright © 2016, Texas Instruments Incorporated

1.4.4 Flashing File System From U-Boot

1. Setup the TFTP server on the PC (assume the IP address of PC is 192.168.1.15).
2. Use the following commands on the uBoot second stage shell prompt to flash the first stage.

   For setting up the TFTP, the following ENV is needed:

```
DM388_IPNC # setenv serverip <TFTP SERVER IP ADDRESS>
DM388_IPNC # setenv ethaddr <DEVICE MAC ADDRESS>
DM388_IPNC # setenv gatewayip <NETWORK GATEWAY ADDRESS>
DM388_IPNC # setenv netmask <NETWORK NETMASK ADDRESS>
DM388_IPNC # setenv dnsip <NETWORK DNS IP ADDRESS>
DM388_IPNC # setenv dnsip2 <ALTERNATE DNS IP ADDRESS>
```

   Choose either of the following steps for setting up the IP address of IPNC during uBoot operations:

   Using static IP address

```
DM388_IPNC# setenv ipaddr '192.168.1.1'
```

   Using DHCP

```
DM388_IPNC# dhcp
```

NOTE: NAND scrubbing is always required before reprogramming theJFFS2 or UBIFS image. This means that uBoot and uImage need to be reprogrammed.

1.4.4.1 Flashing UBIFS File System to NAND

```
DM388_IPNC # nand scrub 0x006C0000 0xB940000
DM388_IPNC# mw.b 0x81000000 0xFF 0x5000000
DM388_IPNC# tftp 0x81000000 ubifs_ipnc.bin
DM388_IPNC# nand erase 0x006C0000 0xB940000
DM388_IPNC# nand write 0x81000000 0x006C0000 0x5000000
```

After these changes, bootcmd and bootargs need to be programmed based on one of the boot methods mentioned in Section 3.

NOTE: Starting from IPNC RDK V2.8, ECC control in the U-boot has been disabled. The first time the U-boot is upgraded from an older version to V2.8 (or any newer version), you still need to run the old commands with ECC in order to successfully flash the U-boot into NAND and reboot. After that, switch to the new commands that are provided. The old commands to copy u-boot.min.nand and u-boot.bin to the device should be only be used once.

1.4.5 Flashing U-Boot From U-Boot via UART

In this mode, the assumption is that the older version of uBoot already exists on the NAND.

1. Connect a RS232 serial port cable to the serial port (COM1) of the Windows host machine where Teraterm is installed. Make sure that the version 4.69 or later of the Teraterm is installed. Connect the free end of the serial RS232 cable to the UART port.
2. Switch on the base EVM and press any key when a countdown appears on the console. U-Boot via UART can be flashed from first or second stage. The following prompt should show up.

```
TI_MIN#
```


Copyright © 2016, Texas Instruments Incorporated

1.4.5.1 Updating First Stage u-boot

1. Flash U-Boot-MIN for NAND (u-boot.min.nand) to NAND by executing the following commands:

```
TI_MIN # mw.b 0x81000000 0xFF 0x20000
TI_MIN # loadb 0x81000000
```

2. From the TeraTerm menu, click 'File' → 'Transfer' → 'Kermit' → 'Send'. Select the first stage u-boot image 'u-boot.min.nand' and click the 'OPEN' button. Note that this should be the same file as used earlier in Section 1.4.1.1 and Section 1.4.4.
3. Wait for the download to complete and then run the following commands in the u-boot prompt:

```
TI_MIN# nand erase 0x0 0x20000
TI_MIN# nand write.i 0x81000000 0x0 0x20000
```

4. If no error messages are displayed, the first stage of NAND boot has been successfully transferred to NAND.

1.4.5.2 Flashing Second Stage u-boot

1. Enter the following command at the prompt:

```
TI_MIN# loadb 0x81000000
```

2. From the TeraTerm menu click 'File' → 'Transfer' → 'Kermit' → 'Send'. Select the second stage u-boot image 'u-boot.bin' built earlier and click the 'OPEN' button. This image is available at the location *<installDir>/Binaries/DMxxx/fullfeature* or *<installDir>/Binaries/DMxxx/lowpower*.
3. Wait for the download to complete and then issue the following command:

```
TI_MIN# go 0x81000000
```

4. You should get the second stage u-boot prompt with the count of 3, 2, 1. Select 'Enter' to stop and get the u-boot prompt.
5. Run the following commands:

```
DM388_IPNC# nand erase 0x20000 0x60000
DM388_IPNC# nand write.i 0x81000000 0x20000 0x60000
```

6. Flashing of the u-boot on IPNC is complete.

1.4.6 Flashing Images to NAND Using SD Boot

It is not necessary to have the older version of uBoot on NAND in this mode.

1. SDCARD boot files can be located at *<install dir>/Binaries/DMxxx/sd*.
2. Copy images from the *<install dir>/Binaries/DMxxx/sd/MLO* file.
3. Copy the *<install dir>/Binaries/DMxxx/sd/u-boot.bin* to your SD Card.
4. Boot using the SD card as described in Section 3.1.2. Do not interrupt the countdown until the `DM388_IPNC#` prompt comes up.
5. Use the commands shown in Section 1.4.6.1 and Section 1.4.6.2 on the uBoot shell prompt.

1.4.6.1 Flashing First Stage to NAND From Second Stage in SD Boot

```
DM388_IPNC# mmc rescan 0
DM388_IPNC# mw.b 0x81000000 0xFF 0x20000
DM388_IPNC$ fatload mmc 0 0x81000000 u-boot.min.nand
DM388_IPNC# nand erase 0x0 0x20000
DM388_IPNC# nand write.i 0x81000000 0x0 0x20000
```

1.4.6.2 Flashing Second Stage to NAND From Second Stage in SD Boot

```
DM388_IPNC# mmc rescan 0
DM388_IPNC# mw.b 0x81000000 0xFF 0x20000
DM388_IPNC$ fatload mmc 0 0x81000000 u-boot.bin
DM388_IPNC# nand erase 0x20000 0x60000
DM388_IPNC# nand write.i 0x81000000 0x20000 0x60000
```


Copyright © 2016, Texas Instruments Incorporated

2 Building IPNC Application

This section provides information on building the IPNC application.

2.1 IPNC Application Build Procedure

To build the application, complete the successful installation as mentioned in Section 1.

2.1.1 PATH and Environment Variable Update

1. Move to the IPNC RDK base directory using the following command: *cd <installDir>/Source/ipnc_rdk*. Typical installation of the IPNC RDK software is shown in the directory structure shown in Figure 8.

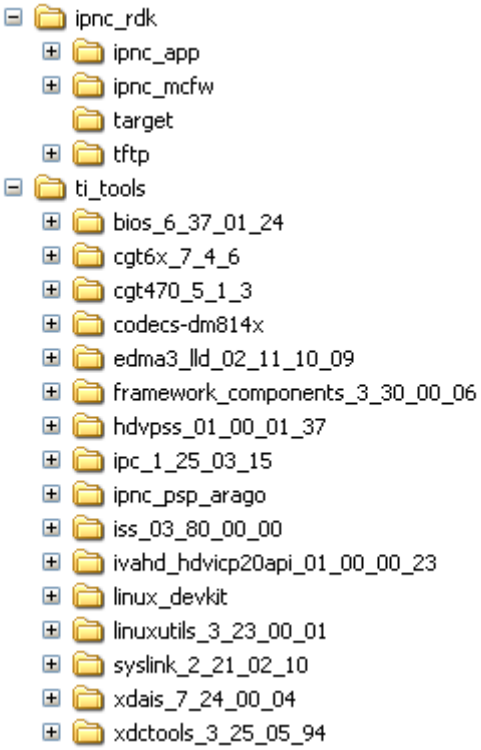

Figure 8. IPNC RDK Directory Structure

NOTE: For example, you can choose different install directories based on convenience. Code or install the directory that does not have any dependency on any hard coded path.

2. If your login is not root, then login using the following commands to prevent errors during installation:

```
#chmod -R a+rwx < IPNC_INSTALL_DIR >
#chown -R <useracct> < IPNC_INSTALL_DIR >
where,
< useracct > is your user login ID on HOST LINUX PC.
< IPNC_INSTALL_DIR > is the directory you set in Rules.make
```

3. Open and edit Rules.make.

   Modify the Rules.make file based on the system's deployment. Rules.make that comes with the release package has default configuration. If you have different paths and need to build the binaries for different configurations, modify them in Rules.make.

4. A simple description about Rules.make to set the correct environment paths on your system is shown in Table 4.

Copyright © 2016, Texas Instruments Incorporated

Table 4. Setting the Correct Environment

Option	Values	Description
SYSTEM_PLATFORM	EVM	SYSTEM_PLATFORM determines the hardware platform.
APP_BUILD_CFG	debug or release	Build configuration on whether debug or release mode of the build is selected
BINARY_MODE	nand or sd	Determines whether the binary build is for running from NAND or SDCARD.
IPNC_DEVICE	DM385, DM388	Selects the platform that the code is built for
IPNC_CONFIG	FULL_FEATURE, LOW_POWER	IPNC_CONFIG selects the camera configuration for low-power or extended-feature mode. In low-power mode, DSP and DSS core are turned off to save power. With the DSS core off, the scaling and noise filtering operations can be achieved on the ISS core. Also low-power mode does not have features to preview the captured image on any display. In full-feature mode, you have flexibility to extend the feature set of IPNC by using DSP and DSS core.
MEMORY_CONFIG	256MB	MEMORY_CONFIG determines the memory bootprint that the application is built for, 256MB is supported.
MTD_UTILS_MODE	64 bit or 32 bit	Select the UBIFS build mode using 32-bit or 64-bit Linux host machine
IMGS_ID	IMGS_MICRON_MT9J003 IMGS_MICRON_AR0330 IMGS_MICRON_AR0331 IMGS_PANASONIC_MN34041 IMGS_OMNIVISION_OV10630 IMGS_MICRON_MT9M034 IMGS_SONY_IMX136 IMGS_SONY_IMX104 IMGS_MICRON_MT9M034 IMGS_SONY_IMX140	Determines the sensor used in the BUILD. Check the sensor connected in the hardware.
IPNC_TILER_ENABLE	YES or NO	This determines wether the IPNC APP will run on TILER mode or not. It enables 90/270 rotation mode via TILER memory access.
CAMERA_TILER_ENABLE	YES or NO	This determines whether to use TILER or not. Needed to be ON for 90/270 rotation.
BTE_ENABLE	YES or NO	Needs to be TURNED ON if TILER is ON
MJPEG_RC_MODE	MJPEG_QPMODE MJPEG_VBR_MODE MJPEG_CBR_MODE	MJPEG_RC_MODE determines the rate control mode with which MJPEG will operate. Variable BItRate(VBR) and Constant BitRate(CBR) are supported. Having option as QPMODE will select the MJPEG encoding without RateControl using Quality factor set from GUI. Note that rate control is applied at frame level.
WDR_ON	YES or NO	If sensor supports WDR mode output then it will be turn on WDR link will replace GLBCE link in the full feature use case
LOW_POWER_OPP100_ MODE	YES or NO	Power optimization mode for 30 fps usecases for OPP100 is selected here. This flag if TRUE will lower clocks for OPP100 configuration in UBOOT Application code (mcfw+app) is not affected by this change Ensure that the samples used are qualified for OPP100 configuration To achieve lower power, voltage can be set to 1.1 V if this mode is turned ON
AES_MODULE_MODE	YES or NO	This is not supported in the current release.
BUILD_WARNINGS_AS_ ERROR	YES or NO	Build warnings will be treated as errors for ipnc_mcfw folder
GUI_ENGINE_MODE	GODB_DEFAULT or VLABS_VLC_OPENSRC	GODB_DEFAULT vendor can be used if the default GUI screens are to be selected VLABS_VLC_OPENSRC vendor can be selected if needed to support on all browsers Note VLABS_VLC_OPENSRC mode is of beta quality (not fully tested)

Copyright © 2016, Texas Instruments Incorporated

Table 4. Setting the Correct Environment (continued)

Option	Values	Description
SENSOR_BOARD_VENDOR	APPRO or LIMG	APPRO: if the appro sensor board is used or LIMG, if the Leopard Imaging Board is used
HARDWARE_REVISION	REV2	Hardware revision: REV2 that the software is to be built for REV2 is the new low-power design that has low-power numbers compared to the older design Note that DM385 and DM388 support REV2 and has 1.8V I/O

Full-feature build sets POWER_OPT_DSP_OFF and POWER_OPT_DSS_OFF to “NO”. This ensures switching DSS and DSP to ON position.

Low-power build needs to set POWER_OPT_DSP_OFF and POWER_OPT_DSS_OFF to “YES”. This ensures switching DSS and DSP to ON position.

The TEARDOWN_LOAD_UNLOAD flag determines whether a complete unload and reload of the usecase is necessary or whether it has to be stop/start of the usecase only.

Make sure that the TFTP_HOME path set in the Rules.make exist as Uboot, kernel binaries and ubifs/jffs2 file system gets installed in TFTP_HOME. If this folder does not exist, ‘make’ will fail at the time of these binaries installation.

5. Build the application using the following command:
```
Make -s sysall
```
This cleans and rebuilds the packages below:

(a) Linux Uboot

(b) Linux PSP

(c) cmem

(d) Syslink

(e) HDVPSS

(f) Iss

(g) Ipnc application binaries

(h) M3 and DSP firmware binary

(i) File system update with binaries

This builds both the A8 host side code as well as BIOS side code and copies all the IPNC files executables to the file system, as mentioned to the output directory located at $(EXEC_DIR) settings in $(installDir)/Rules.make.

6. In order to incrementally or individually build different packages, use the following make targets:

NOTE: “-s” option is used to suppress detailed prints on the command line.

Copyright © 2016, Texas Instruments Incorporated

Table 5. MAKE Command Options

Make Command	Description
make –s sys	Incrementally build all IPNC RDK system and all dependant packages
make –s sysclean	Clean IPNC RDK and all dependant packages
make –s sysall	Clean and rebuild IPNC RDK and all dependant packages
make –s ipncapp	Incrementally build the IPNC application. This includes the hdvpss, iss, mcfw and ipnc_app folder.
make –s ipncappclean	Clean the IPNC application build
make –s ipncappall	Clean and rebuild the IPNC application
make –s mcfw	Incrementally build the mcfw folder for both Linux and bios side software
make –s clean	Clean the McFW builds
make –s all	Clean and rebuild the McFW software on both Linux and bios side
make –s mcfw_bios6	Incrementally build IPNC RDK on the BIOS6 side only
make –s mcfw_bios6_clean	Clean IPNC RDK on the BIOS6 side only
make –s mcfw_bios6_all	Clean and rebuild IPNC RDK on the BIOS6 side only
make –s mcfw_linux	Incrementally build IPNC RDK on the Linux side only
make –s mcfw_linux_clean	Clean IPNC RDK on the Linux side only
make –s mcfw_linux_all	Clean and rebuild IPNC RDK on the Linux side only
make –s lsp	Incrementally build the Linux PSP
make –s lspclean	Clean all of the Linux PSP
make –s lspall	Clean and rebuild the Linux PSP
make –s uboot	Incrementally build the Linux NAND Uboot
make –s ubootclean	Cleam the Linux NAND Uboot
make –s ubootall	Clean and rebuild the Linux NAND Uboot
make –s ubootminsd	Clean and build the Linux SD card boot uboot.min (MLO) and install it in TFTP_HOME directory
make –s ubootbinsd	Clean and build the Linux SD card boot uboot.bin and install it in TFTP_HOME directory
make –s syslink	Incrementally build the syslink (on both the BIOS and the Linux side, including .ko on Linux side)
make –s syslinkclean	Clean all syslink
make –s syslinkall	Clean and rebuild syslink
make –s hdvpss	Incrementally build the HDVPSS drivers (applicable for BIOS side only)
make –s hdvpssclean	Clean all of the HDVPSS drivers
make –s hdvpssall	Clean and rebuild the HDVPSS drivers
make –s iss	Incrementally build the ISS drivers
make –s issclean	Clean all of the ISS drivers
make –s issall	Clean and rebuild the ISS drivers
make –s cmem	Incrementally build the cmem kernel module
make –s cmemclean	Clean the cmem kernel module
make –s cmemall	Clean and rebuild the cmem kernel module
Make –s ubifs	Create the ubifs file system and install it in TFTP_HOME directory

NOTE: This entire package was built and tested with ubuntu 10.04 LTS installed on the host PC.

It is recommended to check the following and create a link after installation of the Linux operating system (OS); on the PC side, avoid compile time errors.

Copyright © 2016, Texas Instruments Incorporated

```
[root@localhost lib]# ls /usr/lib/libstdc++.so.* -al
lrwxrwxrwx 1 root root     14 Nov 24  2007 /usr/lib/libstdc++.so.5 -> libstdc++.so.6
lrwxrwxrwx 1 root root     18 Mar 25  2007 /usr/lib/libstdc++.so.6 -> libstdc++.so.6.0.8
-rwxrwxrwx 1 root root 940972 Oct 11  2006 /usr/lib/libstdc++.so.6.0.8
[root@localhost lib]#
```

2.2 Building U-boot and Kernel Image

1. As mentioned in Section 2.1.1, use the make commands shown below to build the Linux kernel and u-boot:

```
$ make –s lspall
$ make –s ubootall
```

2. u-boot.min.nand, u-boot.bin and uImage are available in <IPNC_RDK_BASE>/tftphome, assuming the default Rules.make path is used.

3 Running IPNC Application

3.1 Running IPNC Application

3.1.1 Boot From NAND

1. Connect the serial port from IPNC to the host PC.
2. Power on IPNC and wait for U-Boot to come up on the serial console.
3. Setup the environment variables.
4. See the Ethernet address and hardware version on the board. This step is not needed if uBoot already shows the right Ethernet address.

```
DM388_IPNC# setenv ethaddr xx:xx:xx:xx:xx:xx
```

5. For the static IP address, set the IP addresses for your environment.

```
DM388_IPNC# setenv ipaddr 192.168.1.168;setenv gatewayip 192.168.1.1;setenv netmask
255.255.255.0;setenv serverip 192.168.1.151
```

NOTE: IPNC RDK uses a pre-decided static IP address of 192.168.1.168 at the time of boot. It ignores the static IP passed via bootargs. If NFS is needed as the file system in static IP mode, make sure the NFS server static IP settings are appropriate.

For example, ipaddr of NFS host – 192.168.1.170

For DHCP:

```
TI8148_IPNC # dhcp
```

6. Use the following step to setup the boot arguments if NFS is used as the root file system.

Boot commands for low power configuration in Rules.make

```
DM388_IPNC#setenv bootcmd 'ipnc_ff_init 0;nboot 0x81000000 0
0x280000; bootm';saveenv
```

Boot commands for full feature configurations

```
DM388_IPNC#setenv bootcmd 'ipnc_ff_init 1;nboot 0x81000000 0
0x280000; bootm';saveenv
```

Bootargs settings for 256MB Memory Map

```
DM388_IPNC # setenv bootargs 'console=ttyO0,115200n8 root=/dev/nfs rw mem=54M
vram=4M notifyk.vpssm3_sva=0xBFD00000
nfsroot=192.168.1.151:/home/work/IPNC_RDK/Release_v3.50.00/Source/ipnc_rdk/
target/filesys,nolock eth=00:0C:0C:A0:07:66 ip=192.168.1.168
cmemk.phys_start=0x83600000 cmemk.phys_end=0x85600000
cmemk.allowOverlap=1 earlyprintk';saveenv
```


Copyright © 2016, Texas Instruments Incorporated

NOTE: Change the nfsroot, IP and ethaddr to your environment.

If the dynamic IP address is to be used, set ip=dhcp in the bootargs above in the 'Bootargs settings for 256MB Memory Map' section where it currently says: ip=192.168.1.168.

7. Use the following step to setup the boot arguments if the kernel image is programmed on NAND and UBIFS is used as root file system.

   Boot commands for low-power configuration in Rules.make

   ```
   DM388_IPNC#setenv bootcmd 'ipnc_ff_init 0;nboot 0x81000000
   0 0x280000; bootm';saveenv
   ```

   Boot commands for full feature configurations

   ```
   DM388_IPNC#setenv bootcmd 'ipnc_ff_init 1;nboot 0x81000000 0
   0x280000; bootm';saveenv
   ```

   Bootargs settings for 256MB Memory Map

   ```
   setenv bootargs 'console=ttyO0,115200n8 rootwait=1 rw
   ubi.mtd=4,2048 rootfstype=ubifs root=ubi0:rootfs init=/init
   mem=54M vram=4M notifyk.vpssm3_sva=0xBFD00000 ip=dhcp
   eth=$(ethaddr) cmemk.phys_start=0x83600000
   cmemk.phys_end=0x85600000 cmemk.allowOverlap=1
   earlyprintk';saveenv
   ```

   NOTE: Change the nfsroot, IP and ethaddr to your environment.

   If the dynamic IP address is to be used, set ip=dhcp in the bootargs above in the 'Bootargs settings for 256MB Memory Map' section where it currently says: ip=192.168.1.168.

8. Save the environment variables.

   ```
   DM388_IPNC # saveenv
   ```

   This saves the current environment variable setting to NAND.

9. Boot the IPNC system using the following command:

   ```
   DM388_IPNC # boot
   ```

   NOTE: After programming, it is always recommended to restart the IPNC for the values mentioned in the above step to take effect

10. Run the “printenv” command to read the saved values that verify the correctness of the environment variables.

    NOTE: Improper programming of the environment variables can make the IPNC non-usable or non-functional.

    The following lines confirm the boot mode:

    ```
    ## Booting image at 81000000 ...
    ...
    Starting kernel ...
    ...
    VFS: Mounted root (nfs filesystem).
    ```


Copyright © 2016, Texas Instruments Incorporated

3.1.2 Boot From SD

1. Change the BINARY_MODE to ‘sd’ mode in Rules.Make
2. Make uboot.min and uboot.bin for the SD card. Uboot.min for the SD card is named as MLO.

```
$ cd <IPNC_RDK_BASE>/ipnc_rdk
$make –s ubootmin
$make –s ubootbin
```

   This makes the MLO and uboot.bin in $TFTP_HOME set in Rules.make.

3. Create NFS tar ball in the filesys directory as shown below:.

```
#cd IPNC_INSTALL_DIR>/target/filesys
#sudo tar cvfz nfs.tar.gz ./*
```

4. Format the SD card.
   Prerequisites: Ensure that the following are available:
   (a) A Linux host with fdisk, sfdisk, mkfs.ext3 and mkfs.vfat utilities is available.
   (b) Copy images MLO, u-boot.bin, uImage and mksd-ti81xx.sh from TFTP_HOME and nfs.tar.gz (created in steps 1 and 2 to a directory on this Linux machine. For subsequent description, it is assumed that these files exists in TFTP_HOME.
      (i) `mksd-ti81xx.sh is located at <install dir>/Utils/sd-script`
   (c) Empty the SD card (at least 256MB, preferably 4GB SDHC).
   (d) An SD memory card reader/programmer to copy files from the Linux host

   NOTE: The SD Boot has some specific restrictions about the format of the first partition and copying the MLO image. Is recommended that the supplied script mksd-ti81xx.sh is used for creating partitions and copying files.

   Steps:
   (e) Connect the SD memory card using the memory card reader to the Linux host.
   (f) Note the name allotted for this device. It is recommended to name the device "/dev/sdd".
   (g) Navigate to the TFTP_HOME directory where all of the mentioned files are copied.
   (h) Ensure that the script mksd-ti81xx.sh has executable permissions.
   (i) The scripts expects arguments in the following format:

```
./mksd-ti81xx.sh <sd-device-name> <sd-1st-stage-bootloader> <kernel-uImage> <tar-gzipped-filesystem-directory>
```

   (j) Note that root/sudo permissions are required.
   (k) In this example, the following command was run:

```
sudo ./mksd-ti81xx.sh /dev/sdd MLO u-boot.bin uImage nfs.tar.gz
```

   (l) When asked about overwriting data, confirm it; the files along with file system will be copied to SD card.
   (m) Note that this script creates two primary partitions:
      (i) The first partition is formatted as FAT32 containing MLO, u-boot.bin, uImage files.
      (ii) The second partition is formatted as ext3 where the file system is extracted in root.

Copyright © 2016, Texas Instruments Incorporated

5. Set the boot mode setting switch S1 as shown in Figure 9. Insert the SD card on the EVM, then reboot.

Figure 9. Boot Mode Order

6. Interrupt the countdown in the second stage and set the environment variable.

   Boot commands for low-power configurations

```
DM388_IPNC#setenv bootcmd 'ipnc_ff_init 0; mmc rescan 0; fatload
mmc 0 0x81000000 uImage; bootm 0x81000000';saveenv
```

   Boot commands for full-feature configurations

```
DM388_IPNC#setenv bootcmd 'ipnc_ff_init 1; mmc rescan 0; fatload
mmc 0 0x81000000 uImage; bootm 0x81000000';saveenv
```

   Set the bootargs for 256MB Memory Map

```
DM388_IPNC# setenv bootargs 'console=ttyO0,115200n8 noinitrd
mem=54M rootwait vram=4M notifyk.vpssm3_sva=0xBFD00000
root=/dev/mmcblk0p2 rw eth=00:0C:0C:02:30:FB ip=172.24.191.14
cmemk.phys_start=0x83600000 cmemk.phys_end=0x85600000
cmemk.allowOverlap=1 earlyprintk';saveenv
```

NOTE: Change the IP and ethaddr to your environment. If the dynamic IP address is to be used, set ip=dhcp in the bootargs above.

Further information on the SD card boot can be downloaded from the following link: http://processors.wiki.ti.com/index.php/TI81XX_PSP_UBOOT_User_Guide#U-Boot_SD_.28Secured_Digital_card.29_Support.

3.2 Troubleshooting

If video streaming is not working, it may be due to config file corruption.

3.3 Network Connections Setting

There are many methods to set the IPNC environment. Connection using dynamic host configuration protocol (DHCP) is discussed in this section.

Follow the steps below to recover from this error:

1. Try removing sysenv.cfg from the location “/mnt/nand” with the following command:

   rm /mnt/nand/sysenv.cfg

2. Re-start the board.

Copyright © 2016, Texas Instruments Incorporated

The above steps will fix any config file corruption issues.

3.3.1 Connection Using DHCP

The IPNC environment can be set using DHCP. Networked devices (clients) use DHCP to obtain various parameters necessary for the clients to operate in an IP network. By using this protocol, devices can be added to the network with minimal or no manual configurations.

There are two methods to build the connecting environment:

- PC connected to DM38x EVM (see Figure 10), you have to set the DHCP server to PC side.

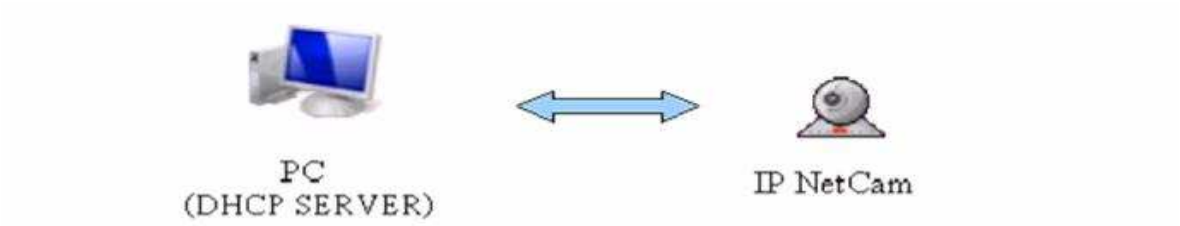

Figure 10. PC Connected to DM38x EVM

- PC connected to DM38x EVM through a router (see Figure 11).

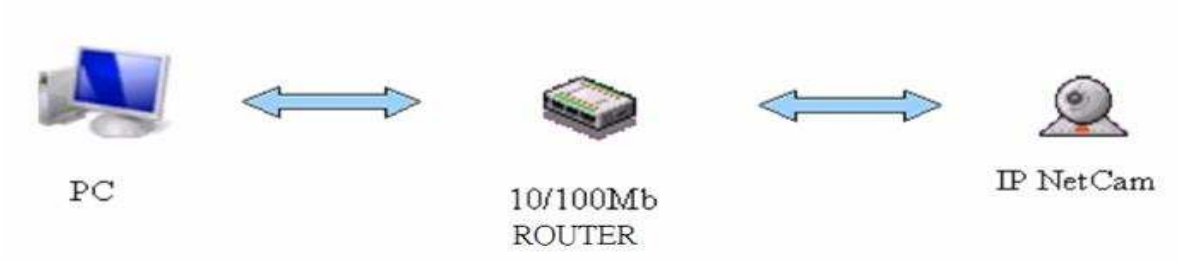

Figure 11. PC Connected to IPNC Through a Router

The following steps needs to be done for connection using DHCP:

- Connect an external power supply to turn on the DM38x EVM. All settings are completed automatically until the LAN LED changes from red to green.
- DHCP server or router assigns a new IP address to DM38x EVM. Ensure to verify that the new IP address is assigned to IPNC.
- Click on the UPnP icon in local PC side to remote monitor with Web browser

Copyright © 2016, Texas Instruments Incorporated

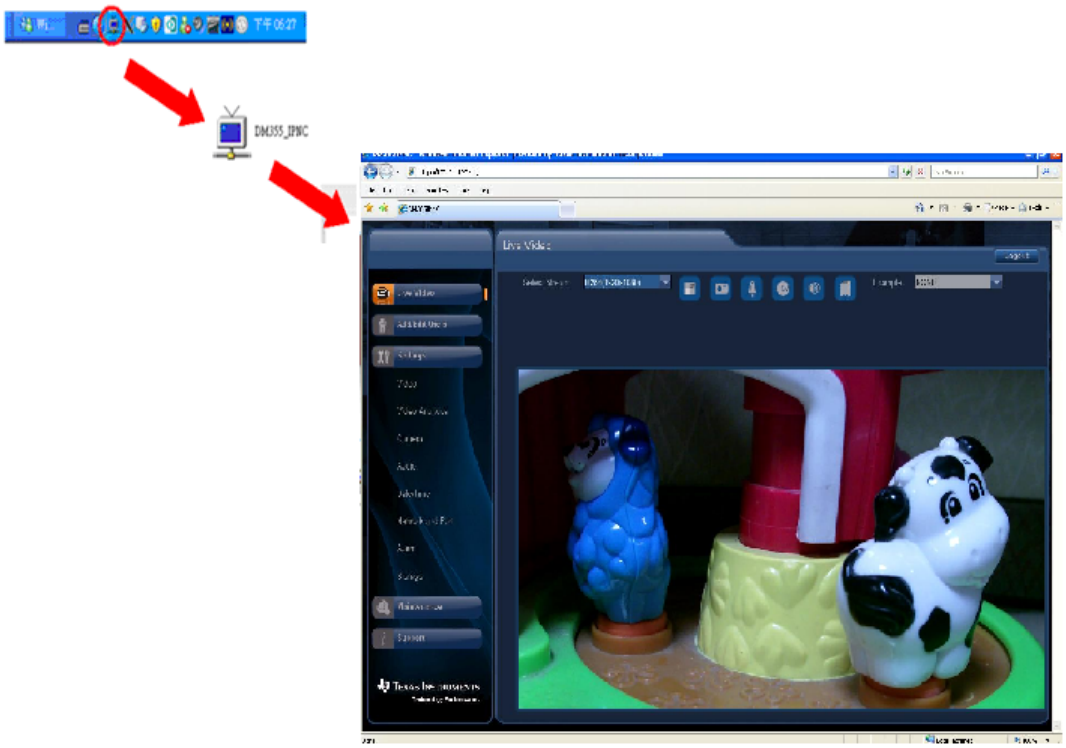

Figure 12. Remote Monitor With Web Browser

- You can also use a new IP address with USB, if you do not want to turn on the UPnP function in your PC side. To do this, connect the DM38x EVM to your local PC with USB cable. When PC detects the new USB device (IPNC), double click the 'usbconf.exe icon. This opens the network data that consist of IP address, Netmask, Gateway HTTP port, and so on.

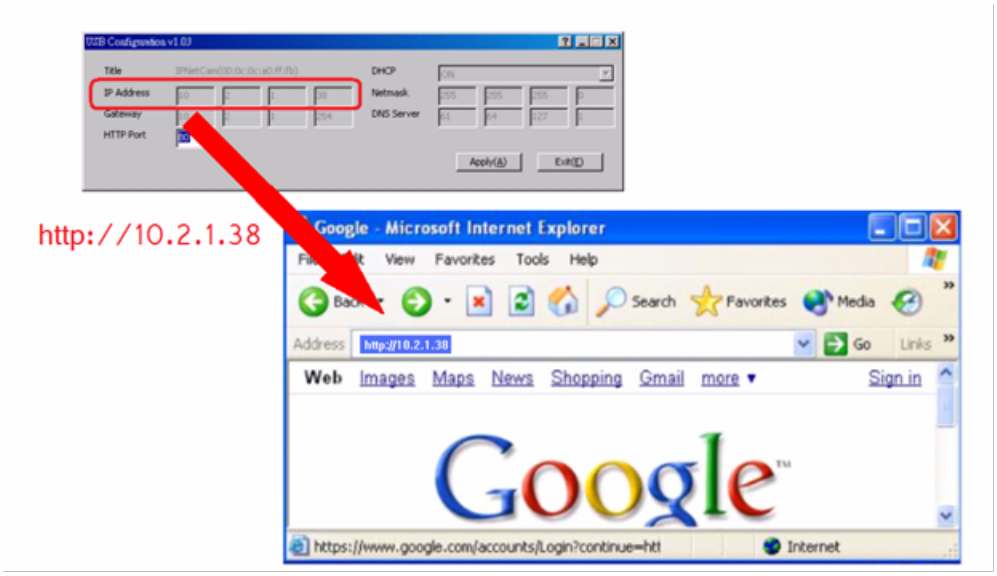

Figure 13. Network Data Display

- In the Internet Explorer window, type the required IP Address in the Address box.

Copyright © 2016, Texas Instruments Incorporated

3.4 Streaming Demo on Web Browser

By default, all of the startup commands are part of /opt/ipnc/autorun.sh, which automatically starts the video capture (from component video port) after power ON. The web browser demo is only supported on windows.

To see the result on Internet Explorer, type the IP address of your device as shown below and press Enter.

```
http:// <IP address>/
```

NOTE: You can get your IP address using the ifconfig command on your device.
autorun.sh, located in file system, provides details on the exact sequence in which the commands needs to be run.
When upgrading to new version, always run command in the camera EVM after login as root → *flash_eraseall /dev/mtd5/*. This erases the old setting and resets to the default setting.

The *Security Warning* dialog box asks whether the ActiveX program needs to be installed.

Figure 14. Security Warning

Install the ActiveX program (select Install).

Clean the software cached on your browser each time there are firmware updates on the IPNC software. The following are the steps to clean the software:

1. Uninstall ActiveX component. To do this, close all Web browsers connected to the IPNC and open Internet Explorer in a blank page.
2. Click on the Tools tab in the Internet Explorer window. Select 'Manage Add-ons' → 'Enable or Disable Add-ons'. The Manage Add-ons dialog box is displayed.

Copyright © 2016, Texas Instruments Incorporated

3. From the Show drop-down list, select 'Downloaded ActiveX Controls (32-bit)' (see Figure 15).

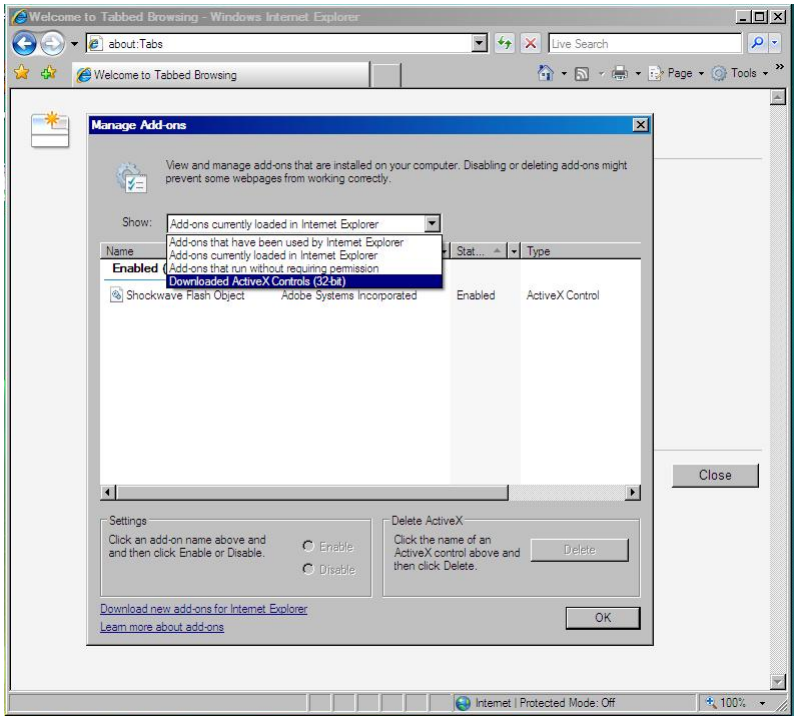

Figure 15. Downloaded ActiveX Controls (32 bit)

You will see an ActiveX Control published by GoDB Tech Private Limited (see Figure 16).

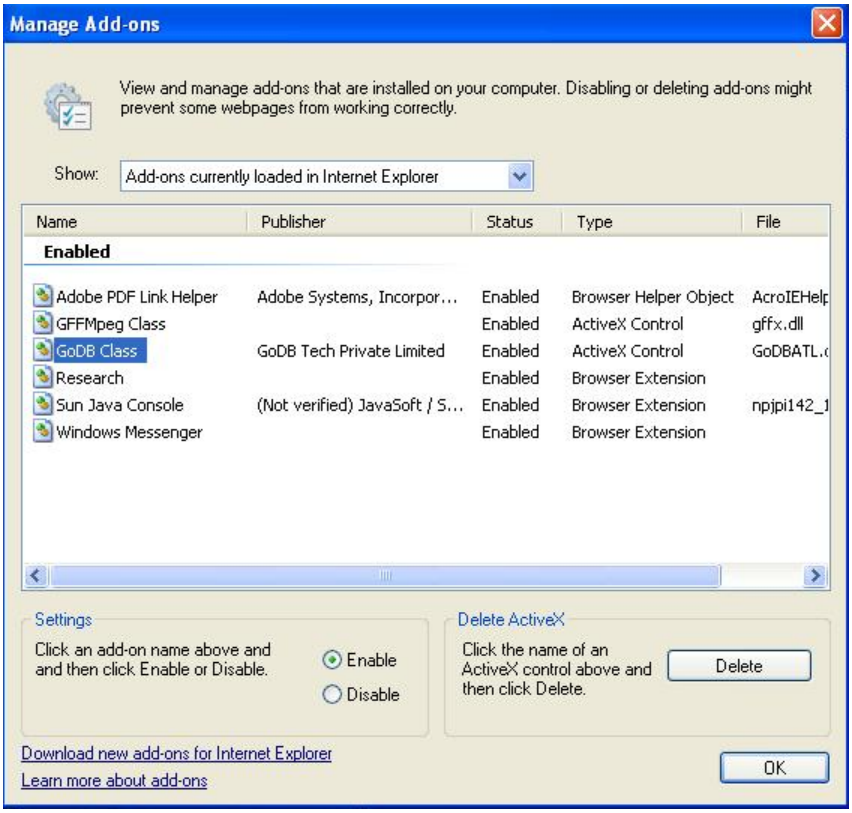

Figure 16. GoDB Tech Private Limited

Copyright © 2016, Texas Instruments Incorporated

4. Select 'ActiveX Control' and click 'Delete' to clean the software cached on your browser (see Figure 17).

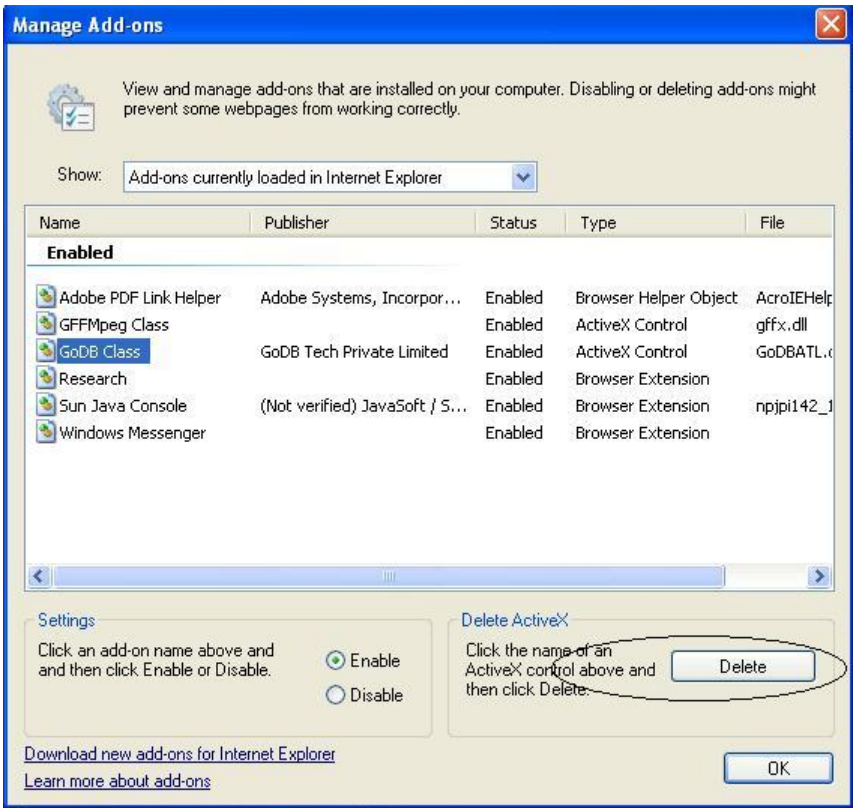

Figure 17. Manage Add-Ons

5. Clean the cookies and temporary files in the IE bowser.
6. Remove files gdobridge.dll, gffx.dll, GoDBATL.dll from C:\WINDOWS folder.
7. Enter the path 'C:\WINDOWS\Downloaded Program Files' in your PC and delete file “GoDB Class” and “GFFMpeg Class”.
8. Restart your IE bowser, as shown in Figure 18.

Figure 18. Restart Browser

Copyright © 2016, Texas Instruments Incorporated

9. If the IPNC is used in IE for the first time, the pop-up will be shown with the required ActiveX player to be installed on your PC. Before this, install the Publisher certificate.

Figure 19. Install the Software

NOTE: This document is prepared based on the screens displayed by a PC running Windows XP with IE7.0 installed. The actual screens displayed may vary based on the operating system and the version of IE installed in the client PC.

10. Click 'Install' for GoDBATL.CAB installation.

    This will install files gdobridge.dll, gffx.dll, GoDBATL.dll in C:\WINDOWS folder.

    “GoDB Class” and “GFFMpeg Class” will be installed on your PC in C:\WINDOWS\Downloaded Program Files.

Copyright © 2016, Texas Instruments Incorporated

11. After installation, press key F5 on the keyboard to refresh the IE active window. This takes you to main Login screen (see Figure 20).

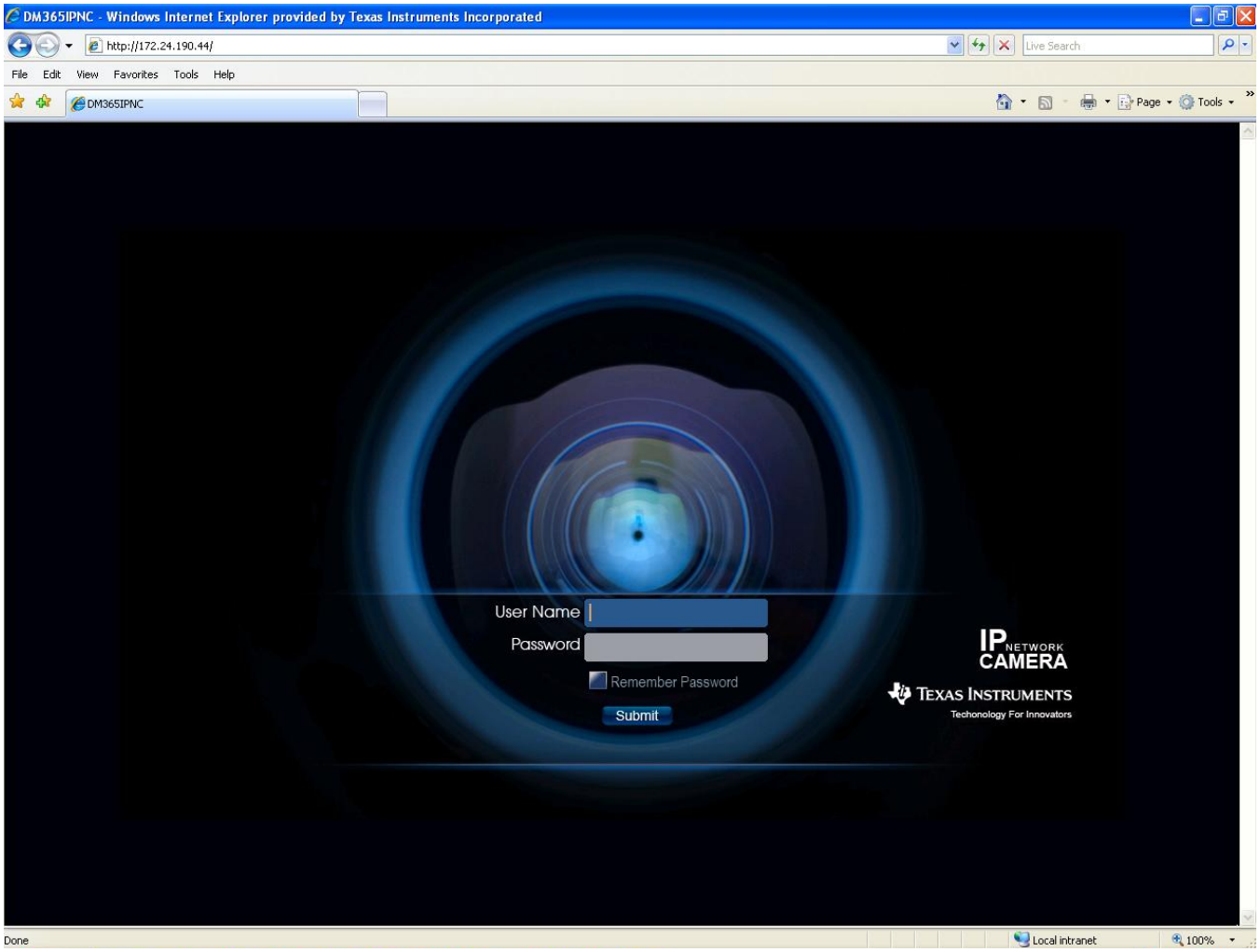

Figure 20. Main Login Screen

Copyright © 2016, Texas Instruments Incorporated

12. Include the default User Name (admin) and Password (9999), and then click 'OK'. The live video screen will be displayed as shown in Figure 20.

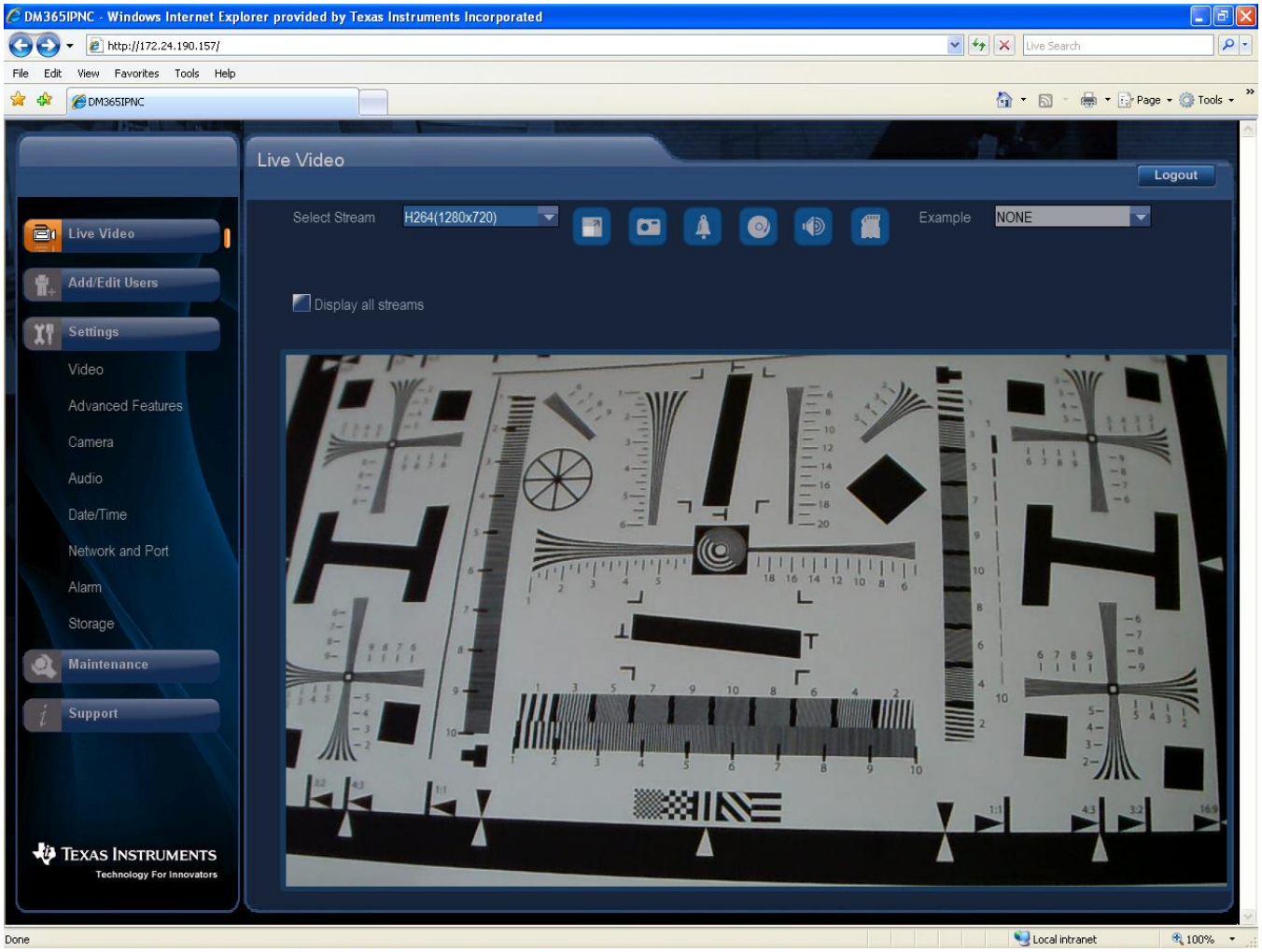

Figure 21. Live Video

3.5 RTSP Streaming Demo on VLC

1. Install the VLC media player on your host PC.

NOTE: You can download the VLC media player at http://www.videolan.org/. This software has been tested with version 2.0 of the VLC player.

By default, all of the startup commands are part of /opt/ipnc/autorun.sh, which automatically starts the video capture from component video input. autorun.sh provides details on the exact sequence in which the commands needs to be run.

When upgrading to a new version, always run command in EVM console after login as root → “flash_eraseall /dev/mtd5/”. This erases old settings and resets them to the default setting.

2. Open the VLC Application.
3. On the Media menu, click 'Open Network Stream'.

Copyright © 2016, Texas Instruments Incorporated

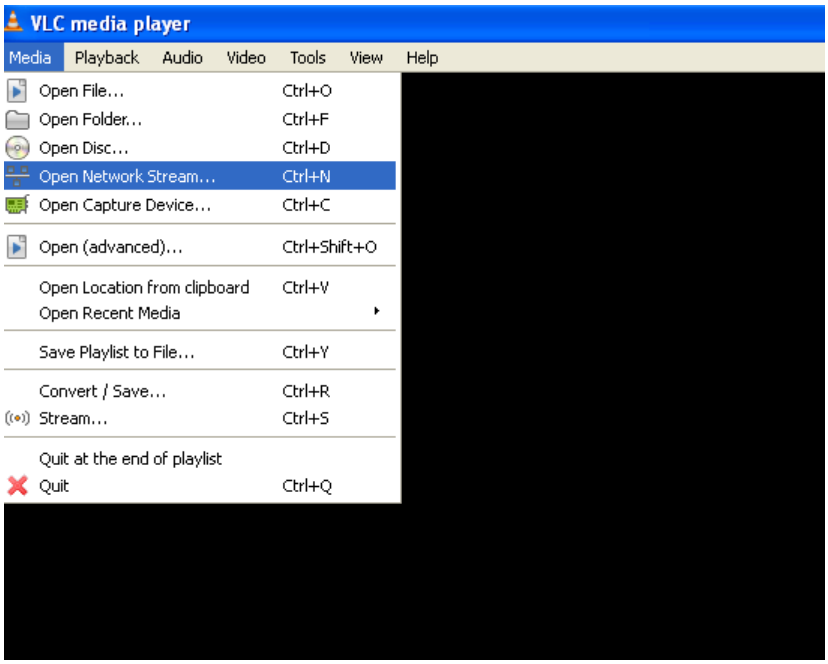

Figure 22. Open Network Stream

4. In the Open dialog box, select the RTSP option. In URL textbox, type the URL and click 'PLAY'.
   For H264 Stream 1:
   `rtsp://<IP_Address>:8557/PSIA/Streaming/channels/2?videoCodecType=H.264`
   For H264 Stream2 :
   `rtsp://<IP_Address>:8556/PSIA/Streaming/channels/2?videoCodecType=H.264`
   For MPEG4 Stream 1:
   `rtsp://<IP_Address>:8553/PSIA/Streaming/channels/1?videoCodecType=MPEG4`
   For MPEG4 Stream 2:
   `rtsp://<IP_Address>:8554/PSIA/Streaming/channels/1?videoCodecType=MPEG4`
   For JPEG Stream :
   `rtsp://<IP_Address>:8555/PSIA/Streaming/channels/0?videoCodecType=MJPEG`

Copyright © 2016, Texas Instruments Incorporated

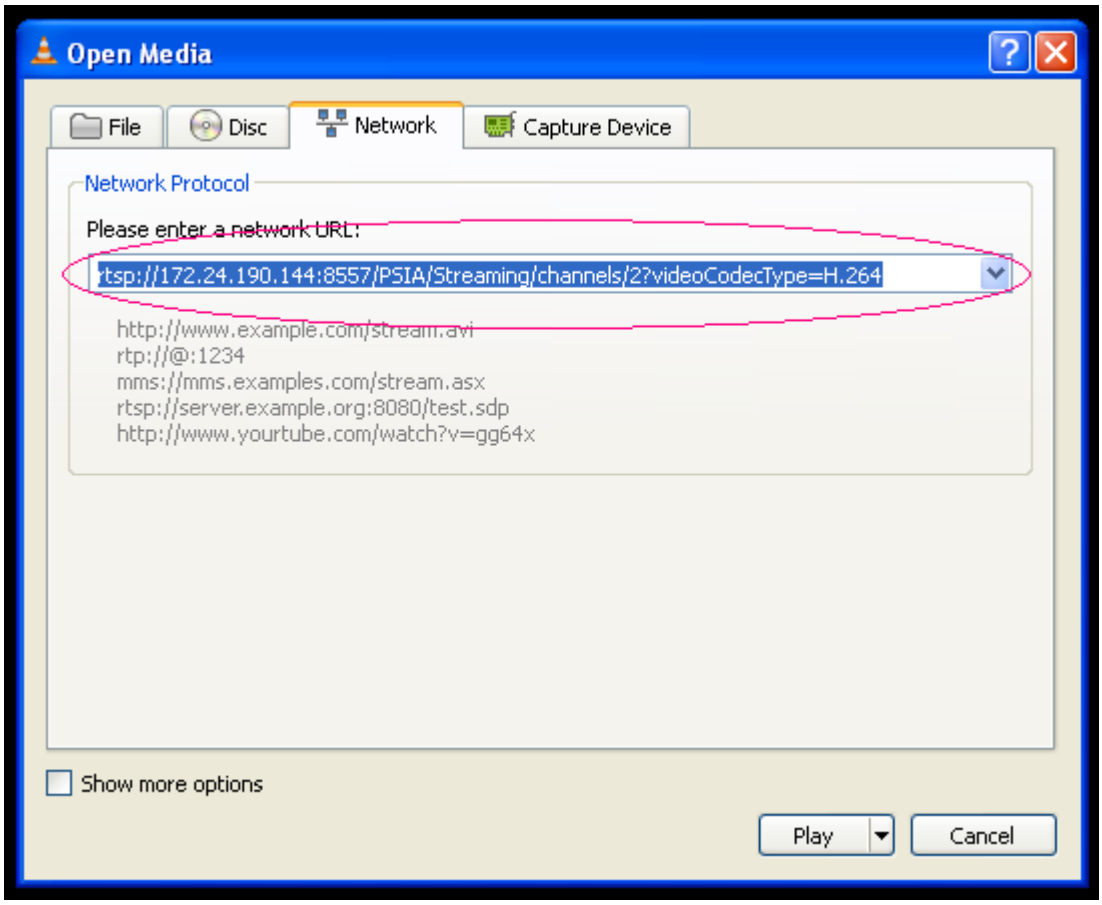

Figure 23. Configure Network Protocol

5. Observe the results on the VLC media player window. The video appears on the browser.

3.5.1 VLC Media Player Setting

The following are steps to set the VLC media player and open a network stream:

1. Install VLC media player on your Host PC.

NOTE: You can download the VLC media player at http://www.videolan.org/. This software has been tested with version 2.0.3 of the VLC player.

2. Open VLC media player.
3. Click 'Tools' → 'Preferences', this displays the Preferences dialog box.

NOTE: Before setting, ensure that all of the parameters are set to default values. To do this, in the Preferences dialog box, click 'Reset Preferences'.

Copyright © 2016, Texas Instruments Incorporated

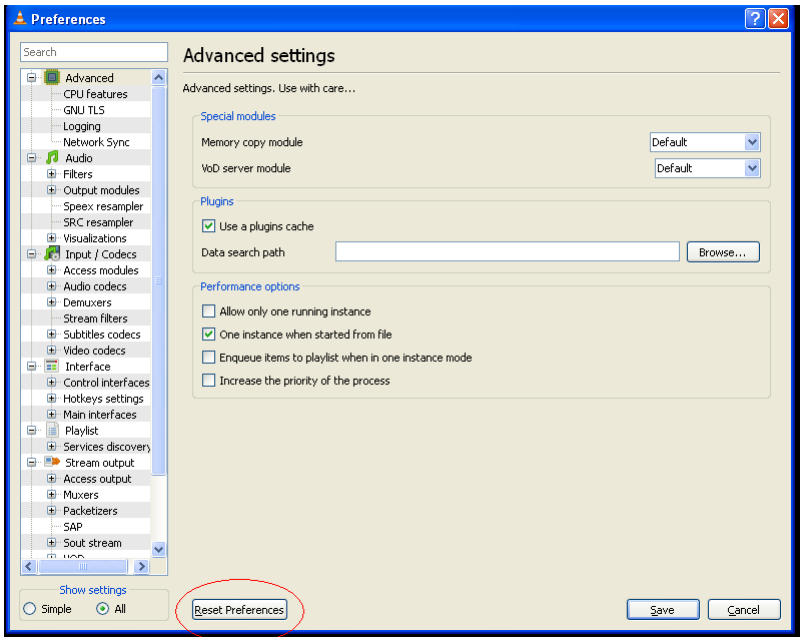

Figure 24. Preferences

4. On the left pane, click 'Video' and select the 'Overlay video output' option, as shown in Figure 25. Also, select 'All' in Show settings.

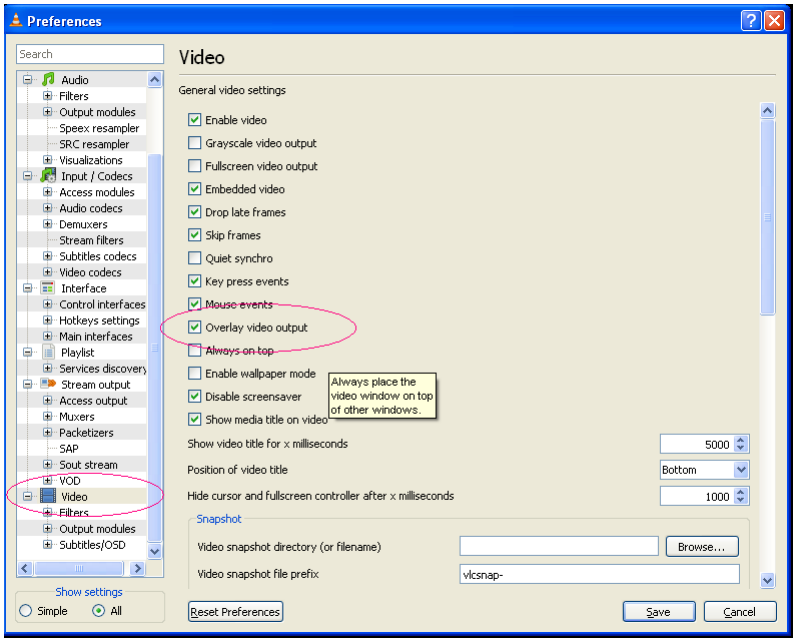

Figure 25. Overlay Video Output

5. On the left pane, click 'Input/Codecs' and scroll down for Advanced options; check the caching value. For low latency, it should be 250 ms based on the PC's performance.

NOTE: If the video is not smooth, set the value as 200 to approximately 1200. This may be due to network congestion, busy PC (memory is not enough or too many processes are running at the same time) or PC's performance does not meet the minimum requirement.

Copyright © 2016, Texas Instruments Incorporated

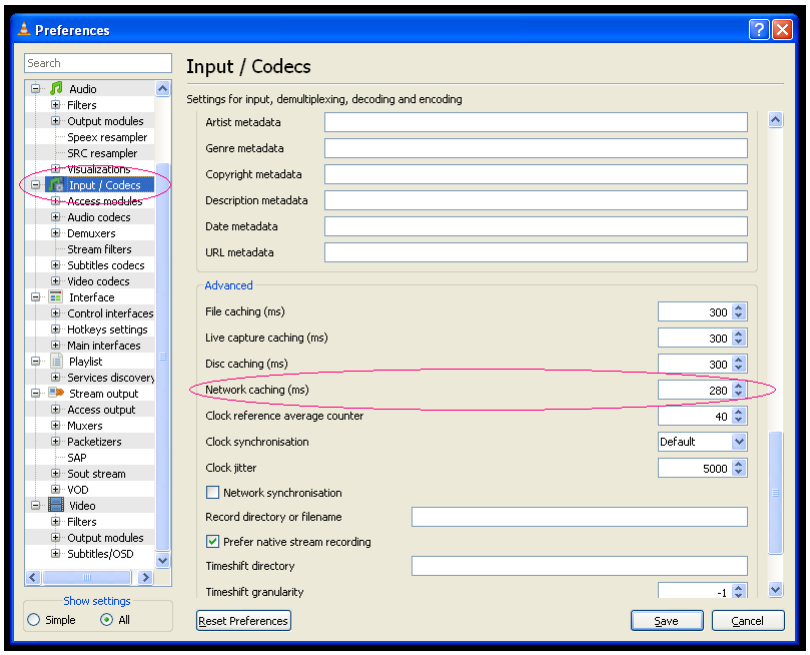

Figure 26. Input/Codecs

6. To save the settings, click 'Save'.
7. In the VLC media player window, on the Media menu, click 'Open Network Stream'. The Open dialog box is displayed.
8. Select the RTSP option. In Address textbox, type the required URL mentioned above and click 'OK'.
9. Once you click 'OK', the video appears on the browser.

3.6 End-to-End Low Latency Measurement

Latency in this context is defined as the time delay between the time where video frame data appears in front of the sensor to the time taken for the same to be displayed at the host PC. This is a summation of the following parameters:

- Capture time
- Encoding time
- Packetization
- Transmission time over network
- Decoding time at host PC
- Display time at host PC

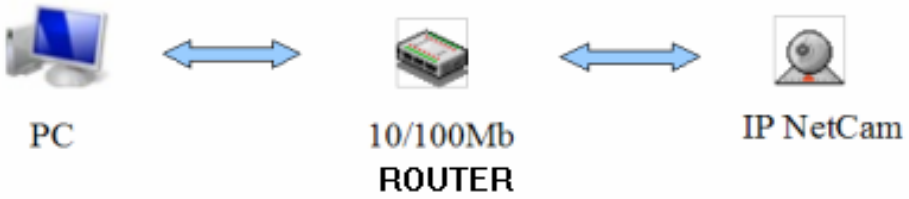

Figure 27. End-to-End Latency

Copyright © 2016, Texas Instruments Incorporated

3.6.1 VLC Player Setting for Low Latency

1. From the Tools menu in the VLC Media Player window, click 'Preferences' (see Figure 28). The Preferences dialog box is displayed.

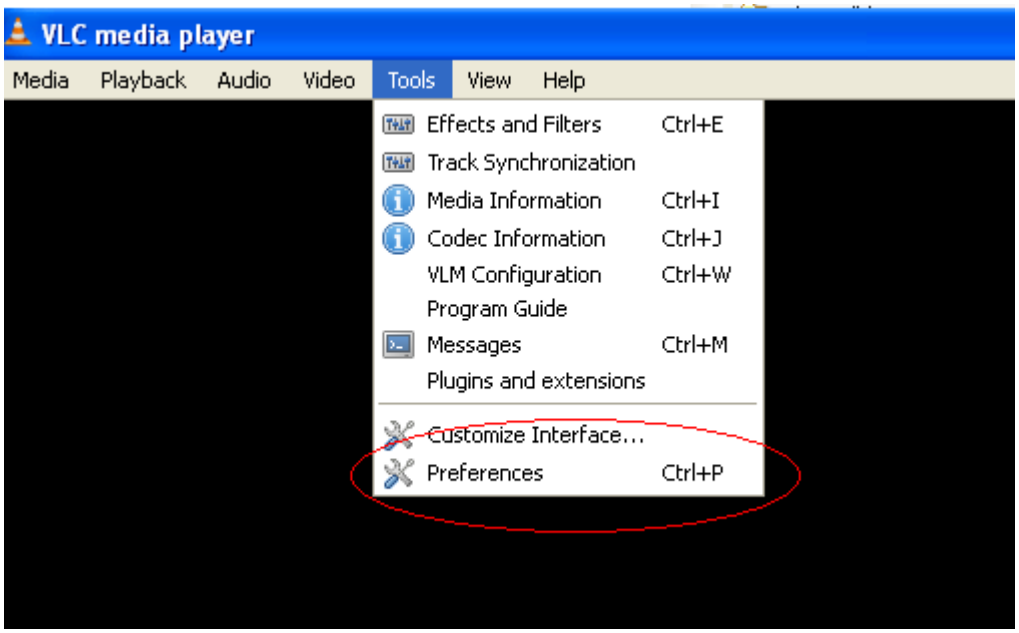

Figure 28. Tools → Preferences

2. In the Preferences dialog box, click the 'Reset Preferences' button (see Figure 29). This resets to default settings.

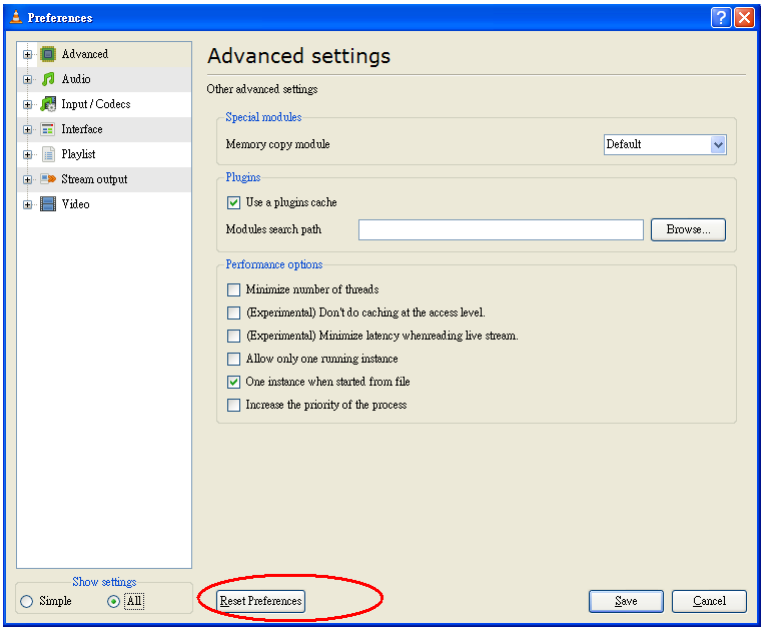

Figure 29. Reset Preferences

Copyright © 2016, Texas Instruments Incorporated

3. On the left pane, click 'Video' and then select 'Overlay video output' option (see Figure 30). Also, select 'All' options.

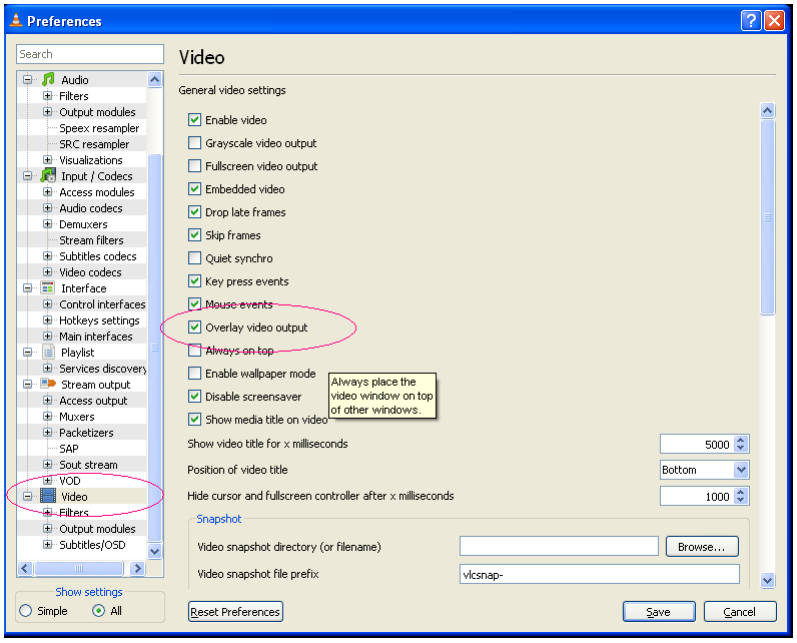

Figure 30. Overlay Video Output

4. On the left pane, click 'Input/Codecs' → 'NetworkCaching' to check the caching value. For low latency, it should be 250 to approximately 300 based on PC's performance.

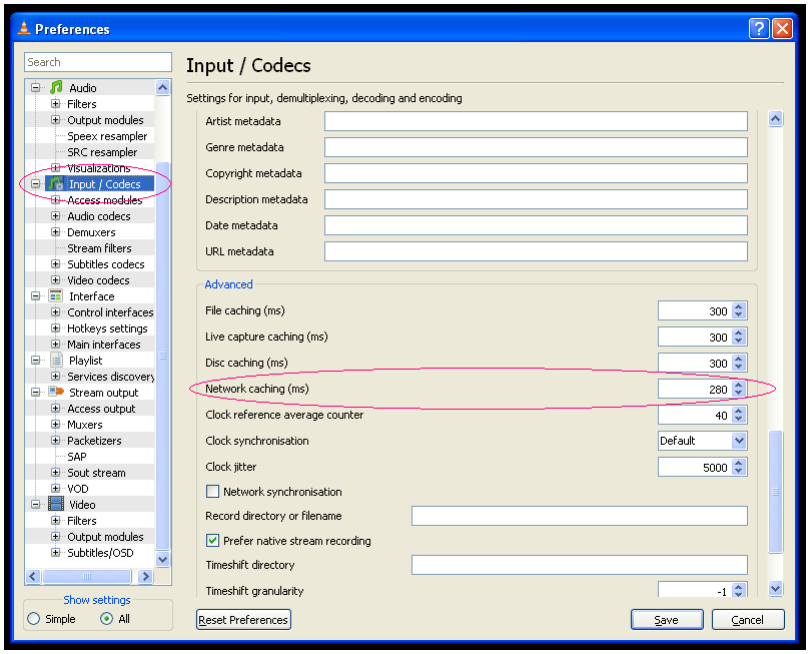

Figure 31. Input/Codecs

Copyright © 2016, Texas Instruments Incorporated

NOTE: If the video is not smooth, then set the value between 300 to approximately 1200. This may be due to the following:

- Network congestion
- Insufficient memory on PC
- Many processes running at the same time on the PC
- PC's performance is not meeting the minimum requirement

5. If you set the frame rate lower than 30 fps on the Web browser or set auto exposure on night mode, you should set higher value (150 to approximately 1200).
   (a) Install stopwatch application with mini seconds in PC.
   (b) Run the PC application and ensure that you get time message on the PC screen.
   (c) You can also access the stopwatch application from the shareware.
   (d) Face IPNC to the PC monitor with time message included.
   (e) Run the IPNC.
   Now the time message can be seen from the IPNC in VLC media player.

Latency : 3.876 – 3.735 = 141ms

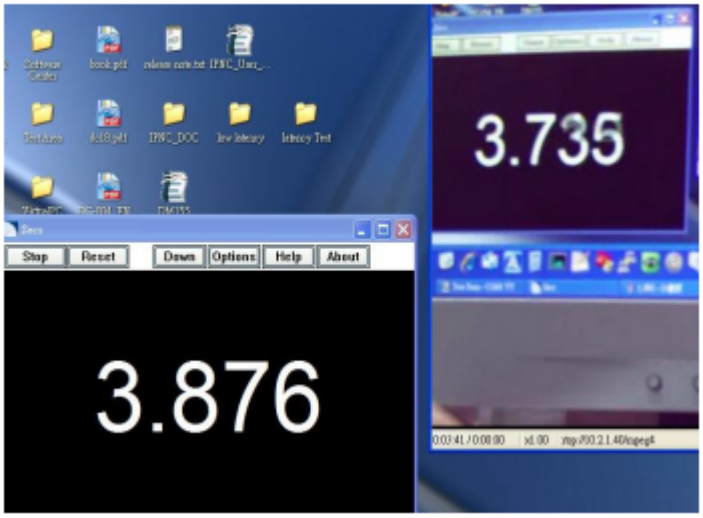

Figure 32. Latency Measurement

6. Subtract the 2 time messages from the PC application and the results through IPNC and the internet in VLC media player to get the latency result.

3.7 ARM® CPU Loading Measurement

To measure CPU loading (CPU usage) of the ARM core, use the arm_loading tool located in the /opt/ipnc directory.

Set up the system condition to be observed. For example, play one or two video streams on the PC client.

Invoke the arm_loading tool to start the measurement. For every one-second interval, a pair of numbers will be seen on the console screen as shown in Figure 21.

Copyright © 2016, Texas Instruments Incorporated

Time in seconds is on the left, CPU loading in percentage is on the right.

```
root@10.2.1.101:/opt/ipnc# ./arm_loading
0     0
1     22
2     21
3     17
4     23
5     21
0     25
```

By default, the test takes 100 frames to complete. You can also specify how many frames this should run at the command line. For example, ./arm_loading 1000 will ensure to run for 1000 frames.

NOTE: This utility is not supported in this release.

3.8 MJPEG Rate Control API Details

The following are details on the usage of the MJPEG rate control mode:

- MJPEG rate control option is available from Rules.Make
- MJPEG_RC_MODE in Rules.Make determines the rate control mode with which MJPEG will operate. Variable BItRate (VBR) and Constant BitRate (CBR) are supported. Having option as QPMODE will select the MJPEG encoding without RateControl using Quality factor set from GUI.
- To change Birate, QP params and other options, user need to change option in file Source/ipnc_rdk/ipnc_mcfw/demos/mcfw_api_demos/stream/stream.c file. Refer to Case STREAM_FEATURE_JPG_QUALITY in function stream_feature_setup,
- For setting BITRATE

```
params.targetBitRate = 8 * 1000 * 1000;
Venc_setDynamicParam(2, 0, &params, VENC_RATECONTROL);
```

- For setting rate control value

```
params.rateControl = RATECONTROL_VBR;
Venc_setDynamicParam(2, 0, &params, VENC_BITRATE);
```

- For setting QP rang value

```
/* if the QP param needs to be tuned, this needs to be used */
params.qpMin    = 1;
params.qpMax    = 51;
params.qpInit   = 28;
Venc_setDynamicParam(2, 0, &params, VENC_QPVAL_I);
```


Copyright © 2016, Texas Instruments Incorporated

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2016, Texas Instruments Incorporated