

MCIMX6Q-SDP Parallel Camera Server/Client Streaming Video

Author: Curtis Wald, Freescale Semiconductor Software FAE

Table of Contents

Introduction.....	1
Resources.....	1
Cameras.....	1
Software.....	2
Documentation.....	2
Linux Setup.....	2
I.MX6Q-SDP Login.....	2
Display Blanking.....	3
Load Linux Kernel Modules.....	3
Show Camera Video on I.MX6Q-SDP Display.....	4
Stream Camera Capture over IP Address.....	4
I.MX6Q-SDP Server	4
Linux Client.....	5

Introduction

The i.MX6Q-SDP (<http://freescale.com/sabresdp>) is a Quad Cortex-A9 Platform which features an Omni Vision OV5642 parallel camera. This paper describes the steps for creating a server using the I.MX6Q-SDP running a Gstreamer pipeline that captures the camera video encoding as motion JPEG using the hardware accelerated Video Processing Unit (VPU) and streaming encoded data to an IP address. A Linux host is the client that receives the stream and employs a Gstreamer command to decode the motion JPEG and display in a window.

Resources

[MCIMX6Q-SDP](#) SABRE Platform for Smart Devices Based on the i.MX 6 Series.

Cameras

The MCIMX6Q-SDP accepts two Omni Vision Camera Modules. For this paper only the parallel camera is used.

OV05640-MRFL-AA0T	MIPI Camera
OV05642-MRSL-A01A	Parallel Camera

Software

[L3.0.35_1.1.0_Source](#) Linux Freescale Board Support Package (BSP)

[L3.0.35_1.1.0_MM_CODECS](#) Linux Freescale Multimedia CODECS

Documentation

[L3.0.35_1.1.0_LINUXDOCS_BUNDLE](#) i.MX 6Quad and i.MX 6Dual Linux BSP Documentation. Includes Release Notes, Reference Manual,

[L3.0.35_1.1.0_LINUX_MMDOCS](#) Linux Multimedia Codecs Documentation. Includes CODECs Release Notes and User's Guide.

V4L2: Video for Linux Driver <http://www.kernel.org/doc/Documentation/video4linux/v4l2-framework.txt>

Linux Setup

Please refer to the Linux docs bundle and the Linux MMDOCS for creating the images to load and run on the SDP.

L3.0.35_1.1.0_Source provides the tools for building the images and all the source code for the boot loader, U-Boot, the Linux Kernel, and the minimum root file system.

L3.0.35_1.1.0_MM_CODECS provides the Gstreamer plug ins that utilize the i.MX6 acceleration CODEC hardware in addition to the basic Gstreamer commands.

I.MX6Q-SDP Login

The login credentials are:

User: root

Password: not required, no password set

```
arm-none-linux-gnueabi-gcc (Freescale MAD -- Linaro 2011.07 --  
Built at 2011/08/10 09:20) 4.6.2 20110630 (prerelease)  
root filesystem built on Wed, 16 Jan 2013 07:17:18 -0600  
Freescale Semiconductor, Inc.
```

```
freescale login: root
login[2814]: root login on 'ttymxc0'
root@freescale ~$
```

Display Blanking

The display has a blanking feature that is enabled after a few minutes. To re-enable the display issue the following command to disable blanking:

```
echo 0 > /sys/class/graphics/fb0/blank
```

Load Linux Kernel Modules

There are two Linux modules that are loaded, one for the camera and the other for the V4l2 driver.

The command `modprobe` is used to install the modules:

```
modprobe ov5642_camera
modprobe mxc_v4l2_capture
```

To see what modules are installed use the `lsmod` command:

```
root@freescale ~$ lsmod
Module                Size  Used by
mxc_v4l2_capture      21436  1
ipu_fg_overlay_sdc   4756   1 mxc_v4l2_capture
ipu_csi_enc          2933   1 mxc_v4l2_capture
ipu_prp_enc          4685   1 mxc_v4l2_capture
ipu_still            1703   1 mxc_v4l2_capture
ipu_bg_overlay_sdc   3921   1 mxc_v4l2_capture
ov5642_camera        74038  0
camera_sensor_clock   765    1 ov5642_camera
```

Show Camera Video on I.MX6Q-SDP Display

To display the camera video on the display use a gstreamer pipeline:

```
gst-launch mfw_v4lsrc ! mfw_v4lsink
```

The mfw_v4lsrc gstreamer plug-in is the v4l2-based camera sensor interface (CSI) camera video source.

The “!” provides the separator for the Gstreamer pipeline.

The mfw_v4lsink gstreamer plug-in description is to display video by using v4l2 interface.

Stream Camera Capture over IP Address

The I.MX6Q-SDP is the server and for this example has an IP address of 10.0.0.10. The Linux host is the client and has an IP address of 10.0.0.20.

Before running the command on the SDP server, ensure the Linux kernel modules are loaded. Refer to section “Load Linux Kernel Modules”.

I.MX6Q-SDP Server

```
gst-launch mfw_v4lsrc capture-mode=4 \  
! vpuenc codec=12 \  
! tcpserver sink host=10.0.0.20 port=5000
```

Gstreamer plug-in description:

Gstreamer Plug-in	Description
mfw_v4lsrc capture-mode=4	v4l2-based camera sensor interface (CSI) camera video source. The capture-mode definitions are found in the Linux driver: L3.0.35_1.1.0_121218/ltib/rpm/BUILD/linux-3.0.35/drivers/media/video/mxc/capture/ov5642.c enum ov5642_mode { ov5642_mode_MIN = 0, ov5642_mode_VGA_640_480 = 0, ov5642_mode_QVGA_320_240 = 1, ov5642_mode_NTSC_720_480 = 2, ov5642_mode_PAL_720_576 = 3, ov5642_mode_720P_1280_720 = 4,

Gstreamer Plug-in	Description
	ov5642_mode_1080P_1920_1080 = 5, ov5642_mode_QSXGA_2592_1944 = 6, ov5642_mode_QCIF_176_144 = 7, ov5642_mode_XGA_1024_768 = 8, ov5642_mode_MAX = 8 };
vpuenc codec=12	I.MX6Q VPU hardware JPEG encoder (12). Use the command “gst-inspect vpuenc” to see all encoders and number mapping.
tcpserver sink host=10.0.0.20 port=5000	Send data as a server over the network via TCP. The IP address is the client the data is sent to. The port is the assigned port number used.

Linux Client

On the Linux host which is the client receiving the data issue the following Gstreamer command:

```
gst-launch tcpclientsrc host=10.0.0.10 port=5000 ! jpegdec ! autovideosink
```

Gstreamer Plug-in	Description
tcpclientsrc host=10.0.0.10 port=5000	Receive data as a client over the network via TCP. The IP address is the server sending the data and using port 5000.
jpegdec	Decode images from JPEG format
autovideosink	Wrapper video sink for automatically detected video sink. A window is opened and decoded JPEG video is shown.