Texas Instruments

Application Report
SPRA937A − July 2004

JPEG Netcam2 on DM642 EVM

Video and Imaging Systems

ABSTRACT

This software demonstration combines real-time D1 Joint Photographic Experts Group (JPEG) encoding of images on the DM642 Evaluation Module (EVM) with networking functionality.

The JPEG standard pertains to compression of still images. Performing JPEG at the rate of 30 frames per second in isolation as individual images is known as motion JPEG (MJPEG). This demonstration uses:

- JPEG encoder library optimized for a DM642 EVM capable of real−time D1 encoding
- JPEG encoder library integrated with the IDMA layer specification
- JPEG encoder library implemented using XDAIS interfaces
- Sample integration of JPEG encoder and decoder under RF−5 framework, to demonstrate JPEG loopback (encode) at programmable quality
- Two different input streams

The addition of networking functionality allows an application to stream JPEG data in and over a network. Applications include security, web cameras, and point−to−point video distribution.

Contents

List of Figures

1 Software Architecture/Data Flow

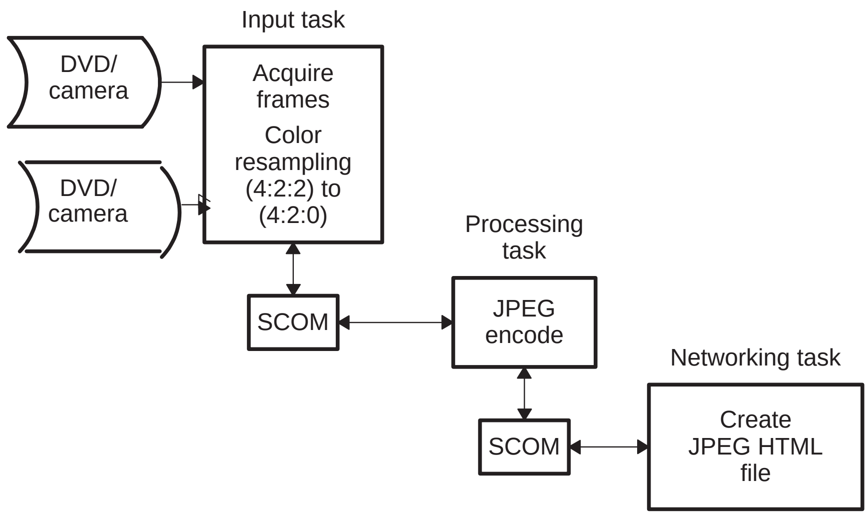

Figure 1. Data Flow Diagram

The data flow in the demonstration reflects the following sequence:

1. A frame is captured alternately from the input source1 and then input source 2 (DVD/Camera), and the acquired frame data, in YUV 4:2:2 format, is resampled to YUV 4:2:0 format.
2. The frame is fed from the input task to the processing task via a SCOM queue. The input source channel is noted in the SCOM message.
3. The JPEG encoder library generates a JPEG encoded image at a desired quality you set. The JPEG file is sent to the networking task via a SCOM queue.
4. The networking task creates an HTML file image of the JPEG, based on its source channel, for viewing over a WEB page.

1.1 Framework Flowchart

This demonstration uses RF-5 framework to integrate the JPEG encoder and decoder library as well as the networking task. The demonstration uses a five task setup. Three of these tasks are shown in the data flow diagram.

The fourth task is a control task, which uses a mailbox to send messages to the processing task. The processing task receives messages from this mailbox and adjusts the application parameters based on the nature of the message. Currently, only the changing of the quality of the encoded image is implemented.

The fifth task is the network initialization task. It is defined in the CDB and manages initializing the networking environment. Once the network is ready, the networking task shown in the data flow diagram is created.

Before coming to the DSP/BIOS™ task scheduler, the demonstration code initializes various modules used in the system. These include:

- Board and processor
  - DSP/BIOS™ initialization and CSL initialization.
  - L2 cache mode is set to 128K cache.
  - EMIFA CE0 and EMIFA CE1 spaces are enabled for caching.
  - DMA priority queue lengths are set to maximum.
  - L2 request priority set as high.
  - DMA manager is initialized with allocated internal and external heap.
- RF-5 modules
  - The system initializes the channel module of RF-5.
  - The system initializes the ICC and SCOM modules of RF-5 required for intercell communication and messaging.
  - Channel setup is performed with the internal, external heap buffers.
- Capture and display channels
  - An instance of capture channel is created and started.
  - An instance of display channel is created and started.

After these initializations, the system enters the five-task system managed by the DSP/BIOS™ scheduler. The tasks use SCOM module of RF-5 to communicate to each other. The five tasks are as follows:

- Input task

The input task acquires the frames from the NTSC/PAL input device. It uses FVID_exchange calls provided by the driver to acquire a frame. The acquired frame is in YUV 4:2:2 format.It is resampled to YUV 4:2:0 format. It then sends a SCOM message to the processing task with the frame pointer embedded in the message. The task then waits for its message to come back from the processing task to continue.

- Processing task

The processing task has one cell in this example. This cell is a JPEG encoder cell that accepts an input YUV 4:2:0 image and produces a compressed JPEG image at a desired quality. The processing task passes the JPEG file to the networking task using an SCOM message.

- Control task

The control task controls the variable parameters within this demonstration. These include the quality factor of the demonstration. The control task checks for a change in value in the parameters defined in a global structure, ExternalControl, (visible to the user in global space). It then copies the values of the changed parameters into a local structure, externalControlPrev and posts messages in a mailbox to the process thread. The processing task periodically checks for messages and calls the corresponding cell’s control function.

- Network initialization task

The network initialization task boots up the networking environment. When the network is ready, the initialization task creates the networking task. The initialization source code for all networking tasks is remarkably similar. Refer to the TMS320C6000 TCP/IP Network Developer's Kit (NDK) User's Guide (SPRU523).

- Networking task

The networking task performs any network functionality required in the system. When the task is initialized, it waits for a SCOM message from the processing task.

In this example, the networking task takes the JPEG image supplied from the Processing task and creates a RAM-based image file (IMAGE*n*.JPG) that the HTTP server recognizes and can serve to HTTP clients. The name of the image file (IMAGE1.JPG or IMAGE2.JPG) is determined by the source channel number of the original captured image. It then sends a SCOM message back to the processing task that it is ready for another JPEG image.

In a more complex system where network data is streamed over sockets, the same task thread structure is used. It is important that networking be handled by a separate task since it spends most of its time in a blocked state by its nature. Thus, during the transmission of a block of network data, the DSP is mostly free. Putting networking in a sparate task allows those free MIPS to be utilized by other processes..

2 System Requirements/Configuration

2.1 Software Requirements

- Microsoft Windows NT (SP6)/Microsoft Windows 2000 (SP1 and SP2)/Microsoft Windows XP
- Code Composer Studio™ Integrated Development Environment (IDE) version 2.21 or greater
- Reference Frameworks (RF 2.20)

2.2 Hardware Requirements

- Pentium machines with 450 MHz, 64MB RAM (minimum)
- DM642 EVM
- NTSC/PAL TV for display purposes
- Camera/DVD for NTSC/PAL capture purposes
- XDS 510/560 emulator

3 Hardware Setup

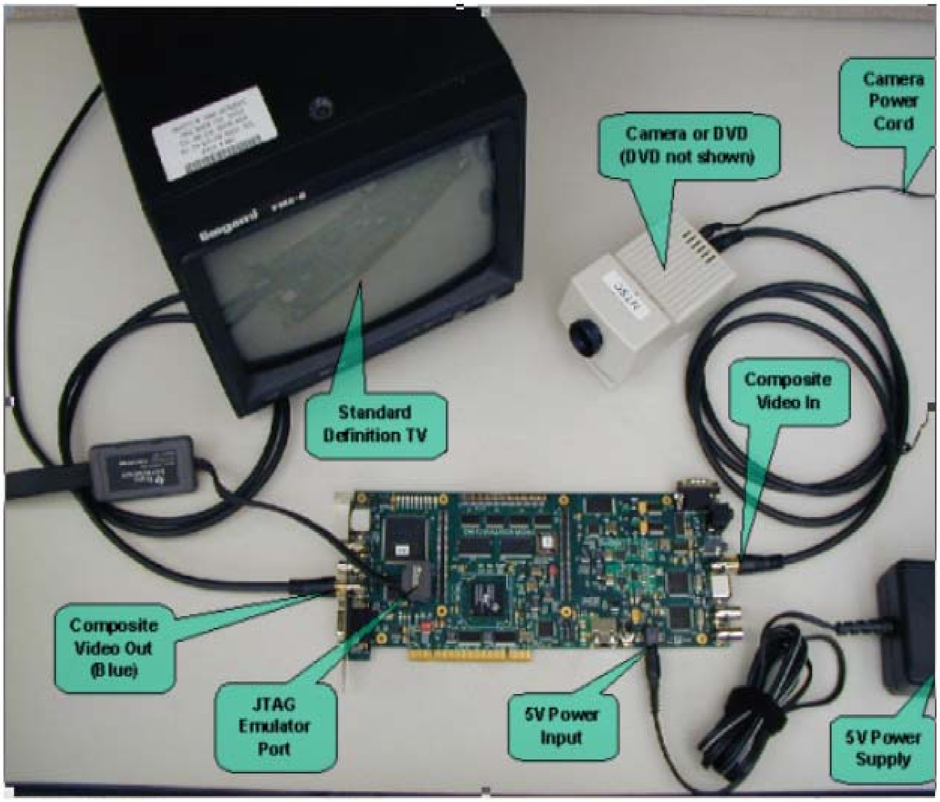

Figure 2. Hardware Setup

To run the demonstration, set up the hardware as shown in Figure 2:

- Connect the DM642 EVM to the appropriate power source.
- Connect the input video port (for composite video) to NTSC/PAL input source (DVD/camera) using RCA cable.
- Connect the output video port (for composite video) to NTSC/PAL output device (SDTV).
- Connect the XDX510/560 emulator to the JTAG pins to download the demonstration code to the board and control it from Code Composer Studio™ IDE.

4 Demonstration Execution

NOTE: Some of the example applications described in this section require a network with support for DHCP. If DHCP is not available, the example must be rebuilt to use a fixed IP configuration. This can be done using Code Composer Studio™ IDE. Refer to the TMS320C6000 TCP/IP Network Developer's Kit (NDK) User's Guide (SPRU523) for details on network application initialization.

To run the demonstration, follow these steps:

1. Connect the NTSC/PAL input device (camera/DVD) using proper RCA cables as shown in the hardware setup figure.
2. Connect the NTSC/PAL output device (SDTV) using proper RCA cables as shown in the hardware setup figure.
3. Power up the EVMDM642 EVM board.
4. Start Code Composer Studio™ IDE version 2.21 or greater.
5. Check the color bar on the output device.
6. Go to the bin folder under the jpeg_netcam2 directory and load jpeg_netcam2_NTSC.out or jpeg_netcam2_PAL.out.
7. Once the program is loaded, go to the Debug Menu in Code Composer Studio ™ IDE and press the Run option (F5).
8. Point a Web browser to the EVM's IP address. Select one of the two input channels, and watch the reconstructed JPEG image, with the TI logo at the top-right corner of the screen.

On a successful execution, one of the status lines printed by the application displays the client IP address (either through DHCP or static configuration). Then, the DSP responds to requests made to its IP address. When using DHCP, the application may be unable to obtain an IP address from a DHCP server. If this happens, the application eventually prints a DHCP status message with the fault condition message.

Once the networking task is ready and an IP address is obtained, you can connect an HTTP browser to that address to see the NETCAM images displayed on the browser window.

NOTE: The simple JAVA application used to implement the NETCAM is not compatible with all JVM's. Specifically, JVM's that cache display images will continuously display the same image. Caching can usually be disabled in the JVM configuration.

4.1 Demonstration Controlling Parameters

- On the Web page with the NETCAM display is a setting that allows the quality of the resulting JPEG decoded image to be varied. The allowed values for quality are of the form 1 <= quality <= 100 with both end values being legal.
- Another setting allows the selection of the desired input channel.

5 Demonstration Code and Build Procedure

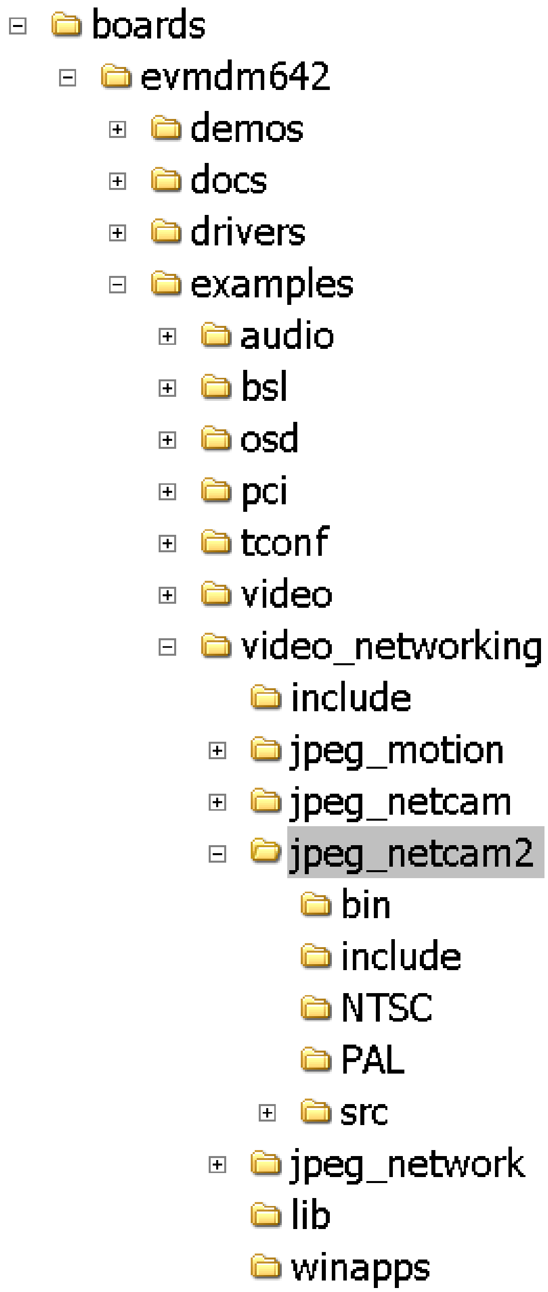

Figure 3. Directory Structure

5.1 Build Procedure

1. Start Code Composer Studio™ IDE version 2.21 or greater.
2. Open the jpeg_netcam2 project (jpeg_netcam2.pjt) in the folder called boards\examples\video_networking\jpeg_netcam2.
3. Go to Project−>Build and rebuild the project.
4. Compiler options used under the pre-processor are CHIP_DM642 = 1, C6000, and UTL_DBGLEVEL = 70.
5. Build the project and load the executable from the bin directory: jpeg_netcam2_NTSC.out or jpeg_netcam2_PAL.out.
6. Press F5 to watch the decoded JPEG image on a Web browser.

6 Known Bugs and Constraints

- The quality factor must be in the range 1 to 100.
- Decoder checks if decoded image is a valid JPEG and returns a negative error code on receiving an incorrect JPEG stream.

7 JPEG Encoder and JPEG Decoder Performance

The performance of JPEG encoder and decoder images is content− and quality−dependent. The figures shown here are for a quality setting of 75 and reasonable complexity images, which are typical.

- D1 4:2:0 encode for typical images at a quality setting of 75% uses 23% of a 600−mHz C64x DSP.
- D1 4:2:0 decode for typical images at a quality setting of 75% uses 20% of a 600−mHz C64x DSP.

8 References

TMS320C6000 TCP/IP Network Developer’s Kit (NDK) User’s Guide (SPRU523)

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

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

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI 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 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. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

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

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

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