

# **JPEG with Motion Detection on the DM642 EVM**

---

*Video and Imaging Systems*

## **ABSTRACT**

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

The JPEG standard pertains to compression of still images. Performing JPEG compression 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 decoder library optimized for a DM642 EVM capable of real-time D1 decoding
- JPEG encoder and decoder library integrated with the IDMA layer specification
- JPEG encoder and decoder library implemented using XDAIS interfaces
- Sample integration of JPEG encoder and decoder under RF-5 framework, to demonstrate JPEG loopback (encode + decode) at programmable quality
- Sample integration of an inline networking module that allows video to be sent to a networking peer or to replace local video with video received by a networking peer
- A Windows "DOS box" example application to record frames with activity and play them back, and send other simple commands to the DSP

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**

<b>1</b>	<b>Software Architecture/Data Flow</b> .....	<b>3</b>
	1.1 Data Flow Diagram for the Demonstration .....	3
	1.2 Framework Flowchart .....	4
<b>2</b>	<b>System Requirements/Configuration</b> .....	<b>6</b>
	2.1 Software Requirements .....	6
	2.2 Hardware Requirements .....	6
<b>3</b>	<b>Hardware Setup</b> .....	<b>6</b>
<b>4</b>	<b>Demonstration Execution</b> .....	<b>7</b>
	4.1 Controlling Parameters of the Demonstration .....	8
<b>5</b>	<b>Demonstration Code and Build Procedure</b> .....	<b>9</b>

Trademarks are the property of their respective owners.

5.1	Build Procedure .....	10
<b>6</b>	<b>Known Bugs and Constraints .....</b>	<b>10</b>
<b>7</b>	<b>Performance of JPEG Encoder and JPEG Decoder .....</b>	<b>10</b>
<b>8</b>	<b>References .....</b>	<b>10</b>

### List of Figures

Figure 1.	Data Flow Diagram .....	3
Figure 2.	Hardware Setup .....	7
Figure 3.	Directory Structure for jpeg_motion .....	9

# 1 Software Architecture/Data Flow

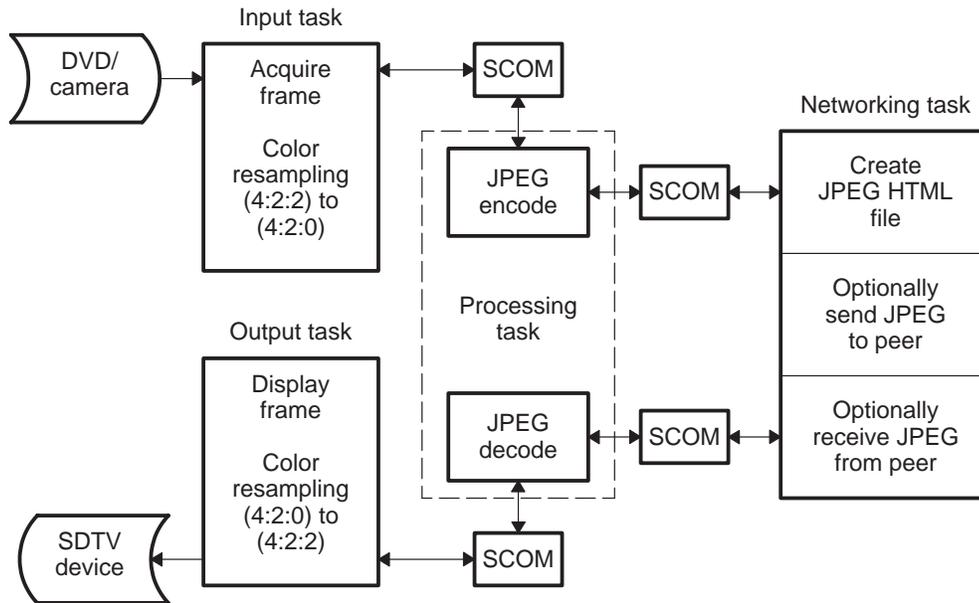


Figure 1. Data Flow Diagram

## 1.1 Data Flow Diagram for the Demonstration

The data flow in the demonstration follows this sequence:

1. A frame is captured from the input source (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 an SCOM queue.
3. The JPEG encoder library generates a JPEG–encoded image at a desired quality set by the user. The raw image is also checked for activity by comparing it to the previous image. The activity (in tenths of percent) is recorded with the JPEG image and is sent to the networking task via SCOM.
4. The networking task creates a local copy of the JPEG for NETCAM functionality. If there is a network peer connected for record and there is activity in the image, the networking task sends the image to a network peer.
5. If there is a network peer connected for playback and the networking task receives a new JPEG image from the network peer that replaces the image supplied by the JPEG encoder, the networking task then sends either the original JPEG or the newly received JPEG back to the processing task via SCOM.
6. The JPEG file is then sent to the JPEG decoder. The decoded JPEG image is fed to the output task via an SCOM queue.
7. The output task converts the decoded YUV 4:2:0 image to a YUV 4:2:2 image and sends it to the TV monitor.

## 1.2 Framework Flowchart

This demonstration uses RF-5 framework to integrate the JPEG encoder and decoder library and the networking task. It also uses a six-task setup, four of which are shown in Figure 1.

The fifth 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 changes the application parameters based on the nature of the message. Currently, only the changing of the quality of the encoded image is implemented.

The sixth 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 is created (see Figure 1).

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

- Board and processor
  - The system performs DSP/BIOS™ initialization and CSL initialization.
  - The L2 cache mode is set to 128K cache.
  - EMIFA CE0 and EMIFA CE1 spaces are enabled for caching.
  - The DMA priority queue lengths are set to maximum.
  - Priority for L2 request is set as high.
  - The 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 and 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 six-task system managed by the DSP/BIOS™ scheduler. These six tasks use the SCOM module of RF-5 to communicate with each other:

- Input task

The input task is responsible for acquiring 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 and is resampled to YUV 4:2:0 format. It ends the message to the processing task with the frame pointer embedded in the message. The task then waits for a message from the processing task to continue.

- Processing task

The processing task has two cells in this demonstration. The first cell is a JPEG encoder cell that accepts an input YUV 4:2:0 image and produces a compressed JPEG image of a desired quality. The second cell is a JPEG decoder cell that accepts a JPEG-encoded image and produces a decoded image. The decoded image produced by the JPEG decoder is in YUV4:2:0 format.

The first responsibility of the processing task is to add the time and date stamp, when enabled, to the input image. Next, the image is passed to the JPEG encoder cell. After the JPEG image is created, the source image is checked for activity, using a simple pixel compare on a fixed grid. The JPEG image and the activity rating are passed to the networking task for further processing.

When the networking task is completed, it passes back a JPEG image to the processing task for JPEG decode. The image may be the exact image passed to the networking task, or it may be an image retrieved off the network. After the completion of the JPEG decode, the processing task displays the activity grid, when enabled, on the output image. The output image is then passed to the output task via a SCOM message.

- Output task

The output task is responsible for displaying the frames on the NTSC/PAL output device. It uses FVID\_exchange calls provided by the driver to display a frame on the NTSC/PAL output device. The acquired frame is in YUV 4:2:0 format and is resampled to YUV 4:2:2 format prior to display. The task then waits for a message from the processing task to continue.

- Control task

The control task is responsible for controlling the parameters that are variable 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 is charged with booting 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) for an in-depth discussion.

- Networking task

The networking task is used to perform any network functionality required in the system. When the task is first initialized it creates a socket to listen on two ports, 3001 and 3002. Port 3001 is used for playback connections when the client wants to send a video stream to the DSP. Port 3002 is used for record connections when the client wants to receive a video stream from the DSP.

The networking task then waits for a SCOM message from the processing task indicating that a new JPEG image is available. The networking task takes the JPEG image and creates a RAM-based image file (IMAGE1.JPG) that the HTTP server recognizes and can serve to HTTP clients.

When a record connection is available, the networking task checks the socket to see if the client has sent any commands. Commands include setting the time and date, whether to display the time and date in the input image, and whether to display the activity grid in the output image. Next, the JPEG image received from the processing task is checked for activity. If there is activity, the image is sent over the record connection. Otherwise, a NULL file size indicator is sent over the record connection to keep the peer synchronized.

Next, if a play connection is available, a new JPEG image is received from this connection. The new JPEG image replaces that passed to the networking task by the processing task.

The networking task then passes the current JPEG image back to the processing task via a SCOM message.

In this system, there is only one networking task. This is possible since the DSP can assume that the client can source and consume an infinite amount of networking bandwidth. In other words, the DSP always assumes the client is ready to send or ready to receive. In other systems (such as a peer-to-peer streaming example), two networking tasks are used: one hooked to JPEG encode for transmit, and the other hooked to JPEG decode for receive. That is because record and playback streams are not necessarily synchronous (as this example assumes).

## 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

To run the demonstration, the hardware must be set up properly, as shown in Figure 2.

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

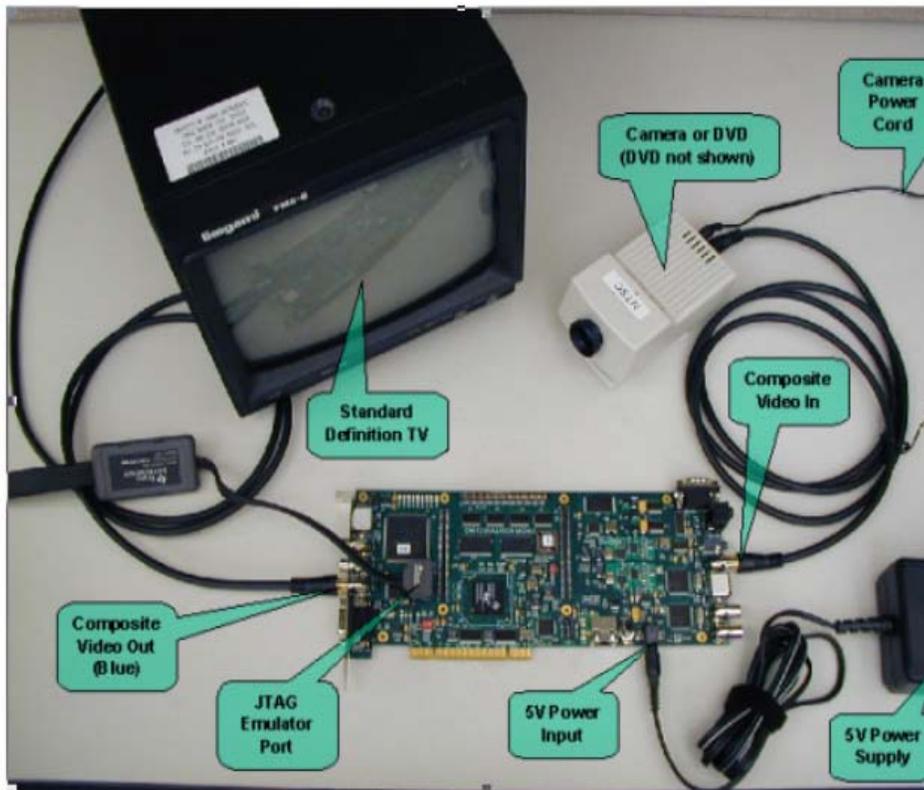


Figure 2. Hardware Setup

## 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:

1. Connect the NTSC/PAL input device (camera/DVD) using proper RCA cables.
2. Connect the NTSC/PAL output device (SDTV) using proper RCA cables.
3. Power up the DM642 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\_motion directory and load jpeg\_motion\_NTSC.out or jpeg\_motion\_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. On the output screen, watch the JPEG reconstructed image with the TI logo at the top-right corner of the screen.

In a successful execution, one of the status lines printed by the application displays the client IP address (either through DHCP or static configuration). Once this address displays, the DSP responds to requests made to its IP address. When using DHCP, it is possible that the application will be unable to obtain an IP address from a DHCP server. In that case, the application eventually prints a DHCP status message with the fault condition.

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.

To record and play back video sequences, a client application is run from a Windows DOS box. From the command line, run the program `mclient` located in the `\winapps` directory with the IP address and time zone adjustment as arguments. For example, if the IP address is 192.168.1.45 and you are at GMT -5 hours (CST), type:

```
mclient 192.168.1.45 5
```

Once the client is connected, the current date and time appear in the EVM output display. Press the spacebar to get a list of `mclient` commands. Using the keyboard, you can record, playback the recorded images, frame step through the previously recorded images as they playback, toggle the date and time display, and toggle the activity grid.

NOTE: `mclient` only records images with activity.

Images recorded by `mclient` are played back only on the output display. Regardless of what you do via `mclient`, the NETCAM always displays the source image (with or without time and date stamp).

NOTE: Do not run NETCAM and `mclient` on the same PC. The NETCAM java application uses up a lot of PC CPU cycles. If you want to run them simultaneously, use two PCs.

## 4.1 Controlling Parameters of the Demonstration

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  $1 \leq \text{quality} \leq 100$  with both end values being legal.

## 5 Demonstration Code and Build Procedure

The directory structure for jpeg\_motion is located under boards\evmdm642\examples\video\_networking (see Figure 3).

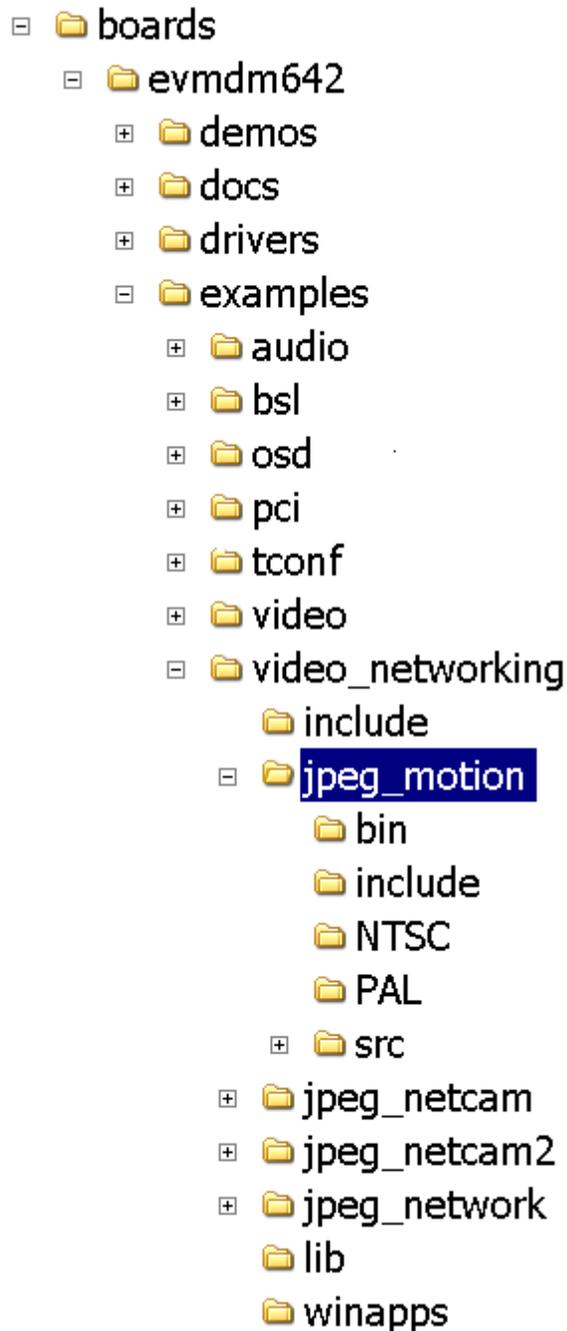


Figure 3. Directory Structure for jpeg\_motion

## 5.1 Build Procedure

1. Start Code Composer Studio™ IDE version 2.21 or greater.
2. Open the jpeg\_motion project (jpeg\_motion.pjt) in the folder called examples\video\_networking\jpeg\_motion.
3. Go to Project->Build and rebuild the project.
4. Compiler options used under the preprocessor are CHIP\_DM642 = 1, C6000, and UTL\_DBGLEVEL = 70.
5. Build the project and load the executable from the bin directory: jpeg\_motion\_NTSC.out or jpeg\_motion\_PAL.out
6. Press F5 to watch the decoded JPEG image.

## 6 Known Bugs and Constraints

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

## 7 Performance of JPEG Encoder and JPEG Decoder

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:

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

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

Copyright © 2004, Texas Instruments Incorporated