MICROCHIP

AN1415

Video Playback and Streaming Solutions Using the PIC® MCU

INTRODUCTION

The required bandwidth for data access from off-chip components can be a bottleneck in the system performance for microcontroller (MCU) video applications. This restriction translates into limiting the number of read and write accesses to the external memory device during a given time period. A popular solution is to use a compression algorithm. The quality of the video depends on, and in most cases, is directly proportional to the complexity of the compression algorithm. Another solution is to adhere to the bandwidth limitations by reading a lower resolution video and to rescale it (i.e., upscaling) during display. There are several methods of upscaling a video.

This application note describes four methods of playing a video file using a PIC® microcontroller (MCU) and hardware solutions available from Microchip. Two major constraints for these solutions are the bandwidth required for the data access and the processor speed. For example, the first method uses video upscaling to achieve the desired performance, while the second method takes advantage of the LCD controller on the Multimedia Expansion Board (MEB) to write the video data directly to the frame buffer. Both of these methods read a video file from a Secure Digital (SD) card and display it on a QVGA LCD screen.

METHOD 1: VIDEO PLAYBACK ON PIC32 USING UPSCALING BY INTERPOLATION

The data throughput required to play the video file can be calculated using the formula in Equation 1.

EQUATION 1:

$$BPS = HRES \cdot VRES \cdot FPS \cdot BPP$$

Where,

BPS = bits per second

$HRES$ = Horizontal resolution

$VRES$ = Vertical resolution

FPS = Frames per second

BPP = bits per pixel

For uncompressed QVGA video at 30 frames per second and 16 bits per pixel, the bandwidth required is calculated, as shown in Equation 2.

EQUATION 2:

$$BPS = 320 \cdot 240 \cdot 30 \cdot 16 \cdot = 36.8Mbps$$

Using the linear interpolation technique, we can reduce this requirement, as shown in Equation 3.

EQUATION 3:

$$BPS = \left(\frac{320}{2}\right) \cdot \left(\frac{240}{2}\right) \cdot 30 \cdot 16 = 9.2Mbps$$

Interpolation is the technique of using known data to estimate values of unknown data. The known data is the data with smaller resolution. The unknown data is the difference in the data between the smaller and the higher resolution images. Figure 1 (A) represents the 16 pixels in an image with a 4 x 4 grid. Figure 1 (B) represents the 64 pixels in an image with an 8 x 8 grid. Each of the pixels in Figure 1 (A) is interpolated in 2D to obtain four pixels in Figure 1 (B). This is the nearest neighbor interpolation. Since this is a linear interpolation technique, this has the least computation cost. This technique assumes high correlation in spatial locality of the same image with different resolutions. Figure 2 shows the visual representation of the linear interpolation technique.

FIGURE 1: MATRIX OF ORIGINAL PIXELS (A) AND INTERPOLATED PIXELS (B)

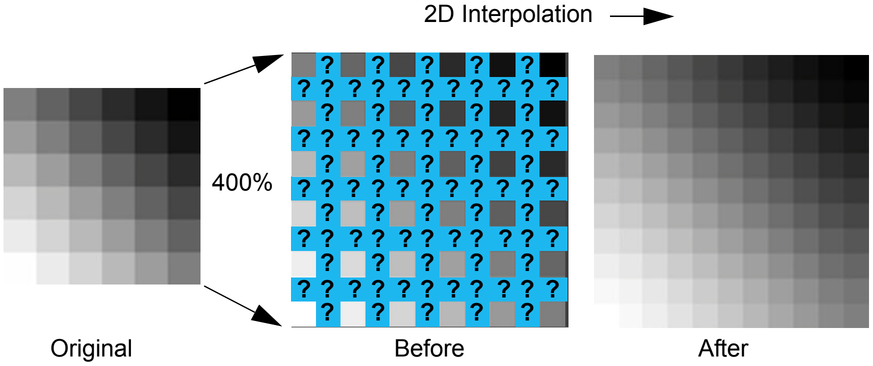
$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$$

(A)

$$\begin{bmatrix} 1 & 1 & 2 & 2 & 3 & 3 & 4 & 4 \\ 1 & 1 & 2 & 2 & 3 & 3 & 4 & 4 \\ 5 & 5 & 6 & 6 & 7 & 7 & 8 & 8 \\ 5 & 5 & 6 & 6 & 7 & 7 & 8 & 8 \\ 9 & 9 & 10 & 10 & 11 & 11 & 12 & 12 \\ 9 & 9 & 10 & 10 & 11 & 11 & 12 & 12 \\ 13 & 13 & 14 & 14 & 15 & 15 & 16 & 16 \\ 13 & 13 & 14 & 14 & 15 & 15 & 16 & 16 \end{bmatrix}$$

(B)

FIGURE 2: LINEAR INTERPOLATION OF PIXELS



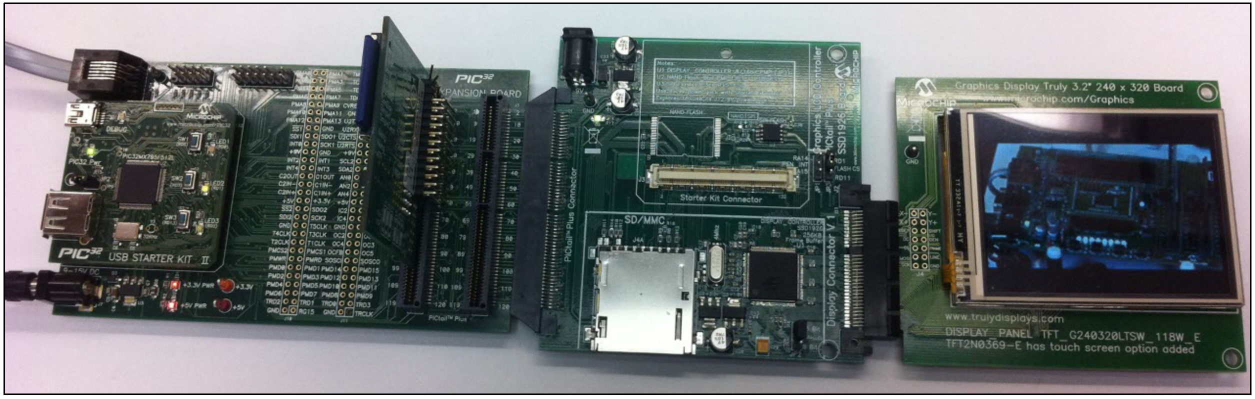
Hardware Requirements

- Starter Kit I/O Expansion Board (DM320002)
- PIC32 Starter Kit: GP (DM320001), USB II (DM320003), or Ethernet (DM320004)
- PICtail™ Daughter Board for SD™ and MMC Cards (AC164122)
- Secure Digital (SD) card
- Graphics LCD Controller PICtail™ Plus SSD1926 Board (AC164127-5)
- Graphics Display Truly 3.2" 240 x 320 Board (AC164127-4)

Board Setup

Figure 3 shows the hardware setup used for Method 1. The SD card that is loaded with the prepared video is inserted into the PICtail Daughter Board for SD and MMC Cards, which is connected to the Starter Kit I/O Expansion Board.

FIGURE 3:



Preparing the Video File

This demonstration plays back an AVI video file from the PICtail Daughter Board for SD and MMC Cards. To ensure proper playback, the video must be formatted correctly before being copied to the SD card. The video format must be:

- 16 BPP RGB
- 160 x 120 (will be upscaled in software)
- No sound

There are several open source software packages available to process and convert video files. In this demonstration example the ffmpeg application is used.

The following procedure describes how to resize a AVI (`.avi`) file using ffmpeg.

1. Download and install ffmpeg from: http://ffmpeg.org/download.html
2. Resize the file using ffmpeg:
    a) In Windows, open a command shell using `cmd.exe`.
    b) Change to the directory that contains your `.avi` file.
    c) In the Command shell, type:

```
ffmpeg -i <fn>.avi -s 160x120
-pix_fmt rgb565 <fn>.avi
```

    where `<fn>` is the prefix of the file name to convert. Figure 4 shows an example dialog.

3. Copy the resulting file to an SD card that has been formatted for the FAT16 file system.
4. Insert the SD card into the PICtail Daughter Board for SD and MMC Cards.

You are now ready to run the demonstration.

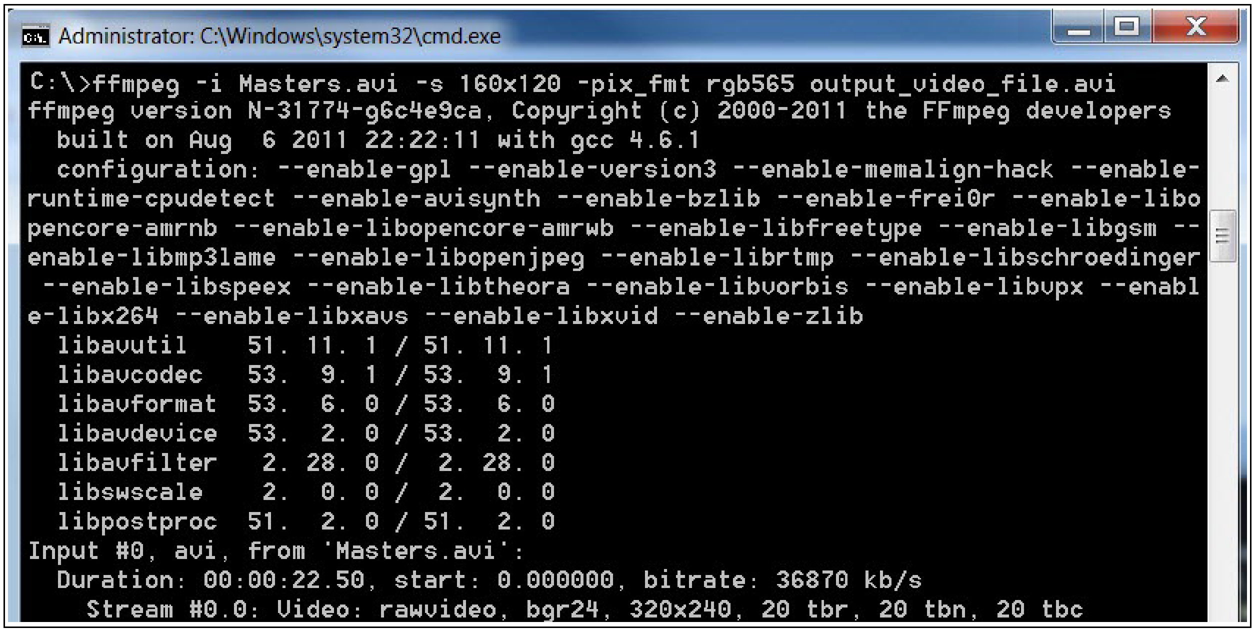
Operation

The Graphical User Interface (GUI) is based on the Microchip Graphics Library and is implemented with the Microchip Graphics Display Designer. The SD card is detected and the file system is initialized using the Microchip Memory Disk Drive (MDD) library. A list of playable files is then shown on the LCD screen. The user selects the file to be played by touching a filename on the touchscreen.

The SPI controller is initialized for data transfer. Data is read from the SD card using the Serial Peripheral Interface (SPI) port. To maximize data throughput, the SD bit clock is running at the maximum supported by the standard, 25 MHz. The data is sent to the SSD1926 using the 16-bit Parallel Master Port (PMP).

Direct Memory Access (DMA) is used to read data from the SD card. Data is read from the SD card using the SPI interface into the SPI buffer, which is monitored by the DMA, thus saving valuable processor time. The SPI buffer is further duplicated into ping-pong buffers. Ping-pong buffers are coupled such that while data is written into one buffer, data is sent to the display from the other buffer. This is done to reduce “sit-on” read time. For example, if 2 bytes of data are to be transferred from the SD card to the display, the first byte can be written into one of the ping-pong buffers. While the second byte is written into another buffer, data can be read into the display from the first buffer.

FIGURE 4:



```
C:\>ffmpeg -i Masters.avi -s 160x120 -pix_fmt rgb565 output_video_file.avi
ffmpeg version N-31774-g6c4e9ca, Copyright (c) 2000-2011 the FFmpeg developers
  built on Aug  6 2011 22:22:11 with gcc 4.6.1
  configuration: --enable-gpl --enable-version3 --enable-memalign-hack --enable-
runtime-cpudetect --enable-avisynth --enable-bzlib --enable-frei0r --enable-libo
pencore-amrnb --enable-libopencore-amrwb --enable-libfreetype --enable-libgsm --
enable-libmp3lame --enable-libopenjpeg --enable-librtmp --enable-libschroedinger
 --enable-libspeex --enable-libtheora --enable-libvorbis --enable-libvpx --enabl
e-libx264 --enable-libxavs --enable-libxvid --enable-zlib
  libavutil    51. 11. 1 / 51. 11. 1
  libavcodec   53.  9. 1 / 53.  9. 1
  libavformat  53.  6. 0 / 53.  6. 0
  libavdevice  53.  2. 0 / 53.  2. 0
  libavfilter   2. 28. 0 /  2. 28. 0
  libswscale    2.  0. 0 /  2.  0. 0
  libpostproc  51.  2. 0 / 51.  2. 0
Input #0, avi, from 'Masters.avi':
  Duration: 00:00:22.50, start: 0.000000, bitrate: 36870 kb/s
    Stream #0.0: Video: rawvideo, bgr24, 320x240, 20 tbr, 20 tbn, 20 tbc
```

METHOD 2: STREAMING RAW VIDEO FROM A micro-SD CARD USING THE SSD1926 LCD GRAPHICS CONTROLLER ON THE MEB

This section describes the implementation of video playback from a micro-SD card using the Multimedia Expansion Board (MEB) and a PIC32/PIC24/dsPIC® DSC starter kit. The demonstration will play a raw video file in RGB565 format from a micro-SD card to the QVGA LCD on the MEB.

Required Hardware

- Multimedia Expansion Board (DM320005)
- A PIC32 starter kit (GP, Ethernet, or USB II)/ PIC24 starter kit/dsPIC DSC starter kit
- micro-SD Card
- Standard-A to mini-B USB cable

Required Software

- MPLAB® 8.x or later
- Raw video file

Preparing the Video File

This demonstration opens a raw video file stored on a micro-SD card and plays it on the QVGA LCD of the MEB. Several software packages are available to convert video files from one type to another. This example describes how to convert a WMV (`.wmv`) file to a raw video file using `ffmpeg`.

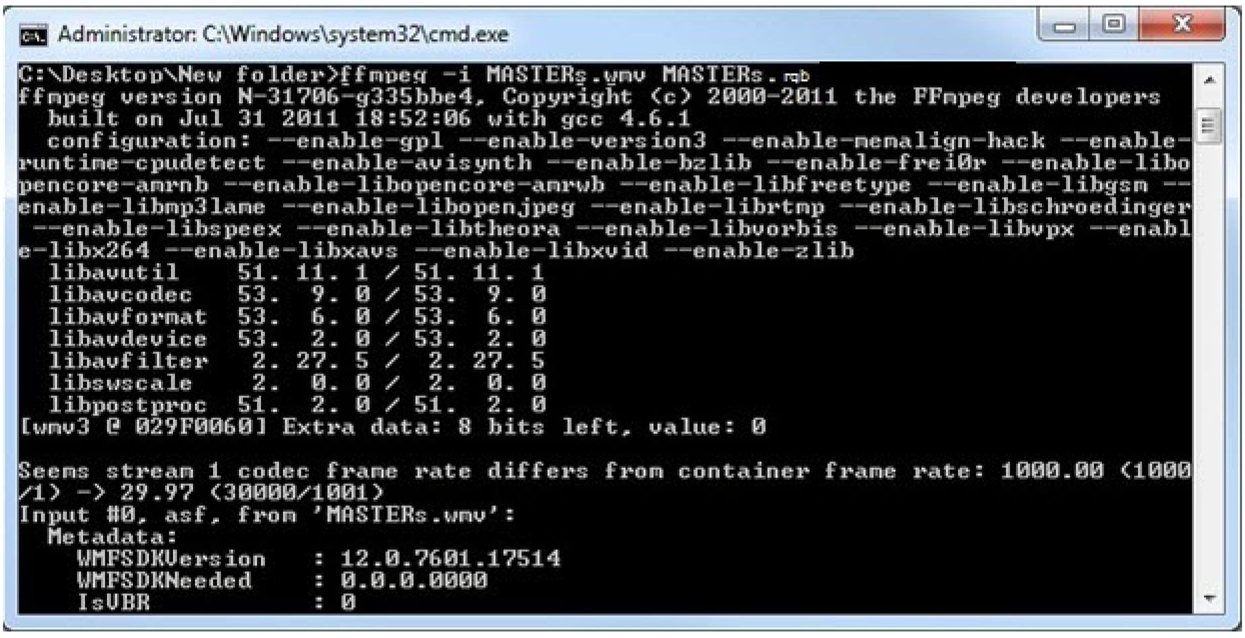
1. Download and install ffmpeg from: http://ffmpeg.org/download.html
2. Convert the file to RGB565 format:
   a) In Windows, open a command shell using `cmd.exe`.
   b) Change to the directory that contains your `.wmv` file.
   c) In the Command shell, type:

   ```
   ffmpeg -i <fn>.wmv -s 320x240
   -pix_fmt rgb565 <fn>.rgb
   ```

   where `<fn>` is the prefix of the file name to convert. Figure 5 shows an example dialog.
3. Copy the resulting file to a micro-SD card that has been formatted for the FAT16 file system.
4. Insert the card into the MEB.

You are now ready to run the demonstration.

FIGURE 5:



```
C:\Desktop\New folder>ffmpeg -i MASTERs.wmv MASTERs.rgb
ffmpeg version N-31706-g335bbe4, Copyright (c) 2000-2011 the FFmpeg developers
  built on Jul 31 2011 18:52:06 with gcc 4.6.1
  configuration: --enable-gpl --enable-version3 --enable-memalign-hack --enable-
runtime-cpudetect --enable-avisynth --enable-bzlib --enable-frei0r --enable-libo
pencore-amrnb --enable-libopencore-amrwb --enable-libfreetype --enable-libgsm --
enable-libmp3lame --enable-libopenjpeg --enable-librtmp --enable-libschroedinger
 --enable-libspeex --enable-libtheora --enable-libvorbis --enable-libvpx --enabl
e-libx264 --enable-libxavs --enable-libxvid --enable-zlib
  libavutil    51. 11. 1 / 51. 11. 1
  libavcodec   53.  9. 0 / 53.  9. 0
  libavformat  53.  6. 0 / 53.  6. 0
  libavdevice  53.  2. 0 / 53.  2. 0
  libavfilter   2. 27. 5 /  2. 27. 5
  libswscale    2.  0. 0 /  2.  0. 0
  libpostproc  51.  2. 0 / 51.  2. 0
[wmv3 @ 029F0060] Extra data: 8 bits left, value: 0

Seems stream 1 codec frame rate differs from container frame rate: 1000.00 (1000
/1) -> 29.97 (30000/1001)
Input #0, asf, from 'MASTERs.wmv':
  Metadata:
    WMFSDKVersion   : 12.0.7601.17514
    WMFSDKNeeded    : 0.0.0.0000
    IsVBR           : 0
```

Running the Demonstration

The demonstration will display a menu containing files on the micro-SD card, as shown in Figure 6. To play a video, select it by touching the name, and it should start playing, as shown in Figure 7. The menu will return when the video has completed playback.

It should be noted that video file to be played in Method 2 has to be in the .RGB format as shown in Figure 6.

FIGURE 6:

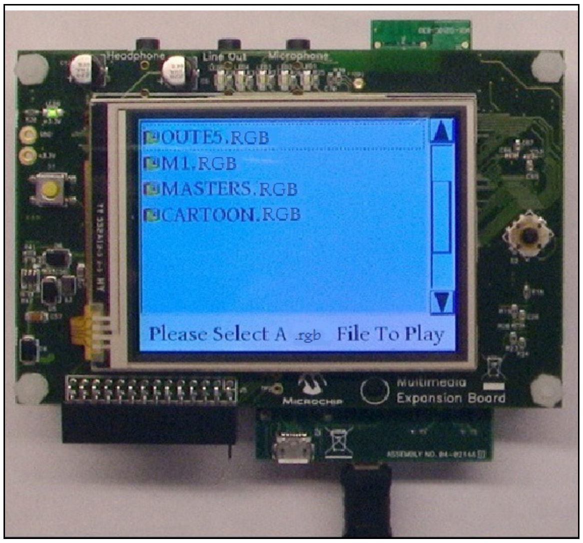
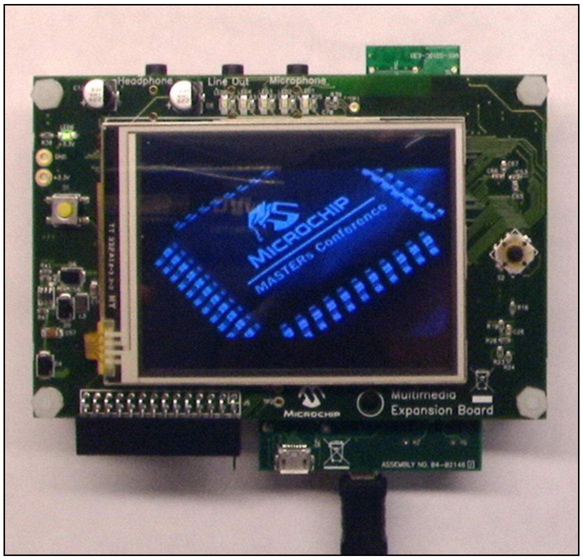


FIGURE 7:

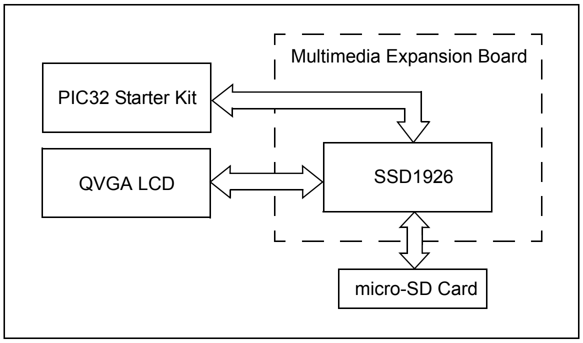


Implementation

This demonstration takes advantage of the Solomon Systech SSD1926 LCD controller to stream raw video data directly from a file on the micro-SD card to the video frame buffer in the SSD1926 on-chip memory. By streaming data directly from the card to the display buffer, the additional overhead of moving data through the microcontroller is avoided.

As shown in Figure 8, the PIC MCU interfaces to the SSD1926 via the MEB. The software utilizes the Microchip Graphics Library to create a GUI for the demonstration. The Microchip MDD file system library is used to locate the file and provide file size and location information. The SSD1926 SD driver from the Microchip Applications Library is used to access the micro-SD card.

FIGURE 8:



Operation

This demonstration is based on the SSD1926 JPEG and micro-SD Card demonstration in the Microchip Applications Library. The GUI is based on the Microchip Graphics Library and is implemented with the Microchip Graphics Display Designer. The Microchip MDD library is used to access the file system on the micro-SD card.

During video playback, data is transferred directly from the micro-SD card to the frame buffer in the SSD1926 one frame at a time using Direct Memory Access (DMA). When a frame has completed transfer, the file pointer is set to the next frame, and the process is repeated until the file has finished playing.

To maximize data throughput, the SD bit clock is running at the maximum speed supported by the standard, 25 MHz. Four-bit data transfer, multi-block transfer and auto-command 12 are enabled. Refer to the SSD1926 data sheet, which can be requested from Solomon Systech for details.

METHOD 3: STREAMING UNCOMPRESSED AVI VIDEO OVER ETHERNET ON THE MEB USING PIC32

This method describes the implementation of AVI video streaming over Ethernet using the MEB and the PIC32 Ethernet Starter Kit.

Video File Requirement

- Uncompressed AVI file in Device Independent Bitmap (DIB) format
- Video resolution of 320 x 240
- Video frame rate of up to 20 frames per second (use a video file with a lower frame rate if network traffic is high)

Hardware Requirement

- Multimedia Expansion Board (DM320005)
- PIC32 Ethernet Starter Kit (DM320004)
- Ethernet (RJ-45) crossover cable
- Standard-A to mini-B USB cable

Software Requirement

- MPLAB 8.x or later

Packet Structure

This demonstration uses a custom-developed PC application (`UDPApp.exe`) to stream the video information over the Ethernet.

The video packets are delivered over the Ethernet using UDP as the transport layer protocol. The PC application delivers three types of packets to the PIC32 device. These are the AVI Information Packet, Video Packet, and Time Sync Packet.

AVI Information Packet: The data field in this packet contains information about the resolution of the video, the number of video frames, and the duration of video frame in microseconds. The PC application extracts this information from the AVI file and delivers them to the PIC32 device. This information is used by the PIC32 device to control the video playback rate and the screen resolution. The AVI information packet is the first packet to be sent from the PC application to the PIC32 device. The structure of this packet is shown in Figure 9.

FIGURE 9: AVI INFORMATION PACKET STRUCTURE

Bytes →	2	2	4	2	2	4	4
	Packet Length	Packet Type	Pixel Address	Size X	Size Y	Number of Frames	Frame Duration
				← Data Field			→

Where:

Packet Length = 20

Packet Type = 4 (AVI Information Packet)

Pixel Address = *not used for this packet*

Size X = 320 (width of screen in pixels)

Size Y = 240 (height of the screen in pixels)

Number of Frames = number of frames supported by the AVI file

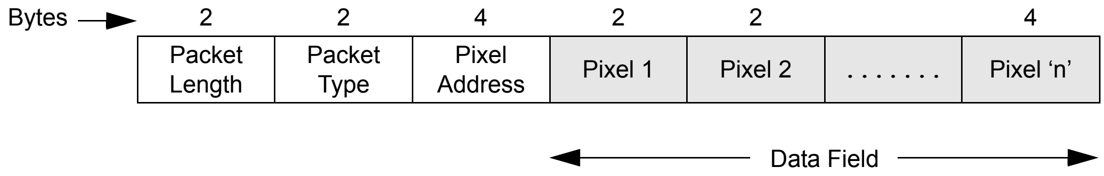
Frame Duration = duration of the video frame in microseconds

Video Packet: The PC application delivers the raw video data to the PIC32 device using video packets. The data field in the video packet contains the pixel data in RGB565 (16-bit) format. Before transmitting the video packets to the PIC32 device, the PC application parses the AVI file and converts the pixel data from RGB888 (24-bit) format to RGB565 (16-bit) format. Converting the pixel data from 24-bit format to 16-bit format saves a considerable amount of Ethernet bandwidth. The video packet also contains the pixel address, which helps the PIC32 device in plotting the pixel data on the LCD. The structure of the video packet is shown in Figure 10.

Time Sync Packet: The PC application transmits the video packets in bursts. Each burst contains the video information for an entire video frame. At the end of each burst, the PC application sends out a time sync packet and waits for an acknowledgement from the PIC32 device. The next burst of video packets from the PC application is transmitted soon after the reception of acknowledgement from the PIC32 device. Insertion of the time sync packet between the bursts of video packets helps in controlling the video playback speed.

Figure 11 shows the structure of the time sync packet. Figure 12 shows the video frame rate control mechanism using the time sync packet.

FIGURE 10: VIDEO PACKET STRUCTURE



Where:

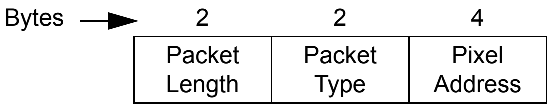
Packet Length = 8 + 2 x (number of pixels)

Packet Type = 1 (Video Packet)

Pixel Address = address of the first pixel in the data field

Pixel 1 . . . . Pixel ‘n’ = Pixels in the data field

FIGURE 11: TIME SYNC PACKET STRUCTURE



Where:

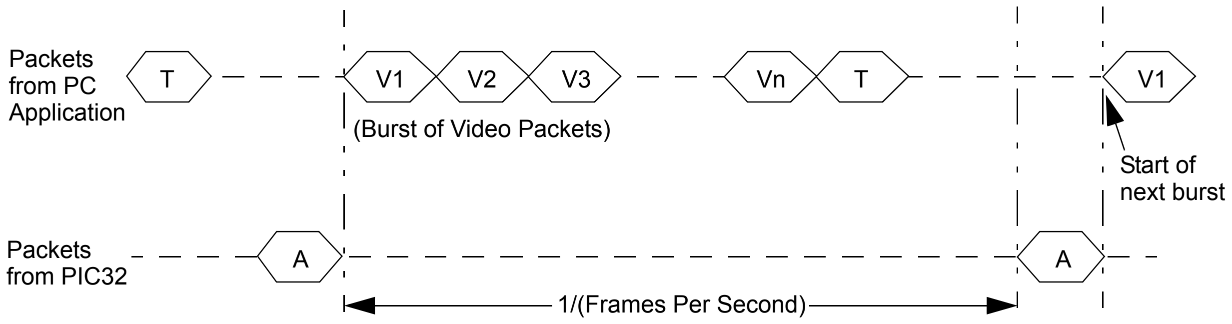
Packet Length = 8

Packet Type = 3 (Time Sync Packet)

Pixel Address = *not used for this packet*

Note: There is no Data Field for this packet type.

FIGURE 12: CONTROLLING VIDEO FRAME RATE USING TIME SYNC PACKET



Where:

T = Time Sync Packet

A = Acknowledgement Packet

V1 . . . Vn = Video Packets

The PC application expects an acknowledgement packet from the PIC32 device for the AVI information packet and the time sync packet. Figure 13 shows the structure of this acknowledgement packet.

FIGURE 13: ACKNOWLEDGEMENT PACKET STRUCTURE

Bytes →	2	2	4	1	1	1
	Packet Length	Packet Type	Pixel Address	‘A’	‘C’	‘K’

← Data Field →

Where:

Packet Length = 11

Packet Type = 5 (Acknowledgement Packet)

Pixel Address = *not used for this packet*

Data Field contains the string “ACK”.

Operation

This demonstration utilizes the TCP/IP stack and the Graphics stack from the Microchip Application Library. Soon after board initialization, the PIC32 device starts to listen for new packets on the Ethernet. The PC application sends out an AVI information packet as the first packet to the PIC32 device. The AVI information packet contains the frame duration information in microseconds. The frame duration information is used by the PIC32 device to play the video at the correct speed. The video packets follows the AVI information packet over the Ethernet. The video packet contains pixel addresses and pixel data. The PIC32 device uses DMA transfer to write pixel data to the SSD1926 (graphics controller) via the PMP interface.

Running the Demonstration

1. Set the IP address and subnet mask of the server PC to 192.168.1.12 and 255.255.255.0, respectively.
2. Connect the PIC32 Ethernet Starter Kit to the MEB.
3. Connect the PC to the starter kit's debugger using the mini-B USB cable.
4. Connect the PIC32 Ethernet Starter Kit to the PC using the Ethernet (RJ-45) cable.
5. Start MPLAB IDE and open the `video_stream_demo.mcp` project.
6. Select **PIC32 Ethernet Starter Kit** as the debugger in MPLAB IDE.
7. Rebuild the project in Debug mode.
8. Program the starter kit and run the demonstration.
9. Ping the PIC32 device to make sure the Ethernet connection is working. This can be done by typing `ping 192.168.1.11` in a command shell on the PC. The default IP address of the PIC32 device is: 192.168.1.11.
10. In a Windows command shell, change to the directory where the PC application, `UDPApp.exe`, is installed.
11. Run the PC application by typing:

    ```
    UDPApp [filename] [delay] 192.168.1.11
    ```

    Where,

    `[filename]` is the video file in AVI format, and

    `[delay]` is the delay between UDP packets in microseconds. Ideally, set this value between 100 and 200. Increasing the delay value plays the video at slower rate. Decreasing the delay value leads to loss of packets and may deteriorate the video quality.

METHOD 4: STREAMING UNCOMPRESSED AVI VIDEO OVER ETHERNET ON WVGA USING PIC32

This method is a variation of Method 3, which uses a different technique to display the video on a larger resolution display, such as a WVGA.

Video File Requirements

- Uncompressed AVI video file in DIB format
- Video resolution of 400 x 240
- Video frame rate of up to 20 FPS (use a video file with a lower frame rate if network traffic is high)

Hardware Requirements

1. Graphics PICtail™ Plus Epson S1D13517 Board (AC164127-7)
2. PIC32 Ethernet Starter Kit (DM320004)
3. Ethernet (RJ-45) cross-over cable
4. Standard-A to mini-B USB cable
5. Graphics Display Truly 7" 800 x 480 Board (AC164127-9)

Software Requirement

- MPLAB 8.x or later

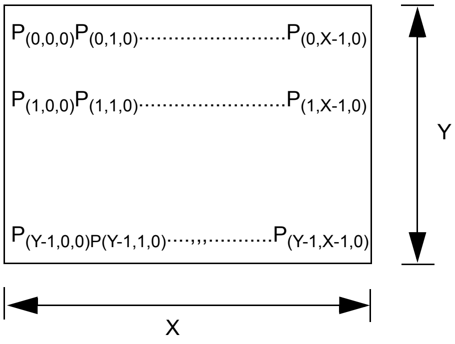
Operation and Running the Demonstration

To operate and run the demonstration, use the information described in Method 3: Streaming Uncompressed AVI Video over Ethernet on the MEB Using PIC32, with the exception of the display.

In Method 4, the technique of horizontal interpolation of pixels is adopted. The video on the graphics display is obtained by interlacing alternate fields (set of all odd and even lines in a frame) from alternate frames. The frame rate from the interpolation can be increased by this method.

Figure 14 describes the interlaced video format for the SD card.

FIGURE 14: SD CARD FORMAT



Where:

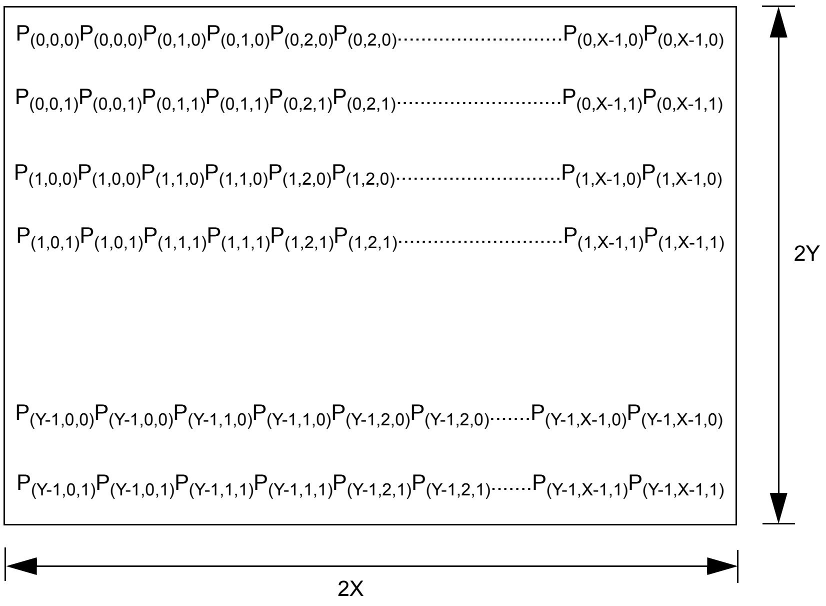
P(a,b,c) = the pixel in the original video stored on the SD card, where (a,b,c) = the pixel indices

a = the vertical index; b = the horizontal index; c = the frame number

X and Y = the horizontal and vertical resolution of the video

Figure 15 describes the interlaced video format for the graphics display.

FIGURE 15: GRAPHICS DISPLAY FORMAT



Where:
P(a,b,c) = the pixel on the graphic display, where (a,b,c) = the pixel indices
a = the vertical index; b = the horizontal index; c = the frame number
2X and 2Y = the horizontal and vertical resolution of the graphics display

CONCLUSION

This application note provides four methods that can be used for video playback and video streaming applications using a PIC32 device.

These methods can be used in situations such as distance education or surveillance cameras, as well as news and entertainment videos for display on the Internet. In addition, video playback from a secure digital card can find uses in situations where data needs to be stored for future review, such as video from surveillance cameras or educational lectures.

APPENDIX A: SOURCE CODE

All of the software covered in this application note is available as a single WinZip archive file. This archive can be downloaded from the Microchip corporate Web site at:

www.microchip.com

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE**.** Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2009

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC^{32} logo, rfPIC and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, REAL ICE, rfLAB, Select Mode, Total Endurance, TSHARC, UniWinDriver, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2011, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

♻ Printed on recycled paper.

ISBN: 978-1-61341-787-4

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hangzhou
Tel: 86-571-2819-3187
Fax: 86-571-2819-3189

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-330-9305

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

08/02/11