ECEE 5623
RT Embedded Systems

Lecture – Continuous Media Projects and Added Feature Overview

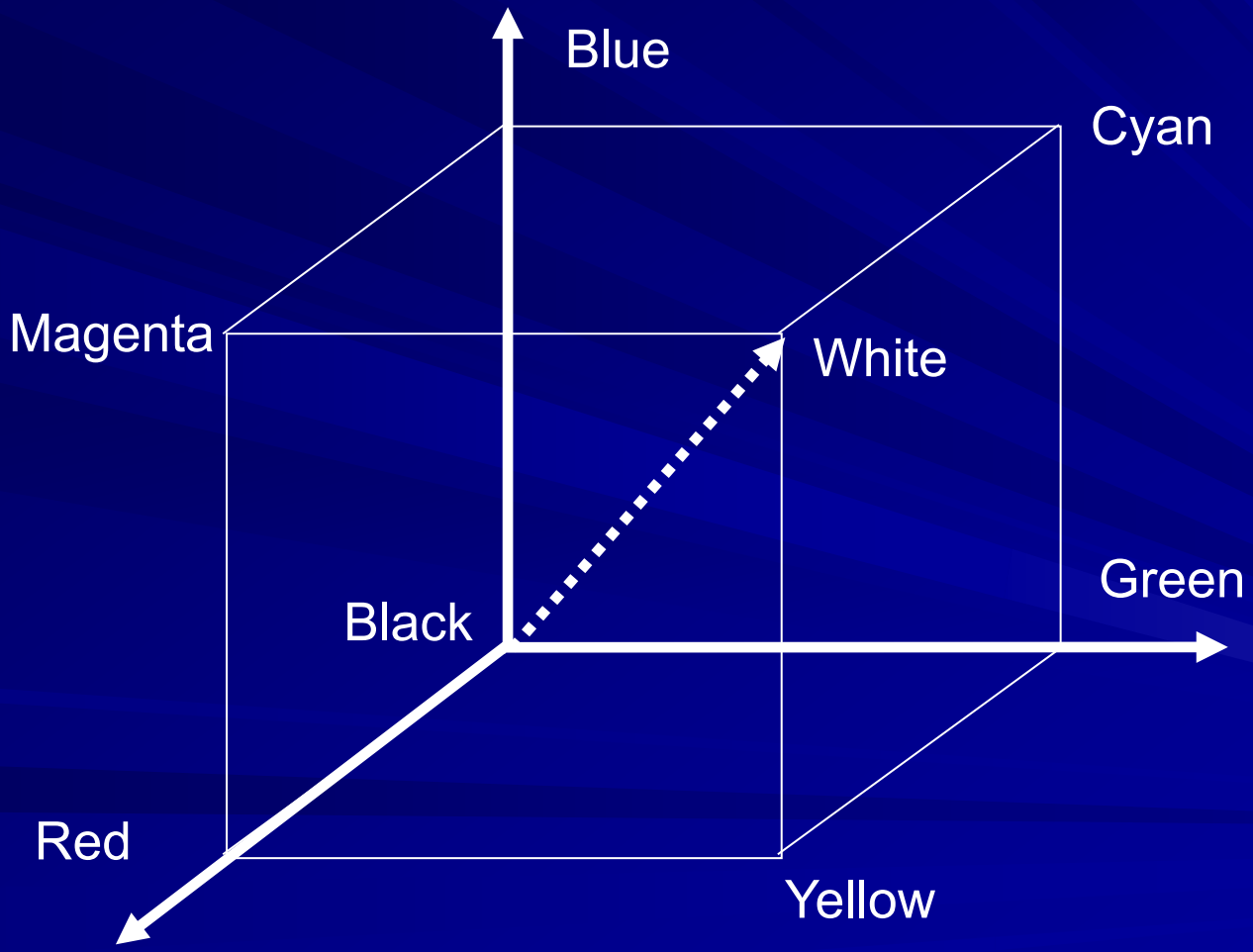
Video Media

Embedding Video Codecs

- Codec = Compression, Decompression
- Build Your Own
  - Run Length Encoding
  - Difference Images
  - Python Viewer (Displays PPM sequences)
  - X-Windows Viewer (Displays PPM sequences)
- Theora/Ogg Open Source Option
  - http://www.theora.org/
  - Stream over Raw TCP to VLC Viewer
- MJPEG Open Source Option
  - http://mjpeg.sourceforge.net/

Notes on Computer Color Encoding

- RGB, 24-bit, [0-255] for each color band
- Each Pixel is a 3-D Vector in RGB Space



YUV/YCrCb ⇔ RGB

- An Alternative to RGB is YUV, Where Y is Luminance and CrCb is Chrominance

- The following 2 sets of formulae are taken from information from Keith Jack's excellent book "Video Demystified" (ISBN 1-878707-09-4).

- **RGB to YUV Conversion (For Computers with RGB [0-255])**
  - Y = (0.257 * R) + (0.504 * G) + (0.098 * B) + 16
  - Cr = V = (0.439 * R) - (0.368 * G) - (0.071 * B) + 128
  - Cb = U = -(0.148 * R) - (0.291 * G) + (0.439 * B) + 128

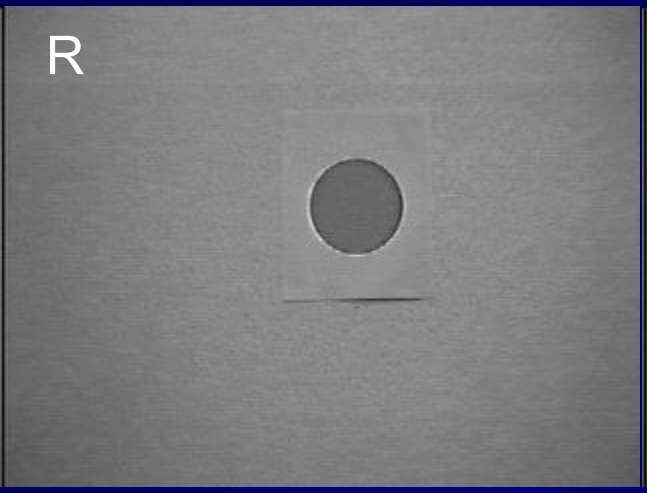
- **YUV to RGB Conversion**
  - B = 1.164(Y - 16) + 2.018(U - 128)
  - G = 1.164(Y - 16) - 0.813(V - 128) - 0.391(U - 128)
  - R = 1.164(Y - 16) + 1.596(V - 128)

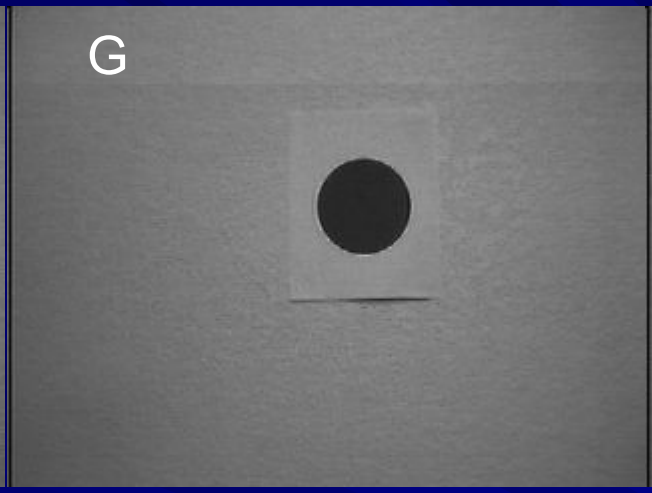
- In both these cases, you have to clamp the output values to keep them in the [0-255] range.

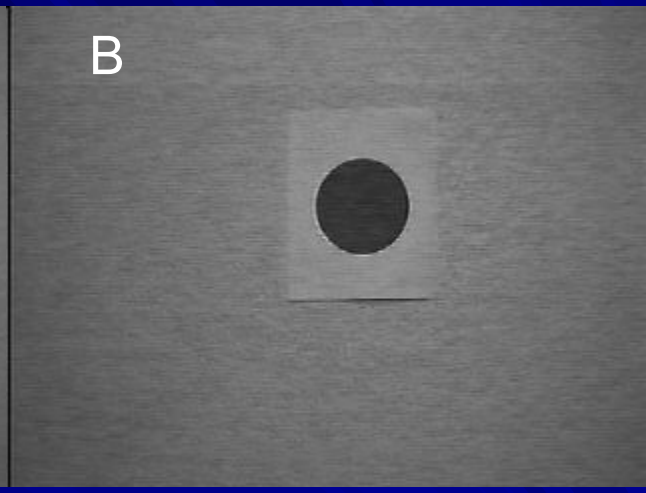
RGB to Grayscale

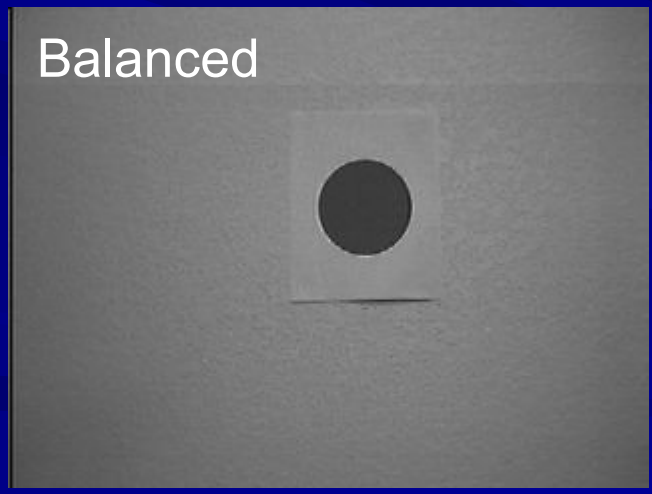
- From 24 bits to 8 bits most often
- Single Color Band from RGB (Y, Cr, or Cb only)
  - Not True Grayscale, but Useful for Computer Vision Applications
  - Some Targets Like a Laser Pointer are Best Seen in Red Band or Green Band Alone
- GIMP Uses a Conversion to 8-bit Luminance
  - $Y = 0.3R + 0.59G + 0.11B$
  - Defined by equal amounts of color the eye is most sensitive to green, then red, and then blue

R, G, or B band only vs. Balance







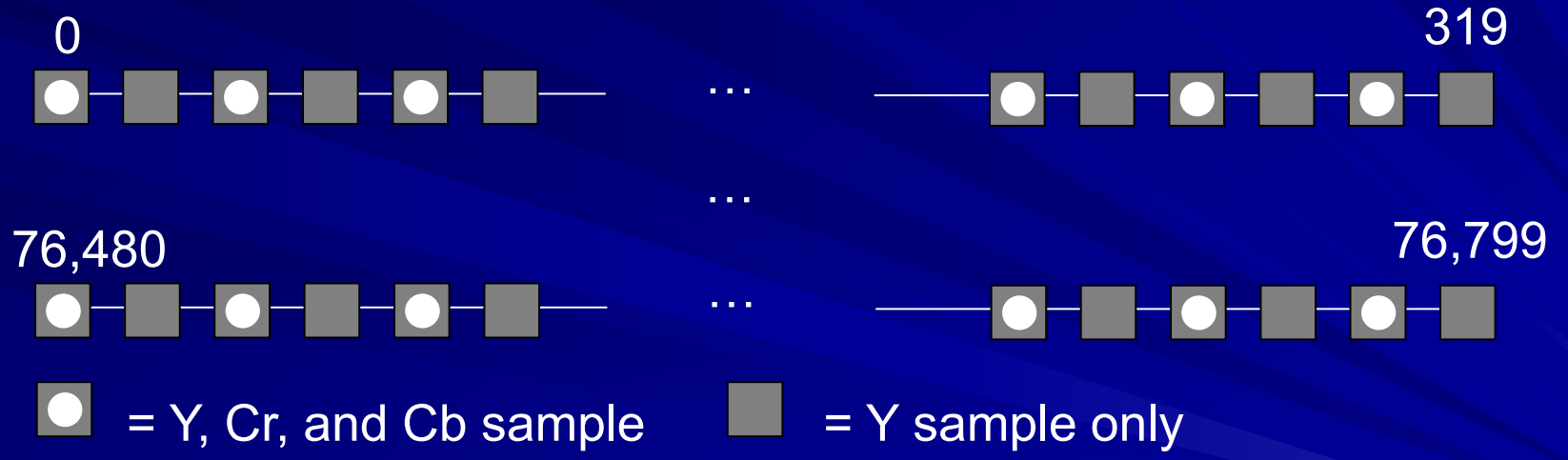


Building Your Own Video Codec

- Video Compression Spaces
  - Color Space
    - RGB (24 bits)
    - YCrCb (16 bits / pixel) – Lossy compared to RGB
    - Grayscale (8 bits) – Lossy
  - XY Dimension
    - As an Image Convolution/Deconvolution (Lossy)
      - Convolution: Moving Average of Pixels to Compress Multiple Pixels to One
      - Deconvolution: Interpolation to Estimate Original Pixel Values Adjancent to Compressed Pixel
    - As A String
      - Run Length Encoding (Lossless)
      - Huffman Encoding (Lossless)
  - Frame to Frame Time Dimension
    - Difference Images (Lossless or Lossy with Thresholds)
      - Pixel Address and data for non-zero ∆pixels
        - Pixel Address for 320x240 = 17 bits
        - Dpixel = 24 bits for RGB
      - Scenes often don't change quickly
      - Transmission of Change-Only Data
      - Threshold on ∆pixel to Compress more (Lossy)
      - Detection of Size Blow-up on Fast Changing Data

YCrCb 4:2:2 16-bit Format

- For every 2 Y samples in a scanline, there is one CrCb sample
  - Each Y, Cr, and Cb Sample is 8 bits each
  - Two RGB Pixes = 48 bits, Whereas Two YCrCb is 32 bits, or 16 bits per pixel vs. 24 bits per pixel (1/3 smaller frame size)



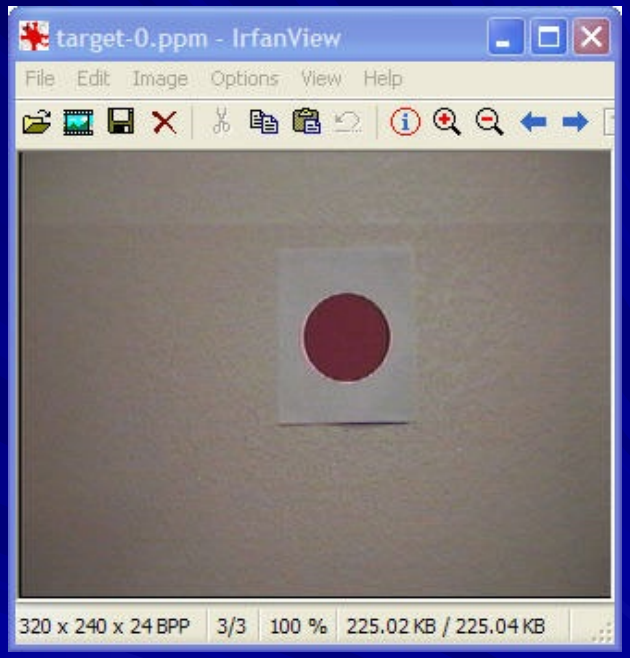
- Pixel-0 = Y7:$Y0_0$, Cb7:$Cb0_0$; Pixel-1 = Y7:$Y0_1$, Cr7:$Cr0_0$
- Pixel-2 = Y7:$Y0_2$, Cb7:$Cb0_1$; Pixel-3 = Y7:$Y0_3$, Cr7:$Cr0_1$
- Pixel-4 = Y7:$Y0_4$, Cb7:$Cb0_2$; Pixel-5 = Y7:$Y0_5$, Cr7:$Cr0_2$

Basic Definitions

- Useful Wikipedia Pages
  - PPM - http://en.wikipedia.org/wiki/Portable_pixmap
  - GIF - http://en.wikipedia.org/wiki/GIF
  - JPEG - http://en.wikipedia.org/wiki/JPEG
  - MPEG - http://en.wikipedia.org/wiki/MPEG
  - Theora - http://en.wikipedia.org/wiki/Theora
- PPM and PGM Info
  - http://netpbm.sourceforge.net/doc/ppm.html (RGB)
  - http://netpbm.sourceforge.net/doc/pgm.html (grayscale)
- MPEG Info
  - http://www.mpeg.org/MPEG/index.html
  - http://www.compression-links.info/MPEG
- DivX Info
  - http://www.divx.com/divx/

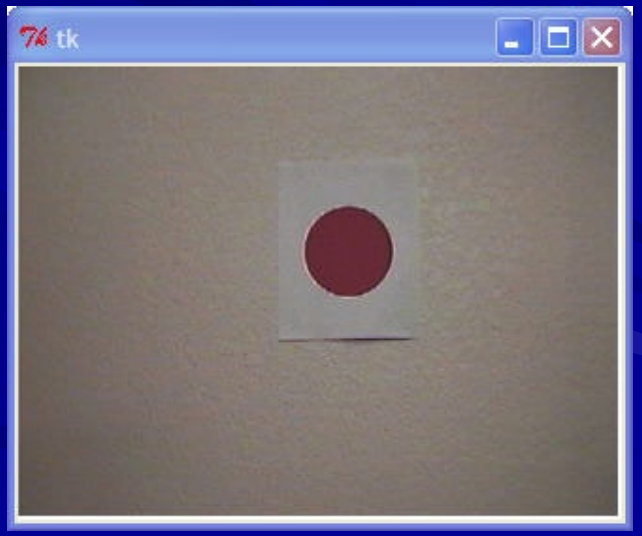
Video Driver and Frame Analysis Resources

- Test Dumping Frame over TSFS
  - Slow, but sure
  - Can load dumped frame to analyze
- Single Frame Viewing and Analysis
  - http://www.irfanview.com/
  - http://www.trilon.com/xv/downloads.html
  - http://www.gimp.org/downloads/
- Image Processing Libraries
  - http://cimg.sourceforge.net/
  - http://sourceforge.net/projects/opencvlibrary/



Using Python PPM Stream Viewer

- Test Your Python and Vpipe Installation Using
  - Run vpipe_display.py first
  - Run frametx_test.py second
- Set up and test your Btvid Bt878 driver and hardware
- Test Image capture using “report” and write_save_buffer to dump PPM image over TSFS
- Write TCP/IP client code to stream 1 frame/sec to Vpipe Display



More on Streaming

- Streaming = Codec + Data Transport
  - E.g. MPEG-4 / RTP
  - Your Codec / UDP

- Transport Protocols
  - UDP – Connectionless Datagrams, No Delivery Guarantee
    - Diversely Routed Data Can Out of Order
    - Datagrams Lost Are Not Re-transmitted
  - TCP – Connection-oriented Messaging, Guarantee for Window
    - All Messages Segmented, Sequenced, and Fully Acknowledged
    - All Messages Re-assembled from Segments and Re-Ordered
    - Any Lost Messages Re-transmitted from Re-Transmission Window
      - Re-transmission Window Based on Bandwidth-Delay, Congestion
      - After a Maximum Number of Retries, TCP Finally Gives Up
  - RTP/UDP – Real-Time Transport
    - Payload type, Sequence Number, Time-stamp, Delivery Monitoring
    - http://www.ietf.org/rfc/rfc1889.txt
  - RTSP – Real-Time Streaming Transport
    - Typically Used to Control RTP Delivery, but can use UDP or other transport
    - http://www.ietf.org/rfc/rfc2326.txt

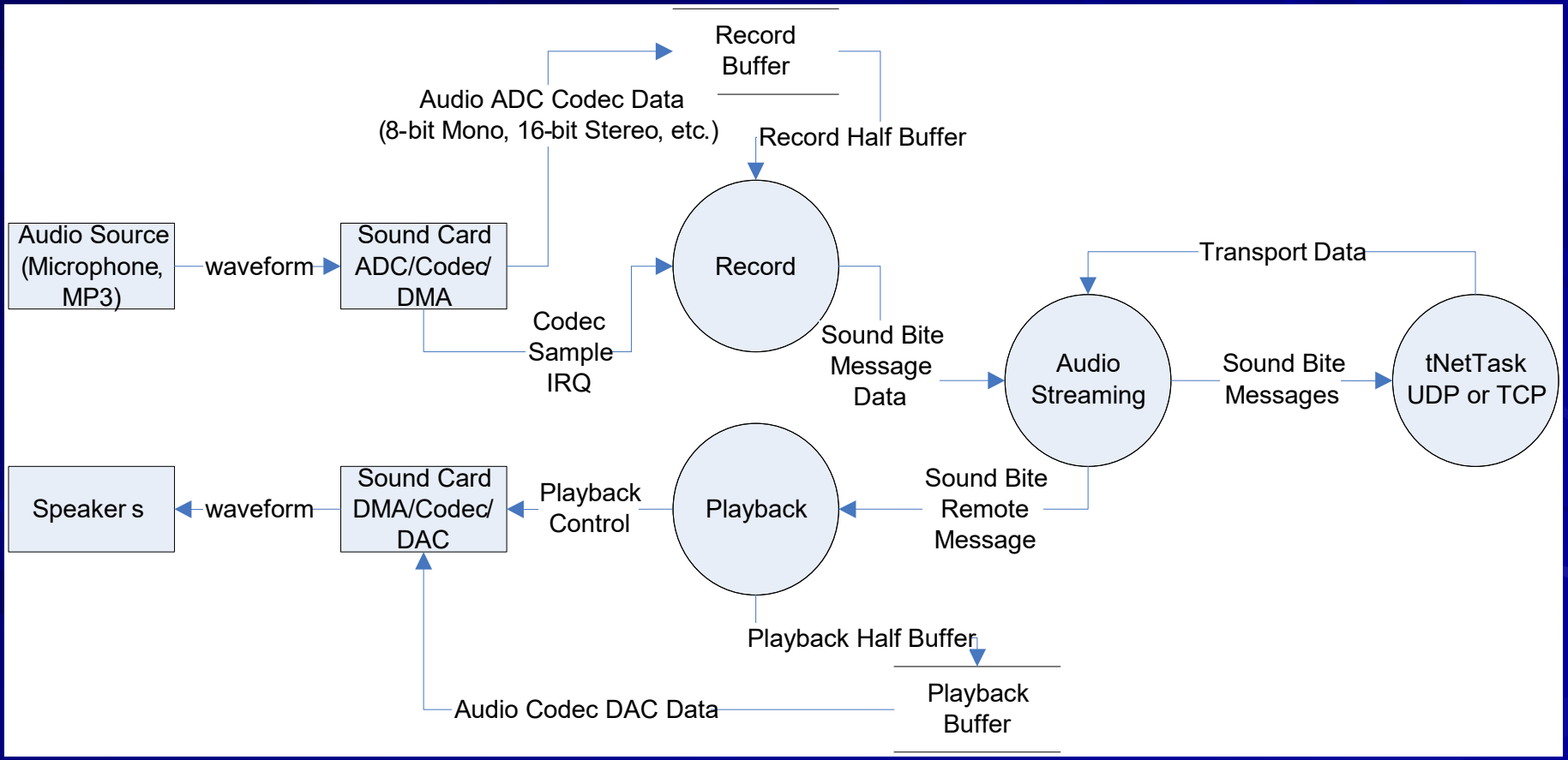
Project Suggestions

- Motion Detection Video Stream Storage and Playback
  - Motion Detection Threshold for Difference Images
  - Compress on Store and Uncompress on Retrieval for Display
- Computer Vision Projects
- Video Editing
- Digital Video Recorder

Audio Media

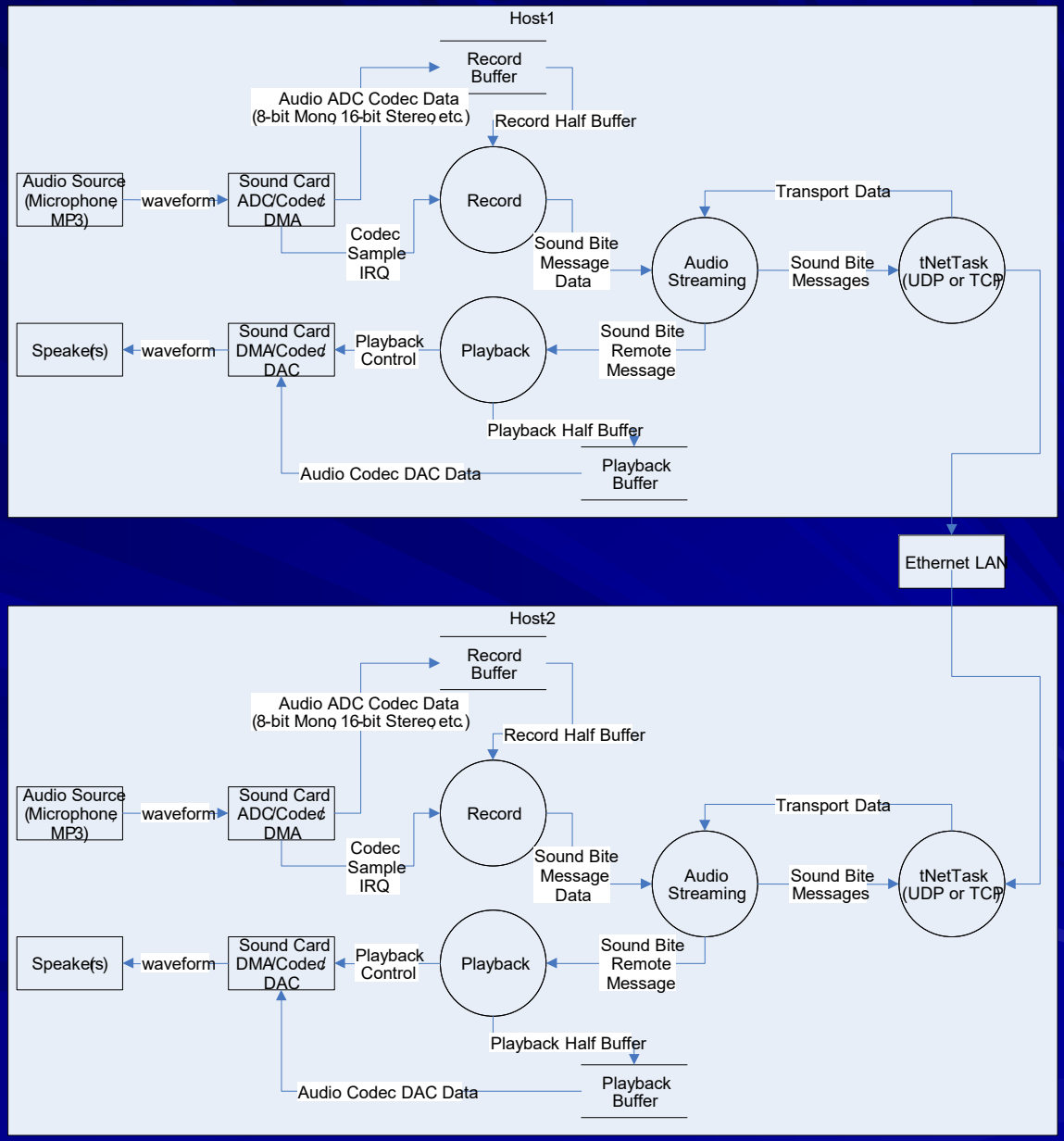
Basic Voice/IP Setup

- Four Tasks (Services) On Each Phone Host
  - Record, Play-back
  - Streaming, Transport
- Two Phone Hosts for Point-to-Point Phone Call



Point-to-Point VOIP

- Replicate Services on Each Host
- Establish Control and Data Transport
- Either Side Should Be Able to Initiate or Answer a Call



VOIP Debugging Tips

- Use “d” windshell command to dump memory from Record and Playback buffers
  - Check for changing data
  - Use Continuous Music Source (e.g. MP3 player) to Test Record
- Use Windview to Verify Codec/DMA Interrupt
- Compare VxWorks Driver to Linux ALSA Driver
  - Chipset documentation errors or difficult to interpret registers
  - Example of audio formats
- Verify Transport Independently from VOIP
- Adjust Half Buffer Size to Improve Quality

More Advanced Audio and VoIP Features

- Teleconferencing Multiple Callers
- Combine with Video for Video Conferencing
- Encryption of the Audio Stream for a Secure Phone
- Voice mail
- Speech to Text Translation
- Text to Speech