EE405 Electronic Design Lab – RoboCam

Lab 5. WebCam and System Integration

Byung Kook Kim
School of Electrical Engineering
Korea Advanced Institute of Science and Technology

I. Purpose

- The purpose of this lab is to
  - design and implement a video functionality using off-the-shelf WebCam on Beaglebone,
  - and perform system integration of RoboCam.

2. Problem Statement

- **Problem 5. WebCam and System integration**
  - Implement ***Video functionality.***
  - Perform ***system integration*** of Video and control functionalities.

Step-by-step improvements.

- **Problem 5A. Test Capture WebCam on Beaglebone.**
- Test capture image from WebCam with V4L2 (Video for Linux 2) on Beaglebone.
- **Problem 5B. Learn SDL2 via Tutorials.**
  - Learn SDL 2 (Simple DirectMedia Layer 2) via Tutorials.
- **Problem 5C. Implement Video functionality.**

  Implement Video functionality composed of
  - Camera on Beaglebone (Capture from WebCam and send video to network) and
  - Viewer on PC (Receive video from network and display video to user) using SDL2.
- **Problem 5D…**

Problem Statement (II)

- **Problem 5D. System Integration.**
  - Perform system integration of Video and control functionalities:
  - A. ***Video functionality*** composed of
    - Camera on Beaglebone (Capture from WebCam and send video to network) and
    - Viewer on PC (Receive video from network and display video to user).
  - B. ***Control functionality*** composed of
    - Commander on PC (Get key input from user and send command packet to network) and
    - Controller on Beaglebone (Receive command packet from network and actuate servos and lights on the robot).
  - For ***recording photos and videos***, Commander on PC sends user command to Viewer, which records photos and videos to storage device.

III. Technical Backgrounds
A. Test WebCam on Bone

- **1. Choosing a Webcam device driver in Linux**
  - Webcam support in Linux is mainly provided by the Linux UVC (USB Video Class) Project's ***UVC driver**.*
  - This aims to provide a universal driver in the same way that a generic driver handles USB storage devices (memory sticks, hard drives, etc.).
  - However, other drivers also exist that may allow more devices to be used.

  - When looking to purchase a webcam for use with Ubuntu, you should look for a *UVC compatible camera*.
  - The Linux-UVC project has a good list of UVC compatible webcams as well as The Quickcam Team for Logitech cameras.

[“Webcam”, https://help.ubuntu.com/community/Webcam**]**

Test WebCam on Bone (II)
2. UVC compatible cameras

- Welcome to the USB Video Class Linux device driver home.
- The goal of this project is to provide all necessary software components to fully support UVC compliant devices in Linux. This include a V4L2 kernel device driver and patches for user-space tools.
- The UVC specification covers webcams, digital camcorders, analog video converters, analog and digital television tuners, and still-image cameras that support video streaming for both video input and output.
- Supported devices

Device ID	Name	Manufacturer	Status
046d:0994	Logitech Quickcam Orbit/Sphere AF	Logitech	V
046d:0805	Logitech Webcam C300	Logitech	V
046d:0819	Logitech Webcam C210	Logitech	V

- Can't find Webcam C110, but we proceed?! // dmesg!

["Linux UVC (USB Video Class) driver and tools", http://www.ideasonboard.org/uvc/]

Video for Linux 2

4. Video for Linux

[“Beaglebone Images, Video and OpenCV”, http://derekmolloy.ie/beaglebone-images-video-and-opencv/]

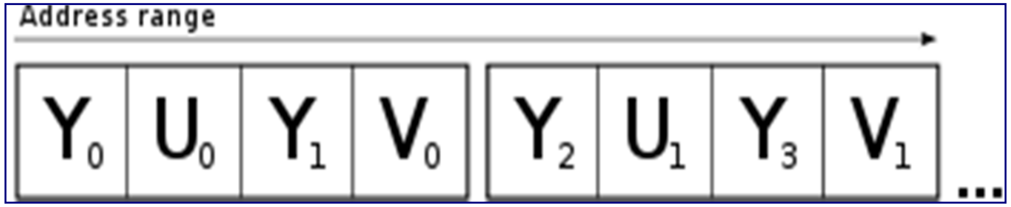
- Beaglebone does not have graphic user interface (GUI). Hence we cannot use GUI programs such as Cheese.
- Using v4l2 (Video for Linux version 2), you can capture images.
  - Video4Linux or V4L is a video capture application programming interface for Linux, supporting many USB webcams, TV tuners, and other devices.
  - Video4Linux is closely integrated with the Linux kernel.
- V4L2 Usage
  - Getting help
    - $ v4l2-ctl --help

Video for Linux 2 (II)

- List supported video formats
  - **# v4l2-ctl --list-formats**
  - ioctl: VIDIOC_ENUM_FMT
  - Index : 0
  - Type : Video Capture
  - Pixel Format: 'YUYV'
  - Name : YUV 4:2:2 (YUYV)
  -
  - Index : 1
  - Type : Video Capture
  - Pixel Format: 'MJPG' (compressed)
  - Name : MJPEG
- To YUYV: (Note: Two '-' before set-fmt-video)
  - **# v4l2-ctl --set-fmt-video=width=640,height=480,pixelformat=0**
- To MJPG:
  - **# v4l2-ctl --set-fmt-video=width=640,height=480,pixelformat=1**

5. What is YUYV?

- ["**YUYV Format**", http://linuxtv.org/downloads/v4l-dvb-apis/V4L2-PIX-FMT-YUYV.html]
- **Name**
  - V4L2_PIX_FMT_YUYV — Packed format with ½ horizontal chroma resolution, also known as YUV 4:2:2
- **Description**
  - In this format each four bytes is two pixels.
  - Each four bytes is two Y's, a Cb and a Cr.
    - Y: Intensity
    - Cb, Cr: Chroma. Color.



Address range

Y_0 U_0 Y_1 V_0 Y_2 U_1 Y_3 V_1 ...

6. What is JPEG?

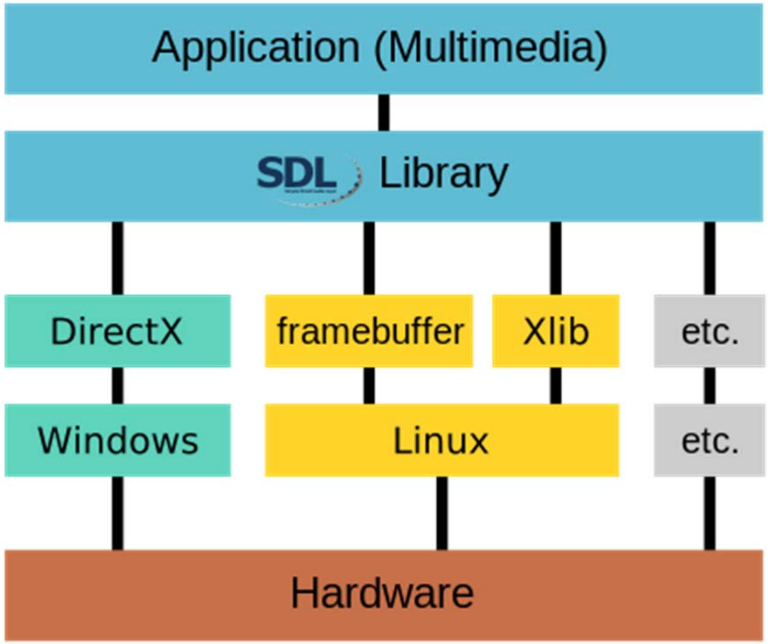
- Joint Photographic Experts Group
  - Lossy compression for digital images, particularly for those images produced by digital photography.
  - The degree of compression can be adjusted, allowing a selectable tradeoff between storage size and image quality.
  - JPEG typically achieves 10:1 compression with little perceptible loss in image quality.
  - Check_Jpeg() function
    - SOI 0xffd8
    - EOI 0xffd9

- **Motion JPEG** (**M-JPEG** or **MJPEG**)
  - A video compression format in which each video frame or interlaced field of a digital video sequence is compressed separately as a JPEG image.

Capture.c

- ***Capture.c***
  - This is a generic example program to capture from WebCam using V4L2.
  - Use this program to capture images with some modifications: Capture2.c

B. Learn SDL via Tutorials



- **Simple DirectMedia Layer (SDL)** is
  - a cross-platform software development library designed to provide a low level hardware abstraction layer to computer multimedia hardware components.
  - Software developers can use it to write high-performance computer games and other multimedia applications that can run on many operating systems such as Android, iOS, Linux, Mac OS X, Windows and other platforms.
  - SDL manages video, audio, input devices, CD-ROM, threads, shared object loading, networking and timers.[5] For 3D graphics it can handle an OpenGL or Direct3D context.

SDL Image

- SDL extension libraries allow you do things like load image files besides BMP, render TTF fonts, and play music. You can set up SDL_image to load PNG files.

- JPEG image can be loaded.
  - rwop = SDL_RWFromFile(filename, “rb”)
  - IMG_LoadJPG_RW(rwop)

- ***SDL Event Handling***

V. Design

Pre-report for first week

- 1. Search internet for C110 specifications. You can visit www.logitech.com.
- 2. Design SDL program for Key Values (Problem 5B)
  - A. Design Get_Key_Var_SDL.cpp
  - *B. Design Key_Value_SDL.cpp*

Design (II)

Pre-report for second week

- **3. Design Video functionality (Problem 5C).**

Video functionality is composed of

  - Camera on Beaglebone (Capture from WebCam and send video to network) and
  - Viewer on PC (Receive video from network and display video to user) using SDL2.

- ***Data flow in detail***

User	**PC**	**Network**	**Beaglebone**	**Robot**
SW:	**Viewer**		**Camera**	
			Capture images ←	← WebCam
			←Send Jpeg images	
		←UDP packets		
	Recv Jpeg images ←			
See video	←Display video SDL			

Design (III) 5C

- ***Camera.c on Bone***
  - Use X.c for cross-gcc compatibility!
  - Modify Capture2.c to add UDP send functionality
  - Additional file: Send_UDP.c for UDP-related routines.
  - *Jpeg format: Use 320x240 Jpeg images!*
- ***Video stream packet***
  - Sequence of Jpeg images (Mjpg video from WebCam)
  - Variable size.
- ***Viewer.cpp using SDL2 on PC***
  - Main loop used for polling events.
  - Require a thread to listen from socket and display.
  - Additional file: Recv_UDP.cpp for UDP-related routines.

Design (IV) 5C

- ***Algorithm for Camera.c on Bone***

```
0. Get argument of Capture: CPORT (4960) to send
          Including -p for ports & -a for ip_addr.
1. Init UDP packet
5. Loop
          Capture Webcam to Jpeg image
          Sendto Jpeg image to UDP datagram
9. Close UDP packet
```

Design (V) 5C

- **Algorithm for Viewer.c on PC (using SDL2)**

```
  Main
1. init()             // Init SDL2
2. Fill the surface with light grey & update the surface
3. Init UDP port with any IP and CPORT (4960) to listen
4. Run a thread RecvDispThread to listen to datagram and display Jpeg image
5. Key event loop
          Just print input key value.

  RecvDispThread
Loop
          A. recvfrom() socket datagram
             print the number of received bytes
          B. Check if Jpeg image (Header & Trailer)
          C. Display Jpeg image using SDL2
```

Design: 4. System Integration (Problem 5D)

- ***Allocate multi-tasking with thread***

HW	PC	Beaglebone
SW: tasks	***Viewer: Thread*** Receive video from network and display video to user (SDL2). Viewer, which records photos and videos to storage device.	***Camera: Task*** Capture from WebCam and send video to network.
	Commander: Task Get key input from user and send command packet to network (SDL2). For recording photos and videos, Commander on PC sends user command to Viewer.	***Controller: Task*** Receive command packet from network and actuate servos and lights on the robot.

Design: 4. System Integration Dataflow with multi-task in detail

User	PC	Bone	Robot
			← See with WebCam
		Camera task	
		Capture WebCam ←	
		← Ssend img datagrams	
	Viewer thread		
	Recv img datagrams ←		
	← Display with SDL2.		
	Record photo/video.		
Watches display ←			
Issue key commands →			
(Servos, Lights, Photo, and Video)			
	Commander task		
	Get user key command in raw mode with SDL2		
	Classify internal/external command		
	Handle internal command (Photo & Video) to Viewer ^		
	Send external command to cmd datagram →		
		Control task	
		→ Recv cmd datagram	
		Control Servos and lights →	
			Move and illuminate

VI. Lab Procedures

First week

- A. Test Image Capture with WebCam on Beagleone
- B. Learn SDL via Tutorial

Second week

- C. Test video functionality
  - Camera on Bone
  - Viewer on PC using SDL
- D. Test System Integration
  - Video functionality
    - Camera on Bone
    - Viewer on PC using SDL
  - Control functionality
    - Commander on PC
    - Controller on Beaglebone