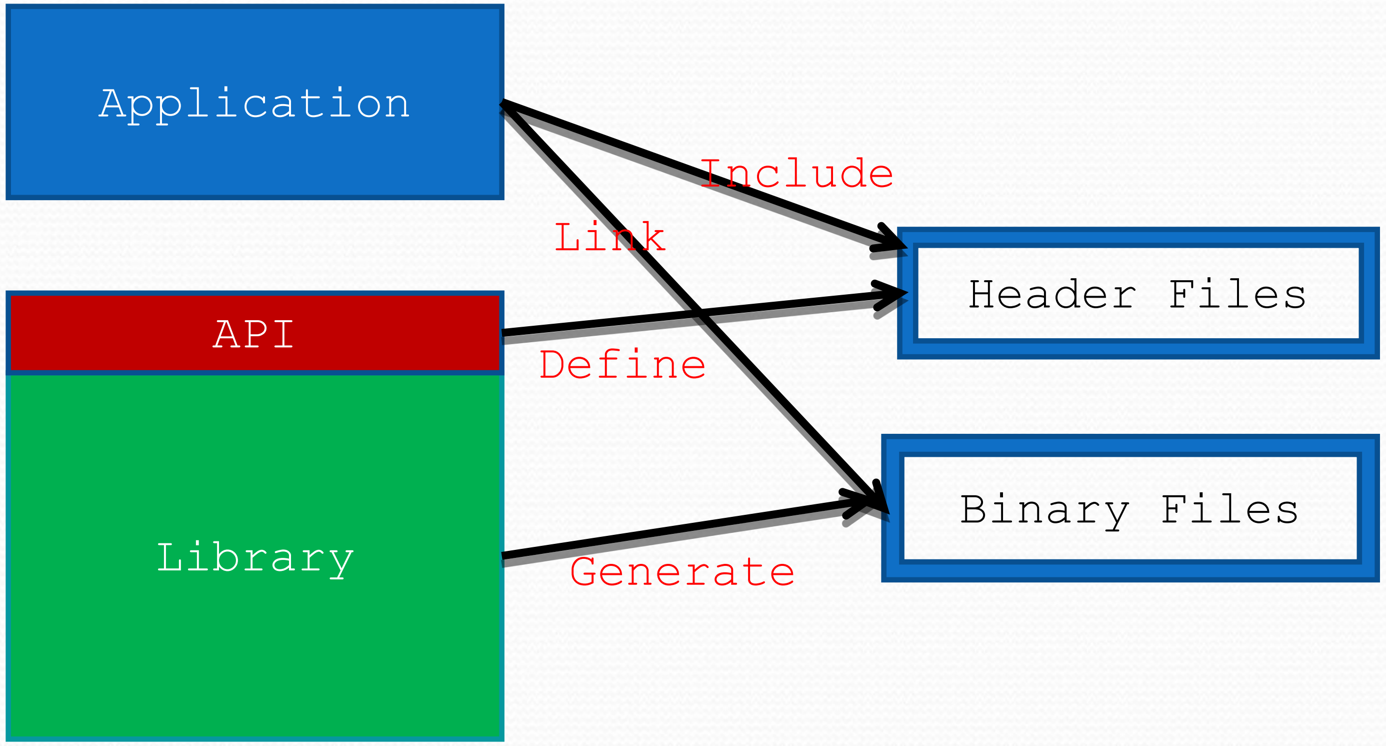
Embedded System & Embedded Linux Development Part 6

Index of today's topic

What is library?

- a collection of resources used to develop software
- contain code and data that provide services to independent programs
- Code reuse

Application/Library/API



C Library

- Standard Library
  - Philosophies between C & Python
- GNU C Library (glibc)
- Math Library (libm)

C standard library

- <assert.h>
- <ctype.h>
- <errno.h>
- <float.h>
- <limits.h>
- <locale.h>
- <math.h>
- <setjmp.h>
- <signal.h>
- <stdarg.h>
- <stddef.h>
- <stdio.h>
- <stdlib.h>
- <string.h>
- <time.h>

C POSIX library header files

<cpio.h>	Magic numbers for the cpio archive format.
<dirent.h>	Allows the opening and listing of directories.
<fcntl.h>	File opening, locking and other operations.
<grp.h>	User group information and control.
<pthread.h>	Defines an API for creating and manipulating POSIX threads.
<pwd.h>	passwd (user information) access and control.
<sys/ipc.h>	Inter-process communication (IPC).
<sys/msg.h>	POSIX message queues.
<sys/sem.h>	POSIX semaphores.
<sys/stat.h>	File information (stat et al.).
<sys/time.h>	Time and date functions and structures.
<sys/types.h>	Various data types used elsewhere.
<sys/utsname.h>	uname and related structures.
<sys/wait.h>	Status of terminated child processes (see wait)
<tar.h>	Magic numbers for the tar archive format.
<termios.h>	Allows terminal I/O interfaces.
<unistd.h>	Various essential POSIX functions and constants.
<utime.h>	inode access and modification times.

Library

- Static Libraries
  - *.a
- Shared Libraries
  - *.so
- Dynamic Linking
  - Dll/so (Shared Object)

Link Library in Linux

- Include the header files in .c
  - **#include <library_file_header.h>**
- Compile
  - **gcc -c <object file> <source file> -I<library_header_path>**
- Link Shared Library
  - **gcc -o <output> <object files> -l<library_name> -L<library_file_path>**
- Link Static Library
  - **gcc -o <output> <object files> <library_file_name.a>**

Index of today's topic

- Use libraries in development
- Use libjpeg to display/save jpeg image
- Use camera on FL6410
- Exercise

libjpeg

- a library to decode & encode the jpeg file
- installed in most of the Linux distribution
- Official website:
    - http://sourceforge.net/projects/libjpeg/

Use libjpeg in your code

- include the jpeglib.h
  - **#include <jpeglib.h>**
- add **-ljpeg** to the linking options
  - **gcc -o test.o -c test.c**
  - **gcc -o test.out test.o -ljpeg**

Example:

- http://www.cim.mcgill.ca/~junaed/libjpeg.php

Key point in writing a JPEG file

- Convert RGB565 buffer to RGB888 buffer
- Write the jpeg file line by line

```
while (cinfo.next_scanline <
  cinfo.image_height) {
   row_pointer[0] =
     &image_buffer[cinfo.next_scanline *
     image_width * 3];
      (void) jpeg_write_scanlines(&cinfo,
  row_pointer, 1);
   }
```

Key point in displaying a JPEG file

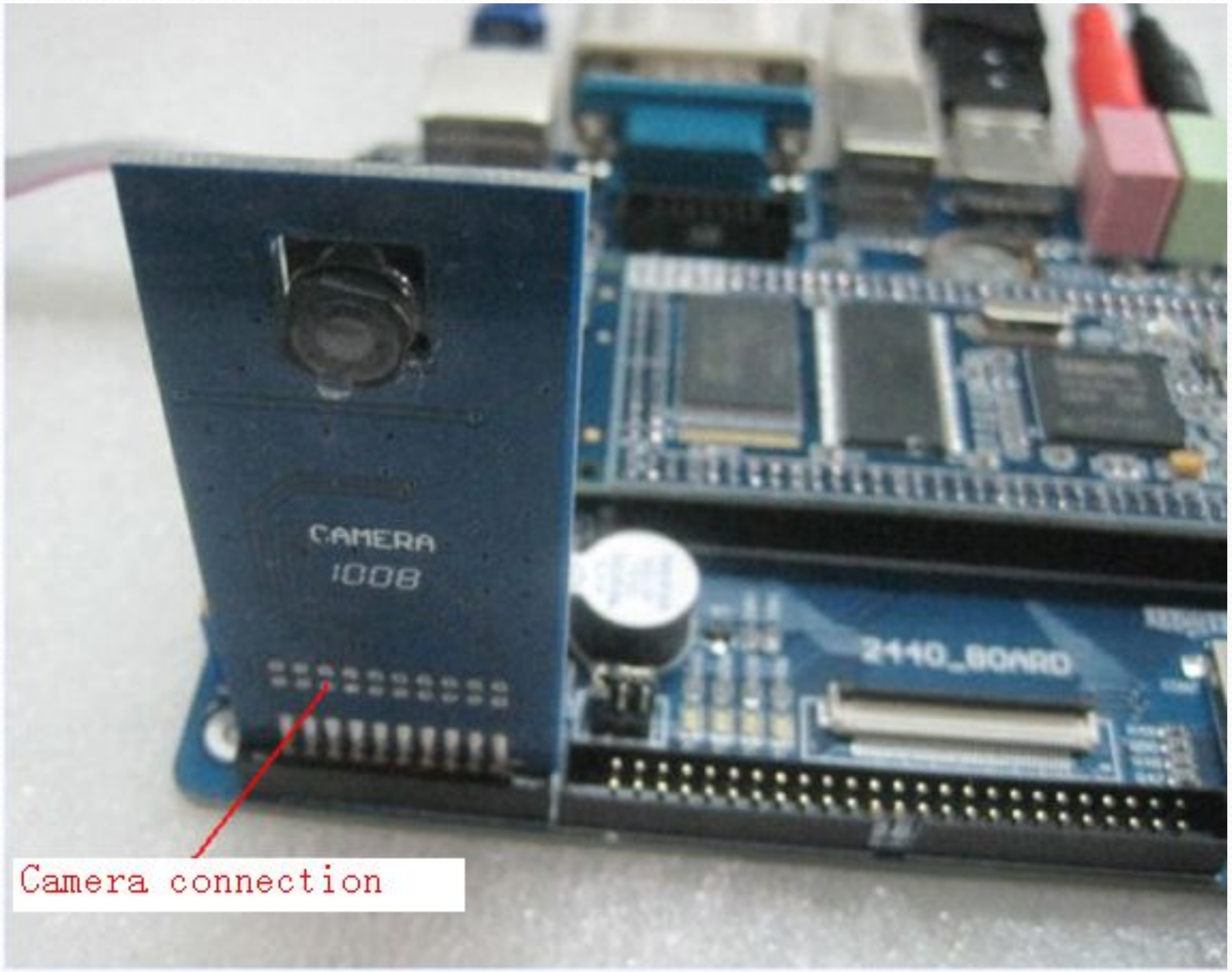
- Read the jpeg file line by line
- Convert RGB888 pixel to RGB565 pixel
- Draw pixel by pixel

```
y = 0;
while (cinfo.output_scanline < cinfo.output_height)
{
	jpeg_read_scanlines(&cinfo, buffer, 1);
	for(x=0; x<cinfo.output_width||
		x<SCREEN_WIDTH; x++)
	{
		draw_pixel(x, y, COLOR(buffer[0][x*3],
			buffer[0][x*3+1], buffer[0][x*3+2]));
	}
	y++;
}
```

Index of today's topic

- Use libraries in development
- Use libjpeg to display/save jpeg image
- Use camera on FL6410
- Exercise

Camera on FL6410



The use of the camera: camera outward, frame angle alignment, then energized.

Shut down power before plug/unplug camera!

Camera driver in Linux

- device name **/dev/video0**
- read() on this device will get the screen pixels with resolution=320*240 in RGB565 format

```
read(camerafd, cambuf, VIDEO_WIDTH * VIDEO_HEIGHT * VIDEO_DEPTH / 8)
```

- you need a buffer with size=320*240*2

Index of today’s topic

- Use libraries in development
- Use libjpeg to display/save jpeg image
- Use camera on FL6410
- Exercise

Exercise tips

Read JPEG

Decompression procedure

1 allocates space for the object and
 initialize JPEG
2 Specify the decompressed data source
3 Getting file information
4 To decompress setting parameters,
 including image size, color space
5 Start decompression
6 Remove the data
7 decompression is completed
8 release resources

```
/* LCD pixel : 480x272 */
/* Camera    : 320x240 */
#define VIDEO_WIDTH 320
#define VIDEO_HEIGHT240
#define VIDEO_DEPTH 16
```

Write JPEG

Compression procedure

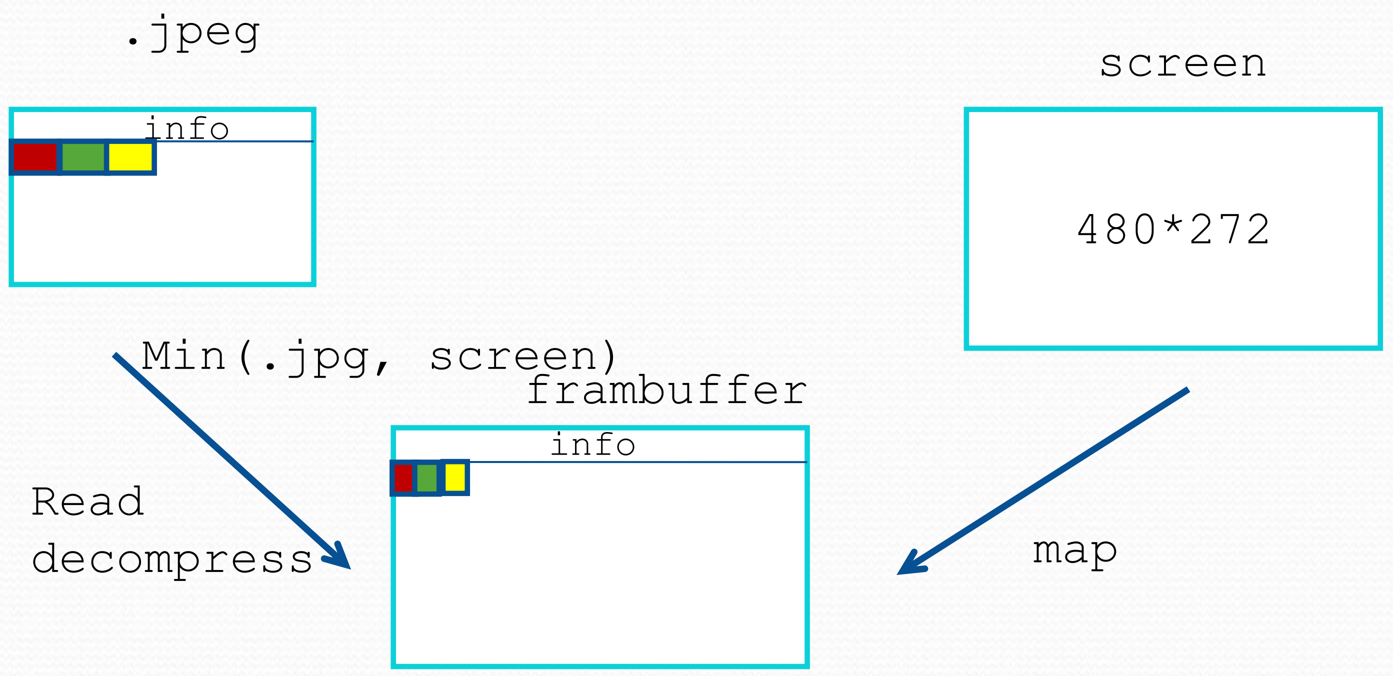
1 allocates space for the object and initialize JPEG
2 Specify the image output target
3 the compression set parameters,
 including the image size, color space
4 compression begins
5 write data
6 compression is completed
7 release resources

```
JSAMPROW row_pointer[1];
/* libjpeg data structure
for storing one row, that
is, scanline of an image
*/
```

Exercise 1

- Display a JPEG image file in the LCD screen

Processing sketch



Exercise 2

- display camera video in the screen
  - open Framebuffer device
  - get screen size, do mmap
  - open camera device
  - while(1)
    - read camera data to buffer
    - copy data to framebuffer
  - you can see what you get in LCD

Tips

- struct v4l2_format fmt;//Frame format, such //as width, height
- int type = V4L2_BUF_TYPE_VIDEO_CAPTURE;

- fmt.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
- fmt.fmt.pix.width = VIDEO_WIDTH;
- fmt.fmt.pix.height = VIDEO_HEIGHT;
- fmt.fmt.pix.depth = VIDEO_DEPTH;
- fmt.fmt.pix.pixelformat= V4L2_PIX_FMT_RGB565;

Tips

/*open camera */

- camerafd = open("/dev/video0", O_RDWR);

/*Set video capture formats*/

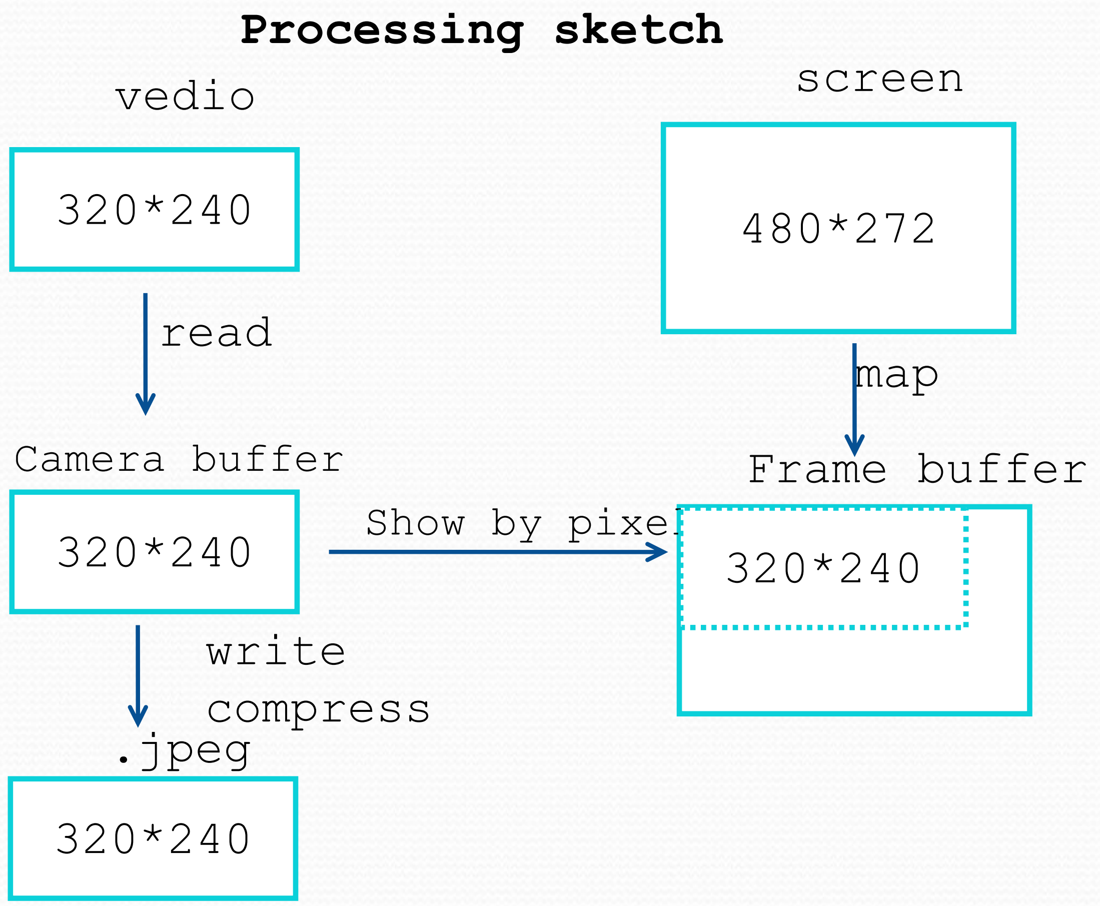
- ioctl(camerafd, VIDIOC_S_FMT, &fmt)

/*Start a video display functions*/

- ioctl(camerafd, VIDIOC_STREAMON, &type)

Exercise 3

- Display Video on Screen
- When user press a button, save current picture to a JPEG file



Processing sketch
vedio
320*240
read
Camera buffer
320*240
write
compress
.jpeg
320*240
Show by pixel
screen
480*272
map
Frame buffer
320*240