EE405A Electronic Design Lab 5 – Robocam

20170275 박정웅 20170616 정희진

1. Purpose

- process images on the beaglebone using webcam.

- capture images & videos by webcam via wifi and display the images on PC.

- integrate the video functionality and system integration.

- multi-thread should be implemented so that TMR can be controlled.

2. Experiment Procedure

1) Problem 5A. Test Capture webCam on Beaglebone

- Check for WebCam device and UVC device driver

먼저 웹캠이 연결되어 있는지 커맨드를 쳐 Logitech C270 웹캠이 연결된 것을 확인하였다.

```
root@beaglebone:~# lsusb
Bus 001 Device 002: ID 1a40:0101 Terminus Technology Inc. 4-Port HUB
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 003: ID 0bda:8176 Realtek Semiconductor Corp. RTL8188CUS 802.11n WLAN Adapter
Bus 001 Device 004: ID 046d:0825 Logitech, Inc. Webcam C270
```

```
root@beaglebone:~# dmesg  | grep 046d:0825
[   18.536570] uvcvideo: Found UVC 1.00 device <unnamed> (046d:0825)
[   18.627280] input: UVC Camera (046d:0825) as /devices/ocp.3/47400000.usb/musb-hdrc.1.auto/usb1/1-1/1-1.4/1-1.4:1.0/input/input1
```

UVC device driver을 확인하였다.

```
root@beaglebone:~# lsmod | grep uvc
uvcvideo               57013  0
videobuf2_vmalloc       2490  1 uvcvideo
```

/dev에 video0이 있는지 확인하였다.

```
root@beaglebone:~# ls /dev | grep video*
video0
```

- Install Video4Linux2 and Test

다음으로 Video4Linux를 설치하였다.

```
root@beaglebone:~# sudo apt-get install v4l-utils
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  v4l-utils
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 111 kB of archives.
After this operation, 258 kB of additional disk space will be used.
Get:1 http://archive.debian.org/debian/ wheezy/main v4l-utils armhf 0.8.8-3 [111 kB]
Fetched 111 kB in 1s (80.6 kB/s)
Selecting previously unselected package v4l-utils.
(Reading database ... 60509 files and directories currently installed.)
Unpacking v4l-utils (from .../v4l-utils_0.8.8-3_armhf.deb) ...
Setting up v4l-utils (0.8.8-3) ...
```

```
root@beaglebone:~# sudo apt-get install libv4l-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  libv4l-dev
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 7,914 B of archives.
After this operation, 57.3 kB of additional disk space will be used.
Get:1 http://archive.debian.org/debian/ wheezy/main libv4l-dev armhf 0.8.8-3 [7,914 B]
Fetched 7,914 B in 0s (13.9 kB/s)
Selecting previously unselected package libv4l-dev:armhf.
(Reading database ... 60517 files and directories currently installed.)
Unpacking libv4l-dev:armhf (from .../libv4l-dev_0.8.8-3_armhf.deb) ...
Setting up libv4l-dev:armhf (0.8.8-3) ...
```

Device를 리스트하고, /dev/video0을 확인하였다.

```
root@beaglebone:~# v4l2-ctl --list-devices
UVC Camera (046d:0825) (usb-musb-hdrc.1.auto-1.4):
        /dev/video0
```

포맷을 리스트하여 YUYV 포맷과 MJPG 포맷을 확인하였다.

```
root@beaglebone:~# 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
```

다음으로 드라이버 정보를 얻었다.

```
root@beaglebone:~# v4l2-ctl --get-priority
Priority: 2
```

```
root@beaglebone:~# v4l2-ctl -D
Driver Info (not using libv4l2):
        Driver name   : uvcvideo
        Card type     : UVC Camera (046d:0825)
        Bus info      : usb-musb-hdrc.1.auto-1.4
        Driver version: 3.8.13
        Capabilities  : 0x84000001
                Video Capture
                Streaming
```

컨트롤 정보를 얻었다.

```
root@beaglebone:~# v4l2-ctl -L
                     brightness (int)    : min=0 max=255 step=1 default=128 value=128
                       contrast (int)    : min=0 max=255 step=1 default=32 value=32
                     saturation (int)    : min=0 max=255 step=1 default=32 value=32
 white_balance_temperature_auto (bool)   : default=1 value=1
                           gain (int)    : min=0 max=255 step=1 default=64 value=64
          power_line_frequency (menu)    : min=0 max=2 default=2 value=2
                                0: Disabled
                                1: 50 Hz
                                2: 60 Hz
      white_balance_temperature (int)    : min=0 max=10000 step=10 default=4000 value=4000 flags=inactive
                      sharpness (int)    : min=0 max=255 step=1 default=24 value=24
         backlight_compensation (int)    : min=0 max=1 step=1 default=0 value=0
                 exposure_auto (menu)    : min=0 max=3 default=3 value=3
                                1: Manual Mode
                                3: Aperture Priority Mode
             exposure_absolute (int)    : min=1 max=10000 step=1 default=166 value=166 flags=inactive
        exposure_auto_priority (bool)   : default=0 value=1
```

카메라에 관한 모든 정보를 리스트하였다.

```
v4l2-ctl -all
v4l2-ctl: invalid option -- 'a'
Unknown argument `-all'
Usage:
Common options:
  --all               display all information available
  -C, --get-ctrl=<ctrl>[,<ctrl>...]
                      get the value of the controls [VIDIOC_G_EXT_CTRLS]
  -c, --set-ctrl=<ctrl>=<val>[,<ctrl>=<val>...]
                      set the controls to the values specified [VIDIOC_S_EXT_CTRLS]
  -D, --info          show driver info [VIDIOC_QUERYCAP]
  -d, --device=<dev> use device <dev> instead of /dev/video0
                      if <dev> is a single digit, then /dev/video<dev> is used
  -F, --get-freq      query the frequency [VIDIOC_G_FREQUENCY]
  -f, --set-freq=<freq>
                      set the frequency to <freq> MHz [VIDIOC_S_FREQUENCY]
  -h, --help          display this help message
  -I, --get-input     query the video input [VIDIOC_G_INPUT]
  -i, --set-input=<num>
                      set the video input to <num> [VIDIOC_S_INPUT]
  -l, --list-ctrls    display all controls and their values [VIDIOC_QUERYCTRL]
  -L, --list-ctrls-menus
                      display all controls, their values and the menus [VIDIOC_QUERYMENU]
  -N, --list-outputs display video outputs [VIDIOC_ENUMOUTPUT]
  -n, --list-inputs  display video inputs [VIDIOC_ENUMINPUT]
  -O, --get-output    query the video output [VIDIOC_G_OUTPUT]
  -o, --set-output=<num>
                      set the video output to <num> [VIDIOC_S_OUTPUT]
  --list-standards    display supported video standards [VIDIOC_ENUMSTD]
  -S, --get-standard
                      query the video standard [VIDIOC_G_STD]
  -s, --set-standard=<num>
                      set the video standard to <num> [VIDIOC_S_STD]
                      <num> can be a numerical v4l2_std value, or it can be one of:
                      pal-X (X = B/G/H/N/Nc/I/D/K/M/60) or just 'pal' (V4L2_STD_PAL)
                      ntsc-X (X = M/J/K) or just 'ntsc' (V4L2_STD_NTSC)
                      secam-X (X = B/G/H/D/K/L/Lc) or just 'secam' (V4L2_STD_SECAM)
  --get-detected-standard
                      display detected input video standard [VIDIOC_QUERYSTD]
  -P, --get-parm      display video parameters [VIDIOC_G_PARM]
  -p, --set-parm=<fps>
                      set video framerate in <fps> [VIDIOC_S_PARM]
  -T, --get-tuner     query the tuner settings [VIDIOC_G_TUNER]
  -t, --set-tuner=<mode>
                      set the audio mode of the tuner [VIDIOC_S_TUNER]
                      Possible values: mono, stereo, lang2, lang1, bilingual
  --list-formats      display supported video formats [VIDIOC_ENUM_FMT]
  --list-formats-mplane
                      display supported video multi-planar formats [VIDIOC_ENUM_FMT]
  --list-formats-ext display supported video formats including frame sizes
                      and intervals
  --list-formats-ext-mplane
                      display supported video multi-planar formats including
                      frame sizes and intervals
```

```
                    Possible values: mono, stereo, lang2, lang1, bilingual
  --list-formats     display supported video formats [VIDIOC_ENUM_FMT]
  --list-formats-mplane
                     display supported video multi-planar formats [VIDIOC_ENUM_FMT]
  --list-formats-ext display supported video formats including frame sizes
                     and intervals
  --list-formats-ext-mplane
                     display supported video multi-planar formats including
                     frame sizes and intervals
  --list-framesizes=<f>
                     list supported framesizes for pixelformat <f>
                     [VIDIOC_ENUM_FRAMESIZES]
                     pixelformat is the fourcc value as a string
  --list-frameintervals=width=<w>,height=<h>,pixelformat=<f>
                     list supported frame intervals for pixelformat <f> and
                     the given width and height [VIDIOC_ENUM_FRAMEINTERVALS]
                     pixelformat is the fourcc value as a string
  -V, --get-fmt-video
                     query the video capture format [VIDIOC_G_FMT]
  -v, --set-fmt-video=width=<w>,height=<h>,pixelformat=<f>
                     set the video capture format [VIDIOC_S_FMT]
                     pixelformat is either the format index as reported by
                     --list-formats, or the fourcc value as a string
  -w, --wrapper      use the libv4l2 wrapper library.
  --list-devices     list all v4l devices
  --silent           only set the result code, do not print any messages
  --verbose          turn on verbose ioctl status reporting

Uncommon options:
  --try-fmt-video=width=<w>,height=<h>,pixelformat=<f>
                     try the video capture format [VIDIOC_TRY_FMT]
                     pixelformat is either the format index as reported by
                     --list-formats, or the fourcc value as a string
  --get-fmt-video-mplane
                     query the video capture format through the multi-planar API [VIDIOC_G_FMT]
  --set-fmt-video-mplane
  --try-fmt-video-mplane=width=<w>,height=<h>,pixelformat=<f>
                     set/try the video capture format using the multi-planar API [VIDIOC_S/TRY_FMT]
                     pixelformat is either the format index as reported by
                     --list-formats-mplane, or the fourcc value as a string
  --list-formats-out display supported video output formats [VIDIOC_ENUM_FMT]
  --get-fmt-video-out
                     query the video output format [VIDIOC_G_FMT]
  --set-fmt-video-out
  --try-fmt-video-out=width=<w>,height=<h>,pixelformat=<f>
                     set/try the video output format [VIDIOC_TRY_FMT]
                     pixelformat is either the format index as reported by
                     --list-formats-out, or the fourcc value as a string
  --list-formats-out-mplane
                     display supported video output multi-planar formats [VIDIOC_ENUM_FMT]
  --get-fmt-video-out-mplane
                     query the video output format using the multi-planar API [VIDIOC_G_FMT]
  --set-fmt-video-out-mplane
  --try-fmt-video-out-mplane=width=<w>,height=<h>,pixelformat=<f>
                     set/try the video output format with the multi-planar API [VIDIOC_S/TRY_FMT]
                     pixelformat is either the format index as reported by
```

- Control with v4l2

먼저 brightness를 128에서 200으로 바꾸어주었다.

```
root@beaglebone:~# v4l2-ctl -L | grep brightness
                    brightness (int)    : min=0 max=255 step=1 default=128 value=128
root@beaglebone:~# v4l2-ctl --set-ctrl brightness=200
root@beaglebone:~# v4l2-ctl -L | grep brightness
                    brightness (int)    : min=0 max=255 step=1 default=128 value=200
```

다음으로 pixelformal으로 YUYV에서 MJPG로 바꾸어주었다.

```
root@beaglebone:~# v4l2-ctl -V
Format Video Capture:
        Width/Height  : 640/480
        Pixel Format  : 'YUYV'
        Field         : None
        Bytes per Line: 1280
        Size Image    : 614400
        Colorspace    : SRGB
root@beaglebone:~# v4l2-ctl --set-fmt-video=width=640,height=480,pixelformat=1
root@beaglebone:~# v4l2-ctl -V
Format Video Capture:
        Width/Height  : 640/480
        Pixel Format  : 'MJPG'
        Field         : None
        Bytes per Line: 0
        Size Image    : 341333
        Colorspace    : SRGB
```

- Edit and cross-compile capture.c

Capture2로 컴파일하기 전에 KLMS에서 다운로드 받은 Lab5_code/a_CaptureBone/capture.c 파일에서 다음과 같이 코드를 바꾸었다. loopIsInfinite 변수를 넣어주고, if 조건문을 추가하고, white 조건문의 조건을 바꾸어주었다.

```
static void mainloop(void)
{
        unsigned int count;
        unsigned int loopIsInfinite = 0;

        if(frame_count == 0) loopIsInfinite = 1;

        count = frame_count;

        //while (count-- > 0) {
        while((count-- > 0) || loopIsInfinite){
                for (;;) {
                        fd_set fds;
                        struct timeval tv;
                        int r:
```

다음으로 capture.c를 Caputure2.c로 이름을 바꾸고 cross-compile하여 Capture2 실행파일을 얻었다.

```
root@beaglebone:~/Lab5_code/a_CaptureBone# ls
Capture2  Capture2.c  Makefile
```

Capture2를 실행하여 help 메시지를 보았다.

```
root@beaglebone:~/Lab5_code/a_CaptureBone# ./Capture2 --help
Usage: ./Capture2 [options]\n\nVersion 1.3\nOptions:\n-d | --device name   Video device name [/dev/video0]\n-h | --help         Print this messagen
-m | --mmap         Use memory mapped buffers [default]\n-r | --read         Use read() calls\n-u | --userp        Use application allocated buffe
rs\n-o | --output       Outputs stream to stdout\n-f | --format       Force format to 640x480 YUYV\n-c | --count        Number of frames to grab [7
0]\nroot@beaglebone:~/Lab5_code/a_CaptureBone#
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7 | VT102 | Online 0:20 | ttyACM0
```

- Capture YUYV as root

다음으로 비디오 width를 640, height를 480, pixelformat을 YUYV로 바꾸고, frame을 1로 하여 cap640a.yuv 파일을 얻었다. 하지만 select timeout이 뜨고 파일을 확인한 결과 데이터가 없는 것을 확인하였다. 크기가 커서 제대로 아웃풋 파일이 안나온 것으로 확인된다.

```
ight=480,pixelformat=0_code/a_CaptureBone# v4l2-ctl --set-fmt-video=width=640,he
root@beaglebone:~/Lab5_code/a_CaptureBone# ./Capture2 -c 1 -o > cap640a.yuv
select timeout\nroot@beaglebone:~/Lab5_code/a_CaptureBone#
```

```
root@beaglebone:~/Lab5_code/a_CaptureBone# ls -la cap*
-rw-r--r-- 1 root root 0 Dec  3 02:49 cap640a.yuv
root@beaglebone:~/Lab5_code/a_CaptureBone#
```

이번에는 비디오 width를 320, height를 240, pixelformat을 YUYV로 바꾸고, frame을 1로 하여

cap320a.yuv 파일을 얻었다. 데이터 크기를 확인한 결과 아웃풋이 잘 나온 것으로 보인다.

```
ight=240,pixelformat=0_code/a_CaptureBone# v4l2-ctl --set-fmt-video=width=320,he
root@beaglebone:~/Lab5_code/a_CaptureBone# ./Capture2 -c 1 -o > cap320a.yuv
.\nroot@beaglebone:~/Lab5_code/a_CaptureBone# ls -la cap*
-rw-r--r-- 1 root root 153600 Dec  3 02:50 cap320a.yuv
-rw-r--r-- 1 root root      0 Dec  3 02:49 cap640a.yuv
```

한번 더 실행해 파일(cap320b.yuv)을 하나 더 얻어보았다.

```
ight=240,pixelformat=0_code/a_CaptureBone# v4l2-ctl --set-fmt-video=width=320,he
root@beaglebone:~/Lab5_code/a_CaptureBone# ./Capture2 -c 1 -o > cap320b.yuv
.\nroot@beaglebone:~/Lab5_code/a_CaptureBone# ls -la cap*
-rw-r--r-- 1 root root 153600 Dec  3 02:50 cap320a.yuv
-rw-r--r-- 1 root root 153600 Dec  3 02:52 cap320b.yuv
-rw-r--r-- 1 root root      0 Dec  3 02:49 cap640a.yuv
```

얻은 cap320a.yuv 파일의 raw data를 16진수로 보았다.

```
root@beaglebone:~/Lab5_code/a_CaptureBone# hd cap320a.yuv | more
00000000  f1 7e f2 7d f1 7e f0 7f  f0 7d f0 7d f3 7d f6 7d  |.~.}.~...}.}.}.}|
00000010  f3 80 f0 7d ef 7d f1 7d  f3 7d f4 7d f6 7d f7 7d  |...}.}.}.}.}.}.}|
00000020  f7 7d f6 7d f5 7d f6 7d  f7 7d f6 7d f7 7c f9 7e  |.}.}.}.}.}.}.|.~|
00000030  f9 7f f7 7d f7 7d f8 7d  f5 7b f5 7d f7 7a fd 7d  |...}.}.}.{.}.z.}|
00000040  ff 7e fe 7d e2 7d d9 7d  dd 7d ff 7e ff 83 ff 7c  |.~.}.}.}.}.~...||
00000050  ff 83 ff 7d ff 83 ff 7d  ff 83 ff 7d ff 83 ff 7d  |...}...}...}...}|
00000060  ff 83 ff 7d ff 83 ff 7d  ff 81 ff 7d ff 80 ff 7d  |...}...}...}...}|
00000070  ff 80 ff 7d ff 83 ff 7d  ff 82 ff 7d ff 82 ff 7d  |...}...}...}...}|
00000080  ff 81 ff 7d ff 80 ff 7d  ff 80 ff 7d ff 82 e7 7d  |...}...}...}...}|
00000090  d1 70 d1 89 d0 70 d0 89  d0 70 d0 89 cf 70 d0 89  |.p...p...p...p..|
000000a0  d0 70 d1 89 d3 70 d2 89  d3 6c d6 89 d8 70 d9 89  |.p...p...l...p..|
000000b0  db 70 db 8c db 71 dc 89  de 70 de 89 e1 70 e1 8b  |.p...q...p...p..|
000000c0  e2 70 e1 8a e0 70 e2 8c  e3 70 e1 89 df 70 e2 89  |.p...p...p...p..|
000000d0  e3 6c e4 89 e4 6f e5 8b  e6 6c e7 8c e8 6c e9 8c  |.l...o...l...l..|
000000e0  e4 6c e2 89 de 70 d8 89  d1 70 c8 89 bf 70 b5 89  |.l...p...p...p..|
000000f0  af 70 a9 89 a0 70 8e 88  64 79 5c 80 5a 7d 5b 7d  |.p...p..dy\.Z}[}|
00000100  5c 7d 5b 7d 5b 80 60 7d  5f 7e 60 80 60 7d 61 80  |\}[}[.`}_~`.`}a.|
00000110  5d 7f 5d 7e 5d 80 5d 7d  5f 80 60 7d 5f 80 5e 7d  |].]~].]}_.`}_.^}|
00000120  60 80 60 7d 5f 80 5f 7d  5e 80 5e 7d 5e 80 62 7d  |`.`}_._}^.^}^.b}|
00000130  67 80 6b 7d 6e 80 71 7d  70 80 71 7d 71 7e 70 7d  |g.k}n.q}p.q}q~p}|
00000140  70 80 72 7d 73 80 72 7d  72 80 75 7d 79 80 7e 7d  |p.r}s.r}r.u}y.~}|
00000150  82 80 84 7d 85 80 85 7d  87 83 89 7b 8b 83 8e 7d  |...}...}...{...}|
00000160  91 83 99 79 9e 83 a3 79  9c 83 9f 79 a5 84 a6 78  |...y...y...y...x|
--More--
```

- Capture JPEG and see

비디오 width를 640, height를 480, pixelformat을 MJPG로 바꾸고, frame을 1로 하여 cap640a.jpg 파일을 얻었다. 데이터 크기를 확인한 결과 아웃풋이 잘 나온 것으로 보인다.

```
ight=480,pixelformat=1_code/a_CaptureBone# v4l2-ctl --set-fmt-video=width=640,he
root@beaglebone:~/Lab5_code/a_CaptureBone# ./Capture2 -c 1 -o > cap640a.jpg
.\nroot@beaglebone:~/Lab5_code/a_CaptureBone# ls -la cap*.jpg
-rw-r--r-- 1 root root 12828 Dec  3 02:56 cap640a.jpg
```

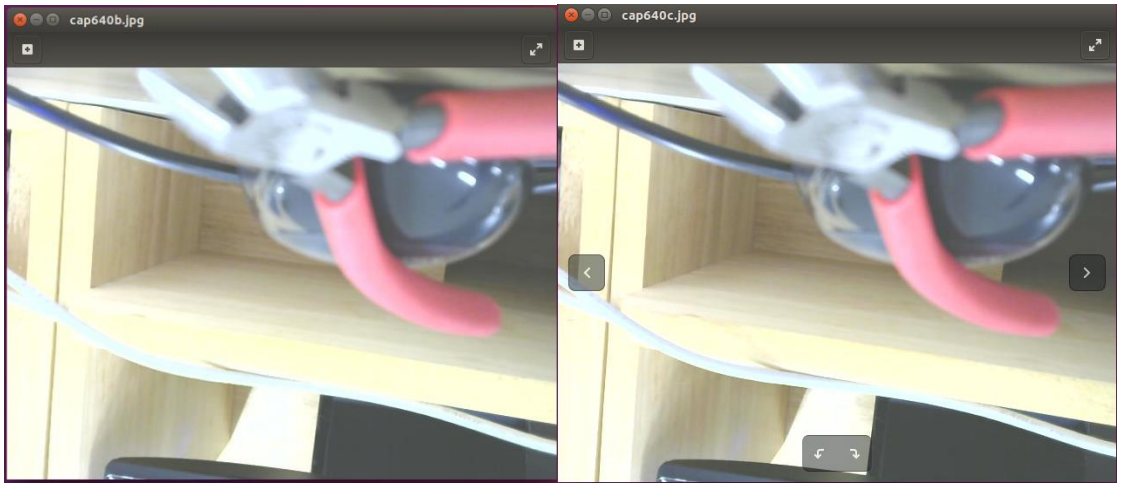
이번에는 먼저 mjpg 파일을 얻어보았다. frame을 100로 하여 cap640a.mjpg 파일을 얻었다. 데이터 크기를 확인한 결과 아웃풋이 잘 나온 것으로 보인다.

```
root@beaglebone:~/Lab5_code/a_CaptureBone# ./Capture2 -c 100 -o > cap640a.mjpg
.....\nroot@beaglebone:~/Lab5_code/a_CaptureBone#
root@beaglebone:~/Lab5_code/a_CaptureBone# ls -la cap*.mjpg
-rw-r--r-- 1 root root 2838760 Dec  3 03:00 cap640a.mjpg
```

그런 다음 frame을 1로 하여 cap640b.jpg, cap640c.jpg 파일을 얻었다. 데이터 크기를 확인한 결과 아웃풋이 잘 나온 것으로 보인다.

```
root@beaglebone:~/Lab5_code/a_CaptureBone# ./Capture2 -c 1 -o > cap640b.jpg
.\nroot@beaglebone:~/Lab5_code/a_CaptureBone# ./Capture2 -c 1 -o > cap640c.jpg
.\nroot@beaglebone:~/Lab5_code/a_CaptureBone# ls -la cap*.mjpg
-rw-r--r-- 1 root root 2838760 Dec  3 03:00 cap640a.mjpg
root@beaglebone:~/Lab5_code/a_CaptureBone# ls -la cap*.jpg
-rw-r--r-- 1 root root 12828 Dec  3 02:56 cap640a.jpg
-rw-r--r-- 1 root root 28449 Dec  3 03:01 cap640b.jpg
-rw-r--r-- 1 root root 28873 Dec  3 03:01 cap640c.jpg
```

얻은 파일을 pc로 옮겨 확인해보았다.



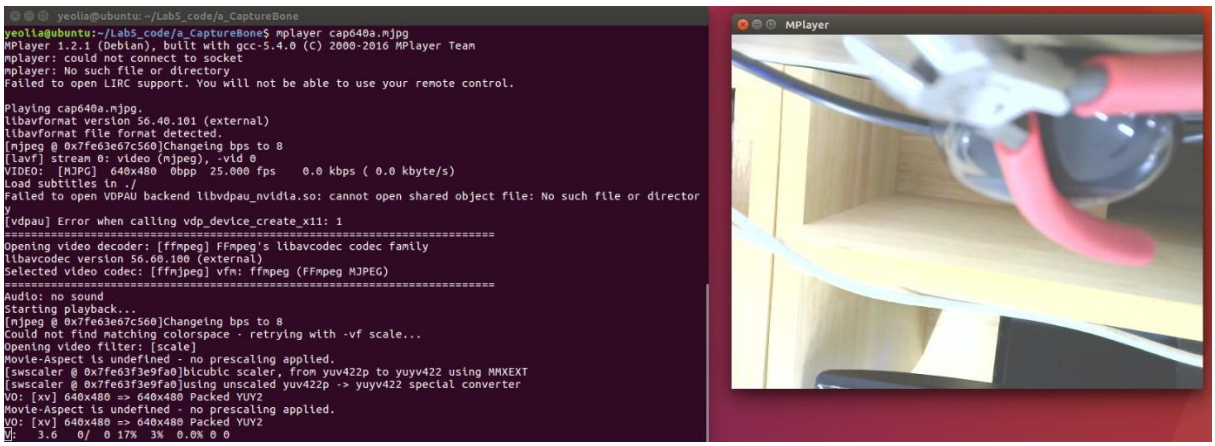
cap640a.jpg가 처음에는 잘 안보였지만 mjpg을 찍고 난 후 cap640b.jpg와 cap640c.jdg는 위와 같이 잘 보였다.

- Capture (record) MJPG video and see

다시 Bone에서 비디오 width를 640, height를 480, pixelformat을 MJPG로 바꾸고, frame을 300로 하여 cap640b.mjpg, cap640c.mjpg 파일을 얻었다. 데이터 크기를 확인한 결과 아웃풋이 잘 나온 것으로 보인다.

```
ight=480,pixelformat=1_code/a_CaptureBone# v4l2-ctl --set-fmt-video=width=640,he
root@beaglebone:~/Lab5_code/a_CaptureBone# ./Capture2 -c 300 -o > cap640b.mjpg
.....
.....
\nroot@beaglebone:~/Lab5_code/a_CaptureBone# ./Capture2 -c 300 -o > cap640c.mjpg
.....
.....
\nroot@beaglebone:~/Lab5_code/a_CaptureBone# ls -la cap*.mjpg
-rw-r--r-- 1 root root 2838760 Dec  3 03:00 cap640a.mjpg
-rw-r--r-- 1 root root 8601327 Dec  3 03:02 cap640b.mjpg
-rw-r--r-- 1 root root 8608725 Dec  3 03:03 cap640c.mjpg
```

PC에서 mplayer을 설치하고 실행해 파일을 보았다.



cap640a.mjpg, cap640b.mjpg, cap640c.mjpg 모두 정상적으로 재생되었다. mjpg 파일을 그냥 열었을 때는 사진만 보였지만 mplayer을 통하여 동영상임을 확인하였다.

2) Problem 5B. Learn SDL via Tutorials

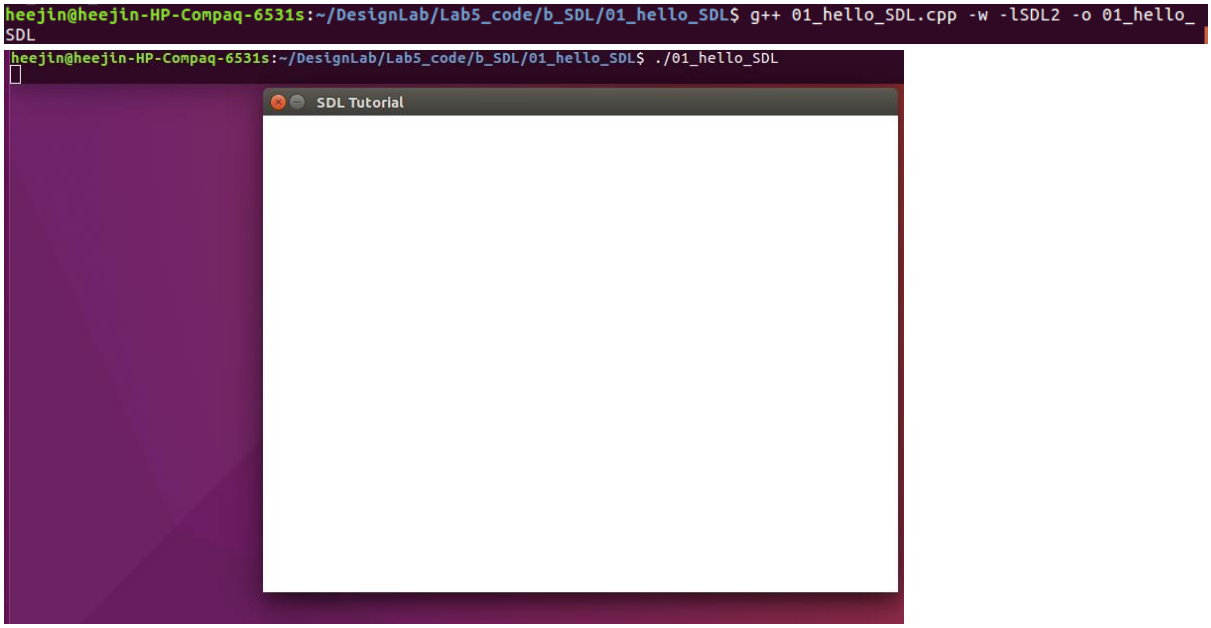
- Install SDL on PC

먼저 SDL를 설치하였다.

```
heejin@heejin-HP-Compaq-6531s:~$ sudo apt-get install libsdl2-*
[sudo] password for heejin:
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'libsdl2-mixer-dbg' for glob 'libsdl2-*'
Note, selecting 'libsdl2-mixer-dev' for glob 'libsdl2-*'
Note, selecting 'libsdl2-image-dbg' for glob 'libsdl2-*'
Note, selecting 'libsdl2-image-dev' for glob 'libsdl2-*'
Note, selecting 'libsdl2-gfx-dbg' for glob 'libsdl2-*'
Note, selecting 'libsdl2-gfx-dev' for glob 'libsdl2-*'
Note, selecting 'libsdl2-gfx-doc' for glob 'libsdl2-*'
Note, selecting 'libsdl2-mixer-2.0-0' for glob 'libsdl2-*'
Note, selecting 'libsdl2-dbg' for glob 'libsdl2-*'
Note, selecting 'libsdl2-dev' for glob 'libsdl2-*'
Note, selecting 'libsdl2-doc' for glob 'libsdl2-*'
Note, selecting 'libsdl2-ttf-dbg' for glob 'libsdl2-*'
Note, selecting 'libsdl2-ttf-dev' for glob 'libsdl2-*'
Note, selecting 'libsdl2-net-2.0-0' for glob 'libsdl2-*'
Note, selecting 'libsdl2-net-dbg' for glob 'libsdl2-*'
Note, selecting 'libsdl2-net-dev' for glob 'libsdl2-*'
Note, selecting 'libsdl2-image-2.0-0' for glob 'libsdl2-*'
Note, selecting 'libsdl2-2.0-0' for glob 'libsdl2-*'
Note, selecting 'libsdl2-gfx-1.0-0' for glob 'libsdl2-*'
```

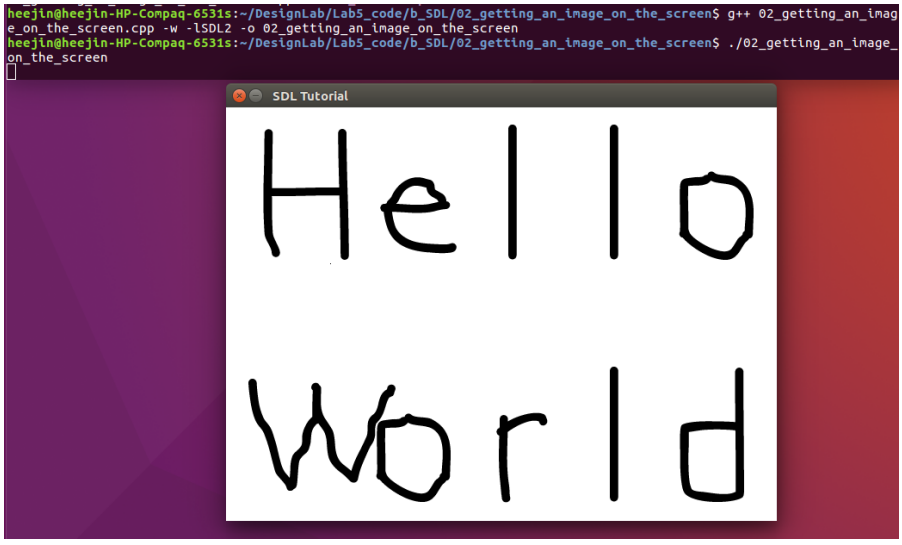
- Create SDL2 window

KLMS에 올려진 Lab5_code/b_SDL에 있는 코드를 모두 컴파일하여 실행해보았다. 먼저 창을 만드는 01_hello_SDL을 실행해보았다



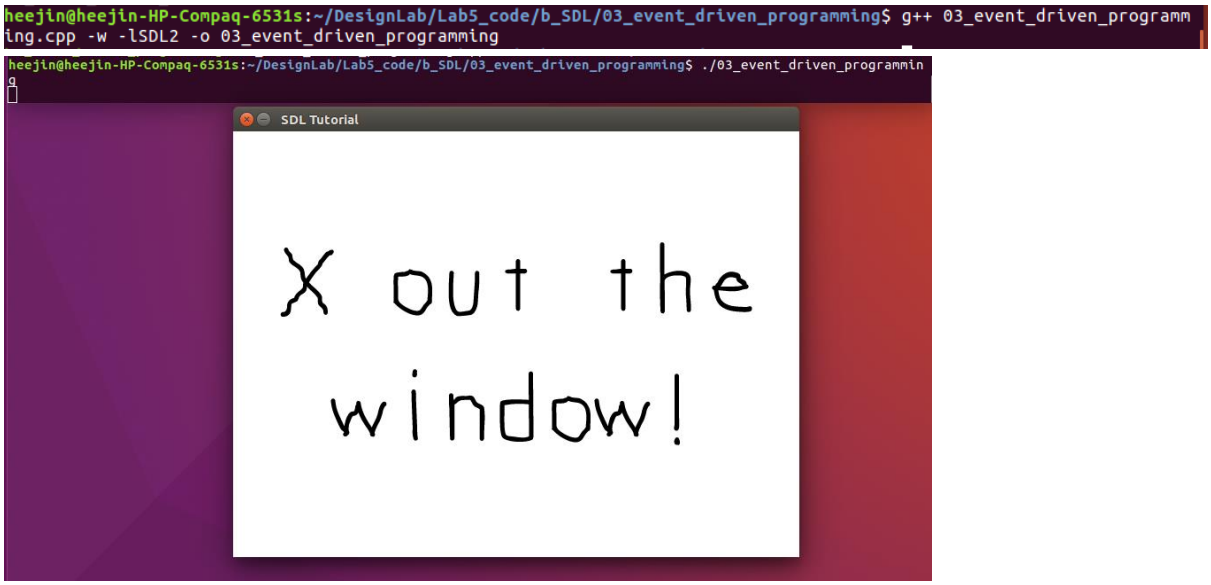
- View a BMP image on screen

다음으로 BMP 이미지 파일을 볼 수 있는 02_getting_an_image_on_the_screen을 실행해보았다.



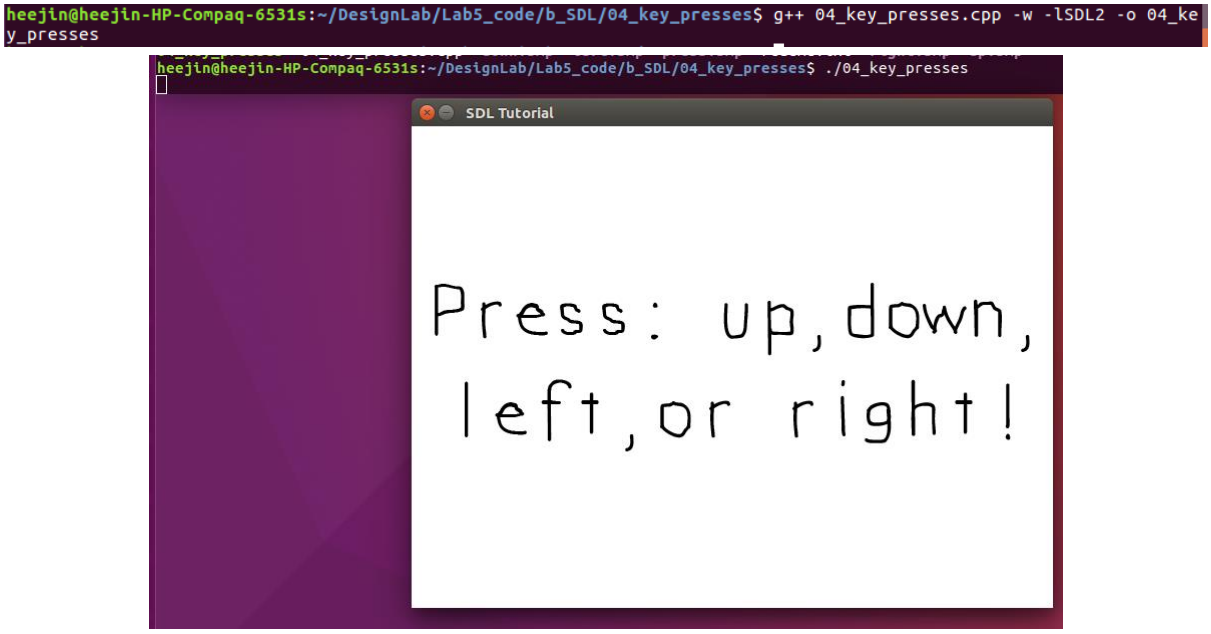
- Event driven programming

다음으로 창에서 x를 누르면 닫아지는 03_event_driven_programming을 실행해보았다.

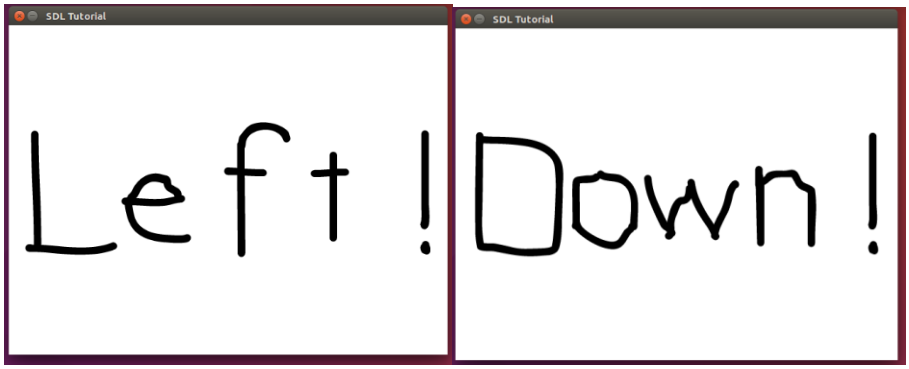


- Key press

다음으로 키보드의 키를 누르면 이미지가 전환되는 04_key_presses를 실행해보았다.



<↑키를 누를 때>　　<→키를 누를 때>

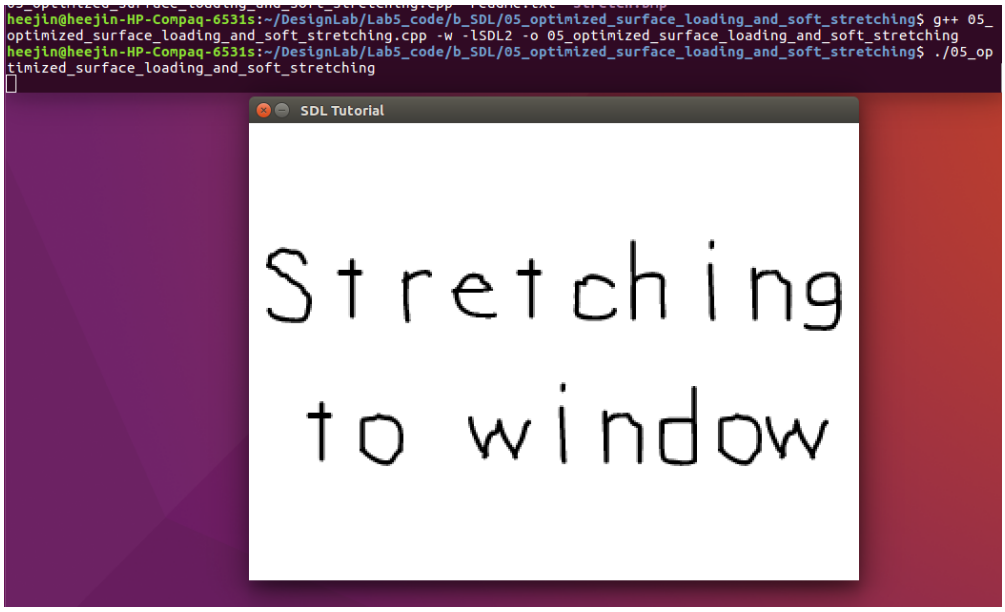


<←키를 누를 때>　　<↓키를 누를 때>

- Image stretching

다음으로 원래의 이미지에서 창에 맞게 이미지를 늘리는

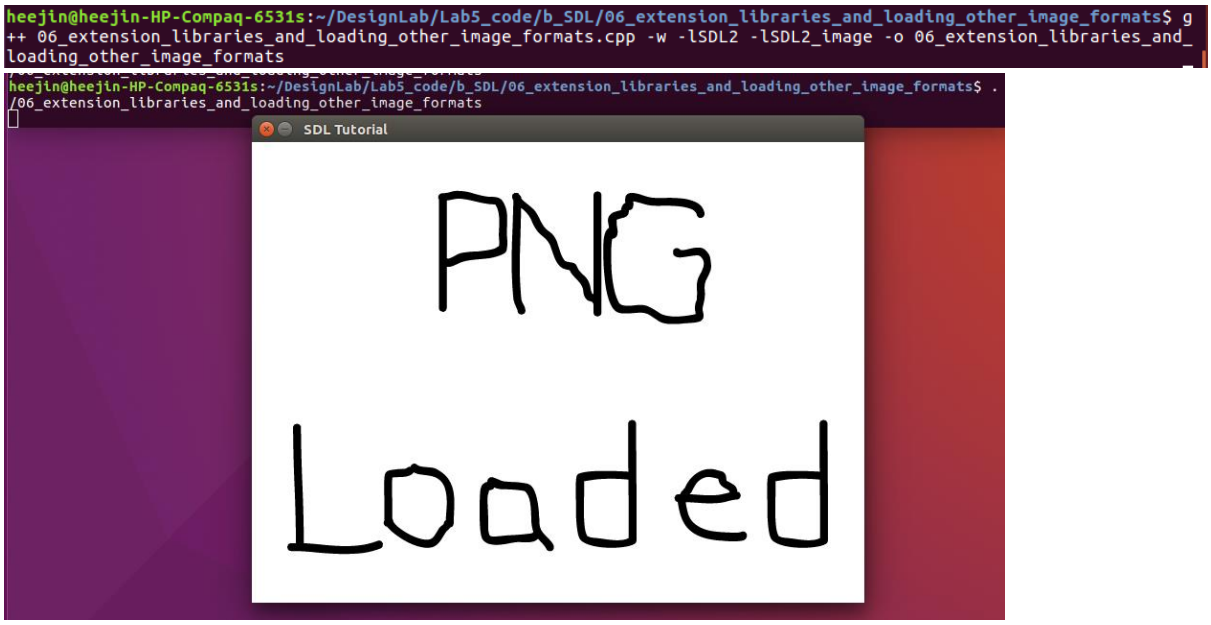
05_optimized_surface_loading_and_soft_streching를 실행해보았다.



- Load PNG image

다음으로 PNG 이미지 파일을 볼 수 있는

06_extension_libraries_and_loading_other_image_formats을 실행해보았다. Jpeg 이미지도 비슷한 방법으로 볼 수 있다.

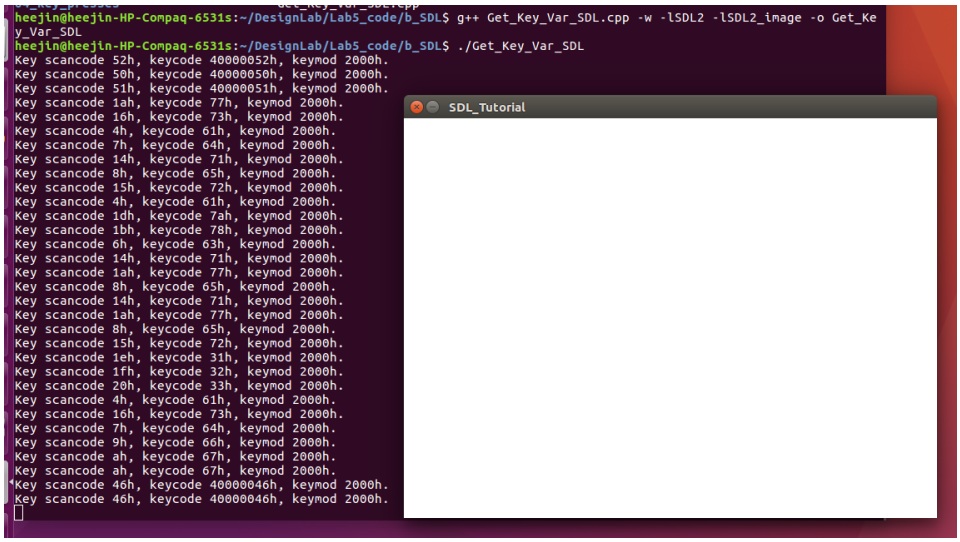


- Thread

46_multithreading도 실행해보았다.

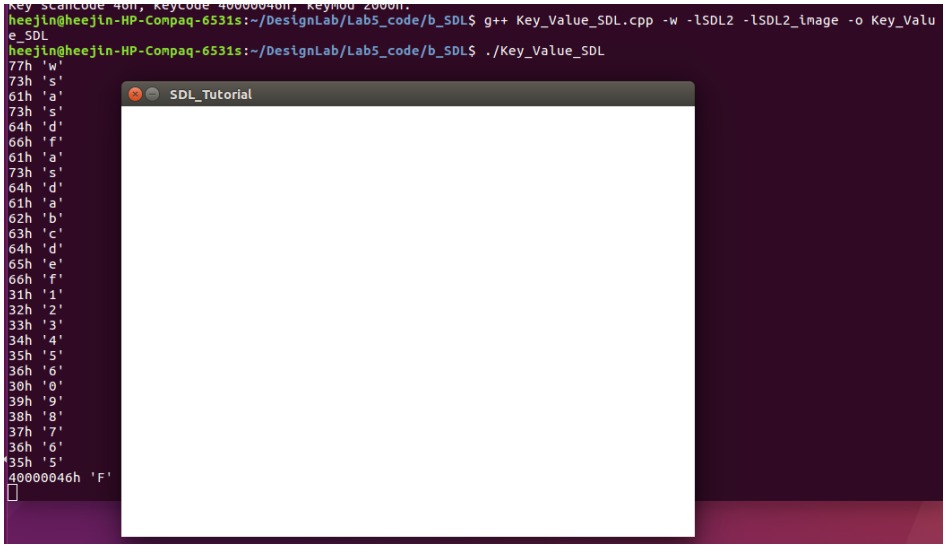
- Test Get_Key_Var_SDL

창을 나타내면서 입력한 키를 나타내는 Get_Key_Var_SDL을 실행해보았다.



- Test Key_Value_SDL

위와 비슷하게 창을 나타내면서 입력한 키를 나타내는 Key_Value_SDL를 실행해보았다. 어떤 키를 입력했는지 한눈에 알아볼 수 있다.



3) Problem 5C. Implement Video functionality & system integration

- Makefile

먼저 모터 제어를 위한 코드인 gpio를 사용하기 위하여 Lab4에서 사용한 gpio_control_test.c 파일을 사용하였다. 따라서 makefile을 수정하여 gpio_control_test.c를 같이 컴파일 할 수 있도록 하였다.

```
PC_CC = g++

BB_CC = arm-linux-gnueabihf-gcc

FLAGS = -g -w

LIBS = -lSDL2 -lSDL2_image -lpthread


all: Control_Camera Control_Viewer


Control_Camera: Control_Camera.c UDP.c

	$(BB_CC) $(FLAGS) -o Control_Camera Control_Camera.c UDP.c gpio_control_test.c -lpthread


Control_Viewer: Control_Viewer.cpp UDP.c

	$(PC_CC) $(FLAGS) -o Control_Viewer Control_Viewer.cpp UDP.c $(LIBS)


clean:

	rm Control_Camera Control_Viewer
```

- Control_Camera

다음은 Control_Camera.c 코드이다.

Control_Camara는 beaglebone에서 실행되는 코드로 총 2가지 역할을 수행한다. 첫 번째는, 카메라로 찍은 영상을 컴퓨터로 전송하는 역할, 두 번째는 컴퓨터에서 카메라를 제어하는 신호를 보내면, 신호를 해석하여 바퀴를 움직이는 역할이다. 이를 위해 main함수 안에서 두 역할이 서로 다른 thread하에 실행된다. 또한 communication에서는 client의 역할을 한다.

헤더 및 변수선언

```
/*
 *  V4L2 video capture example
 *
 *  This program can be used and distributed without restrictions.
 *
 *      This program is provided with the V4L2 API
 * see https://linuxtv.org/docs.php for more information
 */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <assert.h>

#include <getopt.h>             /* getopt_long() */

#include <fcntl.h>              /* low-level i/o */
#include <unistd.h>
#include <errno.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <sys/time.h>
#include <sys/mman.h>
#include <sys/ioctl.h>

#include <linux/videodev2.h>

#include <stdbool.h>
#include <pthread.h>

#include "UDP.h"
#include "gpio_control_test.h"

#define CLEAR(x) memset(&(x), 0, sizeof(x))

struct UDP_socket *sock_udp;

bool quit = false;

enum io_method {
        IO_METHOD_READ,
        IO_METHOD_MMAP,
        IO_METHOD_USERPTR,
};

struct buffer {
        void   *start;
        size_t  length;
};

static char            *dev_name;
static enum io_method   io = IO_METHOD_MMAP;
static int              fd = -1;
struct buffer          *buffers;
static unsigned int     n_buffers;
static int              out_buf;
static int              force_format;
static int              frame_count = 70;
```

다음은 기본 함수들로써 변동사항은 없다.

```
static void errno_exit(const char *s)
{
        fprintf(stderr, "%s error %d, %s\n", s, errno, strerror(errno));
        exit(EXIT_FAILURE);
}

static int xioctl(int fh, int request, void *arg)
{
        int r;

        do {
                r = ioctl(fh, request, arg);
        } while (-1 == r && EINTR == errno);

        return r;
}

static void process_image(const void *p, int size)
{
        static int i = 0;

        if (out_buf)
                        fwrite(p, size, 1, stdout);

        fflush(stderr);
        fprintf(stderr, ".");
        fflush(stdout);

        i++;

        if (UDP_send (sock_udp, p, size) > 0)
        printf ("send frame %d, size: %d\n", i, size);
}

static int read_frame(void)
{
        struct v4l2_buffer buf;
        unsigned int i;

        switch (io) {
        case IO_METHOD_READ:
                if (-1 == read(fd, buffers[0].start, buffers[0].length)) {
                        switch (errno) {
                        case EAGAIN:
                                return 0;

                        case EIO:
                                /* Could ignore EIO, see spec. */

                                /* fall through */

                        default:
                                errno_exit("read");
                        }
                }

                process_image(buffers[0].start, buffers[0].length);
                break;

        case IO_METHOD_MMAP:
                CLEAR(buf);

                buf.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
                buf.memory = V4L2_MEMORY_MMAP;

                if (-1 == xioctl(fd, VIDIOC_DQBUF, &buf)) {
                        switch (errno) {
                        case EAGAIN:
                                return 0;

                        case EIO:
                                /* Could ignore EIO, see spec. */

                                /* fall through */

                        default:
                                errno_exit("VIDIOC_DQBUF");
                        }
                }

                assert(buf.index < n_buffers);

                process_image(buffers[buf.index].start, buf.bytesused);

                if (-1 == xioctl(fd, VIDIOC_QBUF, &buf))
                        errno_exit("VIDIOC_QBUF");
                break;

        case IO_METHOD_USERPTR:
                CLEAR(buf);

                buf.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
                buf.memory = V4L2_MEMORY_USERPTR;
```

```
                if (-1 == xioctl(fd, VIDIOC_DQBUF, &buf)) {
                        switch (errno) {
                        case EAGAIN:
                                return 0;

                        case EIO:
                                /* Could ignore EIO, see spec. */

                                /* fall through */

                        default:
                                errno_exit("VIDIOC_DQBUF");
                        }
                }

                for (i = 0; i < n_buffers; ++i)
                        if (buf.m.userptr == (unsigned long)buffers[i].start
                            && buf.length == buffers[i].length)
                                break;

                assert(i < n_buffers);

                process_image((void *)buf.m.userptr, buf.bytesused);

                if (-1 == xioctl(fd, VIDIOC_QBUF, &buf))
                        errno_exit("VIDIOC_QBUF");
                break;
        }

        return 1;
}

static void mainloop(void)
{
        unsigned int count;
        unsigned int loopIsInfinite = 0;

        if (frame_count == 0) loopIsInfinite = 1;

        count = frame_count;
        //while (count-- > 0) {
        while ((count-- > 0) || loopIsInfinite) {
                if (quit) break;
                for (;;) {
                        fd_set fds;
                        struct timeval tv;
                        int r;

                        FD_ZERO(&fds);
                        FD_SET(fd, &fds);

                        /* Timeout. */
                        tv.tv_sec = 2;
                        tv.tv_usec = 0;

                        r = select(fd + 1, &fds, NULL, NULL, &tv);

                        if (-1 == r) {
                                if (EINTR == errno)
                                        continue;
                                errno_exit("select");
                        }

                        if (0 == r) {
                                fprintf(stderr, "select timeout\n");
                                exit(EXIT_FAILURE);
                        }

                        if (read_frame())
                                break;
                        /* EAGAIN - continue select loop. */
                }
        }
}

static void stop_capturing(void)
{
        enum v4l2_buf_type type;

        switch (io) {
        case IO_METHOD_READ:
                /* Nothing to do. */
                break;

        case IO_METHOD_MMAP:
        case IO_METHOD_USERPTR:
                type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
                if (-1 == xioctl(fd, VIDIOC_STREAMOFF, &type))
                        errno_exit("VIDIOC_STREAMOFF");
                break;
        }
}

static void start_capturing(void)
{
        unsigned int i;
```

```
        enum v4l2_buf_type type;

        switch (io) {
        case IO_METHOD_READ:
                /* Nothing to do. */
                break;

        case IO_METHOD_MMAP:
                for (i = 0; i < n_buffers; ++i) {
                        struct v4l2_buffer buf;

                        CLEAR(buf);
                        buf.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
                        buf.memory = V4L2_MEMORY_MMAP;
                        buf.index = i;

                        if (-1 == xioctl(fd, VIDIOC_QBUF, &buf))
                                errno_exit("VIDIOC_QBUF");
                }
                type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
                if (-1 == xioctl(fd, VIDIOC_STREAMON, &type))
                        errno_exit("VIDIOC_STREAMON");
                break;

        case IO_METHOD_USERPTR:
                for (i = 0; i < n_buffers; ++i) {
                        struct v4l2_buffer buf;

                        CLEAR(buf);
                        buf.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
                        buf.memory = V4L2_MEMORY_USERPTR;
                        buf.index = i;
                        buf.m.userptr = (unsigned long)buffers[i].start;
                        buf.length = buffers[i].length;

                        if (-1 == xioctl(fd, VIDIOC_QBUF, &buf))
                                errno_exit("VIDIOC_QBUF");
                }
                type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
                if (-1 == xioctl(fd, VIDIOC_STREAMON, &type))
                        errno_exit("VIDIOC_STREAMON");
                break;
        }
        printf("start_capturing pass\n");
}

static void uninit_device(void)
{
        unsigned int i;

        switch (io) {
        case IO_METHOD_READ:
                free(buffers[0].start);
                break;

        case IO_METHOD_MMAP:
                for (i = 0; i < n_buffers; ++i)
                        if (-1 == munmap(buffers[i].start, buffers[i].length))
                                errno_exit("munmap");
                break;

        case IO_METHOD_USERPTR:
                for (i = 0; i < n_buffers; ++i)
                        free(buffers[i].start);
                break;
        }

        free(buffers);
}

static void init_read(unsigned int buffer_size)
{
        buffers = calloc(1, sizeof(*buffers));

        if (!buffers) {
                fprintf(stderr, "Out of memory\n");
                exit(EXIT_FAILURE);
        }

        buffers[0].length = buffer_size;
        buffers[0].start = malloc(buffer_size);

        if (!buffers[0].start) {
                fprintf(stderr, "Out of memory\n");
                exit(EXIT_FAILURE);
        }
}

static void init_mmap(void)
{
        struct v4l2_requestbuffers req;

        CLEAR(req);

        req.count = 4;
        req.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
        req.memory = V4L2_MEMORY_MMAP;
```

```
        if (-1 == xioctl(fd, VIDIOC_REQBUFS, &req)) {
                if (EINVAL == errno) {
                        fprintf(stderr, "%s does not support "
                                 "memory mapping\n", dev_name);
                        exit(EXIT_FAILURE);
                } else {
                        errno_exit("VIDIOC_REQBUFS");
                }
        }

        if (req.count < 2) {
                fprintf(stderr, "Insufficient buffer memory on %s\n",
                         dev_name);
                exit(EXIT_FAILURE);
        }

        buffers = calloc(req.count, sizeof(*buffers));

        if (!buffers) {
                fprintf(stderr, "Out of memory\n");
                exit(EXIT_FAILURE);
        }

        for (n_buffers = 0; n_buffers < req.count; ++n_buffers) {
                struct v4l2_buffer buf;

                CLEAR(buf);

                buf.type        = V4L2_BUF_TYPE_VIDEO_CAPTURE;
                buf.memory      = V4L2_MEMORY_MMAP;
                buf.index       = n_buffers;

                if (-1 == xioctl(fd, VIDIOC_QUERYBUF, &buf))
                        errno_exit("VIDIOC_QUERYBUF");

                buffers[n_buffers].length = buf.length;
                buffers[n_buffers].start =
                        mmap(NULL /* start anywhere */,
                              buf.length,
                              PROT_READ | PROT_WRITE /* required */,
                              MAP_SHARED /* recommended */,
                              fd, buf.m.offset);

                if (MAP_FAILED == buffers[n_buffers].start)
                        errno_exit("mmap");
        }
}

static void init_userp(unsigned int buffer_size)
{
        struct v4l2_requestbuffers req;

        CLEAR(req);

        req.count  = 4;
        req.type   = V4L2_BUF_TYPE_VIDEO_CAPTURE;
        req.memory = V4L2_MEMORY_USERPTR;

        if (-1 == xioctl(fd, VIDIOC_REQBUFS, &req)) {
                if (EINVAL == errno) {
                        fprintf(stderr, "%s does not support "
                                 "user pointer i/o\n", dev_name);
                        exit(EXIT_FAILURE);
                } else {
                        errno_exit("VIDIOC_REQBUFS");
                }
        }

        buffers = calloc(4, sizeof(*buffers));

        if (!buffers) {
                fprintf(stderr, "Out of memory\n");
                exit(EXIT_FAILURE);
        }

        for (n_buffers = 0; n_buffers < 4; ++n_buffers) {
                buffers[n_buffers].length = buffer_size;
                buffers[n_buffers].start = malloc(buffer_size);

                if (!buffers[n_buffers].start) {
                        fprintf(stderr, "Out of memory\n");
                        exit(EXIT_FAILURE);
                }
        }
}

static void init_device(void)
{
        struct v4l2_capability cap;
        struct v4l2_cropcap cropcap;
        struct v4l2_crop crop;
        struct v4l2_format fmt;
        unsigned int min;

        if (-1 == xioctl(fd, VIDIOC_QUERYCAP, &cap)) {
                if (EINVAL == errno) {
```

```
                fprintf(stderr, "%s is no V4L2 device\n",
                        dev_name);
                exit(EXIT_FAILURE);
        } else {
                errno_exit("VIDIOC_QUERYCAP");
        }
}

if (!(cap.capabilities & V4L2_CAP_VIDEO_CAPTURE)) {
        fprintf(stderr, "%s is no video capture device\n",
                dev_name);
        exit(EXIT_FAILURE);
}

switch (io) {
case IO_METHOD_READ:
        if (!(cap.capabilities & V4L2_CAP_READWRITE)) {
                fprintf(stderr, "%s does not support read i/o\n",
                        dev_name);
                exit(EXIT_FAILURE);
        }
        break;

case IO_METHOD_MMAP:
case IO_METHOD_USERPTR:
        if (!(cap.capabilities & V4L2_CAP_STREAMING)) {
                fprintf(stderr, "%s does not support streaming i/o\n",
                        dev_name);
                exit(EXIT_FAILURE);
        }
        break;
}



/* Select video input, video standard and tune here. */


CLEAR(cropcap);

cropcap.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;

if (0 == xioctl(fd, VIDIOC_CROPCAP, &cropcap)) {
        crop.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
        crop.c = cropcap.defrect; /* reset to default */

        if (-1 == xioctl(fd, VIDIOC_S_CROP, &crop)) {
                switch (errno) {
                case EINVAL:
                        /* Cropping not supported. */
                        break;
                default:
                        /* Errors ignored. */
                        break;
                }
        }
} else {
        /* Errors ignored. */
}



CLEAR(fmt);

fmt.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
if (force_format) {
        fmt.fmt.pix.width       = 640;
        fmt.fmt.pix.height      = 480;
        fmt.fmt.pix.pixelformat = V4L2_PIX_FMT_YUYV;
        fmt.fmt.pix.field       = V4L2_FIELD_INTERLACED;

        if (-1 == xioctl(fd, VIDIOC_S_FMT, &fmt))
                errno_exit("VIDIOC_S_FMT");

        /* Note VIDIOC_S_FMT may change width and height. */
} else {
        /* Preserve original settings as set by v4l2-ctl for example */
        if (-1 == xioctl(fd, VIDIOC_G_FMT, &fmt))
                errno_exit("VIDIOC_G_FMT");
}

/* Buggy driver paranoia. */
min = fmt.fmt.pix.width * 2;
if (fmt.fmt.pix.bytesperline < min)
        fmt.fmt.pix.bytesperline = min;
min = fmt.fmt.pix.bytesperline * fmt.fmt.pix.height;
if (fmt.fmt.pix.sizeimage < min)
        fmt.fmt.pix.sizeimage = min;

switch (io) {
case IO_METHOD_READ:
        init_read(fmt.fmt.pix.sizeimage);
        break;

case IO_METHOD_MMAP:
        init_mmap();
        break;
```

```
        case IO_METHOD_USERPTR:
                init_userp(fmt.fmt.pix.sizeimage);
                break;
        }
        printf("init device pass\n");
}

static void close_device(void)
{
        if (-1 == close(fd))
                errno_exit("close");

        fd = -1;
}

static void open_device(void)
{
        struct stat st;

        if (-1 == stat(dev_name, &st)) {
                fprintf(stderr, "Cannot identify '%s': %d, %s\n",
                        dev_name, errno, strerror(errno));
                exit(EXIT_FAILURE);
        }

        if (!S_ISCHR(st.st_mode)) {
                fprintf(stderr, "%s is no device\n", dev_name);
                exit(EXIT_FAILURE);
        }

        fd = open(dev_name, O_RDWR /* required */ | O_NONBLOCK, 0);

        if (-1 == fd) {
                fprintf(stderr, "Cannot open '%s': %d, %s\n",
                        dev_name, errno, strerror(errno));
                exit(EXIT_FAILURE);
        }
        printf("open device pass\n");
}

static void usage(FILE *fp, int argc, char **argv)
{
        fprintf(fp,
                 "Usage: %s [options]\n\n"
                 "Version 1.3\n"
                 "Options:\n"
                 "-d | --device name   Video device name [%s]\n"
                 "-h | --help          Print this message\n"
                 "-m | --mmap          Use memory mapped buffers [default]\n"
                 "-r | --read          Use read() calls\n"
                 "-u | --userp         Use application allocated buffers\n"
                 "-o | --output        Outputs stream to stdout\n"
                 "-f | --format        Force format to 640x480 YUYV\n"
                 "-c | --count         Number of frames to grab [%i]\n"
                 "",
                 argv[0], dev_name, frame_count);
}
```

다음은 parsing에 필요한 option 코드이다.

```
static const char short_options[] = "d:hmruofc:a:p:";

static const struct option
long_options[] = {
        { "device", required_argument, NULL, 'd' },
        { "help",   no_argument,       NULL, 'h' },
        { "mmap",   no_argument,       NULL, 'm' },
        { "read",   no_argument,       NULL, 'r' },
        { "userp",  no_argument,       NULL, 'u' },
        { "output", no_argument,       NULL, 'o' },
        { "format", no_argument,       NULL, 'f' },
        { "count",  required_argument, NULL, 'c' },
                // please define variables for ip and port to use UDP
        { "address", required_argument, NULL, 'a'},
        { "port", required_argument, NULL, 'p'},
        { 0, 0, 0, 0 }
};
```

a와 p에 대한 옵션을 추가하였다.

다음은 가장 중요한 코드 중 하나인 로봇 바퀴를 제어하는 코드이다.

```
void* thread_WiFi_Control_TMR(void* vargs){
        FILE *duty0, *duty1, *duty2, *run0, *run1, *run2;
        int numbytes;
        char buf[100];
        char *token;
        int token_num = 0;

        gpio_export(30);
        gpio_export(31);

        gpio_set_dir(30,1);
        gpio_set_dir(31,1);
```

```
        if ((duty0 = fopen("/sys/devices/ocp.3/pwm_test_P9_22.15/duty", "w")) == NULL){
                printf("Error: PWM0 may not have been acquired\n");
                exit(0);
        }
        if ((run0 = fopen("/sys/devices/ocp.3/pwm_test_P9_22.15/run", "w")) == NULL){
                printf("Error: PWM0 may not have been acquired\n");
                exit(0);
        }
        if ((duty2 = fopen("/sys/devices/ocp.3/pwm_test_P9_14.16/duty", "w")) == NULL){
                printf("Error: PWM1 may not have been acquired\n");
                exit(0);
                }
        if ((run2 = fopen("/sys/devices/ocp.3/pwm_test_P9_14.16/run", "w")) == NULL){
                printf("Error: PWM1 may not have been acquired\n");
                exit(0);
        }
        if ((duty1 = fopen("/sys/devices/ocp.3/pwm_test_P8_19.17/duty", "w")) == NULL){
                printf("Error: PWM2 may not have been acquired\n");
                exit(0);
        }
        if ((run1 = fopen("/sys/devices/ocp.3/pwm_test_P8_19.17/run", "w")) == NULL){
                printf("Error: PWM2 may not have been acquired\n");
                exit(0);
        }

        while(1){
                numbytes = UDP_recv(sock_udp, buf, 99);
                if(numbytes < 4)
                        continue;
                printf("listener got packet\n");
                buf[numbytes] = '\0';
                if(strcmp(buf,"End") == 0){
                        quit = true;
                        break;
                }
                token = strtok(buf, ".");
                while (token != NULL){
                        token_num++;
                        token = strtok(NULL, ".");
                        if (token_num == 1){
                                fprintf(duty0, "%d",atoi(token));
                                fprintf(run0, "%d", 1);
                                fflush(duty0);
                                fflush(run0);
                        }
                        else if (token_num == 2){
                                fprintf(duty1, "%d",atoi(token));
                                fprintf(run1, "%d", 1);
                                fflush(duty1);
                                fflush(run1);
                        }
                        else if (token_num == 3){
                                fprintf(duty2, "%d",atoi(token));
                                fprintf(run2, "%d", 1);
                                fflush(duty2);
                                fflush(run2);
                        }
                        else if (token_num == 4){
                                gpio_set_value( 31, atoi(token));
                        }
                        else if (token_num == 5){
                                gpio_set_value( 30, atoi(token));
                        }
                }
                token_num = 0;
                usleep(1000);
        }
        /* Stop PWM */
        fprintf(run0, "%d", 0);
        fprintf(run1, "%d", 0);
        fprintf(run2, "%d", 0);
        /* Close GPIO_LED*/
        gpio_unexport(30);
        gpio_unexport(31);
        /* Close PWM sysfs files*/

        fclose(run0);
        fclose(run1);
        fclose(run2);

        fclose(duty0);
        fclose(duty1);
        fclose(duty2);

        return NULL;
}
```

위 코드는 컴퓨터에서 보내는 제어 신호를 해석하여 바퀴를 움직이는 코드이다.

먼저 LED 제어를 위한 gpio를 설정하고 바퀴를 제어하기 위하여 각각 알맞은 duty, run파일을 읽어 들인다. 그 후 컴퓨터에서 제어 신호를 보내면 udp_recv함수를 통하여 신호를 확인하고, parsing을 하여 각각 바퀴에 LED에 적적한 값을 보낸다. 제어 신호의 형식은

command_num.duty0.duty1.duty2.led0.led1의 형식이다. 예를 들어, 컴퓨터에서 처음 신호를 보낼 때, 제자리에 가만히 정지한체 LED1을 키는 신호를 보내면

1.1500000.1500000.1500000.0.1의 형식으로 신호가 들어온다.

만약 컴퓨터에서 End신호가 들어오게 되면 즉시 열어놓은 모든 파일을 fclose하고 gpio_unexport 한다.

Main 함수

```
// please insert the WiFi_Control_TMR.c which you wrote in lab 4
int main(int argc, char **argv)
{
        pthread_t tid;
        int port;
        char str_ip[32];
        dev_name = "/dev/video0";

        //  please write your own script using UDP_new_socket, UDP_socket_init
        sock_udp = UDP_new_socket();
        UDP_socket_init(sock_udp);

        for (;;) {
                int idx;
                int c;

                c = getopt_long(argc, argv,
                                short_options, long_options, &idx);

                if (-1 == c)
                        break;

                switch (c) {
                case 0: /* getopt_long() flag */
                        break;

                case 'd':
                        dev_name = optarg;
                        break;

                case 'h':
                        usage(stdout, argc, argv);
                        exit(EXIT_SUCCESS);

                case 'm':
                        io = IO_METHOD_MMAP;
                        break;

                case 'r':
                        io = IO_METHOD_READ;
                        break;

                case 'u':
                        io = IO_METHOD_USERPTR;
                        break;

                case 'o':
                        out_buf++;
                        break;

                case 'f':
                        force_format++;
                        break;

                case 'c':
                        errno = 0;
                        frame_count = strtol(optarg, NULL, 0);
                        if (errno)
                                errno_exit(optarg);
                        break;

                //  please write your own script of case 'a' for ip
                case 'a':
                        strcpy(str_ip, optarg);
                        break;
                //  please write your own script of case 'p' for port
                case 'p':
                        port = atoi(optarg);
                        break;

                default:
                        usage(stderr, argc, argv);
                        exit(EXIT_FAILURE);
                }
        }

                //  please write your own script using UDP_set_client, pthread_create
        UDP_set_client(sock_udp, port, str_ip);
        printf("port:%d\n", port);
        printf("ip:%s\n", str_ip);
        pthread_create(&tid, NULL, thread_WiFi_Control_TMR, NULL);

        frame_count = 0;

        printf("reach here\n");

        open_device();
        init_device();
        start_capturing();
        mainloop();
        stop_capturing();
        uninit_device();
        close_device();
        fprintf(stderr, "\n");

                //  please write your own script using pthread_join
        pthread_join(tid, NULL);
        return 0;
}
```

먼저

```
        sock_udp = UDP_new_socket();
        UDP_socket_init(sock_udp);
```

을 통하여 통신을 위한 socket을 만들어 준다.

```
                case 'a':
                        strcpy(str_ip, optarg);
                        break;
                //  please write your own script of case 'p' for port
                case 'p':
                        port = atoi(optarg);
                        break;
```

-a 옵션을 통하여 ip주소를 받아온다. 이번 실험의 경우 컴퓨터의 ip인 192.168.0.16을 입력하여 주었다. -p옵션은 port를 설정하는 값으로, 이번 실험의 경우 4960값을 사용하였다.

```
        UDP_set_client(sock_udp, port, str_ip);
        pthread_create(&tid, NULL, thread_WiFi_Control_TMR, NULL);

        frame_count = 0;

        open_device();
        init_device();
        start_capturing();
        mainloop();
        stop_capturing();
        uninit_device();
        close_device();
        fprintf(stderr, "\n");

                //  please write your own script using pthread_join
        pthread_join(tid, NULL);
        return 0;
```

받아들인 ip, port 값을 통하여 udp client를 설정하여 주고 위에서 설명한 thread_WiFi_Control_TMF함수를 새로운 thread를 생성하여 돌려준다. 그와 동시에 밑의 코드는 카메라에서 찍은 영상을 컴퓨터로 보내는 함수들이다. 컴퓨터에서 end신호를 보내면 즉시 두 thread 모두 종료하게 된다.

- Control_Viewer

Control_Viewer.cpp는 컴퓨터에서 실행되는 코드로 2가지 역할이 있다. 첫 번째는 beaglebone에서 보내는 영상신호를 컴퓨터 화면상에 띄우는 역할이고, 두 번째는 바퀴를 제어하기 위한 신호를 beaglebone으로 보내는 역할이다. 두 역할은 각각 다른 thread하에서 실행된다. 통신에서는 server역할을 맡게된다.

```
#include <stdio.h>
```

```
#include <stdlib.h>
#include <unistd.h>
#include <SDL2/SDL.h>
#include <SDL2/SDL_image.h>
#include <SDL2/SDL_ttf.h>
#include <SDL2/SDL_mixer.h>
#include <string>
#include <sys/time.h>
#include <pthread.h>
#include <netdb.h>
#include "UDP.h"
```

위와 같은 헤더 파일을 include하였다.

```
//Screen dimension constants
const int SCREEN_WIDTH = 320;
const int SCREEN_HEIGHT = 240;

//Starts up SDL and creates window
bool init_SDL();

//Frees media and shuts down SDL
void close_SDL();

//Loads individual image
SDL_Surface* loadSurface(void *buf, size_t size);

//The window we'll be rendering to
SDL_Window* gWindow = NULL;

//The surface contained by the window
SDL_Surface* gScreenSurface = NULL;

//Current displayed PNG image
SDL_Surface* gJPGSurface = NULL;

//Maximum JPG file size//
size_t FRAME_SIZE = 60000;

//frame buffer
char* frame = NULL;

//UDP socket pointer//
struct UDP_socket *sock_udp;

bool quit = false;
```

위와 같은 전역변수를 선언하였다. 이 때, 촬영하는 카메라의 해상도에 따라 Screen dimension constants를 적절히 바꿔 주었다. 또한 640*480의 해상도에서는 한 frame의 사이즈가 대략 50000 언저리로 보내어 졌기 때문에 FRAME_SIZE = 60000으로 설정하였다.

```
bool init_SDL(void)
{
  //Initialization flag
  bool success = true;

  //Initialize SDL
  if( SDL_Init( SDL_INIT_VIDEO ) < 0 )
  {
    printf( "SDL could not initialize! SDL Error: %s\n", SDL_GetError() );
    success = false;
  }
  else
  {
    //Create window
    gWindow = SDL_CreateWindow("Lab5", SDL_WINDOWPOS_UNDEFINED, SDL_WINDOWPOS_UNDEFINED, SCREEN_WIDTH, SCREEN_HEIGHT, SDL_WINDOW_SHOWN );
    if( gWindow == NULL )
    {
      printf( "Window could not be created! SDL Error: %s\n", SDL_GetError() );
      success = false;
    }
    else
    {
      //Initialize JPG loading
      int imgFlags = IMG_INIT_JPG;
      if( !( IMG_Init( imgFlags ) & imgFlags ) )
      {
        printf( "SDL_image could not initialize! SDL_image Error: %s\n", IMG_GetError() );
        success = false;
      }
      else
      {
        //Get window surface
        gScreenSurface = SDL_GetWindowSurface(gWindow);
        //Fill the surface white
        SDL_FillRect(gScreenSurface, NULL, SDL_MapRGB( gScreenSurface->format, 0xFF, 0xFF, 0xFF ) );
        //Update the surface
        SDL_UpdateWindowSurface(gWindow);
      }
    }
  }

  return success;
}

void close_SDL()
{
  //Free loaded image
  SDL_FreeSurface( gJPGSurface );
  gJPGSurface = NULL;

  //Destroy window
```

```
  SDL_DestroyWindow( gWindow );
  gWindow = NULL;

  //Quit SDL subsystems
  IMG_Quit();
  SDL_Quit();
}

SDL_Surface* loadSurface(void *buf, size_t size)
{
  //The final optimized image
  SDL_Surface* optimizedSurface = NULL;

  //Load image at specified path
  SDL_RWops *rw_jpg = SDL_RWFromMem (buf, size);
  SDL_Surface* loadedSurface = IMG_Load_RW(rw_jpg, 1);
  if( loadedSurface == NULL )
  {
    printf( "Unable to load image from buffer! SDL_image Error: %s\n", IMG_GetError() );
  }
  else
  {
    //Convert surface to screen format
    optimizedSurface = SDL_ConvertSurface( loadedSurface, gScreenSurface->format, NULL );
    if( optimizedSurface == NULL )
    {
      printf( "Unable to optimize image! SDL Error: %s\n", SDL_GetError() );
    }

    //Get rid of old loaded surface
    SDL_FreeSurface( loadedSurface );
  }

  return optimizedSurface;
}
```

기본 함수들로 필요한 부분을 적절히 채워주었다.

```
void *thread_viewer (void *vargs)
{
  int n;
  bool isJPG;
  printf("start capturing\n");
  while (!quit) {
    //recv UDP data//
    n = UDP_recv (sock_udp, frame, FRAME_SIZE);

    //check JPG file//
    if (n < 4)
      continue;
    isJPG = true;

    if (frame[0] != 0xFF) isJPG = false;
    if (frame[1] != 0xD8) isJPG = false;
    if (frame[n-2] != 0xFF) isJPG = false;
    if (frame[n-1] != 0xD9) isJPG = false;

    if (!isJPG)
      continue;



    gJPGSurface = loadSurface (frame, n);
    SDL_BlitSurface (gJPGSurface, NULL, gScreenSurface, NULL);
    SDL_FreeSurface (gJPGSurface);
    gJPGSurface = NULL;
    SDL_UpdateWindowSurface (gWindow);
  }

  return NULL;
}
```

thread_viewer함수로 상당히 중요한 역할을 한다. beaglebone에서 보내는 영상 신호를 udp_recv함수를 통하여 받은 다음, jpg형식 체크를 하고 각 frame을 surface상에 띄우는 역할을 한다.

Main

```
int main (int argc, char *argv[])
{
  //Control value//
  int ivx = 0;
  int ivy = 0;
  int iw = 0;
  int LL = 0;
  int RL = 0;
  int command_id = 0;
  struct timeval now;
  struct timeval set;
  int sec, msec;
  pthread_t tid;

  float vx, vy, wr;
  float r, G;
  r = 0.02;
  G = 30000;
  int g2 = 3000000;

  float p[3];
  float w[3];
  float v[3];
  float mT[3][3] =  { {0, -0.67, 0.33*0.08}, {-0.577, 0.33, 0.33*0.08}, {0.577, 0.33, 0.33*0.08}}; //the transfrom matrix
```

```
//Control string//
char pressed_key;
char str_ctrl[100] = {0};
char send_str[100] = {0};
char buffer[100] = {0};

//SDL Event//
SDL_Event e;

//Init UDP socket//
sock_udp = UDP_new_socket();
UDP_socket_init(sock_udp);
UDP_set_server(sock_udp, 4960);



//Init SDL socket//
init_SDL();

//Init frame buffer//
frame = (char*)malloc(sizeof(char)*FRAME_SIZE);

pthread_create(&tid, NULL, thread_viewer, NULL);
gettimeofday(&set, NULL);
//Event//
while (!quit) {
  while (SDL_PollEvent(&e) != 0) {
    if (e.type == SDL_QUIT) {
      quit = true;
    }
    else if (e.type == SDL_KEYDOWN) {

      bool pressed = true;

      switch (e.key.keysym.scancode) {
        case SDL_SCANCODE_F12:
          printf ("pressed: F12 (Quit)\n");
          UDP_send (sock_udp, (void *)"End", 4);
          pressed = false;
          quit = true;
          break;

        case SDL_SCANCODE_1:
          pressed_key = '1';
          break;

        case SDL_SCANCODE_2:
          pressed_key = '2';
          break;

        case SDL_SCANCODE_3:
          pressed_key = '3';
          break;

        case SDL_SCANCODE_Q:
          pressed_key = 'Q';
          LL ^= 0x01;
          break;

        case SDL_SCANCODE_W:
          pressed_key = 'W';
          ivx++;
          if (ivx > 10)
            ivx = 10;
          break;

        case SDL_SCANCODE_E:
          pressed_key = 'E';
          RL ^= 0x01;
          break;

        case SDL_SCANCODE_A:
          pressed_key = 'A';
          ivy++;
          if (ivy < -10)
            ivy = -10;
          break;

        case SDL_SCANCODE_S:
          pressed_key = 'S';
          ivx = 0;
          ivy = 0;
          iw = 0;
          break;

        case SDL_SCANCODE_D:
          pressed_key = 'D';
          ivy--;
          if (ivy > 10)
            ivy = 10;
          break;

        case SDL_SCANCODE_Z:
          pressed_key = 'Z';
          iw++;
          if (iw > 10)
            iw = 10;
          break;

        case SDL_SCANCODE_X:
          pressed_key = 'X';
          ivx--;
          if (ivx < -10)
            ivx = -10;
          break;

        case SDL_SCANCODE_C:
          pressed_key = 'C';
          iw--;
          if (iw < -10)
            iw = -10;
          break;

        default:
          pressed = false;
      }
```

```
      if (pressed) {
        command_id++;

        gettimeofday(&now, NULL);
        msec = (now.tv_sec * 1e6 + now.tv_usec) - (set.tv_sec * 1e6 + set.tv_usec);
        msec /= 1000;
        sec = msec / 1000;
        msec = msec % 1000;

        sprintf (str_ctrl, "%d %d.%03d %d %d %d %d %d",
                 command_id, sec, msec, ivx * 10, ivy * 10, iw * 1000, LL, RL);

        vx = 0.02 * (float)ivx*10;
        vy = 0.02 * (float)ivy*10;
        wr = 0.01 * (float)iw*1000;

        v[0] = vx;
        v[1] = vy;
        v[2] = wr;

        for(int i = 0; i < 3; i++){
          w[i] = 0;
          for(int j = 0; j < 3; j++){
            w[i] += mT[i][j] * v[j];
          }
        }

        w[0] /= r;
        w[1] /= r;
        w[2] /= r;

        p[0] = /*1510095*/ 1500000 - G * w[0];
        p[1] = /*1465450*/ 1500000 - G * w[1];
        p[2] = /*1462500*/ 1500000 - G * w[2];

        if (p[1] < 1500000){
          p[1] -= 500000;
        }
        if (p[2] < 150000){
            p[2] -= 500000;
        }
        if (p[0] < 1500000){
          p[0] -= 500000;
        }
        if (p[1] > 1500000){
            p[1] += 500000;
        }
        if (p[2] > 1500000){
            p[2] += 500000;
        }
        if (p[0] > 1500000){
            p[0] += 500000;
        }
    //  please write your own script using UDP_send
        sprintf(buffer, "%d", command_id);
        strcat(send_str, buffer);
        strcat(send_str, ".");
        sprintf(buffer, "%d", (int)p[1]);
        strcat(send_str, buffer);
        strcat(send_str, ".");
        sprintf(buffer, "%d", (int)p[2]);
        strcat(send_str, buffer);
        strcat(send_str, ".");
        sprintf(buffer, "%d", (int)p[0]);
        strcat(send_str, buffer);
        strcat(send_str, ".");
        sprintf(buffer, "%d", (int)LL);
        strcat(send_str, buffer);
        strcat(send_str, ".");
        sprintf(buffer, "%d", (int)RL);
        strcat(send_str, buffer);

        UDP_send(sock_udp, send_str, strlen(send_str));

        printf ("\t\t\t\t\t%s\n", str_ctrl);
        printf ("\t\t\t\t\t%s\n", send_str);

        memset(send_str, 0, sizeof(send_str));
      }
    }
  }
}
pthread_join (tid, NULL);

free (frame);
frame = NULL;

close_SDL ();
UDP_close (sock_udp);

return 0;
}
```

Main함수이다.

```
  //Control value//
```

```
  int ivx = 0;
  int ivy = 0;
  int iw = 0;
  int LL = 0;
  int RL = 0;
  int command_id = 0;
  struct timeval now;
  struct timeval set;
  int sec, msec;
  pthread_t tid;

  float vx, vy, wr;
  float r, G;
  r = 0.02;
  G = 30000;
  int g2 = 3000000;

  float p[3];
  float w[3];
  float v[3];
  float mT[3][3] =  { {0, -0.67, 0.33*0.08}, {-0.577, 0.33, 0.33*0.08},
{0.577, 0.33, 0.33*0.08}}; //the transfrom matrix


  //Control string//
  char pressed_key;
  char str_ctrl[100] = {0};
  char send_str[100] = {0};
  char buffer[100] = {0};
```

위 값들은 추후 바퀴를 제어하는데 쓰이는 변수들이다.

```
  //SDL Event//
  SDL_Event e;

  //Init UDP socket//
  sock_udp = UDP_new_socket();
  UDP_socket_init(sock_udp);
  UDP_set_server(sock_udp, 4960);


  //Init SDL socket//
  init_SDL();

  //Init frame buffer//
  frame = (char*)malloc(sizeof(char)*FRAME_SIZE);
```

```
  pthread_create(&tid, NULL, thread_viewer, NULL);
```

Control_viewer는 컴퓨터 상에서 실행되고 서버의 역할을 하여야 한다. port 번호는 4960으로 설정하였다. 그 후, 위에서 설명한 thread_viewer함수를 새로운 thread하에서 실행하였다.

```
gettimeofday(&set, NULL);
  //Event//
  while (!quit) {
    while (SDL_PollEvent(&e) != 0) {
      if (e.type == SDL_QUIT) {
        quit = true;
      }
      else if (e.type == SDL_KEYDOWN) {

        bool pressed = true;

        switch (e.key.keysym.scancode) {
          case SDL_SCANCODE_F12:
            printf ("pressed: F12 (Quit)\n");
            UDP_send (sock_udp, (void *)"End", 4);
            pressed = false;
            quit = true;
            break;

          case SDL_SCANCODE_1:
            pressed_key = '1';
            break;

          case SDL_SCANCODE_2:
            pressed_key = '2';
            break;

          case SDL_SCANCODE_3:
            pressed_key = '3';
            break;

          case SDL_SCANCODE_Q:
            pressed_key = 'Q';
            LL ^= 0x01;
            break;

          case SDL_SCANCODE_W:
            pressed_key = 'W';
            ivx++;
            if (ivx > 10)
              ivx = 10;
            break;
```

```
        case SDL_SCANCODE_E:
          pressed_key = 'E';
          RL ^= 0x01;
          break;

        case SDL_SCANCODE_A:
          pressed_key = 'A';
          ivy++;
          if (ivy < -10)
            ivy = -10;
          break;

        case SDL_SCANCODE_S:
          pressed_key = 'S';
          ivx = 0;
          ivy = 0;
          iw = 0;
          break;

        case SDL_SCANCODE_D:
          pressed_key = 'D';
          ivy--;
          if (ivy > 10)
            ivy = 10;
          break;

        case SDL_SCANCODE_Z:
          pressed_key = 'Z';
          iw++;
          if (iw > 10)
            iw = 10;
          break;

        case SDL_SCANCODE_X:
          pressed_key = 'X';
          ivx--;
          if (ivx < -10)
            ivx = -10;
          break;

        case SDL_SCANCODE_C:
          pressed_key = 'C';
          iw--;
          if (iw < -10)
            iw = -10;
          break;

        default:
```

```
            pressed = false;
        }

        if (pressed) {
          command_id++;

          gettimeofday(&now, NULL);
          msec = (now.tv_sec * 1e6 + now.tv_usec) - (set.tv_sec * 1e6 +
set.tv_usec);
          msec /= 1000;
          sec = msec / 1000;
          msec = msec % 1000;

          sprintf (str_ctrl, "%d %d.%03d %d %d %d %d %d",
                   command_id, sec, msec, ivx * 10, ivy * 10, iw * 1000, LL,
RL);

          vx = 0.02 * (float)ivx*10;
          vy = 0.02 * (float)ivy*10;
          wr = 0.01 * (float)iw*1000;

          v[0] = vx;
          v[1] = vy;
          v[2] = wr;

          for(int i = 0; i < 3; i++){
            w[i] = 0;
            for(int j = 0; j < 3; j++){
              w[i] += mT[i][j] * v[j];
            }
          }

          w[0] /= r;
          w[1] /= r;
          w[2] /= r;

          p[0] = /*1510095*/ 1500000 - G * w[0];
          p[1] = /*1465450*/ 1500000 - G * w[1];
          p[2] = /*1462500*/ 1500000 - G * w[2];

          if (p[1] < 1500000){
            p[1] -= 500000;
          }
          if (p[2] < 150000){
              p[2] -= 500000;
          }
          if (p[0] < 1500000){
            p[0] -= 500000;
```

```
        }
        if (p[1] > 1500000){
            p[1] += 500000;
        }
        if (p[2] > 1500000){
            p[2] += 500000;
        }
        if (p[0] > 1500000){
            p[0] += 500000;
        }
  //  please write your own script using UDP_send
        sprintf(buffer, "%d", command_id);
        strcat(send_str, buffer);
        strcat(send_str, ".");
        sprintf(buffer, "%d", (int)p[1]);
        strcat(send_str, buffer);
        strcat(send_str, ".");
        sprintf(buffer, "%d", (int)p[2]);
        strcat(send_str, buffer);
        strcat(send_str, ".");
        sprintf(buffer, "%d", (int)p[0]);
        strcat(send_str, buffer);
        strcat(send_str, ".");
        sprintf(buffer, "%d", (int)LL);
        strcat(send_str, buffer);
        strcat(send_str, ".");
        sprintf(buffer, "%d", (int)RL);
        strcat(send_str, buffer);

        UDP_send(sock_udp, send_str, strlen(send_str));

        printf ("\t\t\t\t\t%s\n", str_ctrl);
        printf ("\t\t\t\t\t%s\n", send_str);

        memset(send_str, 0, sizeof(send_str));
      }
    }
  }
}
pthread_join (tid, NULL);
```

바퀴를 제어하는 코드이다. 키보드가 눌리면 SDL_PollEvent가 실행되는데, 입력된 자판에 따라서 필요한 행동을 취한다. F12의 경우에는 quit신호로 beaglebone으로 End를 보낸다. 또한 main에서는 quit = true값으로 변하여 main함수를 종료할 수 있게 된다. QWEASDZXC는 바퀴 제어 신호이다. 기존의 코드와 약간 달라진 점이 있는데, 제어 신호 duty값을 control_viewer에서 모두 계산한 다음 beaglebone으로 신호를 보내도록 수정하였다. 따라서

```
        vx = 0.02 * (float)ivx*10;
        vy = 0.02 * (float)ivy*10;
        wr = 0.01 * (float)iw*1000;
```

위와 같이 vx,vy,wr의 scale을 변경하여 transform matrix에 값이 전달되도록 하였다. 위 값이 Lab4에서 사용한 값과 동일한 값이고 실험 결과 바퀴가 정상적으로 동작함을 확인할 수 있었다. 제어 신호의 형식은 위 Control_Camera에서 설명하였다.

```
  free (frame);
  frame = NULL;

  close_SDL ();
  UDP_close (sock_udp);

  return 0;
```

만약 quit 값이 true가 되었을 경우 프로그램을 종료하는 코드이다.

- Setting PWMs

먼저 바퀴를 움직이기 위해 먼저 해줘야 하는 세팅을 해주었다. Lab 3에서와 같은 방법으로 P9-22, P9-14, P8-19 바퀴를 이용하여 세팅하였다.

```
root@beaglebone:/home/jungwungpark/lab3/3_MobileRobot/c_PWM_Servo_Shell# ./Acquire_Triple_PWMs.sh
baseboard  driver  modalias  power  slot-4  slot-5  slot-7  slots  subsystem  uevent
Checking for Individual PWM
/sys/devices/ocp.3/pwm_test_P8_19.17:
modalias  power  subsystem  uevent

/sys/devices/ocp.3/pwm_test_P9_14.16:
modalias  power  subsystem  uevent

/sys/devices/ocp.3/pwm_test_P9_22.15:
modalias  power  subsystem  uevent
modalias  power  subsystem  uevent
modalias  power  subsystem  uevent
modalias  power  subsystem  uevent
PWM Acquired
```

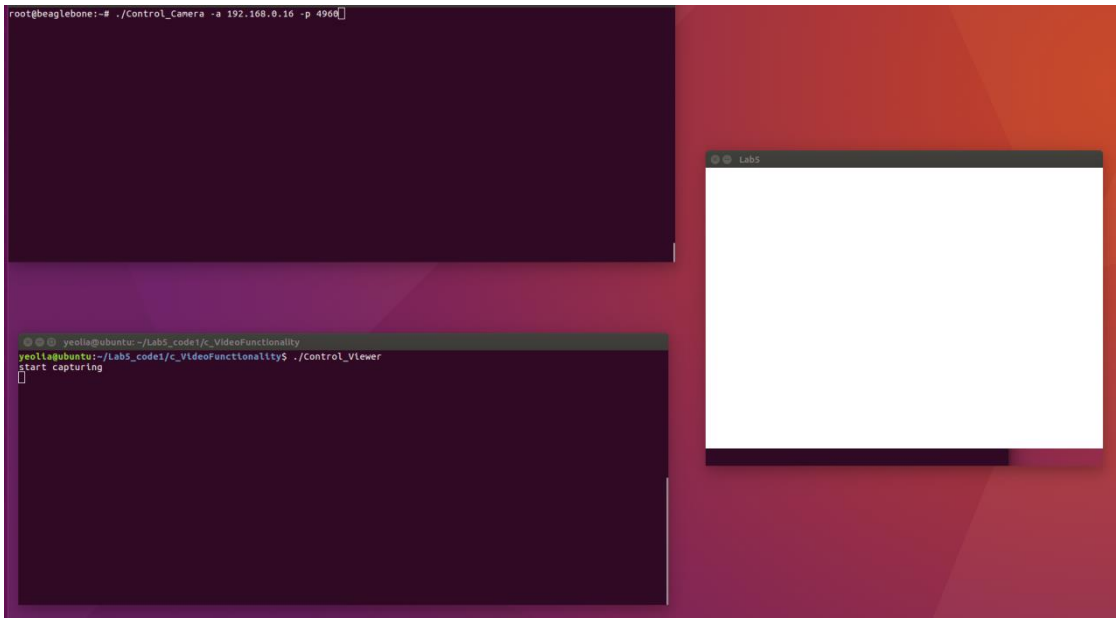
그 이후에도 "cd ../ocp.3/pwm_test_P9_22.15", "echo 3000000 > period", "echo 2000000 > duty" 커맨드를 이용하였고, 다른 바퀴에 대해서도 마찬가지 방법으로 세팅해주었다.

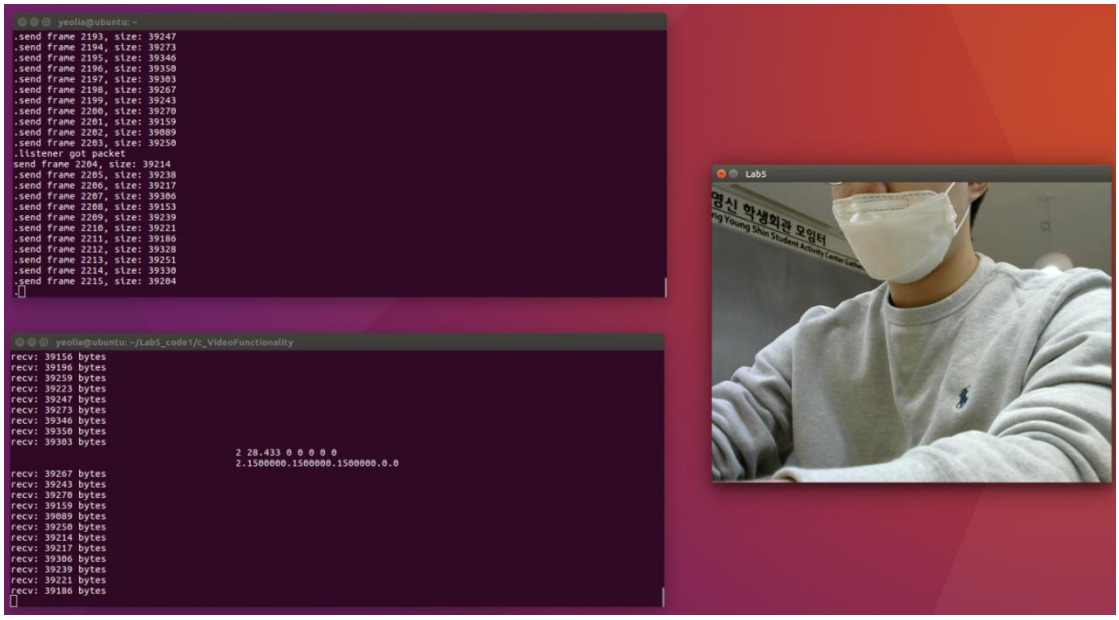
- Test 1

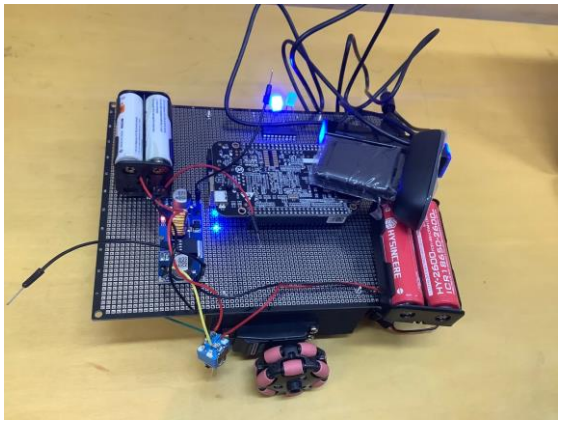
먼저 비디오 width를 640, height를 480, pixelformat을 MJPG로 세팅하였다.

```
root@beaglebone:~# v4l2-ctl --set-fmt-video=width=640,height=480,pixelformat=1
root@beaglebone:~# v4l2-ctl -V
Format Video Capture:
        Width/Height  : 640/480
        Pixel Format  : 'MJPG'
        Field         : None
        Bytes per Line: 0
        Size Image    : 341333
        Colorspace    : SRGB
```

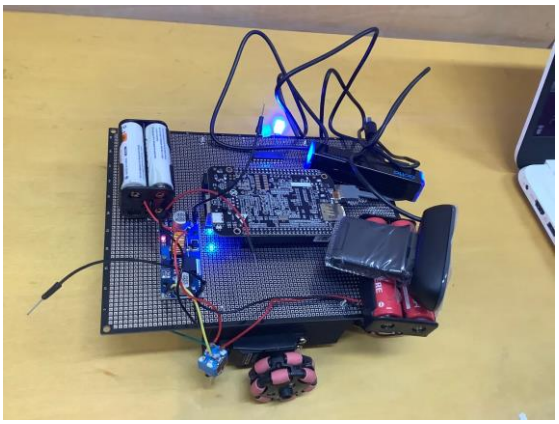
다음으로 컴파일된 실행파일을 실행했더니 프로그램이 잘 실행되었다.







<Q를 눌렀을 때>



<E를 눌렀을 때>

Control_Camera 실행파일에서 웹캠에서 얻은 화면을 Control_Viewer을 통해 볼 수 있었다. Control_Camera를 실행한 커맨드 창에서 "send frame (number), size: (number)"가 보여지고 Control_Viewer를 실행한 커맨드 창에서 "recv: (number) bytes"가 보여지고, 화면이 잘 출력되었기 때문에 두 실행파일 간 카메라 화면 정보를 보내고 받는 통신이 잘 이루어 진 것을 알 수 있다.

또한, Control_Viewer 실행파일에서 계산된 LED, 바퀴 동작 정보를 Control_Camera에서 받아 실제로 동작됨을 확인하였다. Control_Camera를 실행한 커맨드 창에서 "listener got packet"가 보여지고 Control_Viewer를 실행한 커맨드 창에서 바퀴와 LED 동작 정보가 보여지고, 키의 인풋대로 잘 동작했기 때문에 두 실행파일 간 바퀴와 LED 동작 정보를 보내고 받는 통신이 잘 이루어 진 것을 알 수 있다. 바퀴의 동작은 사진으로 보기 어려워 LED의 사진만 첨부하였다.

다음으로 로봇의 LED, 바퀴 값이 실제로 잘 들어가고 있는지 확인하기 위해 Control_Viewer에서 "recv: (number) bytes"을 지우고 보내지는 값을 확인하였다. 키보드의 키를 한번 누를 때마다 2줄이 출력되는데, 첫번째 줄은 "(command ID) (time) (ivx * 10) (ivy * 10) (iw * 1000) (LL) (RL)" 를 나타내고, 두번째 줄은 "(command ID) (P9-22 duty) (P8-19 duty) (P9-14 duty) (LL) (RL)" 를 나타낸다. Lab3과 비교하여 똑같이 LED가 켜지는 여부 값과 바퀴의 duty 값이 들어가고 동작했기 때문에 LED의 켜짐 여부와 바퀴의 속도와 방향이 의도한대로 설계됨을 확인하였다. 해당 키의 순서는 "Q-Q-E-E-W-S-A-S-D-S-X-S-Z-S-C-S-F12"이다.

```
yeolia@ubuntu:~/Lab5_code1/c_VideoFunctionality$ ./Control_Viewer
start capturing
                                        1 11.510 0 0 0 1 0
                                        1.1500000.1500000.1500000.1.0
                                        2 14.905 0 0 0 0 0
                                        2.1500000.1500000.1500000.0.0
                                        3 17.315 0 0 0 0 1
                                        3.1500000.1500000.1500000.0.1
                                        4 19.308 0 0 0 0 0
                                        4.1500000.1500000.1500000.0.0
                                        5 23.448 10 0 0 0 0
                                        5.2173100.1326900.1500000.0.0
                                        6 26.404 0 0 0 0 0
                                        6.1500000.1500000.1500000.0.0
                                        7 29.385 0 10 0 0 0
                                        7.901000.1401000.2201000.0.0
                                        8 32.375 0 0 0 0 0
                                        8.1500000.1500000.1500000.0.0
                                        9 34.415 0 -10 0 0 0
                                        9.2099000.2099000.799000.0.0
                                        10 36.990 0 0 0 0 0
                                        10.1500000.1500000.1500000.0.0
                                        11 38.794 -10 0 0 0 0
                                        11.826900.2173100.1500000.0.0
                                        12 42.703 0 0 0 0 0
                                        12.1500000.1500000.1500000.0.0
                                        13 46.790 0 0 1000 0 0
                                        13.604000.1104000.604000.0.0
                                        14 49.815 0 0 0 0 0
                                        14.1500000.1500000.1500000.0.0
                                        15 51.659 0 0 -1000 0 0
                                        15.2396000.2396000.2396000.0.0
                                        16 54.262 0 0 0 0 0
                                        16.1500000.1500000.1500000.0.0
pressed: F12 (Quit)
```

- Test 2

다음으로 비디오 width를 320, height를 240, pixelformat을 MJPG로 세팅하고 프로그램을 실행하였다. 마찬가지로 잘 실행됨을 확인하였다.