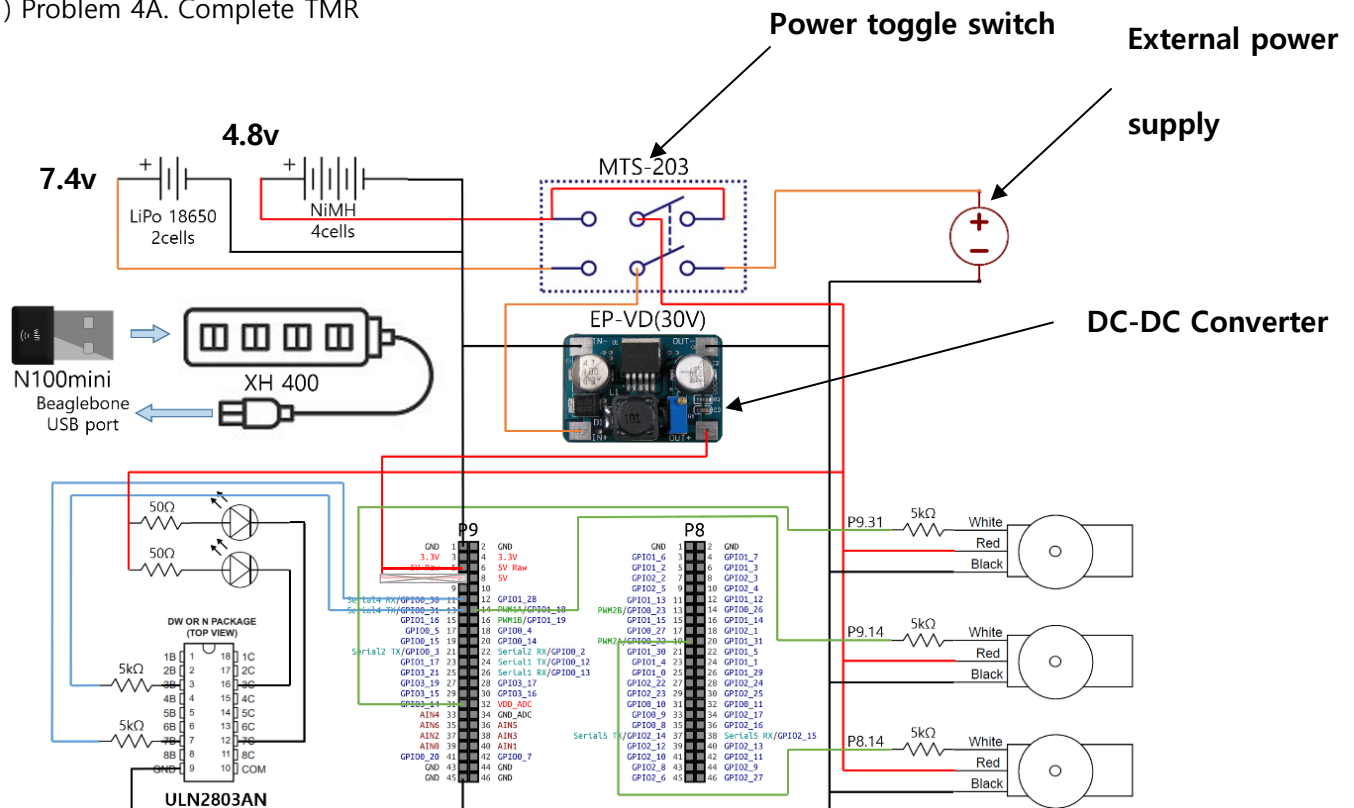


## 1. Purpose

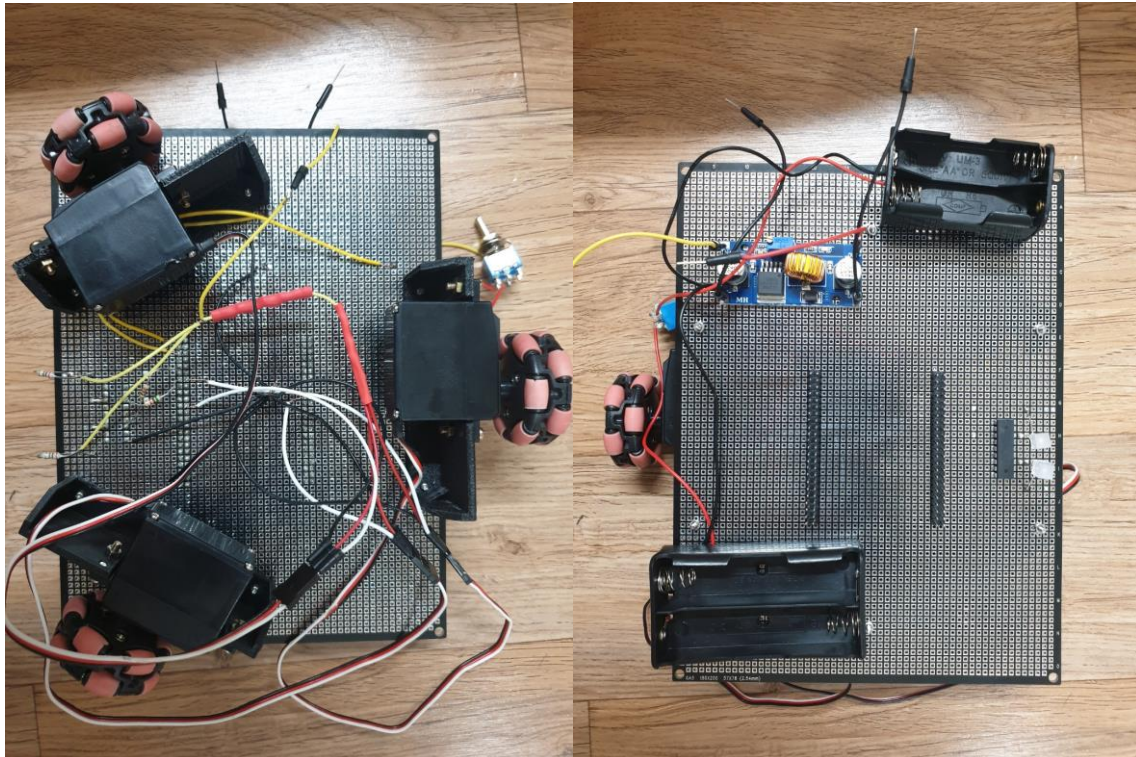
The purpose of this lab is to setup WiFi device driver with security, and control battery-powered threewheeled mobile robot using remote keyboard commander on PC via WiFi.

## 2. Experiment Procedure

## 1) Problem 4A. Complete TMR



위 회로도를 기반으로 회로를 작성하였다. 다만, 모터의 control pin이 위 회로와 다르게 P9\_14, P9\_22, P8\_19 pin을 사용하였다. 또한, 다이오드에 연결되는 저항도 50옴이 아니라 200옴 저항을 사용하였다. External power supply의 경우에는 이번 실험에서는 사용하지 않아, 회로에 추가하지는 않았다. EP-VD의 경우, 초기 out핀의 출력이 7V 정도로 나왔는데, 이를 5V정도로 바꾸어서 비글본에 출력이 정상적으로 공급될 수 있도록 하였다.



2) Problem 4B. Problem 4B. Setup WiFi device driver with security

먼저 WiFi 모듈을 확인하였다.

```
yeolia@ubuntu: ~
lsusb
Bus 001 Device 003: ID 0bda:8176 Realtek Semiconductor Corp. RTL8188CUS 802.11n WLAN Adapter
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
root@beaglebone:~#

jungwungpark@beaglebone:~$ lsmod
Module              Size  Used by
g_multi              50407  2
libcomposite         15028  1 g_multi
omap_rng             4062   0
nfsd                 187513  2
mt7601Usta          458758  0
```

현재 Device 003에 WLAN Adapter가 인식됨을 확인할 수 있었다.

그 후, IP를 확인하였다

```
eth0 Link encap:Ethernet HWaddr 0c:b2:b7:cb:9e:6b
      inet addr:192.168.0.17 Bcast:192.168.0.255 Mask:255.255.255.0
      inet6 addr: fe80::eb2:b7ff:fe6b:9e6b/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:1089 errors:0 dropped:0 overruns:0 frame:0
      TX packets:604 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:189155 (184.7 KiB) TX bytes:54011 (52.7 KiB)
      Interrupt:40
```

```

lo          Link encap:Local Loopback
            inet addr:127.0.0.1  Mask:255.0.0.0
            inet6 addr: ::1/128  Scope:Host
            UP LOOPBACK RUNNING  MTU:65536  Metric:1
            RX packets:1 errors:0 dropped:0 overruns:0 frame:0
            TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:112 (112.0 B)  TX bytes:112 (112.0 B)

usb0       Link encap:Ethernet  HWaddr f0:45:da:81:38:50
            inet addr:192.168.7.2  Bcast:192.168.7.3  Mask:255.255.255.252
            inet6 addr: fe80::f245:daff:fe81:3850/64  Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:88 errors:0 dropped:0 overruns:0 frame:0
            TX packets:34 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:13948 (13.6 KiB)  TX bytes:9320 (9.1 KiB)

wlan0     Link encap:Ethernet  HWaddr 88:36:6c:fe:57:59
            BROADCAST MULTICAST  MTU:1500  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

```

아직 wlan0에는 ip가 할당되지 않았음을 확인하였다.

WiFi 툴을 설치하였다.

```

jungwungpark@beaglebone:~$ sudo apt-get install wireless-tools
[sudo] password for jungwungpark:
Sorry, try again.
[sudo] password for jungwungpark:
Sorry, try again.
[sudo] password for jungwungpark:
Reading package lists... Done
Building dependency tree
Reading state information... Done
wireless-tools is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
jungwungpark@beaglebone:~$ sudo apt-get install wpasupplicant
Reading package lists... Done
Building dependency tree
Reading state information... Done
wpasupplicant is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
jungwungpark@beaglebone:~$

```

wireless-tools, wpasupplicant 모두 최신 상태로 유지하였다.

WiFi reset service를 이용하였다.

```

root@beaglebone:~# ntpdate -b -s -u pool.ntp.org
root@beaglebone:~# apt-get update && apt-get install git
Hit http://archive.debian.org wheezy Release.gpg
Hit http://archive.debian.org wheezy Release
Hit http://archive.debian.org wheezy/main armhf Packages
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@beaglebone:~# git clone https://github.com/adafruit/wifi-reset.git
Cloning into 'wifi-reset'...
remote: Enumerating objects: 16, done.
remote: Total 16 (delta 0), reused 0 (delta 0), pack-reused 16
Unpacking objects: 100% (16/16), done.
root@beaglebone:~# cd wifi-reset/
root@beaglebone:~/wifi-reset# chmod +x install.sh
root@beaglebone:~/wifi-reset# ./install.sh
Installing wifi reset service to /opt/wifi-reset.
Installing systemd service to run at boot.
Enabling systemd service.
ln -s '/lib/systemd/system/wifi-reset.service' '/etc/systemd/system/multi-user.target.wants/wifi-reset.servi
ce'
root@beaglebone:~/wifi-reset# sudo reboot

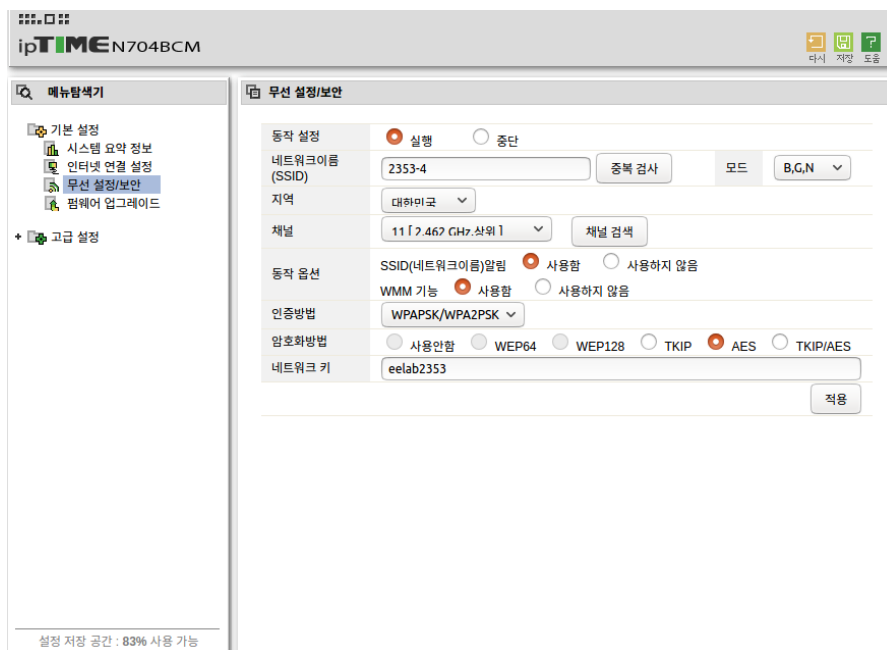
```

ntpdate를 통하여 시간 동기화를 하였고, git을 통하여 wifi-reset 모듈을 불러왔다. 그 후, install.sh 스크립트 파일을 실행하였다.

WiFi network를 설정하였다.

```
yeolia@ubuntu: ~  
wpa_passphrase 2353-4 eelab2353  
network={  
    ssid="2353-4"  
    #psk="eelab2353"  
    psk=3a7faac0420c96d97142e75c231c3e39bde3998286445be6c87c7eddc8d24f50  
}  
root@beaglebone:~#
```

ssid는 2353-4, psk는 eelab2353으로 설정하였다. WiFi network 설정과정에서 iptime 공유기 설정도 일부분 수정하였다.



iptime 공유기 SSID는 2353-4로, 인증방법은 WPAPSK - AES 및 네트워크 키는 eelab2353으로 위 wifi network 설정과 동일하게 맞추었다.

```
#iface wlan0 inet dhcp  
#    wpa-ssid "ssid"  
#    wpa-psk "password"  
  
# Ethernet/RNDIS gadget (g_ether)  
# Used by: /opt/scripts/boot/autoconfigure_usb0.sh  
iface usb0 inet static  
    address 192.168.7.2  
    netmask 255.255.255.252  
    network 192.168.7.0  
    gateway 192.168.7.1  
  
##### Appended for wlan0 WiFi  
auto wlan0  
iface wlan0 inet dhcp  
    wpa-ssid iptime_jwpark_N604R  
    wpa-psk 258822e4b796ff72a9263b251bc452e24e16381c2aa01502fb93a2138b49f902  
    wpa-driver wext  
    wpa-ap-scan 1  
    wpa-protocol RSN  
    wpa-pairwise CCMP  
    wpa-group CCMP  
    wpa-key-mgmt WPA-PSK  
-- INSERT --  
42,25 Bot
```

```
ap_scan = 1
ctrl_interface=DIR=/var/run/wpa_supplicant
network={
    ssid="iptime_jwpark_N604R"
    scan_ssid=0
    psk=258022e4b796ff72a9263b251bc452e24e16381c2aa01502fb93a2138b49f902
    key_mgmt=WPA-PSK
    proto=RSN
    pairwise=CCMP
    group=CCMP
}
```

vi를 통하여 /etc/network/interfaces와 /etc/wpa\_supplicant.conf 파일도 수정하였다. 위 캡처 화면에서는 ssid와 psk가 2353-4, 3a7~~과 다르지만 실제 실험과정에서는 위 두 parameter를 모두 수정하였다.

WiFi 드라이버를 직접 조작하고 WiFi를 확인하였다.

```
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/wlan0/88:36:6c:fe:57:59
Sending on   LPF/wlan0/88:36:6c:fe:57:59
Sending on   Socket/fallback
root@beaglebone:~# sudo ifup wlan0
ioctl[SIOCSIWAP]: Operation not permitted
ioctl[SIOCSIWENCODEXT]: Invalid argument
ioctl[SIOCSIWENCODEXT]: Invalid argument
Internet Systems Consortium DHCP Client 4.2.2
Copyright 2004-2011 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/wlan0/88:36:6c:fe:57:59
Sending on   LPF/wlan0/88:36:6c:fe:57:59
Sending on   Socket/fallback
DHCPDISCOVER on wlan0 to 255.255.255.255 port 67 interval 8
DHCPDISCOVER on wlan0 to 255.255.255.255 port 67 interval 14
DHCPPREREQUEST on wlan0 to 255.255.255.255 port 67
DHCPOFFER from 192.168.0.1
DHCPACK from 192.168.0.1
bound to 192.168.0.20 -- renewal in 2715 seconds.
root@beaglebone:~#
```

Ifup wlan0를 통하여 wlan0가 DHCP로부터 로컬 ip주소인 192.168.0.20을 할당받았다.

```
yeolia@ubuntu: ~
ifconfig
eth0      Link encap:Ethernet  HWaddr 0c:b2:b7:cb:9e:6b
          inet addr:192.168.0.17  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::eb2:b7ff:feb9:9e60/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1089 errors:0 dropped:0 overruns:0 frame:0
          TX packets:604 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:189155 (184.7 KiB)  TX bytes:54011 (52.7 KiB)
          Interrupt:40

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

usb0     Link encap:Ethernet  HWaddr 0c:b2:b7:cb:9e:60
          inet addr:192.168.7.2  Bcast:192.168.7.3  Mask:255.255.255.252
          inet6 addr: fe80::eb2:b7ff:feb9:9e60/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:51 errors:0 dropped:0 overruns:0 frame:0
          TX packets:51 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:8869 (8.6 KiB)  TX bytes:11965 (11.6 KiB)

wlan0    Link encap:Ethernet  HWaddr 88:36:6c:fe:57:59
          inet addr:192.168.0.20  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::8a36:6cfe:fefe:5759/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:63 errors:0 dropped:258 overruns:0 frame:0
          TX packets:49 errors:0 dropped:2 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:11066 (10.8 KiB)  TX bytes:12531 (12.2 KiB)
```

실제로 ifconfig를 해보면, wlan0에 ip주소가 192.168.0.20으로 할당되었음을 확인하였다.

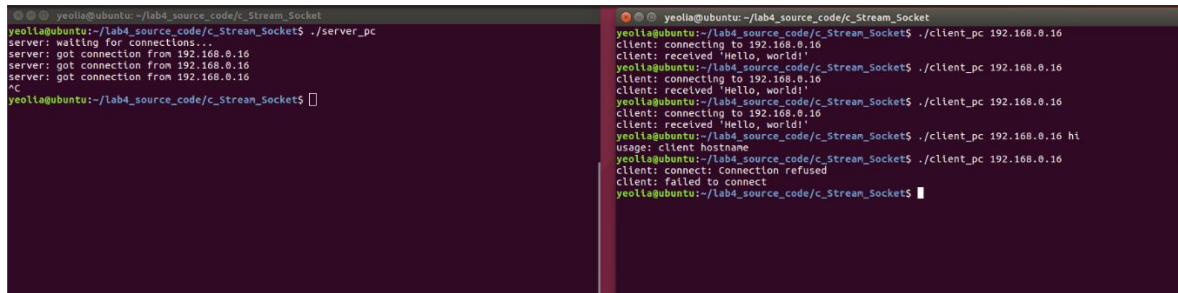
```
root@beaglebone:~# ls /var/run/wpa_supplicant
wlan0
```

wlan0가 정상적으로 잡히고 있다.



### 3) Problem 4C. Test Stream Socket Example.

다음으로 PC 커널에서 각각 server와 client를 실행하였다.

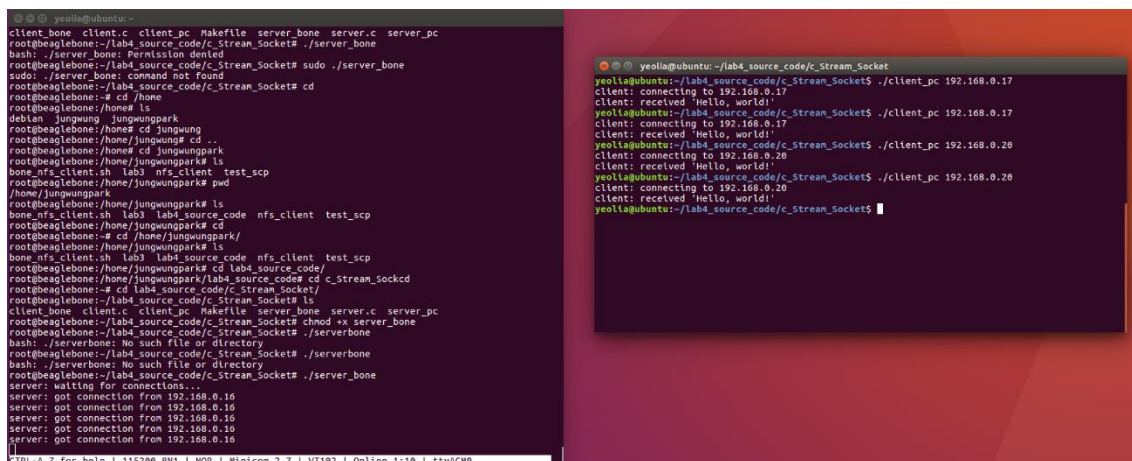


```
yeolla@ubuntu:~/lab4_source_code/c_Stream_Socket$ ./server_pc
server: waiting for connections...
server: got connection from 192.168.0.16
server: got connection from 192.168.0.16
server: got connection from 192.168.0.16
^C
yeolla@ubuntu:~/lab4_source_code/c_Stream_Socket$

yeolla@ubuntu:~/lab4_source_code/c_Stream_Socket$ ./client_pc 192.168.0.16
client: connecting to 192.168.0.16
client: received 'Hello, world!'
yeolla@ubuntu:~/lab4_source_code/c_Stream_Socket$ ./client_pc 192.168.0.16
client: connecting to 192.168.0.16
client: received 'Hello, world!'
yeolla@ubuntu:~/lab4_source_code/c_Stream_Socket$ ./client_pc 192.168.0.16 hi
usage: client hostname
yeolla@ubuntu:~/lab4_source_code/c_Stream_Socket$ ./client_pc 192.168.0.16
client: connect: Connection refused
client: failed to connect
yeolla@ubuntu:~/lab4_source_code/c_Stream_Socket$
```

PC에는 ip주소가 192.168.0.16으로 할당되었다. 따라서 PC에서는 ./server\_pc를 통하여 서버를 가동시키고 다른 터미널을 이용하여 ./client\_pc 192.168.0.16으로 접근하면 서버와 클라이언트가 정상적으로 작동하였다. 만약 서버가 꺼진 상태에서 클라이언트가 192.168.0.16에 접근하여 하면 여러 메시지가 나오는 것도 확인하였다.

다음으로 PC 커널에서 client를 실행하고 Bone 커널에서 server를 실행하였다.



```
client_bone client.c client_pc Makefile server_bone server.c server_pc
root@beaglebone:~/lab4_source_code/c_Stream_Socket$ ./server_bone
bash: ./server_bone: Permission denied
root@beaglebone:~/lab4_source_code/c_Stream_Socket$ sudo ./server_bone
sudo: ./server_bone: command not found
root@beaglebone:~/lab4_source_code/c_Stream_Socket$ cd
root@beaglebone:~$ cd /home
root@beaglebone:/home$ ls
debian jungwung jungwungpark
root@beaglebone:/home$ cd jungwung
root@beaglebone:/home/jungwung$ cd ..
root@beaglebone:/home$ cd jungwungpark
root@beaglebone:/home/jungwungpark$ ls
bone_nfs_client.sh lab3 nfs_client test_scp
root@beaglebone:/home/jungwungpark$ pwd
/home/jungwungpark
root@beaglebone:/home/jungwungpark$ ls
bone_nfs_client.sh lab3 lab4_source_code nfs_client test_scp
root@beaglebone:/home/jungwungpark$ cd
root@beaglebone:~$ cd /home/jungwungpark/
root@beaglebone:/home/jungwungpark$ ls
bone_nfs_client.sh lab3 lab4_source_code nfs_client test_scp
root@beaglebone:/home/jungwungpark$ cd lab4_source_code/
root@beaglebone:/home/jungwungpark/lab4_source_code$ cd c_Stream_Socket
root@beaglebone:/home/jungwungpark/lab4_source_code/c_Stream_Socket$ ls
client_bone client.c client_pc Makefile server_bone server.c server_pc
root@beaglebone:/home/jungwungpark/lab4_source_code/c_Stream_Socket$ chmod +x server_bone
root@beaglebone:/home/jungwungpark/lab4_source_code/c_Stream_Socket$ ./server_bone
bash: ./server_bone: No such file or directory
root@beaglebone:/home/jungwungpark/lab4_source_code/c_Stream_Socket$ ./server_bone
bash: ./server_bone: No such file or directory
root@beaglebone:/home/jungwungpark/lab4_source_code/c_Stream_Socket$ ./server_bone
server: waiting for connections...
server: got connection from 192.168.0.16
server: got connection from 192.168.0.16
server: got connection from 192.168.0.16
server: got connection from 192.168.0.16
server: got connection from 192.168.0.16
^C
root@beaglebone:/home/jungwungpark/lab4_source_code/c_Stream_Socket$

yeolla@ubuntu:~/lab4_source_code/c_Stream_Socket$ ./client_pc 192.168.0.17
client: connecting to 192.168.0.17
client: received 'Hello, world!'
yeolla@ubuntu:~/lab4_source_code/c_Stream_Socket$ ./client_pc 192.168.0.17
client: connecting to 192.168.0.17
client: received 'Hello, world!'
yeolla@ubuntu:~/lab4_source_code/c_Stream_Socket$ ./client_pc 192.168.0.20
client: connecting to 192.168.0.20
client: received 'Hello, world!'
yeolla@ubuntu:~/lab4_source_code/c_Stream_Socket$ ./client_pc 192.168.0.20
client: connecting to 192.168.0.20
client: received 'Hello, world!'
yeolla@ubuntu:~/lab4_source_code/c_Stream_Socket$
```

현재 beaglebone에는 2가지 방법으로 접근할 수 있다. 첫번째로 유선 랜을 통하여 접근하는 방식이고, 이때 ip주소는 192.168.0.17로 할당되었다. 두 번째 방식은 wifi 토글을 이용하여 접근하는 방식이다. 이 때, ip주소는 192.168.0.20으로 할당되었다.

두 가지 방식으로 모두 beaglebone에 연결을 시도하였다. beaglebone에서는 서버를 가동시키고, PC의 terminal cell에서 클라이언트를 사용하여 연결을 시도하였다. IP주소 192.168.0.17, 192.168.0.20 모두 서버와 클라이언트가 정상적으로 동작함을 확인하였다. 즉, 위 두가지 접근 방식 모두 성공적이었음을 확인하였다.

#### 4) Problem 4D. Test Datagram Socket (UDP) example.

##### (1) Implement listener\_pc and talker\_pc

PC 커널에서 각각 listener\_pc와 talker\_pc를 실행하였다. 이때, talker\_pc를 실행할 때, PC의 IP 주소(192.168.0.16)를 넣어주었다.

```
yeolia@ubuntu:~/lab4_source_code/d_Datagram_Socket$ ./listener_pc
listener: waiting to recvfrom...
listener: got packet from 192.168.0.16
listener: packet is 9 bytes long
listener: packet contains "hi,there!"
yeolia@ubuntu:~/lab4_source_code/d_Datagram_Socket$
```

<listener\_pc>

```
yeolia@ubuntu:~/lab4_source_code/d_Datagram_Socket$ ./talker_pc 192.168.0.16 "hi,there!"
talker: sent 9 bytes to 192.168.0.16
yeolia@ubuntu:~/lab4_source_code/d_Datagram_Socket$
```

<talker\_pc>

"hi,there!" 데이터가 listener한테 잘 전달되었다.

##### (2) Implement listener\_bone and talker\_pc

Beaglebone 커널 listener\_bone을 실행하고 PC 커널에서 talker\_pc를 실행하였다. 이때, talker\_pc를 실행할 때, bone의 Ethernet IP 주소(192.168.0.17)를 넣어주었다.

```
root@beaglebone:~/lab4_source_code/d_Datagram_Socket# ./listener_bone
listener: waiting to recvfrom...
listener: got packet from 192.168.0.16
listener: packet is 9 bytes long
listener: packet contains "hi,there!"
root@beaglebone:~/lab4_source_code/d_Datagram_Socket#
```

<listener\_bone>

```
yeolia@ubuntu:~/lab4_source_code/d_Datagram_Socket$ ./talker_pc 192.168.0.17 "hi,there!"
talker: sent 9 bytes to 192.168.0.17
yeolia@ubuntu:~/lab4_source_code/d_Datagram_Socket$
```

<talker\_pc>

"hi,there!" 데이터가 listener한테 잘 전달되었다.

이번에는 bone의 WiFi IP 주소(192.168.0.20)를 넣어주었다.

```
root@beaglebone:~/lab4_source_code/d_Datagram_Socket# ./listener_bone
listener: waiting to recvfrom...
listener: got packet from 192.168.0.16
listener: packet is 9 bytes long
listener: packet contains "hi,there!"
root@beaglebone:~/lab4_source_code/d_Datagram_Socket#
```

<listener\_bone>

```
talker: sent 9 bytes to 192.168.0.17
yeolia@ubuntu:~/lab4_source_code/d_Datagram_Socket$ ./talker_pc 192.168.0.20 "hi,there!"
talker: sent 9 bytes to 192.168.0.20
yeolia@ubuntu:~/lab4_source_code/d_Datagram_Socket$
```

<talker\_pc>

"hi,there!" 데이터가 listener한테 잘 전달되었다.

5) Problem 4E. Remote keyboard control via WiFi and UDP.

(1) Remote\_Commander\_PC.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <sys/time.h>

#define SERVERPORT "4950" // the port users will be connecting to

int main(int argc, char *argv[])
{
```



```

// initialize variables and error print
int sockfd;
struct addrinfo hints, *servinfo, *p1;
int rv;
int numbytes;
char send_str[100] = {0};
char buffer[100];
int cmd_num = 0;
struct timeval init_t;
struct timeval current_t;
double elapsed_t;

    int ivx = 0;
    int ivy = 0;
    int iw = 0;
    int led_one_on = 0;
    int led_two_on = 0;
    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 transform matrix
    int input;

    memset(&hints, 0, sizeof hints);
    hints.ai_family = AF_UNSPEC;
    hints.ai_socktype = SOCK_DGRAM;

// Get argument of destination IP (argv) of TMR
    if ((rv = getaddrinfo(argv[1], SERVERPORT, &hints, &servinfo)) != 0) {
        fprintf(stderr, "getaddrinfo: %s\n", gai_strerror(rv));
        return 1;
    }
}

```

```

// Init datagram socket.
for(p1 = servinfo; p1 != NULL; p1 = p1->ai_next) {
    if ((sockfd = socket(p1->ai_family, p1->ai_socktype,
                        p1->ai_protocol)) == -1) {
        perror("talker: socket");
        continue;
    }
    break;
}

if (p1 == NULL) {
    fprintf(stderr, "talker: failed to create socket\n");
    return 2;
}

printf("Remote_Commander_PC starts...now!\n");
printf("Key input menu (without enter)\n");
printf("-----+\n");
printf("|  Q: LL  |  W: +vx  |  E: RL  |\n");
printf("|-----+-----+-----|\n");
printf("|  A: -vy  |  S: Stop  |  D: +vy  |\n");
printf("|-----+-----+-----|\n");
printf("|  Z: +w   |  X: -vx   |  C: -w   |\n");
printf("-----+\n");
printf("Speed up/down with multiple key strokes\n");
printf("'T' key terminates.\n");

gettimeofday(&init_t, NULL);

while(1)
// Loop start
{
    // Add your own script written in Lab 3 (Loop A. ~. C. in Lecture note page. 20.)
    input = getch();
    if (input == 't')
        goto done;
    switch (input){
        case 'a':

```

```
        ivy += 10;
        break;
case 'd':
    ivy -= 10;
    break;
case 'w':
    ivx += 10;
    break;
case 'x':
    ivx -= 10;
    break;
case 's':
    ivx = 0; ivy = 0; iw = 0;
    break;
case 'z':
    iw += 1000;
    break;
case 'c':
    iw -= 1000;
    break;
case 'q':
    if (led_one_on)
    {
        led_one_on = 0;
    }
    else
    {
        led_one_on = 1;
    }

    break;
case 'e':
    if (led_two_on)
    {
        led_two_on = 0;
    }
    else
    {
        led_two_on = 1;
    }
}
```

```

        }
        break;
    }

    vx = 0.02 * (float)ivx;
    vy = 0.02 * (float)ivy;
    wr = 0.01 * (float)iwr;

    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] < 1500000){
        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;
    }

```

```

gettimeofday(&current_t, NULL);
elapsed_t = (current_t.tv_sec - init_t.tv_sec) + (current_t.tv_usec - init_t.tv_usec)/1000000.0;

```

// Please refer to attached code 'talker.c' for writing your own script (Loop D. ~. E. in Lecture note page. 20.)

```

sprintf(buffer,"%d",cmd_num);
strcat(send_str, buffer);
strcat(send_str, " ");
sprintf(buffer,"%0.3f",elapsed_t);
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)led_one_on);
strcat(send_str, buffer);
strcat(send_str, " ");
sprintf(buffer,"%d",(int)led_two_on);
strcat(send_str, buffer);

if ((numbytes = sendto(sockfd, send_str, strlen(send_str), 0,
                        p1->ai_addr, p1->ai_addrlen)) == -1) {
    perror("talker: sendto");
    exit(1);
}

```



```

        // Print information: key and cmd.
        printf("talker: sent %d bytes to %s\n", numbytes, argv[1]);
        printf("key: %c. cmd: %s\n", input, send_str);

        cmd_num += 1;
        memset(send_str, 0, 100*sizeof(char));

        usleep(1000);
    }

done:
    freeaddrinfo(servinfo);
    close(sockfd);

    return 0;
}

```

## (2) Explanation of Remote\_Commander\_PC.c

먼저 socket type을 datagram 방식으로 지정하였다.

```

memset(&hints, 0, sizeof hints);
hints.ai_family = AF_UNSPEC;
hints.ai_socktype = SOCK_DGRAM;

```

다음으로 IP 주소 정보를 얻는다.

```

if ((rv = getaddrinfo(argv[1], SERVERPORT, &hints, &servinfo)) != 0) {
    fprintf(stderr, "getaddrinfo: %s\n", gai_strerror(rv));
    return 1;
}

```

얻은 정보를 가지고 소켓을 만든다.

```

for(p1 = servinfo; p1 != NULL; p1 = p1->ai_next) {
    if ((sockfd = socket(p1->ai_family, p1->ai_socktype,
                        p1->ai_protocol)) == -1) {
        perror("talker: socket");
        continue;
    }
    break;
}

```

```
}
```

While loop를 들어가기 전에 경과 시간을 print하기 위하여 먼저 초기 시간을 설정한다.

```
gettimeofday(&init_t, NULL);
```

While loop 안에서 입력이 들어오면, 입력에 따라서 속도와 회전속도를 이용해 계산한 duty와 불이 켜졌는지의 여부를 알아낸다.

```
input = getch();
if (input == 't')
    goto done;
switch (input){
    case 'a':
        ivy += 10;
        break;
    case 'd':
        ivy -= 10;
        break;
    case 'w':
        ivx += 10;
        break;
    case 'x':
        ivx -= 10;
        break;
    case 's':
        ivx = 0; ivy = 0; iw = 0;
        break;
    case 'z':
        iw += 1000;
        break;
    case 'c':
        iw -= 1000;
        break;
    case 'q':
        if (led_one_on)
        {
            led_one_on = 0;
        }
        else
        {
```

```

        led_one_on = 1;
    }

    break;

case 'e':
    if (led_two_on)
    {
        led_two_on = 0;
    }
    else
    {
        led_two_on = 1;
    }
    break;
}

vx = 0.02 * (float)ivx;
vy = 0.02 * (float)ivy;
wr = 0.01 * (float)iwr;

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;
    }
}

```

경과된 시간을 초기 시간에서 빼 구한다.

```

gettimeofday(&current_t, NULL);
elapsed_t = (current_t.tv_sec - init_t.tv_sec) + (current_t.tv_usec - init_t.tv_usec)/1000000.0;

```

위에서 구한 모든 정보를 합하여 문자열을 만든다. 만든 문자열은 "(command number) (elapsed time) (P9-22 wheel) (P8-19 wheel) (P9-14 wheel) (p9-13 LED) (P9-11 LED)"으로 이루어져 있다.

```

sprintf(buffer,"%d",cmd_num);
strcat(send_str, buffer);
strcat(send_str, " ");
sprintf(buffer,"%0.3f",elapsed_t);
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)led_one_on);
    strcat(send_str, buffer);
    strcat(send_str, " ");
    sprintf(buffer,"%d",(int)led_two_on);
    strcat(send_str, buffer);

```

문자열을 소켓을 통해 보낸다.

```

    if ((numbytes = sendto(sockfd, send_str, strlen(send_str), 0,
                          p1->ai_addr, p1->ai_addrlen)) == -1) {
        perror("talker: sendto");
        exit(1);
    }

```

'T'키를 누르면 while loop를 빠져나오게 되고 다음의 코드가 실행된다. 열어두었던 파일을 닫고 메모리를 free해준다.

```

done:
    freeaddrinfo(servinfo);
    close(sockfd);

    return 0;

```

### (3) WiFi\_Control\_TMR.c

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <poll.h>
#include <termios.h>
#include "gpio_control_test.h"

```



```

static struct termios old_tio;
static struct termios new_tio;

#define MYPORT "4950" // the port users will be connecting to
#define MAXBUFLEN 100

// pre-define listener function
void *get_in_addr(struct sockaddr *sa)
{
    if (sa->sa_family == AF_INET) {
        return &(((struct sockaddr_in*)sa)->sin_addr);
    }
    return &(((struct sockaddr_in6*)sa)->sin6_addr);
}

// echo macro code
static void echo(char *str, char *file)
{
    int fd = open(file, O_WRONLY);
    if(fd < 0)
    {
        printf("%s:open error.\n", file);
        exit(-1);
    }
    write(fd, str, strlen(str));
    close(fd);
}

int main(int argc, char *argv[])
{
    /*0. Print Title
    1. Set control parameters – gain etc.
    2. Init PWM sysfs.
    3. Init GPIO_LED
    4. Open datagram socket and bind*/

    FILE *duty0, *duty1, *duty2, *run0, *run1, *run2;

    int sockfd;

```

```

struct addrinfo hints, *servinfo, *p;
int rv;
int numbytes;
struct sockaddr_storage their_addr;
char buf[MAXBUFLEN];
socklen_t addr_len;
char s[INET6_ADDRSTRLEN];

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);
}

memset(&hints, 0, sizeof hints);
hints.ai_family = AF_UNSPEC; // set to AF_INET to force IPv4
hints.ai_socktype = SOCK_DGRAM;
hints.ai_flags = AI_PASSIVE; // use my IP

if ((rv = getaddrinfo(NULL, MYPORT, &hints, &servinfo)) != 0) {
    fprintf(stderr, "getaddrinfo: %s\n", gai_strerror(rv));
    return 1;
}

// loop through all the results and bind to the first we can
for(p = servinfo; p != NULL; p = p->ai_next) {
    if ((sockfd = socket(p->ai_family, p->ai_socktype,
                        p->ai_protocol)) == -1) {
        perror("listener: socket");
        continue;
    }
    if (bind(sockfd, p->ai_addr, p->ai_addrlen) == -1) {
        close(sockfd);
        perror("listener: bind");
        continue;
    }
    break;
}

if (p == NULL) {
    fprintf(stderr, "listener: failed to bind socket\n");
    return 2;
}

freeaddrinfo(servinfo);

/* Print Key guide */
printf("+-----+ \n");
printf("|   Q: LL   |   W: +vx   |   E: RL   | \n");
printf("|------+-----+-----| \n");
```

```

printf("|  A: -vy  |  S: Stop  |  D: +vy  |Wn");
printf("|-----+-----+-----|Wn");
printf("|  Z: +w   |  X: -vx   |  C: -w   |Wn");
printf("+-----+-----+-----+Wn");

while(1)
{
    // Please refer to attached code 'listener.c' for writing your own script (Loop A. ~. B. in
Lecture note page. 21.)
    printf("listener: waiting to recvfrom...Wn");

    addr_len = sizeof their_addr;
    if ((numbytes = recvfrom(sockfd, buf, MAXBUFLEN-1 , 0,
        (struct sockaddr *)&their_addr, &addr_len)) == -1) {
        perror("recvfrom");
        exit(1);
    }

    printf("listener: got packet from %sWn",
        inet_ntop(their_addr.ss_family,
            get_in_addr((struct sockaddr *)&their_addr),
            s, sizeof s));
    printf("listener: packet is %d bytes longWn", numbytes);
    buf[numbytes] = '\0';
    printf("listener: packet contains W"%sW"Wn", buf);

    // use strtok() function to parse command to variables
    // use atoi() function to convert a character string to an integer value
    // Add your own script written in Lab 3 (Loop A. ~. C. in Lecture note page. 20.)
    token = strtok(buf, " ");
    while (token != NULL){
        token_num++;
        token = strtok(NULL, " ");
        if (token_num == 2){
            //printf("%dWn",atoi(token));
            fprintf(duty0, "%d",atoi(token));
            fprintf(run0, "%d", 1);
            fflush(duty0);
            fflush(run0);
        }
    }
}

```

```

    }
    else if (token_num == 3){
        //printf("%d\n",atoi(token));
        fprintf(duty1, "%d",atoi(token));
        fprintf(run1, "%d", 1);
        fflush(duty1);
        fflush(run1);
    }
    else if (token_num == 4){
        //printf("%d\n",atoi(token));
        fprintf(duty2, "%d",atoi(token));
        fprintf(run2, "%d", 1);
        fflush(duty2);
        fflush(run2);
    }
    else if (token_num == 5){
        //printf("%d\n",atoi(token));
        gpio_set_value( 31, atoi(token));
    }
    else if (token_num == 6){
        //printf("%d\n",atoi(token));
        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 socket*/
    close(sockfd);
/* 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 0;
}

```

#### (4) Explanation of WiFi\_Control\_TMR.c

먼저 2개의 LED을 제어하기 위해 P9-11, P9-13 pin을 export 했다.

```

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

```

다음으로 세 개의 바퀴(P9-22, P9-14, P8-19 pin)를 제어하기 위한 파일을 넣었다.

```

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);
    }

```

Socket type을 datagram 방식으로 지정하였다.

```

memset(&hints, 0, sizeof hints);
hints.ai_family = AF_UNSPEC; // set to AF_INET to force IPv4
hints.ai_socktype = SOCK_DGRAM;
hints.ai_flags = AI_PASSIVE;

```

IP address의 정보를 얻었다.

```

if ((rv = getaddrinfo(NULL, MYPORT, &hints, &servinfo)) != 0) {
    fprintf(stderr, "getaddrinfo: %s\n", gai_strerror(rv));
    return 1;
}

```

얻은 IP 정보를 이용해 소켓을 만들고 bind 하였다.

```

for(p = servinfo; p != NULL; p = p->ai_next) {
    if ((sockfd = socket(p->ai_family, p->ai_socktype,
        p->ai_protocol)) == -1) {
        perror("listener: socket");
        continue;
    }
    if (bind(sockfd, p->ai_addr, p->ai_addrlen) == -1) {
        close(sockfd);
        perror("listener: bind");
        continue;
    }
    break;
}

```

While loop안에 다음의 코드를 넣어 프로그램이 계속 데이터를 받을 수 있도록 했다. 받은 데이터는 buf 변수 안에 넣었다.

```

if ((numbytes = recvfrom(sockfd, buf, MAXBUFLEN-1, 0,
    (struct sockaddr *)&their_addr, &addr_len)) == -1) {
    perror("recvfrom");
    exit(1);
}

```

buf 변수안에 있는 문자를 잘라, 정보를 각각 얻었다. "(command number) (elapsed time) (P9-22 wheel) (P8-19 wheel) (P9-14 wheel) (p9-13 LED) (P9-11 LED)" 으로 이루어져 있으며 각각의 제어 파일에 값을 넣어주었다.

```
token = strtok(buf, " ");
while (token != NULL){
    token_num++;
    token = strtok(NULL, " ");
    if (token_num == 2){
        //printf("%d\n",atoi(token));
        fprintf(duty0, "%d",atoi(token));
        fprintf(run0, "%d", 1);
        fflush(duty0);
        fflush(run0);
    }
    else if (token_num == 3){
        //printf("%d\n",atoi(token));
        fprintf(duty1, "%d",atoi(token));
        fprintf(run1, "%d", 1);
        fflush(duty1);
        fflush(run1);
    }
    else if (token_num == 4){
        //printf("%d\n",atoi(token));
        fprintf(duty2, "%d",atoi(token));
        fprintf(run2, "%d", 1);
        fflush(duty2);
        fflush(run2);
    }
    else if (token_num == 5){
        //printf("%d\n",atoi(token));
        gpio_set_value( 31, atoi(token));
    }
    else if (token_num == 6){
        //printf("%d\n",atoi(token));
        gpio_set_value( 30, atoi(token));
    }
}
token_num = 0;
```

While loop를 벗어나려면 바퀴를 멈추고 열린 모든 파일을 닫고 정리될 것이다. 하지만, WiFi Control TMR 파일이 계속 실행할 수 있도록 만들어 주었기 때문에, while loop를 벗어나는 일이 없다. Remote Commander PC 파일을 끝내면서 같이 끝내고 싶다면 recvfrom으로 데이터를 받을 때, terminate키 정보를 추가로 얻어야 하고, 다음의 코드를 이용할 수 있을 것이다.

```
/* Stop PWM */
    fprintf(run0, "%d", 0);
    fprintf(run1, "%d", 0);
    fprintf(run2, "%d", 0);
/* Close socket*/
    close(sockfd);
/* 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);
```

#### (5) Implement

먼저 바퀴가 움직일 수 있도록, 전 실험에서 작성한 Acquire\_Triple\_PWMs.sh 파일을 실행해 사전 세팅을 해주었다.

```
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
```

다음으로 LED를 제어해주었다.

```

root@beaglebone:~/lab4_source_code/e_Remote_Commander# ./WiFi_Control_TMR
+-----+
| Q: LL | W: +vx | E: RL |
+-----+
| A: -vy | S: Stop | D: +vy |
+-----+
| Z: +w | X: -vx | C: -w |
+-----+
listener: waiting to recvfrom...
listener: got packet from 192.168.0.16
listener: packet is 35 bytes long
listener: packet contains "0 1.580 1500000 1500000 1500000 1 0"
listener: waiting to recvfrom...
listener: got packet from 192.168.0.16
listener: packet is 35 bytes long
listener: packet contains "1 2.140 1500000 1500000 1500000 1 1"
listener: waiting to recvfrom...

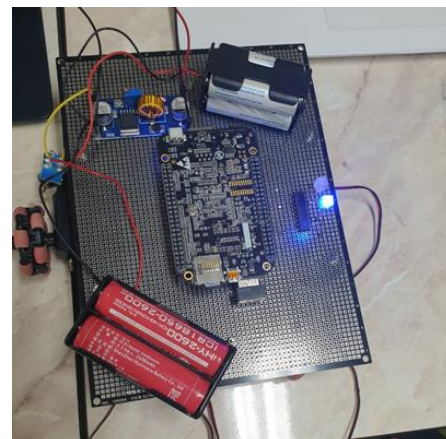
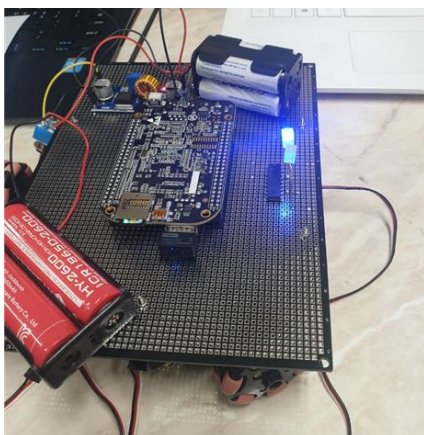
```

```

yeolia@ubuntu: ~/lab4_source_code/e_Remote_Commander
yeolia@ubuntu:~/lab4_source_code/e_Remote_Commander$ ./Remote_Commander_PC_pc 192.168.0.20
Remote_Commander_PC starts...now!
Key input menu (without enter)
+-----+
| Q: LL | W: +vx | E: RL |
+-----+
| A: -vy | S: Stop | D: +vy |
+-----+
| Z: +w | X: -vx | C: -w |
+-----+
Speed up/down with multiple key strokes
'T' key terminates.
talker: sent 35 bytes to 192.168.0.20
key: 'q'. cmd: 0 1.580 1500000 1500000 1500000 1 0
talker: sent 35 bytes to 192.168.0.20
key: 'e'. cmd: 1 2.140 1500000 1500000 1500000 1 1

```

커맨드 창에서 "cmd:"가 나타난 줄의 마지막 두 개의 숫자는 LED가 켜졌는지(1) LED가 꺼졌는지 (0)를 나타낸다. 값이 잘 나타난 것으로 보인다. 첫 번째 숫자는 P9-13 LED를 나타내고, 두 번째 숫자는 P9-11 LED를 나타낸다.



< LED가 모두 꺼진 상태에서 'q'를 눌렀을 때 > < LED가 모두 꺼진 상태에서 'e'를 눌렀을 때 >

'q' 키는 위쪽 LED를 켜졌다 꺼질 수 있도록 제어하고, 'e' 키는 아래쪽 LED를 켜졌다 꺼질 수 있도록 제어한다.

다음으로 바퀴를 제어했다.



```

jungwongpark@beaglebone:~$ ./e_Remote_Commander
listener: got packet from 192.168.0.16
listener: packet is 36 bytes long
listener: packet contains "43 71.333 901000 1401000 2201000 0 0"
listener: waiting to recvfrom...
listener: got packet from 192.168.0.16
listener: packet is 37 bytes long
listener: packet contains "44 72.988 1500000 1500000 1500000 0 0"
listener: waiting to recvfrom...
listener: got packet from 192.168.0.16
listener: packet is 35 bytes long
listener: packet contains "45 74.440 604000 1104000 604000 0 0"
listener: waiting to recvfrom...
listener: got packet from 192.168.0.16
listener: packet is 37 bytes long
listener: packet contains "46 75.980 1500000 1500000 1500000 0 0"
listener: waiting to recvfrom...
listener: got packet from 192.168.0.16
listener: packet is 37 bytes long
listener: packet contains "47 76.597 2396000 2396000 2396000 0 0"
listener: waiting to recvfrom...
listener: got packet from 192.168.0.16
listener: packet is 37 bytes long
listener: packet contains "48 77.679 1500000 1500000 1500000 0 0"
listener: waiting to recvfrom...
^C
root@beaglebone:~/lab4_source_code/e_Remote_Commander#

yeolia@ubuntu: ~/lab4_source_code/e_Remote_Commander
key: 'w'. cmd: 37 61.472 2173100 1326900 1500000 0 0
talker: sent 37 bytes to 192.168.0.20
key: 's'. cmd: 38 63.396 1500000 1500000 1500000 0 0
talker: sent 36 bytes to 192.168.0.20
key: 'x'. cmd: 39 64.084 826900 2173100 1500000 0 0
talker: sent 37 bytes to 192.168.0.20
key: 's'. cmd: 40 66.254 1500000 1500000 1500000 0 0
talker: sent 36 bytes to 192.168.0.20
key: 'd'. cmd: 41 68.321 2099000 2099000 799000 0 0
talker: sent 37 bytes to 192.168.0.20
key: 's'. cmd: 42 70.649 1500000 1500000 1500000 0 0
talker: sent 36 bytes to 192.168.0.20
key: 'a'. cmd: 43 71.333 901000 1401000 2201000 0 0
talker: sent 37 bytes to 192.168.0.20
key: 's'. cmd: 44 72.988 1500000 1500000 1500000 0 0
talker: sent 35 bytes to 192.168.0.20
key: 'z'. cmd: 45 74.440 604000 1104000 604000 0 0
talker: sent 37 bytes to 192.168.0.20
key: 's'. cmd: 46 75.980 1500000 1500000 1500000 0 0
talker: sent 37 bytes to 192.168.0.20
key: 'c'. cmd: 47 76.597 2396000 2396000 2396000 0 0
talker: sent 37 bytes to 192.168.0.20
key: 's'. cmd: 48 77.679 1500000 1500000 1500000 0 0
yeolia@ubuntu:~/lab4_source_code/e_Remote_Commander$

```

커맨드 창에서 "cmd: " 이후 3~5번째 숫자는 세 바퀴의 duty를 나타낸다. 첫 번째 숫자는 P9-22 바퀴를, 두 번째 숫자는 P8-19바퀴를, 세 번째 숫자는 P9-14 바퀴를 제어한다. Duty가 1500000일 때는 바퀴가 멈추었다(s키는 모든 바퀴의 duty를 1500000으로 만들어 주어 움직이지 않도록 해준다). 모든 키를 실행한 결과, 'wxad'는 상하좌우 이동, 'cz'는 회전을 하여 바퀴가 방향대로 잘 움직인 것을 확인했다.