

Table of Contents

[1 일차].....	2
[DSP opencv lib 확인].....	2
[이미지 GRAYSCALE 로 저장 basic_image.cpp].....	3
[이미지 dsp 터치화면에 출력 show_img.cpp].....	4
<빛의 3 원소, 색의 3 원소>.....	5
[컬러필터링 : color_filter.cpp].....	6
[DSP-MCU-LED : traffic_light.cpp MCU_SERIAL_LED_ON.c].....	11
[DSP-MCU-LED : green 은 rgb 값 이상하게 나오나? : 픽셀 위치 때문..].....	16
[DSP 보드 실습 유의사항].....	17
[DSP - 내부망 – PC(CCS) 기본 세팅 방법].....	17

[1 일차]

[DSP opencv lib 확인]

DSP 세팅 – micro sd 부팅 카드 +

AM572X (AM5728) Evaluation Board \$599

<http://www.ti.com/tool/tmdsevm572x?keyMatch=am5728%20evaluation&tisearch=Search-EN-Everything>

```
root@am57xx-evm:/home# cd root
root@am57xx-evm:~# ls
dongju      gihwahong   jangho      minsung     sanghoon   tibt
root@am57xx-evm:~# ls /usr/lib/ | grep opencv
libopencv_aruco.so
libopencv_aruco.so.3.1
libopencv_aruco.so.3.1.0
libopencv_bgsegm.so
libopencv_bgsegm.so.3.1
libopencv_bgsegm.so.3.1.0
libopencv_bioinspired.so
libopencv_bioinspired.so.3.1
libopencv_bioinspired.so.3.1.0
libopencv_calib3d.so
libopencv_calib3d.so.3.1
libopencv_calib3d.so.3.1.0
libopencv_ccalib.so
libopencv_ccalib.so.3.1
libopencv_ccalib.so.3.1.0
libopencv_core.so
libopencv_core.so.3.1
libopencv_core.so.3.1.0
libopencv_cvv.so
libopencv_cvv.so.3.1
libopencv_cvv.so.3.1.0
libopencv_datasets.so
libopencv_datasets.so.3.1
libopencv_datasets.so.3.1.0
libopencv_dnn.so
libopencv_dnn.so.3.1
libopencv_dnn.so.3.1.0
libopencv_dpm.so
libopencv_dpm.so.3.1
libopencv_dpm.so.3.1.0
libopencv_face.so
libopencv_face.so.3.1
libopencv_face.so.3.1.0
libopencv_features2d.so
libopencv_features2d.so.3.1
libopencv_features2d.so.3.1.0
libopencv_flann.so
libopencv_flann.so.3.1
libopencv_flann.so.3.1.0
libopencv_fuzzy.so
libopencv_fuzzy.so.3.1
libopencv_fuzzy.so.3.1.0
libopencv_highgui.so
libopencv_highgui.so.3.1
libopencv_highgui.so.3.1.0
libopencv_imgcodecs.so
libopencv_imgcodecs.so.3.1
libopencv_imgcodecs.so.3.1.0
libopencv_imgproc.so
libopencv_imgproc.so.3.1
libopencv_imgproc.so.3.1.0
libopencv_line_descriptor.so
libopencv_line_descriptor.so.3.1
libopencv_line_descriptor.so.3.1.0
libopencv_ml.so
```

이렇게 라이브러리 쳐서 잘 나와야함.

[이미지 GRAYSCALE 로 저장 basic_image.cpp]

test.jpeg 이미지를 흑백으로 this.jpg 에 저장한다

```
#include <opencv2/opencv.hpp> // 여기말고다른환경에서는 이렇게 참조하는 파일이다를것. 예)윈도우  
using namespace cv;  
using namespace std;  
  
int main(void)  
{  
    Mat srcImg = imread("test.jpeg", IMREAD_GRAYSCALE);  
  
    if(srcImg.empty())  
        return -1;  
    imwrite("this.jpg", srcImg);  
    return 0;  
}
```

컴파일시 아래 텍스트 치면 된다. 그냥 g++로 하면 안됨.

```
g++ -I/usr/local/include/opencv -I/usr/local/include/opencv2 -L/usr/local/lib show_img.cpp  
-lrt -lopencv_core -lopencv_imgproc -lopencv_features2d -lopencv_imgcodecs -lopencv_highgui
```

```
root@am57xx-evm:~# ls  
dongju  gihwahong  jangho  minsung  sanghoon  tibt  
root@am57xx-evm:~/gihwahong# ./a.out  
root@am57xx-evm:~/gihwahong# g++ -I/usr/local/include/opencv -I/usr/local/include/opencv2 -L/usr/local/lib basic_img.cpp -lrt -lopencv_core -lopencv_imgproc -lopencv_features2d -lopencv_imgcodecs -lopencv_highgui
```

컴파일 하고 a.out 실행하면,

같은 폴더에 test.jpeg 놓으면, 아래처럼 this.jpg 가 grayscale 로 생성된다.



[이미지 dsp 터치화면에 출력 show_img.cpp]

```
#include <opencv2/highgui/highgui.hpp>

using namespace cv;

int main(int argc, char **argv)
{
    Mat img= imread(argv[1], -1);
    if(img.empty())
        return -1;

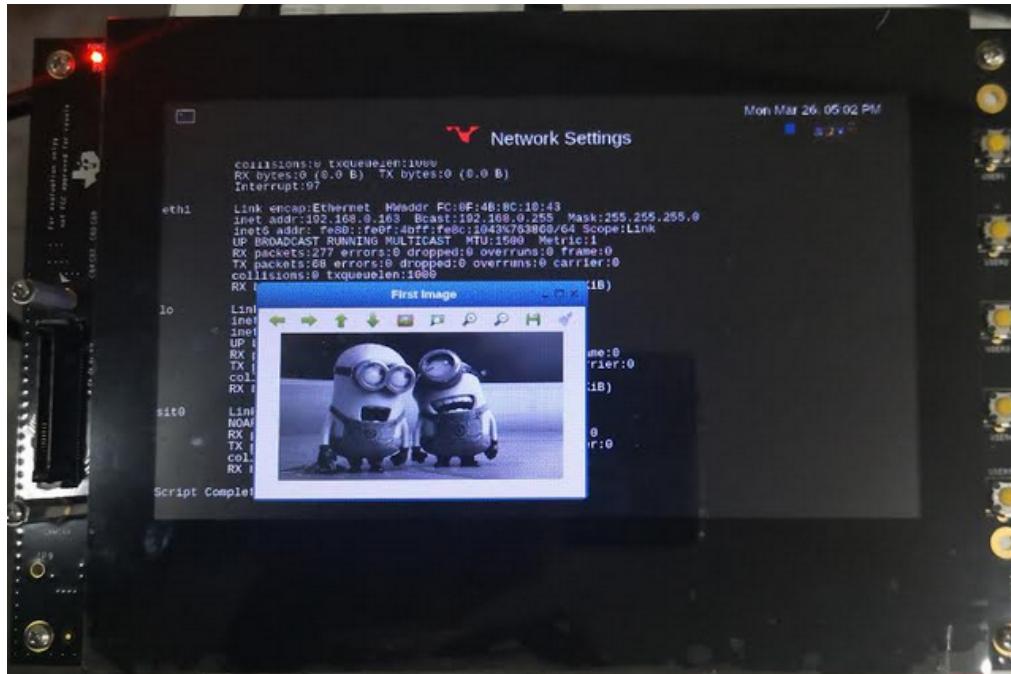
    namedWindow("First Image", cv::WINDOW_AUTOSIZE);
    imshow("First Image", img);

    waitKey(0);

    destroyWindow("First Image");

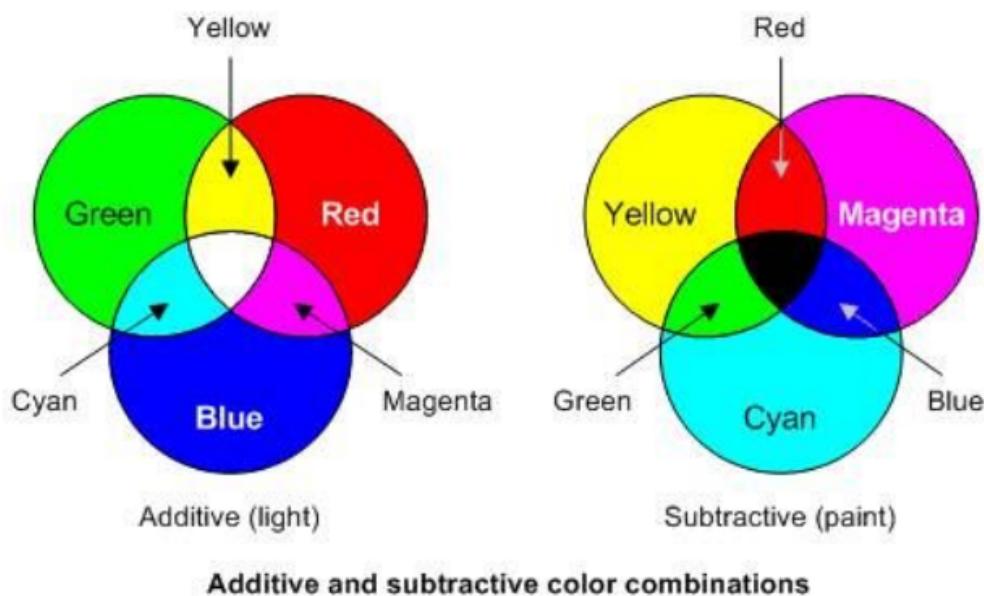
    return 0;
}
```

```
root@am57xx-evm:~/gihwahong# ./a.out this.jpg
init done
Using Wayland-EGL
wlpvrt: PWR Services Initialised+ -I/usr/local/include/opencv -I/usr/local/include
^[[A^[[A^[[B^[[C^[[C
root@am57xx-evm:~/gihwahong# ./a.out this
root@am57xx-evm:~/gihwahong# ./a.out this.jpg
init done
Using Wayland-EGL
wlpvrt: PWR Services Initialised
```



<빛의 3 원소, 색의 3 원소>

우리 눈은 빛(빨파녹 RGB)를 보지만,
카메라는 색(yellow, magenta, cyan)을 본다. 그래서 **변환**을 해 줘야함. (프린트 색상 생각해보자.)



[컬러필터링 : color_filter.cpp]

<코드>

```
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/opencv.hpp>
#include <iostream>

using namespace std;
using namespace cv;

int main(int argc, char **argv)
{
    int x, y, w, h;
    int rgb_threshold[4] = {128, 128, 128};

    Mat img = imread("sample.jpg", -1);
    Mat filter_img(img);

    h = img.rows;
    w = img.cols;

    if(img.empty())
        return -1;

    cout << "img.depth = " << img.depth() << ", "
        << "img.channels = " << img.channels() << endl;

    imshow("Second Image", img);

    cvtColor(img, img, COLOR_BGR2RGB);

    for(y = 0; y < h; y++)
    {
        for(x = 0; x < w; x++)
        {
            if(img.at<Vec3b>(y,x)[0] < rgb_threshold[0])
                filter_img.at<Vec3b>(y, x)[0] = 0;
            if(img.at<Vec3b>(y,x)[0] < rgb_threshold[1])
                filter_img.at<Vec3b>(y, x)[1] = 0;
            if(img.at<Vec3b>(y,x)[0] < rgb_threshold[2])
                filter_img.at<Vec3b>(y, x)[2] = 0;
        }
    }

    imshow("Filter Image", filter_img);

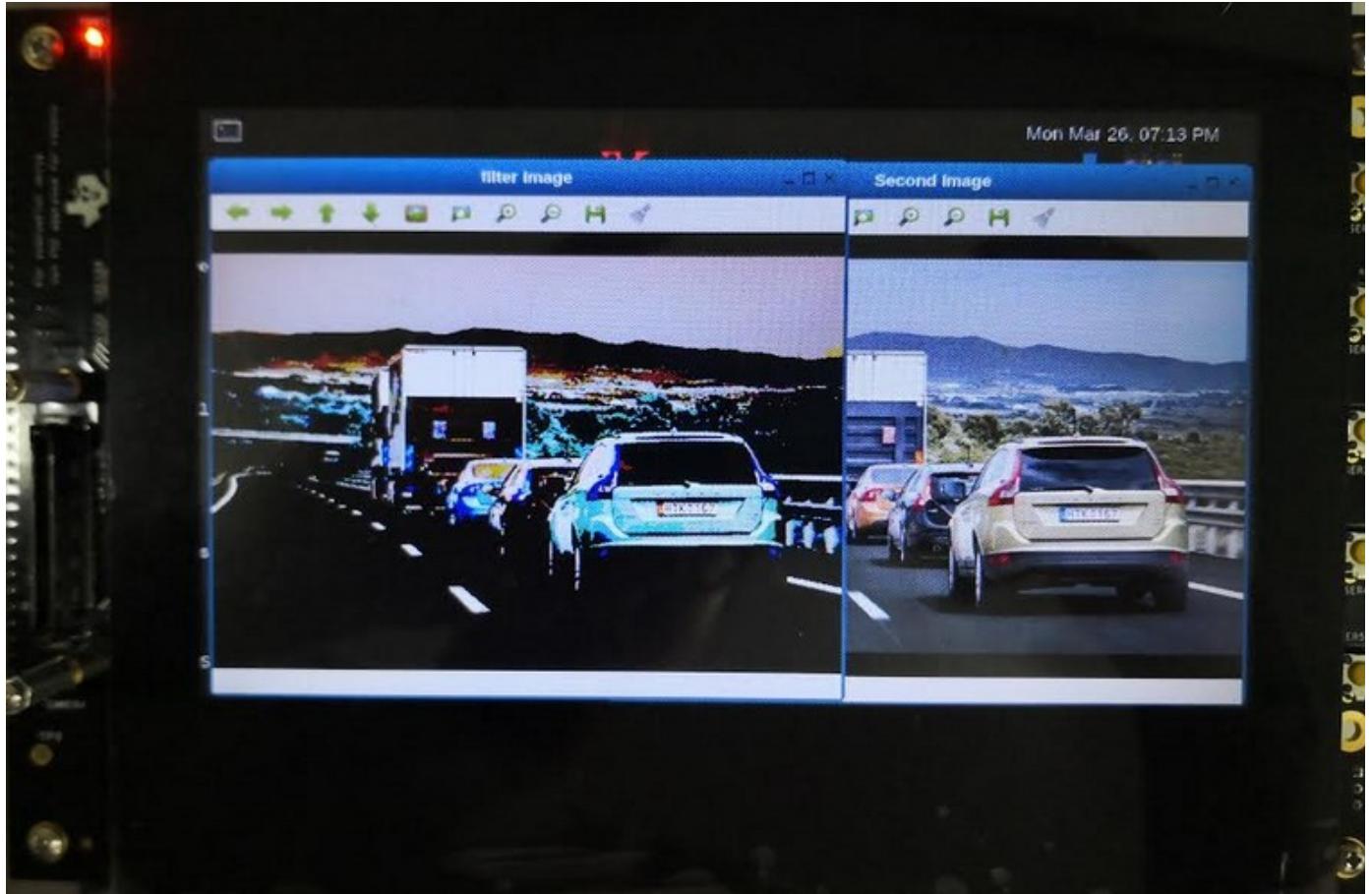
    waitKey(0);

    destroyWindow("Second Image");
    destroyWindow("Filter Image");

    return 0;
}
```

```
root@am57xx-evm:~/gihwahong# g++ -I/usr/local/include/opencv -I/usr/local/include/opencv2 -L/usr/local/lib color_filter.cpp -lrt -lopencv_core -lopencv_imgproc -lopencv_features2d -lopencv_imgcodecs -lopencv_highgui
g++ -I/usr/local/include/opencv -I/usr/local/include/opencv2 -L/usr/local/lib
color_filter.cpp -lrt -lopencv_core -lopencv_imgproc -lopencv_features2d -lopencv_imgcodecs
-lopencv_highgui
```

```
^[[A^Croot@am57xx-evm:~/gihwahong# ./a.out
img.depth = 0, img.channels = 3
init done
Using Wayland-EGL
wlprv: PVR Services Initialised
```



threshold 를 조정해서 128 로 했을때 지금 결과가 나온다.

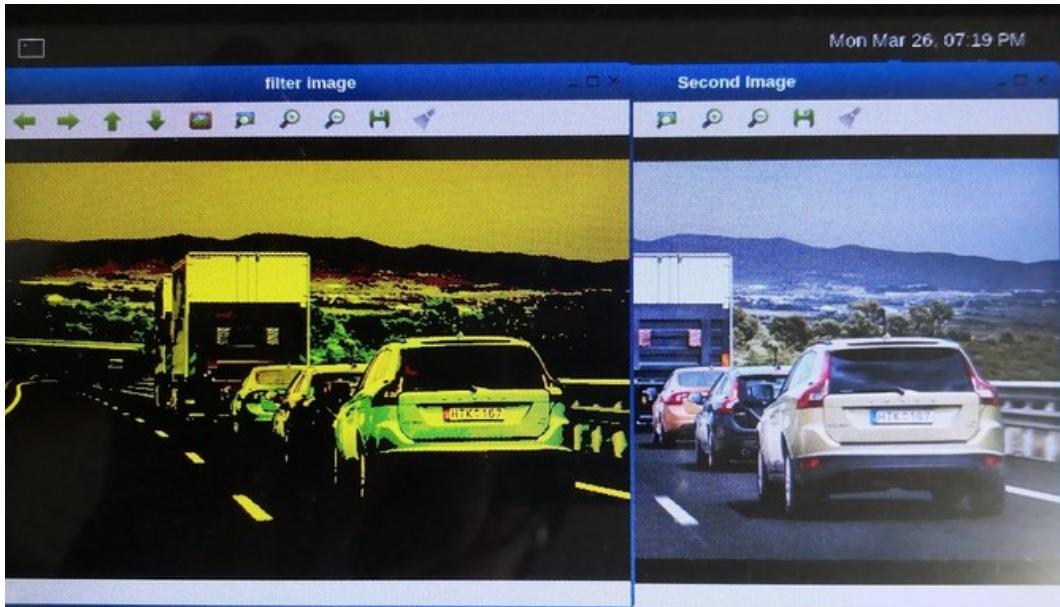
그럼 threshold 조정해서 얻어내는 이미지를 바꿀 수 있다. 이렇게 바꿔서 차선을 인식 할 수 있다.

→ threshold 변경 테스트 결과 아래

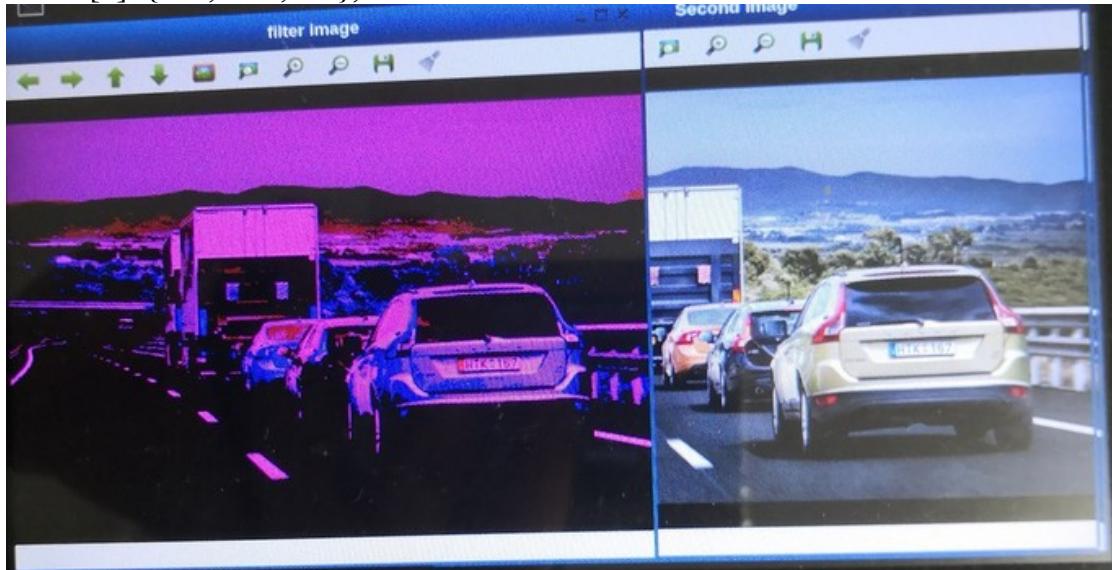
→ 테스트 해본 결과 :

** int rgb_threshold[4]={1024,128,128}; 순서대로 yellow, magenta, sian 값이 클수록 그 높은값으로 해당 색상을 필터링해내니까 해당 색상이 진하게 보인다.

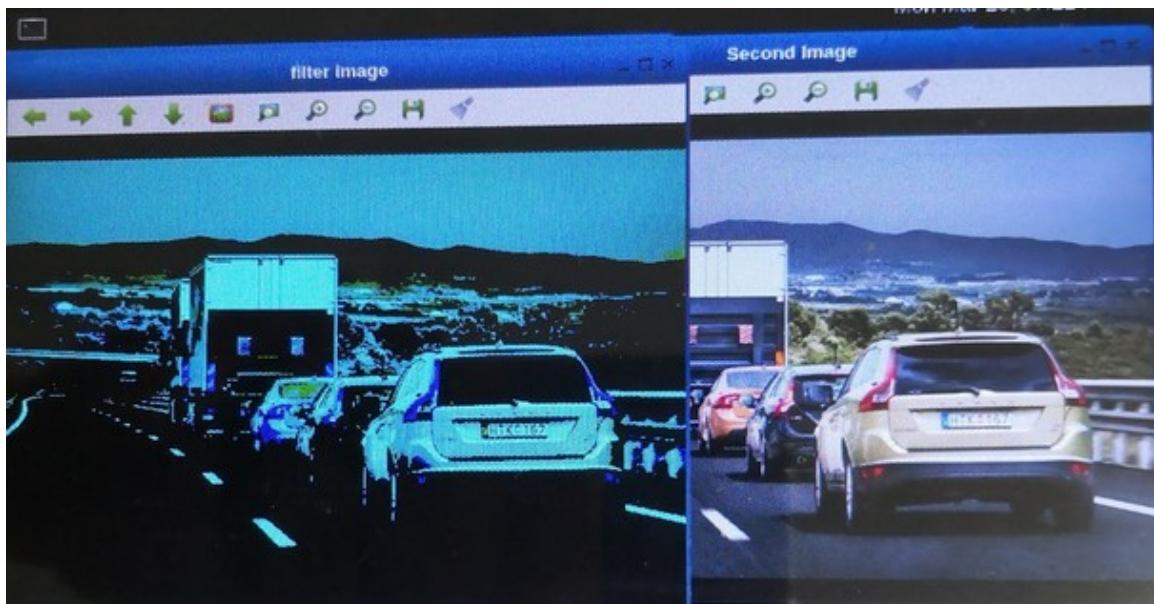
```
** int rgb_threshold[4]={1024,128,128};
```



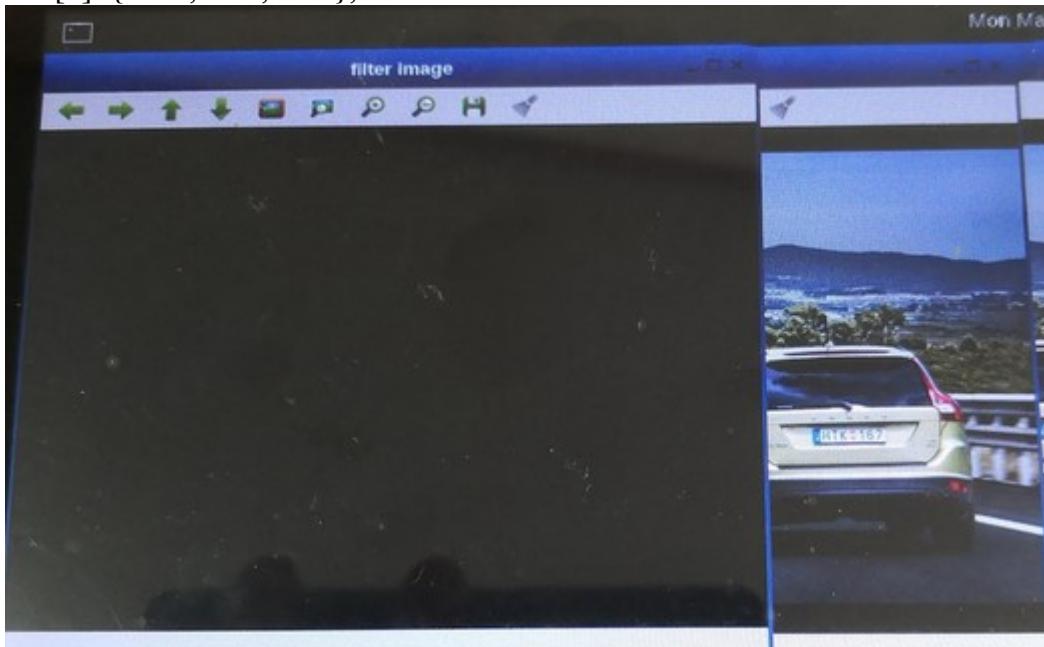
```
** int rgb_threshold[4]={128,1024,128};
```



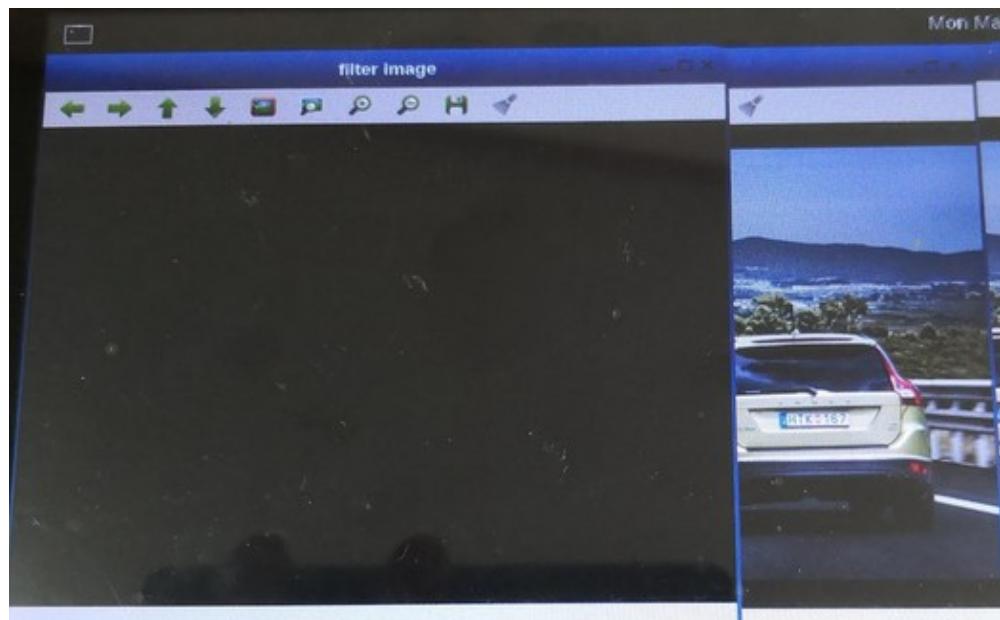
```
** int rgb_threshold[4]={128,128,1024};
```



```
** int rgb_threshold[4]={1024,1024,1024};
```



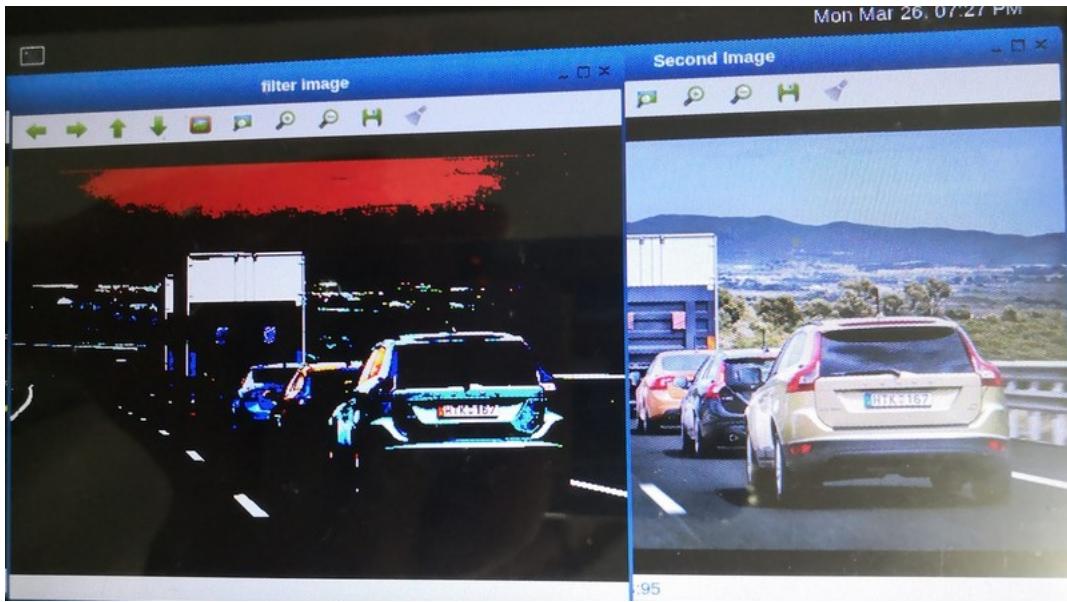
```
** int rgb_threshold[4]={256,256,256};
```



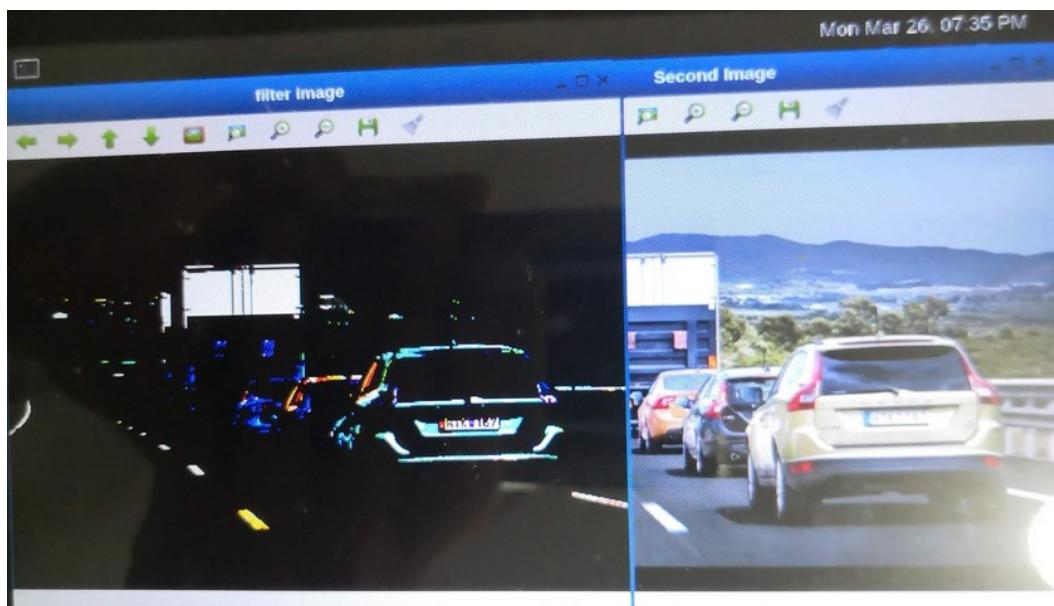
```
** int rgb_threshold[4]={150,150,150};
```



```
** int rgb_threshold[4]={200,200,200};
```



```
** int rgb_threshold[4]={230,230,230};
```



```
** int rgb_threshold[4]={1024,1024,230};
```



[DSP-MCU-LED : traffic_light.cpp MCU_SERIAL_LED_ON.c]

```
<mcu 코드>//깔끔하지않음.  
#include "HL_sys_common.h"  
#include "HL_system.h"  
#include "HL_sci.h"  
#include "HL_gio.h"  
#include <string.h>  
#include <stdio.h>  
  
#define UART sciREG1  
#define TSIZE 7  
  
char txt_buf[100];  
uint8 buf_len;  
uint32 tmp[6];  
char ins[TSIZE] = {'l', 'e', 'd', 'o', 'n', '\n', '\r'};  
  
int tb_len;  
  
void wait(uint32);  
  
void sciSendText(sciBASE_t * , uint8*, uint32);  
  
int main(void)  
{  
    gioInit();  
    wait(10000);  
    sciInit();  
    wait(10000);  
    gioSetDirection(gioPORTA, 0xffffffff);  
    gioSetBit(gioPORTA, 0, 1);  
    gioSetBit(gioPORTA, 1, 1);  
    gioSetBit(gioPORTA, 2, 1);  
  
    sprintf(txt_buf, "enter 1digit number.. \n\r\0");  
    buf_len = strlen(txt_buf);  
    sciSendText(sciREG1, (uint8 *)txt_buf, buf_len);  
  
    for(;;)  
    {  
        tmp[0] = sciReceiveByte(sciREG1);  
        if((tmp[0]>58)|| (tmp[0]<48))  
        {  
            sprintf(txt_buf, "number please.. \n\r\0");  
            buf_len = strlen(txt_buf);  
            sciSendText(sciREG1, (uint8 *)txt_buf, buf_len);  
  
        }  
        sciSendText(sciREG1, &tmp[0], TSIZE);  
  
        if(tmp[0]=='\n' || (tmp[0]<48))  
        ;  
        else  
        //            tmp[0] = tmp[0] - 48;  
        ;  
  
        switch(tmp[0])  
        {  
        case '0':  
            gioSetBit(gioPORTA, 0, 1);  
            gioSetBit(gioPORTA, 1, 1);  
            gioSetBit(gioPORTA, 2, 1);  
  
            gioSetBit(gioPORTA, 0, 0);  
            //gioSetBit(gioPORTA, 1, 0);  
            //gioSetBit(gioPORTA, 2, 0);
```

```

        break;
    case '1':
        gioSetBit(gioPORTA, 0, 1);
        gioSetBit(gioPORTA, 1, 1);
        gioSetBit(gioPORTA, 2, 1);
        //gioSetBit(gioPORTA, 0, 0);
        //gioSetBit(gioPORTA, 1, 0);
        gioSetBit(gioPORTA, 1, 0);
        break;
    case '2':
        gioSetBit(gioPORTA, 0, 1);
        gioSetBit(gioPORTA, 1, 1);
        gioSetBit(gioPORTA, 2, 1);
        //gioSetBit(gioPORTA, 0, 0);
        //gioSetBit(gioPORTA, 1, 0);
        gioSetBit(gioPORTA, 2, 0);
        break;
    }

    wait(33333);
}
return 0;
}

void wait(uint32 delay)
{
    int i;
    for(i=0; i<delay; i++)
    ;
}

void sciSendText(sciBASE_t* sci, uint8 * text, uint32 len)
{
    while(len--)
    {
        while((sci->FLR & 0x04)==4)
            ;
        sciSendByte(sci, *text++);
    }
}

```

<dsp 코드>

```

#include <opencv2/highgui/highgui.hpp>
#include <opencv2/opencv.hpp>
#include <iostream>

#include <sys/types.h>
#include <sys/poll.h>
#include <termios.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>
#include <signal.h>
#include <setjmp.h>
#include <fcntl.h>
#include <stdio.h>

#include "serial.h"

using namespace std;
using namespace cv;

extern char *dev0;
jmp_buf env;
int fd;

```

```

void call_exit(int signo)
{
    longjmp(env, 1);
}

int main(int argc, char **argv)
{
    int nr, fd;
    int x, y, w, h;
    char buf[32] = "";

    int ret;

    signal(SIGINT, call_exit);

    dev0 = "/dev/ttyACM0"; //launchpad는 이게 필요함. ACM0 ↔ ACM1 바꾸면서 해보자.
    fd = serial_config(dev0); // dsp usb dev0
    //fd = 1;    자기 모니터.DSP 아니고 PC로 LINUX TERMINAL에서 하는 경우 켜야한다.

    printf("0: Red, 1: Yellow, 2: Green Light\n");

    if(!!(ret = setjmp(env)))
    {
        for(;;)
        {
            Mat img;
            Mat filter_img;
            nr = read(0, buf, sizeof(buf));

            if(!(strncmp(buf, "0", 1)))
            {
                img = imread("red.png", -1);
                cvtColor(img, img, COLOR_BGR2RGB);
            }
            else if(!(strncmp(buf, "1", 1)))
            {
                img = imread("yellow.jpg", -1);
                cvtColor(img, img, COLOR_BGR2RGB);
            }
            else if(!(strncmp(buf, "2", 1)))
            {
                img = imread("green.jpg", -1);
                cvtColor(img, img, COLOR_BGR2RGB);
            }
            else
            {
                printf("Wrong Number\n");
                continue;
            }

            printf("I read %s", buf);

            printf("r = %d, g = %d, b = %d\n",
                   img.at<Vec3b>(200, 200)[0],
                   img.at<Vec3b>(200, 200)[1],
                   img.at<Vec3b>(100, 100)[2]);

            send_data(fd, buf, 1, 0);
            printf("\n");

#if 0
            red.png -> 219, 17, 29
            yellow.jpg -> 251, 246, 84
            green.jpg -> 145, 145
#endif

            memset(buf, 0x0, 32);
        }
    }
}

```

```

        }

    close_dev(fd);

#ifndef _WIN32
    Mat filter_img(img);

    for(y = 0; y < h; y++)
    {
        for(x = 0; x < w; x++)
        {
            // red
            if(img.at<Vec3b>(y, x)[0] < rgb_threshold[0])
                filter_img.at<Vec3b>(y, x)[0] = 0;
            // green
            if(img.at<Vec3b>(y, x)[1] < rgb_threshold[1])
                filter_img.at<Vec3b>(y, x)[1] = 0;
            // red + green
            if(img.at<Vec3b>(y, x)[2] < rgb_threshold[2])
                filter_img.at<Vec3b>(y, x)[2] = 0;
        }
    }

    imwrite("filtered.jpg", filter_img);
    imshow("Filter Image", filter_img);

    waitKey(0);

    destroyWindow("Second Image");
    destroyWindow("Filter Image");
#endif

    return 0;
}

```

컴파일 명령어 아래와 같다.

```

g++ -std=c++11 -lOpenCL -locl_util -L/usr/share/OpenCV/3rdparty/lib -lopencv_stitchingcv_superres
-lopencv_videostab -lopencv_aruco -lopencv_bgsegm -lopencv_bioinspired -lopencv_ccalib -lopencv_cvv
-lopencv_dnn -lopencv_dpm -lopencv_fuzzy -lopencv_line_descriptor -lopencv_optflow -lopencv_plot
-lopencv_reg -lopencv_saliency -lopencv_stereo -lopencv_structured_light -lopencv_rgbd
-lopencv_surface_matching -lopencv_tracking -lopencv_datasets -lopencv_text -lopencv_face
-lopencv_xfeatures2d -lopencv_shape -lopencv_video -lopencv_ximgproc -lopencv_calib3d -lopencv_features2d
-lopencv_flann -lopencv_xobjdetect -lopencv_objdetect -lopencv_ml -lopencv_xphoto -lopencv_highgui
-lopencv_videoio -lopencv_imgcodecs -lopencv_photo -lopencv_imgproc -lopencv_core -L$(pwd) -Wl,-rpath=$(pwd)
-g -larm_serial traffic_light.cpp

```

컴파일 및 실행 화면 아래와 같다. (경로에 libarmserial.so 및 serial.h 및 이미지(red.jpg, green.jpg, yellow.jpg)가 있다.)

```
d^H^Croot@am57xx-evm:~/gihwahong# g++ -std=c++11 -lOpenCL -locl_util -L/usr/share/OpenCV/3rdparty/lib -lopencv_stitching  
encv_superres -lopencv_videostab -lopencv_aruco -lopencv_bgsegm -lopencv_bioinspired -lopencv_ccalib -lopencv_cvv -lopen  
cv_dnn -lopencv_dpm -lopencv_fuzzy -lopencv_line_descriptor -lopencv_optflow -lopencv_plot -lopencv_reg -lopencv_salienc  
y -lopencv_stereo -lopencv_structured_light -lopencv_rgbd -lopencv_surface_matching -lopencv_tracking -lopencv_datasets  
-lopencv_text -lopencv_face -lopencv_xfeatures2d -lopencv_shape -lopencv_video -lopencv_ximgproc -lopencv_calib3d -lopen  
cv_features2d -lopencv_flann -lopencv_xobjdetect -lopencv_objdetect -lopencv_ml -lopencv_xphoto -lopencv_highgui -lopenc  
v_videoio -lopencv_imgcodecs -lopencv_photo -lopencv_imgproc -lopencv_core -L$(pwd) -Wl,-rpath=$(pwd) -g -larm_serial tr  
affic_light.cpp  
traffic_light.cpp: In function 'int main(int, char**)':  
traffic_light.cpp:40:8: warning: ISO C++ forbids converting a string constant to 'char*' [-Wwrite-strings]  
    dev0= "/dev/ttyACM0";  
    ^~~~~~  
root@am57xx-evm:~/gihwahong# ./a.out  
start  
Ready to start parsing!  
0: Red, 1: Yellow, 2: Green Light  
1  
I read 1  
r = 251, g = 246, b = 84  
  
0  
I read 0  
r = 219, g = 17, b = 29  
  
2  
I read 2  
r = 145, g = 145, b = 145
```

→ DSP 측 설명 : 1 치면, 1 이 DSP usb serial 통해 전송된다(ascii 로 숫자 1 값 송신)
mcu 에서 읽은 값 그대로 쏴 준다.
DSP에서 그걸 읽어서 “read 1” 이라고 표시 해 준다.
0 치면, 0 이 DSP usb serial 통해 전송된다(ascii 로 숫자 0 값 송신)
mcu 에서 읽은 값 그대로 쏴 준다.
DSP에서 그걸 읽어서 “read 0” 이라고 표시 해 준다.
2 치면, 2 이 DSP usb serial 통해 전송된다(ascii 로 숫자 2 값 송신)
mcu 에서 읽은 값 그대로 쏴 준다.
DSP에서 그걸 읽어서 “read 2” 이라고 표시 해 준다.

MCU 측 설명 : main for 문 루프: MCU 에서 UART 값 받아서, 그대로 sendtext 하고,
받은 한 비트(tmp[0]는 switch 문으로 ‘0’, ‘1’,’2’ 에 따라 led PORTA 0,1,2 각각 ON 한다.

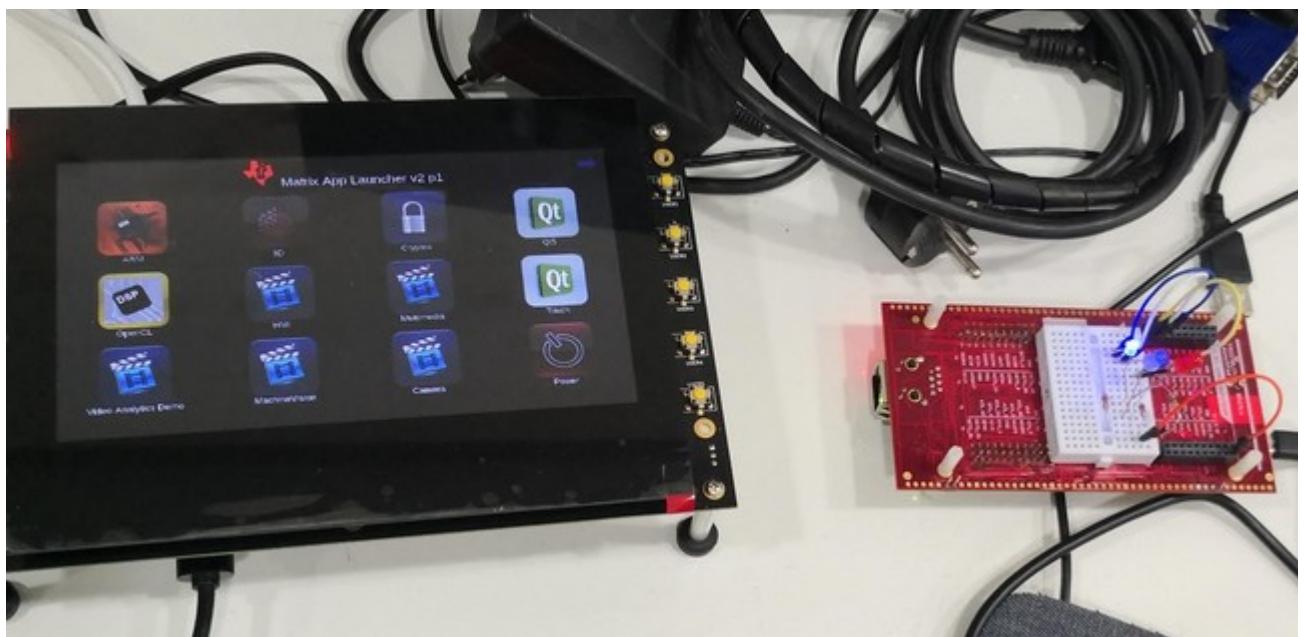
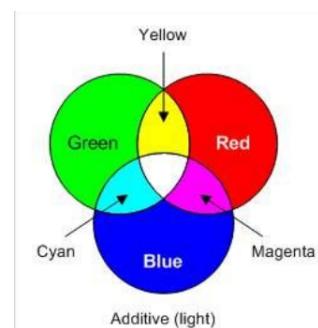
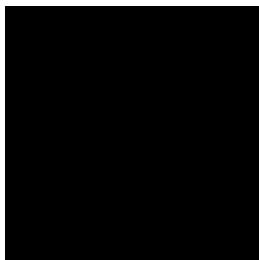


사진 구성 : DSP - USB1 번 (USB 1~3 중 1 번) – MCU– PORTA0,1,2 – LED(-) – LED(+) – 저항 1k – 3.3V
LED 신호 0 이면, ON 된다.

[DSP-MCU-LED : green 은 rgb 값 이상하게 나오나? : 픽셀 위치 때문..]

< 검정색 색상 분석 테스트 >

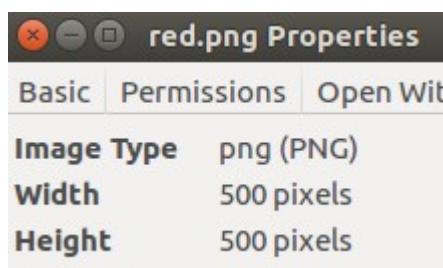
r = 0, g = 0, b = 0



< green 색상 x, y 좌표 변경하여 테스트 (50,50) >

```
img.at<Vec3b>(50, 50)[0],  
img.at<Vec3b>(50, 50)[1],  
img.at<Vec3b>(50, 50)[2]);
```

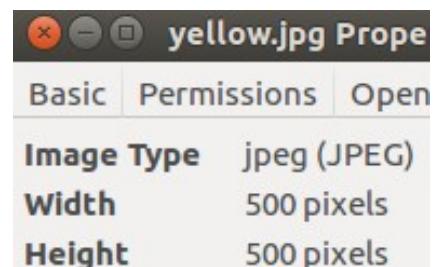
<코드 출력 결과와 RGB 그림판 분석 비교 >



```
r  
I read r  
r = 219, g = 17, b = 29  
buf[0] = 114
```



```
g  
I read g  
r = 1, g = 227, b = 101  
buf[0] = 103
```



```
y  
I read y  
r = 251, g = 246, b = 84  
buf[0] = 121
```

[DSP 보드 실습 유의사항]

DSP 실습 유의사항

1. 전원 관리 : 12V 5A 쓴다. 끌 때 멀티탭 바로 off 해버리면 큰일난다.

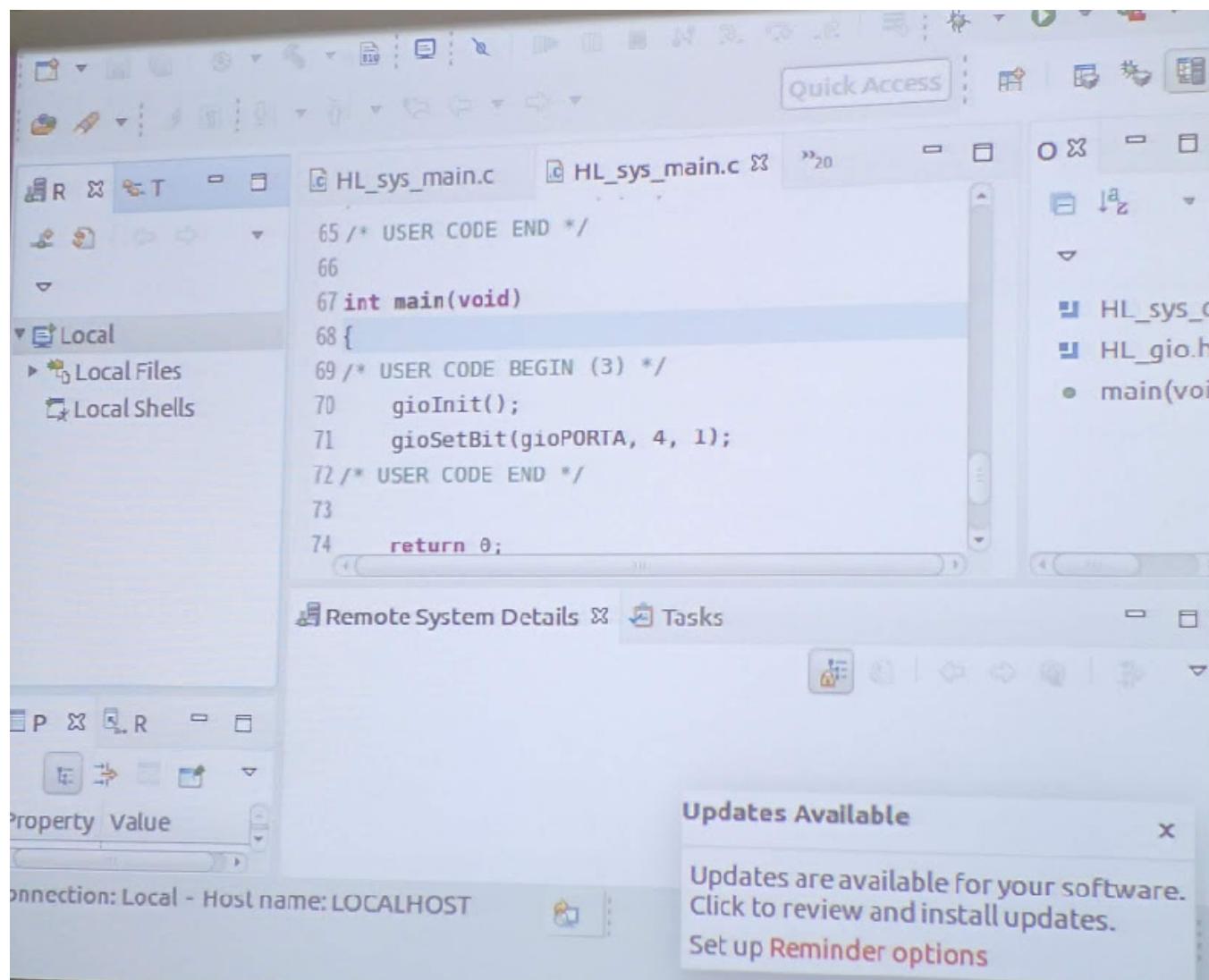
* 전원 on : 전원 on 버튼 눌러야 켜진다.(전원 케이블 바로 옆(터치 lcd 옆 아님))

* 전원 off : 전원버튼 15 초 누르거나(별로), 시스템옵션 - 전원 shutdown – led 꺼질때까지기다림, 5 초정도 기다림. 전원 뽑는다.

2. ETHERNET 선 - 둘 중 하나 쓴다.

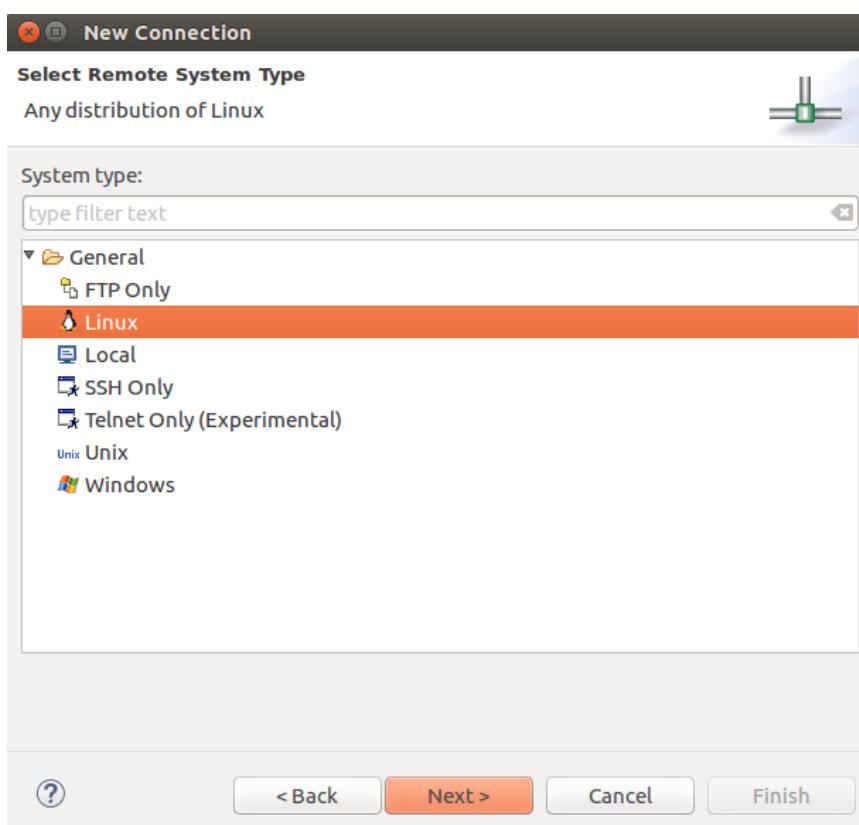
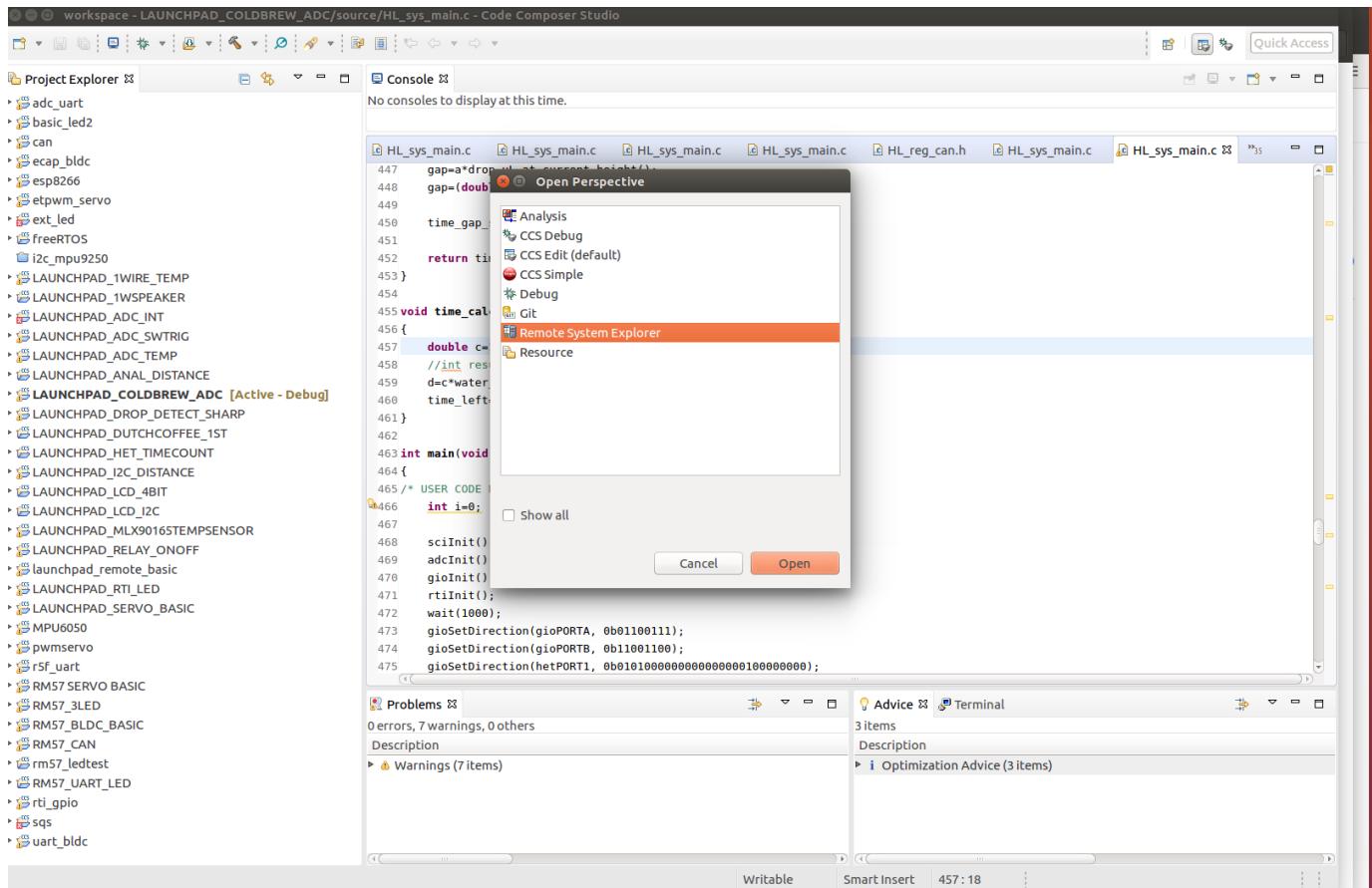
[DSP - 내부망 – PC(CCS) 기본 세팅 방법]

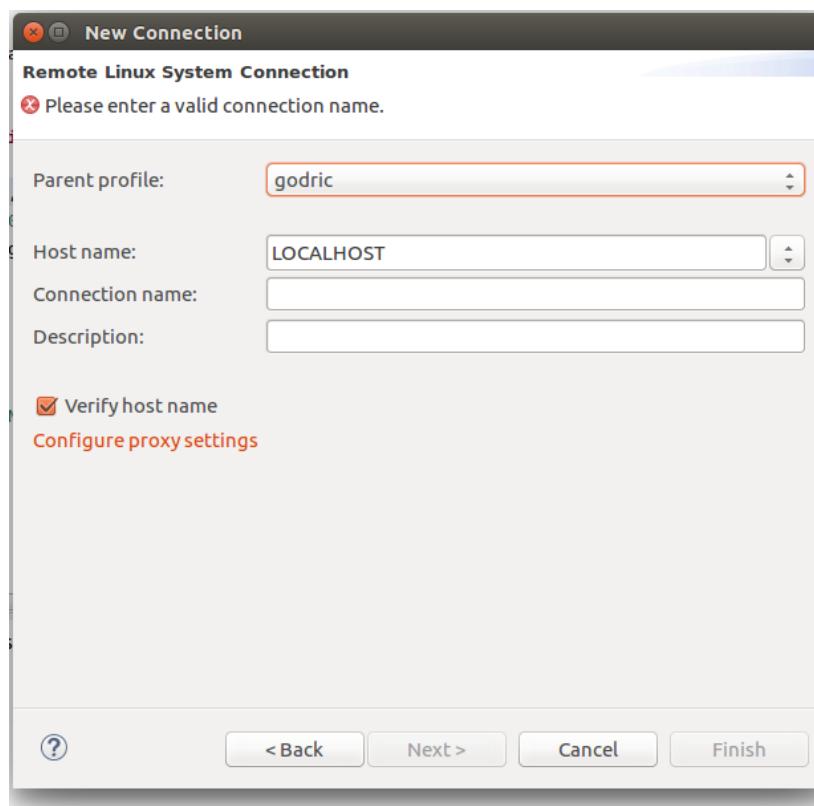
CCS 켠다. 아래 사진에서 창모양 + (quick access 라고 된 옆)를 누른다.



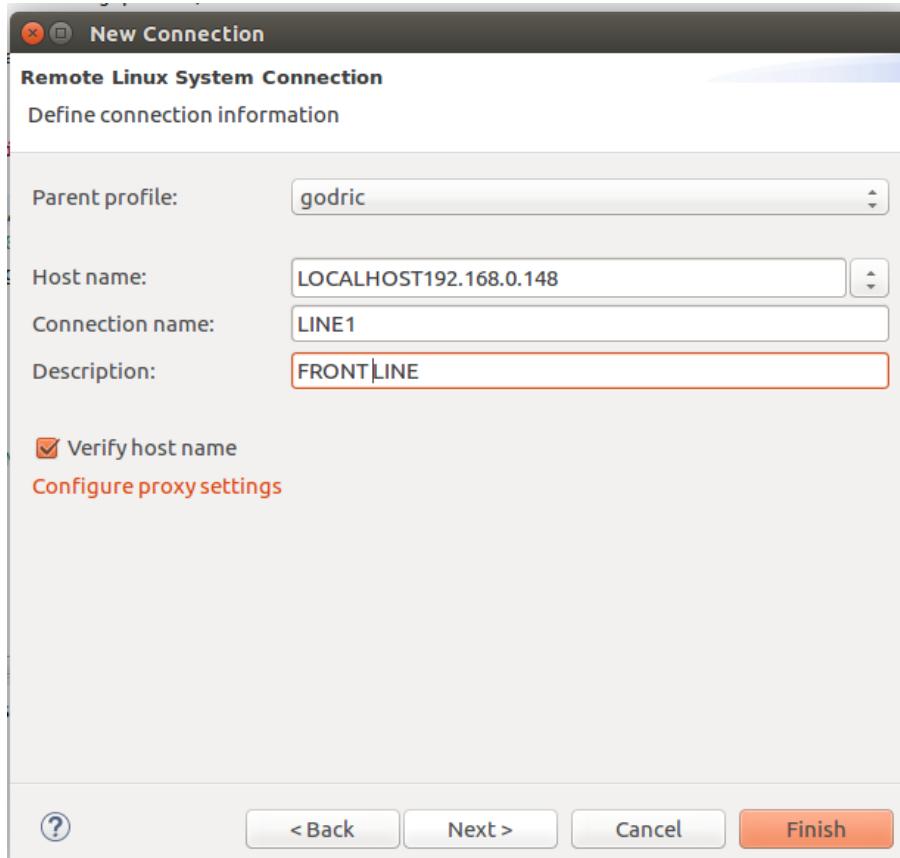
Remote system explorer

리눅스





이제 여기에 dsp 의 물리 주소를 넣어야 한다. (지금 랜포트 2 개 중 1 개에 지역 이더넷에 연결해 놓음. 어떤포트인지확인필),
(godric 아니고 koitt 로 계정 되어있는데 잘못되어있는 것임. 바꿔줘야함)
remote system explorer – connection



SETTING – NETWORK SETTING – ADDRESS 확인해야한다. 그리고 HOST NAME || 넣어줘야함.
(INET ADDR)

```
Network Settings
```

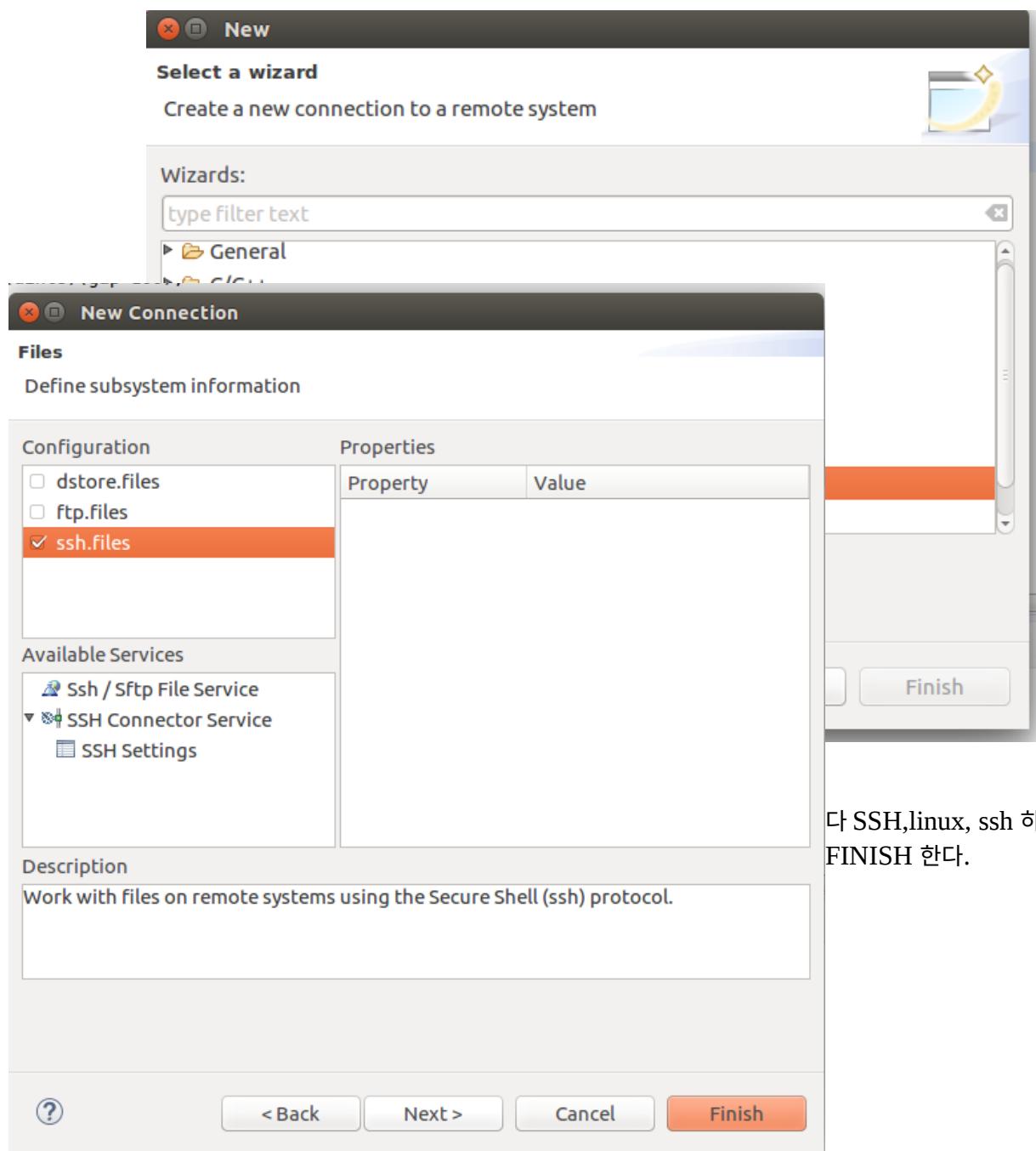
```
eth0      Link encap:Ethernet HWaddr FC:0F:4B:AB:78:E2
          UP BROADCAST MULTICAST MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 frame:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
          Interrupt:97

eth1      Link encap:Ethernet HWaddr FC:0F:4B:AB:78:E3
          inet addr:192.168.0.148 Bcast:192.168.0.255 Mask:255.255.255.0
          inet6 addr: fe80::fe0f:4bff:feab:78e3%763860/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:8443 errors:0 dropped:47 overruns:0 frame:0
          TX packets:356 errors:0 dropped:0 overruns:0 frame:0
          collisions:0 txqueuelen:1000
          RX bytes:993304 (970.0 KiB) TX bytes:55840 (54.5 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1%763860/128 Scope:Host
          UP LOOPBACK RUNNING MTU:65536 Metric:1
          RX packets:38134 errors:0 dropped:0 overruns:0 frame:0
          TX packets:38134 errors:0 dropped:0 overruns:0 frame:0
          collisions:0 txqueuelen:1
          RX bytes:3172415 (3.0 MiB) TX bytes:3172415 (3.0 MiB)

sit0     Link encap:IPv6-in-IPv4
          NOARP MTU:1480 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
```

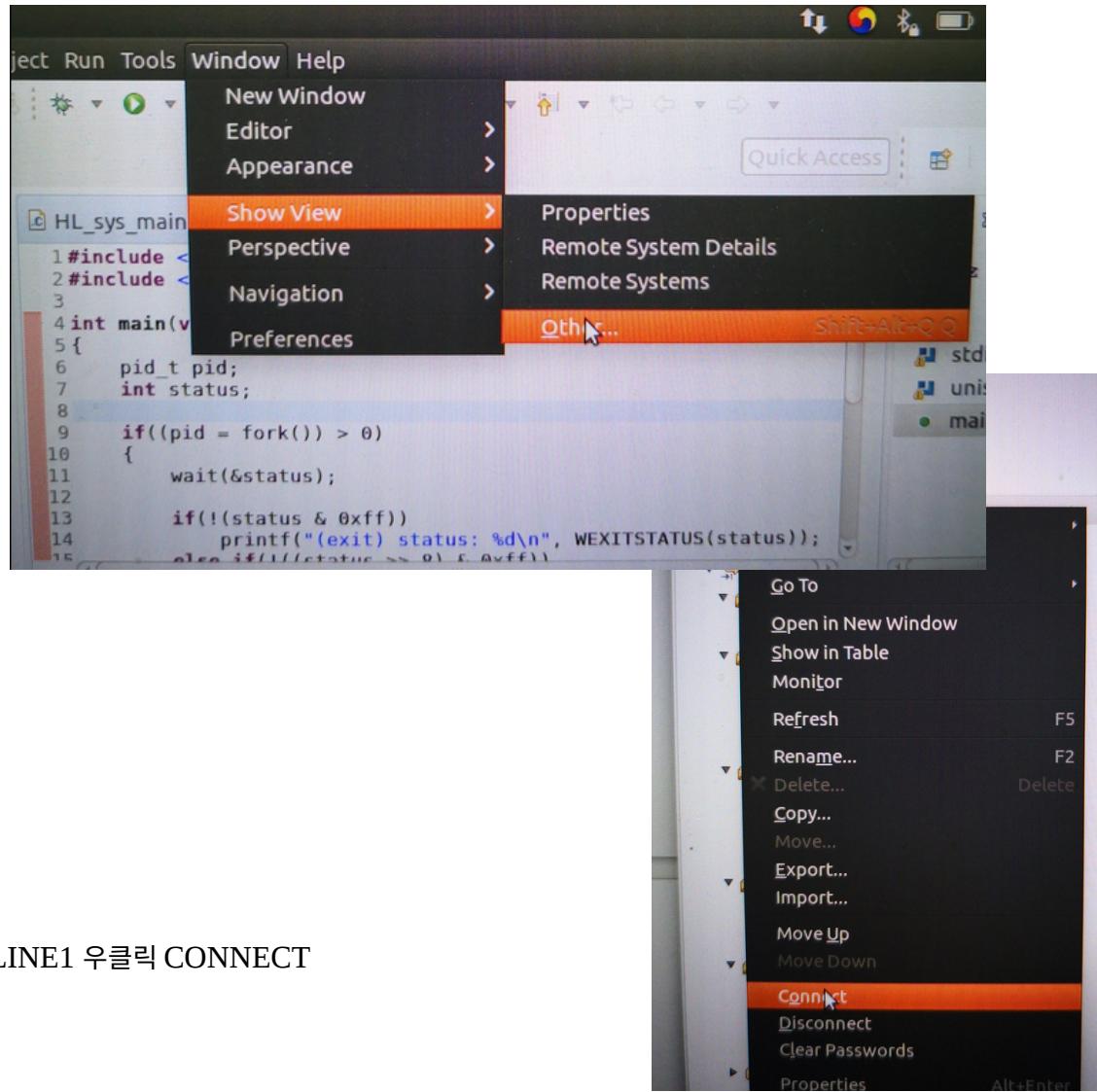
위의 eth1 포트. Inet addr 을 사용한다. 아래에서 LOCALHOST 지우고 192.168.0.148 입력한다.
나머지 이름 설명 등은 알아서 입력한다.



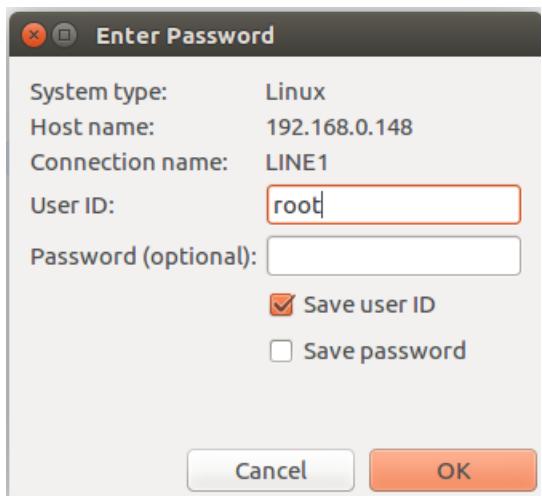
다 SSH,linux, ssh 하고 마지막에
FINISH 한다.

그리고 이제,

WINDOW SHOW OTHER – C C++ PROJECT

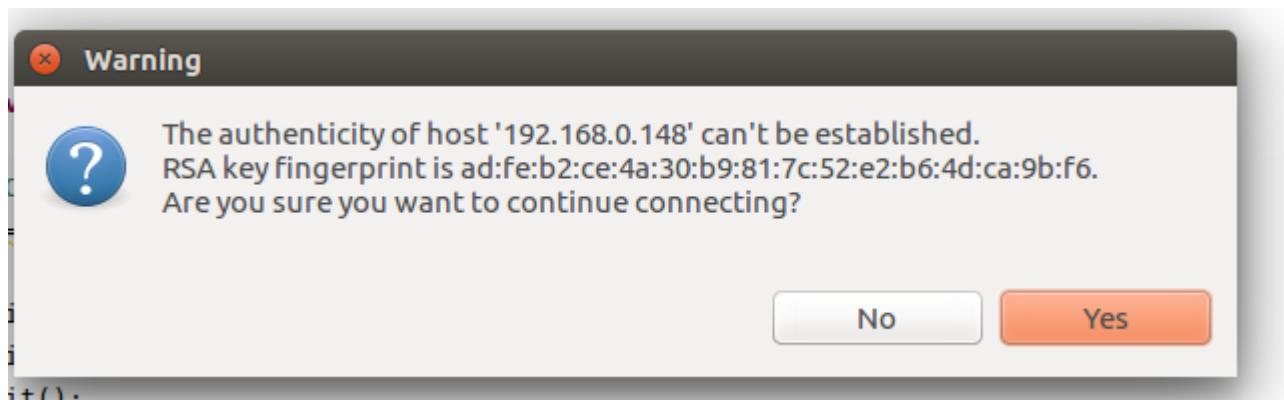


LINE1 우클릭 CONNECT



ok

yes, yes, yes



it();

그 다음, 통신창을 띄운다. → ccs에서 띄울 수 있으나 putty에서도 띄울 수 있다. (putty 설정은 다음페이지에 있다)

SSH Client

Quick Access

192.168.0.148 - PuTTY

```
login as: root
root@am57xx-evm:~# ls
Jangho  dongju  minsung  sanghoon  tibt
root@am57xx-evm:~# cd sanghoon/
root@am57xx-evm:~/sanghoon# ls
root@am57xx-evm:~/sanghoon# ls
root@am57xx-evm:~/sanghoon# ls ..
Jangho  dongju  minsung  sanghoon  tibt
root@am57xx-evm:~/sanghoon# vi fork_test.c
root@am57xx-evm:~/sanghoon# vi fork_chat_test.c
root@am57xx-evm:~/sanghoon# mkfifo myfifo
root@am57xx-evm:~/sanghoon# ls
root@am57xx-evm:~/sanghoon# ls -l
-rw-r--r--  1 root      root          519 Mar 26 15:31 fork_chat_test.c
-rw-r--r--  1 root      root          0 Mar 26 15:31 myfifo
root@am57xx-evm:~/sanghoon#
```

< 로그인 ? >

login as: root 로 한다.

누가누군지 모르고 다같이 작업하면 안되니까

참고 사항 : features : supporting instructions

< 끝 때는?>

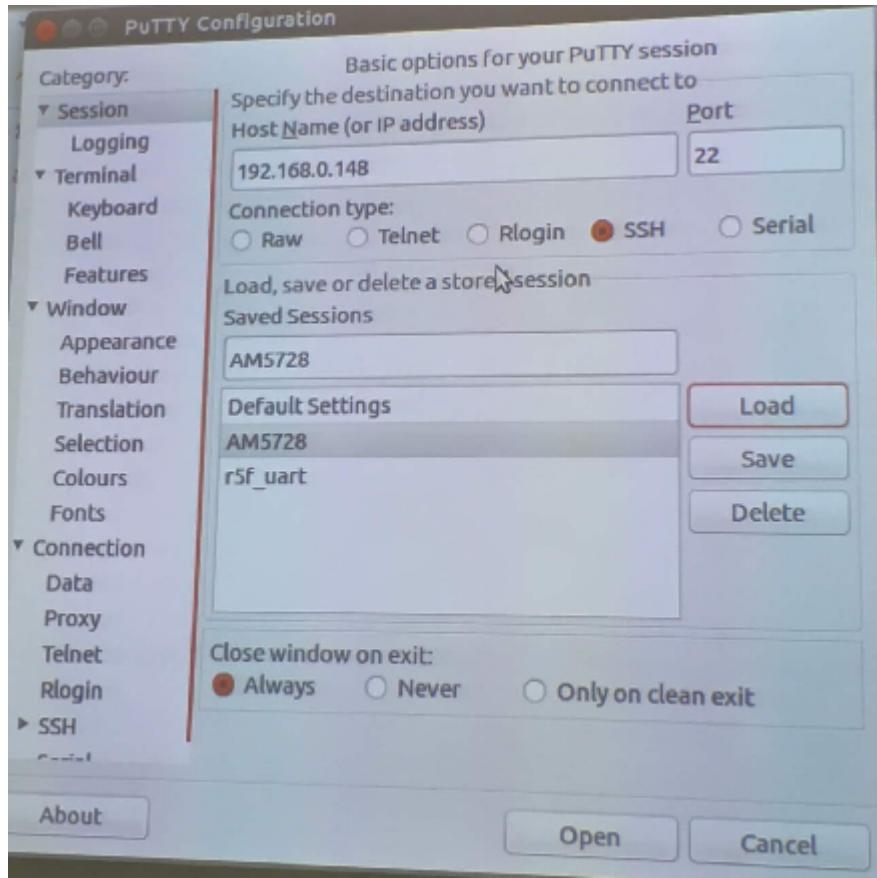
terminal 은 exit 누르면 나가지고

ccs 는 끄면 꺼진다.

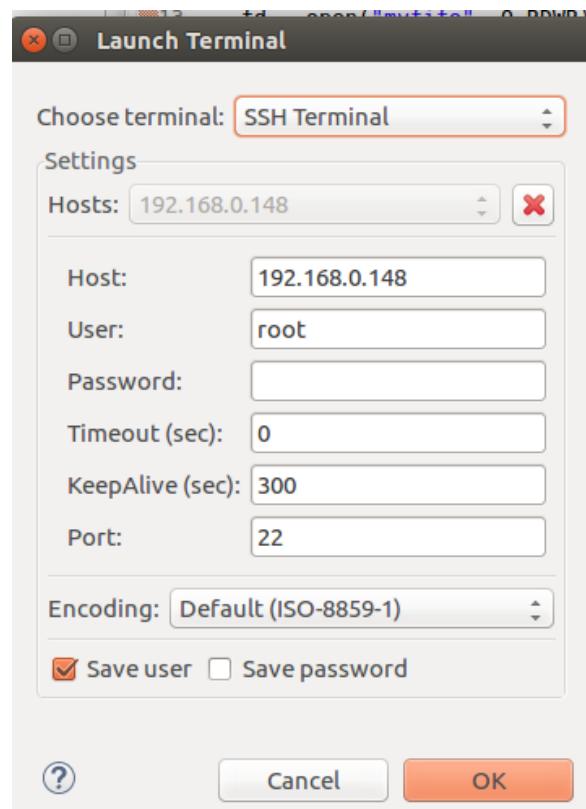
참고 사항 : Dsp 끝때 프로그램 살아있으면 복잡해지니까 프로그램먼저 끄고 dsp 끄자.

터미널 : 참고사항 .. PUTTY 는 텍스트 복사가 안 되지만, CCS 터미널에서는 텍스트 복사가 된다.

창 1 : putty 에서 아래처럼 저장, 실행한다.



창 2 : Ccs에서 아래 모니터 모양 누른다.



이렇게 입력한다. ok 하면 창 나온다. → 코딩한 파일 경로로 간다.

