



웹 서버 연동 LED 제어

조진성

경희대학교 컴퓨터공학과

Mobile & Embedded System Lab.



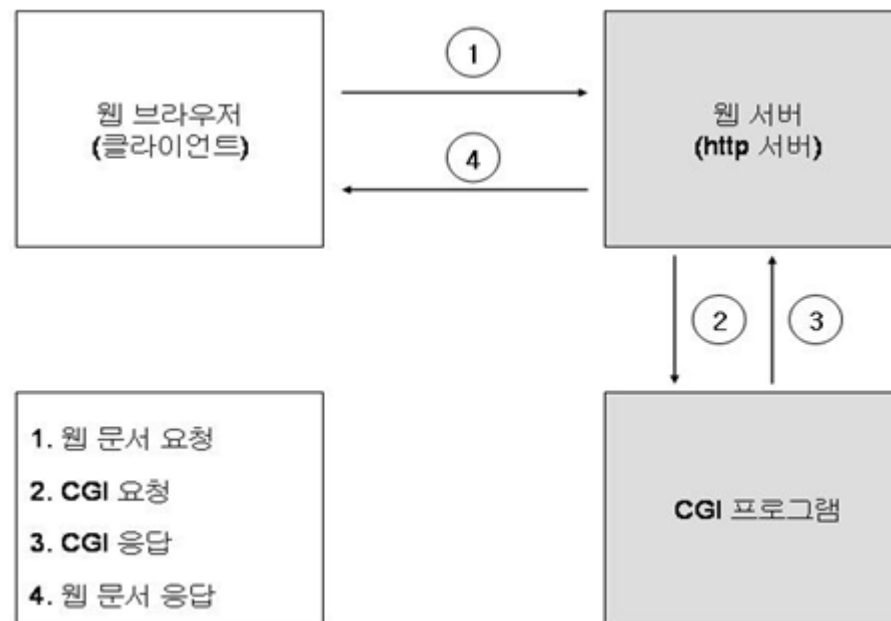
CGI (Common Gateway Interface)

❖ 웹 서버와 응용 프로그램과의 인터페이스를 위한 표준

- HTML은 정적인 파일 형태로 존재하지만, CGI 프로그램은 동적으로 HTML 파일을 생성
- CGI 프로그램은 cgi-bin이라는 특정한 디렉토리에서 관리
- CGI Programming Language
 - C/C++을 포함해 매우 다양

❖ 웹 서버 / 브라우저간의 CGI를 이용한 데이터 흐름

- (2)과정에서 환경변수를 통해 파라미터 전달



실습 환경 구성



❖ 웹 서버 설치 및 설정 [Directory: /home/pi]

Install web server

```
pi@raspberrypi: ~ $ sudo apt-get install apache2
```

Check CGI directory

```
pi@raspberrypi: ~ $ cat /etc/apache2/sites-available/default-ssl.conf | grep cgi
```

```
pi@raspberrypi:~ $ cat /etc/apache2/sites-available/default-ssl.conf | grep cgi
#Include conf-available/serve-cgi-bin.conf
<FilesMatch "\#.(cgi|shtml|phtml|php)$">
<Directory /usr/lib/cgi-bin>
```

CGI enable

```
pi@raspberrypi: ~ $ sudo a2enmod cgi
```

Restart webserver

```
pi@raspberrypi: ~ $ sudo service apache2 restart
```

실습 환경 구성



❖ 라즈베리파이 IP Address 확인

- PC와 라즈베리파이를 새로운 LAN선으로 직접 연결
 - Eth0의 IP Address(아래 사진에서는 169.254.190.194)를 기억하기

```
# Terminal [ Ctrl + Alt + t ]  
pi@raspberrypi: ~ $ ifconfig
```

```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi: ~ $ ifconfig  
eth0      Link encap:Ethernet  HWaddr b8:27:eb:3c:79:c0  
          inet addr:169.254.190.194  Bcast:169.254.255.255  Mask:255.255.0.0  
          inet6 addr: fe80::6214:da8a:99d:1259/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:703 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:54 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:59148 (57.7 KiB)  TX bytes:10591 (10.3 KiB)  
  
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:448 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:448 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1  
          RX bytes:36288 (35.4 KiB)  TX bytes:36288 (35.4 KiB)  
  
wlan0     Link encap:Ethernet  HWaddr b8:27:eb:69:2c:95  
          inet6 addr: fe80::8459:89ac:6471:2ce/64 Scope:Link  
          UP BROADCAST MULTICAST  MTU:1500  Metric:1  
          RX packets:295 errors:0 dropped:295 overruns:0 frame:0  
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:123175 (120.2 KiB)  TX bytes:0 (0.0 B)  
  
pi@raspberrypi: ~ $
```

실습 환경 구성

❖ 윈도우 PC IP Address 변경

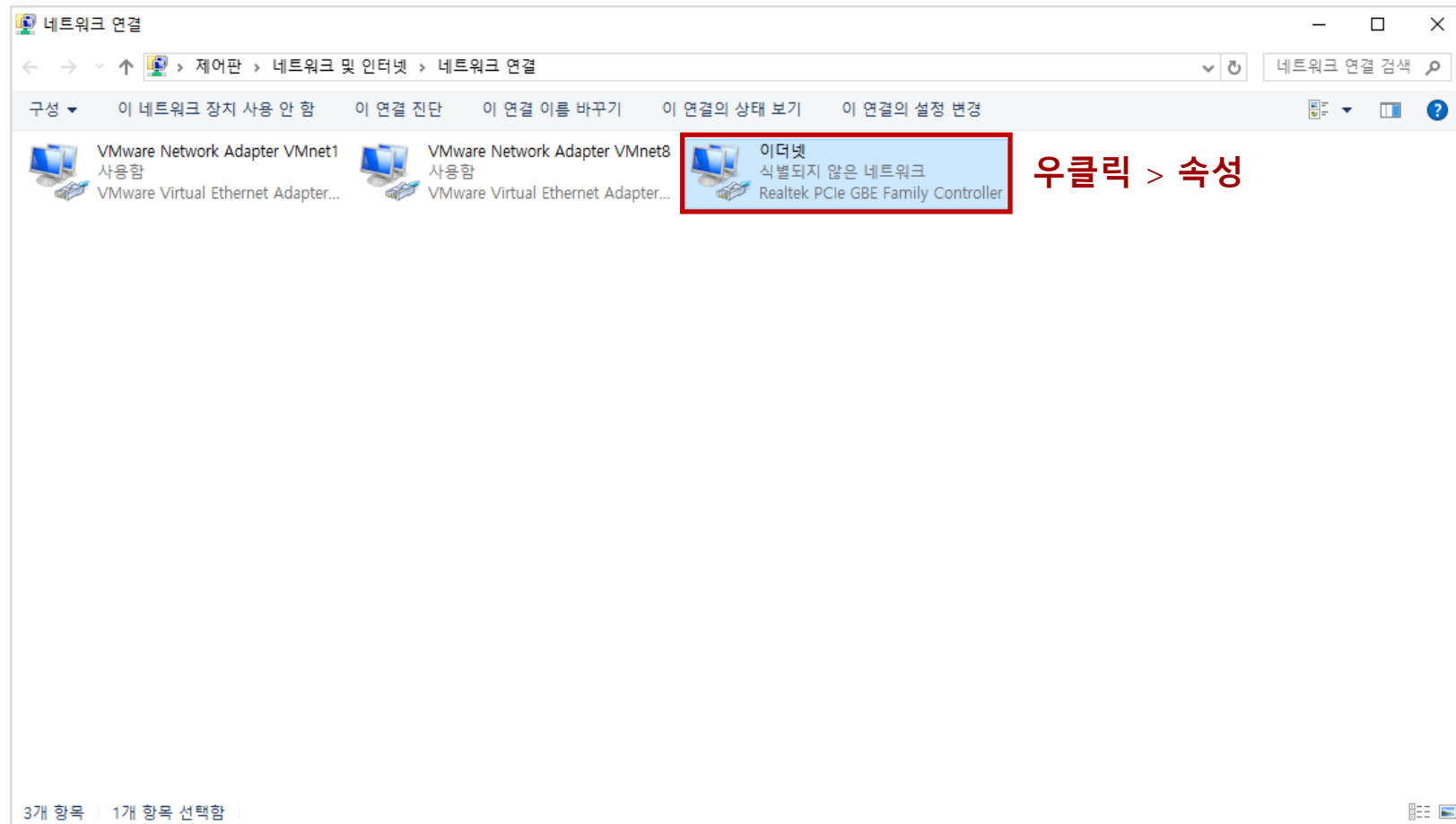
■ 네트워크 및 공유 센터 > 어댑터 설정 변경



실습 환경 구성

❖ 윈도우 PC IP Address 변경

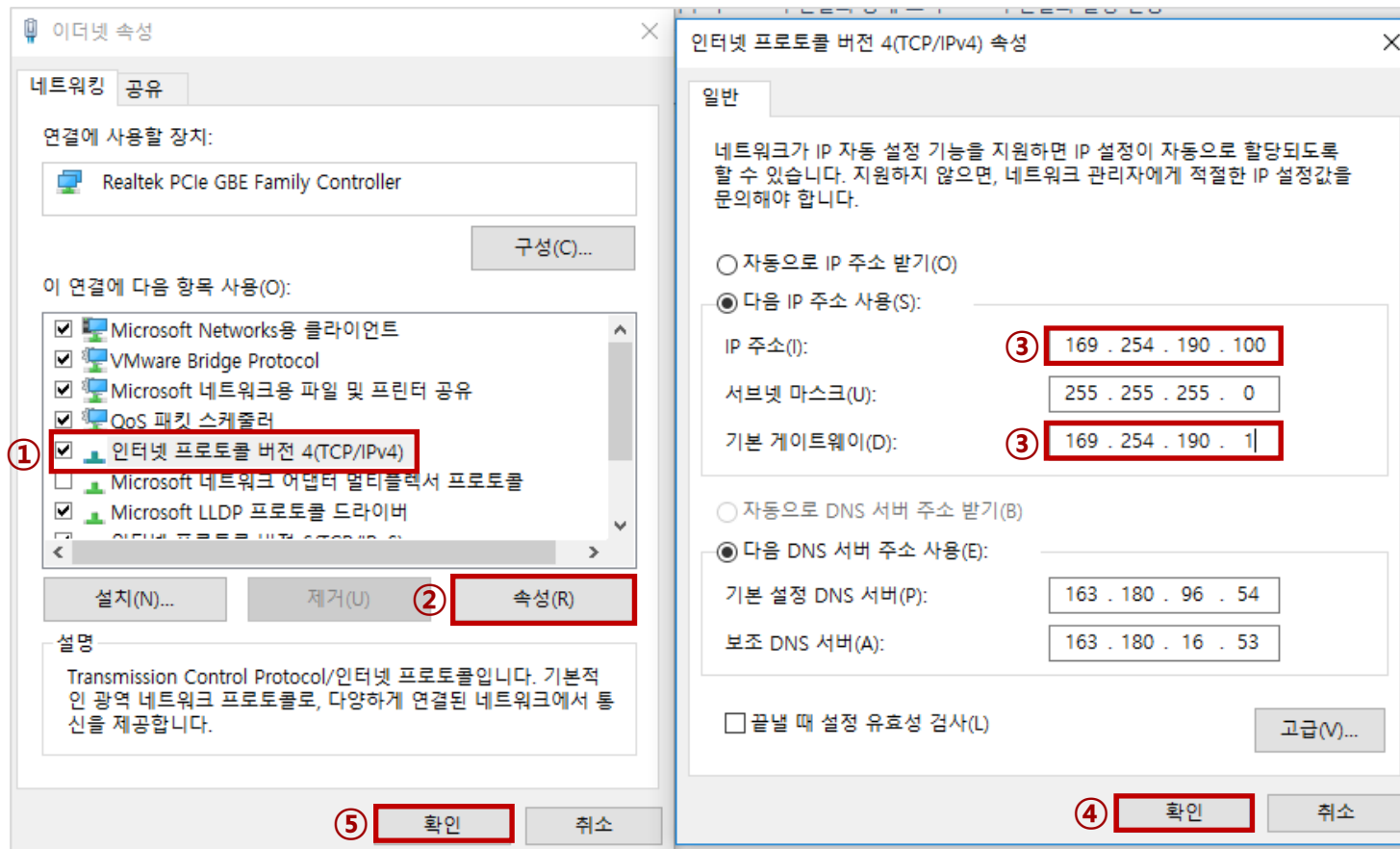
- 이더넷 우클릭 > 속성



실습 환경 구성

❖ 윈도우 PC IP Address 변경

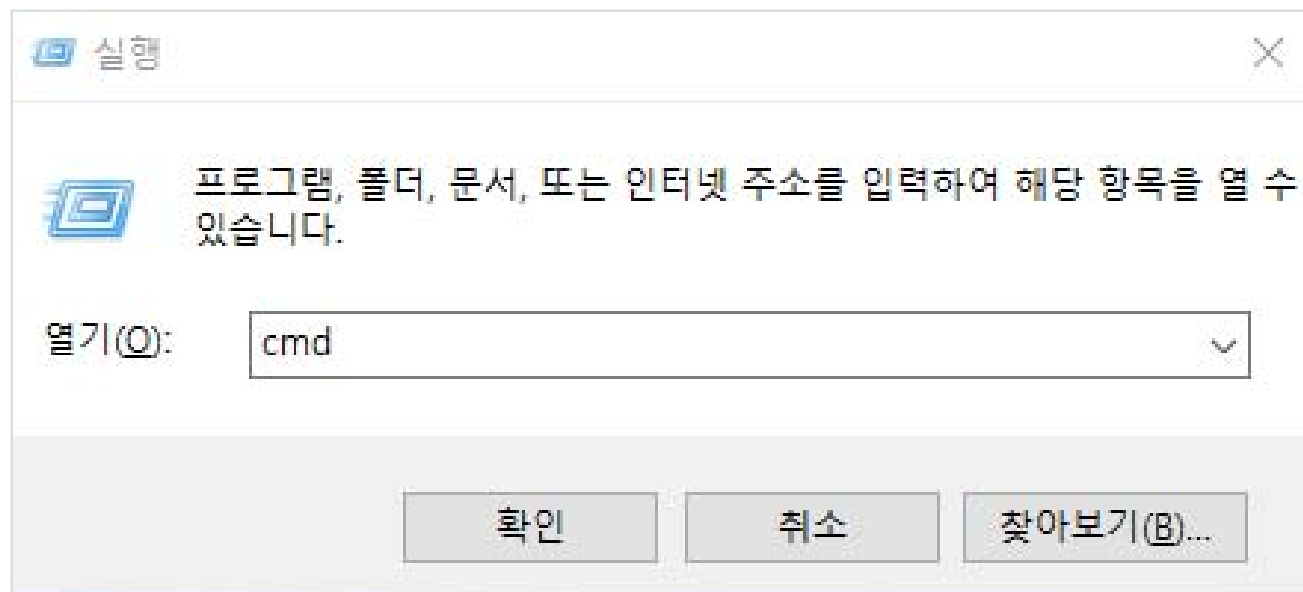
- 인터넷 프로토콜 버전 4(TCP/IPv4) 클릭 > 속성 > IP주소 및 게이트웨이 변경
 - 이때 IP 주소와 게이트웨이는 p.5에서 확인한 라즈베리파이의 IP를 참고하여 변경
 - 기존 IP 주소와 게이트웨이는 잊어버리지 않도록 꼭! 메모



실습 환경 구성

❖ 윈도우 PC IP Address 변경

- 실행(윈도우 키 + r) > cmd



실습 환경 구성

❖ 윈도우 PC IP Address 변경

- ipconfig 입력 > IPv4 주소 및 기본 게이트웨이 확인

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\MESL>ipconfig

Windows IP 구성

이더넷 어댑터 이더넷:

    연결별 DNS 접미사. . . . . :
    링크-로컬 IPv6 주소. . . . . : fe80::8933:d61a-3bd0:3482%13
    IPv4 주소. . . . . : 169.254.190.100
    서브넷 마스크. . . . . : 255.255.255.0
    기본 게이트웨이. . . . . : 169.254.190.1

이더넷 어댑터 VMware Network Adapter VMnet1:

    연결별 DNS 접미사. . . . . :
    링크-로컬 IPv6 주소. . . . . : fe80::b11e:3de:375f:aa6%6
    IPv4 주소. . . . . : 192.168.85.1
    서브넷 마스크. . . . . : 255.255.255.0
    기본 게이트웨이. . . . . :

이더넷 어댑터 VMware Network Adapter VMnet8:

    연결별 DNS 접미사. . . . . :
    링크-로컬 IPv6 주소. . . . . : fe80::e985:d7c9:6531:9980%10
    IPv4 주소. . . . . : 192.168.188.1
    서브넷 마스크. . . . . : 255.255.255.0
    기본 게이트웨이. . . . . :

터널 어댑터 Teredo Tunneling Pseudo-Interface:

    미디어 상태. . . . . : 미디어 연결 끊김
    연결별 DNS 접미사. . . . . :

C:\Users\MESL>
```

실습 환경 구성



❖ 윈도우 PC IP Address 변경

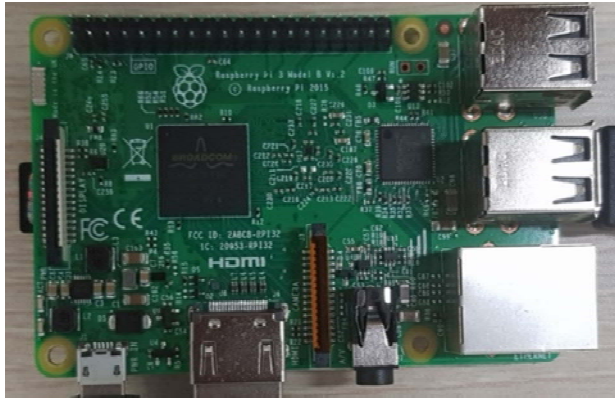
▪ ping [Raspberry pi IP Address]

- 아래 그림과 같이 ping이 정상적으로 나가면 구성 완료

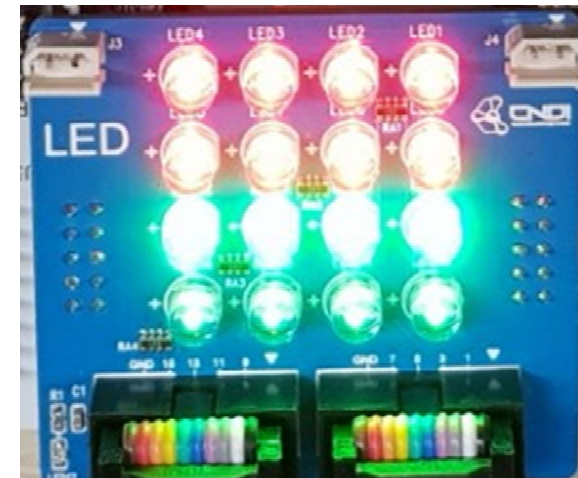
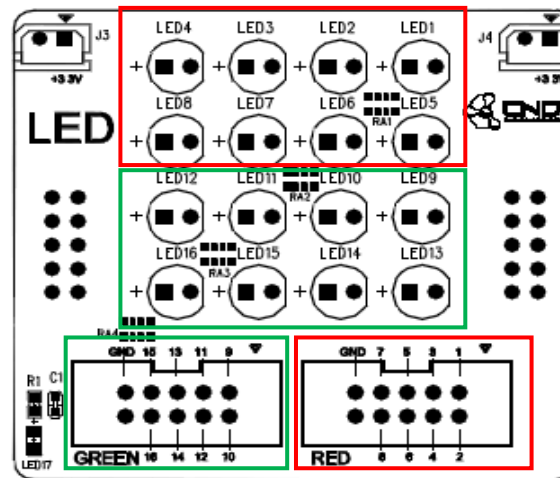
```
C:\Users\MESL>  
C:\Users\MESL>ping 169.254.190.194 ping 테스트  
  
Ping 169.254.190.194 32바이트 데이터 사용:  
169.254.190.194의 응답: 바이트=32 시간<1ms TTL=64  
169.254.190.194의 응답: 바이트=32 시간=1ms TTL=64  
169.254.190.194의 응답: 바이트=32 시간<1ms TTL=64  
  
169.254.190.194에 대한 Ping 통계:  
    패킷: 보냄 = 3, 받음 = 3, 손실 = 0 (0% 손실),  
왕복 시간(밀리초):  
    최소 = 0ms, 최대 = 1ms, 평균 = 0ms  
Control-C  
^C  
C:\Users\MESL>
```

하드웨어 구성

❖ Raspberry Pi 3 Model B



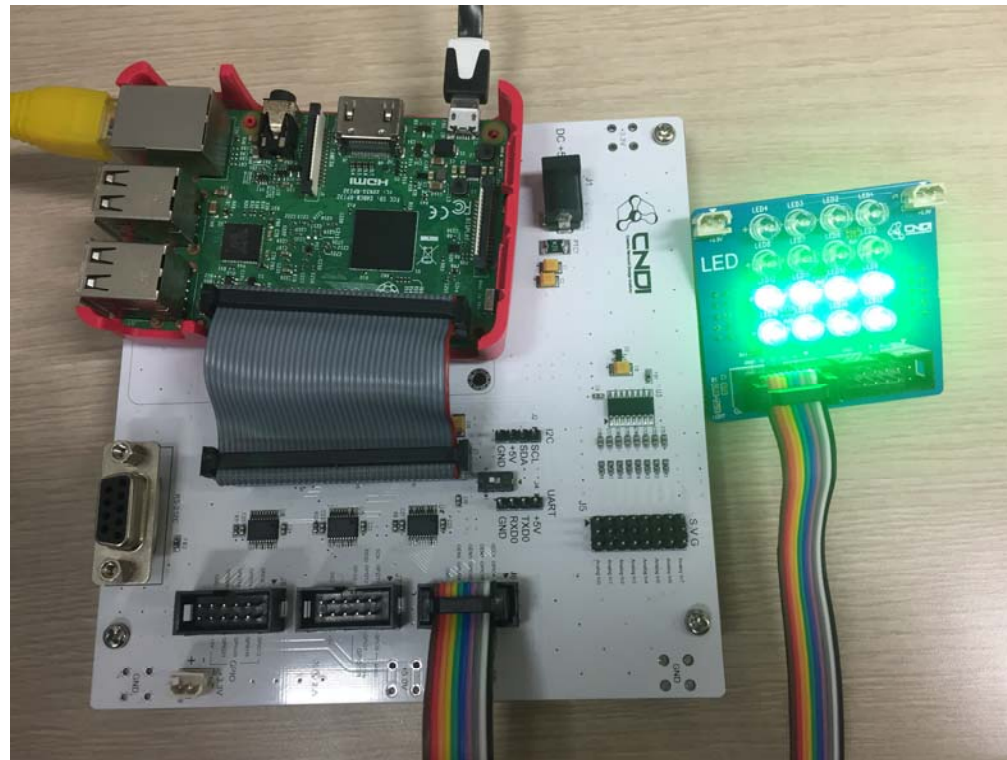
❖ LED 모듈



하드웨어 구성 (1)

❖ 웹 브라우저로 LED 8개 Blink

❖ Lab. 2-2)와 동일



실습 예제 (1)



❖ HTML file [Directory: /var/www/html]

HTML coding

pi@raspberrypi: /var/www/html \$ sudo vi led1.html

```
1 <html>
2   <head>
3     <title>LED1 CGI PROGRAM</title>
4   </head>
5
6   <body>
7     <p>LED START</p>
8     <a href="cgi-bin/led1.cgi">ON</a>
9   </body>
10 </html>
```

실습 예제 (1)



❖ CGI Program using Wiring Pi [Directory: /home/pi]

CGI program coding (using direct GPIO)

pi@raspberrypi: ~ \$ vi led1.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <wiringPi.h>
4
5 const int Led[8] = {
6     4, 17, 18, 27, 22, 23, 24, 25
7 };
8
9 void Setup() {
10     int i;
11
12     if(wiringPiSetupGpio() == -1) {
13         exit(-1);
14     }
15
16     for(i = 0; i < 8; i++) {
17         pinMode(Led[i], OUTPUT);
18         digitalWrite(Led[i], LOW);
19     }
20 }
21
22 void LedTest() {
23     int i;
24
25     for(i = 0; i < 8; i++) {
26         digitalWrite(Led[i], HIGH);
27     }
28     delay(500);
29
30     for(i = 0; i < 8; i++) {
31         digitalWrite(Led[i], LOW);
32     }
33     delay(500);
34 }
35
```

```
36 int main() {
37     printf("Content-type:text/html\n\n");
38     printf("<html>\n<head>\n<title>LED TEST!\n</title>\n</head>\n\n");
39     printf("<body>\n<p>LED BLINK\n</p>\n\n");
40
41     int i;
42
43     Setup();
44
45     for(i = 0; i < 5; i++) {
46         LedTest();
47     }
48
49     printf("</body>\n\n</html>\n\n");
50
51     return 0;
52 }
```


실습 예제 (1)



❖ Install CGI Program [Directory: /usr/lib/cgi-bin]

```
# Compile
pi@raspberrypi: ~ $ gcc -o led1.cgi led1.c -lwiringPi

# Copy CGI program
pi@raspberrypi: ~ $ sudo cp led1.cgi /usr/lib/cgi-bin/
pi@raspberrypi: ~ $ cd /usr/lib/cgi-bin

# Change permission
pi@raspberrypi:/usr/lib/cgi-bin $ sudo chown root:root led1.cgi
pi@raspberrypi:/usr/lib/cgi-bin $ sudo chmod u+s led1.cgi
```

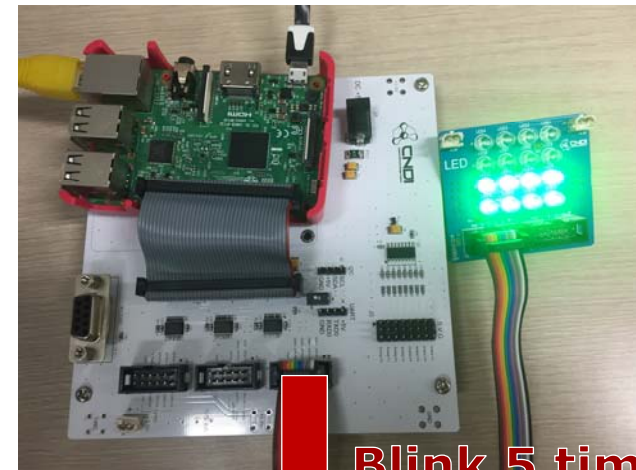
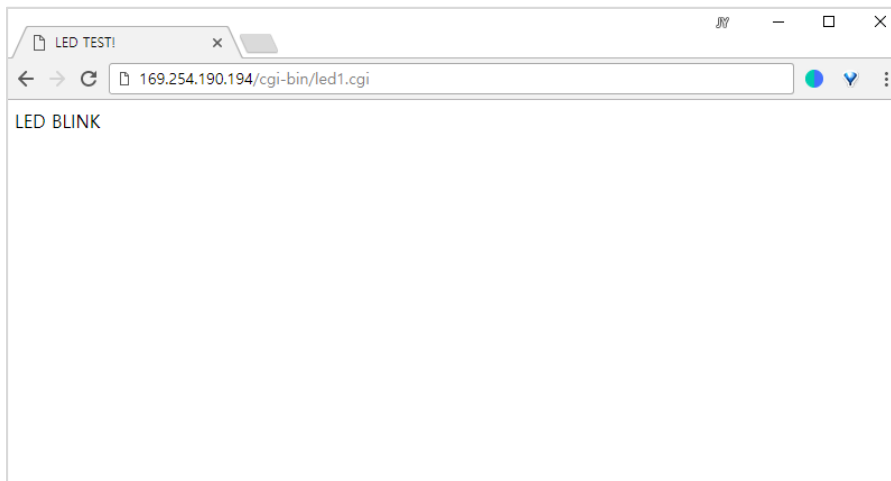
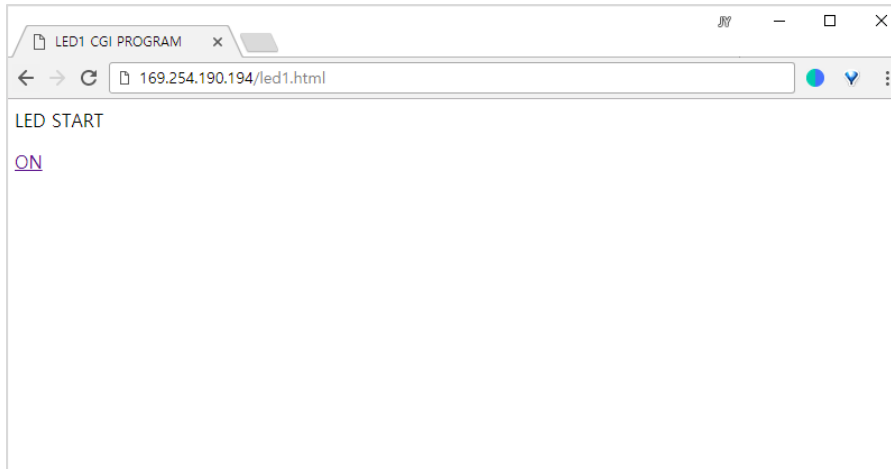
❖ Setuid bit

- `chmod u+s led1.cgi`
- Effective user ID가 file의 owner로 변경됨
- 웹 서버는 pi 권한으로 실행

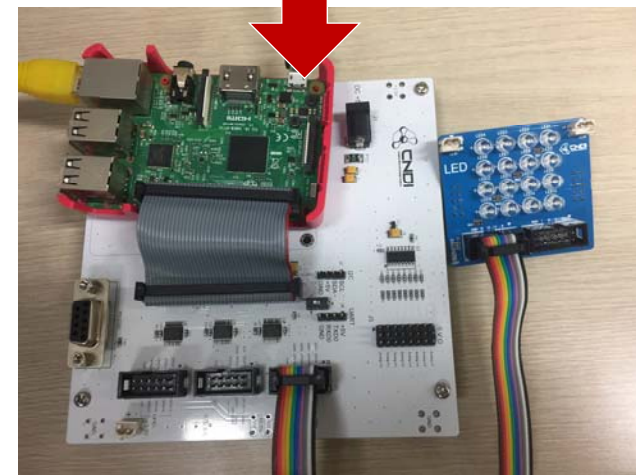
실습 예제 (1)

❖ 동작 화면

- 인터넷 브라우저에서 [라즈베리파이 IP Address]/led1.html 접속

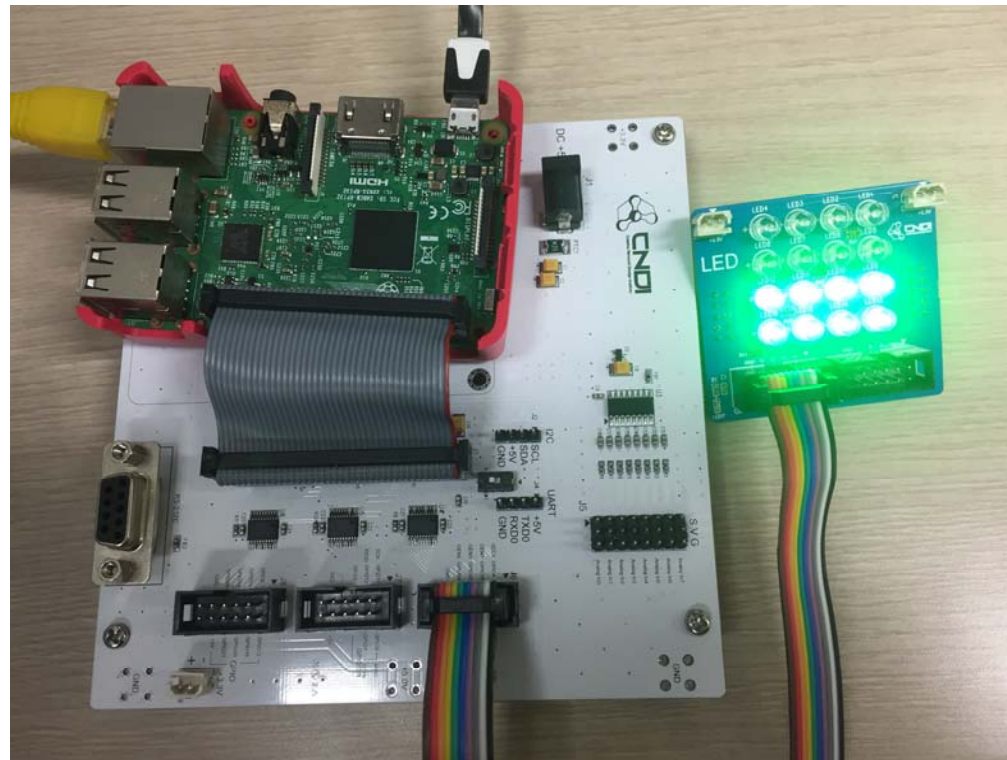


Blink 5 times



하드웨어 구성 (2)

- ❖ 웹 브라우저를 통한 입력을 LED 이진수로 표현
- ❖ Lab 2-2)와 동일



실습 예제 (2)



❖ HTML file [Directory: /var/www/html]

HTML coding

pi@raspberrypi: /var/www/html \$ sudo vi led2.html

```
1 <html>
2   <head>
3     <title>LED2 CGI PROGRAM</title>
4   </head>
5
6   <body>
7     <form method=get action="/cgi-bin/led2.cgi">
8       <p>몇 개의 LED를 ON 하시겠습니까?</p>
9       <input type="text" name="value" maxlength="6" size="6">
10      <input type="submit" name="button" value="input">
11    </form>
12  </body>
13 </html>
```

실습 예제 (2)



❖ CGI Program using Wiring Pi [Directory: /home/pi]

CGI program coding (using WiringPi)
pi@raspberrypi: ~ \$ vi led2.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <wiringPi.h>
5
6 #define MAX_CONTENT_LENGTH 16
7
8 const int Led[16] = {
9     4, 17, 18, 27, 22, 23, 24, 25
10 };
11
12 void getData(char **data) {
13     char *queryString, *delim;
14     int len = 0;
15     int i;
16
17     queryString = getenv("QUERY_STRING");
18     delim = strchr(queryString, '=');
19     for(i = 0; i < sizeof(delim); i++) {
20         delim[i] = delim[i+1];
21     }
22     delim[sizeof(delim)] = '\0';
23
24     if(delim) {
25         len = strlen(delim);
26         *data = malloc(sizeof(char) * len);
27         strcpy(*data, delim);
28     }
29     else {
30         len = 1;
31         *data = malloc(sizeof(char) * len);
32         **data = "\0";
33     }
34 }
35
```

```
36 void Setup() {
37     int i;
38
39     if(wiringPiSetupGpio() == -1) {
40         exit(-1);
41     }
42
43     for(i = 0; i < 16; i++) {
44         pinMode(Led[i], OUTPUT);
45         digitalWrite(Led[i], LOW);
46     }
47 }
48
49 void LedTest(char *data) {
50     int i;
51     int num = atoi(data);
52
53     for(i = 0; i < 4; i++) {
54         digitalWrite(Led[i], ((num >> i)%2));
55     }
56 }
57
58 int main() {
59     printf("Content-type:text/html\n\n");
60     printf("<html>\n<head>\n<title>LED2 TEST!</title>\n</head>\n");
61     printf("<body>\n<p>LED ON</p>\n");
62
63     char *data = (char *)0;
64     getData(&data);
65     Setup();
66     LedTest(data);
67
68     printf("</body>\n</html>");
69
70     return 0;
71 }
```

실습 예제 (2)



❖ Install CGI Program [Directory: /usr/lib/cgi-bin]

```
# Compile
pi@raspberrypi: ~ $ gcc -o led2.cgi led2.c -lwiringPi

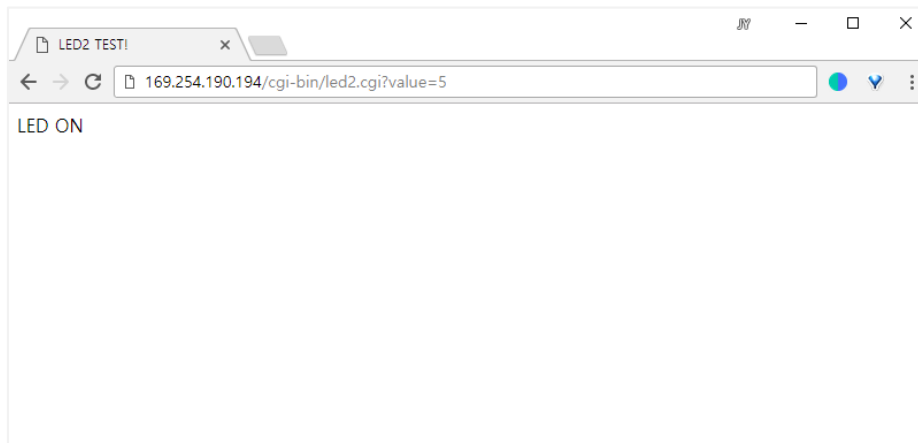
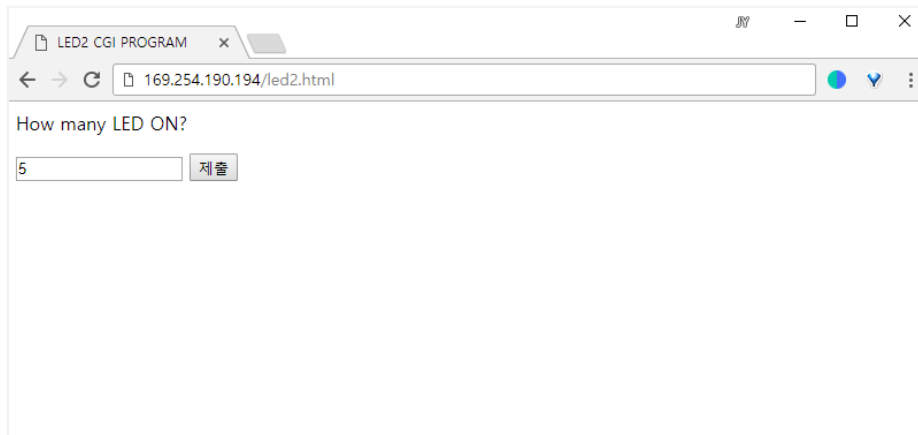
# Copy CGI program
pi@raspberrypi: ~ $ sudo cp led2.cgi /usr/lib/cgi-bin
pi@raspberrypi: ~ $ cd /usr/lib/cgi-bin

# Change permission
pi@raspberrypi: /usr/lib/cgi-bin $ sudo chown root:root led2.cgi
pi@raspberrypi: /usr/lib/cgi-bin $ sudo chmod u+s led2.cgi
```

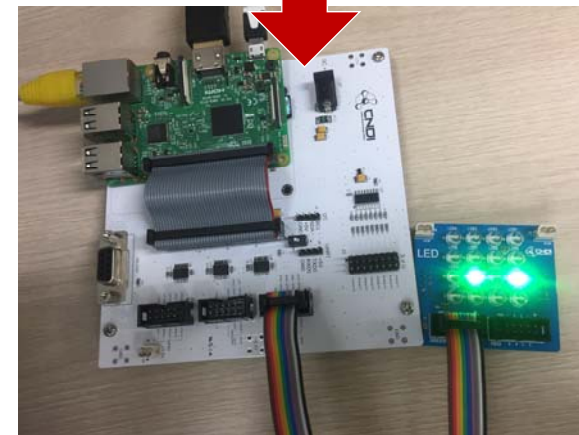
실습 예제 (2)

❖ 동작 화면

- 인터넷 브라우저에서 [라즈베리파이 IP Address]/led2.html 접속



5 = 0b0101



팀 프로젝트 과제 (추가 점수)



❖ 웹 서버 연동 1/100초 Stop Watch 작성 (FND 제어)

- 웹 페이지는 3개의 버튼 존재: 'Clear' 'Start' 'Stop'
 - 'Clear': 0으로 초기화
 - 'Start': 시작 또는 재시작 (1/100초)
 - 'Stop': 정지
- 힌트
 - `<form method=get action="cgi-bin/start.cgi"> <input type="submit" name="button" value="input"> </form>`
 - start.cgi는 child process(FND process)를 fork하여 1/100초 stop watch로 FND를 제어
 - stop.cgi는 FND process에 stop 메시지를 전달 (FIFO 사용)
 - clear.cgi는 FND process에 clear 메시지를 전달 (FIFO 사용)
 - FND process는 FND를 제어하는 thread와 FIFO로 IPC를 수행하는 thread로 구성



Q & A



<http://mesl.khu.ac.kr>