한국외국어대학교
HANKUK UNIVERSITY OF FOREIGN STUDIES

Internet of Things (IoT) Systems

Week 12

Raspberry Pi Programming

Ikram Syed, Ph.D.
Associate Professor
Department of Information and Communication Engineering
Hankuk University of Foreign Studies (HUFS)

Spring – 2025

Remote Control Using PHP

Program to control LED

```
1   #include <wiringPi.h>
2   #include <stdio.h>
3
4    #define PIN 7
5
6   int main(void) {
7       if(wiringPiSetup() == -1) return 1;
8       pinMode(PIN,OUTPUT);
9       digitalWrite(PIN,HIGH);
10      printf("1");
11  }
```

Part	Meaning
-rwsr-sr-x	File permissions and type
1	**Number of hard links** to the file
root	Owner (user) of the file
pi	Group that owns the file

```
pi@raspberrypi:~ $ gcc -o PHP_LEDON PHP_LEDON.c -lwiringPi
pi@raspberrypi:~ $ sudo chown root PHP_LEDON
pi@raspberrypi:~ $ sudo chmod +s PHP_LEDON
```

[Figure 6-6] Compiling File & Providing PHP Execution Permission

```
-rwsr-sr-x 1 root pi
```

[Figure 6-7] Changed Permissions & Owner

Remote Control Using PHP

Program to control LED

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

 #define PIN 7

int main(void)  {

    if(wiringPiSetup() == -1) return 1;
    pinMode(PIN,OUTPUT);
    digitalWrite(PIN,LOW);
    printf("0");
}
```

```
pi@raspberrypi:~ $ gcc -o PHP_LEDOFF PHP_LEDOFF.c -lwiringPi
pi@raspberrypi:~ $ sudo chown root PHP_LEDOFF
pi@raspberrypi:~ $ sudo chmod +s PHP_LEDOFF
```

[Figure 6-9] Compiling File & Providing PHP Execution Permission

```
-rwsr-sr-x 1 root pi
```

[Figure 6-10] Changed Permissions & Owner

Create PHP file

```
pi@raspberrypi:~ $ cd /var/www/html
pi@raspberrypi:/var/www/html $ sudo nano remote_con.php
```

[Figure 6-11] Creating PHP File

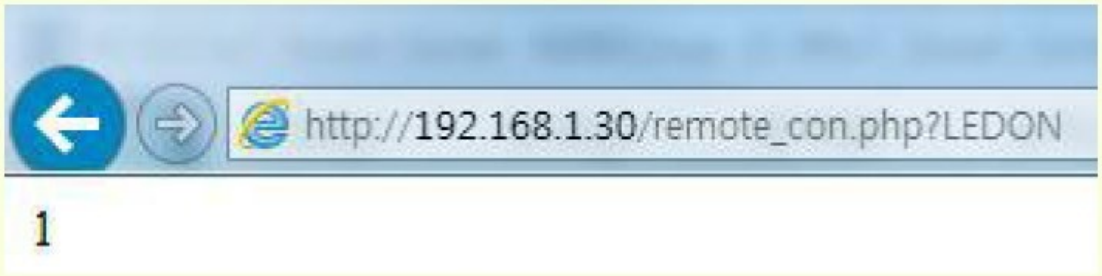
remote_con.php

```
<?php
        if(isset($_GET['LEDON'])){
                $value = shell_exec("/home/pi/PHP_LEDON");
                echo $value;
        }else if(isset($_GET['LEDOFF'])){
                $value = shell_exec("/home/pi/PHP_LEDOFF");
                echo $value;
        }
?>
```

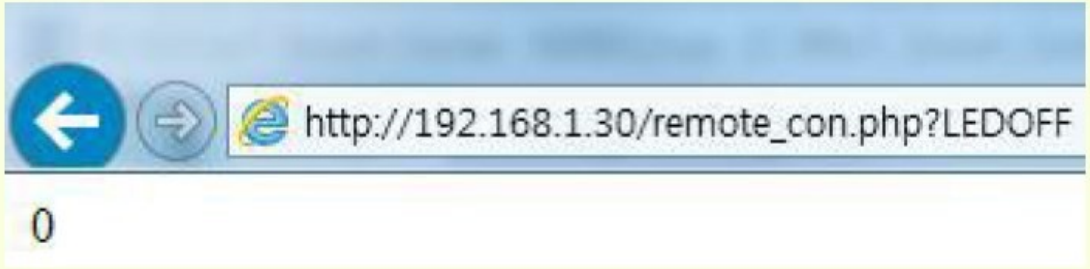
Make Sure:

- The file is executable (chmod +x)
- The path is correct and accessible to PHP
- Ownership and permissions are properly set

Check result



[Figure 6-12] Requesting LED ON



[Figure 6-13] Requesting LED OFF

Remote Control Using PHP

remote_con.php

```
<?php
        if(isset($_GET['LEDON'])){
                $value = shell_exec("/home/pi/PHP_LEDON");
                echo $value;
        }else if(isset($_GET['LEDOFF'])){
                $value = shell_exec("/home/pi/PHP_LEDOFF");
                echo $value;
        }
?>
```

isset function returns true if the variable exists and is not NULL, otherwise it returns false

$_GET can be used to collect data sent in the URL

shell_exec function is used to execute the commands via shell and return the complete output as a string

Program for human detection sensor

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

#define PIN 2

int main(void)  {

    if(wiringPiSetup() == -1) return 1;
    pinMode(PIN,INPUT);
    printf("%d",digitalRead(PIN));
}
```

```
pi@raspberrypi:~ $ gcc -o PHP_PIR PHP_PIR.c -lwiringPi
pi@raspberrypi:~ $ sudo chown root PHP_PIR
pi@raspberrypi:~ $ sudo chmod +s PHP_PIR
```

[Figure 6-15] Compiling File & Providing PHP Execution Permission

```
-rwsr-sr-x 1 root pi
```

[Figure 6-16] Changed Permissions & Owner

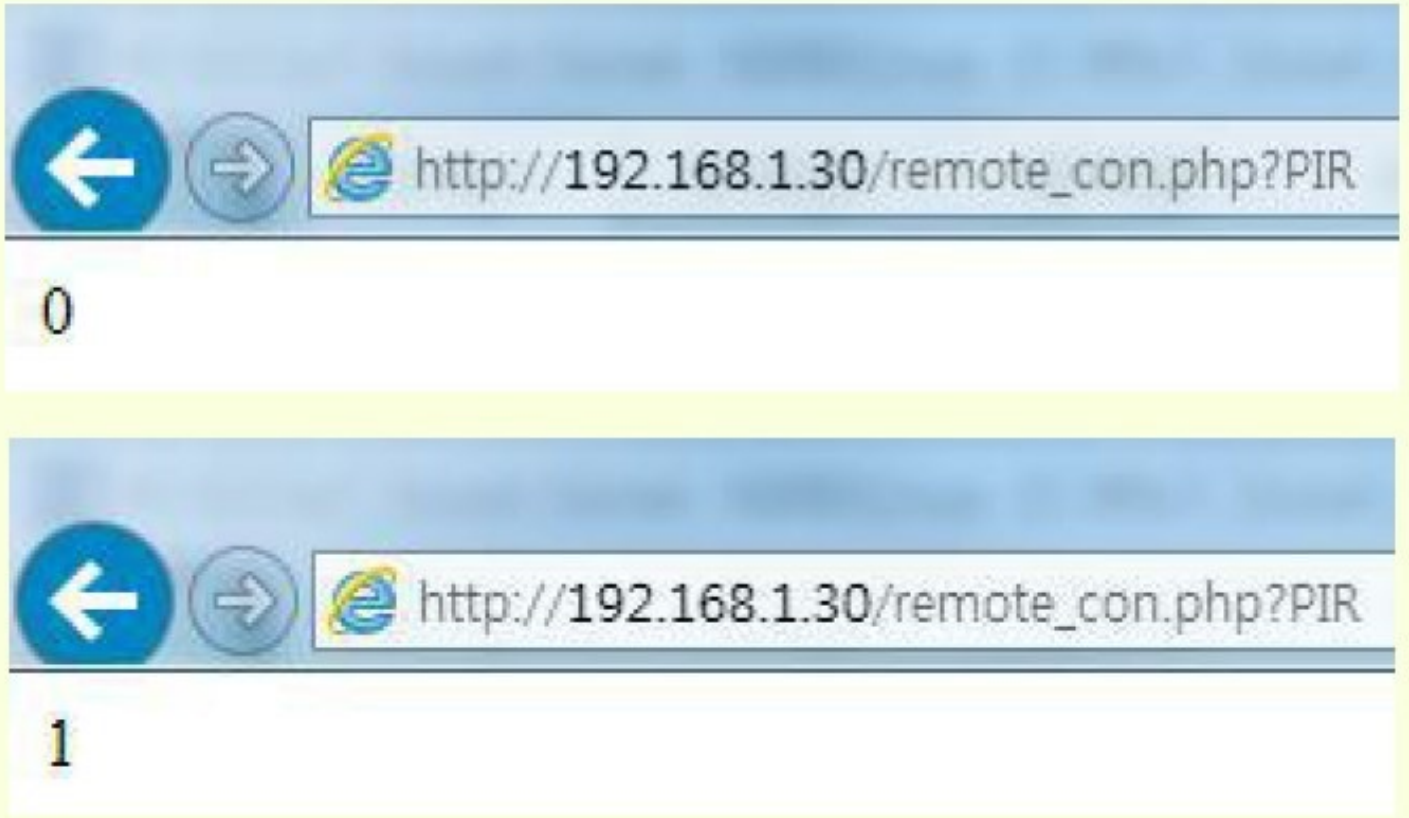
```
pi@raspberrypi:~ $ cd /var/www/html
pi@raspberrypi:/var/www/html $ sudo nano remote_con.php
```

[Figure 6-17] Modifying PHP File

```
<?php
        if(isset($_GET['LEDON'])){
                $value = shell_exec("/home/pi/PHP_LEDON");
                echo $value;
        }else if(isset($_GET['LEDOFF'])){
                $value = shell_exec("/home/pi/PHP_LEDOFF");
                echo $value;
```

```
        }else if(isset($_GET['PIR'])){
                $value = shell_exec("/home/pi/PHP_PIR");
                echo $value;
        }
?>
```

Check result



[Figure 6-18] Requesting the Value of Human Detection Sensor

Program for sound sensor

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

#define SPI_CH 0
#define ADC_CH 2
#define ADC_CS 29
#define SPI_SPEED 500000

int main(void){
    int value=0, i;
    unsigned char buf[3];
    if(wiringPiSetup() == -1) return 1;
    if(wiringPiSPISetup() == -1) return -1;
    pinMode(ADC_CS,OUTPUT);
    buf[0] = 0x06 | ((ADC_CH & 0x04)>>2);
    buf[1] = ((ADC_CH & 0x03)<<6);
    buf[2] = 0x00;
    digitalWrite(ADC_CS,0);
    wiringPiSPIDataRW(SPI_CH,buf,3);
    buf[1]=0x0F & buf[1];
    value = (buf[1]<<8) | buf[2];
    digitalWrite(ADC_CS,1);
    printf("%d",value);
}
```

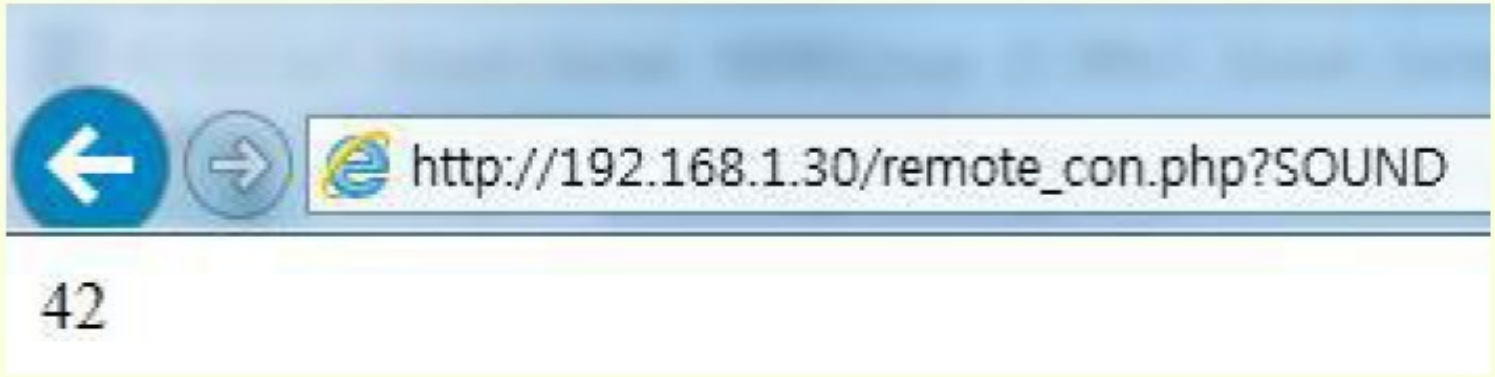
```
pi@raspberrypi:~ $ gcc -o PHP_SOUND PHP_SOUND.c -lwiring
pi@raspberrypi:~ $ sudo chown root PHP_SOUND
pi@raspberrypi:~ $ sudo chmod +s PHP_SOUND
```

[Figure 6-20] Compiling File & Providing PHP Execution Permission

```
-rwsr-sr-x 1 root pi
```

[Figure 6-21] Changed Permissions & Owner

```
        }else  if(isset($_GET['SOUND'])){
                $value  =  shell_exec("/home/pi/PHP_SOUND");
                echo  $value;
        }
?>
```



[Figure 6-23] Requesting the Sensor Value

Program for DC Motor

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

#define PIN 26

int main(void)  {

    if(wiringPiSetup() == -1) return 1;
    pinMode(PIN,OUTPUT);
    digitalWrite(PIN,HIGH);
    printf("1");
}
```

```
pi@raspberrypi:~ $ gcc -o PHP_DCMON PHP_DCMON.c -lwiringPi
pi@raspberrypi:~ $ sudo chown root PHP_DCMON
pi@raspberrypi:~ $ sudo chmod +s PHP_DCMON
```

[Figure 6-34] Compiling File & Providing PHP Execution Permission

```
-rwsr-sr-x 1 root pi
```

[Figure 6-35] Changed Permissions & Owner

Program for DC Motor

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

#define PIN 26

int main(void)  {
    if(wiringPiSetup() == -1) return 1;
    pinMode(PIN,OUTPUT);
    digitalWrite(PIN,LOW);
    printf("0");
}
```

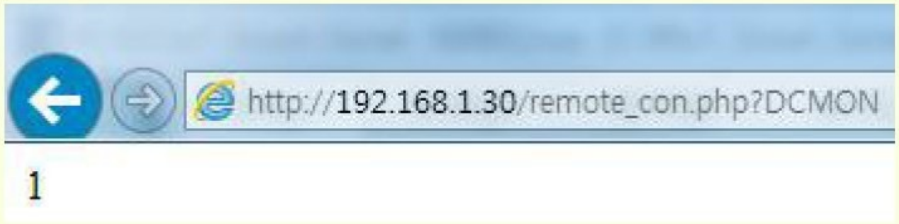
```
pi@raspberrypi:~ $ gcc -o PHP_DCMOFF PHP_DCMOFF.c -lwiringPi
pi@raspberrypi:~ $ sudo chown root PHP_DCMOFF
pi@raspberrypi:~ $ sudo chmod +s PHP_DCMOFF
```

[Figure 6-37] Compiling File & Providing PHP Execution Permission

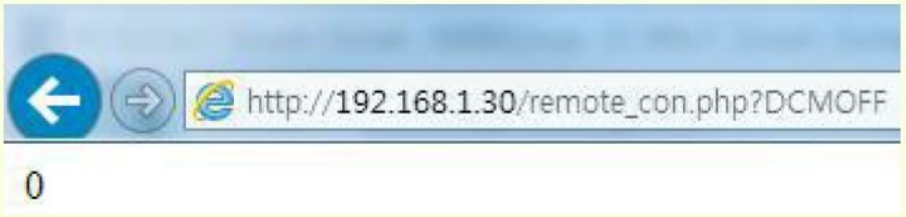
```
-rwsr-sr-x 1 root pi
```

[Figure 6-38] Changed Permissions & Owner

```
else if(isset($_GET['DCMON'])){
        $value = shell_exec("/home/pi/PHP_DCMON");
        echo $value;
}else if(isset($_GET['DCMOFF'])){
        $value = shell_exec("/home/pi/PHP_DCMOFF");
        echo $value;
}
```



[Figure 6-40] Requesting DC Motor ON



[Figure 6-41] RequestingDC motor OFF

Program for Step Motor

```
1   #include <wiringPi.h>
2   #include <stdio.h>
3
4   #define PIN_1A 27
5   #define PIN_1B 0
6   #define PIN_2A 1
7   #define PIN_2B 24
8
9   int main(void)  {
10
11          int i;
12
13          if(wiringPiSetup() == -1) return 1;
14
15          pinMode(PIN_1A,OUTPUT);
16          pinMode(PIN_1B,OUTPUT);
17          pinMode(PIN_2A,OUTPUT);
18          pinMode(PIN_2B,OUTPUT);
```

Continue…

Program for Step Motor

```
20          for(i=0; i<500; i++){
21                  digitalWrite(PIN_1A,HIGH);
22                  digitalWrite(PIN_1B,LOW);
23                  digitalWrite(PIN_2A,LOW);
24                  digitalWrite(PIN_2B,LOW);
25                  usleep(8000);
26                  digitalWrite(PIN_1A,LOW);
27                  digitalWrite(PIN_1B,HIGH);
28                  digitalWrite(PIN_2A,LOW);
29                  digitalWrite(PIN_2B,LOW);
30                  usleep(8000);
31                  digitalWrite(PIN_1A,LOW);
32                  digitalWrite(PIN_1B,LOW);
33                  digitalWrite(PIN_2A,HIGH);
34                  digitalWrite(PIN_2B,LOW);
35                  usleep(8000);
36                  digitalWrite(PIN_1A,LOW);
37                  digitalWrite(PIN_1B,LOW);
38                  digitalWrite(PIN_2A,LOW);
39                  digitalWrite(PIN_2B,HIGH);
40                  usleep(8000);
41          }
42
43          printf("1");
44  }
```

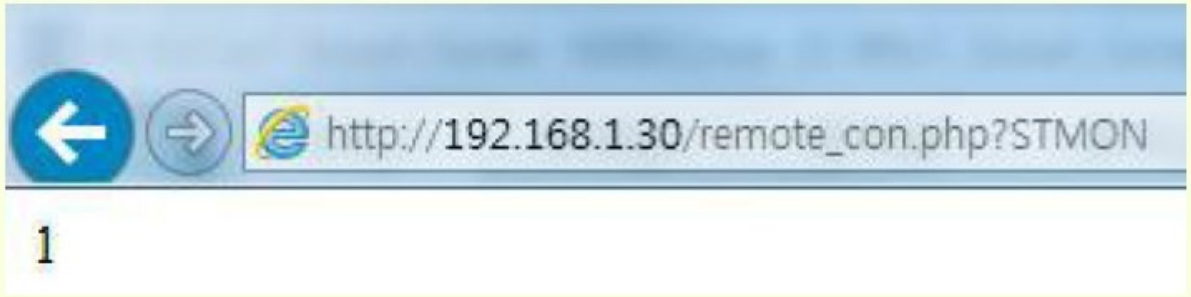
```
pi@raspberrypi:~ $ gcc -o PHP_STMON PHP_STMON.c -lwiringPi
pi@raspberrypi:~ $ sudo chown root PHP_STMON
pi@raspberrypi:~ $ sudo chmod +s PHP_STMON
```

[Figure 6-43] Compiling File & Providing PHP Execution Permission

```
-rwsr-sr-x 1 root pi
```

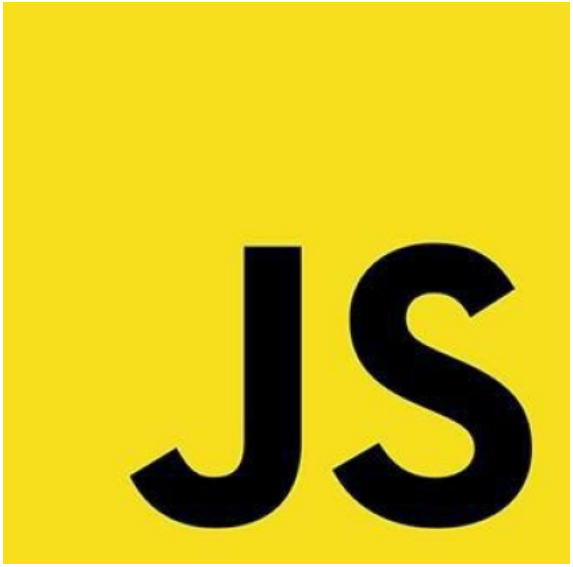
[Figure 6-44] Changed Permissions & Owner

```
else  if(isset($_GET['STMON'])){
            $value  =  shell_exec("/home/pi/PHP_STMON");
            echo  $value;
}
```



[Figure 6-46] Requesting Step Motor ON

JavaScript

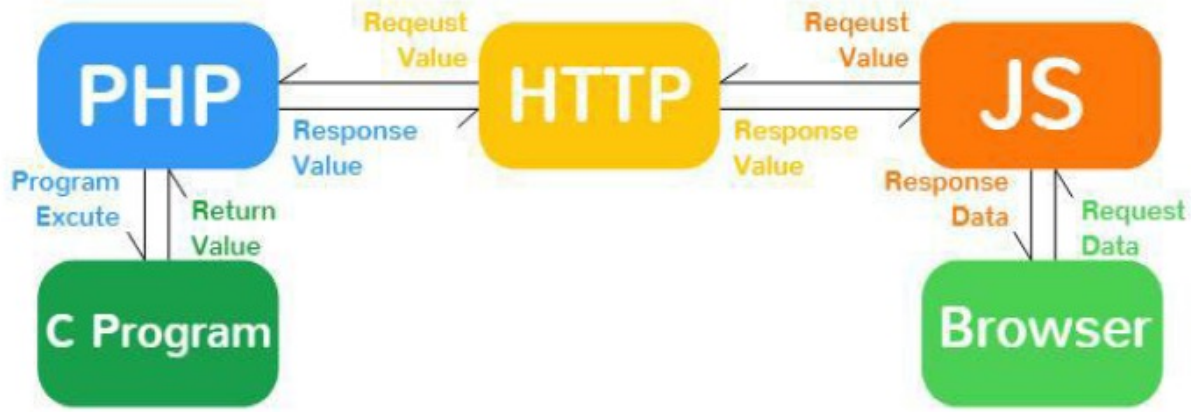


Why Client-Side Script Is Needed?

Without JavaScript: Every time a user clicks a button to turn on an LED, the browser has to **reload the whole page** and send a form to the server.

Feature	Why Client-Side Script (e.g., JavaScript) Is Needed?
Interactivity	Buttons/sliders send GPIO commands instantly
Live Updates	Auto-refresh sensor data without page reload
Form Validation	Catch invalid values before sending them to Pi
AJAX / Fetch	Send/receive data in the background
User Feedback	Show status, loading, or results instantly

AJAX: Asynchronous JavaScript and XML, technique that allows web pages to communicate with a server without refreshing the page.



Web Client

- JavaScript is a lightweight, interpreted programming language
- It is lightweight and most commonly used as a part of web pages

```
<html>
    <body>
        <script language = "javascript" type = "text/javascript">

            document.write("Hello World!")

        </script>
    </body>
</html>
```

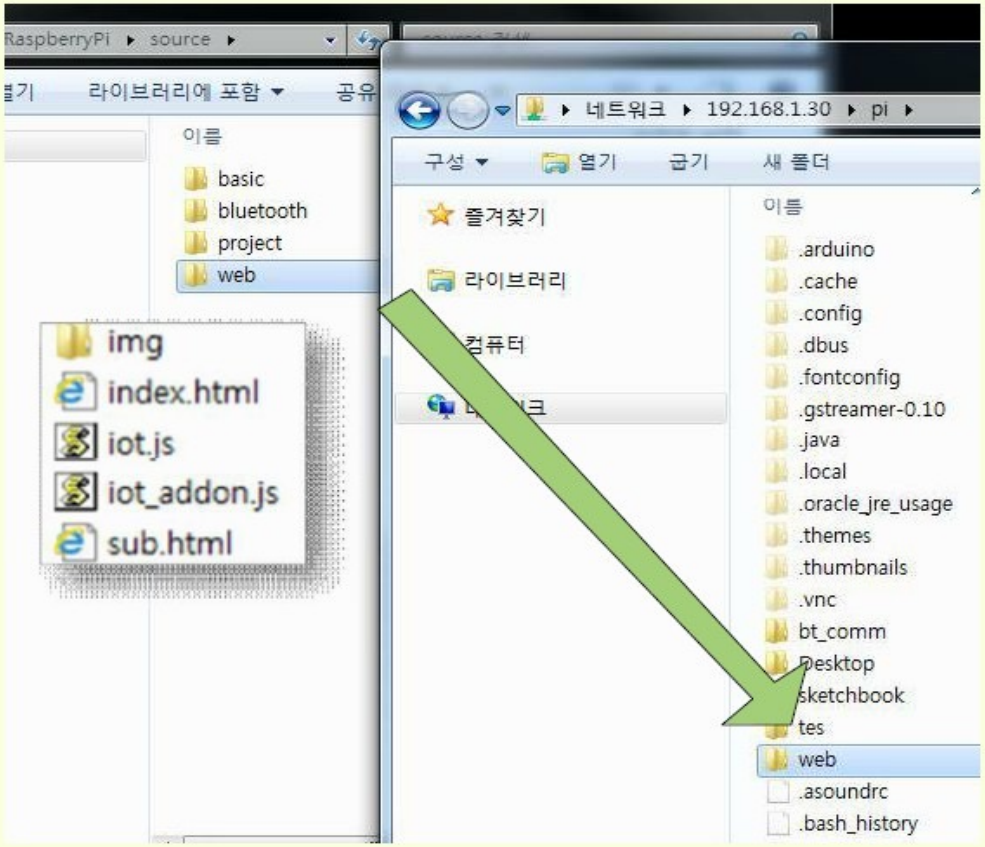
- document.write function writes a string into HTML document

Web Client

- XMLHttpRequest (XHR) is an API in the form of an object whose methods transfer data between a web browser and a web server.
- The object is provided by the browser's JavaScript environment
- To send an HTTP request, create an XMLHttpRequest object, open a URL, and send the request
- After the transaction completes, the object will contain useful information such as the response body and the HTTP status of the result

Term	Type	Description
AJAX	Technique	Asynchronous communication between client & server
XMLHttpRequest	Object	JavaScript object that enables AJAX
Fetch()	Function	Newer way to do AJAX calls in JavaScript

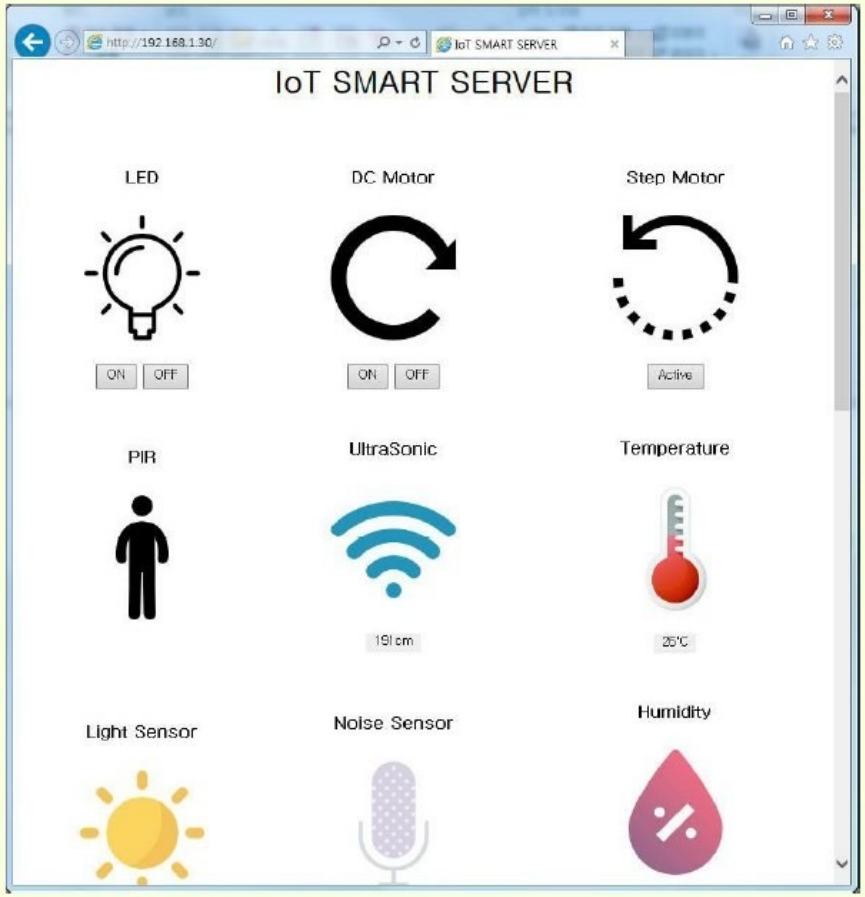
- **Configuring environment**



[Figure 7-4] Web Folder Contents and Copy Process

```
pi@raspberrypi:~ $ cd
pi@raspberrypi:~ $ sudo cp -rf web/. /var/www/html
```

[Figure 7-8] Copy to Web Server Folder



[Figure 7-9] The Screen of http://Raspberry PiIP/

Web Client

- Open index.html

```
pi@raspberrypi:~ $ cd /var/www/html
pi@raspberrypi:/var/www/html $ vi index.html
```

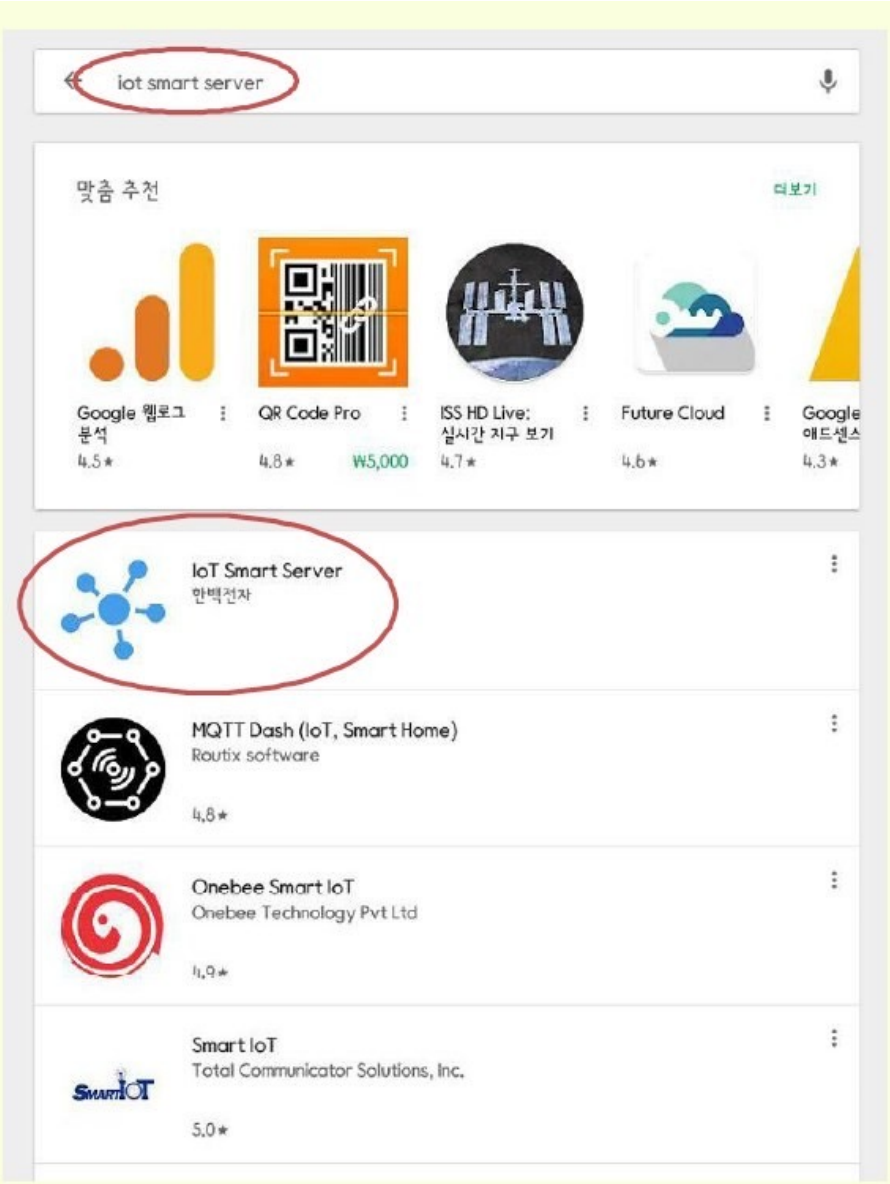
[Figure 7-10] Opening Files in Raspberry Pi

Web Client

- Open iot.js

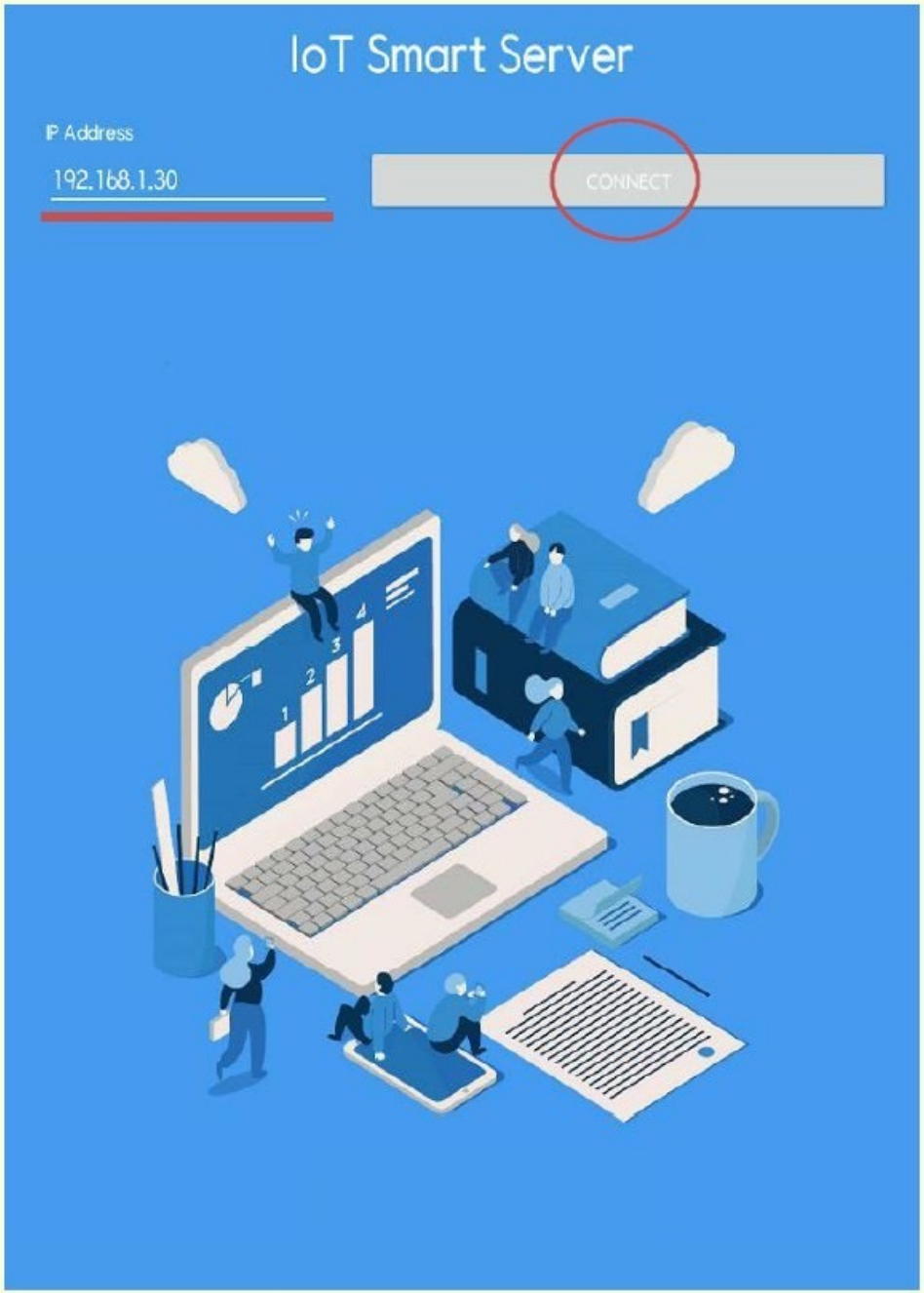
```
pi@raspberrypi:/var/www/html $ vi iot.js
```

[Figure 7-13] Opening File in Raspberry Pi (vi editor)

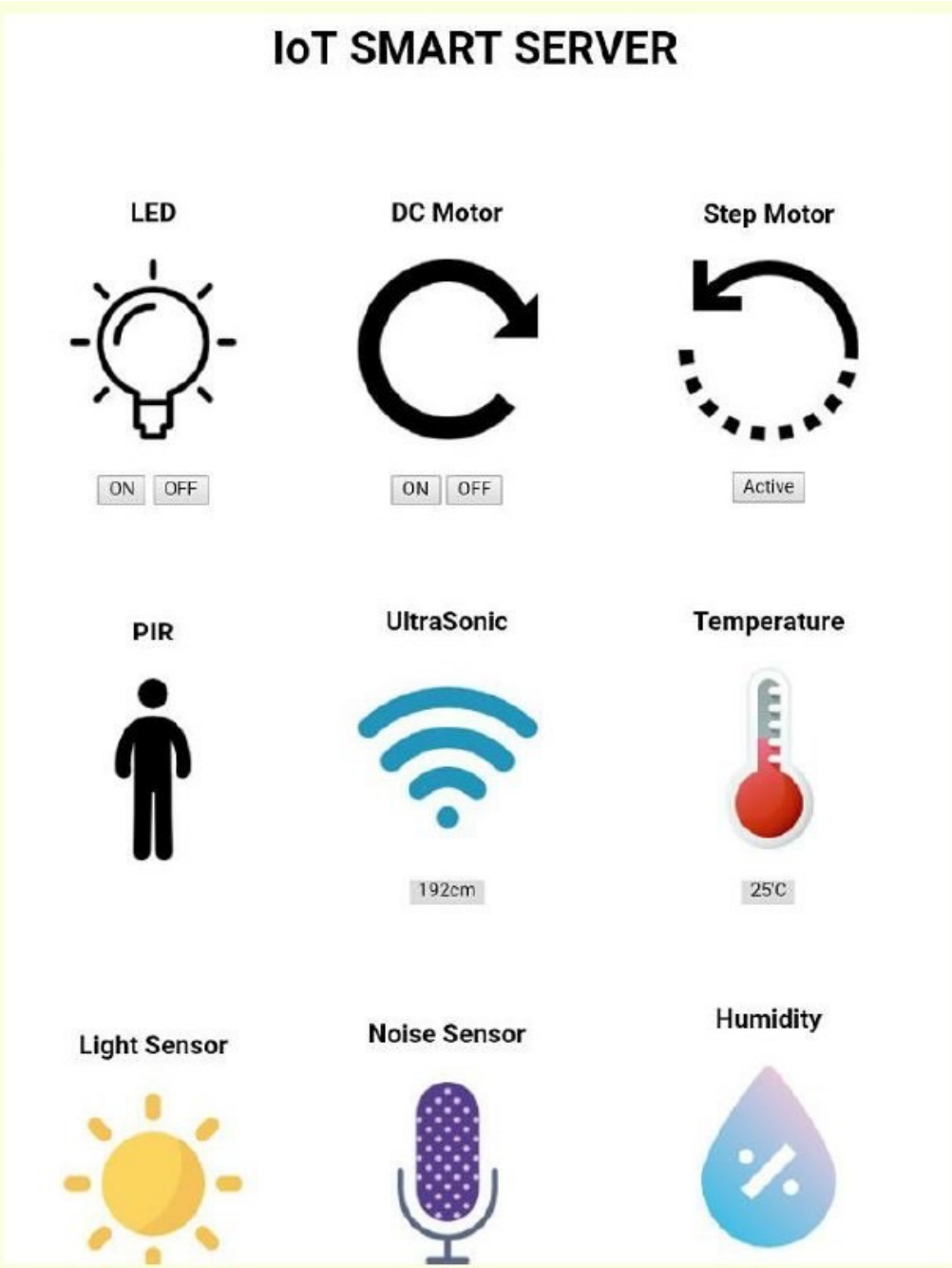


[Figure 7-16] Installing IoT Smart Server

- Run IoT Smart Server App



[Figure 7-17] The Screen of IoT Smart Server



[Figure 7-18] The Screen of IoT Smart Server

〈Table 7-1〉 The Connection Information of 20 Kinds of Main Module

Module	Module Pin Number	Wiring Pi Pin Number	Raspberry Pi Pin Number
LED	LED	7	4
PIR	PIR	2	27
Sound Sensor	SND	ADC2	
Buzzer	BUZZER	15	14
DC Motor	INA	26	12
Step Motor	1A	27	16
	1B	0	17
	2A	1	18
	2B	24	19
Switch Module	SW	3	22
Light Sensor	CdS	ADC0	
Ultrasonic	TRIG	28	20
	OUT	29	21
DHT11	DHT11	25	26
Variable Resistance	VR	ADC1	
Touch Sensor	TOUCH	6	25
Optocoupler	IR	4	23
Bump Sensor	SHOCK	5	24
PSD Sensor	PSD	ADC3	
Laser	LASER	22	6
Mercury Sensor	MERCURY	11	7
Tilt Sensor	TILT	10	8
Flame Sensor	FLAME	16	15
Reed Sensor	REED	9	3

- LED Control Interface

index.html (19~24 line)

```
19 <td><center>
20         <h3>LED</h><br><br>
21         <img src="img/led_off.png" name="LED"><br><br>
22         <input type="button" value="ON" id="LED_ON" onclick="LEDON();">
23         <input type="button" value="OFF" id="LED_OFF" onclick="LEDOFF();">
24 </center></td>
```

iot.js

```
24 function LEDON(){
25         XHR_write('LEDON');
26
27         document.LED.src='img/led_on.png';
28 }
29
30 function LEDOFF(){
31         XHR_write('LEDOFF');
32
33         document.LED.src='img/led_off.png';
34 }
```

- DC Motor Control Interface

```
26 <td><center>
27         <h3>DC Motor</h><br><br>
28         <img src="img/dcm_off.png" name="DCM"><br><br>
29         <input type="button" value="ON" id="DCM_ON" onclick="DCMON();">
30         <input type="button" value="OFF" id="DCM_OFF" onclick="DCMOFF();">
31 </center></td>
```

```
36 function DCMON(){
37         XHR_write('DCMON');
38
39         document.DCM.src='img/dcm_on.png';
40 }
41
42 function DCMOFF(){
43         XHR_write('DCMOFF');
44
45         document.DCM.src='img/dcm_off.png';
46 }
```

Any Questions!