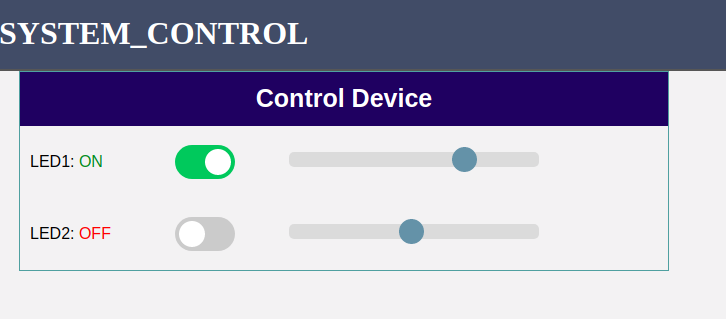
**BÀI 2: Sử dụng Web điều khiển,tăng giảm độ sáng led.**

## Mục tiêu

* Sử dụng arduno để bật tắt,tăng giảm độ sáng led.
* Sử dụng javascript xử lý tác vụ giữa web và arduno.
* Sử dụng HTML/CSS để tạo một giao diện Web đơn giản điều khiển led.
* Tạo được giao diện web điều khiển thiết bị:



## 2.Nội dung

### 2.1 Arduno

-Tạo file bai2.ino là file chứa code arduno.

* **Thêm thư viện,khai báo chân sử dụng:** (ở đây mình sử dụng esp 8266 nodemcu v3)

|  |
| --- |
| 1. #include <ESP8266WiFi.h> 2. #include <ESP8266WebServer.h> 3. #include "index.h" //file html 4. #define dhtPin D1 5. #define lightPin A0 6. #define button1\_pin 14 //D5 7. #define button2\_pin 12 //D6 8. const char\* ssid = "ThuvienPTIT"; 9. const char\* password = "12345678"; 10. ESP8266WebServer server(80); |

-sssid,password:tài khoản mật khẩu wifi kết nối điều khiển.

-Nối led1 với chân D5,led 2 với chân D6 của esp.

* **Hàm xử lý button:**

**-**thêm đoạn code sau tiếp vào file bai2.ino

|  |
| --- |
| void handle\_button() {  String button\_state = server.arg("button\_state");//chuỗi giá trị trả về từ javascript  String curr\_state = "OFF";  if(button\_state == "led1ON"){  digitalWrite(button1\_pin, HIGH);  Serial.println(button\_state);  }  if(button\_state == "led1OFF"){  digitalWrite(button1\_pin, LOW);  Serial.println(button\_state);  }  if(button\_state == "led2ON"){  digitalWrite(button2\_pin, HIGH);  }  if(button\_state == "led2OFF"){  digitalWrite(button2\_pin, LOW);  }  server.send(200, "text/plane", curr\_state);  } |

-Xử lý button ON OFF gửi từ javascript,vd:led1ON (giá trị chuỗi trả về từ javascript)

* **Hàm xử lý tăng giảm độ sáng** (xử lý độ rộng xung:purse width modul)

|  |
| --- |
| void handle\_pwm() {  String pwm\_pin = server.arg("PWNPin");  //nhận tín hiệu chuỗi send từ javascript,sử dụng PWM1 hay PWM2  String pwm\_val = server.arg("PWNval");  //nhận tín hiệu xung để tăng giảm từ javascript  int val = pwm\_val.toInt(); //băm xung sang kiểu int  if(pwm\_pin == "PWM1"){  analogWrite(button1\_pin, val);  Serial.println(val);  }  if(pwm\_pin == "PWM2"){  analogWrite(button2\_pin, val);  Serial.println(val);  }  server.send(200,"text/plane","0"); } |

-Kiểm tra pwm\_pin rồi ghi giá trị val vào chân D5 để tăng giảm độ sáng của led.

* **Thiết lập hàm setup và loop:**

-Trước tiên ta thêm hàm handle\_root để xử lý và gửi dữ liệu lên file html

|  |
| --- |
| void handle\_root() { //tao ra 1 giao dien web  String s = html\_page; //gia tri thu vien index.h tra ve  server.send(200, "text/html", s); //gửi lên web  } |

-Bắt đầu thiết lập hàm setup

|  |
| --- |
| void setup(void){  Serial.begin(115200);  WiFi.begin(ssid, password); //bắt đầu kết nối  pinMode(button1\_pin,OUTPUT);  pinMode(button2\_pin,OUTPUT);  while (WiFi.status() != WL\_CONNECTED) { //loading connect to wifi.  delay(500);  Serial.print(".");  }  Serial.println();  Serial.print("IP address: ");  Serial.println(WiFi.localIP()); //In ra địa chỉ chỉ IP để truy cập web  server.on("/", handle\_root);  server.on("/setPWM", handle\_pwm);  server.on("/setButton", handle\_button);  server.begin();  } |

**2.2 Xử lý html và javascript.**

**-**Tạo file index.h chứa html,javascript

-Đoạn html nằm trong const char html\_page[] PROGMEM = R"==(đoạn html)

* T**ạo table html có nội dung chính như sau:**

|  |
| --- |
| <tr>  <td>  <p>LED1: <span id="led1">OFF</span></p>  </td>  <td>  <label class="switch">  <input type="checkbox" name="button" value="led1" onchange="state\_change(this)">  <!--state\_change để kiểm tra led,khi nhấn button sẽ gọi hàm state train -->  <span class="slider1 round"></span>  </label>  </td>  <td>  <input type="range" min="0" max="255" value="125" class="slider">  <span id="PWM1"></span>  </td>  </tr>  <tr>  <td>  <p>LED2: <span id="led2">OFF</span></p>  </td>  <td>  <label class="switch">  <input type="checkbox" name="button" value="led2" onchange="state\_change(this)">  <span class="slider1 round"></span>  </label>  </td>  <td>  <input type="range" min="0" max="255" value="125" class="slider">  span id="PWM2"></span>  </td>  </tr> |

-thẻ input:khi nhấn vào button sẽ thực hiện onchange.

-thẻ input tăng giảm độ sáng có giá trị min=0,max=255,mặc định value=125.

-Các bạn tự căn chỉnh css để trang web được đẹp hơn,có thể tham khảo css ở cuối paper.

* **Script thực hiện thay đổi giá trị value khi trượt thanh.**

-Thêm đoạn script sau vào file html:

|  |
| --- |
| <script>  var slider = document.getElementsByClassName("slider");  var output0 = document.getElementById("PWM1");  var output1 = document.getElementById("PWM2");  slider[0].onchange = function () {  pwm\_change(output0.id, this.value);  }  slider[1].onchange = function () {  pwm\_change(output1.id, this.value);  }  </script> |

* + **Script xử lý chuỗi và gửi giá trị xung và led được chọn hàm handle\_pwm(arduno)**:

|  |
| --- |
| <script>  function pwm\_change(name, val) {  var xhttp = new XMLHttpRequest(); //request from html  xhttp.open("GET", "setPWM?PWNPin=" + name + "&PWNval=" + val, true);  xhttp.send(); //gửi pwn  }  </script> |

-Thực hiện nối chuỗi name+val:

* **Hàm thực hiện onchange khi nhấn vào button** **và gửi checked về esp:**

|  |
| --- |
| <script>//on off  function state\_change(element) { //thay đổi trang thái khi ấn on off  var xhttp = new XMLHttpRequest();  if (element.checked) {  xhttp.open("GET", "setButton?button\_state=" + element.value + "ON", true);  document.getElementById(element.value).innerHTML = "ON";  document.getElementById(element.value).style.color = "green";  } else if (!element.checked) {  xhttp.open("GET", "setButton?button\_state=" + element.value + "OFF", true);  document.getElementById(element.value).innerHTML = "OFF";  document.getElementById(element.value).style.color = "red";  }  xhttp.send();  }</script> |

xhttp.open("GET", "setButton?button\_state=" + element.value + "ON", true);

-hàm này thực hiện nối chuỗi VD:led1+ON là đúng và gửi về esp để thực hiện điều khiển.

-document.getElementById(element.value).style.color = "green"

-hàm này thực hiện đổi mầu button khi đã check (nằm trong if kiểm tra).

* Có thể tham khảo thêm đoạn css sau:

|  |
| --- |
| .slidecontainer {  width: 100%;  }  .slider {  -webkit-appearance: none;  width: 250px;  height: 15px;  border-radius: 5px;  background: #d3d3d3;  outline: none;  opacity: 0.7;  -webkit-transition: .0s;  transition: opacity .2s;  }  .slider:hover {  opacity: 1;  }  .slider::-webkit-slider-thumb {  -webkit-appearance: none;  appearance: none;  width: 25px;  height: 25px;  border-radius: 50%;  background: #326C88;  cursor: pointer;  }  .slider::-moz-range-thumb {  width: 25px;  height: 25px;  border-radius: 50%;  background: #326C88;  cursor: pointer;  }  /\* Rounded switch \*/  .switch {  position: relative;  display: inline-block;  width: 60px;  height: 34px;  }  .switch input {  opacity: 0;  width: 0;  height: 0;  }  .slider1 {  position: absolute;  cursor: pointer;  top: 0;  left: 0;  right: 0;  bottom: 0;  background-color: #ccc;  -webkit-transition: .4s;  transition: .4s;  }  .slider1:before {  position: absolute;  content: "";  height: 26px;  width: 26px;  left: 4px;  bottom: 4px;  background-color: white;  -webkit-transition: .2s;  transition: .2s;  }  input:checked + .slider1 {  background-color: #3bbb52;  ;  }  input:focus + .slider1 {  box-shadow: 0 0 1px #326C88;  }  input:checked + .slider1:before {  -webkit-transform: translateX(26px);  -ms-transform: translateX(26px);  transform: translateX(26px);  }  .slider1.round {  border-radius: 34px;  }  .slider1.round:before {  border-radius: 50%;  }  \* {  margin: 0px;  padding: 0px;  }  #header {  width: 100%;  height: 70px;  background-color: #404e67;  border-bottom: 2px solid #555555;  }  #header img {  padding: 5px 10px 0px 30px;  float: left;  }  h1 {  padding-top: 16px;  color: #fff;  float: left;  }  #wrap {  width: 100%;  height: 590px;  line-height: 1.5;  background-color: #f3f3f3;  font-family: open sans, sans-serif;  }  #container {  width: 100%;  height: 120px;  padding-top: 20px;  padding-bottom: 20px;  }  .card-block {  width: 280px;  height: 100px;  color: #fff;  margin-left: 20px;  border-radius: 5px;  float: left;  /\* box-shadow: 2px 2px 5px grey; \*/  }  #temp {  background: #c56840;  }  #humi {  background: #729bf3;  }  #light {  background: #dfdd7c;  }  .card-block p {  padding-left: 10px;  font-size: 17px;  border-bottom: 1px solid #f3f3f3;  }  .card-block h4 {  padding-left: 10px;  font-size: 50px;  }  table {  width: 50%;  height: 200px;  margin-left: 20px;  float: left;  border: 1px solid cadetblue;  border-collapse: collapse;  }  table tr th {  font-size: 25px;  color: #fff;  background-color: #0B0B61;  }  table tr td {  padding: 10px 10px;  }  .button {  background-color: #555555;  border: 2px solid #555555;  color: #fff;  padding: 16px 32px;  text-align: center;  text-decoration: none;  font-size: 18px;  margin: 50px 120px;  -webkit-transition-duration: 0.4s;  transition-duration: 0.4s;  cursor: pointer;  }  </style> |

|  |
| --- |
| *Designed By HaNhuThai* |