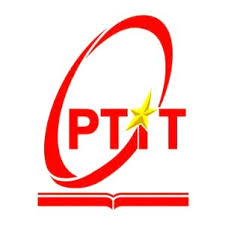
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA KĨ THUẬT ĐIỆN TỬ I**

-----🙞🙜🕮🙞🙜-----



**BÁO CÁO BÀI TẬP LỚN**

**BỘ MÔN: XÂY DỰNG HỆ THỐNG SỐ**

**ĐỀ TÀI:**

**ĐIỀU KIỂN THIẾT BỊ LED VÀ HIỂN THỊ TRẠNG THÁI LED VÀ CÁC THÔNG SỐ CẢM BIẾN NHIỆT ĐỘ, ĐỘ ẨM, KHÍ GAS, BẰNG WEBSERVER, SỬ DỤNG ESP32**

Giảng viên hướng dẫn: TS. Nguyễn Quốc Uy

Sinh viên thực hiện: Nguyễn Văn Đức

Nguyễn Công Minh

Vũ Văn Hậu

Nguyễn Việt Hà

Khóa: D15 KTDT

Hà nội – 2019

MỤC LỤC

PHẦN I: MỞ ĐẦU

GIỚI THIỆU ĐỀ TÀI

PHẦN II: NỘI DUNG

Chương 1:

Chương 2:

Chương 3:

Chương 4: Xây dựng webserver, giao diện hiển thị, giao diện điều khiển LED

## Xây dựng webserver

#### Xây dựng server local bằng nodejs:

#### cài các package cần thiết: express, ejs, socket.io, mqtt, mysql.

#### express: là frameword giúp nodejs tạo trang web với phương thức http và tạo ra một API vô cùng mạnh mẽ.

#### ejs: là 1 view engine giúp chúng ta có thể viết code HTML dễ dàng hơn trong nodejs

#### socket.io: là một thư viện JavaScript cho các ứng dụng web thời gian thực.

#### mqtt: giúp trung ta pub sub dữ liệu giữa server và các client.

#### mysql: giúp chúng ta trung cập vào mysql local từ server.

#### Code:

|  |
| --- |
| **var** express = require("express");  **var** app = express();  **var** server = require("http").Server(app);  **var** io =require("socket.io")(server);  **var** mysql = require('mysql');  **var** mqtt = require('mqtt');  // SERVER  app.use(express.static("public"));  app.set("view engine", "ejs");  app.set("views","./views");  server.listen(3000, **function**(){  console.log('http://localhost:3000/');  });  // WEB  app.get("/", **function**(req, res){  res.render("home");  });  // socket.io CONTROL LED  io.on("connection",**function**(socket){  console.log("User: " + socket.id + " connection!");  socket.on("disconnect", **function**(){  console.log("User: " + socket.id + " disconnected!");  });  socket.on("control-led",**function**(data){  publish(toppic1,data.toString(),options);  **if**(data==1){  publish(toppic2,'LED ON',options);  }  **else** **if**(data==0){  publish(toppic2,'LED OFF',options);  }  **else**{  **var** b = 'Do Sang ' + Math.round(data\*100/255) + '%';  publish(toppic2,b.toString(),options);  }  })  })  // MQTT  **var** options = {  username: 'ducmqtt',  password: 'duc'  }  **var** client = mqtt.connect('mqtt://localhost:1883/', options);  **var** toppic1 = 'Toppic 1';  **var** toppic2 = 'controlled';  **function** **publish**(topic,msg,options){  **if** (client.connected == true){  console.log("publishing",msg);  client.publish(topic,msg,options);  }pronhat  }  // MYSQL  **var** con = mysql.createConnection({  host: "localhost",  user: "ducsql",  password: "duc",  database: "ducdb"  });  con.connect(**function**(error){  **if**(!!error){  console.log('Error connect to Mysql!');  }**else**{  console.log('Connected to Mysql!');  }  }); |

## Xây dựng giao diện hiển thị nhiệt độ và độ ẩm

#### dùng trường hàm setIenterval để thức hiện thay đổi background màu của phần hiển thị nhiệt độ độ ẩn, có thể code thay đổi màu tuỳ ý, và chọn thời gian để thay đổi, trạng thái, trong code ở dưới được đặt là 1000ms = 1s.

|  |
| --- |
| // mau background thanh doi theo gia tri  setIenterval(**function**(){  $.getJSON( '/dataroom1', function( **a** ) {  var temp = document.getElementById('nhietdo');  var hum = document.getElementById('doam');  var dust = document.getElementById('dobui');  if(**a**.result.Temperature <= 20){  temp.style.background = "#00ffed";  }  if(**a**.result.Temperature > 20 && a.result.Temperature <= 30){  temp.style.background = "#00ff7f";  }  if(**a**.result.Temperature > 30 && a.result.Temperature <= 40){  temp.style.background = "#ff5400";  }  if(**a**.result.Temperature > 40){  temp.style.background = "#ff0000";  }  if(**a**.result.Humidity <= 50){  hum.style.background = "#c7e3ff";  }  if(**a**.result.Humidity > 50 && a.result.Humidity <= 85){  hum.style.background = "#8585ff";  }  if(**a**.result.Humidity > 85){  hum.style.background = "#cf3dff";  }  if(**a**.result.Dust <= 30){  dust.style.background = "#c2d6d6";  }  if(**a**.result.Dust > 30 && a.result.Dust <= 50){  dust.style.background = "#b2b299";  }  if(**a**.result.Dust > 50){  dust.style.background = "#6e6e3d";  }  });  },1000); |

## 3. Giao diện và Điều khiển Led

### Giao diện của bóng đèn và nút ấn

#### dùng phương thức getlementById trong javascrip để thay đổi màu của bóng đèn theo các mức độ sáng tương ứng. ở đây là thay đổi màu sắc bằng hình ảnh.

|  |
| --- |
| // chuyen mau bóng den  **function** **getvalue**(){  **var** a = document.getElementById('myRange').value;  **var** b = Math.round(a\*100/255);  **var** hinhanh = document.getElementById("hinhanh");  **var** ledoff = 'img/ledoff.png';  **var** ledon = 'img/ledon.png';  **var** ledon1 = 'img/ledon1.png';  **var** ledon2 = 'img/ledon2.png';  **var** ledon3= 'img/ledon3.png';  **var** ledon4= 'img/ledon4.png';  document.getElementById('demo').innerHTML = a;  document.getElementById('dosang').innerHTML = b;  **if**(0<=b){  hinhanh.src = ledoff;  }  **if**(0<b && b<=20){  hinhanh.src = ledon1;  }  **if**(20<b && b<=40){  hinhanh.src = ledon2;  }  **if**(40<b && b<=60){  hinhanh.src = ledon3;  }  **if**(60<b && b<=80){  hinhanh.src = ledon4;  }  **if**(80<b && b<=100){  hinhanh.src = ledon;  }  }  // hieu ung cua buttom led  var **on** = 'img/ledon.png';  var off = 'img/ledoff.png';  **function** **bat**(){  var onled = document.getElementById("onled");  var offled = document.getElementById("offled");  var hinhanh = document.getElementById("hinhanh");  hinhanh.src = **on**;  onled.style.color = "white";  offled.style.color = "black";  onled.style.background = "green";  offled.style.background = "white";  }  **function** **tat**(){  var onled = document.getElementById("onled");  var offled = document.getElementById("offled");  var hinhanh = document.getElementById("hinhanh");  hinhanh.src = off;  onled.style.color = "black";  offled.style.color = "white";  offled.style.background = "red";  onled.style.background = "white";  } |

### Điều khiển bóng đèn

#### Sử dụng thư viện socket.io trong javascrip để server có thể lắng nghe và bắt được tín hiệu điều hiển từ client(người dùng) một cách ngay lập tức.

|  |
| --- |
| *Function này trờ người dùng ấn nút sau đó báo lên cho server thông quá câu lệnh socket.emit.*  // dieu kien led  **var** socket = io();  $(document).ready(**function**(){  $("#onled").click(**function**(){  socket.emit("control-led","1");  });  $("#offled").click(**function**(){  socket.emit("control-led","0");  });  $("#myRange").change(**function**(){  **var** a = $("#demo").text();  socket.emit("control-led",a);  });  });  *Sau khi nhận được tín hiệu từ người dùng thì server sẽ publish lên 1 topic trên MQTT.*  // socket.io CONTROL LED  io.on("connection",**function**(socket){  console.log("User: " + socket.id + " connection!");  socket.on("disconnect", **function**(){  console.log("User: " + socket.id + " disconnected!");  });  socket.on("control-led",**function**(data){  publish(toppic1,data.toString(),options);  **if**(data==1){  publish(toppic2,'LED ON',options);  }  **else** **if**(data==0){  publish(toppic2,'LED OFF',options);  }  **else**{  **var** b = 'Do Sang ' + Math.round(data\*100/255) + '%';  publish(toppic2,b.toString(),options);  }  })  })  *Sau khi server nhận được tín hiệu từ người dùng và publish lên toppic ở MQTT thì esp sẽ được thực hiện hàm callback này, khi các dữ liệu được server gửi về trùng với yêu cầu bật tắt và điểu chỉnh độ sáng led thì ESP sẽ điều khiển led theo đúng như vậy.*  /\*callback\*/  **void** callback(**char**\* toppic, **byte**\* message, **unsigned** **int** length) {  Serial.print("Topic: ");  Serial.print(toppic);  Serial.print(". Message: ");  **String** messageTemp;  for (**int** i = 0; i < length; i++) {  Serial.print((**char**)message[i]);  messageTemp += (**char**)message[i];  }  Serial.println();  if (**String**(toppic) == "Toppic 1") {  Serial.print("setup led1 ");  if (messageTemp.toInt() == 1) {  ledcWrite(0, 255);  Serial.println("0N");  }  else if (messageTemp.toInt() == 0) {  ledcWrite(0, 0);  Serial.println("OFF");  }  else {  ledcWrite(0, messageTemp.toInt());  }  Serial.println(messageTemp);  }  } |

PHẦN III: KẾT LUẬN