

[Document title]

[Document subtitle]



[Date]

[Company name]

[Company address]

**Design Scenario**

Raining can cause my clothes, my house to be wet with rain.

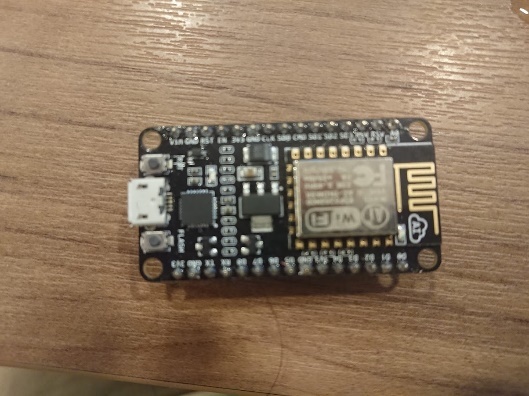
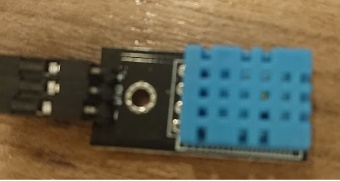
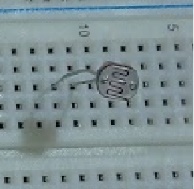
**Motivation of Design**

I saw a Ngee Ann Polytechnic Project on detecting rain and then use a motor to pull the clothes back.

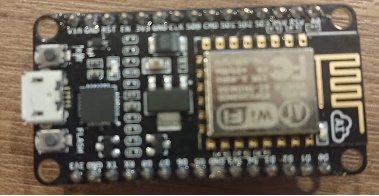
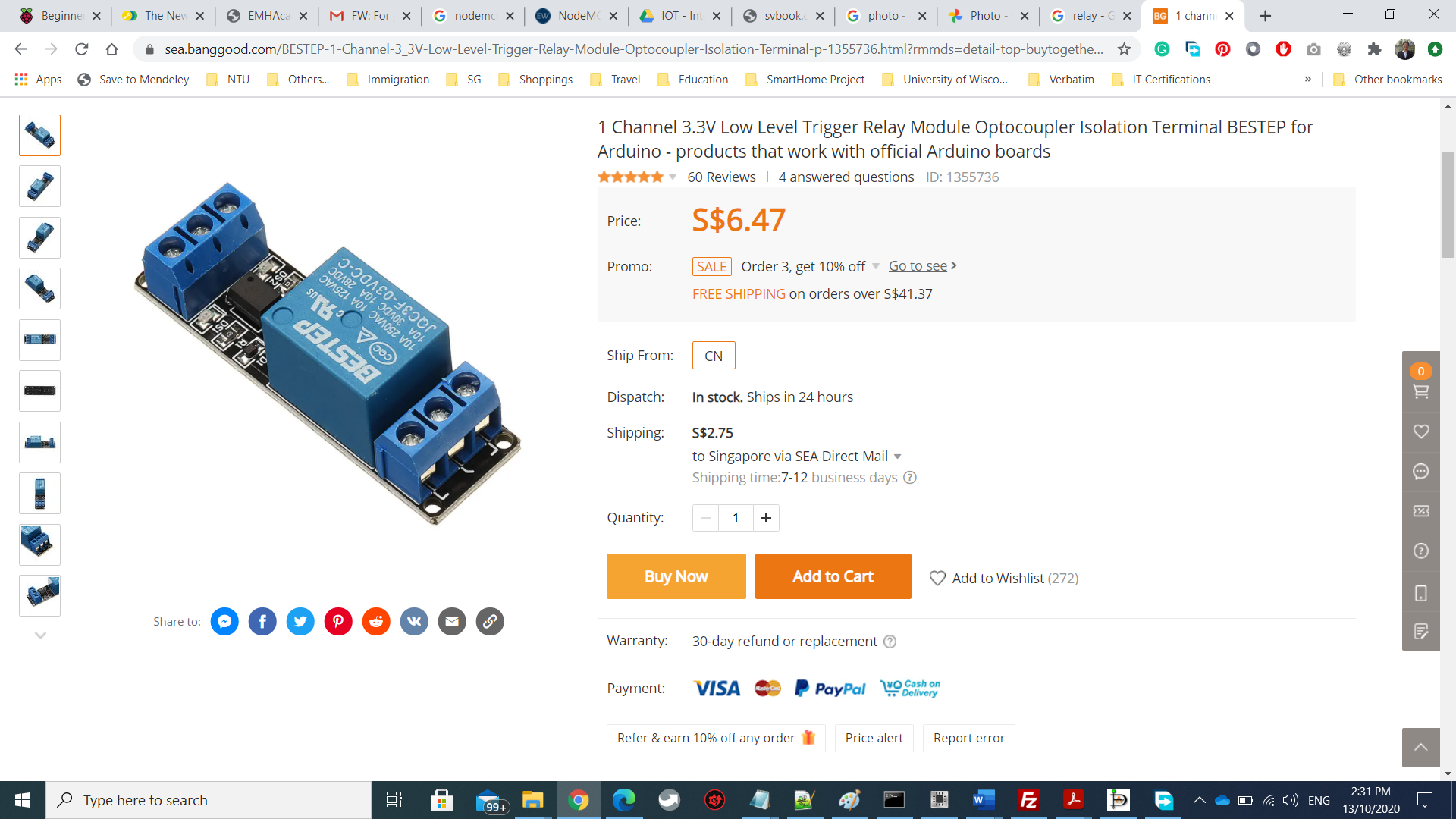
Based on this project, I implement IoT to use sensors to detect light (LDR) and temperature and Humidity (the sensors used in lectures) and detect whether it will be raining. I will then being able to use smartphone to send commands on what to do when it is raining. In class, I learnt about DHT11

**Design Decomposition**

DHT Sensors + LDR + NodeMCU

 +  + 

NodeMCU + switch or relay or motor

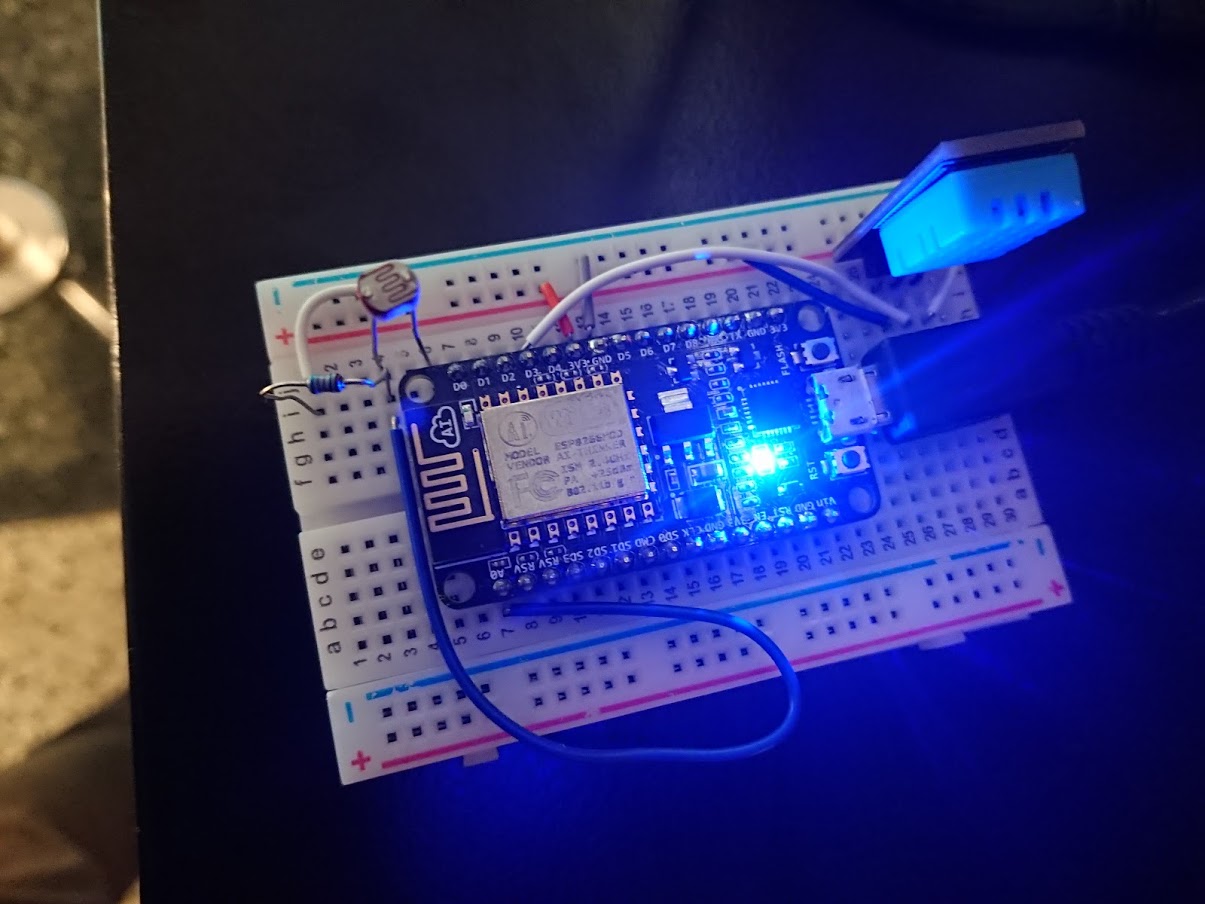
 + 

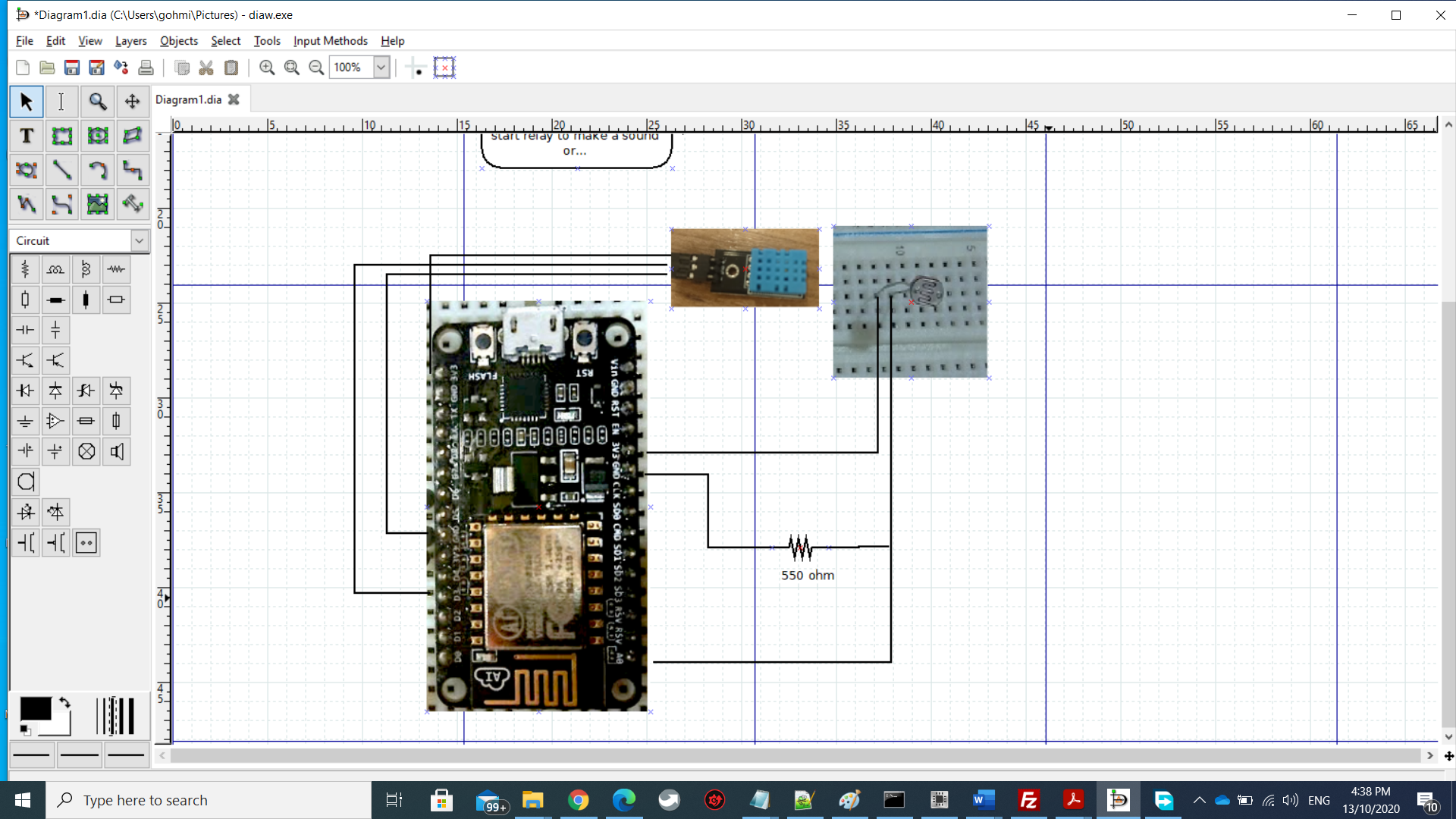
Raspberry Pi for MQTT Server



**Details of Design**

DHT Sensors + LDR + NodeMCU

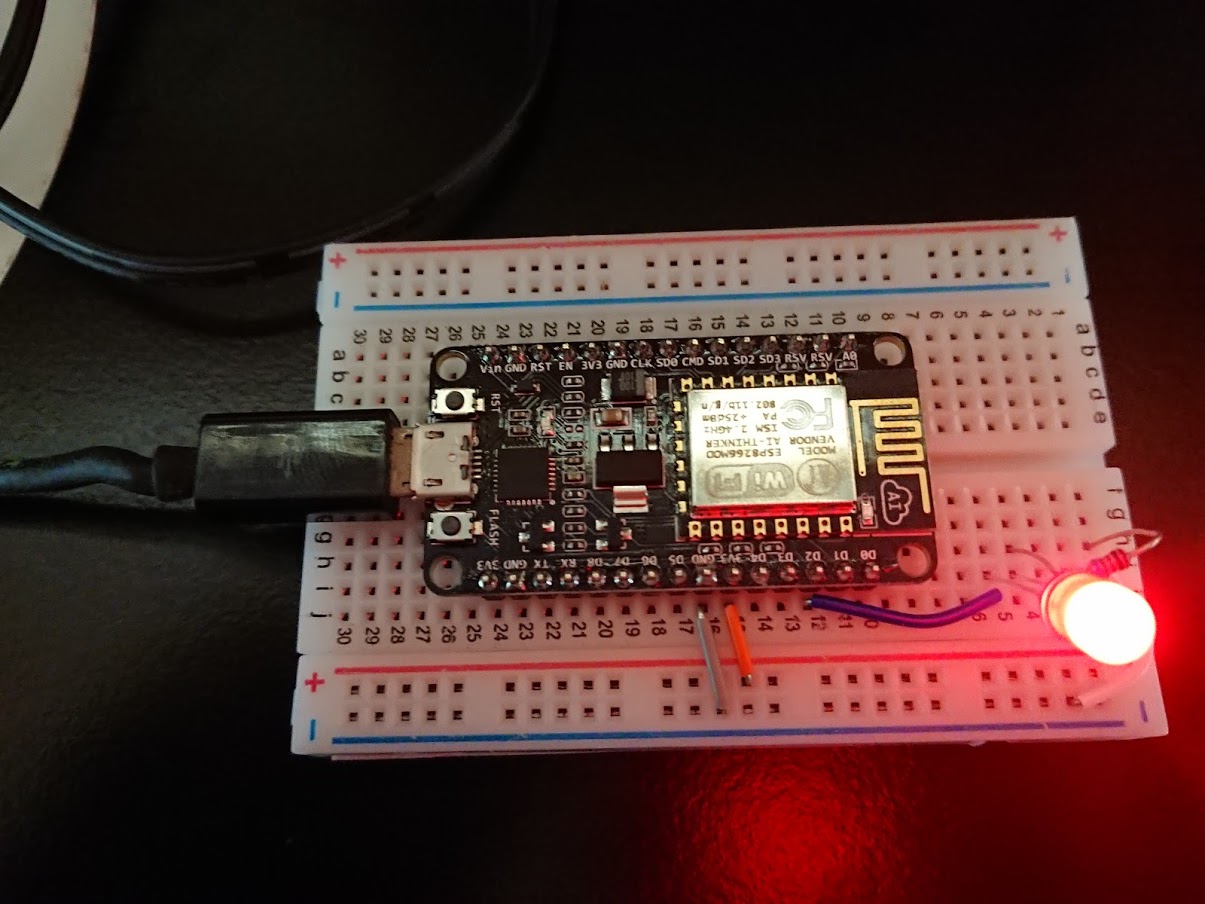


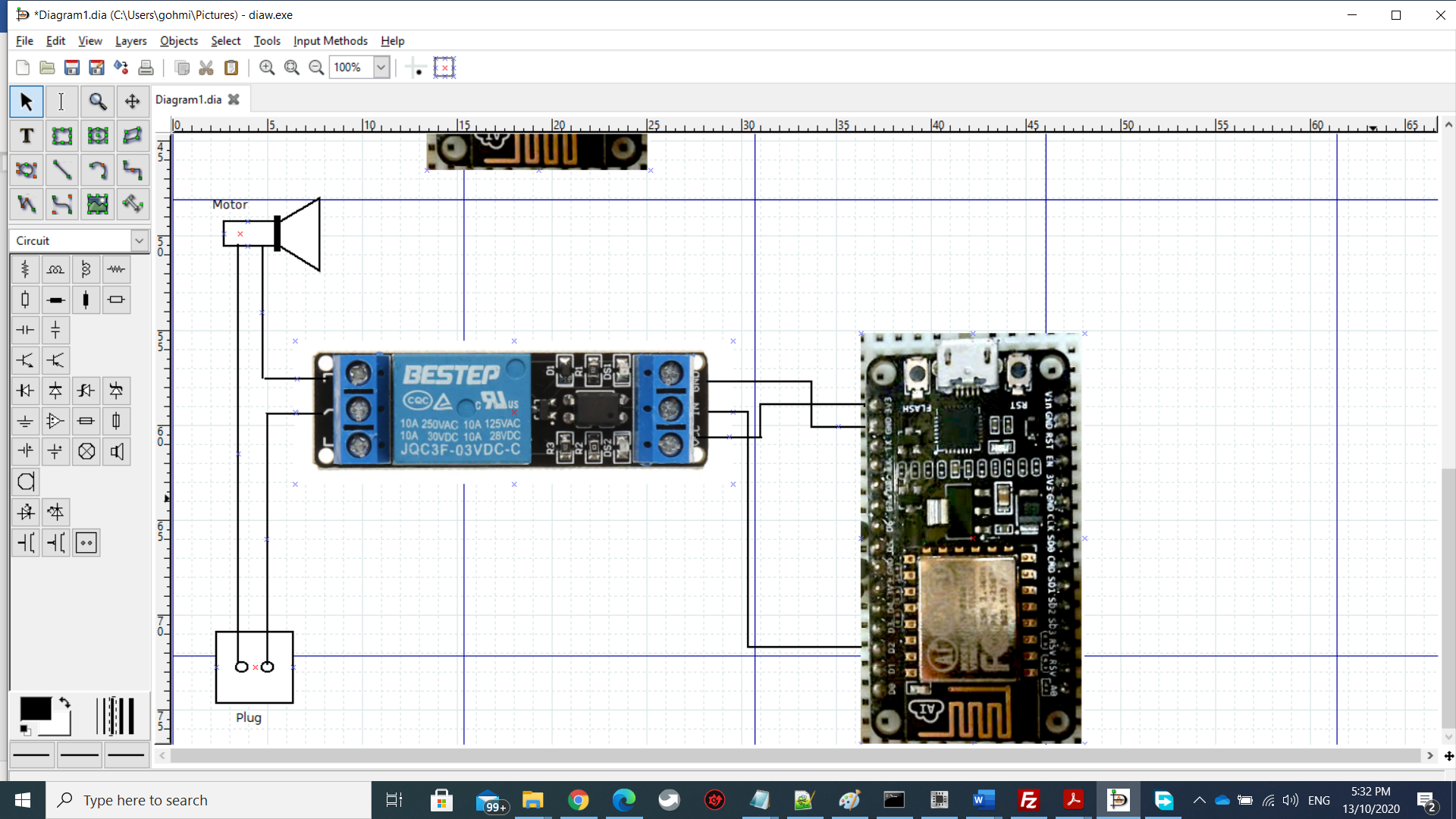


Pseudo Codes

|  |
| --- |
|  |

**NodeMCU + switch or relay or motor**





Pseudo Codes

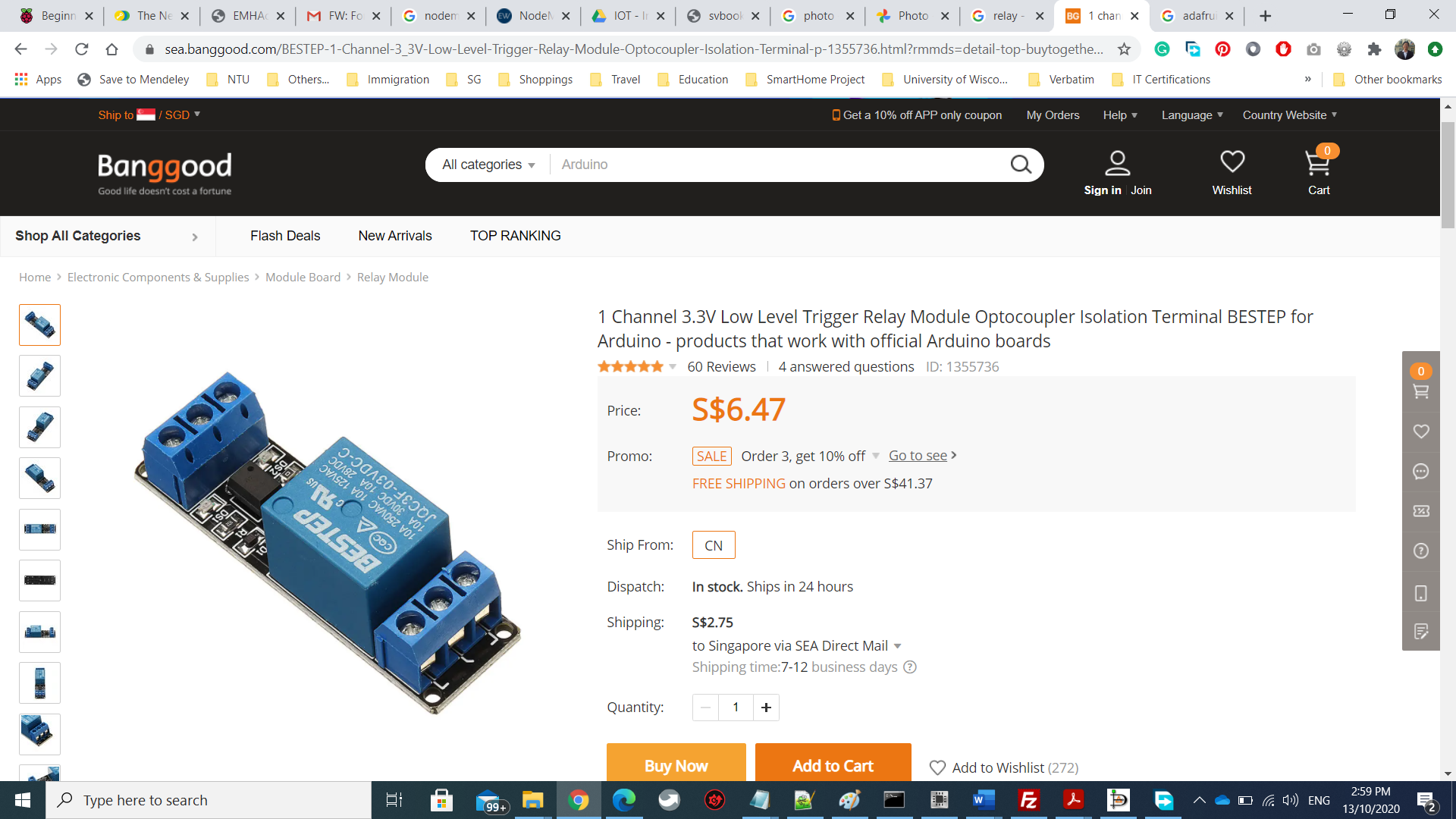
|  |
| --- |
|  |

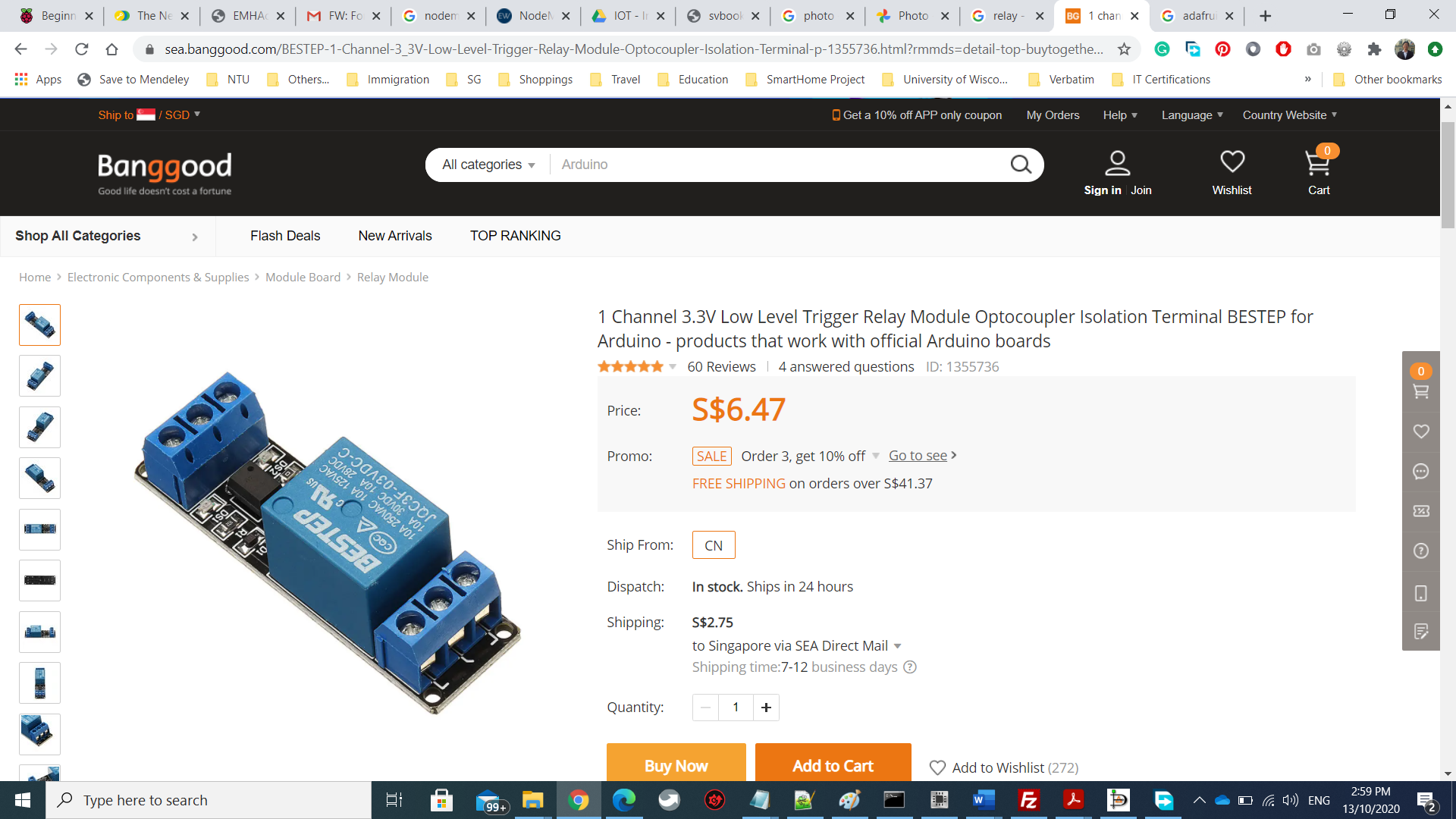
**Comparison**

|  |  |
| --- | --- |
| **NodeMCU** | **Arduino and ESP** |
|  |  |
| Description | Description |

**Justification**

NodeMCU provides 3V3, Arduino provides 5V. To use NodeMCU, the relay must be able to be activated using 3V3. Arduino + ESP need a lot of connections, NodeMCU is cheaper. Chose NodeMCU.





**Description of Resourcses (datasheets, where to buy them…)**

Relay can be found at: <https://sea.banggood.com/BESTEP-1-Channel-3_3V-Low-Level-Trigger-Relay-Module-Optocoupler-Isolation-Terminal-p-1355736.html?rmmds=detail-top-buytogether-auto&cur_warehouse=CN>

NodeMCU can be found at: <https://www.sgbotic.com/index.php?dispatch=products.view&product_id=1857>

**Comparison**

|  |  |
| --- | --- |
| **LDR** | **Other Photoresistors….** |
|  |  |
| Description | Description |

**Justification**

LDR is cheap….

**Description of Resourcses (datasheets, where to buy them…)**

**Comparison**

|  |  |
| --- | --- |
| **DHT** | **Others** |
|  |  |
| Description | Description |

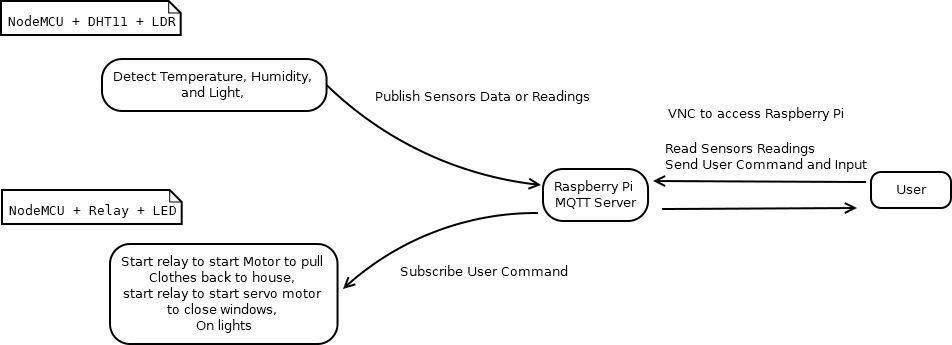
**Justification**

In class….

**Description of Resourcses (datasheets, where to buy them…)**

**Raspberry Pi with MQTT Server for sensors readings**

**Flow Chart**



The NodeMCU + DHT11 + LDR sensors will detect temperature, humidity, light and send it to Raspberry Pi MQTT Server. Raspberry Pi is MQTT server, and Raspberry Pi is the broker. User will receive whether it is raining from Raspberry Pi in smartphone using VNC. Raspberry PI is a PC and VNC access Raspberry Pi as a Remote PC. User then send command to Raspberry Pi, Raspberry Pi then sent to NodeMCU + Relay to start servo motor to pull the clothes or close windows or….

**Cost Estimation**

|  |  |
| --- | --- |
| **Components** | **Price** |
| NodeMCU | 16.50 |
| Relay | 6.47 |
| DHT | 9.90 |
| LDR | 3.21 |
|  | 36.08 |

**Discussion**

**Summary of Design**

…..

**Potential Pitfalls, Advantages and Disadvantages….**

NodeMCU is 3V3, Arduino is 5V. 5V is easier and better.

**Alternative Solutions**

Substitute with Arduino

**Implementation of Your Design**

**Videos**

[**https://drive.google.com/file/d/1uTCclHNaTdk8Qyt9QY0-TBS7NBvaO3ZW/view?usp=sharing**](https://drive.google.com/file/d/1uTCclHNaTdk8Qyt9QY0-TBS7NBvaO3ZW/view?usp=sharing)

**Codes**

|  |
| --- |
| ***NodeMCU + DHT11 + LDR***  #include <PubSubClient.h>  #include <ESP8266WiFi.h>  #include "DHT.h"  #define DHT11PIN D3  #define DHT11TYPE DHT11  DHT dht11(DHT11PIN, DHT11TYPE);  const char\* ssid = "HUAWEI-29D9";  const char\* password = "61031018";  const char\* server = "192.168.8.101";  const char\* topic = "uop/mqtt\_tutorial/LDR";  const char\* topic2 = "uop/mqtt\_tutorial/Temp";  const char\* clientName = "com.dsajhg.nodemcu";  int value;  int percent;  String payload;  WiFiClient wifiClient;  PubSubClient client(wifiClient);  void wifiConnect() {  Serial.println();  Serial.println("Connecting to");  Serial.println(ssid);  WiFi.begin(ssid, password);  while(WiFi.status() != WL\_CONNECTED) {  delay(500);  Serial.print(".");  }  Serial.println("");  Serial.println("WiFi COnnected");  Serial.println("IP Address: ");  Serial.println(WiFi.localIP());  if(client.connect(clientName)) {  Serial.print("Connected to MQTT broker at: ");  Serial.print(server);  Serial.print(" as ");  Serial.println(clientName);  Serial.print("Topic is: ");  Serial.println(topic);    }  else {  Serial.println("MQTT connect failed");  Serial.println("Will reset and try again...");  abort();  }  }  void mqttReConnect() {  while(!client.connected()) {  Serial.print("Attempting MQTT connection...");  if(client.connect(clientName)) {  Serial.println("connected");  client.subscribe(topic);  client.subscribe(topic2);    }  else {  Serial.print("failed, rc=");  Serial.print(client.state());  Serial.println(" try again in 5 seconds");  delay(5000);  }  }  }  void setup() {  Serial.begin(9600);  dht11.begin();  digitalWrite(D0, LOW);  pinMode(D0, OUTPUT);  pinMode(DHT11PIN, INPUT\_PULLUP);    client.setServer(server, 1883);  wifiConnect();  delay(10);    }  void loop() {  value = analogRead(A0);  payload = (String) value;  float h11 = dht11.readHumidity();  float t11 = dht11.readTemperature();  String payload2 = "Humidity: " + (String) h11 + "|Temp: " + (String) t11;  if(client.connected()) {  if(client.publish(topic, (char\*) payload.c\_str())) {  Serial.print("publish ok (");  Serial.print(payload);  Serial.print(")");    }  else {  Serial.println("Published failed. ");  }  if(client.publish(topic2, (char\*) payload2.c\_str())) {  Serial.print("publish ok (");  Serial.print(payload2);  Serial.print(")");    }  else {  Serial.println("Published failed. ");  }    }  else {  mqttReConnect();  }  delay(200);  }  ***NodeMCU + Relay + Light + LED***  #include <PubSubClient.h>  #include <ESP8266WiFi.h>  const char\* ssid = "HUAWEI-29D9";  const char\* password = "61031018";  const char\* server = "192.168.8.101";  const char\* topic = "uop/mqtt\_tutorial/RELAY";  const char\* clientName = "com.dsajhg.nodemcu";  WiFiClient wifiClient;  PubSubClient client(wifiClient);  void wifiConnect() {  delay(10);  // We start by connecting to a WiFi network  Serial.println();  Serial.print("Connecting to ");  Serial.println(ssid);  WiFi.begin(ssid, password);  while (WiFi.status() != WL\_CONNECTED) {  delay(500);  Serial.print(".");  }  Serial.println("");  Serial.print("WiFi connected - ESP IP address: ");  Serial.println(WiFi.localIP());  }  void mqttReConnect() {  while(!client.connected()) {  Serial.print("Attempting MQTT connection...");  if(client.connect(clientName)) {  Serial.println("connected");  client.subscribe(topic);    }  else {  Serial.print("failed, rc=");  Serial.print(client.state());  Serial.println(" try again in 5 seconds");  delay(5000);  }  }  }  int LEDPin = D2;  void callback(String topic, byte\* message, unsigned int length) {  Serial.println("Message arrived on topic: ");  Serial.print(topic);  Serial.print(". Message: ");  String messageTemp;  for(int i = 0; i < length; i++) {  Serial.print((char) message[i]);  messageTemp += (char) message[i];  }  Serial.println();  if(topic == "uop/mqtt\_tutorial/RELAY")  {  if(messageTemp == "1") {  digitalWrite(LEDPin, HIGH);  }  else {  digitalWrite(LEDPin, LOW);  }  }    }  void setup() {  Serial.begin(9600);  pinMode(LEDPin, OUTPUT);  wifiConnect();  client.setServer(server, 1883);  client.setCallback(callback);  }  void loop() {    if(!client.connected()) {  mqttReConnect();  }  client.loop();  } |