

## **DLD Report**

# **Automatic Room**

### **Group Members:**

(Muhammad Usman Yousuf 2012358)

(Syed Ali Sher 2012367)

(Abdul Sami Hameed 1612141)

(Mir Balach Jarwar 1612298)

### Objective :

The objective of this project is to automate the functionality of room utilities to maximize efficiency.

### Required Equipment:

Dht11

Arduino nodemcu

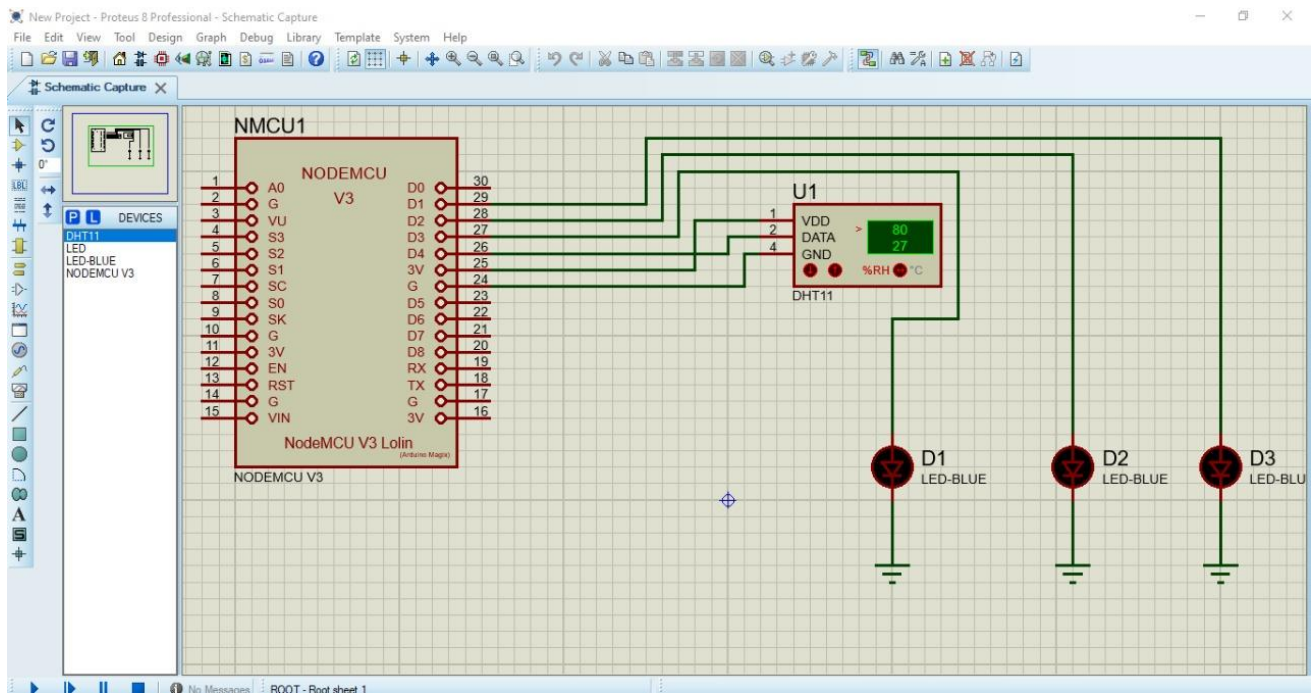
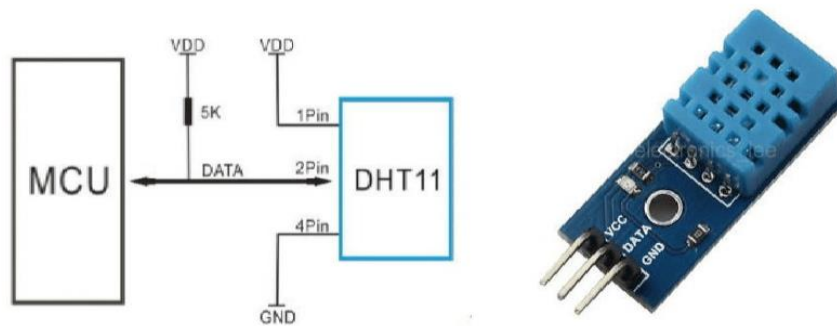
3x LED

### Software:

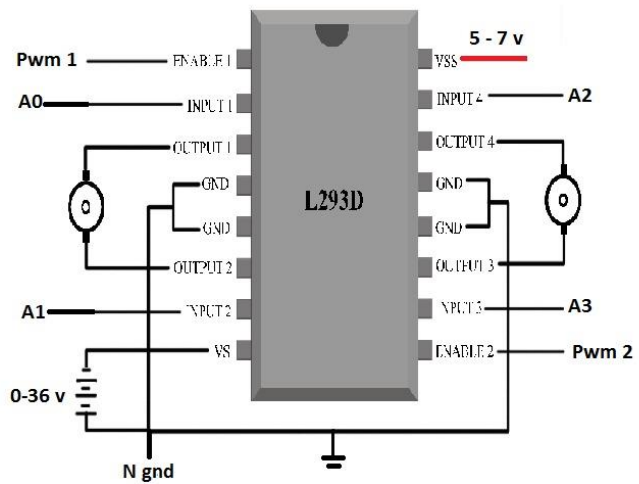
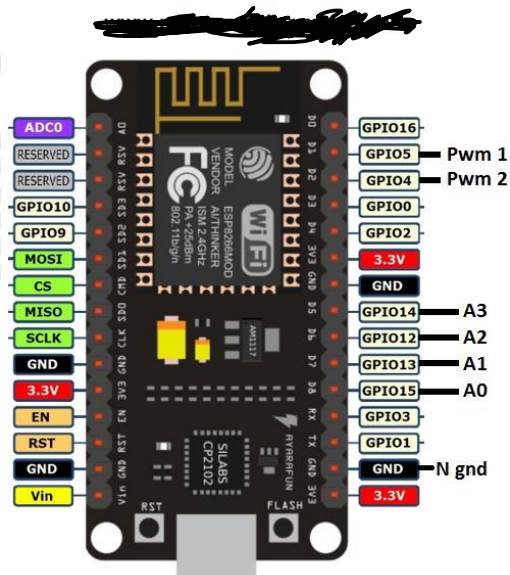
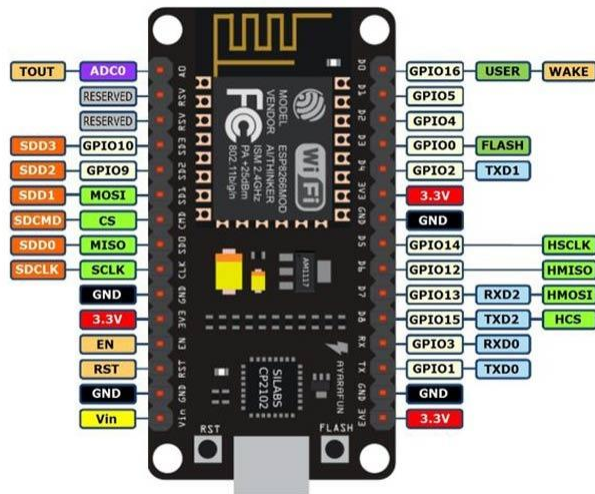
Arduino IDE

Proteus 8.12

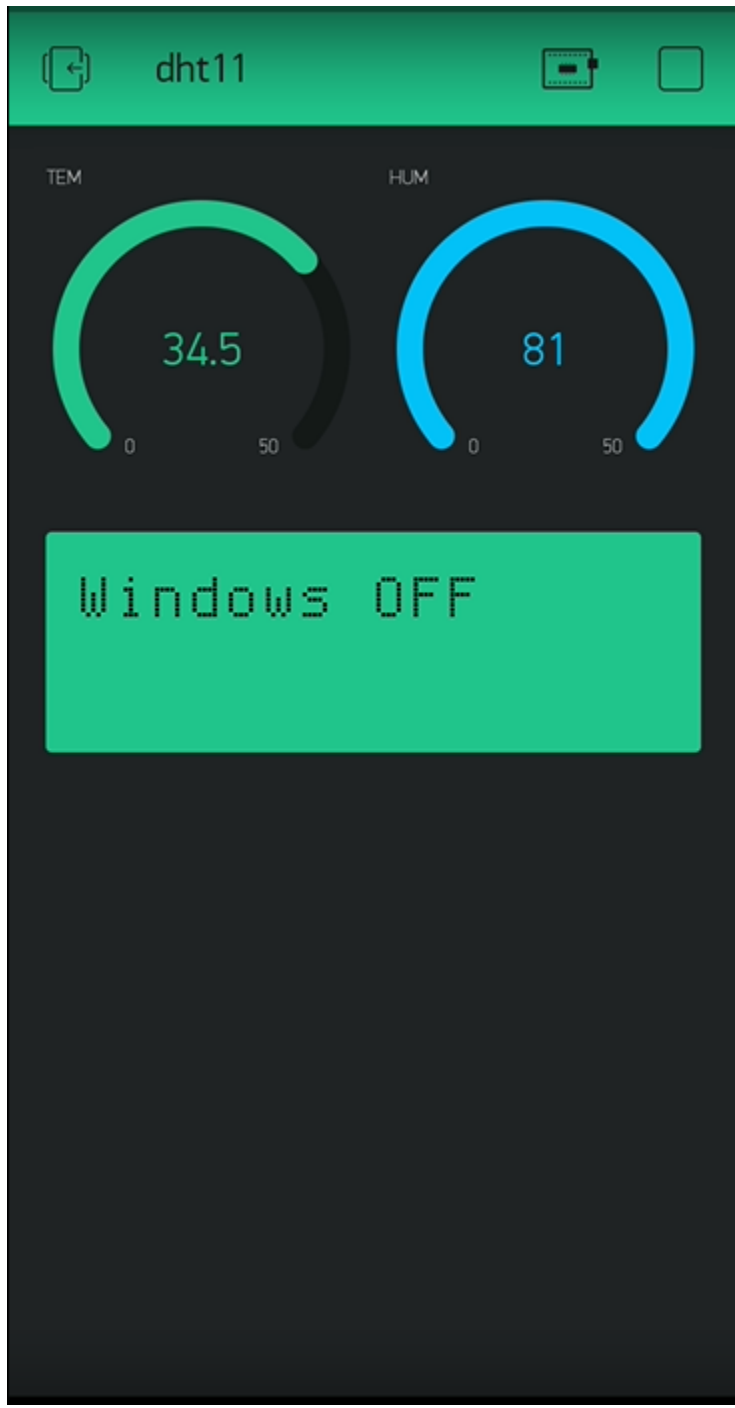
## Circuit Diagrams:



## Arduino NodeMCU:



[Blynk App:](#)



## Snippets:

```
Automatic_Room

#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "hrMH3Om-10mMjhvjQwDbqRF-A1_0wxQK";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "Unknown";
char pass[] = "karachi021";

#define DHTPIN D4           // What digital pin we're connected to
#define ac D3
#define heater D2
#define window D1

// Uncomment whatever type you're using!
#define DHTTYPE DHT11       // DHT 11
// #define DHTTYPE DHT22     // DHT 22, AM2302, AM2321
// #define DHTTYPE DHT21     // DHT 21, AM2301

DHT dht(DHTPIN, DHTTYPE);
BlynkTimer timer;
WidgetLCD lcd(V1);

BlynkTimer timer;
WidgetLCD lcd(V1);

// This function sends Arduino's up time every second to Virtual Pin (5).
// In the app, Widget's reading frequency should be set to PUSH. This means
// that you define how often to send data to Blynk App.
void sendSensor()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature(); // or dht.readTemperature(true) for Fahrenheit

  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  if(t>20 && h<80){
    lcd.clear();
    lcd.print(0, 0, "Air Conditioner ON");
    digitalWrite(ac, HIGH);
    digitalWrite(window, LOW);
    digitalWrite(heater, LOW);
  }
  else if(t<10 && h<80){
    lcd.clear();
    lcd.print(0, 0, "Heater ON");
    digitalWrite(heater, HIGH);
    digitalWrite(ac, LOW);
    digitalWrite(window, LOW);
  }
}
```

---

```

    digitalWrite(window, LOW);

}
else if(h>80){
  lcd.clear();
  lcd.print(0, 0, "Windows OFF");
  digitalWrite(window, HIGH);
  digitalWrite(heater, LOW);
  digitalWrite(ac, LOW);
}
// You can send any value at any time.
// Please don't send more that 10 values per second.
Blynk.virtualWrite(V5, h);
Blynk.virtualWrite(V6, t);
}

void setup()
{
  // Debug console
  Serial.begin(9600);

  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 80);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);

  dht.begin();
  pinMode(ac, OUTPUT);
  pinMode(heater, OUTPUT);

```

---

```

|

  dht.begin();
  pinMode(ac, OUTPUT);
  pinMode(heater, OUTPUT);
  pinMode(window, OUTPUT);

  // Setup a function to be called every second
  timer.setInterval(1000L, sendSensor);
  lcd.clear(); //Use it to clear the LCD Widget
  lcd.print(0, 0, "Distance in cm");
}

void loop()
{
  Blynk.run();
  timer.run();
}

```

---





```

// Uncomment whatever type you're using!

#define DHTTYPE DHT11    // DHT 11

// #define DHTTYPE DHT22  // DHT 22, AM2302, AM2321

// #define DHTTYPE DHT21  // DHT 21, AM2301

DHT dht(DHTPIN, DHTTYPE);

BlynkTimer timer;

WidgetLCD lcd(V1);

// This function sends Arduino's up time every second to Virtual Pin (5).

// In the app, Widget's reading frequency should be set to PUSH. This means

// that you define how often to send data to Blynk App.

void sendSensor()

{

  float h = dht.readHumidity();

  float t = dht.readTemperature(); // or dht.readTemperature(true) for Fahrenheit

  if (isnan(h) || isnan(t)) {

    Serial.println("Failed to read from DHT sensor!");

    return;

  }

  if(t>20 && h<80){

    lcd.clear();

    lcd.print(0, 0, "Air Conditioner ON");

    digitalWrite(ac, HIGH);

    digitalWrite(window, LOW);

    digitalWrite(heater, LOW);

  }

  else if(t<10 && h<80){

```

```
    lcd.clear();

    lcd.print(0, 0, "Heater ON");

    digitalWrite(heater, HIGH);

    digitalWrite(ac, LOW);

    digitalWrite(window, LOW);

}

else if(h>80){

    lcd.clear();

    lcd.print(0, 0, "Windows OFF");

    digitalWrite(window, HIGH);

    digitalWrite(heater, LOW);

    digitalWrite(ac, LOW);

}

// You can send any value at any time.

// Please don't send more that 10 values per second.

Blynk.virtualWrite(V5, h);

Blynk.virtualWrite(V6, t);

}

void setup()

{

    // Debug console

    Serial.begin(9600);

    Blynk.begin(auth, ssid, pass);

    dht.begin();

    pinMode(ac, OUTPUT);
```

```
pinMode(heater, OUTPUT);

pinMode(window, OUTPUT);


// Setup a function to be called every second
timer.setInterval(1000L, sendSensor);

lcd.clear(); //Use it to clear the LCD Widget

lcd.print(0, 0, "Distance in cm");

}


void loop()

{

  Blynk.run();

  timer.run();

}
```