Manufacture and working procedure of Temperature Control Unit (TCU)

F N Sabri*, M Mukhidin, Y S Disastra and B Hasan
Universitas Pendidikan Indonesia, Bandung, Indonesia

*niqosabri@student.upi.edu

Abstract. Internet of Think (IoT) is a system that over time the development of technology makes it easy for users to carry out daily activities. Using the principle of the Internet of Think (IoT) making temperature control devices in a room can be more easily done. This room temperature controller is based on the ESP8266 microcontroller by utilizing an internet connection, users can find out the temperature in the room anywhere and anytime by utilizing the Message Queueing Telemetry Transport (MQTT) protocol. This research is intended to make it easier to supervise, control, and manage a job in this case that is for room temperature. In its application, that is bias for broiler chicken coops, server room temperature regulators, or other temperature monitoring required. Monitoring the temperature of a room in real time and not needing to be directly in place is an advantage of this research. The lowest room temperature can be adjusted as you wish or the standard we want. The output from this tool can be in the form of lights, fans, air conditioners, etc.

1. Introduction
Technological progress is something that cannot be avoided in this life, because technological progress will go according to the progress of science. Every innovation was created to provide positive benefits for human life. Technology also provides many conveniences, as well as a new way of doing human activities. Humans have also enjoyed the many benefits brought about by technological innovations that have been produced in the last decade. In the current era of globalization, technology mastery is the prestige and indicator of a country's progress. Countries are said to be advanced if they have a high level of technological mastery (high technology), whereas countries that cannot adapt to technological advances are often referred to as failed countries [1-5].

The development of technology today is increasingly rapid. This development is not far from the needs needed by humans. Technology has now almost filled all aspects of human life, starting from communication technology, agricultural technology, etc. Therefore, technology is an absolute necessity at this time for humans [6-8].

Internet of things (IoT), which aims to broaden the benefits of continuously connected internet connectivity. As for capabilities such as data sharing, remote control, and so on, including real objects. All levels of society can take advantage of this technological progress in the form of IoT without the exception of the government. The government also uses this IoT to monitor everything that is handled by it [9-11].

The development of telecommunications technology that is very fast has a positive impact on users, can be seen in terms of services from various telecommunications services that can be felt at this time.
This is directly proportional to the larger and better network building business at a more affordable price. With the increase in the size and number of devices built in one network, the monitoring system for each device becomes very important because it will make it easier for administrators to monitor devices that are connected in one network. There are several things that cause problems for monitoring a device, namely how information from the device (tool) that will be monitored provides information to the server as information gatherers. For resolution, it is necessary to design a communication system between the device and the server that receives or requests the information. Communication between the device (device) with the server is required for a protocol for sending data in a data communication network. This research uses the Message Queuing Telemetry Transport (MQTT) protocol. MQTT is one of the existing protocols that has an advantage that is the process of sending data that is fast and easy to operate [12-14].

2. Method
The research method:

2.1. Literature studies
Literature study is a research method to find theories that are relevant to research. In this research, the literature study method is used to find out the theories used namely about the Internet of Think (IoT), ESP8266, DHT11, and MQTT protocol.

2.2. Experimental method
In this research using the experimental method that is by trying to design or make based on the theory of relevance that has been obtained previously.

2.3. Case study method
In this research the case study method is used, namely by systematically collecting data, analysis, and reporting the results.

3. Results and discussion

3.1. Flowchart
The flow chart of this research can be seen in Figure 1.
3.2. Circuit design

The Temperature Control Unit (TCU) assembly uses a circuit like the picture above. The circuit itself uses several components, namely ESP8266 or NodeMCU, DHT, and LED. DHT11 has 4 legs, where foot 1 is connected to vcc 3v from ESP8266, foot number 2 is connected to GPIO16 ESP8266 as data, while foot number 3 DHT11 is not used, and foot number 4 is connected to ESP8266 ground. The positive pole (+) LED is connected to the GPIO5 ESP8266 which functions as data, and the negative pole (-) LED is connected to the ground ESP8266. ESP8266 itself is connected to a PC via a connector that has been provided by micro USB.
3.3. Programming

Making a program on the Temperature Control Unit (TCU) uses Arduino IDE software, using the ESP8266 and DHT11 libraries. The following program is installed on the nodeMCU ESP8266 used:

```c
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <dht.h>

DHT dht;

// Define NodeMCU D3 pin to as temperature data pin of DHT11
#define DHT11_PIN 16
#define DHTTYPE DHT11
#define LED1 5

// Update these with values suitable for your network.
const char* ssid = "xxx";
const char* password = "XxxxxX";
const char* mqtt_server = "postman.cloudmqtt.com";
const int mqttPort = 9999;       //Port number
const char* mqttUser = "qwerty"; //User
const char* mqttPassword = "qwertyuiop"; //Password

WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;

void setup_wifi() {
    delay(100);
    pinMode(LED1, OUTPUT);
    // We start by connecting to a WiFi network
    Serial.print("Connecting to ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("Connected!");
    // Setup the client
    client.begin(mqtt_server, mqttPort, mqttUser, mqttPassword);
    // Connect to the MQTT server
    if (client.connect("nodeMCU", 0)) {
        Serial.println("Connected to MQTT server");
    } else {
        Serial.println("Failed to connect to MQTT server");
    }
}
```

Figure 2. Circuit of Temperature Control Unit (TCU).
void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Command is : ");
  Serial.print(topic);
  int p = (char)payload[0] - '0';
  int chk = DHT.read11(DHT11_PIN);
  // if MQTT comes a 0 message, show humidity
  if (p == 0) {
    Serial.println("to show humidity!");
    Serial.print(" Humidity is: ");
    Serial.print(DHT.humidity, 1);
    Serial.println("%");
  }
  // if MQTT comes a 1 message, show temperature
  if (p == 1) {
    // digitalWrite(BUILTIN_LED, HIGH);
    Serial.println(" is to show temperature!");
    int chk = DHT.read11(DHT11_PIN);
    Serial.print(" Temp is: ");
    Serial.print(DHT.temperature, 1);
    Serial.println("C");
  }
  Serial.println();
}

void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Create a random client ID
    String clientId = "ESP8266";
    clientId += String(random(0xffff), HEX);
    // Attempt to connect
    //if you MQTT broker has clientID,username and password
    //please change following line to if (client.connect(clientId, userName, passWord))
    if (client.connect("ESP8266", mqttUser, mqttPassword)) {
      Serial.println("connected");
      //once connected to MQTT broker, subscribe command if any
    }
  }
}
client.subscribe("TkCommand");
} else {
    Serial.print("failed, rc=");
    Serial.print(client.state());
    Serial.println(" try again in 3 seconds");
    // Wait 3 seconds before retrying
    delay(3000);
}
} //end reconnect()

void setup() {
    Serial.begin(115200);
    setup_wifi();
    client.setServer(mqtt_server, 12550);
    client.setCallback(callback);
    int chk = DHT.read11(DHT11_PIN);
    Serial.print(" Starting Humidity: ");
    Serial.print(DHT.humidity, 1);
    Serial.println("");
    Serial.print(" Starting Temparature ");
    Serial.print(DHT.temperature, 1);
    Serial.println("C");
}

void loop() {
    if (!client.connected()) {
        reconnect();
    }
    client.loop();
    long now = millis();
    // read DHT11 sensor every 6 seconds
    if (now - lastMsg > 1000) {
        lastMsg = now;
        int chk = DHT.read11(DHT11_PIN);
        float suhu = DHT.temperature;
        float lembab = DHT.humidity;
        Serial.print("Temperature = ");
        Serial.println(suhu);
        Serial.print("Humidity = ");
        Serial.println(lembab);
        delay(1000);
        if (suhu >32.00) {
            digitalWrite(LED1, LOW); // LED padam
            Serial.println(" Suhu cukup ");
        }
        else if(suhu <= 32.00) {
            digitalWrite(LED1, HIGH); // LED menyala
            Serial.println(" Suhu kurang ");
        }
    }
}
4. Procedure

The workings of the Temperature Control Unit (TCU) are quite simple. The initial stages of the work of this tool is by connecting ESP8266 with Wi-Fi that has been registered in the ESP8266 program as figure 3.

```c
void setup_wifi() {
  delay(1000);
  pinMode(LED1, OUTPUT);
  // We start by connecting to a Wi-Fi network
  Serial.print("Connecting to ");
  Serial.println("ESP8266");
  WiFi.begin(pwd, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print 
  }
  randomSend(500);
  Serial.println("\n");
  sendMessage("\n");
  Serial.println("\n");
  Serial.println("\n");
}
```

Figure 3. Connecting to Wi-Fi ESP8266.

If the connection from ESP8266 is connected to Wi-Fi that has been registered in the program above, then it can connect to the MQTT server that we have by first registering the MQTT that we have to the ESP8266 program as figure 4.

```c
void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    String clientId = "ESP8266-";
    clientId += String(random(0, 65535));
    // Attempt to connect
    if (!client.connect(clientId.c_str(), userName, password)) {
      Serial.print("\n");
      Serial.println("\n");
      Serial.print("\n");
      if (client.connect("ESP8266", mqttUser, mqttPassword)) {
        Serial.println("\n");
        Serial.println("\n");
      } else {
        Serial.println("\n");
        Serial.println("\n");
        Serial.print("\n");
      }
    }
  }
}
```

Figure 4. Connecting to the MQTT server.

DHT11 component is able to work as an input or temperature reader if ESP8266 is connected to Wi-Fi and connected to the MQTT server, then if it has read the existing room temperature then sent to the
microcontroller ESP8266 through the GPIO16 port ESP8266, then it will be sent to the MQTT protocol according to the program in figure 5.

![Figure 5. DHT11 connecting to MQTT and reading temperature.](image)

Then if the temperature is below then the LED will logic 1 or turn on as an indicator that the temperature in the room is less, according to the existing program as figure 6.

![Figure 6. LED logic program 1.](image)

5. Conclusion
From the research and assignments given by field supervisors in the industry it can be concluded that the process of making a Temperature Control Unit (TCU) requires a long time. Then the results obtained on the serial monitor with the MQTT cloud are real data that cannot be changed or manipulated. This tool also requires sufficient internet connection to send data from the microcontroller to the cloud.

The Temperature Control Unit (TCU) can also be developed into other tools that require real and fast data. Because the use of the Message Queuing Telemetry Transport (MQTT) protocol is a protocol that is quite fast in the data transfer process and has a lightweight or not large size.

References
[1] Arif H S 2014 Perbandingan Akurasi Pengukuran Suhu dan Kelembaban antara Sensor DHT11 dan DHT22 *Jurnal Infotel* 6 2 1-10
[2] Abhishek S B and Jason E M 2017 Connected Home Using MQTT and Common Sensor Framework *IJAREEIE* **6** 4

[3] Nirmala A P and Prasenjit K D 2019 IoT based Automatic Light Control with Temperature Monitoring and Alert mechanism *IJEAT* **8** 6

[4] Bricker D 1995 The challenge of inclusion *Journal of early intervention* **19** 3 179-194

[5] Bell S, Hirvonen M, Kukemelk H, Pancocha K, Vadurova H and Hion P 2014 The inclusion of students with special educational needs in vocational education and training: a comparative study of the changing role of SEN teachers in the European context (Finland, England, The Czech Republic and Estonia) *Socialiniai Tyrimai* **2**

[6] Nasution F 2011 *Mikrokontrolier 5–42*

[7] Lipovski G J 2004 *Introduction to microcontrollers: architecture, programming, and interfacing for the Freescale 68HC12* (Elsevier)

[8] Harshit B, Brij B and Nagesh K 2019 IOT: the Current Scenario and Role of Sensors Involved in Smart Agriculture *IJRTE* **8** 4

[9] M. Weinmann, S. Zeh, and N.-V. Olarescu 2012 *EP2536019A1: Method for controlling an inverter*

[10] Mukesh R S, Muthu K S, Sudha A and Thirumalai N D P 2019 *Environmental Monitoring System Using IoT and Cloud Service at Real-Time* (EasyChair)

[11] Yusuf M 2019 Peran Penting Sensor Dalam Sistem Internet of Things [Online] http://risa.dosen.ittelkom-pwt.ac.id/2019/01/11/peran-penting-sensor-dalam-sistem-internet-of-things/ Accessed on 18 August 2019

[12] Ravi K K and Kopulwar S M 2016 *A low cost implementation of MQTT using ESP8266* IEEE

[13] Raviteja B, Raghudathesh G P, Megha D H, Bindu H V and Madhuri C N 2018 MQTT Based Patient Health Monitoring *IJPAM* **120** 6

[14] Shen Z J and Omura I 2007 Power semiconductor devices for hybrid, electric, and fuel cell vehicles *Proceedings of the IEEE* **95**