**ASSIGNMENT 2 FRONT SHEET**

|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification** | **TEC Level 5 HND Diploma in Computing** | | |
| **Unit number and title** | **Unit 43: Internet of Things** | | |
| **Submission date** | 16/12/2022 | **Date Received 1st submission** |  |
| **Re-submission Date** |  | **Date Received 2nd submission** |  |
| **Student Name** | Do Huu Duy | **Student ID** | GCC200018 |
| **Class** | GCC0903 | **Assessor name** | Luong Hoang Huong |
| **Student declaration**  I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice. | | | |
|  |  | **Student’s signature** | huuduy |

**Grading grid**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| P5 | P6 | P7 | M5 | M6 | D3 | D4 |
|  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **❒ Summative Feedback: ❒ Resubmission Feedback:** | | |
| **Grade:** | **Assessor Signature:** | **Date:** |
| **Internal Verifier’s Comments:** | | |
| **Signature & Date:** | | |

# Assignment Brief 2 (RQF)

## Higher National Certificate/Diploma in Business

|  |  |
| --- | --- |
| **Student Name/ID Number:** |  |
| **Unit Number and Title:** | **Unit 43 – Internet of Things** |
| **Academic Year:** | **2021** |
| **Unit Assessor:** | **Tran Trong Minh** |
| **Assignment Title:** | **Assignment 2 – Internet of Things** |
| **Issue Date:** |  |
| **Submission Date:** |  |
| **Internal Verifier Name:** |  |
| **Date:** |  |

|  |
| --- |
| **Submission Format:** |
| **Format:** This assignment is an Individual assignment and specifically including 1 document: You must use font Calibri size 12, set number of the pages and use multiple line spacing at 1.3. Margins must be: left: 1.25 cm; right: 1 cm; top: 1 cm and bottom: 1 cm. The reference follows Harvard referencing system. The recommended word limit is 2.000-2.500 words. You will not be penalized for exceeding the total word limit. The cover page of the report has to be the Assignment front sheet 2.  **Submission:** Students are compulsory to submit the assignment in due date and in a way requested by the Tutors. The form of submission will be a soft copy posted on <http://cms.greenwich.edu.vn/>  **Note:** The Assignment must be your own work, and not copied by or from another student or from books etc. If you use ideas, quotes or data (such as diagrams) from books, journals or other sources, you must reference your sources, using the Harvard style. Make sure that you know how to reference properly, and that understand the guidelines on plagiarism. If you do not, you definitely get fail |
| **Unit Learning Outcomes:** |
| **LO1** Analyze what aspects of IoT are necessary and appropriate when designing software applications  **LO2** Outline a plan for an appropriate IoT application using common architecture, frameworks, tools, hardware and APIs  **LO3** Develop an IoT application using any combination of hardware, software, data, platforms and services.  **LO4** Evaluate your IoT application and detail the problem your IoT application solves, the potential impact on people, business, society and the end user and the problems it might encounter when integrating into the wider IoT ecosystem |
| **Assignment Brief and Guidance:** |
| You currently work as a product developer for a new startup where you design IoT products for the consumer, corporate, government and defense clients. As part of your role your manager has tasked you to plan and develop a new IoT product, service or application for a potential client. You are required to identify a target user and conduct tests with this user and include this feedback into multiple iterative versions of your product.  **Part 1 (Assignment 1)**: For the first part, you must:   * Plan an IoT application for a specific target end user and the tests you intend to conduct with this user. This plan will be in the form of a document and will include supporting evidence and material, such as user personas and customer journey maps. * Create multiple iterations of your application and modify each iteration with enhancements gathered from user feedback and experimentation. This will follow the pathway outlined in your plan (logbook,)   **Part 2 (Assignment 2)**: For the second part, you must produce a report to prove that:   * Show evidence about Developed IoT application using any combination of hardware, software, data, platforms and services (video or images of your IoT system with code snippet) * Evaluate your IoT application and detail the problem your IoT application solves, the potential impact on people, business, society and the end user and the problems it might encounter when integrating into the wider IoT ecosystem |

|  |  |  |
| --- | --- | --- |
| Learning Outcomes and Assessment Criteria | | |
| Pass | Merit | Distinction |
| **LO3** Develop an IoT application using any combination of hardware, software, data, platforms and services. | | |
| **P5** Employ an appropriate set of tools to develop your plan into an IoT application.  **P6** Run end user experiments and examines feedback. | **M5** Reconcile and evaluate end user feedback and determine advantages and disadvantages of your chosen IoT techniques. | **D3** Critical evaluate security risks that your application might encounter. |
| **LO4** Evaluate your IoT application and detail the problem your IoT application solves, the potential impact on people, business, society and the end user and the problems it might encounter when integrating into the wider IoT ecosystem | | |
| **P7** Evaluate end user feedback from your IoT application. | **M6** Undertake a critical review and compare your final application with the original plan. | **D4** Critique the overall success of your application. Did it solve your problem? What is the potential impact on people, business, society and the end user? What problems might it encounter when integrating into the wider IoT ecosystem? |

Table of Contents

[I. Employ an appropriate set of tools to develop your plan into an IoT application 8](#_Toc121823191)

[1. Problem 8](#_Toc121823192)

[2. Solution 8](#_Toc121823193)

[3. The techniques that include in an application to solve this problem 8](#_Toc121823194)

[3.1 Tool 8](#_Toc121823195)

[3.2 Hardware 9](#_Toc121823196)

[4. Circuit table 10](#_Toc121823197)

[4.1. Circuit table for turn-on and off the light feature 10](#_Toc121823198)

[4.2. Circuit table for turn-on and off the fan feature 10](#_Toc121823199)

[4.3. Circuit table for motion sensor feature 10](#_Toc121823200)

[4.4. Circuit table for temperature, humidity sensor feature 10](#_Toc121823201)

[4.5. Circuit table for rain sensor feature 11](#_Toc121823202)

[II. Run end user experiments and examines feedback 11](#_Toc121823203)

[1. Function description 11](#_Toc121823204)

[2. Code 11](#_Toc121823205)

[3. Image of project 16](#_Toc121823206)

[4. Video 25](#_Toc121823207)

[III. Evaluate end user feedback from your IoT application 26](#_Toc121823208)

[References 27](#_Toc121823209)

Table of Figures

[Figure 1. Interface of website 16](#_Toc121822630)

[Figure 2. Test turn-on and off the light 17](#_Toc121822631)

[Figure 3. Result of test turning on the light 17](#_Toc121822632)

[Figure 4. Test turn-on and off the light 18](#_Toc121822633)

[Figure 5. Result of test turning off the light 18](#_Toc121822634)

[Figure 6. Test turn-on and off the fan 19](#_Toc121822635)

[Figure 7. Result of test turning on the fan 19](#_Toc121822636)

[Figure 8. Test turn-on and off the fan 20](#_Toc121822637)

[Figure 9. Result of test turning off the fan 20](#_Toc121822638)

[Figure 10. Test rain sensor when it rains 21](#_Toc121822639)

[Figure 11. Test rain sensor when it suns 22](#_Toc121822640)

[Figure 12. Test motion when have movement 23](#_Toc121822641)

[Figure 13. Test motion sensor when don't have movement 23](#_Toc121822642)

[Figure 14. Test auto turn-on and off the fan feature based on temperature 24](#_Toc121822643)

[Figure 15. Test the fan auto turn on when the temperature is greater than 27 degrees Celsius 24](#_Toc121822644)

[Figure 16. Test auto turn-on and off the fan feature based on temperature 25](#_Toc121822645)

[Figure 17. Test the fan auto turn off when the temperature is equal or smaller than 27 degrees Celsius 25](#_Toc121822646)

Table of Tables

[Table 1. Tool 8](#_Toc121823299)

[Table 2. Hardware 9](#_Toc121823300)

[Table 3. ESP 32 and Led 10](#_Toc121823301)

[Table 4. ESP 32, Relay, and Fan 10](#_Toc121823302)

[Table 5. ESP 32, Relay, PIR Motion Thermal Sensor HC-SR501, and Led 10](#_Toc121823303)

[Table 6. Arduino Uno, IC2 LCD, Temperature Humidity Sensor DHT11, and ESP 32 11](#_Toc121823304)

[Table 7. Arduino Uno, Rain-water sensor, Servo V9 11](#_Toc121823305)

# I. Employ an appropriate set of tools to develop your plan into an IoT application

## 1. Problem

Before technology developed today, we did not have a lot of technology applications with the devices in our homes to make them smart and we can control them flexibly and conveniently. In the past when we went on a trip with our family for a few days but forgot to turn off appliances such as lights, fans, or air conditioners, how can we turn them off, we can only think of traveling just to return home and turn off them, or we will accept electricity bills for these devices when we are not using them. In addition, when we are at home, how will we turn on fans or air conditioners when we are not near them? For example, when I am watching a favorite movie and I want to turn on the air conditioner or fan but the remote is far from me and I'm too lazy to go to get it and I will miss the good parts of the movie and when I go to get the controller to turn on the fan or air conditioner. This is a problem that I need to solve by applying IoT.

## 2. Solution

I will apply IoT to the devices in the house to make the house smart. I will implement the following functions. First, it is the function of turning on and off the lights remotely via the web by accessing mobile phones or laptops. Second, it is the function of turning on and off the fan or air conditioner remotely via the web by accessing a phone or laptop. Third, it is the function of automatically turning on the fan or air conditioner when the indoor temperature rises, I will use the sensor to measure the indoor temperature and when the temperature rises, the fan will auto-turn on to reduce the temperatures in the house. Fourth, it is the function of automatically turning off and on the lights when someone comes to the house, I will use the body temperature motion sensor and put it in front of the door and when someone comes to the house the sensor will receive the signal and will activate the trigger relay to turn on the light and the off light will automatically turn off when the sensor does not receive a signal of movement in front of the house. Fifth, it's a rain sensor, this function will help me dry clothes easily and conveniently, I will use a rain sensor and servo for this function when it rains the drying pole will automatically turn into the roof to avoid rain and when it's sunny the drying pole will rotate outside the roof to dry clothes. so on when it rains, it turns inside the roof and when it's sunny, it turns out to dry.

## 3. The techniques that include in an application to solve this problem

### 3.1 Tool

Table 1. Tool

|  |  |
| --- | --- |
| **Name** | **Description about tool** |
| Blynk | Blynk is a full suite of software required to prototype, deploy, and remotely manage connected electronic devices at any scale. With Blynk anyone can connect their hardware to the cloud and build web applications to analyze real-time and historical data coming from devices, control them remotely from anywhere in the world, and receive important notifications. Applications made with Blynk are ready for the end-users. Whether it is your family member, an employee, or someone who has purchased your product, they will be able to download the app, connect the device and start using it. In my project, I use Blynk to design a website to control the devices of my smart house model. When designing in Blynk, I feel Blynk is a useful web to perform a project in my subject and Blynk is easy to use (blynk, 2022) |
| Arduino IDE | Arduino is an open-source user-friendly hardware and software prototyping platform. Dedicated to making small single-board computers able to sense and exercise control over the physical world. Arduino employs a set of hardware-software specifications applied to interactive electronics which includes, among others, the Arduino programming language and the Integrated Development Environment (IDE). In my project, I use the Arduino IDE tool to write and run code. Arduino IDE tool is easy to use (avsystem, 2019) |

### 3.2 Hardware

Table 2. Hardware

|  |  |
| --- | --- |
| **Name** | **Description about hardware** |
| Led | The led light is used to perform a turn-on and off the light in the house feature through the web. In addition, the led light is also used for the auto turn-on and off feature when there are people coming to the house. |
| Temperature Humidity Sensor DHT11 | I use the Temperature Humidity Sensor DHT11 to perform measure the temperature in the house and auto turn-on the fan when the temperature is greater than 26 degrees Celsius and auto turn-off when the temperature is smaller than 26-degree Celsius |
| PIR Motion Thermal Sensor HC-SR501 | I use the PIR Motion Thermal Sensor HC-SR501 to perform the auto turn-on and off the light feature when there are people coming to the house. The PIR Motion Thermal Sensor HC-SR501 will get the signal there are people coming to the house and the sensor will send the signal to the ESP 32 to ESP 32 control the relay turn-on to the led is light and opposite when there are no people coming to the house the led will turn off. |
| Relay KY-019 5VDC | I use Relay KY-019 5VDC like a switch to turn-on and off the light and fan |
| LCD display LCD1602 | I use LCD display LCD1602 to present the temperature and humidity that the Temperature Humidity Sensor DHT11 measure to the screen of CLD |
| Fan | The Fan is used like an air conditioner to reduce the temperature in the house when it raises |
| Arduino Uno | I use Arduino Uno to run code to present the temperature and humidity in the house to LCD feature and perform the drying pole auto turn inside the roof when it rains and turn outside the roof when it is a sunny feature |
| Esp32 Nodemcu | I use Esp32 Nodemcu to run code to perform some features such as auto turn-on and off the light when there are people coming to the house, auto turn-on the fan based on the temperature measurements in the house if the temperature is greater than 26 degrees Celsius the fan will turn-on and opposite smaller 26 degree Celsius the fan will turn-off, the turn-on/off the light and fan through the website. |
| Rain-water sensor | I use this sensor to get a signal when it rains to perform the feature auto swivel of the drying pole inside the roof when it rains and swivel outside the roof when it suns |
| Servo V9 | I use Servo V9 to swivel the drying pole clothes |

## 4. Circuit table

4.1. Circuit table for turn-on and off the light feature

Table 3. ESP 32 and Led

|  |  |
| --- | --- |
| **ESP 32** | **Led** |
| GND | - |
| P16 | + |

4.2. Circuit table for turn-on and off the fan feature

Table 4. ESP 32, Relay, and Fan

|  |  |  |
| --- | --- | --- |
| **ESP 32** | **Relay** | **Fan** |
| GND | GND | - |
| 5V | 5V |  |
| P18 | IN/S |  |
|  | NO | + |
| 5V | COM |  |

4.3. Circuit table for motion sensor feature

Table 5. ESP 32, Relay, PIR Motion Thermal Sensor HC-SR501, and Led

|  |  |  |  |
| --- | --- | --- | --- |
| **ESP 32** | **Relay** | **PIR Motion Thermal Sensor HC-SR501** | **Led** |
| GND | GND | GND | - |
| 5V | 5V | VCC |  |
| P2 | IN/S |  |  |
| P4 |  | Out |  |
|  | NO |  | + |
| 5V | COM |  |  |

4.4. Circuit table for temperature, humidity sensor feature

Table 6. Arduino Uno, IC2 LCD, Temperature Humidity Sensor DHT11, and ESP 32

|  |  |  |  |
| --- | --- | --- | --- |
| **Arduino Uno** | **IC2 LCD** | **Temperature Humidity Sensor DHT11** | **ESP 32** |
| GND | GND | - |  |
| 5V | VCC | + |  |
| A5 | SCL |  |  |
| A4 | SDA |  |  |
| 6 |  | Out | P5 |

4.5. Circuit table for rain sensor feature

Table 7. Arduino Uno, Rain-water sensor, Servo V9

|  |  |  |
| --- | --- | --- |
| **Arduino Uno** | **Rain-water sensor** | **Servo V9** |
| GND | GND | Brown wire |
| 5V | 5V | Red wire |
| 5 | DO |  |
| 9 |  | Yellow wire |

# II. Run end user experiments and examines feedback

## 1. Function description

In this project, I perform a smart house with fives features. Firstly, that is a rain-water sensor is used when it rains the clothes drying pole will auto swivel inside the roof to avoid clothes get wet and when it suns the pole will auto swivel outside the roof to drying. Secondly, that is turn on and off the light by the website, I design a website to control the light in house and I can turn on and turn off the light through the button control in the website. Thirdly, that is body temperature motion sensor. When have people come house the sensor will sensor the move and auto turn on the light and when the people leave and do not have anything move front of the sensor the light will auto turn off. Fourth, that is turn-on and off the fan feature by the website. The last one, that is indoor temperature and humidity sensor. The sensor will measure the temperature and humidity in house and notify to the LCD and when the temperature in house bigger than 27 degrees Celsius the fan will auto turn on to make temperature decrease in house and when the temperature is decrease smaller than 26 degrees Celsius the fan will auto turn off.

## 2. Code

* Code run with ESP 32

#define BLYNK\_TEMPLATE\_ID "TMPLe0aeYp71"

#define BLYNK\_DEVICE\_NAME "Smart House Project"

#define BLYNK\_AUTH\_TOKEN "-D-owzgs-JIkkaNQdP3dhZ4IKStLysGf"

#include <BlynkSimpleEsp32.h>

#include <WiFi.h>

#include <WiFiClient.h>

#include "DHTesp.h"

#include "LiquidCrystal.h"

#include <DHT.h>

#include <Wire.h>

#define DHTpin 5

DHTesp dht;

BlynkTimer timer;

char auth[] = BLYNK\_AUTH\_TOKEN;

char ssid[] = "VNPT\_HONG";

char password[] = "";

int sensorPin = 4;

int pirState = LOW;

int val = 0;

int relayPin = 2;

int relayFan = 18;

int light = 16;

int autoFan = 0;

int remoteFan = 0;

byte degree[8] = {

  0B01110,

  0B01010,

  0B01110,

  0B00000,

  0B00000,

  0B00000,

  0B00000,

  0B00000

};

BLYNK\_WRITE(V0){

  int value = param.asInt();

  if(value == 1){

    digitalWrite(16, HIGH);

  }

  if(value == 0){

    digitalWrite(16, LOW);

  }

}

BLYNK\_WRITE(V3){

  int value = param.asInt();

  if(value == 1){

    autoFan = 1;

  }

  if(value == 0){

    autoFan = 0;

  }

}

BLYNK\_WRITE(V4){

  int value = param.asInt();

  if(value == 1){

    remoteFan = 1;

  }

  if(value == 0){

    remoteFan = 0;

  }

}

void sendTempData(){

  delay(dht.getMinimumSamplingPeriod());

  int h = dht.getHumidity();

  int t = dht.getTemperature();

  Blynk.virtualWrite(V1, t);

  Blynk.virtualWrite(V2, h);

  //Auto fan

  if(autoFan == 1){

    if(t > 27){

      digitalWrite(relayFan, HIGH);

    }else{

      digitalWrite(relayFan, LOW);

    }

  }else{

        digitalWrite(relayFan, LOW);

  }

  if(autoFan == 0 && remoteFan == 0){

          digitalWrite(relayFan, LOW);

  }else if(autoFan == 0 && remoteFan == 1){

      digitalWrite(relayFan, HIGH);

  }

}

void sendSensorData(){

  val = digitalRead(sensorPin);

  if (val == HIGH)

  {

    digitalWrite(relayPin, HIGH);

    delay(1000);

  }

  else

  {

    digitalWrite(relayPin, LOW);

    delay(150);

}

}

void setup() {

  pinMode(relayPin, OUTPUT);

  pinMode(sensorPin, INPUT);

  pinMode(light, OUTPUT);

  pinMode(relayFan, OUTPUT);

  Blynk.begin(auth, ssid, password);

  dht.setup(DHTpin, DHTesp::DHT11);

  timer.setInterval(150, sendSensorData);

  timer.setInterval(10000, sendTempData);

}

void loop(){

  Blynk.run();

  timer.run();

}

* Code run with Arduino Uno

#include <DHT.h>

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include <Servo.h>

LiquidCrystal\_I2C lcd(0x27,16,2);

const int DHTPIN = 6;

const int DHTTYPE = DHT11;

DHT dht(DHTPIN, DHTTYPE);

Servo myservo;

int rainSensor = 5;

int pos;

byte degree[8] = {

  0B01110,

  0B01010,

  0B01110,

  0B00000,

  0B00000,

  0B00000,

  0B00000,

  0B00000

};

void setup() {

  pinMode(9, OUTPUT);

  pinMode(rainSensor, INPUT);

  Serial.begin(9600);

  myservo.attach(9);

  lcd.init();

  lcd.backlight();

  lcd.print("Temp: ");

  lcd.setCursor(0,1);

  lcd.print("Humid: ");

  lcd.createChar(1, degree);

  dht.begin();

}

void loop() {

  int value = digitalRead(rainSensor);

  float h = dht.readHumidity();

  float t = dht.readTemperature();

  if (isnan(t) || isnan(h)) {

  }

  else {

    lcd.setCursor(6,0);

    lcd.print(round(t));

    lcd.print("");

    lcd.write(1);

    lcd.print("C");

    lcd.setCursor(7,1);

    lcd.print(round(h));

    lcd.print("%");

  }

  if (value == HIGH) {

    pos = 90;

    myservo.write(pos);

  } else {

    pos = 0;

    myservo.write(pos);

  }

  delay(1000);

}

## 3. Image of project

* Website interface

Graphical user interface, application

Description automatically generated

Figure 1. Interface of website

* Turn on and turn off the light

Now, I will turn on the light

Graphical user interface, application

Description automatically generated

Figure 2. Test turn-on and off the light

A close up of a steering wheel

Description automatically generated with low confidence

Figure 3. Result of test turning on the light

Now I will turn off the light

Graphical user interface, application, Teams

Description automatically generated

Figure 4. Test turn-on and off the light

A picture containing indoor, fan

Description automatically generated

Figure 5. Result of test turning off the light

* Turn on and turn off the fan

Now I will turn on the fan by turn on the Remote Fan control button in the web

Graphical user interface, application

Description automatically generated

Figure 6. Test turn-on and off the fan

A picture containing indoor, floor

Description automatically generated

Figure 7. Result of test turning on the fan

Now I will turn off the fan by turn off the Remote Fan control button in the web

Graphical user interface, application, Teams

Description automatically generated

Figure 8. Test turn-on and off the fan

A picture containing indoor

Description automatically generated

Figure 9. Result of test turning off the fan

* Rain-water sensor

Assume, it rains now



Figure 10. Test rain sensor when it rains

Now, it suns

A picture containing text, indoor

Description automatically generated

Figure 11. Test rain sensor when it suns

When it suns the drying pole swivel outside the roof to drying clothes

* Body temperature motion sensor

Now I will test when there is movement

Text

Description automatically generated

Figure 12. Test motion when have movement

Now, I will test when there is no movement

A picture containing text

Description automatically generated

Figure 13. Test motion sensor when don't have movement

* Temperature sensor, humidity and the fan auto turn on and off

Now, I will test when the temperatures in house bigger than 26 degrees Celsius and the fan will turn on, but to perform this feature I need to turn on this feature by turn on the control button in website.

Graphical user interface, application, Teams

Description automatically generated

Figure 14. Test auto turn-on and off the fan feature based on temperature

Now, the temperature is 29 degrees Celsius, and the fan will auto turn on

A picture containing indoor

Description automatically generated

Figure 15. Test the fan auto turn on when the temperature is greater than 27 degrees Celsius

Now, when the fan makes the house cooling, the temperatures will be decrease and the fan will turn off

Graphical user interface, application, Teams

Description automatically generated

Figure 16. Test auto turn-on and off the fan feature based on temperature

Now, the temperature is 23 degrees Celsius, and the fan will auto turn off

A picture containing indoor

Description automatically generated

Figure 17. Test the fan auto turn off when the temperature is equal or smaller than 27 degrees Celsius

## 4. Video

Link video: [https://drive.google.com/file/d/1WUbtQtexg0ytf36Mv0Ftp193rVSFyMIV/view](https://drive.google.com/file/d/1WUbtQtexg0ytf36Mv0Ftp193rVSFyMIV/view%20)

# III. Evaluate end user feedback from your IoT application

In this project, I perform solution to build smart house with the devices apply IoT that help the house is smart and useful for user. From the problems that I give about the house in the part 1 of I and the solutions in the part 2 of I.

From the problem and the solution that I give and perform, the model smart house responds to the solutions that I perform to solve the problem with full of functions that I have given. The smart house model is completed with suitable solutions to solve the problem before with five main features. Firstly, that is a rain-water sensor is used when it rains the clothes drying pole will auto swivel inside the roof to avoid clothes get wet and when it suns the pole will auto swivel outside the roof to drying. Secondly, that is turn on and off the light by the website, I design a website to control the light in house and I can turn on and turn off the light through the button control in the website. Thirdly, that is body temperature motion sensor. When have people come house the sensor will sensor the move and auto turn on the light and when the people leave and do not have anything move front of the sensor the light will auto turn off. Fourthly, that is turn-on and off the fan feature by the website. I will clink on the Remote Fan button in the website to control the fan turn-on and off. The last one, that is indoor temperature and humidity sensor. The sensor will measure the temperature and humidity in house and notify to the LCD and when the temperature in house bigger than 27 degrees Celsius the fan will auto turn on to make temperature decrease in house and when the temperature is decrease smaller than 26 degrees Celsius the fan will auto turn off.

With this smart house model, I spend 661,000 VND to perform to buy hardware, with this price I think it is suitable and not expensive for a project like this. However, the system of the project such as the devices often gets malfunctioned as the devices get the signal poor, I think it's due to the devices that I use being cheap devices so the quality the poor so there will be some issues.

In terms of commerciality, I think the solutions that I apply to the house are useful and can be implemented in real life to help people in life. With this price for devices apply IoT in smart house I think, it is suitable and cheap for a project like this and the cionvenient that it brings is good for people's life.

Generally, my smart house model completed what I want and solve the problems that I give for the smart house. Besides, the project also supply objective perspective and may be develop and real implementation in the future.

# References

avsystem, 2019. *avsystem.com.* [Online]   
Available at: https://www.avsystem.com/blog/10-open-source-iot-tools/  
[Accessed 13 12 2022].

blynk, 2022. *docs.blynk.io.* [Online]   
Available at: https://docs.blynk.io/en/  
[Accessed 13 12 2022].