NATIONAL INSTITUTE OF BUSINESS MANAGEMENT
BSc (Hons) Computing |Year 2
HIGHER DIPLOMA IN SOFTWARE ENGINEERING – 21.1P

Internet Of Things - Course Work

SMART HOME
(“E-HOME”)

Group Members
Istina Suresh COHDSE211P-004
Peshala Nilmini COHDSE211P-011

Table of Contents

Table of Figures

SMART HOME - “E-HOME”

Abstract

This report describes the construction of a smart home system with a temperature monitoring system named “E-HOME” that has two main functionalities i.e., controlling two bulbs at home from a remote location via mobile app and user can see the current temperature and humidity of his house from any location. The bulbs at home can be turned ON/OFF by the user from any remote location using the mobile app. The bulbs are connected to relay modules which are connected to NodeMCU ESP8266. A DHT11 sensor is also connected to the NodeMCU to update the temperature and humidity of the room to the database. Widgets for each of these components are created on the Blynk dashboard and connected to data streams. This allows the user to control the bulbs and check the temperature and humidity via his mobile app.

Introduction

A smart home refers to a convenient home setup where appliances and devices can be automatically controlled remotely from anywhere with an internet connection using a mobile or other networked device. Devices in a smart home are interconnected through the internet, allowing the user to control functions such as security access to the home, temperature and lighting remotely. Smart homes can be set up through wireless or hardwired systems. Smart home technology provides homeowners with convenience and cost savings. Though full-scale home automation may cost thousands of dollars, smaller individual products costing less than $100 can get homeowners started on smart home products.

E-HOME is a prototype for a simple smart home system that enables the user to control two bulbs in two rooms, remotely via a mobile app. It has a temperature monitoring system that the user can use to view the temperature and humidity of his home from a remote location.

This report discusses the creation of a simple smart home system that possesses the above capabilities.

Methodology

Components used to create this smart home system are as follows:

- **NodeMCU ESP8266** - The NodeMCU (Node MicroController Unit) is an open source software and hardware development environment that is built around a very inexpensive System-on-a-Chip (SoC) called the ESP8266.

- **DHT 11 Sensor** - DHT11 is a low-cost digital sensor for sensing temperature and humidity. This sensor can be easily interfaced with any micro-controller such as Arduino, Raspberry Pi etc.… to measure humidity and temperature instantaneously.

- **Relay Module** - A power relay module is an electrical switch that is operated by an electromagnet. The electromagnet is activated by a separate low-power signal from a micro controller. When activated, the electromagnet pulls to either open or close an electrical circuit.

- **AC Bulbs**

- **Holders**

- **Jumper Cables**

- **Flexible Wires**

Method

The prototype has been assembled using the components mentioned above.

Following figures show the assembled prototype and the mobile app.

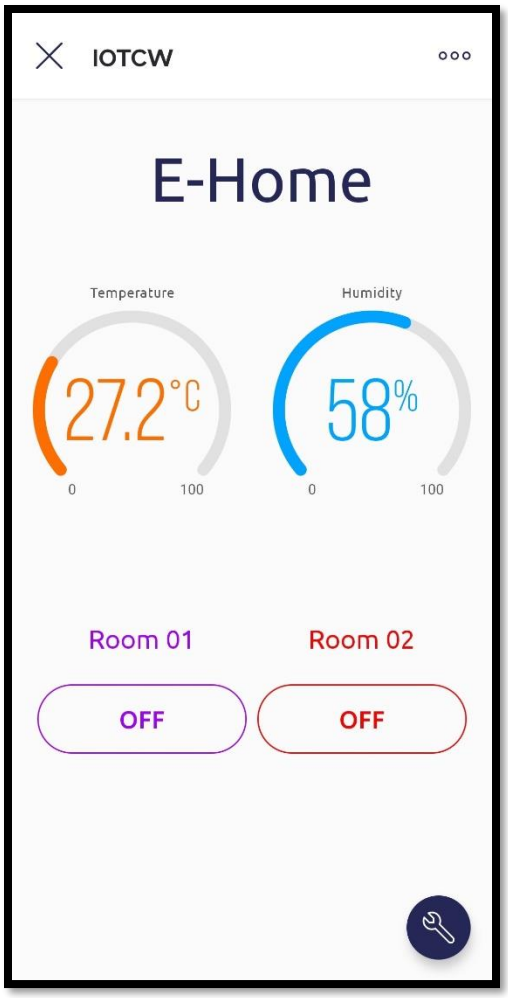

Figure 01 -Mobile App

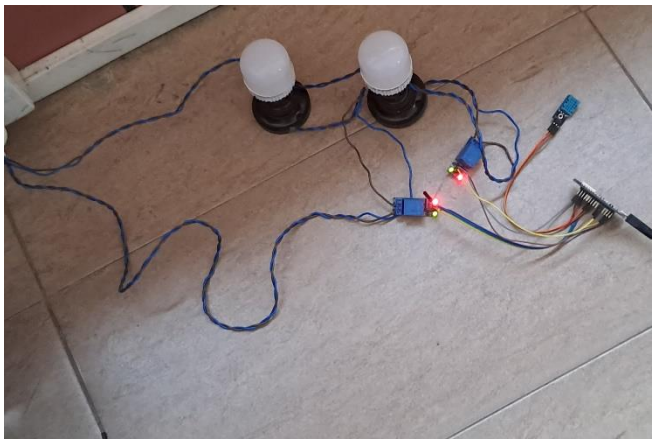

Figure 02 – Assembled Prototype

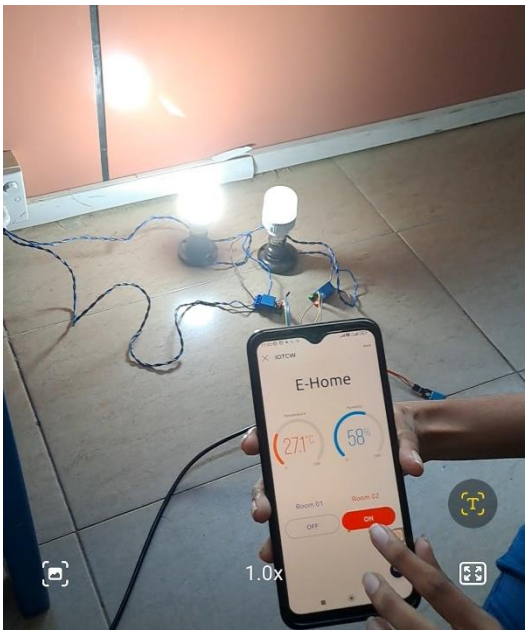

Figure 03 – Switched ON bulb using the mobile app

The workload of the project was carried out under a few phases:

1. **System Design:** We first designed the overall system architecture, which included the ESP8266 module, relay module, DHT11 sensor and bulbs. We chose the ESP8266 module because of its low cost, high processing power, and built-in Wi-Fi capabilities. We then selected a suitable relay module that could handle the required voltage and current of the bulbs.
2. **Hardware Setup:** We connected the bulbs to the relay module, and then connected the relay module and the DHT11 sensor to the ESP8266 module. We also connected a 5V DC power supply to power the system. We used jumper wires and a breadboard to make the connections.
3. **Programming:** We programmed the ESP8266 module using the Arduino IDE. We used the ESP8266WiFi library to connect the module to the local Wi-Fi network. We also used the Blynk IoT platform to create the mobile app and to connect the app with the ESP8266 module.
4. **Testing:** We conducted several tests to ensure the system was working as expected. We tested the connection between the ESP8266 module and the mobile app, and also tested the on/off switch of the bulbs using the mobile app. We tested if the sensor displayed the realtime temperature and humidity.
5. **Refinement:** We refined the system by improving the user interface of the mobile app, and by optimizing the code for better performance and reliability.
6. **Deployment:** We deployed the system in a real-world environment and tested it for a longer period to ensure reliability.

Discussion

In this project, we designed and implemented an smart home system that allows users to control the lighting of a room using their mobile app, as well as monitor the temperature and humidity using a DHT 11 sensor. The primary goal of this project was to create a user-friendly and affordable IoT system that can provide multiple functionalities in a home or small office.

We used an ESP8266 Wi-Fi module as the main microcontroller to connect the bulbs and the DHT 11 sensor with the mobile app. We also used relay modules to control the on/off switch of the bulbs.

To operate the system, the user needs to connect the bulbs to the relay module and the DHT11 sensor to the ESP8266 module. The ESP8266 module is then connected to the local Wi-Fi network. Once the connection is established, the user can download and install the mobile app on their smartphone, which allows them to control the bulbs and monitor the temperature and humidity.

Our smart home system provides an easy and convenient way to control the lighting of a room using a smartphone, as well as to monitor the temperature and humidity using the DHT 11 sensor. It eliminates the need for physical switches or complicated wiring, making it an ideal solution for smart homes or small offices.

Overall, our smart home system provides an affordable and user-friendly solution for controlling the lighting of a room and monitoring the temperature and humidity using a mobile app. By combining hardware and software tools, we were able to develop a functional and reliable smart home system that met our project objectives.

Future Implementations

In the future, we plan to extend the functionality of our IoT system to include more devices, such as air conditioners or heaters.

Our IoT system can be integrated with smart thermostats to automatically adjust the temperature and humidity levels based on the user's preferences. This can lead to greater energy efficiency and cost savings.

Integration with security systems: Our IoT system can be integrated with security systems to provide real-time monitoring of temperature and humidity levels in a home or office. This can help prevent damage caused by leaks or flooding and provide early warning of potential issues.

Integration with voice assistants: Our IoT system can be integrated with popular voice assistants such as Amazon Alexa or Google Assistant, allowing users to control the bulbs and monitor temperature and humidity using voice commands.

Expansion to other rooms and devices: Our IoT system can be expanded to include other rooms and devices, such as ceiling fans, air purifiers, or other appliances. This can provide users with a more comprehensive and integrated smart home solution.

Integration with energy management systems: Our IoT system can be integrated with energy management systems to monitor energy consumption and provide recommendations on how to reduce energy usage. This can help users save money on their utility bills while also reducing their environmental impact.

Overall, by continuously innovating and integrating new technologies, we can create an even more comprehensive and user-friendly smart home solution that meets the needs of modern households and businesses.

Gantt Chart

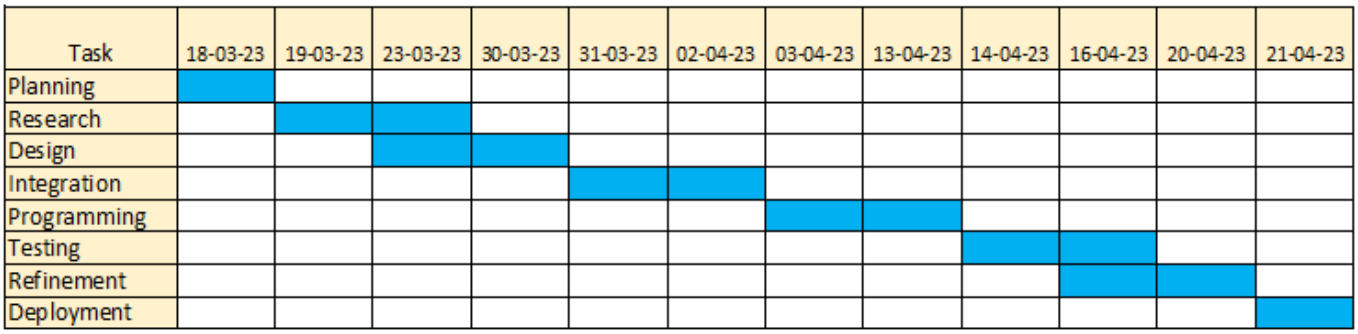

Figure 04

Acknowledgement

Firstly, we would like to thank our lecturer, Mr. A .Supun for his guidance and support.

We would also like to extend our appreciation to our colleagues and friends who provided us with technical assistance and moral support during the development of this project. Their input and feedback were critical in helping us to overcome challenges and achieve our objectives.

Furthermore, we are grateful to NIBM, who provided us with the necessary resources and facilities to carry out this project. Their support was essential in enabling us to complete the project successfully.

Finally, we would like to thank our families for their unwavering support and encouragement throughout this project. Their understanding and encouragement gave us the motivation to complete this project to the best of our abilities.

Once again, we express our heartfelt gratitude to everyone who contributed to the success of this project.

References

A light bulb switch using NodeMCU and the Blynk app (no date) *Gitbook.io*. Available at: https://reacoda.gitbook.io/molemi-iot/introducing-the-nodemcu/a-light-bulb-switch-using-nodemcu-and-the-blynk-app.

Das, M. (no date) *DHT11 and NodeMCU with blynk*, *Hackster.io*. Available at: https://www.hackster.io/Manoranjan2050/dht11-and-nodemcu-with-blynk-10e6b1.

Viral Science-The home of Creativity (2018) *Blynk ESP8266 DHT11 Temperature Sensor*. Youtube. Available at: https://www.youtube.com/watch?v=WLFUwyyPrKo.

Appendix 1

```
#define BLYNK_TEMPLATE_ID "TMPL6kljGaacd"
#define BLYNK_TEMPLATE_NAME "IOTCW"
#define BLYNK_AUTH_TOKEN "mKI1x60wDKzI3ZXJQqDHYtUMJvxRv9HO"

#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

#include <DHT.h>


char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "Istina";
char pass[] = "apple123";


BLYNK_WRITE(V0) {
  digitalWrite(D0, param.asInt());
}

BLYNK_WRITE(V1) {
  digitalWrite(D1, param.asInt());
}


BlynkTimer timer;

#define DHTPIN 4 //D2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);


void sendSensor()
{
  float h = dht.readHumidity();
```

```
  float t = dht.readTemperature(); //temperature shows in celcius

  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read  DHT sensor!");
    return;
  }

    Blynk.virtualWrite(V2, t);
    Blynk.virtualWrite(V3, h);
    Serial.print("Temperature : ");
    Serial.print(t);
    Serial.print("    Humidity : ");
    Serial.println(h);
}


void setup()
{

  Serial.begin(9600);

  pinMode(D0, OUTPUT);
  pinMode(D1, OUTPUT);

  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);


  dht.begin();
  timer.setInterval(7000L, sendSensor);

  }


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