CSE360 Project Report

Project Title:
Full Home Automation & Security System with NodeMCU ESP8266

Team Members
Shuchismita Anwar (ID: 20201067)
Tirthankar Saha Dibbya (ID: 20101409)
Tasfia Ayesha (ID: 20101478)
Tanjim Islam Riju (ID: 20101403)

Date: 20/04/2024

1 Introduction

.Home automation systems fundamentally transform how we interact with our living spaces. They offer unparalleled convenience, enhanced safety, and the potential for significant energy savings. This project focuses on designing and creating a "Full Home Automation and Security System" utilizing the affordable yet powerful NodeMCU ESP8266 board and the user-friendly Blynk mobile application. Our system has the ability to monitor various environmental and security factors. Real-time measurement of temperature and humidity provides crucial data for maintaining optimal indoor comfort. Flame detection plays a vital role in safety, alerting users to potential ignition that could pose a danger [1]. Water level monitoring can help prevent unnecessary overflows in water tanks or other reservoirs [4]. Additionally, incorporating a PIR motion sensor strengthens security by detecting unauthorized entry into the monitored environment [2]. Furthermore, our design takes safety and user experience into account. In the case of rising temperatures, sudden fire, water overflow, or unauthorized entry, the system will trigger a buzzer to alert occupants. The LCD display provides a local visual representation of the sensor readings, facilitating on-site monitoring.

- Temperature and Humidity
- Fire Detect
- Water Levels
- Security (unauthorized presence)

2 Application Area

- **Smart Homes:** Our project aligns with the growing trend of smart homes, making household tasks and monitoring more seamless and efficient.
- **Safety and Security:** The real-time monitoring of gas levels and unauthorized entry enhances safety.
- **Energy Conservation:** Potential for optimization based on temperature, humidity, and usage patterns.
- **Remote Accessibility:** Empowers users with enhanced control from anywhere for a more convenient living experience.

3 Technology and Tools

- **Hardware:**
  - NodeMCU ESP8266 Board

  - DHT11 Temperature and Humidity Sensor
  - Flame Sensor
  - Ultrasonic Sensor
  - LCD I2C Display
  - Buzzer
  - PIR Sensor
  - Breadboard
- **Software:**
  - Blynk App
  - Arduino IDE (for programming the NodeMCU)

4 Programming Language

- C++ (Within the Arduino IDE environment)

5 Working Mechanism of Sensors

The sensors are the frontline data collectors in our home automation and security system. Each type of sensor has a specific role and operates using unique principles:

- **DHT11:** This sensor consists of a capacitive humidity sensing element and a thermistor for measuring the ambient air's humidity and temperature, respectively. It integrates a high-performance 8-bit microcontroller for converting analog signals into a digital signal, which is then sent to the NodeMCU for further processing [6].
- **Ultrasonic Sensor:** By emitting ultrasonic waves at a frequency of 40 kHz, this sensor can measure the distance to an object by calculating the time interval between sending the signal and receiving the echo reflected off the object. The precision of the measurements allows the system to react to changes in proximity [3].
- **Flame Sensor:** Sensitive to a spectrum of wavelengths from 760 to 1100 nm, the flame sensor can detect the presence of a flame or other heat sources within this range [5]. The sensor's output can trigger an immediate alarm, activating the safety protocol if a fire is detected.
- **PIR Sensor:** This sensor detects infrared radiation from objects in its field of view. As living beings emit infrared energy due to body heat, the sensor can pick up motion when an intruder moves within its detection range. It sends a signal to the NodeMCU, initiating an alert if motion is detected [5].

6 Connection with ICs

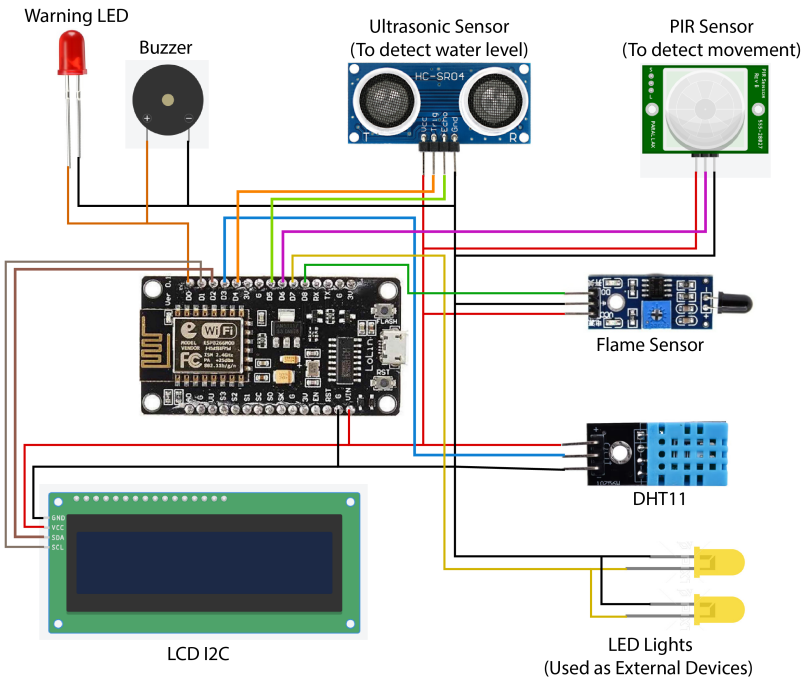


Figure 1: Main Diagram of The Project

7 Data Flow

1. **Sensor Readings:** Sensors continuously gather data (temperature, gas levels, etc.).
2. **Data Processing:** NodeMCU processes sensor data.
3. **LCD Display:** Temperature, Humidity, and water level values are displayed in real-time.
4. **Threshold Alerts:** NodeMCU compares data to preset thresholds. If exceeded, it triggers the buzzer.
5. **Blynk Transmission:** Data is sent to the Blynk app, offering visualization, warning notification, and remote control.
6. **User Input:** Users can send commands via the Blynk app to control appliances.

7. **NodeMCU Execution:** NodeMCU translates Blynk commands to control relays or other actuators.

   The warning LED and buzzer is connected with the D0 pin of ESP8266. It sends high voltage to trigger both LED and buzzer when any warning signal arrives. The trig and echo pins are connected to D4 and D5 respectively. The D4 sends a high signal to the trig pin and the sensor releases the sound through T and R receives the sound and passes the data from echo to D5 and the nodemcu processes the data and displays it to the LCD. The LCD I2C has an SDK pin which is connected to D2 to receive data from nodemcu. The SCL pin is connected to D1. The D1 sends a clock signal to the LCD for synchronization. The clock signal is controlled by the bus master (nodemcu). Furthermore, the PIR sensor has a data pin which is connected to D6. PIR sensor sends digital high to the nodemcu through D1 when it detects any movement. The flame sensor and DHT11 sensor are connected to D8 and D3 respectively. They also send digital high to the pins after detecting fire and temperature respectively. Again, D7 is connected to external devices (two LEDs). The high signal is sent to the device from the D7 to turn on the external devices.

8 Code

```
#include <LiquidCrystal_I2C.h>
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>

#define BLYNK_TEMPLATE_ID "TMPL6K8dXNgjm"
#define BLYNK_TEMPLATE_NAME "Home Automation"
#define BLYNK_AUTH_TOKEN "preE3cRscviPPvt2NwBuP4R_wONm6um2"

//Initialize the LCD display
LiquidCrystal_I2C lcd(0x27, 16, 2);

char auth[] = "preE3cRscviPPvt2NwBuP4R_wONm6um2";//Enter your Auth token
char ssid[] = "Morich";//Enter your WIFI name
char pass[] = "amarjhallage";//Enter your WIFI password

DHT dht(D3, DHT11); //(sensor pin,sensor type)
BlynkTimer timer;
bool pirbutton = 0;

// Define component pins
#define Buzzer D0
#define trig D4
```

```
#define echo D5
#define PIR D6
#define relay1 D7
#define fire D8

//Get buttons values
BLYNK_WRITE(V1) {
  pirbutton = param.asInt();
}

void setup() {
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  pinMode(Buzzer, OUTPUT);
  pinMode(PIR, INPUT);
  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT);
  pinMode(relay1, OUTPUT);
  pinMode(fire, INPUT);
  digitalWrite(relay1, LOW);
  lcd.print("Connecting...");
  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
  lcd.clear();
  dht.begin();

  lcd.setCursor(0, 0);
  lcd.print("Home Automation");
  lcd.setCursor(0, 1);
  lcd.print("Security System");
  delay(4000);
  lcd.clear();

//Call the functions
  timer.setInterval(100L, DHT11sensor);
  timer.setInterval(100L, pirsensor);
  timer.setInterval(100L, ultrasonic);
  timer.setInterval(100L, firesensor);
}

//Get the DHT11 sensor values
void DHT11sensor() {
  float h = dht.readHumidity();
  float t = dht.readTemperature();

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

```
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  Blynk.virtualWrite(V0, t);
  Blynk.virtualWrite(V2, h);

  lcd.setCursor(0, 0);
  lcd.print("T:");
  lcd.print(int(t));
  lcd.print("C");

  lcd.setCursor(8, 0);
  lcd.print("H:");
  lcd.print(int(h));
  lcd.print("%");

}

//Get the PIR sensor values
void pirsensor() {
  bool value = digitalRead(PIR);
  if (pirbutton == 1) {
    if (value == 0) {
      digitalWrite(Buzzer, LOW);
    } else if (value == 1) {
      Blynk.logEvent("theft_alert");
      digitalWrite(Buzzer, HIGH);
    }
  }
}

//Get the ultrasonic sensor values
void ultrasonic() {
  digitalWrite(trig, LOW);
  delayMicroseconds(4);
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);
  long t = pulseIn(echo, HIGH);
  long cm = t / 29 / 2;
  Blynk.virtualWrite(V4, cm);

  lcd.setCursor(0, 1);
  lcd.print("W:");
  lcd.print(cm);
  lcd.print("  ");
```

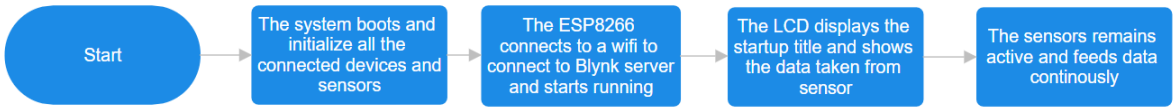
```
}

//Get buttons values
BLYNK_WRITE(V5) {
 bool RelayOne = param.asInt();
  if (RelayOne == 0) {
    digitalWrite(relay1, LOW);
  } else {
    digitalWrite(relay1, HIGH);
  }
}

//Fire sensor
void firesensor(){
int fire_det = digitalRead(fire);
if (fire_det == 1){
  digitalWrite(Buzzer, LOW);
 }
else if(fire_det == 0){
  digitalWrite(Buzzer, HIGH);
  Blynk.logEvent("fire_alert");
 }
 delay(1000);
}

void loop() {
  Blynk.run();//Run the Blynk library
  timer.run();//Run the Blynk timer
}
}
```

9 Estimated Cost Analysis

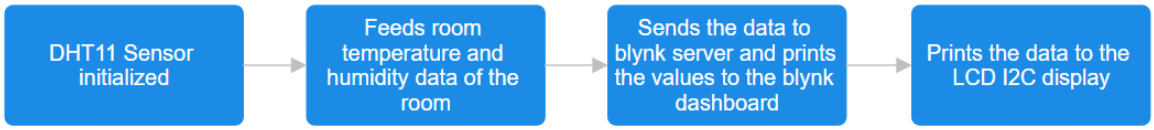
Item	Unit Price (BDT)	Quantity
Flame Sensor	69	1
PIR Sensor HC-SR501	95	1
NodeMCU ESP8266	650	1
DHT11 Sensor HW-036	185	1
Ultrasonic Sensor HC-SR04	93	1
LCD I2C Display	340	1
Buzzer	20	1
Breadboard	150	1
Jumper Wires	150	40
	Total:	**1752 BDT**

10 Workflow of the System

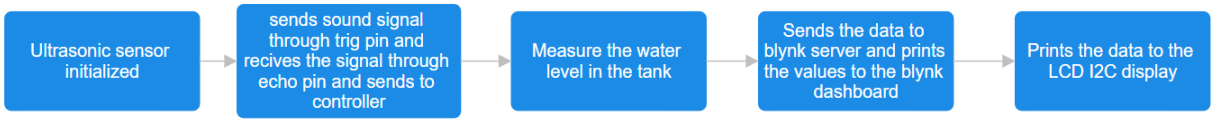
10.1 The whole system



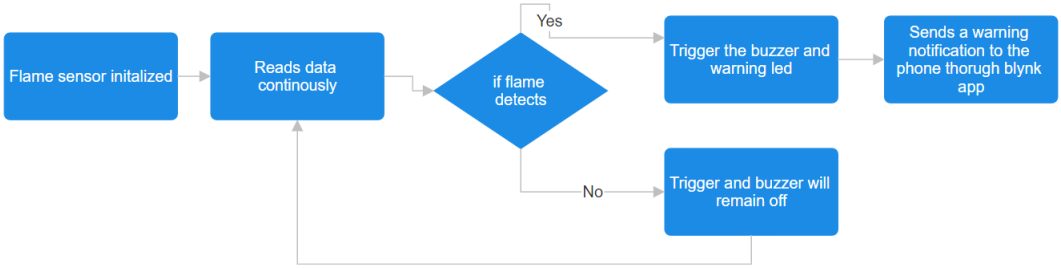
10.2 DHT 11 Sensor



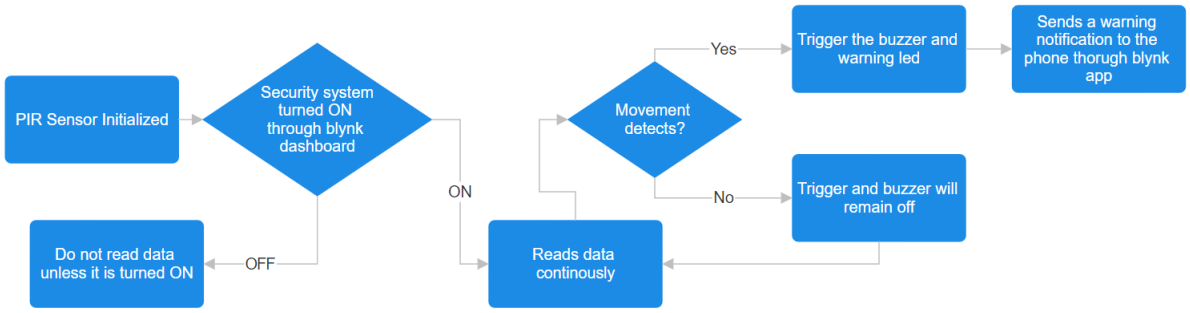
10.3 Flame Sensor



10.4 Ultrasonic Sensor

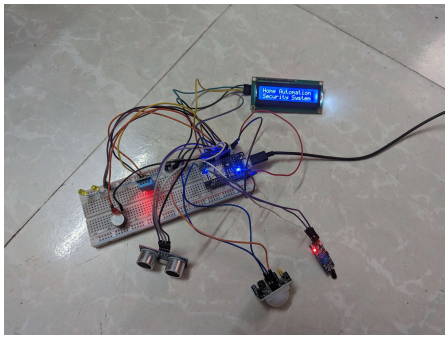


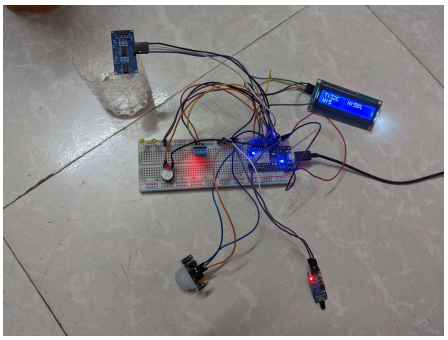
10.5 PIR Sensor

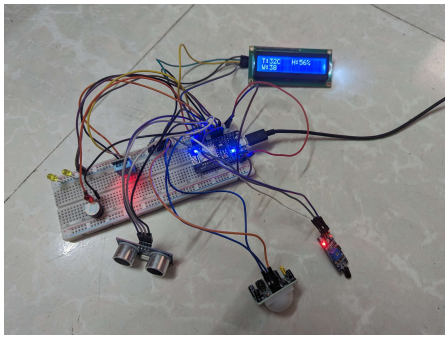


11 Pictures of The Project

ESP8266 is connecting to the wifi. After connecting to the Blynk server the system boots up and the title screen shows up.

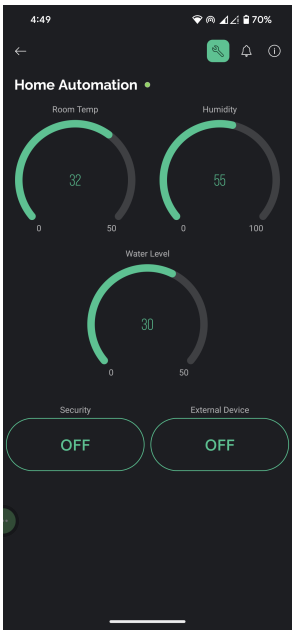


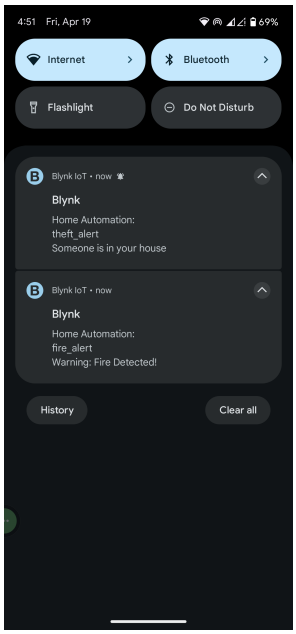


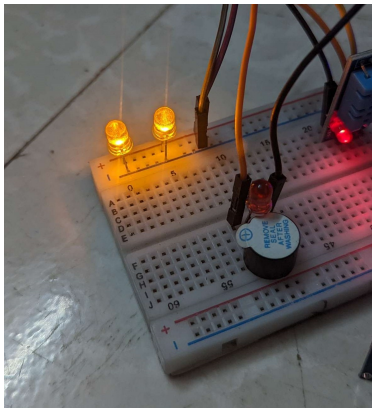


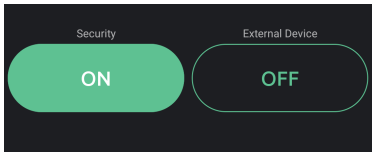
The system then runs all the sensors and the data are shown in the LCD display.

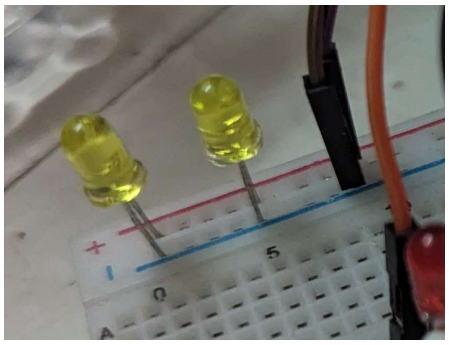
12 Blynk Interface



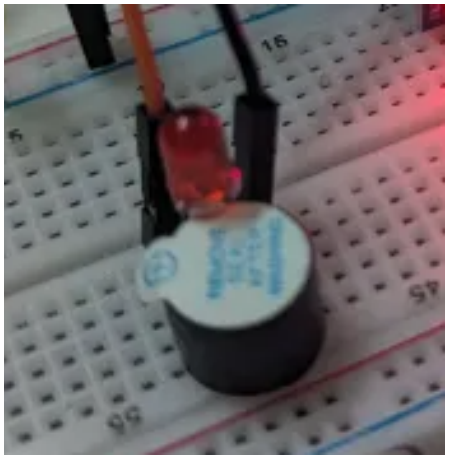


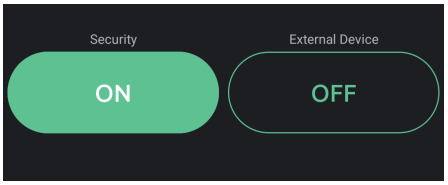




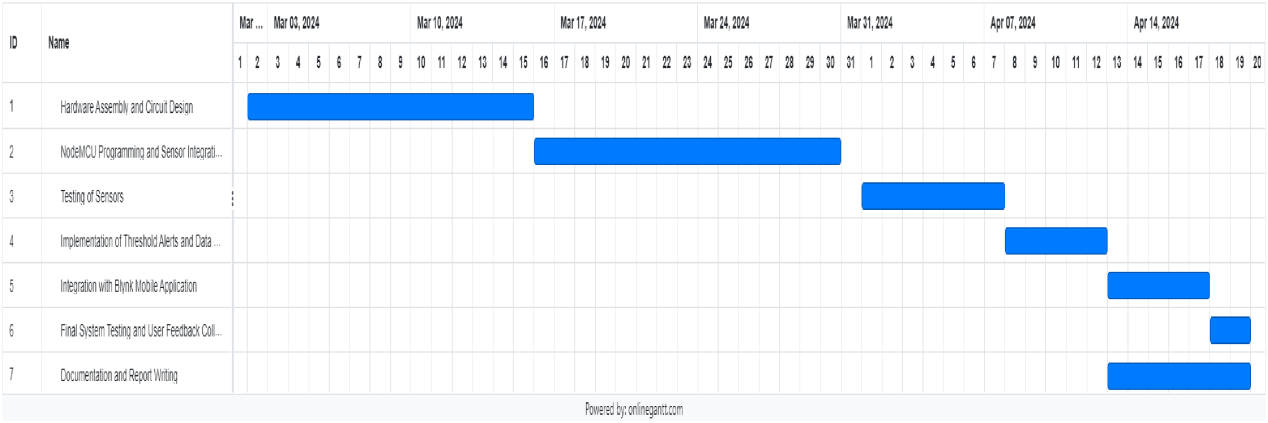



Security
External Device
OFF
OFF




Security
External Device
ON
OFF

13 Work Flow (Gantt Chart)



14 Responsibilities of Each Member

The successful completion of our project required a collaborative effort where each team member focused on different aspects:

- **Tirthankar Saha Dibbya, Tasfia Ayesha:**
  - Hardware assembly
  - Circuit design
- **Tanjim Islam Riju, Suchismita Anwar:**
  - NodeMCU programming
  - Sensor integrations
- **All Members:**
  - Project documentation
  - Report writing

15 Conclusion

This project presents a comprehensive home automation solution by showcasing effective sensor integration, proficient data processing, and seamless IoT connectivity utilizing the NodeMCU board and Blynk application. Looking forward, the system has the potential for numerous expansions such as the integration of voice control capabilities, the implementation of energy optimization algorithms, and the incorporation of additional types of sensor data to enhance functionality and user experience.

References

[1] Marina Artiyasa et al. "APLIKASI SMART HOME NODE MCU IOT UNTUK BLYNK". In: *Jurnal Rekayasa Teknologi Nusa Putra* 7.1 (Mar. 2021), pp. 1–7. DOI: 10.52005/rekayasa.v7i1.59. URL: https://rekayasa.nusaputra.ac.id/article/view/59.

[2] "Design and Implementation of an IOT Based Smart Home Monitoring and Control System Using NodeMCU". In: 25 (). DOI: 10.9734/jerr/2023/v25i2881. URL: https://mail.journaljerr.com/index.php/JERR/article/view/881.

[3] Ipin Prasojo, Phong Nguyen, and Omar Tanane. "Design of Ultrasonic Sensor and Ultraviolet". In: *International Journal of Robotics and Control* (Apr. 2021).

[4] K. Lova Raju et al. "Home Automation and Security System with Node MCU using Internet of Things". In: Mar. 2019, pp. 1–5. DOI: 10.1109/ViTECoN.2019.8899540.

[5] B. Ugur Toreyin et al. "Flame detection using PIR sensors". In: *2008 IEEE 16th Signal Processing, Communication and Applications Conference*. 2008, pp. 1–4. DOI: 10.1109/SIU.2008.4632660.

[6] Man-Ching Yuen et al. "A low-cost IoT smart home system". In: *International Journal of Engineering and Technology(UAE)* 7 (Jan. 2018), pp. 3143–3147. DOI: 10.14419/ijet.v7i4.18862.