Home automation through Google Voice assistant

Sher Ali, *Semester IV, EE-1497*
Hassan Shakeel Semester IV EE-1462

***Abstract*—This is a post-completion analysis report for a custom Home Automation system using the Node MCU Micro-Controller. We had to make a few amendments to our original concept as to appear more appealing to potential customers and to cut production costs. Most of the originally intended features were implemented with some minor hardware and software revisions. We have personally used this device and consider to be an exceptionally useful addition to our daily lives. The cost was kept under the proposed budget at 80% cheaper than market competitors.**

I. INTRODUCTION

THIS report outlines the features of our product and its comparison to market competitors while keeping the costs in mind.

II. Features

The Main objective of this project is providing a cheaper alternative to Smart home systems that are already available on the market. Our implementation uses the Node MCU ESP8266 integrated with the Blynk API and IFTTT for Google Voice integration.

1. Temperature and Humidity Monitoring and Logging.
2. Supports backup battery power and External power through an adapter.
3. Smart Control for HVAC systems with Target Temperature control.
4. Smoke Detection system with “in app notifications”.
5. Voice commands

III. Comparison with Google Home

Feature	Home Automation -/1400 Rs	Google Home -/6999 Rs
Appliance Power Control	✓	✓
Temperature Monitoring	✓	✓
Temperature Control	✓	✓
CO2 Monitoring	✓	X

Further Possible additions

There were further features that we couldn’t add due to time limitations. Some of them have been listed below:

1. Automatic Light Control using presence detection.
2. Daily weather report using speakers.
3. Task Reminders.

IV. FUTURE TIMELINE

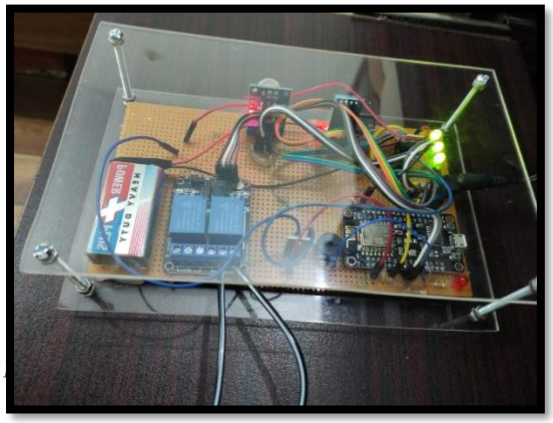
Addition of the remaining features and further optimizations should take 1-2 weeks. We will work towards further improvements and will try to implement the suggestions from our instructor and peers.

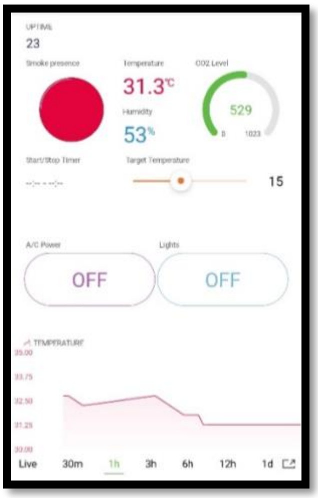
V. COST ANALYSIS

Component	Cost	Mass Production Cost
Node MCU	450	350
Relay Module	200	180
DHT-11 Sensor	220	200
MQ-135 Sensor	280	250
LM7805 IC	15	15
Buzzer	25	25
LEDs	20	20
IRF740 MOSFET	65	60
Vero Board	50	40
Power Adapter	150	100
Casing	150	80
Wiring / Misc.	100	80
TOTAL	1,725	1,400

80% cheaper compared to commercially available units with better functionality and customization.

VI. PRODUCT PICTURES





UPTIME
23

Smoke presence

Temperature
31.3℃

Humidity
53%

CO2 Level
529

Start/Stop Timer
--:-- - --:--

Target Temperature
15

A/C Power
OFF

Lights
OFF

Figure 2 The App GUI

VII. ESP8266 Code:

```
//Main Code EDC
#define BLYNK_TEMPLATE_ID "TMPL6PxbI2l-"
#define BLYNK_DEVICE_NAME "Template"
#define BLYNK_AUTH_TOKEN "LebNqMgyfELn3EQv3ub0RuwIx6t2aPZN"
#define BLYNK_PRINT Serial

#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>
#define DHTPIN D4
#define AcRelay D5
#define GasRelay D1
#define LightRelay D6
#define DHTTYPE DHT11
#define Buzzer D2

int mq135 = A0;
int mqR = 22000;
int data = 0;

#define irPin1 D2
#define irPin2 D3
int count = 0;
int i = 1;
unsigned long int irTime1 = 0;
unsigned long int irTime2 = 0;

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

char auth[] = BLYNK_AUTH_TOKEN;

char ssid[] = "Hmmmm";
char pass[] = "PASSWORD1126";

BlynkTimer timer;
DHT_Unified dht(DHTPIN, DHTTYPE);

// This function is called every time the Virtual Pin 0 state changes

BLYNK_WRITE(V0)
{
  // Set incoming value from pin V0 to a variable
  int value = param.asInt();
  // Update state
  Blynk.virtualWrite(V1, value);
}
BLYNK_WRITE(V8)
{
  int Ac = param.asInt(); // assigning incoming value from pin V1 to a variable
  Serial.print("V8 Button value is");
  Serial.println(Ac);
  if (Ac == 1) {digitalWrite(AcRelay, HIGH);}
  else {digitalWrite(AcRelay, LOW);}}
BLYNK_WRITE(V11)
{int Light = param.asInt(); // assigning incoming value from pin V1 to a variable

  Serial.print("V11 Button value is");
  Serial.println(Light);
  if (Light == 1) { digitalWrite(LightRelay, HIGH); }
  else {digitalWrite(LightRelay, LOW);}}
BLYNK_WRITE(V12)
{int TargetTemp = param.asInt(); // assigning incoming value from pin V1 to a
variable
  Serial.print("V12 Button value is");
  Serial.println(TargetTemp);
  sensors_event_t event;
  dht.temperature().getEvent(&event);
  float t = event.temperature;
  if (TargetTemp == 0) {}
  else if (TargetTemp == t) {}
  else if (TargetTemp == t + 1) {}
  else if (TargetTemp == t - 1) {}
  //else if(TargetTemp==t+2){}
  //else if(TargetTemp==t-2){}
  else if (TargetTemp > t) {
    digitalWrite(AcRelay, HIGH);}
```

```
  }
}

// This function sends Arduino's uptime every second to Virtual Pin
2.
void myTimerEvent()
{
  Blynk.virtualWrite(V2, millis() / 1000);
}
void mq135readsend()
{
  data = analogRead(mq135);
  data = (data * 675) / 1023;
  Blynk.virtualWrite(V9, data);
  Serial.print(data);
  if (data > 550) {
    Blynk.virtualWrite(V10, 1);
    digitalWrite(GasRelay, HIGH);
    digitalWrite(Buzzer, HIGH);
    delay(250);
    digitalWrite(Buzzer, LOW);
    delay(250);
  }
  else {
    Blynk.virtualWrite(V10, 0);
    digitalWrite(GasRelay, LOW);  }}
void dhtreadsend(){  sensors_event_t event;
  dht.temperature().getEvent(&event);
  if (isnan(event.temperature)) {
    Serial.println(F("Error reading temperature!"));}
  else { float t = event.temperature;
    Blynk.virtualWrite(V6, t);  //V6 is for Temperature
    //Serial.print(F("Temperature: "));
    //Serial.print(event.temperature);
    //Serial.println(F("Â°C"));  }
  dht.humidity().getEvent(&event);
  if (isnan(event.relative_humidity)) {
    Serial.println(F("Error reading humidity!")); }
  else {  //Serial.print(F("Humidity: "));
    //Serial.print(event.relative_humidity);
    //Serial.println(F("%"));
    float h = event.relative_humidity;
    Blynk.virtualWrite(V5, h);  //V5 is for Humidity
  }}
void setup()
{
 // Debug console
  Serial.begin(115200);
  tone(D2, 4000, 100);
  delay(100);
  tone(D2, 4000, 100);
  Blynk.begin(auth, ssid, pass);
  pinMode(irPin1, INPUT);
  pinMode(irPin2, INPUT);
  pinMode(AcRelay, OUTPUT);
  pinMode(LightRelay, OUTPUT);
  pinMode(GasRelay, OUTPUT);
  pinMode(Buzzer, OUTPUT);
  digitalWrite(AcRelay, HIGH);
  digitalWrite(LightRelay, HIGH);
  //digitalWrite(Buzzer,HIGH);
  //delay(100);
  //digitalWrite(Buzzer,LOW);
  //delay(100);

  // Setup a function to be called every second
  dht.begin();
  timer.setInterval(1000L, myTimerEvent); // V2 used
  timer.setInterval(1000L, mq135readsend);  // V4 mq135
  timer.setInterval(1000L, dhtreadsend); // V5 H , V6 T}
void loop()
{ Blynk.run();
  timer.run();
}
```