**ANNA UNIVERSITY**

**MIT CAMPUS**

**DEPARTMENT OF COMPUTER TECHNOLOGY**

EE6351

BASICS OF ELECTRICAL AND ELECTRONIC ENGINEERING

SMART HOME

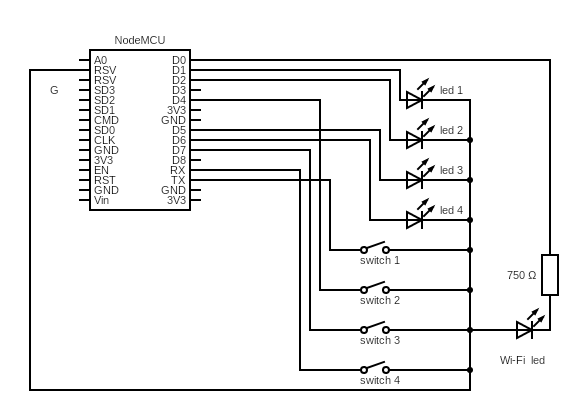
PROJECT REPORT

*SMART HOME*

NTRODUCTION:

* A smart home allows homeowners to control appliances, thermostats, lights, and other devices remotely using a smartphone or tablet through an internet connection.
* Smart homes can be set up through wireless or hardwired systems.
* Smart home technology provides homeowners with convenience and cost savings.
* Putting in a wireless home automation system with features such as smart lighting, climate control, and security can cost several thousand dollars, making it very cost-friendly.

CIRCUIT DIAGRAM:



APPARATUS REQUIRED:

* BreadBoard
* LEDs
* Switches
* Connecting wires
* NodeMCU
* Voltage supply

CONSTRUCTION:

1. Using Aurdino IDE, program your NodeMCU with required C++ codes to perform your desired task.
2. Upload the codes to the nodeMCU.
3. Take a breadboard and connect the components like LEDs, switches and NodeMCU with connecting wires (male pins and female pins) as given in the circuit diagram.
4. Additionaly we use a blue LED to check whether the microcontroller is connected with external wifi or not.
5. Install Blynk IOT app in our smart phone.
6. Connect the nodeMCU with voltage supply.
7. Check whether the circuit is working wirelessly via internet connection or not.

WORKING:

After successful completion of coding in Aurdino IDE with C++ codes we upload it in nodeMCU. Using breadboard we connect the components as given in the circuit. Blynk app is installed in our smart phone for having control over the devices wirelessly at any distance through internet. After completing successfully it is possible to turn ON or OFF through our smart phone and it is also possible to control manually with switches.

PROGRAM:

#define BLYNK\_TEMPLATE\_ID "TMPLIQRz8A8e"

#define BLYNK\_DEVICE\_NAME "BEEE NodeMCU"

#define BLYNK\_AUTH\_TOKEN "ht1dG8x1dYJONGmm1Kf-\_OfvFaGs9nEz"

char ssid[] = "darshanking";

char pass[] = "techy1234";

bool fetch\_blynk\_state = true;

//#define BLYNK\_PRINT Serial

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

#include <AceButton.h>

using namespace ace\_button;

#define echoPin1 9 //S2

#define echoPin2 10 //S3

#define trigPin1 0 //S0

#define trigPin2 15 //S1

long duration1;

int distance1;

long duration2;

int distance2;

bool u1 = false;

bool u2 = false;

// define the GPIO connected with Relays and switches

#define RelayPin1 5 //D1

#define RelayPin2 4 //D2

#define RelayPin3 14 //D5

#define RelayPin4 12 //D6

#define SwitchPin1 1 //TX

#define SwitchPin2 2 //D4

#define SwitchPin3 13 //D7

#define SwitchPin4 3 //RX

#define wifiLed 16 //D0

//Change the virtual pins

#define VPIN\_BUTTON\_1 V5

#define VPIN\_BUTTON\_2 V6

#define VPIN\_BUTTON\_3 V7

#define VPIN\_BUTTON\_4 V8

#define VPIN\_BUTTON\_5 V9

// Relay State

bool toggleState\_1 = LOW; //Define integer to remember the toggle state for relay 1

bool toggleState\_2 = LOW; //Define integer to remember the toggle state for relay 2

bool toggleState\_3 = LOW; //Define integer to remember the toggle state for relay 3

bool toggleState\_4 = LOW; //Define integer to remember the toggle state for relay 4

int wifiFlag = 0;

char auth[] = BLYNK\_AUTH\_TOKEN;

ButtonConfig config1;

AceButton button1(&config1);

ButtonConfig config2;

AceButton button2(&config2);

ButtonConfig config3;

AceButton button3(&config3);

ButtonConfig config4;

AceButton button4(&config4);

void handleEvent1(AceButton\*, uint8\_t, uint8\_t);

void handleEvent2(AceButton\*, uint8\_t, uint8\_t);

void handleEvent3(AceButton\*, uint8\_t, uint8\_t);

void handleEvent4(AceButton\*, uint8\_t, uint8\_t);

BlynkTimer timer;

// When App button is pushed - switch the state

BLYNK\_WRITE(VPIN\_BUTTON\_1) {

toggleState\_1 = param.asInt();

digitalWrite(RelayPin1, toggleState\_1);

}

BLYNK\_WRITE(VPIN\_BUTTON\_2) {

toggleState\_2 = param.asInt();

digitalWrite(RelayPin2, toggleState\_2);

}

BLYNK\_WRITE(VPIN\_BUTTON\_3) {

toggleState\_3 = param.asInt();

digitalWrite(RelayPin3, toggleState\_3);

}

BLYNK\_WRITE(VPIN\_BUTTON\_4) {

toggleState\_4 = param.asInt();

digitalWrite(RelayPin4, toggleState\_4);

}

BLYNK\_WRITE(VPIN\_BUTTON\_5) {

toggleState\_1 = 0; digitalWrite(RelayPin1, LOW); Blynk.virtualWrite(VPIN\_BUTTON\_1, toggleState\_1); delay(100);

toggleState\_2 = 0; digitalWrite(RelayPin2, LOW); Blynk.virtualWrite(VPIN\_BUTTON\_2, toggleState\_2); delay(100);

toggleState\_3 = 0; digitalWrite(RelayPin3, LOW); Blynk.virtualWrite(VPIN\_BUTTON\_3, toggleState\_3); delay(100);

toggleState\_4 = 0; digitalWrite(RelayPin4, LOW); Blynk.virtualWrite(VPIN\_BUTTON\_4, toggleState\_4); delay(100);

}

void checkBlynkStatus() {

bool isconnected = Blynk.connected();

if (isconnected == false) {

wifiFlag = 1;

Serial.println("Blynk Not Connected");

digitalWrite(wifiLed, LOW);

digitalWrite(RelayPin1,LOW);

digitalWrite(RelayPin2,LOW);

digitalWrite(RelayPin3,LOW);

digitalWrite(RelayPin4,LOW);

}

if (isconnected == true) {

wifiFlag = 0;

if (!fetch\_blynk\_state){

Blynk.virtualWrite(VPIN\_BUTTON\_1, toggleState\_1);

Blynk.virtualWrite(VPIN\_BUTTON\_2, toggleState\_2);

Blynk.virtualWrite(VPIN\_BUTTON\_3, toggleState\_3);

Blynk.virtualWrite(VPIN\_BUTTON\_4, toggleState\_4);

}

digitalWrite(wifiLed, HIGH);

Serial.println("Blynk Connected");

}

}

BLYNK\_CONNECTED() {

// Request the latest state from the server

if (fetch\_blynk\_state){

Blynk.syncVirtual(VPIN\_BUTTON\_1);

Blynk.syncVirtual(VPIN\_BUTTON\_2);

Blynk.syncVirtual(VPIN\_BUTTON\_3);

Blynk.syncVirtual(VPIN\_BUTTON\_4);

}

}

void setup()

{

Serial.begin(9600);

pinMode(trigPin1, OUTPUT);

pinMode(echoPin1, INPUT);

pinMode(trigPin2, OUTPUT);

pinMode(echoPin2, INPUT);

pinMode(RelayPin1, OUTPUT);

pinMode(RelayPin2, OUTPUT);

pinMode(RelayPin3, OUTPUT);

pinMode(RelayPin4, OUTPUT);

pinMode(wifiLed, OUTPUT);

pinMode(SwitchPin1, INPUT\_PULLUP);

pinMode(SwitchPin2, INPUT\_PULLUP);

pinMode(SwitchPin3, INPUT\_PULLUP);

pinMode(SwitchPin4, INPUT\_PULLUP);

//During Starting all Relays should TURN OFF

digitalWrite(RelayPin1, !toggleState\_1);

digitalWrite(RelayPin2, !toggleState\_2);

digitalWrite(RelayPin3, !toggleState\_3);

digitalWrite(RelayPin4, !toggleState\_4);

digitalWrite(wifiLed, LOW);

config1.setEventHandler(button1Handler);

config2.setEventHandler(button2Handler);

config3.setEventHandler(button3Handler);

config4.setEventHandler(button4Handler);

button1.init(SwitchPin1);

button2.init(SwitchPin2);

button3.init(SwitchPin3);

button4.init(SwitchPin4);

//Blynk.begin(auth, ssid, pass);

WiFi.begin(ssid, pass);

timer.setInterval(2000L, checkBlynkStatus); // check if Blynk server is connected every 2 seconds

Blynk.config(auth);

delay(1000);

if (!fetch\_blynk\_state){

Blynk.virtualWrite(VPIN\_BUTTON\_1, toggleState\_1);

Blynk.virtualWrite(VPIN\_BUTTON\_2, toggleState\_2);

Blynk.virtualWrite(VPIN\_BUTTON\_3, toggleState\_3);

Blynk.virtualWrite(VPIN\_BUTTON\_4, toggleState\_4);

}

}

void loop()

{

Blynk.run();

timer.run(); // Initiates SimpleTimer

button1.check();

button2.check();

button3.check();

button4.check();

ultrasonic();

}

void button1Handler(AceButton\* button, uint8\_t eventType, uint8\_t buttonState) {

Serial.println("EVENT1");

switch (eventType) {

case AceButton::kEventReleased:

Serial.println("kEventReleased");

digitalWrite(RelayPin1, !toggleState\_1);

toggleState\_1 = !toggleState\_1;

Blynk.virtualWrite(VPIN\_BUTTON\_1, toggleState\_1);

break;

}

}

void button2Handler(AceButton\* button, uint8\_t eventType, uint8\_t buttonState) {

Serial.println("EVENT2");

switch (eventType) {

case AceButton::kEventReleased:

Serial.println("kEventReleased");

digitalWrite(RelayPin2, !toggleState\_2);

toggleState\_2 = !toggleState\_2;

Blynk.virtualWrite(VPIN\_BUTTON\_2, toggleState\_2);

break;

}

}

void button3Handler(AceButton\* button, uint8\_t eventType, uint8\_t buttonState) {

Serial.println("EVENT3");

switch (eventType) {

case AceButton::kEventReleased:

Serial.println("kEventReleased");

digitalWrite(RelayPin3, !toggleState\_3);

toggleState\_3 = !toggleState\_3;

Blynk.virtualWrite(VPIN\_BUTTON\_3, toggleState\_3);

break;

}

}

void button4Handler(AceButton\* button, uint8\_t eventType, uint8\_t buttonState) {

Serial.println("EVENT4");

switch (eventType) {

case AceButton::kEventReleased:

Serial.println("kEventReleased");

digitalWrite(RelayPin4, !toggleState\_4);

toggleState\_4 = !toggleState\_4;

Blynk.virtualWrite(VPIN\_BUTTON\_4, toggleState\_4);

break;

}

}

**TEAM MEMBERS**

Ajay Kumar K (2021503003)

Babith Sarish S (2021503009)

Darshan B (2021503509)

Siva Jegadeesh CB (2021503559)

Raj Kumar M (202503039)

Prasanna M (202503535)