```
#define BLYNK_TEMPLATE_ID "TMPL30S9KaGCX"
#define BLYNK_TEMPLATE_NAME "alertsys"
#define BLYNK_AUTH_TOKEN "Nt0tB8h3gwMYuvKr4SfFAxHKfSz5f2H_"

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


#include <DHT.h>
 #include <LiquidCrystal_I2C.h>



LiquidCrystal_I2C lcd(0x27, 16, 2);



  byte degree_symbol[8] =

            {

              0b00111,

              0b00101,

              0b00111,

              0b00000,

              0b00000,

              0b00000,

              0b00000,

              0b00000

            };



char auth[] = BLYNK_AUTH_TOKEN;

char ssid[] = "Himadri2g";  // type your wifi name
char pass[] = "Chyavan2003";  // type your wifi password

BlynkTimer timer;

 //Connect Out pin to D2 in NODE MCU
#define DHTTYPE DHT11
DHT dht(D6, DHTTYPE);


void sendSensor()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  float m=analogRead(A0);
```

```
  int f= digitalRead(D5);// or dht.readTemperature(true) for Fahrenheit

  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  // You can send any value at any time.
  // Please don't send more that 10 values per second.
    Blynk.virtualWrite(V0, t);
    Blynk.virtualWrite(V1, h);
     Blynk.virtualWrite(V2, m);
     Blynk.virtualWrite(V3, f);
    Serial.print("Temperature : ");
    Serial.print(t);
    Serial.print("    Humidity : ");
    Serial.println(h);
}
void setup()
{
    lcd.begin();
  lcd.backlight();
  Serial.begin(115200);
  pinMode(D8,OUTPUT);

  Blynk.begin(auth, ssid, pass);
  dht.begin();
  timer.setInterval(100L, sendSensor);

  }

void loop()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  float m=analogRead(A0);
  int f= digitalRead(D5);
    lcd.setCursor(0, 0);
  lcd.print("humidity");
   lcd.setCursor(0, 1);
  lcd.print(h);
  delay(1000);
  lcd.clear();
      lcd.setCursor(0, 0);
  lcd.print("Temp");
   lcd.setCursor(0, 1);
  lcd.print(t);
  delay(1000);
    lcd.clear();
      lcd.setCursor(0, 0);
  lcd.print("air quality");
   lcd.setCursor(0, 1);
  lcd.print(m);

  delay(1000);
     lcd.clear();
      lcd.setCursor(0, 0);
  lcd.print("fire");
   lcd.setCursor(0, 1);
  lcd.print(f);
  delay(1000);
```

```
   lcd.clear();
  if(f==0)
  {
digitalWrite(D8,HIGH);
  }
  else if(f==1){
  digitalWrite(D8,LOW);
  }
  Blynk.run();
  timer.run();
 }
```

```
#include <AFMotor.h>

//defining pins and variables
#define left A0
#define right A1

//defining motors
AF_DCMotor motor1(1, MOTOR12_1KHZ);
AF_DCMotor motor2(2, MOTOR12_1KHZ);
AF_DCMotor motor3(3, MOTOR34_1KHZ);
AF_DCMotor motor4(4, MOTOR34_1KHZ);



void setup() {
  //declaring pin types
  pinMode(left,INPUT);
  pinMode(right,INPUT);
  //begin serial communication
  Serial.begin(9600);

}

void loop(){
  //printing values of the sensors to the serial monitor
  Serial.println(digitalRead(left));

  Serial.println(digitalRead(right));

  //line detected by both
  if(digitalRead(left)==0 && digitalRead(right)==0){
    //Forward
    motor1.run(FORWARD);
    motor1.setSpeed(100);
    motor2.run(FORWARD);
    motor2.setSpeed(100);
    motor3.run(FORWARD);
    motor3.setSpeed(100);
    motor4.run(FORWARD);
    motor4.setSpeed(100);
  }
  //line detected by left sensor
  else if(digitalRead(left)==0 && !analogRead(right)==0){
    //turn left
```

```
    motor1.run(FORWARD);
    motor1.setSpeed(200);
    motor2.run(FORWARD);
    motor2.setSpeed(200);
    motor3.run(BACKWARD);
    motor3.setSpeed(200);
    motor4.run(BACKWARD);
    motor4.setSpeed(200);

  }
  //line detected by right sensor
  else if(!digitalRead(left)==0 && digitalRead(right)==0){
    //turn right
    motor1.run(BACKWARD);
    motor1.setSpeed(200);
    motor2.run(BACKWARD);
    motor2.setSpeed(200);
    motor3.run(FORWARD);
    motor3.setSpeed(200);
    motor4.run(FORWARD);
    motor4.setSpeed(200);

  }
  //line detected by none
  else if(!digitalRead(left)==0 && !digitalRead(right)==0){
    //stop
    motor1.run(RELEASE);
    motor1.setSpeed(0);
    motor2.run(RELEASE);
    motor2.setSpeed(0);
    motor3.run(RELEASE);
    motor3.setSpeed(0);
    motor4.run(RELEASE);
    motor4.setSpeed(0);

  }

}




#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Keypad.h>
#include <Servo.h>

// Initialize LCD with I2C address 0x27
LiquidCrystal_I2C lcd(0x27, 16, 2);

// Servo objects
Servo servo1;
Servo servo2;

// Keypad setup
const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
```

```
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};
byte rowPins[ROWS] = {2, 3, 4, 5}; // Connect to R1-R4
byte colPins[COLS] = {6, 7, 8, 9}; // Connect to C1-C4
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

void setup() {
  // Servo setup
  servo1.attach(10);  // Servo 1 pin
  servo2.attach(11); // Servo 2 pin
  servo1.write(0);   // Initial position
  servo2.write(0);

  // LCD setup
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Servo Control");
}

void loop() {
  char key = keypad.getKey();

  if (key) {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Key Pressed: ");
    lcd.setCursor(12, 0);
    lcd.print(key);

    if (key == '1') {
      lcd.setCursor(0, 1);
      lcd.print("Patient 1");
      rotateServo(servo1);
    } else if (key == '2') {
      lcd.setCursor(0, 1);
      lcd.print("Patient 2");
      rotateServo(servo2);
    }
  }
}

// Function to rotate servo from 0 to 90 and back
void rotateServo(Servo &servo) {
  servo.write(90);         // Rotate to 90 degrees
  delay(5000);             // Wait for 5 seconds
  servo.write(0);          // Return to 0 degrees
  delay(500);              // Small delay to stabilize
}
```