| Tên cảm biến | Chân | Read value |
| --- | --- | --- |
| Cảm biến ánh sáng LM393 | VCC GND D0 (sáng = LOW) A0 ( sáng >> thấp) | 1.#include <dht.h>  DHT dht(DHT\_PIN, DHT11);  float h = dht.readHumidity();;  float t = dht.readTemperature(); 2. write trig low, delay 2us, trig high, delay 10us, trig low.  int d = pulseIn(echo, 1);  double cm = d / 2 / 29.1;  D~ 3 5 6 9 10 11 |
| Cảm biến nhiệt độ analog LM35 | VCC GND A0 ( t > value >) |
| Cảm biến dò line | VCC GND D0 A0 ( trắng càng cao) |
| Cảm biến siêu âm HC-SR04 (2) | VCC GND TRIG 6 ECHO 7 |
| Joystick sw input\_pullup | VCC GND (VRxVRy-A) SW(D) |
| Cảm biến Gas MQ2 | VCC GND A0 D0 |

MotorDC và L298N

OUT1,2 -> Motor1 VCC GND 5V ENA9 (IN1 HIGH | IN2 LOW | Quay tới) (IN1 LOW | IN2 HIGH | lui)

| const int enA = 9;  const int in1 = 8;  const int in2 = 7;  void setup() {  pinMode(enA, OUTPUT);  pinMode(in1, OUTPUT);  pinMode(in2, OUTPUT);  // lúc đầu tắt motor  digitalWrite(in1, LOW); digitalWrite(in2, LOW);} | void loop() {  // quay tới ở max tốc độ  digitalWrite(in1, HIGH); digitalWrite(in2, LOW);  analogWrite(enA, 255);  delay(2000);  // quay lui, tốc độ tăng dần  digitalWrite(in1, LOW); digitalWrite(in2, HIGH); | for (int sp=0; sp<256; sp++) {  analogWrite(enA, sp);  delay(20);  }  // tắt động cơ trong 2s  digitalWrite(in1, LOW); digitalWrite(in2, LOW);  delay(2000);  } |
| --- | --- | --- |

Stepper 28BYJ-28 IN1-4 -> D8-11 MotorConnection VCC GND và ULN2003

| #include <Stepper.h>  #define STEPS 2038 // số bước trong một vòng quay  // Số trên là hằng số với mỗi loại Stepper Motor.  int degreeToSteps(int degree, int STEPS = 2038) {  if (degree == 0) return 0;  return STEPS / (360 / degree); | Stepper stepper(STEPS, 8, 10, 9, 11);  void setup() {  // không cần làm gì cả, thư viện Stepper sẽ lo setup động cơ cho ta}  void loop() {  stepper.setSpeed(3); // 3 rpm | stepper.step(STEPS); // quay một vòng cùng chiều đồng hồ  delay(1000);  stepper.setSpeed(6); // 6 rpm  stepper.step(-STEPS); // quay một vòng ngược chiều đồng hồ  delay(1000);  } |
| --- | --- | --- |

SERVO SG90: GND nâu 5V đỏ Control vàng D9~

| #include <Servo.h>  Servo myservo;  const int ServoPin = 9;  void setup() { myservo.attach(ServoPin);} | void loop() {  for (pos = 0; pos <= 180; pos += 1) {  myservo.write(pos);  delay(15); } | for (pos = 180; pos >= 0; pos -= 1) {  myservo.write(pos);  delay(15); }}  int pos = 0; |
| --- | --- | --- |

Hồng ngoại: LED phát (D3, GND), LED thu (D7, GND, VCC)

| #include <IRremote.h>  IRsend irsend;  void setup() { Serial.begin(9600);}  int val=0;  void loop() { val = val ^ 1; // đổi giá trị val  Serial.println("Sending: " + String(val) + "...");  irsend.sendRC5(val, 8);  delay(500);} | #include <IRremote.h> const int rpin = 7;  IRrecv irrecv(rpin); decode\_results rs;  void setup() { irrecv.enableIRIn();  Serial.begin(9600); pinMode(13, OUTPUT);  }  void loop() { if (irrecv.decode(&rs)) { int value = rs.value; Serial.println(value);  digitalWrite(13, value); irrecv.resume(); }delay(1); } |
| --- | --- |

Bluetooth HC-05: VCC, GND, RX TX (11,10) EN vào cmdmode

| #include <SoftwareSerial.h>  SoftwareSerial BTSerial(10, 11); // **slave**  void setup(){Serial.begin(9600);BT.begin(9600);  pinMode(13, OUTPUT);}  void loop() { if (BTSerial.available()) {  char input=BTSerial.read();  Serial.write(input);  if (input == '1') digitalWrite(13, HIGH);  if (input == '0') digitalWrite(13, LOW); }} | #include <SoftwareSerial.h>  SoftwareSerial BTSerial(10, 11);  void setup() { Serial.begin(9600);  BTSerial.begin(9600);  }void loop() { BTSerial.write('1');  delay(2000);  BTSerial.write('0');  delay(2000);}  // parseInt(), readString(), print\_ln |
| --- | --- |

GPRS VCC GND, TX 3 RX 2, RST reset, RING có call, MIC+- mic , SPK+- loa, DTR sleepmode

| // SEND MESSAGE  #include <SoftwareSerial.h>  SoftwareSerial mySerial(3, 2);// TX 3 RX 2  void setup()  { Serial.begin(9600);  mySerial.begin(9600);  Serial.println("Initializing...");  delay(1000);  mySerial.println("AT"); updateSerial();  mySerial.println("AT+CMGF=1"); updateSerial();  mySerial.println("AT+CMGS=\"+ZZxxxxxxxxxx\""); updateSerial();  mySerial.print("noi dung"); updateSerial();  mySerial.write(26);}  void loop(){}  void updateSerial()  { delay(500); while (Serial.available()) {  mySerial.write(Serial.read()); }  while(mySerial.available()) {  Serial.write(mySerial.read()); }} | // RECIEVE MESSAGE  #include <SoftwareSerial.h>  SoftwareSerial mySerial(3, 2);  void setup()  { Serial.begin(9600);  mySerial.begin(9600);  Serial.println("Initializing...");  delay(1000);  mySerial.println("AT"); updateSerial();  mySerial.println("AT+CMGF=1"); updateSerial();  mySerial.println("AT+CNMI=1,2,0,0,0"); updateSerial();}  void loop(){ updateSerial();}  void updateSerial(){  delay(500); while (Serial.available()) {  mySerial.write(Serial.read()); }  while(mySerial.available()) {  Serial.write(mySerial.read()); }} |
| --- | --- |
| // MAKE CALL  #include <SoftwareSerial.h>  SoftwareSerial mySerial(3, 2); void setup()  { Serial.begin(9600); mySerial.begin(9600);  Serial.println("Initializing..."); delay(1000);  mySerial.println("AT"); updateSerial();  mySerial.println("ATD+ +ZZxxxxxxxxxx;"); updateSerial();  delay(20000); // wait for 20 seconds...  mySerial.println("ATH"); //hang up  updateSerial();}  void loop(){}  void updateSerial()  { delay(500);  while (Serial.available()) {  mySerial.write(Serial.read()); }  while(mySerial.available())  { Serial.write(mySerial.read()); }} | // RECIEVE CALL  #include <SoftwareSerial.h>  SoftwareSerial mySerial(3, 2);  void setup()  { Serial.begin(9600);  mySerial.begin(9600);  Serial.println("Initializing...");  }  void loop(){ updateSerial();}  void updateSerial()  { delay(500);  while (Serial.available()) {  mySerial.write(Serial.read()); }  while(mySerial.available()) {  Serial.write(mySerial.read()); }} |

NODE MCU SYNC WEB

| #include <ESP8266WiFi.h>  #include <ESP8266WebServer.h>  #include "Servo.h"  // WiFi network  const char\* ssid = "Han";  const char\* password = "12345678";  ESP8266WebServer server ( 80 );  Servo myservo;  char htmlResponse[3000];  void handleAngle() {  if (server.arg("goc") != "") {  Serial.println("Goc: " + server.arg("goc"));  int goc = ((String)server.arg("goc")).toInt();  myservo.write(goc);  }}  void xmlRequest() {  snprintf ( htmlResponse, 3000,  "<!DOCTYPE html>\  <html lang=\"en\">\  <head>\  <meta charset=\"utf-8\">\  <meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">\  </head>\  <body>\  <h1>Dien toc do 0 - 180</h1>\  <input type='text' name='goc' id='goc' size=2 autofocus> hh \  <div>\  <br><button id=\"goc\_button\">OK</button>\  </div>\  <script>\ document.getElementById('goc\_button').onclick = function() {\  var xmlHttp = new XMLHttpRequest();\  var angle = document.getElementById('goc').value;\  xmlHttp.open( \"GET\", '/speed?s=' + angle, true ); \  xmlHttp.send( null );\  };\  </script>\  </body>\  </html>");  server.send ( 200, "text/html", htmlResponse );} | void setup() {  pinMode(LED\_BUILTIN, OUTPUT);  pinMode(In3, OUTPUT);  pinMode(In4, OUTPUT);  // Start serial  Serial.begin(115200);  delay(10);  myservo.attach(servo\_pin);  // Connecting to a WiFi network  Serial.println();  Serial.println();  Serial.print("Connecting to ");  Serial.println(ssid);  WiFi.begin(ssid, password);  while (WiFi.status() != WL\_CONNECTED) {  delay(500);  Serial.print(".");  }  Serial.println("");  Serial.println("WiFi connected");  Serial.println("IP address: ");  Serial.println(WiFi.localIP());  server.on("/angle", handleAngle);  server.on("/xml", xmlRequest);  server.begin();  Serial.println ( "HTTP server started" );  }  void loop() {  server.handleClient();  } |
| --- | --- |

NODEMCU ASYNC WEB

| // Import required libraries  #include <Arduino.h>  #include <ESP8266WiFi.h>  #include <Hash.h>  #include <ESPAsyncTCP.h>  #include <ESPAsyncWebServer.h>  #include <Adafruit\_Sensor.h>  #include <DHT.h>  const char\* ssid = "Han";  const char\* password = "12345678";  #define DHTPIN 5  #define DHTTYPE DHT11 // DHT 11  DHT dht(DHTPIN, DHTTYPE);  int t = 0; float h = 0.0;  const char index\_html[] PROGMEM = R"rawliteral(  <!DOCTYPE HTML><html>  <head>  <meta name="viewport" content="width=device-width, initial-scale=1">  </head>  <body>  <h2>ESP8266 DHT Server</h2>  <p>  <i class="fas fa-thermometer-half" style="color:#059e8a;"></i>  <span class="dht-labels">Co nguoi 1 </span>  <br>  <span class="dht-labels">K co nguoi 0 </span>  <br>  <span id="temperature">%TEMPERATURE%</span>  </body>  <script>  setInterval(function ( ) {  var xhttp = new XMLHttpRequest();  xhttp.onreadystatechange = function() {  if (this.readyState == 4 && this.status == 200) { | document.getElementById("temperature").innerHTML = this.responseText;  }  };  xhttp.open("GET", "/temperature", true);  xhttp.send();  }, 10000 ) ;  setInterval(function ( ) {  var xhttp = new XMLHttpRequest();  xhttp.onreadystatechange = function() {  if (this.readyState == 4 && this.status == 200) {  document.getElementById("humidity").innerHTML = this.responseText;  }  };  xhttp.open("GET", "/humidity", true);  xhttp.send();  }, 10000 ) ;  </script>  </html>)rawliteral";  // Replaces placeholder with DHT values  String processor(const String& var) { //Serial.println(var);  if (var == "TEMPERATURE") { return String(t);  } return String();  }  void setup() {  Serial.begin(115200);  dht.begin();  WiFi.begin(ssid, password);  Serial.println("Connecting to WiFi");  while (WiFi.status() != WL\_CONNECTED) {  delay(1000);  Serial.println("."); }  Serial.println(WiFi.localIP());  server.on("/", HTTP\_GET, [](AsyncWebServerRequest \* request) {  request->send\_P(200, "text/html", index\_html, processor);  });  server.begin();  }  void loop() { } |
| --- | --- |

Cảm biến hồng ngoại số HC-SR501

Cảm biến vật cản hồng ngoại FC-51 Obstacle sensor

Cb Hall KY-024, cảm biến cháy KY-026

Cảm biến rung SW-420