곽진 21900031

1. ESP32 Module

1) Wifi모듈을 통해 홈페이지에서 모터 제어할 수 있는 코드를 작성하고, 주석을 첨부하시오

답)

|  |  |
| --- | --- |
| **Arduino Code** | **코드 설명** |
| #include <Servo.h>  #include <ESP8266WiFi.h>  #define left 0  #define right 180  #define center 90  #define attch D0  const char\* ssid= "Mschool" ;  const char\* password = "Mschool123";  Servo myservo;  int now=0;  WiFiServer server(80);  int stop=82;  void setup() {    Serial.begin(9600);    delay(10);    WiFi.begin(ssid, password);    while (WiFi.status() != WL\_CONNECTED) { //Wifi 접속까지        delay(500);        Serial.print(".");    }    Serial.println("");    Serial.print("Connecting to ");    Serial.println(ssid);    Serial.print("IP address: ");    Serial.println(WiFi.localIP());    server.begin();    Serial.println("Server started");    myservo.attach(attch); //Pin Number    myservo.write(center);  //Set the motor angle to 0    delay(10);  }  void loop() {    WiFiClient client = server.available();    // 웹사이트에 접속했을 때    if(!client) return;      Serial.println("새로운 클라이언트");    client.setTimeout(5000);    // 클라이언트 전송 후 5초 초과 시 타임 아웃    String request = client.readStringUntil('\r');    // 전송받은 데이터 즉,URL을 알기 위해 사용    Serial.println("request: ");    Serial.println(request);    client.print("<title>Servo Control </title>");    client.print("</head>");    client.print("<body>");    client.print("<h2>Servo Control Webpage</h2>");//제목    client.println("<br>");    client.println("<a href='/Left'><button>left </button></a>");    client.println("<br>");    client.println("<a href='/Center'><button>center </button></a><br />");    client.println("<a href='/Right'><button>right </button></a>");    client.println("<br>");    client.print("HTTP/1.1 200 OK");    client.print("Content-Type: text/html\r\n\r\n");  // 해더& 구분    client.print("<!DOCTYPE HTML>");                  //HTML로 만들어진 문서 선언    client.print("<html>");    client.println("<br>");    client.print("Servo Status : ");    if(request.indexOf("/Left") != -1) { //왼쪽버튼      client.print("Left");      if(now == left) {        myservo.write(stop);      }//이미 왼쪽이라면 스탑      else if(now == right){        myservo.write(0);        delay(500);      }      else{ //Center일 때 및 예외        myservo.write(left);      }      delay(500);      myservo.write(stop);      now =left;    }    else if(request.indexOf("/Right") != -1) {  //오른 쪽      client.println("Right");      if(now == left){        myservo.write(right);        delay(500);      }      else if(now == center){        myservo.write(right);      }      else{        myservo.write(stop);      }      delay(500);      myservo.write(stop);      now = right;    }    else if(request.indexOf("\Center") != -1){ //center      client.println("Center");      if(now == center){        myservo.write(stop);        delay(500);      }      else if(now == left){        myservo.write(right);      }      else{        myservo.write(left);      }      delay(500);      myservo.write(stop);      now = center;    }    else{      Serial.println("invalid request");      myservo.write(stop);      delay(1000);      // 현재 상태 유지    }    while(client.available()) {      client.read(); // 버퍼 비움 효과    }    client.print("<head>"); //    client.print("<meta&nbsp;charset=\"UTF-8\">");    client.println("<br>");    client.println("<br>");    client.print("</body>");    client.print("</html>");    Serial.println("클라이언트 연결 해제");  } | Servo를 위해 사용한 Servo 헤더 파일과 와이파이 통신을 위한 EPS 헤더 파일  좌측으로 도는 것을 0  우측으로 도는 것을 180  중앙을 90으로 맞춰놓았다.  D0로 회로를 연결하였다.  와이파이는 엠스쿨 와이파이를 사용하였다.  0~180 사이에서 돌아야하기 때문에,  now라는 int 자료형을 사용해 매번 마지막에 눌린 값을 저장하여 90도 이상으로 도는 것을 방지하였다.  Stop은 경험적으로 모터를 출력해보았을 때 가장 덜 도는 숫자라서 멈췄을 때 servo.write문에 stop으로 출력하였다.  0~180에서만 모터가 돌아가기 위하여 특정 버튼이 눌렸을 때 그 안의 조건문으로 상황에 따른 모터 출력을 주었다. |