علیرضا سلطانی نشان ا 98111033302016

درس سخت افزار

استاد افشاری

کد درس 2328

مثال اول) روشن کردن LED با استفاده از اعلان استفاده شده در پروتکل UDP، بطوری که turnon و turnoff اگر صدا شود چراغ توسط برد آردواینو و جواستیک و خازن روشن یا خاموش می‌شود.

```
#include <ESP8266WiFi.h>
#include <WiFiUdp.h>
 const char* ssid = "sofa" ;const char* password = "AL20re/*";

WiFiUDP Udp;
unsigned int localUdpPort = 4210;  // local port to listen on
char incomingPacket[255];  // buffer for incoming packets
char  replyPacket[] = "Hi there! Got the message :-)";  // a reply string to send back


void setup()
{
  Serial.begin(115200);
  Serial.println();

  Serial.printf("Connecting to %s ", ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
  Serial.println(" connected");
  Udp.begin(localUdpPort);
  Serial.printf("Now listening at IP %s, UDP port %d\n", WiFi.localIP().toString().c_str(),
localUdpPort);
}


void loop()
{
  int packetSize = Udp.parsePacket();
  if (packetSize)
  {
    // receive incoming UDP packets
//    Serial.printf("Received %d bytes from %s, port %d\n", packetSize,
Udp.remoteIP().toString().c_str(), Udp.remotePort());
    int len = Udp.read(incomingPacket, 255);

    if (len > 0)
    {
      incomingPacket[len] = 0;
    }
    Serial.println(incomingPacket[5]);

    char turnOnChar [255] = "turnOn";
    char turnOffChar [255] = "turnOff";

    Serial.println(incomingPacket);

    if (incomingPacket[5] == turnOnChar[5]) {
      digitalWrite(LED_BUILTIN, LOW);
      Serial.println("THE =>>> Turn On LED");
    } else if (incomingPacket[5] == turnOffChar[5]) {
      digitalWrite(LED_BUILTIN, HIGH);
      Serial.println("THE =>>> Turn off LED");
    }
//    Serial.printf("UDP packet contents: %s\n", incomingPacket);

    // send back a reply, to the IP address and port we got the packet from
    Udp.beginPacket(Udp.remoteIP(), Udp.remotePort());
    Udp.write(replyPacket);

    Udp.endPacket();
  }
}
```

مثال دوم) ارسال اطلاعات جیسون به سمت برد که اطلاعات را به آن قالب توسط پروتکل HTTP دریافت می‌کند.

```
/*
 *  ESP8266 JSON Decode of server response
 *  - Alireza Soltani Neshan | 98111033302016
 *  - project 5
 *  - IoT
 */

#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266HTTPClient.h>
#include <ArduinoJson.h>

const char* wifiName = "iPhone";
const char* wifiPass = "AL20re/*";

//Web Server address to read/write from
const char *host = "http://172.20.10.9:4001/5";

void setup() {

  Serial.begin(115200);
  delay(10);
  Serial.println();

  Serial.print("Connecting to ");
  Serial.println(wifiName);

  WiFi.begin(wifiName, wifiPass);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());   //You can get IP address assigned to ESP
}

void loop() {
  HTTPClient http;    //Declare object of class HTTPClient

  Serial.print("Request Link:");
  Serial.println(host);

  http.begin(host);     //Specify request destination

  int httpCode = http.GET();            //Send the request
  String payload = http.getString();    //Get the response payload from server

  Serial.print("Response Code:"); //200 is OK
  Serial.println(httpCode);   //Print HTTP return code

  Serial.print("Returned data from Server:");
  Serial.println(payload);    //Print request response payload

  if(httpCode == 200)
  {
    // Allocate JsonBuffer
    // Use arduinojson.org/assistant to compute the capacity.
    const size_t capacity = JSON_OBJECT_SIZE(10) + JSON_ARRAY_SIZE(8) + 60;
    DynamicJsonBuffer jsonBuffer(capacity);

   // Parse JSON object
    JsonObject& root = jsonBuffer.parseObject(payload);
    if (!root.success()) {
      Serial.println(F("Parsing failed!"));
      return;
    }

    // Decode JSON/Extract values
    Serial.println("===============Your Res as Info===============");
    Serial.println(F("Response:"));
    Serial.println("Name of Country:");
    Serial.println(root["name"].as<char*>());
    Serial.println(root["country"].as<char*>());
    Serial.println("Description:");
    Serial.println(root["description"].as<char*>());
    Serial.println("Latitude:");
    Serial.println(root["latitude"].as<char*>());
    Serial.println("Longitude:");
    Serial.println(root["longitude"].as<char*>());
  }
  else
  {
    Serial.println("Error in response");
  }

  http.end();  //Close connection

  delay(5000);  //GET Data at every 5 seconds
}
```

مثال سوم) روشن و خاموش کردن چراغ توسط برد آردواینو توسط GPIO

```
/*
 *  This sketch demonstrates how to set up a simple HTTP-like server.
 *  The server will set a GPIO pin depending on the request
 *    http://server_ip/gpio/0 will set the GPIO2 low,
 *    http://server_ip/gpio/1 will set the GPIO2 high
 *  server_ip is the IP address of the ESP8266 module, will be
 *  printed to Serial when the module is connected.
 */

#include <ESP8266WiFi.h>

  const char *ssid = "sofa";  //ENTER YOUR WIFI SETTINGS
  const char *password = "AL20re/*";

// Create an instance of the server
// specify the port to listen on as an argument
WiFiServer server(80);

void setup() {
  Serial.begin(9600);
  delay(10);

  // prepare GPIO2
  pinMode(D4, OUTPUT);
  digitalWrite(2, 0);

  // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
   //------------------------------------
 Serial.print("ESP Board MAC Address:  ");
  Serial.println(WiFi.macAddress());
  //------------------------------------
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  // Start the server
  server.begin();
  Serial.println("Server started");

  // Print the IP address
  Serial.println(WiFi.localIP());
//  server.on("/", handleRoot);
}

void loop() {
 bool x=0;
  // Check if a client has connected
  WiFiClient client = server.available();
  if (!client) {
    return;
  }

  // Wait until the client sends some data
  Serial.println("new client");

  while(!client.available()){
    delay(1);
  }

  // Read the first line of the request
  String req = client.readStringUntil('\r');
    Serial.println("req=---------------------");
  Serial.println(req);
  client.flush();
  //http://192.168.1.100/gpio/1
  // Match the request
  if (req.indexOf("/gpio/0") != -1)
  {
    digitalWrite(2, HIGH);//off
    x=1;
  }
  else if (req.indexOf("/gpio/1") != -1)
  {
    digitalWrite(2, LOW);//on
    x=0;
  }
  else {
    Serial.println("invalid request");
    client.stop();
    return;
  }

  // Set GPIO2 according to the request


  client.flush();

    String convertableLocalIp = WiFi.localIP().toString().c_str();
    String url0 = "/gpio/0";
      String url1 = "/gpio/1";
  //    String toggleUrl = x== 1 ? '/gpio/1' : x == 0? '/gpio/0': '';

  Serial.println(x);


    // Prepare the response
    String s = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n<!DOCTYPE HTML>\r\n<html> <button onclick='toggle()'>Toggle</button> <button onclick='on0()'>gpio0</button> <button onclick='on1()'>gpio1</button>  \r\nGPIO is now <script> function on0(){location.replace(\""+url0+"\")} function on1(){location.replace(\""+url1+"\")} function toggle(){location.replace(\""+x+"\"== 1 ? '/gpio/1' : \""+x+"\" == 0? '/gpio/0': '')} </script> ";
    s += (x)?"high":"low";
    s += "</html>\n";

    // Send the response to the client
    client.print(s);
      Serial.println(s);
    delay(1);
    Serial.println("Client disonnected");

    // The client will actually be disconnected
    // when the function returns and 'client' object is detroyed
  }

```