# **Υποδράση 3**

## **Προγραμματισμός ESP32s Serial Communication with Arduino**

#include <AsyncTCP.h>

// this sample code provided by www.programmingboss.com

#define RXp2 16

#define TXp2 17

//define for GAS Sensor

#define LED 2

#define SENSOR 4

#define DELAY 500

int sensorValue = 0; // variable to store the value coming from the sensor

#include "WiFi.h"

#include "ESPAsyncWebServer.h"

#define RELAY\_NO true

// Set the number of relays you want to use

#define NUM\_RELAYS 4

// Assign each GPIO to a relay

int relayGPIOs[NUM\_RELAYS] = {5, 18, 19, 21};

//Edit according to your need

String relayNames[NUM\_RELAYS] = { "Room Lights", "Magnetic Door", "Gas Sensor", "Alarm System" };

// Enter your wifi SSID and password

const char\* ssid = "SSID\_NETWORK"; //Connection to Wifi Network

const char\* password = "password!"; //Give password for Wifi Network

// Set your Static IP address

IPAddress local\_IP(192, 168, 0, 119);

// Set your Gateway IP address

IPAddress gateway(192, 168, 0, 1);

IPAddress subnet(255, 255, 255, 0);

IPAddress primaryDNS(8, 8, 8, 8); //optional

IPAddress secondaryDNS(8, 8, 4, 4); //optional

const char\* PARAM\_INPUT\_1 = "relay";

const char\* PARAM\_INPUT\_2 = "state";

AsyncWebServer server(80);

const char index\_html[] PROGMEM = R"rawliteral(

<!DOCTYPE HTML><html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

html {font-family: Arial; display: inline-block; text-align: center;}

h3 {

margin:0;

padding:0;

font-family: sans-serif;

text-align:center;

color:#fff;

font-size:16px;

padding:15px 0;

text-transform: uppercase;

letter-spacing:4px;

}

h4 {

margin:0;

padding:0;

font-family: sans-serif;

text-align:center;

color:#fff;

font-size:13px;

padding:12px 0;

text-transform: uppercase;

letter-spacing:2px;

}

body {

margin:0;

padding:0;

background:#151515;

}

.center {

position:absolute;

left:50%;

top:50%;

transform:translate(-50%, -50%);

}

input[type="checkbox"] {

margin:10px;

position:relative;

width:120px;

height:40px;

-webkit-appearance: none;

background: linear-gradient(0deg, #333, #000);

outline: none;

border-radius: 20px;

box-shadow: 0 0 0 4px #353535, 0 0 0 5px #3e3e3e, inset 0 0 10px rgba(0,0,0,1);

}

input:checked[type="checkbox"]:nth-of-type(2) {

background: linear-gradient(0deg, #e67e22, #f39c12);

box-shadow: 0 0 0 4px #353535, 0 0 0 5px #3e3e3e, inset 0 0 10px rgba(0,0,0,1);

}

input:checked[type="checkbox"]:nth-of-type(1) {

background: linear-gradient(0deg, #70a1ff, #1e90ff);

box-shadow: 0 0 0 4px #353535, 0 0 0 5px #3e3e3e, inset 0 0 10px rgba(0,0,0,1);

}

input[type="checkbox"]:before {

content:'';

position:absolute;

top:0;

left:0;

width:80px;

height:40px;

background: linear-gradient(0deg, #000, #6b6b6b);

border-radius: 20px;

box-shadow: 0 0 0 1px #232323;

transform: scale(.98,.96);

transition:.5s;

}

input:checked[type="checkbox"]:before {

left:40px;

}

input[type="checkbox"]:after{

content:'';

position:absolute;

top:calc(50% - 2px);

left:70px;

width:4px;

height:4px;

background: linear-gradient(0deg, #6b6b6b, #000);

border-radius: 50%;

transition:.5s;

}

input:checked[type="checkbox"]:after {

left:110px;

}

</style>

</head>

<body>

<h3>ESP32 Web Server</h3>

%BUTTONHOLDER%

<script>function toggleCheckbox(element) {

var xhr = new XMLHttpRequest();

if(element.checked){ xhr.open("GET", "/update?relay="+element.id+"&state=1", true); }

else { xhr.open("GET", "/update?relay="+element.id+"&state=0", true); }

xhr.send();

}</script>

</body>

</html>

)rawliteral";

// Replaces placeholder with button section in your web page

String processor(const String& var){

//Serial.println(var);

if(var == "BUTTONHOLDER"){

String buttons ="";

for(int i=1; i<=NUM\_RELAYS; i++){

String relayStateValue = relayState(i);

buttons+= "<h4>" + relayNames[i-1] + " - GPIO " + relayGPIOs[i-1] + "</h4><label class=\"switch\"><input type=\"checkbox\" onchange=\"toggleCheckbox(this)\" id=\"" + String(i) + "\" "+ relayStateValue +"><span class=\"slider\"></span></label>";

}

return buttons;

}

return String();

}

String relayState(int numRelay){

if(RELAY\_NO){

if(digitalRead(relayGPIOs[numRelay-1])){

return "";

}

else {

return "checked";

}

}

else {

if(digitalRead(relayGPIOs[numRelay-1])){

return "checked";

}

else {

return "";

}

}

return "";

}

void setup() {

pinMode(LED, OUTPUT);

Serial.println("Sensor start");

// put your setup code here, to run once:

Serial.begin(115200);

Serial2.begin(9600, SERIAL\_8N1, RXp2, TXp2);

// Set all relays to off when the program starts - if set to Normally Open (NO), the relay is off when you set the relay to HIGH

for(int i=1; i<=NUM\_RELAYS; i++){

pinMode(relayGPIOs[i-1], OUTPUT);

if(RELAY\_NO){

digitalWrite(relayGPIOs[i-1], HIGH);

Serial.println(relayGPIOs[i-1]);

}

else{

digitalWrite(relayGPIOs[i-1], LOW);

}

}

// Configures static IP address

if (!WiFi.config(local\_IP, gateway, subnet, primaryDNS, secondaryDNS)) {

Serial.println("STA Failed to configure");

}

// Connect to Wi-Fi

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(1000);

Serial.println("Connecting to WiFi..");

}

// ESP32 IP Address

Serial.println(WiFi.localIP());

// Route for root / web page

server.on("/", HTTP\_GET, [](AsyncWebServerRequest \*request){

request->send\_P(200, "text/html", index\_html, processor);

});

// Send GET request to <ESP\_IP>/update?relay=<inputMessage>&state=<inputMessage2>

server.on("/update", HTTP\_GET, [] (AsyncWebServerRequest \*request) {

String inputMessage;

String inputParam;

String inputMessage2;

String inputParam2;

// GET input1 value on <ESP\_IP>/update?relay=<inputMessage>

if (request->hasParam(PARAM\_INPUT\_1) & request->hasParam(PARAM\_INPUT\_2)) {

inputMessage = request->getParam(PARAM\_INPUT\_1)->value();

inputParam = PARAM\_INPUT\_1;

inputMessage2 = request->getParam(PARAM\_INPUT\_2)->value();

inputParam2 = PARAM\_INPUT\_2;

if(RELAY\_NO){

Serial.print("NO ");

digitalWrite(relayGPIOs[inputMessage.toInt()-1], !inputMessage2.toInt());

}

else{

Serial.print("NC ");

digitalWrite(relayGPIOs[inputMessage.toInt()-1], inputMessage2.toInt());

}

}

else {

inputMessage = "No message sent";

inputParam = "none";

}

Serial.println(inputMessage + inputMessage2);

request->send(200, "text/plain", "OK");

});

// Start the server

server.begin();

}

void loop() {

Serial.println("Message Received: ");

Serial.println(Serial2.readString());

// start the Gas sensor measures

digitalWrite(LED, HIGH); // turn the LED on (HIGH is the voltage level)

delay(DELAY); // wait for a second

// read the value from the sensor:

sensorValue = analogRead(SENSOR);

Serial.print("Value: "); Serial.println(sensorValue);

digitalWrite(LED, LOW); // turn the LED off by making the voltage LOW

delay(DELAY);

}