

Nama : _____

Kelas : _____

Date : _____

Guru : _____

LEMBAR KERJA PESERTA DIDIK



ELEMEN PEMROGRAMAN DAN APLIKASI MIKROKONTROLER

KELAS 11 SEMESTER GENAP 2023 / 2024

SMK KONSENTRASI KEAHLIAN TEKNIK AUDIO VIDEO

Webserver With ESP 32

Tujuan Pembelajaran

Pada akhir Fase F, Peserta Didik Mampu memahami pemrograman mikrokontroler; membuat program aplikasi sistem pengendali berbasis mikrokontroler.

Alur Tujuan Pembelajaran

1. Peserta didik mampu merencanakan proyek mikrokontroler berbasis web server ESP32
2. Peserta didik mampu membuat proyek mikrokontroler berbasis web server ESP32
3. Peserta didik mampu membuat program proyek mikrokontroler berbasis web server ESP32
4. Peserta didik mampu mengimplementasikan proyek yang telah dibuatnya di kehidupan sehari-hari

A. MATERI

KENDALI LAMPU DENGAN ESP 32 DAN WEBSERVER



Control Multiple Relays with ESP32 Web Server



Di bagian ini, kami telah membuat contoh server web yang memungkinkan Anda untuk mengontrol relai sebanyak yang Anda inginkan melalui server web apakah mereka dikonfigurasi sebagai biasanya dibuka atau ditutup secara normal. Anda hanya perlu mengubah beberapa baris kode untuk menentukan jumlah relai yang ingin Anda kontrol dan penetapan pin.

Untuk membangun server web ini, kami menggunakan [ESPAsyncWebServer library](#)

Menginstal Library ESPAsyncWebServer

Ikuti langkah-langkah selanjutnya untuk menginstal [ESPAsyncWebServer](#) library:

1. [Klik di sini untuk mengunduh library ESPAsyncWebServer](#). Anda harus memiliki folder .zip di folder Unduhan Anda
2. Buka zip folder .zip dan Anda akan mendapatkan folder *master ESPAsyncWebServer*
3. Ganti nama folder Anda dari *ESPAsyncWebServer-master* ke *ESPAsyncWebServer*
4. Pindahkan folder *ESPAsyncWebServer* ke folder library instalasi Arduino

Atau, di Arduino IDE Anda, Anda bisa pergi ke **Sketch > Include Library > Add .ZIP library...** dan pilih library yang baru saja Anda unduh.

Menginstal Library Async TCP untuk ESP32

[Pustaka ESPAsyncWebServer](#) memerlukan library [AsyncTCP](#) agar berfungsi. Ikuti langkah-langkah selanjutnya untuk menginstal library itu:

1. [Klik di sini untuk mengunduh library AsyncTCP](#). Anda harus memiliki folder .zip di folder Unduhan Anda
2. Buka zip folder .zip dan Anda akan mendapatkan folder *master AsyncTCP*
3. Ganti nama folder Anda dari *AsyncTCP-master* ke *AsyncTCP*
4. Pindahkan folder *AsyncTCP* ke folder libraries instalasi Arduino IDE Anda
5. Terakhir, buka kembali IDE Arduino Anda

Setelah menginstal Library yang diperlukan, salin kode berikut ke Arduino IDE Anda.

```
//MONSTERCHIP INDONESIA
// Import required libraries
#include "WiFi.h"
#include "ESPAsyncWebServer.h"

// Set to true to define Relay as Normally Open (NO)
#define RELAY_NO true

// Set number of relays
#define NUM_RELAYS 5

// Assign each GPIO to a relay
int relayGPIOs[NUM_RELAYS] = {2, 26, 27, 25, 33};

// Replace with your network credentials
const char* ssid = "REPLACE_WITH_YOUR_SSID";
const char* password = "REPLACE_WITH_YOUR_PASSWORD";

const char* PARAM_INPUT_1 = "relay";
const char* PARAM_INPUT_2 = "state";

// Create AsyncWebServer object on port 80
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;}
    h2 {font-size: 3.0rem;}
    p {font-size: 3.0rem;}
    body {max-width: 600px; margin:0px auto; padding-bottom: 25px;}
    .switch {position: relative; display: inline-block; width: 120px; height: 68px}
    .switch input {display: none}
    .slider {position: absolute; top: 0; left: 0; right: 0; bottom: 0; background-color:
#ccc; border-radius: 34px}
    .slider:before {position: absolute; content: ""; height: 52px; width: 52px; left: 8px;
bottom: 8px; background-color: #fff; -webkit-transition: .4s; transition: .4s; border-
radius: 68px}
    input:checked+.slider {background-color: #2196F3}
    input:checked+.slider:before {-webkit-transform: translateX(52px); -ms-transform:
translateX(52px); transform: translateX(52px)}
  </style>
</head>
<body>
  <h2>ESP Web Server</h2>
  %BUTTONPLACEHOLDER%
<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 == "BUTTONPLACEHOLDER"){
    String buttons = "";
```

```

    for(int i=1; i<=NUM_RELAYS; i++){
        String relayStateValue = relayState(i);
        buttons+= "<h4>Relay #" + String(i) + " - 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(){
    // Serial port for debugging purposes
    Serial.begin(115200);

    // 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);
        }
        else{
            digitalWrite(relayGPIOs[i-1], LOW);
        }
    }
    // Connect to Wi-Fi
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(1000);
        Serial.println("Connecting to WiFi..");
    }

    // Print ESP32 Local 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 a 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 server
server.begin();
}

void loop() {
}
```

Tentukan Konfigurasi Relay

Ubah variabel berikut untuk menunjukkan apakah Anda menggunakan relai dalam konfigurasi biasanya terbuka (NO) atau biasanya tertutup (NC). Mengatur RELAY_NO variabel ke **true** untuk os yang biasanya terbuka diatur ke **false** untuk biasanya tertutup.

```
#define RELAY_NO true
```

Tentukan Jumlah Relai (Saluran)

Anda dapat menentukan jumlah relai yang ingin Anda kendalikan pada NUM_RELAYS variabel. Untuk tujuan demonstrasi, kami menyetelnya ke 5.

```
#define NUM_RELAYS 5
```

Tentukan Tugas Pin Relay

Dalam variabel array berikut, Anda dapat menentukan GPIO ESP32 yang akan mengontrol relay:

```
int relayGPIOs[NUM_RELAYS] = {2, 26, 27, 25, 33};
```

Jumlah relai yang disetel pada NUM_RELAYS variabel harus sesuai dengan jumlah GPIO yang ditetapkan dalam relayGPIOHimpunan.

Kredensial Jaringan

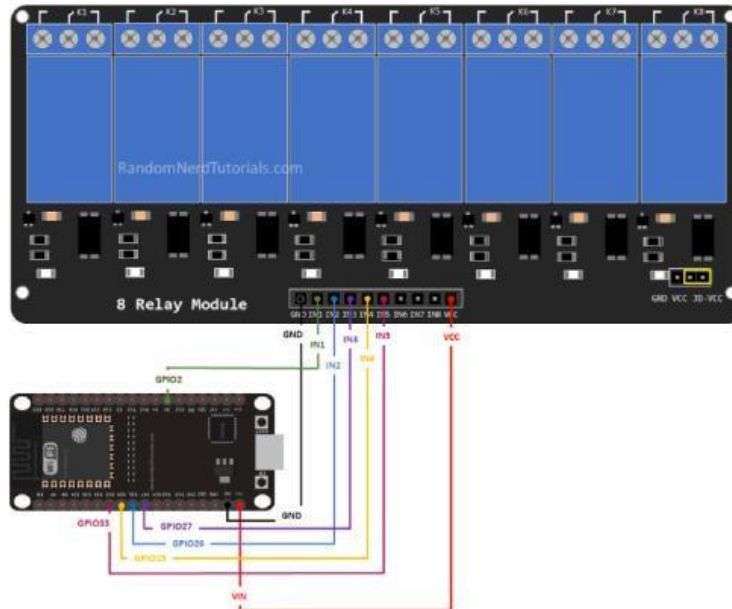
Masukkan kredensial jaringan Anda dalam variabel berikut.

```
const char* ssid = "REPLACE_WITH_YOUR_SSID";
```

```
const char* password = "REPLACE_WITH_YOUR_PASSWORD";
```

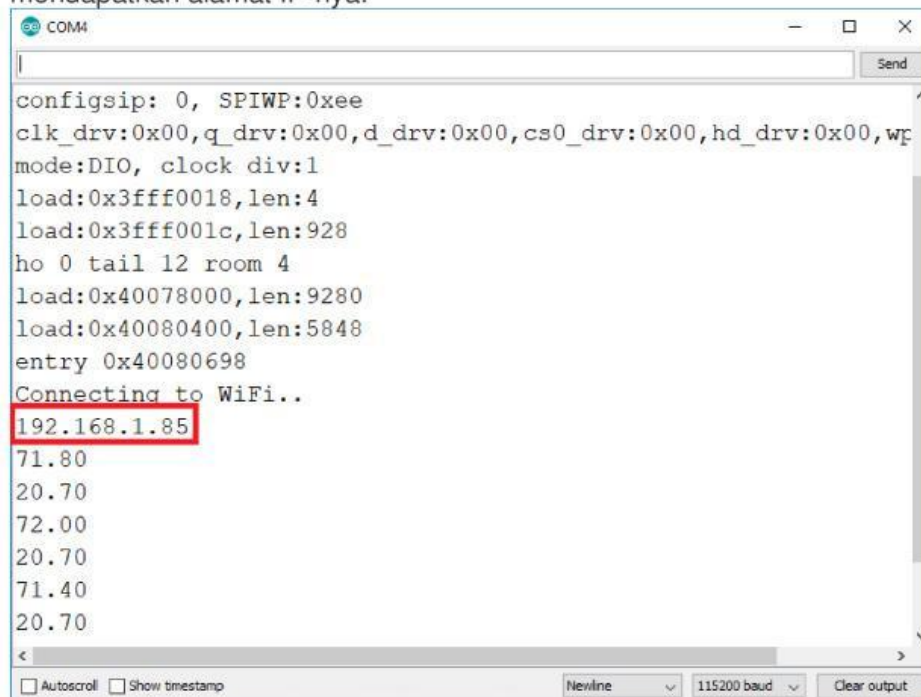
Rangkaian Relay 8 Channel ke ESP32

Untuk tujuan demonstrasi, kami mengendalikan 5 saluran relai. Hubungkan ESP32 ke modul relai seperti yang ditunjukkan pada diagram skema berikut.



Demonstrasi

Setelah melakukan perubahan yang diperlukan, unggah kode ke ESP32 Anda. Buka Serial Monitor pada baud rate 115200 dan tekan tombol ESP32 EN untuk mendapatkan alamat IP-nya.



Kemudian, buka browser di jaringan lokal Anda dan ketik alamat IP ESP32 untuk mendapatkan akses ke server web.

B. TANGGAPAN

BERIKAN TANGGAPAN KAMU PADA KOLOM DI BAWAH INI SETELAH MENYAKSIKAN PENJELASAN DALAM VIDEO DAN MEMPELAJARI ARTIKEL TADI BERKAITAN DENGAN MATERI

C. TENTUKAN JAWABAN YANG TEPAT

1. Pada video tutorial ESP32 dengan webserver, terlihat bentuk fisik ESP 32 yang digunakan. berikut ini Kode yang tepat sesuai dengan ESP 32 yang digunakan di video adalah

2. Pada Video tersebut terlihat untuk ESP 32 berbeda dengan arduino uno, terdapat 2 tombol, salah satunya adalah EN untuk mereset ESP32. adapun tombol satunya lagi yang tepat adalah....

3. Pada artikel diatas, Ketika akan membuat program webserver, Kita membutuhkan 2 library yang harus terinstall, library yang tepat adalah

4. didalam struktur program webserver di arduino IDE, selain menggunakan bahasa pemrograman C++, juga menggunakan sketch program dari

5. Setelah Kamu coba dan berhasil, silahkan masukkan IP Address project Kamu di Kolom sebelah Kanan ini !

D. REFLEKSI HASIL KEGIATAN

Diakhir e-LKPD ini Kamu diminta untuk merefleksikan seluruh Kegiatan yang telah dilakukan. adapun refleksinya menggunakan metode Traffic Light Reflection, dengan cara kamu memasukkan dikolom Komentar sesuai dengan instruksi yang telah disediakan dibawah ini.

Selebrasi

Apa yang harus dirayakan ?



Stop

Apa yang harus dihentikan ?



Lanjut

Apa yang harus dilanjutkan ?



Mulai

Apa yang harus dimulai ?

