

CÓDIGO DE PROGRAMAÇÃO UTILIZADO - IDE DO ARDUÍNO

Código feito na IDE do Arduíno com comandos para controlar a iluminação, a engrenagem (portão eletrônico), os sensores e as chaves de fim de curso.

<pre>#include <WiFi.h> // Definindo as portas que estão conectadas no ESP32 #define led1 5 // Led 1 esta na porta 5 #define led2 13 // Led 2 esta na porta 13 #define led3 12 // Led 3 esta na porta 12 #define ldr 36 // Ldr (sensor de luminosidade) esta na porta 36 #define led4 14 // Led 4 esta na porta 14 #define led5 27 // Led 5 esta na porta 27 #define led6 26 // Led 6 esta na porta 26 #define pinled 26 // Led 6 esta na porta 26 (ldr) #define abrir 25 // Girar motor para direita na porta 25 (abrir portão) #define chavel 39 // Parar motor na porta 39 (chave de fim de curso) #define fechar 33 // Girar motor para esquerda na porta 33 (fechar portão) #define chave2 35 // Parar motor na porta 35 (chave de fim de curso) const char* ssid = "Blessed"; //Nome da rede const char* password = "belchior153759"; //Senha int sensorValue; //Sensor de luminosidade int fimdecurso; //Chave de fim de curso de um lado do portao int fimdecurso2; //Chave de fim de curso do outro lado do portao WiFiServer server(80); void setup() { client.print("<!DOCTYPE html>"); client.print("<html lang='pt-br'>"); client.print("<head>"); client.print("<title>Automação Gabi</title>"); //Nome que irá aparecer quando executar ip client.print("<meta charset='utf-8'>"); client.print("</head>"); client.print("<body>"); client.print("ON Luz 1.
"); //Definir Comando H para acender LED1 client.print("OF Luz 1.
"); //Definir Comando L para apagar LED1 client.println(); client.print("ON Luz 2.
"); //Definir Comando A para acender LED2 client.print("OF Luz 2.
"); //Definir Comando B para apagar LED2 client.println(); client.print("ON Luz 3.
"); //Definir Comando X para acender LED3 client.print("OF Luz 3.
"); //Definir Comando Y para apagar LED3 client.println(); client.print("ON Luz 4.
"); //Definir Comando K para acender LED4 client.print("OF Luz 4.
"); //Definir Comando J para apagar LED4 client.println(); client.print("ON Luz 5.
"); //Definir Comando O para acender LED5 client.print("OF Luz 5.
"); //Definir Comando P para apagar LED5 client.println(); client.print("ON Luz 6.
"); //Definir Comando Q para acender LED6 client.print("OF Luz 6.
"); //Definir Comando W para apagar LED6 client.println(); Serial.begin(9600); pinMode(led1, OUTPUT); pinMode(led2, OUTPUT); pinMode(led3, OUTPUT); pinMode(led4, OUTPUT); pinMode(led5, OUTPUT); pinMode(led6, OUTPUT); pinMode(ldr, INPUT); pinMode(pinled, OUTPUT); pinMode(abrir, OUTPUT); pinMode(fechar, OUTPUT); pinMode(chavel, INPUT); delay(10); // Começa a conectar a rede WiFi Serial.println(); Serial.println(); Serial.print("Connecting to "); Serial.println(ssid); WiFi.begin(ssid, password); while (WiFi.status() != WL_CONNECTED) { delay(500); Serial.print("."); }</pre>	<pre>client.println(); client.print("ON Todas as luzes.
"); //Definir Comando T para acender TODOS OS LEDS client.print("OF Todas as luzes.
"); //Definir Comando R para apagar TODOS OS LEDS client.println(); client.print("ABRIR Portão.
"); //Definir Comando M para abrir portão client.println(); client.print("FECHAR Portão.
"); //Definir Comando N para fechar portão client.print("</body>"); client.print("</html>"); client.println(); break; } else { currentLine = ""; } } else if (c != '\r') { currentLine += c; } // Verifica se a solicitação do cliente foi "GET /H" or "GET /L": GET/H liga e o GET/L apaga if (currentLine.endsWith("GET /H")) { digitalWrite(led1, HIGH); } if (currentLine.endsWith("GET /L")) { digitalWrite(led1, LOW); } // Verifica se a solicitação do cliente foi "GET /A" or "GET /B": GET/A liga e o GET/B apaga if (currentLine.endsWith("GET /A")){ digitalWrite(led2, HIGH); } } if (currentLine.endsWith("GET /B")){ digitalWrite(led2, LOW); } // Verifica se a solicitação do cliente foi "GET /X" or "GET /Y": GET/X liga e o GET/Y apaga if (currentLine.endsWith("GET /X")){ digitalWrite(led3, HIGH); } } if (currentLine.endsWith("GET /Y")){ digitalWrite(led3, LOW); } // Verifica se a solicitação do cliente foi "GET /K" or "GET /J": GET/K liga e o GET/J apaga if (currentLine.endsWith("GET /K")){ digitalWrite(led4, HIGH); } } if (currentLine.endsWith("GET /J")){ digitalWrite(led4, LOW); } // Verifica se a solicitação do cliente foi "GET /O" or "GET /P": GET/O liga e o GET/P apaga if (currentLine.endsWith("GET /O")){ digitalWrite(led5, HIGH); } } if (currentLine.endsWith("GET /P")){ digitalWrite(led5, LOW); } } // Verifica se a solicitação do cliente foi "GET /Q" or "GET /W": GET/Q liga e o GET/W apaga if (currentLine.endsWith("GET /Q")){ digitalWrite(led6, HIGH); } } if (currentLine.endsWith("GET /W")){ digitalWrite(led6, LOW); } }</pre>	<pre>// Verifica se a solicitação do cliente foi "GET /M": GET/M abre o portão if (currentLine.endsWith("GET /M")){ digitalWrite(abrir, HIGH); } // Verifica se a solicitação do cliente foi "GET /N": GET/N fecha o portão if (currentLine.endsWith("GET /N")){ digitalWrite(fechar, HIGH); } } // Verifica se a solicitação do cliente foi "GET /T" or "GET /R": GET/T liga todas e o GET/R apaga todas if(currentLine.endsWith("GET /T")){ digitalWrite(led1, HIGH); digitalWrite(led2, HIGH); // LIGAR TODAS AS LAMPADAS digitalWrite(led3, HIGH); digitalWrite(led4, HIGH); digitalWrite(led5, HIGH); digitalWrite(led6, HIGH); } } if(currentLine.endsWith("GET /R")){ digitalWrite(led1, LOW); digitalWrite(led2, LOW); // DESLIGAR TODAS AS LAMPADAS digitalWrite(led3, LOW); digitalWrite(led4, LOW); digitalWrite(led5, LOW); digitalWrite(led6, LOW); } } } client.stop(); Serial.println("Client Disconnected."); } sensorValue = analogRead(36); // Ler o pino de entrada analógica 36 Serial.print(sensorValue, DEC); // Imprimir o valor lido Serial.print(" \n"); // Imprimi um espaço entre os números delay(500); // Aguarde 50ms para a próxima leitura if(sensorValue <= 2600) // Se o valor do sensor for menor ou igual a 2600 { digitalWrite(pinled, HIGH); } else { digitalWrite(pinled, LOW); } fimdecurso = analogRead(39); // Ler o pino de entrada 34 Serial.print(fimdecurso, DEC); // Imprimir o valor lido Serial.print(" \n"); // Imprimi um espaço entre os números delay(500); // Aguarde 50ms para a próxima leitura if(fimdecurso != 0) // Se o valor do sensor for igual a 0 { digitalWrite(abrir, LOW); Serial.print("desliga motor abrir"); } fimdecurso2 = analogRead(35); // Ler o pino de entrada 34 Serial.print(fimdecurso2, DEC); // Imprimir o valor lido Serial.print(" \n"); // Imprimi um espaço entre os números delay(500); // Aguarde 50ms para a próxima leitura if(fimdecurso2 != 0) // Se o valor do sensor for igual a 0 { digitalWrite(fechar, LOW); Serial.print("desliga motor fechar"); } }</pre>
---	---	---