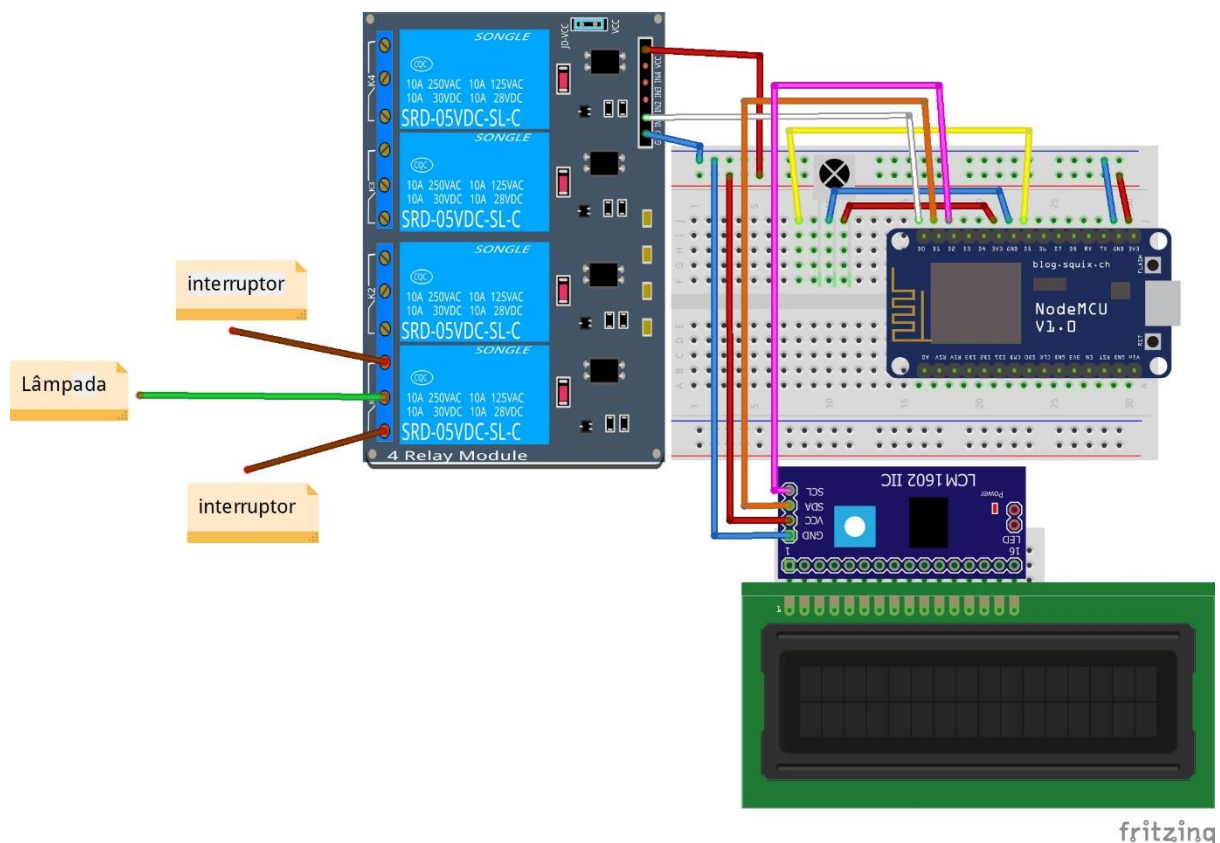


Projeto controle de lâmpada via web, controle remoto e interruptor

Pedro Felipe Gonçalves de Arruda

O projeto visa o controle (acionar/desligar) de uma lâmpada (podendo ser outras cargas) por meio de interface web, controle remoto ou o interruptor. Assim trazendo maior comodidade e flexibilidade ao usuário que for utilizar.

Para o funcionamento do projeto, haverá a necessidade dos seguintes componentes, um esp8266 NodeMCU, um display LCD 16x2 com modulo I2C, um sensor infravermelho e um interruptor paralelo.



A programação é feita pela IDE do Arduino, o programa faz com que o esp8266NodeMCU se conecte ao WIFI e assim crie uma página web, para que se possa interagir com ele usando a internet por meio da rede local, a programação do sensor infravermelho é feita de modo para quando identificar um botão específico do controle remoto interaja com a lâmpada e a programação do display serve para mostrar o ip do esp8266NodeMCU para que o usuário possa conectar nele e assim acessar a pagina web por ele gerado.

Código:

```
#include <IRremoteESP8266.h>

#include <IRremoteInt.h>

#include <ESP8266WiFi.h>

#include <Wire.h> // responsável pela comunicação com a interface i2c

#include <LiquidCrystal_I2C.h> // responsável pela comunicação com o display LCD


// Inicializa o display no endereço 0x27

//os demais parâmetros, são necessários para o módulo conversar com o LCD

//porém podemos utilizar os pinos normalmente sem interferência

//parâmetro: POSITIVE > > Backligh LIGADO | NEGATIVE > > Backlight desligado


LiquidCrystal_I2C lcd(0x27,2,1,0,4,5,6,7,3, POSITIVE);


#define ledPin D0


const char* ssid = "Servidor"; // Nome da Rede
const char* password = "Custom@00"; //Password da rede
int value = LOW;
WiFiServer server(80); // Porto 80


int RECV_PIN = D5;


IRrecv irrecv(RECV_PIN);


decode_results results;


void setup()
{
    //inicializa o display (16 colunas x 2 linhas)
    lcd.begin (16,2);
```

```
lcd.setCursor(0,0);  
lcd.print("Pedro Arruda");  
delay(1000);  
Serial.begin(115200); //  
Serial.println("Enabling IRin");//  
irrecv.enableIRIn(); // Inicia o IR receiver  
Serial.println("Enabled IRin");  
delay(400);  
pinMode(ledPin, OUTPUT); // Define o ledPin como saída  
digitalWrite(ledPin, LOW); // O LED começa desligado  
lcd.clear();  
  
// Comunicação com a rede WiFi  
Serial.println();  
Serial.println();  
Serial.print("Conectado a "); // Mensagem apresentada no monitor série  
Serial.println(ssid); // Apresenta o nome da rede no monitor série  
  
lcd.setCursor(0,0);  
lcd.print("Conectendo a ");  
  
lcd.setCursor(0,1);  
lcd.print(ssid);  
delay(1000);  
  
WiFi.begin(ssid, password); // Inicia a ligação a rede  
  
while (WiFi.status() != WL_CONNECTED) {  
  
    delay(500);
```

Serial.print("."); // Enquanto a ligação não for efectuada com sucesso é apresentado no monitor série uma sucessão de “.”

lcd.clear();

}

lcd.clear();

Serial.println("");

Serial.println("WiFi connected"); // Se a ligação é efectuada com sucesso apresenta esta mensagem no monitor série

lcd.setCursor(0,0);

lcd.print("Conectado WIFI");

delay(1000);

lcd.clear();

// Servidor

server.begin(); // Comunicação com o servidor

Serial.println("Servidor iniciado"); //é apresentado no monitor série que o servidor foi iniciado

lcd.setCursor(0,0);

lcd.print("Servidor");

// Impressão do endereço IP

Serial.print("Use o seguinte URL para a comunicação: ");

Serial.print("http://");

Serial.print(WiFi.localIP()); //Abrindo o Brower com este IP acedemos á pagina HTML de controlo dos LED's, sendo que este IP só está disponível na rede à qual o ESP8266 se encontra ligado

Serial.println("/");

lcd.setCursor(0,1);

lcd.print(WiFi.localIP());

```
}
```

```
void loop() {
```

```
    int estado = 0;
```

```
    if (irrecv.decode(&results)) {
```

```
        Serial.println(results.value, DEC);
```

```
        irrecv.resume(); // Receive the next value
```

```
        Serial.println(value);
```

```
        if (results.value == 3380809001) {
```

```
            if (value == LOW) {
```

```
                digitalWrite(ledPin, HIGH); // Se o pedido no LedPin for LED=ON, acende o LED
```

```
                value = HIGH;
```

```
            } else {
```

```
                digitalWrite(ledPin, LOW); // Se o pedido no LedPin for LED=OFF, apaga o LED
```

```
                value = LOW;
```

```
            }
```

```
        }
```

```
    }
```

```
    delay(500);
```

```
    // Verificação se o cliente está conectado
```

```
    WiFiClient client = server.available();
```

```
    if (!client) { // Verifica se o cliente está conectado ao servidor, executa este ciclo até estar conectado
```

```
        return;
```

```
    }
```

```
    // Espera até o cliente enviar dados
```

```
Serial.println("novo cliente"); //Apresenta esta mensagem quando o cliente se liga ao servidor
```

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

```
// Ler a primeira linha do pedido  
String request = client.readStringUntil('\r');  
Serial.println(request);  
client.flush();
```

```
// Operação do pedido
```

```
if (request.indexOf("/LED") != -1) {  
    if (value == LOW) {  
        digitalWrite(ledPin, HIGH); // Se o pedido no LedPin for LED=ON, liga o relé  
        value = HIGH;  
    } else {  
        digitalWrite(ledPin, LOW); // Se o pedido no LedPin for LED=OFF, desliga o relé  
        value = LOW;  
    }  
}
```

```
// Inicialização da página HTML
```

```
// Retorno do resposta  
client.println("HTTP/1.1 200 OK");  
client.println("Content-Type: text/html");  
client.println(""); // do not forget this one  
client.println("<!DOCTYPE HTML>");  
client.println("<html>");
```

```
client.print("LAMPADA");
```

```
client.println("<br><br>");
```

```
client.println("<a href=\" /LED\\\"><button style='height: 60px; width: 120px; font-size:15px'>ON/OFF</button></a>");// Ligar ou delisga a lampada
```

```
client.println("</html>");
```

```
delay(1);
```

```
Serial.println("Cliente desconectado"); // Depois do cliente efectuar o pedido apresenta esta mensagem no monitor série
```

```
Serial.println("");
```

```
}
```