



**Arduino e módulo ESP8266**  
**Sistemas de Comunicação – Noturno, Msc. Ivandro Ribeiro**

Equipe: Ramon Temporim de Lacerda Segundo RA: 21502162

Rodrigo da Silva Moraes RA: 21505004

Universidade Centro de Ensino Unificado de Brasília Campus Taguatinga II,  
Brasília, novembro de 2017.

## Resumo

A transmissão da rede WI-FI é feita por sinais de radiofrequência, que se propagam pelo ar e podem variar o alcance do sinal. O módulo WI-FI ESP8266 ESP-01 foi especialmente desenvolvido para que possamos conectar o nosso microcontrolador a uma conexão Wi-Fi de forma fácil e eficaz. Dessa forma, utilizamos no experimento para simular uma transferência de dados entre o computador e o Arduino com auxílio do módulo e a propagação da rede Wi-Fi.

*Palavras-Chaves: Wi-Fi, ESP8266 ESP-01, Arduino.*

## Introdução

As redes Wi-Fi fazem uso de ondas de rádio comuns para transmitir as informações de Internet, assim como acontece com a televisão, rádio e celular, por exemplo. Essas redes funcionam através de ondas de rádios transmitidas por meio de um adaptador, o roteador, que recebe os sinais, decodifica e os emite a partir de uma antena, sendo a parte principal do Wi-Fi. Os sinais de Internet emitidos pelos roteadores podem chegar via cabo, linha telefônica ou ondas de rádio, como é o caso do “3G”/ “4G” até os aparelhos que suportam essa tecnologia. Para que um dispositivo tenha acesso a esses sinais, é preciso que esteja dentro do “hotspot”, ou seja, dentro de um determinado raio de ação. A troca de informações acontece em uma das duas frequências disponíveis pelos governos, a de 2.4 GHz ou a de 5GHz. Quanto mais alta a frequência, maior também a capacidade do sinal carregar um alto número de informações. Sabendo que a rede não se resume apenas a internet.

O módulo Wi-Fi ESP8266 ESP-01 foi especialmente desenvolvido para que você possa conectar o seu microcontrolador a uma conexão WiFi de forma fácil e eficaz. O módulo WiFi ESP8266 suporta as redes 802.11 b/g/n, muito usadas atualmente, podendo trabalhar como um Ponto de Acesso (Access Point) ou como uma Estação (Station), enviando e recebendo dados. A comunicação do módulo com o Arduino pode ser feita via serial utilizando os pinos RX e TX, podendo ser configurada através de comandos AT. Porém, o Adaptador WiFi ESP-01 já possui o conversor de nível e pode ser conectado diretamente a protoboard. Dessa forma, é possível enviar dados de um componente para o outro através da rede Wi-Fi como detalhado abaixo.

## **Objetivos**

- Utilizar o arduíno junto a uma rede Wi-Fi;
- Entender o funcionamento do módulo ESP8266 – ESP01;
- Controlar um Led através da utilização de uma rede Wi-Fi.

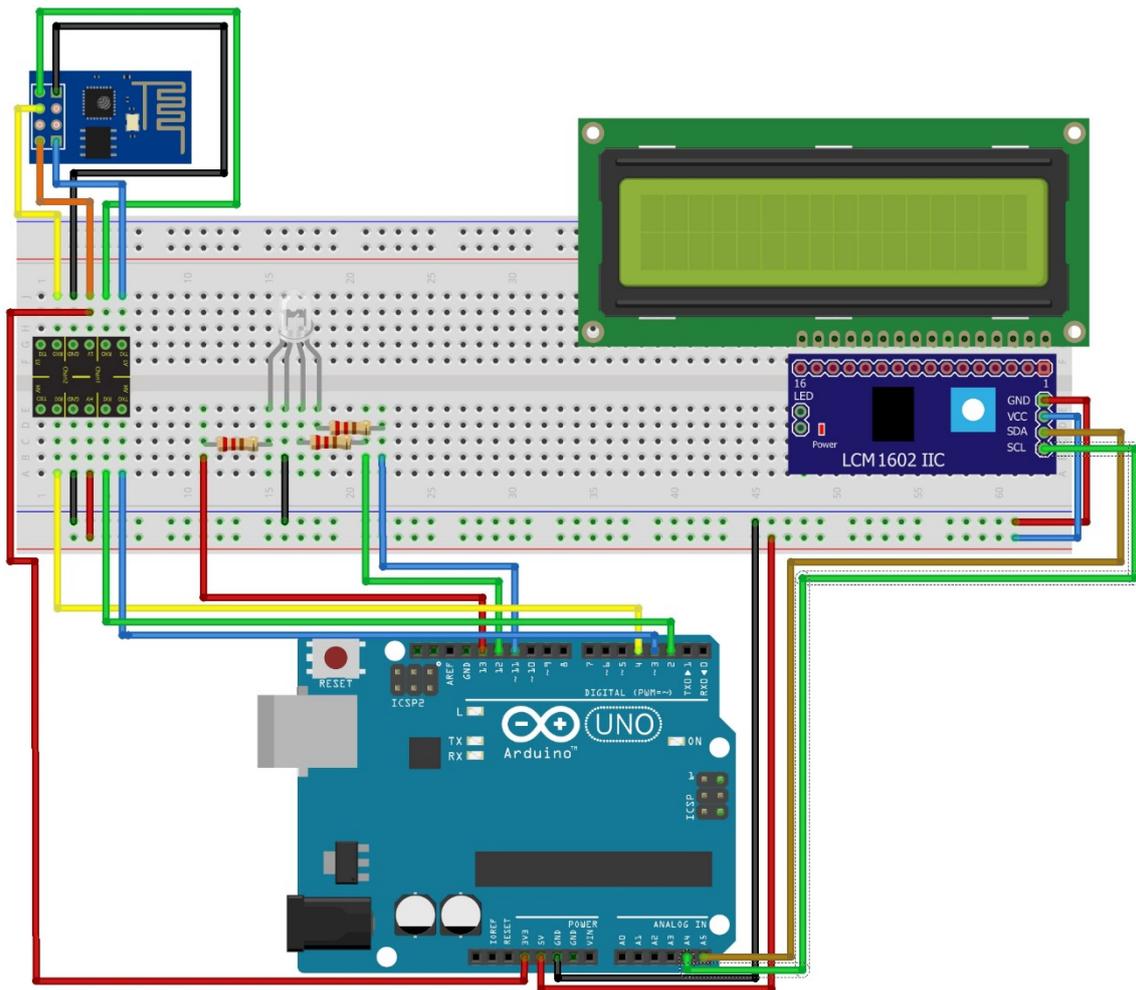
## **Materiais**

- Arduíno Uno
- Módulo ESP8266 – ESP01
- Módulo I2C
- Display LCD 16x2
- Led RGB
- 3 resistores de  $300\Omega$
- Protoboard
- Conversor de Nível Lógico RC
- Alguns jumpers
- Wi-Fi
- Computador

## **Procedimento Experimental**

Com a utilização dos materiais listados acima, foi possível montar um circuito capaz de acender um Led RGB ligado ao arduíno através de uma rede Wi-Fi.

Para começar, com a ajuda do data sheet encontrado na internet montou-se um circuito como o mostrado na figura 1.



fritzing

**Figura 1 - Circuito montado. Fonte: Própria.**

É importante saber que o módulo ESP8266 utilizado funciona através de uma corrente de 3.3V e como o arduíno gera uma corrente de 5V necessitou-se de um conversor de nível lógico. Outro item para levar em consideração é o módulo I2C, já que ao utiliza-lo diminui-se a necessidade de utilizar uma quantidade maior de jumpers.

Após a montagem do circuito, para ele funcionar é necessário a utilização de um código criado na IDE do arduíno, e além disso, um código em html que será utilizado para criação de uma pagina local que será responsável pela ação de ligar ou desligar o Led. É importante observar que o módulo ESP8266 precisa ser conectado em uma rede Wi-Fi e essa conexão é feita através do comando `AT+CWJAP` no código arduíno. Outro comando importante é o `AT+CIFSR` que retorna o IP que o módulo está utilizando para comunicação, esse IP será necessário no código html, já que sem ele não é possível mandar o comando de ligar ou desligar para o Led.

Com a finalização da criação dos códigos, é possível ligar e testar o funcionamento do circuito. Ao ligar o arduíno e compilar o código criado será necessário fazer as modificações cabíveis para o IP e a rede conectada, após isso será possível abrir a página html local e realizar a ligação ou desligamento do Led.

## **Resultados**

Foi um processo demorado desde o início até a finalização do projeto. Para chegar ao resultado final, foi necessário bastante testes e modificações, tanto no código quanto no circuito.

O ESP8266 é uma ótima ferramenta para conectar o arduíno à internet Wi-Fi, além de ser mais barato que alguns outros circuitos que prometem o mesmo resultam, ele possui um tamanho bastante reduzido, facilitando sua utilização. Durante o primeiro teste conseguimos observar que seria necessário utilização de algumas bibliotecas, como mostra o código abaixo.

Outra coisa observada, foi que era necessário a utilização de comando AT, esses comandos ajudam na conexão do ESP a internet, além de realizar diversas outras funções, todas elas explicadas através do data sheet do módulo ESP.

Após entender o funcionamento do ESP e montado o primeiro código, encontramos nosso primeiro erro, o módulo não conseguia conectar quando utilizávamos uma rede de um roteador. Como só conseguimos testar com um roteador do celular, pensamos que para o módulo funcionar era necessário um roteador de mesa e com acesso a internet, após a utilização do modem de casa conseguimos fazer o projeto funcionar, confirmando a teoria de que não funcionaria com roteador móvel. Até esse ponto tudo certo, mas ao testar em outro modem, obtivemos o mesmo erro anterior e ao testar novamente com o modem de casa que tinha funcionado antes, erro novamente.

Demorou um bom tempo até que durante uma análise ao código html no navegador percebemos que uma linha do código era executada de maneira errada, para mandar a ação o código html precisa fazer uma conexão com o módulo ESP e essa conexão é feita através do IP e da porta, por exemplo, nos

testes realizado utilizávamos o IP 192.168.43.76 e a porta 80, então no código precisaria ficar 192.168.43.76:80/, mas por algum motivo durante a compilação da página o IP era separado da porta e essa conexão não se realizava.

Após recriar um novo código html com o erro resolvido, conseguimos realizar o projeto sem nenhum outro problema. Abaixo é possível ver o código utilizado no Arduíno e o código da página html.

- **Código para Arduíno**

```
1 #include <SoftwareSerial.h>
2 #include <LiquidCrystal_I2C.h> // Inclui a biblioteca para utilizar o LCD.
3 #define DEBUG true
4
5 SoftwareSerial esp8266(2,3); //Especifica que o TX do esp8266 ta no pino 2 e o RX
no pino 3
6 LiquidCrystal_I2C lcd(0x3F, 16, 2); //Especifica o ldc utilizando o modulo i2c
7
8 int vermelho =11; //declara que o led vermelho ta no pino 11 do arduino
9 int verde =12; //declara que o led verde ta no pino 12 do arduino
10 int azul =13; //declara que o led azul ta no pino 13 do arduino
11 const int RST = 4; //declara o reset do modulo
12
13 void setup(){
14 Serial.begin(9600); //ajusta a velocidade da taxa de transferencia de dados da
porta serial para 9600
15 esp8266.begin(9600); //ajusta a velocidade da taxa de transferencia de dados do
esp8266 para 9600
16
17 //declara os pinos do LED RGB
18 pinMode(vermelho, OUTPUT);
19 pinMode(verde, OUTPUT);
20 pinMode(azul, OUTPUT);
21 delay(2000);
22
23 //Pulso em Reset para aceitar comunicacao
24 pinMode(RST, OUTPUT);
25 digitalWrite(RST, LOW);
26 delay(300);
27 digitalWrite(RST, HIGH);
28
29 //Comandos AT podem ser encontrado no site:
https://www.espressif.com/sites/default/files/documentation/4a-esp8266\_at\_instruction\_set\_en.pdf
30 sendData("AT+RST\r\n", 500, DEBUG); //AT+RST um comando utilizado para resetar o
modulo. Usado para aplicar as alteraçães realizadas.
31 sendData("AT+CWMODE=3\r\n", 500, DEBUG); //AT+CWMODE utilizado para configurar o modo
do esp, o =3 significa que ele está configurado como ponto de acesso e estáãéo.
Basicamente significa que ele pode conectar em uma rede ou ser conectado como em uma.
32 sendData("AT+CWJAP=\"teste\", \"12345678\"r\n", 10000, DEBUG); //AT+CWJAP ão utilizado
para conectar o esp em uma rede, onde necessãrio colocar o nome da rede e a senha.
33 sendData("AT+CIFSR\r\n", 2000, DEBUG); //AT+CIFSR ão um comando para retonar o ip que
o modulo está conectado.
34 sendData("AT+CIPMUX=1\r\n", 2000, DEBUG); //AT+CIPMUX habilita multiplas conexães
35 sendData("AT+CIPSERVER=1,80\r\n", 2000, DEBUG); //AT+CIPSERVER habilita a porta 80
36 }
37
38
39 void loop(){
40 lcd.init(); //inicia o lcd
41 lcd.clear(); //limpa a tela do lcd
42 lcd.backlight(); //liga o display
43 lcd.setCursor(0, 0); // Move o cursor do display para a primeira linha.
44 lcd.print("Aguardando sinal"); //mostra a mensagem
45
46 if(esp8266.available()){ //verifica se o esp8266 esta enviando mensagem e esta
disponivel
47 lcd.clear(); //limpa o lcd
```

```

48 lcd.backlight(); // liga o display
49 lcd.setCursor(0, 0); // Move o cursor do display para a primeira linha.
50 lcd.print("Recebendo sinal" ); // Manda o texto para a tela do display
51
52 if(esp8266.find("+IPD,")){
53 delay(1000);
54 int connectionId = esp8266.read()-48; //recupera apenas comando ASCII
55
56 esp8266.find("led="); //procura pela palavra led no sinal enviado pelo esp8266
57 int led = (esp8266.read()-48); //pega o primeiro numero enviado pelo esp e armazena
58 int onoff =(esp8266.read()-48); //armazena o segundo dado enviado, sendo ele o
responsvel pelo ligar ou desligar o led
59
60 lcd.clear(); //limpa o lcd
61 lcd.setCursor(0, 0); // Move o cursor do display para a segunda linha.
62 lcd.print("Sinal Recebido." ); // Manda o texto para a tela do display
63
64 //verifica qual o led que deve ser ligado ou desligado
65 if(led==1){ //se o dado salvo no led for 1
66 digitalWrite(vermelho, onoff); //vai realizar a aÃ§Ã£o para o led vermelho
67 }
68 if(led==2){ //se o dado salvo no led for 2
69 digitalWrite(verde, onoff); //vai realizar a aÃ§Ã£o para o led verde
70 }
71 if(led==3){ //se o dado salvo no led for 3
72 digitalWrite(azul, onoff); // digitalWrite(pinNumber, statusLed); //vai realizar a
aÃ§Ã£o para o led azul
73 }
74
75 // finaliza o comando no esp8266
76 String closeCommand = "AT+CIPCLOSE=";
77 closeCommand+=connectionId;
78 closeCommand+="\r\n";
79 //Encerra a conexao
80 sendData(closeCommand,1000,DEBUG);
81 }}}
82
83 //Metodo que envia os comandos para o esp8266
84 String sendData(String comando, const int timeout, boolean debug){
85 String resposta = ""; //variavel de resposta do esp8266
86
87 esp8266.println(comando); //manda a leitura dos caracteres para o esp8266
88
89 long int time = millis();
90 while( (time+timeout) > millis()){
91 while(esp8266.available()){
92 char c = esp8266.read(); //Concatena caracter por caractere recebido do modulo
esp8266
93 resposta+=c;}}
94
95 if(debug){ //debug de resposta do esp8266
96 Serial.println("Arduino : " + resposta);} //Imprime o que o esp8266 enviou para o
arduino
97
98 return resposta;}
99
100

```

- **Código em html**

```

1 <html lang="en">
2 <head>
3   <meta charset="utf-8">
4   <title>ESP8266</title>
5   <style>
6 <!-- edita o paragrafo -->
7   p {
8     color: red;
9     margin: 5px;
10    cursor: pointer;
11  }
12  p:hover {
13    background: yellow;

```

```

14     }
15 </style>
16 <script src="https://code.jquery.com/jquery-1.10.2.js"></script>
17 </head>
18 <body>
19   <center>
20     <!-- Titulos -->
21     <h1>Projeto de Sistema de Comunica  o utilizando ESP8266 e Arduino</h1>
22     <h2>Controle dos LED</h2>
23
24     <!-- Criar os botoes para cor vermelha -->
25     <button id="11" class="led">Liga Led vermelho</button>
26     <button id="10" class="led">Desliga Led vermelho</button>
27     </p>
28
29     <!-- Criar os botoes para cor verde -->
30     <button id="21" class="led">Liga Led verde</button>
31     <button id="20" class="led">Desliga Led verde</button>
32     </p>
33
34     <!-- Criar os botoes para cor azul -->
35     <button id="31" class="led">Liga Led azul</button>
36     <button id="30" class="led">Desliga Led azul</button>
37     </p>
38   </center>
39   <script>
40     // criar uma a  o pra quando clicar no botao
41     $( ".led" ).click(function() { // cria a a  o para todos da classe led
42       var p = $(this).attr('id');
43
44       $.get("http://192.168.43.76:80/", {led:p}); //manda o dado com o id do lad
correspondente para o Ip+porta utilizado
45
46
47     });
48   </script>
49 </body>
50 </html>

```

## Conclus  o

Dessa forma, antes de atingir o resultado esperado ocorreram algumas complica  es no decorrer do projeto. Entretanto, com um estudo mais detalhado sobre o m  dulo utilizado e conhecimento adquiridos no decorrer das aulas, tornou-se mais f  cil a manipula  o do sistema. Contudo, o procedimento foi completado com   xito. Um dado ser mandado por um componente e recebido por outro atrav  s de uma rede Wi-Fi, fazendo um Led ligar ap  s o recebimento do dado.

## Referências bibliográficas

CURVELLO, A. Apresentando o módulo ESP8266. Disponível em: <<https://www.embarcados.com.br/modulo-esp8266/>>. Acesso em: 8 de novembro de 2017.

ESP8266 Community Forum. Disponível em: <<http://www.esp8266.com/>>. Acesso em: 8 de novembro de 2017.

ESP8266EX Datasheet. Disponível em: <[https://www.mikrocontroller.net/attachment/231858/0A-ESP8266\\_Specifications\\_v4.pdf](https://www.mikrocontroller.net/attachment/231858/0A-ESP8266_Specifications_v4.pdf)>. Acesso em: 8 de novembro de 2017.

MINATEL, P. 2014. Utilizando o módulo ESP8266 com comandos AT. Disponível em: <<http://pedrominatel.com.br/pt/esp8266/utilizando-o-modulo-esp8266-com-comandos-at/>>. Acesso em: 8 de novembro de 2017.

PIXININE, J. 2015. Como um Wi-Fi funciona? Entenda a tecnologia. Disponível em: <<http://www.techtudo.com.br/noticias/noticia/2015/02/como-um-wi-fi-funciona-entenda-tecnologia.html>>. Acesso em: 12 de novembro de 2017.

RoboCore. Módulo WiFi - ESP8266. Disponível em: <<https://www.robocore.net/loja/produtos/modulo-wifi-esp8266.html>>. Acesso em: 8 de novembro de 2017.

THOMSEN, A. 2015. Tutorial Módulo Wireless ESP8266 com Arduino. Disponível em: <<https://www.filipeflop.com/blog/esp8266-arduino-tutorial/>>. Acesso em: 8 de novembro de 2017.