Material de apoio - Modulo 3

Conectando a ESP32 na rede WiFi

É muito simples conectar a ESP32 ao WiFi, basta utilizar a seguinte função passando o SSID da rede e a senha como argumento:

```
WiFi.begin("Nome_da_rede", "senha123");
```

Para verificar se a conexão foi bem sucedida, você pode utilizar a as funções `WiFi.status()` , que retorna WL_CONNECTED se a conexão foi estabelecida, e `WiFi.localIP()` , que retorna o endereço IP do seu dispositivo na rede.

```
// Aguardando a conexão ser bem sucedida
while(WiFi.status() != WL_CONNECTED);

// Quando conectar, envia o endereço pela serial
Serial.println(WiFi.localIP());
```

Ainda, na aula vimos outras funções que podem ser úteis dependendo de sua necessidade. A primeira faz um scan nas redes e retorna a quantidade de redes que a placa encontrou e que estão disponíveis para serem conectadas

```
WiFi.scanNetworks();
```

Depois de escanear, você pode analisar informações de acordo com o número da rede retornado pelo scan. Algumas das funções que exibem informações da rede são essas:

```
WiFi.SSID(numero_da_rede);            // Retorna o nome da rede
WiFi.RSSI(numero_da_rede);            // Retorna a qualidade do sinal
WiFi.encryptionType(numero_da_rede);  // Retorna o tipo de encryptação da rede
```

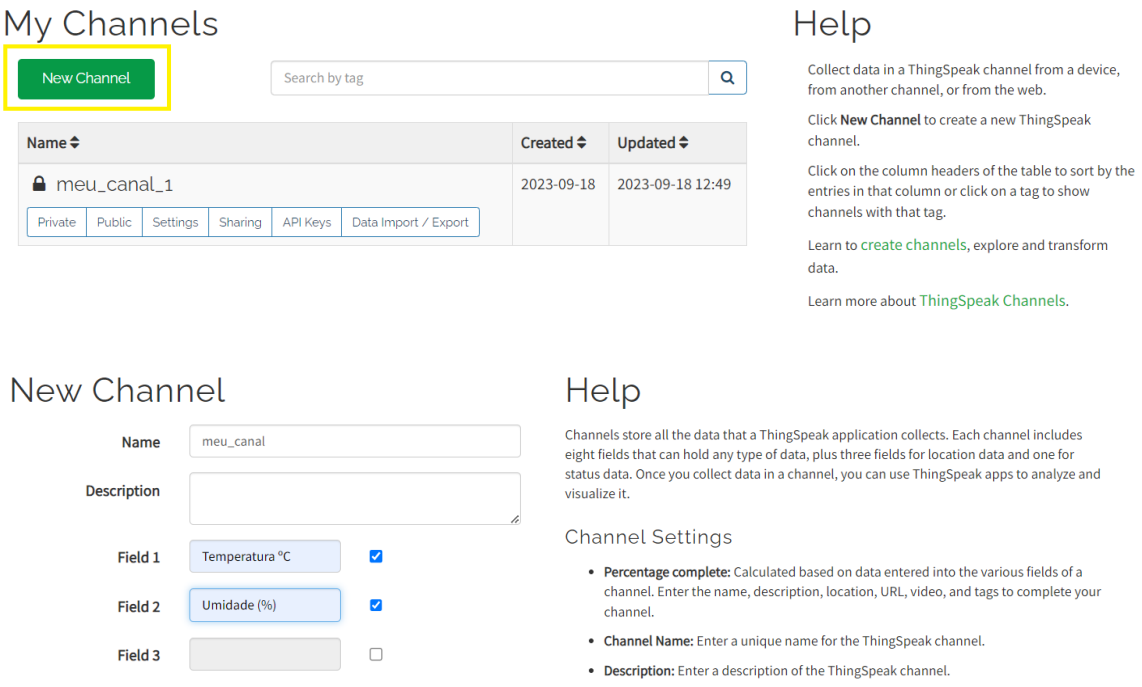
Depois de feito o scan, você deve excluir essas informações da memória:

```
WiFi.scanDelete();
```

Enviando leitura de temperatura e umidade para a plataforma ThingSpeak

ThingSpeak é uma plataforma da Internet das Coisas (IoT) que permite aos usuários coletar, analisar e visualizar dados de sensores em tempo real. Foi desenvolvida pela MathWorks, a empresa por trás do MATLAB, e é comumente usada em projetos de IoT e monitoramento ambiental.

Crie uma conta na plataforma e abra um novo canal:



Cada “Field” se traduzirá em um gráfico dentro do canal. Você precisará fornecer dados a este campo por meio de requisições HTTP que serão geradas pela ESP 32

Protocolo HTTP

Uma requisição HTTP (Hypertext Transfer Protocol) é um pedido feito por um cliente a um servidor da web para solicitar informações ou ações específicas. O HTTP é o protocolo padrão para a comunicação entre navegadores da web (clientes) e servidores da web (servidores) na Internet. Essas requisições são a base da comunicação na World Wide Web (WWW) e são usadas sempre que você acessa uma página da web, faz uma pesquisa online, envia um formulário ou realiza qualquer outra interação com um servidor da web.

Como funciona a requisição HTTP?

Ao digitar uma URL em um navegador, você está, na verdade, enviando dados a um servidor e esperando uma resposta. Ao digitar “google.com”, por exemplo, você está solicitando a página inicial do google para um servidor do google. Esta solicitação se chama **requisição** e é gerada pelo navegador. Veja o passo:

- O usuário insere a URL “google.com”
- O navegador gera a requisição e envia através da rede até o servidor:

```
GET / HTTP /1.1
Host: www.google.com.br
```

- O servidor responde dizendo que entendeu a requisição e retorna o que foi solicitado (que, no caso, é a página do google)

```
HTTP/1.1 200 OK
content-Type: text/html; charset=UTF-8
date: Mon, 30 Abril 2018 17:00:02 GMT
connection: close
content-Length: 438
```

- Ao chegar a resposta, o browser constrói a página. São três linguagens que montam a página web que você enxerga: o HTML, CSS e JavaScript;

Para uma utilização comum da plataforma ThingSpeak, você não precisa conhecer HTML, CSS e JavaScript, pois não precisará construir sites nesse momento. Você precisará apenas conhecer as requisições HTTP do tipo GET.

As requisições do tipo GET permitem enviar dados ao servidor por meio da URL. Veja a requisição do tipo GET feita abaixo para pesquisar por “Escola Instructiva” no google.

```
https://www.google.com/search?q=escola+instructiva&pt-BR
```

Criando a requisição GET na ESP32

Você precisará gerar o texto da requisição que será enviado ao servidor (ThingSpeak). Para isso, **você precisará do seu Channel ID e da API KEY**, além do nome do Host, que neste caso é `api.thingspeak.com` . A requisição ficará desta forma:

```
GET /update?api_key=5KR0WV0KZZHD5VTX&field1=25.0 HTTP/1.1
Host: api.thingspeak.com
Connection: close
```

Os espaços entre as linhas devem ser considerados no código com `\r\n` .

Para realizar a requisições, a ESP32 funcionará como cliente, por isso criamos o objeto client e enviando o texto de requisição por meio dele. O arquivo <WiFi.h> é quem possui esta definição, por isso não é necessário inserir nenhuma biblioteca.

```
const char* host = "api.thingspeak.com";   // Este é o servidor
const int httpPort = 80;                   // Esta é a porta onde ocorrerá a comunicação

WiFiClient client;

if (!client.connect(host, httpPort)){      // Estabelece a conexão
    return;
}

client.print("Texto da requisição");        // Envia a request
```

Ao encerrar o arquivo de requisição **Connection: close**, a conexão é automaticamente encerrada. Portando, para realizar outra comunicação é necessário restabelecê-la.

ESP32 como server

É muito simples transformar a ESP32 em um Web Server. Primeiro, vamos instanciar um novo servidor dentro do código, utilizamos a porta 80 como padrão.

```
WiFiServer server(80);
```

Depois de conectado ao wifi, esse server pode ser inicializado.

```
server.begin();
```

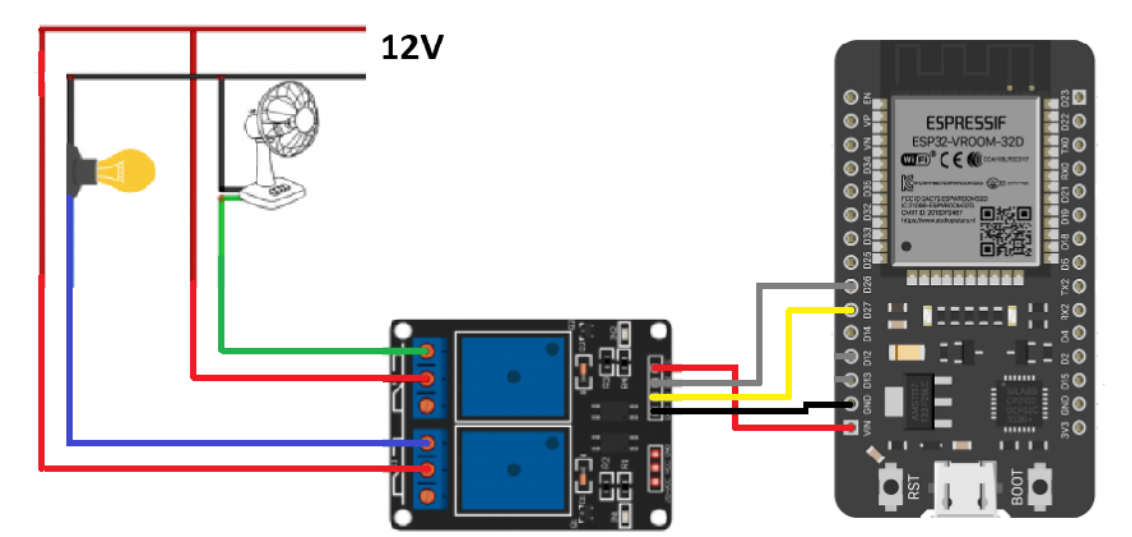
Se um cliente se conectar, ele se torna available. Então o objeto client pode ser utilizado para fazer leituras e escritas neste canal HTTP.

```
WiFiClient client = server.available();
char c = client.read();
```

```
client.println("conteudo");
client.stop();
```

Enviando comandos via GET

Vamos utilizar o protocolo HTTP para realizar este acionamento:



Basicamente, teremos que receber um comando que estará junto do cabeçalho da requisição. Por isso, você pode utilizar a função `indexOf` que retorna a posição de um caractere dentro da string. Se não houver aquele caractere dentro da string, ela retornará 0. Se você utilizar `/led/on` como comando, por exemplo, `indexOf` deverá procurar por este comando dentro da requisição:

```
if(header.indexOf("GET /led/on") >= 0 ){
    // Realiza algum comando
}
```

Onde `header` é a string que contém toda a requisição vinda do navegador. O comando em questão é acionar o relé. Mas você pode colocar o comando que quiser. Agora, obrigatoriamente, o usuário deverá enviar a requisição desta forma para enviar o comando:

```
http://192.168.248.33/led/on
```

Lembre-se de utilizar o endereço correspondente da sua ESP32.