Acionamento de LED com Web Socket

Nesse tutorial, será apresentado como controlar dois Leds conectados ao Arduino, via rede ethernet local.

Obs.: Esse exemplo pode ser utilizado para qualquer projeto que você queira desenvolver com Arduino, e os códigos estão disponíveis para download.

Requisitos:

- Uma placa Arduino (nesse exemplo foi utilizado o Arduino Mega 2560);
- Um módulo de rede ethernet ENC28J60;
- Biblioteca UIPEthernet.h para o módulo de rede;
- Instalar servidor XAMPP e IDE do Arduino;
- Cabo USB;
- Cabo Rede;
- Infraestrutura de rede local;

Conexões:

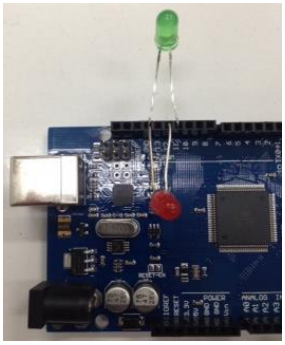
LEDS X Arduino:

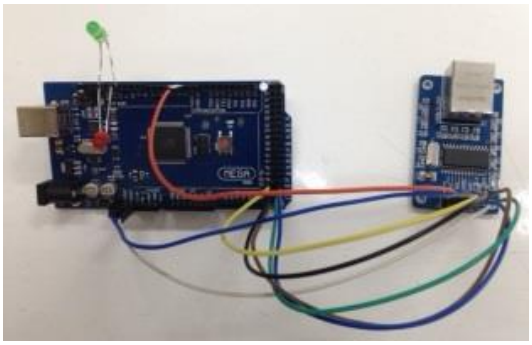
LED Verde	ARDUINO	LED Vermelho	ARDUINO
GND	GND	GND	GND
VCC	Pino digital 11	VCC	Pino digital 12

Módulo X Arduino

Módulo	Arduino
INT	Pino digital 2
SO	Pino digital 50
SCK	Pino digital 52
RESET	RESET
GND	GND
VCC	3.3V
CS	Pino digital 53
SI	Pino digital 51

Conexões Prontas:





Procedimentos

Passo 1: Instalação do Servidor WEB.

- Para que os serviços web funcionem, ou seja, as requisições enviadas através da Interface web (index.html) para o socket (arduino.php) e do socket para o Arduino, será necessário instalar os servidores que fornecem esses serviços, ou seja, Apache e PHP. Para facilitar essa instalação, sugiro o XAMPP que se trata de uma ferramenta que une esses serviços em apenas uma instalação.

  - Faça o download do XAMPP através desse endereço e instale-o: https://www.apachefriends.org/pt_br/download.html
  - Após a instalação, clique na opção **Start** da coluna **Action** na linha do serviço **Apache**;
  - Após inicializar o serviço Apache, o XAMPP pode ser minimizado.

Passo 2: Entendendo os Códigos.

- *controleLed.ino* – Esse arquivo contém o código que deverá ser gravado no Arduino, e possui comentários referentes as funções utilizadas;
- Insira o endereço IP de acordo com a faixa utilizada em sua rede, e é claro, um endereço que esteja disponível.

```
// Biblioteca necessária para utilização do Módulo Ethernet ENC28J60;
#include <UIPEthernet.h>

// Definição dos Pinos utilizados para os Leds;
#define gpio_11 11
#define gpio_12 12

// Instanciamento do serviço de rede com atribuição de um nome
// para o servidor(server) e da porta(80) utilizada;
EthernetServer server = EthernetServer(80);

void setup()
{
    // Deifinindo os pinos como pinos de saída;
    pinMode(gpio_11, OUTPUT);
    pinMode(gpio_12, OUTPUT);

    //As funções abaixo são nativas da biblioteca UIPEthernet.h

    // Endereço MAC para o módulo de rede(Não precisa ser alterado
    //caso tenha apenas este módulo em sua rede);
    uint8_t mac[6] = {0x00,0x01,0x02,0x03,0x04,0x05};
    // Endereço IP para o Arduino;
    IPAddress myIP(10,1,52,46);
    // Função que envia os endereços MAC e IP para o módulo de rede;
    Ethernet.begin(mac,myIP);
    // Função que inicia o módulo de rede de acordo com as atribuições anteriores;
    server.begin();

}

void loop()
{
    // Declaração da variável client que receberá as requisições de server.available();
    EthernetClient client = server.available();

    // Estrutura de seleção que receberá as requisições do socket criado em arduino.php;
    if(client)
    {
        // client.read() - Função nativa da biblioteca UIPEthernet.h que recebrá o valor do scoket;
        switch (client.read())
        {
            // '1' - Valor recebido do socket que acionará a função digitalWrite(gpio_11, HIGH);
            case '1':
              digitalWrite(gpio_11, HIGH);
              break;
            case '2':
              digitalWrite(gpio_12, HIGH);
              break;
            case '3':
              digitalWrite(gpio_11, LOW);
              break;
            case '4':
              digitalWrite(gpio_12, LOW);
              break;
            default:
              delay(1);
        }
    }
    // client.stop() - Função que para o serviço de rede até que chegue
    //uma nova requisição via socket;
    client.stop();
}
```

- arduino.php – Esse arquivo contém a criação do web socket e possui comentários referentes as funções utilizadas.

```php
<?php

// $sock - Variável declarada para ciração do socket;
// socket_create - Função nativa do PHP para habilitar o novo socket;
// http://php.net/manual/pt_BR/function.socket-create.php - Tutorial completo sobre a função socket_create e seus parâmetros;
$sock = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);

// socket_connect - Função que irá conectar o web socket desenvolvido em PHP com o Arduino;
// Parâmetros do socket_connect - (Nome do scoket, IP do Arduino, Porta);
socket_connect($sock,"10.1.52.46", 80);

// Estrutura de seleção que receberá as requisições da interface web(index.html) e associar com o parâmetro do $_POST['estadoLed'];
if ($_POST['estadoLed']=="lverde_ligado")
{
    // socket_write - Função nativa do PHP que enviará via socket o valor("1") para estrutura de seleção(switch (client.read()) do arduino,
    // para acionamento do LED;
    socket_write($sock, "1");
}
if ($_POST['estadoLed']=="lvermelho_ligado")
{
    socket_write($sock, "2");
}
if ($_POST['estadoLed']=="lverde_desligado")
{
    socket_write($sock, "3");
}
if ($_POST['estadoLed']=="lvermelho_desligado")
{
    socket_write($sock, "4");
}

// header - Função nativa do PHP que retorna a index.html após acionamento dos botões;
header("Location: index.html");

// socket_close - Função nativa do PHP que encerra a conexão com o socket;
socket_close($sock);
?>
```

- index.html – Esse arquivo contém a estrutura da interface web.

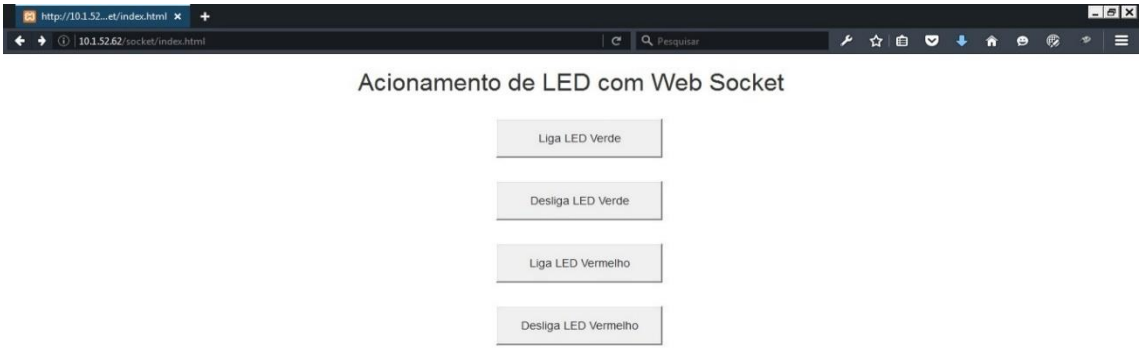
```html
<!DOCTYPE html>
<html>
<head>
    <title></title>
    <link rel="stylesheet" type="text/css" href="js/bootstrap/bootstrap.css">

    <!-- Tag <style> criada para alinhamento e padronização dos botões -->
    <style>
    .body
    {
        text-align: center;
    }
    .botao
    {
        width: 200px;
        height: 50px;
        margin-top: 20px;
        margin-left: 20px;
    }
    </style>

</head>
<body>
    <div class="body">
        <h2>Acionamento de LED com Web Socket</h2>
        <!-- POST - Método utilizado para enviar as informações para o socket criado no arquivo arduino.php -->
        <form method="POST" action="arduino.php">
        <p><button class="botao" type="submit" value="lverde_ligado" name="estadoLed">Liga LED Verde
        </button>
        </p>
        </form>

        <form method="POST" action="arduino.php">
        <p><button class="botao" type="submit" value="lverde_desligado" name="estadoLed">Desliga
        LED Verde</button>
        </p>
        </form>

        <form method="POST" action="arduino.php">
        <p><button class="botao" type="submit" value="lvermelho_ligado" name="estadoLed">Liga LED
        Vermelho</button>
        </p>
        </form>

        <form method="POST" action="arduino.php">
        <p><button class="botao" type="submit" value="lvermelho_desligado" name="estadoLed">Desliga
        LED Vermelho</button>
        </p>
        </form>
    </div>
</body>
</html>
```

Passo 3: Enviando os códigos para o servidor web e gravando código no Arduino.

- ❖ Tendo os códigos criados, será necessário envia-los para o servidor, ou seja, os mesmos precisam existir no servidor para que os serviços web façam a integração entre a interface web e o Arduino.
- ➲ Crie uma pasta e copie os arquivos index.html e arduino.php para dentro dessa pasta. Dê o nome que preferir para pasta, nesse caso, a nomeei como socket. Junto com os arquivos dos códigos para download, há uma pasta chamada **js**, copie-a também para dentro da pasta que você criou, pois, a mesma possui a biblioteca do **CSS** que estamos utilizando para alinhar os botões no arquivo index.html;
- ➲ Copie a pasta, maximize o XAMPP e clique no botão **Explore** na coluna de botões localizada a direita da tela do mesmo;
- ➲ A pasta **htdocs** abrirá automaticamente, com isso, cole a pasta que você criou para dentro dela;
- ➲ Conecte o Arduino ao seu computador através do cabo USB, conecte o cabo de rede no módulo de rede e no seu modem ou roteador;
- ➲ Abra o arquivo controleLed.ino, e execute a gravação do código para o Arduino;
- ➲ Aguarde a gravação concluir.

Passo 4: Testando a aplicação.

- ❖ Se os procedimentos foram realizados e concluíram sem erros, a aplicação está pronta para ser testada.
- ➲ Abra o seu navegador e na barra de endereços digite: http://endereço_IP_do_seu_computador/socket/index.html
- ➲ O navegador deverá abrir a página conforme a imagem abaixo, se isso acontecer o projeto foi implementado com sucesso, PARABÉNS!



Atenciosamente,
Ricardo Jr. Abreu da Silva
Técnico em Redes de Computadores – IFSC – São José - SC
Estudante de Ciência da Computação – UNIVALI – São José - SC
55 48.99574635(WhatsApp) – ricardojras19@gmail.com