

## TUTORIAL: PASSO A PASSO - CONTROLE DE VENTILADOR E CAFETEIRA

**OBS: SE ESTIVER DIFÍCIL DE ENXERGAR AS IMAGENS, BASTA APERTAR AS TECLAS CTRL E + SIMULTANEAMENTE PARA DAR ZOOM NO PDF.**

### O que é Android?

*Android é o sistema operacional do Google para dispositivos móveis baseado no Linux.*

### O que é Arduino?

*Arduino é um projeto totalmente aberto de protótipos de eletrônica baseados numa plataforma de hardware e software flexível e de fácil utilização. É destinado a artistas, designers, hobbyistas e qualquer tipo de pessoa interessada em criar objetos ou ambientes interativos. É um projeto que engloba software e hardware e tem como objetivo fornecer uma plataforma fácil para prototipação de projetos interativos, utilizando um microcontrolador. Ele faz parte do que chamamos de computação física: área da computação em que o software interage diretamente com o hardware, tornando possível integração com sensores, motores e outros dispositivos eletrônicos.*

### Automação utilizando Arduino em conjunto com o Android:

*Uma combinação perfeita quando se fala de automação. Os passos para fazer o Android comandar um Arduino são bem simples, e não requer muita programação em ambos. Em poucas linhas de código, você consegue criar uma aplicação Android que comunique com seu Arduino, e ao mesmo tempo consegue programar seu Arduino para executar e responder aos comandos enviados pelo Android. Esse tutorial visa mostrar o controle de um ventilador e uma cafeteira através da aplicação Android, onde os dois equipamentos estarão ligados em um módulo relé 5V de 2 canais e os equipamentos poderão se ligados e desligados e, o status atual de cada um deles será mostrado na aplicação Android.*

Caso você se interesse em começar a programar aplicações Android para interagir com o Arduino, irá precisar das seguintes ferramentas:

**IDE Eclipse (Ambiente de Desenvolvimento):** Essa ferramenta é utilizada para desenvolvimento de aplicações na linguagem Java. A programação para Android é basicamente o Java, portanto o Eclipse com o devido plugin Android instalado, permite o desenvolvimento de Aplicações para dispositivos Android.

**Android SDK:** Esse pacote trás consigo todo conteúdo necessário para programação de aplicativos Android, desde bibliotecas até as API's disponibilizadas pela Google. Além de bibliotecas e API's, o Android SDK trás também os AVD's (Android Virtual Device), que são os emuladores virtuais de um dispositivo físico rodando o Android como plataforma. Os AVD's servem para que você teste as aplicações durante o desenvolvimento.

Abaixo seguem dois links (**alternativos**) para que você possa baixar e configurar o Eclipse corretamente, de forma que ele trabalhe com o Android:

LINK 1: <http://goo.gl/nWiFxd>

LINK 2: <http://goo.gl/MNZjJL>

**IDE Arduino (Ambiente de Desenvolvimento):** Essa ferramenta é utilizada para programação do Arduino. Além de programar, ela é responsável por introduzir o código dentro do Arduino.

Acesse o DVD e entre no seguinte caminho: "Drivers e IDE de Desenvolvimento Arduino" dentro da pasta está a IDE pronta para ser executada. Copie a pasta para algum lugar em seu PC. Para abrir a IDE, basta executar o arquivo "arduino.exe".

Gostaria de ressaltar uma coisa: a aplicação android que está anexada a este material e que você irá utilizar, é uma aplicação simples e que precisa ser melhorada, contudo é uma aplicação funcional que mostrará a você como é simples automatizar utilizando Arduino e Android.

A partir daqui muita atenção para execução dos passos.

**IMPORTANTE: OS PASSOS DESCRITOS ABAIXO FORAM BASEADOS NO ARDUINO MEGA 2560, PORÉM SÃO APLICADOS NOS KIT'S COM ARDUINO UNO. LEMBRE-SE: O KIT ADQUIRIDO É PARA APRENDIZAGEM, PORTANTO SERÁ NECESSÁRIO QUE VOCÊ PESQUISE E APRENDA CASO ENCONTRE DIFICULDADES.**

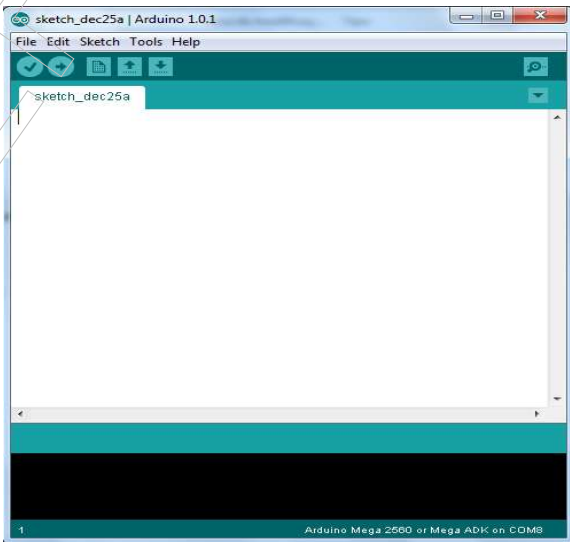
LEMBRE-SE: Não precisa ter pressa, seja bastante paciente, mantenha a calma sempre. Se por ventura você não conseguir fazer funcionar ao final das explicações, volte e refaça tudo novamente com mais calma e lembre-se que o Google é seu maior amigo online. Nada nessa vida é difícil, se fizermos tudo com dedicação e paciência, as coisas mais complicadas se tornam muito simples.

Inicialmente você precisa conectar o Ethernet Shield ao Arduino. Veja na imagem abaixo como eles ficam quando conectados.



Agora ligue o cabo USB ao arduino, em seguida conecte a outra ponta ao seu computador. Aguarde a instalação do arduino (**CASO O WINDOWS NÃO ENCONTRE OS DRIVERS DE INSTALAÇÃO, FAÇA A INSTALAÇÃO MANUAL, INDICANDO A PASTA DRIVERS QUE ESTÁ NA PASTA DA IDE DE DESENVOLVIMENTO DO ARDUINO**). Conecte uma ponta do cabo de rede no ethernet shield, e a outra ponta ligue em uma das portas do seu roteador .

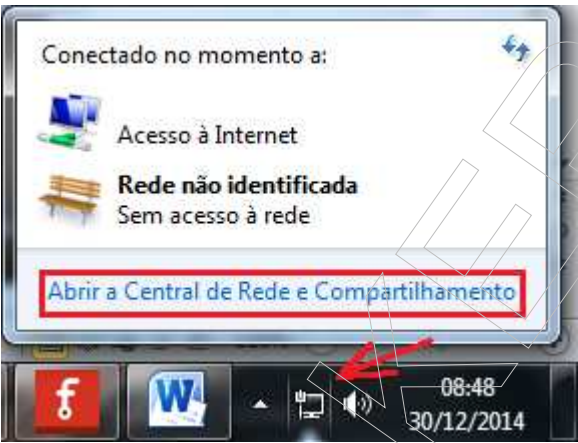
Com seu arduino instalado e acoplado ao ethernet shield, execute a IDE de desenvolvimento (**arduino.exe**).



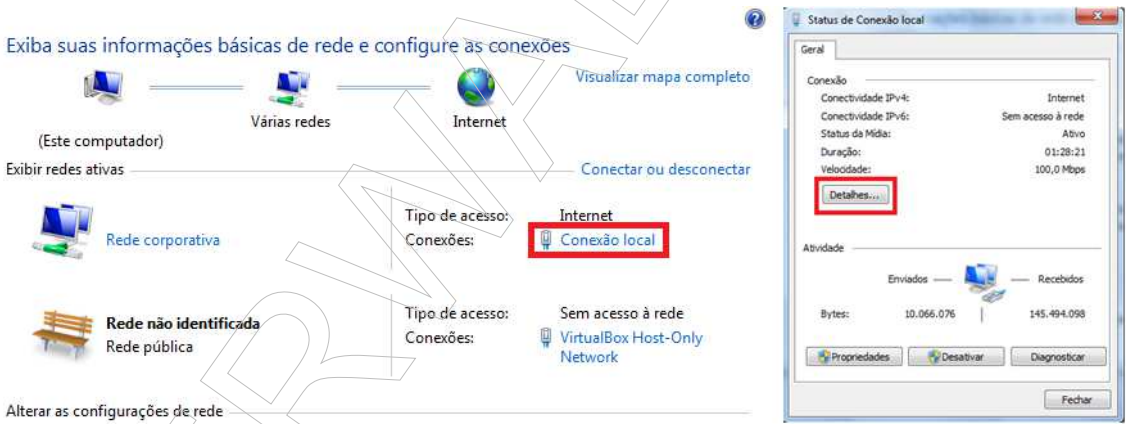
Em seguida clique no menu File>Open, e vá até a pasta “Android e Arduino” em seguida entre na pasta “PASSO a PASSO - Controle de Ventilador e Cafeteira” e depois “Exemplo Ventilador Cafeteira”. Dentro dessa pasta tem um arquivo que é reconhecido pela IDE do Arduino, dê um duplo clique nele e aguarde ele abrir na IDE de desenvolvimento do Arduino.

Com o arquivo aberto você terá que fazer algumas edições para que ele fique configurado de acordo com o seu roteador.

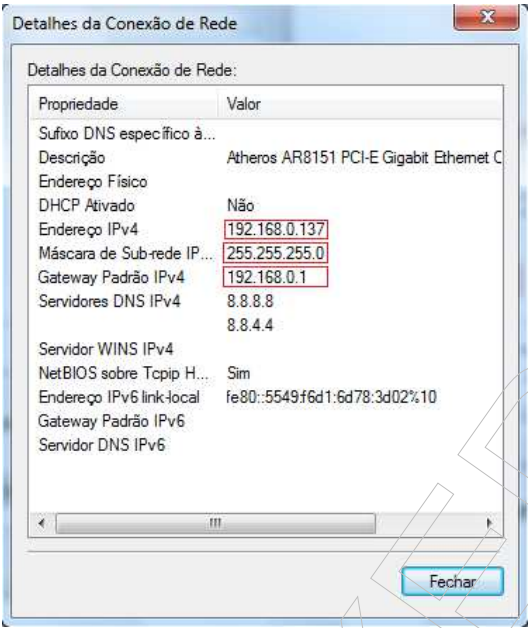
Primeiramente vá até a barra de tarefas do Windows e clique sobre a sua conexão de internet e em seguida em “Abrir a Central de Rede e compartilhamento”.



Clique em “Conexão local” e em seguida em “Detalhes”.



Após clicar em “Detalhes” você terá informações importantes para configuração posterior do Ethernet Shield.



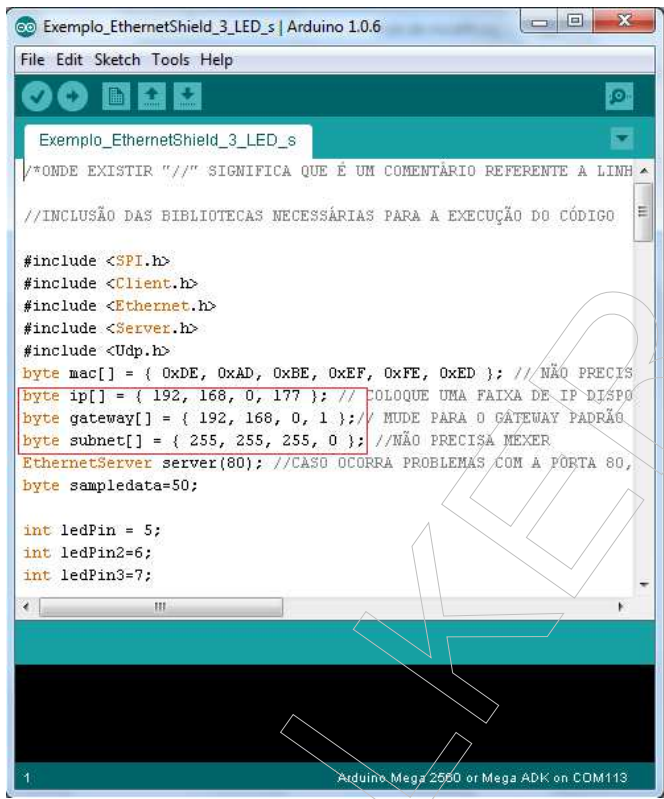
Veja que na imagem estão destacados as informações que são necessárias para configuração do Ethernet Shield. Vale ressaltar que seu IP pode possuir um “1” ou “2” no lugar do “0” (Exemplo: 192.168.2.200 / 192.168.1.150). Pode ocorrer de a rede possuir faixa de IP 10.2.x.xxx(Exemplo: 10.2.2.102 / 10.2.1.200). Isso não vai influenciar em nada, o importante é que você configure corretamente no Ethernet Shield:

**Endereço IPv4** - No Ethernet Shield será o “byte ip[]”. Deve ser um IP na mesma faixa da sua rede. Na imagem acima o IP é o “192.168.0.137”, portanto no “byte ip[]” posso colocar qualquer IP(QUE JÁ NÃO ESTEJA SENDO USADO) entre “192.168.0.100” e “192.168.0.254”.

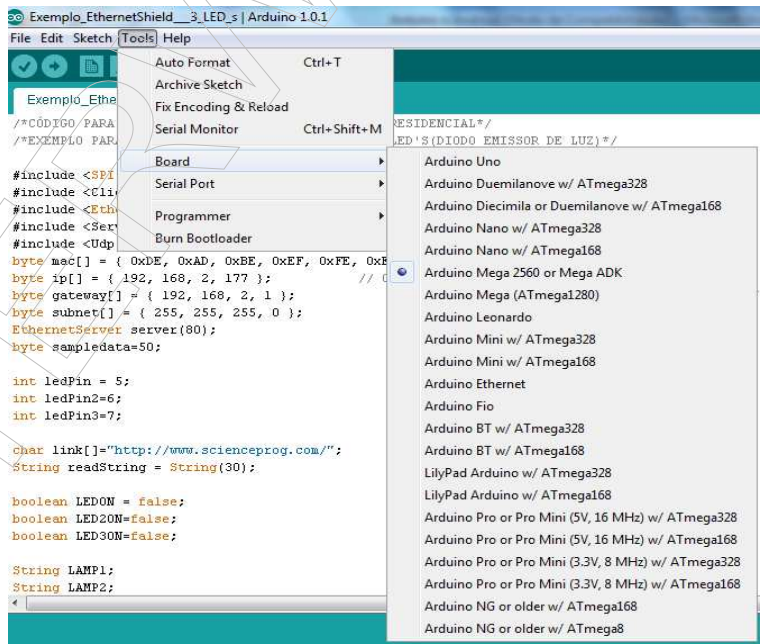
**Gateway Padrão IPv4** - No Ethernet Shield será o “byte gateway[]”. Você deverá colocar o valor do seu “Gateway Padrão IPv4” no “byte gateway[]”.

**Máscara de Sub-rede IPv4** - No Ethernet Shield será o “byte subnet[]”. Na maioria das redes domésticas será “255.255.255.0”. Você deverá colocar o valor da sua “Máscara de Sub-rede IPv4” no “byte subnet[]”.

Recapitulando: Deixe as informações de IP separadas em um bloco de notas para utilizar na configuração do Ethernet Shield. Na imagem abaixo você pode ver o código já configurado com as informações da minha rede. Basta que você substitua as informações do código pelas informações da sua rede que foram identificadas nos passos atrás.

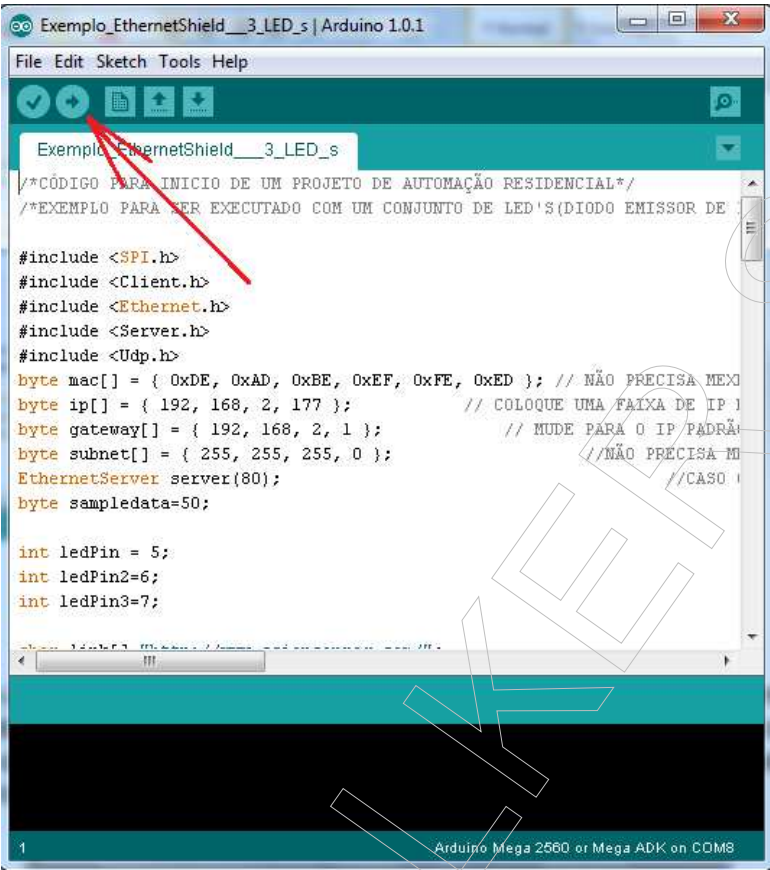


Após terminar de configurar as informações de IP o código está pronto para ser utilizado. Antes de carregar o código no arduino, verifique se a placa setada no programa é a correta e se a porta também é a correta na opção **Serial Port**.





Agora carregue o programa no arduino.

A screenshot of the Arduino IDE interface. The title bar reads 'Exemplo\_EthernetShield\_\_3\_LED\_s | Arduino 1.0.1'. The menu bar includes 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. Below the menu bar is a toolbar with icons for opening, saving, uploading, and downloading. A red arrow points to the upload icon (a right-pointing arrow). The main text area contains C++ code for an Ethernet-based LED control project. The code includes headers for SPI, Client, Ethernet, Server, and UDP. It defines MAC and IP addresses, a gateway, and a subnet. It also sets up an EthernetServer on port 80 and defines three LED pins (5, 6, and 7). The status bar at the bottom indicates '1' and 'Arduino Mega 2560 or Mega ADK on COM8'.

```
Exemplo_EthernetShield__3_LED_s

/*CÓDIGO PARA INICIO DE UM PROJETO DE AUTOMAÇÃO RESIDENCIAL*/
/*EXEMPLO PARA SER EXECUTADO COM UM CONJUNTO DE LED'S(DIODO EMISSOR DE

#include <SPI.h>
#include <Client.h>
#include <Ethernet.h>
#include <Server.h>
#include <Udp.h>

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; // NÃO PRECISA MEMO
byte ip[] = { 192, 168, 2, 177 }; // COLOQUE UMA FAIXA DE IP
byte gateway[] = { 192, 168, 2, 1 }; // MUDE PARA O IP PADRÃO
byte subnet[] = { 255, 255, 255, 0 }; // NÃO PRECISA MU
EthernetServer server(80); //CASO
byte sampledata=50;

int ledPin = 5;
int ledPin2=6;
int ledPin3=7;
```

Feche a IDE de desenvolvimento.

Agora você terá que fazer as ligações do módulo relé 5V no Arduino e do ventilador e da cafeteira. Lembre-se de sempre efetuar as ligações com seu Arduino desconectado da porta USB.

**ATENÇÃO: VOCÊ ESTÁ PRESTES A TRABALHAR COM CORRENTE/ TENSÃO ALTERNADA(127V / 220V ) EM UMA PARTE DO CIRCUITO. É IMPORTANTE QUE VOCÊ ESTEJA CIENTE DO RISCO DE LEVAR CHOQUE, FAZER UMA LIGAÇÃO ERRADA E GERAR UM CURTO CIRCUITO OU DANIFICAR COMPONENTES DEVIDO AS LIGAÇÕES ERRADAS.**

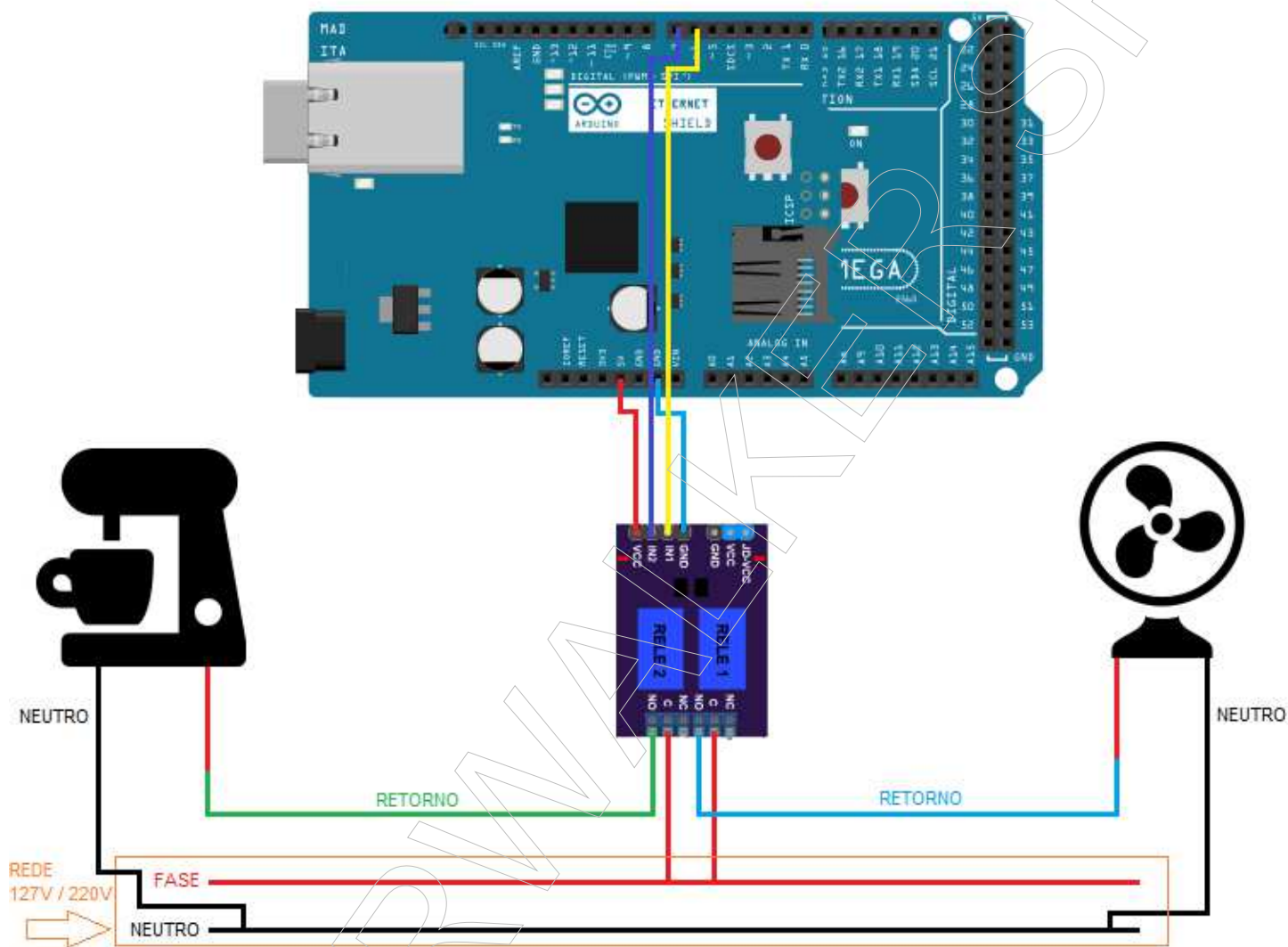
Basicamente: preocupe-se em **NÃO** ligar **FASE** com **NEUTRO(127V)** ou **FASE** com **FASE(220V)**.

Material necessário:

- Arduino Mega 2560 R3
- Ethernet Shield
- Módulo relé 5V de 2 CANAIS
- Ventilador 127V/220V
- Cafeteira 127V/220V
- Fita isolante
- Chave de teste (PARA DESCOBRIR FASE E NEUTRO EM REDES 12V)
- Cabos para conexão(jumper's e cabos de no mínimo 2,5mm para ligar os equipamentos)
- Plug's fêmea

Ligue de acordo com a imagem abaixo:

**OBS: MANTENHA TODO O CIRCUITO SEM ENERGIA ENQUANTO FAZ AS LIGAÇÕES. PARA QUALQUER MODIFICAÇÃO FÍSICA DO CIRCUITO, NUNCA SE ESQUEÇA DE DESLIGAR A ENERGIA. LIGUE A FASE E O NEUTRO POR ÚLTIMO.**



**Lembre-se de sempre utilizar fita isolante para isolar a fase do neutro ou uma fase de outra fase.**

Para facilitar a ligação da cafeteira e do ventilador, utilize plug's fêmea iguais ao da imagem abaixo. Basta desparafusar o plug e conectar os cabo(de 2,5mm)s nos contatos e marcar o que será fase e o que será neutro/fase em seguida poderá conectar a tomada de cada um dos equipamentos nos plug's:



**IMPORTANTE: UTILIZE UMA CHAVE DE TESTE PARA IDENTIFICAR A FASE E O NEUTRO NA REDE 127V ALTERNADA EM QUE ESTIVER EXECUTANDO A MONTAGEM DOS EQUIPAMENTOS. NO CASO DAS LIGAÇÕES ESTAREM SENDO FEITAS NUMA REDE 220V VOCÊ NÃO PRECISA IDENTIFICAR FASES, BASTANDO EFETUAR AS LIGAÇÕES CONFORME DESCRIÇÃO ABAIXO, SUBSTITUINDO O NEUTRO PELA FASE. CHAVES DE TESTE PODEM SER ADQUIRIDAS EM LOJAS DE ELÉTRICA OU ELETRÔNICA.**

#### **Resumo das ligações:**

**RELÉ:** pino VCC ligado aos 5V do Arduino, pino GND ligado ao GND do Arduino, pino IN1 ligado na porta digital 6 do Arduino e pino IN2 ligado na porta digital 7 do Arduino. Contato central do CANAL 1(K1) do relé é ligado na fase da rede 127V ou 220V, contato NO(ou normalmente aberto) é ligado numa das pontas da tomada do ventilador. Contato central do CANAL 2(K2) do relé é ligado na fase da rede 127V ou 220V, contato NO(ou normalmente aberto) é ligado numa das pontas da tomada da cafeteira.

**VENTILADOR:** uma das pontas da tomada já está ligada ao retorno que vem do relé no CANAL 1(K1). A outra ponta da tomada deve ser ligada no neutro/fase da rede 127V / 220V.

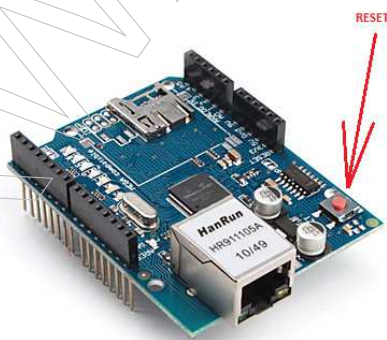
**CAFETEIRA:** uma das pontas da tomada já está ligada ao retorno que vem do relé no CANAL 2(K2). A outra ponta da tomada deve ser ligada no neutro/fase da rede 127V / 220V.

Após efetuar a montagem de acordo com a imagem acima, revise todo o circuito para eliminar qualquer possibilidade de ligações erradas. Repito: preocupe-se em **NÃO** ligar **FASE** com **NEUTRO**. Se possuir um multímetro, utilize o teste de continuidade para verificar se a **FASE** e o **NEUTRO** não estão interligados.

Após as ligações, seu arduino já estará pronto para comunicar com a aplicação android. Ligue novamente o cabo USB no Arduino e no PC para alimentá-lo.

**OBS: SE COMEÇAR A SENTIR CHEIRO DE QUEIMADO DESLIGUE O CIRCUITO IMEDIATAMENTE E REVISE AS LIGAÇÕES.**

**OBS: PRESSIONE O BOTÃO DE RESET DO ETHERNET SHIELD POR 3 SEGUNDOS E SOLTE.**





Dentro do DVD entre na pasta “Android e Arduino”, em seguida entre na pasta “PASSO a PASSO - Controle de Ventilador e Cafeteira” e depois “Instalador - App Controle de Ventilador e Cafeteira”. Copie o APK que está lá e coloque no seu dispositivo Android. Instale a aplicação e execute a mesma. Você verá uma tela semelhante com a da imagem abaixo:



Insira o IP (byte ip[ ]) que você atribuiu ao Ethernet Shield, em seguida clique em conectar.



OBS: SE O IP DIGITADO FOR DIFERENTE DO IP ATRIBUÍDO AO byte ip[ ], A APLICAÇÃO NÃO IRÁ FUNCIONAR E PODERÁ FICAR TRAVADA.



VENTILADOR



CAFETEIRA



Copyright© MasterWalker Shop

STATUS  
VENTILADOR: LIGADO  
CAFETEIRA DESLIGADA

Copyright© MasterWalker Shop

Agora basta pressionar os botões da tela da aplicação para controlar o ventilador e a cafeteira. Veja que o status de cada um dos equipamentos é mostrado na tela da aplicação Android.

### ALGUMAS CONSIDERAÇÕES:

- \* Esta aplicação só funciona dentro de uma determinada rede WiFi (SEM FIO), portanto se você pensa em ativar o 3G do seu smartphone e enviar os comandos, já te adianto que não será possível a não ser que você pesquise sobre atribuição de IP fixo para acesso de qualquer lugar do mundo ou fazer [NAT](#) no seu roteador.*
- \* A aplicação pode sofrer variações quando executada em tablets, e smartphones de baixa resolução de tela.*
- \* Sintam-se a vontade para explorar ao máximo da aplicação e modificá-la o quanto quiser.*
- \* Esse é um exemplo simples e prático, e que não costuma ser encontrado em qualquer lugar.*

O código fonte da aplicação Android segue em anexo com o restante do material. Caso se interesse ou já tenha conhecimento de programação para android, basta importar a aplicação para o eclipse e fazer as modificações que julgar necessário.