

TUTORIAL: SENSOR DE UMIDADE E TEMPERATURA DHT11 - UTILIZANDO ANDROID E ARDUINO

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 um Sensor de Umidade e Temperatura ligado ao Arduino, onde o mesmo vai apresentar na tela da aplicação a umidade e temperatura ambiente que forem medidas pelo sensor conectado ao Arduino.

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/nWjFxd>

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.

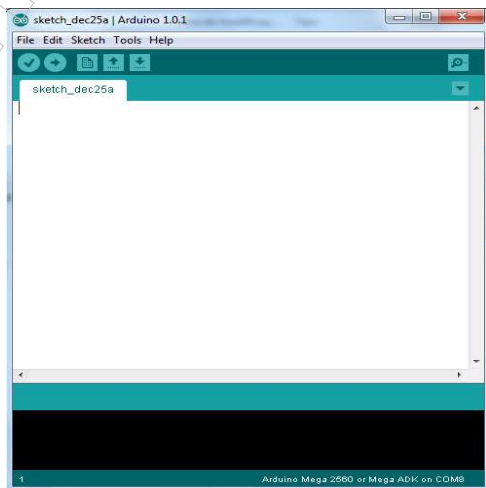
Inicialmente você precisa conectar o Ethernet Shield ao Arduino. Veja na imagem abaixo como eles ficam quando conectados.



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 .

OBS: Para continuar a executar os passos abaixo é necessário que você tenha efetuado os passos do arquivo DVD > Android e Arduino > LEIA-ME ANTES DE COMEÇAR - PASSO a PASSO - Importando Bibliotecas.

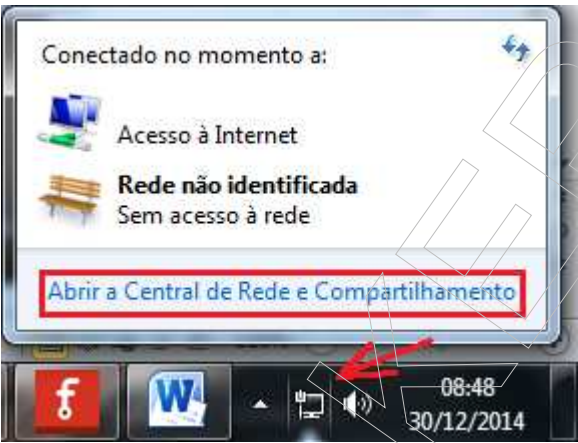
Com seu arduino instalado e acoplado ao ethernet shield, execute a IDE de desenvolvimento (**arduino.exe**).



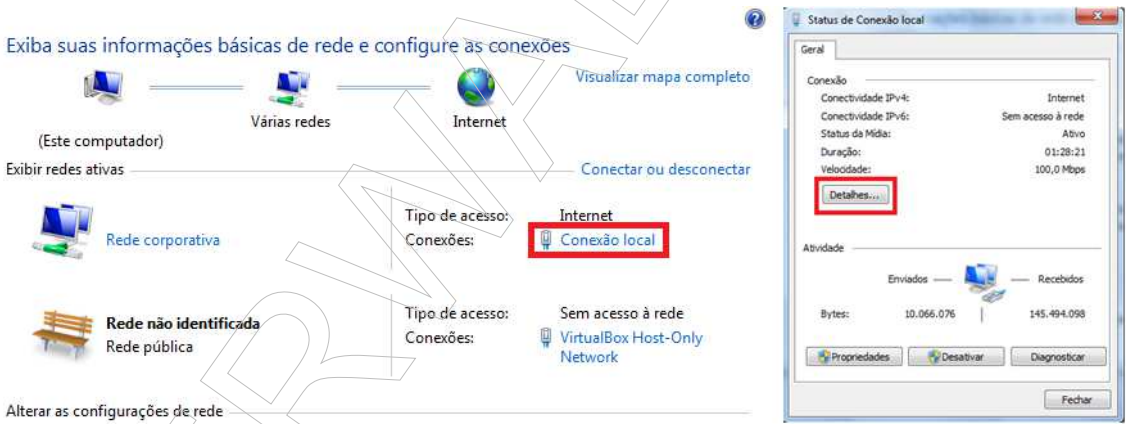
Em seguida clique no menu File>Open, e vá até a pasta “Android e Arduino”, em seguida entre na pasta “PASSO a PASSO - Sensor de Umidade e Temperatura DHT11” e depois “Exemplo_EthernetShield_Umidade_Temperatura”. Dentro dessa pasta tem um arquivo que é reconhecido pela IDE do Arduino, dê um duplo clique 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.

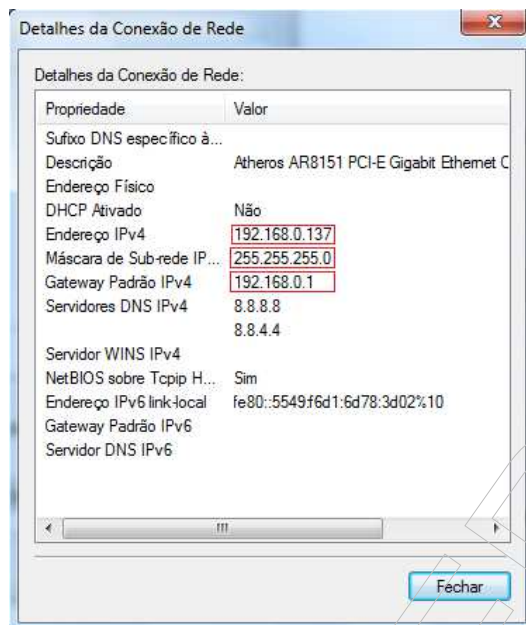
Primeiramente vá até a barra de tarefas do Windows e clique sobre a sua conexão de internet e em seguida em “Abrir a Central de Rede e compartilhamento”.



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



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



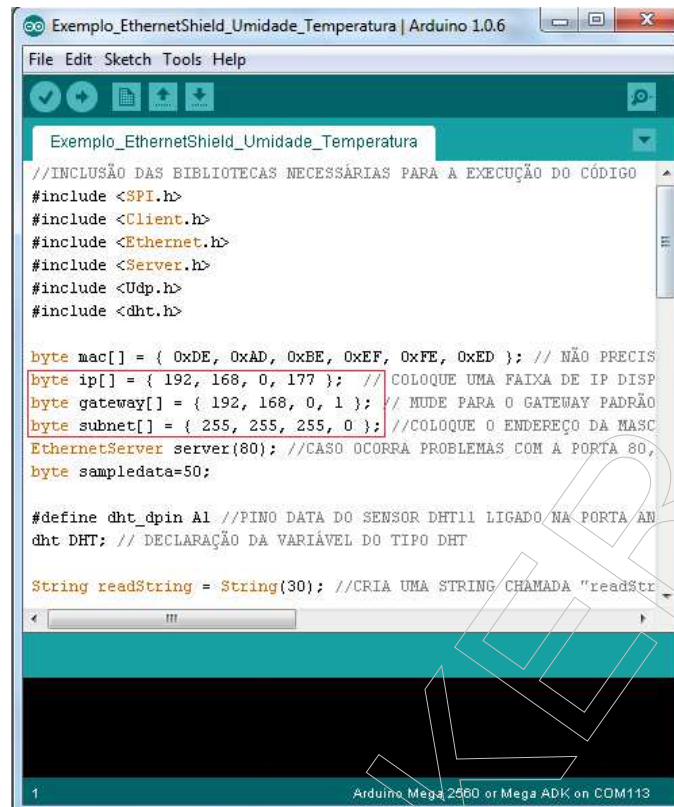
Veja que na imagem estão destacados as informações que são necessárias para configuração do Ethernet Shield. Vale ressaltar que seu IP pode possuir um "1" ou "2" no lugar do "0" (Exemplo: 192.168.2.200 / 192.168.1.150). **Pode ocorrer de a rede possuir faixa de IP 10.2.x.xxx(Exemplo: 10.2.2.102 / 10.2.1.200).** 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.



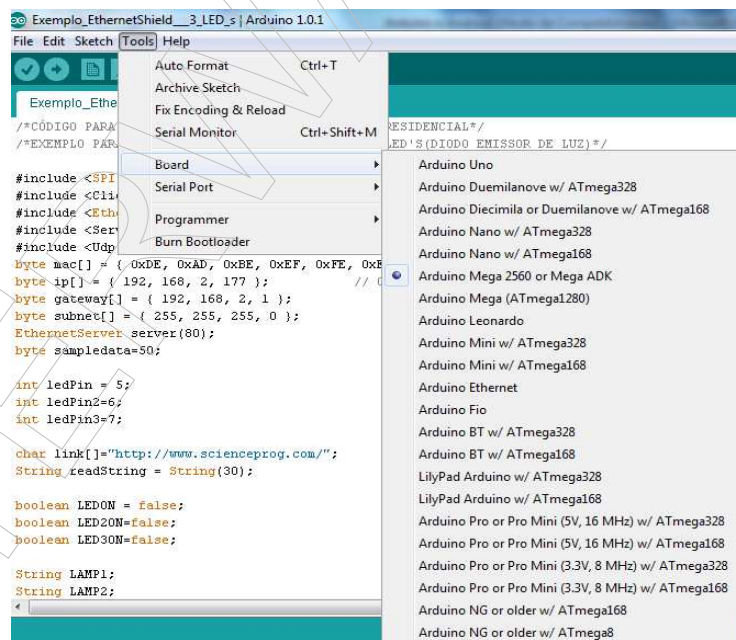
```
//INCLUSÃO DAS BIBLIOTECAS NECESSÁRIAS PARA A EXECUÇÃO DO CÓDIGO
#include <SPI.h>
#include <Client.h>
#include <Ethernet.h>
#include <Server.h>
#include <Udp.h>
#include <dht.h>

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; // NÃO PRECIS
byte ip[] = { 192, 168, 0, 177 }; // COLOQUE UMA FAIXA DE IP DISP
byte gateway[] = { 192, 168, 0, 1 }; // MUDE PARA O GATEWAY PADRÃO
byte subnet[] = { 255, 255, 255, 0 }; //COLOQUE O ENDEREÇO DA MASC
EthernetServer server(80); //CASO OCORRA PROBLEMAS COM A PORTA 80,
byte sampledata=50;

#define dht_dpin A1 //PINO DATA DO SENSOR DHT11 LIGADO NA PORTA AN
dht DHT; // DECLARAÇÃO DA VARIÁVEL DO TIPO DHT

String readString = String(30); //CRIA UMA STRING CHAMADA "readStr
1 Arduino Mega 2560 or Mega ADK on COM113
```

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**.



```
Exemplo_EthernetShield_3_LED_s | Arduino 1.0.1
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Board
Serial Port
Programmer
Burn Bootloader
/*CÓDIGO PARA
/*EXEMPLO PARA
#include <SPI.h>
#include <Client.h>
#include <Ethernet.h>
#include <Server.h>
#include <Udp.h>
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
byte ip[] = { 192, 168, 2, 177 };
byte gateway[] = { 192, 168, 2, 1 };
byte subnet[] = { 255, 255, 255, 0 };
EthernetServer server(80);
byte sampledata=50;

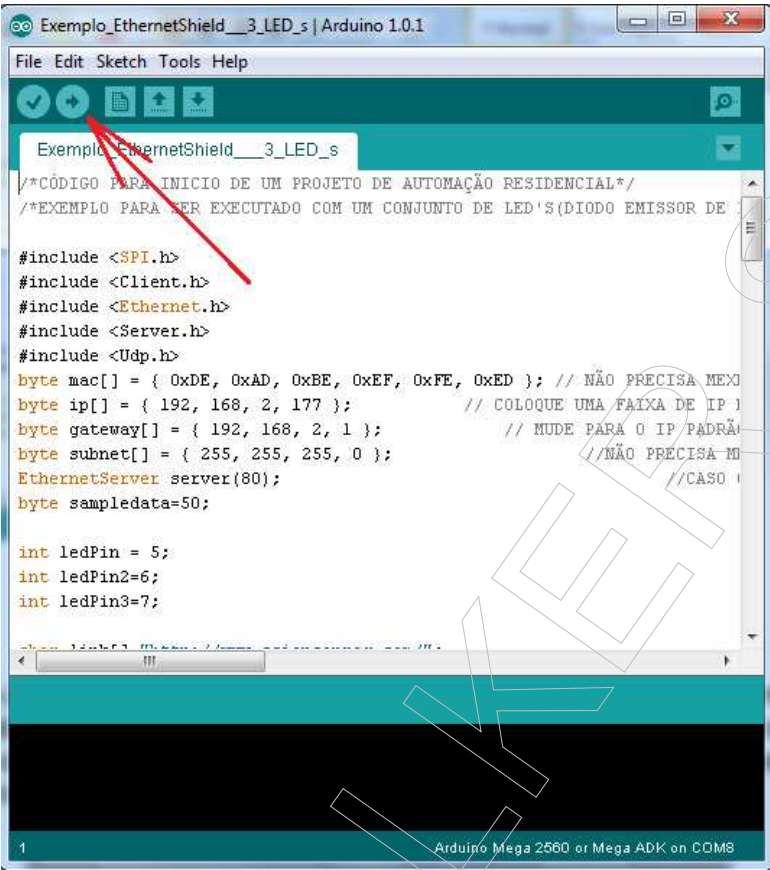
int ledPin = 5;
int ledPin2=6;
int ledPin3=7;

char link[]="http://www.scienceprog.com/";
String readString = String(30);

boolean LEDON = false;
boolean LED2ON=false;
boolean LED3ON=false;

String LAMP1;
String LAMP2;
```


Agora carregue o programa no arduino.



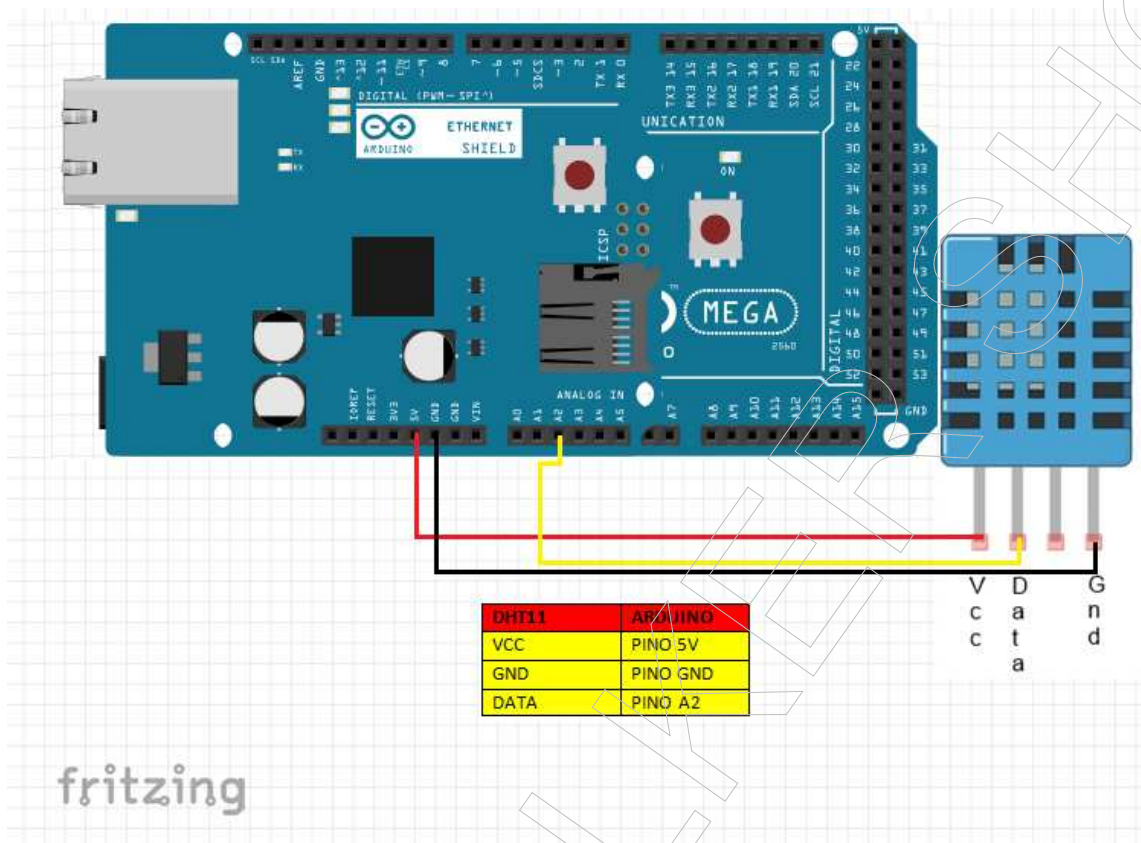
Feche a IDE de desenvolvimento.

Agora você terá que fazer a ligação do Sensor de Umidade e Temperatura no Arduino.

Material necessário:

- Arduino Mega 2560 R3
- Ethernet Shield
- Protoboard
- Sensor de Umidade e Temperatura DHT11
- Cabos para conexão

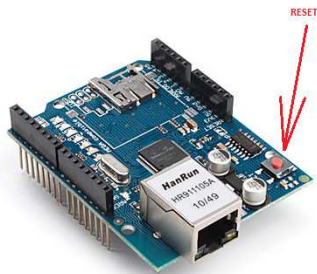
Ligue de acordo com a imagem abaixo:



Note que foi utilizado o pino analógico A2 do Arduino ao invés do A0 ou A1. Quando se utiliza o Ethernet Shield W5100 acoplado ao Arduino, o Ethernet utiliza os pinos analógicos A0 e A1 para controle do SDCard que pode ser conectado ao Ethernet. Portanto, ao utilizar o Ethernet Shield W5100, só trabalhe com os pinos analógicos a partir do A2 em diante.

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: 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 - Sensor de Umidade e Temperatura DHT11” e depois “Instalador - App Umidade e Temperatura”. 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:



carregando... aguarde um momento.

By MasterWalker Shop
Copyright 2014

Copyright© MasterWalker Shop

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.



UMIDADE
47.00%

TEMPERATURA
29.00°

Copyright© MasterWalker Shop

Veja que a aplicação já exibe a umidade e temperatura atual do ambiente.

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.