

TUTORIAL

Arduino Nano

Conceito:

O **Arduino Nano** é uma plataforma de prototipagem eletrônica de *hardware* livre e de placa única, projetada com um microcontrolador Atmel AVR com circuitos de entrada/saída embutido que podem ser facilmente conectados à um computador e programada via IDE (*Integrated Development Environment*, ou Ambiente de Desenvolvimento Integrado) utilizando uma linguagem baseada em C/C++, sem a necessidade de equipamentos extras além de um cabo mini-USB.

Aplicação:

O objetivo do **Arduino Nano** é criar ferramentas que são acessíveis, com baixo custo, flexíveis, funcionais e fáceis de programar, sendo dessa forma acessível a estudantes, projetistas, artistas e amadores. Além disso, foi adotado o conceito de *hardware* livre, o que significa que qualquer um pode montar, modificar, melhorar e personalizar o Arduino, partindo do mesmo *hardware* básico.

Depois de programado o Arduino ele pode ser usado de forma independente, ou seja, você pode colocá-lo em processos industriais como *M2M (Machine to Machine)*, automação e controle de aparelhos, em monitoramento de sensores, etc.

Protocolo:

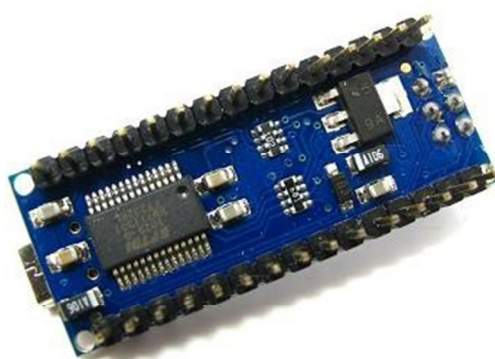
O **Arduino Nano** foi construído originalmente para se comunicar de forma serial através de seus pinos digitais D0 (RX) e D1 (TX), que ao serem conectados a um computador através de um cabo mini-USB, permite a transferência de dados pela chamada interface UART (TTL), mesmo enquanto trabalha em outras tarefas, desde que haja espaço disponível no buffer serial de 64 byte. Assim, no Arduino existe uma parte do hardware específica para a conversão Serial/USB. Disso resulta a criação de uma porta COM virtual no computador para a comunicação com o Arduino.

Arduino Nano:

O Arduino Nano V3.0 é uma placa de microcontrolador baseada no ATmega328. Ele possui 14 pinos de entradas/saídas digitais, 8 entradas analógicas, 1 UARTs (porta serial de *hardware*), um oscilador de cristal de 16 MHz, uma conexão mini-USB, uma entrada de alimentação, uma conexão ICSP e um botão de reset.



Vista de Frente



Vista do Verso

TUTORIAL Arduino Nano

Informações úteis sobre o Arduino Nano:

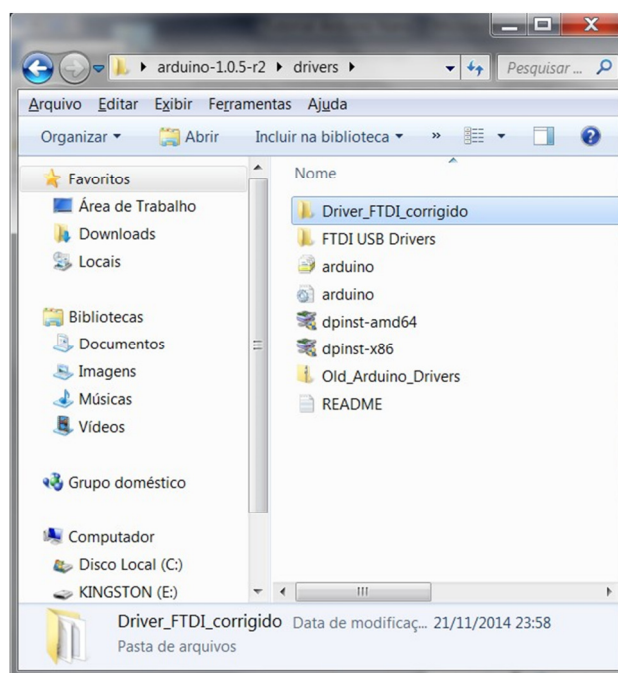
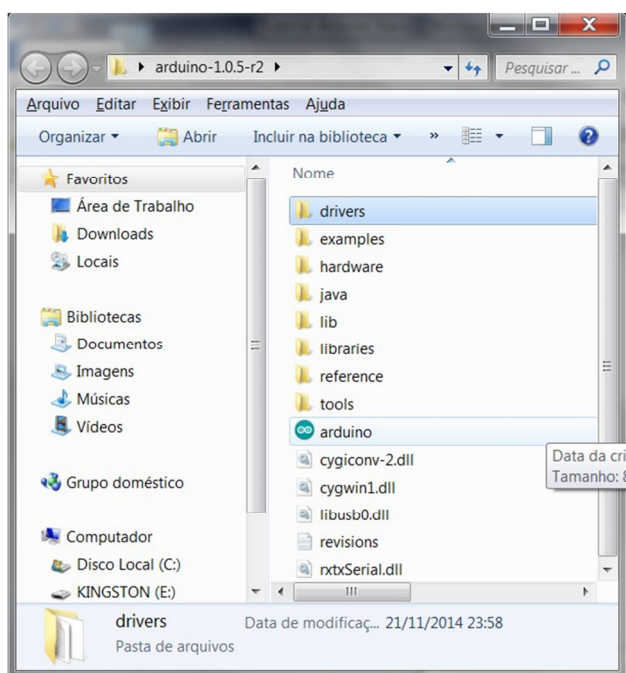
- Tamanho: 17,78mm x 43,18mcm;
- Microcontrolador: ATmega328;
- Tensão de operação: 5V;
- Tensão de entrada (recomendada): 7-12V;
- Tensão de entrada (limites): 6-20V;
- Pinos de entrada/saída (I/O) digitais: 14 (dos quais 6 podem ser saídas como PWM);
- Pinos de entrada analógicas: 8;
- Corrente DC por pino I/O: 40mA;
- Corrente DC para pino 3,3V: 50mA;
- Memória Flash: 32KB (dos quais, 2KB são usados pelo *bootloader*);
- SRAM: 2KB;
- EEPROM: 1KB;
- Velocidade de Clock: 16MHz.

Passo a passo para instalação do Arduino Nano no Computador (via Windows):

O **Arduino Nano** foi construído originalmente para se comunicar de forma serial através de seus pinos digitais D0 (RX) e D1 (TX), que ao serem conectados a um computador através de um cabo mini-USB, permite a transferência de dados pela chamada interface *UART (TTL)*, mesmo enquanto trabalha em outras tarefas, desde que haja espaço disponível no buffer serial de 64 byte. Assim, no Arduino existe uma parte do *hardware* específica para a conversão Serial/USB. Essa conversão é realizada por um chip denominado FTDI.

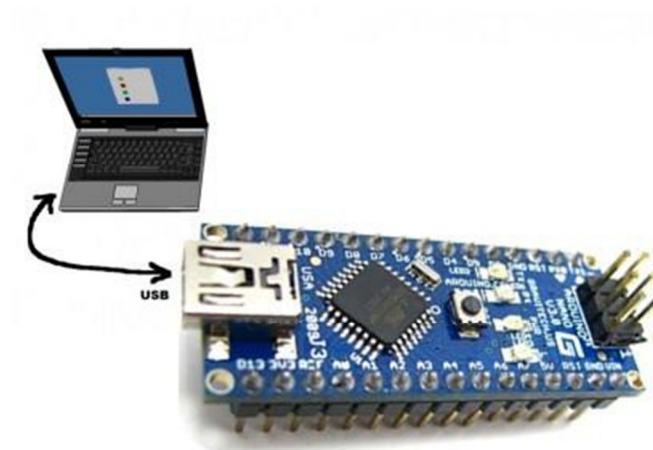
Caso você tenha alguma dificuldade em fazer seu **Arduino Nano** ser reconhecido automaticamente pelo driver já pré-instalado no seu Windows, sugerimos seguir os passos abaixo exatamente na mesma sequência que forem apresentados, pois caso contrário correrá o risco de não funcionar:

1. Baixar o Driver FTDI corrigido na aba DOWNLOAD do link a seguir e descompactá-lo na pasta *Driver* do Arduino: http://www.martaduino.com.br/images/TechDocs/BibliotecasArduino/FTDI_X86_Modificado.rar

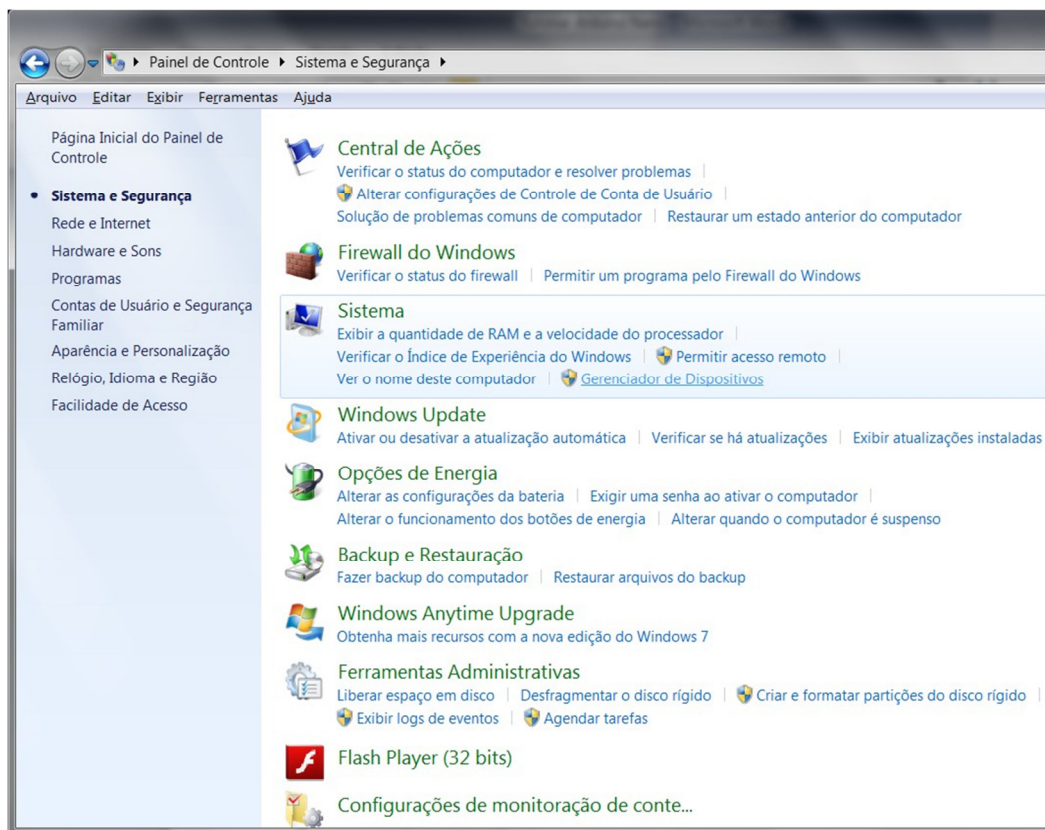


TUTORIAL Arduino Nano

2. Plugar o cabo mini-USB no **Arduino Nano** e conectar no seu computador;



3. Abrir o Gerenciador de Dispositivos. Para isso clicar no botão **Iniciar**, clicar em **Painel de Controle**, clicar em **Sistema e Segurança** e, em **Sistema**, clicar em **Gerenciador de Dispositivos**. É necessário ter permissão de administrador. Se você for solicitado a informar uma senha de administrador ou sua confirmação, digite a senha ou forneça a confirmação.



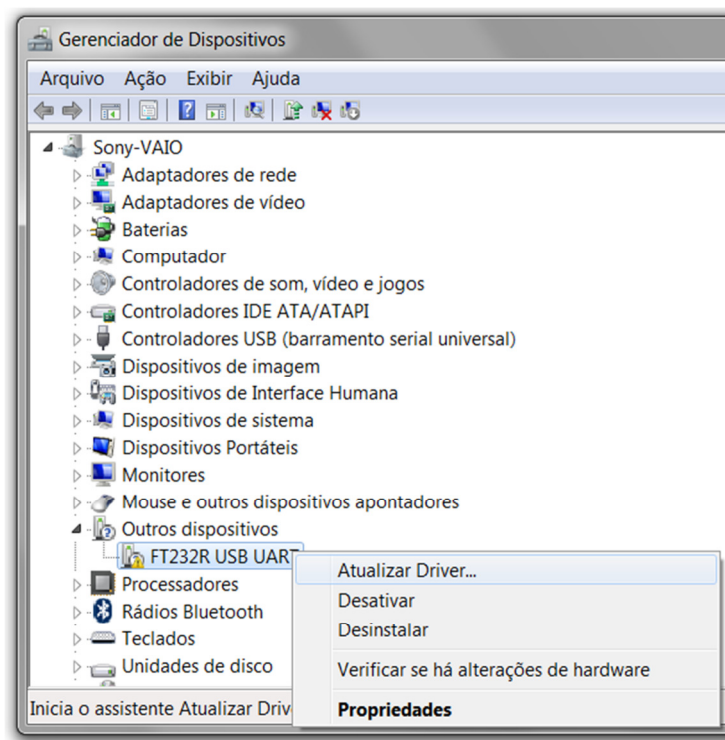
TUTORIAL

Arduino Nano

4. Caso você esteja utilizando o Windows 8 ou mais novo, você vai precisar desabilitar a verificação da assinatura do Driver, antes de atualiza-lo, para isso, siga as instruções deste tutorial complementar:

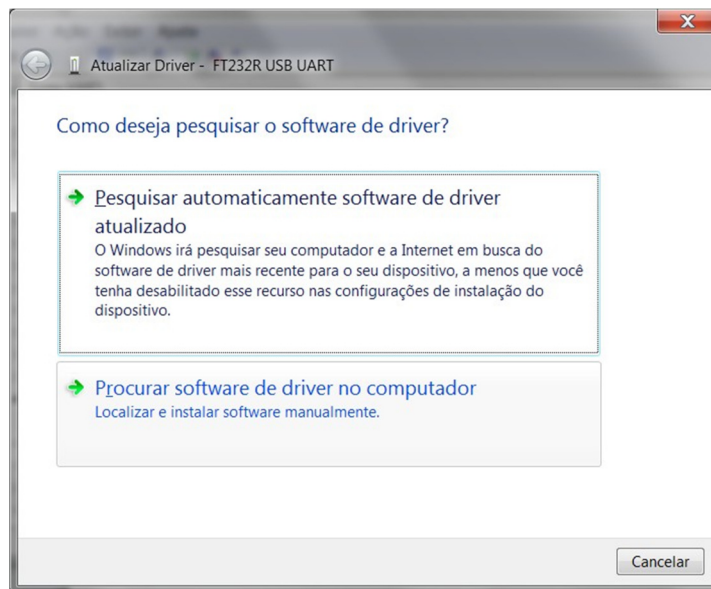
Desabilitando Assinatura de Drivers Windows 8

5. Ao abrir a janela do Gerenciador de Dispositivos, clicar com o botão direito do mouse em **FT232R USB UART**, selecionar "**Atualizar Driver...**";

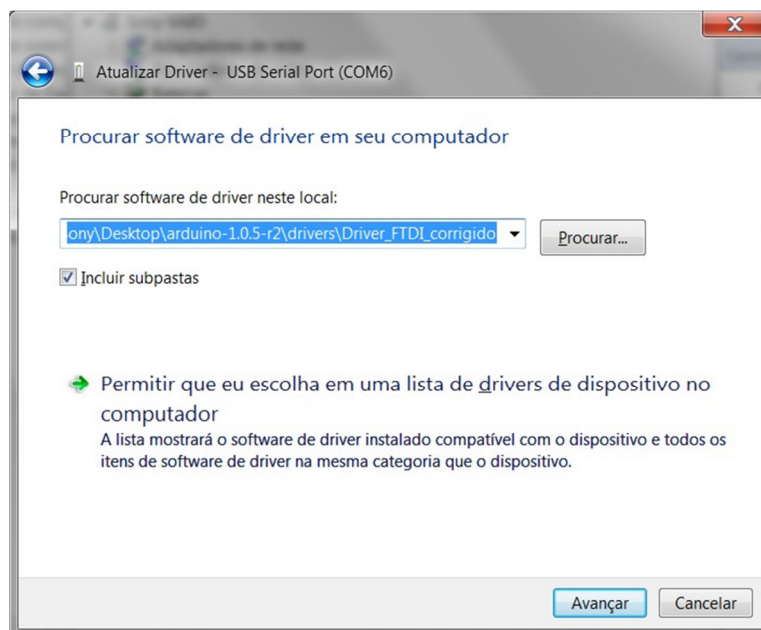


TUTORIAL Arduino Nano

6. Selecionar a opção: **"Procurar software de driver no computador"**;



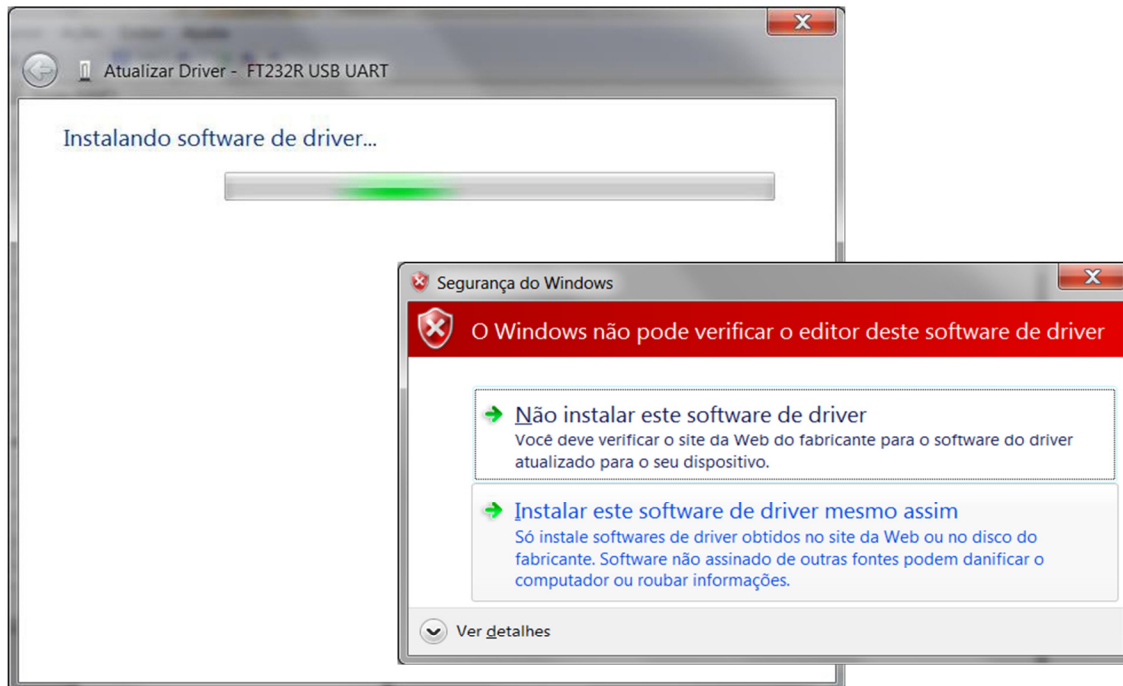
7. Clicar no botão **"Procurar..."** e indicar o caminho onde você descompactou o Driver FTDI corrigido, ou seja, na pasta Driver do seu Arduino;



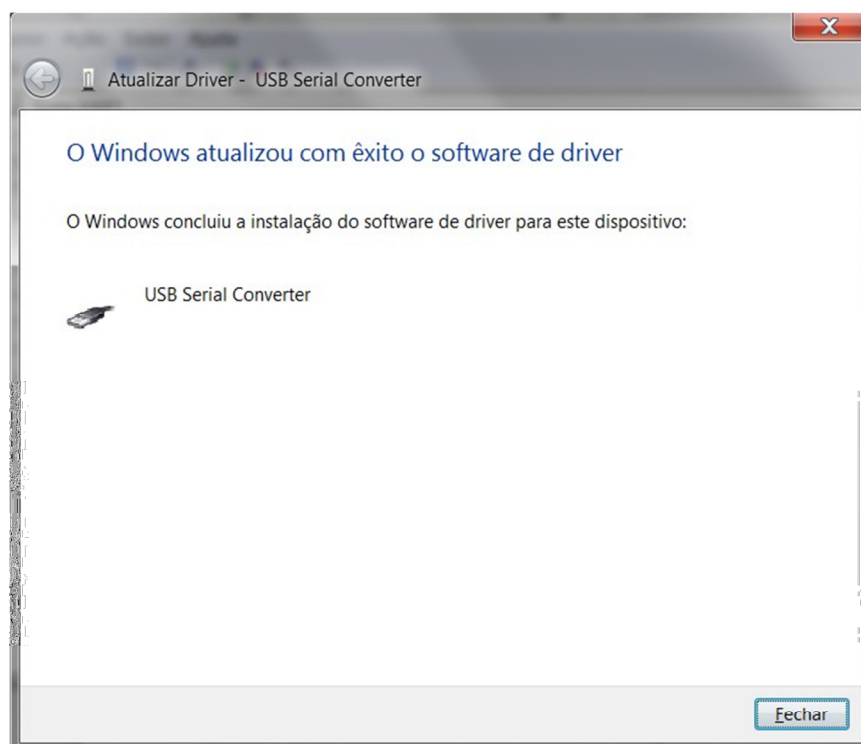
TUTORIAL

Arduino Nano

8. Por fim clicar no botão "**Avançar**" e se o Windows apresentar algum empecilho clicar na opção "**Instalar este software de driver mesmo assim**";



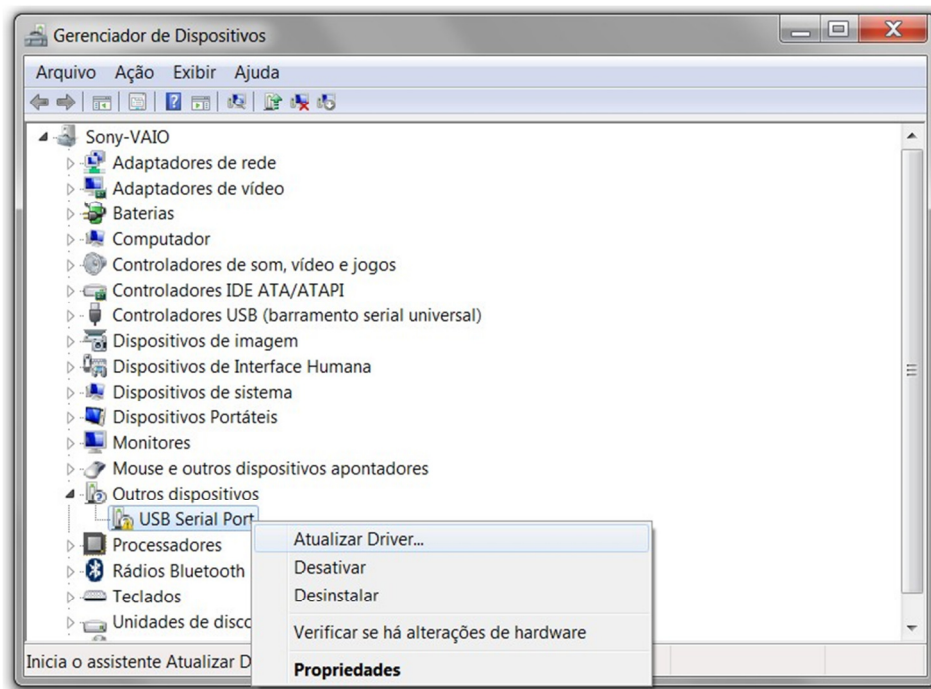
9. Caso tudo ocorra bem será apresentada a tela abaixo:



TUTORIAL

Arduino Nano

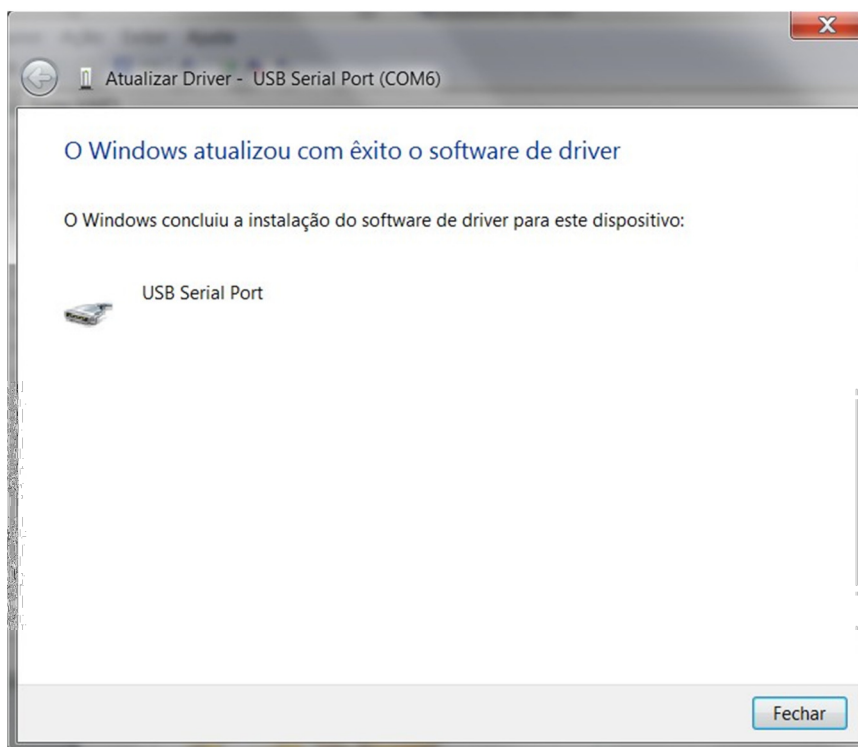
10. Volte a abrir o Gerenciador de Dispositivos para verificar se uma porta COM foi criada para o seu **Arduino Nano**. Caso contrário repita os passos anteriores só que agora clicando com o botão direito do mouse em **USB Serial Port** e selecionar **"Atualizar Driver..."**;



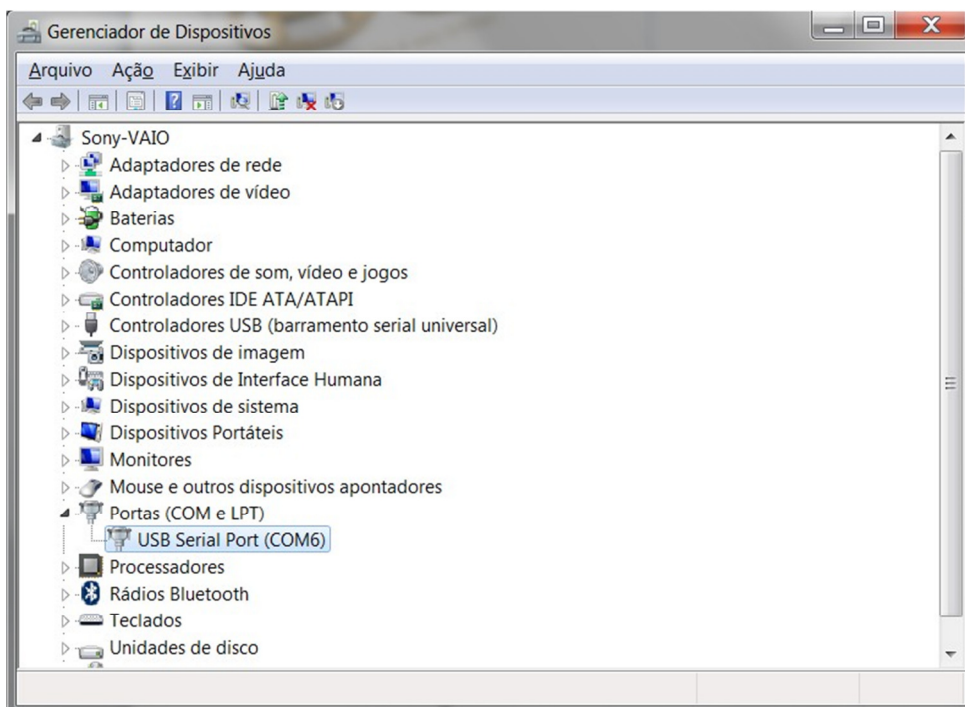
TUTORIAL

Arduino Nano

11. Verifique agora que no título da tela abaixo foi definido, para o caso do nosso exemplo, a porta **COM6** para nosso **Arduino Nano**;



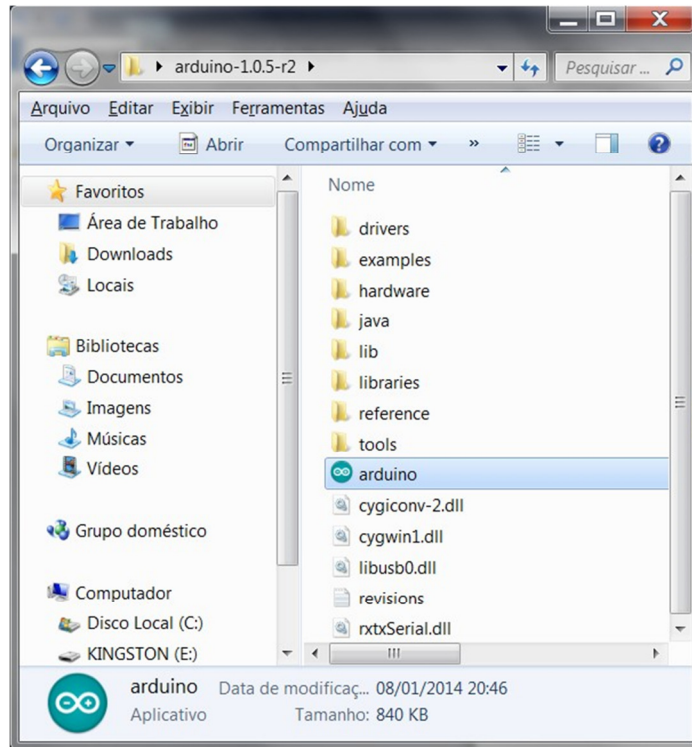
12. Pode fechar a tela anterior e voltar ao Gerenciador de Dispositivos, onde poderá ser confirmada a criação da porta **COM6**. É nessa porta que nosso **Arduino Nano** conseguirá se comunicar com a IDE do Arduino;



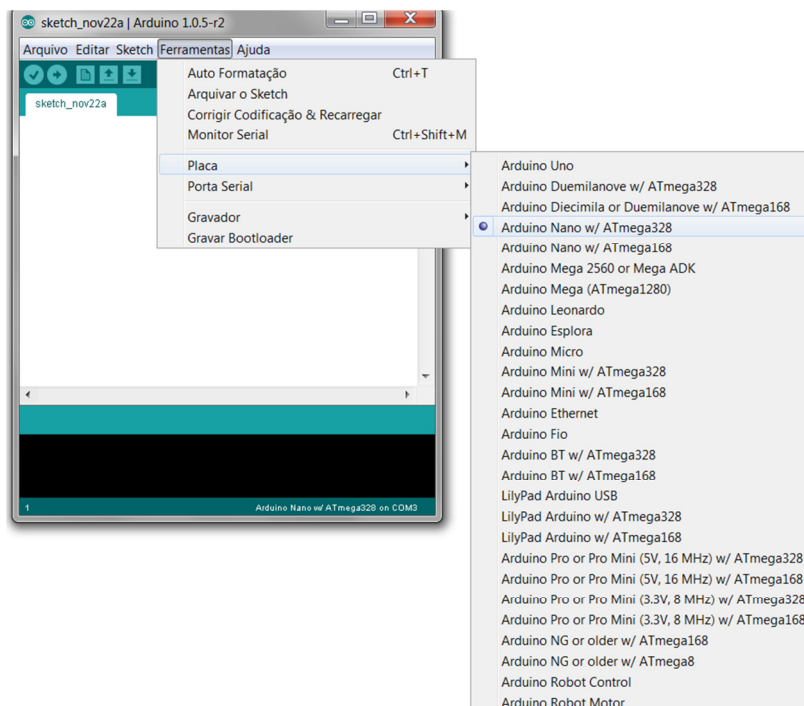
TUTORIAL

Arduino Nano

13. Para testar vamos abrir a IDE (Interface de Desenvolvimento Integrada) do Arduino clicando no aplicativo "arduino.exe";



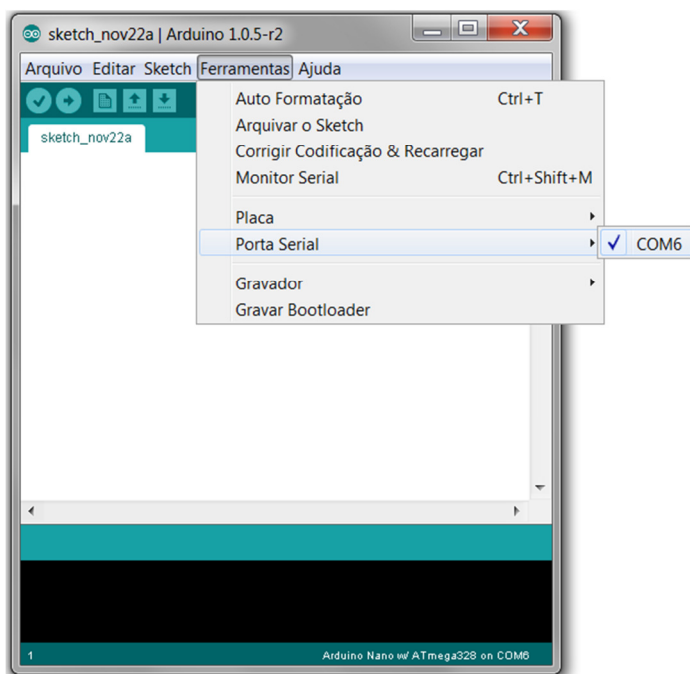
14. No menu da IDE selecione a opção "**Ferramentas (Tools)**", "**Placa (Board)**" e escolha o tipo de Arduino que você está utilizando. No caso desse exemplo estamos utilizando o **Arduino Nano com ATmega328**;



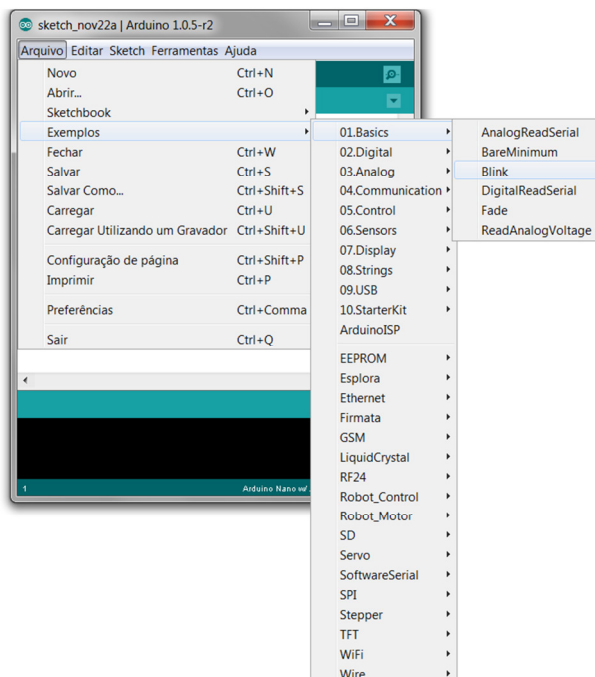
TUTORIAL

Arduino Nano

15. No menu da IDE selecione a opção "**Ferramentas (Tools)**", "**Porta Serial (Serial Port)**" e defina qual porta **COM (Communication)** o Arduino está utilizando. No caso desse exemplo foi definida a **COM6**;



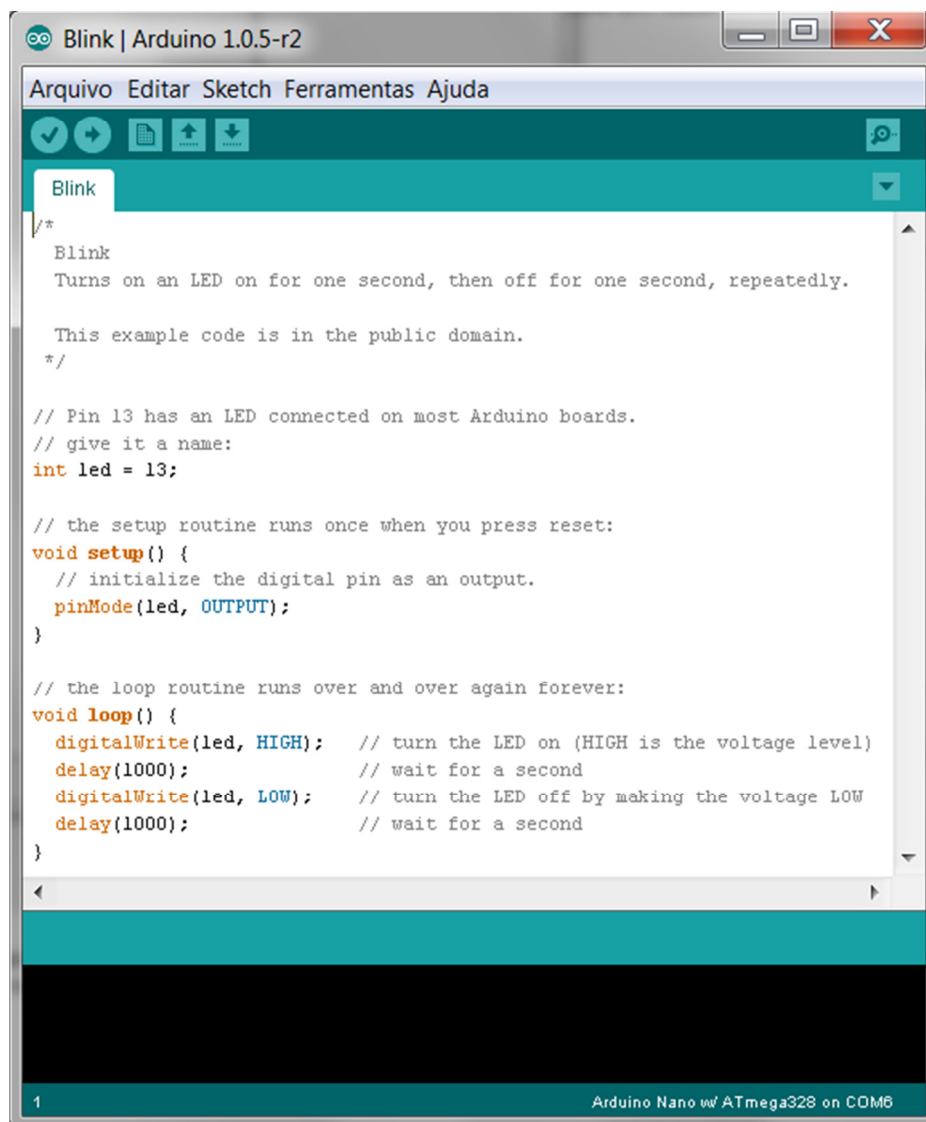
16. Como é só um teste então vamos abrir um dos programas de exemplo já existentes na IDE chamado *Blink*. Deve-se selecionar a opção "**Exemplos (Examples)**" no menu "**Arquivo (Files)**", em seguida selecionar "**Basics**" e clicar no nome do programa "**Blink**";



TUTORIAL

Arduino Nano

17. Após abrir o programa *Blink.ino* deve-se carregá-lo no Arduino, pressionando o botão “**Carregar (Upload)**” logo abaixo da barra de menu. Aguarde até uma mensagem de concluído (*Done*) ser exibida na IDE. Aconselhamos a fazer uma leitura dos comentários inseridos em cada linha do programa, pois isso facilitará o seu entendimento de como o programa funciona.



```
Blink | Arduino 1.0.5-r2
Arquivo  Editar  Sketch  Ferramentas  Ajuda

Blink
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
  */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);  // turn the LED on (HIGH is the voltage level)
  delay(1000);              // wait for a second
  digitalWrite(led, LOW);   // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second
}

1
Arduino Nano w/ ATmega328 on COM6
```

18. Após carregar o programa no **Arduino Nano** o LED conectado de fábrica ao seu pino digital 13 irá piscar, ficando ora aceso por um segundo e ora apagado também por um segundo;
19. A modificação do driver que disponibilizamos em nosso site (http://www.martaduino.com.br/images/TechDocs/BibliotecasArduino/FTDI_X86_Modificado.rar) se faz necessária porque a empresa **FTDI** (*Future Technology Devices International*) responsável pela fabricação do chip de conversão serial assíncrona para USB lançou atualizações de drivers capazes de reconhecer chips FTDIs não fabricados por eles, impossibilitando que várias placas, inclusive alguns **Arduinos Nano**, não sejam reconhecidas pelo sistema operacional do seu computador (PC). E a maneira que a empresa FTDI vem conseguindo fazer isso é modificando alguns códigos

TUTORIAL

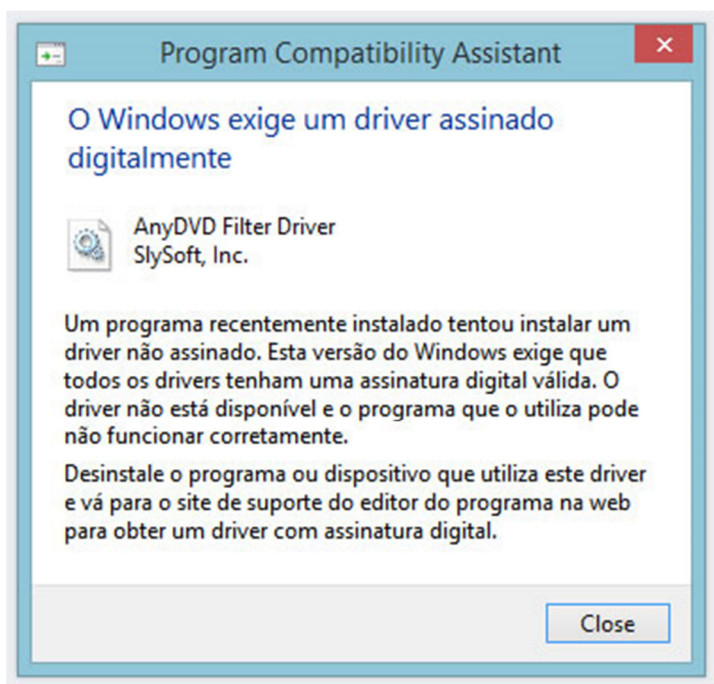
Arduino Nano

que existem na comunicação de um dispositivo USB. São eles: o VID (*vendor id*) e o PID (*product id*). Esses códigos identificam o fabricante e o modelo do dispositivo e são comercializados pelo USB-IF (*USB Implementor Forum*), sendo também usados pelos sistemas operacionais para escolher o driver apropriado. Mas felizmente, existe uma maneira de reprogramar o valor do PID, como você pode acompanhar nos passos abaixo. E foi isso que fizemos no Driver FTDI corrigido que está disponível em nosso site:

- Baixar os drivers FTDI do site oficial: <http://www.ftdichip.com/Drivers/VCP.htm>;
- No caso do Windows, a associação entre o VID/PID e o driver é feita através de um arquivo **.INF**. Então basta abrir esses arquivos **.INF** do arquivo que você baixou no passo anterior e substituir a sequência numérica 6001 por 0000;
- Conectar o seu dispositivo, no nosso caso o **Arduino Nano**, e utilizar para instalação do seu dispositivo esse driver corrigido.

20. Desabilitando Assinatura de Drivers Windows 8

Assim como no Windows Vista e 7 as edições de 64 bits requerem drivers assinados digitalmente. O problema é que muitos dispositivos e periféricos são fornecidos com seus drivers de instalação como webcams, impressoras mais antigas, tablets e canetas para desenhos etc. Ao instalar tal dispositivo no Windows 8.x é mostrada a impossibilidade por estes não serem assinados digitalmente. Hoje, iremos mostrar como instalá-los de qualquer maneira no Windows 8 & Windows 8.1.



Geralmente isso acontece porque a desenvolvedora/fabricante dos drives e periféricos não pagam "Royalties" à Microsoft. Na teoria um driver não assinado não influencia em nada a compatibilidade entre o sistema operacional e periféricos. Já os drivers assinados digitalmente incluem uma impressão digital eletrônica que indica que o driver da empresa foi "produzido/licenciado por", mais ou menos como uma roupa de marca por exemplo.

Como desabilitar a verificação de assinatura de driver do Windows 8.x 64 bits

Se preferir pressione a combinação de teclas **Win + C** para abrir a "**Charms Bar**", em seguida, clique no botão **Configurações**.

TUTORIAL

Arduino Nano



Quando o Painel de Controle é aberto, mude para a seção "Atualização e recuperação" e em seguida, clique na opção de recuperação, ao lado esquerdo. Uma vez selecionado, você verá a seção de **Inicialização avançada** aparecer no lado direito. Então clique sobre o botão "**Reiniciar agora**".

Inicialização avançada

Inicialize de um dispositivo ou disco (como unidade USB ou DVD), mude as configurações de inicialização do Windows ou restaure o Windows a partir de uma imagem do sistema. O computador será reiniciado.



Quando o computador for reiniciado selecione a opção "Solução de Problemas".



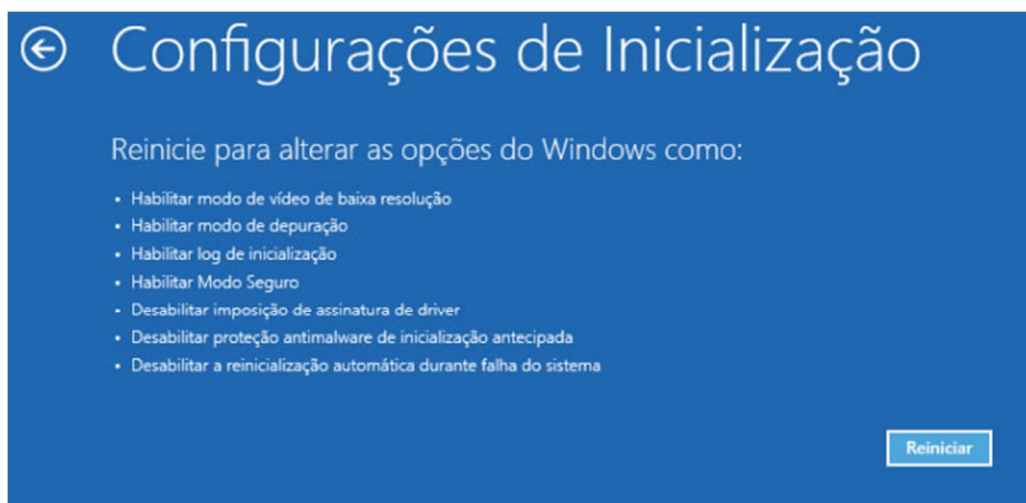
Em seguida, vá para as opções avançadas escolhendo a opção "**Configurações de inicialização**".

TUTORIAL

Arduino Nano



Por estarmos modificando as definições de configuração em tempo de boot, você terá que reiniciar o computador pela última vez.



*Finalmente, será dada uma lista de configurações de inicialização que podemos alterar. O que estamos procurando é "**Desabilitar imposição da assinatura de driver**". – cada opção do menu é uma tecla "**F**" diferente, para escolher a configuração que desabilita assinaturas de drivers no Windows 8 & 8.1 você terá que pressionar a tecla **F7**.*

TUTORIAL

Arduino Nano

Configurações de Inicialização

Pressione um número para escolher as opções abaixo:

Use teclas numéricas ou de função F1-F9.

- 1) Habilitar depuração
- 2) Habilitar log de inicialização
- 3) Habilitar vídeo de baixa resolução
- 4) Habilitar Modo de Segurança
- 5) Habilitar Modo de Segurança com Rede
- 6) Habilitar Modo de Segurança com Prompt de Comando
- 7) Desabilitar Imposição de Assinatura de Driver
- 8) Desabilitar proteção antimalware de início antecipado
- 9) Desabilitar reinício automático após falha

Pressione F10 para obter mais opções

Pressione Enter para retornar ao seu sistema operacional

Seu PC irá reiniciar pela última vez e você será capaz de instalar drivers não assinados de impressoras e outros periféricos sem qualquer mensagem de erro no Windows 8.x de 64bits.

Links úteis sobre o Arduino Nano:

- <http://arduino.cc/en/Main/ArduinoBoardNano>