

Arduino e Proteus (Tutorial)

Por Felipe Gomes Flores



Para quem trabalha com projetos ou apenas os faz por hobby sabe que há situações em uma boa idéia surge do nada e às vezes não há como reproduzi-la no momento, muitas vezes por não ter o equipamento (hardware) necessário para a montagem de um certo circuito e aí acabam não conseguindo visualizar de alguma forma se a tal idéia realmente funciona ou não.

Bom, para aqueles que querem desenvolver projetos com o Arduino sem ter a necessidade de montar circuitos físicos e/ou gravar diversas vezes um sketch no chip (coisas que levam um certo tempo) existe uma maneira de simular os circuitos virtualmente. Neste tutorial veremos como fazê-lo utilizando um software bastante conhecido e poderoso em simulações de circuitos eletrônicos: o Proteus.

Como isso funciona?

Algumas pessoas perguntam coisas como: “O Proteus lê Arduino?” ou “Tenho que instalar algum plug-in ou extensão do Arduino para o Proteus?”. A verdade que é as coisas são muito mais simples do que parecem, e a resposta para as duas perguntas acima é “NÃO!”.

O Proteus é um software muito poderoso capaz de simular alguns tipos de microcontroladores como PIC e 8051 e , assim como esses dois primeiros, o Arduino nada mais é que uma placa com um microcontrolador chamado AVR (ATmega8, ATmega168, ATmega328 etc). Portanto iremos simular um circuito usando um microcontrolador AVR, porém vamos utilizar a IDE do Arduino para criar o programa. Para o Proteus não interessa qual é a linguagem que você está programando (se é de alto ou baixo nível), mas sim o arquivo em hexadecimal que é gerado para gravar no chip. Então, é isso o que faremos.

Gerando o Arquivo em Hexadecimal!

A seguir será mostrado um passo-a-passo de como simular o Arduino no Proteus. Se tudo der certo, ao final deste tutorial você terá simulado o programa mais simples em Arduino: piscar um LED.

1º Passo: Abrir o código

Abra a IDE do Arduino (neste caso estou utilizando a versão 0017*), então vá em “File/Examples/Digital/Blink”, como mostra a figura 1.

**Com a IDE 0021 não funcionou*

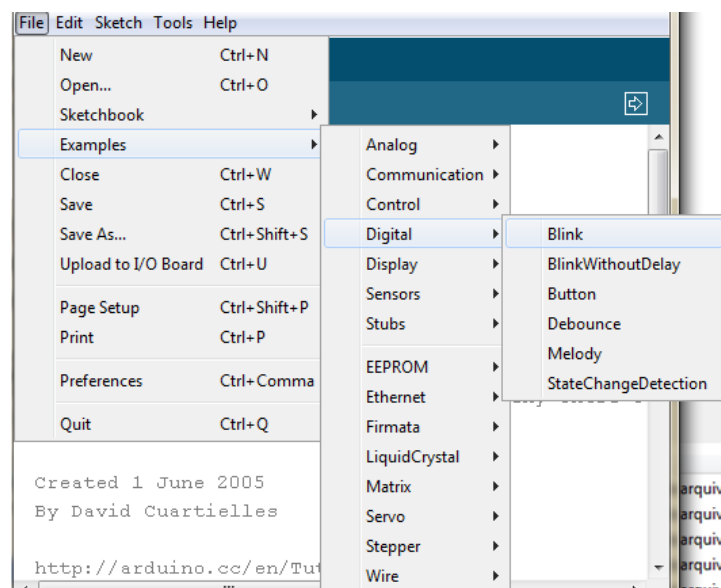


Figura 1 - Abrindo o Sketch “Blink”

O código abaixo então aparecerá na IDE

```
/*  
  Blink  
  
  Turns on an LED on for one second, then off for one second, repeatedly.  
  
  The circuit:  
  * LED connected from digital pin 13 to ground.  
  
  * Note: On most Arduino boards, there is already an LED on the board  
  connected to pin 13, so you don't need any extra components for this example.
```

Created 1 June 2005
By David Cuartielles

<http://arduino.cc/en/Tutorial/Blink>

based on an original by H. Barragan for the Wiring i/o board

```
*/  
  
int ledPin = 13;    // LED connected to digital pin 13  
  
// The setup() method runs once, when the sketch starts  
  
void setup() {  
  // initialize the digital pin as an output:  
  pinMode(ledPin, OUTPUT);  
}  
  
// the loop() method runs over and over again,  
// as long as the Arduino has power  
  
void loop()  
{  
  digitalWrite(ledPin, HIGH);  // set the LED on  
  delay(1000);                 // wait for a second  
  digitalWrite(ledPin, LOW);   // set the LED off  
  delay(1000);                 // wait for a second  
}
```

Aperte **Ctrl+Shift+S** para “Salvar Como”. Isso é feito para você saber onde estará o arquivo gerado. Neste caso o arquivo será salvo com o nome de “PiscaLED” em “Meus Documentos/Arduino” como mostra a figura 2.

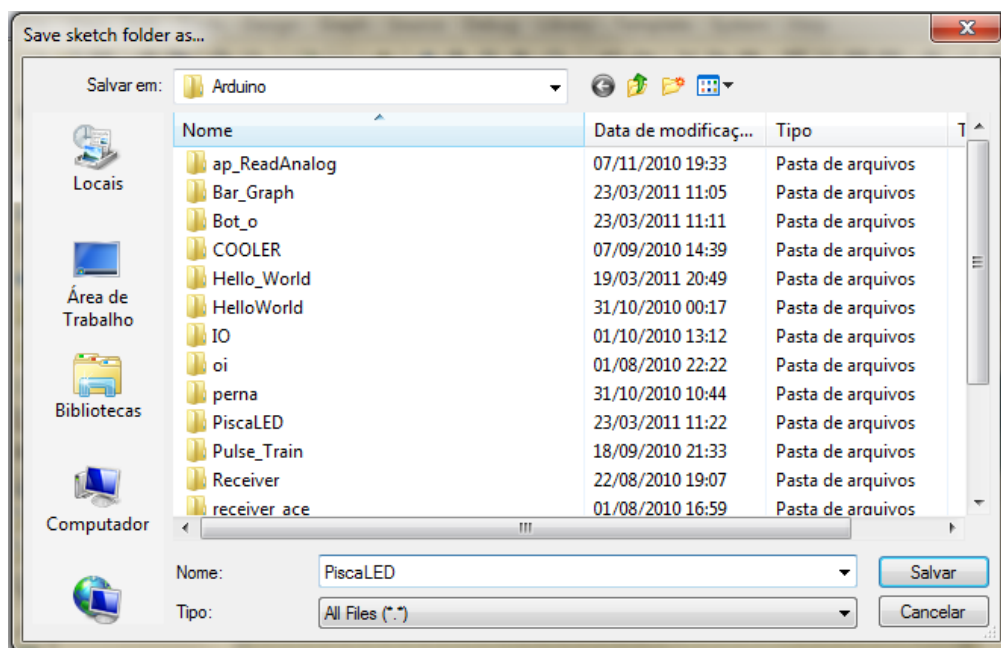


Figura 2 - Salvando o Sketch em um Local Conhecido.

2º Passo: Fazer o Upload do Sketch

Faça como se fosse gravar o programa no Arduino exceto pelo fato que você não irá conectá-lo ao computador: escolha o modelo da placa (Neste caso *Arduino Duemilanove or Nano w/ ATmega328*) e a porta com a qual irá gravar o sketch (Neste caso *COM1*) e então aperte o botão *Upload*.

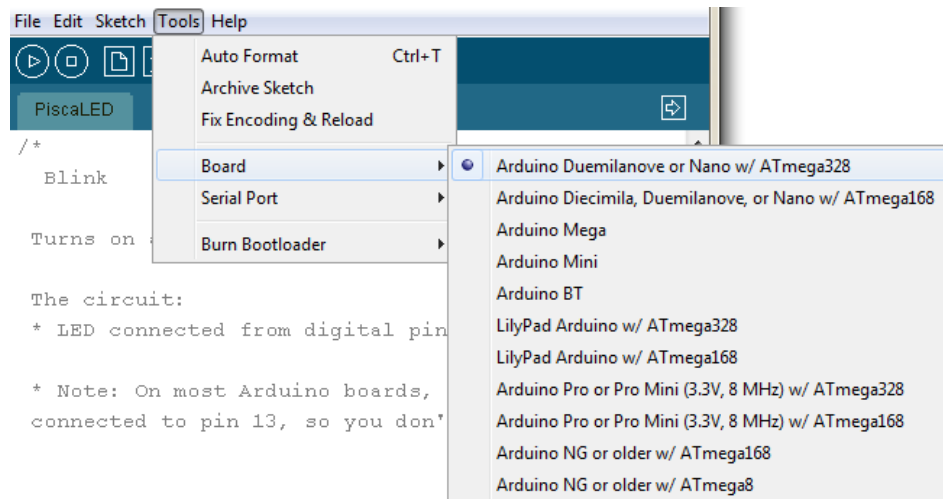


Figura 3 - Escolhendo a placa correta

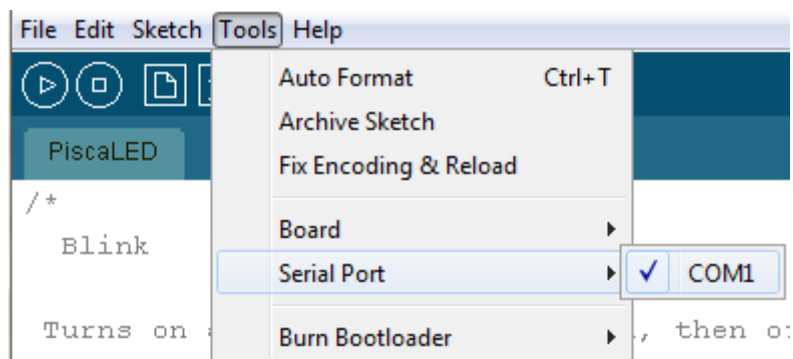


Figura 4 - Escolhendo a porta correta



Figura 5 - Painel de Botões

Alguns segundos depois que o Upload começa ele será interrompido e aparecerá uma mensagem de erro como a da figura 6. Isso obviamente ocorrerá porque não há uma placa conectada ao computador.

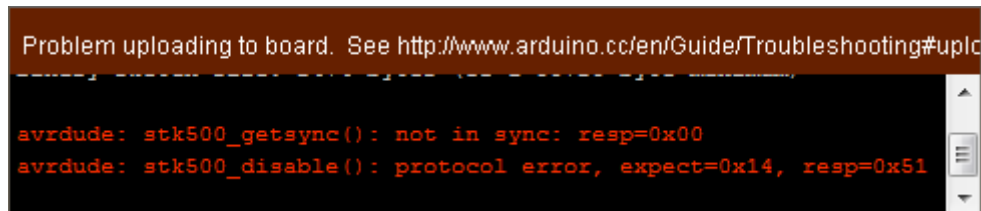


Figura 6 - Mensagem de erro exibida

Não se preocupe com esse erro, pois o que realmente importa é que o arquivo em hexadecimal foi gerado. Dentro da pasta *Meus Documentos/Arduino/PiscaLED* aparecerá uma pasta chamada *applet*. Abra essa pasta e veja se esta contém o arquivo “*PiscaLED.cpp.hex*”.

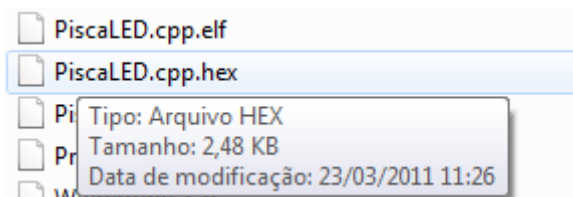


Figura 7 - Visualização do Arduino .hex na pasta

3º Passo: Monte o Circuito no Proteus


Abra o ISIS Proteus . A direita da janela do programa você verá dois botões (“P” e “L”) como o mostrado na figura abaixo. Clique em “P”.



Figura 8 - Opções “Pick from libraries” (Botão “P”) e “Manage Library” (Botão “L”)

Aparecerá então a Janela “Pick Devices”, como mostrado na figura 9.

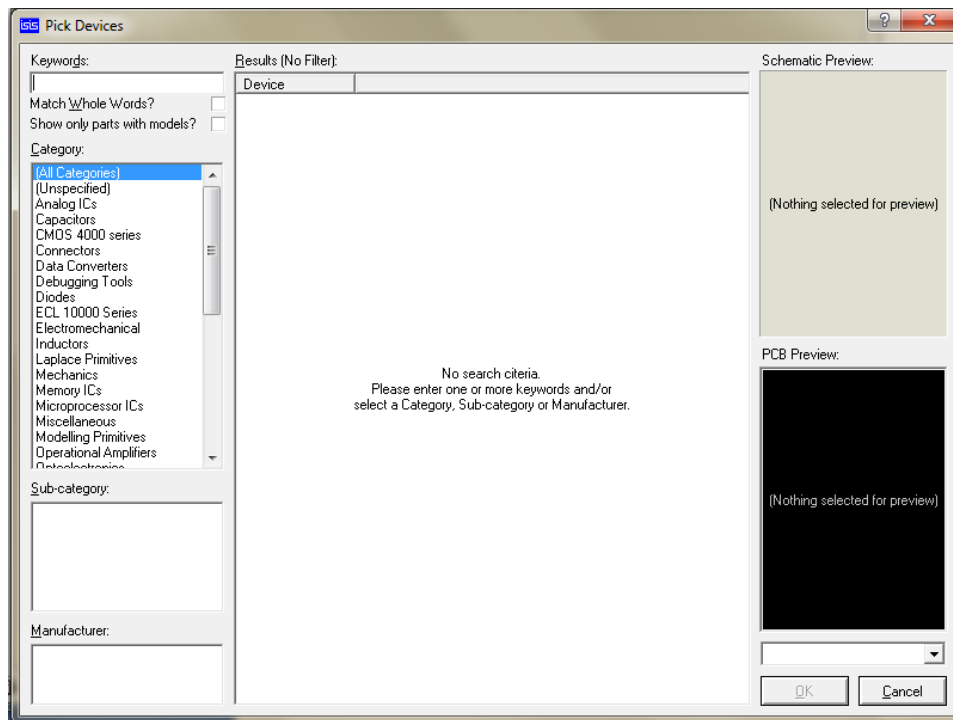


Figura 9 - Janela "Pick Devices" do Proteus

Busque os componentes com as seguintes *Keywords*:

- ATMEGA328P – Microcontrolador do Arduino;
- CRYSTAL – Cristal Oscilador de Quartzo;
- LED-RED - LED Vermelho Animado;
- MONORES22P – capacitor 22 pF;

Na lista de ícones a direita da Janela do Proteus clique em *Terminals Mode*, então clique em GROUND e depois clique dentro da janela de montagem do circuito para inserir o GND.

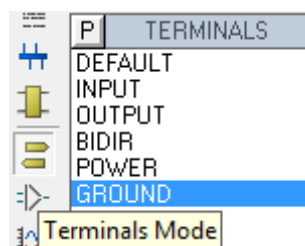


Figura 10 - Terminals Mode

Faça o mesmo para POWER. Para voltar à lista de componentes, clique no botão *Component Mode* que está um pouco acima o botão *Terminal Mode*.

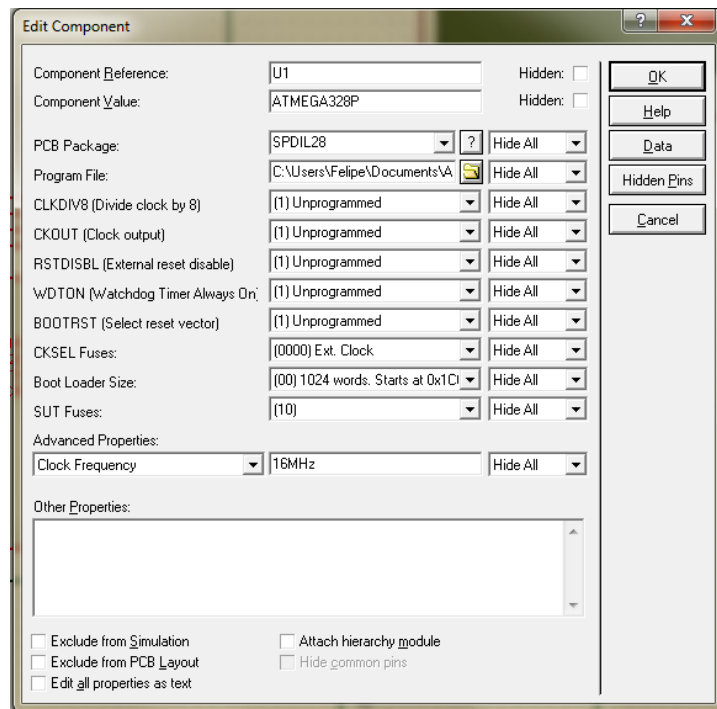


Figura 13 - Configurando o ATmega

No item *Program File* clique na pasta a direita, encontre o arquivo “PiscaLED.cpp.hex” e dê um duplo clique sobre ele. Depois clique OK!

No canto inferior esquerdo da janela do Proteus estão os seguintes botões de simulação. Clique em **Play**.



Figura 14 - Botões de Simulação

O LED começará a piscar!!