Aula 9: Oficina de Programação e Robótica

Bem-vindos a 2º módulo da nossa oficina!

Nesta parte vamos construir alguns projetos com um dispositivo chamado Arduino. Vamos começar com coisas mais básicas, até chegarmos à construção de um robô simples. Vocês verão que, apesar do tamanho, o Arduino nos possibilita criar muitos projetos legais.

O material elaborado para este módulo da nossa oficina se baseia fortemente no livro *Arduino Básico*, de Michael McRoberts (Editora Novatec, 2011), e nos tutoriais do site do Arduino (http://arduino.cc).

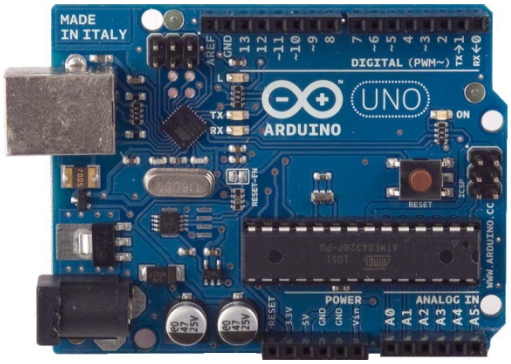
O que é o Arduino

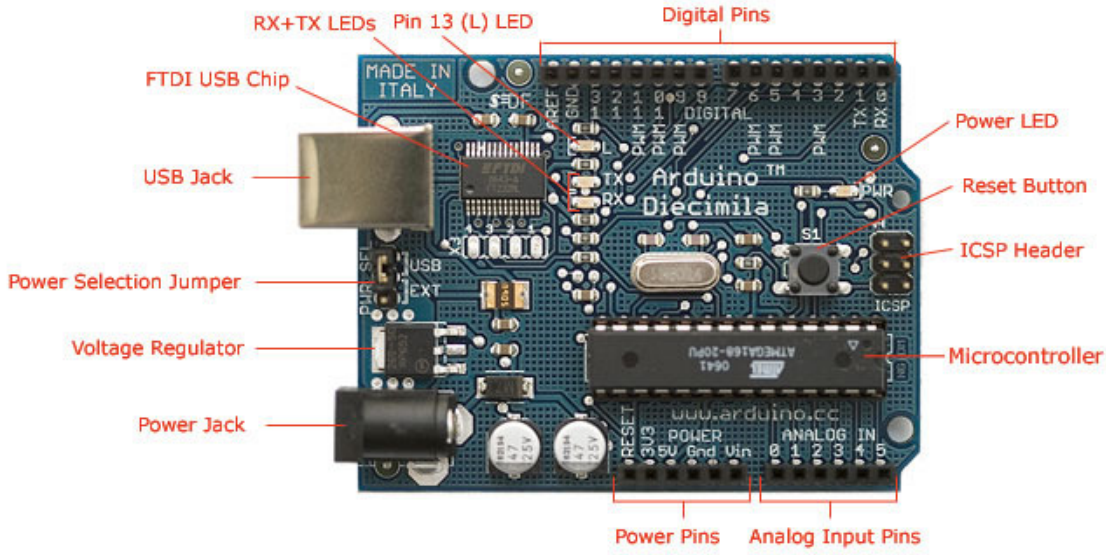
Em termos práticos, um Arduino é um pequeno computador onde você pode conectar componentes externos, e que você pode programar para processar informações desses componentes. O Arduino é o que chamamos de **plataforma de computação embarcada**, ou seja, **um sistema que pode interagir com seu ambiente por meio de hardware e software**.

Geralmente ele é usado em conjunto com outros dispositivos, em especial sensores, e tem várias aplicações. Alguns exemplos:

- http://info.abril.com.br/noticias/blogs/zonalivre/hardware/5-projetos-matadores-com-arduino/
- http://hacknmod.com/hack/top-40-arduino-projects-of-the-web/

Com tantos usos diferentes, quanto vocês acham que ele tem de memória? Nós vamos usar o **Arduino Uno**, que tem só **32KB de memória flash**. Parece muito pouco, não é? Mas é o suficiente para a maioria das aplicações que vimos. Há outras versões do Arduino, algumas com mais memória; vejam em http://arduino.cc/en/Main/Products.





Photograph by SparkFun Electronics. Used under the Creative Commons Attribution Share-Alike 3.0 license.

Configurando o ambiente de programação (IDE) do Arduino

Antes de começar a programar o Arduino, precisamos dizer ao seu IDE que tipo de placa vamos usar. Isso é feito no menu *Tools* → *Board*, onde existe uma lista das placas. Você deve escolher a placa que estiver usando no momento. No nosso caso, é a **"Arduino Uno"**.

Agora o IDE precisa saber em que porta do computador o Arduino está conectado. Clique novamente no menu *Tools* e, em seguida, em *Serial Port*. Escolha a porta apropriada e você já pode começar a usar o Arduino!

Uma dica: o IDE mostra no lado direito do seu rodapé o tipo de placa e a porta que estão selecionados atualmente.

A seguir, vamos testar se a instalação do Arduino deu certo com um programa bem simples.

Upload do seu primeiro sketch

"Sketch" (rascunho em inglês) é como o IDE do Arduino chama os programas. Ele já possui diversos exemplos prontos, que estão listados no menu *File* → *Examples*. Vamos começar por um dos mais simples, chamado **"Blink"**. Clique no menu *01.Basics* → *Blink* para abri-lo (uma nova janela será aberta).

Agora você precisa carregar esse código no Arduino. Faça isso clicando no 2º botão da barra verde abaixo dos menus (barra de ferramentas). Ele tem o desenho de uma seta para a direita, e se chama **"Upload"**. Observe algumas coisas que acontecem agora...

- Existe uma barra verde abaixo da área de codificação. Ela é chamada **"barra de status"**. Nela aparece a frase *"Compiling sketch..."* do lado esquerdo. Isso significa que o IDE está compilando o código, ou seja, verificando se existe algum erro nele.
- Do lado direito da barra de status aparece uma barra de progresso enquanto o IDE compila o código e faz o upload dele para o Arduino.
- No Arduino existe 1 LED chamado **RX** e outro chamado **TX**. **"RX"** é um termo usado para indicar **recepção de dados**, enquanto **"TX"** indica **transmissão (envio) de dados**. Ou seja, o LED RX pisca quando o Arduino está recebendo dados, e o LED TX pisca quando ele está transmitindo dados. Observe que esses 2 LEDs devem começar a piscar para mostrar a troca de informações entre o computador e a placa.
- Quando o upload do sketch tiver terminado, a frase *"Done uploading"* (upload completo) aparecerá na barra de status do IDE e as luzes RX e TX pararão de piscar na placa.

Depois de alguns segundos, observe o LED acima dos LEDs RX e TX (identificado por uma letra L). Ele é chamado de **LED 13** (em breve você saberá por que) e deve estar piscando, acendendo e apagando em intervalos de 1 segundo. Se isso estiver acontecendo, parabéns! O Arduino está funcionando direitinho no seu computador!

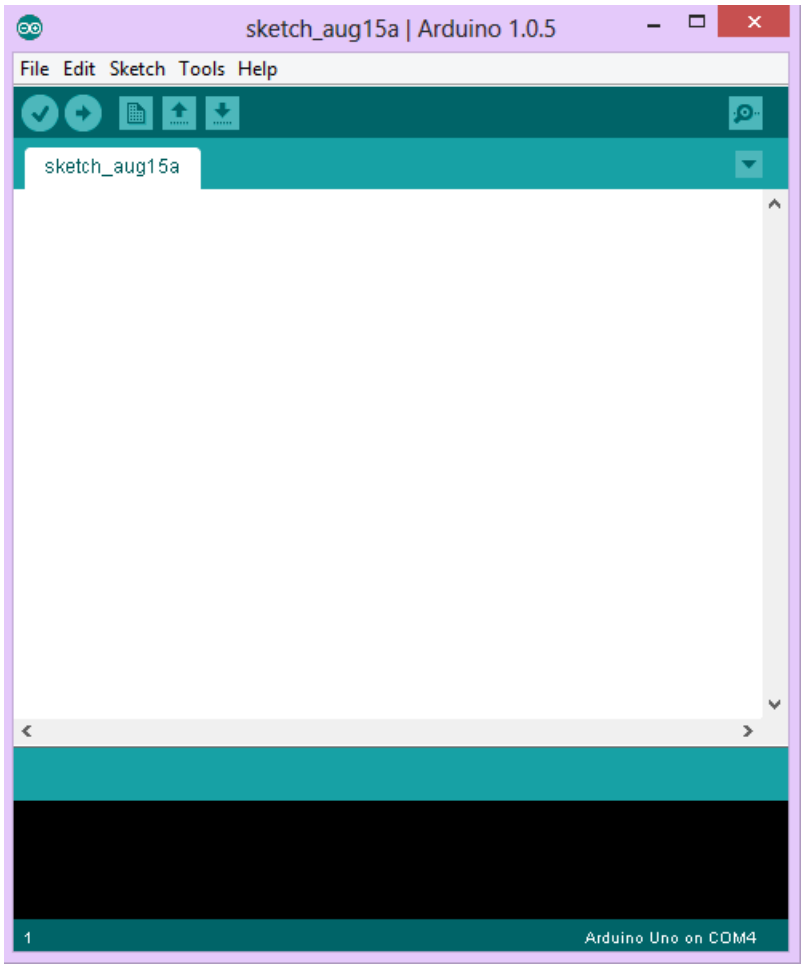
Mas por que o LED 13 está piscando? Porque foi isso que o sketch Blink mandou o Arduino fazer: acender e apagar o LED 13 em intervalos de 1 segundo. E agora esse programa que você carregou permanecerá na memória do Arduino até que outro programa seja carregado. Ou seja, na memória do Arduino cabe somente um programa por vez.

O programa que foi carregado no Arduino será executado sempre que a placa for energizada, o que é indicado pelo LED **"ON"** aceso. Há duas formas de energizar (ou alimentar) o Arduino. A 1ª nós já sabemos: através do cabo USB-serial que o conecta ao computador. A 2ª é através de uma bateria com tensão de 7V a 12V.

Antes de começarmos nosso primeiro projeto, vamos ao último passo: conhecer a IDE do Arduino.

Conhecendo o IDE do Arduino

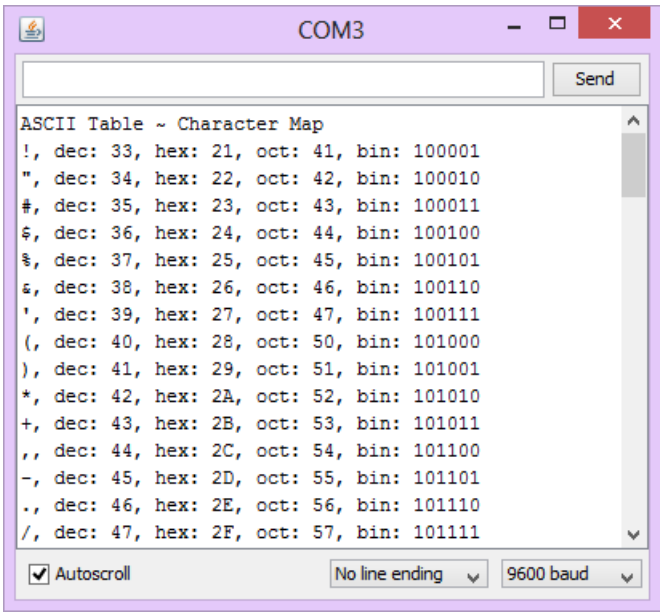
No topo da janela fica a **barra de menus**. Já entramos em duas das suas opções: *File* e *Tools*. Em seguida temos a **barra de ferramentas**, onde fica o botão *Upload*, que também já usamos. Abaixo temos uma **barra de abas (ou guias)** e a **área de codificação (Sketch Window)**. Depois dela, uma **área de mensagens**.



Vamos detalhar os botões da barra de ferramentas, que são uma forma mais rápida de usar algumas funções que também estão nos menus. Veja a função de cada botão na tabela a seguir.

Verify	Compila o código (verifica se há erros nele). Você pode usá-lo enquanto constrói o programa e antes de carregar o código (sketch) no Arduino.
Upload	Carrega o código no Arduino (e compila o código antes).
New	Cria um sketch em branco.
Open	Mostra a opção para abrir um sketch externo e a lista de sketches de exemplo.
Save	Salva o código aberto.
Serial Monitor	Exibe os dados seriais enviados do Arduino.

A figura abaixo mostra um exemplo de janela do monitor serial, que exibe os dados enviados pelo Arduino. O monitor é uma ferramenta muito útil, principalmente para depuração de código, ou seja, para encontrar erros nele. Na parte inferior da janela temos, à esquerda, a opção "*Autoscroll*". Quando ela está marcada, o texto rola automaticamente para cima sempre que ultrapassar o limite da janela. A caixa ao lado configura a forma como o texto é exibido, mas em geral não é modificada. A segunda caixa altera a taxa de transmissão (**Baud Rate**) de dados entre o computador e o Arduino. O baud rate é a taxa em que as informações são trocadas entre o computador e o Arduino por segundo. A configuração padrão é 9.600 baud.



A caixa de texto no topo, ao lado do botão "*Send*", serve para enviar dados seriais para o Arduino. Para isso seu código deve estar preparado para recebê-los, mas, por enquanto, não vamos usar essa forma de envio de dados.

A área central da janela exibe os dados seriais enviados pelo Arduino através da sua porta serial. **Mas, afinal, o que é essa porta serial?** Em termos práticos, é a porta onde você conecta o cabo que liga o Arduino ao computador. Em breve vamos conectar o Arduino a outros dispositivos eletrônicos e mecânicos, e você verá que eles se comunicam entre si, ou seja, trocam informações. Mas o Arduino também pode trocar informações com o computador, e eles "conversam" através dessa porta serial.

No nosso exemplo, o Arduino está executando o sketch *ASCIITable*, da coleção de exemplos *04.Communications*. Esse programa faz a saída de caracteres ASCII do Arduino pela porta serial, e o monitor serial os exibe.