Arduino + Brino para a robótica educacional

Gabriel Rodrigues Pacheco, Giulia Fricke Galice, Mateus Berardo de Souza Terra, Rafael Mascarenhas Dal Moro, Victor Rodrigues Pacheco.

Brasília, 2016

**Índice**

1. Introdução ...................................................................................................... 3
2. Começando .................................................................................................... 4

**Introdução**

A tecnologia está, atualmente, presente em todos os aspectos da vida humana, desde rastreadores de sono e outras tecnologias vestíveis até ferramentas educacionais e os mais tradicionais computadores e smartphones. Dentro desse contexto, o desenvolvimento de habilidades especificas se torna cada vez mais importante, mesmo para quem não trabalha na área de TI, a ponto de saber uma linguagem de programação ser tão fundamental quanto o inglês daqui poucos anos.

            Uma linguagem de programação é uma série de instruções, com regras sintáticas e semânticas, que são executadas por um processador. Praticamente todas as linguagens existentes são feitas utilizando o inglês como base. Observando a dificuldade de jovens e de quem não sabe inglês nós criamos uma linguagem de programação em português para o Arduino, o Brino, linguagem que almeja facilitar o entendimento da lógica de programa, sendo caracterizada como linguagem intuitiva e de simples compreensão.

O Arduino é uma placa com um microcontrolador que pode ser utilizada para prototipagem de sistemas e máquinas de forma fácil e rápida. Não apenas ao hardware, mas também um conjunto de software se refere o nome. A placa é baseada em um processador Atmel AVR e é feita em hardware livre. Portanto, é possível acessar a página do arduino e baixar os esquemáticos da placa para montar o seu próprio clone.

O arduino é atualmente utilizado em diversas áreas. Não apenas no ramo da tecnologia e desenvolvimento, pessoas adeptas do movimento maker (Faça você mesmo, do inglês DIY - Do it yourself) adotam esta placa pela curva de aprendizagem relativamente pequena. Além de hobbistas, artistas estão aproveitando as possibilidades da placa para construir obras interativas e responsivas.

A versatilidade da plataforma fez com que ela se tornasse muito popular. Nosso primeiro contato com o Arduino aconteceu no ano de 2014. Desde lá, desenvolvemos diversos projetos desde alguns voltados para IoT (Internet of things- internet das coisas), até alguns jogos e robôs autônomos. Para aprender utilizamos alguns livros e muita internet! O próprio fórum do arduino é um ótimo local para buscar auxílio assim como outros pela WEB. Sites como o instructables também é uma mina de ouro para quem busca um projeto interessante e não sabe por onde começar.

Os projetos disponíveis na internet são programados utilizando a própria linguagem do Arduino, se você está utilizando o Brino pode buscar nosso auxílio por email, na página do facebook ou até utilizar o código nativo do arduino em seu rascunho (o Brino oferece suporte ao código nativo do Arduino).

Tudo bem, você já leu uma folha inteira de teoria sobre o que é o Arduino, linguagens de programação, etc, etc... Se você, como nós, tem um espírito maker, já está de saco cheio de texto e quer começar a desenvolver seu projeto e fazer seu arduino ser útil, então vamos pôr a mão na massa! Antes disso, uma pequena dica: não seja levado completamente por sua impulsividade, aproveite um tempo antes de cada projeto para elaborar um bom planejamento e avaliar as possibilidades, isso evitará que você perca tempo e dinheiro.

**Começando**

Para começar você vai precisar, claro, de um Arduino. Seja um original ou um clone, a placa que você preferir provavelmente funcionará! Nessa apostila utilizaremos o arduino nano como referência. Recomendamos que o resto dos componentes seja adquirido quando você começar a desenvolver o projeto do capítulo, comprar todos juntos pode ser custoso. Algumas instruções para habilidades necessárias para o desenvolvimento de alguns projetos podem ser encontradas no final Vardo livro, como confecção de placas de circuito impresso e soldagem.

Primeiro, vamos configurar o seu computador para que você possa programar sua placa. Entre no site do arduino [arduino.cc] e vá na aba downloads para baixar a IDE mais recente que na ocasião era a versão 1.6.7. Siga as imagens para completar a instalação:

Depois de instalar a IDE do arduino, você está pronto para instalar o Brino. Para isso, basta acessar a página do GitHub: ratosdepc.github.io/Brino ; Na área de primeiros passos você encontra o link para a página de releases e os nossos contatos no final. Na página de releases, baixe o zip da versão mais recente e descompacte-o em algum lugar (de preferência que você possa encontrar facilmente). Depois disso, basta abrir o arquivo BrinoIDE.exe e você estará pronto para começar!

**Básico**

Para que possamos criar nossos próprios códigos é necessário que aprendamos palavras chaves e suas estruturas. Nesse capitulo iremos abordar assuntos como:

* Variáveis;
* Comentários;
* Instruções de controle;
* Loops;
* Como criar funções.

1. **Variáveis:**

Variáveis são muito usadas na programação pois elas são capazes de armazenar dados. Uma das formas mais simples de se pensar em uma variável é como uma caixa ou balde onde o computador pode armazenar ou ler dados. O uso de variáveis torna o código mais fácil de se entender e mais fácil de ser mantido. Por exemplo, se você possuir uma variável chamada *pinoLED* que define o pino 13 como uma saída para um LED e, mais a frente, decidir usar o pino 8 para isso, será fácil a troca sendo necessária apenas a troca do valor 13 pelo valor 8.

Esse exemplo poderia ser aplicado no segmento a seguir:

|  |
| --- |
| Numero pinoLED = 13 <= Aqui está a nossa variável!  Configuracao() {  Pino.definirModo(pinoLED, Saida);  }  Principal() {  Pino.ligar(pinoLED); // liga a porta digital acedendo o LED  esperar(1000); // espera por um segundo  Pino.desligar(pinoLED); // desliga a porta digital apagando o LED  esperar(1000); // espera por um segundo  } |

Agora que entendemos qual a utilidade das variáveis em nossos programas vamos examinar os seus tipos:

* Numero: As variáveis do tipo Numero são muito usadas pois elas são capazes de, como o próprio nome sugere, armazenar números inteiros entre

-32.768 a 32.767, ou seja, um numero de 16 bits.

Ex.: Numero minhaVariavel = 3600;

* NumeroDecimal: Esse tipo de variável é capaz de armazenar números de até 32 bits com um ponto decimal.

Ex.: NumeroDecimal Pi = 3,1415;

* Letra: Essa variável armazena um caractere ASCII (iremos abordar isso mais a frente), ou seja, ela é capaz de armazenar qualquer caractere (a, A, 1, &, entre outros). Operações aritméticas podem ser aplicadas sobre esses dados (também estudaremos isso mais a fundo). Seu dado deve vir entre aspas simples (‘ ‘).

Ex.: Letra nota = ‘A’;

* Palavra: Esse tipo especial de variável pode ser comparada a uma serie de caracteres. Ela é usada para armazenar palavras e frases. Seu dado deve vir entre aspas duplas (“ ”).

Ex.: Palavra saudacao = “oi”;

* Condicao: A menor variável que vamos estudar é usada para guardar apenas dois valores, Verdadeiro ou Falso, e será muito usada em operações logicas e como controle.

Ex.: Condicao chovendo = Falso;

Outro ponto importante quanto as variáveis são os nomes que elas podem receber, que pode ser quase qualquer coisa, desde que não seja nenhuma palavra palavra-chave. Mesmo com toda essa liberdade recomendamos fortemente que não usem acentos ou caracteres especiais, como ç, @, “ e coisas do tipo, e que seja dada a preferência a nomes sugestivos (não, uhsefiuhfwiu não é nada sugestivo) para que o código possa ser mais facilmente entendido pelos outros e por você mesmo. São exemplos de bons nomes: valorSensor, motorDireito, portaLED e leituraDistancia.

1. **Comentários:**
2. **Instruções de controle:**
3. **Loops:**
4. **Como criar funções:**