

UNIVERSIDADE FEDERAL DE ALAGOAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS GRADUAÇÃO EM MODELAGEM COMPUTACIONAL DO
CONHECIMENTO

ALEXANDRE JOSÉ BRAGA DA SILVA

**UM MODELO DE BAIXO CUSTO PARA AULAS DE ROBÓTICA EDUCATIVA
USANDO A INTERFACE ARDUINO**

MACEIÓ
2014

ALEXANDRE JOSÉ BRAGA DA SILVA

**UM MODELO DE BAIXO CUSTO PARA AULAS DE ROBÓTICA EDUCATIVA
USANDO A INTERFACE ARDUINO**

Dissertação DE Mestrado apresentada ao Programa de Pós Graduação em Modelagem Computacional de Conhecimento da Universidade Federal de Alagoas, como requisito parcial para obtenção do grau de Mestre em Modelagem Computacional do Conhecimento.

Orientador(a): Profa. Dra. Eliana da Silva Almeida

Coorientador: Prof. Dr. Luiz Marcos Gonçalves

MACEIÓ
2014

Catálogo na fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico
Bibliotecária: Maria Auxiliadora G. da Cunha

S586m Silva, Alexandre José Braga da.
Um modelo de baixo custo para aulas de robótica educativa usando a interface Arduino / Alexandre José Braga da Silva. – 2014.
109 f. : il.

Orientadora: Eliana Silva de Almeida.
Dissertação (Mestrado em Modelagem Computacional do Conhecimento) – Universidade Federal de Alagoas. Instituto de Computação. Maceió, 2014.

Bibliografia: f. 55-58.
Anexos: f. 59-109.

1. Arduino. 2. Robotica educativa. 3. Interfgace. 4. Controle.
I. Título.

CDU: 004.5:372.8



UNIVERSIDADE FEDERAL DE ALAGOAS/UFAL
Programa de Pós-Graduação em Modelagem Computacional de Conhecimento
Avenida Lourival Melo Mota, Km 14, Bloco 09, Cidade Universitária
CEP 57.072-900 – Maceió – AL – Brasil
Telefone: (082) 3214-1364



Membros da Comissão Julgadora da Dissertação de Mestrado de Alexandre José Braga da Silva, intitulada: “Um Modelo de Baixo Custo para Aulas de Robótica Educativa usando a Interface Arduino”, apresentada ao Programa de Pós-Graduação em Modelagem Computacional de Conhecimento da Universidade Federal de Alagoas em 16 de maio de 2014, às 14h00min, na sala de aula do Mestrado em Modelagem Computacional de Conhecimento da UFAL.

COMISSÃO JULGADORA

Prof. Dra. Eliana Silva de Almeida
UFAL – Instituto de Computação
Orientadora

Prof. Dr. Luiz Marcos Garcia Gonçalves
UFRN – Departamento de Engenharia de Computação e Automação
Coorientador

Prof. Dr. Leonardo Viana Pereira
UFAL – Instituto de Computação
Examinador

Prof. Dr. Roberto Jorge Vasconcelos dos Santos
Faculdade Estácio de Alagoas
Examinador

Maceió, maio de 2014.

AGRADECIMENTOS

Agradeço primeiramente a DEUS por ter me ajudado a chegar até esta etapa da jornada, a minha estimada orientadora Dra. Eliana Almeida, ao meu co-orientador Dr. Luiz Marcos, que mesmo de longe me ajudou com informações valiosas e aos demais professores do Programa de Pós-Graduação em Modelagem Computacional de Conhecimento. Aproveito ainda para felicitar e me congratular com meus colegas de mestrado, agradeço também a minha família e a todos aqueles que de alguma forma contribuíram com este trabalho.

RESUMO

Este trabalho descreve um método de integração entre a interface de controle *Arduino* como base tecnológica e dispositivos de baixo custo que possuam um grau de simplicidade adequado para que seja usado por crianças, adolescentes e pessoas que estejam se iniciando em projetos de robótica, automação e controle. Este estudo se baseia em protótipos desenvolvidos em oficinas e aulas de robótica educativa em uma escola de ensino fundamental e médio do estado de Alagoas. Os resultados alcançados através dos experimentos realizados, usando materiais reaproveitáveis, sucata e componentes eletrônicos de fácil aquisição no mercado, em conjunto com softwares integrados e utilizados pelos professores e alunos, demonstraram excelentes resultados em termos de aproveitamento das aulas, interesse, participação e melhorias nos conhecimentos ministrados, de acordo com as avaliações feitas por meio de questionários com os alunos.

Palavras chave: Arduino, robótica educativa, interface, controle

ABSTRACT

This paper describes a method of integration between the control interface Arduino as a low cost technological base that has an appropriate degree of simplicity to be used by children, teenagers and people who are starting projects in robotics, automation and control devices. This study is based on prototypes developed in workshops and educational robotics classes at a school for elementary and middle level in the state of Alagoas. The results obtained through experiments using reusable materials, scrap and electronic components easy to purchase in the market, set of integrated circuits and used by teachers and students, software showed excellent results in terms of exploiting classes, interest, participation and improvements in knowledge given, according to the assessments made by questionnaires with students.

Keywords: Arduino, educational robotics, interface, control

LISTA DE ILUSTRAÇÕES

Figura 1 - Interface de controle para robôs com várias portas de I/O	22
Figura 2 - Interface Arduino e seus componentes em detalhes	24
Figura 3 - Especificações do sensor ultrassônico HC-SR04	27
Figura 4 - Funcionamento do sensor de toque	28
Figura 5 - Esquema de ligação do sensor de temperatura LM35	28
Figura 6 - Esquema de ligação do sensor de luz no Arduino	29
Figura 7 - Esquema de ligação do acelerômetro MMA7361 no Arduino	30
Figura 8 - Comunicação de dados entre a Interface Arduino e um computador	31
Figura 9 - Material didático ZOOM e kit de peças da linha educativa LEGO	36
Figura 10 - Kit educacional Modelix	37
Figura 11 - Kit básico de peças para as aulas de robótica educativa	50
Figura 12 - Peças físicas das marcas Pololu e Tamiya	51
Figura 13 - Peças alternativas e de sucata usadas para as aulas de robótica	52
Figura 14 - Trechos do material de apoio ao professor	54
Figura 15 - Trechos do material didático do aluno	56
Figura 16 - Modelo de troca de mensagens entre página web e Arduino	60
Figura 17 - Página web com exemplo de conexão remota ao Arduino	63
Figura 18 - Montagem prática do projeto garra com mini servo motor	64
Figura 19 - Robô com esteiras para plantação automática de semente	71
Figura 20 - Projeto de robô com quatro patas	71
Figura 21 - Projeto de veículo autônomo com sensor de distância	72
Figura 22 - Aula e treinando para competições de robótica	77

LISTA DE TABELAS

Tabela 1 - Especificações técnicas da interface Arduino	26
Tabela 2 - Versões do Software Modelix System	44
Tabela 3 - Componentes usados no kit básico para aulas práticas	51
Tabela 4 - Principais configurações do SerialProxy	61
Tabela 5 - Comparativo de custos entre kits usados no projeto	70

SUMÁRIO

1	Introdução	11
1.1	Justificativa	12
1.2	Motivações	12
1.3	Materiais e Métodos	14
1.4	Revisão da Literatura	14
1.5	Estrutura do Trabalho	18
2	Robótica e Interfaces de Controle	20
2.1	Tipos de Interface	22
2.2	A Plataforma Arduino	23
2.3	Arquitetura e Funcionalidades do Arduino	24
2.4	Sensores e Atuadores	26
2.5	Ferramentas de Programação	30
2.6	A Robótica Educativa	32
2.7	Metodologias de Ensino de Robótica	34
2.8	As Competições de Robótica	37
3	Metodologias de Ensino da Robótica	40
3.1	Os Parâmetros Curriculares Nacionais	40
3.2	Legó Zoom	42
3.3	Modelix	43
3.4	RoboEduc	44
3.5	O PNCA Alfa	45
3.6	Plataforma Vex	46
4	O Modelo Proposto de Baixo Custo Usando a Interface Arduino	47
4.1	O Hardware Usado no Modelo Proposto	49
4.2	Material de Apoio ao Professor	52
4.3	Material Didático para o Aluno	54
4.4	Iniciando seu Próprio Projeto	57
4.5	A IDE WebSockets	58
4.6	A Integração entre os Elementos Propostos	60
4.7	A Aplicação do Modelo na Prática	69

5 Conclusões e Resultados Obtidos	76
5.1 Trabalhos Futuros	78
Referências	80
Anexos	85

1. Introdução

O principal objetivo deste trabalho é demonstrar um modelo genérico de ensino da robótica pedagógica baseado em dispositivos de baixo custo, material descartável, sucata e fazendo uso de software livre, integrando-os de forma específica a cada nível escolar dentro dos padrões pedagógicos dos Parâmetros Curriculares Nacionais do Ensino Fundamental e Médio. Sua relevância consiste em definir um modelo que sirva de base para qualquer instituição de ensino pública ou privada que tenha interesse em implementar aulas de robótica educativa e que precisem de um roteiro específico para tal. O público-alvo deste estudo são professores e facilitadores das escolas interessadas em implantar este tipo de atividade ou aquelas que já possuem aulas de robótica, mas gostariam de fazer mudanças no modelo atualmente em uso. Já existem algumas propostas neste sentido que podem ser adotadas, o diferencial do que estamos propondo é justamente fazer uso das melhores práticas usadas atualmente, com recursos financeiros mínimos, que não inviabilizem o projeto, na aquisição de hardware e software proprietário, juntamente com uma proposta de experimentação dos projetos executados no laboratório da escola, virtualmente, através de um site na Internet acoplado à interface *Arduino*. Como objetivos específicos deste trabalho, temos ainda:

- A elaboração de um roteiro mínimo, que compreende as noções básicas de eletrônica e mecânica, lógica de programação e conhecimentos preliminares sobre sensores e atuadores, que pode ser adotado pelos professores do Ensino Fundamental e Médio.
- Uma descrição de como integrar a interface *Arduino* com material de sucata ou de baixo custo de fácil aquisição no mercado local.
- Exemplos de utilização de sensores e atuadores, bem como seus limites de uso em situações reais ou em preparação para competições de robótica educativa.
- Um método para utilizar os recursos didáticos e controlar os dispositivos robóticos através do navegador Web fazendo a ponte entre a porta física (USB) na qual o *Arduino* está ligado com a porta lógica que estará em modo de escuta para interceptar os dados enviados e recebidos pelo projeto em uso.

As técnicas sugeridas neste trabalho foram aplicadas nos anos letivos de 2012 e 2013, durante as aulas regulares de robótica em uma escola particular do estado de Alagoas que já tinha em seu currículo regular, desde o ano de 2002, aulas de robótica educativa usando uma

plataforma de kits baseados na linha de produtos da LEGO *Mindstorms* e que vem ao longo do tempo desenvolvendo o seu próprio material didático, assim como alguns dos projetos práticos apresentados aos alunos.

1.1 Justificativa

disseminação cada vez maior de dispositivos robóticos e a sua utilização por instituições de ensino como ferramenta de ensino/aprendizagem de conceitos físicos, matemáticos e de lógica faz com que este importante recurso não seja utilizado da melhor maneira e muitas vezes aqueles que querem desenvolver ou mesmo utilizar tais recursos não tenham em mãos as formas corretas de utilizar estes recursos. Desta forma, ter um modelo mais genérico que possa auxiliar neste desenvolvimento se torna uma alternativa bastante viável na medida em que será implementado um modelo de ensino lúdico que faz uso de materiais alternativos, uma interface simples de utilizar e ao mesmo tempo bastante versátil, integração das habilidades cognitivas (raciocínio lógico e solução de problemas) e interações Sujeito-Sujeito (trabalho em equipe), Sujeito-Objeto e Sujeito-Cultura. Além disso, o modelo proposto incentiva a criação de situações problemas do mundo real, culminando em discussões interdisciplinares e experimentações práticas em sala de aula, com o objetivo de testar as soluções criadas pelos alunos.

1.2 Motivações

A robótica como ferramenta de ensino/aprendizagem tem surgido nas instituições brasileiras desde o final dos anos 90 com a linguagem LOGO, que fazia uso de uma tartaruga virtual para ilustrar os conceitos de movimentação no plano XY com comandos simples. Iniciativas como o projeto Roboticando, da UFRGS (ROBOTICANDO, 2004) e SIROS da UNICAMP, fizeram uso de ambientes virtuais de simulação para introduzir os conceitos de robótica em laboratórios de informática através de robôs virtuais (SIROS, 2004). Empresas como ARS Consult, que foi a primeira a desenvolver um kit de robótica educacional brasileiro, o *Super Robby*, e *Cyberbox* que também fazia uso de material de sucata, foram alguns dos precursores no uso de material de sucata para o ensino de robótica nas escolas. Ainda no início deste

século, a empresa dinamarquesa LEGO desenvolveu um kit de robótica chamado *Mindstorms*, que era importado e distribuído no Brasil pela empresa Edacom e até hoje é muito utilizado por escolas e universidades para experimentos e competições de robótica. As aplicações em robótica são atualmente baseadas em modelos tradicionais de linguagens de programação, e neste contexto muitas das aplicações tornam-se ineficientes na utilização de recursos escassos do hardware e software. Quando mencionamos recursos escassos nos referimos a quantidade limitada (na faixa dos kilobytes) de memória RAM e do processador mais lento destas interfaces. Conforme (COSTELHA & LIMA, 2007), a maioria dos modelos adotados para representar tarefas robóticas não se baseia em abordagens formais, mas adaptados à mão a cada tarefa. Desta forma, utiliza-se as técnicas tradicionais de programação, como orientação a objetos e programação utilizando blocos de instruções, para criar ambientes de programação para robôs. A criação de um modelo de ensino que faz uso de uma interface de controle com recursos escassos como o *Arduino*, em conjunto com os referenciais teóricos construtivistas pregados por teóricos como Piaget e Papert que pregam a interação do aprendiz com objetos físicos (Sujeito-Objeto) nos trás desafios complexos, na medida em que esta integração passa também pela supervisão de professor capacitado e um projeto pedagógico alinhado com a proposta da instituição de ensino e os Parâmetros Curriculares Nacionais de cada série escolar. Para (SCHONS ET AL., 2004), a robótica educativa “[...] constitui nova ferramenta que se encontra à disposição do professor, por meio da qual é possível demonstrar na prática muitos dos conceitos teóricos, às vezes de difícil compreensão, motivando tanto o professor como principalmente o aluno”. Nesta visão de Schons podemos acrescentar que além da motivação é necessário que o modelo adotado seja adequado ao nível dos alunos envolvidos e, além disso, identificando as potencialidades e competências de cada aluno de acordo com a sua área de atuação (hardware ou software) e focando também no incentivo à criatividade dos alunos e professores no desenvolvimento de projetos práticos, utilizando materiais de baixo custo e reaproveitamento de peças usadas.

Outra motivação para a elaboração deste trabalho foi tentar responder a algumas perguntas:

- Seria a robótica uma ferramenta apropriada para auxiliar no desenvolvimento de habilidades e aprimorar os conhecimentos de estudantes do nível fundamental e médio?
- As atividades lúdicas envolvendo robótica poderiam motivar mais os alunos a buscar novos conhecimentos?

- Quais os conteúdos abordados no currículo tradicional das escolas poderiam servir de plano de fundo durante as atividades de robótica educativa?

1.3 Materiais e Métodos

A fundamentação teórica para a elaboração deste trabalho se baseia na área de indexação de periódicos científicos, bem como o referencial obtido em livros, busca em repositórios de conhecimento como o *Science Advisor*, mecanismo de busca do Google acadêmico páginas web de instituições de ensino, pesquisa e páginas dos desenvolvedores das aplicações e ferramentas usadas neste projeto. As referências foram pesquisadas com base principalmente no fator de impacto dos periódicos pesquisados, em seguida a quantidade de citações de cada publicação e por fim o grau de relevância para o tema abordado. Na abordagem técnica deste trabalho, será utilizada a plataforma *Arduino*, bem como a linguagem de programação Java, juntamente com a linguagem C++ e as páginas *web* desenvolvidas neste projeto utilizam a linguagem HTML5 e CSS3. Ao término do projeto, uma das contribuições desse trabalho é a produção de um roteiro completo de como implementar um modelo de aula de robótica educativa para alunos do Ensino Fundamental e Médio, além de um conjunto de ferramentas de software integradas para auxiliar nas experimentações práticas através do ambiente da Internet. A ferramenta proposta, como parte dos aplicativos utilizados nos desenvolvimento desse trabalho poderá ser implementada em várias plataformas e devido ao seu caráter didático será de código aberto com licença de uso *GNU General Public License (GPL)*.

1.4 Revisão da Literatura

Uma das primeiras iniciativas do governo brasileiro que incentivou o uso da informática na educação foi o PRONINFE (programa Nacional de Informática Educativa). Este programa tinha por objetivo principal estimular as iniciativas para o uso da informática nos três níveis de educação, incluindo posteriormente também a educação especial. Pretendia-se com este programa o desenvolvimento da informática em todo o território brasileiro por meio dos Centros de Informática Educativa nas Universidades Federais, através dos CIES (Centros de Informática na Educação Superior) e Escolas Técnicas Federais. Outro foco do programa era

capacitar de forma contínua e permanente professores, técnicos e pesquisadores em informática educativa (ANDRADE, 1996). Este programa apresentou resultados bastante significativos entre 1989 e 1995, mas por dificuldades orçamentárias e de gestão foi substituído oficialmente em 1997 pelo PROINFO (Programa Nacional de Informática na Educação), cuja estrutura era mais centrada na Secretaria de Educação a Distância (SEED) e nas Secretarias de Educação dos Estados. Neste novo modelo, a capacitação dos professores ficava a cargo dos NTEs (Núcleos de Tecnologia Educacional) de cada Estado e não mais nas Universidades. Atualmente o PROINFO chama-se Programa Nacional de Tecnologia Educacional e não está focado apenas em treinamentos, mas também em levar computadores, equipamentos e conteúdos educacionais para as escolas públicas.

Em termos de robótica educativa pelo mundo podemos destacar a *Carnegie Mellon University* com a *Robotics Academy*, uma iniciativa que visa “desenvolver ferramentas para os professores que tornam mais fácil implementar aulas de robótica em sala de aula para alunos do ensino médio e superior”. (ROBOTICS ACADEMY, 2014). Eles trabalham em parceria com instituições como National Science Foundation, The *Department of Education* e *LEGO Group*, ministrando treinamentos presenciais em *Pittsburgh, PA* e certificações online na linha de produtos *LEGO Mindstorms* e *Vex*. A NASA também possui um projeto em parceria com várias instituições de ensino chamado *The Robotics Alliance Project* que engloba estudantes da educação infantil, passando pelo ensino fundamental, médio, superior e pós-graduação, com projetos voltados para o treinamento de estudantes que participam de competições de robótica usando também os kits da linha LEGO (ROBOTICS PROJECT, 2014). Opções como os MOOCs (*Massive Open Online Courses*) oferecem vários cursos online a nível mundial, porém voltados para estudantes de nível superior e pós-graduação na área de robótica e mecatrônica, não havendo nenhuma opção disponível para crianças e adolescentes. Observa-se nas pesquisas realizadas, que grande parte do conteúdo publicado e disponibilizado neste tema se baseia em treinamentos e certificações voltadas ao preparo dos professores e alunos para participar de competições de robótica a nível mundial, assim como para a utilização de kits de robótica comerciais, com pouca ênfase ao uso de outros produtos.

A informática é a porta de entrada para o aprendizado de outros recursos tecnológicos e neste sentido a robótica educacional tem um papel bastante importante para que o indivíduo consiga ser mais participativo na construção do conhecimento. De acordo com o pensamento Construcionista, o aprendizado é mais significativo para o aprendiz quando este é um produto de sua própria criação. Desta forma, a robótica é uma ferramenta no processo educacional na

qual o aluno atua na construção do próprio conhecimento, aumentando o seu leque de aprendizagem (CURCIO, 2008).

Um dos elementos formadores da robótica é a interface de controle a ser usada para controlar o hardware do robô. A noção de interface, adaptada da informática, está relacionada aos dispositivos tecnológicos desenvolvidos para viabilizar o contato entre meios heterogêneos (PAIS, 2002). Neste sentido observa-se que a maioria das placas usadas como interface de controle se baseiam em microcontroladores comuns no mercado mundial. Isto é interessante na medida em que simplifica a construção e aquisição destas interfaces ou ainda a compra de uma interface já pronta. De acordo com Lewin (2006), as interfaces usadas em projetos de robótica podem ser classificadas em dois grandes tipos de acordo com a sua utilização:

- Interface embarcada
- Interface não embarcada

As interfaces embarcadas geralmente são usadas para controle do hardware em tempo real com respostas rápidas a estímulos do meio ambiente no qual o robô está inserido. Este tipo de interface requer um microprocessador mais rápido e meios de comunicação sem fio também mais rápidos para coleta e tratamento de dados (LEWIN, 2006).

As interfaces não embarcadas são usadas em projetos onde há uma interação mais estreita entre a interface de controle e um computador externo. Estas interfaces podem ser mais lentas em termos de processamento e podem usar meios físicos de comunicação como cabos de rede ou USB (LEWIN, 2006).

Lewin cita ainda os microcontroladores mais usados neste tipo de interface: PIC da Microchips, AVR da Atmel e Advantech PCM em versões embarcadas ou não.

Diferentemente de outras aplicações da robótica, como industrial, onde o produto final é o que mais interessa a robótica educacional foca o processo de construção e a elaboração do pensamento do aluno, não tendo tanto interesse no produto final, mas sim no caminho que é feito até chegar a ele (CASTILHO, 2002). Konzen concorda com Castilho na sua afirmação:

“... a robótica educativa visa ao processo de construção e elaboração do pensamento do aluno. Na robótica educativa o caminho percorrido pelo aluno até a chegada a um determinado produto é a fase mais importante” (KONZEN, 2007).

Miranda (2006) cita dois problemas que são fatores importantes que impedem a expansão da robótica educacional no Brasil: o alto custo dos equipamentos e a limitação tecnológica de alguns produtos comercializados aqui, já que existe muita carência quanto ao desenvolvimento de metodologias e materiais para criação de projetos pedagógicos aplicados à robótica. Já Ribeiro (2006) afirma que o desenvolvimento do trabalho com a robótica utilizando material de sucata é bastante atraente ao aluno, porém exige conhecimentos que não se tornam atraentes ao professor. Desta forma, há de se tentar encontrar um equilíbrio entre estas duas visões tornando as aulas de robótica educativa mais atraente para todos os atores envolvidos nesta atividade.

Outra vantagem com relação as aulas de robótica diz respeito ao trabalho em equipe, pois cada projeto proposto deve obrigatoriamente ser implementado em grupos de alunos, o que estimula o raciocínio lógico em grupo e contribui significativamente para a união entre conhecimento teórico e experimentação prática (MIRANDA, 2007). Além destas contribuições, (ZILLI, 2004) cita outras também importantes:

Investigação e compreensão

- Trabalho com pesquisa
- Resolução de problemas por meio de acertos e erros
- Capacidade crítica.

Conforme citado anteriormente, um fator de impacto significativo para implantação de qualquer modelo de aula de robótica educativa é o custo. A utilização de um modelo que faça uso em sua maioria de materiais de baixo custo e de software livre surge como um contraposto a este problema. O pesquisador e programador Richard Stallman, do MIT (Instituto de Tecnologia de Massachusetts formalizou o conceito de Software Livre em um manifesto no qual define a licença de uso de um programa livre: a licença GNU, também conhecida como Licença Pública Geral (GPL). Para defender seus ideais, Stallman criou a *Free Software Foundation* (FSF), que defende o conceito de software livre. Com base nesta plataforma é possível implementar um curso de robótica educativa baseado em software livre

sob qualquer arquitetura de Sistema Operacional (Windows ou Linux). A utilização tanto de software como de material didático produzido sob licença GNU é uma grande vantagem pois não implica no gasto com pagamento de direitos autorais ou de licenças de software.

Com a crescente evolução das tecnologias usadas na internet, surgiram soluções interessantes como os laboratórios remotos, que possibilitam o acesso a elementos físicos, coleta de dados de forma remota e análise de resultados remotamente via internet. Segundo Gasparetti, a utilização deste tipo de laboratório traz benefícios tanto para o ensino presencial como para o desenvolvimento de conteúdo on-line,

Os benefícios dos laboratórios remotos estão em sua flexibilidade, sua independência de tempo e local, possibilidade de acesso de um maior número de alunos a uma plataforma experimental, acesso a equipamentos caros que de outro modo não estariam disponíveis, e permitir acesso de estudantes com deficiências físicas. (GASPARETTI, 2004)

Com a união de um modelo de laboratório remoto e aulas tradicionais temos um melhor aproveitamento dos recursos didáticos na medida em que o aluno pode agregar conteúdo posteriormente às aulas na escola através de uma página web e esta página será compartilhada com outros alunos da mesma escola e de outras. A possibilidade de integrar projetos presenciais com material complementar na internet também facilita que os próprios alunos sejam criadores de conteúdo através de blogs, redes sociais e wikis, por exemplo.

1.5 Estrutura do Trabalho

Este trabalho está dividido em seis capítulos, começando pela introdução a qual descreve sua justificativa, objetivos, ferramentas e metodologia adotada.

No capítulo 2, temos uma abordagem da representação de modelos de ensino de robótica atualmente em uso em várias partes do país e como estes modelos podem servir de base para uma integração dos vários dispositivos tratados neste trabalho.

No capítulo 3 é abordado um modelo genérico de ensino de robótica pedagógica que faz uso da interface *Arduino* e de outros materiais de baixo custo, associado a uma

implementação de software que faz uso do ambiente da Internet e do próprio ambiente de programação da interface.

No capítulo 4 temos um roteiro prático de como implementar estes recursos para cada nível escolar de acordo com as habilidades cognitivas dos alunos e com a supervisão do professor e acompanhamento do material didático interdisciplinar, assim como o funcionamento de forma prática deste modelo e os seus resultados aplicados com os alunos e professores.

Por fim, no capítulo 5 são dadas as conclusões obtidas com este trabalho e as sugestões para melhorias e aperfeiçoamentos no modelo e na API para perspectivas futuras.

2. Robótica e Interfaces de Controle

Neste capítulo é mostrada uma definição sucinta da robótica e dos principais tipos de controladores e interfaces usados em vários tipos de aplicações, alguns dos tipos de sensores comumente utilizados nas atividades de robótica, além de uma perspectiva do funcionamento de um robô e seus muitos campos de atuação.

O termo robô foi utilizado pela primeira vez por Karel Capek ao escrever uma peça teatral. Esta peça tinha como personagens seres mecânicos com forma humana, que realizavam tarefas de forma automática. Já a palavra robótica foi criada pelo escritor de ficção científica Isaac Assimov, no seu romance “Eu, robô”, de 1948. A robótica é um ramo da tecnologia que faz uso integrado de sistemas computacionais (software) e dispositivos elétricos e mecânicos (hardware) para atuar em um ambiente (atuadores) ou obter informações deste ambiente (sensores) (BURNS, 1997).

Em uma definição mais abrangente e de acordo com o *Robot Institute of America*, um robô é:

“Um manipulador programável multifuncional, projetado para mover materiais, objetos, ferramentas ou aparelhos específicos através de vários movimentos programados com vista à realização de determinadas tarefas.”
(TEIXEIRA, 2006)

Para que um dispositivo robótico venha a funcionar corretamente e tenha seu comportamento de acordo com a função para a qual foi projetado é necessário que este dispositivo seja controlado por uma interface ou por algum dispositivo de controle. Existem muitas interfaces que são usadas para controlar robôs, desde os mais simples indicados para iniciantes e *hobbistas* até as interfaces complexas usadas, por exemplo, para controlar automóveis que se autoconduzem. Qualquer sistema desenvolvido para controlar robôs tende a ser extremamente complexo por ter que lidar com eventos em tempo real e ambientes dinâmicos e repletos de incertezas. Além disso, estes sistemas precisam lidar com informações vindas de múltiplos sensores atuando simultaneamente e de atuadores que, em certos tipos de robôs, precisam agir com precisão (robôs cirurgiões, por exemplo).

Atualmente as interfaces de controle em geral para controle de robôs tendem a ser modulares, ou seja, permitem que outras funcionalidades sejam acrescentadas à uma placa básica

expandindo as suas funcionalidades. A interface *Serializer 3.0* é um exemplo de placa modular que faz uso do *Framework .NET* ou da linguagem C++ e que permite o controle de vários sensores, atuadores e acesso a redes sem fio Bluetooth, wifi e zigbee (ROBOTICS CONNECTION, 2012). A figura 1 ilustra esta interface.



Figura 1- Interface de controle para robôs com várias portas de I/O

(http://www.roboticsconnection.com/skins/skin_1/images/rc_controllers/SerializerIsoLarge.png)

Muitas outras interfaces de controle comerciais e *Opensource* podem ser facilmente encontradas na Internet. Estas interfaces possuem como componente principal um processador mais simples do tipo ARM, *Intel Atom*, *Motorola*, PIC ou *Atmel*. Estes processadores se caracterizam por serem baratos, confiáveis e de arquitetura simplificada para lidar com vários tipos de dispositivos de entrada e saída. As plataformas baseadas em ARM, PIC e *Atmel* são as mais usadas atualmente por vários fabricantes, pois a quantidade de dispositivos e complementos disponíveis no mercado é muito maior para estas plataformas, assim como a disponibilidade de programas e ambientes de programação.

2.1 Tipos de Interface

Os principais tipos de interface para controle de robôs, dispositivos mecatrônicos e aplicações em redes de sensores podem ser divididos em quatro grupos:

Interface autônoma – Este tipo de interface não requer necessariamente ligação com um computador externo. Possui sistema operacional (*firmware*) embarcado e permite a programação diretamente na interface com a utilização de algum *display* e teclado externo. São interfaces mais complexas e maiores, com mais memória e poder de processamento usado para controlar dispositivos maiores e com mais mobilidade. Podemos citar como exemplos de interfaces autônomas:

- Interface semi autônoma – São aquelas interfaces que necessitam de conexão a um computador externo para receber a programação através de um cabo ou de uma conexão sem fio de curto alcance (infravermelho, Bluetooth, etc.), mas que depois de feita a transferência do programa do computador para a memória interna da interface, a mesma pode ser desconectada do computador e operar de forma autônoma até que alguma mudança no programa seja necessária e neste caso necessite novamente de uma conexão a um computador externo. Alguns exemplos de interface semi autônomas: Microcontrolador LEGO RCX e NXT, *Arduino*, Microcontrolador da linha VEX.
- Interface embarcada – É um tipo específico de interface desenvolvida pelo fabricante de algum equipamento usado para prototipagem, linha de produção, equipamentos industriais e também em computação ubíqua, onde há a possibilidade de controlar dispositivos embarcados em dispositivos móveis. Este tipo de interface engloba em um único circuito integrado a memória, o núcleo de processamento e *firmware*, responsável pela inicialização e funcionamento da lógica do dispositivo. Uma interface embarcada pressupõe nenhuma intervenção externa, e, portanto espera-se que funcione por anos ininterruptamente, contando com recursos que permitem a sua recuperação autônoma em caso de erros. Alguns exemplos de interface embarcada são: Processador embarcado Nios II,
- Interface rádio controlada – Esta categoria de interface de controle é usada para permitir o controle humano a dispositivos mecatrônicos ou robóticos por meio de ondas de rádio de médio e curto alcance. Este tipo de interface é geralmente usado em

competições onde os robôs são controlados por equipes de pessoas e também na indústria do cinema para controle de animatrônicos – bonecos com movimentos controlados por meio de rádio frequência.

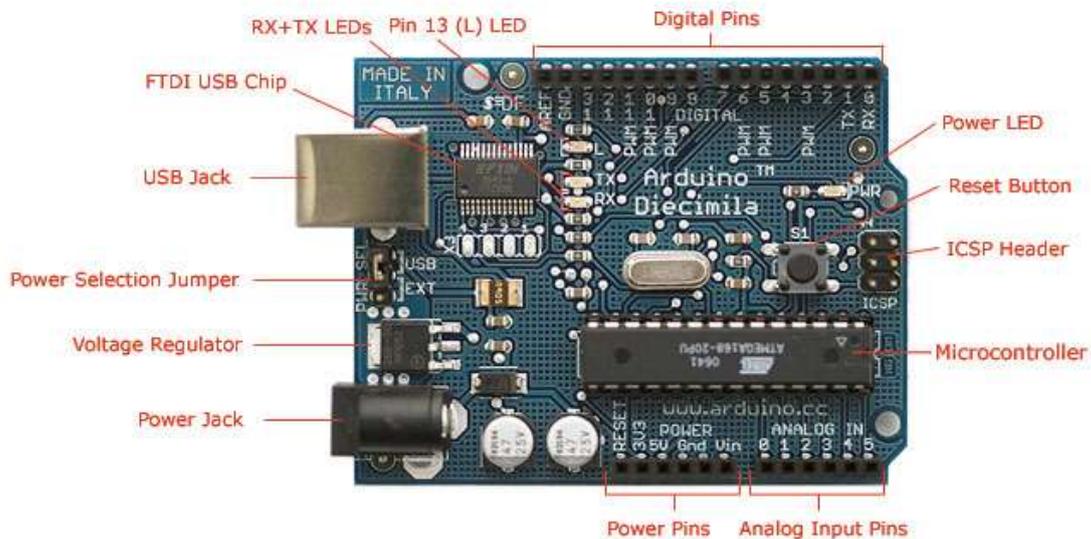
2.2 A Plataforma Arduino

Apesar de haver um grande número de interfaces de controle no mercado, o foco principal deste trabalho é a interface *Arduino* em função da sua flexibilidade de uso e das muitas aplicações que esta plataforma oferece principalmente no âmbito da robótica educativa. De acordo com McRoberts, um *Arduino* é um pequeno computador programável para processar entradas e saídas entre o dispositivo e os componentes externos ligados a ele (MCROBERTS, 2011). O projeto do *Arduino* surgiu na cidade de Ivrea, na Itália, no ano de 2005 e seu objetivo era criar uma plataforma de prototipagem simples, barata e livre que qualquer pessoa com o mínimo de experiência com eletrônica pudesse montar e operar. Devido ao projeto simples a plataforma *Arduino* rapidamente se tornou conhecida mundialmente e foi adotada por muitas instituições de pesquisa como uma interface base para projetos de inovação tecnológica. Devido a sua simplicidade de hardware, confiabilidade e versatilidade, o *Arduino* pode ser usado de forma autônoma, controlando diretamente vários atuadores e recebendo estímulos de vários sensores, ou pode ser usado ligado a um computador repassando informações para ele e recebendo instruções de volta (MCROBERTS, 2011).

Para que se possa desenvolver aplicações para a plataforma *Arduino* foi desenvolvida uma IDE (*Development environment*) que atualmente encontra-se disponível no Google Code sob licença GNU GPL/2 e *Creative Commons*. O software do *Arduino* é formado pela IDE, escrita em Java e baseada no ambiente de desenvolvimento *Processing* e por um conjunto de bibliotecas complementares escritas em linguagem C e C++ (PROJETO GOOGLE CODE, 2012).

2.3 Arquitetura e Funcionalidades do Arduino

O hardware do *Arduino* é formado por um microprocessador Atmel AVR, associado a um oscilador de cristal que especifica sua velocidade de processamento de 16Mhz, um circuito regulador de tensão que limita sua operação a 5 volts, um circuito conversor TTL/FTDI e uma porta USB para comunicação com o PC. A figura 2.0 mostra uma interface *Arduino* e seus principais componentes.



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

Figura 2 - Interface Arduino e seus componentes em detalhes.

(<http://arduino.cc/en/Main/ArduinoBoardDiecimila>)

Na figura 2 destacam-se os elementos físicos da placa de circuitos do *Arduino*:

- Pinos de força – São usados para alimentação de energia externa aos componentes a serem ligados na placa (sensores, atuadores, outros circuitos, etc.). O pino de Reset é usado para reiniciar um determinado elemento, os pinos de 3,3V e 5V são ligados ao terminal positivo dos componentes e o pino Grd (terra) é ligado ao terminal negativo dos componentes.

- Pinos analógicos – São ao todo seis pinos identificados de 0 à 5 e são usados para ligar dispositivos analógicos como: motores simples, sensores analógicos, etc. Este pinos podem receber alimentação a partir da porta USB ou diretamente dos pinos de força.
- Pinos digitais – São ao todo treze pinos identificados de 0 à 12 e são usados para ligar dispositivos digitais como servo motor, motor de passo, sensores digitais, etc. Os pinos digitais 0 e 1 possuem ainda uma função especial são os pinos RX e TX, respectivamente, e são usados para comunicação serial com outras placas ou computador. Os pinos digitais 3, 5, 6, 9, 10 e 11 são ainda usados para controlar dispositivos que trabalham com PWM¹ e o pino 13 é também ligado a um LED na própria placa do *Arduino* que serve para fazer testes como por exemplo ligar e desligar (MCROBERTS, 2011).
- Pinos ICSP – Este pinos são usados para enviar códigos de programação diretamente para o Microcontrolador. ICSP (*In Circuit Serial Programmer*) é a forma mais simples e rápida para enviar instruções simples para Microcontroladores do tipo PIC e *Atmel*, mas requer conhecimentos específicos dos cabeçalhos e instruções em *Assembler* para este tipo de dispositivo (INSTRUCTABLES, 2012).
- Botão de Reset – É usado para provocar uma interrupção temporária na execução do programa atualmente em uso na memória RAM do Microcontrolador. Esta ação é feita em casos onde é preciso reiniciar uma aplicação ou forçar a saída de um *Loop* infinito.

A tabela 1 ilustra as especificações técnicas básicas do modelo padrão da interface *Arduino*. Estas especificações se referem ao modelo *duemilanove*, podendo haver algumas mudanças nas versões mais recentes e nas versões nano e mega. Para efeitos práticos, no entanto, estas especificações são suficientes para a grande maioria dos projetos.

¹ PWM (*Pulse Width Modulation*) ou Modulação por Largura de Pulso é um conjunto de pulsos digitais em uma largura medida em milissegundos. Os pulsos típicos têm entre 40Hz e 50Hz.

Tabela 1 – Especificações técnicas da interface Arduino modelo Duemilanove.

Microcontrolador	ATmega328
Voltagem operacional	5V
Voltagem de alimentação (recomendada)	7-12V
Voltagem de alimentação (limites)	6-20V
Pinos I/O digitais	14 (dos quais 6 podem ser saídas PWM)
Pinos de entrada analógica	6
Corrente contínua por pino I/O	40 Ma
Corrente contínua para o pino 3.3V	50 Ma
Memória flash	32 KB (2KB usados para o bootloader)
SRAM	2 KB
EEPROM	1 KB
Velocidade de clock	16 MHz

Fonte: www.arduino.cc.

Em alguns modelos de placas, como o *Arduino Mega*, existem mais componentes e mais pinos de entradas e saídas. No entanto, a descrição básica citada neste capítulo abrange a grande maioria de placas desta plataforma. Os pinos de conexão estão bem identificados na placa, porém deve-se tomar muito cuidado no momento de criar circuitos externos e fazer sua conexão à placa do *Arduino*, pois em alguns casos uma ligação errada pode provocar danos irreparáveis à placa ou aos dispositivos externos ligados nela, especialmente outros circuitos integrados.

2.4 Sensores e Atuadores

Dentro os muitos componentes que podem ser usados em conjunto com a interface *Arduino* para a criação de robôs e dispositivos mecatrônicos, os sensores são os elementos com os quais o seu dispositivo “sentem” o mundo a sua volta. Dependendo da forma como são feitos os sensores convertem informações captadas no ambiente ao redor do robô em sinais elétricos que podem ser interpretados pela interface *Arduino* de duas formas diferentes: analógico ou digital (MARGOLIS, 2011, p.166). Os sistemas de sensores podem ser classificados de acordo com a sua atuação em dois tipos: sensores ativos e sensores passivos. Os sensores

ativos são aqueles que possuem sua própria fonte de radiação eletromagnética (como radares, lasers e acústicos). Os sensores passivos são aqueles que medem a radiação emitida pelos elementos externos (como câmeras, sensores de luz e de calor).

Atualmente os sensores mais utilizados em conjunto com o *Arduino* são:

- Sensores de distância ou ultrassônico – baseados em um emissor e receptor de sinal ultrassônico semelhante ao dos morcegos. Um sensor ultrassônico mede o tempo que o pulso de som leva para ir do emissor, bater em um obstáculo e retornar ao sensor (MARGOLIS, 2011, p.174). Um dos modelos mais fáceis e barato do mercado é o HC-SR04 da *cytron Technologies*, que possui alcance entre 2 e 40 centímetros com precisão de aproximadamente 3 milímetros. Este módulo sensor possui uma tensão de operação de 5v e consumo de 15mA, o que o torna excelente para trabalhar nas portas digitais do *Arduino*. A figura 3 mostra as dimensões do sensor e o seu raio de alcance.

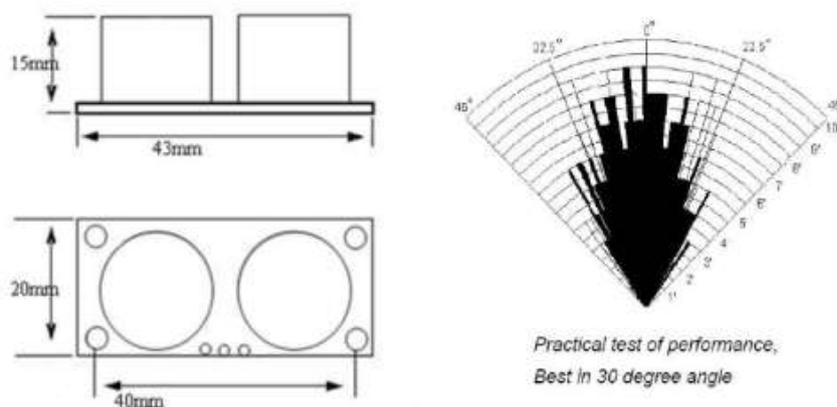


Figura 3 - Especificações do sensor ultrassônico HC-SR04. (https://docs.google.com/document/d/1Y-yZnNhMYy7rwhAgyL_pfa39RsB-x2qR4vP8saG73rE/)

- Sensores de toque – são conhecidos também como sensores de pressão ou chave de final de curso. Este tipo de sensor é semelhante a uma chave liga/desliga e desta forma enviam apenas dois estados para o *Arduino*: ligado (5v) ou desligado (0v) quando suas partes móveis fazem contato. Este tipo de sensor pode ser usado para que o robô desvie de algum obstáculo próximo (dentro do alcance de toque do sensor). Na figura 4 podemos ver uma forma de ligar o sensor de toque aos pinos do *Arduino*.

sensores podem ser ligados diretamente nas portas analógicas ou em série com um resistor de aproximadamente 150 ohms.

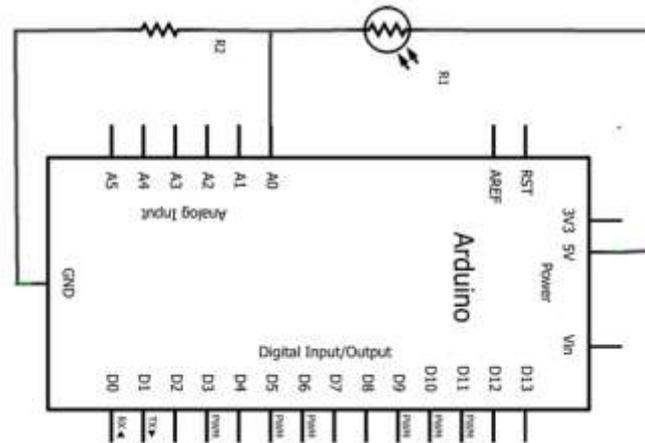


Figura 6 - Esquema de ligação do sensor de luz no Arduino.

(<http://web.djodjo.org/article/articles/electrical/arduino/general/ldr>)

- **Acelerômetro** – este componente é muito utilizado em projetos que precisam de orientação espacial, no sentido em que este tipo de sensor mede as oscilações nos eixos X, Y e Z. De fato um acelerômetro mede a aceleração de um corpo ou objeto em relação à gravidade gerando uma variação de tensão de aproximadamente 800mV/g para cada um dos eixos. Uma das vantagens deste dispositivo é que ele é muito pequeno, leve e consome pouca energia (400 μ A) quando está em operação. A sua ligação ao *Arduino* pode ser feita diretamente aos pinos analógicos e recebendo diretamente energia dele a 3,3V. A figura 7 ilustra a ligação de um acelerômetro modelo MMA7361, um dos mais comuns e baratos do mercado, nas portas do *Arduino*.

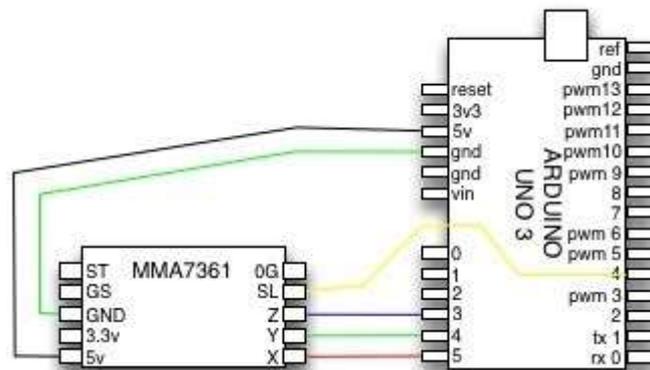


Figura 7 - Esquema de ligação do acelerômetro MMA7361 no Arduino.

2.5 Ferramentas de Programação

O modelo de API gráfica *Opensource* do *Arduino* foi concebido para trabalhar com qualquer tipo de linguagem de programação compatível com o seu Microcontrolador *Atmel*. Dependendo do sistema operacional usado é possível até mesmo controlar a placa através de linha de comando pelo console isto ocorre, por exemplo, no Linux ou Mac OS. Para facilitar o uso pelos desenvolvedores iniciantes, os criadores do projeto distribuem gratuitamente um ambiente de desenvolvimento de aplicações desenvolvido em *Java* e baseado na linguagem C para o *Arduino*. Por ser baseado em C, este ambiente de desenvolvimento possui a mesma sintaxe de programação além de algumas bibliotecas específicas para lidar, por exemplo, com os sensores e atuadores. Este ambiente se baseia na IDE do *Processing*, que é um ambiente de desenvolvimento criado em 2001 na forma de um “*sketchbook*” para o desenvolvimento de outras aplicações (PROCESSING, 2012).

A IDE usada como *Arduino* é bastante simples, de fácil utilização e se baseia no conceito de “*sketch*” que é uma espécie de protótipo para programação. Mas como existem muitas variações de placas e versões disponíveis, algumas configurações são necessárias para que a comunicação entre o programa e a interface ocorra de forma correta. As configurações mais importantes são: a porta de comunicação serial onde a placa está conectada e o tipo da placa usada. A interface gráfica do programa é dividida basicamente em quatro partes:

- A barra de menus – Contém os comandos básicos de configuração, manipulação de arquivos e gerenciamento da comunicação com a placa.

- A barra de ferramentas – possui um conjunto de poucos botões com algumas das funções mais usadas pelo programa como: abrir, fechar, criar projetos, enviar programas para a placa e abrir a janela do monitor serial.
- A área de programação – é a janela principal da interface onde será digitado o código do programa que irá ser enviado para a interface *Arduino*.
- A barra de status – nesta área são mostrados os avisos e informações referentes a execução do programa atual, mensagens de erros, de envio e recepção de dados por comunicação serial.

Esta interface possui recursos para comunicação serial diretamente com vários tipo de interface *Arduino*. Na comunicação serial, um bit de cada vez é enviado através de um canal de comunicação ou em um barramento de dados. Esta comunicação é síncrona, isto é, permite o envio e recepção de dados no formato *master-slave*, onde cada elemento aguarda o final da requisição para iniciar uma nova requisição ou enviar uma resposta. Como a comunicação síncrona é mais lenta, faz-se necessário o uso de uma banda de comunicação um pouco mais alta para compensar e evitar problemas nas leituras de dados vinda dos sensores. Nesta API de software, a velocidade de comunicação serial se dá a partir de 9600 até 19200 bits por segundo. O esquema da figura 8 ilustra a comunicação de dados entre a placa de circuito do *Arduino* e um computador.

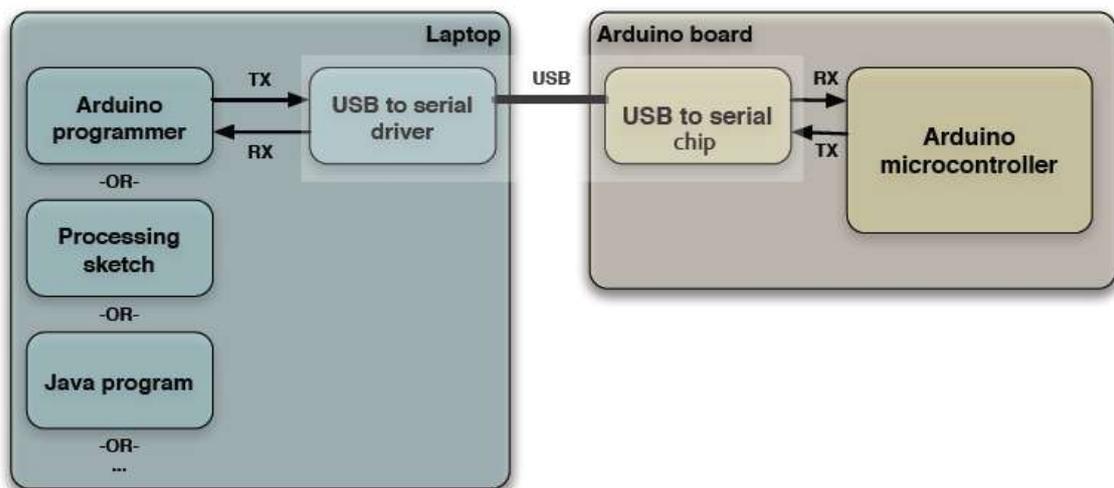


Figura 8 - Esquema representativo da comunicação de dados entre a Interface Arduino e um computador.

(http://todbot.com.s3.amazonaws.com/bionicarduino/bionic_arduino_class2.pdf)

Conforme visto no esquema da Figura 8.0 o programa *opensource* desenvolvido para o *Arduino* pode ser usado em conjunto com outras linguagens de programação como *Processing* ou Java e através de um *driver* padrão faz a conversão de dados no padrão USB para o padrão serial quando a placa da interface estiver conectada ao PC. Do outro lado, a conversão USB/Serial é feita por um circuito dedicado na própria placa e a comunicação interna dela já é feita naturalmente de forma serial. Devido a suas características físicas, a comunicação entre o microcontrolador e o canal USB/serial apresenta por si só uma latência da ordem de 2 milissegundos durante a comunicação de dados. Quando se faz uso de um programa em C para o envio de 1 byte, por exemplo, a latência sobe para 4,1 segundos e quando se utiliza um programa escrito em Java a latência sobe para aproximadamente 20 segundos. Estas taxas se mantêm em níveis mais ou menos constantes com blocos de até 128 bytes, acima deste valor e até alguns milhares de bytes a latência entre a linguagem C e o Java se igualam. Desta forma, quando se trabalha com projetos que utilizam mais de um sensor para controlar o movimento de vários atuadores, o atraso na comunicação serial se torna um complicador. Para evitar um atraso considerável entre a leitura dos sensores, o envio destes dados para o PC e o processamento destes dados é recomendado o uso de uma taxa de *bound rate* o mais alta possível para reduzir o tempo na transmissão dos dados.

2.6 A Robótica Educativa

Pode-se definir a robótica educativa como sendo um ambiente formado pelo computador, componentes eletrônicos, mecânicos e software, no qual o usuário integra estes elementos a fim de construir e programar dispositivos automatizados que podem ser usados para simular e testar conceitos de outras áreas do conhecimento, bem como coletar dados e enviá-los ao computador onde podem ser processados e/ou analisados (D'ABREU, 2002). Outra forma de utilização do computador com componentes eletrônicos e mecânicos é através da mecatrônica. Segundo Acar, da Universidade de Longhborough na Inglaterra (1996), a mecatrônica é uma filosofia de projeto, baseada na integração de microeletrônica, computação e controle em sistemas mecânicos, para a obtenção da melhor solução de projeto e produtos com um certo grau de “inteligência” e “flexibilidade”. Neste contexto definido por Acar, a mecatrônica se assemelha muito com a robótica, o que a torna também um elemento bastante usado nas aulas de robótica educativa na medida em que os projetos das aulas utilizam

conceitos de inteligência comportamental dos robôs, integração dos componentes e flexibilidade na utilização de vários tipos de materiais.

Como ferramenta de apoio ao aprendizado, a robótica atualmente é empregada em instituições de ensino como uma atividade lúdica que auxilia o entendimento de conceitos de matemática, física, ciências e noções de lógica de programação. Por ser uma atividade essencialmente de grupo, o uso da robótica em aulas também estimula a colaboração e trabalho em equipe dos aprendizes de uma mesma série ou de séries diferentes. Com o auxílio da robótica educativa, os educadores podem desenvolver projetos como veículos autônomos, braços mecânicos, giroscópios e uma infinidade de outros elementos de simulação para que os alunos compreendam, na prática, como funcionam leis fundamentais da física e também conceitos de mecânica e movimento.

A robótica educativa já é bem utilizada em alguns países. Holanda e Alemanha já usam em 100% das escolas públicas; países como Canadá e Estados Unidos já possuem um grande número de escolas que fazem uso de aulas de robótica. Na América Latina, além do Brasil, países como México e Peru estão iniciando com esta atividade (QUINTANILHA, 2008). Atualmente, não há estatísticas nacionais de quantas escolas efetivamente possuem aulas de robótica em seus currículos no Brasil, porém, a cada ano cresce o número de alunos que participam de competições de robótica como, por exemplo, a OBR (Olimpíada Brasileira de Robótica). Isto denota que há um crescente interesse por parte das instituições de ensino fundamental e médio em implantar em seus currículos aulas de robótica educativa. Um exemplo deste crescimento pode ser observado no estado da Paraíba onde cerca de trinta mil alunos estiveram envolvidos em projetos de robótica educativa no ano de 2008 (DAOUN, 2008). Além da iniciativa da OBR existem outras modalidades de competição de robótica, como a *First Lego League*, que incentiva e estimula a cooperação e a troca de experiência entre os alunos. Esta competição, a nível mundial, é patrocinada pela LEGO e por empresas parceiras com o objetivo de incentivar a competição entre alunos de todo o mundo sobre temas que influenciam a vida cotidiana, como a escassez de alimentos, a nano tecnologia, etc. Esta competição se baseia exclusivamente em kits de brinquedo da linha NXT da LEGO, e, portanto nem todas as instituições de ensino têm condições de participar. Além do investimento em materiais, existe também a necessidade da formação de professores e tutores especificamente para lidar com este tipo de material, além do material didático que é fundamental para que tanto os professores como os alunos obtenham um suporte mínimo para desenvolver os projetos práticos.

No caso específico deste tipo de prática no estado de Alagoas, observamos uma evolução gradativa nos últimos anos no engajamento das escolas, dos alunos e dos profissionais envolvidos nestas atividades. Esta evolução se mede através do aumento do número de escolas do estado que possuem algum tipo de atividade de robótica educativa e também da participação dos alunos destas escolas em competições de robótica. Um dos fatores que podemos destacar para este crescimento é a maior divulgação deste tipo de atividade e também o apoio dados pela Universidade Federal de Alagoas (UFAL) e Instituto Federal de Alagoas (IFAL).

2.7 Metodologias de Ensino de Robótica

O modelo adotado atualmente pela maioria das escolas é implantar a robótica com uma aula semanal de 50 minutos em turmas do ensino fundamental e médio. Existem aquelas que fazem parcerias com empresas que fornecem o material didático e os kits de peças e existem aquelas que optam por desenvolver um material didático próprio e utilizar material de sucata ou kits educacionais vendidos em lojas especializadas. Qualquer que seja o modelo adotado, o foco principal é levar o aluno a procurar soluções e resolver problemas por conta própria ou em grupos, sair da teoria para a prática e isto segue a linha da teoria construtivista, conforme (PIAGET, 1974) que afirma: o conhecimento é entendido como ação do sujeito com a realidade, e em ambientes de robótica educacional é simulado o que aconteceria em situações reais.

Em seu estágio atual a robótica educativa é uma ferramenta importante no incentivo ao aprendizado das ciências exatas, pois permite ao aprendiz ver na prática conceitos teóricos que não seriam tão facilmente assimilados. Entretanto, a falta de um modelo de educação tecnológica voltada especificamente para o ensino de robótica e até mesmo de informática nas escolas diminui a eficácia desta importante ferramenta, na medida em que cada escola transmite estes conhecimentos da maneira que acha mais adequada e se um aprendiz muda de escola, não há garantias de que haverá uma correta continuidade em seus estudos de robótica e informática ou se os conteúdos por ele aprendidos podem ser aproveitados em outra escola. A utilização das aulas de robótica em escolas da rede pública ainda é muito inferior às escolas da rede particular e isso ocorre por diversos fatores, entre eles:

- Falta de professores capacitados para dar início e continuidade a este tipo de atividade nas escolas públicas;
- Baixo interesse por parte dos gestores públicos em implementar projetos que envolvem ciência e tecnologia;
- O custo para aquisição de kits e material didático para as aulas de robótica educativa é um fator que atrapalha, mas não é determinante;
- Ainda há pouca divulgação e pouco engajamento com as comunidades locais, o que acaba não disseminando os ganhos adquiridos por este tipo de atividade.

O professor James McLurkin, da Universidade Rice, nos Estados Unidos, propõe: “um currículo onde cada estudante tenha seu próprio robô e possa estudar lições individuais e também trabalhar em equipe, usando seus robôs coletivamente em sistemas multi-robóticos” (INOVAÇÃO TECNOLÓGICA, 2011). Esta abordagem do professor McLurkin poderia ser uma iniciativa a nível mundial para se criar um modelo de robótica educativa e colaborativa na medida em que um padrão de robôs que seguem um mesmo modelo facilita a troca de conhecimento e integração entre alunos de várias partes do mundo através da Internet, por exemplo. Ao redor do mundo muitos pesquisadores procuram estabelecer modelos do ensino de robótica para crianças e adolescentes dentro de padrões mais ou menos universais, mas sem levar em conta as especificidades de cada país ou região. No Brasil, especificamente, podemos destacar duas soluções de aprendizagem de robótica que obtiveram um êxito de alcançar muitas regiões; o programa curricular Zoom que visa relacionar o desenvolvimento de competências ligadas à robótica com os conteúdos disciplinares de cada série ou nível, adequando-os às exigências do MEC (Ministério da Educação e Cultura) (LEGO EDUCACIONAL, 2011). O modelo Zoom faz uso de kits educacionais da linha LEGO NXT *for Schools* e de um material didático preparado por professores brasileiros. Um dos principais entraves ao uso do modelo Zoom é o seu alto custo em função dos kits serem importados e isto deixa de fora as escolas menores ou em regiões de menor poder aquisitivo.



Figura 9 - Material didático ZOOM e kit de peças da linha educativa LEGO

(www.edacom.com.br)

A outra solução tecnológica que se destaca é fornecida pela empresa Modelix que desenvolve soluções de robótica educativa para crianças e adultos. O modelo adotado pela Modelix, *Physical Computing*², é pioneiro no Brasil e atualmente é adotado por mais de 50 instituições de ensino com o uso de uma plataforma de software e hardware *open source*, o que reduz bastante o custo de implantação e manutenção deste tipo de aula já que as estruturas físicas usadas neste kit são de alumínio e plástico e produzidas no Brasil, como se pode ver na figura 10.0. Outra vantagem deste modelo de ensino é o uso de material de sucata em conjunto com os kits obtidos, o que permite aplicações mais diversificadas em sala de aula. Além disso, a empresa oferece também um material didático de apoio ao professor que inclui alguns exemplos de uso, projetos de robôs e dispositivos mecatrônicos em níveis variáveis de complexidade que varia do iniciante ao experiente (MODELIX ROBOTICS, 2012).

² Physical Computing foi definido por Dan O’Sullivan como sendo um conjunto de técnicas que permitem “sentir” e controlar o mundo físico através dos computadores.

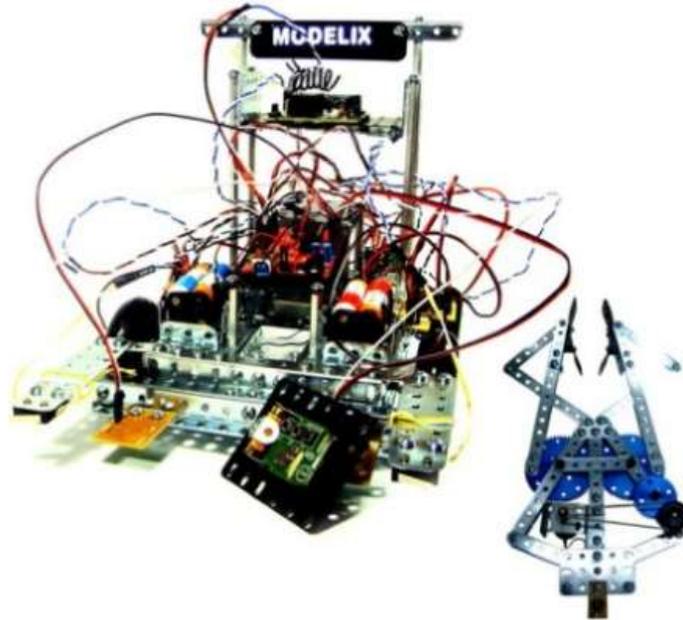


Figura 10 - Kit educacional Modelix

(<http://modelix.lojavirtualfc.com.br/prod,IDLoja,2284,Y,6500765222174,idproduto,3080657,linha-xt-xt75>)

Conforme foi visto neste capítulo, a robótica educativa ainda é uma atividade iniciante na realidade brasileira, mas com um potencial de crescimento bastante interessante à medida que mais empresas e mais escolas se engajam em desenvolver conteúdo e consumir este conteúdo, respectivamente. Foi visto também que as iniciativas atuais ainda não contam com políticas públicas e normas específicas voltadas para o ensino da robótica e de técnicas de programação nos Parâmetros Curriculares Nacionais, o que dificulta a padronização das aulas nas instituições de ensino públicas e privadas.

2.8 As Competições de Robótica

As atividades de robótica educativa em sala de aula ou nos laboratórios das escolas tem a finalidade primária de introduzir estes conhecimentos às crianças e jovens de forma mais lúdica possível através de experimentações práticas. Mas além deste caráter lúdico das aulas de robótica educativa muitas escolas tem procurado estimular seus alunos a realizarem competições entre os grupos de alunos de uma mesma escola e também de alunos de outras escolas na mesma região que também ministram aulas de robótica. Além das iniciativas locais existem também as competições de robótica a nível regional, nacional e internacional. Atualmente a maioria das escolas que implementam aulas de robótica educativa também

participa de eventos que englobam competições entre alunos, como a Olimpíada Brasileira de Robótica (OBR), por exemplo. Um dos objetivos deste tipo de evento é estimular a competição saudável e a socialização do conhecimento adquirido entre os alunos de escolas públicas e privadas de todo o país (SILVA & ALMEIDA, 2012). As olimpíadas científicas, a exemplo da OBR, são uma iniciativa para a popularização e difusão da ciência e tecnologia em especial para crianças e adolescentes, além de servir como um processo de atualização para professores e escolas, e que ocorre em muitos países do mundo. As competições de olimpíadas do conhecimento tiveram início no Brasil em 1978, mas apenas a partir de 2002, passaram a ser apoiadas pelo poder público através de editais. Especificamente a Olimpíada Brasileira de Robótica, que é um evento apoiado pelo CNPq, teve seu início oficialmente em 2008 e abrange dois tipos distintos de competição: a modalidade teórica, onde o aluno responde individualmente a uma prova de múltiplas escolhas sobre assuntos de conhecimentos gerais associada com a temática da robótica e a modalidade prática, que contempla equipes de até quatro alunos para resolver desafios como resgate de vítimas utilizando kits de robótica educativa e materiais diversos. Os principais objetivos desta olimpíada é estimular o interesse pela robótica a partir das crianças, estimulando o desenvolvimento de tecnologias e formando novos recursos humanos; e o outro objetivo importante é disseminar os conhecimentos adquiridos pelos jovens de forma que este conhecimento possa ser utilizado por outros jovens em todo o país (OBR, 2014).

Desde o começo da olimpíada no Brasil, uma parcela significativa de equipes utilizam os kits da linha *LEGO Educational* para desenvolver seus robôs de competição. Isto se observa especialmente nos primeiros anos do ensino fundamental onde os alunos estão tendo seus primeiros contatos com a robótica educativa. O uso destes kits facilita muito o processo de montagem dos robôs, porém, limita a utilização de peças, componentes e recursos de programação ao que está disponível no referido kit. Outros kits como o *Vex* e interfaces de controle como *Arduino* e *Raspberry pi* também vêm sendo utilizados nas últimas edições da OBR e com resultados bastante interessantes na medida em que eles deixam os alunos com mais opções de utilização de peças, sensores, atuadores e programas mais diversificados e com mais recursos. Com esta grande variedade de componentes utilizados, as regras da olimpíada tem sofrido algumas mudanças nos últimos anos para permitir que equipes utilizem materiais já prontos, mas que devem ter a programação feita exclusivamente pelos alunos, o que acaba desvirtuando os objetivos da competição, que são incentivar a descoberta do conhecimento por parte dos alunos e o processo de aprendizagem por conta própria.

Além da Olimpíada Brasileira de Robótica (OBR) existe ainda uma outra competição a nível mundial chamada *First Lego League* (FLL), que surgiu em 1998 através dos esforços do inventor e empreendedor Dean Kamen e do grupo LEGO. Os objetivos da FLL são promover o trabalho em equipe entre crianças de 9 a 16 anos focado na solução de problemas atuais utilizando a robótica, na forma de kits da linha LEGO (FLL, 2014). As competições da FLL ocorrem anualmente em etapas regionais, nacionais e mundiais, envolvendo crianças e jovens de escolas públicas e privadas, porém, como trabalham com kits específicos os custos para participação desta competição, mesmo a nível regional e nacional são muito altos, o que limita a participação apenas de uma pequena parcela de escolas.

Especialmente quanto ao uso de interfaces como *Arduino* e *Raspberry pi*, percebe-se que a maior dificuldade encontrada pelos alunos é se familiarizar com linguagens de programação como C e Java, pois estas exigem uma curva de aprendizado mais alta, mesmo contando com uma IDE um pouco mais amigável, assim como a integração de componentes, às vezes de fabricantes totalmente diferentes, e material fabricado pelos próprios alunos e professores. A participação dos alunos em competições pode ser considerada como a culminância das aulas de robótica. É uma maneira lúdica e estimulante de fazer com que os alunos ponham em prática a experiência adquirida em suas aulas de robótica durante o ano letivo.

3. Metodologias de Ensino de Robótica

Neste capítulo são abordadas as metodologias mais usadas atualmente para o ensino da robótica em instituições públicas e privadas com base nos relatos de profissionais de várias regiões do país, assim como as principais características de cada uma delas.

Durante os últimos anos têm sido testadas várias metodologias para o ensino de robótica nas escolas. A quase totalidade destas metodologias se baseiam nas teorias construtivistas de Piaget, Vigotsky e Papet onde o foco é o aprender fazendo ou o aprendiz é o ator principal do seu próprio aprendizado, tornando o professor um facilitador deste aprendizado. Os kits comerciais são atualmente a base da maioria das aulas de robótica e são poucos os materiais didáticos de apoio desenvolvidos com base nos parâmetros brasileiros (CÉSAR, 2008). De acordo com Jean Piaget, é importante que o aprendiz construa seu próprio conhecimento interagindo com o ambiente ao seu redor, proporcionando um maior desenvolvimento em sua aprendizagem (PIAGET, 1972).

As propostas mais utilizadas em escolas e outras instituições de ensino têm o seu foco na utilização de kits vendidos comercialmente tanto importados como nacionais com suporte de material didático impresso ou digital produzido pela própria instituição de ensino ou de parcerias feitas com representantes dos kits. Para trabalhar com a parte lógica da robótica, ou seja, a programação comportamental do robô, também são utilizados softwares específicos para cada kit adotado pela escola. Paralelo ao uso deste material e dependendo da forma como as aulas são ministradas, os próprios alunos acabam produzindo seu próprio conhecimento agregando novas soluções e disponibilizando esse conteúdo no site da escola, em blogs e em redes sociais, o que estimula a troca de conhecimento entre estudantes de robótica educativa do Brasil e de todo o mundo.

3.1 Os Parâmetros Curriculares Nacionais

Os Parâmetros Curriculares Nacionais são um referencial de qualidade para a educação no ensino fundamental em todo o país. Sua função é orientar e garantir a coerência dos investimentos no sistema educacional, socializando discussões, pesquisas e recomendações, subsidiando a participação de técnicos e professores brasileiros, principalmente daqueles que

se encontram mais isolados, com menor contato com a produção pedagógica atual (PARÂMETROS CURRICULARES NACIONAIS, 1997).

De acordo com o MEC (Ministério da Educação), uma das principais funções dos PCNs é servir como elemento catalisador de ações na busca de melhorias na qualidade da educação brasileira. A utilização destes parâmetros norteia as disposições básicas dos currículos ministrados no ensino fundamental e estabelece uma referência curricular comum para escolas públicas e privadas do país. Apesar de apresentar uma estrutura curricular completa, os Parâmetros Curriculares Nacionais são abertos e flexíveis, uma vez que, por sua natureza, exigem adaptações para a construção do currículo de uma Secretaria ou mesmo de uma escola. Também pela sua natureza, eles não se impõem como uma diretriz obrigatória: o que se pretende é que ocorram adaptações, por meio do diálogo, entre estes documentos e as práticas já existentes, desde as definições dos objetivos até as orientações didáticas para a manutenção de um todo coerente (PARÂMETROS CURRICULARES NACIONAIS, 1997).

Apesar do tema robótica educativa não constar explicitamente nos Parâmetros Curriculares Nacionais, até pelo fato de ser uma tecnologia relativamente nova a ser utilizada como ferramenta pedagógica, observa-se que a preocupação com a utilização das novas tecnologias na educação está presente, conforme o trecho:

“É indiscutível a necessidade crescente do uso de computadores pelos alunos como instrumento de aprendizagem escolar, para que possam estar atualizados em relação às novas tecnologias da informação e se instrumentalizarem para as demandas sociais presentes e futuras.” (PARÂMETROS CURRICULARES NACIONAIS, 1997).

Em seu trecho que fala sobre as ciências exatas e da terra, os PCNs listam as necessidades de se estabelecer uma interdisciplinaridade no sentido de explorar as várias formas de interagir os conteúdos e fazer com que o aprendizado seja melhor absorvido pelo aluno. Neste contexto, a robótica tanto quanto a informática, pode ser uma ferramenta valiosa na absorção do conhecimento devido a sua característica lúdica e prática além de servir como laboratório de experimentação para conceitos teóricos explorados nas disciplinas de matemática, ciências, física, biologia e outras.

Como as aulas de robótica educativa podem ser ministradas tanto em escolas públicas como em escolas privadas e estas escolas podem estar em qualquer parte do país, a adequação

da metodologia de ensino aos moldes dos PCNs é importante para tentar garantir que os conteúdos ministrados sejam adequados para cada nível escolar, além de permitir que estes conteúdos possam ser compartilhados por outras escolas sem que precise haver muitas mudanças estruturais nestes conteúdos para adequar a realidade de cada região. A utilização destas novas tecnologias por uma quantidade grande de escolas espalhadas pelo Brasil podem ainda servir de incentivo para melhorias e aperfeiçoamentos nas próximas atualizações dos próprios PCNs.

3.2 Lego Zoom

A utilização de kits de robótica desenvolvidos em parceria pela empresa Lego e o Media Lab do MIT começou por volta de 1998 com o surgimento do Kit Lego Mindstorms RCX. Este kit agregava as peças tradicionais da Lego com motores, sensores e uma interface de controle baseada em um microcontrolador Hitachi H8.

Devido ao grande sucesso desta plataforma, muitas escolas adotaram o kit educacional da Lego para aulas de robótica educativa. No Brasil esta disseminação ocorreu principalmente no eixo sul-sudeste do país, mas com boa participação também de estados do centro-oeste e nordeste. A empresa Edacom, que desde o ano 2000 representa os kits Lego no país, começou a desenvolver uma metodologia de ensino de robótica baseado nestes kits e focado nos seguintes conceitos: aprendendo a fazer, aprendendo a ser, aprendendo a conviver, aprendendo a conhecer e aprendendo a agir. (LEGO EDUCACIONAL, 2011).

O modelo Zoom se baseia, portanto, em incorporar as diversas habilidades e capacidades psicomotoras por meio do desenvolvimento de projetos sugeridos nos fascículos que são vendidos às instituições de ensino que contratam estes serviços. Este material didático de apoio ao professor tem sido elaborado por equipes multidisciplinares que se baseiam nos preceitos dos Parâmetros Curriculares Nacionais ilustrados por meio de situações-problema. Atualmente o modelo de ensino Zoom se divide em dois eixos centrais:

- Curricular – Abrange desde a educação infantil até o ensino médio, com estrutura pedagógica e material de apoio adequada para cada faixa etária.

- Extracurricular – É mais voltado para o ensino de elementos básicos de robótica, liderança e empreendedorismo, baseado em desafios estimulantes para jovens entre 10 e 15 anos (LEGO ZOOM, 2011).

Além de introduzir os conceitos da robótica para estudantes das séries iniciais de forma mais didática, a proposta do Zoom é também auxiliar as escolas que pretendem participar da *First Lego League*, competição à nível mundial que a cada ano propõe desafios nas várias áreas do conhecimento tecnológico utilizando peças da linha Lego e estimulando a participação e socialização de alunos de todo o mundo.

As Vantagens deste modelo são focadas nas situações-problema criadas por seus desenvolvedores, o material didático ricamente ilustrado e a possibilidade de apoios especializados na própria escola. As principais desvantagens são a questão do custo altíssimo, pois é necessário que a escola firme contrato com a Edacom e também o foco centrado apenas nos kits educacionais da Lego, deixando pouca margem para utilização de outros kits e materiais de menor custo.

3.3 Modelix

A plataforma de ensino da robótica desenvolvido pela empresa *Modelix Robotics* abrange uma solução composta por hardware e software em grande parte de baixo custo. A empresa trabalha com o princípio *open source* (plataforma aberta), apoiando a disseminação do conhecimento para a criação de alta tecnologia e não apenas à sua utilização.

Atuando no mercado de robótica educativa desde 2007, a *Modelix Robotics* atualmente difunde a robótica em escolas de todo o Brasil fazendo uso de materiais em plástico ABS e alumínio produzidos localmente e faz uso da interface de controle *Arduino* para integrar seus sensores e atuadores, em sua maioria montados ou feitos no país. Por utilizar materiais intercambiáveis de fácil aquisição esta plataforma usada pela *Modelix* é interessante, pois as próprias escolas podem produzir suas peças usando as mesmas especificações estruturais. Os kits disponibilizados pela empresa possuem material didático de apoio em português e um software chamado *Modelix System* adaptado para programar os robôs usando uma interface intuitiva e em português, onde crianças a partir dos seis anos de idade podem usar, além de

fornecer capacitação *in-loco* para professores e facilitadores das escolas. A tabela 2.0 ilustra as diferenças entre as duas versões atuais deste software.

Tabela 2 – Diferenças entre as versões Starter e Pro do software Modelix System.

Recursos	Modelix System Starter	Modelix System Pro
Licença	Pessoa Física	Uso Educacional (empresas)
Validade da licença	1 ano (paga)	1 ano (paga)
Número de instalações	1 máquina fixa	4 máquinas fixas
Plataforma	Windows XP ou superior	Windows XP ou superior Linux sob pedido especial
Comunicação	USB	USB
Idioma	Português	Português
Interface usada	Arduino	Arduino
Simulador de cenários	Não	Sim
Material didático	Não	Sim
Modo robô autônomo	Sim	Sim

Fonte: <http://www.modelix.com.br/modelixsystem.pdf>

Uma das grandes desvantagens desta plataforma é o alto custo de implantação nas escolas visto que requer grandes investimentos em kits, software e material didático. Porém, com o tempo e a utilização de material de sucata ou criado na própria escola a tendência é uma redução dos custos.

3.4 RoboEduc

De acordo com o site dos criadores, *RoboEduc* foi desenvolvido por pesquisadores da Universidade Federal do Rio Grande do Norte nas áreas de educação e computação, com foco na inovação da educação tecnológica, comprometida com o aprendizado de alunos desde o ensino infantil até o ensino superior, criando valor para a sociedade (ROBOEDUC, 2012).

A proposta desta metodologia de ensino de robótica educativa leva em conta uma solução de software chamado sistema *RoboEduc* em conjunto com oficinas de capacitação para os professores, material didático elaborado durante as oficinas e a utilização de kits de robótica da linha Lego *Mindstorms*. O foco desta metodologia é, portanto, resgatar o “aprender brincando”. Como o projeto teve seu início no âmbito acadêmico, seu foco atual é atender as escolas públicas da região onde foi implantado, o que é um grande incentivo para que outros colégios também adotem este modelo. Apesar de estar em estágio ainda inicial, o modelo proposto pelo *RoboEduc* tem seu mérito no suporte da academia e na interação mais estreita com as escolas públicas, mas atualmente sua maior desvantagem é não utilizar kits de plataforma aberta, o que dificulta a utilização de materiais alternativos e de baixo custo.

3.5 O PNCA Alfa

De acordo com o site oficial, a PNCA Alfa Robótica e Eletrônica, situada no interior paulista, é uma empresa especializada no desenvolvimento de dispositivos eletrônicos para serem utilizados no mercado educacional. Esta iniciativa já se encontra em uso por aproximadamente 35 mil alunos em várias regiões do país entre escolas públicas e privadas (PNCA ALFA, 2012).

Da mesma forma que as outras plataformas, o PNCA Alfa fornece kits educativos baseados no trabalho em equipe, material didático baseado nos PCNs e uma linguagem de programação em português chamada LEGAL, semelhante à linguagem natural e um simulador de ambiente 2D baseado na *Web*. Este modelo também possui um kit físico composto por peças em alumínio, plástico e um módulo de controle programável MC 2.5 baseado no microcontrolador Atmega32 e possui comunicação USB, entradas analógicas e digitais para atuadores e sensores que podem ser aqueles que acompanham o próprio kit, de outros kits e ainda o uso de atuadores e sensores retirados de material de sucata (PNCA ALFA, 2012).

Tanto a sua proposta pedagógica quanto na aparência física dos kits, a plataforma do PNCA Alfa e do *Modelix* são praticamente idênticos, o que varia é a questão da interface e o software usados para controlar os robôs (MÓDULO MC, 2012). Além disso, o modelo PNCA faz uso de uma linguagem gratuita e o custo de implantação em uma instituição de ensino é

bem menor do que o *Modelix*, porém com resultados similares, de acordo com os casos disponíveis nos sites das empresas. Uma outra diferença deste modelo é o uso de um ambiente de simulação onde os estudantes podem ver o comportamento do seu robô através de uma página no navegador da internet, permitindo ao aluno fazer pequenas experiências em casa. Este simulador imita as principais características do robô físico, mas está disponível apenas para escolas e alunos associados ao projeto, o que caracteriza uma desvantagem por não socializar com a comunidade as simulações geradas pelos alunos.

3.6 Plataforma Vex

A plataforma de robótica Vex é baseada em um grande conjunto de materiais em metal e plástico resistente usado por escolas, centros de pesquisa e instituições de todo o mundo. Por ser uma empresa multinacional, a *Vex Robotics* possui uma grande variedade de produtos voltados e especialmente desenhados para competições de robótica, para a área educacional e comercial. Uma das grandes desvantagens deste modelo é o altíssimo custo de implantação e manutenção, pois todo o material é importado e não haver um representante único no Brasil (VEX ROBOTICS, 2012).

Atualmente este modelo é muito utilizado por estudantes que participam de competições a nível mundial como a *First Lego League* e a própria *Vex Robotics Competition*, entre outras. Os kits da linha Vex seguem o mesmo modelo dos kits mencionados anteriormente como: incentivar o trabalho em equipe, utilização de material impresso de apoio (em inglês), possibilidade de uso de material de sucata para complementar o kit e utilização de uma interface de controle baseada no microcontrolador Cortex-M3 da ARM. Algumas das desvantagens da utilização deste modelo no Brasil são: o alto custo de importação deste material, podendo chegar na faixa dos milhares de reais, a falta de um representante que dê suporte a este modelo no país e o fato do material de apoio não ser baseado nos padrões curriculares brasileiros e não é escrito em português, o que dificulta o seu uso, por exemplo, nas escolas públicas do país.

Neste capítulo foram abordados os principais aspectos relacionados aos modelos de aula de robótica educativa mais usados atualmente no país. Estas metodologias são usadas por escolas para o ensino de robótica e também para preparar equipes de alunos para participar de competições nacionais e internacionais na área da robótica.

4. O Modelo Proposto de Baixo Custo Usando a Plataforma Arduino

Neste capítulo será mostrado o modelo proposto para aulas de robótica educativa que faz uso de materiais de baixo custo, material didático baseado nos parâmetros curriculares Nacionais e na utilização de ferramentas on-line para socializar os conteúdos desenvolvidos nas atividades práticas tanto para os alunos participantes do projeto como para aqueles interessados em implantar esta metodologia em sua instituição de ensino.

Nos capítulos anteriores foram abordados os conceitos de robótica, do aprendizado deste conhecimento pelas crianças e adolescentes e os aspectos inerentes à utilização de alguns modelos tanto de kits educacionais como de metodologias de ensino da robótica no ensino brasileiro. Estes modelos possuem suas particularidades e semelhanças, onde cada um oferece vantagens e desvantagens com relação ao uso da tecnologia, da metodologia e de sua afinidade com os parâmetros do ensino fundamental e médio proposto pelo Ministério da Educação (MEC). Em sua totalidade os Parâmetros Curriculares Nacionais não são claros com relação ao uso da tecnologia da informação na educação. São citados apenas os benefícios que a tecnologia pode trazer ao agregar recursos audiovisuais na educação. A informática é citada apenas como sendo uma ferramenta pedagógica de apoio não havendo menção ao seu uso de uma forma específica, nem sugerindo conteúdos a serem incorporados às aulas em laboratório. Isto deixa claro que a intenção é que cada região do país faça uso da tecnologia da forma como achar melhor. Isto abre um leque amplo, mas deixa a lacuna para que não exista, por enquanto, um padrão a ser adotado.

Se a informática nas escolas é um tema pouco abordado e conseqüentemente pouco explorado nos PCNs, a robótica educativa por ser ainda mais recente nem sequer é mencionada nos textos, visto que sua utilização nas escolas remonta a menos de dez anos. Desta forma, a sua interação aos currículos escolares se dá de uma forma ainda experimental onde cada escola procura se adequar ao seu uso da melhor forma possível e muitas vezes inclusive, trocando experiências e aproveitando os casos de sucesso usados por outras escolas.

Em sua dissertação de mestrado (CURCIO, 2008), procura demonstrar uma metodologia muito interessante de robótica educacional de baixo custo utilizando em sua maior parte material de sucata e onde podemos notar a sua ênfase em uma base de

treinamento para os professores envolvidos neste tipo de atividade. Além disso, Curcio faz uso de vários materiais alternativos como garrafas plásticas, pedaços de madeira e metal, papelão, entre outros associados com motores e sensores retirados de brinquedos usados e outros equipamentos eletrônicos de baixo custo. Os exemplos citados por Curcio nesta dissertação são voltados para escolas públicas. Sem a utilização de uma interface de controle para que seus robôs tenham alguma inteligência comportamental, é demonstrado apenas um módulo usado para acionar motores através da porta paralela do computador, o que nos dias de hoje não seria muito útil, pois a maioria das aplicações faz uso da porta USB ou da porta ETHERNET para comunicação de dados e a proposta de software utilizada neste modelo é o SUPERLOGO, uma ferramenta bastante limitada em termos de recursos e que não é mais tão utilizada nos ambientes de ensino. Mas apesar disso, são exemplos bastante válidos para um modelo básico para séries iniciais e com pouca experiência com este tipo de atividade.

Com base nas experiências adquiridas em quase dez anos de aulas práticas de robótica educativa em escolas da cidade de Maceió, estado de Alagoas, observamos que para aplicar uma metodologia realmente eficiente em termos de aprendizado e atraente tanto para o aluno, como para o professor, é fundamental atingir três objetivos básicos:

- Um material didático de fácil entendimento de acordo com a faixa etária dos alunos envolvidos nas aulas.
- Um conjunto de componentes eletrônicos e mecânicos que estimulem a criatividade dos alunos e ao mesmo tempo consiga criar robôs e mecanismos funcionais e estimulantes da criatividade.
- O software usado para a programação da inteligência comportamental do robô e os softwares de apoio para publicação e disseminação dos projetos serem de fácil aprendizado e de preferência possam ser usados livremente e distribuídos também livremente.

Neste contexto, o método para ensino de robótica educativa proposto neste trabalho baseia-se na aplicação dos conceitos teóricos abordados nos capítulos 3 e 5, e levando em conta também a utilização de materiais recicláveis, de baixo custo ou ainda encontrados facilmente no comércio local. A proposta aqui explanada também permite utilizar formulários e questionários para que o professor e os responsáveis pelo projeto possam avaliar os resultados obtidos durante e após as aulas de modo que ajustes possam ser feitos para melhorar ainda mais os resultados almejados. Além, disso, como o aluno é também o

protagonista do seu aprendizado, é mostrada também uma maneira de divulgar e colaborar os projetos práticos feitos em sala de aula em um site na Web para que os alunos possam rever as atividades práticas realizadas durante as aulas, acrescentar novas ideias e ainda compartilhar com estudantes do Brasil e de todo o mundo seus projetos além de poder trocar informações via blog e rede social. Um estudo comparativo foi feito utilizando dois grupos de alunos: um dos grupos formado por alunos do primeiro ano do ensino médio que já haviam feito aulas de robótica educativa desde o quarto ano até o sétimo ano do ensino fundamental utilizando kits LEGO MINDSTORMS e um grupo de alunos também do primeiro ano do ensino médio, mas que não tiveram aula de robótica nos anos anteriores.

Os resultados indicaram que o uso desta metodologia se aplica melhor para alunos que já tenham alguma experiência prévia com kits de robótica. Além disso, o nível de maturidade dos alunos do ensino médio se adéqua muito mais a utilização da interface *Arduino* e dos demais componentes utilizados neste projeto.

4.1 O Hardware Usado no Modelo Proposto

O modelo proposto neste trabalho tem como base a interface de controle *Arduino*, além de componentes elétricos, eletrônicos e mecânicos de baixo custo, fácil aquisição e em muitos casos matérias recicláveis como isopor, papelão, pedaços de madeira leve, garrafas Pet, dentre outros.

A figura 11 a seguir ilustra um kit básico usado para as aulas práticas. Neste kit nós temos os elementos listados em detalhes na tabela 3.0 suficientes para que um grupo de até cinco alunos possam criar seus projetos.



Figura 11 - Kit básico de peças para as aulas de robótica educativa usando a interface Arduino.

No kit da figura anterior, são mostrados apenas os componentes elétricos e eletrônicos. Além deles é necessário o uso outros materiais como chaves do tipo Philips e de fenda, alicate de corte e de bico, suporte para quatro ou seis pilhas do tipo AA e em alguns projetos também pode ser necessário o uso de ferro de solda, porcas e parafusos para prender as estruturas físicas usadas pelos alunos nos projetos propostos, bem como é preciso ter um computador *desktop* ou *notebook* equipado com sistema operacional *Windows* ou *Linux*, com pelo menos uma porta USB livre e conexão com a Internet para cada grupo de quatro ou cinco alunos.

Tabela 3.0 – Lista de componentes usados no kit básico para aulas práticas.

Componente	Quantidade	Valor Aproximado em R\$
Arduino UNO ou equivalente	01	85,00
Protoboard	01	20,00
Servomotor de rotação contínua	02	100,00
Micro servomotor de posição	01	16,00
Sensor Ultrasônico HC-SR04	02	136,00
Sensor de luz LDR	02	6,00
Resistores de 220 Ohms	06	3,00
LEDs de várias cores	06	16,00
Fios elétricos	20	10,00
Barra de pinos	01	2,00
Suporte para 06 pilhas AA	01	12,00

Além dos componentes elétricos e eletrônicos, são utilizados também vários elementos físicos e mecânicos usados para a estrutura física dos robôs. Uma opção de baixo custo é usar as bases móveis com rodas e esteiras das marcas *Tamiya* e Pololu que são facilmente integradas com outros componentes e podem ser encontrados em sites de componentes eletrônicos e vendidos no Brasil. A figura 12 mostra alguns exemplos de bases móveis deste tipo.



Figura 12 - Peças físicas das marcas Pololu e Tamiya para as aulas de robótica educativa

(www.pololu.com).

Neste modelo usamos peças de alumínio, papelão rígido e engrenagens e polias plásticas retiradas de drives de CD/DVD quebrados. A figura 13 mostra alguns componentes físicos em detalhes usados para criar as estruturas físicas dos robôs e mecanismos usados nos projetos, bem como materiais de sucata como tampas de garrafão de água mineral usados como rodas, suporte de motor feito com papelão e garra feita com palitos de madeira.

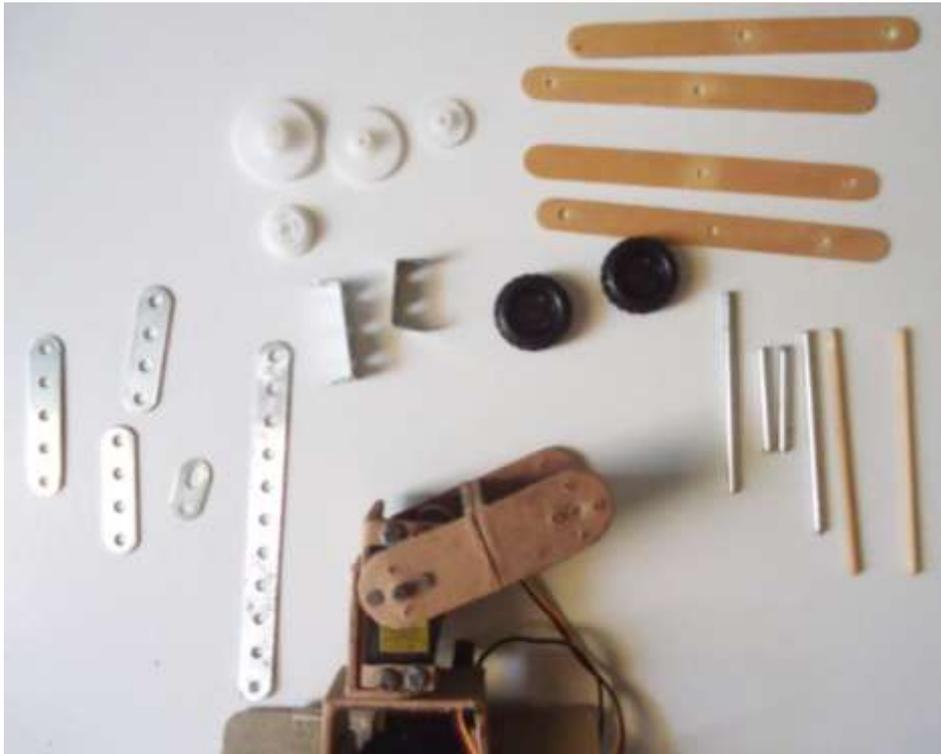


Figura 13 - Peças alternativas e de sucata usadas para as aulas de robótica educativa.

4.2 Material de Apoio ao Professor

Para que as aulas de robótica tenham um maior aproveitamento e também para que haja uma continuidade no projeto de qualquer instituição de ensino mesmo que haja mudanças no corpo docente, é preciso haver uma memória, ou ainda uma biblioteca de apoio ao professor com o mínimo de informações úteis para quem está entrando agora neste tipo de atividade e também para lembrar alguma informação útil para quem já leciona robótica há mais tempo. Este material de apoio não é um manual estático; ele é dinâmico no sentido em que a cada nova descoberta ou pesquisa feita pelo professor, esta informação é agregada ao material já

existente. Neste contexto, a elaboração de um material de apoio para o professor é algo fundamental. Mas é importante salientar que o material didático de apoio ao professor não substitui um treinamento adequado e uma capacitação complementar para o professor, principalmente quanto às habilidades lógicas e de programação do professor, algo fundamental para o desenvolvimento do comportamento dos robôs e este conhecimento deve ser transmitido ao aluno de forma clara e objetiva.

Os itens que devem ser considerados como o mínimo para um material de apoio ao professor nas aulas de robótica seriam:

- Informações técnicas como memória, velocidade de processamento, quantidade de portas analógicas e digitais, limites de tensão e corrente e formas de conexão ao PC sobre a interface usada no projeto.
- Informações técnicas e formas de ligação na interface de controle dos sensores e atuadores, assim como suas especificações elétricas, polarização e usabilidade.
- Exemplos de uso dos componentes mecânicos como engrenagens, polias, estruturas físicas e meios de tração, assim como a forma de conectar estes componentes uns aos outros e montar os componentes elétricos e eletrônicos neles.
- Alguns conceitos básicos de física, matemática, mecânica e eletrônica para auxiliar o professor nas demonstrações práticas. Estes conceitos se concentram mais nas leis de Ohm, circuitos abertos e fechados, associação de resistores, utilização de pilhas, baterias e noções de circuito aberto e fechado.

Os trechos do material didático de apoio ao professor mostrados na figura 14 são considerados um conteúdo mínimo que pode ser melhorado com o uso por parte do professor que agrega mais conteúdo na medida em que as aulas ocorrem. A utilização deste material por parte do professor irá auxiliar principalmente aqueles que tem pouca ou nenhuma experiência nas aulas de robótica educativa servindo também como um material de treinamento pois agrega muitos conceitos básicos. Para os professores mais experientes, o material de apoio serve como guia de referência para lembrar algum conteúdo esquecido e também para que o professor mais experiente possa agregar mais conteúdo às suas aulas. Os anexos no final deste trabalho trazem de forma detalhada como o material didático para o professor foi estruturado.

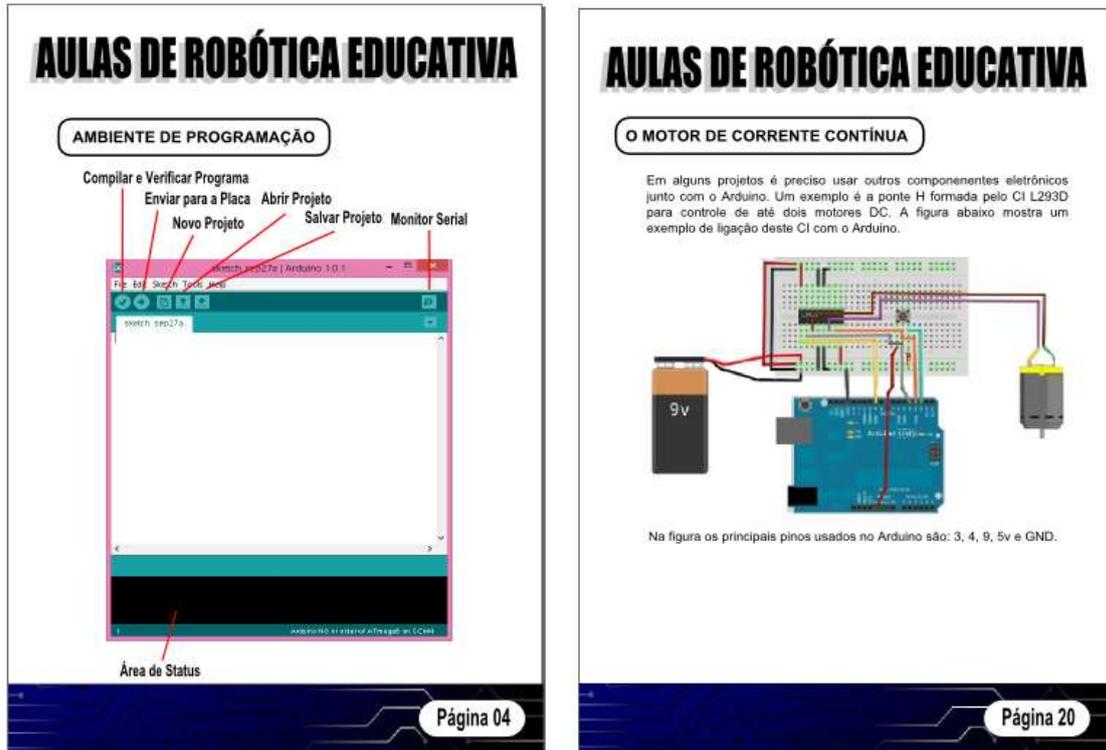


Figura 14 - Trechos do material de apoio ao professor.

Fonte: O autor

O exemplo mostrado na figura 14.0 mostra tanto o ambiente de programação do *Arduino* com descrição de suas funcionalidades como o esquema eletrônico e físico para montagem de um projeto.

4.3 Material Didático para o Aluno

O material didático a ser disponibilizado ao aluno é um dos componentes mais importantes deste modelo de aula de robótica educativa na medida em que é através deste material didático que o aluno pode experimentar na prática os projetos propostos pelo professor, além dele próprio aprimorar ou modificar os projetos contidos no material didático através de sugestões de mudanças nos projetos existentes ou mesmo propondo novas ideias que podem ser posteriormente adicionadas ao material didático existente. A elaboração deste material segue alguns elementos básicos, tais como:

- Os projetos mostrados no material didático devem seguir um grau de dificuldade crescente, ou seja, deve-se começar com projetos mais simples e ir aumentando o grau de dificuldade gradativamente.
- Associado a cada projeto deve haver também um exemplo de programação para dar a ideia básica ao aluno de como o robô irá funcionar.
- Cada projeto deve conter também uma lista de peças com uma descrição sucinta de cada uma.
- Os projetos devem conter partes físicas e componentes eletrônicos de baixo custo e se possível aproveitado de sucatas e equipamentos fora de uso.
- Ao final de cada projeto deve haver uma área destinada para que o aluno descreva de forma resumida o que aprendeu com este projeto, as dificuldades que encontrou e alguma possível sugestão de melhorias no projeto.

O material didático proposto neste projeto foi idealizado para ser trabalhado por grupos de no máximo quatro alunos, onde o professor pode atribuir tarefas a cada aluno do grupo como, por exemplo, programador, montador, organizador, etc. É importante que a cada novo projeto ou mesmo durante a execução de projetos mais complexos haja um rodízio nas funções de determinado membro do grupo para que todos vivenciem as dificuldades de cada etapa do projeto. Cada projeto possui uma sequência na qual o aluno primeiramente é apresentado às ideias básicas como, por exemplo, o funcionamento de um servo motor, de um sensor de distância, de um sensor de luz, etc. Após adquirir este conhecimento, o material didático sugere um projeto passo a passo de como montar um robô ou dispositivo mecânico que faça uso dos conhecimentos mostrados. As imagens das etapas da montagem procuram mostrar os detalhes, mas cabe ao professor conhecer muito bem cada etapa para auxiliar o aluno quando ele tiver alguma dúvida. Após concluir as etapas de montagem, cada grupo deve observar com atenção e verificar se o seu projeto está de acordo com o projeto mostrado no material didático. A última etapa antes de concluir o projeto é a programação sugerida no material didático, a qual cada grupo deve digitar na interface gráfica do *Arduino*, fazer o *download* para a interface e verificar seu funcionamento, fazendo mudanças e correções de erros, caso seja necessário. A programação contida no material didático do aluno após cada projeto é uma maneira de fazer, mas o professor deve incentivar os alunos a modificar a programação e observar os resultados, fazer melhorias ou ainda criar sua própria programação. Ao final de cada projeto existe uma página no material didático onde o aluno

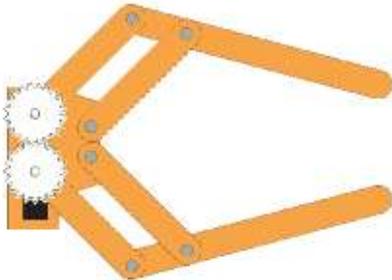
irá fazer um pequeno relatório de tudo que aprendeu, suas dificuldades e anotar quaisquer mudanças feitas na programação.

A sequência de projetos sugeridos segue um grau de dificuldade crescente além de incluir desde os componentes eletrônicos e mecânicos básicos até os mais complexos e que necessitam de conhecimentos e competências mais aprofundados por parte dos alunos, como noções de eletrônica e mecânica. É importante notar que a forma como as informações e as figuras estão dispostas no material didático influencia diretamente no grau de aprendizado, pois a clareza delas faz com que o aluno consiga aprender corretamente a forma de montar e testar o projeto. A figura 15 ilustra partes de um projeto contido no material didático disponibilizado para o aluno. Nos anexos deste trabalho está disponível o material didático completo para o aluno.

AULAS DE ROBÓTICA EDUCATIVA

GARRA COM SERVO MOTOR

A atividade proposta neste projeto tem como objetivo criar uma garra que abre e fecha para segurar pequenos objetos feita com pedaços de madeira leve utilizando apenas um servo motor pequeno e duas engrenagens. O esquema da figura abaixo ilustra como este tipo de mecanismo pode funcionar.



LISTA DE MATERIAL:

- 01 Servo motor pequeno
- 08 Pedaços de madeira leve
- 01 Palito de churrasco
- 02 Engrenagens médias
- 06 Parafusos pequenos
- 06 Porcas pequenas
- 01 Interface Arduino

Página 15

Figura 15 - Trechos do material didático disponibilizado para o aluno.

Os projetos atualmente disponibilizados no material didático proposto neste estudo contemplam atividades que utilizam dispositivos como sensores de distância, de luz, de pressão, servo motores, motores de corrente contínua, engrenagens de vários tipos e tamanhos, estruturas de apoio com uso de material de baixo custo e objetos do dia a dia como palito de picolé, papelão e isopor.

4.4 Iniciando seu Próprio Projeto

Quando uma instituição de ensino planeja iniciar um projeto de robótica educativa é altamente recomendado que este planejamento inclua alguns passos para que o projeto venha a trazer os resultados esperados e não traga frustração para os alunos, professores e gastos excessivos para a escola. As etapas a seguir são recomendadas para qualquer tipo de instituição de ensino e com o mínimo de uma aula semanal de pelo menos 50 minutos cada:

- O primeiro passo é utilizar o material didático mostrado neste projeto ou utilizar como base para que a escola elabore o seu próprio material didático que será disponibilizado para o professor e para os alunos. Caso a escola já tenha um material deste tipo esta etapa pode ser desconsiderada.
- O próximo passo é definir a lista de materiais e componentes que serão utilizados para as aulas. Estes materiais representam o maior custo para a escola e baseado na quantidade de peças por aluno, se consegue uma grande otimização e redução de custos se for levada em conta a relação mostrada na equação 1, com as seguintes variáveis.

N_{kt} : número mínimo de kits de robótica.

N_{al} : número total de alunos por turma.

N_{sl} : número de aulas por semana.

N_{gr} : número de alunos por grupo.

$$N_{kt} = \frac{N_{al} \times N_{sl}}{N_{gr} \times 2} \quad \text{equação 1}$$

- O passo seguinte é realizar uma oficina ou workshop com os profissionais (professores, coordenadores de área, direção da escola, etc.) que serão responsáveis pelas aulas de robótica. Neste momento serão apresentados os objetivos do projeto, o material didático a ser utilizado, a metodologia e interdisciplinaridade e os componentes físicos
- Após estes passos iniciais e quando as aulas já tenham tido início, o professor primeiramente deve mostrar aos alunos os objetivos do projeto de robótica, fazer uma demonstração dos cuidados com as peças e o manuseio dos componentes, explicar detalhadamente como funciona a robótica e como será feita a distribuição de componentes para cada grupo de alunos. Este passo geralmente ocorre em duas aulas.
- No passo seguinte com a utilização do material didático proposto, o professor deve sempre fazer uma associação entre o projeto que será montado pelos alunos em cada aula com os conhecimentos que ele já viu ou está prestes a ver nas aulas regulares de outras disciplinas.
- E por fim, no último passo, o professor deve apresentar o projeto de cada aula, seguir as instruções de montagem auxiliando os alunos quando necessário, testando a integridade da montagem, explicando como será feita a programação, testando o funcionamento final do robô e ao final de cada projeto incentivar os alunos de cada grupo a escrever um pequeno relatório enumerando o que aprendeu e as dificuldades encontradas durante a realização do projeto. Estes procedimentos devem ser repetidos em cada aula que ocorra um novo projeto. Em alguns projetos mais complexos, pode se estender para duas ou três aulas a mais.

4.5 A IDE WebSockets

Desde a sua concepção que a web se baseia no paradigma de solicitação/resposta através de HTTP (*Hyper Text Transfer Protocol*). Sempre que um dispositivo cliente solicita uma informação, esta solicitação é repassada a um dispositivo servidor que irá processar a solicitação e enviar uma resposta. Este recurso de requisição/resposta pode ser implementado de várias formas dependendo da tecnologia *web* a ser utilizada, mas em todos eles os mecanismos de envio e recebimento de mensagens são bem semelhantes. Com o avanço das tecnologias e a atualização dos navegadores, muitos novos recursos estão surgindo; um dos

mais promissores atualmente é o HTML5, que permite estender as funcionalidades das páginas HTML com poderosos recursos dispensando muitas vezes a utilização de outras tecnologias complementares.

O W3C, consórcio gestor responsável por padronizar as especificações usadas pela maioria dos navegadores web possui uma lista grande dos novos recursos HTML5 em funcionamento e alguns ainda em fase de implementação e ainda não são suportados por todos os navegadores. Dentre estas especificações abordaremos o *WebSocket*, uma API que possibilita às páginas *web* usar o protocolo *WebSocket*, o qual é usado para comunicação via *socket* bidirecional com um host remoto. (W3C, 2013).

Deve-se levar em conta que esta especificação só pode ser utilizada nas versões mais recentes dos navegadores web que suportam HTML5. Uma das grandes vantagens do *WebSocket* é detectar a presença de um servidor de *Proxy* e gerar um túnel para negociar acesso através do *Proxy* requisitando a abertura de uma conexão TCP/IP para um *host* e porta específicos. O construtor da classe *WebSocket* usa apenas dois argumentos: *url* e *protocols* para especificar o endereço do host junto com a porta a ser usada e opcionalmente uma lista de subprotocolos. Uma vez que o construtor entra em execução, ele obtém as informações necessárias para estabelecer a comunicação. Algumas rotinas de validação e tratamento de erros podem ser adicionadas ao código para tornar a página web mais confiável. A utilização deste *WebSocket* é bastante simples e pode ser feita através de *JavaScript* diretamente em uma página *web* comum, que posteriormente será hospedada em um provedor de acesso qualquer. O Apêndice 3 no final deste trabalho contém os códigos usados para implementar uma solução comentada de página *web* com troca de mensagens com a interface *Arduino*. A utilização desta solução em conjunto com o *serialProxy* pode ser usada para criar uma biblioteca de projetos didáticos disponibilizados e acionados pela tela do navegador em qualquer parte, desde que seja informado o endereço IP do computador no qual a parte física (interface, componentes eletrônicos e estruturas) dos projetos está ligado. Os professores podem agendar aulas virtuais para alunos que estejam distantes ou que não foi possível comparecer a aula por algum motivo qualquer.

4.6 A Integração entre os Elementos Propostos

Além dos elementos mostrados nos itens anteriores, o modelo de robótica educativa proposto neste trabalho também contemple uma forma de integrar os projetos e conhecimentos mostrados em um ambiente na *web*, que permita ao aluno e ao professor socializar os projetos e as ideias discutidas em aula com outros grupos e também acessar alguns dos projetos através da Internet por meio de uma página *web* que pode enviar e receber dados para a interface *Arduino*. Esta integração se dá através da criação de um blog contendo os projetos discutidos em sala de aula, os relatórios de desempenho dos alunos, fotos e vídeos da montagem e teste dos projetos, além de uma página desenvolvida em *Java* ou *PHP* para permitir controlar alguns dos projetos através da Internet enviando e recebendo dados do *Arduino*.

A interface *Arduino* se comunica com o PC no qual está conectado por meio da porta USB. Esta comunicação se dá através de um módulo conversor serial/USB na própria placa. O computador no qual o *Arduino* está conectado deve ter um *driver* instalado que irá criar uma porta USB virtual emulando uma porta serial. Desta forma, é possível enviar e receber comandos do *Arduino* através desta porta tanto para ser interceptado pelo navegador quanto para qualquer outra implementação que fique “escutando” as mensagens que passam pela porta especificada. O esquema mostrado na figura 16 ilustra uma maneira de estabelecer uma comunicação serial/USB através de uma página *web* com a interface.

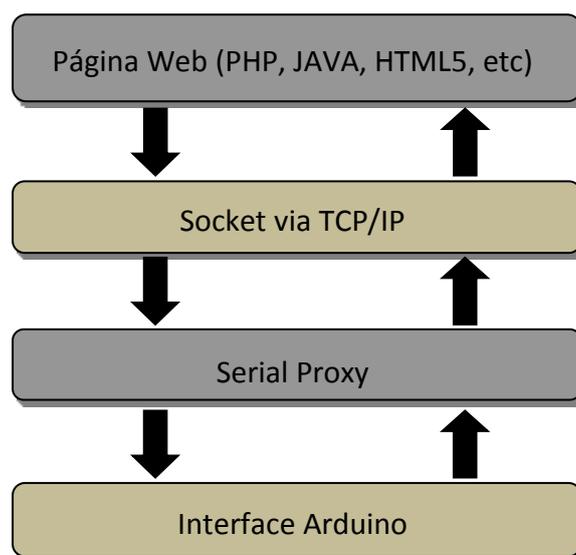


Figura 16 - Modelo de troca de mensagens entre uma página Web e a interface Arduino.

Para que este modelo funcione corretamente é necessário a utilização de um pequeno aplicativo de código aberto chamado *SerialProxy* que é um Proxy multitarefa sob licença GPL para plataforma *Linux*, *Mac* e *Windows*. O programa foi desenvolvido por Stefano Busti, em 1999 e posteriormente modificado por David A. Mellis em 2005. Este aplicativo permite redirecionar conexões de *sockets* de rede de/para um link serial, nos casos em que o link serial remoto não reconhece diretamente uma conexão TCP/IP. Quando estiver sendo executado, o *SerialProxy* entra em modo de escuta (*listen*) à espera de alguma conexão remota a uma faixa específica de portas TCP/IP. Uma vez que a conexão é estabelecida, os dados são recebidos e enviados para uma porta serial definida no arquivo de configuração do aplicativo. (SERIAL PROXY, 2013).

A configuração do *SerialProxy* para sua utilização com a interface *Arduino* é relativamente simples em qualquer um dos sistemas operacionais. As principais modificações necessária no arquivo *serproxy.cfg* (Windows) é praticamente a mesma nas outras plataformas e tem a ver com a definição da porta serial e porta TCP/IP a ser usada, além da taxa de transferência de dados. A tabela 4 contém os principais comandos que podem ser configurados para adaptar o *SerialProxy* à plataforma que está sendo usada pelo *Arduino*.

Tabela 4 – Lista das principais configurações do SerialProxy.

Linha de comando	Descrição
<code>comm_ports=<n></code>	Define quais portas seriais serão usadas pela interface Arduino. Valores aceitos: 1, 2, 3...9.
<code>comm_baud=<n></code>	Taxa de comunicação entre o computador e a interface em bits por segundo. Admite valores entre 9600 e 19200 bps.
<code>comm_databits=<n></code>	Define a quantidade de bits transferida por cada pacote de dados. O valor padrão geralmente é 8.
<code>comm_stopbits=<n></code>	Este comando informa quais serão os bits de parada que finalizam o pacote de dados. O valor padrão geralmente é 1.
<code>comm_parity=<p></code>	Define o bit de paridade na porta de comunicação. Os valores possíveis são <i>none</i> , <i>even</i> e <i>odd</i> .
<code>timeout=<n></code>	Informa o tempo máximo que o cliente pode ficar em modo de espera antes de ser desconectado. Admite qualquer valor

	em milissegundos.
net_port1=<n>	Define qual porta serial será mapeada através da porta TCP. Isto irá criar a comunicação TCP/IP-serial.

Fonte: Arquivo readme do SerialProxy.

Para uma interface conectada na porta COM3, taxa de transferência de 9600 bps e conversão de porta para 8081, um exemplo típico de configuração para o *serialProxy* seria:

```
comm_ports=3
comm_baud=9600
comm_databits=8
comm_stopbits=1
comm_parity=none
timeout=3000
net_port3=8081
```

No exemplo a seguir, temos um projeto no qual o aluno deve construir uma garra com pedaços de madeira, parafusos e porcas usando um mini servo motor para fazer a garra abrir e fechar. Este mesmo movimento pode ser feito também através da página na Internet clicando no comando `ABRIR GARRA` ou o comando `FECHAR GARRA`, equivalentes respectivamente às letras A e F que serão enviadas remotamente através do *SerialProxy* para a interface *Arduino* que controla a garra.

Devemos observar que a interface *Arduino* foi criada de forma a simplificar o máximo possível o seu uso com os mais diversos componentes eletrônicos disponíveis no mercado. Desta forma, com apenas algumas exceções é possível fazer a ligação destes componentes diretamente nas portas digitais e analógicas do *Arduino*. Sensores são ligados de acordo com a leitura dos seus dados (analógico ou digital) e atuadores como servo motores e motores de passo podem ser ligados diretamente ao *Arduino* e sua fonte de alimentação. Recomenda-se por uma questão de proteção utilizar um diodo em série com motores do tipo DC para evitar

que parte da carga elétrica usada para alimentar o motor retorne e danifique a porta do *Arduino* (PINHEIRO, 2012).

Não se deve utilizar cargas superiores a 50mA ligadas em cada porta do *Arduino* pois isso pode provocar a queima do microcontrolador. Além disso, é altamente recomendável ligar um resistor de pelo menos 200 Ohms em série com a porta de sua interface quando estiver utilizando cargas que drenam muita energia para evitar sobrecargas na placa. A figura 17 mostra a tela da página *web* usada para controlar a garra e a figura 18 ilustra a montagem física da garra já conectada a um *laptop* através do *Arduino*.



The screenshot shows a web browser window with the URL `www.admacademico.com.br/index.html`. The page title is **ROBÓTICA EDUCATIVA**. The navigation menu includes [INÍCIO](#), [SOBRE](#), [MATERIAL DIDÁTICO](#), and [CONTATO](#). A large graphic features an Arduino board inside a stylized head shape, with the text 'COMANDO REMOTO' above it. A red message 'Comando enviado.' is displayed on the right. Below the graphic, the page is divided into two sections: 'PROJETO: Garra com Servo Motor' and 'ACIONAMENTO'. The 'PROJETO' section describes the gripper's construction and operation. The 'ACIONAMENTO' section includes an IP address field (192.168.1.177.8081) and two buttons: 'Abrir Garra' and 'Fechar Garra'. Below these are two diagrams of the gripper: 'Garra Fechada' and 'Garra Aberta'.

ROBÓTICA EDUCATIVA

INÍCIO SOBRE MATERIAL DIDÁTICO CONTATO

COMANDO REMOTO

Comando enviado.

PROJETO: Garra com Servo Motor

Este projeto permite abrir e fechar uma garra feita com pedaços de madeira e um servo motor através de uma página na Internet.

FUNIONAMENTO: O aluno(a) deve informar o endereço IP e a porta da máquina onde o projeto está ligado e em seguida clicar em um dos botões: **ABRIR GARRA** ou **FECHAR GARRA**.

ACIONAMENTO

Endereço IP: 192.168.1.177.8081

Garra Fechada Garra Aberta

Figura 17 - Página Web com exemplo de conexão remota ao Arduino.

Fonte: O autor

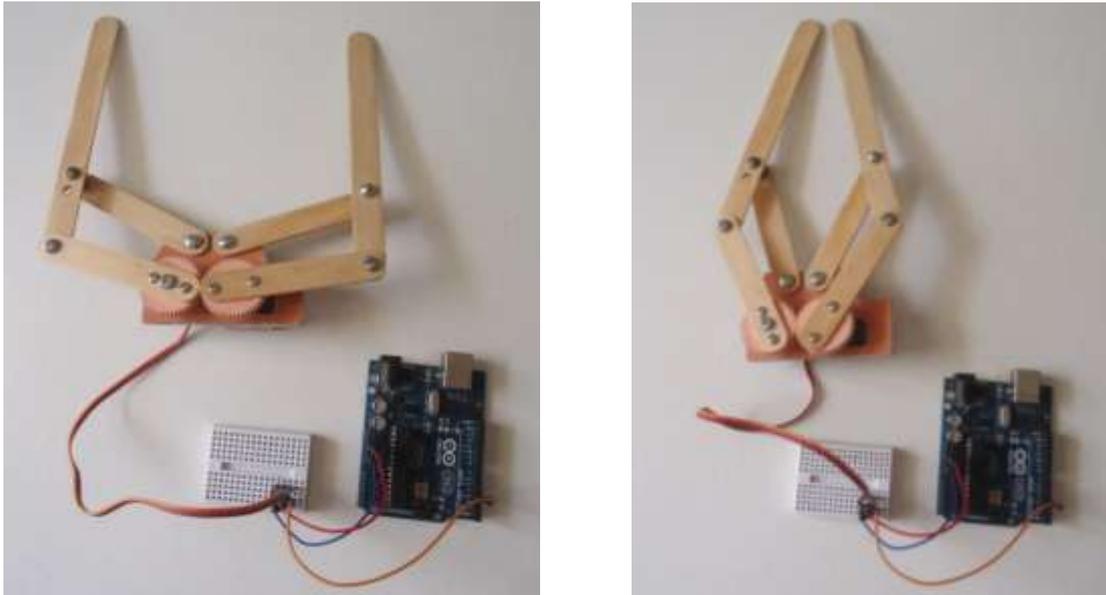


Figura 18 - Montagem prática do projeto garra com mini servo motor usada no exemplo prático abordado neste capítulo.

Fonte: O autor

O projeto utiliza uma interface *Arduino*, uma mini matriz de contatos para ligar os fios aos componentes, três fios para ligação elétrica, um mini servo motor, duas engrenagens aproveitadas de brinquedos usados, porcas, parafusos, pedaços de madeira aproveitados de palitos de picolé e pedaços de papelão rígido.

O código fonte a seguir representa a página web utilizada para acesso remoto a alguns dos projetos práticos desenvolvidos durante as aulas e o código fonte do *Arduino* para o projeto de exemplo utilizando uma garra com mini servo motor.

1. Página web em *javascript* que faz uso do *WebSocket* para enviar/receber dados do *Arduino* através do *serverProxy* (www.admacademico.com.br).

```
<!DOCTYPE html>
<html lang="pt">
  <head>
    <title> Aulas de Robótica Educativa - Projetos Online </title>
    <meta name="viewport" content="width=device-width, minimum-scale=1.0,
maximum-scale=1.0" />
    <link href="styles/main.css" type="text/css" rel="stylesheet">
    <!--[Se lt IE 9]>
```

```

<script src="//html5shiv.googlecode.com/svn/trunk/html5.js">
</script>
<![endSe]-->
<script type='text/javascript' src='scripts/respond.min.js'></script>

<script language="javascript" type="text/javascript">
    function init(){
        var saida = document.getElementById("saida");
    }

    function abreGarra() {
        if ("WebSocket" in window){
            var wsURL = "ws:"+document.formulario.txt_ip.value
            websocket = new WebSocket(wsURL);
            websocket.onopen = saida.innerHTML = "Conectado!";
            websocket.onclose = function(evt) { Fechar(evt) };
            websocket.onmessage = function(evt) { msgAbre(evt) };
            websocket.onerror = saida.innerHTML = "<span style='color:
red;'>Comando enviado.</span>";
        } else {
            alert("Seu navegador não suporta WebSockets.");
        }
    }

    function fechaGarra() {
        if ("WebSocket" in window){
            var wsURL = "ws:"+document.formulario.txt_ip.value
            websocket = new WebSocket(wsURL);
            websocket.onopen = saida.innerHTML = "Conectado!";
            websocket.onclose = function(evt) { Fechar(evt) };
            websocket.onmessage = function(evt) { msgFecha(evt) };
            websocket.onerror = saida.innerHTML = "<span style='color:
red;'>Comando enviado.</span>";
        } else {
            alert("Seu navegador não suporta WebSockets.");
        }
    }

    function msgAbre(evt){
        saida.innerHTML = "<p>Conectado!</p>";
        websocket.send("A");
    }

```

```

    }
    function msgFecha(evt) {
        saida.innerHTML = "<p>Conectado!</p>";
        websocket.send("F");
    }
    function fechar(evt) {
        saida.innerHTML = "<span style='color: red;'>Conexão
encerrada.</span>";
        websocket.close();
    }

    window.addEventListener("load", init, false);
</script>
</head>

<body onload="init()">
    <div id="wrapper">
        <header>
            <nav id="skipTo">
                <ul><li>
                    <a href="#main" title="Skip to Main Content">Skip to Main
Content</a>
                </li></ul>
            </nav>

            <h1>robotic</h1>
            <nav>
                <ul>
                    <li><a href="#" title="Página Inicial">Início</a></li>
                    <li><a href="#" title="Sobre">Sobre</a></li>
                    <li><a href="#" title="Material de apoio">Material
Didático</a></li>
                    <li><a href="#" title="Contatos">Contato</a></li></ul>
                </nav>
                <div id="banner">
                    
                </div>
            </header>
            <section id="main">
                <h1>PROJETO: Garra com Servo Motor</h1>

```

<p>Este projeto permite abrir e fechar uma garra feita com pedaços de madeira e um servo motor através de uma página na Internet.</p>

<p>FUNCIONAMENTO: O aluno(a) deve informar o endereço IP e a porta da máquina onde o projeto está ligado e em seguida clicar em um dos botões: ABRIR GARRA ou FECHAR GARRA.</p>

</section>

<aside>

<h1>ACIONAMENTO</h1>

<p><form action="" id="formulario" name="formulario">

<fieldset>

Endereço IP: <input type = "text" id="txt_ip" required />

<button type ="button" onclick = "abreGarra()">

Abrir Garra

</button>

<button type ="button" onclick = "fechaGarra()">

Fechar Garra

</button>

</fieldset>

</form>

</aside>

</div>

<div id="saida" style="LEFT: 860px; WIDTH: 276px; POSITION: absolute; TOP: 375px; HEIGHT: 105px; width: 540">

</div>

<div class=side style="LEFT: 660px; WIDTH: 376px; POSITION: absolute; TOP: 675px; HEIGHT: 105px; width: 540">

<p> Garra Fechada</p>

</div>

<div class=side style="LEFT: 790px; WIDTH: 376px; POSITION: absolute; TOP: 675px; HEIGHT: 105px; width: 540">

<p> Garra Aberta</p>

</div>

</body>

```
</html>
```

2. *Sketch* do *Arduino* responsável por controlar o movimento de abrir e fechar da garra usando a biblioteca servo.

```
// Bibliotecas usadas para os servos e o sensor de distancia
#include <Servo.h>
#include <Ultrasonic.h>

// Duas instancias do objeto servo para mover o veiculo
Servo servo1;
Servo servo2;

// Define as portas digitais 12 e 13 para o sensor ultrassonico
Ultrasonic ultra(12,13);

void setup(){
  // Os servos motores ligados nas portas digitais PWM 3 e 6
  servo1.attach(3);
  servo2.attach(6);
}

void loop(){
  long dist;
  float centim;

  dist = ultra.timing(); // Armazena o valor lido pelo sensor.
  centim = ultra.convert(dist,Ultrasonic::CM); // Armazena o valor
convertido em centímetros lido pelo sensor de distância.

  if (centim < 20){
    // se a distância medida for menor que 20cm o veiculo anda um pouco
para trás e em seguida vira para o lado para desviar do obstáculo.
    servo1.write(90);
    servo2.write(0);
    delay(500);
    servo1.write(0);
    servo2.write(0);
    delay(300);
  }
}
```

```
// após desviar, o robô volta a andar para frente.  
servo1.write(0);  
servo2.write(90);  
delay(1000);  
}
```

4.7 A Aplicação do Modelo na Prática

A tarefa de implantar aulas de robótica educativa em uma instituição de ensino em qualquer nível depende muito da forma como estas aulas serão ministradas. Porém, se forem seguidos os passos mencionados no capítulo 4.4 deste trabalho, a tarefa se torna muito mais fácil e evita que a escola tenha gastos desnecessários e gere frustração por parte dos alunos e professores envolvidos no projeto de robótica.

Este modelo vem sendo utilizado desde o final do ano de 2012 em uma escola de ensino fundamental e médio da cidade de Maceió como forma de estimular os alunos que já participam do projeto de robótica educativa a alguns anos utilizando os kits educacionais da linha LEGO. Temos percebido que após alguns anos utilizando este material didático, há uma certa saturação por parte dos alunos e muitos deles acabam por se desestimular para participar das aulas. Em virtude disto e de uma tentativa para reduzir custos das atividades de robótica, foi implementado o modelo proposto neste artigo para um grupo de alunos da primeira série do Ensino Médio. Estes alunos já haviam participado das aulas de robótica desde o sétimo ano do Ensino Fundamental e se sentiram mais motivados a utilizar novos materiais e novas metodologias. A utilização deste material serviu de base para alguns projetos simples como veículos, garras e mecanismos simples motorizados. Esta utilização se deu por um período de aproximadamente seis meses, com uma aula semanal e uma aula aos sábados a cada quinze dias. Houve uma significativa redução de custos para a escola conforme mostrado na tabela 5 que ilustra um comparativo dos custos da escola com kits de robótica educacional proprietário da linha *Legó Mindstorms* e com os materiais adotando software livre e material de baixo custo.

Tabela 5 – Comparativo de custos entre kits usados no projeto.

Produto	Custo Unitário em R\$	Descrição
Lego Mindstorms NXT 2.0 (importado)	2.000,00	Kit proprietário que inclui 619 peças e software Robolab.
Kit Modelix Escolar 411 (nacional)	2.500,00	Kit proprietário de fabricação nacional com material didático e proposta pedagógica incluso.
Kit Arduino intermediário opensource	450,00	Interface de controle com vários itens eletrônicos, mecânicos, sensores e atuadores.

Fonte: O autor

Conforme os dados na tabela 5 observa-se que há uma diferença bastante grande entre os kits proprietários comumente encontrados no mercado brasileiro e o uso de kits montados com peças avulsas e aproveitando materiais de baixo custo. Enquanto que cada kit proprietário contém peças suficientes para se trabalhar com grupos de quatro alunos, a aquisição de um conjunto de peças básicas trariam uma redução de custos de aproximadamente 65% quando se adota a proposta indicada neste trabalho, sem contar com a utilização de materiais de sucata e a impressão do material didático. Os valores em real são referentes a compra dos produtos no Brasil, no período entre julho e outubro de 2012 e não inclui valores de frete. Além dos ganhos óbvios em termos de redução de custos neste modelo proposto, existe ainda o ganho em termos de flexibilidade na utilização de materiais alternativos por parte dos alunos, onde com muita criatividade, muitos deles inclusive trouxeram para as aulas outras formas de integrar os materiais adquiridos com peças de sucata como, por exemplo, palitos de madeira, isopor e papelão. Como o projeto ainda se encontra em caráter experimental tanto o material didático como o material de apoio ao professor possuem poucos projetos, porém já contamos com algumas ideias por parte dos alunos envolvidos no projeto, conforme mostrado nas figuras 19, 20 e 21.

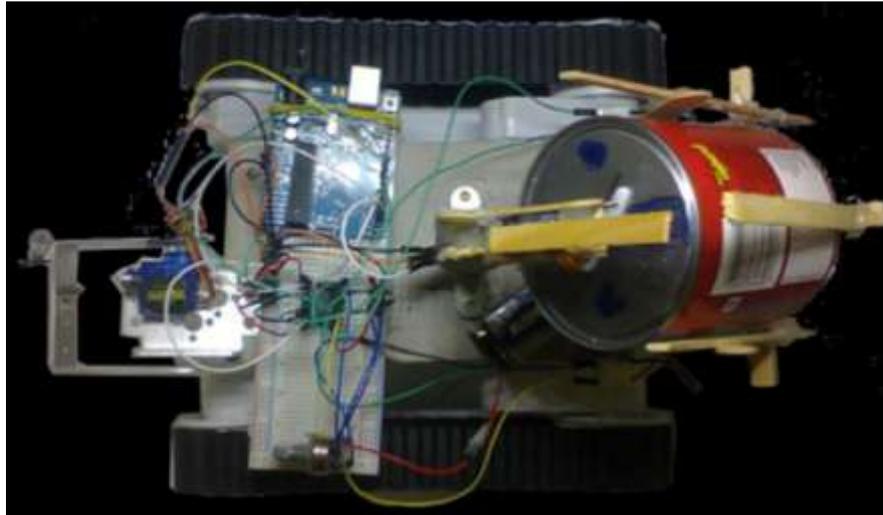


Figura 19 - Montagem do projeto robô com esteiras para plantação automática de sementes criado por alunos do 2º Ano do Ensino Médio

Fonte: O autor

O robô mostrado na figura 19 serviu de base para um dos artigos apresentado pelo alunos do Ensino Médio no qual é relatado o seu funcionamento e os objetivos deste projeto: criar um robô autônomo com uma interface gráfica em *Processing* onde o usuário informa os dados do terreno (largura, comprimento e quantidade de covas) e o robô irá fazer os furos, lançar as sementes nas covas e em seguida arrastar terra sobre elas.

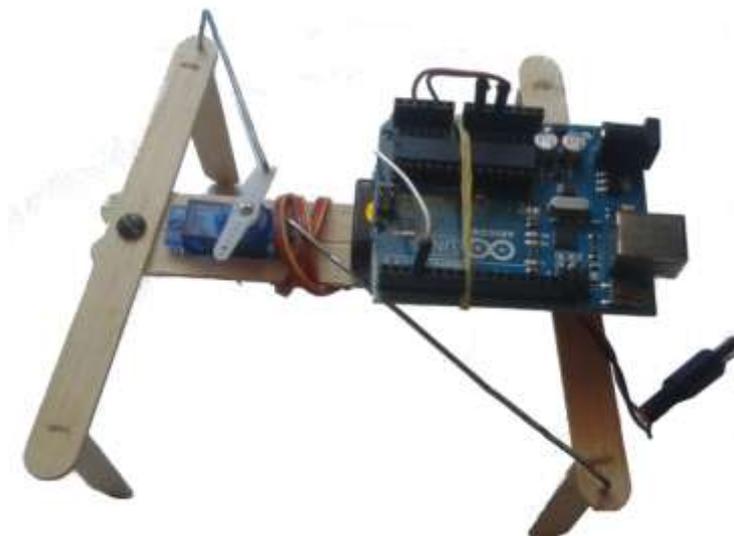


Figura 20 - Montagem do projeto robô com patas criado por alunos do 7º Ano do Ensino Fundamental

Fonte: O autor.

No projeto mostrado na figura 20, os alunos procuraram usar uma estrutura simples formada por palitos de madeira (palito de picolé), cliques de papel e fios para montar um robô com quatro patas.

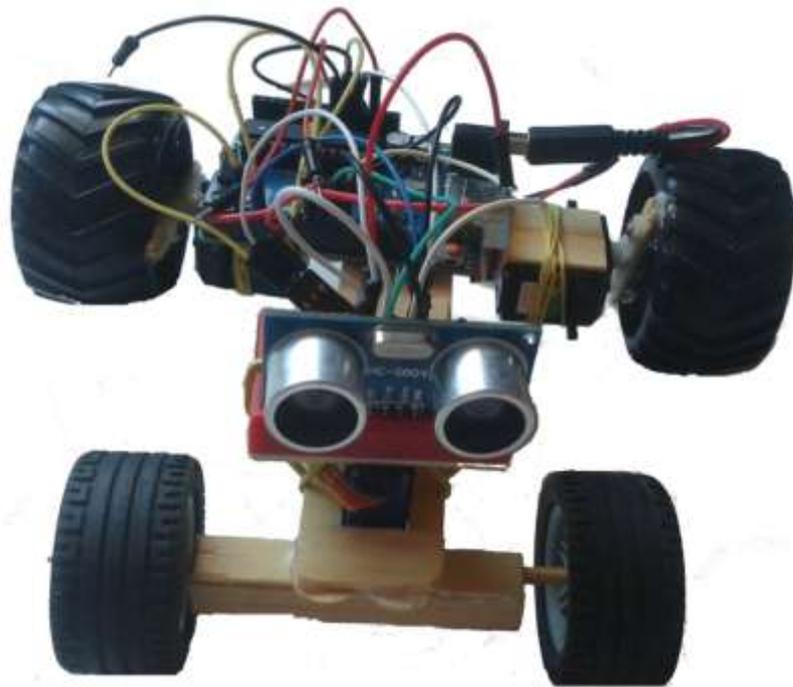


Figura 21 - Projeto veículo autônomo com sensor de distância criado por alunos do 2º Ano do Ensino Médio

Fonte: O autor

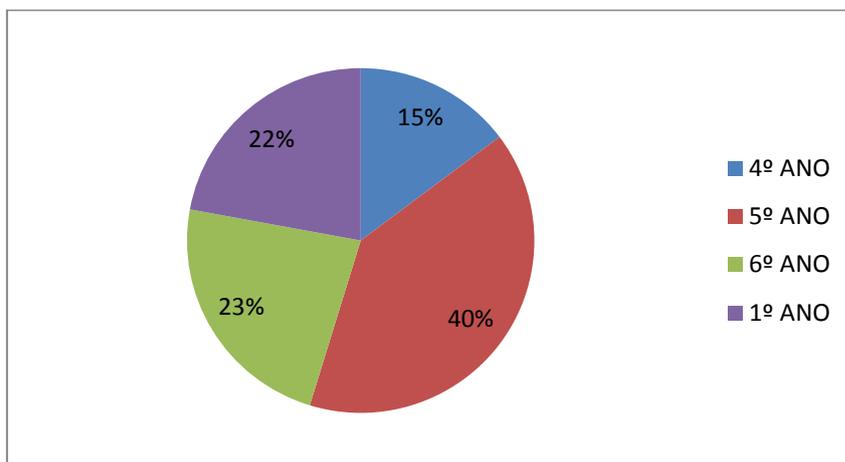
No projeto mostrado na figura 21, os alunos criaram um veículo autônomo usando palitos de madeira (palitos de picolé) com sensor de distância para evitar obstáculos e um micro servo motor na parte da frente para dar o sentido de direção do veículo.

Como já foi mencionado anteriormente as aulas de robótica educativa na escola em estudo ocorrem em tempos de 50 minutos cada, uma vez por semana abrangendo todos os alunos do 4º ao 7º ano do ensino fundamental e apenas um grupo de 21 alunos do 1º ano do ensino médio. Ao final do período letivo utilizamos uma pesquisa estatística como instrumento para medir o grau de eficácia e satisfação desta metodologia aplicando dois questionários aos alunos participantes do projeto no qual foram avaliados os seguintes critérios:

- O grau de interesse dos alunos nas atividades realizadas;
- O nível de aprendizagem dos conteúdos em disciplinas como matemática, ciências e física;
- A participação dos alunos nos projetos.

Um dos questionários foi aplicado aos alunos do 4º, 6º e 7º ano do ensino fundamental que tiveram aulas de robótica educativa usando os kits educacional da linha LEGO *Mindstorms* no ano letivo de 2013. Os alunos do 4º ano, na faixa etária entre 10 e 11 anos, representam o grupo de entrada no projeto, pois é a partir desta série que o colégio inicia as aulas de robótica. Enquanto que os alunos do 7º ano, com faixa etária entre 13 e 14 anos, representa o grupo final nesta etapa do projeto pois a escola finaliza as aulas de robótica nesta série. Ao todo 14 alunos do 4º ano, 22 alunos do 6º ano e 38 alunos do 7º ano responderam o questionário. Após a análise dos dados coletados percebemos que dos 74 alunos pesquisados, 44,7% demonstraram uma boa capacidade de manusear os materiais e 4,1% dos alunos tiveram dificuldade em manusear os materiais usados nos projetos.

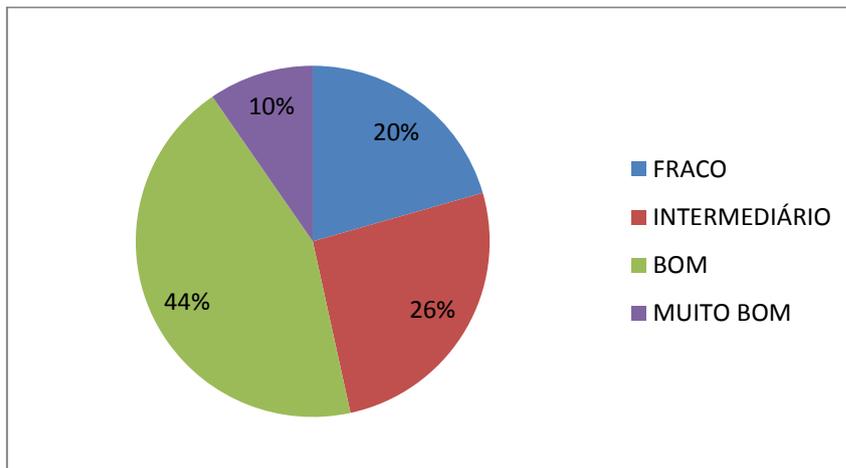
Gráfico 1 – Alunos que responderam os questionários de avaliação.



Outra informação importante nos dados indica que 39,2% dos alunos pesquisados demonstraram muito interesse em aprender mais sobre ciência e tecnologia após as aulas de robótica, o mesmo percentual de alunos afirma ter melhorado o seu grau de envolvimento e

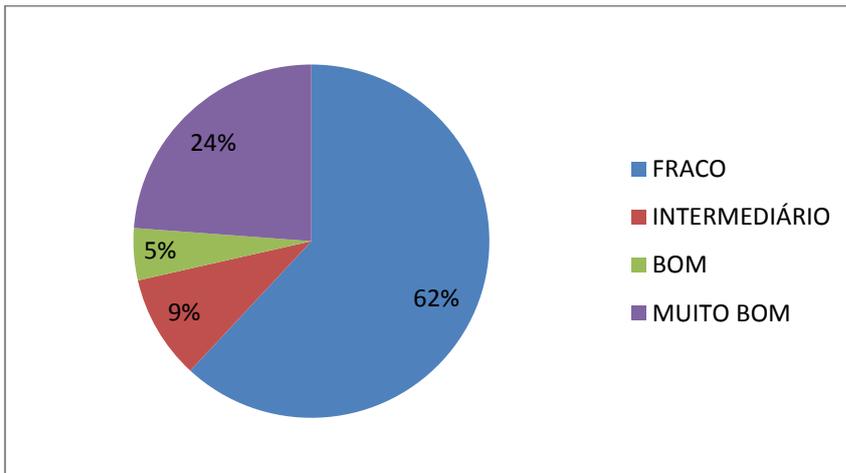
trabalho em equipe e 43,3% dos alunos demonstraram um bom conhecimento quanto ao ambiente de programação do LEGO NXT.

Gráfico 2 – Conhecimento do ambiente de programação LEGO NXT.



O segundo questionário foi aplicado a um grupo de 21 alunos do 1º ano do ensino médio que participaram do projeto de aulas de robótica educativa usando a interface *Arduino* e materiais de sucata. Neste segundo grupo de alunos 29% não participaram das aulas de robótica usando os kits LEGO *Mindstorms*. Dos alunos analisados nesta pesquisa, 76,5% demonstraram um bom manuseio dos materiais e da interface, o que demonstra que a participação nas aulas de robótica em séries anteriores facilitou a utilização por parte dos alunos. No entanto, apenas 29% dos alunos pesquisados tiveram um bom ou excelente aproveitamento quanto ao aprendizado da linguagem de programação C++, usada com a interface *Arduino*, o que demonstra a necessidade em focar mais as aulas no ensino de lógica e linguagem de programação para esta faixa etária. O grau de interesse demonstrado pelos alunos em utilizar material de sucata, componentes eletrônicos e o próprio *Arduino* foi de 95,6%, o que deixou evidente o interesse dos alunos por criar experiências utilizando materiais alternativos.

Gráfico 3 – Conhecimento da linguagem de programação do Arduino.



5. Conclusões e Resultados Obtidos

Neste capítulo serão mostradas as conclusões e os resultados obtidos com a aplicação deste modelo de ensino de robótica educativa em turmas de alunos do ensino fundamental e médio, além das fases de desenvolvimento da parte prática e da integração dos elementos propostos através da web.

Os estudos desenvolvidos neste trabalho abordaram de forma clara através de uma revisão bibliográfica e de um levantamento de tecnologias e metodologias utilizadas atualmente que as aulas de robótica educativa são uma ferramenta importante na formação acadêmica e até mesmo na formação profissional futura dos alunos dos anos finais do Ensino Fundamental, Ensino Médio e Técnico. A importância de inserir estas competências aos alunos se dá pelo estímulo ao aprendizado das ciências exatas, do aprendizado pela experiência prática, pelo prazer na descoberta de soluções para problemas que podem ser resolvidos através da experimentação e também pela socialização dos conhecimentos adquiridos planejando e executando os projetos em grupo. Apesar de se encontrar ainda em um estágio inicial, as aulas de robótica educativa se mostram uma excelente ferramenta tecnológica, associada às aulas de informática e aprendizado da lógica, no apoio ao ensino e aprendizagem. Porém, para que este tipo de atividade tenha um maior avanço nas escolas, alguns fatores importantes se fazem necessários:

- Um maior interesse por parte das escolas, tanto públicas quanto privadas, no incentivo para que os seus alunos tenham acesso a este tipo de atividade;
- O apoio aos professores que, uma vez apresentados a estas tecnologias, se sintam motivados a ministrar este tipo de aula e desenvolver material didático para seus alunos;
- O desenvolvimento de uma grade curricular que contemple o uso das tecnologias da informação, especialmente a robótica educativa;
- O envolvimento também dos pais e da comunidade no entorno da escola promovendo atividades como competições, feiras e socializações de projetos abertos ao público em geral;
- A utilização sempre que possível de kits e produtos nacionais e material de sucata.

- Um meio de incentivar o aluno a praticar fora do ambiente da escola os ensinamentos aprendidos, se possível, de forma prática usando matérias encontrados em sua própria casa ou através de simulações disponibilizadas pela Internet.

As recomendações acima e a aplicação na prática das atividades propostas durante as aulas tiveram um resultado promissor na medida em que houve um interesse maior dos alunos além de um aproveitamento maior durante as aulas de robótica com a utilização de novos materiais. Durante a elaboração deste trabalho observou-se também que o grau de envolvimento dos alunos e dos professores em todas as etapas dos projetos contribuiu imensamente para aumentar o interesse pelas aulas além de promover a disseminação dos conhecimentos adquiridos entre os alunos que não participaram das atividades e em alguns casos havendo interesse também por parte de alunos de outras escolas.

A figura 22 ilustra alguns momentos de aula com os alunos, o desenvolvimento dos projetos no laboratório e treino para participação na modalidade prática da Olimpíada Brasileira de Robótica (OBR).



Figura 22 - Fotos dos alunos em aula durante os projetos e treinando para competições de robótica.

Fonte: O autor

Durante o período de desenvolvimento deste trabalho foram produzidos artigos que foram apresentados em eventos a nível regional e nacional como a Escola Regional Bahia Alagoas e Sergipe (ERBASE), com o título “A Olimpíada Brasileira de Robótica Educativa como Ferramenta de Ensino/Aprendizagem no Ensino Fundamental”, que ganhou o primeiro lugar neste mesmo evento. Além disso, podemos citar outro artigo importante e que ficou entre os cinco melhores do Simpósio Brasileiro de Robótica, parte integrante do *Latin American Robotics Symposium*, com o título “Integração de Múltiplas Plataformas Robóticas no Ensino Fundamental e Médio”. Mais recentemente um dos alunos do Ensino Médio, participante das aulas de robótica apresentou um artigo na Mostra Nacional de Robótica,

ficando entre os melhores artigos do evento e obtendo com isso uma bolsa de iniciação científica júnior pelo CNQp.

Os bons resultados e a experiência adquiridos com este projeto, além dos dados coletados sobre a percepção dos alunos nos tem dado condições de investir ainda mais na participação de competições, eventos e submissão de artigos e projetos a nível nacional e eventualmente até mesmo internacional, não apenas com relação aos professores mas também incentivando aos alunos do ensino médio a também escrever seus próprios artigos e submetê-los a congressos e publicações.

5.1 Trabalhos Futuros

Os resultados obtidos com este trabalho atingiram o esperado na medida em que estão sendo usados nas aulas de uma escola particular, porém no decorrer dos estudos deste trabalho observou-se que os seguintes pontos ainda podem ser melhorados, aperfeiçoados ou implementados:

- Os alunos ainda sentem dificuldades para trabalhar com as peças de sucata e para lidar com componentes eletrônicos pequenos e frágeis.
- Como não há uma boa base lógica a parte da programação baseada na IDE do *Arduino* também oferece muita resistência por parte dos alunos que demonstram dificuldades em assimilar a lógica de programação.
- Após criar uma página web de testes com alguns dos projetos disponíveis no material didático, alguns alunos relataram que não conseguiram se comunicar com o *Arduino* via web e estamos estudando o problema em busca de possíveis soluções. Acreditamos que alguns provedores de acesso estejam bloqueando os comandos por acharem se tratar de um programa malicioso.
- Uma boa sugestão dos alunos é implementar um acesso à *webcam* na página onde está o projeto a ser acionado *online*, desta forma o aluno pode acionar e ao mesmo tempo ver pela webcam o robô em funcionamento.
- Por fim, pode-se desenvolver uma forma de alterar a própria programação do projeto online e enviar estas alterações pelo navegador para o *bootloader* do *Arduino*, recompilando e executando as mudanças no código.

Além dos ganhos evidenciados, é importante também salientar que nos últimos anos a participação dos alunos nas aulas de robótica educativa também renderam boas participações dos alunos envolvidos em eventos como a Olimpíada Brasileira de Robótica (OBR), na qual obtivemos medalhas de prata e bronze a nível estadual, medalhas de prata a nível nacional na modalidade teórica também da OBR e bolsa de iniciação científica júnior do CNPq para alunos que tiveram artigo científico na área de robótica educativa aceito e apresentado na Mostra Nacional de Robótica (MNR).

Embora ainda esteja no começo, a utilização do modelo proposto neste trabalho já mostra seus bons resultados como ferramenta de pesquisa e de meio de referência para que outros grupos possam dar continuidade e até mesmo desenvolvendo novos modelos com novas funcionalidades expandindo ainda mais a utilização da robótica educativa para ajudar a melhorar o desempenho e o interesse pelas ciências exatas de nossos alunos por todo o país.

Referências

- ACAR, M.; PARKIN R. M. **Engineering Education for Mechatronics**. IEEE Transactions on Industrial Electronics, vol. 43, no. 1, p. 106-112, 1996.
- ANDRADE, P.F. **Programa Nacional de Informática Educativa. A utilização da Informática na escola pública brasileira**. (1970-2004); MEC: Secretaria de Educação a Distância, 1996.
- BURNS, Edward. M. **História da civilização ocidental: do homem das cavernas até a bomba atômica**. 2. Ed. Editora Globo. (Original Publicado em 1941), 1997.
- CASTILHO, Maria. I. **Robótica na educação: com que objetivos?** Porto alegre, 2002. Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Disponível em <<http://www.pucrs.br/eventos/desafio/mariaines.php>>. Acesso em 12 dez. 2013.
- CÉSAR, Danilo R. **O que é Robótica Livre?** Disponível em: <<http://www.roboticalivre.org/portal/?q=node/1>> Acesso em: 28 dez. 2012.
- COSTELHA, Hugo; LIMA, Pedro. **Modelling, analysis and execution of robotic tasks using petri nets**. IEEE International Conference on Intelligent Robots and Systems, 2007.
- CURCIO, Cristina. P. C. **Proposta de Método de Robótica Educacional de Baixo Custo**. Curitiba, 2008. Dissertação (Mestrado) – Instituto de Tecnologia para o Desenvolvimento (LACTEC), Programa de Pós-Graduação em Desenvolvimento de Tecnologia, 2008.
- DAOUN, Michel. **Alunos criativos, robôs idem**. Revista Carta na Escola. Rio de Janeiro, n. 25, abril, 2008.
- D'ABREU, João. V. V. **Integração de dispositivos mecânicos para ensino-Aprendizagem de conceitos na área de controle e automação**. Tese de Doutorado, Universidade

Estadual de Campinas – UNICAMP, Faculdade de Engenharia Mecânica – FEM, Campinas, 2002.

FLL. First Lego League. Online. Disponível em <<http://www.firstlegoleague.org>>. Acesso em 20 jan. 2014.

GASPARETTI, Antonio. C. **Laboratório Remoto para Ensino de Engenharia**. Online. Disponível em <ftp://ftp.usjt.br/pub/revint/353_39.pdf>. Acesso em 16 jan. 2014.

INOVAÇÃO TECNOLÓGICA. **Cientistas propõem um robô por aluno**. 14/11/2011. Online. Disponível em www.inovacaotecnologica.com.br/noticias/noticia.php?artigo=um-robo-por-aluno>. Acesso em 18 jan. 2012.

INSTRUCTABLES. **Understanding ICSP for PIC Microcontrollers**. Online. Disponível em <<http://www.instructables.com/id/Understanding-ICSP-for-PIC-Microcontrollers>>. Acesso em 18 abr. 2012.

KONZEN, I. M. G. et al. **Kit de robótica educativa: desenvolvimento e aplicação metodológica**. In: Encontro da Escola Regional de Licenciatura em Computação, II, 2007, Santa Cruz. Anais: II Encontro da Universidade de Santa Cruz do Sul. Santa Cruz: Unisc, Departamento de Informática, 2007. p. 1 - 4.

LDR. **Light Dependent Resistor**. Online. Disponível em <http://www.gta.ufrj.br/grad/01_1/contador555/ldr.htm>. Acesso em 19 fev. 2014.

LEGO EDUCACIONAL. **Edacom Tecnologia**. Online. Disponível em <<http://www.edacom.com.br/Default.aspx?tabid=80>>. Acesso em 18 dez. 2011.

LEGO ZOOM. **Edacom Tecnologia**. Online. Disponível em <<http://www.edacom.com.br/solucoesdeaprendizagem.aspx>>. Acesso em 18 dez. 2011.

LEWIN, Edwards. A. R. W. **Open-source robotics and process control cookbook**. Elsevier Books. Oxford, 2006.

MARGOLIS, Michael. **Arduino Cookbook**. O'Reilly Media. 2ª. Ed, 2011.

MCROBERTS, Michael. **Arduino Básico**. tradução Rafael Zanolli. Editora Novatec. São Paulo, 2011.

MIRANDA, Leonardo. C. **Robofácil: especificação e implementação de artefatos de hardware e software de baixo custo para um kit de robótica educacional**. Rio de Janeiro, 2006. f. 124 .Dissertação (Mestrado em Informática) – Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro.

MODELIX ROBOTICS. **Modelix**. Online. Disponível em <<http://www.modelix.cc/pt-br/>>. Acesso em 20 fev. 2012.

MÓDULO MC. **Módulo de Controle MC 2.5**. Online. Disponível em <http://www.pnca.com.br/downloads/Dados%20T%C3%A9cnicos/datasheet_mc2.5_v2.6.1.5.pdf>. Acesso em 18 abr. 2012.

OBR. **Olimpíada Brasileira de Robótica**. Online. Disponível em <<http://www.obr.org.br/>>. Acesso em 19 fev. 2014.

PROCESSING. **Processing**. Online. Disponível em <<http://www.processing.org/>>. Acesso em 20 fev. 2012.

PAIS, L. C. **Educação escolas e as tecnologias da informática**. Belo Horizonte: Autêntica, 2002.

PARÂMETROS CURRICULARES NACIONAIS. Introdução aos parâmetros curriculares nacionais. Secretaria de Educação Fundamental. – Brasília : MEC/SEF, 1997.

PIAGET, Jean. **Aprendizagem e conhecimento**. Freitas Bastos, 1974.

PINHEIRO, Fernando. K. **Controlando Motores DC com Arduino**. Rio Grande do Sul. Disponível em: <<http://ferpinheiro.wordpress.com/2012/05/20/controlando-motores-dc-com-arduino/>>. Acesso em: 12 fev. 2014.

PNCA ALFA. **PNCA Alfa Robótica e Eletrônica**. Online. Disponível em

<http://www.pnca.com.br/index.php?option=com_content&view=article&id=19&Itemid=27>. Acesso em 18 abr. 2012.

PROJETO GOOGLE CODE. **Google Code**. Online. Disponível em

<<http://code.google.com/p/arduino>>. Acesso em 15 abr. 2012.

QUINTANILHA, Leandro. **Irresistível robô**. Revista A Rede, São Paulo, ano 3, n. 34, p.10-17, mar. 2008.

RIBEIRO, Célia R. **Robôcarochinha: um estudo qualitativo sobre a robótica educativa no 1º ciclo do ensino básico**. Braga, 2006.f. 207 Dissertação (Mestrado em Educação) - Instituto de Educação e Psicologia, Universidade do Minho.

SCHONS C., PRIMAZ E. WIRTH G. A. P. **A Robótica Educativa na Instituição Escolar para alunos do Ensino Fundamental da disciplina de Língua Espanhola através das Novas Tecnologias de Aprendizagem**. In Anais do I Workshop de Computação da Região Sul. 2004

SERIAL PROXY. **SerialProxy versão 0.14**. Online. Disponível em

<<http://cote.cc/blog/serialproxy-v014-can-use-com-ports-above-9>>. Acesso em 25 fev. 2013.

SILVA, Alexandre. J. B. ALMEIDA, Eliana. **Integração de Múltiplas Plataformas**

Robóticas no Ensino Fundamental e Médio. In Anais do III Workshop de Robótica Educacional – Simpósio Brasileiro de Robótica, 2012. Online. Disponível em:

<<http://www.natalnet.br/lars/wre2012/pdf/105833.pdf>>. Acesso em 26 dez. 2013.

SIROS. **Sistemas Robóticos com SuperLogo**. Online. Disponível em

<<http://www.nied.unicamp.br/?q=content/sistemas-rob%C3%B3ticos-com-superlogo-siros>>. Acesso em 16 set. 2013.

TEIXEIRA, José C. **Aplicações da Robótica no Ensino Secundário: O Sistema Lego Mindstorms e a Física**. 187p. Dissertação (Mestrado em Ensino da Física) – Departamento de Física, Universidade de Coimbra, 2006.

ROBOEDUC. **RoboEduc**. Online. Disponível em <<http://www.roboeduc.com/>>. Acesso em 21 fev. 2012.

ROBOTICANDO. **RobotIcanDo**. Laboratório de Estudos Cognitivos da Universidade Federal do Rio Grande do Sul (UFRGS/LEC). Disponível em: <http://www.pgie.ufrgs.br/portalead/rosane/fortaleza/oea_lec/roboticando/index.ht>. Acesso em 28 dez. 2012.

ROBOTICS ACADEMY. **Carnegie Mellon University Robotics Academy**. Online. Disponível em <<http://www.education.rec.ri.cmu.edu/index.htm>> Acesso em 12 mai. 2014.

ROBOTICS CONNECTION. **Serializer Robot Controller**. Online. Disponível em <<http://www.roboticsconnection.com/p-16-serializer-robot-controller.aspx>>. Acesso em 20 jan. 2012.

ROBOTICS PROJECT. **The Robotics Alliance Project**. Online Disponível em <<http://robotics.nasa.gov/>> Acesso em 12 mai. 2014.

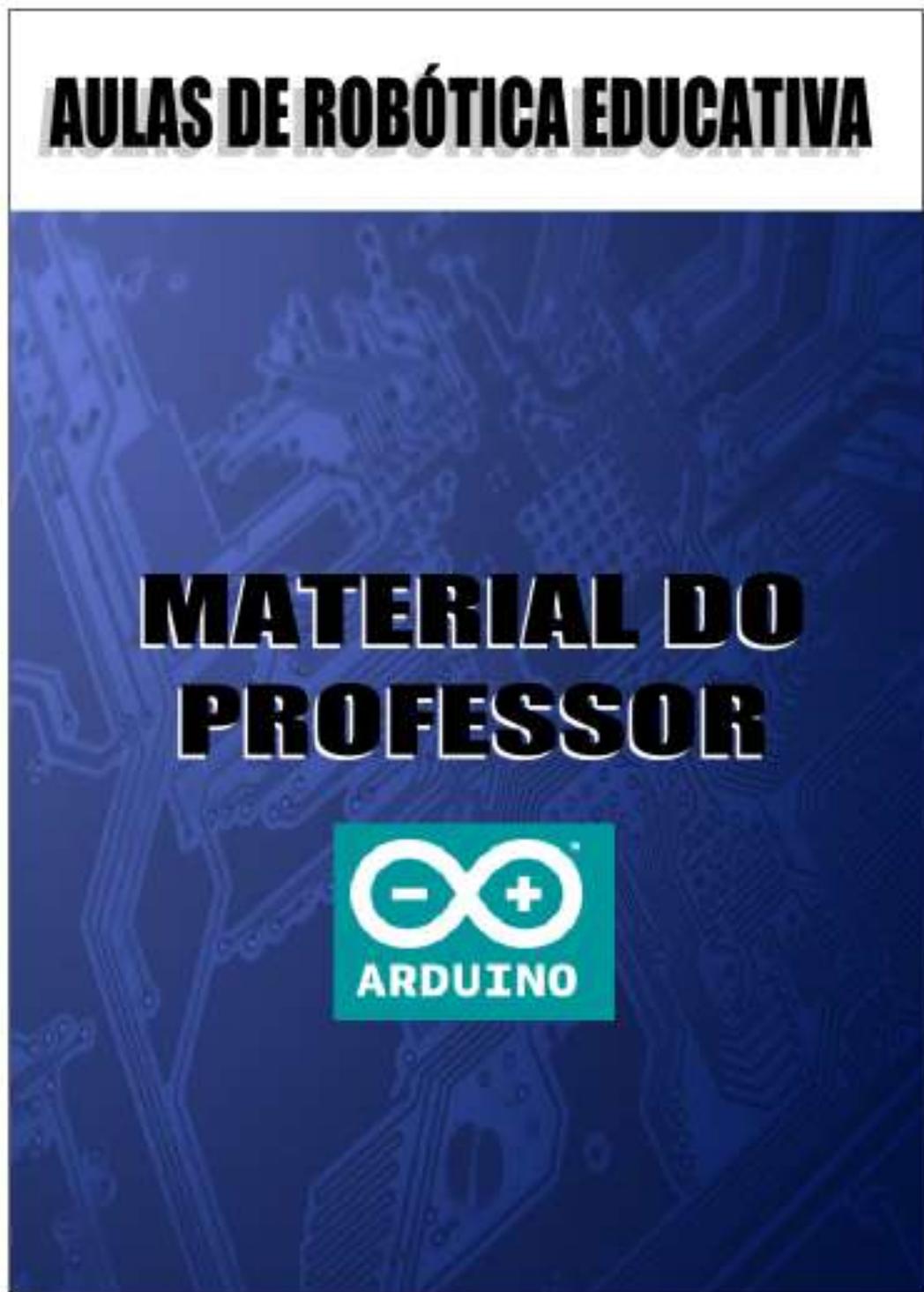
VEX ROBOTICS. **Vex Robotics**. Online. Disponível em <<http://www.vexrobotics.com/vex>>. Acesso em 27 dez. 2012.

W3C. **W3C**. Online. Disponível em <<http://dev.w3.org/html5/websockets/>>. Acesso em 09 abr. 2013.

ZILLI, Silvana. R. **A Robótica Educacional no Ensino Fundamental: Perspetivas e Prática**. Dissertação de Mestrado apresentada à Universidade Federal de Santa Catarina, Florianópolis, 2004.

Anexos

Material completo de apoio desenvolvido para os professores proposto neste modelo de ensino. Há de se destacar que este material de apoio ao professor está em constante atualização.



AULAS DE ROBÓTICA EDUCATIVA

INDICE

Conhecendo as Peças	02
Cuidados com os Componentes	03
Ambiente de Programação	04
Algumas Fórmulas Úteis	10
A Interface Arduino	11
Diodo Emissor de Luz	12
Sensor Ultrassónico	15
Sensor de Luz	16
Sensor de Toque	18
Sensor de Temperatura	19
O Servo Motor	21
Motor de Corrente Contínua	22
Outras Ligações para Motor DC	25
Display de Cristal Líquido	26
Sons com o Arduino e Buzzer	28
Acelerómetro de 3 eixos	30

AULAS DE ROBÓTICA EDUCATIVA

CONHECENDO AS PEÇAS

SENSOR ULTRASSÔNICO HC-SR04



SERVO MOTOR



MICRO SERVO MOTOR



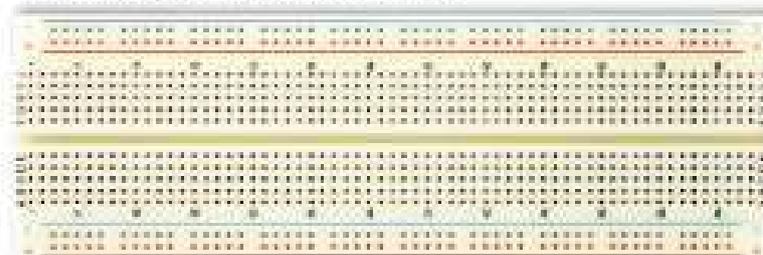
INTERFACE ARDUINO UNO



SENSOR DE TOQUE



MATRIZ DE CONTATOS (PROTOBOARD)



AULAS DE ROBÓTICA EDUCATIVA

CUIDADOS COM OS COMPONENTES

Os componentes que exigem mais cuidados são:

1. A interface Arduino - Deve-se tomar cuidado para não inverter os fios como positivo (VCC) e terra (GND). Além disso, deve-se tomar cuidado para não derrubar a interface de locais como mesas e nem ligar componentes com a placa energizada.
2. O servo motor - O principal cuidado é ligar corretamente os três fios do servo na interface. O fio preto ligado no terra (GND), o fio vermelho ligado ao positivo (5v) e o fio de sinal que pode ser amarelo ou branco ligado em uma das portas PWM da interface.
3. O sensor ultrassônico - Possui quatro pinos na ordem: positivo (5v), Trigger (disparo), Echo (retorno) e terra (GND). Deve-se apenas observar a ligação correta do positivo e do terra.
4. Motor de corrente contínua (DC) - Não deve ser ligado diretamente no Arduino pois cada porta fornece apenas 50mA e um motor deste tipo consome em torno de 500mA. Deve haver um circuito amplificador de corrente ligando este tipo de motor à interface Arduino.

AULAS DE ROBÓTICA EDUCATIVA

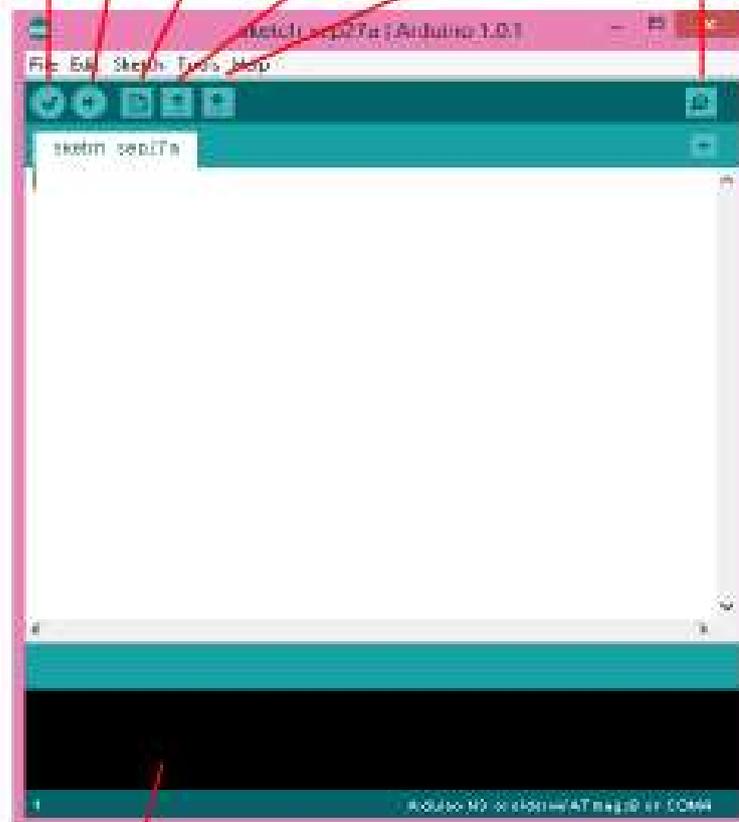
AMBIENTE DE PROGRAMAÇÃO

Compilar e Verificar Programa

Enviar para a Placa Abrir Projeto

Novo Projeto

Salvar Projeto Monitor Serial



Área de Status

AULAS DE ROBÓTICA EDUCATIVA

AMBIENTE DE PROGRAMAÇÃO

Principais elementos da linguagem de programação do Arduino:

#include - Serve para inserir um código escrito em Assembly ou uma função externa na linguagem C.

setup() - É um dos blocos principais e executá apenas uma vez no tempo de vida do programa. No bloco setup inserimos os códigos de configuração das portas do Arduino, de comunicação serial com o PC e todas as demais configurações possíveis no programa.

Exemplo:

```
void setup() {
  bloco de comandos a serem executados apenas uma vez;
}
```

loop() - É o bloco principal do programa que é executado sem parar enquanto o Arduino estiver ligado e energizado. Neste bloco são inseridos todos os comandos, funções e procedimentos.

Exemplo:

```
void loop() {
  bloco de comandos a serem executados;
}
```

pinMode() - Define se uma determinada porta do Arduino será usada para entrada ou saída.

Sintaxe: pinMode(pino, modo);

Exemplo:

```
int ledPin = 13;
void setup() {
  pinMode(ledPin, OUTPUT);
}
```

digitalWrite() - Envia um sinal alto (5v) ou baixo (0v) para uma determinada porta digital do Arduino.

Sintaxe: digitalWrite(pino, valor);

Exemplo:

```
int ledPin = 13;
void setup() {
  pinMode(ledPin, OUTPUT);
}
void loop() {
  digitalWrite(ledPin, HIGH);
}
```

AULAS DE ROBÓTICA EDUCATIVA

AMBIENTE DE PROGRAMAÇÃO

Principais elementos da linguagem de programação do Arduino:

digitalRead() - Recebe um sinal alto (5v) ou baixo (0v) a partir de uma porta digital específica do Arduino.

```
Sintaxe: digitalRead(pino);  
Exemplo:  
int ledPin=13;  
int val=0;  
void setup() {  
  pinMode(ledPin,OUTPUT);  
}  
void loop() {  
  val=digitalRead(ledPin);  
}
```

analogRead() - Recebe um sinal analógico na faixa entre 0 e 1023 a partir de uma das portas analógicas do Arduino. Como cada pino pode fornecer no máximo 5v, então a resolução é de 5/1024 ou 4,9mV (0,0049v).

```
Sintaxe: analogRead(pino);  
Exemplo:  
int analogPin=3;  
int val=0;  
void setup() {  
  Serial.begin(9600);  
}  
void loop() {  
  val=analogRead(analogPin);  
  Serial.println(val);  
}
```

AULAS DE ROBÓTICA EDUCATIVA

AMBIENTE DE PROGRAMAÇÃO

Principais elementos da linguagem de programação do Arduino:

analogWrite() - Envia um sinal analógico na forma de um sinal de pulso de aproximadamente 490hz (PWM) através de uma das portas analógicas.

Sintaxe: `analogWrite(pino, valor);`

Exemplo:

```
int ledPin=8;
int val=0;
void setup() {
  pinMode(ledPin, OUTPUT);
}
void loop() {
  analogWrite(ledPin, val / 4);
}
```

Serial.begin() - Inicia a comunicação através da porta serial com uma velocidade especificada na função. A comunicação serial ocorre pela porta USB ou pelos pinos digitais 0 (RX) e 1 (TX).

Sintaxe: `Serial.begin(velocidade);`

Exemplo:

```
void setup() {
  Serial.begin(9600);
}
```

delay() - Provoca uma interrupção na execução do programa por um tempo de n milissegundos (existem 1000 milissegundos em 1 segundo).

Sintaxe: `delay(tempo);`

Exemplo:

```
int ledPin=13;
void setup() {
  pinMode(ledPin, OUTPUT);
}
void loop() {
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```

AULAS DE ROBÓTICA EDUCATIVA

AMBIENTE DE PROGRAMAÇÃO

Principais elementos da linguagem de programação do Arduino:

millis() - Retorna o número de milissegundos desde que o Arduino começou a executar o programa atual. Este número irá zerar após aproximadamente 50 dias.

```
Sintaxe: var=millis();
Exemplo:
unsigned long tempo;
void setup() {
  Serial.begin(9600);
}
void loop() {
  Serial.print("Tempo: ");
  tempo=millis();
  Serial.println(tempo);
}
```

if() - É um bloco condicional usado para testar uma determinada condição e executar um ou mais comandos caso a condição seja satisfeita ou não.

```
Sintaxe: if(variavel > valor) { bloco de codigos }
Exemplo:
int ledPin=13;
void loop() {
  if (x>120) {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }
}
```

random() - Gera um valor aleatório qualquer ou dentro de uma faixa de valores mínimo e máximo.

```
Sintaxe: random(max); ou random(min,max);
Exemplo:
long numero;
void setup() {
  Serial.begin(9600);
}
void loop() {
  numero=random(300);
  Serial.println(numero);
}
```

AULAS DE ROBÓTICA EDUCATIVA

AMBIENTE DE PROGRAMAÇÃO

Principais elementos da linguagem de programação do Arduino:

for() - Repete um bloco de declarações dentro de uma quantidade definida de vezes.

Sintaxe: `for (int var = valor; var <operador> valor; var++ ou var--)`

Exemplo:

```
int pino = 10;
void loop() {
  for (int i=0; i<255; i++) {
    analogWrite(pino, i);
    delay(10);
  }
}
```

while() - Executa indefinidamente uma repetição até que uma expressão dentro dos parênteses seja falsa.

Sintaxe: `while(expressão) { bloco de declarações }`

Exemplo:

```
int var=0;
while (var < 200) {
  var++;
}
```

switch() - Semelhante ao comando if, mas permite o teste de múltiplas condições, comparando o valor de uma variável com valores explicitados em cada declaração "case". Após cada "case" deve-se colocar uma declaração "break" para sair do bloco.

Sintaxe: `switch (variavel) { case teste_variavel: bloco de código }`

Exemplo:

```
int var=0;
switch (var) {
  case 1:
    //bloco a executar
    break;
  case 2:
    //bloco a executar2
    break;
  default:
    //este bloco é opcional
}
```

AULAS DE ROBÓTICA EDUCATIVA

ALGUMAS FÓRMULAS ÚTEIS

Cálculo de resistor para ligar um LED:

Para ligar um ou mais LEDs (diodo emissor de luz) numa determinada fonte de energia é fundamental colocar um resistor em série com o LED para limitar o valor da corrente evitando a queima do LED. Na maioria dos projetos a fonte de energia fica em torno de 6v e um LED consome em média 1,6v com uma corrente máxima de 0,02A (200mA).

Primeiro deve-se subtrair a queda de tensão provocada pelo LED:

$$V_x = V - V_{led} \quad V_x = 6 - 1,8 \quad V_x = 4,2$$

Depois encontramos o valor aproximado do resistor:

$$R = V / I \quad R = 4,2 / 0,02 \quad R = 210 \text{ Ohms}$$

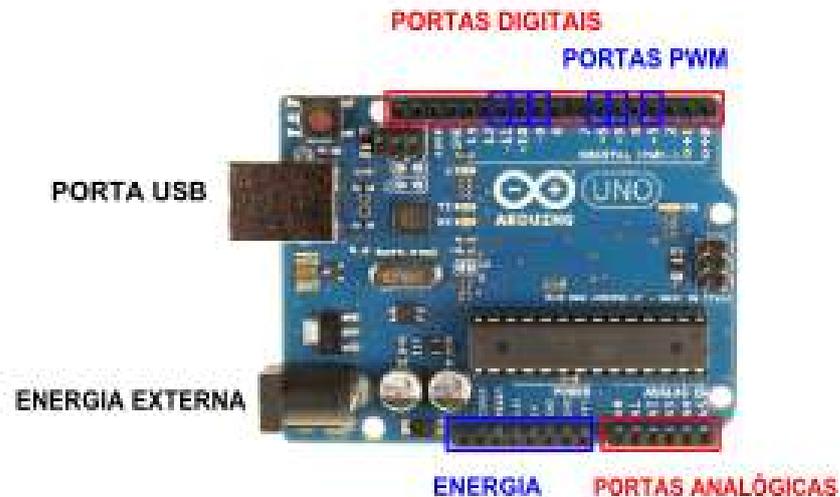
Cálculo para converter temperaturas usando o sensor LM35.

Para cada grau celsius recebido, a saída do sensor aumenta 10mV, gerando 1024 possíveis valores para a temperatura. Na prática ele lê valores de até 500 graus:

$$\text{Temp} = (500 * \text{sensor}) / 1023$$

AULAS DE ROBÓTICA EDUCATIVA

A INTERFACE ARDUINO



A figura acima mostra os principais elementos do Arduino. As portas digitais e analógicas são usadas para ligar os motores e sensores. Os conectores de energia são usados para ligar um pino positivo de 5v e outro de 3,3v, dois pinos para o terra e outro para a fonte de energia externa que pode chegar até 12v.

Além disso, as portas digitais 0 e 1 também são usadas para transmissão (TX) e recepção (RX) de bits.

ATENÇÃO: A fonte de energia externa deve ter no máximo 12v para evitar danos ao Arduino.

AULAS DE ROBÓTICA EDUCATIVA

DIODO EMISSOR DE LUZ



O diodo emissor de luz ou LED (Light Emission Diode) é um componente eletrônico dos mais comuns e úteis para emitir luz em várias cores e intensidades. É um componente de fácil aquisição, barato e que economiza muita energia. Porém, devemos tomar alguns cuidados ao utilizar um LED:

1. Ele é um diodo, portanto, tem polaridade (lado positivo e negativo) e se for ligado ao contrário pode queimar.
2. Ele sempre deve ser ligado em série com um resistor ao seu circuito para evitar queima do componente.
3. Dependendo do tamanho e da cor, o consumo de energia do LED varia.

Na página 9 temos as formas de uso do LED e exemplos de ligação com o Arduino.

AULAS DE ROBÓTICA EDUCATIVA

DIODO EMISSOR DE LUZ

Código de Cores
Este resistor tem uma tolerância superior para o valor de Ω.

Cor	1ª Faixa	2ª Faixa	3ª Faixa	Multiplicador	Tolerância
Preto	0	0	0	x 10 ⁰	± 1%
Marrão	1	1	0	x 10 ¹	± 1%
Vermelho	2	2	0	x 10 ²	± 2%
Laranja	3	3	0	x 10 ³	± 3%
Amarelo	4	4	0	x 10 ⁴	± 4%
Verde	5	5	0	x 10 ⁵	± 0,5%
Azul	6	6	0	x 10 ⁶	± 0,2%
Violeta	7	7	0	x 10 ⁷	± 0,1%
Preto	8	8	0	x 10 ⁸	± 0,05%
Branco	9	9	0	x 10 ⁹	± 0,01%
Prateado				x 10 ⁻¹	± 10%

Para ligar um LED a qualquer circuito é preciso saber qual a tensão e a corrente que seu LED precisa para funcionar. Neste exemplo vamos usar o Arduino ligado diretamente na porta USB, que fornece 5V. Usaremos um LED vermelho comum que provoca uma queda de tensão de 1,8V e corrente de 0,02A. Se o LED causa uma queda de tensão de 1,8V, então a tensão nos terminais dele é a que sobra da fonte de energia, ou seja, $5 - 1,8 = 3,2$. Usando uma fórmula simples podemos calcular o valor do resistor a ser utilizado:

Lei de Ohm:

$$\text{RESISTÊNCIA} = \frac{\text{TENSÃO}}{\text{CORRENTE}}$$

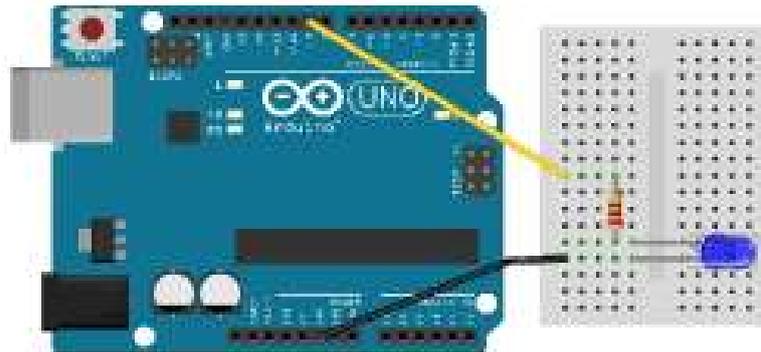
Substituindo os valores:

$$R = \frac{3,2}{0,02} \Rightarrow R = 160$$

Como o circuito possui uma margem para mais ou para menos, podemos utilizar um resistor comum de 220 ohms sem problemas.

AULAS DE ROBÓTICA EDUCATIVA

DIODO EMISSOR DE LUZ



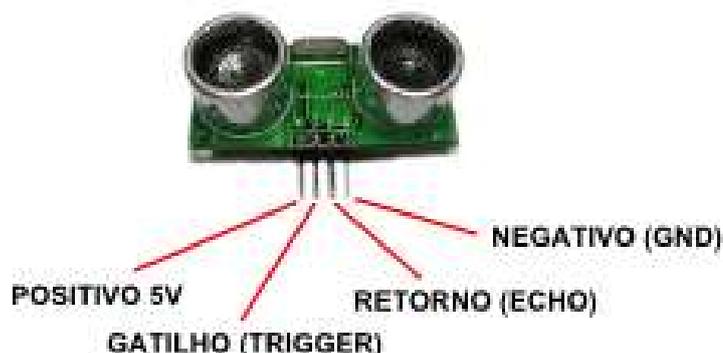
A imagem acima ilustra uma forma simples de ligar um LED comum em série com um resistor a uma das portas digitais do Arduino. Para ligar mais LEDs basta repetir o esquema mostrado neste exemplo, lembrando de compartilhar o pino terra (GND) e cada LED ligar em uma porta digital ou analógica do Arduino.

LISTA DE PEÇAS:

- 01 Arduino
- 01 LED comum
- 01 Resistor de 220 ohms
- 02 Fios de ligação

AULAS DE ROBÓTICA EDUCATIVA

SENSOR ULTRASSONICO



O sensor ultrassonico é usado para medir distâncias. Este é o sensor HC-SR04 que possui 4 conectores na ordem mostrada na figura. O retorno e o gatilho podem ser ligados em qualquer porta digital do Arduino. O exemplo de código a seguir mostra como medir distâncias em centímetros e polegadas.

```
#define gatilho 13
#define retorno 12

void setup() {
  Serial.begin(9600);
  pinMode(gatilho, OUTPUT);
  pinMode(retorno, INPUT);
}

void loop() {
  float valor, dist;
  digitalWrite(gatilho, LOW);
  delayMicroseconds(2);
  digitalWrite(gatilho, HIGH);
  delayMicroseconds(10);
  digitalWrite(gatilho, LOW);
  valor=pulseIn(retorno, HIGH);
  dist=(valor/2) / 28.1;
  Serial.print(dist);
  Serial.println();
}
```

AULAS DE ROBÓTICA EDUCATIVA

SENSOR DE LUZ



A figura acima mostra um sensor de luz do tipo LDR (light Dependent Resistor), ou resistor variável dependente de luz, que deve ser usado geralmente em pares para criar, por exemplo, um robô seguidor de linha. Este sensor deve ser soldado a dois fios ou ligado na matriz de contatos e ligado a qualquer porta analógica do Arduino. O exemplo de código a seguir mostra como usar o LDR como sensor de luz.

```
int sensor = 0;
int valor = 0;

void setup() {
  Serial.begin(9600);
  pinMode(sensor, INPUT);
}

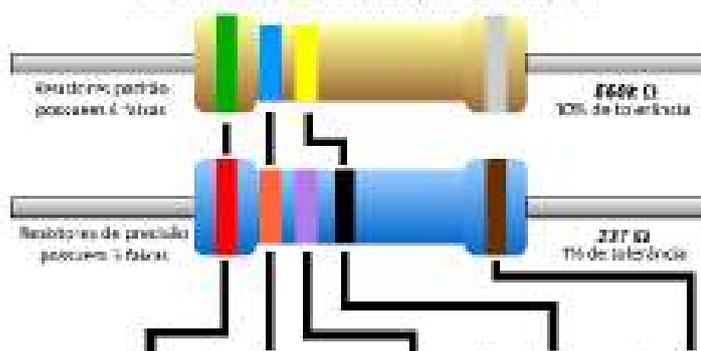
void loop() {
  valor=analogRead(sensor);
  Serial.print("O valor do LDR é igual a: ");
  Serial.print(valor);
  delayMicroseconds(500);
}
```

AULAS DE ROBÓTICA EDUCATIVA

SENSOR DE LUZ

Código de Cores

A resistência com mais faixas deve apontar para a esquerda.



Cor	1ª Faixa	2ª Faixa	3ª Faixa	Multiplicador	Tolerância
Preto	0	0	0	x 1 Ω	
Marron	1	1	1	x 10 Ω	+/- 1%
Vermelha	2	2	2	x 100 Ω	+/- 2%
Laranja	3	3	3	x 1K Ω	
Amarelo	4	4	4	x 10K Ω	
Verde	5	5	5	x 100K Ω	+/- 5%
Azul	6	6	6	x 1M Ω	+/- 20%
Violeta	7	7	7	x 10M Ω	+/- 1%
Cinza	8	8	8		+/- 0,5%
Branco	9	9	9		
Outorado				x 1 Ω	+/- 5%
Prateado				x 0,1 Ω	+/- 10%

A sensibilidade de um sensor de luz gera uma variação de resistência muito baixa para que a porta analógica do Arduino consiga ler de forma correta. Para termos uma leitura mais confiável e acertada é preciso colocar em série um resistor com o sensor de luz (LDR) criando um divisor de tensão. Para este exemplo usamos um resistor de 10K. A tabela acima mostra como identificar as cores em qualquer resistor.

AULAS DE ROBÓTICA EDUCATIVA

SENSOR DE TOQUE



A figura acima mostra uma chave de fim de curso que pode ser usada como sensor de toque para um robô que "sinta" quando bate em algum obstáculo. Este sensor deve ser soldado a dois fios ou ligado na matriz de contatos e ligado a qualquer porta digital do Arduino. O exemplo de código a seguir mostra como usar o sensor de toque ligado na porta digital 2 para ligar ou desligar um motor de corrente contínua ligado na porta 5 do Arduino.

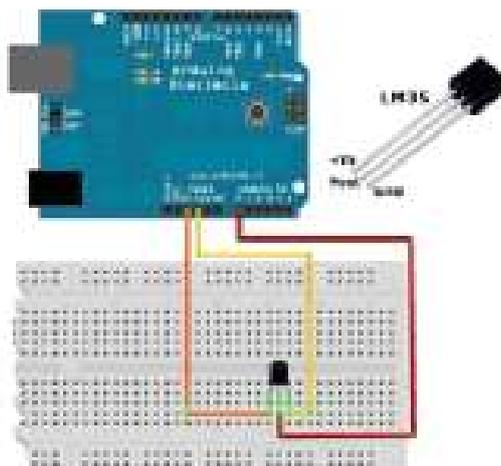
```
int sensor = 2;
int motor = 5;
int valor = 0;

void setup() {
  Serial.begin(9600);
  pinMode(sensor, INPUT);
  pinMode(motor, OUTPUT);
}

void loop() {
  valor = digitalRead(sensor);
  if (valor == 0) {
    digitalWrite(motor, HIGH);
    delayMicroseconds(1000);
  } else {
    digitalWrite(motor, LOW);
    delayMicroseconds(1000);
  }
  delayMicroseconds(500);
}
```

AULAS DE ROBÓTICA EDUCATIVA

SENSOR DE TEMPERATURA



A figura acima mostra um transistor LM35 usado como sensor de temperatura e seu esquema de ligação. Este sensor deve ser soldado a 3 fios ou conectado a uma matriz de contatos e em seguida a qualquer porta analógica do Arduino.

Para ser lido pelo Arduino necessita que haja uma conversão, pois o Arduino só lê valores inteiros entre 0 e 1023. Assim, sabendo que este sensor tem uma resolução de 10mV para cada 1°C o sensor será alimentado pelo Arduino que fornece 5V, como as entradas analógicas têm uma resolução de 10 bits, ou seja, $2^{10} = 1024$ e cada grau corresponde a 10mV então a temperatura em função do valor lido na porta analógica do Arduino será:

$$\text{temperatura} = (\text{valor lido} \times (5 / 1023)) \times 100$$

$$\text{temperatura} = 0,0048875855 \times \text{valor lido}$$

AULAS DE ROBÓTICA EDUCATIVA

SENSOR DE TEMPERATURA

O código abaixo ilustra um exemplo de utilização do sensor de temperatura para medir a temperatura do ambiente em graus celsius:

```
int sensor = 0;
int valorLido = 0;
float temperatura = 0;

void setup() {
  serial.Begin(9600);
}

void loop() {
  valorLido = analogRead(sensor);
  temperatura = (valorLido*0.00488);
  temperatura = temperatura * 100;
  Serial.print("Temperatura atual: ");
  Serial.println(temperatura);
  delayMilliseconds(1000);
}
```

AULAS DE ROBÓTICA EDUCATIVA

O SERVO MOTOR



A figura acima mostra o servo motor e seus conectores. O fio preto deve ser ligado ao terra (GND), o fio vermelho ao positivo (5v) e o fio amarelo ou branco a qualquer porta PWM do Arduino. Para usar este tipo de motor é necessário fazer uma chamada a biblioteca `servo.h` que se encontra na pasta do Arduino. O exemplo de código a seguir mostra como movimentar um servo motor de posição.

```
#include <Servo.h>
Servo servo1;

void setup() {
  servo1.attach(9);
  serial.Begin(9600);
  servo1.write(0);
}

void loop() {
  delayMilliseconds(1000);
  servo1.write(90);
  delayMilliseconds(1000);
  servo1.write(-90);
  delayMilliseconds(1000);
  servo1.write(180);
  delayMilliseconds(1000);
  servo1.write(-180);
}
```

AULAS DE ROBÓTICA EDUCATIVA

O MOTOR DE CORRENTE CONTÍNUA



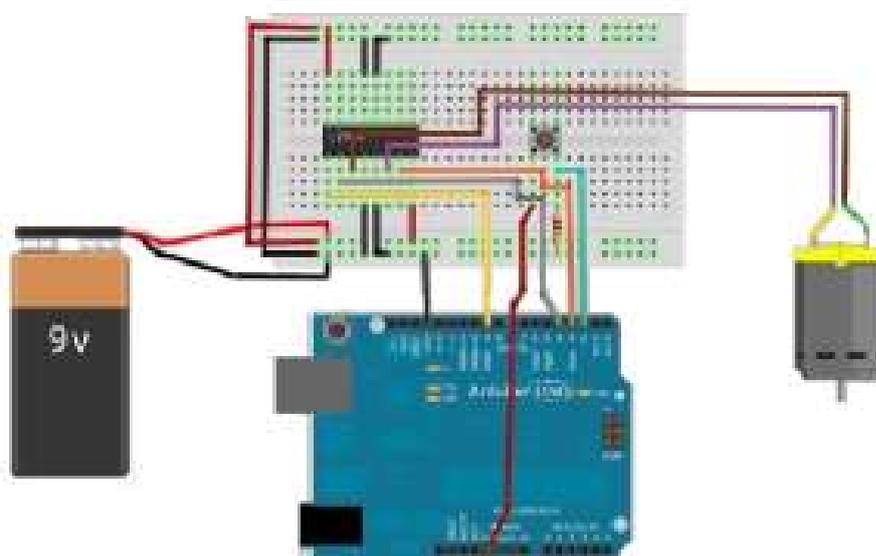
A figura acima mostra um motor de corrente contínua (DC). Não é possível ligar o motor diretamente em uma das portas da interface Arduino pois cada motor drena em torno de 500mA, enquanto que cada porta só fornece no máximo 50mA. É preciso ter um circuito amplificador de corrente para ligar qualquer motor de corrente contínua no Arduino. O exemplo da página 16 ilustra como fazer funcionar um motor DC ligado a um circuito integrado chamado ponte H que irá controlar o funcionamento do motor DC.

ATENÇÃO: LIGAR O MOTOR DC DIRETO NO ARDUINO PODE QUEIMAR A SUA PLACA!

AULAS DE ROBÓTICA EDUCATIVA

O MOTOR DE CORRENTE CONTÍNUA

Em alguns projetos é preciso usar outros componentes eletrônicos junto com o Arduino. Um exemplo é a ponte H formada pelo CI L293D para controle de até dois motores DC. A figura abaixo mostra um exemplo de ligação deste CI com o Arduino.



Na figura os principais pinos usados no Arduino são: 3, 4, 9, 5v e GND.

AULAS DE ROBÓTICA EDUCATIVA

O MOTOR DE CORRENTE CONTÍNUA

O exemplo de código a seguir ilustra uma maneira de acionar o motor DC nos dois sentidos de rotação:

```
int motorPin1 = 5;
int motorPin2 = 6;
int entrada = 0;

void setup() {
  Serial.begin(9600);
  pinMode(motorPin1, OUTPUT);
  pinMode(motorPin2, OUTPUT);
}

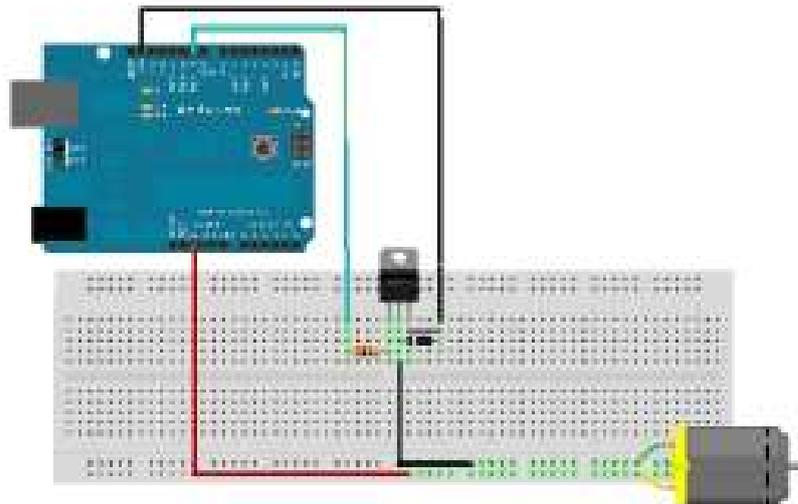
void loop() {
  if (Serial.available() > 0) {
    entrada = Serial.read();
    if (entrada == 'e') {
      esquerda();
    } else if (entrada == 'd') {
      direita();
    } else {
      parar();
    }
  }
}

void esquerda() {
  digitalWrite(motorPin1, 1); // Motor para esquerda
  digitalWrite(motorPin2, 0); // Motor parado.
}

void direita() {
  digitalWrite(motorPin1, 0); // Motor para direita
  digitalWrite(motorPin2, 1); // Motor parado.
}
```

AULAS DE ROBÓTICA EDUCATIVA

OUTRA LIGAÇÃO PARA MOTOR DC



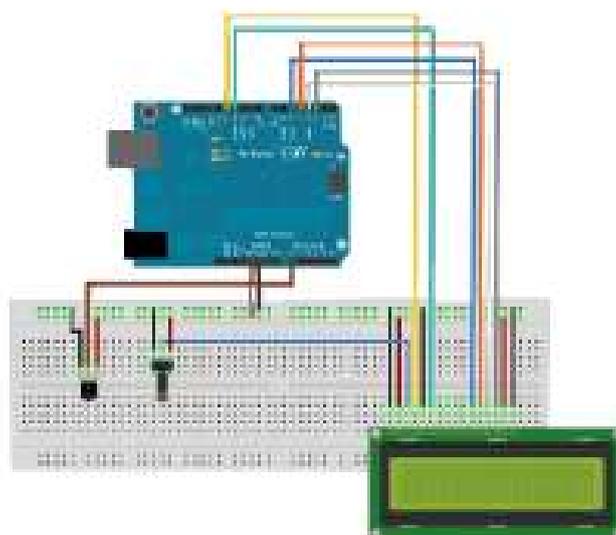
Na figura acima temos outra maneira de ligar um motor DC ao Arduino sem a utilização de uma ponte H. Para conseguir amplificar a corrente neste projeto foi utilizado um transistor TIP 122 ou TIP 120. Também deve-se utilizar um diodo que servirá como proteção para a placa do Arduino, pois uma parte da carga poderá voltar para a porta que controla o motor e danifica-la.

Lista de Materiais:

- 01 Arduino
- 01 Motor DC 5v
- 01 Transistor TIP 122 ou 120
- 01 Resistor de 2,2k
- 01 Diodo IN4004 ou IN4007
- Alguns pedaços de fio para ligação.

AULAS DE ROBÓTICA EDUCATIVA

DISPLAY DE CRISTAL LÍQUIDO



Este componente é usado para exibir informações em uma pequena tela digital. Existem display de vários tamanhos e tipos, onde geralmente varia a cor e a quantidade de caracteres que pode ser exibida de cada vez. Usaremos uma biblioteca própria do Arduino para controlar este dispositivo que possui 2 linhas e 16 caracteres por linha.

Lista de Materiais:

- 01 Arduino
- 01 Sensor de temperatura LM35
- 01 Display LCD
- 01 Protoboard
- 01 Potenciômetro de 10K
- Alguns pedaços de fio para ligação.

AULAS DE ROBÓTICA EDUCATIVA

DISPLAY DE CRISTAL LÍQUIDO

O exemplo de código a seguir ilustra uma maneira de utilizar o display LCD com um sensor de temperatura:

```
#include <LiquidCrystal.h>
// Define o pino A0 como sensor
#define sensor 0
// Variável para ler o sinal do sensor
int Ventrada;
// Variável que recebe o valor convertido para temperatura
float Temperatura;
// pinos do Arduino que serão usados pelo LCD
LiquidCrystal lcd (12, 11, 5, 4, 3, 2);

void setup() {
  // informa que o LCD possui 2 linhas com 16 caracteres
  lcd.begin(16, 2);
  lcd.print("Temperatura: ");
}

void loop() {
  Ventrada = analogRead (sensor);
  // converte o valor de tensão em temperatura
  Temperatura=(500*Ventrada)/1023;
  // move o cursor do LCD para a próxima linha
  lcd.setCursor(0, 1);
  lcd.print(Temperatura);
  lcd.print(" C");
  // aguarda 1 segundo para fazer nova leitura
  delay(1000);
}
```

Material didático completo para os alunos proposto neste modelo de ensino. Da mesma forma que o material do professor, este conteúdo para os alunos também está em constante atualização.



AULAS DE ROBÓTICA EDUCATIVA

INDICE

Conhecendo as Peças	03
Cuidados com as Peças	05
A Interface do Arduino	06
Projeto 01: Diodo Emissor de Luz	07
Projeto 01: Semáforo	09
Projeto 02: Mecanismo de Direção	09
Projeto 03: Garra com Servo Motor	12
Projeto 04: Veículo com Sensor de Distância	15
Projeto 05: Robô Quadrupede com servo motor	19
Projeto 06: Braço Robótico	22

AULAS DE ROBÓTICA EDUCATIVA

CONHECENDO AS PEÇAS

ATUADORES

São elementos responsáveis por movimentar as partes de um robô ou mecanismo. Os atuadores podem ser motores, músculos eletrônicos, pistões hidráulicos ou pneumáticos.



Servo Motor



Motor de Passo



Pistão Pneumático

SENSORES

São componentes eletrônicos e mecânicos usados para que o robô "sinta" o meio ambiente ao seu redor. Os sensores podem ser externos quando ligam o robô ao ambiente ou internos quando servem para informar ao robô sobre o funcionamento de alguma de suas partes.



Sensor de Luz



Sensor de Distância



Sensor de Toque

COMPONENTES ELETRÔNICOS

São os vários elementos usados em conjunto com os sensores, atuadores e interface para formar a parte eletrônica do robô. Estes componentes podem ser adquiridos em lojas especializadas ou aproveitados de material de sucata.



Resistor



Diodo



Transistor



Circuito Integrado



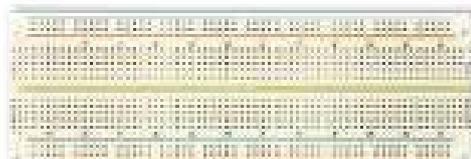
Painel Solar

AULAS DE ROBÓTICA EDUCATIVA

CONHECENDO AS PEÇAS

MATRIZ DE CONTATOS (PROTOBOARD)

São elementos responsáveis por movimentar as partes de um robô ou mecanismo. Os atuadores podem ser motores, músculos eletrônicos, pistões hidráulicos ou pneumáticos.



INTERFACE ARDUINO

São componentes eletrônicos e mecânicos usados para que o robô "sinta" o meio ambiente ao seu redor. Os sensores podem ser externos quando ligam o robô ao ambiente ou internos quando servem para informar ao robô sobre o funcionamento de alguma de suas partes.



FERRAMENTAS

São usadas para ajudar a manipular alguns dos outros componentes, para ajustar encaixes, cortar fios, medir grandezas elétricas, etc.



AULAS DE ROBÓTICA EDUCATIVA

CUIDADOS COM AS PEÇAS

Os componentes que exigem mais cuidados são:

1. A interface Arduino - Deve-se tomar cuidado para não inverter os fios como positivo (VCC) e terra (GND). Além disso, deve-se tomar cuidado para não derrubar a interface de locais como mesas e nem ligar componentes com a placa energizada.
2. O servo motor - O principal cuidado é ligar corretamente os três fios do servo na interface. O fio preto ligado no terra (GND), o fio vermelho ligado ao positivo (5v) e o fio de sinal que pode ser amarelo ou branco ligado em uma das portas PWM da interface.
3. O sensor ultrassônico - Possui quatro pinos na ordem: positivo (5v), Trigger (disparo), Echo (retorno) e terra (GND). Deve-se apenas observar a ligação correta do positivo e do terra.
4. Motor de corrente contínua (DC) - Não deve ser ligado diretamente no Arduino pois cada porta fornece apenas 50mA e um motor deste tipo consome em torno de 500mA. Deve haver um circuito amplificador de corrente ligando este tipo de motor à interface Arduino.

AULAS DE ROBÓTICA EDUCATIVA

A INTERFACE DO ARDUINO

Compilar e Verificar Programa

Enviar para a Placa

Abrir Projeto

Novo Projeto

Salvar Projeto

Monitor Serial



Área de Status

AULAS DE ROBÓTICA EDUCATIVA

DIODO EMISSOR DE LUZ



O diodo emissor de luz ou LED (Light Emission Diode) é um componente eletrônico dos mais comuns e úteis para emitir luz em várias cores e intensidades. É um componente de fácil aquisição, barato e que economiza muita energia. Porém, devemos tomar alguns cuidados ao utilizar um LED:

1. Ele é um diodo, portanto, tem polaridade (lado positivo e negativo) e se for ligado ao contrário pode queimar.
2. Ele sempre deve ser ligado em série com um resistor ao seu circuito para evitar queima do componente.
3. Dependendo do tamanho e da cor, o consumo de energia do LED varia.

Na página 9 temos as formas de uso do LED e exemplos de ligação com o Arduino.

AULAS DE ROBÓTICA EDUCATIVA

DIODO EMISSOR DE LUZ

Código de Cores
A identificação das faixas de cores indica o valor da resistência

Cor	1ª Faixa	2ª Faixa	3ª Faixa	Multiplicador	Tolerância
Preto	0	0	0	x 1	± 1%
Marrom	1	1	1	x 10	± 1%
Vermelho	2	2	2	x 100	± 2%
Amarelo	3	3	3	x 1.000	± 5%
Verde	4	4	4	x 10.000	± 0,5%
Azul	5	5	5	x 100.000	± 0,2%
Violeta	6	6	6	x 1.000.000	± 0,1%
Laranja	3	3	3	x 100	± 0,05%
Branco	9	9	9	x 10	± 0,1%
Dourado				x 0,1	± 5%
Prateado				x 0,01	± 10%

Para ligar um LED a qualquer circuito é preciso saber qual a tensão e a corrente que seu LED precisa para funcionar. Neste exemplo vamos usar o Arduino ligado diretamente na porta USB, que fornece 5V. Usaremos um LED vermelho comum que provoca uma queda de tensão de 1,8V e corrente de 0,02A. Se o LED causa uma queda de tensão de 1,8V, então a tensão nos terminais dele é a que sobra da fonte de energia, ou seja, $5 - 1,8 = 3,2$. Usando uma fórmula simples podemos calcular o valor do resistor a ser utilizado:

Lei de Ohm:

$$\text{RESISTÊNCIA} = \frac{\text{TENSÃO}}{\text{CORRENTE}}$$

Substituindo os valores:

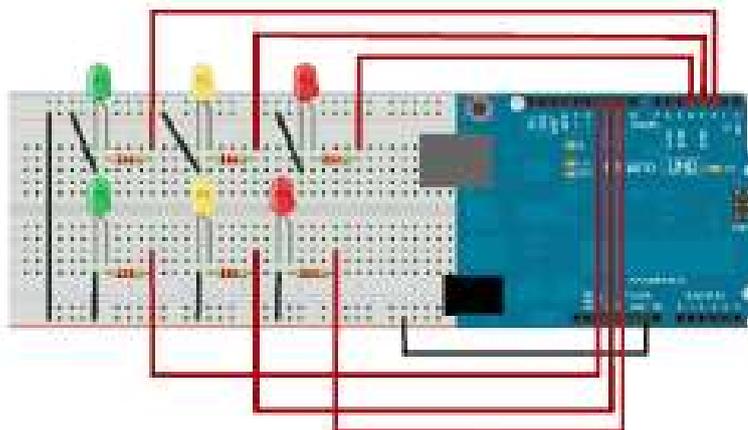
$$R = \frac{3,2}{0,02} \implies R = 160$$

Como o circuito possui uma margem para mais ou para menos, podemos utilizar um resistor comum de 220 ohms sem problema.

AULAS DE ROBÓTICA EDUCATIVA

SEMÁFORO TEMPORIZADO

A atividade proposta neste projeto é criar um modelo de semáforo inteligente com controle de tempo através da interface Arduino. Com o uso de LEDs coloridos, além de alguns resistores de 220 Ohms é possível fazer uma simulação de cruzamento de ruas, por exemplo. A imagem abaixo ilustra as conexões elétricas entre a interface e os dois semáforos.



LISTA DE MATERIAL:

- 02 LEDs vermelhos
- 02 LEDs verdes
- 02 LEDs amarelos
- Pedaços de fio para ligações
- 01 Matriz de contatos
- 01 Interface Arduino
- 06 Resistores de 220 Ohms

AULAS DE ROBÓTICA EDUCATIVA

SEMÁFORO TEMPORIZADO

PROGRAMAÇÃO:

```

// Projeto Semáforo
const int Verde1 = 2;
const int Amarelo1 = 3;
const int Vermelho1 = 4;
const int Verde2 = 8;
const int Amarelo2 = 10;
const int Vermelho2 = 11;
int tempo = 0;

void setup() {
  pinMode(Verde1, OUTPUT);
  pinMode(Amarelo1, OUTPUT);
  pinMode(Vermelho1, OUTPUT);
  pinMode(Verde2, OUTPUT);
  pinMode(Amarelo2, OUTPUT);
  pinMode(Vermelho2, OUTPUT);
}

void loop() {
  switch(tempo) {
    case 0: // verde1 aceso e vermelho 2 apagado
      digitalWrite(Verde1, HIGH);
      digitalWrite(Amarelo1, LOW);
      digitalWrite(Vermelho1, LOW);
      digitalWrite(Verde2, LOW);
      digitalWrite(Amarelo2, LOW);
      digitalWrite(Vermelho2, HIGH);
      tempo++;
      delayMicroseconds(1000);
      break;
    case 1: // Amarelo1 e 2 aceso, verde1 e vermelho 2 apagado
      digitalWrite(Verde1, LOW);
      digitalWrite(Amarelo1, HIGH);
      digitalWrite(Vermelho1, LOW);
      digitalWrite(Verde2, LOW);
      digitalWrite(Amarelo2, HIGH);
      digitalWrite(Vermelho2, LOW);
      tempo++;
      delayMicroseconds(1000);
      break;
    case 2: // verde1 apagado e vermelho 2 apagado
      digitalWrite(Verde1, LOW);
      digitalWrite(Amarelo1, LOW);
      digitalWrite(Vermelho1, HIGH);
      digitalWrite(Verde2, HIGH);
      digitalWrite(Amarelo2, LOW);
      digitalWrite(Vermelho2, LOW);
      tempo++;
      delayMicroseconds(1000);
      break;
  }
}

```

AULAS DE ROBÓTICA EDUCATIVA

SEMÁFORO TEMPORIZADO

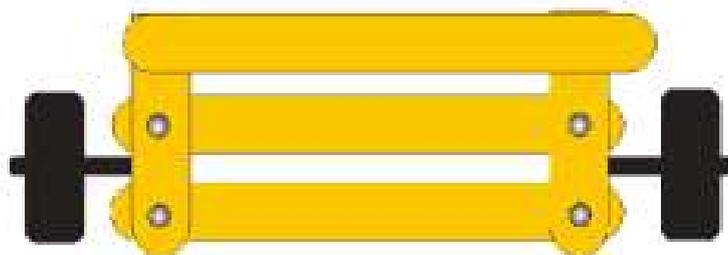
RELATÓRIO DE APRENDIZADO E DÚVIDAS DA EQUIPE:

Escreva aqui o que a equipe aprendeu, suas dúvidas e eventuais sugestões de mudanças ou melhorias neste projeto.

AULAS DE ROBÓTICA EDUCATIVA

MECANISMO DE DIREÇÃO

A atividade proposta neste projeto tem como objetivo criar um mecanismo semelhante ao sistema de direção de um automóvel, onde é possível fazer a mudança de direção para esquerda e para a direita utilizando apenas um servo motor pequeno. O esquema da figura abaixo ilustra como este tipo de mecanismo pode funcionar.



LISTA DE MATERIAL:

- 01 Servo motor pequeno
- 08 Pedacos de madeira leve
- 02 Palitos de churrasco
- 02 Pneus plásticos ou de borracha
- 08 Parafusos pequenos
- 08 Porcas pequenas
- 01 Interface Arduino

AULAS DE ROBÓTICA EDUCATIVA

MECANISMO DE DIREÇÃO

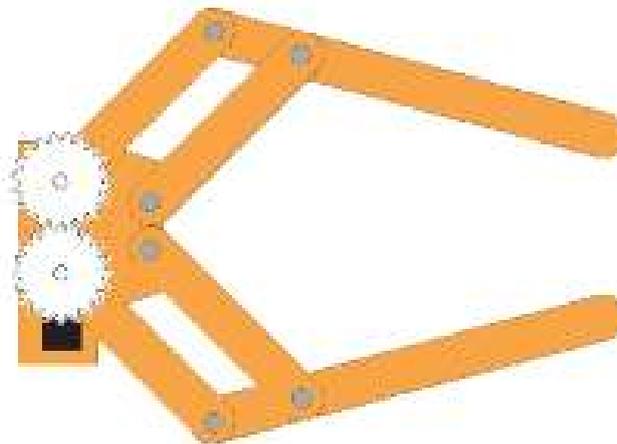
RELATÓRIO DE APRENDIZADO E DÚVIDAS DA EQUIPE:

Escreva aqui o que a equipe aprendeu, suas dúvidas e eventuais sugestões de mudanças ou melhorias neste projeto.

AULAS DE ROBÓTICA EDUCATIVA

GARRA COM SERVO MOTOR

A atividade proposta neste projeto tem como objetivo criar uma garra que abre e fecha para segurar pequenos objetos feita com pedaços de madeira leve utilizando apenas um servo motor pequeno e duas engrenagens. O esquema da figura abaixo ilustra como este tipo de mecanismo pode funcionar.



LISTA DE MATERIAL:

- 01 Servo motor pequeno
- 08 Pedaços de madeira leve
- 01 Palito de churrasco
- 02 Engrenagens médias
- 06 Parafusos pequenos
- 06 Porcas pequenas
- 01 Interface Arduino

AULAS DE ROBÓTICA EDUCATIVA

GARRA COM SERVO MOTOR

PROGRAMAÇÃO:

```

// Projeto Sensores
const int Verde1 = 2;
const int Amarelo1 = 3;
const int Vermelho1 = 4;
const int Verde2 = 5;
const int Amarelo2 = 10;
const int Vermelho2 = 11;
int inicio = 0;

void setup() {
  pinMode(Verde1, OUTPUT);
  pinMode(Amarelo1, OUTPUT);
  pinMode(Vermelho1, OUTPUT);
  pinMode(Verde2, OUTPUT);
  pinMode(Amarelo2, OUTPUT);
  pinMode(Vermelho2, OUTPUT);
}

void loop() {
  switch(inicio) {
    case 0: // Verde1 aceso e Vermelho 2 apagado
      digitalWrite(Verde1, HIGH);
      digitalWrite(Amarelo1, LOW);
      digitalWrite(Vermelho1, LOW);
      digitalWrite(Verde2, LOW);
      digitalWrite(Amarelo2, LOW);
      digitalWrite(Vermelho2, HIGH);
      inicio++;
      delayMicroseconds(1000);
      break;

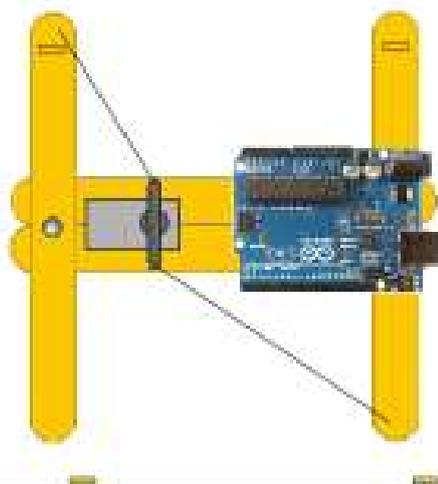
    case 1: // Amarelo1 e 2 acesos, Verde1 e Vermelho 2 apagados
      digitalWrite(Verde1, LOW);
      digitalWrite(Amarelo1, HIGH);
      digitalWrite(Vermelho1, LOW);
      digitalWrite(Verde2, LOW);
      digitalWrite(Amarelo2, HIGH);
      digitalWrite(Vermelho2, LOW);
      inicio++;
      delayMicroseconds(1000);
      break;

    case 2: // Verde1 apagado e Vermelho 2 apagado
      digitalWrite(Verde1, LOW);
      digitalWrite(Amarelo1, LOW);
      digitalWrite(Vermelho1, HIGH);
      digitalWrite(Verde2, HIGH);
      digitalWrite(Amarelo2, LOW);
      digitalWrite(Vermelho2, LOW);
      inicio=0;
      delayMicroseconds(1000);
      break;
  }
}

```

AULAS DE ROBÓTICA EDUCATIVA

ROBÔ QUADRUPEDE COM SERVO



Lista de Material:

- | | |
|---------------------------------|----------------------------|
| 05 Palitos de picolé | Alicate de bico |
| 02 Parafusos pequenos | Estilete |
| 02 Clips de papel tamanho médio | Cola branca ou de madeira |
| 01 Micro servo motor de 9g | chave de fenda |
| 01 Arduino | régua |
| 01 Bateria de 9V | lâpis |
| 01 Clip para bateria de 9V | Furadeira ou mini retífica |
| 03 Pedacos de fio fino | |
| 01 Liga de elástico | |

Passo1: Separe 1 palito de picolé e corte-o com o estilete em 4 pedaços conforme a figura 01 abaixo. CUIDADO AO MANUSEAR O ESTILETE POIS HÁ RISCO DE CORTES!

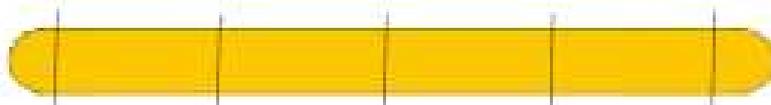


Figura 01

AULAS DE ROBÓTICA EDUCATIVA

ROBÔ QUADRUPEDE COM SERVO

Passo 2: Separe 2 palitos de picolé e cole um do lado do outro usando 2 pedaços de palito que foram cortados no passo 1, conforme a figura 2 abaixo.



Figura 02

Passo 3: Separe 2 palitos de picolé e com ajuda da régua marque um ponto no centro de cada um dos palitos e usando a furadeira com broca fina, faça um furo do mesmo diâmetro do parafuso que você separou. Observe a figura 3 abaixo.



Figura 03

Passo 4: Pegue o servo motor de 9g e coloque sua base sobre os dois palitos de picolé que você colou no passo 2. Marque com um lápis a posição do servo nos 2 palitos conforme a figura 4 abaixo e use o estilete para cortar a madeira no tamanho exato da base do servo motor. CUIDADO AO MANUSEAR O ESTILETE, PODE HAVER RISCO DE CORTES!



Figura 04

AULAS DE ROBÓTICA EDUCATIVA

ROBÔ QUADRUPEDE COM SERVO

Passo 5: Coloque os 2 palitos que você furou no passo 3 sobre os outros 2 palitos que você colou e cortou no formato do servo motor. Use o furo existente como guia para furar também esses dois palitos conforme a figura 5 abaixo. Após furar, use os 2 parafusos para prender tudo.

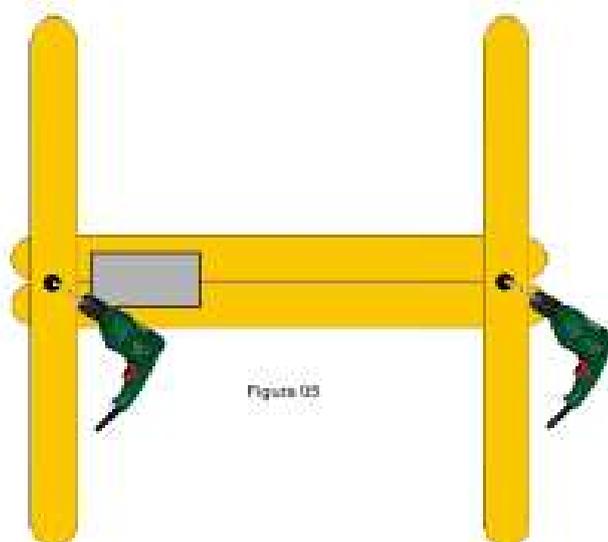


Figura 05

Passo 6: Separe 1 palito de picolé e com ajuda da régua marque exatamente o meio do palito. Cada metade dos palitos deverá encaixar na base do robô que já está pronta e aparafusada e para isso use o estilete para entalhar as quinas dos palitos conforme a figura 6 abaixo.



Figura 06

Repita este passo 6 mais 3 vezes para entalhar as 4 pernas do robô.

AULAS DE ROBÓTICA EDUCATIVA

ROBÔ QUADRUPEDE COM SERVO

Passo 7: Pegue a base do robô já colada e parafusada e usando o estilete corte pequenas ranhuras nas pontas das pernas do robô para encaixar as peças de madeira cortadas no passo 8.

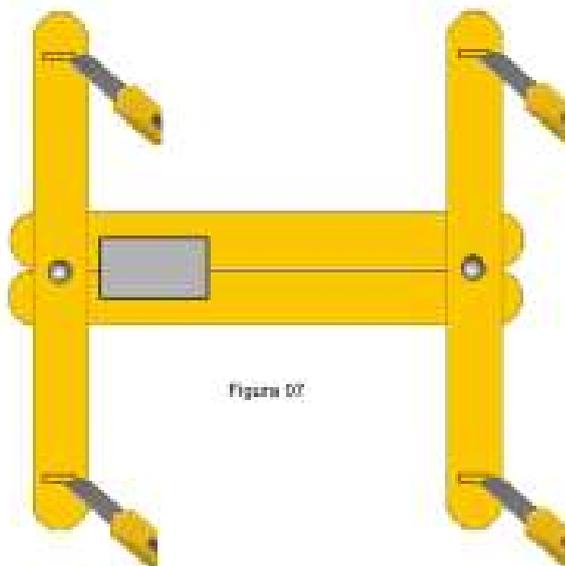


Figura 07

Passo 8: Separe 1 palito de picolé, com ajuda da régua marque 3 pedaços de 3cm e corte com o estilete. Separe o último palito de picolé, marque com a régua 3 pedaços de 4cm e corte com o estilete. CUIDADO AO MANUSEAR O ESTILETE, PODE HAVER RISCO DE CORTES!

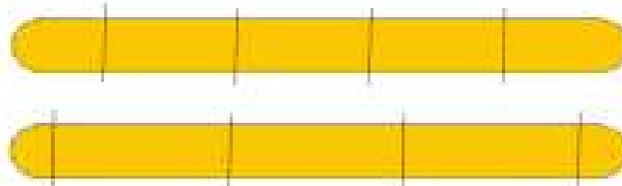


Figura 08

AULAS DE ROBÓTICA EDUCATIVA

ROBÔ QUADRUPEDE COM SERVO

Passo 9: Pegue os 3 pedaços de madeira de 3cm que você cortou no passo 8 e cole conforme a figura 9a. Pegue 1 pedaço de madeira de 4cm e corte ao meio com o estilete no sentido do veio da madeira. Agora cole esses pedaços mais fino nos outros 2 pedaços de madeira de 4cm conforme a figura 9b.

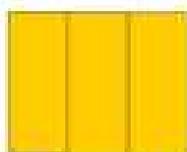


Figura 09a

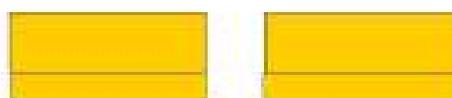


Figura 09b

Passo 10: Pegue os 3 pedaços de madeira de 3cm que você colou e cole com os outros 2 pedaços de madeira com 4cm na forma de U. Observe a figura 10 abaixo.



Figura 10

Passo 11: Pegue o aparato em forma de U que você colou no passo 10, posicione-o sobre o seu robô no lado oposto ao servo motor e cole as suas laterais no corpo do robô conforme a figura 11 abaixo.

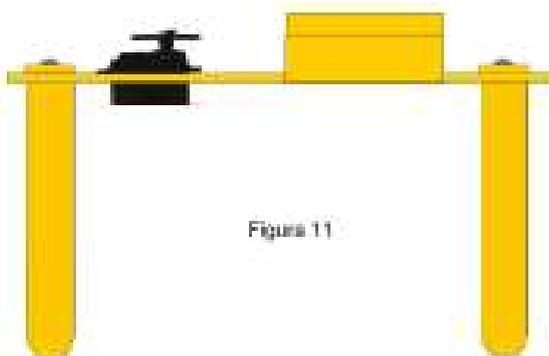
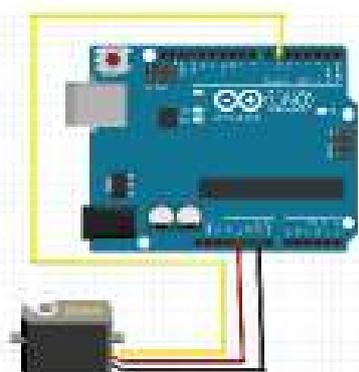


Figura 11

AULAS DE ROBÓTICA EDUCATIVA

ROBÔ QUADRUPEDE COM SERVO

Passo 12: Coloque a bateria de 9V dentro da caixa em U colada no passo 11. Coloque sobre a caixa em U a placa Arduino e prenda com a liga elástica e verifique que ela está firmemente presa. Ligue o dip na bateria e ligue os fios do servo motor no arduino conforme esquema elétrico abaixo.



Passo 13: Agora que a montagem do robô está concluída, vamos programar e testar. Abra a interface de programação do Arduino e digite o programa abaixo.

```
#include <Servo.h>
```

```
Servo servo1; // cria uma instância do objeto servo
int pos = 0; // variavel que armazenara a posicao do servo
void setup() {
  servo1.attach (6); // servo ligado ao pino 6 do Arduino
}
```

```
void loop() {
  // vai de 0 a 180 graus em passos de 1 grau
  for (pos = 0; pos < 180; pos += 1) {
    // avisa ao servo para se posicionar no valor de 'pos'
    servo1.write (pos);
    // espera 15ms para o servo atingir a posicao
    delay (15);
  }
}
```

```
delay (500);
// vai de 180 a 0 graus em passos de 1 grau
for (pos = 180; pos >= 1; pos -= 1) {
  // avisa ao servo para se posicionar no valor de 'pos'
  servo1.write (pos);
  // espera 15ms para o servo atingir a posicao
  delay (15);
}
delay (500);
}
```

Exemplos dos questionários aplicados aos alunos do ensino fundamental e médio que participaram das aulas de robótica educativa.

LOGOMARCA DA ESCOLA	NOME DA ESCOLA www.enderecoeletronicodaescola.com.br
------------------------	--

Avaliação de Aulas – Robótica Educativa

1. Assinale quais séries que você teve aulas de robótica.

4º ANO 5º ANO 6º ANO 7º ANO

2. Você participou das aulas de robótica educativa com kits educacionais da LEGO?

Sim Não

Obs.: Nos próximos itens selecione uma das opções de acordo com o grau de intensidade de 1 a 4, onde 1 é o grau mais fraco e 4 o grau mais forte. Marque um X no grau desejado.

3. Facilidade em manusear os componentes e materiais.

(1) (2) (3) (4)

4. Clareza no entendimento de como funciona a interface NXT e demais componentes.

(1) (2) (3) (4)

5. Conhecimento do ambiente de programação do LEGO NXT.

(1) (2) (3) (4)

6. Qual o seu interesse em estudar mais sobre ciência e tecnologia?

(1) (2) (3) (4)

7. Você consegue trabalhar melhor em equipe depois das aulas de robótica?

(1) (2) (3) (4)

Nome do(a) Aluno(a): _____

Série: _____ Data: _____ / _____ / _____

LOGOMARCA DA ESCOLA	NOME DA ESCOLA www.enderecoeletronicodaescola.com.br
------------------------	--

Avaliação de Aulas – Robótica Educativa

1. Você participou da oficina de robótica educativa utilizando material de sucata e Arduino?

Sim Não

2. Você participou das aulas de robótica educativa com kits educacionais da LEGO?

Sim Não

Obs.: Nos próximos itens selecione uma das opções de acordo com o grau de intensidade de 1 a 4, onde 1 é o grau mais fraco e 4 o grau mais forte. Marque um X no grau desejado.

3. Facilidade em manusear os componentes e materiais.

(1) (2) (3) (4)

4. Clareza no entendimento de como funciona a interface Arduino e demais componentes.

(1) (2) (3) (4)

5. Conhecimento da linguagem de programação do Arduino.

(1) (2) (3) (4)

6. O seu interesse em utilizar material de sucata, componentes eletrônicos e Arduino para criar robôs.

(1) (2) (3) (4)

Nome do(a) Aluno(a): _____

Série: _____ Data: _____ / _____ / _____