

unesp  **UNIVERSIDADE ESTADUAL PAULISTA**
“JÚLIO DE MESQUITA FILHO”
CAMPUS DE GUARATINGUETÁ

MATHEUS DE FELIPPE FERREIRA

**ELABORAÇÃO DE EXPERIMENTOS DE ROBÓTICA VOLTADOS PARA
ALUNOS DO ENSINO MÉDIO**

Guaratinguetá - SP
2016

MATHEUS DE FELIPPE FERREIRA

ELABORAÇÃO DE EXPERIMENTOS DE ROBÓTICA VOLTADOS PARA ALUNOS DE
ENSINO MÉDIO

Trabalho de Graduação apresentado ao Conselho de Curso de Graduação em Engenharia Elétrica da Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista, como parte dos requisitos para obtenção do diploma de Graduação em Engenharia Elétrica.

Orientador: Prof. Dr. Leonardo Mesquita

Guaratinguetá - SP
2016

F383e	<p>Ferreira, Matheus de Felipe Elaboração de experimentos de robótica voltados para alunos do ensino médio / Matheus de Felipe Ferreira – Guaratinguetá, 2016. 89 f : il. Bibliografia: f. 80-84</p> <p>Trabalho de Graduação em Engenharia Elétrica – Universidade Estadual Paulista, Faculdade de Engenharia de Guaratinguetá, 2016. Orientador: Prof. Dr. Leonardo Mesquita</p> <p>1. Robótica. 2. Eletrônica. 3. Educação. 4. Ensino médio. I. Título</p> <p style="text-align: right;">CDU 007.52</p>
-------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

MATHEUS DE FELIPPE FERREIRA

ESTE TRABALHO DE GRADUAÇÃO FOI JULGADO ADEQUADO COMO
PARTE DO REQUISITO PARA A OBTENÇÃO DO DIPLOMA DE
GRADUADO EM ENGENHARIA ELÉTRICA

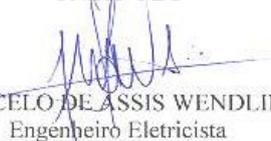
APROVADO EM SUA FORMA FINAL PELO CONSELHO DE CURSO DE
GRADUAÇÃO EM ENGENHARIA ELÉTRICA


Prof. Dr. Leonardo Mesquita
Coordenador

BANCA EXAMINADORA:


Prof. Dr. LEONARDO MESQUITA
Orientador/UNESP-FEG


Prof. Dr. GALENO JOSÉ DE SENNA
UNESP-FEG


Eng. JOSÉ MARCELO DE ASSIS WENDLING JÚNIOR
Engenheiro Eletricista

Dezembro de 2016

DADOS CURRICULARES

MATHEUS DE FELIPPE FERREIRA

NASCIMENTO	05.04.1993 – GUARATINGUETÁ / SP
FILIAÇÃO	Ricardo Jorge de Abreu Ferreira Maria Célia Hasmann Coelho Ferreira
2008/2010	Curso Técnico em Eletrônica na <i>Universidade Estadual Paulista – Colégio Técnico Industrial de Guaratinguetá</i>
2011/2016	Curso de Graduação em Engenharia Elétrica na <i>Universidade Estadual Paulista – Campus de Guaratinguetá</i>

dedico este trabalho, de modo especial, aos meus pais Ricardo e Maria Célia, minha irmã Valeska e minha avó Geny, que sempre me deram força nesta jornada, aos meus amigos e à equipe de robótica “FEG Robótica” pelas ideias e amparo na concretização deste trabalho.

AGRADECIMENTOS

Em primeiro lugar agradeço a Deus, fonte da vida e da graça. Agradeço pela minha vida, minha inteligência, minha família e as oportunidades que Ele me concedeu,

ao meu orientador, *Prof. Dr. Leonardo Mesquita* que sempre colocou desafios para me incentivar e me tornar um profissional mais flexível,

aos meus pais *Ricardo e Maria Célia*, que apesar das dificuldades enfrentadas, sempre incentivaram meus estudos,

a minha irmã *Valeska*, que sempre me serviu como exemplo de perseverança na vida,

a minha avó *Geny*, que sempre me ajudou e me fortaleceu durante esta árdua jornada,

aos meus amigos de intercâmbio de *Newcastle University*, que mesmo conhecendo-os durante um breve ano, proporcionaram experiências únicas e que revolucionaram minha vida,

aos meus amigos de faculdade, em especial ao *Everton Lincon, Thiago Andrade Prado, Fernando Thiago, Murilo Soldi, Marina Aparecida e Helen Vendrameto*, que foram ótimas companhias durante o curso e cujas amizades são verdadeiras relíquias,

a equipe de robótica *FEG Robótica* e seus integrantes atuais e passados, que construíram grandes lembranças e que me fizeram aprender lições de vida importantíssimas em competições e em projetos internos do time,

ao curso preparatório pré-vestibular *FEGVEST*, e em especial ao *Bruno Lemes* que aceitou minha integração no mesmo, que me fez descobrir e desenvolver minha vocação como professor, bem como em perceber a importância do professor na vida de alunos de diferentes realidades,

a empresa *GENERAL MOTORS* de São José dos Campos, em especial aos estagiários da minha turma e aos meus colegas da engenharia de manufatura, que me deu a oportunidade de colocar meus conhecimentos em prática, aprender como funciona o planejamento de uma grande empresa e como são tomadas decisões em um mercado extremamente competitivo, além de todo o suporte e auxílio que me foi fornecido durante todo o estágio,

e por fim, agradeço todos os professores do *Colégio Técnico Industrial de Guaratinguetá* e da *Universidade Estadual Paulista “Júlio de Mesquita Filho”*, por me instruírem da melhor forma possível para me tornar um profissional digno e íntegro, em especial ao *Prof. Dr. Rubens Alves Dias*, professor cuja experiência sempre serviu de inspiração para mim, e ao *Prof. Dr. Samuel Euzédice de Lucena*, cujo carinho e suporte sempre me ajudaram a escolher o melhor caminho do meu futuro profissional.

“A felicidade não está em fazer o que a gente quer, mas sim em querer o que a gente faz.”

Jean-Paul Sartre

FERREIRA, M. de F. **Elaboração de Experimentos de Robótica Voltados para Alunos de Ensino Médio**. 2016. 89p. Trabalho de graduação (Graduação em Engenharia Elétrica) – Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista, Guaratinguetá, 2016.

RESUMO

O extenso contato com aparelhos eletrônicos inteligentes e a implementação massiva da chamada “Internet das Coisas” fez com que crianças, desde idade bem prematuras, interagissem com tais aparelhos cada vez mais frequentemente, influenciando desde seus hábitos e gostos até a predisposição a absorver novos conceitos e ideias. Um movimento crescente no cenário global é a integração do aprendizado tecnológico juntamente com as matérias obrigatórias do ensino, de forma a fazer com que os estudantes ponham em prática os conhecimentos obtidos nas aulas, fomentando o raciocínio lógico e estimulando o interesse nas matérias de exatas, onde muitos estudantes encontram-se desencorajados diante de problemas mais complexos. O presente trabalho visa propor um material de um curso de robótica para alunos de ensino médio, podendo ser adaptado para demais faixas etárias, de forma que o mesmo aborde os temas e conceitos de eletrônica e programação mais críticos, em uma plataforma de fácil compreensão, usando materiais eletrônicos básicos disponíveis no mercado nacional e com carga horária condizente para não comprometer o ensino das demais disciplinas, fazendo com que os alunos compreendam o funcionamento de dispositivos eletrônicos que o circundam e, ao longo prazo, contribuindo para a formação de profissionais mais completos e flexíveis para o mercado de trabalho.

PALAVRAS-CHAVE: Robótica. Educação. Programação. Eletrônica. Desenvolvimento de Projetos.

FERREIRA, M. de F. **Development of Robotics Experiments Oriented to High School Students**. 2016. 89p. Graduate work (Bachelor in Electrical Engineering) – Faculdade de Engenharia do Campus de Guaratinguetá, Universidade Estadual Paulista, Guaratinguetá, 2016.

ABSTRACT

The extensive contact with intelligent electronic appliances and the massive implementation of the “Internet of Things” made children, from very premature ages, interact with those devices more and more oftenly, affecting from habits and tastes to new ideas and concepts absorption readiness. An increasing movement in the global scope is the integration of technological learning with core academic subjects, so the students can put their knowledge to the test, fostering their logical reasoning and stimulating the interest in Maths, where lots of students find themselves discouraged in face of more complex problems. This work aims to propose a robotics course material for high school students, so that it contains the most critical electronics and programming concepts, in an easy to understand platform, using basic electronic material available in the national market, and with befitting workload in order to not compromise other educational subjects, making students understand how electronic devices around them work and, in the long term, contributing to the development of more complete and flexible professionals in the job market.

KEYWORDS: Robotics. Education. Programming. Electronics. Project Development.

LISTA DE ILUSTRAÇÕES

Figura 1 – Robô móvel LEGO montado usando kit MINDSTORMS RCX.....	21
Figura 2 – Interface principal da IDE do LEGO Mindstorms NXT.....	22
Figura 3 – Placa de desenvolvimento Arduino Uno R3 com microcontrolador Atmel Atmega 328p..	23
Figura 4 – Interface gráfica do Arduino Software (IDE)	24
Figura 5 – Interface gráfica do Scratch for Arduino (S4A)	25
Quadro 1 – Comparação entre Plataformas de Ensino de Robótica.....	26
Figura 6 – Figura didática explicando o conceito de corrente elétrica.....	29
Figura 7 – Exercício sobre lógica: O problema do fazendeiro, a raposa, o pato e o milho.....	30
Figura 8 – Exemplo de fluxograma para uma situação cuja solução consiste no loop de ações previamente executadas.....	31
Quadro 2 – Operadores matemáticos e lógicos básicos no Arduino	32
Figura 9 – Exemplo de uso de estruturas condicionais cascadeadas em forma de fluxograma.	33
Figura 10 – Simbologia de portas lógicas fundamentais e suas respectivas tabelas de operação.	34
Figura 11 – Diferença entre (a) um sinal analógico e (b) um sinal digital.....	36
Figura 12 – Ilustração do processo de modulação PWM a partir de uma onda triangular.....	36
Figura 13 – Exemplo de circuito que utiliza um microfone para acender LEDs de um display tipo “bargraph” na representação da intensidade sonora captada.....	38
Figura 14 – Representação das estruturas básicas de um motor CC (estator e rotor) usado em brinquedos e seu arranjo.....	39
Figura 15 – Representação do circuito elétrico didático de uma “ponte H” para o controle do sentido de rotação de um motor CC.....	40
Figura 16 – Exemplos de servomotores.	41
Figura 17 – Pinagem do circuito integrado L293D (à esquerda) e seu esquemático equivalente para suas saídas (à direita).....	42
Figura 18 – Pinagem de um display de sete segmentos cátodo comum e os respectivos segmentos relativos à cada pino.....	43
Figura 19 – Display de cristal líquido 16 colunas por 2 linhas convencional.....	44
Figura 20 – Protoboard (<i>Breadboard</i>) convencional (à esquerda) e suas conexões internas (à direita)	47

Figura 21 – Código de cores para resistores de baixa potência.....	48
Figura 22 – Associação de resistores em série (a), em paralelo (b) e associação mista (c)	49
Figura 23 – Circuito misto didático para ensinado uso do multímetro	49
Figura 24 – Exemplo de fluxograma para uma situação cuja solução consiste no loop de ações previamente executadas.....	51
Figura 25 – Exemplo de programa na IDE Arduino contendo estrutura condicional para controlar um LED com apenas um botão.	53
Figura 26 – Exemplo de programa contendo estrutura de repetição de comandos para o acendimento sequencial de LEDs conectados em vários pinos da plataforma Arduino.....	54
Figura 27 – Circuito de demonstração de regulação de brilho de um LED	55
Figura 28 – Exemplo de uso do monitor serial para teste de programas no software Arduino.....	57
Figura 29 – Exemplo de circuito com LDR para funcionar como um “sensor crepuscular”, acendendo o LED na ausência de luz e apagando-o haja luz suficiente	58
Figura 31 – Circuito elétrico para o acionamento “Liga/Desliga” de um motor CC utilizando o circuito integrado L293D	61
Figura 32 – Circuito elétrico para o controle de sentido de giro de um motor CC utilizando o circuito integrado L293D	63
Figura 33 – Programa de exemplo que realiza a varredura do eixo de um servomotor de 0 até 180 graus, e então de 180 até 0 graus ciclicamente.....	64
Figura 34 – Exemplo de circuito para conexão do display LCD com a placa Arduino, incluindo potenciômetro para controle de contraste do texto exibido.....	65
Figura 35 – Programa de inicialização de um display LCD 16x2.....	66
Figura 36 – Plataforma educacional de robótica baseado em um carro contemporâneo	69
Figura 37 – Placas de circuito integrado montadas na plataforma de robótica	72
Figura 38 – Circuito da versão básica “stand-alone” do Arduino	72
Figura 39 – Mapeamento dos pino do microcontrolador Atmel ATMEGA328p vistos pela IDE Arduino	73
Figura 40 – Comparação entre dificuldade dos softwares da oficina de robótica de acordo com os alunos	76
Figura 41 – Resultados referentes à competência de trabalho em equipe e aprendizagem em equipe .	77

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
BCD	Binary Coded Decimal
CC	Corrente Contínua
CD	Compact Disc
CI	Circuito Integrado
CMOS	Complementary Metal-Oxide Semiconductor
CNC	Computer Numerically Controlled
CNI	Confederação Nacional da Indústria
CONFEA	Conselho Federal de Engenharia e Agronomia
CREA	Conselho Regional de Engenharia e Agronomia
DVD	Digital Versatile Disc
EPI	Equipamento de Proteção Individual
EXPRECI	Exposição Regional de Engenharia e Ciências
FEBRACE	Feira Brasileira de Ciências e Engenharia
FLL	FIRST LEGO League
GND	Ground
GPIO	General Purpose Input/Output
I2C	Inter-Integrated Circuit
ICSP	In-Circuit Serial Programming
IBGE	Instituto Brasileiro de Geografia e Estatística
IDE	Integrated Development Environment
IEC	International Electrotechnical Commission
IPEA	Instituto de Pesquisa Econômica Aplicada
ISO	International Organization for Standardization
LCD	Liquid Crystal Display
LDR	Light Dependent Resistor
LED	Light Emitting Diode
MIT	Massachusetts Institute of Technology
NBR	Norma Brasileira Regulamentadora
OBR	Olimpíada Brasileira de Robótica
OECD	Organisation for Economic Co-operation and Development
PCB	Printed Circuit Board
PIC	Peripheral Interface Controller
PISA	Programme for International Student Assessment
PLC	Programmable Logic Controller
PWM	Pulse Width Modulation
RFID	Radio-Frequency Identification
SD	Secure Digital
TTL	Transistor-Transistor Logic
UNESP	Universidade Estadual Paulista

SUMÁRIO

1	INTRODUÇÃO	16
1.1	ORIGEM DA ROBÓTICA E SUA IMPORTÂNCIA NAS INDÚSTRIAS	16
1.2	OBSTÁCULOS DO ATUAL SISTEMA EDUCACIONAL BRASILEIRO	17
1.3	MOTIVAÇÃO PARA O ENSINO DE ROBÓTICA NO ENSINO MÉDIO	18
1.4	ORGANIZAÇÃO DO TRABALHO	19
2	PLATAFORMAS DE ROBÓTICA EDUCACIONAL.....	20
2.1	LEGO MINDSTORMS	20
2.2	ARDUINO	23
2.2.1	Arduino Software	24
2.2.2	Scratch for Arduino (S4A).....	25
2.3	COMPARAÇÃO E FOCO DE PLATAFORMAS DE ENSINO DE ROBÓTICA	26
3	CONCEITUAÇÃO TEÓRICA	28
3.1	INTRODUÇÃO À ELETRÔNICA	28
3.2	INTRODUÇÃO À LÓGICA E FLUXOGRAMAS.....	30
3.3	APROFUNDAMENTO NA LINGUAGEM DE PROGRAMAÇÃO DA PLATAFORMA ARDUINO ..	32
3.4	OPERAÇÕES LÓGICAS E MATEMÁTICAS.....	34
3.5	MANIPULAÇÃO DE DADOS ANALÓGICOS.....	35
3.5.1	Modulação por Largura de Pulso (PWM)	36
3.6	SENSORES ANALÓGICOS E SUAS APLICAÇÕES.....	37
3.7	ATUADORES ROTATIVOS E SEUS ACIONAMENTOS	38
3.7.1	Motores CC	38
3.7.2	Servomotores.....	40
3.7.3	O Circuito Integrado L293D	41
3.8	DISPLAYS E MOSTRADORES DIGITAIS	42
3.8.1	Display de Sete Segmentos	43
3.8.2	Display de Cristal Líquido (LCD).....	44
4	ELEMENTOS PRÁTICOS DO CURSO	46
4.1	AULA INTRODUTÓRIA.....	46
4.1.1	Montagem de Circuitos Eletrônicos.....	46
4.1.2	Leitura de Resistores e Demais Componentes Eletrônicos.....	47
4.1.3	Associação de Resistores	48
4.1.4	Equipamentos de Medição em Circuitos Eletrônicos.....	49
4.2	INTRODUÇÃO À PROGRAMAÇÃO	50
4.2.1	Utilizando o Software Arduino IDE	51
4.2.2	Comandos Básicos na Plataforma Arduino	52
4.3	IMPLEMENTAÇÃO DAS ESTRUTURAS CONDICIONAIS E DE REPETIÇÃO	52
4.4	LEITURA DE DADOS ANALÓGICOS E MODULAÇÃO PWM NO ARDUINO	55
4.4.1	Uso do Monitor Serial da IDE Arduino	56
4.5	UTILIZAÇÃO DE SENSORES ANALÓGICOS EM APLICAÇÕES DINÂMICAS	57
4.6	UTILIZAÇÃO E CONTROLE DE MOTORES CC E SERVOMOTORES	60
4.6.1	Controle Tipo “Liga/Desliga” (ON/OFF) de um Motor CC.....	61
4.6.2	Controle Bidirecional de um Motor CC.....	62

4.6.3	Controle de Servomotores	63
4.7	UTILIZAÇÃO DE DISPLAYS DE CRISTAL LÍQUIDO	65
5	DESENVOLVIMENTO DA PLATAFORMA	69
5.1	DESENVOLVIMENTO DO CIRCUITO.....	70
5.2	SOLDAGEM DOS COMPONENTES	71
5.3	PROGRAMAÇÃO DA PLATAFORMA	73
5.4	TESTES E CONSIDERAÇÕES	74
6	APLICAÇÃO.....	76
7	CONCLUSÃO.....	79
	REFERÊNCIAS	80
	APÊNDICE A – Esquemático Elétrico da Plataforma Educacional de Robótica.....	85
	APÊNDICE B –Lista de Componentes para Construção da Plataforma Educacional de Robótica.....	86
	APÊNDICE C – Código-Fonte Final da Plataforma Educacional de Robótica	87

1 INTRODUÇÃO

1.1 Origem da Robótica e sua Importância nas Indústrias

A palavra “robô” é derivado da palavra tcheca *robot*, que significa servo ou escravo, e foi cunhado nas primeiras décadas do século XX para denominar sistemas de manipulação da indústria de manufatura e em livros clássicos de ficção científica, notavelmente, pelo dramaturgo Karel Capek, que usou tal termo pela primeira vez em 1921 (MURRAY; LI; SASTRY, 1994). O termo “robótica”, por sua vez, foi usado pela primeira vez pelo escritor de ficção Isaac Asimov na década de 1940, como um meio de simbolizar o avanço rápido destes sistemas robotizados, que se tornavam cada vez mais flexíveis, executando mais operações e, conseqüentemente, sendo os substitutos cada vez mais comuns aos operários nas indústrias.

Segundo o dicionário Aurélio, robótica é definido como “Conjunto dos estudos e das técnicas tendentes a conceber sistemas capazes de substituírem o homem em suas funções motoras, sensoriais e intelectuais.” (FERREIRA, 2010). Isto significa que, mesmo que não tendo necessariamente forma semelhante com a de um ser humano, o robô é capaz de responder a estímulos externos, executar tarefas e ter desenvoltura lógica como um humano.

Ainda que na primeira metade do século XX robôs executavam tarefas muito simples, não possuíam muita eficiência energética e necessitavam de manutenção constante, a tecnologia implementada em robôs deu um salto após a Segunda Guerra Mundial. Notavelmente, o desenvolvimento de robôs operados à distância pela General Electric, o desenvolvimento de máquinas CNC pela Unimation que possibilitaram a manufatura de peças extremamente precisas para aviões e carros, e o primeiro robô industrial, Unimate, desenvolvido por Joseph Engelberger e George Devol para a General Motors (BOCK, 2009) para a solda de partes da carroceria de automóveis, estão entre as principais criações que revolucionaram os sistemas robotizados.

A crescente evolução dos sistemas robotizados foi impulsionada de acordo com as necessidades das companhias em atender demandas cada vez maiores, reduzir custos de produção, atender o nível de qualidade exigido pelos consumidores e a complexidade na manipulação de peças e elementos químicos. Tais exigências não só eram benéficas para as grandes empresas, que desenvolviam produtos que proporcionavam maior conforto ao consumidor final, como também para os operários destas companhias, visto que robôs poderiam operar em temperaturas e condições prejudiciais aos seres humanos, bem como diminuir o número de acidentes do trabalho na manipulação de peças e produtos químicos.

Tal evolução foi possível devido à evolução de microprocessadores e microcontroladores, acentuada a partir de 1970 pelas empresas Intel e Texas Instruments, no desenvolvimento e posterior barateamento de microprocessadores, com capacidades de memória e velocidade de execução de instruções cada vez maiores. Melhorias nos métodos de produção de microprocessadores também contribuíram significativamente para o desenvolvimento de sistemas menores e mais confiáveis, reduzindo o número de falhas e perdas de dados por decorrência da alta frequência exigida pelos microprocessadores e os efeitos elétricos deste chaveamento rápido (WANG; STROUD; TOUBA, 2007).

Nos dias de hoje, células automatizadas com robôs e a integração destas em diversas plantas industriais é algo já comum para que as empresas permaneçam competitivas no mercado. A massificação de tais sistemas acabou por necessitar de normas e padrões de qualidade, tanto globais quanto regionais, de forma a garantir a certificação de robôs em diversas áreas, dentre elas, o nível de segurança necessário de acordo com o risco da operação a ser realizada, o tempo de resposta de comandos manuais e de emergência, a padronização da fabricação e dimensionamento de componentes de automação internos ao robô e ao respectivo controlador. Algumas das normas em destaque são a ISO 10218:2011 (Robôs e Dispositivos Robóticos - Requisitos de Segurança para Robôs Industriais), ISO 15066:2016 (Robôs e Dispositivos Robóticos – Robôs Colaborativos), ABNT NBR 14153:2013 (Segurança de Máquinas – Partes de Sistemas de Comando Relacionados à Segurança – Princípios Gerais para Projetos) e a IEC 60204:2005 (Segurança de Máquinas – Equipamentos Elétricos de Máquinas).

1.2 Obstáculos do Atual Sistema Educacional Brasileiro

Diante do cenário atual da complexidade de sistemas robotizados, a formação de um profissional flexível o bastante para compreender sistemas de diferentes fabricantes e capaz de ter raciocínio lógico para aglutinar inúmeros fatores na solução de problemas industriais é difícil de se obter com o atual sistema de ensino brasileiro.

Segundo o censo mais recente do IBGE, apenas 10,8% das pessoas acima de 25 anos possuem ensino superior completo (2010). Entre 2001 e 2011, apenas 44,41% dos estudantes matriculados nos cursos de engenharia obtiveram a titulação nesta área (CONFEDERAÇÃO NACIONAL DA INDÚSTRIA, 2014). Em números inteiros, foram cerca de 288 de engenheiros formados por milhão de habitantes em 2010, número muito abaixo se comparado

com outros países da OECD (“*Organisation for Economic Co-operation and Development*”), como Estados Unidos (496), Itália (544), México (701) e Reino Unido (909).

Ainda que, atualmente, somam-se mais de 1,3 milhão de engenheiros ativos registrados no CREA (CONFEA, 2016), a qualidade da formação destes também é relevante para se perceber o cenário real da educação no Brasil. Segundo o IPEA, diferentes setores da economia como construção, extração mineral, infraestrutura e indústria de transformação possuem requerimentos técnicos diferentes entre si (MACIENTE; ARAÚJO, 2011). Logo, a capacidade de assimilação de novos conhecimentos, interpretação de dados, análise lógica e elaboração de um plano de ação são essenciais para a formação de profissionais que saibam lidar com as especificidades de cada setor.

Tais habilidades não se desenvolvem apenas durante a idade adulta ou no ensino superior. Ademais, o desenvolvimento destas habilidades em idades mais prematuras afetam diretamente no interesse posterior na área de engenharia por parte do indivíduo e no desenvolvimento de suas habilidades em Matemática. Esta performance é, atualmente, deficiente, segundo avaliação do PISA (*Organisation for Economic Co-operation and Development*, [2012?]), que deu a nota de 391 para a média de conhecimentos em Matemática, Leitura e Ciência em estudantes de 15 anos no Brasil. Tal nota deixa o Brasil abaixo da média OECD, comparável com Albânia (394), Jordão (386) e Argentina (388).

1.3 Motivação para o Ensino de Robótica no Ensino Médio

Para estimular o aprendizado das ciências exatas em alunos do ensino fundamental e médio, desenvolveram-se diversos métodos práticos e interativos que visam a aplicação dos conhecimentos teóricos de Ciências, Matemática e Física em ações e respostas concretas. O uso de robôs para tais fins educativos é chamado de robótica educacional, que tem por objetivo estimular a criatividade dos alunos de forma interativa para se obter maior envolvimento com o ensino tradicional (GOMES, 2007).

A integração da robótica educacional no sistema público de ensino já é realidade em países desenvolvidos mundo afora. Holanda e Alemanha já possuem a robótica educacional implementada em 100% das escolas públicas, sendo que Inglaterra, Itália, Canadá e Estados Unidos estão avançando para atingir tal marca (QUINTANILHA, 2008). Notavelmente, todos os países supracitados desempenharam melhores pontuações no PISA 2012 se comparados com o Brasil (*Organisation for Economic Co-operation and Development*, 2014).

No Brasil, o ensino de robótica nas escolas públicas é raro. Nas escolas particulares, a robótica já é mais comum no ensino fundamental, médio e técnico, mas como apenas as escolas de maior porte contém tal estrutura, isto acaba por excluir boa parte de estudantes em regiões carentes de recursos.

Ainda assim, o Brasil caminha para mudar tal cenário incentivando cada vez mais os estudantes a desenvolver seus próprios projetos para apresentação em eventos patrocinados por órgãos governamentais na forma de competições e feiras. É o caso da Olimpíada Brasileira de Robótica, competição que incentiva alunos do ensino fundamental e médio em modalidades teóricas e práticas, o Torneio de Robótica *FIRST LEGO League*, competição onde crianças de 9 a 16 anos devem buscar soluções para diversos cenários de forma criativa, e a FEBRACE, feira anual que estimula e premia projetos inovadores, visando o fomento de novas tecnologias em território brasileiro.

1.4 Organização do Trabalho

O capítulo 1 apresenta uma breve introdução do cenário educacional do Brasil frente a países desenvolvidos, evidenciando a necessidade do aprendizado de robótica para preparar o cidadão para o mercado de trabalho e melhorar suas habilidades matemáticas.

O capítulo 2 visa comparar plataformas de robótica e verificar qual é a mais indicada para ensino médio de forma a incentivar o estudante a persistir na carreira em engenharia.

O capítulo 3 discorre sobre o conteúdo teórico do curso necessário para que os estudantes compreendam a estrutura, lógica e funcionamento dos dispositivos usados na plataforma em questão, em ordem compreensível e com propostas de reforço conceitual.

O capítulo 4 trata do conteúdo prático, contendo propostas de atividades, projetos e exercícios a serem executados durante as aulas práticas do curso.

O capítulo 5 mostra a aplicação dos conceitos teóricos e práticos na montagem de uma plataforma de robótica, onde experimentos vinculados às áreas de Ciências, Física ou Matemática podem ser realizados com base em diferentes tópicos de cada disciplina.

O capítulo 6 apresenta resultados da aplicação de um curso de robótica educacional de curta duração a alunos do ensino médio das escolas da região do Vale do Paraíba realizado nas dependências do campus de Guaratinguetá da UNESP, com o intuito de testar o uso do material desenvolvido.

O capítulo 7 conclui o trabalho, colocando propostas de melhorias conforme interesses, tempo e recursos disponíveis do educador e da instituição que irá aplicar o curso.

2 PLATAFORMAS DE ROBÓTICA EDUCACIONAL

O primeiro passo para se desenvolver um curso de robótica educacional é realizar a escolha de qual plataforma deve-se usar como base, visto que diferentes plataformas desenvolvem diferentes habilidades e áreas do conhecimento nos estudantes. Naturalmente, também deve ser considerada a faixa etária dos estudantes que estarão matriculados no curso, conhecimento prévio em Matemática e Lógica, e a capacidade da escola em fornecer os materiais aos alunos em quantidade suficiente.

Plataformas robóticas educacionais basicamente se dividem entre kits patenteados por empresas, tais como os da LEGO (2006), VEX (2016) e Fischertechnik (2014), e plataformas de código de acesso livre (*open source*) compatíveis com componentes eletrônicos discretos disponíveis no mercado, tais como o Arduino (2016a) e Raspberry Pi (2016).

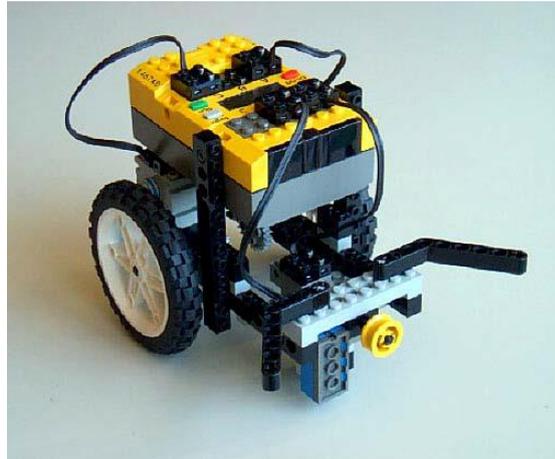
Por haver diversos tipos de plataformas voltadas à robótica educacional, serão considerados neste trabalho apenas as mais populares atualmente, bem como aquelas com maior suporte, seja do fabricante ou por meio de usuários em comunidades online, disponível em Português.

2.1 LEGO Mindstorms

Talvez a plataforma de robótica educacional mais popular de todas, o LEGO Mindstorms é uma evolução de um projeto entre a LEGO, empresa de brinquedos dinamarquesa, e o Instituto Tecnológico de Massachusetts (MIT), que objetivava a criação de um “tijolo inteligente” (*intelligent brick*) que pudesse ser programado para interagir com o ambiente externo através de sensores e atuadores eletrônicos ligados ao mesmo (RESNICK et al., 1996).

Dentro da linha de kits educativos LEGO Mindstorms, existem diferentes gerações, cada uma implementando melhorias nas peças inclusas nos kits das gerações anteriores e melhorias na unidade de processamento central da plataforma, possibilitando a aplicação em experimentos mais complexos. Dentre alguns projetos possíveis usando o Mindstorms estão robôs seguidores de linha, guitarras elétricas, empilhadeiras e robôs móveis em geral, como o mostrado pela Figura 1.

Figura 1 –Robô móvel LEGO montado usando kit MINDSTORMS RCX



Fonte: (GONÇALVES, 2007).

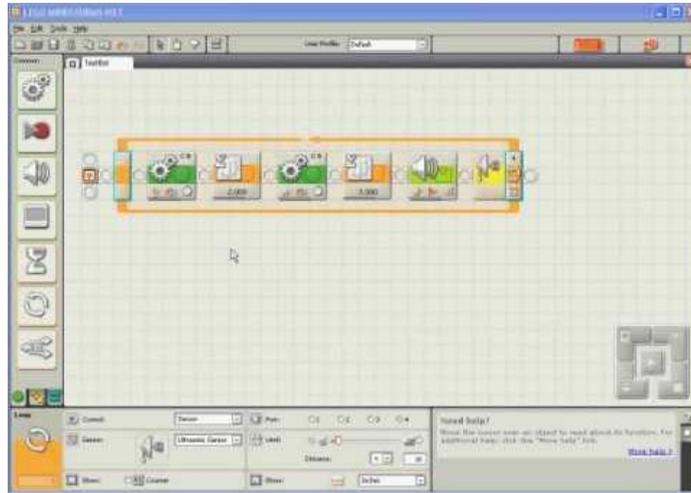
Recomendado para crianças acima de 11 anos de idade (KNUDSEN, 1999), a plataforma LEGO Mindstorms desenvolve a criatividade das crianças e adolescentes de forma lúdica e simples, não necessitando de qualquer conhecimento prévio de mecânica, eletrônica ou programação, sendo o ponto de entrada ideal para desenvolver conhecimentos básicos nessas áreas.

A interface de desenvolvimento (IDE) usada na linha Mindstorms é um software desenvolvido juntamente com a National Instruments (LEGO, 2006) envolvendo a programação em blocos. Este tipo de programação divide diferentes interações entre a central de processamento e seus sensores e atuadores em blocos que desenvolvem diferentes funções. Dentro das configurações de cada bloco, pode-se indicar onde o sensor está conectado, a sensibilidade do sensor, a velocidade e sentido de giro de um motor, número de giros completos que o motor deve realizar, emitir um som ou música, entre outras inúmeras funções. Todas essas funções estão separadas de maneira clara e de fácil seleção através de botões e barras de rolagem, fazendo com que a criação e mudança das funções dos blocos seja fácil e intuitiva.

A programação em blocos também envolve operações matemáticas, lógicas, temporais e de deslocamento de dados, podendo assim controlar por quanto tempo um motor permanece ativo, comparar distâncias, medir temperaturas em graus Celsius ou Fahrenheit e armazenar dados em um cartão micro SD. A geração mais recente, EV3, permite ainda que celulares e tablets sejam usados para programar a unidade de processamento, geração e armazenamento de gráficos de sensores e comunicação entre unidades de processamento via Bluetooth. A

Figura 2 ilustra o ambiente de desenvolvimento usado na programação do LEGO Mindstorms.

Figura 2 – Interface principal da IDE do LEGO Mindstorms NXT



Fonte: Autoria própria.

A *LEGO Education*, divisão da LEGO responsável pelos kits educacionais de robótica, atende 23 estados brasileiros (FEITOSA, 2013), dando suporte técnico e de vendas para kits adquiridos no território nacional por parceiros autorizados. A principal fornecedora de kits é a EDACom Tecnologia, sendo que cerca de 70% de seus clientes são prefeituras, totalizando 3,5 mil escolas públicas e privadas no Brasil (QUINTANILHA, 2008).

Embora seja uma excelente plataforma de robótica educacional, os kits LEGO Mindstorms possuem um preço de aquisição elevado por serem produtos importados e envolverem diversas patentes e tecnologias proprietárias. Não só isso, os kits só funcionam com sensores e atuadores oficiais, com disponibilidade restrita no Brasil a poucos fornecedores, além de quase nenhuma flexibilidade de tipos de sensores e atuadores diferentes que possam ser usados. Isto impossibilita o uso, por exemplo, de displays LCDs, LEDs, sensores de radiação, sensor de umidade, sensor de flexão, módulo RFID e outros sensores e sistemas mais complexos.

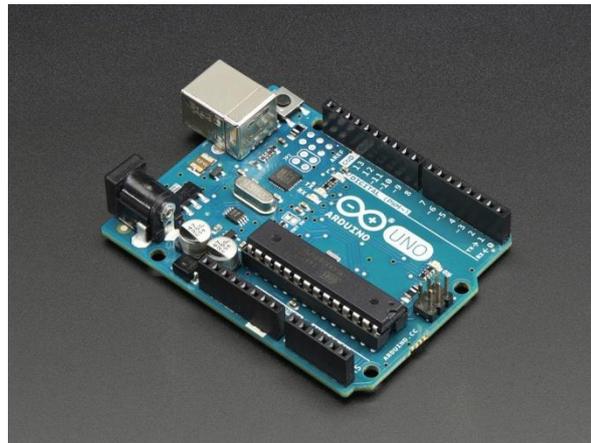
Além destes fatores, devido à linguagem de alto nível usada na programação da unidade de processamento do LEGO Mindstorms, programas criados nesta plataforma possuem lenta velocidade de execução, por se tratar de um código pouco otimizado quando convertido em linguagem binária, que é a linguagem nativa de qualquer processador (MAINI, 2007).

2.2 Arduino

Arduino (2016a) refere-se a uma plataforma de código aberto usada para a prototipagem de projetos eletrônicos, com linguagem e IDE próprias, que pode trocar informações com diversos componentes eletrônicos.

Baseado em uma tese de mestrado que visava uma plataforma em que artistas e designers pudessem trabalhar com projetos eletrônicos de uma maneira mais abstrata (KUSHNER, 2011), as placas Arduino começaram a ser produzidas em 2005 e ganharam imensa popularidade graças à sua facilidade de utilização, compatibilidade com inúmeros sistemas operacionais e preço bem atraente frente a plataformas de desenvolvimento similares. Dentre as placas comerciais mais famosas de Arduino, destaca-se o Arduino Uno R3, mostrado pela Figura 3.

Figura 3 – Placa de desenvolvimento Arduino Uno R3 com microcontrolador Atmel Atmega 328p



Fonte: (ADAFRUIT, 2016).

A vantagem da plataforma ser de código aberto refere-se à disponibilidade do circuito eletrônico de placas comerciais e suas placas de expansão (*shields*) em meios eletrônicos, tais como fóruns e websites. Isto possibilita o projetista basear-se nestas placas para criar a sua própria placa de desenvolvimento, seja para otimizar sua velocidade de processamento, garantir operação em condições climáticas adversas, ou mesmo para garantir funcionalidades extras como comunicação Bluetooth e Wireless diretamente da placa.

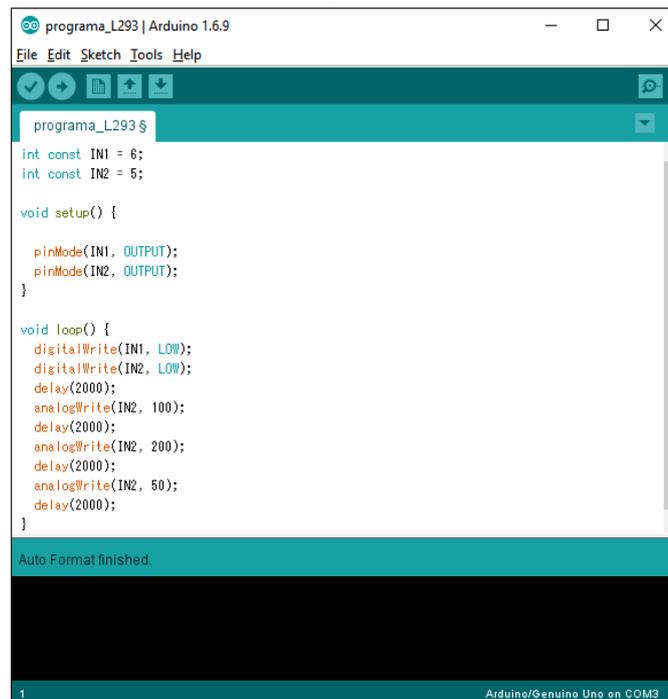
As placas Arduino utilizam microcontroladores Atmel como centro de processamento e controle de informações dos dispositivos conectados ao mesmo. Para a programação do dito microcontrolador, há diversos softwares disponíveis em meios eletrônicos, cada um com suas

especificidades na linguagem. Contudo, será dado enfoque apenas aos softwares mais usados na programação de placas Arduino, com suporte nativo da interface do programa no idioma Português.

2.2.1 Arduino Software

A IDE clássica do Arduino, ilustrada pela Figura 4, é denominada simplesmente como *Arduino Software* e usa uma linguagem derivada da linguagem C++, onde o programa é escrito através de linhas de código e são declaradas variáveis, funções e comandos que ditam as funcionalidades do projeto.

Figura 4 – Interface gráfica do Arduino Software (IDE)



```

programa_L293 $
int const IN1 = 8;
int const IN2 = 5;

void setup() {
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
}

void loop() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  delay(2000);
  analogWrite(IN2, 100);
  delay(2000);
  analogWrite(IN2, 200);
  delay(2000);
  analogWrite(IN2, 50);
  delay(2000);
}

```

Auto Format finished.

1 Arduino/Genuino Uno on COM3

Fonte: Autoria Própria.

A vantagem do uso da IDE clássica está na similaridade com a estrutura de programas de linguagens comumente usadas nos cursos de engenharia e ciências da computação, tais como C++, Java e Python. Tal familiarização ajuda o aluno a entender como o microcontrolador processa informações e como testar seu funcionamento adequadamente com base nos comandos que foram inseridos.

Além de poder programar em uma vasta gama de microcontroladores, esta IDE também conta com um monitor serial via software, que pode ser configurado para ajudar nos

testes de programas mediante *feedback* de informações importantes como monitoramento de variáveis de programa e acionamento de interrupções, sem necessitar de um display LCD físico para ler tais informações.

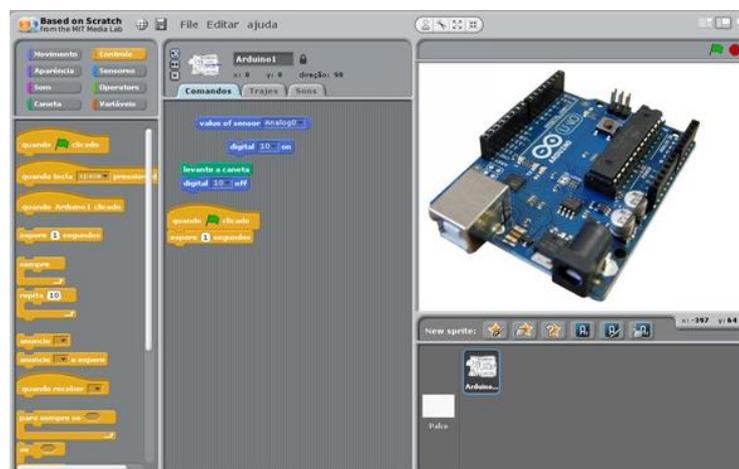
Outra vantagem está no extenso número de bibliotecas disponíveis para este programa, que possibilita suporte fácil para sensores e dispositivos mais complexos como displays LCD, motores de passo e interfaceamento I2C, através de comandos e argumentos próprios de cada biblioteca. Novamente, sendo uma plataforma de código livre, qualquer um tem a liberdade de acessar o conteúdo de tais bibliotecas ou mesmo criar sua própria biblioteca para uso no projeto, dando grande flexibilidade para o projetista.

2.2.2 Scratch for Arduino (S4A)

O Scratch (MIT, 2016) é uma linguagem de programação criada de um projeto do grupo *Lifelong Kindergarten* do MIT, usada especialmente para crianças de 8 a 16 anos de idade, com o intuito de facilitar o aprendizado de linguagem de programação e demais matérias do ciclo básico de educação através de uma interface chamativa e usando linguagem de blocos na construção de programas, aplicativos e jogos.

Já o Scratch for Arduino, ou S4A, é uma IDE para Arduino utilizando uma linguagem Scratch modificada, possibilitando o controle dos componentes conectados ao Arduino semelhante à maneira com que se programam aplicativos e jogos no Scratch. Uma imagem desta interface é ilustrada pela Figura 5.

Figura 5 – Interface gráfica do Scratch for Arduino (S4A)



Fonte: Autoria Própria.

Como a programação usando este software é mais intuitiva e visual que no software convencional, o S4A faz com que o aprendizado de robótica para crianças mais novas seja descomplicado e fluido. Para crianças que não possuem boas noções de lógica ou de programação, tal plataforma pode ser usada para desenvolver tais conhecimentos antes de usar a IDE tradicional do Arduino, de maneira a obter melhores resultados no aprendizado do aluno (MELO et al., 2011).

2.3 Comparação e Foco de Plataformas de Ensino de Robótica

Frente às vantagens e desvantagens citadas nas seções anteriores, pode-se montar o Quadro 1, que se trata de um comparativo das plataformas de robótica educacionais mais comuns eo nível de ensino mais recomendado aos alunos de acordo com cada plataforma.

Quadro 1 – Comparação entre Plataformas de Ensino de Robótica

Plataforma	Microcontrolador	Linguagem	Complexidade	Custo Inicial	Bibliografia Disponível	Nível de Ensino Recomendado
Mindstorms	ARM	Blocos	Baixa	Alto	Alto	Ensino Fundamental/Médio
Arduino	ATMEL	Blocos/Linhas de programação	Média	Baixo	Alto	Ensino Médio/Superior

Fonte: Autoria Própria.

A plataforma Arduino torna-se mais atraente para o ensino de robótica a alunos do ensino médio e superior uma vez que esta plataforma possibilita projetos mais complexos que os possíveis no LEGO Mindstorms, porém com um preço inicial menor, ampla disponibilidade no mercado e extensa literatura em Português.

O LEGO Mindstorms pode ser usado como um primeiro contato à robótica para crianças do ensino fundamental e médio, uma vez que é pouco complexo, intuitivo e atraente a tais públicos. Entretanto, a plataforma Arduino possibilita a construção de sistemas lógicos e sensoriais bem mais diversos, estimulando o conhecimento nas áreas de computação, eletrônica e mecânica de maneira mais eficiente e livre de restrições (OLIVEIRA et al., 2011).

Devido às restrições de aprendizado da plataforma LEGO Mindstorms ao abordar apenas conceitos de programação e mecânica, o material proposto para o curso de robótica será baseado na plataforma Arduino, uma vez que esta explora conceitos de eletrônica e ainda explora outras competências criativas dos estudantes ao incentivar o auto-desenvolvimento e pesquisa independente sobre novas tecnologias aplicáveis à plataforma. Este fator também é

de suma importância ao educador, que pode diversificar seus experimentos e aplicações de forma evolutiva, adaptando-se a tendências tecnológicas e até ampliar o material para cursos mais avançados e específicos.

3 CONCEITUAÇÃO TEÓRICA

Após o estabelecimento da plataforma a ser usada durante o curso de robótica, o próximo passo é a elaboração do material teórico necessário para a compreensão e uso adequado da mesma. O material aplicado deve levar em conta a faixa etária dos estudantes, exemplos didáticos a serem apresentados, treinamento por parte do educador e o tempo de aula necessário para a solidificação do conhecimento pelos alunos, de forma a tornar o curso eficaz e engajador (MATARIC; KOENIG; FEIL-SEIFER, 2007).

Espera-se que o educador tenha um considerável conhecimento prévio da plataforma de ensino de robótica em questão, bem como desenvolva antecipadamente os projetos que planeja fornecer como atividade aos alunos, de forma a prever os erros mais comuns a serem executados pelos mesmos. Também é fundamental a documentação de diferentes aproximações possíveis para a mesma situação-problema e providenciar componentes de reposição para aqueles que forem mais sensíveis ao manuseio.

Foi considerado o material necessário para um curso de 40 horas, divididas entre aulas teóricas e aulas práticas, de forma que, no final do curso, o estudante tenha o conhecimento necessário para compreender a lógica de programação, desenvolver projetos básicos e compreender o funcionamento de dispositivos mais complexos.

A parte teórica deste curso considerará que o estudante tenha noções básicas de Matemática, porém sem nenhuma noção de eletrônica e programação. A noção de lógica básica deve ser reforçada nas primeiras aulas para aperfeiçoar o raciocínio lógico e facilitar o aprendizado no restante do curso, necessitando assim o engajamento dos estudantes por meio de atividades, perguntas e exemplos lúdicos.

O material usado no desenvolvimento da sequência de ensino teórico baseia-se nos livros mais usados em instituições de ensino nos Estados Unidos e no Brasil, bem como os mais vendidos dentre as maiores livrarias destes países (BANZI; SHILOH, 2014) (MCROBERTS, 2011) (MONK, 2010) (MONK, 2012), além de livros usados nos primeiros anos de ensino técnico de eletrônica e informática para o desenvolvimento de conceitos de eletrônica e programação (BOYLESTAD, 2012) (FARRELL, 2012) (MALVINO, 1996).

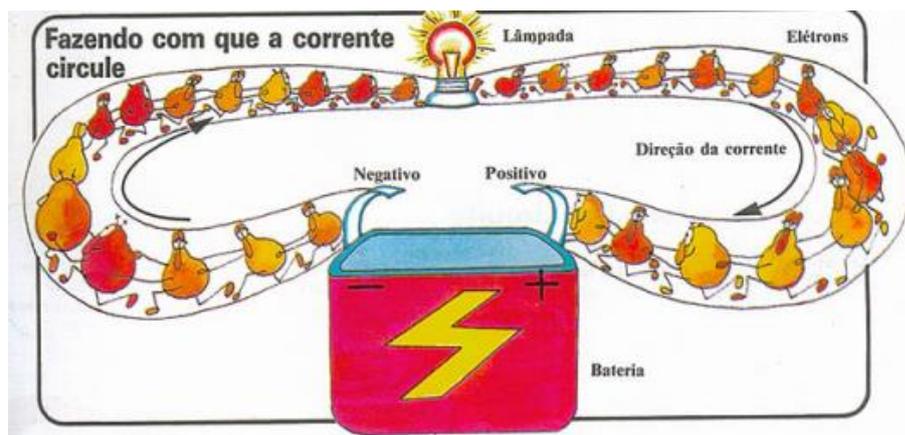
3.1 Introdução à Eletrônica

Em um primeiro instante, é necessário que os estudantes aprendam o funcionamento de circuitos eletrônicos e os termos usados nesta área do conhecimento. Para isto, é

imprescindível que o educador seja capaz de ensinar conceitos de Física de maneira simplificada e com auxílio de imagens, animações, vídeos ou experimentos para que os estudantes absorvam este conhecimento de maneira eficiente.

Preferencialmente, o livro-base de Física a ser usado nas explicações e exemplos também deve abordar o funcionamento de dispositivos eletrônicos básicos, tais como baterias e LEDs, bem como conceitos básicos de eletricidade como resistência, tensão e corrente elétrica, de forma a “casar” os conceitos técnicos com os conceitos previstos na ementa do curso de Física da escola. Livros como (VILLATE, 2012), (GASPAR, 2013) e (MOURA, 2011) são ótimos exemplos de livros a serem usados que abordam tais conceitos do ponto de vista da Física, com linguagens bem visuais e exemplos lúdicos, similarmente como o ilustrado na Figura 6.

Figura 6 – Figura didática explicando o conceito de corrente elétrica



Fonte: (INTRODUÇÃO, 1985)

Durante a primeira aula do curso, o foco deve ser a conexão entre os conceitos aprendidos em tópicos básicos de Matemática e Física para a compreensão de conceitos em eletrônica, incluindo termos mais utilizados, equipamentos e dispositivos básicos, familiarização com a representação de circuitos elétricos e analogias lúdicas sobre o funcionamento básico de circuitos.

Caso seja de interesse do educador, tal explicação pode ser reforçada também na primeira aula de laboratório envolvendo tal tópico introdutório, onde a explicação de conexões de protoboards, manuseio de multímetros e identificação de resistores e demais componentes serão aprofundados. O reforço por meio de exemplos realizados pelo educador e exercícios teóricos e práticos é de suma importância durante as primeiras aulas para que os

alunos exercitem suas competências enquanto se mantêm motivados mesmo quando surgirem exercícios mais complexos em tópicos futuros.

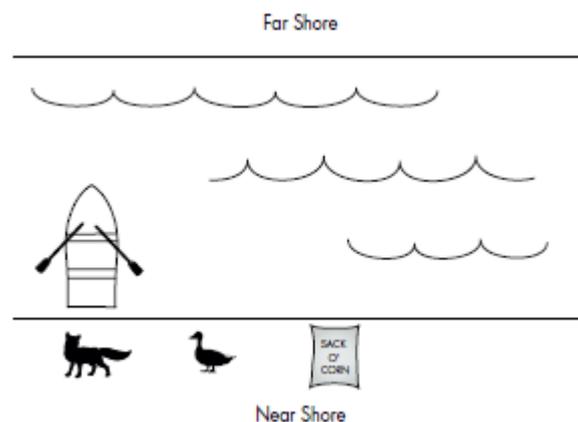
3.2 Introdução à Lógica e Fluxogramas

Tendo-se explicado os conceitos básicos de circuitos elétricos, o foco da aula seguinte deve ser o estudo da lógica, visando o apuramento do raciocínio lógico, capacidade de dedução e pensamento crítico dos estudantes para a solução de situações-problema, competência pouco desenvolvida no sistema de educação brasileira atual (HORTA, 2013).

O aprendizado de lógica deve ser ministrado de forma que os estudantes possam comparar situações cotidianas como pequenos problemas de lógica, visando apresentar soluções segundo uma certa ordem de raciocínio. O uso de parábolas também poderá incentivar o estudante a abstrair seus pensamentos, fazendo com que este consiga chegar em uma solução a partir de um conjunto de informações elementares do problema.

A Figura 7 ilustra um clássico problema de lógica, onde um fazendeiro deve cruzar um rio de barco, transportando uma raposa, um pato e uma saca de milho de uma margem à outra. Neste problema, o barco pode levar apenas um dos itens consigo, além do próprio fazendeiro, sob risco de afundar. Há também restrições quanto aos itens que podem ser deixados nas margens sem a supervisão do fazendeiro: a raposa não pode ser deixada junto com o pato (do contrário a raposa comeria o pato), enquanto o pato não pode ser deixado junto com a saca de milho (do contrário o pato comeria todo o milho).

Figura 7 – Exercício sobre lógica: O problema do fazendeiro, a raposa, o pato e o milho.

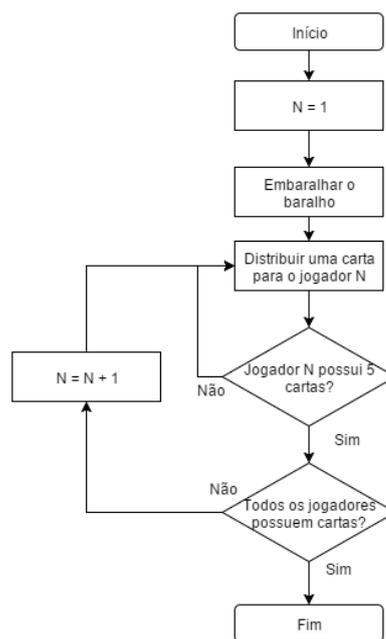


Fonte: (SPRAUL, 2012)

O educador pode usar este problema para incentivar a participação dos alunos, bem como desenvolver suas soluções passo a passo. Este problema também incentiva o pensamento crítico dos estudantes, visto que, em um primeiro olhar, alguns estudantes podem achar o problema impossível por se prender no fato que o fazendeiro deve levar um dos itens até a margem oposta e voltar sem levar nenhum dos objetos de volta.

Outro recurso de extrema utilidade e que deve ser reforçado na aula sobre lógica é o uso de fluxogramas, como mostrado na Figura 8, na elaboração de uma sequência de ações lógicas que compõem a solução para um problema qualquer. Tal estruturação auxiliará o estudante a estabelecer uma relação de causa e efeito de suas ações na solução final, bem como aproximará seu raciocínio com o funcionamento de comandos básicos em microcontroladores em geral.

Figura 8 – Exemplo de fluxograma para uma situação cuja solução consiste no loop de ações previamente executadas.



Fonte: Autoria Própria

Novamente, deve-se incentivar o uso constante da montagem de fluxogramas pelos estudantes, tanto para situações lúdicas quanto situações corriqueiras, bem como fornecer os meios necessários para facilitar a abstração e a generalização de fluxogramas para diferentes situações-problema.

Cenários úteis para a assimilação de tais capacidades podem incluir jogos de cartas, quebra-cabeças (sudoku, torre de Hanoi, cubo mágico), jogos de palavras (caça-palavras,

força, shiritori) e procedimentos em geral (como dirigir um carro, como ler um e-mail, como fazer um bolo).

3.3 Aprofundamento na Linguagem de Programação da Plataforma Arduino

Com as bases teóricas em eletrônica e lógica estabelecidas, progride-se para a familiarização com os comandos nativos da plataforma Arduino. Esta parte teórica reforçará a experiência obtida nas primeiras aulas práticas onde os estudantes já terão aprendido o básico da interface do programa, funções e alguns comandos específicos desta plataforma.

Isto deve incluir explicações sobre tipos de variáveis, tipos de operadores existentes e inicialização de variáveis, caso tais conceitos não tenham sido formalizados anteriormente, tais como os descritos na Tabela 1 e pelo Quadro 2.

Tabela 1 – Tipos de variáveis mais comuns no Arduino

Tipo	Faixa de valores	Número de Bits
boolean	0 ou 1	1
char	-128 até 127	8
int	-32.768 até 32.767	16
long	-2.147.483.647 até 2.147.483.648	32
float	-3,4028235.10 ³⁸ até 3,4028235.10 ³⁸	32

Fonte: Autoria Própria.

Quadro2 – Operadores matemáticos e lógicos básicos no Arduino

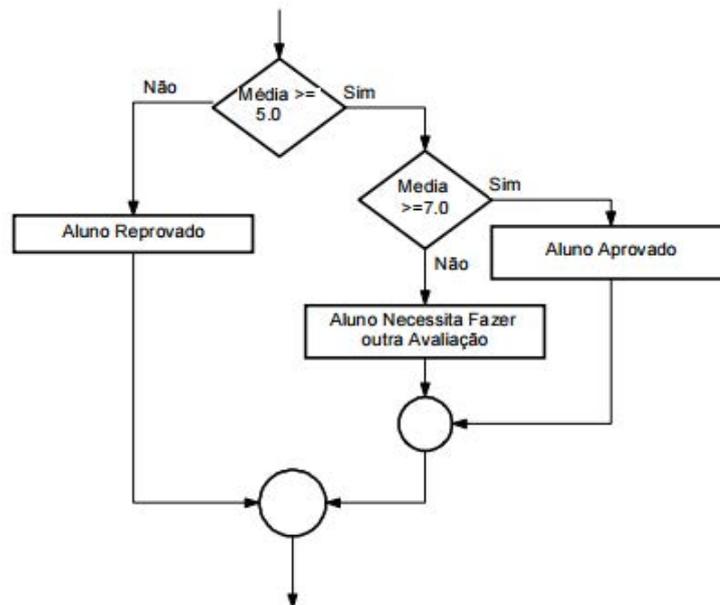
Símbolo	Descrição	Símbolo	Descrição
+	Adição	>	Maior que
-	Subtração	<	Menor que
*	Multiplicação	==	Igual a
/	Divisão	!=	Diferente de
**	Exponenciação	>=	Maior ou igual a
		<=	Menor ou igual a

Fonte: Autoria Própria.

Além do reforço aos comandos básicos, deve ser dada atenção especial no ensino de estruturas condicionais e de repetição de comandos, inspirando os estudantes a escolherem as estruturas que melhor identificarem para usar na solução de problemas propostos durante a aula e dedicados para dever de casa, sendo estes mais difíceis e elaborados para incentivar o autodidatismo e aprofundamento pessoal. Nota-se aqui que o educador deve ter confiança e

domínio das diferentes estruturas, bem como possivelmente a solução para os problemas propostos em mais de uma estrutura para flexibilizar o raciocínio dos alunos, como no fluxograma ilustrado na Figura 9.

Figura 9 – Exemplo de uso de estruturas condicionais cascadeadas em forma de fluxograma.



Fonte: (MORAES, 2000)

É importante salientar que, durante as aulas que este tópico for abordado, o uso de estruturas condicionais e de repetição não necessariamente são de uso único, ou seja, tais estruturas podem estar aninhadas entre si para a execução de programas mais complexos como apresentado na Figura 9.

No caso da plataforma Arduino, deve ser feita a comparação entre as estruturas “*if-else*” e “*switch-case*”, tanto no âmbito útil quanto nos melhores cenários que cada estrutura deve ser usada, uma vez que é recomendado o uso do “*if-else*” para poucas condicionais que dependem de condições lógicas, enquanto o “*switch-case*” é mais usado quando diferentes rotinas de programa dependem de valores específicos de uma variável.

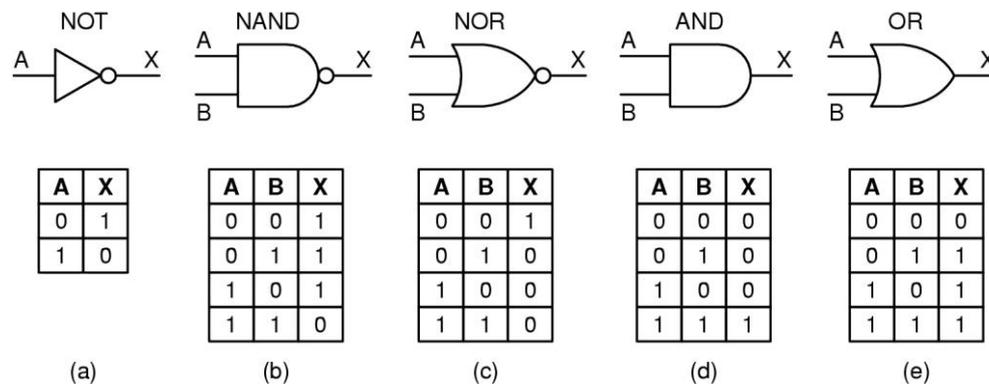
Nas estruturas de repetição de comandos, o foco principal deve ser a estrutura “*for*”, mas as estruturas “*while*” e “*do-while*” devem ser abordadas e comparadas. Aconselha-se que o educador reforce tais conceitos também no laboratório com o uso de programas já feitos para detalhar a sintaxe de cada estrutura e suas particularidades.

3.4 Operações Lógicas e Matemáticas

Durante essa aula, o educador deve se basear nos conceitos explicados nas aulas anteriores para ensinar aos alunos como as operações lógicas funcionam e como construir tabelas da verdade (tabelas de operação) para solucionar problemas lógicos simples. É recomendável também que o educador explique como é a ordem das operações lógicas e matemáticas na plataforma, reforçando os conceitos já estudados pelos estudantes em Matemática por meio de exemplos.

É necessário que o educador dê atenção especial ao ensino de lógica, visto que é um assunto pouquíssimo abordado tanto no ensino fundamental quanto no ensino médio público (SÃO PAULO (Estado), 2011). Portanto, os estudantes podem levar um certo tempo para sedimentarem estes novos conceitos, o que poderá ser avaliado pelo desempenho dos mesmos na aula prática deste tema (discutido na seção 4.3). Introduzir conceitos de eletrônica digital ao mostrar a simbologia, conforme a Figura 10, das diferentes operações lógicas pode ser usado para simplificar tal aprendizado.

Figura 10 – Simbologia de portas lógicas fundamentais e suas respectivas tabelas de operação.



Fonte: (FELGUEIRAS, 2003)

Ainda que a conceituação de lógica voltada à operação do microcontrolador seja de grande valia, deve haver um cuidado para não sobrecarregar os estudantes com tais conceitos de forma a não desinteressá-los ao exigir construção de tabelas verdade de circuitos lógicos complexos. O educador deve se ater a situações práticas e lúdicas, envolvendo poucas variáveis e focando no raciocínio necessário para se desenvolver uma solução no âmbito da eletrônica digital.

Caso haja tempo hábil no curso para maiores explicações, algumas abordagens interessantes neste tópico incluem a exibição de como a unidade microcontroladora da

plataforma executa operações matemáticas complexas ao decompô-las em operações de incremento, decremento e comparação básica, bem como alusões das operações lógicas fundamentais aos circuitos equivalentes usando-se chaves eletrônicas e lâmpadas. Tais demonstrações podem ser feitas na(s) parte(s) prática(s) deste tópico no curso, ou mesmo através de vídeos feitos pelo educador contendo tais circuitos básicos e seus equivalentes em portas lógicas básicas.

3.5 Manipulação de Dados Analógicos

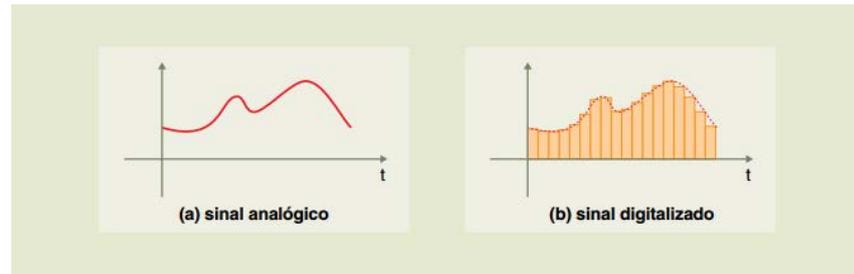
Antes de prosseguir para o estudo de dados analógicos, é necessário que os alunos estejam familiarizados com a manipulação lógica e matemática de dados digitais através das aulas anteriores. Reforçar tais conceitos com exercícios, tanto na sala de aula quanto como dever de casa, bem como a realização de exercícios práticos no laboratório ou com software de simulação apropriado são alguns dos planos de ação necessários para que os estudantes não se desencorajem neste novo tópico.

Neste tema, é necessário que o educador seja claro e conciso na diferença entre dados digitais e analógicos, visto que é comum que haja confusão nesta diferenciação. Logo, é necessário se ater ao conceito básico:

- Informações digitais são aquelas cuja transição é dada de maneira discreta, ou seja, a informação não pode possuir valor intermediário entre dois valores consecutivos.
- Informações analógicas apresentam valores contínuos (FLOYD, 2007), logo, em qualquer instante de tempo, virtualmente qualquer valor dentre seus limites inferior e superior pode ser atribuído à variável em questão.

A título informativo, o educador pode explicar que qualquer sinal analógico pode ser discretizado de forma a criar um sinal digitalizado de formato similar, como ilustrado pela Figura 11. No entanto, o foco deste tópico deve ser a compreensão das diferenças entre informações analógicas e digitais, visto que a conversão de sinais analógicos em sinais digitais será abordado apenas de maneira resumida.

Figura 11 – Diferença entre (a) um sinal analógico e (b) um sinal digitalizado



Fonte: (DIAGO; AMARAL, 2011)

3.5.1 Modulação por Largura de Pulso (PWM)

A modulação por largura de pulso denomina um processo que tem por objetivo a obtenção de uma tensão CC menor ou igual do que o valor fornecido pela fonte CC de entrada (POMILIO, 2014). De maneira simplificada, a modulação PWM consiste no chaveamento, em uma determinada frequência entre dois níveis lógicos de forma a fornecer para a carga uma tensão média de valor menor ou menor que o valor de tensão referente original.

Tal chaveamento pode ser obtido através da comparação entre um sinal de controle e uma onda de tensão periódica, onde o chaveamento ocorre quando ambos os sinais se igualam ou quando completa-se o período do sinal PWM gerado. Uma ilustração deste comportamento pode ser visualizado na Figura 12.

Figura 12 – Ilustração do processo de modulação PWM a partir de uma onda triangular



Fonte: (POMILIO, 2014)

Na Figura 12, o sinal ilustrado na parte superior trata-se de um sinal de tensão analógico, que pode ter qualquer formato, simétrico ou não. Quando o valor de tensão do sinal analógico é igual ou superior ao valor de v_c (valor crítico), o sinal resultante após a modulação PWM atinge o valor de v_o , que é o nível lógico alto que a plataforma trabalha. Similarmente, se o valor do sinal analógico de origem for menor que o valor crítico, o sinal modulado terá valor de tensão igual ao nível lógico baixo do processo de conversão. O valor

V_o , por sua vez, trata-se do valor médio da tensão resultante do sinal modulado via PWM, que é alterado de acordo com o valor crítico estabelecido e pela forma de onda do sinal analógico de entrada.

Para o curso, o importante a se destacar é que a frequência do sinal PWM gerado pelo microcontrolador da plataforma Arduino sempre será fixa, mas o valor médio de tensão do sinal PWM será definido de acordo com a aplicação, seja por meio de valores fixos via software ou dependentes de componentes variáveis como potenciômetros e sensores.

Uma observação a ser feita na abordagem deste tema é que, para se saber com exatidão o tempo em que o sinal de saída permanecerá em nível lógico alto ou baixo, bem como a tensão média do sinal gerado, é necessário a aplicação de algumas fórmulas matemáticas e da análise do sinal por meio de um osciloscópio. Portanto, é esperado que os alunos estejam familiarizados com operações matemáticas envolvendo frações e porcentagens, bem como consigam manipular os osciloscópios disponíveis no curso. Caso haja limitação quanto ao uso de osciloscópios, gráficos de exemplos podem esclarecer tais conceitos, bem como o uso de programas de teste que gerem diferentes sinais PWM e a constatação da tensão calculada com auxílio de um voltímetro.

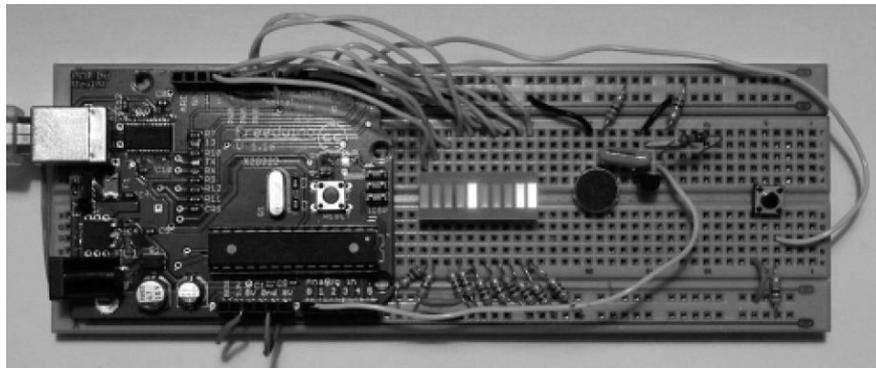
O conhecimento da geração de ondas PWM na plataforma Arduino será de grande valia nos exercícios práticos deste tópico, onde os estudantes terão a oportunidade de controlar, por exemplo, o brilho de um LED conectado ao microcontrolador. Futuramente, tal conceito poderá ser expandido para o controle de velocidade de motores e no controle de frequência sonora emitido por um buzzer, o que abrirá um grande leque de possibilidades para desenvolvimento de projetos mais interativos.

3.6 Sensores Analógicos e Suas Aplicações

Em um primeiro momento, deve-se listar os tipos de sensores existentes, bem como separá-los entre sensores digitais e sensores analógicos, podendo inclusive haver uma comparação entre as vantagens e desvantagens de cada tecnologia. Também deve-se enfatizar que o mundo representa um sistema analógico, e que qualquer grandeza física pode ser medido por sensores analógicos, dentre os quais citam-se: temperatura, luz, distância, som e pressão.

Uma aplicação simples de sensores é ilustrado na Figura 13, que utiliza um display como forma de visualização da intensidade sonora captada por um microfone de eletreto, ilustrando o conceito de conversão analógico/digital e a utilidade de sensores.

Figura 13 – Exemplo de circuito que utiliza um microfone para acender LEDs de um display tipo “bargraph” na representação da intensidade sonora captada



Fonte: (MONK, 2010)

Caso haja tempo suficiente, o educador pode explicar sobre o funcionamento dos sensores na conversão de uma grandeza física em grandeza elétrica, no entanto, o foco maior deve ser dado em entender a proporcionalidade entre o que está se medindo e o valor de tensão lido pelo próprio sensor, bem como transladar a visualização de dados analógicos por meio de LEDs ou com o auxílio da própria IDE do Arduino através do monitor serial presente no mesmo.

3.7 Atuadores Rotativos e seus Acionamentos

Como foco deste tópico, os estudantes deverão aprender o funcionamento básico dos diferentes tipos de motores elétricos disponíveis, bem como seus usos mais indicados, métodos de controle e cuidados no chaveamento dos mesmos para evitar danos aos componentes eletrônicos mais sensíveis a variações de tensão.

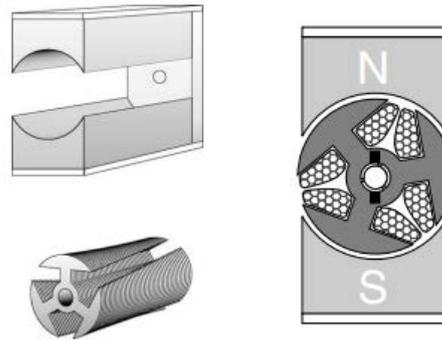
O educador deve se limitar a usar uma conceituação básica, a fim de engajar os estudantes no estudo de motores e suas aplicações em inúmeros eletrodomésticos atuais. Caso os estudantes já tenham estudado o funcionamento de diodos comuns, pode-se aproveitar disto na explicação de circuitos de proteção usados em motores e como tais circuitos se comportam durante o chaveamento e inversão de sentido de corrente no motor.

3.7.1 Motores CC

Motores CC, cuja estrutura é ilustrada na Figura 14, consistem em um conjunto de bobinas, chamado de enrolamentos de armadura, dentro de um conjunto de ímãs permanentes,

denominado estator. Ao se aplicar tensão elétrica nas bobinas, há a produção de torque mecânico na armadura, resultando em movimento do rotor, que é mantido graças ao arranjo comutador que mantém os campos magnéticos gerados pelo rotor e pelo estator em um ângulo reto (HAMBLEY, 2010).

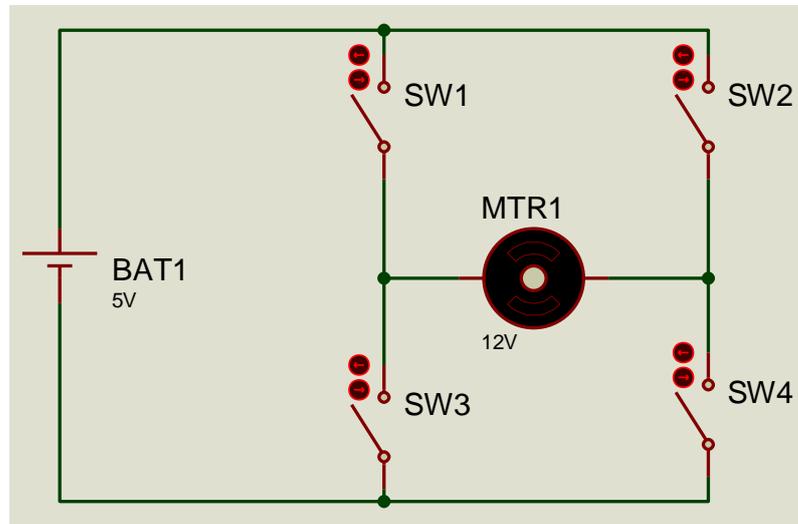
Figura 14 – Representação das estruturas básicas de um motor CC (estator e rotor) usado em brinquedos e seu arranjo.



Fonte: (HUGHES, 2006)

A explicação do funcionamento básico do motor deve se ater apenas a animações ou vídeos bem simplificados por se tratar de um assunto repleto de cálculos, gráficos e termos técnicos vistos no ensino superior. No entanto, faz-se necessário que os estudantes entendam como se faz o controle de sentido de rotação de um motor CC, algo que pode ser explicado com o uso de circuitos que utilizam chaves comutadoras em um arranjo denominado “ponte H”, como o mostrado pela Figura 15.

Figura 15 – Representação do circuito elétrico didático de uma “ponte H” para o controle do sentido de rotação de um motor CC.



Fonte: Autoria Própria

No circuito da figura 15, caso as chaves SW1 e SW4 sejam fechadas enquanto as chaves SW2 e SW3 estejam abertas, haverá corrente circulando no motor e fará com que o mesmo gire em um dado sentido. Caso esta situação seja invertida, ou seja, as chaves SW2 e SW3 sejam fechadas enquanto as chaves SW1 e SW4 estejam abertas, o fluxo de corrente no motor se dará no sentido inverso que no primeiro caso, fazendo com que o motor gire seu eixo no sentido oposto.

Ainda que deva ser explicado o motivo de não se usar chaves mecânicas para o controle de motores CC no mundo real (risco de curto-circuitar a fonte de tensão, possibilidade de danificar as chaves devido à corrente induzida pela bobina interna do motor e impossibilidade de automatização do processo de chaveamento), tal circuito facilita a explicação e o entendimento do conceito, bem como poderá ser tomado como base na explicação de acionadores integrados, como o L293D descrito na seção 3.7.3.

3.7.2 Servomotores

Os servomotores se referem a motores que, ao contrário dos motores CC, não mantêm seu eixo em rotação constante quando aplicada tensão em seus terminais. Ao invés disto, o servomotor mantém seu eixo em um determinado ângulo com o auxílio de um sinal PWM (MONK, 2010).

A grande maioria de servomotores possuem três pinos: um deles recebe a alimentação positiva de tensão (normalmente de +5V a +12V), outro recebe a alimentação de referência (GND), e o terceiro pino é usado para controlar a posição do eixo do servomotor (BASTOS FILHO, 2004). Exemplos de servomotores podem ser vistos na Figura 16.

Figura 16 – Exemplos de servomotores.



Fonte: (BASTOS FILHO, 2004)

O educador deve manter o foco na relação entre a posição do rotor e a largura de pulso necessária do sinal PWM que controla o motor. Para isto, pode-se fazer uso de gráficos exemplificando diferentes posições do eixo e seus respectivos sinais de controle, ou mesmo com o auxílio de softwares de simulação de circuitos elétricos para interação entre um dispositivo de controle (potenciômetro e botões) e um servomotor ligado ao microcontrolador.

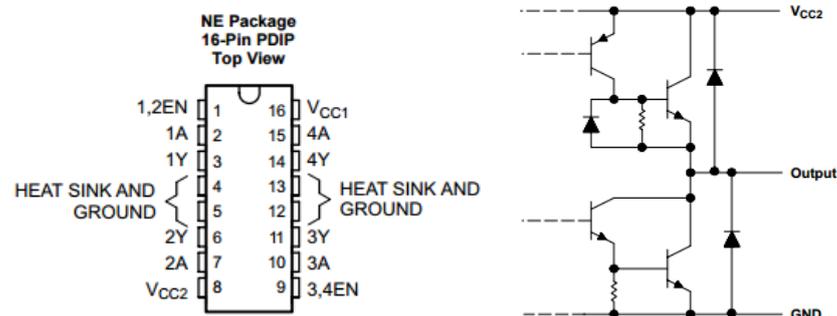
Como a biblioteca utilizada na IDE do Arduino facilita o processo de controle de servomotores através de comandos e argumentos intuitivos, não é necessário que o educador explique os pormenores do funcionamento intrínseco dos comandos usados nesta biblioteca.

3.7.3 O Circuito Integrado L293D

Para facilitar o controle de motores CC, aconselha-se fazer o uso de circuitos integrados especializados para tal, visto que estes CIs já possuem dispositivos de proteção contra ruídos externos, diodos de proteção contra transientes indutivos (TEXAS INSTRUMENTS, 2016a) e proteção contra o efeito *shoot-through*, efeito causado quando um par de chaves de uma ponte H de um mesmo ramal (SW1 e SW3, ou SW2 e SW4 no circuito da Figura 15) são fechados simultaneamente, o que resultaria em um curto-circuito da fonte de tensão do motor (FAIRCHILD SEMICONDUCTOR, 2003).

Se comparado com outros acionamentos (via transistores ou relés), circuitos integrados dedicados para o acionamento de motores possuem um custo-benefício muito superior, já que facilitam a montagem do circuito pelos estudantes, economizam espaço físico e possuem folhas de dados (*datasheets*) explicando o funcionamento dos mesmos com gráficos, tabelas e circuitos de teste. A arquitetura interna e pinagem do L293D pode ser vista na Figura 17.

Figura 17 – Pinagem do circuito integrado L293D (à esquerda) e seu esquemático equivalente para suas saídas (à direita).



Fonte: (TEXAS INSTRUMENTS, 2016a)

Um CI dedicado para acionamento de motores CC é o L293D, da Texas Instruments, que possui quatro metades de uma ponte H integradas em um pacote de 16 pinos, possibilitando o controle “liga-desliga” de até quatro motores em apenas um sentido, ou até dois motores com controle de sentido de rotação. A corrente máxima que o CI pode fornecer para o acionamento de cada motor é de 600mA, valor mais que suficiente para o acionamento de motores CC até 12V usados em brinquedos e em aparelhos leitores de CD e DVD.

O funcionamento do circuito integrado deve ser explicado de tal maneira a formar uma analogia entre o conceito de ponte H e os pinos do circuito integrado. Logo, atendo-se apenas ao aspecto teórico, o educador deve justificar o uso do circuito integrado no controle do acionamento de motores, visto que o microcontrolador usado nas placas Arduino não possui corrente suficiente para acionar cargas que demandam correntes maiores que 40mA, sob risco de danificar a confiabilidade do microcontrolador (ATMEL, 2015).

3.8 Displays e Mostradores Digitais

Neste tópico, deverão ser ensinados os diferentes tipos de mostradores eletrônicos aos estudantes, de forma a fazer com que o microcontrolador da plataforma possa transmitir informações através de elementos visuais mais complexos que simples LEDs. O foco deverá

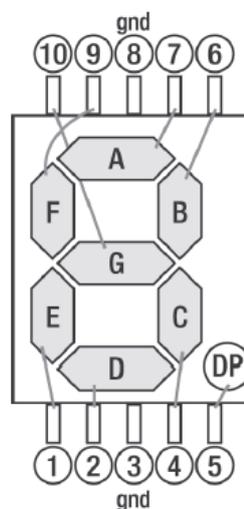
ser dado nas aplicações dos diferentes tipos de mostradores, bem como o método para usá-los de forma coerente e também utilizá-los para apresentar informações em forma de texto

3.8.1 Display de Sete Segmentos

O display de sete segmentos é um dos mostradores eletrônicos mais básicos existentes, tratando-se de um encapsulamento com 8 LEDs, sendo 7 usados para formar um dígito numérico de 0 à 9, e o último LED representando o ponto decimal para a separação da parte fracionária de números decimais. Há também, no mesmo encapsulamento, dois pinos comuns entre si, que podem ser alimentados com a tensão positiva (normalmente +5V) ou negativa (0V, ou GND) dependendo do tipo do display de sete segmentos em questão.

A Figura 18 ilustra a pinagem de um display de sete segmentos do tipo cátodo comum, mostrando qual pino está relacionado à qual segmento do mesmo. Nota-se que, como os terminais comuns do display correspondem ao GND (nível lógico baixo), o pino relativo ao segmento que se deseja acender deverá receber nível lógico alto. Ao se usar um display de anodo comum, ainda que a relação entre os pinos de cada segmento do display não mude, o sinal comum passa a ser uma tensão de +5V (nível lógico alto), e assim, para se acender um segmento, o respectivo pino deverá receber nível lógico baixo.

Figura 18 – Pinagem de um display de sete segmentos cátodo comum e os respectivos segmentos relativos à cada pino



Fonte: (MCROBERTS, 2011)

O controle de displays sete segmentos pode ser feito de duas formas: com a escrita segmento por segmento do valor a ser mostrado pelo mesmo, ou com o auxílio de circuitos integrados que convertam um vetor binário em um valor numérico no respectivo display.

A escrita segmento por segmento é mais didática, porém requer longos programas para o controle simples do display, o que pode soar desinteressante aos alunos. Por outro lado, CIs que fazem a decodificação do código binário para um código compatível com o display sete segmentos (como o SN74LS47 e o CD4511B, ambos da Texas Instruments) realizam tal conversão de maneira mais simples e possuem funções extras para o teste dos displays.

Caso tais CIs sejam usados, será necessário que o educador descreva o processo de conversão entre números de base decimal para base binária de acordo com o sistema BCD8421, uma vez que tal conversão deverá estar presente de alguma forma em programas desenvolvidos pelos alunos para o correto uso do CI utilizado.

No entanto, como as aplicações de displays 7 segmentos se restringem, de um modo geral, apenas a aplicações que envolvem escrita de números, o educador pode optar em dar um foco maior no ensino do display LCD, já que este possui mais aplicações práticas.

3.8.2 Display de Cristal Líquido (LCD)

O display de cristal líquido, também conhecido pela sigla LCD, é um tipo de mostrador eletrônico complexo, visto que cada caractere exibido é composto de uma matriz de 5x7 pixels, podendo assim reproduzir não só números como também caracteres minúsculos e maiúsculos, caracteres especiais, símbolos e caracteres de alfabetos de outras línguas como o Japonês.

Tais displays são normalmente controlados pelo CI HD44780, da Hitachi, dedicado não só a exibir caracteres como mostrado na Figura 19, como também controlar o brilho e contraste dos caracteres para facilitar a leitura, efeitos de animação do cursor e controle da posição de entrada de caracteres (HITACHI, 1998).

Figura 19 – Display de cristal líquido 16 colunas por 2 linhas convencional



Fonte: (ELECTROSOME, 2013)

Ao contrário do display de 7 segmentos, o controle em baixo nível do display LCD, isto é, o controle através de comandos simples e até então já praticados pelos alunos, requer extensa leitura do *datasheet* do controlador HD44780 e demanda grande tempo de adaptação do educador e dos alunos, algo não aconselhável para o ensino de robótica na carga horária definida. No entanto, a IDE do Arduino já dispõe de bibliotecas pré-instaladas que contém comandos de fácil compreensão e aplicação para o controle de tais displays, atentando-se apenas às conexões do display com o microcontrolador e ao tipo do display usado nos comandos de inicialização.

Devido ao controle do mostrador LCD ser feito com mais fios do que nas aplicações do controlador de motor L293D, espera-se que os alunos tenham dificuldade durante a fase prática deste tópico e, assim, é aconselhável que tanto o educador quanto os alunos estejam já familiarizados com o uso do monitor serial presente na IDE Arduino, visto que os comandos usados para a escrita de dados no LCD são bem similares aos comandos de escrita no próprio monitor serial, tornando-o uma ferramenta que deve ser usada para a previsão dos dados a serem exibidos no display físico, servindo como um meio de auxílio na detecção de problemas nos exercícios práticos.

4 ELEMENTOS PRÁTICOS DO CURSO

Estabelecendo-se a base teórica do curso, o educador deve ter em mente que aulas práticas devem ser realizadas paralelamente com as aulas teóricas visando a sedimentação do conhecimento aos alunos e a familiarização com os instrumentos, equipamentos, componentes e programas necessários para a realização das atividades propostas.

Com exceção das primeiras aulas introdutórias, o foco deverá ser a apresentação de conceitos voltados ao que será efetivamente usado durante as respectivas aulas, ou seja, no desenvolvimento de programas e circuitos que realizem a função desejada. Espera-se que os alunos consigam, ao final de cada aula prática, compreender os mecanismos de funcionamento da plataforma Arduino, bem como dos circuitos eletrônicos adicionais no experimento e saibam flexibilizar os programas desenvolvidos para diferentes cenários.

É aconselhável que o educador disponha de um aparelho projetor de imagem (“*datashow*”) para mostrar aos estudantes em tempo real o desenvolvimento de programas e possivelmente montagem dos circuitos necessários por meio de câmeras conectadas ao projetor. Este arranjo possibilita que o educador faça sua explicação simultaneamente para todos os alunos, economizando tempo que será dedicado às montagens de circuitos e ao desenvolvimento de programas. Caso isso não seja possível, o educador pode disponibilizar os programas e desenhos de circuitos de cada aula prática em plataforma online (*intranet*, blogs, pastas compartilhadas) que serão acessadas durante as aulas.

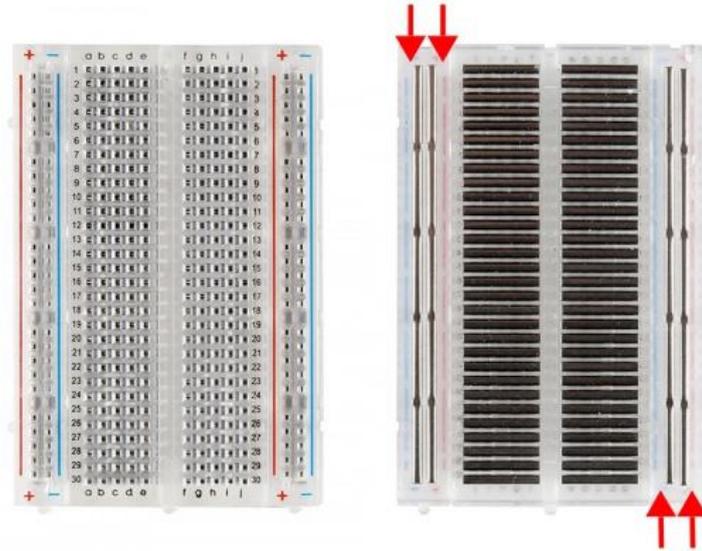
4.1 Aula Introdutória

Na primeira aula, serão usadas aplicações puramente eletrônicas, envolvendo os componentes mais simples possíveis e não exigindo nenhuma programação do Arduino, de forma a familiarizar o aluno aos conceitos básicos da eletrônica, dispositivos usados, procedimento de montagem utilizando protoboard e o uso do multímetro.

4.1.1 Montagem de Circuitos Eletrônicos

A montagem de circuitos eletrônicos sempre será feita em protoboards ou placas similares no decorrer do curso, não entrando no escopo a soldagem de placas eletrônicas. O educador deverá ensinar como utilizar o protoboard, como são dispostas suas conexões internas (conforme Figura 20) e os cuidados a serem tomados para conservá-lo.

Figura 20 – Protoboard (*Breadboard*) convencional (à esquerda) e suas conexões internas (à direita)



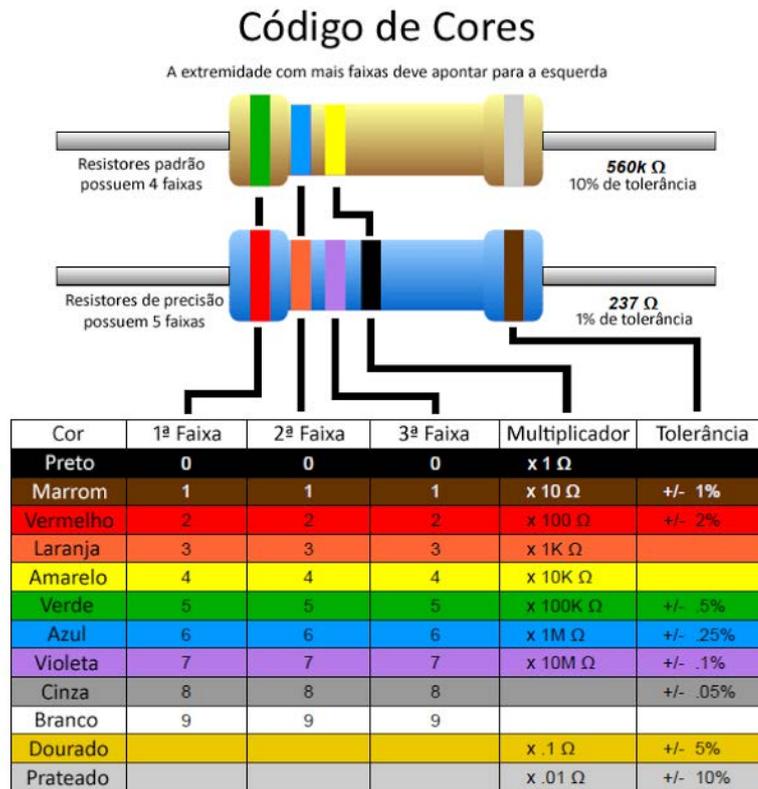
Fonte: (SPARKFUN, 2016)

Além disso, é interessante que o educador ensine sobre as boas práticas de montagem de circuitos eletrônicos, como a conexão dos níveis de tensão positivo e negativo nas respectivas trilhas do protoboard, como inserir componentes corretamente, como descascar fios e uso preferencial de cores para fios e cabos de alimentação (ELENCO, 2012).

4.1.2 Leitura de Resistores e Demais Componentes Eletrônicos

Ao mostrar os diferentes dispositivos eletrônicos que serão manipulados durante o curso, o educador pode aproveitar desta ocasião para falar sobre o código de cores usado em resistores de baixas potências, mostrando o procedimento da leitura e assegurando que os alunos entendam tal procedimento. A ilustração da Figura 21 deve estar, preferencialmente, disponível em sua forma colorida para os alunos, em meio físico ou digital de fácil acesso para que seja possível a consulta durante todas as aulas práticas do curso.

Figura 21 – Código de cores para resistores de baixa potência



Fonte: (UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO, [2015?])

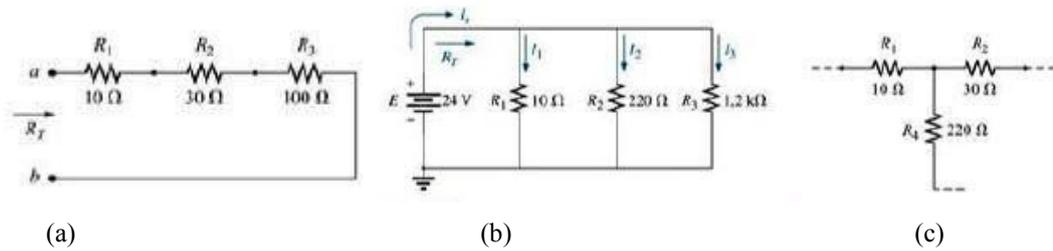
No caso da identificação de demais elementos eletrônicos (circuitos integrados, potenciômetros e sensores), deve-se alertar os alunos que a identificação se dá no próprio componente, bastando apenas identificar o código do respectivo componente e diferenciá-lo da numeração de série ou demais códigos relativos à fabricação do mesmo.

4.1.3 Associação de Resistores

O educador deve reforçar os conceitos de conexão série e conexão paralela com resistores, pois tais conceitos são muito usados em eletrônica e muito úteis para a análise de circuitos que contenham muitos componentes conectados a uma mesma fonte de tensão.

De uma forma bem resumida, a associação em série para resistores ocorre quando há apenas uma conexão em comum entre dois, e apenas dois resistores adjacentes, enquanto a associação em paralelo de resistores ocorre quando dois resistores tiverem dois pontos de conexão em comum entre si (BOYLESTAD, 2012). Deve-se tomar cuidado na explicação da associação em série, visto que três ou mais resistores que compartilham um mesmo ponto de conexão em comum não estarão necessariamente em série entre si, conforme Figura 22 (c).

Figura 22 – Associação de resistores em série (a), em paralelo (b) e associação mista (c)



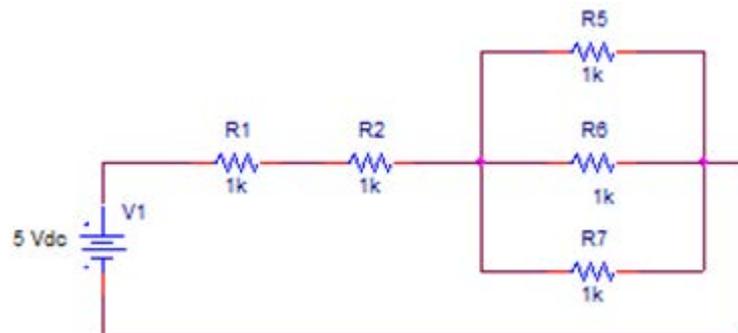
Fonte: (BOYLESTAD, 2012)

As consequências de cada associação são de extrema importância, visto que resistores em série possuem o mesmo valor de corrente circulando entre todos os elementos em série, enquanto resistores em paralelo possuem o mesmo valor de tensão entre suas extremidades para todos os resistores em paralelo entre si. Tais consequências são de grande valia para se determinar a corrente e tensão em uma dada carga.

4.1.4 Equipamentos de Medição em Circuitos Eletrônicos

Ainda durante a aula introdutória, preferencialmente após a montagem de alguns circuitos de testes envolvendo resistores e/ou lâmpadas, o educador deve explicar sobre o uso de multímetros na medição de tensão e corrente, de forma a conferir a aplicação das equações da Lei de Ohm em circuitos práticos, como no circuito da Figura 23.

Figura 23 – Circuito misto didático para ensinado uso do multímetro



Fonte: Autoria própria.

O educador deve explicar sobre a correta posição dos bornes vermelho e preto para a medição de tensão e corrente, a seleção da escala mais adequada de acordo com o que será mensurado (devido à influência da resistência interna do multímetro para cada escala),

posicionamento das pontas de prova na medição da grandeza escolhida (em série com o componente, em caso de medição de corrente, ou em paralelo com o componente, em caso de medição de tensão) e a necessidade do circuito estar energizado ou não para a medição (medição de corrente e tensão necessitam que a fonte esteja ligada, porém para a medição de resistência, a fonte deve estar desligada).

Aconselha-se que também sejam abordados os cuidados na montagem e manuseio de circuitos, como sempre manter a fonte de tensão desligada durante a montagem e sempre que for necessário a retirada de algum componente do circuito. Ainda que dificilmente haja perigos reais aos alunos uma vez que os experimentos envolvem tensões de corrente contínua abaixo de 120V em ambiente seco, respeitando assim o item 5.1.2.5.1 da NBR 5410 (ABNT, 2004), deveser a abordagem dos possíveis riscos à saúde e complicações que choques elétricos podem resultar para o corpo humano, como forma de conscientização dos perigos elétricos dependendo do nível de tensão a que as pessoas se expõem sem equipamentos de proteção.

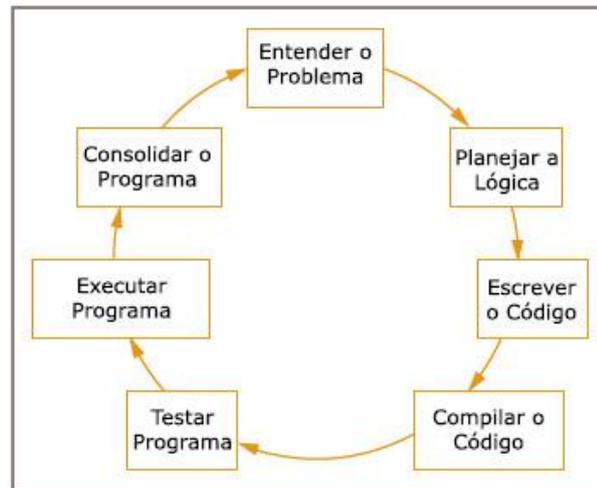
4.2 Introdução à Programação

Passadas as aulas introdutórias práticas sobre os conhecimentos básicos de eletrônica e montagem de circuitos eletrônicos, o educador prosseguirá com o ensino e a aplicação de lógica no âmbito da programação.

Em um primeiro instante, para auxiliar os alunos no método de resolução de problemas e no desenvolvimento do pensamento crítico, o educador deve usar exemplos lúdicos tais como os usados nas aulas teóricas do tópico a fim de realizar um procedimento passo a passo de ações e materiais necessários para resolver o problema. Relacionar fluxogramas com a explicação é de grande valia para contextualizar a explicação e reforçar o uso desta ferramenta.

A Figura 24 ilustra um ciclo a ser seguido no desenvolvimento de programas, onde percebe-se que há sempre espaço para a melhoria contínua nos programas e que não há apenas um único meio para resolver os exercícios propostos ao longo do curso.

Figura 24 – Exemplo de fluxograma para uma situação cuja solução consiste no “loop” de ações previamente executadas.



Fonte: Adaptado de (FARRELL, 2012).

A explicação da lógica deve levantar palavras-chaves como “constantes” e “variáveis” dependentes de um problema, “valores” de quantificação das variáveis e “condições” de resposta do sistema. Assim, o educador deve mostrar que o processo de solução de um problema é similar ao desenvolvimento de um algoritmo na programação da plataforma Arduino, onde a lógica de programação deverá relacionar variáveis, valores e condições por meio de instruções, funções e estruturas condicionais de forma a executar todo o procedimento elaborado.

4.2.1 Utilizando o Software Arduino IDE

Em momento oportuno, o educador deve dividir os alunos em grupos em função da quantidade de computadores disponíveis na sala de laboratório do curso e ensiná-los como abrir a interface de desenvolvimento Arduino e executar ações básicas no mesmo, tais como: abrir programas já feitos, abrir exemplos pré-instalados no software, estabelecer a comunicação entre o software e a placa física da plataforma Arduino utilizada e como realizar o *upload* do programa para a placa.

Espera-se que o educador tenha previamente instalado os drivers necessários em todos os computadores para que estes possam reconhecer a placa Arduino, bem como solucionar problemas básicos de comunicação.

4.2.2 Comandos Básicos na Plataforma Arduino

Recomenda-se que, para os primeiros programas de cada aula prática a partir deste tópico, o educador forneça aos alunos programas funcionais previamente desenvolvidos ou exemplos já pré-instalados no software Arduino para o ensino de novos comandos da plataforma e seus parâmetros.

Durante a primeira aula de programação efetiva por parte dos alunos, o educador pode utilizar o programa “*blink*” pré-instalado no Arduino como uma primeira aproximação às primeiras funções e comandos usados no Arduino. Uma vez que há um LED conectado internamente ao pino 13 da placa Arduino Uno (BANZI; SHILOH, 2014), não será necessário montagem alguma por parte dos alunos. Caso outra placa seja utilizada, pode ser que o educador necessite explicar o processo de montagem e conexão de um LED e um resistor limitador de corrente do *protoboard* ao Arduino.

Deve ser ressaltada a diferença entre as funções *setup* e *loop*, onde a primeira é executada primeiro e apenas uma única vez quando o programa é executado pela plataforma, enquanto os comandos inseridos na função *loop* são repetidos indefinidamente após todos os comandos da função *setup* serem executados. Os diferentes comandos como “*pinMode*”, “*digitalWrite*” e “*delay*” também devem ser explicados, sendo que o educador deve citar os diferentes parâmetros que cada comando utiliza por meio dos significados das palavras reservadas em inglês usadas nos mesmos e o que cada valor numérico efetivamente significa no comportamento destes comandos.

Ainda neste primeiro contato com comandos e instruções usadas no Arduino, aconselha-se motivar os alunos a realizarem algumas mudanças básicas no código para verificar a diferença no comportamento do programa. Para o exemplo do programa “*blink*”, pode-se alterar o tempo em que o LED permanecerá aceso ou apagado, o pino que será tratado como saída (necessitando a montagem de um circuito básico que conecte um resistor limitador de corrente e um LED na porta de saída especificada) e a inclusão de mais comandos “*digitalWrite*” para que múltiplos LEDs pisquem ao mesmo tempo.

4.3 Implementação das Estruturas Condicionais e de Repetição

Ao longo deste tópico, o objetivo é que o aluno consiga aplicar as estruturas condicionais e de repetição de comandos na linguagem da plataforma Arduino, estabelecendo

uma relação entre fluxogramas que descrevam o comportamento pretendido e os comandos e parâmetros que devem ser incluídos no programa para a execução de tal comportamento.

Primeiramente, é necessária a compreensão da estrutura “*if-else*”, que se trata da estrutura condicional básica de programação. Nela, o programador pode estipular uma determinada condição e fazer com que o programa execute diferentes blocos de instruções caso tal condição seja verdadeira ou falsa. Estabelecer a condição da estrutura envolve fazer a declaração correta de variáveis (uma vez que o parâmetro usado deve ser condizente à variável, sendo que tais parâmetros são diferentes para variáveis numéricas e lógicas) e o uso do operador lógico correto de acordo com o comportamento pretendido.

No programa de exemplo mostrado na Figura 25, nota-se que há a inclusão de diversos comentários, o que auxilia os alunos a entenderem o comportamento do programa linha por linha. Deve-se encorajar os estudantes a usarem comentários em seus programas sempre que o comportamento do programa for alterado, de forma a facilitar a leitura do programa futuramente, bem como na identificação de erros de lógica.

Figura 25 – Exemplo de programa na IDE Arduino contendo estrutura condicional para controlar um LED com apenas um botão.

```

LEDControlado
// Programa 3 - LED Controlado por Botão

void setup() {
  pinMode(1, OUTPUT); // Pino 1 será tratado como saída de dados.
  pinMode(2, INPUT);  // Pino 2 será tratado como entrada digital de dados
}

void loop() {
  if (digitalRead(2) == HIGH) {
    // Estrutura condicional "if": Se a condição entre parênteses for verdadeira, executam-se os
    // comandos escritos entre chaves. O comando "digitalRead" serve para ler o nível lógico de um pino do Arduino.
    // Neste caso, lê-se o nível lógico do pino 2.
    // Aqui, a condição do "if" é verificar se o pino 2 do Arduino está em nível lógico alto.
    // Não se esqueça de abrir da estrutura "if" !!
    digitalWrite(1, HIGH); // Se o nível lógico do pino 2 for nível lógico alto, acende-se o LED conectado no pino 1.
  } // Fechando-se as chaves do condicional "if", encerra-se a descrição dos comandos que serão executados
  // caso a condição seja verdadeira.
  // Caso a condição do "if" seja falsa, o programa ignorará as linhas de comando escritas entre as chaves do "if".
  else // Caso a condição "if" descrita anteriormente seja falsa, executar-se-ão os comandos escritos entre chaves.
  {
    digitalWrite(1, LOW); // Se o nível lógico do pino 2 for baixo, apaga-se o LED conectado no pino 1 do Arduino.
  }

  // Não se esqueça de fechar as chaves da função loop!!
}

```

Fonte: Autoria Própria

Já para a estrutura “*for*”, o educador deve tomar cuidado na explicação dos argumentos desta estrutura, uma vez que pode haver confusão quanto à ordem ou à utilidade de cada argumento ao longo do curso. Além disso, algo interessante a se desenvolver nesta estrutura é

a parametrização de comandos com base nos valores armazenados por variáveis do programa, conhecimento tal que possibilita que comportamentos complexos sejam realizados de maneira simples e com poucos comandos atrelados a essa estrutura.

Um exemplo de código que utiliza a estrutura “*for*” de maneira simples está presente na Figura 26. Nota-se neste programa que há o uso de declaração de variáveis e incremento de variáveis, conceitos não utilizados no programa da Figura 25. Cabe ao educador explicar os benefícios do uso de variáveis para facilitar a leitura do programa, explicando as boas práticas a serem seguidas na declaração de variáveis de acordo com a sua função no programa, além de restrições quanto ao uso de acentos nestas variáveis (ressaltando também que nomes de programas também não podem incluir acentos para evitar erros por parte do software).

Figura 26 – Exemplo de programa contendo estrutura de repetição de comandos para o acendimento sequencial de LEDs conectados em vários pinos da plataforma Arduino.

```

DeslocamentoLED $
// Programa 5 - Deslocamento de LEDs Usando Estrutura "for"

int numero; //Definição de uma variável auxiliar inteira "numero".

void setup() {
  pinMode(0, OUTPUT); // Define pino 0 como saída.
  pinMode(1, OUTPUT); // Define pino 1 como saída.
  pinMode(2, OUTPUT); // Define pino 2 como saída.
  pinMode(3, OUTPUT); // Define pino 3 como saída.
  pinMode(4, OUTPUT); // Define pino 4 como saída.
}

void loop() {
  for (numero = 0; numero <=4 ; numero++) {
    //Estrutura for: for("condição inicial" ; "condição para repetição dos comandos" ; "incremento").
    //
    // Esta estrutura significa que, partindo-se da condição inicial,
    //enquanto a condição final não for atingida, os comandos escritos entre chaves serão executados em loop.
    // Neste caso, a variável auxiliar será inicializada com o valor zero, e a cada iteração do loop, seu valor será incrementado.
    // Em outras palavras, serão executados os comandos entre chaves 5 vezes
    //(após a quinta execução, "numero" será igual a 5, tornando falsa a condição de repetição).
    digitalWrite(numero, HIGH); // Acende-se o LED referente ao valor de "numero".
    delay(500); // Mantém este estado por meio segundo.
    digitalWrite(numero,LOW); // Apaga-se o mesmo LED.
  } //Não se esqueça de fechar as chaves da estrutura "for"!!

}

```

Fonte: Autoria Própria

A exemplo da estrutura condicional básica “*if-else*”, os estudantes já seriam capazes de estabelecer a lógica de um sistema de semáforo em um cruzamento com trânsito de carros, com suporte para travessia de pedestres mediante pressionamento de um botão. Ainda que possa ser montado uma maquete para implementação de um programa como este, o foco maior deve ser motivar os estudantes a aplicar seus conhecimentos já obtidos em diferentes

aplicações, não se esquecendo de desenhar fluxogramas previamente para facilitar no processo de escrita do programa.

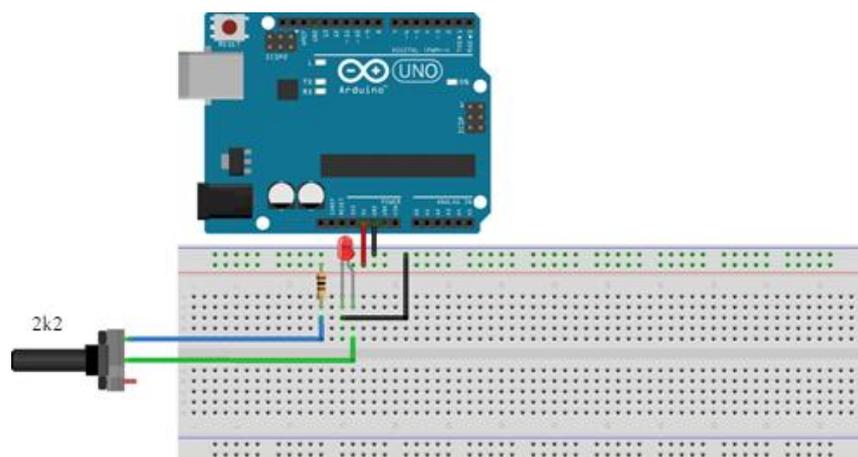
4.4 Leitura de Dados Analógicos e Modulação PWM no Arduino

Neste tema, o educador deve utilizar os conceitos já explicados na parte teórica do curso referente a dados analógicos e o processo de modulação PWM, para então explicar os comandos usados na plataforma de modo a realizar a leitura e, por consequência, gerar a onda modulada de tensão para uma dada carga.

Talvez o exemplo mais simples possível, que trate ambos os conceitos, é o uso de um potenciômetro ligado a uma entrada analógica do Arduino que controle o brilho de um ou vários LEDs conectados em pinos de saída digital que possuam suporte à geração de onda PWM (em se tratando do Arduino Uno, é o caso dos pinos 3, 5, 6, 9, 10 e 11).

O educador pode optar por, primeiramente, realizar tal exemplo com um circuito que independa de qualquer programação do Arduino ao conectar o potenciômetro em série com um LED conforme ilustrado na Figura 27. Nota-se que neste circuito há um resistor limitador de corrente de 100 Ohms conectado em série com o potenciômetro para evitar danos ao LED quando o aluno colocar o potenciômetro no menor valor possível, já que LEDs comerciais possuem tensão típica entre 1,5V e 2,5V, e correntes entre 10 e 50mA (MALVINO, 1996).

Figura 27 – Circuito de demonstração de regulação de brilho de um LED



Fonte: Autoria Própria

É importante que o educador saliente as vantagens do controle de brilho de um LED via PWM ao invés de se usar um potenciômetro conectado em série com o mesmo

(automatização do processo com base em outros dispositivos conectados ao microcontrolador, controle fino e preciso do brilho do LED, controle de múltiplos LEDs simultaneamente para um mesmo brilho, etc.), bem como atentar os alunos que o comando “*analogWrite*” não funcionará corretamente em pinos que não possuam suporte PWM.

Deve ser dada atenção especial na explicação dos valores retornados pelos comandos “*analogRead*” e “*analogWrite*”, uma vez que estes comandos retornam, respectivamente, valores entre 0 e 1023, e valores entre 0 e 255. Tal diferença se deve ao fato do Arduino possuir módulo de conversão Analógico/Digital (conversor A/D) de 10 bits, enquanto o módulo interno de geração de onda PWM possui 8 bits (ATMEL, 2015). Logo, simplesmente realizar a leitura através do comando “*analogRead*”, armazenar este valor em uma variável e atribuí-la como argumento de um comando “*analogWrite*” pode gerar erros ou comportamentos imprevisíveis, e portanto o educador deve abordar tal distinção e como contornar tal impasse.

4.4.1 Uso do Monitor Serial da IDE Arduino

Um importante recurso da IDE Arduino a ser introduzido aos alunos é o uso do monitor serial, onde será possível realizar o monitoramento de diversos dispositivos de entrada e de saída conectados à plataforma por meio de uma janela de texto capaz de reproduzir valores de variáveis do programa em execução, auxiliando no processo de desenvolvimento de programas e adequações lógicas quando os alunos depararem com comportamentos não esperados.

Para usar o monitor serial, o programa deve primeiramente iniciar a comunicação serial entre o microcontrolador e o computador por meio do comando “*Serial.begin(9600)*” que deve ser colocado dentro da função *main*, sendo que o argumento 9600 refere-se ao *baud rate*, termo em inglês que indica a velocidade de transmissão de dados em sistemas de comunicação serial (MICROCHIP, 2003). Para a comunicação eficiente entre a placa Arduino e o computador, recomenda-se que apenas o *baud rate* de 9600 seja utilizado em todos os programas que utilize o monitor serial para evitar perda de informações e falha de sincronismo em aplicações.

Em seguida, pode-se escrever textos e valores de variáveis através do comando “*Serial.print()*” e “*Serial.println()*”, sendo que textos estáticos devem ser inseridos entre os parênteses destes comandos, começando e terminando com aspas duplas, enquanto para valores de variáveis basta colocar o nome da variável entre os parênteses destes comandos.

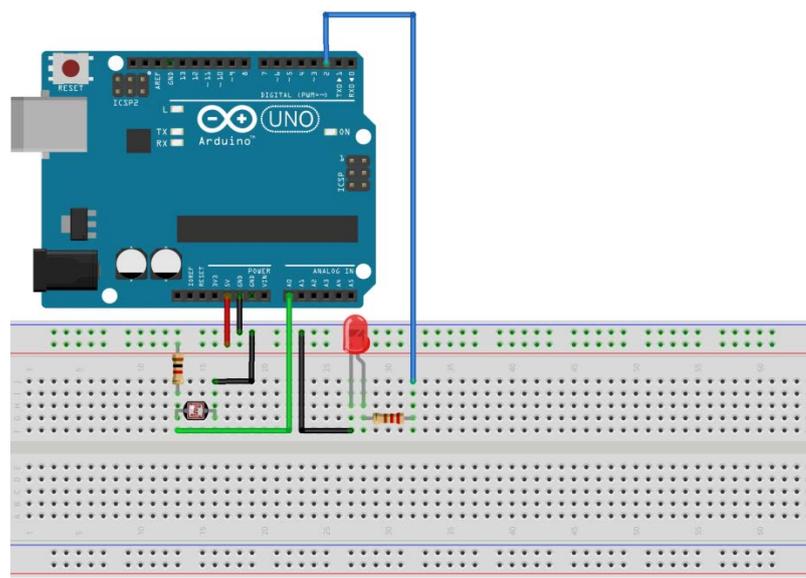
resistores cujo valor de resistência varia proporcionalmente de acordo com a luminosidade do local) por serem os mais fáceis de serem implementados e com circuitos relativamente simples de montar.

Por não haver novos comandos a serem aprendidos pelos alunos na utilização de sensores, o educador deve focar em ensinar os conceitos de divisor de tensão entre resistores e LDRs, bem como os primeiros passos necessários para o planejamento e construção de circuitos adequados para cada sensor, que deve envolver a construção de circuitos protótipos intermediários para otimizar a leitura e escrita de informações.

O circuito da Figura 29 ilustra um divisor de tensão feito com um resistor e um LDR. Devido a variações na construção de LDRs, haverá leituras de tensões diferentes para dois LDRs distintos, mesmo implementando-se um mesmo resistor divisor de tensão em série com este. Por isso, o valor de limiar usado para o programa distinguir entre um ambiente claro ou escuro dependerá da leitura de tensão em ambas as situações, e, portanto, os estudantes deverão chegar nestes valores para o melhor funcionamento possível do programa.

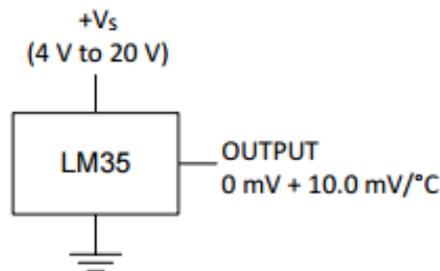
Naturalmente, ainda se tratando do circuito da Figura 29, caso o LED e o LDR sejam posicionados muito próximos um do outro, o programa do sensor crepuscular piscará o LED de maneira rápida em ambientes escuros ao invés acender permanentemente, uma vez que a própria luz emitida pelo LED interferirá na tensão entre os terminais do LDR.

Figura 29 – Exemplo de circuito com LDR para funcionar como um “sensor crepuscular”, acendendo o LED na ausência de luz e apagando-o haja luz suficiente



Outro exemplo de sensor a ser implementado é o LM35, que é um sensor de temperatura ambiente muito simples de ser usado, como pode ser constatado pela Figura 30, uma vez que, com alimentação direta entre 4V e 20V em corrente contínua, o sensor já é capaz de medir temperaturas entre 2°C e 150°C (TEXAS INSTRUMENTS, 2016b). Caso estejam disponíveis fios jumpers (também conhecidos como cabos dupont) do tipo macho/fêmea, a implementação pode até mesmo ser feita sem o uso do protoboard, conectando os fios jumpers correspondentes diretamente da placa Arduino para o sensor, bastando a conexão do pino de saída do sensor em uma das portas de entrada analógica da placa.

Figura 30 – Uso básico do sensor LM35 para medição de temperaturas entre 2°C e 150°C



Fonte: (TEXAS INSTRUMENTS, 2016b)

Um ponto importante do uso deste sensor está na relação entre a temperatura medida e a tensão lida no pino correspondente do sensor. Os alunos podem pensar erroneamente que a leitura direta do comando “*analogRead()*” já resultará na leitura em graus Celsius do sensor, porém o educador deve esclarecer aos alunos que é necessário a prévia conversão do valor de leitura do sensor em um valor na escala de graus Celsius.

Tal conversão deve ser feita em duas etapas, relacionando a tensão de saída do LM35 em um valor inteiro de 0 até 1023, e relacionando este valor a um valor de temperatura.

O microcontrolador tratará a tensão de 0V (tensão do pino GND) como uma leitura de valor 0 do conversor A/D, enquanto a tensão de 5V (tensão medida do pino 5V da placa Arduino) terá uma leitura de valor 1023. Utilizando-se de uma regra de 3 simples, é possível escrever a Equação 1:

$$V_{min} = \frac{V_{cc}}{N_{valores_possiveis}} = \frac{5}{1023} \cong 4,89mV \quad (1)$$

Sendo V_{min} o valor de tensão correspondente mínimo detectável pelo microcontrolador da placa Arduino, e V_{cc} o valor de tensão correspondente ao nível lógico alto da plataforma.

Em seguida, como 10mV lidos do sensor LM35 correspondem a um grau Celsius (TEXAS INSTRUMENTS, 2016b), pode-se dizer que, com certa aproximação, basta dividir o valor lido do conversor A/D por 2 que resultará no valor de temperatura em graus Celsius. Realizando a conversão de maneira mais exata, tem-se a Equação 2:

$$T = N_{lido} \cdot \frac{V_{min}}{(10 \cdot 10^{-3})} = N_{lido} \cdot \frac{5}{(1023 \cdot 10 \cdot 10^{-3})} = N_{lido} * \frac{5}{10,23} \quad (2)$$

Sendo N_{lido} o valor retornado pelo comando “*analogRead()*” na programação do Arduino, e T o valor da temperatura medida pelo sensor, em graus Celsius.

Devido a tais conversões, é necessário que os alunos compreendam o procedimento utilizado pelo educador durante a explicação da mesma e dos passos necessários para se chegar na Equação 2. Para se obter uma exatidão ainda maior entre o valor de temperatura medida pelo LM35, recomenda-se substituir o valor “5” da Equação 2 pelo valor de tensão real medido no pino “5V” da placa Arduino. Cabe ao educador explicar que nem sempre o regulador de tensão interno da plataforma gera uma tensão de exatos 5V para alimentação do microcontrolador devido a diversos fatores, sendo, dentre os mais comuns, a quantidade de corrente consumida pelo circuito total conectado ao microcontrolador, qualidade do cabo USB utilizado, comprimento do cabo USB e variações de temperatura de operação do regulador de tensão.

4.6 Utilização e Controle de Motores CC e Servomotores

Sabendo-se o quão diferentes motores CC são de servomotores, naturalmente, seus métodos de controle pelo microcontrolador se dão de formas diferentes, conforme explicado na parte teórica acerca deste tópico. Com isso, o educador deve separar tempo adequado de suas aulas práticas para o ensino de acionamento de motores CC, e, em outra aula, o acionamento de servomotores, de forma a fornecer tempo suficiente para a absorção dos conceitos por parte dos alunos e para evitar confusões entre os métodos de controle.

Como motores CC requerem correntes maiores que o máximo fornecido pelas portas programáveis de entrada e saída digital de dados do microcontrolador (também conhecidas como GPIOs), conforme discutido na seção 3.7.3, tal controle deve ser feito através de

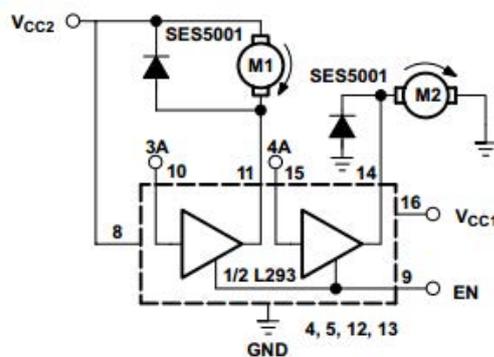
dispositivos auxiliares, seja por meio de relés, transistores ou circuitos integrados especiais. Para facilitar o aprendizado e por questões de economia de tempo do curso, serão abordadas apenas soluções que utilizem o circuito integrado L293D.

4.6.1 Controle Tipo “Liga/Desliga” (ON/OFF) de um Motor CC

O controle “Liga/Desliga” (ON/OFF) para um motor CC apenas se preocupa com a energização e a desenergização do mesmo, baseado em um comando temporizado ou controlado do microcontrolador da placa Arduino. Portanto, este tipo de controle não se preocupa com a comutação no sentido de giro do motor, controle tal que deve ser abordado após o entendimento deste acionamento mais simplório.

A Figura 31 ilustra o circuito necessário para este acionamento, extraído do próprio *datasheet* do fabricante do componente. Nota-se aqui que os diodos reversamente polarizados “SES5001” tratam-se de dispositivos de proteção para os motores, também conhecidos como diodos de *flyback*, utilizados para o escoamento de picos de corrente induzida na desenergização das bobinas internas destes motores de forma a prolongar a vida útil do circuito integrado e demais componentes sensíveis a impulsos de corrente (HAMLIN, 2013). Tais diodos já são integrados no CI L293D, porém, caso algum CI alternativo seja implementado e não possua tais diodos integrados em seu encapsulamento, aconselha-se usá-los para acostumar os alunos com as boas práticas de montagem de circuitos envolvendo motores CC e evitar danos aos materiais eletrônicos usados.

Figura 31 – Circuito elétrico para o acionamento “Liga/Desliga” de um motor CC utilizando o circuito integrado L293D



Fonte: (TEXAS INSTRUMENTS, 2016a)

A entrada 3A (pino 10) do circuito integrado L293D controla o motor M1 (conectado no pino 11), enquanto a entrada 4A (pino 15) controla o motor M2 (pino 14). Nota-se que o circuito integrado deve estar alimentado com uma tensão entre 4,5V e 36V no barramento VCC2 (que será a tensão de alimentação dos motores), enquanto o barramento VCC1 deve estar alimentado de 4,5V até 7V (que será a alimentação dos dispositivos lógicos internos do CI). Portanto, visando a preservação do funcionamento esperado do CI, recomenda-se que ambas as alimentações VCC1 e VCC2 sejam levadas à tensão de 5V da própria placa Arduino, tensão mais que suficiente para acionar motores de dimensões pequenas usados no curso de robótica.

Este circuito se comporta da seguinte forma: enquanto o pino 9, ou seja, a entrada “EN” (abreviação de “*Enable*”, que significa “Habilitação”) estiver em nível lógico baixo, os motores M1 e M2 não girarão, independente do nível lógico presente nas entradas 3A e 4A. No entanto, enquanto o pino EN estiver em nível lógico alto, se a entrada 3A receber nível lógico baixo, o motor M1 girará em um dado sentido, ao passo que se esta mesma entrada receber nível lógico alto, o motor M1 irá parar sua rotação. Tal lógica se inverte para o motor M2: inserindo-se nível lógico alto na entrada 4A, o motor M2 girará, enquanto nível lógico baixo no mesmo pino fará o motor M2 parar.

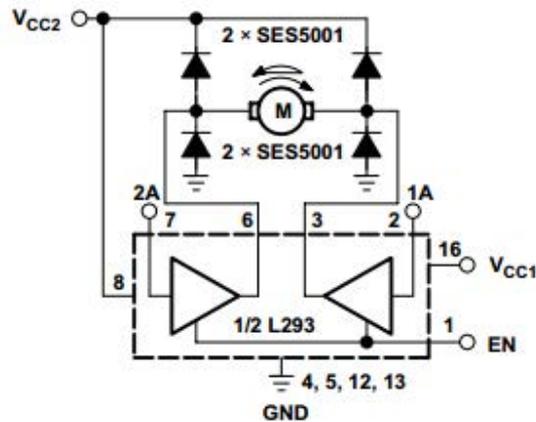
Ainda que tal circuito pareça básico aos alunos, o educador deve ressaltar a importância de aprender tal método de controle, visto que derivados destes circuitos estão presentes até nos dias hoje em fontes de potência de computadores, geladeiras, exaustores e sistemas de refrigeração de ar. Tal discussão é de vital importância para desenvolver o pensamento crítico dos alunos, bem como implementação dos conceitos de sensores vistos no tópico anterior para a criação de novos sistemas que possam utilizar este controle simples de motores CC.

4.6.2 Controle Bidirecional de um Motor CC

Após as atividades envolvendo o controle “Liga/Desliga” de motores convencionais CC, o educador deve guiar os alunos para a montagem de um circuito de acionamento com controle de sentido de giro de um motor CC, tomando-se como base os conceitos vistos sobre “Ponte H” presentes na seção 3.7.1.

A Figura 32 ilustra o circuito que possibilita tal controle. Caso os diodos “SEN5001” não sejam utilizados, os terminais do motor CC deverão estar conectados entre os pinos 3 e 6 do L293D, deixando o barramento de tensão VCC2 conectado apenas no pino 8 do mesmo CI.

Figura 32 – Circuito elétrico para o controle de sentido de giro de um motor CC utilizando o circuito integrado L293D



Fonte: (TEXAS INSTRUMENTS, 2016a)

Similarmente como visto no funcionamento do circuito da Figura 31, enquanto a entrada “EN” (pino 1) do L293D estiver em nível baixo, o motor não girará em nenhum sentido, independente dos níveis lógicos inseridos nas entradas 2A e 1A. Mantendo-se a entrada “EN” em nível lógico alto, caso as entradas 2A e 1A possuírem níveis lógicos opostos entre si, o motor girará para um dado sentido, sendo que tal sentido é invertido caso os níveis lógicos nestas entradas sejam invertidas. Caso os níveis lógicos das entradas 2A e 1A forem iguais, o motor não irá girar em nenhum sentido, uma vez que a alimentação do mesmo será removido devido à lógica interna do circuito integrado.

É necessário ressaltar que, para o circuito integrado L293D, há duas entradas “EN” para cada CI: a entrada “EN” localizada no pino 1 (chamada de “EN1” no software Proteus ou “1,2EN” segundo o *datasheet* do fabricante) controla a habilitação dos drivers 1A e 2A, enquanto a entrada “EN” do pino 8 (chamada de “EN2” ou “3,4EN”) controla os drivers 3A e 4A. Assim, cada par de drivers internos do CI são independentes entre si e devem ser habilitados individualmente de acordo com a aplicação.

4.6.3 Controle de Servomotores

Para o controle de servomotores, utiliza-se uma biblioteca chamada “*Servo.h*”, já pré-instalada no software Arduino. Nesta biblioteca, referencia-se um objeto da mesma logo após a sua inclusão, como pode ser visto no programa de exemplo da Figura 33, na linha “*Servo meuServo;*”. Isto significa que um objeto chamado “*meuServo*” foi criado, e que este está

atrelado a um único servomotor, cujo pino é definido pelo comando “*meuServo.attach()*”. Caso deseja-se instalar um novo servomotor na placa Arduino, deverá ser referenciado um novo objeto da biblioteca “*Servo.h*” e, atrelado a este, um novo pino com suporte PWMa ser escrito entre parênteses pelo comando “*attach*” neste objeto.

Figura 33 – Programa de exemplo que realiza a varredura do eixo de um servomotor de 0 até 180 graus, e então de 180 até 0 graus ciclicamente

```
// Programa 14 - Controle de Servo-Motores

#include <Servo.h> //Inclui biblioteca para dar suporte a comandos avançados
// No caso, a biblioteca Servo.h contém comandos específicos para servos.
Servo meuServo; // Cria um objeto que referencia a biblioteca Servo.
int angulo; // Variável auxiliar que controlará o ângulo do servo.
const int IN2 = 8; // Define o valor da constante inteira IN2.

void setup() {
  meuServo.attach(IN2); //Instrução necessária para que o objeto "meuServo"
  // saiba que o pino IN2 controlará a posição do servo-motor.
}
void loop() {
  for (angulo = 0; angulo < 180; angulo++)
    // Este laço "for" fará o motor girar seu eixo de 0 até 180 graus
    {
      meuServo.write(angulo); // Instrução para forçar o servo a
      // posicionar seu eixo em uma posição fixa, em graus.
      delay(15); // Tempo de espera para o servo fixar sua posição.
    }
  delay (1000);
  for (angulo = 180; angulo > 0; angulo--)
  {
    meuServo.write(angulo); // Desta vez, o servo irá girar seu
    delay(15); // eixo de 180 graus até a posição de 0 graus.
  }
  delay (1000);
}
```

Fonte: Autoria Própria

Pelo programa de exemplo da Figura 33, além do próprio uso de bibliotecas no programa que devem ser incluídas pelo parâmetro “*#include*”, outro comando básico para controlar a posição do eixo do servomotor é o comando “*meuServo.write()*”, onde o programador pode diretamente incluir entre os parênteses deste comando o valor da posição que deseja mover o servo. Este valor, por definição do próprio código-fonte da biblioteca, deve ser dado em graus e só pode incluir valores inteiros. Para a grande maioria de servomotores de pequeno porte, a faixa de valores aceitável para o comando “*write()*” é de 0 até 180.

Ainda que apenas os comandos “*attach*” e “*write*” sejam usados, o educador deve ressaltar a importância de poder controlar a posição do eixo em servomotores em diversas

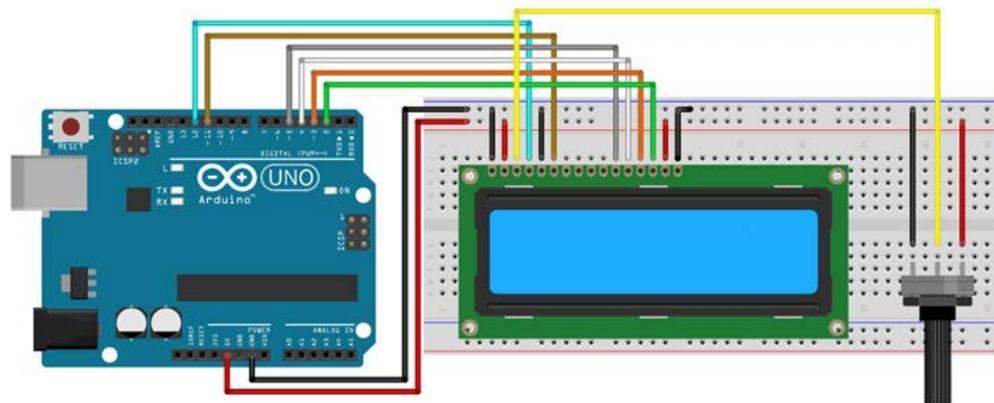
aplicações, bem como quais aplicações podem ser realizadas tendo conhecimento destes comandos. Alguns exemplos incluem relógios de ponteiros, controle de abertura de persianas e abertura e fechamento automatizado de portas.

4.7 Utilização de Displays de Cristal Líquido

Devido às limitações de displays do tipo 7 segmentos explanados na seção 3.8.1, este curso tratará, em sua parte prática, apenas da aplicação de displays de LCD, mais especificamente displays do tipo 16x2 (16 colunas de caracteres x 2 linhas de informações).

A Figura 34 ilustra a disposição das conexões entre o display LCD e a placa Arduino. Por medida de segurança e preservação da vida útil do display, recomenda-se que este seja alimentado por uma tensão de 3,3V ao invés de 5V, visto que displays projetados para funcionar em 5V apenas terão o brilho de suas telas afetados, ao passo que displays projetados para funcionar com 3,3V podem ter seus componentes discretos danificados caso sejam submetidos a tensões mais elevadas. Tal fato se dá devido à tecnologia implementada no display (TTL ou CMOS) com suas respectivas faixas de tensão de trabalho (DIAGO; AMARAL, 2011).

Figura 34 – Exemplo de circuito para conexão do display LCD com a placa Arduino, incluindo potenciômetro para controle de contraste do texto exibido



Fonte: Autoria Própria

Recomenda-se que o educador tenha montado o circuito da Figura 34 em pelo menos uma placa, de forma a preservá-la e utilizá-la como uma “placa de referência” caso alguns grupos de estudantes não consigam montar este circuito, encorajando os estudantes a testarem a exibição de caracteres por meio do monitor serial como discutido na seção 3.8.2. Tal

prevenção economiza tempo de laboratório e de solução de problemas encontrados durante a programação envolvendo displays pelos alunos, otimizando assim o tempo de desenvolvimento do curso.

Para utilizar os comandos de escrita no display LCD, similarmente como no controle de servomotores, deve-se primeiramente incluir a biblioteca de controle de displays LCD. No caso, esta biblioteca chama-se “*LiquidCrystal.h*” e também se encontra pré-instalada no software Arduino. Depois, cria-se um objeto desta biblioteca, seguindo a sintaxe descrita na Figura 35.

Figura 35 – Programa de inicialização de um display LCD 16x2

```
// Programa 15 - Controle Simples de um LCD

#include <LiquidCrystal.h>

//Define os pinos que serão utilizados para ligação ao display
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
//É gerado uma instância (objeto) da biblioteca LiquidCrystal.
//No caso, foram definidos, respectivamente, em quais pinos do Arduino
//serão conectados os pinos RS, E, D4, D5, D6 e D7 do LCD

void setup()
{
  //Define o número de colunas e linhas do LCD
  lcd.begin(16, 2); //Inicializa o LCD, contendo 16 colunas e 2 linhas
  lcd.cursor(); //Mostra o cursor permanentemente no display.
}
```

Fonte: Autoria Própria

No programa de exemplo da Figura 35, contendo apenas os comandos necessários para a definição do objeto da biblioteca “*LiquidCrystal.h*” e inicialização do display LCD, destacam-se dois pontos:

- O objeto referenciado da biblioteca “*LiquidCrystal.h*” deve possuir argumentos inseridos entre parênteses de acordo com o modo de operação desejado do display, contendo os pinos certos de conexão entre a placa Arduino e determinados pinos do display. Para este caso em específico, colocou-se o display para operar em modo 4 bits, com os números dos argumentos referentes ao circuito ilustrado pela Figura 34;
- O comando “*lcd.begin()*” (lembrando que “*lcd*” é o objeto da biblioteca “*LiquidCrystal.h*”) é usado para inicializar o LCD. Entre parênteses, devem

estar inclusos, respectivamente, o número de colunas e o número de linhas do display, separados entre si por vírgula.

Após a inicialização do display LCD, alguns comandos podem ser usados para controlar o display. Assim como no caso do comando “*lcd.begin()*”, todos os comandos abaixo precisam ser precedidos do objeto referenciado pela biblioteca, havendo um ponto de separação entre o objeto e o comando desejado.

- *clear()*: Usado para “limpar” o display, apagando todos os caracteres escritos anteriormente;
- *print()*: Similarmente quando se usa o comando “*Serial.print()*”, este comando escreve informações, sejam textos pré-definidos entre aspas duplas ou valores de variáveis do programa, em uma mesma linha do display LCD. Caso o texto a ser escrito for maior que o espaço restante da linha onde o cursor está localizado, os caracteres excedentes não serão exibidos no display.
- *setCursor(X, Y)*: Move o cursor para uma dada posição. A posição inicial (primeira coluna da primeira linha) de qualquer display LCD é a posição (0,0), também conhecida como posição *home*. O valor de X dita a coluna em que o cursor será posicionado, enquanto o valor de Y dita a linha em que o cursor será posicionado. Logo, a posição (7,1) significa que o cursor estará localizado na 8ª coluna da 2ª linha do display.

Para mais comandos envolvendo displays LCD, (MONK, 2012) inclui explicações de comandos mais avançados, tais como fazer o cursor piscar, desligar e ligar o display para economia de energia e deslocamento do texto escrito para a esquerda e para a direita.

Após os estudantes terem compreendido o uso básico do display, o educador deve passar aos alunos um exercício mais elaborado, buscando exercícios de tópicos anteriores cujo retorno de informação se dá no display LCD. Um exemplo de tal programa é o monitoramento da luz ambiente através de um LDR, com o display LCD mostrando o texto “dia” ou “noite” de acordo com o valor lido do LDR. Pode-se também optar pelo monitoramento do valor discreto obtido diretamente no display para acompanhamento juntamente com o monitor serial.

Neste tópico, o mais importante para o educador se preocupar é a inclusão de “delays” (atrasos) entre comandos de escrita no display de forma a evitar erros do texto mostrado no

display. Mau contato nas conexões de fios e jumpers que realizam o interfaceamento entre o display e o microcontrolador da placa Arduino também podem causar falhas nas operações de escrita no display, portanto os estudantes devem ser instruídos para tomarem cuidado durante a montagem de componentes nas proximidades do componente.

5 DESENVOLVIMENTO DA PLATAFORMA

Tendo-se estabelecido todo o conteúdo teórico e prático do curso, o objetivo de se montar uma plataforma educacional de robótica é a de facilitar o processo de aprendizado por parte dos alunos, bem como incentivá-los a estudar o conteúdo proposto para aplicá-lo em exercícios de programação elaborados pelo educador.

Visando aproximar a aplicação da plataforma a um produto que seja familiar a todos os alunos e de forma a fazer analogias com tecnologias recentemente implementadas em produtos, será utilizado como base as tecnologias embarcadas em carros recentemente disponibilizados para o mercado no Brasil. Uma imagem da plataforma montada está ilustrada pela Figura 36.

Ainda que seja inviável a integração de todas as tecnologias de um carro em uma plataforma didática, as características e opções de maior relevância em carros atualmente vendidos podem ser replicadas sem grandes problemas em um protótipo educacional cujas funções sejam controladas pelo microcontrolador da placa Arduino. Alguns exemplos dessas funções são: controle de velocidade constante do carro, computador de bordo com monitoramento dos sensores, controle de brilho dos faróis, sensor de estacionamento, acionamento automático de faróis em condições noturnas, dentre outros.

Figura 36 – Plataforma educacional de robótica baseado em um carro contemporâneo



Fonte: Autoria Própria

No caso da plataforma desenvolvida, todas as funcionalidades descritas no parágrafo anterior foram implementadas e instaladas dentro de um carro de brinquedo adaptado para comportar todos os sensores, atuadores e displays da plataforma. A escolha de se usar um carro de brinquedo como base para a plataforma barateia o custo de montagem do projeto

frente a kits e chassis de “carros inteligentes” vendidos em grandes lojas de componentes eletrônicos, bem como cria-se uma réplica mais fidedigna do produto no qual a plataforma foi baseada a fim de interessar os alunos do curso.

5.1 Desenvolvimento do Circuito

Durante a fase de desenvolvimento do projeto, os seguintes pontos foram observados:

- Implementação de, no mínimo, três tipos de dispositivos de saída diferentes;
- Uso de, no mínimo, dois tipos de sensores diferentes;
- Uso de componentes e sensores amplamente disponíveis no mercado nacional;
- Montagem modular através de conectores e soquetes especiais;
- Controle de certas funções da plataforma à distância;
- Flexibilidade da plataforma;
- Desenvolvimento da própria placa Arduino com microcontrolador Atmel.

Os dois primeiros pontos são os mais essenciais para a plataforma, visto que a mesma tem como objetivo familiarizar os alunos com diferentes tipos de dispositivos eletrônicos, tanto digitais quanto analógicos.

A disponibilidade dos componentes usados no mercado nacional também foi fator importante para possibilitar ao educador comprar em diferentes fornecedores, sejam estes locais ou com possibilidade de compra pela internet. Tal facilidade não só encurta o tempo de espera para o recebimento de componentes como também possibilita uma maior pesquisa de preço dos mesmos para adquirir os componentes no menor preço possível.

A ideia da montagem modular se deu como uma boa prática observada durante o estágio realizado na indústria, onde o cabeamento entre atuadores, sensores e sistema controlador central nunca são soldados entre si de maneira definitiva, facilitando assim o processo de manutenção e troca destes componentes.

O controle à distância foi projetado para que os estudantes percebam que o mesmo microcontrolador usado no curso também pode ser usado para tipos de controles mais complexos, fomentando o interesse dos mesmos no autodidatismo e na curiosidade por novas tecnologias compatíveis com o microcontrolador.

A flexibilidade da plataforma possibilita que a mesma possa ser utilizada não só no ensino de robótica, mas também em exercícios didáticos de Matemática e Física, como durante a abordagem de temas como movimento retilíneo uniforme, movimento retilíneo uniformemente variável, plano inclinado, força de atrito, entre outros.

Por fim, o desenvolvimento da própria placa Arduino foi escolhido para mostrar aos alunos que é possível confeccionar placas que melhor se adequem à aplicação escolhida e ao espaço disponível para fixação da placa. Além disso, optar por desenvolver a placa completa, incluindo reguladores de tensão e osciladores a cristal, ao invés de fabricar um “*shield*” de uma placa Arduino colabora para baratear ainda mais a plataforma, necessitando apenas de uma placa comercial para programação de todos os microcontroladores ante a compra de placas comerciais Arduino para cada plataforma a ser utilizada no curso.

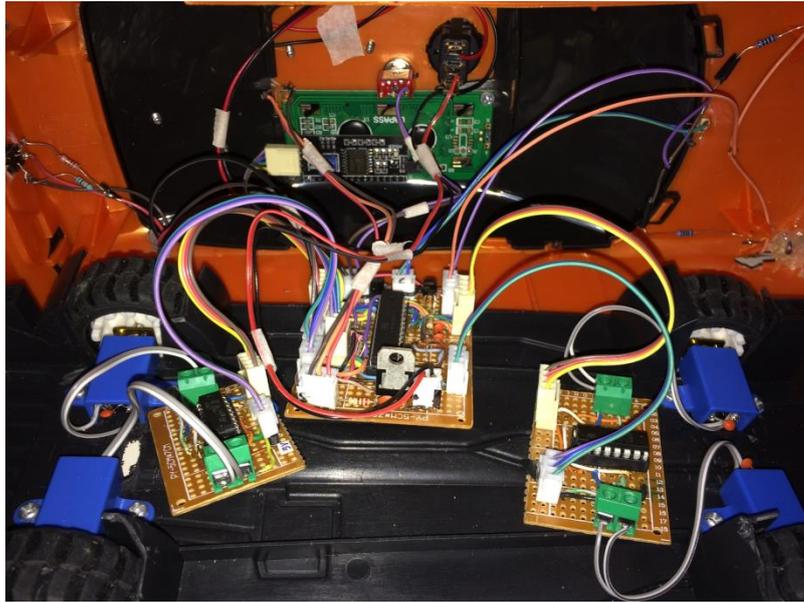
O desenho do circuito da plataforma de robótica pode ser conferido no Apêndice A deste trabalho, enquanto a lista de materiais para o mesmo se encontra no Apêndice B. É importante ressaltar que podem ser usados componentes similares aos descritos para os casos de transistores, sensores e atuadores conforme o enfoque do curso e o resultado final esperado. No entanto, o responsável pela montagem deve ficar atento às mudanças necessárias nos valores de resistores e capacitores que podem derivar do uso de componentes diferentes do especificado de forma a preservar o funcionamento original da plataforma.

5.2 Soldagem dos Componentes

Antes de soldar os componentes à placa, deve-se ter cuidado especial quanto ao arranjo dos componentes, visto que a placa estará confinada dentro de um carro de brinquedo (aproximadamente 100x150mm de área útil para as placas) e, portanto, deve-se otimizar o uso de espaço disponível. Para tal, usou-se uma placa de prototipagem de dimensões 60x80mm para a montagem do microcontrolador e os soquetes de seus sensores e atuadores, deixando a parte de acionamento dos motores isolados ao utilizar placas 60x40mm, cada uma contendo um CI L293D e os conectores para cada par de motores.

A Figura 37 mostra a disposição das placas de circuito impresso dispostas dentro da plataforma de robótica. Foram utilizados conectores especiais para a conexão de dispositivos eletrônicos de forma a otimizar eventuais manutenções e substituições na plataforma, evitando a dessoldagem de sensores e motores defeituosos.

Figura 37 – Placas de circuito integrado montadas na plataforma de robótica

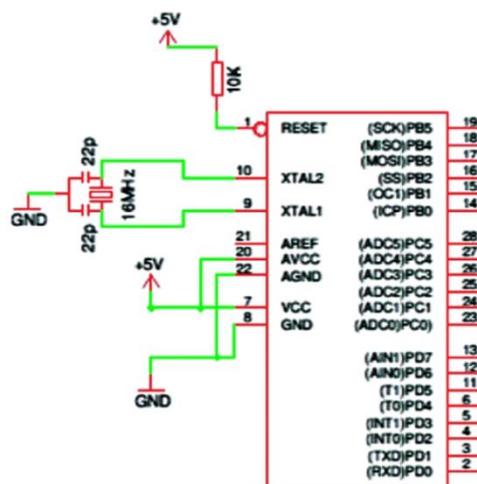


Fonte: Autoria Própria

A placa maior, por ser encarregada de fornecer a alimentação para o microcontrolador, sensores, atuadores e display, deve possuir um regulador de tensão para adequar a tensão de 9V oriunda da bateria, usada para a alimentação dos motores, para níveis aceitáveis de alimentação dos demais componentes eletrônicos.

A Figura 38 ilustra a montagem da versão “*stand-alone*” do microcontrolador na placa de circuito impresso, visando economizar espaço físico frente a utilizar a placa comercial Arduino Uno R3.

Figura 38 – Circuito da versão básica “*stand-alone*” do Arduino



Fonte: (OPENELECTRONICS, 2016)

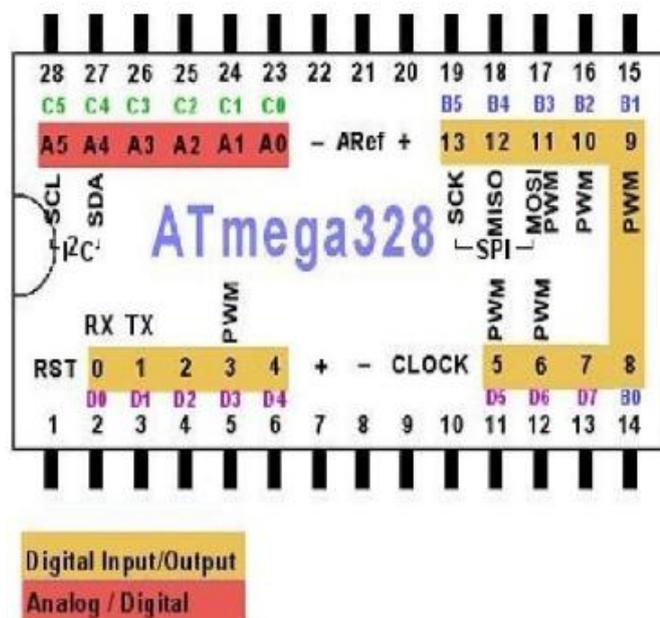
Após montar o circuito da Figura 38 (além do regulador de tensão, cujo circuito já está ilustrado no Apêndice A) e soldá-lo na placa principal, deve-se testar o mesmo para certificar que o nível adequado de tensão está chegando ao microcontrolador, bem como avaliar a forma de onda do oscilador a cristal para verificar se a mesma está adequada de acordo com os parâmetros aceitáveis do microcontrolador.

A partir daí, montam-se os circuitos de cada sensor, atuador e display, um de cada vez, preferencialmente utilizando um programa de teste para analisar se os mesmos estão operando adequadamente. Ainda que tal processo seja mais longo que a montagem completa, trata-se de uma boa prática para evitar problemas devido a mau contato dos fios de conexão e facilitar o processo de eventuais correções.

5.3 Programação da Plataforma

Tendo-se montado todo o circuito da plataforma educacional, o programador deverá, associar os dispositivos conectados ao microcontrolador de maneira correta de acordo com o mapeamento dos pinos do microcontrolador, como ilustrado na Figura 39.

Figura 39 – Mapeamento dos pino do microcontrolador Atmel ATMEGA328p vistos pela IDE Arduino



Fonte: (MADEKAR et al., 2016)

Isto garantirá que o programador referencie corretamente as portas atreladas aos sensores e atuadores da plataforma, exatamente da mesma forma que se estes dispositivos fossem conectados a uma placa Arduino comercial.

Outro ponto a ser ressaltado é que, como a plataforma educacional não possui circuitaria que possibilite a programação via ICSP (modo serial), deve-se ter de prontidão uma placa comercial Arduino para se programar o microcontrolador usado na plataforma. Evidentemente, tal placa comercial deve possibilitar a troca de microcontroladores, ao invés de ter o mesmo soldado na placa.

Além disso, o microcontrolador a ser programado já deve possuir o seu *bootloader* instalado para ser capaz de reconhecer comandos de linguagem Arduino sem necessitar de hardware externo (ARDUINO, 2016b). Diversas lojas de componentes eletrônicos já vendem o microcontrolador Atmel ATMEGA328p com o *bootloader* pré-instalado, o que economiza tempo de programação do microcontrolador, no entanto, por se tratar de uma plataforma de código aberto, pode-se encontrar o programa do *bootloader* Arduino em fóruns e demais fontes juntamente com o circuito e procedimento para instalação do mesmo.

5.4 Testes e Considerações

Os testes isolados de cada sensor e atuador se deram sem maiores problemas. A sequência de testes se deu primeiramente no acionamento dos *drivers* dos motores para garantir que estes tivessem torque suficiente para deslocar o carro, além da fixação mecânica dos mesmos por meio de braçadeiras de nylon para minimizar vibrações indesejadas. Em seguida, implementou-se o controle de velocidade e sentido de deslocamento por meio de um controle remoto convencional, a fim de testar o tempo de resposta dos comandos. Então, testou-se o display LCD para certificar que a comunicação I2C estava sendo realizada de maneira adequada pelo microcontrolador, testando também a leitura de variáveis do microcontrolador e a escrita destes valores de maneira visível no mostrador digital.

Após certificar do funcionamento destes sistemas, foram elaborados programas separados para o teste de cada sensor (LDR e CNY70) e atuador (LED e buzzer) para garantir suas funcionalidades, até finalmente implementar um programa que utilize todos os dispositivos empregados na plataforma, implementando também algumas otimizações de código para os sensores usados de forma a terem seus funcionamentos adequados para o objetivo inicial da plataforma. O código-fonte deste programa final encontra-se no Apêndice C deste trabalho.

Um ponto a ser levantado ao longo dos testes é que, devido à biblioteca utilizada para interpretar as informações recebidas do receptor de infravermelho, um dos três timers (temporizadores internos do microcontrolador) disponíveis do ATMEGA 328p acabam sendo reservados para tal função, o que compromete a funcionalidade de PWM em dois dos pinos do microcontrolador regidos por tal timer (ATMEL, 2016). Isto significa que, para o controle PWM dos faróis e do buzzer da plataforma, é necessário a exclusão da biblioteca controladora do receptor de infravermelho da plataforma, eliminando assim o uso de controle remoto nestas aplicações.

Uma possível proposta de melhoria seria a implementação de um módulo wireless, como o ESP8266, ao invés de um receptor infravermelho, para assim possibilitar o acionamento e o monitoramento remoto da plataforma por meio de computadores e dispositivos móveis. Ainda que tal aplicação possa ser implementada no Arduino por meio de bibliotecas específicas, a implementação destes módulos é muito mais complexa e de difícil absorção para estudantes da faixa etária a que o curso se destina, e, portanto, para ser aplicado, deverá haver uma reestruturação do curso e das bases de conhecimento necessárias.

Outra proposta de melhoria que pode ser levada em conta seria a implementação de comunicação ICSP na plataforma, onde possibilitaria a programação da placa diretamente via interface USB ao invés de depender de placa comercial Arduino para realizar a programação, além da conveniência de não precisar remover e inserir o microcontrolador na placa da plataforma educacional toda vez que alterações de *software* forem necessárias.

Há placas comerciais que realizam a ponte de comunicação entre microcontroladores e computadores, sendo a comunicação para com microcontroladores via padrão ICSP e a comunicação com computadores via USB de diversos formatos (micro USB e USB Tipo-B estão entre os mais comuns). Tais placas podem também ser soldadas na mesma placa principal que a plataforma educacional, proporcionando assim maior comodidade durante o desenvolvimento de programas para a mesma.

6 APLICAÇÃO

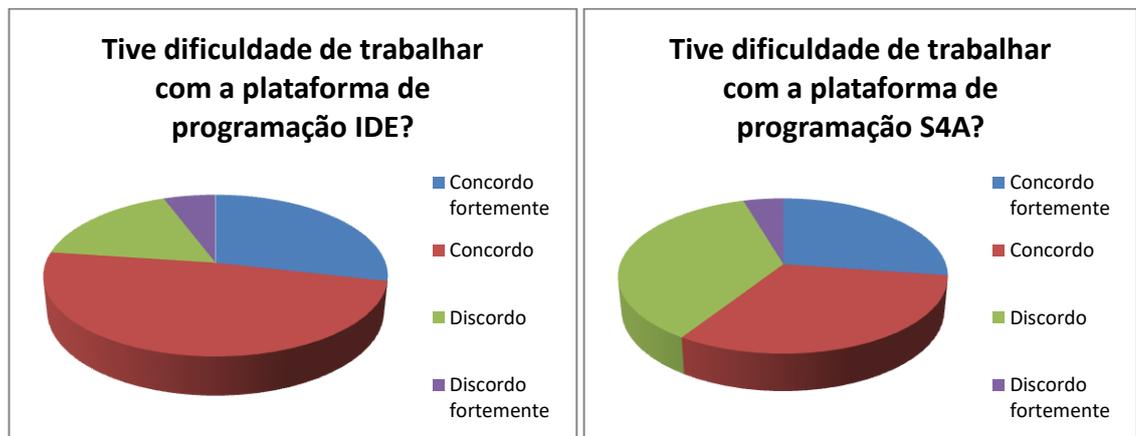
Parte do material teórico e prático descrito neste trabalho foi aplicado na segunda edição da Exposição Regional de Engenharia e Ciências, EXPRECI, realizado no campus de Guaratinguetá da UNESP, na forma de uma oficina de robótica de 2 horas voltada a estudantes do ensino médio que, em sua maioria, não haviam tido qualquer contato anterior com Arduino ou plataformas de robótica similares.

Devido à restrição no tempo da oficina, apenas os conceitos mais cruciais foram passados aos alunos, focando mais nas atividades práticas e na solução de dúvidas durante o desenvolvimento de programas usando o software de programação do microcontrolador.

Houve ao todo três turmas para as quais o mesmo objetivo da oficina foi passado, sendo que em duas delas foi utilizado o Arduino IDE como software de programação, enquanto a turma restante utilizou o S4A como software principal. No total, 61 alunos estavam presentes no dia da oficina de robótica.

Por meio dos questionários preenchidos pelos alunos presentes no início e no término da oficina, cujos resultados podem ser vistos na Figura 40 e na Figura 41, foram retirados importantes informações sobre o que os alunos esperavam aprender e o que os motivaram a estarem na oficina e, no final, se suas expectativas foram alcançadas e se a oficina conseguiu motivar os alunos a aprenderem mais sobre Arduino e conceitos de Eletrônica e Programação.

Figura 40 – Comparação entre dificuldade dos softwares da oficina de robótica de acordo com os alunos



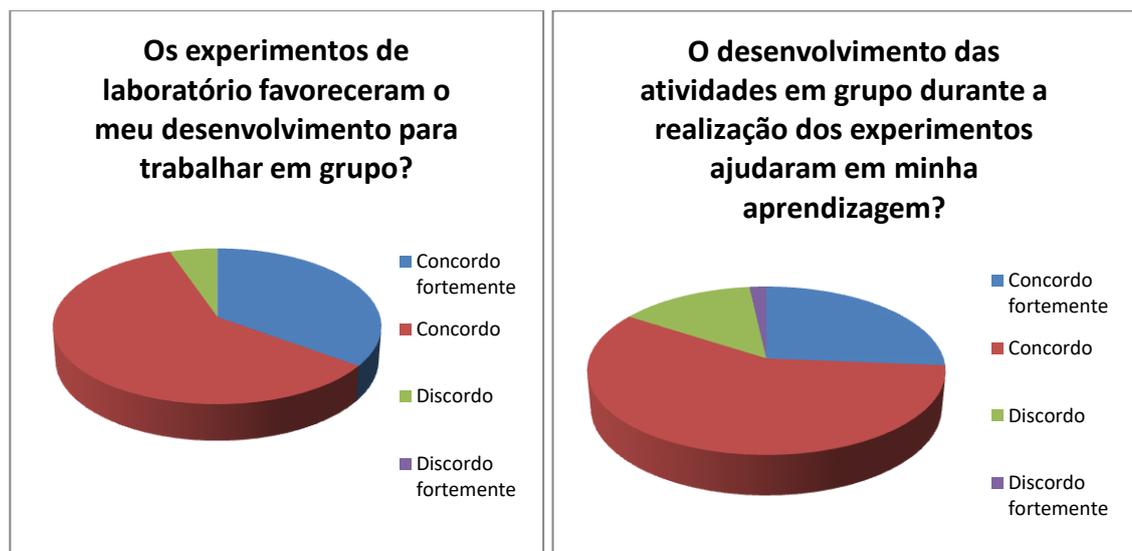
Fonte: Autoria Própria

Como esperado, a interface S4A cativou mais alunos iniciantes que a interface IDE, o que reforça o ponto citado em 2.2.2, onde o S4A pode servir como um bom primeiro contato

dos alunos a conceitos de eletrônica e lógica de programação. No entanto, ao comparar com alunos que já possuíam algum conhecimento de ambientes de programação em suas escolas, notou-se que a interface IDE obteve resultados até melhores no que se refere à dificuldade de aprendizado durante o curso, mesmo comparado com alunos iniciantes manipulando o S4A.

Analisando o desenvolvimento do trabalho em grupo, a grande maioria dos alunos concordaram que o mesmo se fazia necessário para resolver os exercícios práticos propostos na oficina, independente do software utilizado. Além disso, o trabalho em grupo também ajudava os alunos a compreenderem melhor os conceitos de lógica de programação, montagem dos circuitos elétricos e na solução de problemas. Tais aspectos favorecem a evolução dos estudantes para o preparo ao exigente mercado de trabalho atual, bem como se tornam facilitadores para o aprendizado de conceitos mais complexos.

Figura 41 – Resultados referentes à competência de trabalho em equipe e aprendizagem em equipe



Fonte: Aatoria Própria

As atividades da oficina envolviam uma breve introdução sobre robótica, projetos possíveis que podem ser feitos usando o mesmo microcontrolador do Arduino, uma explicação rápida sobre os componentes eletrônicos e a explicação dos comandos da IDE utilizada em cada turma juntamente com exercícios para solidificar os conhecimentos pelos alunos.

Os estudantes tiveram suas dúvidas prontamente resolvidas pelos assistentes e voluntários presentes, e, no geral, houve um bom aproveitamento da oficina, ainda que o tempo não tenha sido suficiente para que os alunos pudessem desenvolver seus próprios

projetos com os materiais fornecidos. Houve também sinais animadores oriundos das respostas dos questionários, demonstrando interesse futuro em demais oficinas similares e levando a crer que o aprendizado focado nas atividades práticas e em exercícios simples de alteração de lógica é fundamental para a concretização do conhecimento em diversas áreas, tanto acadêmicas quanto sociais.

7 CONCLUSÃO

Ao longo do desenvolvimento deste trabalho, teve-se em mente a construção de uma plataforma de robótica que utilizasse o microcontrolador de maneira extensiva, visando um curso de no mínimo 40 horas sobre conhecimentos em engenharia, eletrônica e programação, de forma que os alunos no final do curso pudessem construir e programar a plataforma por conta própria, além de instigar a criatividade dos mesmos para possíveis aplicações com os conhecimentos adquiridos.

Valeressaltar que a plataforma também conta, como ponto-chave na fase de projeto, com aplicações voltadas para Matemática e Física. Aplicações práticas sobre movimento retilíneo uniforme, uniformemente variável e até movimento circular uniforme são possíveis graças ao controle independente da velocidade de cada um dos quatro motores, além de experimentos envolvendo frequência de ondas sonoras (como experimento de efeito Doppler) e óptica (projeção de sombras em objetos conforme a distância do ponto luminoso e intensidade luminosa da fonte de luz).

Com recentes movimentos políticos envolvendo reforma do sistema educacional brasileiro, visando a flexibilização do currículo de estudantes do ensino médio, faz-se necessário que o ensino de conceitos técnicos envolvendo eletrônica e informática seja transmitido de maneira eficaz e abrangente pelo educador, que além de prever dificuldades dos alunos nos diversos tópicos que ministrará, precisa de um material de apoio completo e de qualidade para basear suas aulas.

O ensino de robótica muitas vezes é, erroneamente, visto apenas como uma ferramenta exclusiva para o desenvolvimento de conhecimentos na área de exatas. No entanto, nas mãos de um educador experiente e com o auxílio de plataformas microcontroladas especiais, diferentes áreas do conhecimento podem ser beneficiadas com tal aprendizado, tais como Música, Artes, Linguagens e Códigos, Ciências, dentre outras. Sabendo-se ensinar à nova geração de estudantes como utilizar conceitos de lógica de maneira correta e fomentando a busca constante por conhecimento em novas tendências tecnológicas, será possível o enriquecimento cultural da nação ao motivar estudantes a serem profissionais mais competitivos no futuro.

REFERÊNCIAS

- ADAFRUIT.**Arduino Uno R3**. 2016. Disponível em: <<https://www.adafruit.com/product/50>>. Acesso em: 10 jul. 2016.
- ARDUINO.**Arduino**. 2016a. Disponível em: <<https://www.arduino.cc/>>. Acesso em: 03 jul. 2016.
- ARDUINO.**Bootloader Development**. 2016b. Disponível em: <<https://www.arduino.cc/en/Hacking/Bootloader?from=Tutorial.Bootloader>>. Acesso em: 17 set. 2016.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 5410**: Instalações Elétricas de Baixa Tensão. Rio de Janeiro, 2004.
- ATMEL.**ATMEL 8-Bit Microcontroller with 4/8/16/32Kbytes In-System Programmable Flash**: Datasheet. 2015. Disponível em: <http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf>. Acesso em: 05 ago. 2016.
- BANZI, M.; SHILOH, M. **Getting Started With Arduino: The Open Source Electronics Prototyping Platform**. 3. ed. Sebastopol: Maker Media Inc., 2014.
- BASTOS FILHO, T. F. **Apostila de Eletrônica Básica II**. 2004. Disponível em: <<http://www.iwatufes.com/Basica2/apostila.pdf>>. Acesso em: 05 ago. 2016.
- BOCK, T. Turning Points in Construction. In: INTERNATIONAL SYMPOSIUM ON AUTOMATION AND ROBOTICS IN CONSTRUCTION, 26., 2009, Austin. **Proceedings...**[s.n.], 2009. p.1. Disponível em: <http://www.iaarc.org/publications/fulltext/Turning_Points_in_Construction.pdf>. Acesso em: 2 jul. 2016.
- BOYLESTAD, R. L. **Introdução à Análise de Circuitos**. 12. ed. São Paulo: Pearson Prentice Hall, 2012.
- CONFEDERAÇÃO NACIONAL DA INDÚSTRIA. **Recursos Humanos para Inovação: engenheiros e tecnólogos**. Brasília: CNI, 2014. 43 p.
- CONSELHO FEDERAL DE ENGENHARIA E AGRONOMIA. **Profissionais**: Por Crea. Disponível em: <<http://ws.confed.org.br:8080/EstatisticaSic/ModEstatistica/Pesquisa.jsp?vw=TotalCrea>>. Acesso em: 2 jul. 2016.
- DIAGO, R.; AMARAL, V. M. **Eletrônica: Eletrônica Digital**. São Paulo: Fundação Padre Anchieta, 2011.
- ELECTROSOME.**Interfacing LCD with Atmega32 Microcontroller Using Atmel Studio**.2013. Disponível em: <<https://electrosome.com/interfacing-lcd-atmega32-microcontroller-atmel-studio/>>. Acesso em: 05 ago. 2016.

ELENCO. **Basic Electronic Experiments**. 2012. Disponível em: <http://www.elenco.com/admin_data/pdffiles/PK-101_REV-G.pdf>. Acesso em: 30 ago. 2016.

FAIRCHILD SEMICONDUCTOR.**AN-6003**: “Shoot-through” in Synchronous Buck Converters. 2003. Disponível em: <<https://www.fairchildsemi.com/application-notes/AN/AN-6003.pdf>>. Acesso em: 05 ago. 2016.

FARRELL, J. **Programming Logic and Design**.7.ed.Boston: Cengage Learning, 2012.

FEITOSA, J. G (Org).**Manual Didático Pedagógico**. Curitiba: ZOOM Editora, 2013.

FELGUEIRAS, C. A. **Organização Estruturada de Computador**: Aula 5. 2003. Disponível em: <http://www.dpi.inpe.br/~carlos/Academicos/Cursos/ArqComp/aula_5bn1.html>. Acesso em: 30 jul. 2016.

FERREIRA, A. B. de H.**Dicionário da língua portuguesa**.5. ed. Curitiba: Positivo, 2010. 2222 p.

FISCHERTECHNIK.**Building Blocks For Life**. 2014. Disponível em: <<http://www.fischertechnik.de/PortalData/1/Resources/downloads/documents/prospectuses/HK-engl.pdf>>. Acesso em: 03 jul. 2016.

FLOYD, T. L. **Sistemas Digitais: Fundamentos e Aplicações**. 9. ed. Porto Alegre: Bookman, 2007.

GASPAR, A. **Compreendendo a Física: Ondas, Óptica e Termodinâmica**. 2. ed. São Paulo: Ática, 2013. 456p.

GOMES, M. C. Reciclagem Cibernética e Inclusão Digital: Uma Experiência em Informática na Educação. In: LAGO, C. (Org.). **Reescrevendo a Educação**. Chapecó: Sinproeste, 2007.

GONÇALVES, P. C. **Protótipo de um Robô Móvel de Baixo Custo para Uso Educacional**. 2007. 86 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Estadual de Maringá, Maringá, 2007.

HAMBLEY, A. R. **Electrical Engineering: Principles and Applications**. 5. ed. [s.l]: Pearson, 2010. 912 p.

HAMLIN. **Application Note AN108A**: Inductive Load Arc Suppression. 2013. Disponível em: <<http://www.hamlin.com/specSheets/AN108A-Inductive-Load-Arc-Suppression.pdf>>. Acesso em: 09 set. 2016.

HITACHI.**HD44780U (LCD-II)**: Dot Matrix Liquid Crystal Display Controller/Driver. 1998. Disponível me: <<https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>>. Acesso em: 26 ago. 2016.

HORTA, H. **Education in Brazil: Access, Quality and STEM**. 2013. Disponível em: <<http://www.acola.org.au/PDF/SAF02Consultants/Consultant%20Report%20-%20Brazil.pdf>>. Acesso em: 24 jul. 2016.

HUGHES, A. **Electric Motors and Drives: Fundamentals, Types and Applications**. 3. ed. Burlington: Newnes, 2006. 384 p.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **Censo Demográfico 2010**. Disponível em: <<http://www.ibge.gov.br/home/presidencia/noticias/imprensa/ppts/00000011363712202012375418902674.pdf>>. Acesso em: 02 jul. 2016.

INTRODUÇÃO à Eletrônica. Rio de Janeiro: Editora Lutécia, 1985. Disponível em: <<http://www.apostilastecnicas.com/Outros/Eletronica/Curso%20De%20Eletronica%20Ilustrado%5B1%5D.pdf>>. Acesso em: 24 jul. 2016.

KNUDSEN, J. B. **The Unofficial Guide to LEGO MINDSTORMS Robots**. Sebastopol: O'Reilly, 1999. 249 p.

KUSHNER, D. **The Making of Arduino: How five friends engineered a small circuit board that's taking the DIY world by storm**. 2011. Disponível em: <<http://utmechatronics.ir/wp-content/uploads/The-Making-of-Arduino-IEEE-Spectrum.pdf>>. Acesso em: 10 jul. 2016.

LEGO. **History of LEGO Robotics**. 2016. Disponível em: <<http://www.lego.com/en-us/mindstorms/history>>. Acesso em: 09 jul. 2016.

LEGO. **NXT User Guide**. 2006. Disponível em: <http://cache.lego.com/downloads/education/9797_LME_UserGuide_US_low.pdf>. Acesso em: 03 jul. 2016.

MACIENTE, A. N.; ARAÚJO, T. C. Demanda por Engenheiros e Profissionais Afins no Mercado de Trabalho Formal. **Radar: Tecnologia, Produção e Comércio Exterior**, Brasília, n. 12, p.43-54, fev. 2011.

MADEKAR, K. et al. Automatic Mini CNC Machine for PCB Drawing and Drilling. **International Research Journal of Engineering and Technology**, v.3, n.2, p.1106-1110, fev. 2016.

MAINI, A. K. **Digital Electronics: Principles, Devices and Applications**. Chichester: Wiley, 2007.

MALVINO, A. P. **Eletrônica**, v. 1. 4. ed. São Paulo: Pearson Education, 1996.

MATARIC, M.; KOENIG, N.; FEIL-SEIFER, D. Materials for Enabling Hands-On Robotics and STEM Education. In: **AAAI SPRING SYMPOSIUM ON ROBOTS AND ROBOT VENUES**, 2007, Stanford. **Proceedings...** Stanford: Resources for AI Education, 2007, p. 1-4.

MCROBERTS, M. **Arduino Básico**. São Paulo: Novatec Editora, 2011.

MELO, F. E. N. et al. Do Scratch ao Arduino: Uma Proposta para o Ensino Introdutório de Programação para Cursos Superiores de Tecnologia. In: CONGRESSO BRASILEIRO DE EDUCAÇÃO EM ENGENHARIA, 39., 2011, Blumenau. **Anais...**Blumenau: FURB, 2011.

MICROCHIP.AN774: Asynchronous Communications with the PICmicro USART. 2003. Disponível em: <<http://ww1.microchip.com/downloads/en/AppNotes/00774a.pdf>>. Acesso em: 07 set. 2016.

Massachusetts Institute of Technology.**About Scratch**.2016. Disponível em: <<https://scratch.mit.edu/about>>. Acesso em: 12 jul. 2016.

MONK, S. **30 Arduino Projects for the Evil Genius**. [s.l]: McGraw-Hill, 2010.

MONK, S. **Programming Arduino: Getting Started With Sketches**. [s.l]: McGraw-Hill, 2012.

MORAES, P. S. **Lógica de Programação**. 2000. Disponível em: <<http://www.inf.ufsc.br/~vania/teaching/ine5231/Logica.pdf>>. Acesso em: 25 jul. 2016.

MOURA, C. S. **Física para o Ensino Médio: Gravitação, Eletromagnetismo e Física Moderna**. Porto Alegre: EDIPUCRS, 2011. 284 p.

MURRAY, R. M.; LI, Z.; SASTRY, S. S. **A Mathematical Introduction to Robotic Manipulation**. Berkeley: Crc Press, 1994. 480 p.

OLIVEIRA, G. S. et al. Análise Comparativa entre os Kits Proprietários e de Robótica Livre tendo como Parâmetro o seu Custo e Características de Ensino. In: ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, 31., 2011, Belo Horizonte. **Anais...**Belo Horizonte: ENEGEP, 2011. p. 1-9.

OPENELECTRONICS.**Arduino ISP (In System Programming) and Stand-Alone Circuits**.2016.Disponível em: <<http://www.open-electronics.org/arduino-isp-in-system-programming-and-stand-alone-circuits/>>. Acesso em: 16 set. 2016.

ORGANISATION FOR ECONOMIC CO-OPERATION AND DEVELOPMENT.**Results from PISA 2012: Brazil**. Paris: OECD. [2012?]. 12 p. Disponível em: <<https://www.oecd.org/brazil/PISA-2012-results-brazil.pdf>>. Acesso em: 02 jul. 2016.

ORGANISATION FOR ECONOMIC CO-OPERATION AND DEVELOPMENT.**PISA 2012 Results in Focus: What 15-Year-Olds Know and What They Can Do With What They Know**. Paris: OECD. 2014. 44 p. Disponível em: <https://www.oecd.org/pisa/keyfindings/pisa-2012-results-overview.pdf>. Acesso em: 03 jul. 2016.

POMILIO, J. A. **Técnicas de Modulação de Potência**. 2014. Disponível em: <<http://www.dsce.fee.unicamp.br/~antenor/pdf/eltpot/cap2.pdf>>. Acesso em: 31 jul. 2016.

QUINTANILHA, L. Irresistível Robô. **Revista ARede**, São Paulo, Ano 3, n. 34, p. 10-17. 2008.

RASPBERRY PI. **Raspberry Pi**. 2016. Disponível em: <<http://www.raspberrypi.org/about>>. Acesso em: 03 jul. 2016.

RESNICK, M. et al. **Programmable Bricks: Toys to Think With**. IBM Systems Journal, vol. 35, no. 3-4, p. 443-452, 1996.

SÃO PAULO (Estado). Secretaria da Educação. **Currículo do Estado de São Paulo: Matemática e suas Tecnologias**. 1. ed. São Paulo, 2011. 72 p.

SPARKFUN. **How to Use a Breadboard**. 2016. Disponível em: <<https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard>>. Acesso em: 28 ago. 2016.

SPRAUL, V. A. **Think Like a Programmer: An Introduction to Creative Problem Solving**. San Francisco: No Starch Press, 2012. 256 p.

STMICROELECTRONICS. **L298: Dual Full-Bridge Driver**. 2000. Disponível em: <<http://pdf.datasheetcatalog.com/datasheet2/2/052daje928cw7pc0uqs1ipyryppy.pdf>>. Acesso em: 28 ago. 2016.

TEXAS INSTRUMENTS. **L293x Quadruple Half-H Drivers**. 2016a. Disponível em: <<http://www.ti.com/lit/ds/symlink/l293.pdf>>. Acesso em: 05 ago. 2016.

TEXAS INSTRUMENTS. **LM35 Precision Centigrade Temperature Sensors**. 2016b. Disponível em: <<http://www.ti.com/lit/ds/symlink/lm35.pdf>>. Acesso em: 07 set. 2016.

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO. Centro Universitário Norte do Espírito Santo. **Física Experimental II: Roteiros de Experiências**. Espírito Santo, [2015?].

VEX. **VEX Robotics**. 2016. Disponível em: <<http://www.vexrobotics.com>>. Acesso em: 03 jul. 2016.

VILLATE, J. E. **Física 2. Eletricidade e Magnetismo**. 2012. Disponível em: <http://www.villate.org/doc/fisica2/fisica2_20120908.pdf>. Acesso em: 24 jul. 2016.

WANG, L.; STROUD, C. E.; TOUBA, N. A. **System-on-Chip Test Architectures: Nanometer Design for Testability (Systems on Silicon)**. Burlington: Morgan Kaufmann, 2007. 896 p.

**APÊNDICE B –LISTA DE COMPONENTES PARA CONSTRUÇÃO DA
PLATAFORMA EDUCACIONAL DE ROBÓTICA**

Componente	Quantidade
Resistor 10kΩ 1/4W	4
Resistor 100Ω 1/4W	6
Resistor 220Ω 1/4W	1
Capacitor eletrolítico 2,2μF	1
Capacitor eletrolítico 100μF	1
Capacitor cerâmico 1μF	1
Capacitor cerâmico 22pF	2
Cristal piezoelétrico 16,000MHz	1
Circuito integrado 7805	1
Circuito integrado L293D	2
Transistor MOSFET 2N7000	2
LED de alto brilho azul 1/2W	2
Alarme piezoelétrico (buzzer) CC tipo ativo	1
Motor elétrico N20 300RPM eixo 3mm	4
Push-button tipo NA	1
Chave 2 posições tipo ON/OFF	1
Display LCD 16x2 azul	1
Módulo Interfaceamento LCD I2C PCF8574	1
Receptor infravermelho TSOP4838	1
Sensor de luminosidade LDR 5mm	1
Sensor óptico reflexivo CNY70	1
Suporte bateria 9V	1
Roda de borracha 43x19mmx3mm c/ furo D	4
Suporte motor N20 com porca e parafuso	4
Microcontrolador Atmel ATMEGA328p	1
Soquete para circuito integrado DIP 28 vias	1
Soquete para circuito integrado DIP 18 vias	2
Conector bloco de terminais parafusados 2P	4
Placas de circuito impresso 60x80mm	2
Kit conectores XH2.54 2P/3P/4P	*quantos forem necessários*
Kit conectores XF2510-2P/3P/4P	*quantos forem necessários*
Cabos AWG 28	*quantos forem necessários*

APÊNDICE C – CÓDIGO-FONTE FINAL DA PLATAFORMA EDUCACIONAL DE ROBÓTICA

```

#include <IRremote.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd (0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); // Inicializa o display no endereço 0x27

int const BOTAOIHM = 12; //Entrada Digital para botão controlador da IHM
int const IHM = 0;

//Declaração faróis dianteiros e sensor crepuscular
int const LDR = 2; //(ENTRADA ANALOG IN)
int const LED = 3; //I/O DIGITAL

//Declaração sensor de estacionamento (CNY70, buzzer e variáveis)
int const SENSORRE = 3; //Entrada Analog In
int const BUZZER = 11; //I/O Digital
int const limiar_CNY = 15;
unsigned int v_aproximacao, t_adic;
unsigned long t_atual, t_anterior = 0;

//Declaração motores
int const IN1 = 9;
int const IN2 = 10;
int const IN3 = 5;
int const IN4 = 6;

//Declaração controle remoto IR
int const IR_REMOTE = 13;
IRrecv irrecv(IR_REMOTE);
decode_results results;

//Declaração variáveis LDR (sensor crepuscular)
int leitura_LDR;

//Declaração variáveis motores
int velocidade = 0;

//Declaração variáveis controle remoto IR
long int valor_lido;

void setup() {
//Serial.begin (9600); //Inicializa comunicação serial com Baud rate 9600bps

lcd.begin (16,2); //Iniciar LCD 16x2
lcd.backlight();
lcd.setCursor(0,0);
  lcd.print("Plataf. Robotica");
  lcd.setCursor(0,1);
  lcd.print("Por: Matheus F.");

irrecv.enableIRIn(); // Inicializa receptor IR do controle remoto

pinMode(IN1, OUTPUT);
pinMode(IN2, OUTPUT);
pinMode(IN3, OUTPUT);
pinMode(IN4, OUTPUT);

digitalWrite(IN1,LOW);
digitalWrite(IN2,LOW);
digitalWrite(IN3,LOW);
digitalWrite(IN4,LOW);

pinMode(LDR, INPUT);
pinMode(LED, OUTPUT);

pinMode(BOTAOIHM, INPUT);

pinMode(BUZZER, OUTPUT);

delay(1500); //Delay para inicialização
lcd.clear();

```

```

}

void loop() {

  sensorCrepuscular();
  velocityControl();

  if (velocidade < 0)
    sensorEstacionamento();

  atualizarIHM();

  // Serial.println("");
  delay(10);
}

void sensorCrepuscular()
{
  leitura_LDR = analogRead(LDR);
  if (leitura_LDR <= 850) digitalWrite(LED, 0);
  else digitalWrite(LED, 1);
}

void velocityControl()
{
  if (irrecv.decode(&results)) {
    valor_lido = results.value; //Captura valor lido pelo receptor IR

    if (valor_lido == 0xFF22DD) { //SETA PARA ESQUERDA
      digitalWrite(IN1,LOW);
      digitalWrite(IN2,HIGH);
      digitalWrite(IN4,LOW);
      digitalWrite(IN3,HIGH);
    }

    if (valor_lido == 0xFFC23D) { //SETA PARA DIREITA
      digitalWrite(IN2,LOW);
      digitalWrite(IN1,HIGH);
      digitalWrite(IN3,LOW);
      digitalWrite(IN4,HIGH);
    }

    if (valor_lido == 0xFF629D) { //SETA PARA CIMA
      velocidade += 50;
      if (velocidade >= 250) velocidade = 250;
      decideSentido(velocidade);
    }

    if (valor_lido == 0xFFA857) { //SETA PARA BAIXO
      velocidade -= 50;
      if (velocidade <= -250) velocidade = -250;
      decideSentido(velocidade);
    }

    if (valor_lido == 0xFF02FD) { //BOTÃO OK
      digitalWrite(IN1,LOW);
      digitalWrite(IN2,LOW);
      digitalWrite(IN3,LOW);
      digitalWrite(IN4,LOW);
      velocidade = 0;
    }

    irrecv.resume(); // Recebe próximo valor
  }
}

void decideSentido (int velocidade)
{
  if (velocidade >= 0)
  {
    digitalWrite(IN1,LOW);
    analogWrite(IN2,velocidade);
    digitalWrite(IN3,LOW);
    analogWrite(IN4,velocidade);
  }
  else

```

```

    {
    digitalWrite(IN2,LOW);
    analogWrite(IN1,-velocidade);
    digitalWrite(IN4,LOW);
    analogWrite(IN3,-velocidade);
    }
}

void sensorEstacionamento ()
{
    v_aproximacao = analogRead(SENSORRE);
    delay(5);

    if (v_aproximacao <= limiar_CNY)
    {
    digitalWrite(BUZZER, LOW);
    }
    else if (v_aproximacao <= (10*limiar_CNY))
        t_adic = 200;
    else t_adic = 50;

    t_atual = millis();

    if ((t_atual - t_anterior >= t_adic) && (v_aproximacao > limiar_CNY))
    {
        t_anterior = t_atual;
        if (digitalRead(BUZZER) == HIGH)
            digitalWrite(BUZZER, LOW);
        else
            digitalWrite(BUZZER, HIGH);
    }
}

void atualizarIHM()
{
    if (digitalRead(BOTAOIHM) == 0)
    {
        lcd.clear();
        contIHM++;
        delay(300);
    }
}

if (contIHM == 4) contIHM = 0;

switch (contIHM) {
case 0:
    lcd.setCursor(0,0);
    lcd.print("Sensor Distancia");
    lcd.setCursor(0,1);
    lcd.print(analogRead(SENSORRE));
    lcd.print(" ");
    break;
case 1:
    lcd.setCursor(0,0);
    lcd.print("Sensor Crepuse.");
    lcd.setCursor(0,1);
    lcd.print(leitura_LDR);
    lcd.print(" ");
    break;
case 2:
    lcd.setCursor(0,0);
    lcd.print("Velocidade Atual");
    lcd.setCursor(0,1);
    lcd.print(velocidade);
    lcd.print(" ");
    break;
case 3:
    lcd.setCursor(0,0);
    lcd.print("Tecla Pression.");
    lcd.setCursor(0,1);
    lcd.print(valor_lido,16);//Imprime valor HEX da tecla pressionada
    lcd.print(" ");
    break;
}
}
}

```