

CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA
Faculdade de Tecnologia de Jundiaí – “Deputado Ary Fossen”
Curso Superior de Tecnologia em Sistemas Embarcados

Alex Bertoloti do Carmo Grellet
João Vitor Comoti Pires dos Santos
Nicolas Kevin Fagundes Silva

**B-DIRO: Um Robô para uso didático com Dispositivos Lógicos
Programáveis e VHDL**

**Jundiaí
2023**

Alex Bertoloti do Carmo Grellet
João Vitor Comoti Pires dos Santos
Nicolas Kevin Fagundes Silva

B-DIRO: Um Robô para uso didático com Dispositivos Lógicos Programáveis e VHDL

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Jundiaí - “Deputado Ary Fossen” como requisito parcial para a obtenção do título de Tecnólogo em Sistemas Embarcados, sob a orientação do Professor Mestre Peter Jandl Jr.

Jundiaí
2023

Este trabalho é dedicado aos docentes da Fatec – Jundiaí que atuaram não só como porta de entrada nos conhecimentos apresentados neste documento, mas também desempenharam um papel de suporte e auxílio nas atividades desempenhadas.

AGRADECIMENTOS

Primeiramente agradecemos a colaboração e empenho da equipe, que em todas as dificuldades se manteve proativos e trabalhando em prol de alcançar o objetivo traçado, na elaboração desse projeto.

Agradecemos também o suporte e a estrutura disponibilizada pela instituição de ensino FATEC Jundiaí Deputado Ary Fossen, e pela equipe de professores idealizadores e fundadores do curso de Sistemas Embarcados, Mestre Cláudio Luís Vieira Oliveira, Dr. Humberto Augusto Piovesana Zanetti e Mestre Peter Jandl Junior, os quais em todas as hipóteses, sempre se colocaram à disposição em nos ajudar no decorrer de todo o curso. Um agradecimento especial ao coordenador do curso de Sistemas Embarcados já citado anteriormente, Mestre Cláudio Luís Vieira Oliveira e ao nosso orientador Mestre Peter Jandl Junior.

Por fim, agradecemos a todos os familiares da equipe, que foram o principal suporte, e ponto de estabilidade mental para a equipe se manter centrada, com a cabeça no lugar para a execução do trabalho acadêmico.

RESUMO

GRELLET, Alex. SANTOS, João; SILVA, Nicolas. **Projeto de um Robô para uso didático com Dispositivos Lógicos Programáveis e VHDL**. 30 p. Trabalho de Conclusão de Curso de Tecnólogo em Sistemas Embarcados. Faculdade de Tecnologia de Jundiaí - “Deputado Ary Fossen”. Centro Estadual de Educação Tecnológica Paula Souza. Jundiaí. 2023.

O atual trabalho acadêmico promove uma iniciativa de um projeto com a pretensão de desenvolver um robô com propósitos didáticos, para fomentar o aprendizado prático de DLPs (Dispositivos Lógicos Programáveis), utilizando a linguagem HDL (Linguagem de Descrição de Hardware) e CUPL (Linguagem de programação da Universidade de Cambridge), colocando em prática o que foi feito em um simulador. Será abordado, desde a construção do circuito eletrônico, baseado em um protótipo feito com portas lógicas e a transformação para um circuito com DLP e reprodução da lógica aplicada com o escopo proposto de comportamento atrativo e repulsivo, através de sensores de luz (LDR), e adição de sensores reflexivos para possibilitar o uso como robô seguidor de linha. O propósito principal se trata de oferecer uma solução de custo reduzido e o mais simples possível, que viabilize a utilização de circuitos reais em salas de aula, tornando o aprendizado mais atrativo e prático.

Palavras-chave: Robô. Dispositivo Lógico Programável (DLP). Linguagem de Descrição de Hardware (VHDL). Sensores

ABSTRACT

GRELLET, Alex; SANTOS, João; SILVA, Nicolas. **Project of a robot for didactic use with programmable logic devices and vhdl**. 30 p. End-of-course paper in technologist degree in embedded systems. Faculdade de tecnologia de Jundiaí - “deputado Ary Fossen”. Centro estadual de educação tecnológica Paula Souza. Jundiaí. 2023.

The current academic work promotes a project initiative with the aim of developing a robot with didactic purposes, to encourage practical learning of PLDs (Programmable Logic Devices), using the HDL language (Hardware Description Language) and CUPL (Cambridge University Programming Language), putting into practice what was done in a simulator. It will be covered, from the construction of the electronic circuit, based on a prototype made with logic gates and the transformation to a circuit with PLD and reproduction of the applied logic with the proposed scope of attractive and repulsive behavior, through light sensors (LDR), and addition of reflective sensors to enable use as a line-following robot. The main purpose is to offer a low-cost and as simple solution as possible, which enables the use of real circuits in classrooms, making learning more attractive and practical.

Keywords: Robot. Programmable Logic Devices (PLD). Hardware Description Language (VHDL). Sensors.

LISTA DE ILUSTRAÇÕES

Figura 1 - Circuito PROM.....	11
Figura 2 - Circuito PAL	12
Figura 3 - Circuito PLA.....	13
Figura 4 - Circuito FPGA	14
Figura 5 - Circuito do Gravador	19
Figura 6- Esquema do gravador	20
Figuras 7 - Pinagem do ATF16	21
Figura 8- Diagrama ATF16	21
Figura 9- Pinagem Arduino Nano	23
Figura 10- Circuito B-DIRO	Erro! Indicador não definido.
Figura 11- Esquema B-DIRO	Erro! Indicador não definido.
Figura 12- Módulo CI MX1616.....	24
Figura 13- Motor DC 6V Eixo Duplo	Erro! Indicador não definido.
Figura 14- ensor Óptico Reflexivo.....	25
Figura 15- Esquema Sensor Óptico Reflexivo.....	25
Figura 16- Sensor Foto Resistor (LDR).....	25
Figura 17- Programação no Software Wincupl	27
Figura 18- Compilação no Software Wincupl.....	Erro! Indicador não definido.
Figura 19- Inicialização do Afterburner.....	Erro! Indicador não definido.
Figura 20- Configuração de Tensão	29
Figura 21- Configuração de Tensão	32
Figura 22- – Tabela Verdade Robô Repulsivo e Atrativo	Erro! Indicador não definido.

SUMÁRIO

1	INTRODUÇÃO	9
2	DISPOSITIVOS LÓGICOS PROGRAMÁVEIS.....	10
2.1.1	Definição de DLP	10
2.1.2	Tipos de DLPs e suas Funcionalidades	10
2.1.3	Programação de DLPs	14
2.1.4	Sintaxe do HDL	15
2.1.5	A Linguagem AHDL	16
2.1.6	Integração com a linguagem VHDL.....	17
2.1.7	A Linguagem CUPL.....	18
3	CIRCUITOS ELETRÔNICOS	19
3.1	Circuito do Gravador	19
3.1.1	Módulo ATF16	20
3.1.2	Conversor DC/DC StepUp (Boost)	22
3.1.3	Arduino Nano	22
3.2	Circuito Principal - B-DIRO (Bertoloti Didatic Robot).....	23
4	AMBIENTE DE DESENVOLVIMENTO E PROGRAMAÇÃO.....	27
4.1	Gravação de Comandos no Circuito.....	27
4.1.1	Calibração da Tensão de Gravação	29
4.1.2	Etapa de Gravação	30
5	APLICAÇÃO DIDÁTICA DO CIRCUITO DE APRENDIZAGEM	31
5.1	Aplicações sugeridas.....	31
5.1.1	Um robô seguidor de linha que percorre um trajeto estabelecido	31
5.1.2	Um robô com o modo repulsivo e atrativo	33
6	ANÁLISES E RESULTADOS	36
	REFERÊNCIAS	37

1 INTRODUÇÃO

A eletrônica digital é um elemento presente em grande parte dos dispositivos que cercam a humanidade. Os circuitos lógicos integrados contribuem proporcionando avanços e criando um mercado volátil que fabrica inúmeros apetrechos úteis ao cotidiano humano.

Os primórdios da eletrônica digital originaram-se dos fundamentos desenvolvidos pelo matemático George Boole. A chamada álgebra booleana tem como cerne operações utilizando apenas valores 0 e 1, que representam sempre dois estados, como “verdadeiro” e “falso”. Esse conceito foi usado como base na criação das funções lógicas, que são blocos de operações básicas que ao receberem um ou mais sinais de entrada, aplicam uma operação lógica sobre esses sinais e produzem um sinal de saída que expressa o resultado da operação. Cada tipo de função tem uma tabela-verdade associada, que exterioriza quaisquer combinações prováveis de valores e os resultados condizentes.

O desenvolvimento das funções lógicas e seu posterior aperfeiçoamento permitiu a criação dos chamados circuitos digitais, que são sistemas eletrônicos formados por dispositivos cuja tecnologia combinada possibilita a realização de tarefas diversas como aritmética, processamento e armazenagem de dados.

No decorrer dos avanços nos estudos e na elaboração dos circuitos digitais elevou-se a necessidade de operações cada vez mais complexas e amplas, o que fomentou a criação dos dispositivos lógicos programáveis (DLPs). Segundo Tocci (2017), os DLPs são configurados eletronicamente, formando um circuito lógico customizável pelo usuário, que pode moldar seu funcionamento e desempenho por meio de programação. A pesquisa presente neste trabalho tem como objetivo dissertar acerca dos dispositivos lógicos programáveis e propor uma solução à carência do seu estudo prático no âmbito educacional.

O foco principal visa a construção de um robô de nome B-DIRO, utilizando circuitos lógicos programáveis de forma simples e acessível, para implementação em salas de aula. A equipe optou pela abordagem do tema, devido à curiosidade aguçada durante as aulas de eletrônica digital, ministradas no curso de Sistemas Embarcados.

2 DISPOSITIVOS LÓGICOS PROGRAMÁVEIS

2.1.1 Definição de DLP

Os DPLs são circuitos integrados (CI) que não possuem uma função lógica definida, esta função lógica pode ser configurada pelo usuário, proporcionando adaptabilidade ao circuito, o tornando muito útil para projetos, pois inibe a necessidade de mudar o circuito a cada mudança de escopo de operação, garantindo não apenas praticidade, mas também reduzindo os custos do projeto. O funcionamento de um DLP (Programmable Logic Device) depende inteiramente da função lógica desenvolvida pelo usuário, ou seja, um programa que é compilado por um software, e gravado no circuito integrado.

2.1.2 Tipos de DLPs e suas Funcionalidades

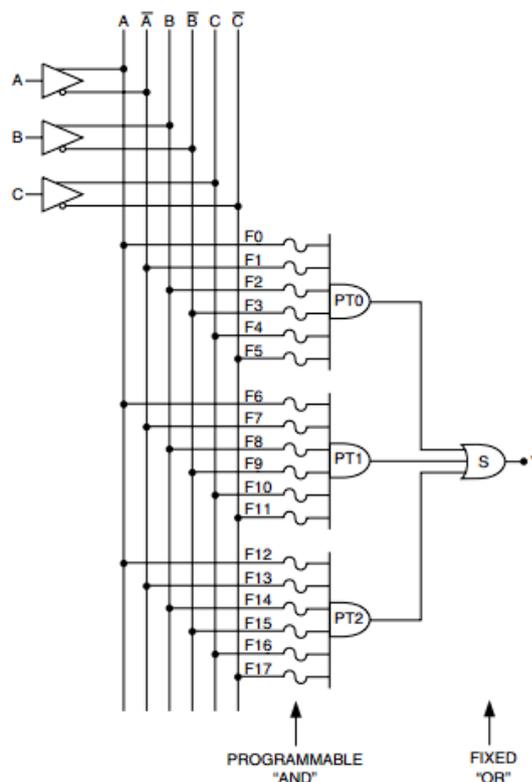
2.1.2.1 Programmable Read Only Memory (PROM)

PROMs são Memórias Somente Leitura Programáveis. Embora o nome não implique em lógica programável, as PROMs são, de fato, lógicas. A descrição no manual fornecido pela ATMEL (2006), menciona a arquitetura da maioria das PROMs, no que normalmente consiste em um número fixo de termos na matriz AND, que alimentam uma matriz OR programável. Elas são principalmente usadas para decodificar combinações específicas de entrada em funções de saída, como mapeamento de memória em ambientes de microprocessadores.

2.1.2.3 Generic Array Logic (GAL)

GALs são dispositivos projetados para emular muitas PALs comuns através do uso de macrocélulas. Caso um usuário tiver um projeto implementado usando várias PALs comuns, ele pode configurar vários GALs iguais para emular cada um dos outros dispositivos, em concordância a ATMEL (2006). Isso reduzirá o número de dispositivos diferentes em estoque e aumentará a quantidade adquirida. Geralmente, uma grande quantidade do mesmo dispositivo deve reduzir o custo individual dele. Além disso, esses dispositivos são eletricamente apagáveis, o que os torna muito úteis para engenheiros de design.

Figura 2 - Circuito PAL

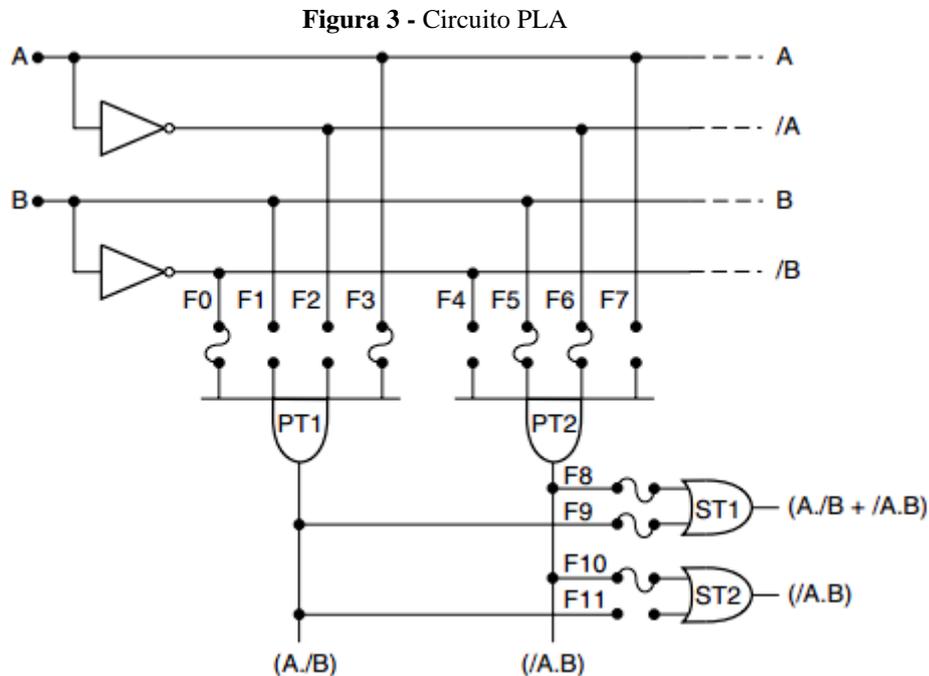


Fonte: ATMEL WINCUPPL, Disponível em: [Atmel Wincupl Manual](#). Acesso em: 30/07/2023

2.1.2.4 Programmable Logic Array (PLA)

PLAs são dispositivos que possuem aspectos programáveis tanto do tipo E (AND) quanto do tipo OU (OR), o que possibilita todo aspecto do tipo E alimente qualquer aspecto do tipo OU. As PLAs provavelmente têm a maior flexibilidade, em comparação com os outros dispositivos em termos de funcionalidade lógica. As PLAs retratado por ATMEL (2006),

possuem feedback da matriz do tipo OU para a matriz do tipo E, que pode ser usado para implementar máquinas de estados assíncronas.



Fonte: ATMEL WINCUPPL, Disponível em: [Atmel Wincupl Manual](#). Acesso em: 30/07/2023

2.1.2.5 PLDs Complexos

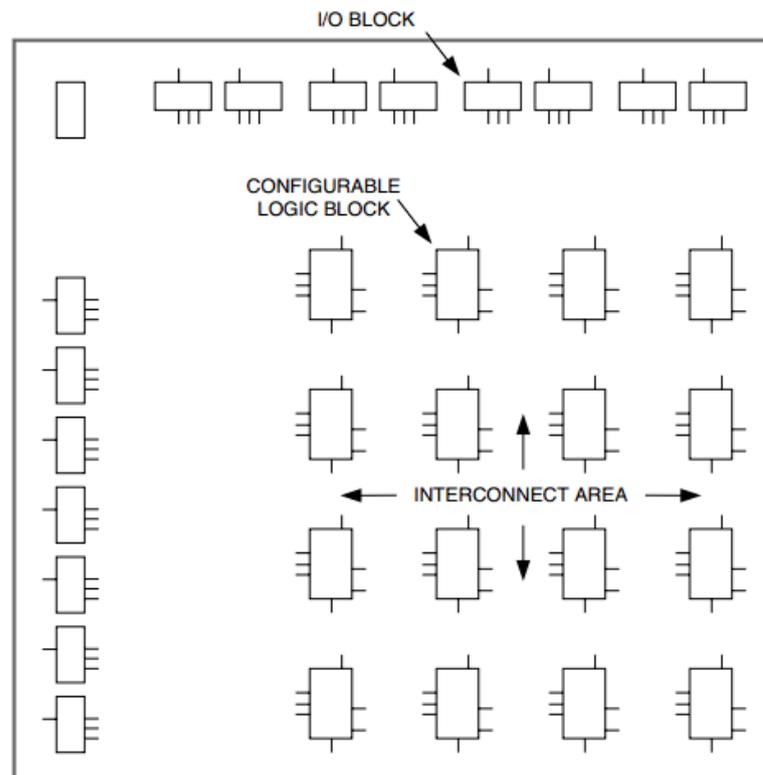
Os PLDs complexos são considerados como dispositivos de Matriz Lógica Programável muito grandes que têm algumas características de PLAs. O ATMEL (2006), explica que a organização básica é muito semelhante a uma PAL, com a capacidade de aumentar a quantidade de termos do tipo E, para qualquer termo fixo do tipo OU. Isso é realizado, seja utilizando termos do tipo E adjacentes, ou recorrendo a termos do tipo E de uma matriz expansora. Isso permite que praticamente qualquer projeto, seja implementado dentro desses dispositivos.

2.1.2.6 Field Programmable Gate Array (FPGA)

FPGAs são circuitos integrados (CIs) programáveis eletricamente, que contêm múltiplos níveis de lógica. As FPGAs apresentam alta densidade de portas, com alto desempenho, de muitas entradas e saídas definíveis pelo usuário, por meio de um esquema de interconexão flexível e um ambiente de design semelhante ao de uma matriz de portas. Elas não estão limitadas à típica matriz AND-OR. Em vez disso, contêm uma matriz interna de elementos

lógicos configuráveis (CLBs – Configurable Logic Blocks) e um anel circundante de blocos de entrada e saída digitais (IOBs – Input/Output Blocks). Cada CLB possui lógica combinatória programável e registradores de armazenamento. A seção de lógica combinatória de cada bloco, é capaz de implementar qualquer função booleana de suas variáveis de entrada. Ele também contém *flip-flops*, que podem ser usados para armazenar entradas e saídas. As chaves programáveis conectam as entradas e saídas dos IOBs e CLBs às linhas próximas. Linhas longas percorrem todo o comprimento ou largura do dispositivo, evitando intercâmbios para fornecer a distribuição de sinais críticos com o mínimo de atraso ou desvio (ATMEL, 2006).

Figura 4 - Circuito FPGA



Fonte: ATMEL WINCUPL, Disponível em: [Atmel Wincupl Manual](#). Acesso em: 30/07/2023

2.1.3 Programação de DLPs

Originalmente para configurar um DLP, era preciso atribuir as entradas um e zero em cada uma das suas milhares de grades de conexões programáveis, sendo um processo complexo e pouco prático. Porém, com a introdução do detalhamento da linguagem do hardware, tornou-se

possível descrever a operacionalidade dos circuitos por meio de programação em alto nível, ou seja, um formato mais compreensível ao programador.

A linguagem HDL (Hardware Description Language), permite caracterizar todos os comportamentos importantes de um circuito lógico, em diversos níveis de abstração e independentemente da complexidade (CODÁ, 2014). Nos tópicos seguintes serão apresentados os dois modelos de HDL e sua relevância quando aplicada na programação dos DLPs.

2.1.4 Sintaxe do HDL

Define-se sintaxe como a estrutura de um código pautada por um conjunto de leis, que regem determinada linguagem. Na descrição de hardware, a sintaxe não é muito diferente das demais utilizadas tanto em alto, quanto em baixo nível de máquina. Dentre as regras que compõem o básico da parametrização de qualquer circuito, destacam-se as definições de entrada e saída (Input/Output) e as operações com as variáveis de entrada, que irão resultar nas saídas correspondentes. A construção de sintaxe da linguagem HDL, é dada seguindo um fluxo dividido em três partes: documentação, definições I/O e descrição funcional (TOCCI, 2017).

2.1.4.1 Documentação

A fase da documentação trata-se da atribuição de uma nomenclatura ao bloco do circuito e respectivamente ao arquivo no formato HDL. Para nomear um circuito são utilizadas palavras com características que variam dependendo do modelo de linguagem empregada. No entanto, a principal exigência quando é usada a descrição de hardware, é de que tanto o bloco, quanto o arquivo HDL, possuam o mesmo termo como nome.

2.1.4.2 Definição de Entradas e Saídas (I/O)

De forma análoga às outras linguagens de programação, em HDL tem-se a necessidade de nomear e definir quais serão as entradas e saídas esperadas ('ports'), respeitando a sua origem no sistema universal adotado (Ex: pode ser um único bit 0 ou 1). Outros dois termos importantes na fase de definição são o "modo" e o "tipo".

O modo de um port define se é entrada, saída ou ambos. O tipo se refere ao número de bits e a como esses bits são agrupados e interpretados. Se o tipo de entrada for um número binário de 4 bits transmitido por um teclado numérico, ele pode ter 16 valores diferentes ($0000_2 - 1111_2$). (TOCCI, 2014, p.109).

2.1.4.3 Descrição Funcional

A etapa refere-se à descrição das operações lógicas que envolverão as entradas, a fim de gerar as saídas desejadas, dependendo da empregabilidade do DLP. As operações são determinadas utilizando uma coletânea de símbolos e termos contidos no dicionário da linguagem.

2.1.5 A Linguagem AHDL

A AHDL, ou Altera Hardware Description Language, é uma narração de hardware desenvolvida pela Altera Corporation, uma empresa reconhecida por seus dispositivos lógicos programáveis, como FPGAs (Field-Programmable Gate Arrays). Essa linguagem de nível alto é utilizada para descrever circuitos digitais de forma clara e estruturada, permitindo que os projetistas criem e projetem circuitos lógicos complexos.

A AHDL possui uma sintaxe baseada em texto que facilita a conexão de blocos lógicos e a definição de suas funcionalidades. Essa linguagem é orientada a eventos, o que torna a modelagem de sistemas assíncronos e síncronos mais fácil e eficiente.

Uma das principais vantagens da AHDL é a sua integração com as ferramentas de desenvolvimento da Altera, o que permite a síntese e implementação direta em seus dispositivos FPGA (TOCCI, 2017). Isso agiliza o processo de prototipagem e validação de projetos, tornando-a uma escolha popular entre os projetistas.

A AHDL também é compatível com outras linguagens de descrição de hardware, como a VHDL e Verilog, o que dá aos projetistas a flexibilidade de escolher a expressão mais apropriada às suas necessidades e conhecimentos.

Embora a AHDL seja uma opção poderosa para projetos em dispositivos FPGA da Altera, é importante observar que ela é específica dessa empresa, o que pode limitar sua portabilidade para outros dispositivos FPGA de diferentes fabricantes.

Ao longo do tempo, a Altera tem continuado a aprimorar e atualizar a AHDL para melhorar sua funcionalidade e eficiência. A empresa também oferece suporte técnico e documentação abrangente, para ajudar os projetistas a utilizarem a linguagem de forma eficaz.

Em resumo, a AHDL é uma linguagem de descrição de hardware, que facilita o projeto e a implementação de circuitos digitais em dispositivos FPGA da Altera. Com sua sintaxe estruturada e integração com as ferramentas da empresa, ela oferece uma solução eficiente para criar sistemas digitais complexos. No entanto, é importante estar ciente das limitações e requisitos específicos da linguagem ao considerar seu uso e explorar outras opções, caso seja necessário trabalhar com dispositivos FPGA de outros fabricantes.

2.1.6 Integração com a linguagem VHDL

VHDL é uma poderosa linguagem de descrição de hardware que pode ser utilizada tanto para a síntese quanto para a simulação de circuitos. Na síntese, a linguagem descreve o comportamento ou a estrutura desejada de uma unidade digital, a partir da qual um circuito físico compatível é inferido pelo compilador. Já na simulação, a linguagem gera estímulos e os aplica ao circuito codificado, verificando posteriormente seu comportamento através da comparação das respostas obtidas com os resultados esperados, facilitando imensamente a verificação (e documentação) do circuito em desenvolvimento.

TOCCI (2017), cita que as principais aplicações do VHDL incluem a síntese/implantação de circuitos digitais em chips de matriz de portas programáveis em campo (FPGA) e a geração de layout/máscara para a fabricação de circuitos integrados específicos de aplicação (ASIC), juntamente com simulações funcionais e temporais correspondentes.

VHDL possui uma sintaxe estruturada, que permite aos projetistas de hardware descrever circuitos digitais de forma concisa e legível. Sua natureza de descrição comportamental ou estrutural torna-o adequado para descrever sistemas complexos e de larga escala, além de permitir a representação de múltiplos níveis de abstração.

A linguagem oferece suporte a tipos de dados personalizados, permitindo aos projetistas definir tipos específicos de acordo com as necessidades do projeto. Além disso, VHDL possui uma biblioteca rica de funções e operadores pré-definidos que facilitam a manipulação de sinais e a realização de operações lógicas e aritméticas.

Outra característica importante do VHDL é sua capacidade de modelar o comportamento de sistemas em diferentes níveis de abstração, permitindo aos projetistas validar o projeto em etapas iniciais do desenvolvimento. Isso ajuda a identificar problemas e corrigi-los antes de avançar para a implementação física, economizando tempo e recursos.

O VHDL também possui suporte para hierarquia de design, o que significa que os circuitos podem ser divididos em blocos funcionais menores, facilitando a reutilização e a manutenção de código. Esse produto torna-se especialmente útil em projetos de grande porte e complexos, onde a modularidade é essencial para garantir a eficiência e escalabilidade.

Uma das principais vantagens do VHDL é sua independência de plataforma, o que significa que os projetos desenvolvidos na linguagem podem ser sintetizados para diferentes tecnologias de chips e fabricantes de dispositivos, aumentando a flexibilidade e a portabilidade do projeto.

Apesar de sua poderosa funcionalidade, o VHDL também apresenta desafios, especialmente para projetistas iniciantes. A curva de aprendizado pode ser íngreme, e a complexidade dos projetos pode exigir um conhecimento sólido de eletrônica digital e lógica programável.

Em resumo, o VHDL é uma linguagem essencial para o projeto de circuitos digitais complexos e é amplamente utilizado na indústria de design de hardware. Sua capacidade de descrever tanto o comportamento quanto a estrutura de circuitos, juntamente com suas características de hierarquia, modularidade e independência de plataforma, o tornam uma ferramenta poderosa e versátil para engenheiros de hardware em todo o mundo.

2.1.7 Cambridge University Programming Language (CUPL)

A sigla CUPL refere-se a uma linguagem de descrição de hardware, muito útil para projetar circuitos lógicos programáveis como CPLDs (Complex Programmable Logic Devices) e FPGAs (Field-Programmable Gate Arrays). O que torna esta linguagem útil na confecção de circuitos DLP é a facilidade em descrever o comportamento e a interconexão dos componentes de um circuito lógico em sintaxes simples. A linguagem permite que os desenvolvedores descrevam a funcionalidade de um circuito lógico, abrangendo portas lógicas, *flip-flops*, multiplexadores e demais elementos, por meio de uma notação específica. Essa descrição pode ser usada para criar uma representação textual do projeto do circuito

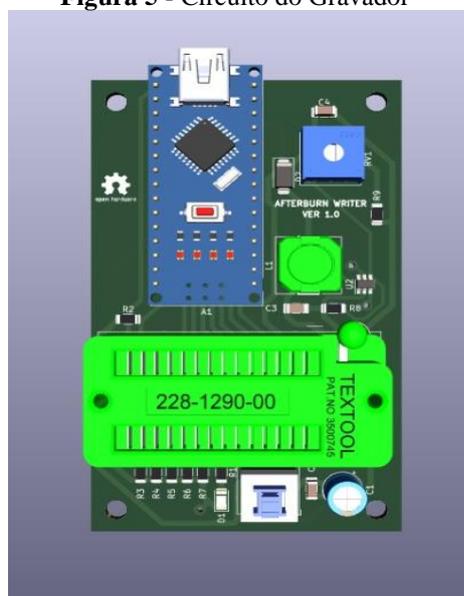
Em suma, é possível concluir que CUPL se trata de um conjunto de ferramentas usadas, para projetar e programar dispositivos lógicos programáveis, permitindo a descrição e implementação de circuitos personalizados, para uma vasta diversidade de investimentos eletrônicos.

3 CIRCUITOS ELETRÔNICOS

Para o projeto foram construídos dois circuitos, um gravador de comandos elétricos e um robô com dois modos de operação.

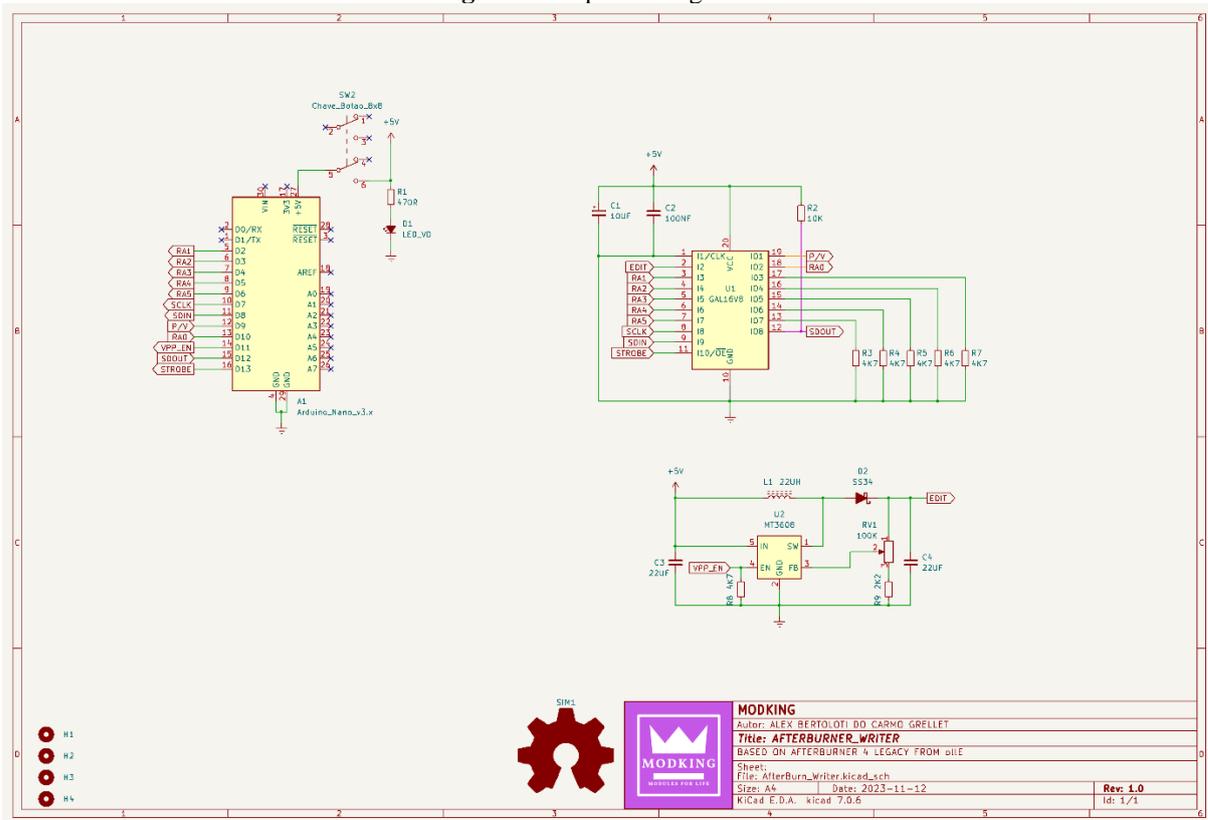
3.1 Circuito do Gravador

Figura 5 - Circuito do Gravador



Fonte: Arquivo pessoal

Figura 6- Esquema do gravador



Fonte: Arquivo pessoal

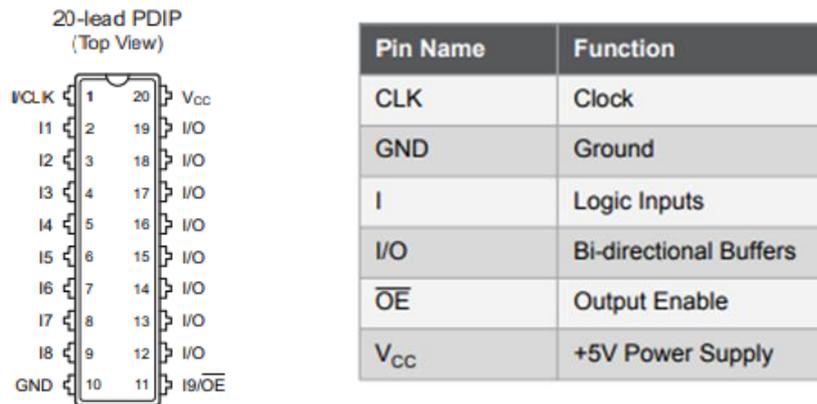
O circuito elaborado é composto por três elementos soldados na placa de circuito impresso, que serão descritos nos subtópicos seguintes.

3.1.1 Módulo ATF16

O circuito integrado ATF16V8B-15PU da empresa Atmel. Trata-se de um Dispositivo Lógico Programável de Alta Performance em tecnologia CMOS (EE PLD) que utiliza a memória Flash apagável eletricamente. Esse dispositivo é sistematizado para operar em uma imensa faixa de velocidade e sua capacidade de fornecer uma solução de baixo consumo de energia notável. Possui três modos diferentes de operação, que podem ser configurados automaticamente no software, ampliando ainda mais suas capacidades de implementação lógica avançada.

O dispositivo possui as pinagens configuradas conforme a distribuição abaixo:

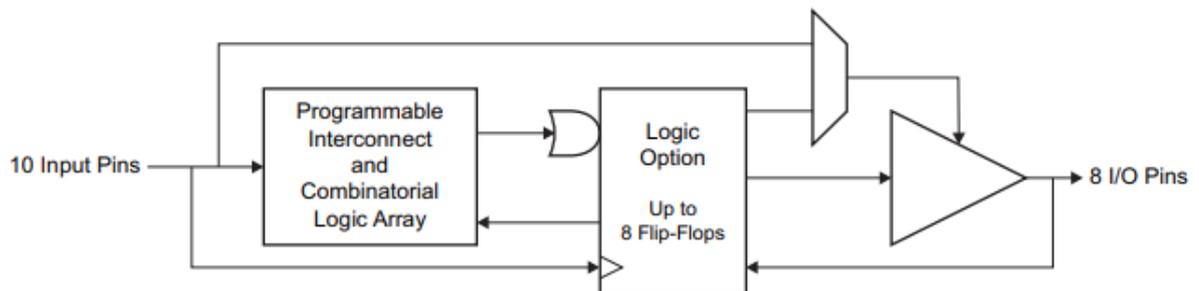
Figuras 7 - Pinagem do ATF16



Fonte: ATMEL (2014). Disponível em: <https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-0364-PLD-ATF16V8B-8BQ-8BQL-Datasheet.pdf>, Acesso em: 01/07/2023.

Conforme pode-se observar nas imagens, o pino número 1 é referente ao clock, um sinal elétrico periódico usado para sincronizar o funcionamento dos elementos presentes no microchip. A sequência formada pelos pinos 2 a 7 são as entradas lógicas. O intervalo das pinagens 14 a 19 é definido pelos buffers direcionais, que são responsáveis por direcionar o fluxo de dados nas entradas e nas saídas.

Figura 8- Diagrama ATF16



Fonte: ATMEL (2014). Disponível em: [ATF16V8B-8BQ-8BQL-Datasheet](#), Acesso em: 01/07/2023.

3.1.2 Conversor DC/DC StepUp (Boost)

Outro componente utilizado no circuito é o conversor DC/DC do tipo Boost, ou elevador de tensão. É um conversor chaveado que tem como característica, a partir de um nível de tensão baixo elevar seu valor entre uma faixa de trabalho.

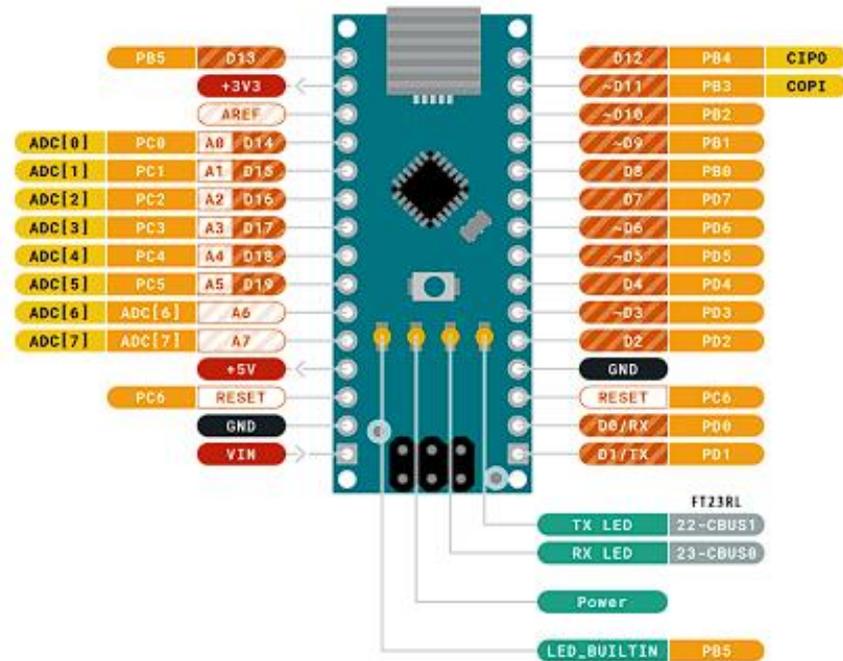
Os indutores também possuem um papel crucial nesses conversores, onde a corrente varia com o tempo. Os indutores armazenam energia no campo magnético criado pela corrente, o que permite gerar tensões maiores do que a aplicada, especialmente em conversores boost. Não somente os indutores, como também os transistores MOSFET de canal N, são essenciais na operação dos conversores chaveados, onde o transistor é chaveado continuamente para criar uma corrente pulsante através do diodo e, posteriormente, obter uma tensão contínua na carga, assegurando o funcionamento eficiente destes circuitos.

3.1.3 Arduino Nano

O Arduino Nano é uma placa de amadurecimento consistente, elaborada com o objetivo de facilitar a criação de protótipos que possuem dimensionamentos reduzidos, fornecendo um alto desempenho. O microcontrolador ATmega328p é a base da placa e oferece interfaces suficientes para a execução do projeto. O processador de 8 bits, conta com um baixo consumo de energia e opera a uma frequência de *clock* de 16 MHz.

Em termos de conectorização e pinagem, o Arduino Nano não possui uma entrada para alimentação DC, sendo necessário utilizar um cabo USB Mini-B para fornecer energia para a placa. Ele oferece 20 pinos de entrada/saída digitais, 8 pinos analógicos e uma porta mini-USB para comunicação com o PC.

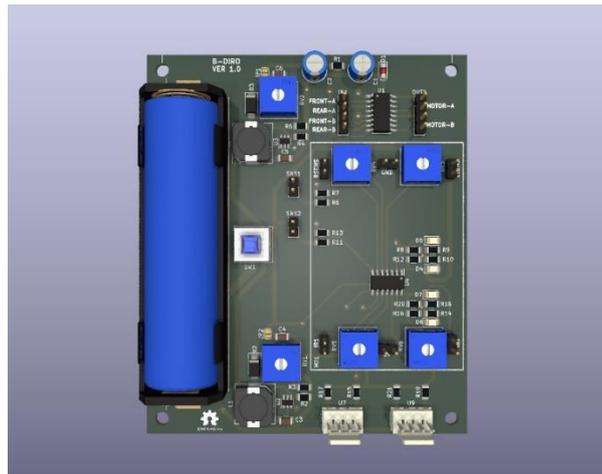
Figura 9- Pinagem Arduino Nano



Fonte: ARDUINO ONLINE SHOP, Disponível em: [Arduino Nano](#). Acesso em: 08/11/2023

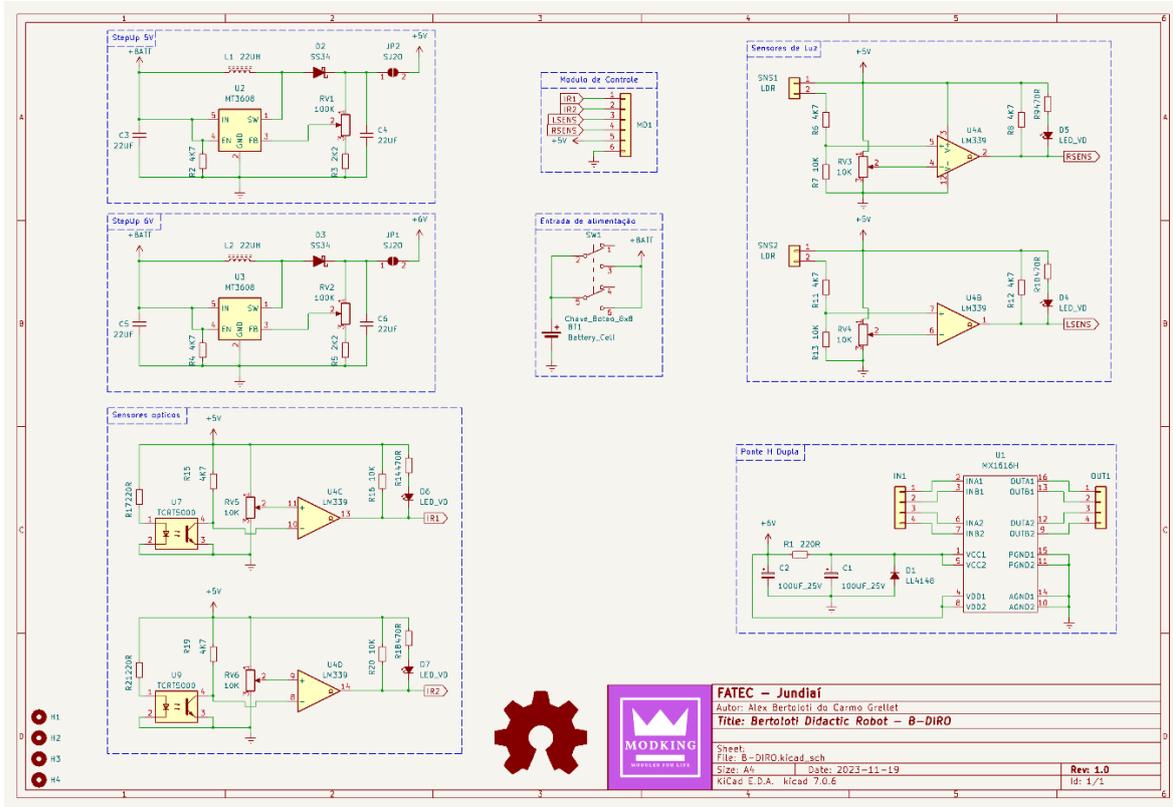
3.2 Circuito Principal - B-DIRO (Bertoloti Didatic Robot)

Figura 10- Circuito B-DIRO



Fonte: Arquivo pessoal

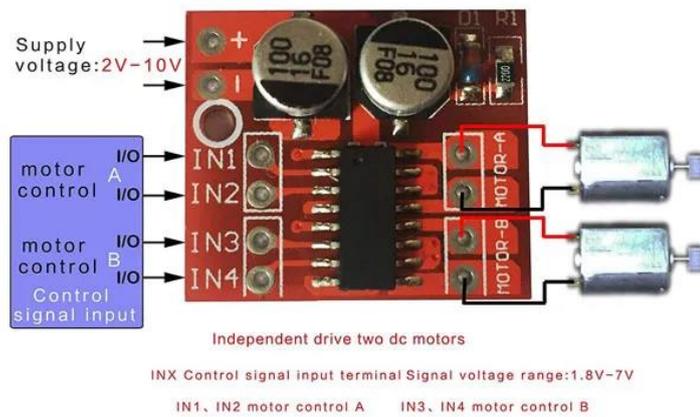
Figura 11- Esquema B-DIRO



Fonte: Arquivo Pessoal

O circuito é constituído por dois sensores ópticos reflexivos, dois LDRs, dois motores DC de 6V com caixa de diminuição e apoio duplo ligados em ponte H num módulo CI MX1616H.

Figura 10- Módulo CI MX1616



Fonte: ART OF CIRCUITS, Disponível em: Art of Circuits – [MX1616 Module](#). Acesso em: 08/11/2023

Os módulos CI MX1616 e o CI MX1508 são ideais para circuitos que trabalham com dois motores, pois são compactos e comercializados por um preço acessível. Os motores DC 6V

com caixa de redução e eixo duplo apresentam um ótimo custo-benefício em relação a desempenho e custo, e atendem todas as expectativas dentro do propósito do projeto.

Figura 13- Motor DC 6V Eixo Duplo



Fonte: MONTENEGRO (2021), Disponível em: [Repostório - UFC](#). Acesso em: 08/11/2023

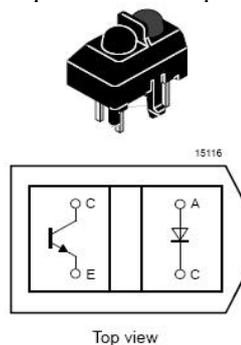
Pode-se considerar que os dois sensores óticos reflexivos e os LDRs são o núcleo de todo o funcionamento do robô, captando as variações do ambiente e transpondo para os motores, portanto sua utilização é indispensável para o projeto.

Figura 11- Sensor Óptico Reflexivo



Fonte: PEREIRA (2017), Disponível em: [Repositório - Unitau](#). Acesso em: 08/11/2023.

Figura 12- Esquema Sensor Óptico Reflexivo



Fonte: PEREIRA (2017), Disponível em: [Repositório - Unitau](#). Acesso em: 08/11/2023.

Figura 13- Sensor Foto Resistor (LDR)



Fonte: WENDLING (2010), Disponível em: [Unidade Estadual Paulista - Sensores](#). Acesso em: 08/11/2023

Os sensores possuem um comparador LM 339, uma integração de quatro comparadores que podem ser ajustados para trabalhar na faixa de frequência de luminosidade desejada, permitindo que a reflexão em relação a luminosidade seja considerada apenas no espectro de uma cor ajustada para o sensor óptico.

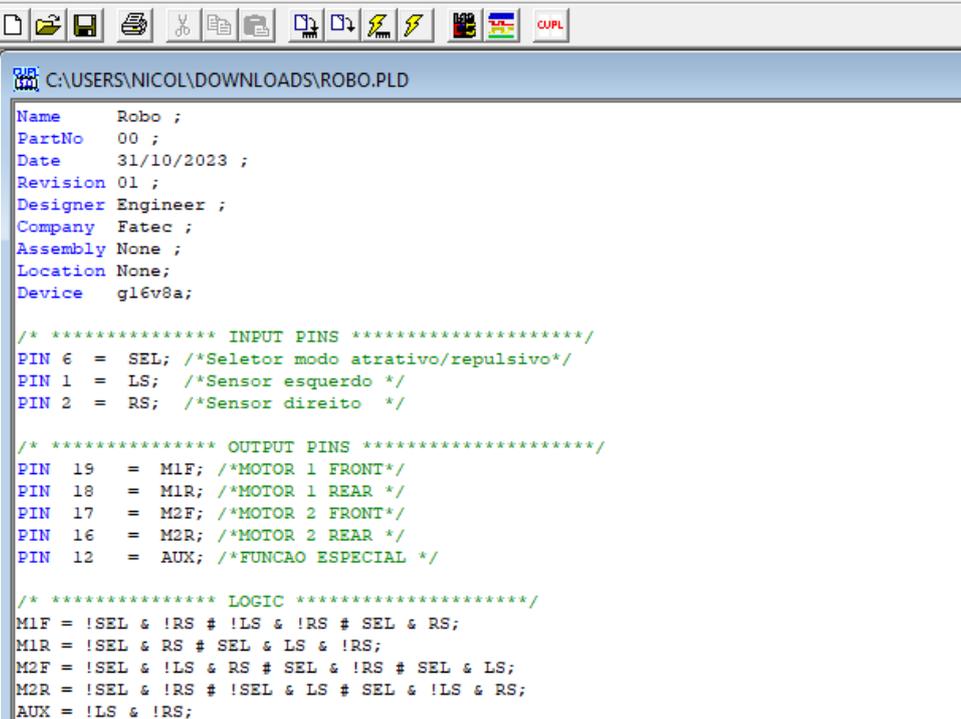
4 AMBIENTE DE DESENVOLVIMENTO E PROGRAMAÇÃO

4.1 Gravação de Comandos no Circuito

O dispositivo gerado pelo circuito do gravador é programado inteiramente na linguagem CUPL, utilizando o Windows Powershell e o software WinCupl que serve para realizar a compilação do código e o design do funcionamento do CPLD.

No ambiente de desenvolvimento do software, primeiramente os pinos de *input* e *output* devem ser declarados, posteriormente é estabelecido a lógica “booleana” que irá ser gravada no circuito, esta etapa é muito importante, pois é neste momento que é definido todo o comportamento do circuito digital.

Figura 14- Programação no Software Wincupl



```

File Edit View Options Run Utilities Window Help
C:\USERS\NICOL\DOWNLOADS\ROBO.PLD
Name Robo ;
PartNo 00 ;
Date 31/10/2023 ;
Revision 01 ;
Designer Engineer ;
Company Fatec ;
Assembly None ;
Location None;
Device gl6v8a;

/* ***** INPUT PINS *****/
PIN 6 = SEL; /*Seletor modo atrativo/repulsivo*/
PIN 1 = LS; /*Sensor esquerdo */
PIN 2 = RS; /*Sensor direito */

/* ***** OUTPUT PINS *****/
PIN 19 = M1F; /*MOTOR 1 FRONT*/
PIN 18 = M1R; /*MOTOR 1 REAR */
PIN 17 = M2F; /*MOTOR 2 FRONT*/
PIN 16 = M2R; /*MOTOR 2 REAR */
PIN 12 = AUX; /*FUNCAO ESPECIAL */

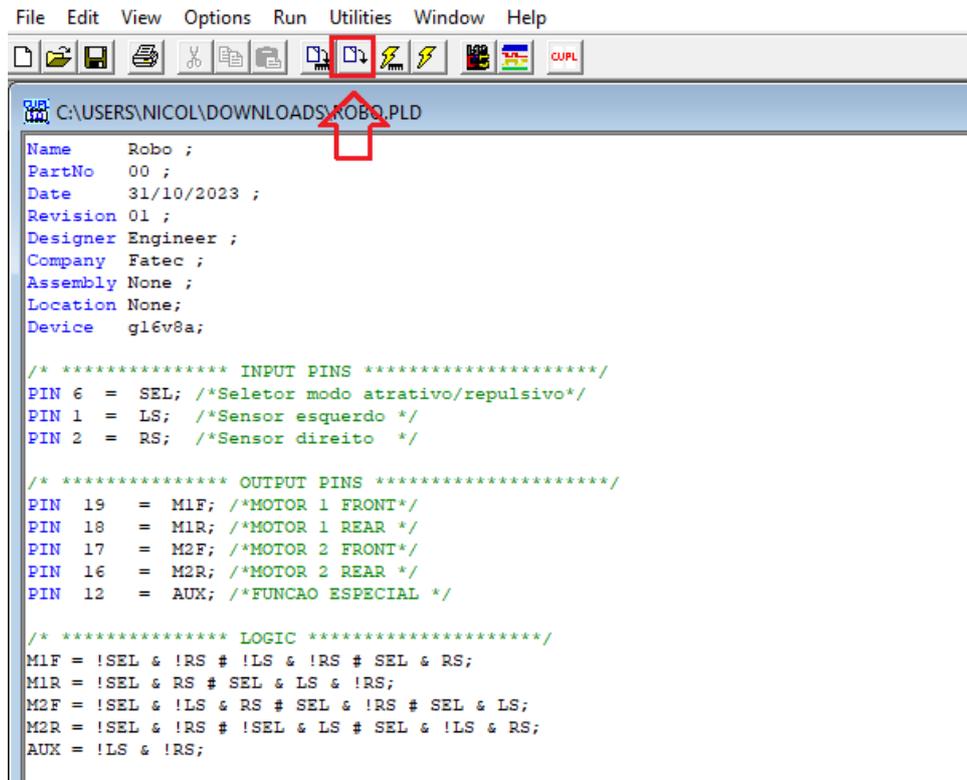
/* ***** LOGIC *****/
M1F = !SEL & !RS # !LS & !RS # SEL & RS;
M1R = !SEL & RS # SEL & LS & !RS;
M2F = !SEL & !LS & RS # SEL & !RS # SEL & LS;
M2R = !SEL & !RS # !SEL & LS # SEL & !LS & RS;
AUX = !LS & !RS;

```

Fonte: Arquivo pessoal

Com a lógica já desenvolvida, a última etapa desempenhada no software é a compilação do programa clicando no botão na barra de tarefas.

Figura 18- Compilação no Software Wincupl



```

File Edit View Options Run Utilities Window Help
C:\USERS\NICOL\DOWNLOADS\ROBO.PLD
Name      Robo ;
PartNo    00 ;
Date      31/10/2023 ;
Revision  01 ;
Designer  Engineer ;
Company   Fatec ;
Assembly  None ;
Location  None;
Device    gl6v8a;

/* ***** INPUT PINS ***** */
PIN 6 = SEL; /*Seletor modo atrativo/repulsivo*/
PIN 1 = LS;  /*Sensor esquerdo */
PIN 2 = RS;  /*Sensor direito */

/* ***** OUTPUT PINS ***** */
PIN 19 = M1F; /*MOTOR 1 FRONT*/
PIN 18 = M1R; /*MOTOR 1 REAR */
PIN 17 = M2F; /*MOTOR 2 FRONT*/
PIN 16 = M2R; /*MOTOR 2 REAR */
PIN 12 = AUX; /*FUNCAO ESPECIAL */

/* ***** LOGIC ***** */
M1F = !SEL & !RS # !LS & !RS # SEL & RS;
M1R = !SEL & RS # SEL & LS & !RS;
M2F = !SEL & !LS & RS # SEL & !RS # SEL & LS;
M2R = !SEL & !RS # !SEL & LS # SEL & !LS & RS;
AUX = !LS & !RS;

```

Fonte: Arquivo pessoal

Após compilar e gravar o firmware de gravação do DLP, basta abrir o ambiente de desenvolvimento do Arduino (IDE) e descompactar o afterburner na pasta releases. Em seguida o arquivo executável do afterburner foi renomeado para “afterburner.exe”, para facilitar o acesso ao comando. O afterburner é um código que permite a programação de chips no processador do Arduino. A fim de habilitar o uso do executável em qualquer diretório, foi necessário copiar o arquivo para a pasta “System32” do Windows.

Seguinte à configuração inicial, é possível conectar o gravador em uma porta USB qualquer e executar o POWERSHELL do Windows, atentando-se a detectar a porta em que o gravador está conectado para poder colocar na linha de comando, pois a porta padrão é setada como COM1.

Enviar o comando “Get-PnpDevice -PresentOnly | Where-Object {\$_. InstanceId -match '^USB'}” no POWERSHELL para saber em qual porta está localizado o gravador. Em seguida é importante configurar a tensão conforme o chip utilizado.

4.1.1 Calibração da Tensão de Gravação

Ao digitar o comando *afterburner*, uma lista dos comandos aparece na tela.

Figura 19- Inicialização do Afterburner

```

Afterburner v.0.4.2 a GAL programming tool for Arduino based programmer
more info: https://github.com/ole00/afterburner
usage: afterburner command(s) [options]
commands: ierwvs
  i : read device info and programming voltage
  r : read fuse map from the GAL chip and display it, -t option must be set
  w : write fuse map, -f and -t options must be set
  v : verify fuse map, -f and -t options must be set
  e : erase the GAL chip, -t options must be set
  s : sets VPP ON to check the programming voltage. Ensure the GAL is NOT inserted.
options:
  -v : verbose mode
  -t <gal_type> : the GAL type. use GAL16V8 GAL20V8 GAL22V10 ATF16V8B ATF22V10B ATF22V10C
  -f <file> : JEDEC fuse map file
  -d <serial_device> : name of the serial device. Default is: COM1
                    serial params are: 38400, 8N1
  -nc : do not check device GAL type before operation: force the GAL type set on command line
  -sec: enable security - protect the chip. Use with 'w' or 'v' commands.
examples:
  afterburner i -t ATF16V8B : reads and prints the device info
  afterburner r -t ATF16V8B : reads the fuse map from the GAL chip and displays it
  afterburner wv -f fuses.jed -t ATF16V8B : reads fuse map from file and writes it to
                    the GAL chip. Does the fuse map verification at the end.
hints:
  - use the 'i' command first to check and set the right programming voltage (VPP)
    of the chip. If the programming voltage is unknown use 10V.
  - known VPP voltages as tested on Afterburner with Arduino UNO:
    Lattice GAL16V8D, GAL22V10D: 12V
    Atmel ATF16V8B, AFT16V8C, ATF22V10C: 10V

```

Fonte: Arquivo pessoal

Para fazer o ajuste da tensão foi programado o comando “afterburner s -d COM3 -t GAL16V8”. O parâmetro “s” ativa a tensão de programação. Com um voltímetro foi medida a tensão de saída do módulo MT3608 e ajustado o trimpot para a tensão do chip. O parâmetro “-d” define qual porta será utilizada, no caso a COM3. O termo “-t” define qual chip está sendo utilizado. Todo o procedimento é feito sem o chip conectado e com um jumper fechando o sinal.

Figura 15- Configuração de Tensão

```

Lattice GAL16V8D, GAL22V10D: 12V
Atmel ATF16V8B, AFT16V8C, ATF22V10C: 10V

```

Fonte: Arquivo pessoal

4.1.2 Etapa de Gravação

Após a remoção do *jumper*, o soquete foi posicionado junto a um fio para alimentar o circuito. O dispositivo está pronto para identificação por meio do comando abaixo: `afterburner i -d COM3 -t GAL16V8`

O parâmetro “i” lê as informações do dispositivo e a tensão de programação. Com o dispositivo reconhecido, agora é só fazer a programação, porém antes todo chip, mesmo novo é necessário apagar ele todo, utilizando o comando:

`afterburner e -d COM3 -t GAL16V8`

Nenhum aviso é dado ao apagar, porém o *prompt* é liberado para inserir novos comandos.

Para programar, é necessário já ter o arquivo de extensão “.jed”, gerado no CUPL e utilizar o comando: `afterburner wv -d COM3 -t GAL16V8 -f xxx.jed`

Com o parâmetro “wv” o gravador grava o código e verifica ao final da gravação. O parâmetro “-f” seleciona o arquivo a ser gravado. Após a gravação nenhum aviso é dado e é necessário o teste no circuito.

5 APLICAÇÃO DIDÁTICA DO CIRCUITO DE APRENDIZAGEM

O estudo da eletrônica digital e portas lógicas faz parte da grade curricular de diversos cursos superiores relacionados a tecnologia, além de muitos cursos técnicos relacionados não só a tecnologia, mas também a elétrica. Os docentes organizam seu material teórico, somado a plataformas de simulação para transmitir e fixar os conceitos estudados. Esta dinâmica apesar de ter uma efetividade minimamente satisfatória, afasta os alunos da realidade das atividades reais encontradas no dia a dia de um profissional da área de eletrônica, resultando em um déficit no aprendizado do aluno, o qual encontrará dificuldade nas atividades a realizar no início de sua carreira profissional. A fim de cobrir esta brecha no sistema de ensino de eletrônica, é proposto um modelo de um circuito de baixo custo, de fácil aquisição e montagem, para que os laboratórios, locais onde se é ensinado eletrônica, possuam um equipamento que permita o ensino prático no mundo físico da eletrônica digital, trabalhando em conjunto com softwares de simulação.

O circuito proposto proporciona aos docentes uma gama de atividades e de aplicações para fixar o entendimento da eletrônica digital, abordando aplicações e situações comuns do dia a dia da indústria, ou até mesmo promovendo competições entre alunos, com o propósito de instigar a criatividade e o desenvolvimento das habilidades na programação de portas lógicas. O robô pode ser configurado de diversas formas diferente, e desempenhando inúmeras funções práticas de aplicação, atendendo as necessidades da prática proposta em aula.

5.1 Aplicações sugeridas

Devido à versatilidade do circuito apresentado, serão propostas especificações de como desenvolver algumas experiências práticas para abordagem em laboratório.

5.1.1 Um robô seguidor de linha que percorre um trajeto estabelecido

Com o intuito de implementar a funcionalidade de um robô seguidor de linha no circuito apresentado, basta criar um trajeto e elaborar uma tabela verdade que expresse a condição em que quando os receptores dos sensores ópticos reflexivos detectem alta frequência em relação

a sua emissão do infravermelho, ele desligue os motores (seus outputs) conforme a tabela verdade abaixo:

Tabela verdade considerando 0 como dentro do trajeto e 1 como fora do trajeto

Figura 16- Configuração de Tensão

SE	SD	M1F	M2F	M1R	M2R
0	0	1	1	1	1
0	1	1	0	1	0
1	0	0	1	0	1
1	1	0	0	0	0

Fonte: Arquivo pessoal

Na tabela acima determinamos que os sensores ópticos "SE" representa o (input) sensor alocado ao lado esquerdo do robô, "SD" representa o (input) sensor alocado ao lado direito do robô, "M1F" representa o (output) motor alocado a parte esquerda do robô deslocando-se ao sentido horário (frente), "M2F" representa o (output) motor alocado a parte direita do robô deslocando-se ao sentido horário (frente), "M1R" representa o (output) motor alocado esquerda do robô deslocando-se ao sentido anti-horário (trás), e "M2R" representa o (output) motor alocado a parte direita do robô deslocando-se ao sentido anti-horário (trás).

Nesta configuração o circuito será capaz de perseguir um tracejado escuro, no qual enquanto o emissor de infravermelho do sensor estiver sobre o tracejado escuro, e conseqüentemente o receptor detecte um baixo nível de frequência, resultando na ativação dos respectivos motores (M1F, M2F, M1R e M2R) do lado que o sensor (SE, SD) está alocado, e ao sair do tracejado, os respectivos motores sejam desligados.

Após construir a tabela verdade é preciso elaborar um código em VHDL ou CUPL que represente o funcionamento descrito na tabela. Exemplo:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity LineFollowerRobot is
    Port ( Sensor1 : in STD_LOGIC;
          Sensor2 : in STD_LOGIC;
          Motor1  : out STD_LOGIC;
          Motor2  : out STD_LOGIC);
end LineFollowerRobot;

architecture Behavioral of LineFollowerRobot is
    signal Velocidade : STD_LOGIC := '1'; -- Você pode ajustar isso conforme necessário

begin
    process(Sensor1, Sensor2)
    begin
        if (Sensor1 = '0' and Sensor2 = '0') then
            -- Ambos os sensores detectam branco
            Motor1 <= Velocidade;
            Motor2 <= Velocidade;
        end if;
    end process;
end Behavioral;
```

```

    elsif (Sensor1 = '1' and Sensor2 = '0') then
        -- Sensor1 detecta preto, Sensor2 detecta branco
        Motor1 <= '0';
        Motor2 <= Velocidade;
    elsif (Sensor1 = '0' and Sensor2 = '1') then
        -- Sensor1 detecta branco, Sensor2 detecta preto
        Motor1 <= Velocidade;
        Motor2 <= '0';
    else
        -- Condição de segurança para qualquer outra combinação de entradas
        Motor1 <= '0';
        Motor2 <= '0';
    end if;
end process;

end Behavioral;

```

Nesse exemplo em VHDL foi utilizada a biblioteca padrão IEEE que possui o mesmo nome do instituto que a criou, IEEE (Institute of Electrical and Electronic Engineers) e que segundo a Codá, Luiza, padroniza as tarefas de documentação, descrição, síntese, simulação, teste e verificação formal da linguagem VHDL. Portanto ao incluir o termo “library IEEE” no início de um arquivo VHDL, está indicando que deseja usar os pacotes e definições padrão fornecidos pela IEEE para VHDL.

Em comparação ao CUPL utilizado no projeto, a linguagem VHDL é mais amplamente utilizada, e conseqüentemente existem mais guias nos registros web para promover a integração e aprendizado da sua aprendizagem, entretanto os softwares para compilar a linguagem VHDL são pagos e menos acessíveis que o já citado anteriormente WinCupl, que compila a linguagem CUPL. De modo geral as duas linguagens podem ser utilizadas, e fica a critério do professor escolher os softwares e as linguagens na hora do planejamento da aula. Vale ressaltar que existem softwares como o VHDL Editor que fazem a conversão entre as linguagens VHDL e CUPL, portanto é indicado que o aluno se habitue em utilizar VHDL, pois a linguagem CUPL é baseada na linguagem VHDL, e as divergências de semântica entre elas podem superadas com a utilização de softwares.

Após desenvolver o programa no software desejado, basta gravar a sua configuração do circuito conforme o explicado no tópico **4 Ambiente de Desenvolvimento e Programação**.

5.1.2 Um robô com o modo repulsivo e atrativo

Com o intuito de implementar a funcionalidade de um robô com dois modos, um que persegue luminosidade, e o outro que se afasta da luminosidade é necessário desenvolver uma tabela verdade com um *input* para efetuar a mudança de modo, e os *outputs* com os dois

comportamentos inversos, o modo que persegue a luminosidade é bastante similar a experiência apresentada anteriormente, conforme a tabela verdade abaixo:

Figura 22- Tabela Verdade Robô Repulsivo e Atrativo

SEL	LS	RS	M1F	M1R	M2F	M2R
0	0	0	1	0	0	1
0	0	1	0	1	1	0
0	1	0	1	0	0	1
0	1	1	0	1	0	1
1	0	0	1	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	1	0
1	1	1	1	0	1	0

Fonte: Arquivo pessoal.

Na tabela acima determinamos que os sensores ópticos "SE" representa o (input) sensor alocado ao lado esquerdo do robô, "SD" representa o (input) sensor alocado ao lado direito do robô, "M1F" representa o (output) motor alocado a parte esquerda do robô deslocando-se ao sentido horário (frente), "M2F" representa o (output) motor alocado a parte direita do robô deslocando-se ao sentido horário (frente), "M1R" representa o (output) motor alocado esquerda do robô deslocando-se ao sentido anti-horário (trás), e "M2R" representa o (output) motor alocado a parte direita do robô deslocando-se ao sentido anti-horário (trás).

Nesta configuração o circuito será capaz de adotar 2 modos de operação, e variar entre perseguir e afastar-se da luminosidade.

Após construir a tabela verdade é preciso elaborar um código em VHDL ou CUPL que represente o funcionamento descrito na tabela. Exemplo:

```
Name      Robo ;
PartNo    00 ;
Date      31/10/2023 ;
Revision  01 ;
Designer  Engineer ;
Company   Fatec ;
Assembly  None ;
Location  None;
Device    g16v8a;

/* ***** INPUT PINS *****/
PIN 6 = SEL; /*Seletor modo atrativo/repulsivo*/
PIN 1 = LS;  /*Sensor esquerdo */
PIN 2 = RS;  /*Sensor direito */

/* ***** OUTPUT PINS *****/
PIN 19 = M1F; /*MOTOR 1 FRONT*/
PIN 18 = M1R; /*MOTOR 1 REAR */
PIN 17 = M2F; /*MOTOR 2 FRONT*/
PIN 16 = M2R; /*MOTOR 2 REAR */
PIN 12 = AUX; /*FUNCAO ESPECIAL */
```

```
/* ***** LOGIC ***** */  
M1F = !SEL & !RS # !LS & !RS # SEL & RS;  
M1R = !SEL & RS # SEL & LS & !RS;  
M2F = !SEL & !LS & RS # SEL & !RS # SEL & LS;  
M2R = !SEL & !RS # !SEL & LS # SEL & !LS & RS;  
AUX = !LS & !RS;
```

Nesse exemplo em CUPL, o software WinCupl foi utilizado para compilar o programa, declara as informações de instalação e nome do respectivo programa que está sendo desenvolvido, depois é declarado os pinos de input e output, e por fim é desenvolvida a lógica booleana que posteriormente será gravada no circuito conforme o explicado no tópico 4 Ambiente de Desenvolvimento e Programação.

6 ANÁLISES E RESULTADOS

Com o aprendizado e os desafios encontrados durante a confecção e configuração do B-DIRO, ficou mais evidente ainda os grandes ganhos em trabalhar com projetos verdadeiramente tangíveis, a experiência proporcionada em desenvolver projetos com ferramentas físicas promovem um aprendizado de DLPs mais amplo. O projeto B-DIRO possibilita que com poucos recursos, seja possível deslumbrar o resultado dos seus esforços bem diante dos seus olhos, ou seja, aumenta a interatividade do aluno com a experiência desenvolvida em sala de aula, o que muitas vezes o motiva a aprender mais.

Desempenhando atividades práticas foi possível constatar que para o estudo de DLPs, distanciar-se de práticas em simuladores é essencial para fixar o conhecimento adquirido em aula. Portanto trazer ao grupo estudantil experiências em laboratório com ferramentas físicas, aproximando-os ao máximo do ambiente encontrado no mercado de trabalho em primeira instância mostra-se um grande desafio, entretanto, com as ferramentas certas alcançamos o almejado sem muitos sacrifícios. O modelo de robô desenvolvido mostrou-se uma ferramenta bem eficiente em viabilizar os estudos de eletrônica digital de forma prática e física, tendo como seu único limitador a aquisição dos softwares de compilação de VHDL, os quais grande parte deles são pagos, todavia, a possibilidade de utilizar CUPL e o software compilador WinCupl, que é totalmente acessível, permite que todas as barreiras para alcançar uma rotina de aulas com estudos práticos em ferramentas físicas sejam superadas.

REFERÊNCIAS

- CODÁ, L. M. R.; Dispositivos Lógicos Programáveis. Departamento de Engenharia Elétrica e de Computação. EESC – USP, São Paulo. Disponível em: https://edisciplinas.usp.br/pluginfile.php/5512480/mod_resource/content/2/DISPOSITIVOS%20L%C3%93GICOS%20PROGRAM%C3%81VEIS_2014.pdf. Acesso em: 28/11/2023.
- FARIA, R.M.; MELCHER, E.U.K.; TURNELLI, M.F.Q.V. Ensinando a organização de computadores e sistemas digitais usando hardware reconfigurável e baixo custo. Universidade Federal de Campina Grande, São Paulo, março de 2003. Disponível: <https://copec.eu/congresses/icece2003/proc/pdf/T141.pdf>. Acesso em: 03/11/2023.
- LOZADA, J. C. H.; ROBLES, J.C.G. Tecnologia de dispositivos programáveis. **Artigo CIDETEC IPN**, Disponível em: https://embebidos-cidetec.com.mx/profesores/jcrls/doctos/tec_programables_cidetec.pdf. Acesso em: 01/07/2023.
- MONTENEGRO, L. P.. Controle digital de velocidade de um motor CC de baixa potência, utilizando arduino. Universidade Federal do Ceará, Fortaleza, Ceará, 2021. Disponível em: https://repositorio.ufc.br/bitstream/riufc/62150/3/2021_tcc_lpmontenegro.pdf. Acesso em: 08/11/2023.
- PEDRONI, V.A. **Circuit Design With VHDL**. Massachusetts Institute of Technology, Massachusetts, 2020. Disponível em: ISBN 9780262042642 Acesso em: 01/07/2023.
- PEREIRA, L.L.S.; GONÇALVES, W.I.O. Estudo da automação e controle aplicados em estacionamentos. Universidade de Taubaté, Taubaté, 2017. Disponível em: <http://repositorio.unitau.br/jspui/bitstream/20.500.11874/5403/1/Lays%20L.%20da%20Silva%20Pereira%20e%20Weslei%20I.%20de%20O.%20Goncalves.pdf>. Acesso em: 08/11/2023.
- SANTOS, A.T.; CAPORALI, A.S.. Sistemas instrumentados de segurança implementados em dispositivos lógicos reconfiguráveis: uma revisão sistemática. *Manhuaçu*, v.19, n.2, p. 584 – 596, maio – setembro, 2021. Disponível em: <https://www.pensaracademico.unifacig.edu.br/index.php/pensaracademico/article/view/2437/1997>. Acesso em: 01/07/2023.
- TOCCI, R. J.; WIDMER, N. S.; MOSS, G.L.. *Sistemas Digitais princípios e aplicações*., Pearson Education, São Paulo, 2017, 11° ed.. Disponível em: https://d1wqtxts1xzle7.cloudfront.net/64388384/Tocci_Sistemas_Digitais_11_edlibre.pdf?1599635966=&response-contentdisposition=inline%3B+filename%3DTocci_Sistemas_Digitais_11_ed.pdf&Expires=1

701099544&Signature=Nh9XkaBkzKhIkMyvpsE9U9j9JKLwt3PDTRLygWQzY3qqPgmw7
 TecANiZRu9e4z6Cy1bn5kai79AvWW2lL6hhLZTdWGbfgbDNooNsUmCDLBlhQfUnLDbc
 CTFGxPII~yzUcdscUzTNjEtyWKC76gtWNzds7d6mmRScNBssX6jO/Ld2~qkbs~f~387XLz
 br9UqQ3mmm560LbFjDgDvWTUMnXeJMqrUab5sQjf6C88gyQwF8WcALHVmvjJnQ~t3
 D9PwA9K3Dj0AJirb0I01ZPHEPmhvJAxyiVG23n6D4wbPICIyr5qavfngR1pPvhG7uPGX~
 X6sI75gMVvj4kkN7K6g__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA. Acesso em:
 01/07/2023. ISBN 978-85-4300-694-9.

TRIANA, M. A. P.. Controle Baseado em comportamento aplicado a robótica móvel, udando
 hardware reconfigurável. Atmel. ATF16V8B, ATF16V8BQ*, and ATF16V8BQL. 2014.
 Disponível em: Atmel-0364-PLD-ATF16V8B-8BQ-8BQL-Datasheet.pdf. Acesso em:
 01/07/2023.

WENDLING, M.. Sensores. **Universidade Estadual Paulista**. Campus Guaratinguetá.
 Guaratinguetá, 2010. Disponível em: https://d1wqtxts1xzle7.cloudfront.net/55868161/4---sensores-v2.0-libre.pdf?1519275237=&response-content-disposition=inline%3B+filename%3Dunesp_UNIVERSIDADE_ESTADUAL_PAULISTA.pdf&Expires=1701226311&Signature=DIKCG541zMtl~zncF2MR3uhi1SuoZ3yHAMfVV5zsBnZILbUEpMKuVBUqhd3py5Wo6zaS9afCr435Eoj5yaQ1bUhCIoMLoQun-WIgjU08hLbNDzCdUDmt-CzqZgTJJs-xCH4JyjciHUZG2kgyq3RUdCnUu8VTn4-Wi6cRN4rs1hXWtB~iomAY-2fmnix~fNXv1i4GiJKnDoEhaH6cHSITdgJnBJIgzycYHfoMn0haKRRosevZZjitPAHFK9nEPNmCrGGosA-sOb5Rw7jzfttelan2Ai1VMGYI-QZE0i~XqooJ3qDAXgAxFThUI6NzhDZTIOHkk-19AtCJ6Usaij-UQ__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA Acesso em: 08/11/2023.