

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO

JULIANA MARTINS PEREIRA
KAREN D. PARENTE RIBEIRO
MISAEAL ROSA DA COSTA

ROBÔ SEGUIDOR DE LINHA COM DETECÇÃO DE OBSTÁCULOS

CURITIBA
2019

JULIANA MARTINS PEREIRA
KAREN D. PARENTE RIBEIRO
MISAEAL ROSA DA COSTA

ROBÔ SEGUIDOR DE LINHA COM DETECÇÃO DE OBSTÁCULOS

Relatório Final elaborado na disciplina Oficina de Integração 1 do curso de Engenharia de Computação, ofertado pelos Departamento Acadêmico de Informática e Departamento Acadêmico de Eletrônica no Campus Curitiba da Universidade Tecnológica Federal do Paraná.

Orientadores: Gustavo Benvenuto Borba e Ronnier Frates Rohrich

CURITIBA
2019

SUMÁRIO

1	INTRODUÇÃO	3
1.1	MOTIVAÇÃO	3
1.2	TRABALHOS RELACIONADOS	4
1.3	OBJETIVOS	4
2	MATERIAIS	5
3	HARDWARE	6
3.1	SENSORES, MOTORES E MÓDULO UTILIZADOS	6
3.1.1	Sensor Óptico Reflexivo TCRT5000	6
3.1.2	Sensor Ultrassônico HC-SR04	6
3.1.3	Módulo Driver Ponte H L298N	7
3.1.4	Motor DC	7
3.1.5	Arduíno Uno R3	8
3.2	ESQUEMÁTICO DO CIRCUITO	8
4	SOFTWARE	10
4.1	DETECÇÃO DE OBSTÁCULOS	10
4.2	DESVIO DE OBSTÁCULOS	10
4.3	DETECÇÃO DE LINHA	11
4.4	CONTROLE DE DIREÇÃO E VELOCIDADE	12
4.5	MELHORIAS E REFINAMENTO	13
4.5.1	Vantagens e desvantagens da lógica Fuzzy	13
5	RESULTADOS	15
5.1	CÁLCULOS	16
5.1.1	Ultrassônico	16
5.1.2	Velocidade	16
6	CONCLUSÃO	18

1 INTRODUÇÃO

Nos dias atuais, cada vez se faz mais presente o uso de microprocessadores e microcontroladores. Com a pauta dos veículos autônomos em voga no mercado, resolveu-se, por meio deste trabalho, integrar de forma primitiva esses recursos a fim de desenvolver um robô seguidor de linha com detecção e desvio de obstáculos. Como já sugere o nome auto-explicativo, a proposta do trabalho consiste em um robô composto principalmente por um chassi de acrílico que anda sobre uma linha preta, realizando curvas e desviando de obstáculos ao detectá-los.

Para tal, foram utilizadas duas placas de Arduino Uno, contendo um microcontrolador cada, além de sensores de refletância e ultrassônico. Por fim, para realizar o controle de velocidade e direção, foi utilizado um driver ponte H em conjunto com quatro motores PWM. Mais informações a respeito do Hardware serão apresentados no Capítulo 2.

1.1 MOTIVAÇÃO

Cada vez mais, dispositivos utilizam sensores para monitorar o ambiente e, com os dados obtidos, gerar uma resposta para determinadas situações. Por fazer uso de dois tipos diferentes de sensores, é possível analisar as diferentes formas de aplicação do projeto sob o viés de cada um deles.

O primeiro diz respeito ao sensor de infravermelho, que detecta a luminescência refletida (ou não) por elementos ao seu redor. Essa tecnologia é a empregada pra fazê-lo andar sobre uma linha preta (detectando a absorção de luz pela fita). Entretanto, é possível fazer uso desse sensor, ainda, em campos como a medicina, por exemplo. Nesse caso, ele pode ser utilizado para detectar a radiação térmica emitida pela pele humana (FLESCHE, 2019), possibilitando a apuração de doenças e o melhor estudo do metabolismo humano através do mapeamento de sua temperatura. Outra aplicação desse mesmo sensor é na detecção de colisões em trens (CHILTON, 2019), por exemplo, emitindo luz que deverá ser refletida de volta caso haja um objeto próximo. Assim, o condutor pode ser avisado acerca de objetos no meio do percurso, a segurança é redobrada em cruzamentos e ainda pode-se verificar a temperatura dos trilhos (prevenindo seu superaquecimento), de acordo com a lei de Boltzmann.

Já o sensor de ultrassom atua de maneira análoga ao sensor de infravermelho, porém emitindo ondas sonoras ao invés de eletromagnéticas. Por isso, além de possuir utilidade na detecção de colisões, pode ser utilizado em outros campos que independam da emissão de luz. Pode ser utilizado para calcular distâncias até elementos transparentes, por exemplo, como vidros e líquidos. Há projetos de cidades inteligentes que utilizam sensores ultrassônicos para avaliar o acúmulo de lixo em lixeiras. Isso permite que elas só sejam esvaziadas quando estiverem cheias (isto é, a distância da tampa - onde o sensor é colocado - até o corpo detectado for a menor

possível), poupando funcionários e até combustível. Outra aplicação desse sensor é, também, na medicina. Os aparelhos de “ultrassom” são capazes de realizar exames não invasivos apenas através do mapeamento dos órgãos internos por meio de ondas sonoras emitidas e recebidas.

Portanto, devido às múltiplas possibilidades de aplicação dos sensores utilizados, decidiu-se implementar algumas delas neste projeto. Entretanto, elas podem ser estendidas para outros campos como o da saúde, indústria e inteligência artificial.

1.2 TRABALHOS RELACIONADOS

É reconhecido que houve projetos semelhantes que ajudaram a embasar e serviram de motivação para o desenvolvimento desse. A plataforma online Robo Core (ROBOCORE, 2019) disponibiliza um tutorial para a montagem do robô, além de um conjunto de códigos para a detecção de linha. Eles permitiram a melhor compreensão da lógica a ser empregada, mas não foram diretamente aplicados pois os componentes utilizados não eram os mesmos. Outro site consultado foi o Filipe Flop (FILIPEFLOP, 2019) que conta com um tutorial para a construção do robô e também com diversos outros para a identificação da linha, o controle da velocidade e a detecção de obstáculos.

Entretanto, não foi observada uma grande quantidade de projetos que integrassem as duas tecnologias de acompanhamento de linha e identificação de obstáculos. Portanto, esse se configura como um dos pioneiros nesse quesito.

1.3 OBJETIVOS

Os objetivos deste trabalho foram integrar os conhecimentos já adquiridos com conhecimentos a serem obtidos. De forma que, ao final, tenha-se um robô seguidor de linha e que desvia de obstáculos.

2 MATERIAIS

A Tabela 1 contém todos os materiais utilizados e seus custos, assim como o custo total do projeto.

Tabela 1 – Custos

Quantidade	Peça	Preço (total)
1	Kit Chassi 4WD	R\$ 105,00
2	Placa Arduino UNO R3	R\$ 109,80
1	Módulo Ponte H L298N	R\$ 25,50
3	Sensor de refletância	R\$ 10,50
1	Sensor Ultrasônico HC SR04	R\$ 12,00
8	Pilhas	R\$ 11,00
5	Resistores	R\$ 2,50
2	Placas perfuradas	R\$ 10,80
1	Kit Porcas, parafusos e extensores	R\$ 12,90
1	Kit Jumpers	R\$ 25,00
1	Cartolina	R\$ 7,20
1	Ferro de solda	R\$ 15,03
1	Tubo de Solda Estanho	R\$ 6,99
2	Protoboard	R\$ 60,00
1	Fita isolante	R\$ 3,00
Total		R\$ 417,22

3 HARDWARE

O Robô desenvolvido no projeto é composto por um Módulo Driver Ponte H L298N (FILIPEFLOP, 2019a), que controla 4 motores DC aos pares, ou seja, os motores da esquerda recebem as mesmas "ordens"; os da direita comportam-se da mesma forma. Além disso, fazem parte do robô 3 Sensores Ópticos Reflexivos TCRT5000 (MASTERWALKER, 2019) cuja função é fazer a verificação da linha. Foi usado também um Sensor Ultrassônico HC-SR04 (FILIPEFLOP, 2019c) cujo objetivo é detectar obstáculos que estejam em frente ao robô. O gerenciamento dos sensores e controle da velocidade, direção, movimento do robô é realizado por meio de dois arduínos UNO R3 (BAUDAELETRONICA, 2019).

3.1 SENSORES, MOTORES E MÓDULO UTILIZADOS

3.1.1 Sensor Óptico Reflexivo TCRT5000

Este sensor opera por meio de um sistema de reflexão infravermelho, onde um LED emissor IR e um fototransistor IR ficam lado a lado. Quando o sensor é aproximado de um objeto, a luz infravermelha do LED emissor é refletida e ativa o fototransistor que gera uma saída. A tensão da saída varia conforme a cor do objeto. Quando esse sensor é lido como sinal digital, cores mais claras geram sinal baixo e cores escuras, como o preto, geram sinal alto. O sensor TCRT5000 pode ser visto na figura 1.



Figura 1 – Sensor Óptico Reflexivo TCRT5000.

Fonte – (MASTERWALKER, 2019)

3.1.2 Sensor Ultrassônico HC-SR04

Para realizar a medição de distância, é necessário deixar o pino Trigger em nível alto por 10 microssegundos. Dessa forma o sensor emite uma onda, que, quando encontra um obstáculo, rebate e volta em direção ao sensor. Entre o tempo que o sinal é emitido e é recebido

de volta, o pino Echo fica em nível alto. Com isso é possível calcular a distância que a onda percorreu até a voltar, visto que a mesma possui a velocidade do som e o tempo que ela demorou pra voltar foi registrado. O Sensor Ultrassônico HC-SR04 pode ser visualizado na figura 2.



Figura 2 – Sensor Ultrassônico HC-SR04.

Fonte – (FILIPEFLOP, 2019c)

3.1.3 Módulo Driver Ponte H L298N

Este módulo recebe uma tensão de até 12V e a regula para fazer o controle dos motores DC. O mesmo possui 4 pinos de entrada, dos quais dois a dois controlam a direção de rotação dos motores da direita e da esquerda. Além disso apresenta dois pinos cuja função é o controle da velocidade dos motores da esquerda e da direita. O Módulo Driver Ponte H L298N possui ainda uma saída de 5V, usada nesse projeto para alimentar os arduínos e demais componentes usados. O módulo acima citado está representado na figura 3.

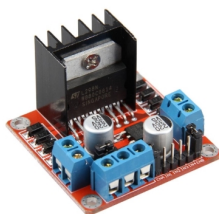


Figura 3 – Módulo Driver Ponte H L298N.

Fonte – (FILIPEFLOP, 2019a)

3.1.4 Motor DC

Os motores DC usados neste projeto operam numa tensão de 3 a 6V, possuem uma caixa de redução com redução 1:48, além de apresentar eixo duplo. O motor DC usado no projeto pode ser visualizado na figura 4.



Figura 4 – Motor DC 3-6V com caixa de redução.

Fonte – (FILIPEFLOP, 2019b)

3.1.5 Arduíno Uno R3

O Arduíno UNO R3 é baseado no microcontrolador ATmega328. Possui 14 pinos que podem servir como entrada ou saída digital, destes 6 podem ser usados para saídas do tipo PWM. Além disso, o mesmo possui 6 entradas analógicas. Ele tem seu software carregado via USB e pode ser usado com uma bateria, ligado a uma fonte ou diretamente ligado na entrada USB. Por ser uma placa com produção "open-source" existem várias versões da mesma placa, Uma delas pode ser vista na figura 5.



Figura 5 – Arduíno Uno R3.

Fonte – (BAUDAELETRONICA, 2019)

3.2 ESQUEMÁTICO DO CIRCUITO

Na imagem a seguir (figura 6) está exposto o esquemático do circuito que foi desenvolvido e montado para este projeto.

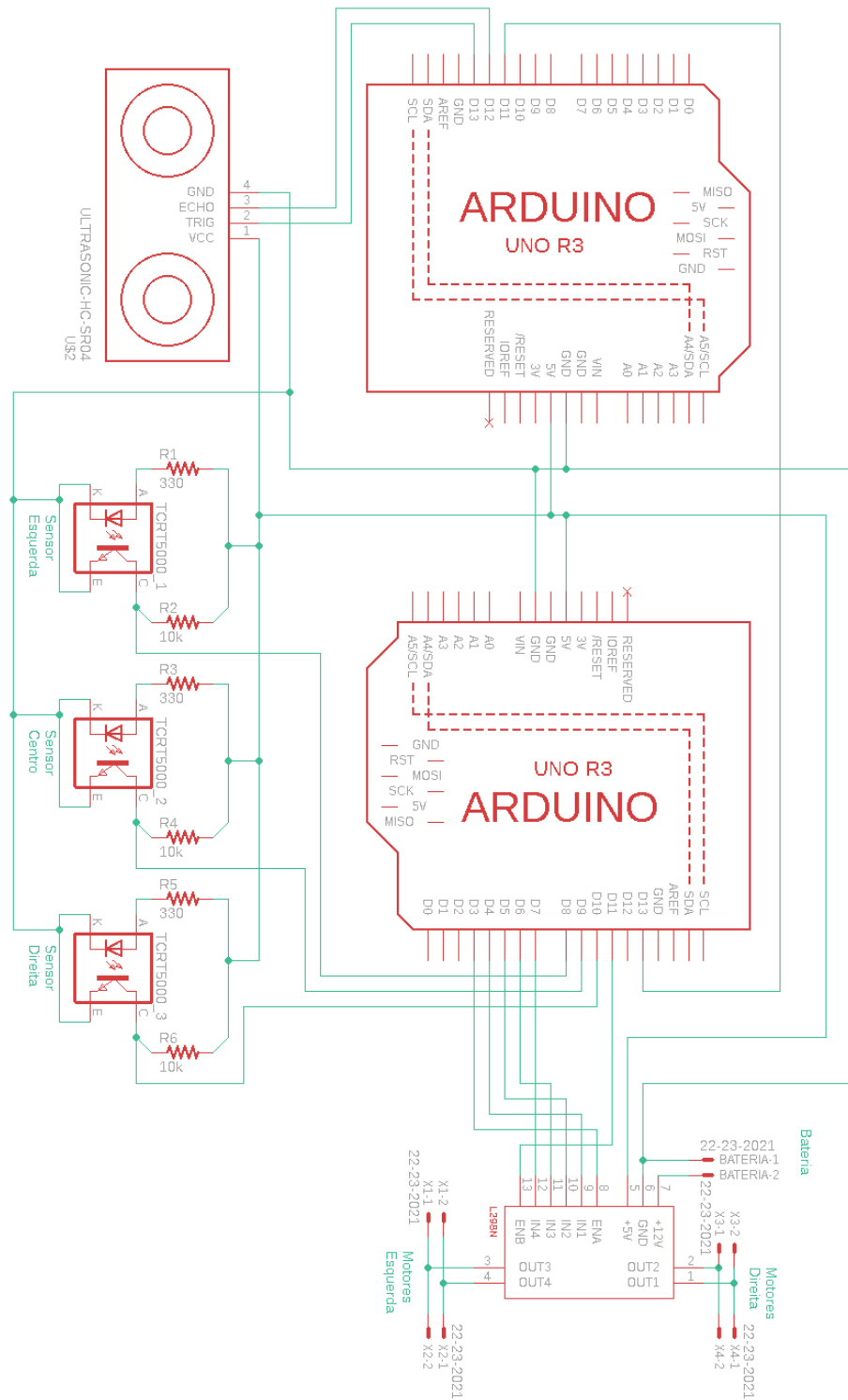


Figura 6 – Esquemático do circuito montado.

Fonte – Autoria Própria

4 SOFTWARE

Por fazer uso de duas placas Arduino UNO, foi necessária a implementação de dois programas. Assim, modularizou-se o controle de cada tipo do sensor, sendo que uma placa ficou encarregada dos sinais enviados pelo sensor ultrassônico (Placa secundária ou *master*) e a outra, dos sensores de refletância (Placa principal ou *slave*), além da tomada de decisões a partir do sinal enviado pela primeira placa.

Em um primeiro momento, tentou-se utilizar a comunicação serial entre ambas, mas a comunicação serial de um dos Arduinos com o computador acabava por interferir nesse processo e vice-versa. Então, optou-se por fazer com que a placa que detecta obstáculos envie um pulso de tensão para a placa principal. Dessa forma, se é detectado um objeto a menos de 20 cm., a tensão na saída da primeira é alterada para HIGH STATE e o programa principal executa as funções referentes ao desvio de obstáculos.

4.1 DETECÇÃO DE OBSTÁCULOS

Quanto ao primeiro programa (*master*), é realizado por ele o controle e processamento dos sinais emitidos e enviados, respectivamente, pelo sensor HC-SR04. Para tal, foi utilizada a biblioteca “Ultrassonico.h”, disponibilizada pela plataforma Filipe Flop (FILIPEFLOP, 2019c), como base para a construção de um algoritmo próprio. Ele foi feito à luz dos princípios da programação orientada a objetos, onde é criada uma classe para o gerenciamento do sensor. Nela, a função *distance()* é responsável por emitir através do pino *trigger* um sinal alto de tensão e converter o tempo em que o sinal retorna ao sensor para uma distância, calculada em centímetros. Então, se a distância identificada é menor do que 20 cm., é enviado um outro sinal, porém dessa vez para a placa principal, indicando que há um obstáculo próximo.

4.2 DESVIO DE OBSTÁCULOS

Já na placa principal, encontra-se um loop em que, primeiramente, é identificado se há um obstáculo próximo ou não. Em caso de afirmativa, trata-se de desviar do obstáculo com base em uma linha a ser seguida. Nessa situação, existe no trajeto pré-definido uma linha que segue em direção ao objeto e uma outra rota alternativa, que desvia dele. Cabe ao robô, portanto, girar para a direita até que essa última seja detectada.

Caso não exista um objeto do qual desviar, o programa entra no caso de *else*, em que apenas deverá seguir a linha. Abaixo, a maneira como isso é feito é melhor detalhada. Foi utilizado como base para o algoritmo final de detecção de linha e controle de velocidade dos

motores o código disponibilizado pela plataforma Robo Core (ROBOCORE, 2019).

4.3 DETECÇÃO DE LINHA

Para que o robô possa seguir a linha, foram instalados três sensores de refletância, já detalhados anteriormente. Portanto, a resposta deles sobre a luminosidade é um pico de tensão (leitura digital igual a 1) caso se encontrem na ausência de luz e uma tensão mínima (leitura digital igual a 0) quando há luz sendo refletida de volta. Isto é, quando o sensor está sobre a linha preta, não há luz sendo refletida, portanto a resposta da leitura é 1. Já quando ele está fora dela, a resposta é 0. Dessa forma, quando ambos os sensores da esquerda e da direita estiverem fora da linha, o carrinho deverá andar para frente. Já quando o sensor da esquerda estiver sobre a linha e o da direita, fora, ele deverá virar para esquerda. O contrário ocorre quando o sensor da direita estiver sobre a linha e o da esquerda, fora dela (Figura 7).

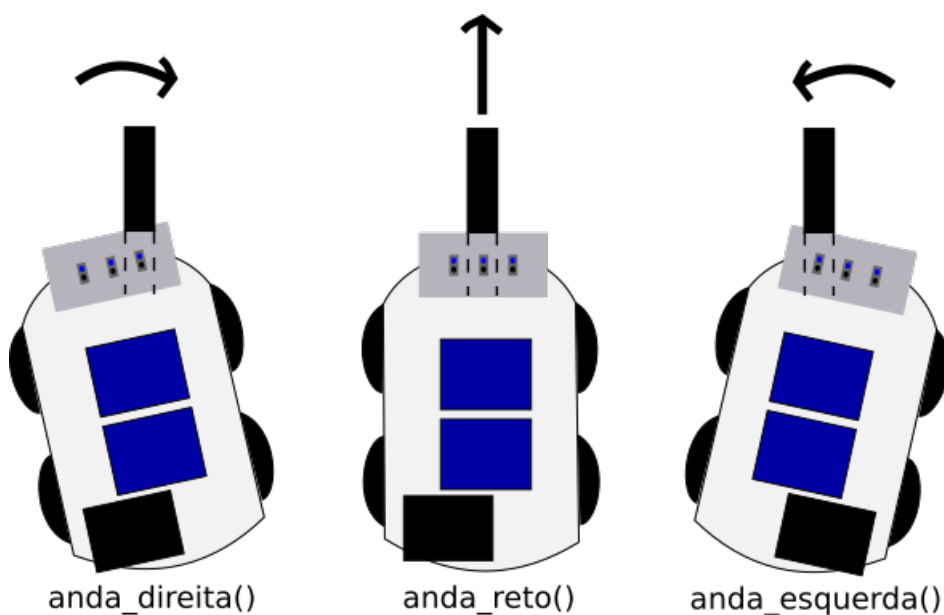


Figura 7 – Detecção e acompanhamento da linha.

Fonte – Autoria própria

Para essas três direções em que o carrinho deverá andar, reto, para a esquerda e para a direita, respectivamente, foram criadas três funções (`anda_frente()`, `anda_esquerda()` e `anda_direita()`). O que permite essa mudança de direção e sentido do movimento, por sua vez, é o controle de velocidade dos motores, explicitado a seguir.

4.4 CONTROLE DE DIREÇÃO E VELOCIDADE

Para possibilitar a movimentação do robô, foram utilizados quatro motores (para quatro rodas). Entretanto, o controle de velocidade e de direção é realizado dois a dois, um para os da direita e outro para os da esquerda. Neste caso, o lado direito é designado M1 e o esquerdo, M2. Nos pinos de controle de direção, as saídas dependem da tensão gerada em duas das portas de cada motor. As combinações de tensão (LOW, LOW) e (HIGH, HIGH) levam ao travamento das rodas. Já a (HIGH, LOW) faz com que se movimentem para frente e (LOW, HIGH), para trás.

O controle de velocidade, por sua vez, é realizado com base em um sinal PWM (Pulse with Modulation). Essa técnica permite simular uma saída de tensão analógica através de pulsos de tensão digitais (isto é, que possuem apenas um valor mínimo e máximo). Assim, mantendo-se a mesma frequência de pulsos e variando apenas a duração de cada um deles, é possível modular a tensão média, como ilustra a Figura 8. A função *analogWrite()* possui como parâmetros a porta de saída e um valor que varia de 0 a 255, fazendo com que essa tensão varie de 0 V a 5 V. Esse valor de tensão chega, então aos motores, que se movimentam mais ou menos rapidamente de acordo com ele.

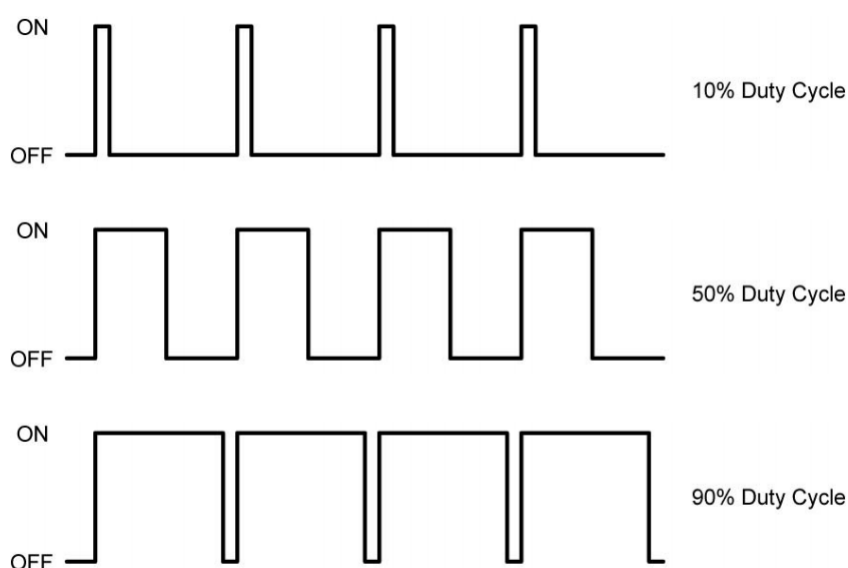


Figura 8 – Princípios do sinal PWM.

Fonte – (MECAWEB, 2019)

Desse modo, a velocidade normal de percurso é de 200, o que gera uma tensão esperada de 3,9 V. Porém, ao realizar o desvio de obstáculos, no momento em que o robô procura a linha alternativa, ela é reduzida pela metade, valendo 100 (1,95 V). Para realizar as curvas à direita, inverte-se o sentido de rotação dos motores da direita. Já para virar à esquerda, inverte-se o sentido dos motores da esquerda. Em um primeiro momento, optou-se apenas por reduzir

a velocidade dos motores (ao invés de alterar o sentido do giro) quando estivessem realizando uma curva. Entretanto, verificou-se que em curvas com ângulos menores, o carrinho acabava perdendo a linha e não a seguia corretamente. Portanto, fez-se necessário modificar o algoritmo que estava sendo usado.

4.5 MELHORIAS E REFINAMENTO

No decorrer do projeto, inúmeras modificações e melhorias foram cogitadas. A primeira delas foi a implementação de uma lógica *Fuzzy*, para a detecção da linha. A premissa dessa lógica é a de que as variáveis assumam valores que variem de 0 a 1, ao contrário da lógica clássica booleana muito aplicada na computação (NETO, 2014). Neste caso, mais do que trabalhar com valores numéricos, é possível trabalhar com conceitos difusos, como quando o robô estivesse "parcialmente" sobre a linha. Isso, somado a um indicativo de que estivesse "um pouco" rotacionado à esquerda, levaria a um giro "levemente" à direita, por exemplo.

4.5.1 Vantagens e desvantagens da lógica Fuzzy

Abaixo, estão enumeradas algumas das maiores vantagens e desvantagens encontradas na implementação dessa proposta.

Vantagens:

- Permite uma análise de todas as combinações possíveis entre os *inputs* dos três sensores.
- Torna o código sintaticamente mais compreensível.
- Permite especificações parciais (sem "arredondamentos").

Desvantagens:

- Implementação extensa.
- Especificações podem tornar-se subjetivas.
- Não reduz os casos condicionais.

Apesar de ser uma abordagem bastante interessante do problema, a lógica não foi implantada no programa. Alguns códigos exemplificando esse uso foram estudados e levado em consideração, sobre os quais pode-se constatar que seria um artifício demasiadamente extenso e trabalhoso para a produção de um resultado muito semelhante ao que se obteve sem ele. Os casos de condições, por exemplo, teriam de ser igualmente declarados no programa, além de a definição de cada função de pertinência. Em nome da otimização de tempo, de memória e de processamento, portanto, a lógica difusa não foi diretamente aplicada.

Outra melhoria proposta, mas também não adotada diz respeito ao processamento em

paralelo dos sinais vindos do sensor ultrassônico e dos de refletância. Para evitar que o envio do sinal de ultrassom interferisse na detecção da linha, tentou-se utilizar o Paradigma Orientado a Notificações (??), muito utilizado também no controle dos robôs desenvolvidos pelo Laboratório Avançado de Sistemas Embarcados ((LASER, 2016)).

Por meio desse paradigma, ocorreria o processamento dos sensores de refletância na placa principal aliado à avaliação lógica do sinal emitido pelo sensor ultrassônico. Essa premissa seria notificada caso fosse verificado que a distância até um objeto fosse menor do que 20 centímetros. E em seguida, seria ativada a regra para o desvio desse obstáculo. Entretanto, por falta de tempo e de uma pesquisa mais rigorosa, não foi possível adotar essa outra lógica.

5 RESULTADOS

Durante o desenvolvimento do projeto, foi necessário a realização de testes e cálculos para se determinar certos parâmetros. Como, por exemplo, o calculo da velocidade ideal, o que em certas aulas foi um problema a ser enfrentado.

O principal problema a ser resolvido era encontrar uma velocidade que fosse capaz de vencer o atrito, ou seja, que os motores tivessem força o suficiente para que o carrinho andasse. Mas que também fosse uma velocidade ideal para a detecção da linha preta e, conseqüentemente, a realização de curvas e ajuste do percurso.

Primeiramente, buscou-se através de testes encontrar a velocidade mínima inicial, ou seja, aquela que desse o impulso inicial mínimo para o carrinho andar. Constatou-se que é 70/255, ou seja, 11,36 cm/s. Encontrado este valor, tentou-se setar essa como a velocidade para tanto andar em linha reta quanto realizar as curvas. O carrinho se perdia nas curvas, pois virava devagar demais. Então, a velocidade para fazer as curvas foi aumentada para 120/255. Dessa maneira que o marco 2 foi apresentado, o carrinho seguia a linha e ajeitava, mas dependia muito da angulação em relação a linha.

Depois de mais alguns testes mexendo com as velocidades, e com a ajuda de um dos projetos similares, o (ROBOCORE, 2019), descobriu a velocidade ideal. Na qual a velocidade para andar reto é 200/255, ou seja, 32,47 cm/s, e, para fazer as curvas, um par de rodas gira pra frente e o outro, para trás.

Apos vencido o empecilho da velocidade. Tivemos problemas com o sensor ultrassônico. O código apenas do ultrassom funcionava, porém ao juntar com o código de seguir linha o carrinho perdia a linha. Demorou algum tempo para a equipe perceber que o tempo da emissão e recepção da onda sonora era grande o suficiente para atrapalhar o código de seguir linha. Esse problema foi resolvido adicionando um outro Arduíno que apenas faz a leitura do ultrassom. E, caso a distancia seja inferior a 20 cm manda um sinal para o outro Arduíno que faz a curva para desviar do obstáculo. Por utilizar a biblioteca do sensor ultrassônico, a distancia setada no programa já está em centímetros.

Durante o projeto houve também vários problemas relacionados ao mau contato e jumpers. Várias vezes por conta do transporte, por mais que esse tenha sido feito com cuidado, jumpers quebraram ou saiam do lugar. De forma que perdia-se algum tempo sempre que íamos trabalhar no projeto.

Inicialmente os circuitos foram montados em protoboards, o intuito era posteriormente passar para uma placa perfurada. O que foi feito no meio do projeto, contudo por uma falta de experiencia, a primeira tentativa não ficou boa. Como ainda era necessário trabalhar outros requisitos do projeto. Deixou-se por ultimo a parte de soldar na placa novamente.

Atividade esta que trouxe alguns problemas também. Apesar de da segunda fez a solda ficar menos grossa. O que aparentemente evitava o contato entres componentes que não

deveriam ter contanto. Problema este que ocorreu da primeira vez. Em alguns pontos que aparentemente estavam certos, a solda não tinha aderido a placa. Algumas soldas soltaram por estar muito pouco grudadas. E a falta de experiencia da equipe dificultou a descoberta do problema.

Todavia, o grupo confessa que soldar desde que os sensores já estavam fazendo a leitura correta poderia ter salvado algum tempo, que foi perdido por conta de mau contato.

5.1 CÁLCULOS

5.1.1 Ultrassônico

Utilizando a equação 5.1 descobre-se o tempo gasto para o sinal do ultrassônico voltar.

$$t = (2 * d)/v \quad (5.1)$$

Onde t é o tempo que o sinal demora para voltar pro ultrassônico, v é a velocidade do som (340 m/s) e d é a distancia que o obstáculo se encontra.

Considerando a menor distância do obstaculo que o carrinho irá chegar, ou seja 20 cm, tem-se que o menor tempo gasto é:

$$t = (2 * 20)/340 \quad (5.2)$$

$$t = 0.12s \quad (5.3)$$

Tempo esse grande o suficiente para atrapalhar a detecção da linha.

5.1.2 Velocidade

Na Tabela 2 encontram-se os valores de tempo gasto para se andar 1 m com a velocidade setada em 200/255.

$$v = 1/\Delta t \quad (5.4)$$

Tabela 2 – Tempos

Tempo 1 (s)	2,79
Tempo 2 (s)	3,28
Tempo 3 (s)	3,37
Tempo 4 (s)	2,88
Tempo Médio (s)	3,08

Logo, a velocidade é 32,47 cm/s.

6 CONCLUSÃO

Ao final do projeto, foi possível experienciar inúmeras situações da parte prática do curso de Engenharia de Computação, o que agregou uma vivência bastante enriquecedora. Com a realização do trabalho, foi possível constatar a importância das boas práticas da eletrônica e da programação, especialmente no que tange a organização do projeto. A grande parte dos erros cometidos não se deu na lógica em si do robô, que é relativamente simples, mas em sua execução. Cabos com mau contato, soldas mal feitas e falta de capricho na hora da montagem dos componentes foram as maiores dificuldades e os maiores causadores de erros e danos ao trabalho.

Por outro lado, o aprendizado que pode ser obtido foi imensurável. O fato de ter-se trabalhado com dois tipos de sensores possibilitou um amplo conhecimento acerca das ligações entre eles, deles com as placas e das placas entre si. Os conceitos de *input* e *output*, imprescindíveis para as respostas dadas pelo programa, puderam ser melhor consolidados, por exemplo.

Por fim, a união desses dois sensores permite uma análise bastante rica dos arredores do “robô” a ser construído, mesmo enquanto ocorre sua movimentação. O fato de ele seguir uma linha permite delimitar um “circuito” a ser percorrido, mesmo com a interrupção dele por obstáculos. Isso poderia resultar em uma aplicação industrial, como duas máquinas que precisem se comunicar através de um trajeto pré-definido (e sujeito a descontinuidades). Ou mesmo demonstrar de maneira primitiva a tecnologia empregada em carros autônomos. Em suma, os conhecimentos adquiridos através do desenvolvimento desse projeto são de extrema utilidade em inúmeros campos da saúde, indústria e inteligência artificial. Sendo assim, tende a agregar conceitos e experiências de extrema valia.

REFERÊNCIAS

BAUDAELETRONICA. **Arduíno Uno R3**. 2019. <<http://www.baudaeletronica.com.br/arduino-uno-r3.html>>. Acessado em: 2019-07-07.

CHILTON, A. **Using Defect Sensors to Detect Signaling Problems and Prevent Train Collisions**. 2019. <<https://www.azosensors.com/article.aspx?ArticleID=237>>. Acessado em: 2019-07-06.

FILIPEFLOP. **Como conectar o sensor HC-SR04 ao Arduino**. 2019. <<https://www.filipeflop.com/blog/sensor-ultrassonico-hc-sr04-ao-arduino/>>. Acessado em: 2019-03-30.

_____. **Driver Motor Ponte H baseado no chip L298N**. 2019. <<https://www.filipeflop.com/produto/driver-motor-ponte-h-l298n/>>. Acessado em: 2019-07-07.

_____. **Motor DC 3-6V com Caixa de Redução e Eixo Duplo**. 2019. <<https://www.filipeflop.com/produto/motor-dc-3-6v-com-caixa-de-reducao-e-eixo-duplo/>>. Acessado em: 2019-07-07.

_____. **Sensor de Distância Ultrassônico HC-SR04**. 2019. <<https://www.filipeflop.com/produto/sensor-de-distancia-ultrassonico-hc-sr04/>>. Acessado em: 2019-07-07.

FLESCH, U. **The Application Of Infrared-Sensors In Medicine**. 2019. <<https://www.spiedigitallibrary.org/conference-proceedings-of-spie/0395/0000/The-Application-Of-Infrared-Sensors-In-Medicine/10.1117/12.935204.short?SSO=1>>. Acessado em: 2019-07-06.

LASER. **Laboratório Avançado de Sistemas Embarcados e Robótica**. 2016. <<http://laser.dainf.ct.utfpr.edu.br/doku.php>>. Acessado em: 2019-07-06.

MASTERWALKER. **Como usar com Arduino – Sensor Óptico Reflexivo TCRT5000**. 2019. <<http://blogmasterwalkershop.com.br/arduino/arduino-utilizando-o-sensor-reflexivo-tcrt5000/>>. Acessado em: 2019-07-07.

MECAWEB. **PWM - Modulação Por Largura de Pulso**. 2019. <http://www.mecaweb.com.br/eletronica/content/e_pwm>. Acessado em: 2019-07-06.

NETO, A. de M. **Lógica Fuzzy**. 2014. <<https://www.ime.usp.br/~adao/LOGICAFUZZY2017F.pdf>>. Acessado em: 2019-07-07.

ROBOCORE. **Robô Seguidor de Linha**. 2019. <<https://www.robocore.net/tutorials/robo-seguidor-de-linha.html>>. Acessado em: 2019-03-30.