



Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Departamento de Engenharia Elétrica e Informática

Projeto de instrumentação e controle de um veículo autoguiado

Matheus Martins Monteiro

Campina Grande, PB
Dezembro de 2018

Matheus Martins Monteiro

Projeto de instrumentação e controle de um veículo autoguiado

*Relatório de Estágio apresentado à
Coordenação do Curso de Graduação
em Engenharia Elétrica da Univer-
sidade Federal de Campina Grande,
Campus Campina Grande, como
parte dos requisitos necessários para
a obtenção do grau de Bacharel em
Engenharia Elétrica.*

Área de Concentração: Controle e Automação

Orientador: Rafael Bezerra Correia Lima

Campina Grande, PB
Agosto de 2018

Matheus Martins Monteiro

Projeto de instrumentação e controle de um veículo autoguiado

*Relatório de Estágio apresentado à
Coordenação do Curso de Graduação
em Engenharia Elétrica da Univer-
sidade Federal de Campina Grande,
Campus Campina Grande, como
parte dos requisitos necessários para
a obtenção do grau de Bacharel em
Engenharia Elétrica.*

Aprovado em 20/12/2018

Péricles Rezende Barros

Universidade Federal de Campina Grande
Supervisor

Rafael Bezerra Correia Lima

Universidade Federal de Campina Grande
Orientador

*Sempre que te perguntarem se podes fazer um trabalho,
respondas que sim e te ponhas em seguida a aprender como se faz.*

Agradecimentos

Agradeço primeiramente a Deus, que é a base de tudo. À minha família, que esteve comigo em todos os momentos, sempre me incentivando a correr atrás dos meus objetivos e que nunca mediram esforços para manter-me firme em busca do meu propósito. Aos meus professores, que me ensinaram todo o essencial necessário para eu me tornar um excelente profissional. Em especial, agradeço ao meu orientador Rafael Correia, pela paciência na orientação e incentivo, e ao suporte técnico de Simões Toledo, que tornou possível a conclusão deste trabalho. Agradeço também meus amigos, que me aconselharam a seguir trilhando no caminho certo.

Resumo

Este relatório apresenta as atividades realizadas durante o estágio supervisionado no Laboratório de Instrumentação Eletrônica e Controle (LIEC). Como trabalho principal, foi realizada uma atualização da plataforma didática de um veículo autoguiado, para que seja utilizada futuramente em projetos de controle e automação neste laboratório. Dessa forma, esse trabalho apresenta desde a fundamentação teórica das tecnologias envolvidas, até a solução adotada para a nova configuração do veículo, bem como testes para averiguar os resultados dos estudos no período vigente de estágio. Foram abordados todas as características e tecnologias contidas no veículo, dispostos em vários tópicos, e organizados de forma a dar uma visão ampla dos sistemas básicos para o funcionamento do veículo, como os sistemas de acionamento dos motores, de detecção de obstáculos e de correção de trajetória.

Palavras chave: AGV; Veículo Autoguiado;

Abstract

This report presents the activities performed during a supervised internship held in Electronics and Instrumentation Laboratory Control (LIEC). As major activity, an update was done on the didactic platform of a self-guided vehicle, to be used in future control and automation projects in this laboratory. In this way, this work presents from the theoretical foundation of the technologies involved, to the solution adopted for the new configuration of the vehicle, as well as for the results of the studies performed in the current period. The subjects dealt include all the characteristics and technologies contained in the vehicle, arranged in various topics, and organized in such a way as to give a broad overview of the basic systems for the operation of the vehicle, such as engine drive systems, obstacle detection and correction trajectory.

Keywords: AGV; Self-guided vehicle.

Lista de Figuras

1	Foto da fachada do Laboratório	14
2	AGV - Primeira Versão	17
3	Funcionamento do sensor Ultrassônico	18
4	Funcionamento do Motor CC	19
5	Funcionamento Encoder	20
6	Relação taxa de transmissão x comprimento do barramento CAN.	21
7	Influência dos bits Dominantes e Recessivos nos níveis de tensão no barramento ao longo do tempo.	22
8	Mensagens no padrão CAN 2.0A.	22
9	Mensagens no padrão CAN 2.0b.	23
10	Arquitetura física do AGV.	25
11	Primeira versão do novo Veículo	26
12	Versão Final do AGV	26
13	Microcontroladores utilizados no projeto	27
14	Motor 34:1 Metal 25Dx52L mm HP 6V com Encoder 48 CPR de quadratura	29
15	Módulo Dual LS7366R Quadrature Encoder Buffer	30
16	Shield Confeccionado para Arduino Mega	31
17	Driver Dual VNH2SP30 Motor Driver Carrier MD03A	32
18	Sensor Ultrassônico	33
19	Disposição dos sensores de Ultrassom ao redor do veículo	34
20	Módulo CAN MCP2515	35
21	Modificação feita no módulo CAN para utilização no Microcontrolador Raspberry Pi	36
22	Imagem após processamento	38
23	Interface Gráfica do controle remoto do Raspberry no Computador	39
24	Teste dos sensores de distância	40
25	Teste dos sensores de distância	41
26	Teste dos sensores de distância	42
27	Experimento 01	43
28	Experimento 02	43
29	Experimento 03	44
30	Experimento 04	44
31	Teste de identificação da linha	45
32	Sistema de detecção de obstáculos	49
33	Sistema de leitura dos encoders	50
34	Sistema de acionamento dos motores	51
35	Sistema de comunicação por rede CAN	52

Lista de Tabelas

1	Especificações Gerais do Motor	29
2	Cores dos fios do encoder e suas respectivas funcionalidades	30
3	Especificações Gerais do Driver	33
4	Especificações Gerais do módulo CAN	35

Lista de Abreviaturas e Siglas

A Ampere

ACK Acknowledgment Error Check

AGV Automated Guided Vehicles

CC Corrente Contínua

CRC Cyclic Redundancy Check

CSMA/CD with NDA Carrier Sense Multiple Access / Collision Detection with Non-Destructive Arbitration

DC Direct Current

LIEC Laboratório de Instrumentação Eletrônica e Controle

Mbps Mega Bits per Second

NRZ Non Return to Zero

OpenCV Open Source Computer Vision Library

oz·in Ounce per inch

PWM Pulse Width Modulation

RPM Rotações por Minuto

UFMG Universidade Federal de Campina Grande

V Volts

Sumário

1	Introdução	13
1.1	Justificativa e Problemática	13
1.2	Objetivos	14
1.3	O laboratório de Instrumentação Eletrônica e Controle - LIEC	14
1.4	Organização do Documento	15
2	Revisão Bibliográfica	15
2.1	Veículos Guiados Automaticamente	15
2.1.1	Veículo do LIEC	16
2.2	Intrumentação	17
2.2.1	Sensor de Distância	17
2.2.2	Motor CC	18
2.2.3	Encoder	19
2.3	Rede CAN	20
3	Atividades Realizadas	24
3.1	Arquitetura do AGV	25
3.2	Alimentação do Sistema	28
3.3	Motores de Corrente Contínua e Encoders	28
3.3.1	Bibliotecas	30
3.3.2	Buffer	30
3.4	Driver Para motor	31
3.5	Detecção de Obstáculos	33
3.6	Módulos CAN	34
3.7	Definição da trajetória por Processamento de Imagem	36
3.7.1	Biblioteca OpenCV	36
3.8	Estação de Trabalho: Computador	38
4	Resultados	40
4.1	Detecção de Obstáculos	40
4.2	Acionamento dos Motores	41
4.3	Identificação de Linha	45
4.3.1	Algoritmo de Controle	46
5	Conclusão	47
5.1	Trabalhos Futuros	47
	Apêndice	49

1 Introdução

Esse relatório de estágio supervisionado tem por objetivo expor as tarefas realizadas e os resultados obtidos durante o seu período de vigência, que compreende do dia 24/09/2018 até 26/11/2018, totalizando 182 horas (6 créditos, mínimo 180 horas), dispostos em 20 horas semanais, sob a supervisão do professor Péricles Rezende Barros e orientado pelo professor Rafael Bezerra Correia Lima. O estágio faz parte da grade curricular obrigatória do curso de Engenharia Elétrica da Universidade Federal de Campina Grande e de suma importância para a formação profissional, pois consolida na prática os conhecimentos teóricos adquiridos durante o curso de graduação.

1.1 Justificativa e Problemática

Historicamente, o homem sempre foi o principal vetor para o desenvolvimento de processos artesanais e industriais, sendo usado como força bruta de trabalho para tal. Desta forma, além de todas as alienações e problemas sociais consequentes desse sistema, a queda de produção e eficiência de energia, devido à rusticidade das operações realizadas, impactava negativamente na sociedade e no desenvolvimento da mesma.

Com a revolução industrial, as máquinas assumiram o papel do homem nas tarefas, em especial as de cunho repetitivo, no qual uma certa operação, que não necessitava necessariamente da intervenção humana, deveria ser realizada diversas vezes. Como os equipamentos utilizados eram capazes de realizar este tipo de operação em um curto espaço de tempo, houve por consequência imediata um aumento da produção do produto final que, aliado ao menor custo de operação e a ausência de direitos trabalhistas para o maquinário, ocasionou uma crescente considerável do número e qualidade do maquinário utilizado nas indústrias, que foram substituindo os homens em tarefas específicas, fazendo-os focar apenas nas tarefas intelectuais, inerentes ao ser humano.

De fato, esse foi um divisor de águas na história da humanidade, visto que não somente naquele momento influenciou bastante o estilo de vida como um todo, como até hoje podemos perceber cada vez mais uma ascensão das máquinas e equipamentos de pequeno e grande porte operando nas indústrias como um todo.

Com o desenvolvimento da tecnologia, as tarefas específicas que os dispositivos eram capazes de realizar se expandiram para todos os campos do trabalho humano, desde o desenvolvimento de matérias primas, até os avanços mais recentes da medicina, agregando cada vez mais estes aparelhos no dia-a-dia do homem moderno.

Entre outras aplicações, temos os veículos autoguiados que surgiram para o transporte de produtos diversos no ambiente industrial, sem a necessidade de intervenção humana. Esses veículos são conhecidos como *Automated Guided Vehicles*, ou simplesmente AGV. O mercado global de AGV tem estimativas de alcançar a marca de 7.3 bilhões de dólares até 2025, com uma taxa de crescimento anual de 16.5% [1]. Uma crescente melhoria da segurança nas indústrias, aumento da produção e redução de custos operacionais aparecem como os principais motivos para o crescimento desses veículos no mercado de automação industrial.

Segundo essa motivação, foi desenvolvido em laboratório uma primeira versão de um AGV seguidor de linha, utilizando para isso um sistema de instrumentação básico, porém funcional, com diversos dispositivos e sensores arranjados de tal forma que fosse possível a realização da tarefa desejada, com visualização de dados importantes em um computador que serve como estação de trabalho. Porém, o modelo era uma versão primária, cuja concepção tinha um caráter mais didático.

Devido a isso tecnologias bastante utilizadas no meio industrial foram deixadas em segundo plano, como por exemplo a implementação de comunicação por rede CAN entre os microcontroladores do veículo, dentre outras atualizações no modo de como adquirir informações do ambiente através dos sensores.

1.2 Objetivos

No contexto apresentado, o trabalho teve como objetivo a realização dos itens a seguir:

- Atualização da Plataforma Didática (AGV);
- Substituição do hardware do Wi-Fi;
- Implemento dos módulos CAN;
- Projeto de Sistema de Controle.

Dessa forma, se espera uma melhoria significativa do veículo de estudo, de forma que o mesmo seja utilizado em projetos futuros pelo laboratório.

1.3 O laboratório de Instrumentação Eletrônica e Controle - LIEC

Localizado na Universidade Federal de Campina Grande - UFCG, no campus de Campina Grande, O LIEC é um laboratório do Departamento de Engenharia Elétrica trabalhos nas áreas de Automação, Eletrônica e Controle.



Figura 1: Foto da fachada do Laboratório

Fonte: **O próprio Autor**

Com uma área de aproximadamente 600 m², o LIEC (Figura 1) conta com oito laboratórios de desenvolvimento, duas salas de apoio técnico, sala para apresentação de trabalhos, salas para pós-graduação e professores [2].

Dentre outras atividades, podemos destacar:

- Redes Industriais;
- RFID;
- Ultrassom;
- Aplicações Wireless;
- Automação Industrial;
- Controle e Otimização;
- Instrumentação Eletrônica.

No LIEC, os alunos da UFCG encontram um ambiente adequado para o desenvolvimento de seus arcabouços teóricos em atividades práticas, por meio de pesquisas e outras atividades realizadas no laboratório.

1.4 Organização do Documento

Diante do exposto, a organização do documento é descrita a seguir. No Capítulo 2 temos uma breve revisão sobre o embasamento teórico dos hardwares utilizados, bem como uma abordagem sobre alguns conceitos básicos que a acompanham, dentre outros podemos citar as tecnologias de hardware utilizadas, neste caso os microcontroladores Arduino Mega e Raspberry Pi 3B. Já no Capítulo 3, é apresentada a proposta de solução adotada, com sua arquitetura física e lógica detalhada, apresentando de forma detalhada qual o procedimento adotado para o cumprimento dos objetivos impostos ao trabalho. No Capítulo 4, temos por objetivo mostrar todo o desenvolvimento do projeto, bem como a validação do trabalho como um todo, conrmando que a solução adotada é de fato possível e eciente no que se propõe a fazer.

Por último, temos a conclusão deste trabalho, abordando algumas discussões sobre os resultados obtidos, as dificuldades durante o projeto, bem como as perspectivas futuras acerca desta tecnologia, no contexto de automação industrial.

2 Revisão Bibliográfica

Para a reformulação do veículo autoguiado, inicialmente foi feito um estudo sobre o AGV presente no laboratório, a fim de descobrir o que já estava implementado em termos de dispositivos no veículo, com o intuito de melhorá-los, bem como inserir novas tecnologias.

2.1 Veículos Guiados Automaticamente

Veículos Autoguiados ou AGVs (*Automated Guided Vehicles*), são veículos que funcionam de forma autônoma, ou seja sem interferência humana direta, seguindo um certo padrão pré-programado. Para o controle de rotina desses veículos, é geralmente implementado um sistema

embarcado conectado a alguns sensores, que realizam leituras do ambiente a fim de identificar a trajetória a ser seguida, trajetória essa que pode ser traçada de diferentes maneiras, dependendo de que tipo de sensor e de qual aplicação se deseja executar com veículo. Dentre as mais utilizadas, temos:

- Demarcação por fitas de cor escura, a qual são percebidas pela diferença de contraste com o piso;
- Fitas magnéticas que diferem da anterior, utilizam do campo magnético como guia;
- Cabos que transmitem sinais de Radio Frequência;
- Objetos detectados por meio de lasers.

Dessa maneira, através da identificação do caminho a ser seguido, uma rotina é programada para segui-la, com parâmetros que variam conforme a necessidade. Os sensores, além de identificar o trajeto, também são comumente utilizados para medir a posição relativa do veículo com relação ao caminho a ser seguido, auxiliando a permanência na trajetória.

Geralmente são utilizados em meio industrial para locomoção de cargas e equipamentos, armazenagem e distribuição de produtos, recuperação de objetos, dentre outras aplicações. Como principais vantagens, podemos apontar sua melhora na organização dos processos industriais, bem como a eficiência no transporte de cargas e redução de custos com mão de obra para tal, o que aumenta a produção da indústria como um todo, tornando os processos produtivos cada vez mais eficientes [3].

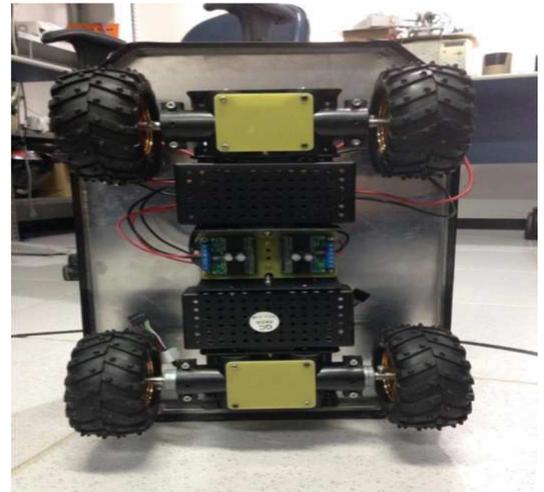
2.1.1 Veículo do LIEC

Devido a importância do veículo para o setor industrial, o Laboratório em cooperação com a então aluna de graduação em Engenharia Elétrica Clarice Barros, desenvolveu em 2016 a primeira versão de um AGV funcional, programado para seguir uma trilha de cor escura por diferença de contraste com o chão, utilizando para isso um arranjo de sensores, que emitiam luz infravermelha no piso e analisavam a resposta, os quais conseguiam analisar posição e angulação do veículo em relação à trilha. Os dispositivos que se encontravam no carrinho no momento do início deste estágio são destacados a seguir:

- 6x Sensores de Ultrassom HC-SR04;
- 2x Arduínos Mega;
- 12 sensores TCRT5000(L) arranjados em uma placa de fenolite;
- Conversor DC-DC;
- Bateria 12v 7Ah;
- 2x motores 34:1 Metal 25Dx52L mm HP 6V;
- 2x motores 34:1 Metal 25Dx52L mm HP 6V com Encoder 48 CPR de quadratura;
- 2x Drivers Dual VNH2SP30 para Motor com Suporte MD03A;



(a) Veículo vista A



(b) Veículo vista B

Figura 2: AGV - Primeira Versão

Fonte: **O próprio Autor**

- 4x Rodas Dagu Wild Thumper (12060mm chrome);
- 1x Buffer Dual de Encoder de Quadratura LS7366R;
- Módulo SparkFun WiFly GSX Breakout para conexão Wi-Fi com estação de trabalho;
- Fios, chaves e Leds em geral.

Dessa maneira, visando melhorar essa versão, o AGV foi totalmente desmontado, afim de implementar mudanças em seu chassi e dispositivos utilizados. Alguns dispositivos foram mantidos e outros substituídos, visando a atualização da plataforma didática. A seguir, será abordado todos os dispositivos que se encontram atualmente no veículo.

2.2 Instrumentação

Os fundamentos físicos dos sensores utilizados no projeto do AGV são descritos nos tópicos a seguir.

2.2.1 Sensor de Distância

Sensores ultrassônicos ao redor de todo o veículo funcionam como sensores de distância. Seu princípio de funcionamento é o seguinte: um pequeno sinal acústico, com duração de cerca de 0.2 ms e frequência ultrassônica de 40 KHz, é emitido e o tempo entre a transmissão e o retorno do eco para o sensor é medido [4].

O tempo medido é uma proporção equivalente a duas vezes a distância do objeto mais próximo do cone de atuação do sensor, dessa maneira, a distância é calculada dividindo-se o tempo total por 2 (ida e volta do sinal). Dessa maneira, multiplicando o tempo que restou pela velocidade do som (aproximadamente 340m/s), podemos deduzir a que distância um determinado objeto está do

sensor. Caso nenhum sinal seja retornado até um determinado limite de tempo, podemos deduzir que nenhum objeto está dentro do cone de alcance do sensor.

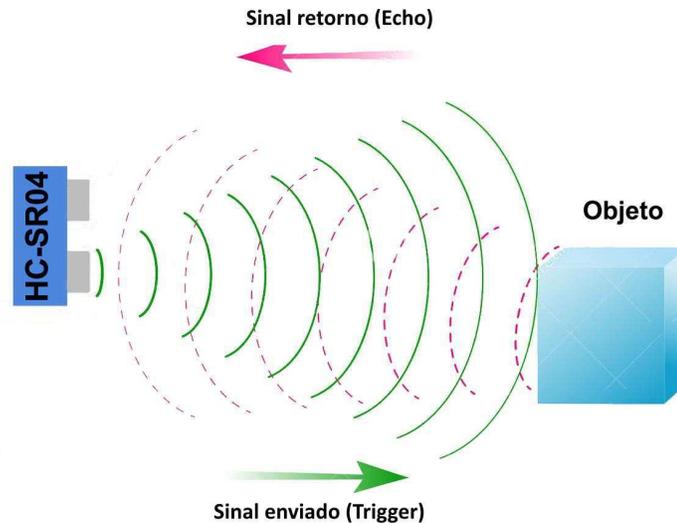


Figura 3: Funcionamento do sensor Ultrassônico

Fonte: <https://www.filipeflop.com/blog/sensor-ultrassonico-hc-sr04-ao-arduino/>

Existem dois possíveis fenômenos que podem ocorrer quando trabalhamos com medidas de distância através de ultrassom: reflexão do som ou interferência sonora. No primeiro caso, podemos obter uma leitura errada da distância real do objeto, caso haja outros objetos refletindo o sinal sonoro enviado antes do mesmo retornar ao sensor. Na interferência, o conjunto de vários sonares trabalhando ao mesmo tempo no mesmo ambiente pode ocasionar esse fenômeno, visto que um sensor pode ler o sinal enviado pelo outro e vice-versa, resultando em erros nas leituras.

O sensor neste projeto foi o HCSR04, capaz de medir distâncias de no mínimo 2 centímetros e com alcance máximo de 4 metros, mantendo uma boa precisão. Foram tomadas medidas para evitar os fenômenos de reflexão e interferência descritos acima, desta forma sendo possível obter medidas satisfatórias de distância, com erros desprezíveis.

2.2.2 Motor CC

Motores de Corrente Contínua (CC) foram os utilizados para este projeto. Dentre outros aspectos positivos, podemos citar o baixo ruído emitido, torque elevado e facilidade de controle por PWM (*Pulse Width Modulation*), o que permite a esse motor executar uma série de atividades.

O funcionamento desses motores se baseia no princípio eletromagnético clássico que diz que um condutor que conduz uma corrente elétrica, se exposto a um campo magnético, fica submetido a uma força eletromagnética.

Utilizando a figura 4 como referência, podemos notar a condição relatada acima, com um condutor submetido a uma corrente I com sentido indicado na figura, ao mesmo tempo que é submetido a um campo magnético B , com sentido também explicitado. Podemos notar o surgimento de duas forças de mesmo módulo em EG e DC, porém com sentidos opostos, o que resultará na

rotação do condutor da figura em torno do eixo OP. Para aumentar a sensibilidade do motor, é comumente utilizado uma técnica que usa diversas espiras enroladas em um cilindro, aumentando assim a rotação da espira.

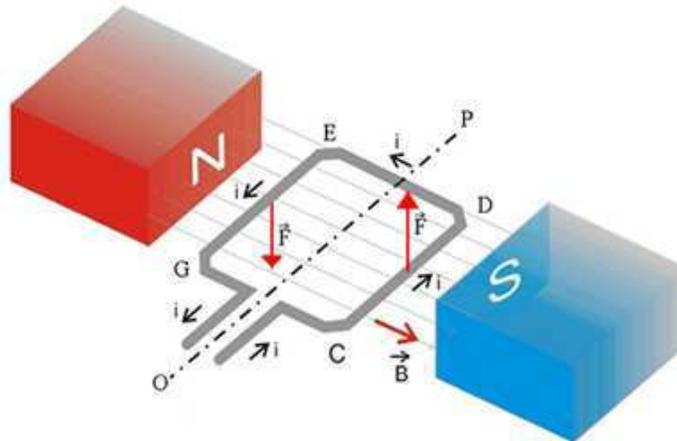


Figura 4: Funcionamento do Motor CC

Fonte: <https://brasilecola.uol.com.br/fisica/aplicacoes-forca-magnetica-um-condutor.htm>

Como podemos notar, o sentido de rotação é influenciado pelo sentido da corrente e o sentido do campo magnético, dessa forma caso seja necessário inverter o sentido de rotação de um motor CC, basta inverter o sentido de uma das duas grandezas citadas acima. No mercado, temos circuitos de eletrônica de potência denominados ponte H, do tipo chopper classe E, em outras palavras, atua nos quatro quadrantes, convertendo um sinal oriundo de uma fonte de alimentação CC com polaridades de corrente e tensão definidos em sinais com polaridades de tensão e corrente variáveis, dessa forma tornando possível o controle da polaridade dessas grandezas nos terminais do motor.

2.2.3 Encoder

Encoders, são dispositivos eletromecânicos capazes de converter movimentos de deslocamento ou rotação de forma linear em sinais elétricos de onda quadrada. São geralmente utilizados na realimentação do controle de motores ou para localizar veículos por odometria, técnica baseada na ideia de que as rotações da roda podem ser convertidas em deslocamento linear com relação ao chão.

Existem diversos tipos de encoders, e o utilizado neste trabalho foi o tipo óptico, que utiliza um disco marcado com vários segmentos de cores preto e branco, de forma alternada entre cada segmento, como ilustrado na figura 5b, associado a um led e um fotodiodo. O fotodiodo recebe a luz que é refletida pelo segmento branco, e não recebe nada durante a passagem pelo segmento preto, de tal forma que o sensor recebe a quantidade de pulsos equivalente à quantidade de pares de segmentos brancos e pretos para cada revolução do motor [4].

Para uma melhor precisão de leituras, o encoder se localiza diretamente no eixo do motor. Porém um sensor só é capaz de contar a quantidade de segmentos em cada rotação completa do eixo do motor, não sendo capaz de determinar qual o sentido de rotação do mesmo. Desta forma, outros dois sensores ópticos são equipados, de forma que possuam um pequeno deslocamento de fase um

do outro. Assim, independente do sentido de rotação, teremos sempre um dos sensores recebendo o sinal antes do outro, possibilitando assim a determinação de qual sentido de rotação do motor, além de saber a velocidade com que o eixo está rotacionando. Na figura 5a, se o Sinal A acontece antes do B, o sentido de giro é horário, caso contrário, o sentido de rotação do eixo é anti-horário.

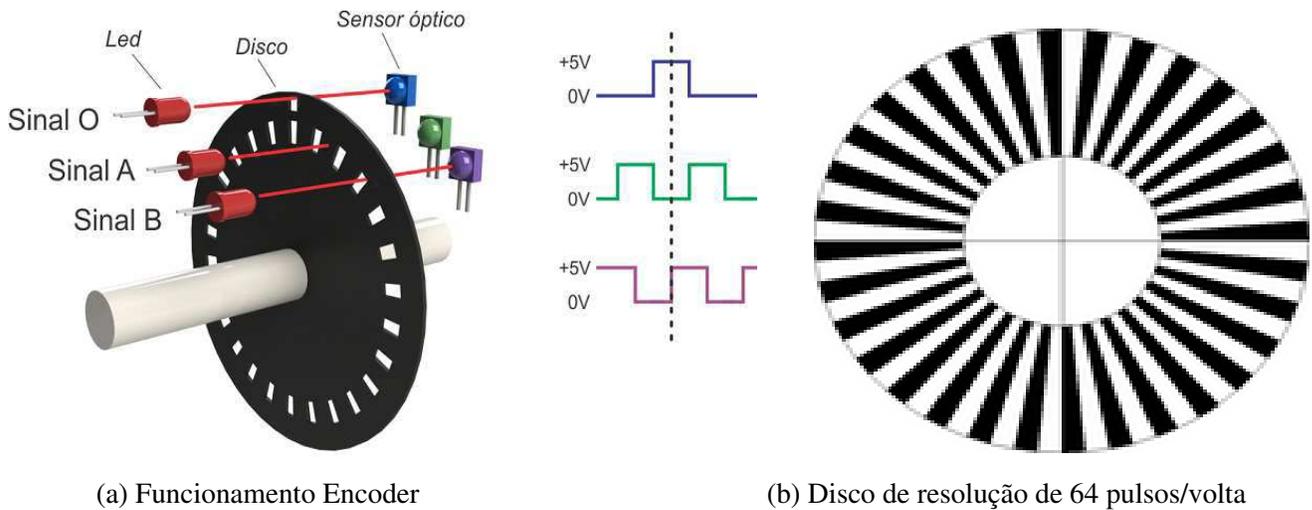


Figura 5: Funcionamento Encoder

Fonte: <https://www.hitecnologia.com.br/blog/o-que-é-encoder-para-que-serve-como-escolher-como-interfacear/>

2.3 Rede CAN

A Can Bus (ou Barramento Controller Area Network), é um protocolo de comunicação serial síncrono criado pela empresa alemã BOSCH na década de 80, originalmente idealizado para uso em sistemas automotivos. Seu princípio teórico de sincronismo se estabelece no início do envio de cada mensagem enviada ao barramento de dados, o que sempre ocorre em intervalos de tempo regulares e conhecidos.

Utiliza o conceito de múltiplos mestres, onde qualquer módulo da rede pode ser mestre ou escravo em qualquer momento, além disso toda e qualquer mensagem é enviada para todos os módulos conectados, caracterizando assim o regime de *multicast*.

Uma característica peculiar desse protocolo é trabalhar segundo o conceito de CSMA/CD with NDA (*Carrier Sense Multiple Access / Collision Detection with Non-Destructive Arbitration*). Isso quer dizer que todos os módulos checam o barramento por transmissões iniciadas por outros módulos e suas respectivas prioridades, para filtrar apenas as mensagens de maior prioridade no momento. Ao identificar qual delas tem prioridade em ser enviada, o módulo encerra a transmissão da mensagem de menor prioridade e a mensagem de maior prioridade continua a ser transmitida a partir deste ponto, sem perdas e sem necessidade de reiniciar a rotina de comunicação.

Também se utiliza do conceito do NRZ (*Non Return to Zero*), onde os bits são transmitidos por valores específicos de corrente e tensão. A velocidade em que os dados são transmitidos é inversamente proporcional ao comprimento do barramento de dados. A figura 6 demonstra o efeito

do comprimento do barramento na velocidade de transmissão de dados na Rede Can. A maior taxa de transmissão para um barramento de 40 metros é de 1 Mbps.

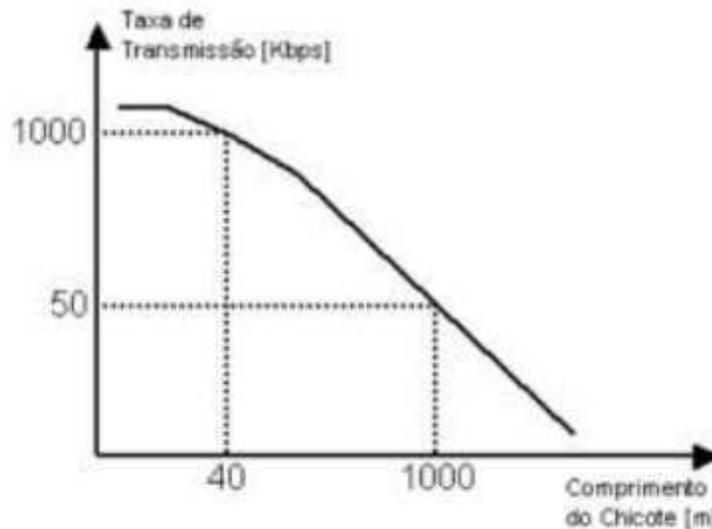


Figura 6: Relação taxa de transmissão x comprimento do barramento CAN.

Fonte: http://www.alexag.com.br/outros_artigos.html

Existem três formas de se construir o barramento na rede CAN utilizando fios, diferenciando apenas a quantidade utilizada.

- Barramento com um fio: Contém apenas o fio de dados, chamado também de linha CAN.
- Barramento com dois fios: Utilizam os sinais de dados CAN_H (CAN High) e CAN_L (CAN Low).
- Barramento com quatro fios: Além dos sinais CAN_H e CAN_L, utiliza dois fios para alimentação das terminações ativas da rede.

Uma peculiaridade desse barramento é que seus condutores devem ser trançados ao invés de blindados. Isso se deve ao fato de os dados enviados serem interpretados por diferença de potencial entre CAN_H e CAN_L, por este motivo o barramento CAN é classificado como par trançado diferencial. Por estar trançado, o barramento atenua os efeitos de possíveis interferências eletromagnéticas externas, pois qualquer interferência em um fio também será aplicada ao outro, causando certa flutuação com mesma intensidade e sentido para ambos os sinais, e devido ao fato do sinal ser interpretado por diferença de potencial entre os condutores, a comunicação não é afetada [6].

No barramento CAN, os dados não são representados por bits (0 e 1). Ao invés disso, eles são representados por bits dominantes e recessivos, devido à construção do barramento via CAN_H e CAN_L. A figura 7 ilustra os níveis de tensão em uma rede CAN, bem como os bits Dominantes e Recessivos.

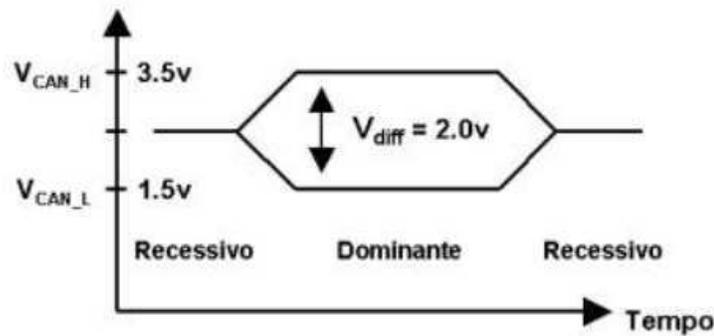


Figura 7: Influência dos bits Dominantes e Recessivos nos níveis de tensão no barramento ao longo do tempo.

Fonte: http://www.alexag.com.br/outros_artigos.html

Como exposto no início deste tópico, todos os módulos da rede CAN podem ser mestre e, assim, enviar suas mensagens. O protocolo prevê o envio de mais de uma mensagem ao mesmo tempo e consegue evitar colisão entre mensagens, utilizando uma técnica de arbitragem bit a bit não destrutiva. Tomando como exemplo dois módulos enviando mensagens diferentes, após o envio do primeiro bit, cada módulo que é mestre naquele momento verifica o barramento para confirmar se outro módulo conectado à rede está sobrescrevendo o bit enviado, pois bits dominantes sobrescrevem eletricamente um bit recessivo. Caso perceba que seu bit foi sobrescrito, o módulo mestre em questão interrompe imediatamente a transmissão, parar liberar o barramento ao módulo com mensagem de maior prioridade, sendo esta decidida pelos bits dominantes. O módulo o qual sobrescreveu o bit do outro e, portanto com maior prioridade no barramento, continua sua transmissão normalmente. Essa característica permite o protocolo CAN ser utilizado mesmo em sistemas de tempo real.

No que diz respeito ao formato das mensagens, temos dois formatos a serem citados:

- CAN 2.0A – Mensagens com ID de 11 bits. Capacidade de até 2048 mensagens, o que pode limitar algumas aplicações.
- CAN 2.0B – Mensagens com ID de 29 bits. Capacidade de até 537 milhões de mensagens. Os 18 bits a mais garantem uma grande capacidade de troca de mensagens, porém aumentam o tempo de transmissão, o que pode ser um problema em aplicações de tempo real.

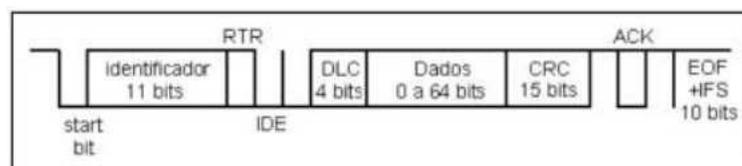


Figura 8: Mensagens no padrão CAN 2.0A.

Fonte: http://www.alexag.com.br/outros_artigos.html

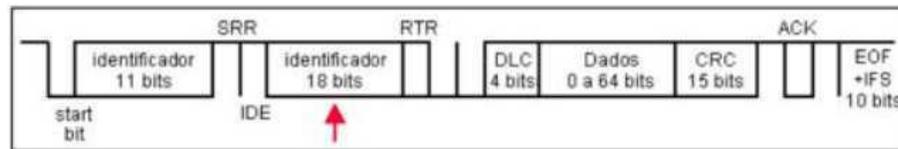


Figura 9: Mensagens no padrão CAN 2.0b.

Fonte: http://www.alexag.com.br/outros_artigos.html

No que diz respeito à detecção de falhas, podemos evidenciar uma das maiores vantagens desse protocolo, devido a sua robustez e capacidade de se recuperar em condições de falha. Podemos classificar as falhas de uma rede CAN em três categorias, cada uma delas com suas subcategorias específicas.

- **Nível Físico:** Para os barramentos com mais de um fio, caso aconteça algum problema físico, a rede continuará operando sob uma espécie de modo de segurança. Os problemas físicos possíveis de se acontecer ao barramento da rede CAN e que permitem ainda as atividades na rede são listados a seguir.
 - Curto do CAN_H ou CAN_L para os pinos de alimentação;
 - Curto entre os fios de dados da rede CAN;
 - Rompimento do CAN_H ou CAN_L;
- **Nível de Bit**
 - *Bit Monitoring:* Após o envio de um bit dominante, o transmissor verifica o barramento. Se o bit lido for recessivo, significa que há erros no barramento.
 - *Bit Stuffing:* Até cinco bits consecutivos podem ter o mesmo valor, sendo ele dominante ou recessivo. Se a aplicação deseja transmitir em sequência seis ou mais bits de mesmo valor, o módulo transmissor irá inserir um bit de valor contrário após cada grupo de cinco bits iguais. O receptor irá retirar este bit, conhecido como Stuff Bit. Se determinada mensagem for recebida com pelo menos seis bits iguais e consecutivos, temos a indicação de erro no barramento.
- **Nível de Mensagem**
 - *CRC ou Cyclic Redundancy Check:* O transmissor calcula um valor em função dos bits da mensagem e o transmite junto com ela. Os receptores recalculam este CRC e verificam se este é igual ao transmitido com a mensagem.
 - *Frame Check:* Os receptores analisam o conteúdo de alguns bits da mensagem recebida. Estes bits não mudam de mensagem para mensagem e são determinados pelo padrão CAN.
 - *Acknowledgment Error Check:* Os módulos receptores respondem a toda mensagem recebida, escrevendo um bit dominante em ACK de uma resposta que é enviada ao transmissor. Caso não seja recebida nenhuma resposta pelo transmissor significará que, ou a mensagem estava corrompida, ou nenhum módulo a recebeu.

3 Atividades Realizadas

A principal atividade deste estágio foi a atualização da plataforma didática do AGV, que envolveu além de mudanças físicas, com alteração do chassi do veículo, como mudanças nas tecnologias embarcadas e protocolos de comunicação.

Como o objetivo inicial do veículo era a simulação de funcionalidades em ambiente industrial, manteve-se os requisitos mínimos anteriormente adotados, sendo eles os mecanismos de locomoção e a definição da trajetória a ser seguida. A comunicação e leitura de dados remota da plataforma de trabalho com o AGV se manteve, porém com algumas melhorias que serão explicitadas no decorrer deste capítulo. Outro item que se manteve foram os sensores de distância, pela real necessidade de identificação e contorno de obstáculos, a fim de evitar colisões. *Encoders* foram utilizados para controle e leitura de velocidade de cada motor.

Em termos gerais, as atividades realizadas no estágio são listadas a seguir:

- Sistema de locomoção refeito, mantendo o modo de acionamento dos motores;
- Sistema de detecção de obstáculos reorganizado;
- Sistema de comunicação com plataforma de trabalho modificado;
- Implementação dos módulos CAN para comunicação entre dispositivos;
- Implementação de Processamento de Imagem para definição de trajetória por meio de faixa guia colorida;

Abaixo será explicitado um resumo de cada tópico discutido acima, explicitando o trabalho realizado durante o período de estágio.

3.1 Arquitetura do AGV

A arquitetura do AGV é ilustrado na figura 10. As conexões gerais dos circuitos e Microcontroladores são apresentadas no Apêndice A.

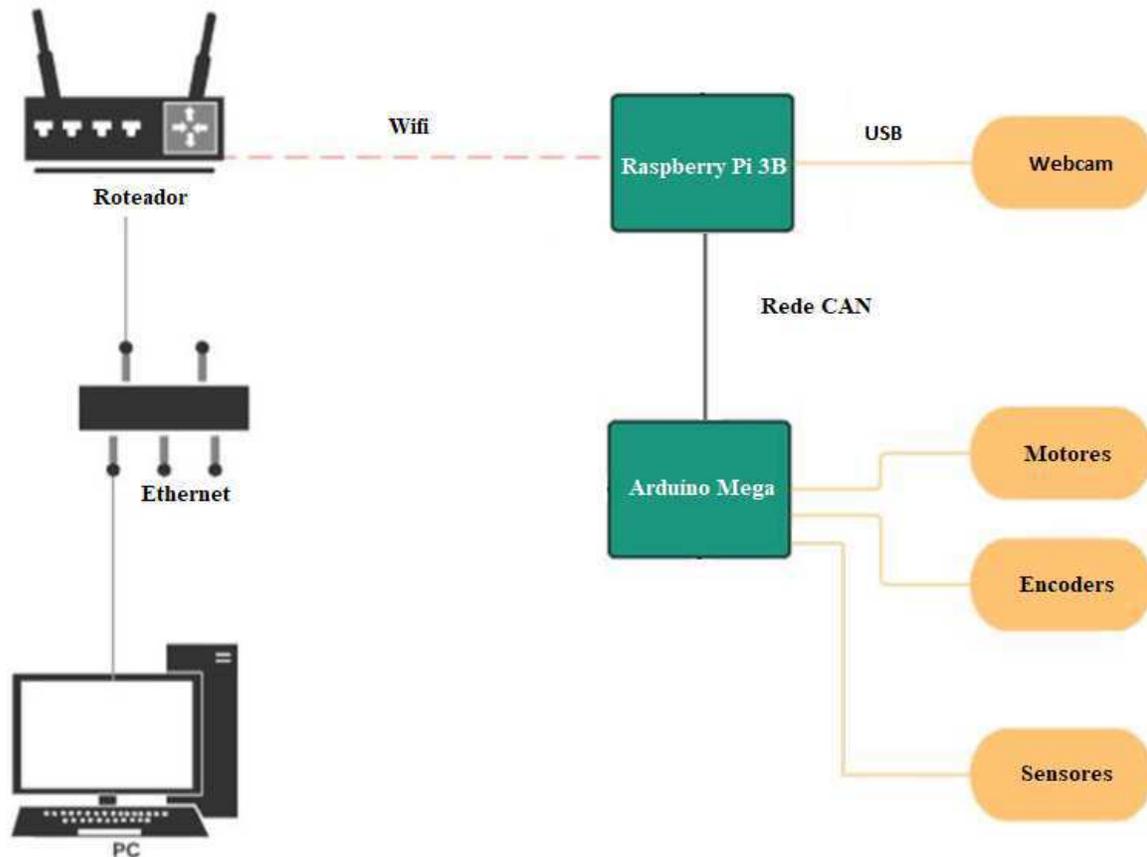


Figura 10: Arquitetura física do AGV.

Fonte: **Autor**

A figura 11 foi produzida em um momento anterior à adição dos andares no AGV, e demonstra como estão arranjados os dispositivos no chassi, com exceção de um driver de motor que foi retirado da versão final e dos sensores de distância, que foram realocados para o primeiro andar do veículo, adicionado posteriormente.

Já a figura 12 ilustra a versão final do AGV, já com dois andares de alumínio acoplados à estrutura do veículo, assim como a versão anterior, para o suporte de novas estruturas e tecnologias a serem introduzidas em trabalhos futuros.

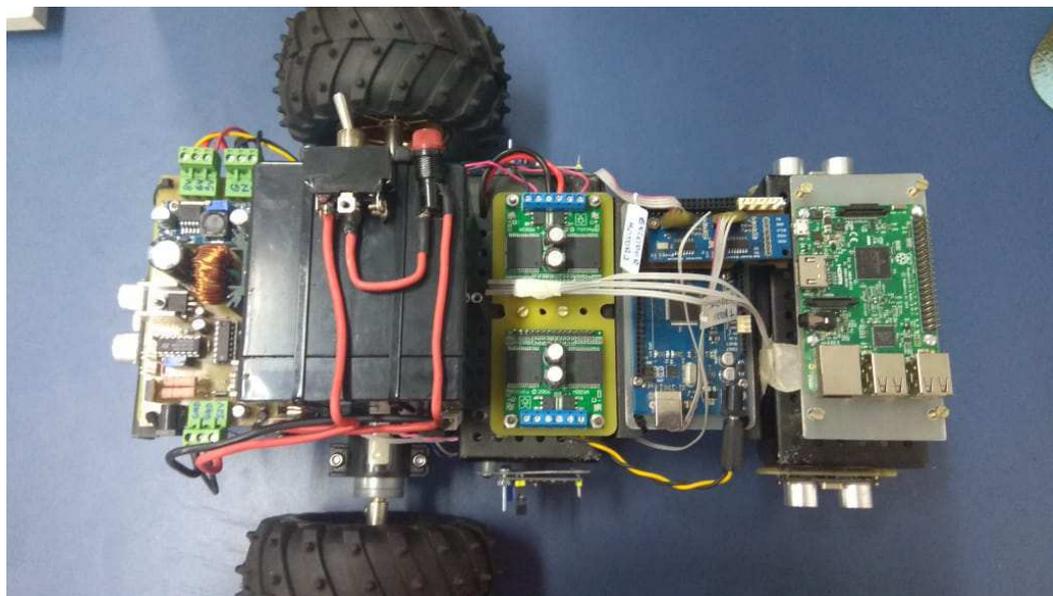


Figura 11: Primeira versão do novo Veículo

Fonte: **Autor**



Figura 12: Versão Final do AGV

Fonte: **Autor**

Neste trabalho, dois microcontroladores foram utilizados. O primeiro é o Arduino Mega, ilustrado na figura 13a, responsável por toda a instrumentação do veículo, coletando dados dos sensores de distância e agindo de acordo para evitar colisões, e acionando os motores ao receber tal ordem do raspberry, bem como lendo os dados dos encoders e informando à plataforma de desenvolvimento por intermédio do raspberry, caso necessário. A escolha do arduíno Mega se deve principalmente à quantidade de porta de entrada e saída disponíveis para trabalho, tanto analógicas como digitais, bem como por satisfazer os requisitos de processamento necessários ao funcionamento do sistema. Por último temos o Raspberry Pi 3B, ilustrado na figura 13b, incubido de intermediar ações manuais do usuário com o arduíno e por consequência a instrumentação e acionamento dos motores do veículo, além de colher dados de trajetória da câmera acoplada à parte frontal do veículo e enviar comandos para o arduíno via rede CAN sobre como e quando acionar os motores. O principal motivo para a inclusão do raspberry neste trabalho se deve ao seu alto poder de processamento, capaz de conseguir uma alta taxa de frames da câmera em questão, além de dar margem de sobra para futuros sistemas que devem ser incluídos ao veículo, que necessitaram de um sistema embarcado para controlá-los.



(a) Arduíno Mega



(b) Raspberry Pi 3B

Figura 13: Microcontroladores utilizados no projeto

Fonte: **Autor**

Uma lista completa dos materiais utilizados para a confecção da versão final do novo veículo é mostrado abaixo:

- 2x motores 34:1 Metal 25Dx52L mm HP 6V com Encoder 48 CPR de quadratura;
- 1x driver Dual VNH2SP30 para Motor com Suporte MD03A;
- 2x Rodas Dagu Wild Thumper (120x60 mm - cromo);
- 2x Encoders 48 CPR de quadratura (acoplados a dois dos motores do Sistema de Acionamento de Motores);
- 1x Buffer Dual de Encoder de Quadratura LS7366R;

- 1x Conversor DC-DC
- 1x Bateria 12V 5Ah;
- 6 sensores HC-SR04
- 1x Arduíno Mega
- 1x Raspberry Pi 3B
- 1x Chave ON/OFF
- 1x Fusível
- Fios Condutores
- Ferro de Solda

3.2 Alimentação do Sistema

Uma bateria de 12V 5Ah é responsável por alimentar o veículo, associado a uma chave liga-desliga e um fusível de proteção para curto circuitos no sistema. Por ser o componente mais pesado presente na estrutura, a bateria foi disposta perto dos pneus, para concentrar o peso do carro onde o mesmo tem sua tração, balanceando o veículo. Para a conversão dos níveis de tensão necessário para a alimentação de todos os módulos acoplados ao AGV, foi-se utilizado um conversor DC-DC produzido pelo próprio laboratório, o qual era responsável por converter os 12 Volts provenientes da bateria em 5 Volts, 6 Volts e 8 Volts, de forma eficiente e segura, garantindo uma boa fonte de alimentação para todos os componentes e uma autonomia maior da bateria.

3.3 Motores de Corrente Contínua e Encoders

Para o deslocamento do veículo, foram utilizados motores com encoders ópticos acoplados diretamente ao eixo do motor. O motor selecionado foi o de modelo 34:1 Metal 25Dx52L mm com Encoder 48 CPR de quadratura. A figura 14a ilustra o motor com o encoder acoplado a seu eixo, enquanto a figura 14b mostra o encoder utilizado neste trabalho.



Figura 14: Motor 34:1 Metal 25Dx52L mm HP 6V com Encoder 48 CPR de quadratura

Fonte: <https://www.pololu.com/product/1573>

O motor de corrente contínua tem baixo consumo e trabalha com tensão nominal de 6 Volts. Opera com uma caixa de redução de relação 34:1 acoplada ao mesmo que, com o encoder de resolução de 48 pulsos por volta localizado no eixo do motor, fornecem aproximadamente 1632 pulsos por revolução. Detalhes sobre a especificação técnica do motor se encontram na tabela 1.

Tabela 1: Especificações Gerais do Motor

Relação de Redução	34.014:1
Velocidade à vazio	280 rpm
Corrente à vazio	550 mA
Corrente com rotor bloqueado	6500 mA
Torque com rotor bloqueado	95 oz.in

Fonte: Adaptado de <https://www.pololu.com/product/1573>

O encoder trabalha com tensão de entrada de 3.5 a 20 V, corrente contínua, e consome uma corrente máxima de 10mA. No encoder, temos duas saídas, A e B, que são ondas quadradas de mesma amplitude da tensão de entrada, defasadas entre si aproximadamente 90 graus. Medindo-se a frequência e ordem das transições de saída podemos medir a velocidade e direção do motor, respectivamente. A tabela 2 demonstra a funcionalidade dos fios do encoder.

Tabela 2: Cores dos fios do encoder e suas respectivas funcionalidades

Cor	Função
Vermelho	Alimentação do Motor
Preto	Alimentação do Motor
Verde	GND do encoder
Azul	VCC do Encoder
Amarelo	Saída A do Encoder
Branco	Saída B do Encoder

Fonte: **Autor**

3.3.1 Bibliotecas

Para o controle dos motores presentes no veículo, foi reutilizada uma biblioteca feita pela desenvolvedora do primeiro AGV, Clarisse Barros [7]. A biblioteca contém dois arquivos, Motor.h e Motor.cpp, os quais contêm todas as funções necessárias para o controle de velocidade de cada motor de forma simples. Para a leitura dos encoders, também foi reutilizado uma biblioteca feita pela mesma autora anteriormente citada, contendo os arquivos Encoder.h e Encoder.cpp, responsáveis pela leitura dos encoders e reinício das contagens, de forma simples e objetiva.

3.3.2 Buffer

Uma placa de expansão para arduino Mega, ou shield, foi confeccionada para acoplar o módulo Dual LS7366R Quadrature Encoder Buffer. O módulo é ilustrado na figura 15, e o shield confeccionado para receber o módulo é ilustrado na figura 16. O buffer é composto por dois contadores de 32 bits, interagindo diretamente com a uma saída do encoder. A comunicação entre o módulo e o Arduino Mega é feita pelo barramento SPI, simplificando a ação de monitorar o número total de pulsos. Dessa forma, o Arduino não precisa dispor de parte de sua capacidade de processamento para contar os pulsos, deixando esta função ao buffer, focando assim em outras tarefas.

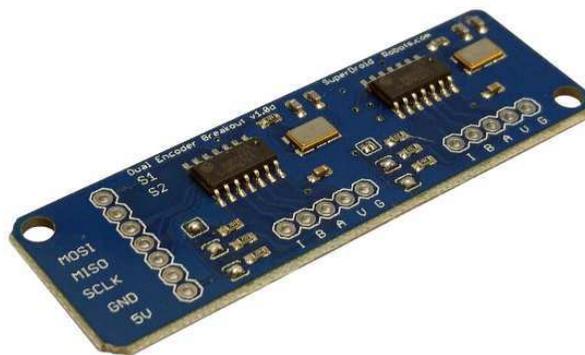


Figura 15: Módulo Dual LS7366R Quadrature Encoder Buffer

Fonte: <http://www.superdroidrobots.com/shop/item.aspx/dual-ls7366r-quadrature-encoder-buffer/1523/>

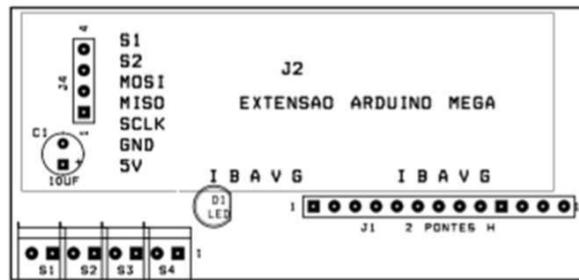
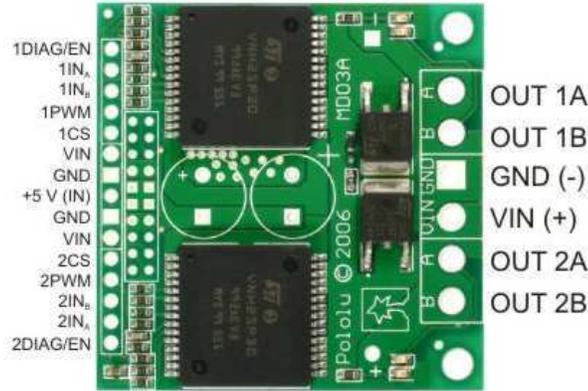


Figura 16: Shield Confeccionado para Arduino Mega

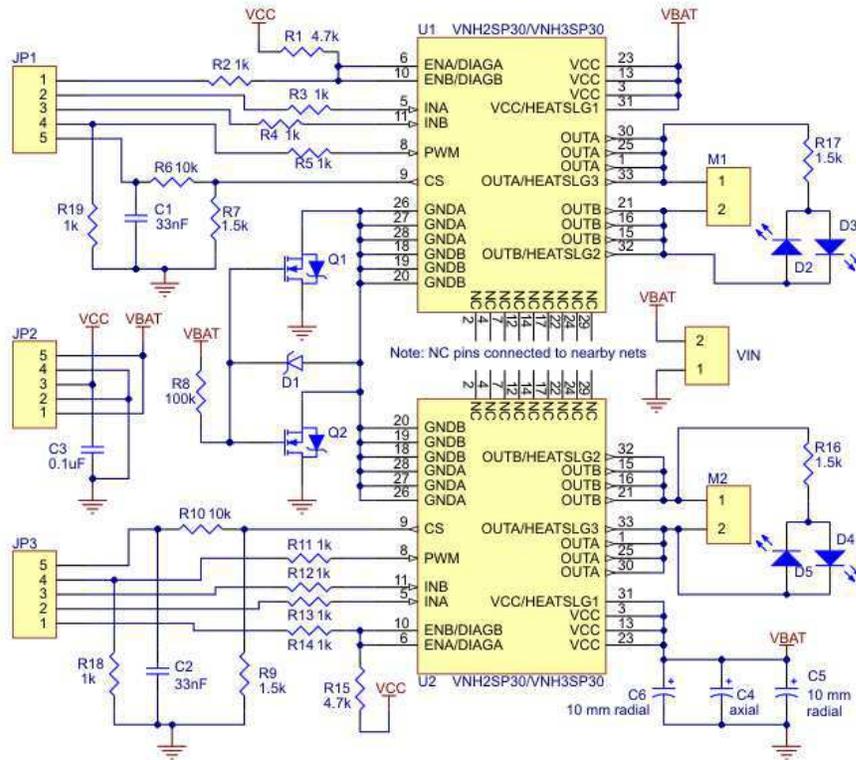
Fonte: **Autor**

3.4 Driver Para motor

O driver escolhido para este projeto foi o módulo Dual VNH2SP30 Motor Driver Carrier MD03A do fornecedor Pololu Robotics & Electronics, ilustrado na figura 17a. Esse módulo é capaz de controlar dois motores de alta potência, com corrente máxima de 14 A, utilizando para isso uma ponte H interna ao módulo, possibilitando o funcionamento do motor em ambos os sentidos de rotação. O driver também conta com monitoramento da corrente, fornecendo uma saída proporcional ao valor da mesma, além de proteções contra sobretensão. O esquemático do circuito interno do driver se encontra na figura 17b.



(a) Placa do Driver



(b) Esquemático do Driver

Figura 17: Driver Dual VN12SP30 Motor Driver Carrier MD03A

Fonte: <https://www.pololu.com/product/708>

As características elétricas do Driver são indicados na tabela 3.

Tabela 3: Especificações Gerais do Driver

Tensão de Alimentação	5.5 - 16 V
Máxima corrente temporária de atuação	30 A
Resistência do MOSFET	19mΩ
Máxima Frequência do PWM	20KHz
Sensor de Corrente	Aproximadamente 0.13 V/A
Desligamento por Sobretensão	Mínimo de 16V (Típico de 19V)
Corrente acionada sem limite de tempo	14 A

Fonte: Adaptado de <https://www.pololu.com/product/708>

3.5 Detecção de Obstáculos

Para evitar a colisão do AGV com obstáculos, um arranjo de sensores HD-SR04 foram distribuídos ao redor do veículo. O módulo do sensor de ultrassom utilizado é mostrado na figura 18. A disposição dos sensores na estrutura do primeiro andar do veículo é ilustrado na figura 19.



Figura 18: Sensor Ultrassônico

Fonte: <https://www.filipeflop.com/produto/sensor-de-distancia-ultrassonico-hc-sr04/>

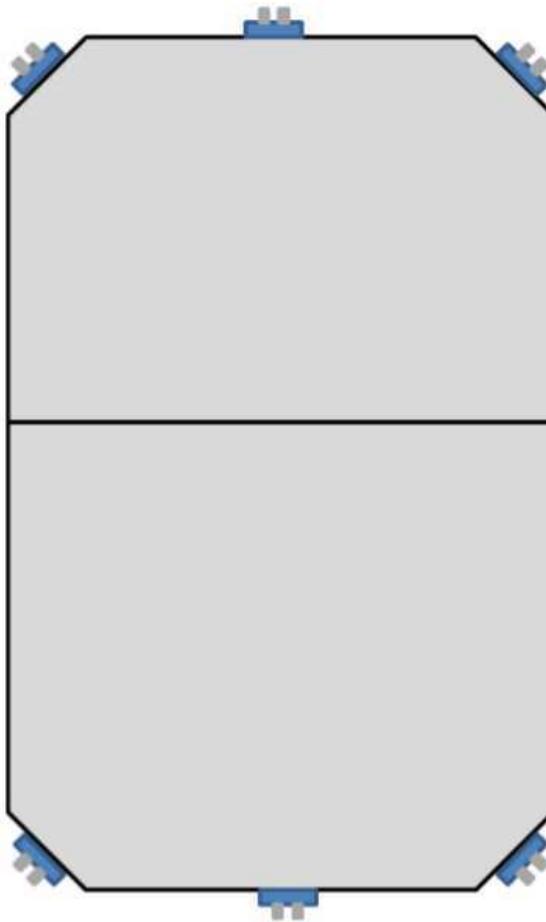


Figura 19: Disposição dos sensores de Ultrassom ao redor do veículo

Fonte:

<https://sites.google.com/a/dee.ufcg.edu.br/cgee/RelatorioEstagioClarisseFinal.pdf?attredirects=0>

3.6 Módulos CAN

Para a comunicação por Rede CAN, foi escolhido o módulo de comunicação ilustrado na figura 20. Alguns detalhes sobre suas especificações gerais se encontra presente na tabela 4.

Tabela 4: Especificações Gerais do módulo CAN

Tensão de Alimentação da placa	5 V (adaptado 3.3V)
Corrente de Operação	5mA (standby 1uA)
Temperatura de Operação	-40 a 85 graus Celsius
Máxima Frequência do PWM	20KHz
Dimensões	40 x 28 x 11,5mm

Fonte: Adaptado de <http://ww1.microchip.com/downloads/en/devicedoc/20001801h.pdf>

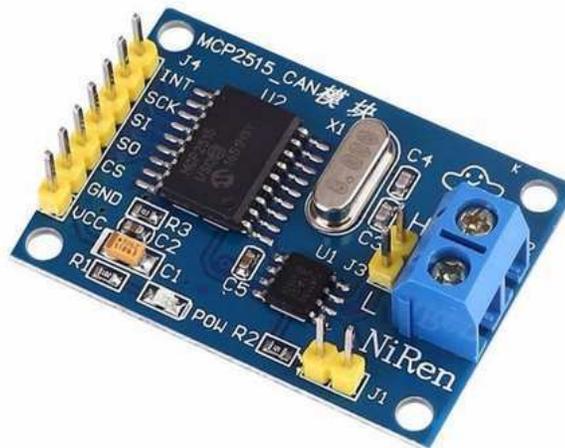


Figura 20: Módulo CAN MCP2515

Fonte: **Autor**

O controlador CAN encontra-se no CI MCP2515, com tensões de alimentação que variam entre 2.7 a 5.5 V. Já o transceptor da placa está localizado no TJA1050, que trabalha apenas com 5V. Como os módulos realizam a comunicação entre o arduino Mega e o Raspberry pi, e o Raspberry Pi não é tolerante a 5V em suas portas SPI, uma modificação em um dos módulos, o qual iria se conectar à SPI do raspberry, se fez necessária. O módulo CAN utilizado no trabalho conta com apenas uma entrada para alimentação, ao qual se deve fazer com 5V, devido ao transceptor CAN. Dessa forma, uma segunda alimentação foi improvisada, rompendo a trilha que alimentava o MCP2515 na placa e soldando um fio no pino de alimentação deste CI, que iria ligar diretamente no pino de 3.3V localizando no raspberry. A figura 21 ilustra a modificação realizada, bem como as ligações do módulo ao microcontrolador de exemplo Raspberry zero.

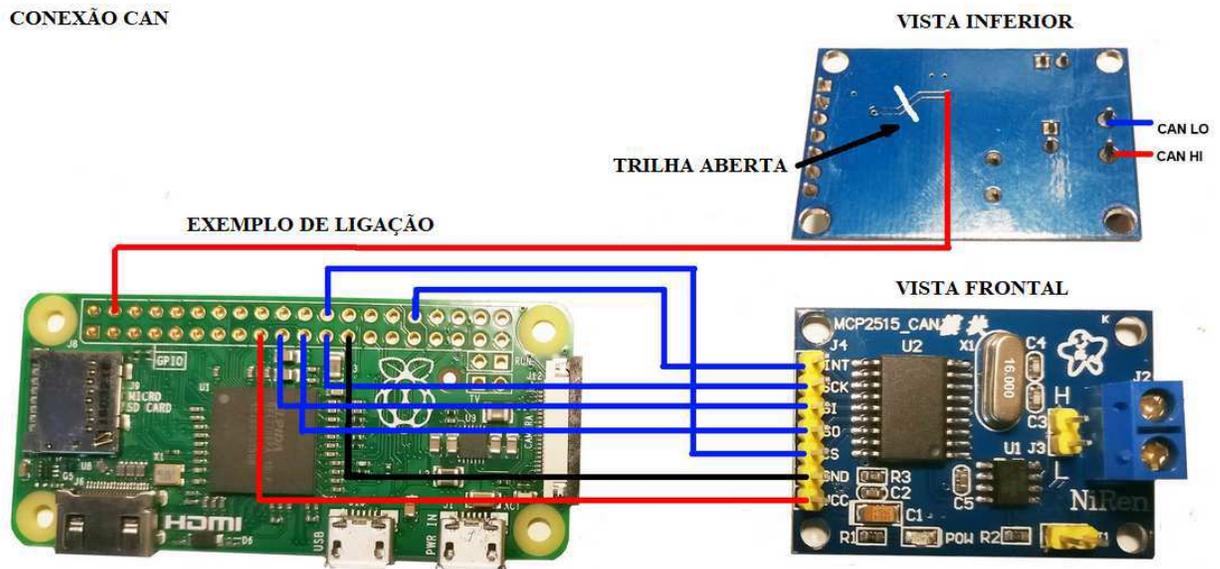


Figura 21: Modificação feita no módulo CAN para utilização no Microcontrolador Raspberry Pi

Fonte: Adaptado de <http://cosmostreamer.com/wired/howtomake>

Toda rede CAN possui 2 Terminadores. Estes terminadores são resistores de 120 ohm, conectados à rede, a fim de garantir a propagação dos sinais elétricos pelos seus fios. Localizados nas duas pontas da rede, eles garantem a reflexão dos sinais no barramento e o funcionamento da rede CAN. No módulo em questão, isso é feito apenas adicionando um jumper no terminal J1 da placa. Como a rede do AGV é composta apenas de dois módulos CAN, este procedimento foi realizado nas duas placas. A biblioteca utilizada no arduino para leitura e envio de dados na rede foi a biblioteca CAN, licenciada pela *Massachusetts Institute of Technology* e distribuída pelo membro Sandeep Mistry pela plataforma Github. A biblioteca é encontrada facilmente pelo gerenciador de bibliotecas do próprio arduino IDE.

3.7 Definição da trajetória por Processamento de Imagem

O método de definição de trajetória escolhido para o carrinho foi o da faixa guia, o que consiste em uma faixa de cor escura, preferencialmente preta, disposta em um piso de cor clara, preferencialmente branco, no qual a diferença de contraste de cores pudesse dar ao AGV a noção de como localizar a trilha, e dessa forma, qual caminho seguir. Para a coleta desses dados, foi utilizada uma webcam usb conectada ao raspberry, que é responsável por registrar frames do piso, e enviá-las ao raspberry, a fim de processar a imagem e encontrar o caminho a ser seguido.

3.7.1 Biblioteca OpenCV

A biblioteca utilizada para o processamento de imagens no Raspberry foi a OpenCV (Open Source Computer Vision Library), gratuito para uso acadêmico e comercial. Trabalha com as linguagens de programação C++, Python e Java, nos sistemas operacionais Windows, Linux, Mac OS, iOS, e android. OpenCV foi designado para eficiência computacional com foco em aplicações

em tempo real [10]. Pode usar múltiplos núcleos do processador e aceleração de hardware para agilizar as tarefas implementadas ¹. Neste trabalho foi utilizado a biblioteca com o código em Python, executando no Raspberry.

O passo a passo para tratamento da imagem e definição da trajetória é mostrado a seguir:

- Configuração de câmera e resolução adotada para os frames, no caso deste trabalho a resolução adotada foi de 160x120, a fim de obter uma taxa de frames por segundo superior.
- Captura dos frames e recorte da área de interesse. O recorte ajuda ao AGV a focar apenas na parte da linha próxima a ele.
- Conversão da imagem em escala de cinza, pois o veículo está interessado apenas em linhas negras na superfície branca. Após isso, é aplicado uma *gaussian blur*, que ajuda a eliminar quaisquer ruídos da imagem.
- Limitar a imagem. Nessa passo, limitamos a imagem convertendo-a para uma imagem booleana. Dessa forma, qualquer pixel abaixo de 60 na escala de cinza se tornará branco, e qualquer pixel acima de 60 se tornará preto.
- Achar os contornos da trilha. Precisamos que o microcontrolador detecte todas as formas que estão com a cor preta (neste caso a faixa guia), para serem processados nos próximos passos. Como a função do OpenCV edita a imagem que é inserida, faz-se necessário criar uma cópia da imagem originada do passo anterior, para aplicarmos essa função, visto que a imagem original é comumente utilizada como debug durante o desenvolvimento do código.
- Achar o centro do maior contorno. Primeiro é checado se algum contorno é detectado, caso a resposta seja afirmativa, é calculado o contorno com a maior área, então é encontrado as coordenadas x e y do centro deste contorno.
- Linhas ortogonais são desenhadas no centro do maior contorno na imagem inicial recortada.
- De acordo com a posição do centro da imagem, é possível comandar o AGV para seguir em frente ou manobrar para um dos lados, a fim de tentar manter sempre o centro do contorno no centro dos frames recebidos. A figura 22 ilustra o frame após todo o processamento descrito acima.

Dessa forma, temos uma lógica computacional simples e eficiente no que diz respeito em encontrar o centro da faixa guia e comandar o AGV para se manter no trajeto desejado, a partir das coordenadas de um ponto no centro desta faixa.

¹Tradução Livre do Autor

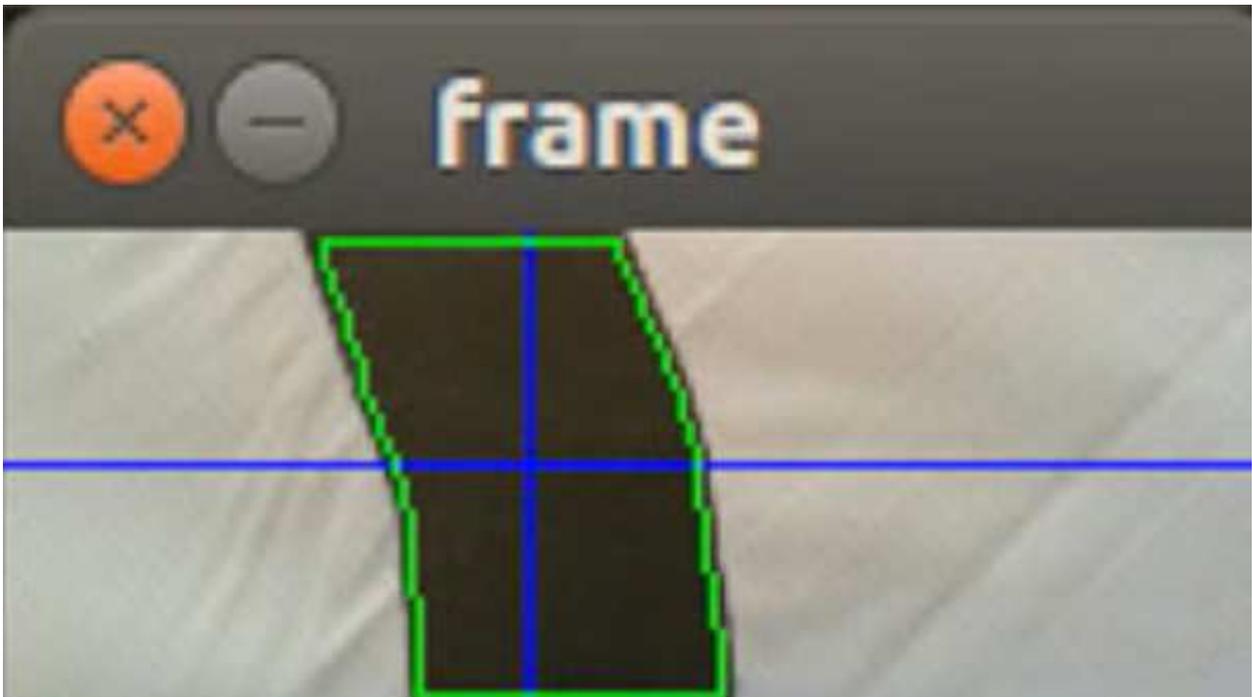


Figura 22: Imagem após processamento

Fonte: **Autor**

3.8 Estação de Trabalho: Computador

A comunicação do AGV com o computador se dá através da rede wireless, presente na estação de trabalho e no raspberry do veículo. A interface de comunicação utilizada foi o *VNC Server* para o raspberry e o *VNC Viewer* para o computador.

O VNC nada mais é do que um software que permite compartilhar em tempo real a tela transmitida por um dispositivo remoto conectado em qualquer lugar, mesmo que em redes de internet distintas, utilizando os servidores do VNC como intermediários. Porém além de compartilhar a tela do dispositivo em questão, o VNC permite o controle do dispositivo, tal como ele estivesse conectado diretamente à estação de trabalho.

Ao tomar controle do raspberry por meio do VNC, podemos controlá-lo de duas maneiras distintas, dependendo do que se deseja. Para o controle manual do veículo, utilizando as setas do teclado do computador, deve-se abrir um prompt de comando no sistema operacional do raspberry e executar o arquivo denominado *manual.py*. Para acionar o modo automático do veículo, em que o mesmo seguirá a trajetória marcada pela faixa preta no piso, o mesmo procedimento é realizado, porém agora o arquivo executado é denominado *automatico.py*.

A interface do programa em questão capaz de controlar o raspberry, e por consequencia, o AGV, é mostrado na figura 23.

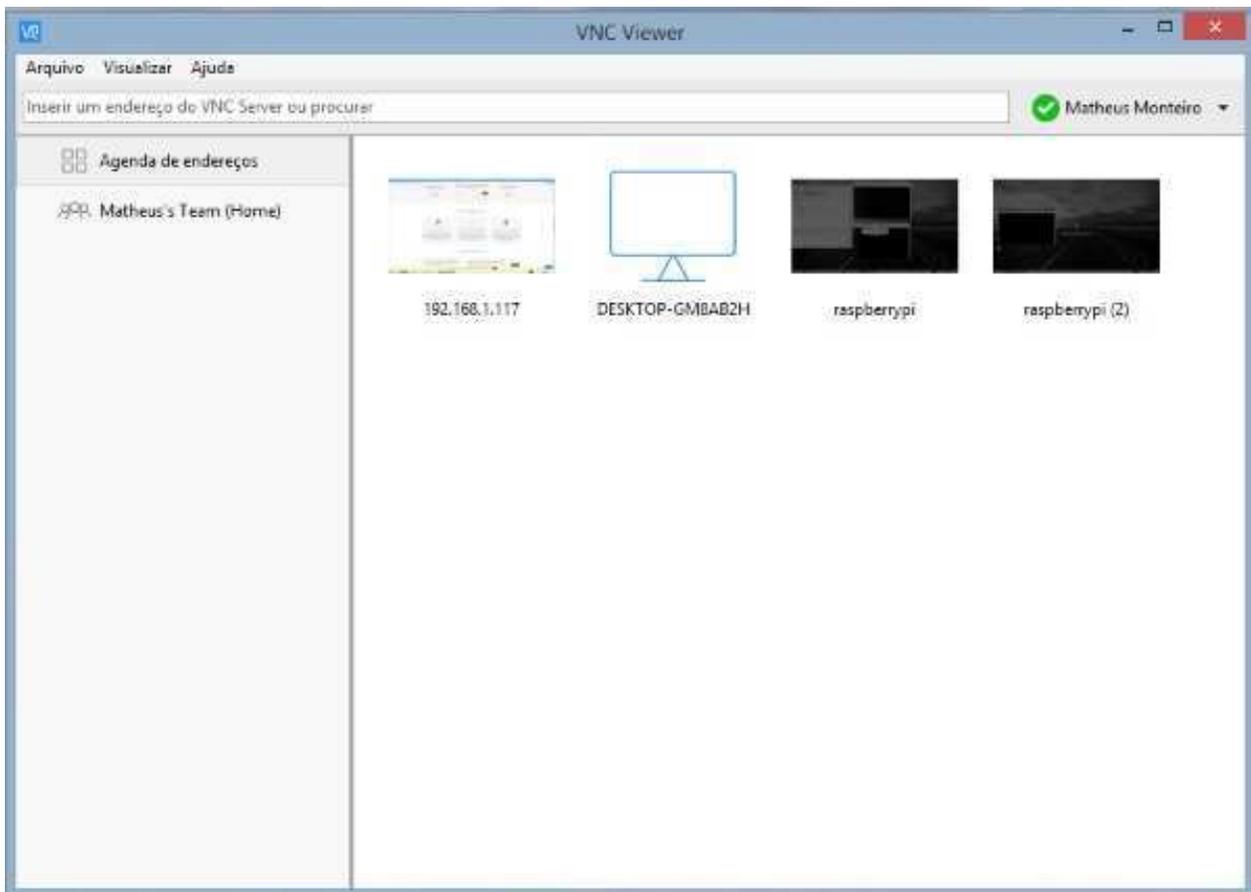


Figura 23: Interface Gráfica do controle remoto do Raspberry no Computador

Fonte: **Autor**

Neste capítulo foi apresentado os sistemas utilizados no veículo durante o estágio supervisionado, nos quais os mesmo conseguem suprir as exigências básicas para o funcionamento de um AGV, e contém suporte para aplicações extras. O próprio raspberry, por exemplo, foi implementado no veículo com a intenção de deixar sobrando poder de processamento no AGV, a fim de uma implementação de novas funcionalidades em trabalhos futuros, ampliando ainda mais a gama de aplicações possíveis para o veículo.

4 Resultados

Neste capítulo é apresentado o resultado da implementação das tecnologias discutidas nos capítulos anteriores, desta forma alguns testes de desempenho e funcionalidade foram idealizados, e os resultados são discutidos nos tópicos a seguir.

4.1 Detecção de Obstáculos

Em comparação à primeira versão do veículo, nada foi modificado no sistema de detecção de obstáculos, mantendo-se fixa as posições anteriores dos sensores e sua tecnologia. Por este motivo, os resultados do teste de desempenho desse sistema, idealizados e executados pela aluna responsável pelo primeiro veículo, Clarisse Barros [7], não diferem dos resultados de testes anteriormente realizados. Dito isto, o teste em questão é descrito a seguir.

Variou-se o posicionamento de um obstáculo de 1 em 1 centímetro, da posição 2 cm a 9 cm. Em verde estão as leituras feitas pelo sensor e em azul o posicionamento real da faixa.

Os resultados são ilustrados na figura 24. Em verde estão as leituras feitas pelo sensor e em azul o posicionamento real da faixa.

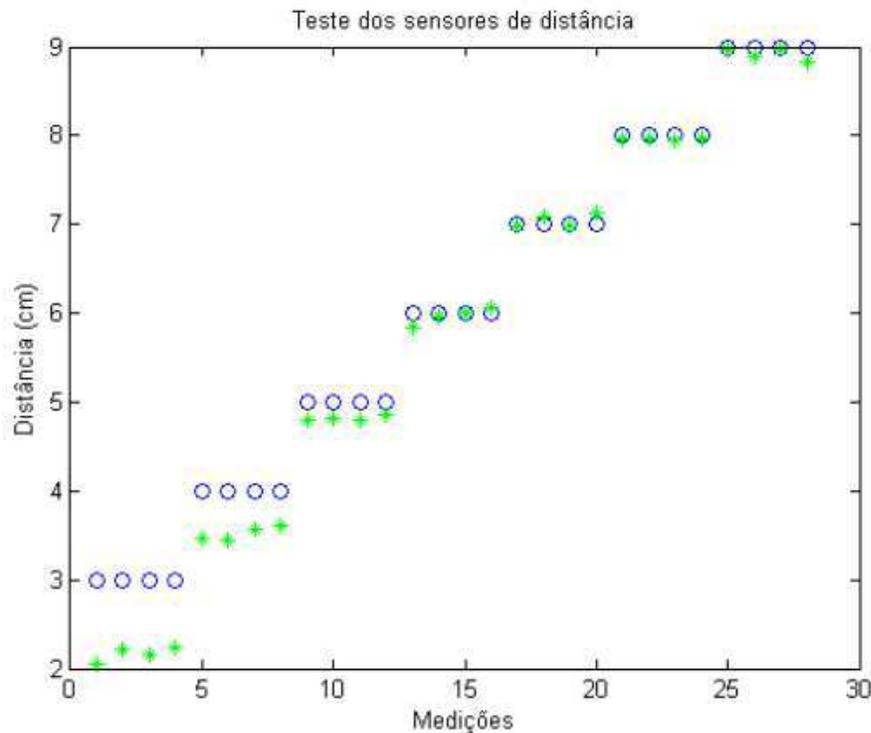


Figura 24: Teste dos sensores de distância

Fonte: [7]

Em seguida, variou-se o posicionamento de um obstáculo de 10 em 10 centímetros, da posição 20 cm a 120 cm. Novamente em verde estão as leituras feitas pelo sensor e em azul o posicionamento real da faixa. O resultado do teste é ilustrado na figura 25.

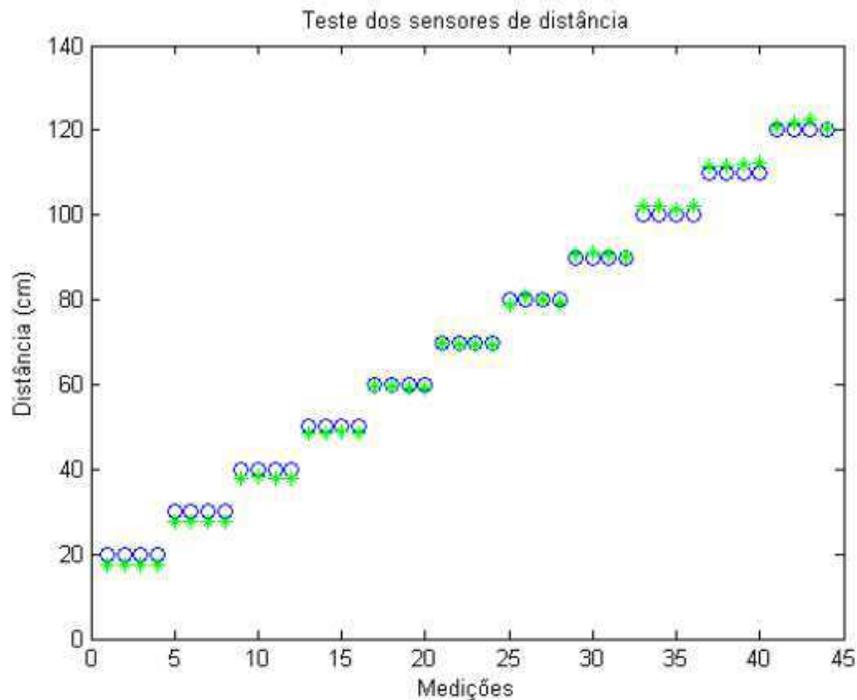


Figura 25: Teste dos sensores de distância

Fonte: [7]

Apesar de notarmos que medidas inferiores a 5 centímetros não serem lidas com precisão, podemos observar que em distâncias superiores as leituras são bem próximas dos valores reais, o que torna o sistema eficiente para evitar colisões de forma geral.

4.2 Acionamento dos Motores

Para atestar o correto funcionamento dos motores e de seu acionamento, foi utilizado a leitura do encoder presente em cada motor, o que nos dá o espaço percorrido por cada roda, que aliado ao tempo medido pelo Arduino durante o trajeto, informa a velocidade aplicada nos motores.

Os valores de tensão aplicados variam entre 0 e 6 Volts, controlados pelo microcontrolados Arduino, por intermédio do driver Dual VNH2SP30, utilizando para isso valores de PWM que variam entre 0 a 255. Os resultados foram enviados via rede CAN ao raspberry, o qual enviava em tempo real os dados por Wi-Fi à estação de trabalho, utilizando o software VNC Viewer, apresentado anteriormente.

Para verificar o comportamento dos motores, foram idealizados quatro experimentos, os quais verificaram as velocidades dos motores ao se deslocar para frente e para trás, em condições de atrito e motor livre. Para simular o ambiente com atrito na bancada de testes, foi utilizado uma base com rolo para treinamento, ilustrado na figura 26.

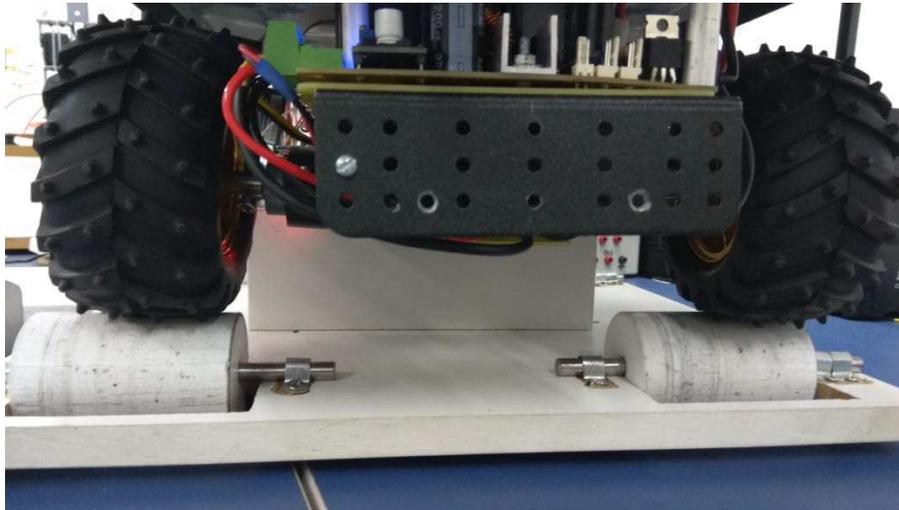


Figura 26: Teste dos sensores de distância

Fonte: **Autor**

As condições de cada experimento são explicitadas a seguir:

- Experimento 01 (Figura 27):
 - Motor sem atrito deslocando-se para frente
 - Velocidade Inicial: 0 PWM
 - Velocidade Aplicada: 200 PWM (78.43% de Torque)
 - Quantidade de Leituras: 21 Leituras
- Experimento 02 (Figura 28):
 - Motor com atrito deslocando-se para frente
 - Velocidade Inicial: 0 PWM
 - Velocidade Aplicada: 200 PWM (78.43% de Torque)
 - Quantidade de Leituras: 21 Leituras
- Experimento 03 (Figura 29):
 - Motor com atrito deslocando-se para trás
 - Velocidade Inicial: 0 PWM
 - Velocidade Aplicada: 200 PWM (78.43% de Torque)
 - Quantidade de Leituras: 21 Leituras
- Experimento 04 (Figura 30):
 - Motor sem atrito deslocando-se para trás
 - Velocidade Inicial: 0 PWM

- Velocidade Aplicada: 200 PWM (78.43% de Torque)
- Quantidade de Leituras: 21 Leituras

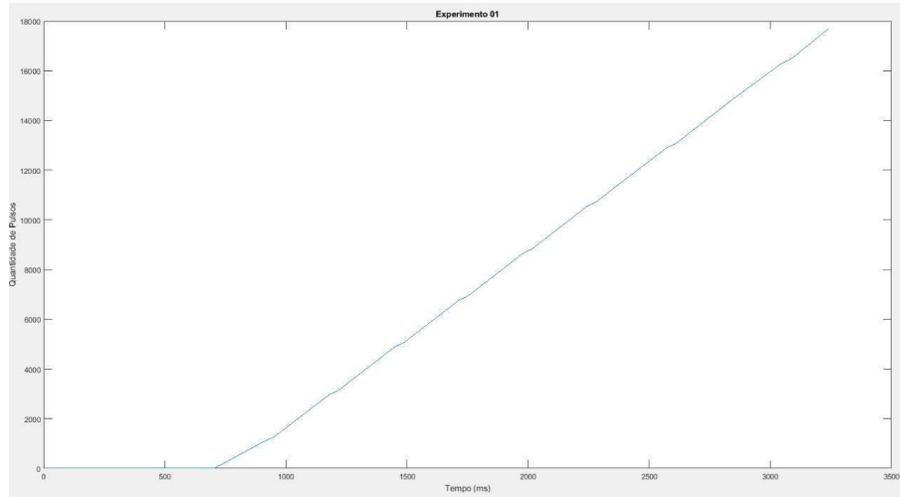


Figura 27: Experimento 01

Fonte: **Autor**

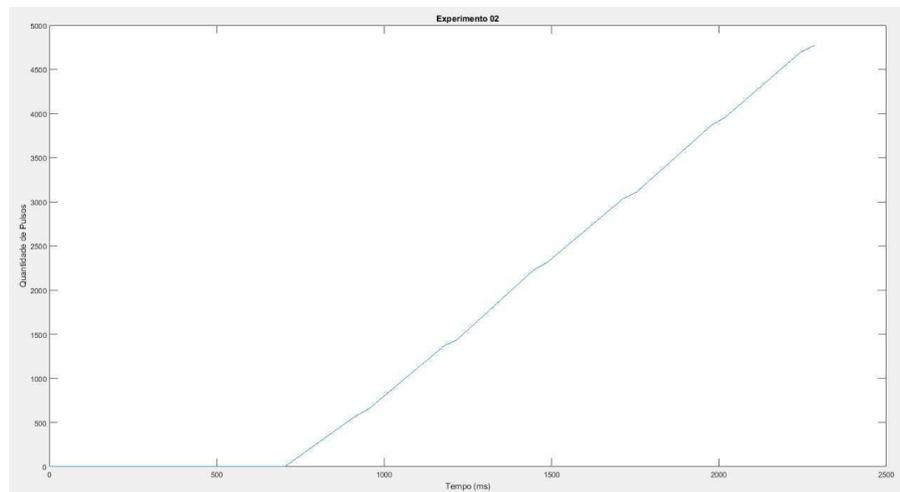


Figura 28: Experimento 02

Fonte: **Autor**

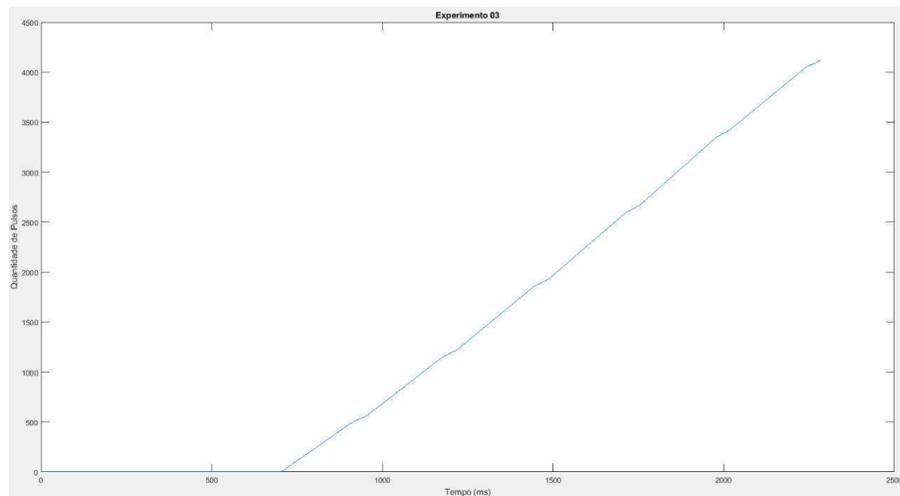


Figura 29: Experimento 03

Fonte: **Autor**

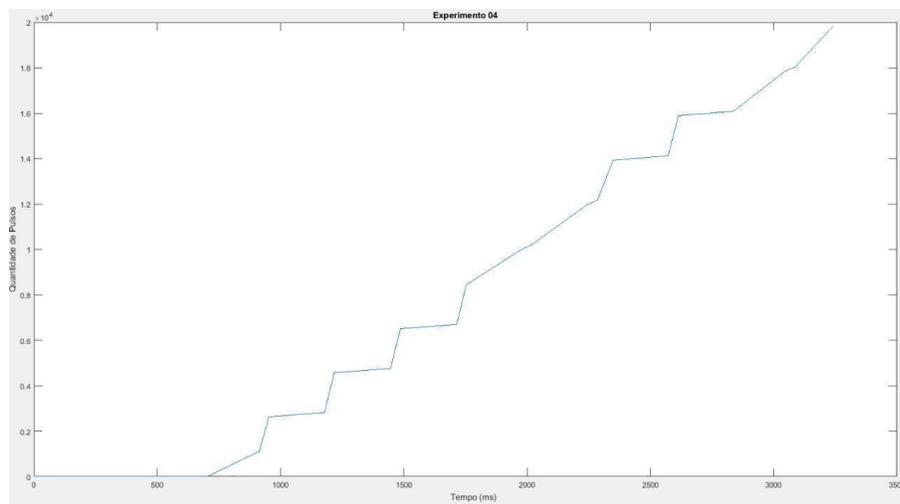


Figura 30: Experimento 04

Fonte: **Autor**

Analisando os gráficos acima, podemos observar que os motores tem uma resposta rápida à aplicação de novas velocidades, principalmente nos casos em que não há atrito. Podemos notar também que os motores alcançam rotações menores em superfícies com atrito, devido também ao peso da estrutura que requer um certo torque, limitando um pouco a velocidade máxima nessas condições, apesar da aceleração em ambos os casos serem praticamente iguais.

4.3 Identificação de Linha

Para este teste, utilizamos uma trajetória em linha reta com uma faixa preta feita com fita isolante de 3 centímetros de largura, com piso de cerâmica branco. Uma câmera USB foi disposta na parte superior do veículo, apontada para baixo, capturando os frames do trajeto e indicando ao sistema que caminho tomar, podendo seguir em frente, ou girar para a esquerda ou direita.

A figura 31 ilustra o comportamento do veículo durante toda a trajetória. Inicialmente o veículo foi posicionado fora do centro da trajetória, a fim de observar como o mesmo se comportava ao tentar corrigir o trajeto. As linhas em roxo e vermelho delimitam os limites da linha, enquanto a linha azul referencia o centro da trajetória.

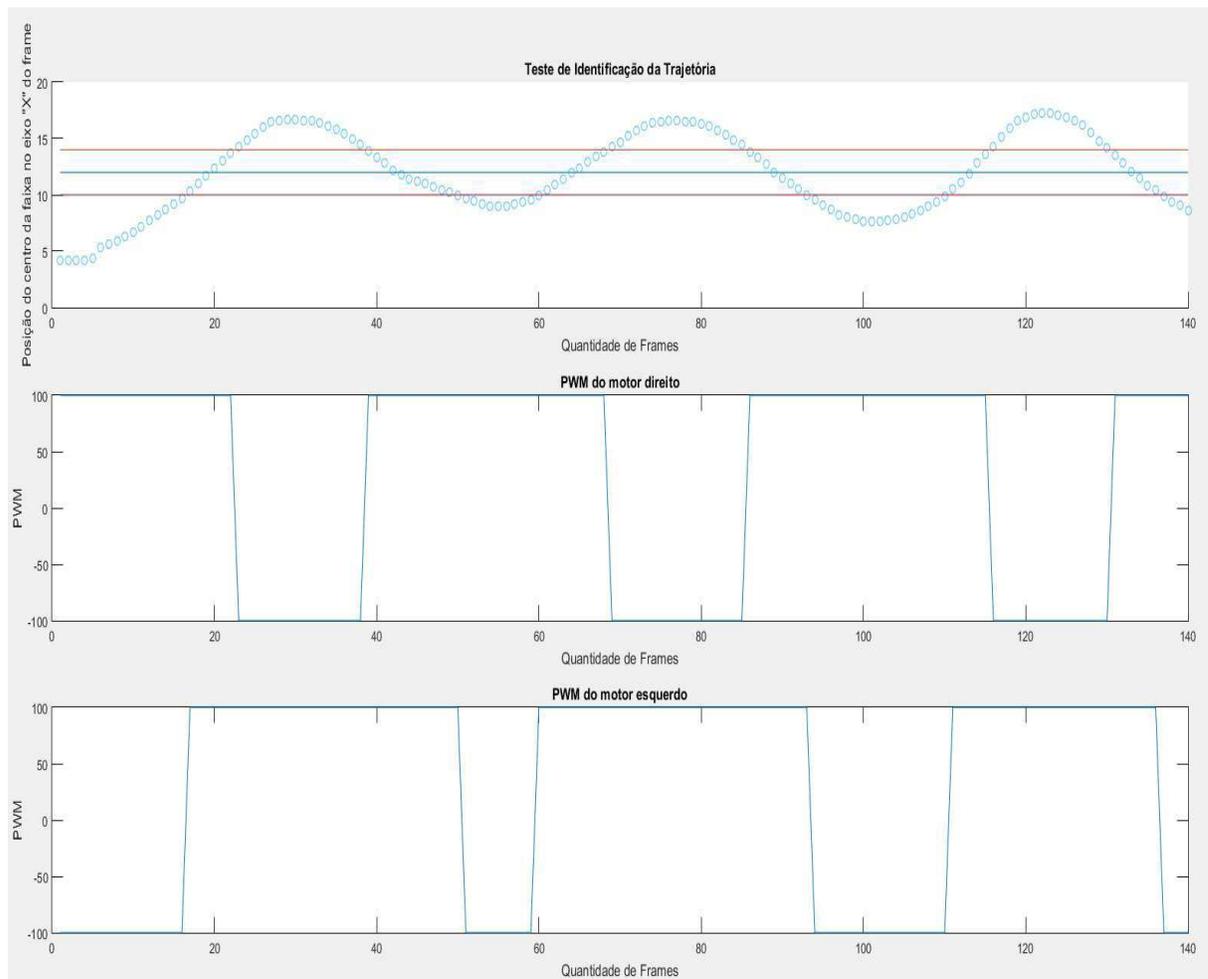


Figura 31: Teste de identificação da linha

Fonte: **Autor**

Podemos observar que o AGV consegue seguir a linha, porém com um comportamento ondulatório em relação à trajetória. Até o fim do período de estágio não foi possível encontrar uma solução com exatidão para o correto controle do trajeto, porém a causa é conhecida e se deve ao fato do veículo não conseguir reagir aos comandos a tempo.

4.3.1 Algoritmo de Controle

O microcontrolador Raspberry Pi é responsável pelo controle de trajetória do veículo, executando uma rotina feita em Python para tal. No caso deste projeto, foi inicialmente planejado o uso do controle do tipo *ON-OFF* para correção da trajetória. O controle recebe como entrada o centroide da faixa, resultado do processamento de imagem. A partir daí, a depender da posição do centroide nos pixels do eixo x dos frames que a câmera capta, o veículo assume uma direção específica. As saídas do controle são as seguintes:

- Caso o centroide detectado esteja no centro do frame ou aos arredores, com um determinado limite à direita e à esquerda, o veículo deve seguir em frente. Dessa forma, temos a aplicação de um PWM igual a 100 em ambos os motores, o qual é o mínimo necessário para quebrar a inércia e fazer o veículo se deslocar com mínima velocidade.
- Caso o centroide detectado esteja mais à esquerda do frame capturado, o veículo deve corrigir a trajetória, devendo girar à direita para isso. Para isso, aplicamos PWM igual a 100 no motor da direita, e PWM igual a -100 no motor da esquerda, assim fazendo o veículo girar para a direita, com ângulo de curvatura igual equivalente ao seu eixo.
- Caso o centroide detectado esteja mais à direita do frame capturado, o veículo deve corrigir a trajetória, porém desta vez girando à esquerda. Para tal, o PWM do motor da direita deve ser agora igual a -100, enquanto que o motor da esquerda deve ser o inverso, ou seja, com PWM igual a 100. Assim, o carro irá girar à esquerda.
- Caso o centroide não seja detectado pela câmera, quer dizer que o trajeto chegou ao fim e o AGV deve parar. Dessa forma, é aplicado o PWM igual a zero em ambos os motores, estacionando o veículo.

Quando o veículo está levemente inclinado a um dos lados, o controle de trajeto faz o mesmo girar para o sentido contrário, a fim de corrigir a trajetória. Porém devido à velocidade e ao eixo de curvatura do veículo, o mesmo ultrapassa um pouco o centro da linha ao executar o comando, o que ocasiona o controle novamente indicar que o mesmo deve corrigir novamente a trajetória girando o veículo para o outro lado, ficando preso nesse ciclo de correção em torno da referência.

5 Conclusão

Este trabalho teve como intuito apresentar as atividades realizadas durante o período de estágio supervisionado, como exigência para conclusão do curso de Bacharel em Engenharia elétrica pela Universidade Federal de Campina Grande.

O estágio tinha por objetivo desenvolver uma nova versão funcional para o AGV, e como resultado direto obteve-se uma versão atualizada a plataforma didática que já existia, com implementações de novos recursos de tecnologias da área, o que se mostrou bastante desafiador em alguns aspectos, fazendo-se necessário o conhecimento adquirido em diversas disciplinas do curso para a realização das tarefas envolvidas.

Por fim, é notória a contribuição do programa de estágio para a formação profissional do estagiário, visto que se pôs em prática boa parte dos conhecimentos adquiridos durante o curso, devido aos desafios apresentados, bem como expôs o dia-a-dia dos trabalhos exercidos em um laboratório, oferecendo ainda a oportunidade de elaborar uma solução única e completa para um real problema de engenharia.

5.1 Trabalhos Futuros

Como proposta para trabalhos futuros, é sugerido o uso de outras técnicas de controle para o devido ajuste da trajetória, como por exemplo um controle PI para elevar o PWM do motor na partida, a níveis suficiente para a quebra de inércia, para no regime diminuir a um nível que seja possível obter a mínima velocidade possível com o veículo em movimento, possibilitando maior tempo para correção da trajetória pelo AGV, e obtendo assim maior precisão do seu trajeto, em relação à referência, dada os parâmetros físicos do AGV em questão. Além disso, podemos utilizar novas técnicas para definição de trajetória, como trajetos marcados por fitas magnéticas, por exemplo, o que é possível devido ao poder de processamento adicional que o Raspberry dispõe na nova plataforma didática. Técnicas de odometria para o desenvolvimento de um sistema de localização espacial para o veículo se mostra também uma adição considerável ao projeto.

Referências

- [1] *Automated Guided Vehicles Market Size Worth \$7.3 Billion By 2025*. Disponível em: <https://www.grandviewresearch.com/press-release/global-automated-guided-vehicle-agv-market>
Acesso: 30 de Novembro de 2018
- [2] LIEC, 2015. *Laboratório de Instrumentação Eletrônica e Controle..* Disponível em: <http://lic.ufcg.edu.br>
Acesso: 30 de Novembro de 2018
- [3] KIM, W. C.; TANCHOCO, J. M. A. *AGV dispatching based on workload balancing*. 27. ed. [S.l.]: *International Journal of Production Research*, 1999..
- [4] BRÄUNL, T. *Embedded Robotics: Mobile Robot Design and Applications with Embedded Systems*. 2. ed. [S.l.]: Springer, 2006..
- [5] VAN HERREWEGE, A., SINGELLE, D., VERBAUWHEDE, I. *CANAuth-a simple, backward compatible broadcast authentication protocol for CAN bus*. In *ECRYPT Workshop on Lightweight Cryptography (Vol. 2011)*..
- [6] Alexandre de A. Guimarães, MSc *CAN Bus (Controller Area Network): Entendendo e Implementando um Rede de Dados*. Disponível em: http://www.alexag.com.br/outros_artigos.html
Acesso: 30 de Novembro de 2018
- [7] BARROS, Clarisse. *Desenvolvimento da instrumentação e comunicação para um veículo autoguiado*. Disponível em: <https://sites.google.com/a/dee.ufcg.edu.br/cgee/RelatorioEstagioClarisseFinal.pdf?attredirects=0>
Acesso: 30 de Novembro de 2018
- [8] DIAS, Charles. *Relatório de Estágio Supervisionado*. Disponível em: <https://www.google.com/url?q=https://sites.google.com/a/dee.ufcg.edu.br/cgee/> Acesso: 30 de Novembro de 2018
- [9] Einsteinium Studios. *Building a Line Following BeagleBone Robot with openCV* . Disponível em: <http://einsteiniumstudios.com/beaglebone-opencv-line-following-robot.html>
Acesso: 30 de Novembro de 2018
- [10] *OpenCV Website com descrição OpenCV*, Disponível em: <https://opencv.org/>
Acesso: 30 de Novembro de 2018

Apêndice A - Circuitos do Projeto

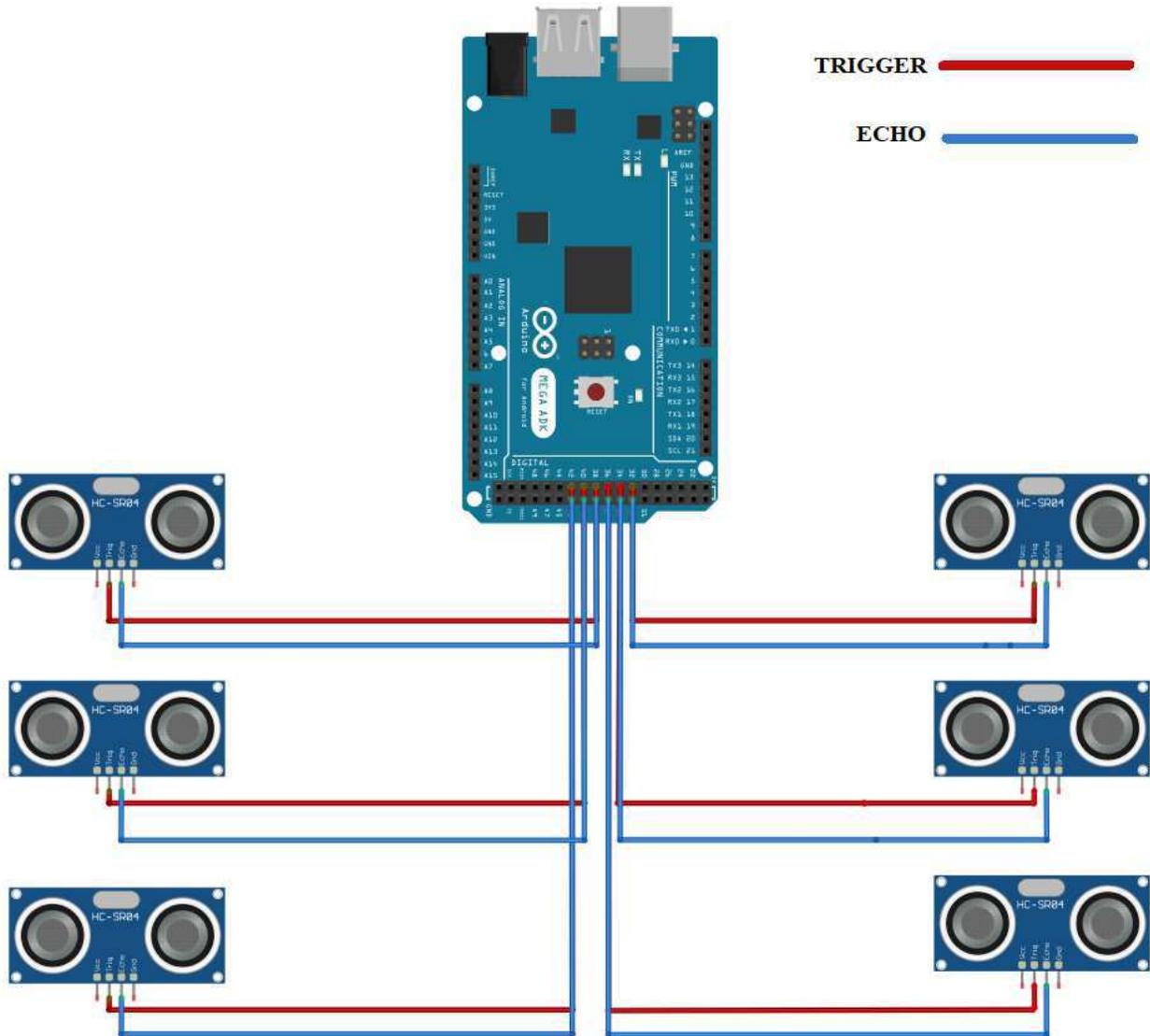


Figura 32: Sistema de detecção de obstáculos

Fonte: **Autor**

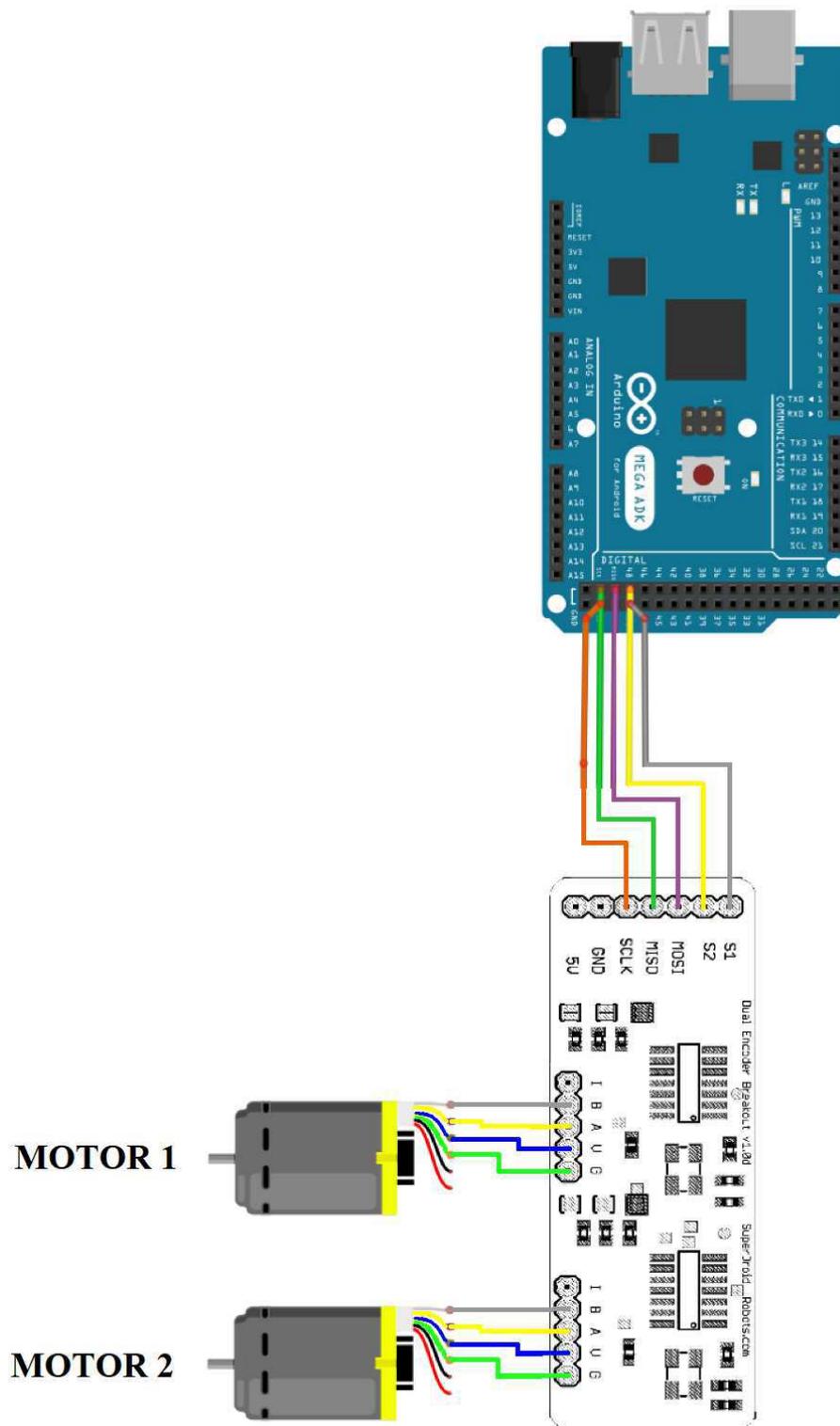


Figura 33: Sistema de leitura dos encoders

Fonte: **Autor**

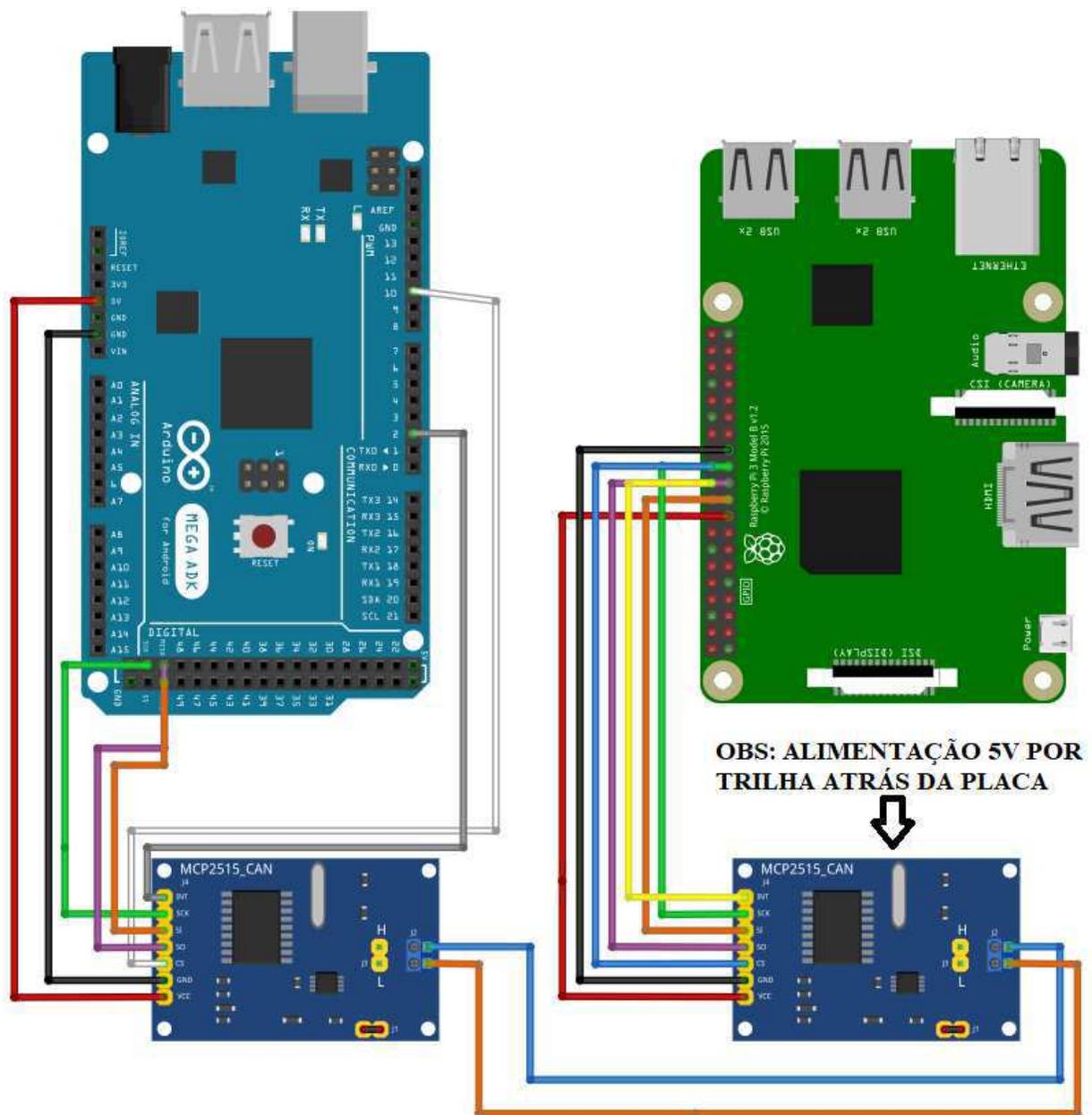


Figura 35: Sistema de comunicação por rede CAN

Fonte: Autor