

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETROTÉCNICA
CURSO SUPERIOR DE TECNOLOGIA EM AUTOMAÇÃO INDUSTRIAL

RENAN KEITYRO YAOITA
TIAGO BUGAN DEBS

AGV COM CONEXÃO WI-FI

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA
2019

RENAN KEITYRO YAOITA
TIAGO BUGAN DEBS

AGV COM CONEXÃO WI-FI

Projeto apresentado à disciplina de Projeto Integrador II, do curso de Tecnologia em Automação Industrial da Universidade Tecnológica Federal do Paraná, como requisito parcial na obtenção de aprovação na disciplina de Projeto Integrador II.

Orientadora: Profa. Mariana Aguiar Furucho

CURITIBA
2019

RESUMO

DEBS, Tiago Bugar; YAOITA, Renan Keityro. **CARRINHO SEGUIDOR DE LINHA COM CONTROLE DE VELOCIDADE**. 2017. 50f. Projeto Integrador (Projeto Integrador II) - Departamento Acadêmico de Eletrotécnica - Curso Superior de Tecnologia em Automação Industrial, Universidade Tecnológica Federal do Paraná, Curitiba, 2017.

Neste trabalho apresenta-se a construção de um protótipo de um veículo autoguiado com a intenção de integrar os conhecimentos adquiridos até o momento. Dentre os modelos de veículos autônomos, preferiu-se construir um carrinho seguidor de linha. Para não ficar limitado somente a essa funcionalidade, adicionou-se o controle de velocidade por cor da linha e um sistema de anticolisão, parando totalmente o protótipo caso haja um obstáculo na frente. Neste projeto utilizou-se um microcontrolador Arduino nano 3.x para controlar os periféricos (motores DC, sensores reflexivos infravermelhos e sensor ultrassônico).

Palavras-chaves: Veículo autoguiado; Seguidor de linha; Controle de velocidade por cor; Sistema anticolisão.

LISTA DE FIGURAS

Figura 1: Funcionamento de motor DC.	12
Figura 2: Motor DC utilizado no projeto.	13
Figura 3: Acionamento ponte H.	13
Figura 4: Driver ponte H.	14
Figura 5: Funcionamento sensor reflexivo.	15
Figura 6: Sensor de distância ultrassônico HC-SR04.	15
Figura 7: Princípio de funcionamento do sensor ultrassônico.	16
Figura 8: Portas Arduino Nano.	17
Figura 9: diagrama de blocos do projeto.	18
Figura 10: Chassi do AGV desmontado.	19
Figura 11: Início da montagem chassi AGV.	20
Figura 12: 2º passo da montagem chassi AGV.	21
Figura 13: Chassi AGV montado.	22
Figura 14: Esquemático da placa de sensores.	23
Figura 15: PCB com layout transferido placa de sensores.	24
Figura 16: Placa de sensores montada	25
Figura 17: Placa de sensores fixada ao AGV.	26
Figura 18: Conexão driver ponte H com motores	27
Figura 19: Integração Arduino - placa de sensores	28
Figura 20: Vista superior AGV completo	29
Figura 21: Placa de sensores AGV.	30
Figura 22: Vista frontal AGV sem o sensor anticolisão.	31
Figura 23: Vista frontal AGV com o sensor anticolisão.	31

SUMÁRIO

1 INTRODUÇÃO	6
1.1 TEMA	6
1.2 DELIMITAÇÃO DO TEMA	7
1.3 PROBLEMA	7
1.4 OBJETIVOS.....	8
1.4.1 <i>Objetivo Geral</i>	8
1.4.2 <i>Objetivos Específicos</i>	8
1.5 JUSTIFICATIVA	9
1.6 PROCEDIMENTOS METODOLÓGICOS	9
1.6.1 <i>Classificação da Pesquisa</i>	9
1.6.2 <i>Propósito dos Pesquisadores</i>	10
1.6.3 <i>Método</i>	10
1.7 EMBASAMENTO TEÓRICO	10
1.8 ESTRUTURA DO TRABALHO	10
1.9 CRONOGRAMA	11
2 FUNDAMENTAÇÃO TEÓRICA	12
2.1 MOTOR DC.....	12
2.2 DRIVE PONTE H	13
2.3 SENSOR ÓPTICO REFLEXIVO	14
2.4 SENSOR DE DISTÂNCIA ULTRASSÔNICO	15
2.5 MICROCONTROLADOR ARDUINO NANO	16
3 DESENVOLVIMENTO	18
3.1 VISÃO GERAL DO PROJETO	18
3.2 MONTAGEM.....	18
3.3 CÓDIGO DE PROGRAMAÇÃO	32
4 GERÊNCIA DE MANUTENÇÃO	37
4.1 NORMA REGULAMENTADORA 12	37
4.2 REQUISITOS DE MANUTENÇÃO.....	39
4.2.1 <i>Metodologia de Análise do Tipo e Efeito de Falha (FMEA)</i>	39
4.2.2 <i>Análise de Árvore de Falha (FTA)</i>	39
4.2.3 <i>Instrução de Trabalho</i>	40
4.2.4 <i>Plano de Manutenção</i>	40

5 CONCLUSÃO.....	41
REFERÊNCIAS	42
APÊNDICE A – DIAGRAMA DE GANTT	44
APÊNDICE B – DIAGRAMA DE LIGAÇÃO	45
APÊNDICE C – TAGS	46
APÊNDICE D – FMEA	47
APÊNDICE E – FTA.....	48
APÊNDICE F – INSTRUÇÃO DE TRABALHO	49
APÊNDICE G – PLANO DE MANUTENÇÃO.....	50

1 INTRODUÇÃO

Neste capítulo irá apresentar uma visão geral do projeto desenvolvido, de forma a esclarecer e definir o tema, objetivos, problemas, justificativa e estrutura do trabalho. Este trabalho visa a construção de um protótipo de AGV (*Auto Guided Vehicle*, ou Veículo Autoguiado em português) para transporte de peças dentro de uma linha de montagem, atualmente as indústrias necessitam de um grau de automação maior para poder aumentar sua competitividade no mercado. Uma forma de aumentar a produtividade de uma indústria é reduzir, ou eliminar o tempo onde não se agrega valor à um produto, o transporte de peças é uma operação onde não tem valor agregado ao produto final. Neste cenário um AGV diminui o tempo de transporte, pois ele irá realizar esta tarefa levando as peças de um ponto a outro do trajeto estabelecido ininterruptamente.

Devido à alta competitividade das indústrias e a crescente demanda por informações em tempo real, hoje em dia a área de IoT (Internet of Things, ou Internet das Coisas em português) está se expandindo cada vez mais. Segundo Warren Saichatien, “Em 2019, veremos amplamente divulgadas as redes 5G, que irão diretamente resultar na emergência de novos e interessantes casos para tecnologias associadas a indústria 4.0 - AR/VR, carros autônomos, AI/ML e IoT” (tradução livre do Inglês).

Um AGV tem como função principal transporte de cargas, sendo por vezes implementado em empresas de logística para coordenar a movimentação de cargas dentro de seu armazém.

1.1 TEMA

O projeto desenvolvido foi um protótipo de AVG para seguir um trajeto demarcado por uma faixa de cor preta e navega entre estações de cores variadas, que devem ser sobrepostas sobre um fundo branco, para simular uma planta de fábrica, o AGV recebe instruções remotamente de um dispositivo com conexão à internet. Os AVGs têm um amplo uso na indústria, principalmente em logística e linha de montagem. Nestes ambientes eles são utilizados em diversas tarefas, desde transporte de produtos entre estações de trabalho, até o manuseio de mercadorias delicadas e seu armazenamento no local designado para as mesmas.

Um AGV consiste de um veículo controlado por um sistema embarcado, que normalmente é alimentado por baterias e circula em um trajeto definido, há vários tipos de determinar a navegação de um AGV em algum ambiente, através de faixas ópticas no chão (MHI AGVS, 2016).

Devido a essa grande sofisticação nos sistemas de navegação é possível alterar com facilidade o trajeto de um AVG, pois não é necessária nenhuma instalação prévia na planta, dado que existe a possibilidade de utilizar lasers e faixas ópticas para delimitar a área coberta pelo veículo, exceto quando utiliza por campo magnético (indução).

1.2 DELIMITAÇÃO DO TEMA

Neste projeto irá desenvolver um AGV com navegação através de faixas ópticas e em escala reduzida para demonstrar seu uso de forma simplificada, simulando uma aplicação de transporte de carga na indústria. Para isso será utilizado um microcontrolador Arduino nano, que será responsável por todos os controles do veículo, um NodeMCU que irá receber e enviar as informações para o servidor, de forma a comandar e relatar o estado em que se encontra o AGV, um sensor ultrassônico responsável pela “visualização” do trajeto, uma placa com sensores óticos para detecção da faixa, um driver de ponte H (possui duas ponte H na sua construção eletrônica) para acionamento de dois motores que irão tracionar o carro e uma terceira roda livre para dar apoio ao chassi e auxiliar o veículo a seguir os comandos de direção dados à ambos os motores de tração. A comunicação entre o Arduino nano e o NodeMCU se dará através do protocolo de comunicação serial UART (Universal Asynchronous Receiver/Transmitter ou Receptor/Transmissor Universal Assíncrono).

Como não foram feitos estudos ou testes com cargas para o carro transportar, e não se têm uma plataforma adequada para o transporte, não irá fazer um teste com real transporte de carga. Como citado anteriormente (MHI AGVS, 2016), os AGV normalmente são alimentados eletricamente através de uma bateria, neste projeto foram usadas quatro baterias de 4.5V X-BalG modelo 18650, sendo duas para alimentar o drive de ponte H e a outras duas para alimentar o Arduino, NodeMCU, a placa de sensores infravermelho e sensor ultrassônico.

1.3 PROBLEMA

A identificação da faixa óptica e de obstáculos e a integração de hardware programável são partes complexas na construção de um AGV. Na indústria a movimentação de peças e produtos é uma atividade muito presente, porém, muitas vezes ineficiente. Com a crescente necessidade de acesso remoto à produtos, a comunicação estabelecida pelo NodeMCU com o Wi-Fi se faz necessária para adequar-se aos requisitos da indústria 4.0

1.4 OBJETIVOS

Nesta seção irá apresentar os objetivos do projeto, eles estão divididos em objetivo geral e objetivos específicos. O objetivo geral trata sobre grande marco do projeto, já os objetivos específicos são tarefas mais fáceis, mas de vital importância para alcançar o objetivo geral do projeto.

1.4.1 Objetivo Geral

O principal objetivo do projeto é desenvolver um AGV capaz de seguir uma faixa de determinada cor em uma superfície branca, receber instruções e enviar informações relevantes ao processo, e caso haja um obstáculo nesta faixa, parar o movimento imediatamente, de forma totalmente autônoma.

1.4.2 Objetivos Específicos

Os objetivos específicos do projeto estão listados abaixo, e são:

- Montar o chassi do AVG;
- Testar a percepção dos sensores ao espectro de cores visíveis;
- Definir valores de leitura dos sensores que indicam a cor da faixa;
- Testar a resposta de rotação dos motores aplicando uma tensão de forma direta;
- Definir valores de tensão a serem aplicados nos motores;
- Integrar motores ao driver ponte H;
- Fazer placa de circuito impresso com sensores reflexivos;

- Definir valores de distância para o sensor anticolisão;
- Integrar os sensores e driver ponte H com o Arduino através de hardware;
- Desenvolver programa para funcionamento correto do Arduino;
- Testar tempo de resposta dos motores;
- Reavaliar a tensão que deve ser aplicada nos motores dependendo do tempo de resposta deles.
- Desenvolver programa para garantir comunicação entre o NodeMCU e o servidor
- Configurar o servidor de forma a estabelecer uma conexão estável com a plataforma
- Integrar o Arduino com o NodeMCU através de hardware

1.5 JUSTIFICATIVA

Este projeto pode ser justificado tendo em vista que as indústrias cada vez mais se veem obrigadas a utilizar AGV no intuito de agilizar o processo de transporte e estocagem de produtos e matérias-primas, portanto o estudo do funcionamento e da construção de um protótipo em escala reduzida é muito importante para posterior aplicação dos conhecimentos adquiridos em um projeto real de aplicação de um AGV.

1.6 PROCEDIMENTOS METODOLÓGICOS

Este tópico tem como objetivo classificar a natureza da pesquisa, o propósito e o método utilizado para a obtenção de dados da pesquisa.

1.6.1 Classificação da Pesquisa

Este trabalho consiste em uma pesquisa científica aplicada. Pois, segundo a Organização para a Cooperação e Desenvolvimento Econômico (2002, p. 30), “a pesquisa científica aplicada consiste [...] em trabalhos originais realizados para adquirir novos conhecimentos; no entanto está dirigida fundamentalmente para um objetivo prático específico”.

1.6.2 Propósito dos Pesquisadores

De acordo com Gil (2002, p.42), esta pesquisa é descritiva, já que tem como objetivo descrever as características de certo acontecimento. Este acontecimento ou evento é a descrição do funcionamento de um carrinho seguidor de linha e desviador de obstáculos.

1.6.3 Método

A obtenção da pesquisa é bibliográfica. De acordo com Gil (2002, p. 44), uma pesquisa desta classificação “é desenvolvida com base em material já elaborado, constituído principalmente de livros e artigos científicos”. Por esta descrição que se classificou este trabalho, que teve a aquisição de dados por meio de Trabalhos de Conclusão de Cursos, artigos de periódicos, *datasheet* dos componentes e experimentos práticos disponibilizados.

1.7 EMBASAMENTO TEÓRICO

Para a montagem do projeto será utilizado o *chassi* de um kit robô que possui dois motores de corrente contínua (motor DC) para a locomoção e direção e uma roda livre. Para alimentar os motores com corrente elétrica demandada, o circuito será alimentado por quatro baterias de 4.5V e haverá um drive de ponte H. A parte lógica do circuito e todas suas limitações, serão implementadas através de um Arduino nano, que segundo o site oficial da empresa FILIPEFLOP, é “uma plataforma *open-source* de eletrônica, baseada em *hardware* e *software* fáceis de usar”. Por fim os sensores utilizados para a detecção de faixa foi o C9L3, um fototransistor que funciona como uma chave; e o sensor anticolisão utilizado foi o HC-SR04, um sensor de distância ultrassônico. Para comunicação com o servidor será utilizado um NodeMCU que irá fazer as requisições seguindo o protocolo MQTT (Message Queuing Telemetry Transport)

1.8 ESTRUTURA DO TRABALHO

Esta seção descreve como está estruturado e dividido o trabalho apresentado.

O primeiro capítulo é a introdução. Nela serão apresentados o tema e a delimitação do trabalho, em seguida estão descritos os problemas e pressupostos, objetivos e a justificativa do projeto. Além disso, possui os procedimentos metodológicos utilizados na pesquisa, um breve embasamento teórico, a própria estrutura do trabalho e, por fim, o cronograma.

Já o capítulo dois, que é a fundamentação teórica, descreve o funcionamento dos componentes primordiais utilizados no projeto.

O terceiro capítulo é a descrição do desenvolvimento prático, ou seja, a construção do protótipo, desde a ideia de controle e ligação até a montagem final do projeto.

No quarto capítulo descreve sobre a gerência de manutenção do projeto, ou seja, a questão de segurança do projeto, as possíveis falhas que pode apresentar no protótipo, as possíveis manutenção que se pode ser realizado caso haja a falha, e a instrução de trabalho de como realizar uma manutenção em um determinado componente.

E o último capítulo são as considerações finais, mostrando os resultados obtidos com o projeto, se foi conseguido alcançar todos os objetivos, e propostas para próximos trabalhos que embasem neste projeto.

1.9 CRONOGRAMA

O cronograma foi feito usando o diagrama de Gantt, cujo gráfico usado para ilustrar o avanço das diferentes etapas do projeto. Para isso, foi feito pelo *software* smartsheet, e resultou no Apêndice A.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo descreve o funcionamento dos componentes de maior relevância no projeto. Dentre eles estão: o motor DC, o drive ponte H, o sensor óptico reflexivo, o sensor ultrassônico e o microcontrolador.

2.1 MOTOR DC

O funcionamento de um motor DC ou motor CC (motor elétrico de corrente contínua) se baseia no magnetismo, utilizando as propriedades básicas de atração e repulsão de polos para gerar a movimentação da rotação (**Erro! Fonte de referência não encontrada.**). O sentido de rotação é estabelecido pela polaridade de tensão aplicada entre seus terminais.

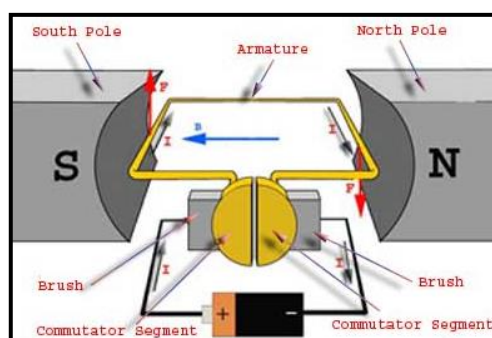


Figura 1: Funcionamento de motor DC.

Para que isso ocorra, o motor DC contém dois enrolamentos, um que fica no estator (enrolamento de campo que induz) e o outro no rotor (enrolamento de armadura que é induzido). Além disso, o motor possui escovas (comutadores) para o estabelecimento de contato elétrico girante ao enrolamento da armadura (GONZAGA, JULIANI, 2008).

Os motores no protótipo (Figura 2) serão responsáveis pela tração e direção.



Figura 2: Motor DC utilizado no projeto.

2.2 DRIVE PONTE H

Para controlar o motor de forma que não cause danos no microcontrolador, irá inverter as polaridades utilizando um circuito simples composto por chaves (reles ou transistores) que podem inverter a tensão aplicada nos terminais do motor quando acionadas corretamente (PATSKO, 2006).

A ponte H é constituída por quatro chaves dispostas em 'H', sendo que o motor é posicionado no meio. Para que o motor funcione, basta acionar diagonalmente as chaves, fazendo com que a corrente atravessa o motor. Para inverter o sentido da rotação, desligam-se as chaves acionadas e aciona-se o outro par de chaves, fazendo com que a corrente que passa pelo motor siga em direção oposta (PATSKO, 2006). Esse acionamento é mostrado na Figura 3.

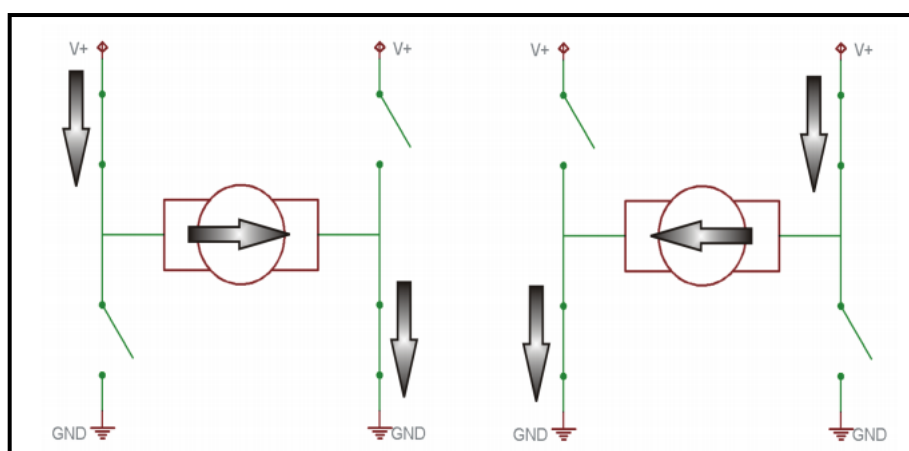


Figura 3: Acionamento ponte H.

Para o projeto, resolveu-se usar o *driver* de ponte H (Figura 4) da empresa FILIPEFLOP. Este drive contém um CI L298n, componente que possui duas pontes H à disposição e pode controlar a corrente de saída com o uso de resistores nos pinos de SENSE e SENSEB.

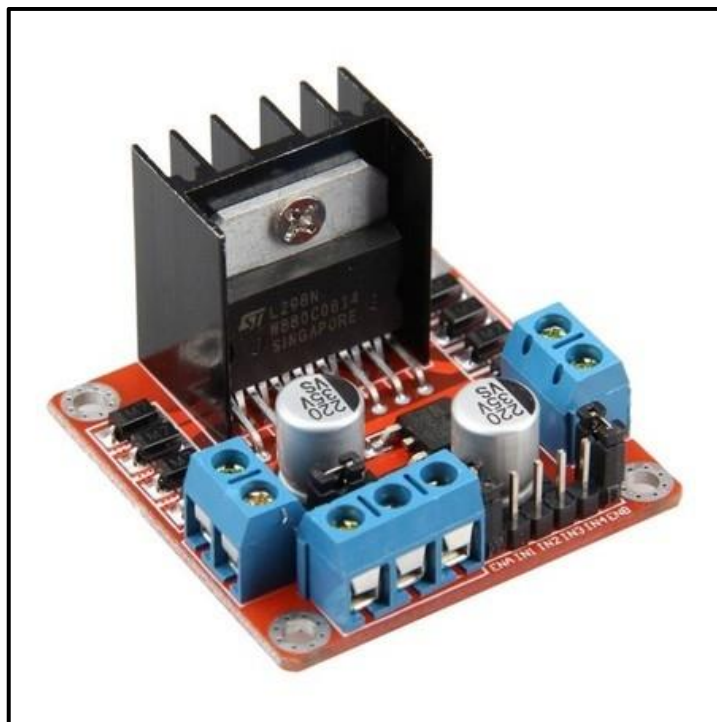


Figura 4: *Driver* ponte H.

2.3 SENSOR ÓPTICO REFLEXIVO

De acordo com o site FILIPEFLOP, este tipo de sensor (utiliza-se no projeto o sensor C9L3) possui acoplado no mesmo dispositivo um LED infravermelho (emissor) e um fototransistor (receptor). Foi especialmente projetado para bloquear outras faixas de luz que não seja a do próprio emissor, evitando que iluminações do ambiente venham causar alguma interferência.

O LED emite um feixe de luz infravermelha que é refletida pelo objeto que estiver à frente do sensor e que é, por sua vez, detectada pelo fototransistor (Figura 5). A cor e o material do objeto podem interferir no funcionamento do sensor. Existem materiais que refletem menos luz do que outros. Um exemplo disso são os objetos de cor preta que não são bons refletores, especialmente se sua superfície não for bem lisa (MOREIRA, 2002).

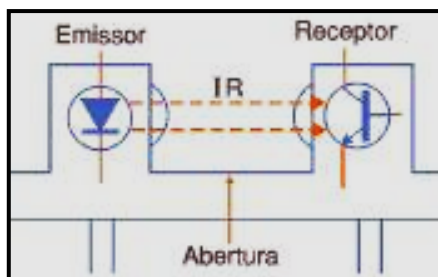


Figura 5: Funcionamento sensor reflexivo.

2.4 SENSOR DE DISTÂNCIA ULTRASSÔNICO

Para que o protótipo cumprisse um dos objetivos-gerais, de parar caso haja um obstáculo no trajeto, necessitou colocar um sensor de presença ou distância na frente do protótipo. O tipo escolhido para essa aplicação foi o sensor de distância ultrassônico HC-SR04 (Figura 7).



Figura 6: Sensor de distância ultrassônico HC-SR04.

Este sensor consiste, segundo o site Automatize, em emitir uma onda sonora de alta frequência (não-audível) (como na Figura 7), e o tempo de retorno do eco ao chocar com um objeto, pode-se calcular a distância estimada. O eco, ao retornar no receptor, é convertido em sinal elétrico.

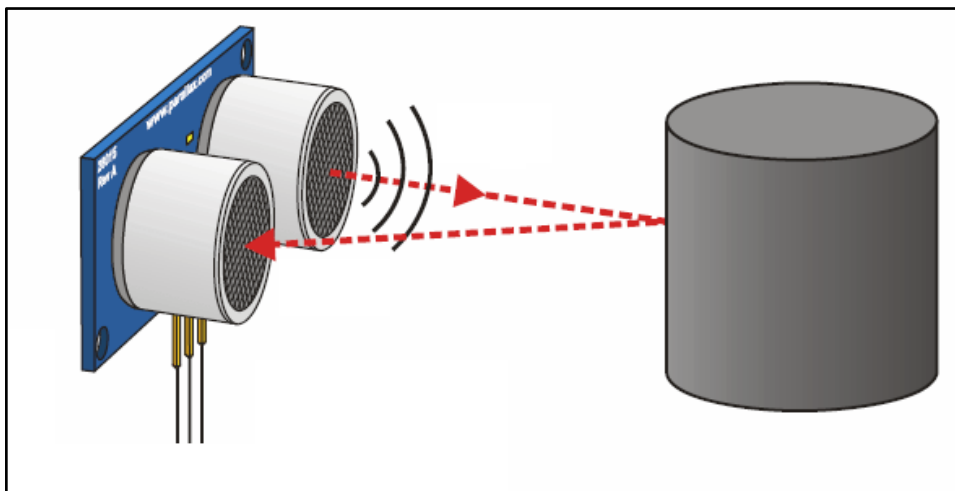


Figura 7: Princípio de funcionamento do sensor ultrassônico.

Para utilizar o sensor, é necessário aplicar um pulso de 10 μ s no pino Trig, que, automaticamente, o módulo emissor enviará oito (08) pulsos de 40 KHz. Espera-se o retorno no módulo receptor, caso haja, a distância é calculada da seguinte maneira:

$$Distância = \frac{\text{tempo do nível elétrico alto no receptor} \cdot \text{velocidade do som}}{2}$$

Sendo que a velocidade do som equivale a 340 m/s, e a divisão por dois (02) é necessária pois está medindo o tempo de ida e volta do sinal/onda.

2.5 MICROCONTROLADOR ARDUINO NANO

Neste projeto foi utilizado uma placa Arduino nano V3.x (Figura 8) para fazer o controle lógico das operações realizadas pelo projeto como um todo. Esta placa foi projetada e fabricada pela Gravitech e possui como microcontrolador um chip Atmega328, algumas de suas principais características são:

- Possui 14 pinos de I/O Digital.
- Tem 8 pinos de I/O Analógico.
- Capacidade de utilizar até 6 saídas com modulação de pulso (PWM).
- *Clock* de 16MHz.
- Possui dois *timers* à disposição.
- Suporta comunicação SPI.
- Suporta comunicação I2c.
- Tensão de operação 5Vdc.

Para garantir que os controles de todas as funções do projeto fossem executados, foi usado duas das seis saídas de PWM, cinco entradas analógicas e quatro saídas digitais para leitura dos sensores que detectam a faixa e acionamento do *driver* ponte H.

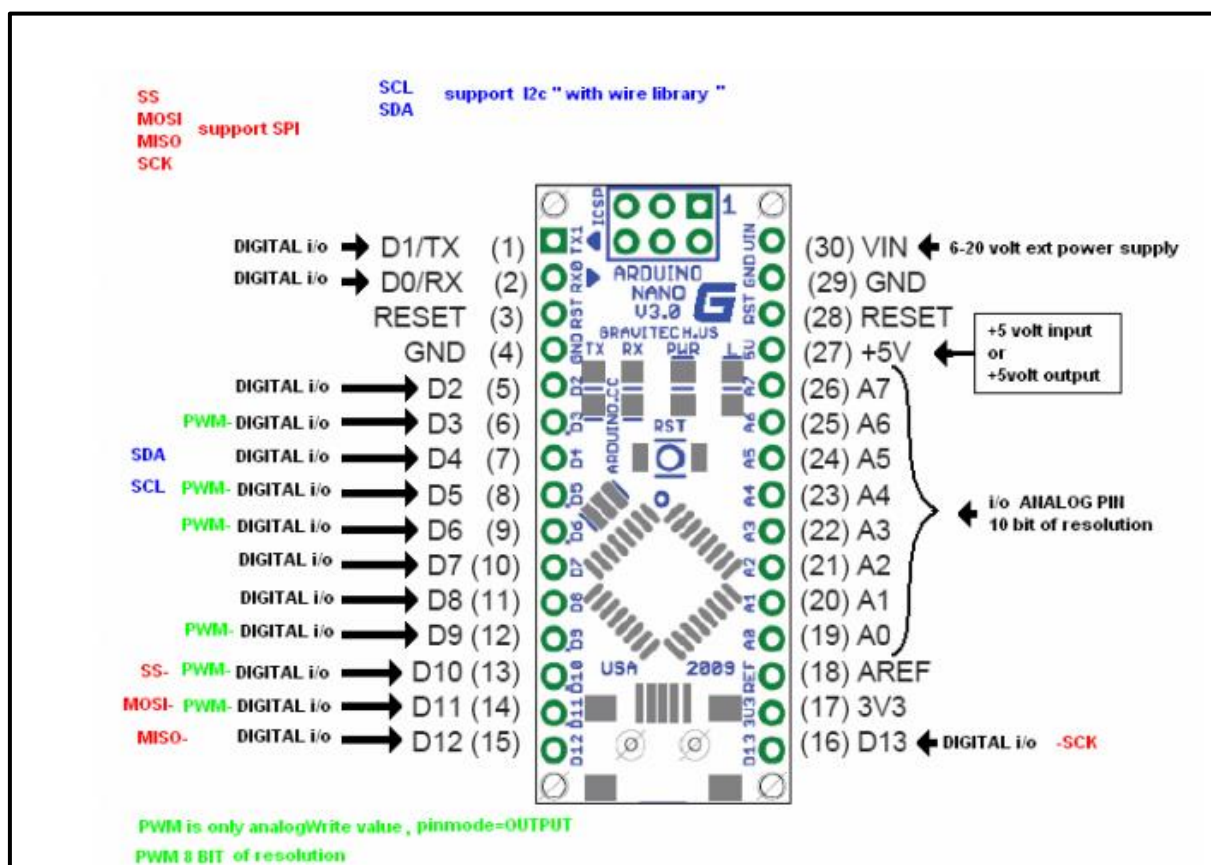


Figura 8: Portas Arduino Nano.

3 DESENVOLVIMENTO

Este capítulo irá descrever toda a parte da construção do projeto. Está dividido em visão geral do projeto, montagem e código de programação.

3.1 VISÃO GERAL DO PROJETO

O projeto desenvolvido tem como função primária seguir uma faixa óptica localizada em uma superfície branca, a função secundária do projeto é identificar a cor da faixa para fazer alteração de sua velocidade, e a última função é verificar a existência de obstáculo no percurso.

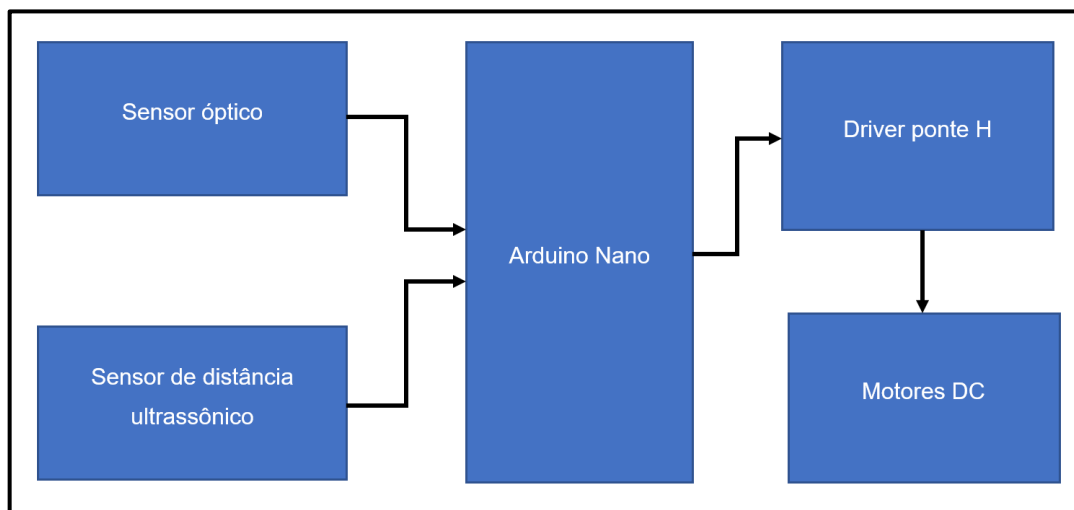


Figura 9: diagrama de blocos do projeto.

No diagrama de blocos da Figura 9 (acima) pode-se observar o funcionamento geral do projeto, é possível observar que a entrada do projeto é constituída pela placa de sensores contendo cinco sensores óticos C9L3 e pelo sensor de distância. Com os sinais que estes mandam para o Arduino Nano o mesmo é capaz de acionar o *driver* ponte H, este acionamento depende do sinal que o Arduino está recebendo dos sensores e também de sua programação interna.

Por fim o *driver* ponte H aciona os motores conforme ordenado pelo Arduino e desta forma o AGV irá mover-se para frente ou parar, ou fazer curvas dependendo do sinal que foi enviado ao *driver* ponte H.

3.2 MONTAGEM

Para montar o chassi do protótipo de AGV foi utilizado um Kit Chassi 2WD Robô para Arduino. Este kit possui uma base de acrílico para o AGV, essa base tem alguns furos para fixação de componentes, como o motor, a roda livre e um suporte de pilhas, além dos furos específicos possuem furos destinados a fixar os componentes de projeto de quem estiver utilizando este kit. Na Figura 10 a seguir é possível observar o kit desmontado.



Figura 10: Chassi do AGV desmontado.



Figura 11: Inicio da montagem chassi AGV.

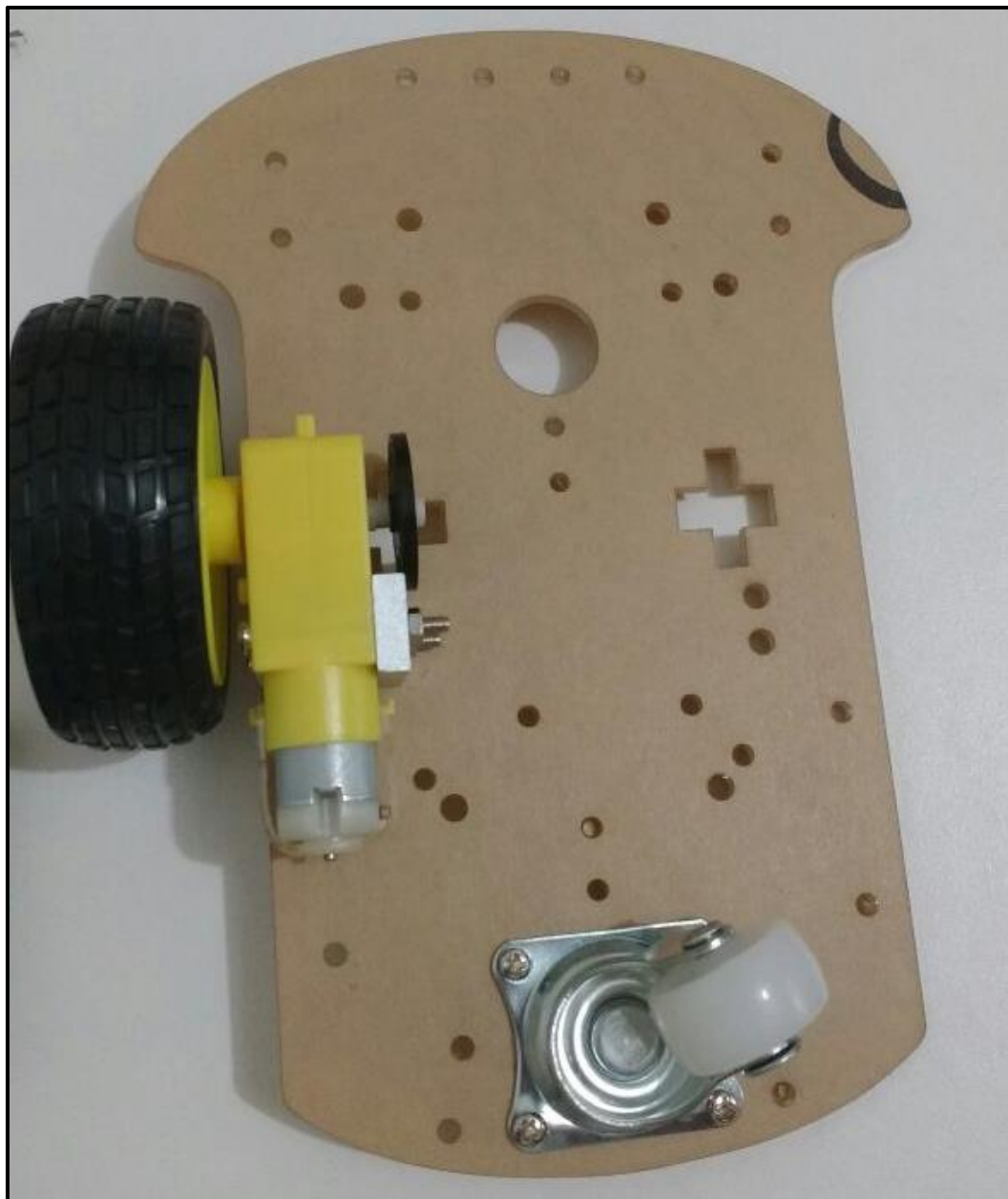


Figura 12: 2º passo da montagem chassi AGV.

Após a montagem do kit o AVG já está com seu chassi pronto, com todas as rodas posicionadas (Figura 13, próxima página).



Figura 13: Chassi AGV montado.

Na figura acima se pode observar o chassi do AVG ainda sem a integração feita com o Arduino e os sensores de anticolisão e de seguidor de linha, nas próximas etapas da montagem será mostrado esta integração.

Para fazer a placa de sensores do projeto foi utilizada uma PCB (*Printed Circuit Board*, ou Placa de Circuito Impresso em português) com um (01) oz (aproximadamente 28,35 gramas) de cobre em sua superfície. Elaborou-se um esquemático do circuito da placa de sensores tendo em vista que esta placa iria estar fixada na parte inferior do chassi do AGV e precisaria enviar um sinal de retorno dos valores lidos por cada sensor para que o Arduino fosse capaz de controlar os motores de forma adequada. O esquemático (Figura 14, próxima página) foi elaborado no *software* Proteus.

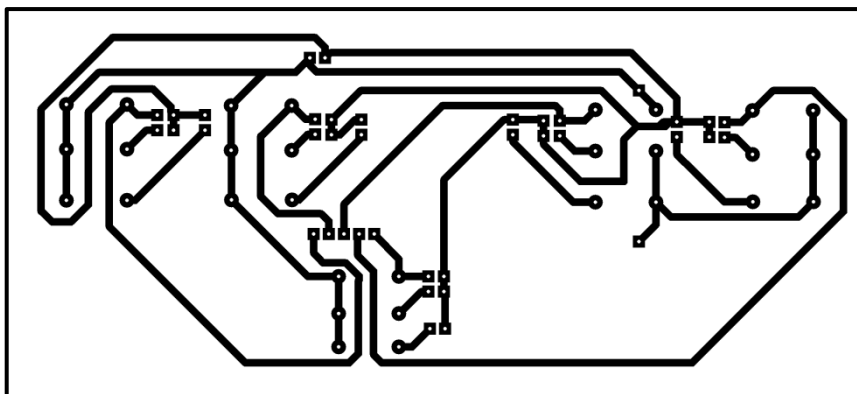


Figura 14: Esquemático da placa de sensores.

Antes de elaborar este novo esquemático foram realizados testes com uma placa anterior para determinar qual o valor de leitura era obtido em cada cor de faixa óptica diferente, a princípio foi decidido que três cores de faixa seriam utilizadas, porém após a confecção desta placa os valores de leitura obtidos nos testes não condiziam com a leitura da nova placa e por isso decidiu-se utilizar apenas duas cores de faixa óptica, sendo elas azuis e pretas.

Já com o esquemático impresso em papel couche 120g foi utilizado o processo de transferência de layout por calor, neste processo é utilizado um ferro de passar roupa em temperatura máxima para fazer a transferência do layout do papel para a placa de cobre. Antes de transferir o esquemático é necessário limpar bem a placa, por isso foi utilizado uma esponja de aço para limpar a superfície da placa e em seguida feita uma limpeza com álcool para remover as impurezas deixadas pela esponja de aço. Feito isso o papel foi cortado e posicionado sobre o cobre da placa e então se utilizou o ferro de passar roupa para esquentar o papel e transferir o layout para a placa em si (Figura 15, próxima página).

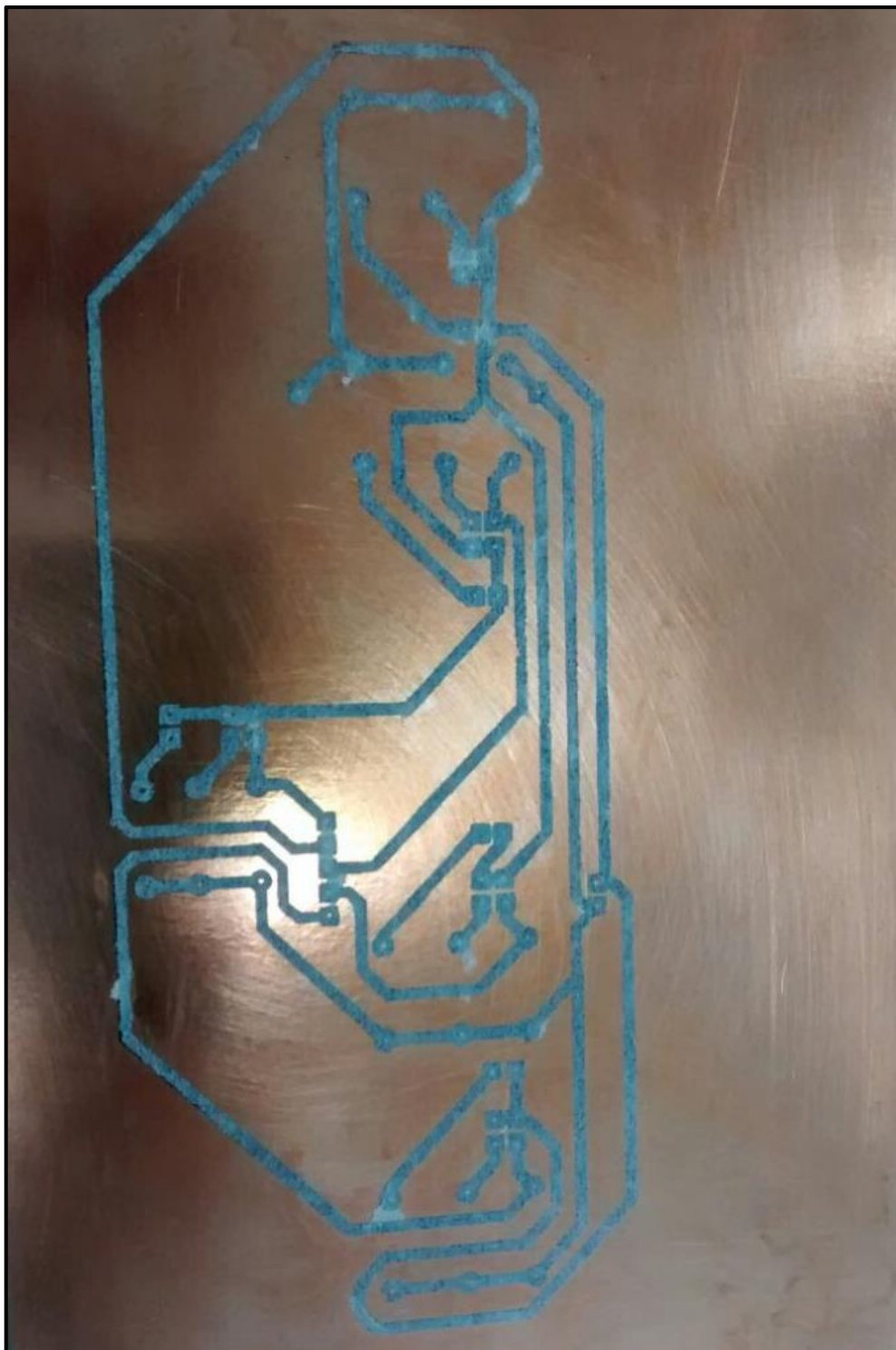


Figura 15: PCB com layout transferido placa de sensores

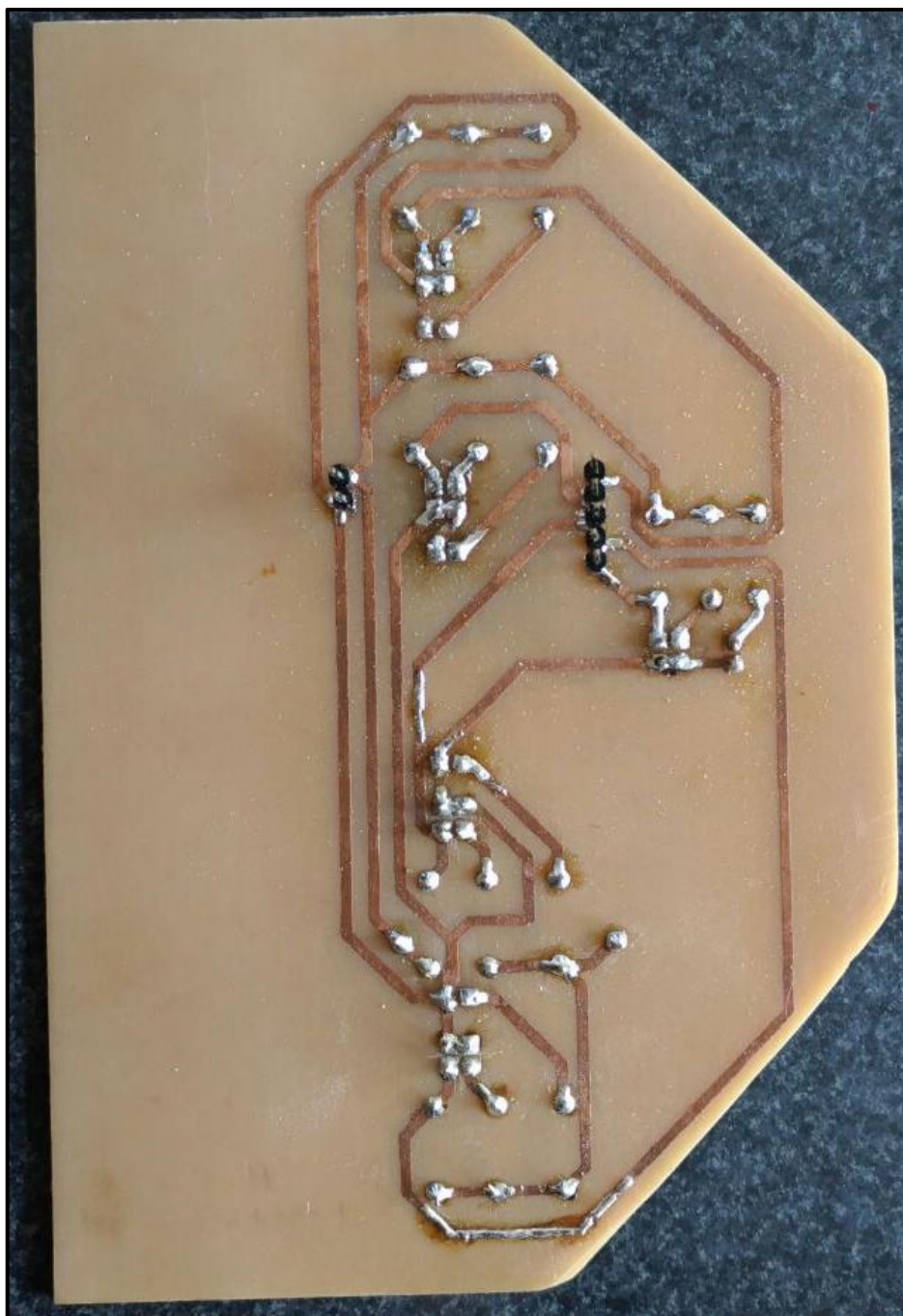


Figura 16: Placa de sensores montada



Figura 17: Placa de sensores fixada ao AGV

Para a integração entre os sensores e o Arduino foram utilizados terminais “machos” na saída da placa de sensores e também na entrada do Arduino, então foi feito um cabo com conectores “fêmeas” em cada uma de suas extremidades para poder fazer a conexão Arduino – Sensores.

Através desta conexão o Arduino é capaz de receber os valores de leitura dos cinco sensores presentes na placa, sendo que quatro deles estão responsáveis pelo direcionamento do carrinho e o quinto por identificar a cor da faixa para determinar a velocidade que o AGV deve andar. Para fazer a conexão entre o Arduino e o *driver* ponte H também foram utilizados os terminais macho do Arduino e no *driver* ponte H, portanto o cabo tem em suas extremidades conectores fêmea. O terminal responsável por habilitar o uso do *driver* ponte H está sendo controlado por um PWM para que o controle de velocidade seja efetuado de forma precisa.

Para assegurar que teria contato elétrico entre o *driver* ponte H e os motores a conexão destes no driver foi feita através de terminais pino com estanho na parte interna para garantir contato entre o terminal e o cabo de alimentação do motor.

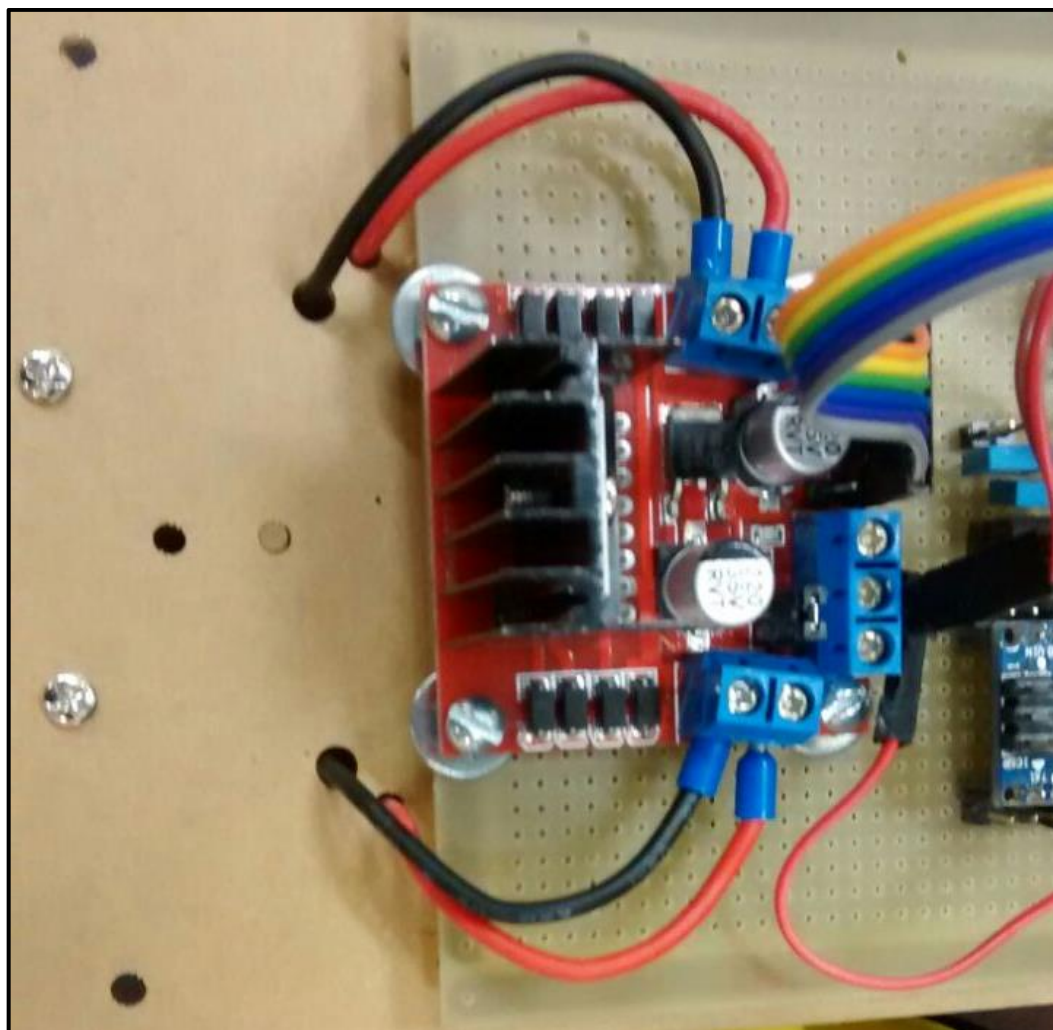


Figura 18: Conexão driver ponte H com motores

Para finalizar a montagem, o Arduino e o driver ponte H foram fixados em uma placa universal 10x10 cm para que eles pudessem ser fixados através de parafusos nos furos presentes no chassi, além disso, foi adicionada uma chave de alimentação do circuito para que tivesse um controle de quando começar a operação e, também, para economizar energia.

O módulo do sensor ultrassônico foi implementado depois de apresentar o projeto sem o sistema de anticollisão à disciplina de Projeto Integrador I. Para alcançar o objetivo atual do projeto, de não colidir com obstáculos no percurso, foi colocado o sensor HC-SR04 em um suporte de isopor e fixado na frente do AVG (como se pode ver na Figura 23). A comunicação entre o Arduino é de apenas um envio de sinal de comando e o sinal de retorno.

O esquema de ligação do projeto pode ser visto no Apêndice B, o qual está mostrando, por meio de tags (Apêndice C), onde cada pino deve ser conectado por

um cabo com o pino dos componentes correspondentes. A seguir vê-se o projeto montado.

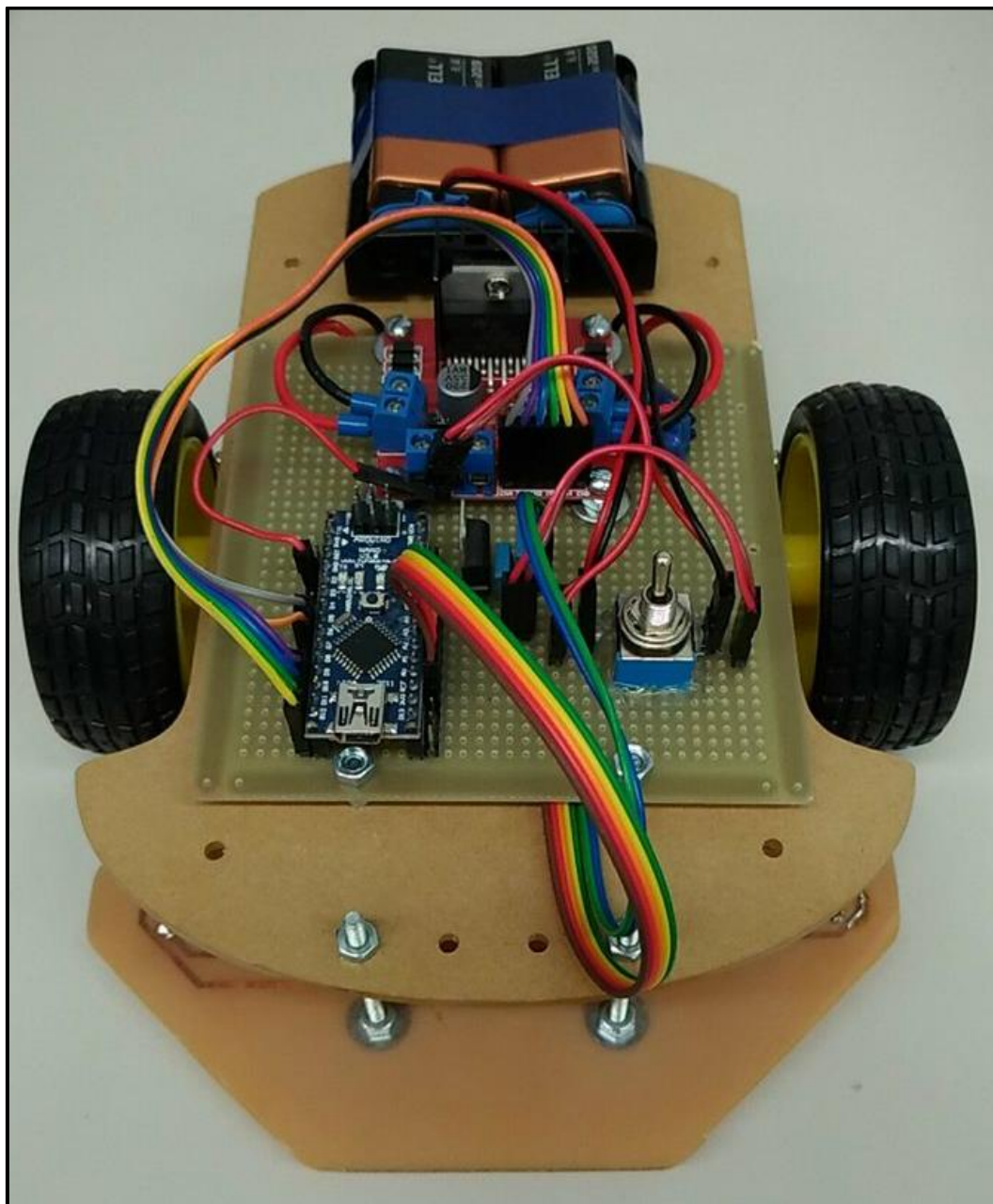


Figura 19: Integração Arduino - placa de sensores

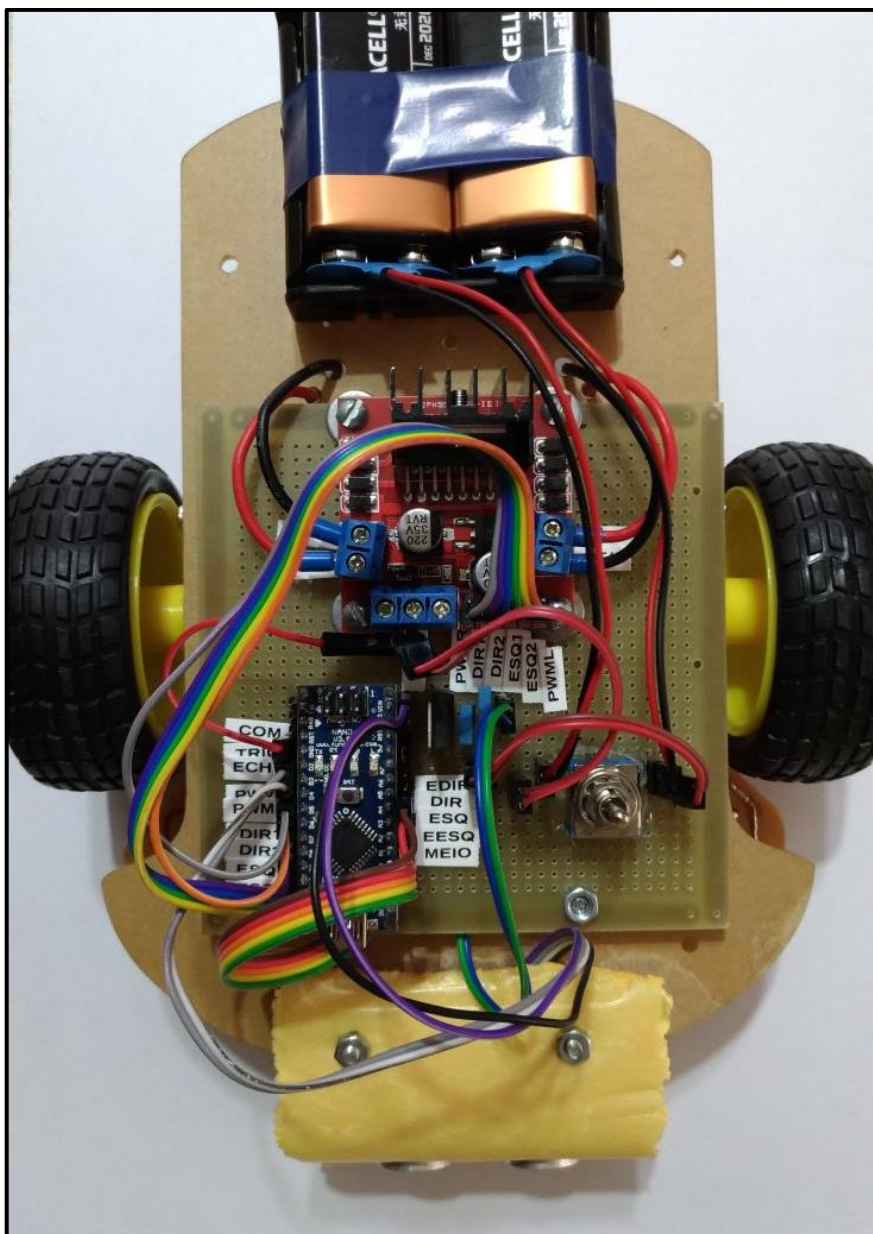


Figura 20: Vista superior AGV completo

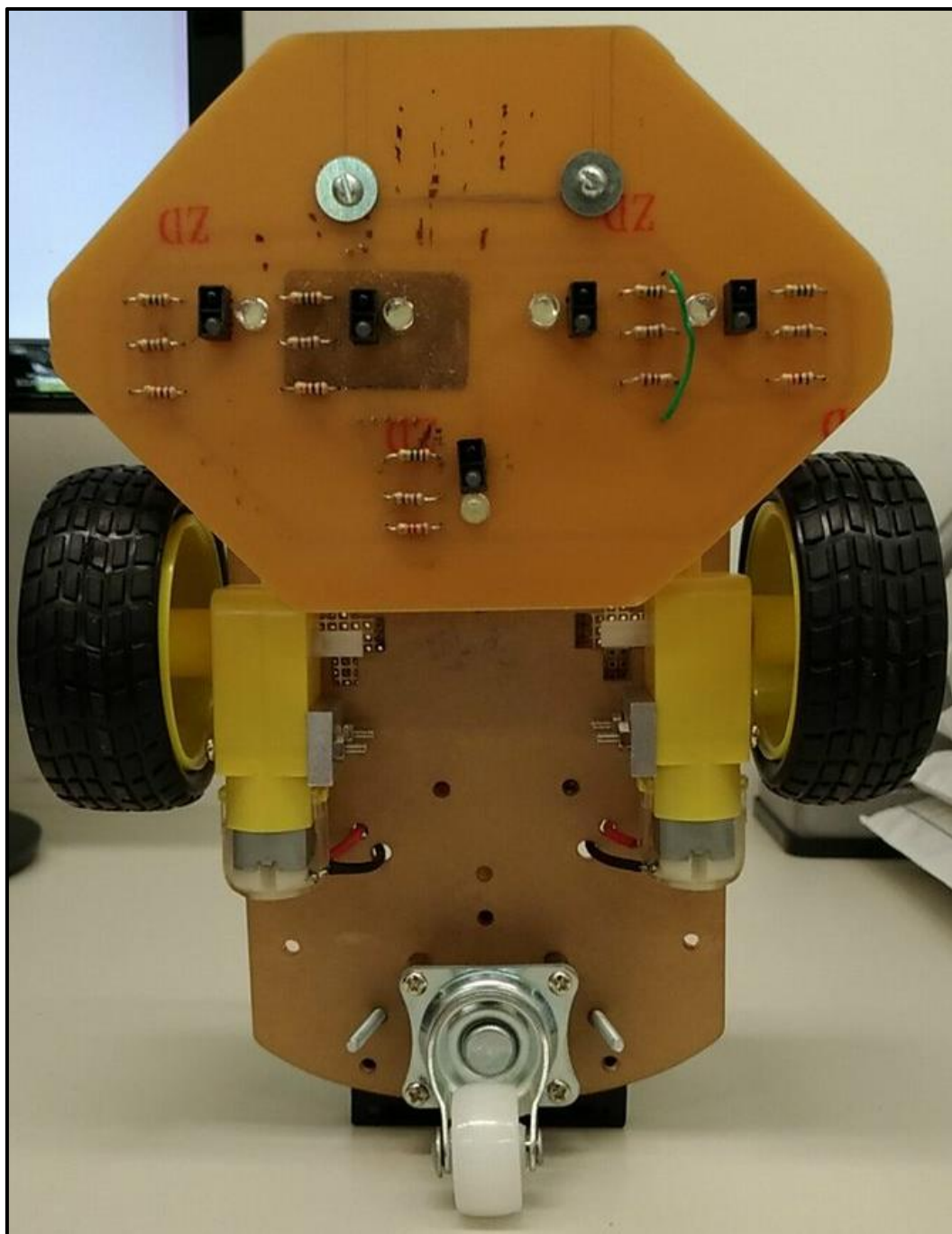


Figura 21: Placa de sensores AGV

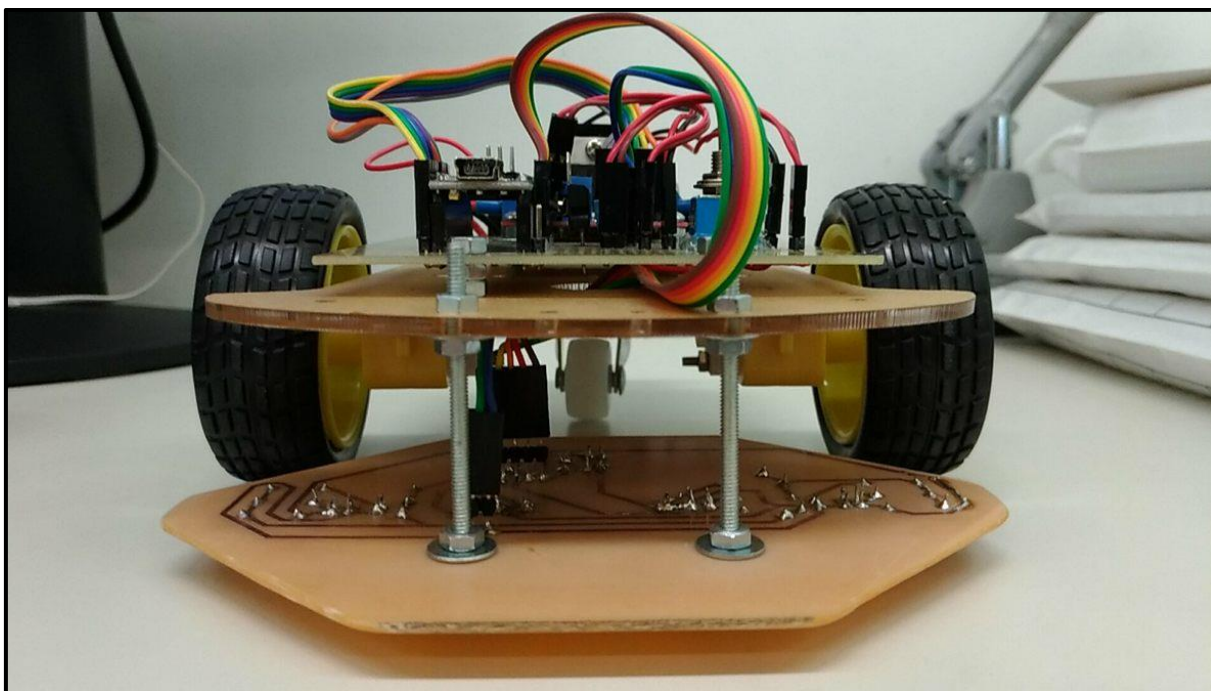


Figura 22: Vista frontal AGV sem o sensor anticolisão.

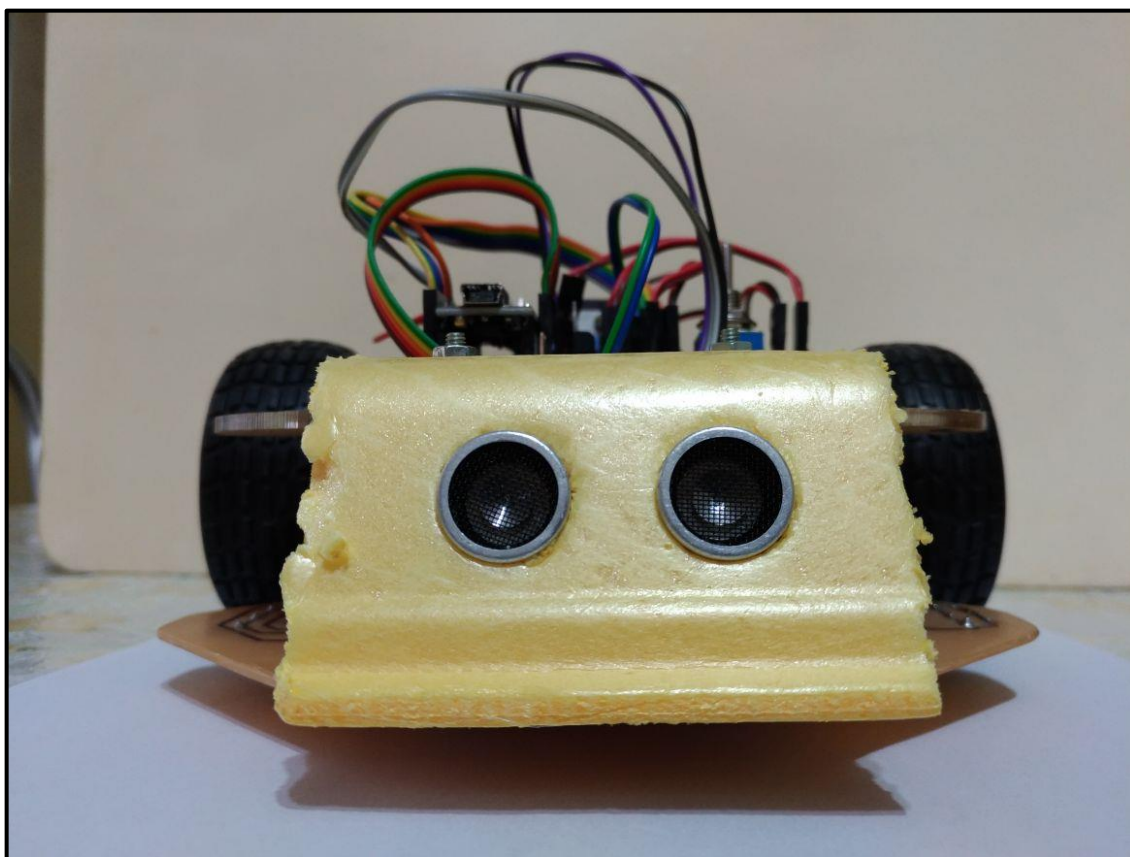


Figura 23: Vista frontal AGV com o sensor anticolisão.

3.3 CÓDIGO DE PROGRAMAÇÃO

O código do projeto foi realizado no *software* Arduino, e tentou deixar o mais simples possível para não dificultar a interpretação dos programadores caso ocorra um problema. Abaixo está o código do projeto com alguns comentários para facilitar na interpretação das ações de cada função.

```
//Código do Seguidor de Linha com Controle de Velocidade e Anticolisão
//variáveis para os motores
const int analogInPin0 = A0; //sensor meio - marrom
const int analogInPin1 = A1; //sensor esquerdo extremidade - vermelho
const int analogInPin2 = A2; //sensor esquerdo centro - laranja
const int analogInPin3 = A3; //sensor direito centro - amarelo
const int analogInPin4 = A4; //sensor direito extremidade - verde
const int velr = 5; //pwm direito - cinza
const int vell = 6; //pwm esquerdo - laranja
int i, para = 0;

//variáveis para o sensor ultrassônico
int gatilho = 2; // pino TRIG do sensor ultrassônico
int echo = 3; // pino ECHO do sensor ultrassônico
float tempo; // para armazenar o tempo de ida e volta do sinal em microsegundos
float distancia_cm; // para armazenar a distância em centímetros

//variáveis para o sensor infravermelho
int sensorValue0 = 0;
int sensorValue1 = 0;
int sensorValue2 = 0;
int sensorValue3 = 0;
int sensorValue4 = 0;
#define linha 380 //cor máxima para a superfície
#define LinhaBaixa 400 //cor mínima para a faixa azul
#define LinhaAlta 700 //cor mínima para a faixa preta
```

```

void setup() {
  Serial.begin(38400);

  // Configuração do Arduino para o comando do Driver Motor L298N
  pinMode(9, OUTPUT); // Define pino D9 como saída - entrada I1 - motor direito -
  roxo
  pinMode(10, OUTPUT); // Define pino D10 como saída - entrada I2 - motor direito -
  azul
  pinMode(12, OUTPUT); // Define pino D12 como saída - entrada I3 - motor
  esquerdo - verde
  pinMode(11, OUTPUT); // Define pino D11 como saída - entrada I4 - motor
  esquerdo - amarelo
  pinMode(vclr, OUTPUT); // Define pino D5 como saída
  pinMode(vell, OUTPUT); // define pino D6 como saída

  // Configuração do Arduino para o sensor ultrassônico HC-SR04
  pinMode(gatilho, OUTPUT); // Configura pino GATILHO como saída
  digitalWrite(gatilho, LOW); // Deixa pino em LOW
  delayMicroseconds(10);
  pinMode(echo, INPUT); // Configura pino ECHO como entrada
}

void loop() {

  sensorValue0 = analogRead(analogInPin0);
  sensorValue1 = analogRead(analogInPin1);
  sensorValue2 = analogRead(analogInPin2);
  sensorValue3 = analogRead(analogInPin3);
  sensorValue4 = analogRead(analogInPin4);

  // Disparo do pulso ultrassônico
  digitalWrite(gatilho, HIGH);
  delayMicroseconds(10);

```

```
digitalWrite(gatilho, LOW);
```

```
tempo = pulseIn(echo, HIGH); // Mede o tempo de ida e volta do pulso ultrassônico
distancia_cm = tempo / 29.4117647 / 2; // Calcula as distâncias em centímetros
```

```
// Verificação e comando do protótipo de AGV
```

```
if (distancia_cm > 10) {
```

```
    if (sensorValue0 < LinhaBaixa) {
```

```
        //condição de arranque
```

```
        analogWrite(vclr, 102); //PWM configurado para 2V (40%) para o motor A
```

```
        analogWrite(vell, 102); //PWM configurado para 2V (40%) para o motor B
```

```
    }
```

```
    if (sensorValue0 >= LinhaBaixa && sensorValue0 < LinhaAlta) {
```

```
        analogWrite(vclr, 102); //PWM configurado para 2V (40%) para o motor A
```

```
        analogWrite(vell, 102); //PWM configurado para 2V (40%) para o motor B
```

```
    }
```

```
    else if (sensorValue0 >= LinhaAlta) {
```

```
        analogWrite(vclr, 120); //PWM configurado para 3V (60%) para o motor A
```

```
        analogWrite(vell, 120); //PWM configurado para 3V (60%) para o motor B
```

```
    }
```

```
    if (sensorValue2 < linha && sensorValue3 < linha) {
```

```
        //anda para frente
```

```
        digitalWrite(9, HIGH);
```

```
        digitalWrite(10, LOW);
```

```
        digitalWrite(11, HIGH);
```

```
        digitalWrite(12, LOW);
```

```
    }
```

```
    else if (sensorValue2 > linha && sensorValue3 < linha) {
```

```
        //vira para esquerda
```

```
        digitalWrite(9, HIGH);
```

```
        digitalWrite(10, LOW);
```

```

    digitalWrite(11, LOW);
    digitalWrite(12, LOW);
}
else if (sensorValue2 < linha && sensorValue3 > linha) {
    //vira para direita
    digitalWrite(9, LOW);
    digitalWrite(10, LOW);
    digitalWrite(11, HIGH);
    digitalWrite(12, LOW);
}

if (sensorValue1 > linha && sensorValue4 < linha) {
    //vira para esquerda (90º)
    analogWrite(vclr, 102); //PWM configurado para 2V (40%) para o motor A
    analogWrite(vell, 102); //PWM configurado para 2V (40%) para o motor B
    digitalWrite(9, HIGH);
    digitalWrite(10, LOW);
    digitalWrite(11, LOW);
    digitalWrite(12, HIGH);
}
else if (sensorValue1 < linha && sensorValue4 > linha) {
    //vira para direita (90º)
    analogWrite(vclr, 102); //PWM configurado para 2V (40%) para o motor A
    analogWrite(vell, 102); //PWM configurado para 2V (40%) para o motor B
    digitalWrite(9, LOW);
    digitalWrite(10, HIGH);
    digitalWrite(11, HIGH);
    digitalWrite(12, LOW);
}
else if (sensorValue1 > linha && sensorValue4 > linha) {
    //para
    analogWrite(vclr, 0); //PWM configurado para 0V (0%) para o motor A
    analogWrite(vell, 0); //PWM configurado para 0V (0%) para o motor B
    if (para == 0){

```

```
    delay(5000); //tempo de espera em uma estação de trabalho
    //frente
    analogWrite(vlcr, 120); //PWM configurado para 3V (60%) para o motor A
    analogWrite(vell, 120); //PWM configurado para 3V (60%) para o motor B
    digitalWrite(9, HIGH);
    digitalWrite(10, LOW);
    digitalWrite(11, HIGH);
    digitalWrite(12, LOW);
    delay(500);
    para = 0;
  }
}
}
else if (distancia_cm < 10) {
  //para
  analogWrite(vlcr, 0); //PWM configurado para 0V (0%) para o motor A
  analogWrite(vell, 0); //PWM configurado para 0V (0%) para o motor B
}
}
```

4 GERÊNCIA DE MANUTENÇÃO

Este capítulo irá descrever as questões de segurança do protótipo com a Norma Regulamentadora 12 e os requisitos de manutenção para o projeto, que são o FMEA, FTA e a Instrução de Trabalho.

4.1 NORMA REGULAMENTADORA 12

As Normas Regulamentadoras (NRs) são conjunto de requisitos e procedimentos que regulamentam e fornecem orientações sobre procedimentos relativos a segurança e medicina do trabalho e do trabalhador.

Inicialmente, existiam 28 normas, mas, atualmente, possuem 36 Normas Regulamentadoras aprovadas pelo Ministério do Trabalho e Emprego. A norma que será tratada nesse capítulo será a NR12, a qual descreve sobre a Segurança no Trabalho em Máquina e Equipamento.

Segundo a NR12, estabelece que as áreas de circulação próximas aos locais de instalação de máquinas e equipamentos devem ser demarcadas (12.6) e permanentemente obstruídas (12.6.2), sendo assim, na maquete da simulação de transporte de carga, deve-se ter uma delimitação de área de circulação para o trabalhador em tamanho reduzido próximos a estações de carga e descarga.

A partir do item 12.14 a 12.23, a norma descreve sobre instalações e dispositivos elétricos. De acordo com essa parte da norma, o protótipo de AGV precisaria ser projetado a não oferecer perigos como choque elétrico, incêndio e outros tipos de acidentes, pois tem grande risco de choque elétrico ao operador/trabalhador, já que possui partes expostas que conduzem eletricidade. A bateria se encontra em local de fácil acesso, facilitando a manutenção e a troca do mesmo (12.22.a), e não há deslocamento acidental pois está bem fixado (12.22.b).

Ainda na parte de instalações e dispositivos elétricos, é proibido a utilização de chave geral como dispositivo de partida e parada (12.21.a) e que funcionem automaticamente ao serem energizadas (12.25), cujos modos estão sendo usados no protótipo. Entretanto, o dispositivo de partida não está localizado em zonas perigosas (12.24) e podem ser acionados ou desligados em caso de emergência por outra pessoa que não seja o operador (12.24.b).

O sistema anticolisão do AGV é um sistema de segurança que visa a paralisação dos movimentos perigosos e demais riscos quando ocorrem falhas ou

situações anormais de trabalho (12.39.f), que no caso seria a obstrução de passagem do protótipo no percurso. O protótipo, após ter sido acionado o comando de parada pelo sistema de segurança, mantém a condição de parada até que existam condições seguras para o rearme, do mesmo modo que está escrito no item 12.40.1 da NR12.

De acordo com a NR12, item 12.42, é considerado dispositivo de segurança os componentes que reduzam os riscos de acidentes e de outros agravos a saúde, e podem ser classificados em seis (06) categorias, sendo que um deles são os sensores de segurança. Esses são dispositivos detectores de presença que atuam quando uma pessoa ou parte do corpo entra na zona de detecção, enviando um sinal para interromper ou impedir o início de funções perigosas. Com essa definição, o sensor HC-SR04 ultrassônico do projeto é um sensor de segurança, que visa a parar o movimento do protótipo ao detectar algo na zona de detecção.

Além disso, seria interessante a implantação de mais sensores de presença em volta do protótipo para garantir maior segurança dos trabalhadores, da máquina e do ambiente de trabalho.

Caso necessite de uma parada de emergência no projeto, isso será feito pelo dispositivo de partida, cujo é a chave que energiza todo o projeto. Segundo a norma, este dispositivo de parada de emergência está totalmente errado, pois, o item 12.56 estabelece que deve possuir pelo menos um dispositivo e não devem ser utilizados ao mesmo tempo como dispositivo de partida (12.56.1).

Do item 12.85 ao item 12.93 da NR12 está relacionado com os transportadores de materiais, o qual se encaixa a utilização do AGV nas indústrias. De acordo com a NR12, os transportadores devem ser utilizados para o tipo e capacidade de carga para os quais foram projetados, sendo que este projeto não priorizou o transporte de carga, mas sim no deslocamento e no sistema de anticolisão frontal.

Deve-se enfatizar que, segundo o item 12.90, que, quando os transportadores (nesse caso os AGV) não forem projetados para carregarem pessoas, é proibida a permanência e a circulação de pessoas sobre partes, ou que possam ficar em movimento, dos transportadores.

A norma prevê o uso de sinais luminosos e sonoros intermitentes, indicando a iminência de um acontecimento perigoso, como partida (12.121). As implementações destes meios de sinalização seriam de grande valia para o

protótipo, já que não possui nenhum, para indicar a sua movimentação pelo trajeto e a parada de emergência.

No item 12.130, diz que devem ser elaborados procedimentos de trabalho e segurança específica com descrição de cada tarefa, passo a passo. Para isto, foi feito a Instrução de Trabalho, FMEA e o Plano de Manutenção.

4.2 REQUISITOS DE MANUTENÇÃO

Visando facilitar a manutenção do projeto, foram desenvolvidas ferramentas para ajudar a identificar possíveis defeitos no projeto desenvolvido, com a intenção de aumentar a manutenibilidade do projeto. Para isso, foram utilizadas ferramentas FMEA, FTA e a Instrução de Trabalho.

4.2.1 Metodologia de Análise do Tipo e Efeito de Falha (FMEA)

É uma ferramenta desenvolvida para analisar as possíveis falhas e suas causas dentro de um equipamento, processo ou atividade. O FMEA tem como objetivo aumentar a confiabilidade de uma máquina ou processo, reduzindo os fatores que implicam em falha do mesmo, há 3 componentes que formam o indicador de confiabilidade, Ocorrência (O), Severidade (S) e Detecção (D).

A ocorrência é responsável por avaliar o quão frequente aquele fator tende a causar uma falha, quanto mais vezes um fator ocorrer, maior o grau de relevância deste, os 3 componentes do FMEA são classificados em uma escala de 1 a 10.

A Severidade determina o quão grave para o processo, máquina ou operador, será caso aquela falha ocorra, quanto pior para o operador, ou equipamento, mais alta será a relevância deste fator.

A detecção determina o quanto um fator pode ser percebido, antes de ocorrer a falha devido este, quanto mais fácil for detectar um fator antes deste gerar falha menor será sua importância pois mais cedo poderá ser realizada uma ação corretiva ou preventiva.

Sendo assim, foi feito o FMEA do projeto, possuindo quatro (04) subitem, como pode ser visto no Apêndice D.

4.2.2 Análise de Árvore de Falha (FTA)

O FTA é um método de análise de falhas quantitativo, diferente do FMEA que além de quantificar determina um nível de importância para cada falha. No FTA é montado um diagrama lógico utilizando simbologias de funções lógicas para determinar que fatores podem levar a determinada falha, a falha percebida pelo operador vira o centro principal da análise, caso aquilo seja percebido pelo operador, quais as possíveis falhas do sistema que ocasionariam aquele efeito.

O projeto possui o FTA no Apêndice E, demonstrando as possíveis falhas que podem ocorrer em determinados componentes.

4.2.3 Instrução de Trabalho

A instrução de trabalho foi desenvolvida, para ajudar a equipe de manutenção a realizar um reparo específico em algum equipamento. Uma instrução de trabalho contém o passo a passo para a realização de uma função de reparo, desta forma, mesmo que o manutentor não conheça as especificidades de determinada máquina, o fabricante indica como realizar o reparo da forma correta, garantindo que após a operação o equipamento voltara a condição normal de operação.

Neste projeto, a instrução de trabalho está apresentada no Apêndice F, descrevendo passo a passo da substituição do sensor ultrassônico, responsável pelo sistema de anticolisão do projeto.

4.2.4 Plano de Manutenção

O plano de manutenção é um documento onde se registram as atividades de manutenção preventiva, bem como, a frequência, periodicidade, materiais e peças que serão utilizadas e quem são os profissionais responsáveis pela execução das atividades.

No Apêndice G, tem o plano de manutenção do projeto com quatro subitem, sendo todos de extrema importância para o projeto.

5 CONCLUSÃO

O projeto foi capaz de executar suas funções que são seguir uma faixa ótica e alterar a velocidade conforme a cor da faixa que o mesmo está seguindo, entretanto, o controle fino ainda não está bem calibrado; e parar o “transporte” caso haja um obstáculo no caminho percorrido. Quando o AGV está em sua maior velocidade o mesmo encontra dificuldade para realizar algumas curvas, pois para tornar perceptível a diferença de velocidade optou-se por deixar a maior velocidade dele em 80% da velocidade máxima do mesmo.

A placa de sensores desenvolvida para este projeto não está muito precisa em sua leitura de cores, portanto seria necessário alterar o resistor acoplado à perna do coletor do sensor C9L3 para que a leitura se assemelhasse com a leitura obtida durante os testes à sensibilidade de cor com a placa que havia sido desenvolvida anteriormente, isto iria possibilitar o uso de três velocidades diferentes ao invés de duas, apesar de que talvez fosse um pouco menos perceptível a mudança de velocidade dado que o mínimo que pode ser atribuído para que o motor tenha torque o suficiente para começar a andar é 30% da sua velocidade máxima.

Sugestões para trabalhos futuros:

- Regular ajuste fino de PWM.
- Refazer placa de sensores tornando-a mais precisa.
- Adicionar plataforma de transporte de carga.
- Fazer estudo de carga máxima possível de ser transportada.
- Adicionar sensores de presença em torno do AGV.
- Adicionar sistema sonora para indicação de movimentação.

REFERÊNCIAS

ANDRADE, Daniel Spillere. **Robô Seguidor de Linha**. 2013. 35 f. Projeto de Eletrônica Aplicada (Graduação em Engenharia Elétrica) – Departamento de Engenharia Elétrica, Universidade Federal de Santa Catarina Centro Tecnológico, Florianópolis, 2013.

ARDUINO. **Start with Arduino Nano**. Arduino.org, 2016. Disponível em: <<http://www.Arduino.org/learning/getting-started/getting-started-with-Arduino-nano>>. Acesso em: 20 de setembro de 2016.

_____. **Arduino Nano**. Arduino.cc, 2016. Disponível em: <<https://www.Arduino.cc/en/Main/ArduinoBoardNano>>. Acesso em: 20 de setembro de 2016.

CARDOSO, Daniel. **Ponte H L298N – Aprenda a controlar a velocidade de um motor DC**. vidadesilicio.com, 2016. Disponível em: <<http://blog.vidadesilicio.com.br/Arduino/ponte-h-l298n-controle-velocidade-motor/>>. Acesso em: 20 de setembro de 2016.

COSTA, Ricardo Fantin da; GIOPOPO, Lucas Longen; HIGASKINO, Marcelo Massao Kataoka; MEIRA, William Hitoshi Tsunoda. **ROBÔ SEGUIDOR DE LINHA**. 2009. 36 f. Monografia (Graduação em Engenharia da Computação) – Departamento Acadêmico de Eletrônica/Informática, Universidade Tecnológica Federal do Paraná, Curitiba, 2009.

FILIPEFLOP. **Kit Chassi 2WD Robô para Arduino**. filipeflop.com, 2016. Disponível em : <<http://www.filipeflop.com/pd-9dd47-kit-chassi-2wd-robo-para-Arduino.html?ct=93347&p=1&s=1>>. Acesso em: 15 de agosto de 2016.

_____. **MOTOR DC COM DRIVER PONTE H L298N**. filipeflop.com, 2016. Disponível em : <<http://blog.filipeflop.com/motores-e-servos/motor-dc-Arduino-ponte-h-l298n.html>>. Acesso em: 15 de agosto de 2016.

_____. **Sensor Óptico Reflexivo TCRT5000**. filipeflop.com, 2015. Disponível em: <<http://www.filipeflop.com/pd-6b912-sensor-optico-reflexivo-x2-unidades.html?ct=&p=1&s=1>>. Acesso em: 20 de outubro de 2015.

GIL, Antonio Carlos. **Como Elaborar Projetos de Pesquisa**. 4. ed. São Paulo: Atlas, 2002.

GONZAGA, D. P.; JULIANI, A. D. P. **Notas de aula em Máquinas Elétricas de Corrente Contínua**. São Carlos: [s.n.], 2008.

LEE, SHANNON apud SAICHATIEN, WARREN. **Where is IoT Headed in 2019?**. Disponível em <<https://www.iotforall.com/where-is-iot-headed-2019/>>. Acesso em: 18 de janeiro de 2019.

MHI. **AGVS**. mhi.org, 2016. Disponível em: <<http://www.mhi.org/agvs>>. Acesso em: 05 de dezembro de 2016.

OLIVEIRA, José Amilton de. **DESENVOLVIMENTO DE UM PROTÓTIPO DE VEÍCULO AUTO GUIADO PARA CÉLULA PRODUTIVA**. 2011. 79f. Projeto de Dissertação (Pós-graduação em Desenvolvimento de Tecnologia) – PRODETEC, Lactec, Curitiba, 2011.

ORGANISATION DE COOPÉRATION ET DE DÉVELOPPEMENT ÉCONOMIQUES (OCDE). **Manuel de Frascati 2002**: Méthode type proposée pour les enquêtes sur la recherche et le développement expérimental. Paris, 2002, 254 p.

PATSKO, Luís Fernando. **Tutorial Montagem da Ponte H**. Maxwell Bohr Instrumentação Eletrônica. [S.l.]. 2006.

PEREZ, Anderson Luiz Fernandes; SILVA, Igo Rafael Alves. DESENVOLVIMENTO DE UM VEÍCULO AUTÔNOMO INTELIGENTE PARA TAREFAS DE NAVEGAÇÃO LIVRE DE COLISÕES. ***Evolvere Scientia***: UNIVASF (Universidade Federal do Vale do São Francisco), Juazeiro, vol. 1, n. 1, p. 6-13, 2013.

PSCHEIDT, Élio Rubens. **ROBÔ AUTÔNOMO – MODELO CHÃO DE FÁBRICA**. 2007. 58 f. Monografia (Graduação em Engenharia da Computação) – Núcleo de Ciências Exatas e Tecnológicas, Centro Universitário Positivo, Curitiba, 2007.

APÊNDICE A – Diagrama de Gantt

APÊNDICE B – Diagrama de Ligação

APÊNDICE C – TAGS

APÊNDICE D – FMEA

APÊNDICE E – FTA

APÊNDICE F – Instrução de Trabalho

APÊNDICE G – Plano de Manutenção